KPMG

Blockchain Consensus

Immutable agreement for the Internet of value

kpmg.com

About the authors

Sigrid Seibold
Principal, Advisory Capital Markets, KPMG LLP

Sigrid looks back at 25 years of working in the banking and capital markets industry. She primarily focuses on the major investment banks, leveraging her areas of specialization, such as data management and digital technologies, including financial and blockchain. As a respected industry thought leader, she has published a variety of white papers covering a variety of capital markets topics, including the use of blockchain for investment banks, as well as articles in major newspapers, such as the *Wall Street Journal*.

George Samman
Blockchain Consultant and Adviser

George is a blockchain consultant/ adviser and entrepreneur in residence at Startupbootcamp for blockchain and bitcoin. He also writes a blog on blockchain technology and use cases at SAMMANTICS. He has co-founded and worked for various startups in the bitcoin and blockchain space since 2013. He is also a contributing writer for various blockchain publications. He was a Senior portfolio manager and market strategist with a Wall Street firm, and a technical analyst. He holds the designation chartered market technician (CMT).

Local China contacts

James McKeogh
Partner, Advisory

James is a Partner within KPMG's Advisory practice and has over 17 years experience with the firm and 3 years with Barclays Wealth in London. With much of his experience coming from the Financial sector, he has also worked in pharmaceuticals, manufacturing, petroleum and gas industries, public sector and retail.

James has been working in Hong Kong for over 6 years specialising in KPMG's services with emerging technologies in Data & Analytics, Digital and Payments. James also leads KPMG's FinTech agenda involved with multiple aspects of the ecosystem from incubators and accelerators, through to investors and the major Corporates.

Raymond Cheong
FinTech and Innovation Partner

Raymond has over 25 years of professional experience in the consulting industry. He has significant business and technical expertise in core banking, enterprise transformation, and business process reengineering and credit risk management. Among others, he worked on large-scale Transformation projects, including Blockchain. ITSP, Branch operations, CRM, Lean Operation Improvement, Digital - Internet banking, KPI and Data and Analytics.

KPMG

Contents

1	Seizing opportunity – Blockchain and beyond
2	The basics behind blockchain
3	Consensus
10	Key observations
14	Is blockchain right for your organization
15	Maneuvering the road ahead
17	Appendix 1: Key terminology
19	Appendix 2: Consensus mechanism valuation questionnaire
24	Appendix 3: Questionnaire response set
25	Acknowledgments

Seizing opportunity - blockchain and beyond

Blockchain, the underlying technology behind the decentralised crypto-currency Bitcoin, may have gone largely unnoticed when it was first pioneered, although this is no longer the case. Some of the world's largest banks and technology firms are investing huge amounts of resources into this technology, which they recognise as potentially the most disruptive technological development to emerge since the Internet.

With potential applications from establishing digital identities through to automating traditionally paper intensive processes like trade financing, blockchain technology is to some the panacea for future financial services whilst many others view it with much more caution. One thing is clear, you cannot ignore it. As part of our engagement with clients, KPMG China is continuing to research and assess different use cases and this report represents a part of that investment. We believe that while financial services organisations are right to be aware of its impact, they should take a measured approach when considering how best to incorporate this technology into their day-to-day operations.

The terms

Blockchain, distributed ledgers, and consensus mechanisms are sometimes used interchangeably. For purposes of this paper, we use the following definitions:

Blockchain: A type of distributed ledger database that maintains a continuously growing list of transaction records ordered into blocks with various protections against tampering and revision.

Distributed ledger: A digital record of ownership that differs from traditional database technology, since there is no central administrator or central data storage; instead, the ledger is replicated among many different nodes in a peer-to-peer network, and each transaction is uniquely signed with a private key.

Consensus mechanism: A method of authenticating and validating a value or transaction on a Blockchain or a distributed ledger without the need to trust or rely on a central authority. Consensus mechanisms are central to the functioning of any blockchain or distributed ledger.

Nodes: Members or systems of a consensus network or a server that holds a replicated copy of the ledger and can have varying roles: to issue, verify, receive, inform, etc. For all intents and purposes, a node can be a virtual machine (VM) instance.

The basics behind blockchain

Blockchains are a specific type of a distributed ledger and a way of ordering and verifying transactions into blocks with various protections against tampering and revision. A network of computers maintains and validates a record of consensus of those transactions via a cryptographic audit trail. A distributed ledger means that no single centralized authority, like a clearinghouse, verifies and executes transactions. Instead, participants have computers that serve as "nodes" within the network.

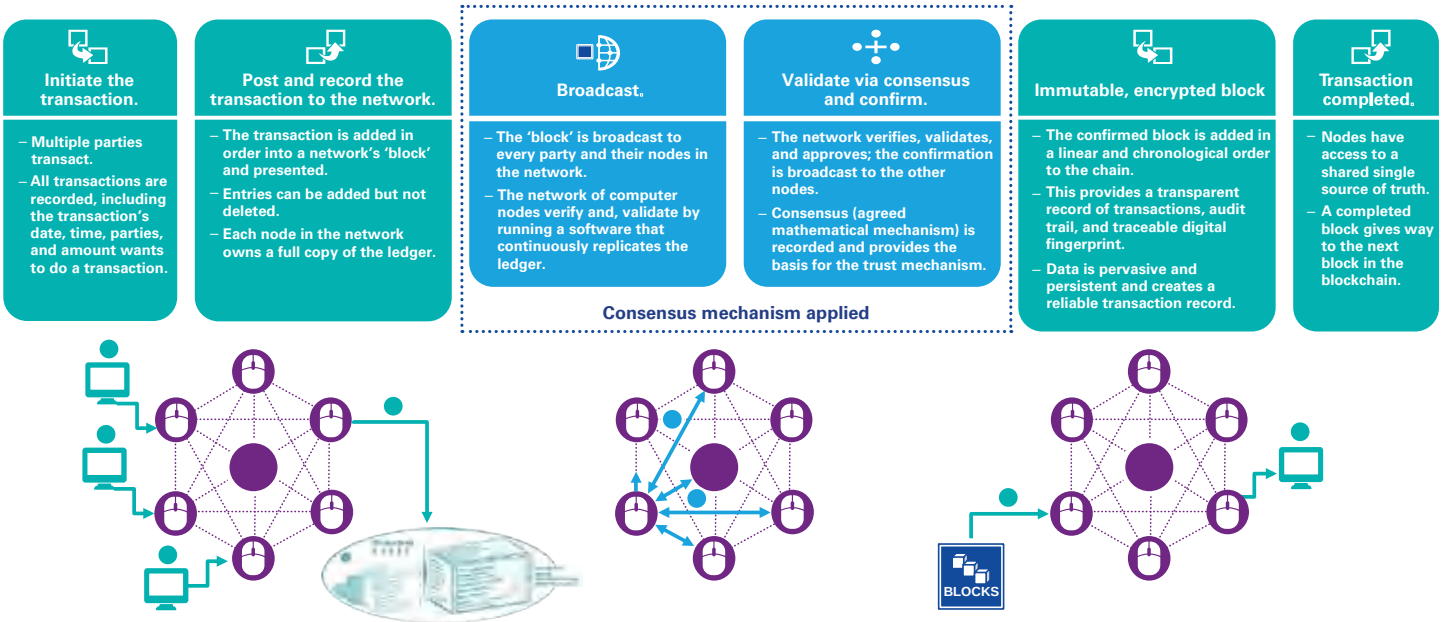
Some or all of these nodes verify and, if appropriate, execute proposed transactions according to an agreed-upon algorithm called the consensus mechanism. The transactions are then encrypted and stored in linked blocks on the nodes, creating an audit trail.

There's no need for a middleman between the parties in a transaction. There's also no need for trust from one peer to the next, since the technology, running on the participants' nodes, provides all the confidence needed. If a blockchain is well-implemented, the resulting advantages include speed, privacy, reliability, and much lower costs.

At the heart of a blockchain is consensus among the participants (refer to steps three and four in **Figure 1.**) Consensus is key, because without a central authority, the participants have to agree on the rules and how to apply them; using these rules, they have to agree to accept and record a proposed transaction.

Figure 1: What exactly are blockchains?

Blockchains are a way of ordering and verifying transactions in a distributed ledger, where a network of computers maintains and validates a record of consensus of those transactions with a cryptographic audit trail.



As highlighted in Figure 1, the transaction, once created and posted to the network, is signed with the signature of the transaction's initiator, which indicates the authorization to spend the money, create the contract, or pass on the data parameters associated with the transactions. If the transaction is signed, it is valid and contains all the information needed to be executed.

The transaction is sent to a node connected to the blockchain network, which knows how to validate the transaction based on predefined criteria. Invalid transactions are discarded, while valid transactions are propagated to another three or four other connected nodes, which will further validate the transactions and send them to their peers until a transaction reaches every node in the network. This flooding approach guarantees a valid transaction will reach the whole network within a few seconds. The senders do not need to trust the nodes they use to broadcast the transactions, as long as they use more than one to ensure that the transaction propagates. The recipients do not need to trust the senders either, because the transactions are signed or contain no confidential information or credentials, such as private keys.

Once a transaction is validated and included in a block, it is then propagated to the network. Once the whole network reaches a consensus and the other nodes of the network accept the new block, it is chained into the blockchain. Once recorded on the blockchain and confirmed by sufficient subsequent blocks, the transaction becomes a permanent part of the public ledger and is accepted as valid in principle by all nodes within the blockchain network.

There are many different mechanisms that can build this consensus, and programmers and companies are constantly working on new ones. Which consensus mechanism a blockchain uses is at the core of what most defines it.

In the pages that follow, we'll look at some of the most important consensus mechanisms out there. As you will see, not all of these consensus mechanisms are blockchains. Some can also work "off-chain," as bilateral agreements, and we'll take a closer look at some of those too. Note: that there's a glossary at the end, which provides definitions for some terms that may not be familiar to a nonspecialist.

Consensus

Consensus. Old and new

Certainly, building consensus is not a new concept. Consensus has been around for as long as human beings have lived together. In its most basic form, it's just a way for a diverse group to make decisions without conflict. According to Edward Shils' "The Concept of Consensus," three things are needed for a consensus:

– The common acceptance of laws, rules, and norms

– The common acceptance of institutions that apply these laws and rules

– A sense of identity or unity, so group members accept that they're equal in respect to the consensus.

Consensus began as a concept for societies, but it's now an important part of computer science too. In the last 30 years, consensus mechanisms in the computer world have gone from an abstract idea to the backbone of distributed ledger technology.

In distributed ledgers, a consensus mechanism is the way in which a majority (or, in some mechanisms, all) of network members agree on the value of a piece of data or a proposed transaction, which then updates the ledger. In other words, a consensus mechanism is a set of rules and procedures that maintains a coherent set of facts among the participating nodes.[1]

Consensus algorithms allow connected machines to work together as a group that can even survive if some of its members fail. This tolerance of failure is another big advantage of blockchains and distributed ledgers, which have a kind of redundancy built in.

Consensus protocols or consensus platforms lie at the core of distributed ledger technologies. There is a great diversity of algorithms for building consensus based on requirements like performance, scalability, consistency, data capacity, governance, security, and failure redundancy.

1 http://www.ofnumbers.com/wp-content/uploads/2015/04/Permissioned-distributed-ledgers.pdf

Consensus

How consensus mechanisms work

Basic parameters that define a consensus mechanism:

- **Decentralized governance:** A single central authority cannot provide transaction finality.
- **Quorum structure:** Nodes exchange messages in predefined ways, which may include stages or tiers.
- **Authentication:** This process provides means to verify the participants' identities.
- **Integrity:** It enforces the validation of the transaction integrity (e.g., mathematically through cryptography).
- **Nonrepudiation:** This provides means to verify that the supposed sender really sent the message.
- **Privacy:** It helps ensure that only the intended recipient can read the message.
- **Fault tolerance:** The network operates efficiently and quickly, even if some nodes or servers fail or are slow.
- **Performance:** It considers throughput, liveness, scalability, and latency

Within these parameters, there are significant differences between one consensus mechanism and another. We'll look at some of these differences when we describe specific mechanisms below. A number of the parameters above are implemented through four main techniques within cryptography that use mathematical formulas to try to ensure security and privacy. These techniques include private keys, public keys, hashing functions, and hierarchical deterministic keys.

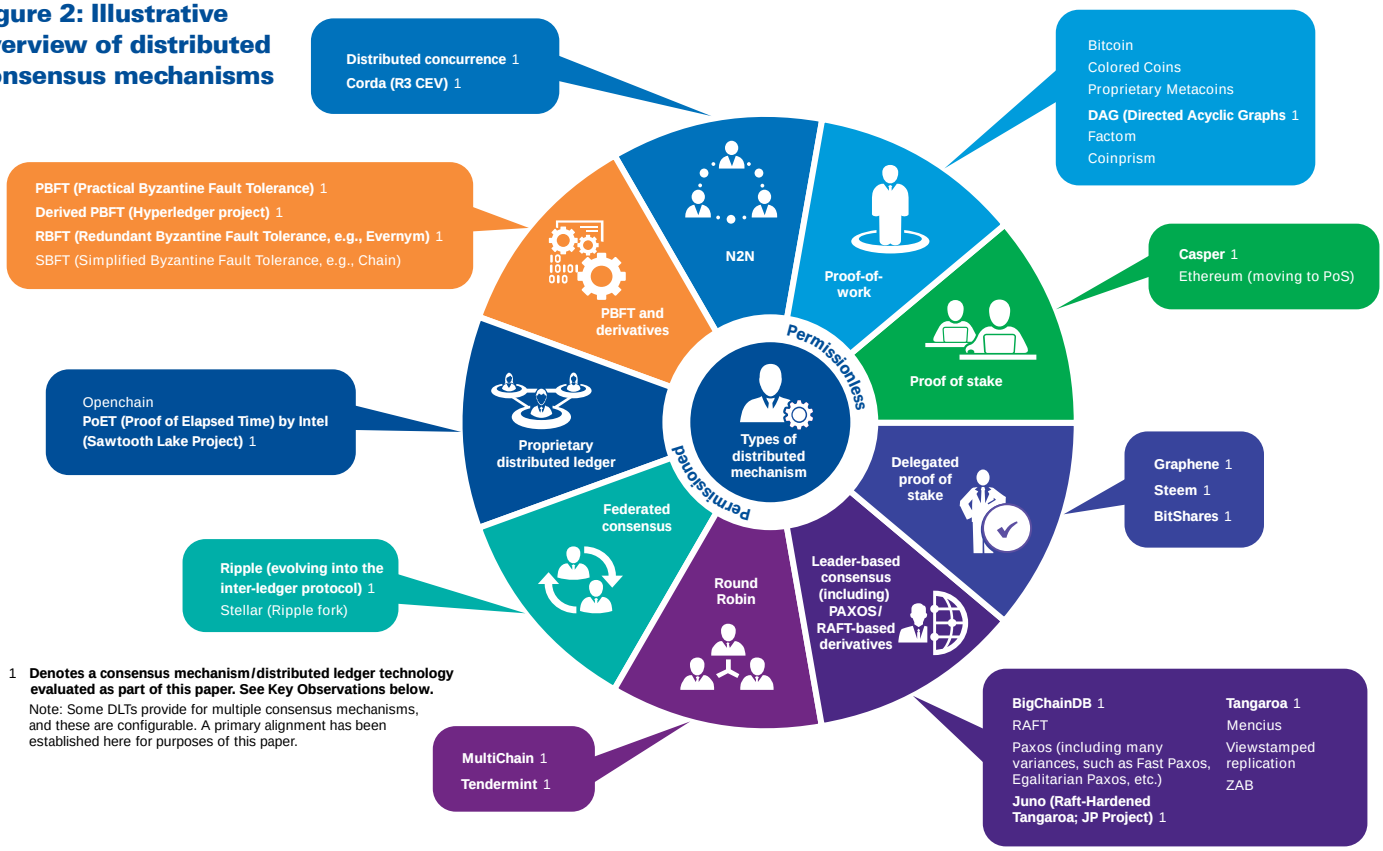
Overview of consensus mechanisms and distributed ledger technologies

Figure 2, provides a visual summary of the key distributed ledger technologies we are seeing in the market right now.

Note: See appendix 1 for definitions of key terms

The scope and description of the various consensus mechanisms can only be a snapshot in time (April/May 2016) as the landscape is evolving quickly. This paper does not aim to be a complete overview of the existing technology consensus options but is geared toward giving a fair representation of those propositions, which currently are being actively explored and discussed as technical options for building blockchains. For the sake of transparency, it also should be stated that many of these consensus mechanisms have been used before blockchain and distributed ledgers came into existence. We have not included any traditional centralized databases for our evaluation.

Figure 2: Illustrative overview of distributed consensus mechanisms



1 **Denotes a consensus mechanism/distributed ledger technology evaluated as part of this paper. See Key Observations below.**

Note: Some DLTs provide for multiple consensus mechanisms, and these are configurable. A primary alignment has been established here for purposes of this paper.

The Byzantine generals' problem

The basis for modern consensus mechanisms came in 1962, when an engineer at the RAND Corporation, Paul Baran, came up with the idea for cryptographic signatures in a paper called "On Distributed Communications Networks." These digital signatures soon became a way to authenticate users when they amended data or files.

Twenty years later, a trio of scholars published a paper that outlined the problem of reliability in a decentralized system. In "The Byzantine Generals' Problem[2]," authors Leslie Lamport, Robert Shostak, and Marshall Pease proposed a thought experiment: Imagine that a group of generals, each commanding part of the Byzantine army, surrounds an enemy city. The generals can only communicate by messenger, but in order to conquer the city, they have to agree on a battle plan.

The problem is that one or more of the generals might be a traitor who will try to distort the messages and sabotage the plan. The question is, how many traitorous generals can the army have and still function as a unified force?

There's a direct analogy to digital currencies, the custody of assets, and the transference of values when there isn't a central authority to verify these assets and transactions. In distributed ledgers, the different participants' nodes are like generals. They have to decide on an acceptable fault level: How many transactions can be malicious (how many generals can be traitors) without the system having to refuse a transaction? This is because a certain number of failures may not damage the overall system's reliability.

In the scenario these authors described, with oral messengers connecting each pair of generals, it's possible to develop an algorithm so that the system (the Byzantine army) will be reliable if it's certain that two-thirds or more of the generals are loyal.

For distributed financial transactions on computers, the question is more complex. For a while, it seemed unsolvable.

The Byzantine generals' solution – And bitcoins

The solution came in 1999, when Miguel Castro and Barbara Liskov introduced the practical byzantine fault tolerance (PBFT) algorithm. PBFT can process an enormous number of direct peer-to-peer (or distributed) messages with minimal latency. That means that programmers can build secure and resilient private distributed networks. Since 1999, PBFT has been implemented in many ways, and it's been further developed into various technical iterations.

The first way, developed in 1999, was "proof-of-work." Proof-of-work means that the system's users have to repeatedly run algorithms to validate the transactions of the system's other participants. Today, it's still the most publicly proven method to achieve consensus.

Proof-of-work systems maintain their blockchains with a decentralized peer-to-peer cryptographic protocol. They don't have any central authority, but they do assume that "honest" nodes control at least a majority of the system's computer power. (At least half the army is in the hands of loyal generals.) They're public or permission-less systems: The nodes don't need to know who the other nodes are.

Bitcoin is the best-known use of a proof-of-work system. A person or team who worked under the name Satoshi Nakamoto published the Bitcoin technology in October 2008 in a paper called "Bitcoin: A Peer-to-Peer Electronic Cash System."[3] It was quickly implemented as open-source code and released in January 2009 as the now-famous digital currency. It's based on "mining": Participants' computers verify and add transactions to the public ledger and, as a reward, earn new bitcoins.

Many other methods have since followed. **Figure 3,** (on the following page) gives a visual representation of the technology's development before and after bitcoins. In the following pages, we look at some of the forks in this road.

Another way of mining bitcoins

Proof-of-stake came in 2012. The method here is to create a mechanism that punishes nodes that don't follow the consensus protocol. Participants have to bet a predefined amount of digital assets (bitcoins) on a consensus outcome. If the outcome doesn't take place, the malicious nodes lose these assets.

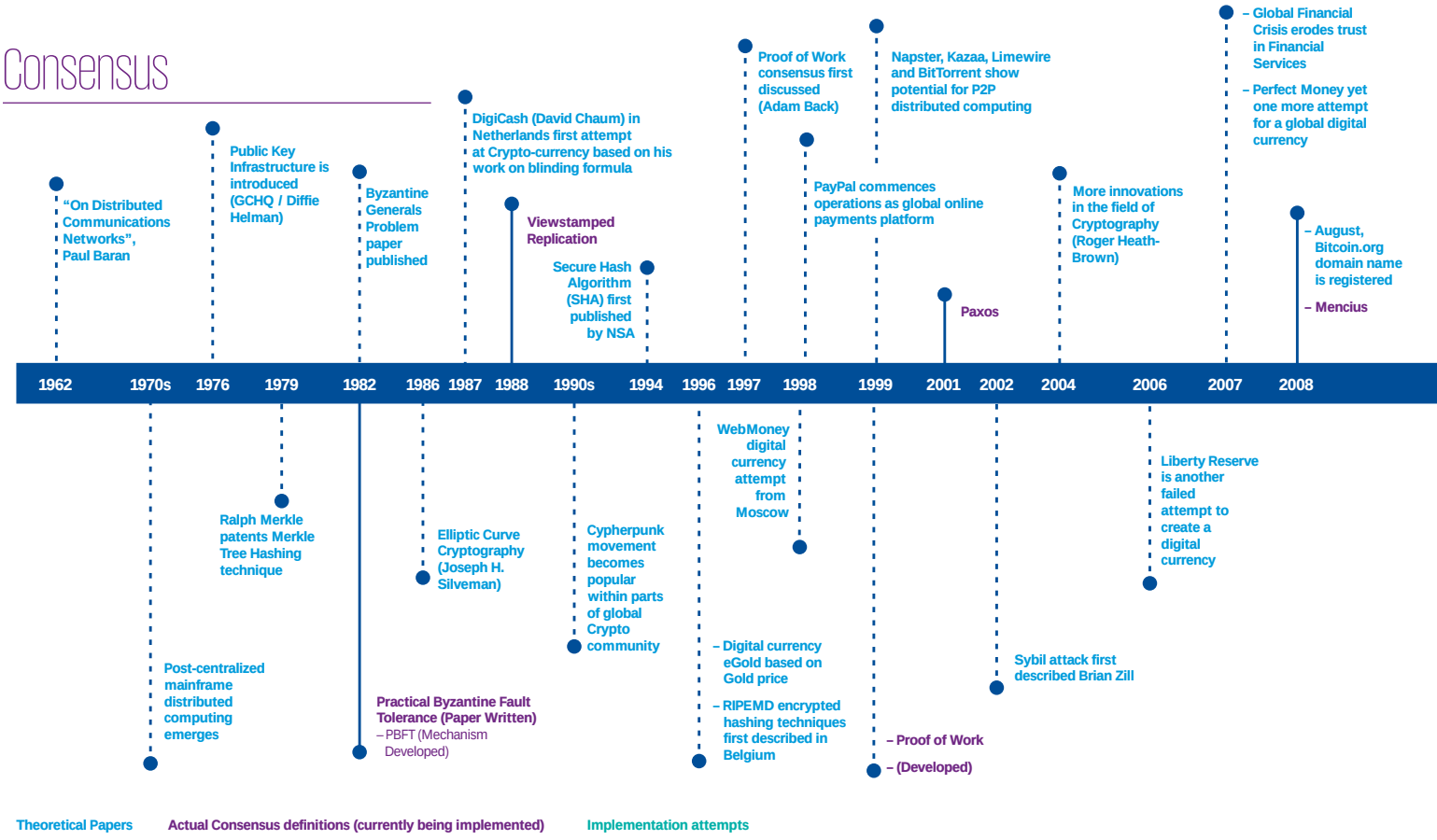
In proof-of-stake bitcoin systems, where mining requires the participant to "put up a stake," a participant can mine new coins (or enter new transactions) in accordance with how many coins they already have. In proof-of-work systems, mining successfully depends on actually doing the computational work.

The advantage of proof-of-stake over proof-of-work is that it requires fewer laborious computations. Since these computations are usually expensive, their reduction lowers the cost of the system and the barriers to entry.[4] The more coins are held in the digital wallet and the higher the degree of controlled computing power, the greater is the probability of winning a block.

2 LAMPORT, L., Shostak, R., and Pease, M. The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems, 4, 3 (July 1982), 382–401

3 https://bitcoin.org/bitcoin.pdf

4 http://bitfury.com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf



A variant system called delegated proof-of-stake (DPOS) tries to combine proof-of-stake and proof-of-work characteristics. DPOS uses a decentralized voting process through what are known as witnesses as a way to mitigate against potential network centralization.

The next generation after bitcoin

Developers have since presented new mechanisms meant to improve bitcoin. In 2014, the French entrepreneur Flavien Charlon launched Coinprism, which uses "colored coins," an open-source protocol to create digital assets on top of the bitcoin blockchain. That lets the bitcoin blockchain be used for more than just currency. Several big financial market players, including Citigroup[5] and NASDAQ,[6] began experimenting with colored coins in 2015.

Metacoins have also emerged–coins sitting on top of another blockchain as a "meta" layer.[7]

But for all the apparent potential, it soon became clear that it wasn't feasible for financial institutions, which are heavily regulated, to adopt either of these technologies for these reasons[8]:

– The security system inherited from bitcoin and other proof-of-work-based blockchains doesn't work well for regulated financial settlements (its incentives are distorted)

– There's not enough legal finality around settlements.

– Regulatory risks remain high.

Alternatives to blockchains

The search for consensus mechanisms that are reliable for financial institutions and acceptable to regulators led developers to systems that don't depend on bitcoin and proof-of-work. Ripple, developed in 2012, was the first significant new one. Ripple's code base is based on the bitcoin blockchain but does not use proof-of-work consensus. Instead, the ripple network uses a "ripple consensus ledger" which has these features:

– Its participants and history define it, not the underlying technology.

– It relays messages with open peer-to-peer broadcasts.

– Instead of relying on mining, it uses a system of tokens called XRP as currency.

– Consensus subnetworks of collective trust called "unique node lists" (UNL) exist within the larger network, so the system is a kind of federation

5 http://www.ibtimes.co.uk/codename-citicoin-banking-giant-built-three-internal-blockchains-test-bitcoin-technology-1508759

6 https://www.theguardian.com/technology/2015/may/13/nasdaq-bitcoin-blockchain

7 http://explainbitcoin.com/what-is-a-meta-coin/

8 https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/564ca429e4b0a9e90a947ba2/1447863337472/watermarked-tokens-and-pseudonymity-on-public-blockchains-swanson.pdf

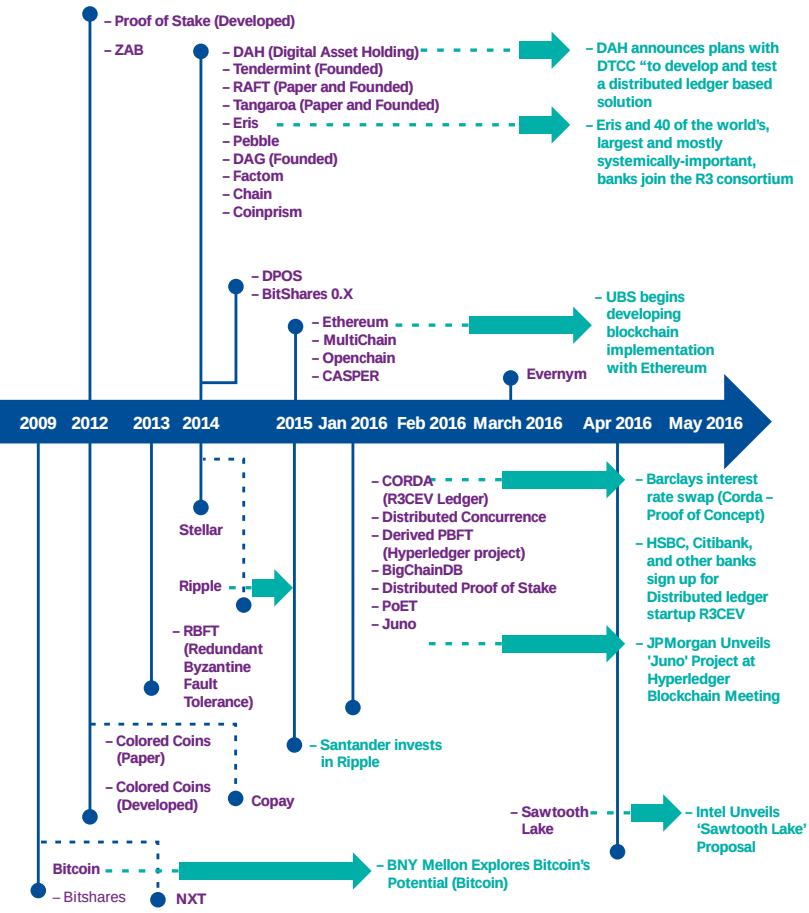


Figure 3: Illustrative body of work before bitcoin and the history of consensus mechanisms

– Each participating server maintains its own UNL based on how its operator configured it

– A server can be configured either as a node that participates in validating proposed transactions or as a follower node, in which case participants just use it to submit payments or make inquiries.

Ripple's consensus mechanism requires that an 80 percent supermajority of nodes in the UNL subnetwork (not in the larger system) agree to validate a transaction.[9] That means transactions can take place in seconds, rather than the 10 minutes or more required in proof-of-work systems.

This was a major breakthrough on the path to distributed ledgers. Instead of mining, a native token is used primarily to throttle 'spam-like' transactions. The progression of ripple has led to the current evolvement of the Inter Ledger Protocol, which in essence provides a way to connect legacy ledgers of the past, with the distributed ledgers of the future.

Stellar followed in 2014. The stellar consensus protocol (SCP) is based on what Stellar calls a federated byzantine agreement, which uses "quorum slices": Each node chooses which other nodes to trust. The sum of all these individual choices is a system-level quorum of consensus. These slices bind the system together in much the same way that individual networks' decisions unify the Internet.

9 https://ripple.com/files/ripple_consensus_whitepaper.pdf

More consensus mechanisms emerge

Since PBFT was developed in 1999, many other consensus mechanisms have emerged. Some consensus mechanisms utilize tokens; others have evolved to function token-less and without mining respectively. There's been a general shift from permission-less systems like bitcoin, where anonymous nodes validate transactions, to permissioned systems. Here, nodes must be legally known and identified to validate transactions. Since the nodes are known and can demonstrate their assets, there's no need for "mining" to create currency.

Where bitcoin is an open, censorship-resistant system, we have seen an evolution for certain capital markets transactions to the exact opposite of that: bilateral consensus mechanisms, node to node (N2N). The two counterparties in a transaction validate it between themselves, node-to-node, without others in the ledger involved, unless the counterparties choose to permit it. R3CEV's new ledger, Corda, has developed such a solution for its banking consortium. Corda essentially creates, with a set of rules that all participants have agreed to, an environment in which everyone has access to the same data. Each hash is then recorded so that it can't be disputed.

RAFT, which was developed and published in 2014 to improve an earlier system called Paxos, works through a transitory centralization: The nodes temporarily elect one node to be a leader. The leader is then responsible for validating transactions. A variation on RAFT called Tangaroa came the year after, meant to better protect against malicious attacks and software errors.

In 2014, Digital Asset Holdings was founded as a blockchain technology company that provides settlement and ledger services for financial assets. In March 2015, the company appointed Blythe Masters as chief executive officer and made headlines in early 2016, raising more than $60 million in their series A funding round.

Ethereum came in July 2015 as another attempt to extend blockchain use beyond bitcoin's peer-to-peer money system. The smart contract concept of distributed data computation on distributed ledgers was introduced—an entity can represent value from tangible or financial goods in a contract and then use blockchains to distribute it. Ethereum is a step toward combining traditional financial contracts with the blockchain technology.

Eris Industries has offered a way to integrate Ethereum's tool kit with the technology stack (the various layers of software that form a computer's infrastructure.) An Ethereum VM sits at the top of the stack. Further down are different consensus mechanisms that can be swapped in and out. With this system, each organization, depending on its needs at a given time, can use one consensus mechanism or another. There is no need to use one monolithic consensus layer.

Also in 2015, Coin Sciences launched MultiChain as the first freely available off-the-shelf blockchain platform. At present, MultiChain allows the issuing and tracking of assets on a network level and includes a permissions management system to enable privacy, mining control and specific counterparties.

Consensus

In February 2016, the Linux Foundation's Hyperledger project introduced templates based PBFT meant to serve as foundations for blockchains. The idea is to create a cross-industry open standard and an open-source development library so that business users can build custom distributed ledger solutions. Hyperledger's templates can customize a given transaction and then record it through a private blockchain or other registries.

Big companies increasing their involvement

In March of this year, JP Morgan announced its own consensus mechanism, which it had begun working on in 2015. Its Juno12 project, like RAFT and Tangaroa (which inspired it), achieves consensus by electing a temporary leader. The client-side node then gives this leader node a command, which it distributes to the system.[11,12]

Also this year, Intel® released details on its Sawtooth Lake project, based on its PoET platform for distributed ledgers. Intel describes it as follows:

"Sawtooth Lake abstracts the core concepts of consensus, isolates consensus from transaction semantics, and provides two consensus protocols with different performance trade-offs. The first, called Proof of Elapsed Time (PoET), is a lottery protocol that builds on trusted execution environments provided by Intel's SGX to address the needs of large populations of participants. The second, Quorum Voting, is an adaptation of the ripple protocol and SCP and serves to address the needs of applications that require immediate transaction finality."[13]

Chinese companies are also moving forward. The ChinaLedger Alliance was announced at the start of May: 11 commodity, equity, and financial asset exchanges led by Wanxiang Blockchain Labs are working to create an open-source blockchain protocol and to set standards across the industry to ensure regulatory compliance.[14]

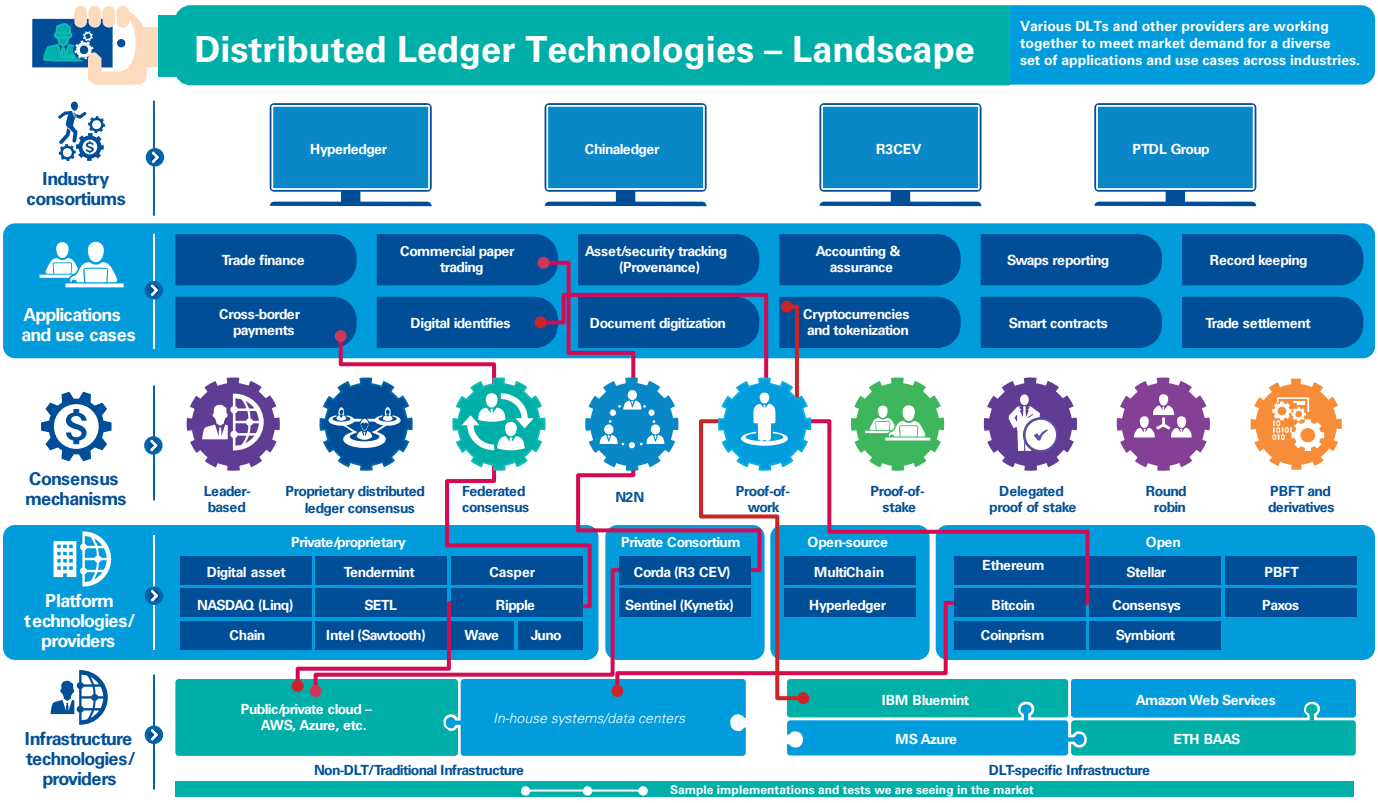
11 http://www.the-blockchain.com/docs/JP-Morgan-Juno-Distributed-Cryptoledger.pdf

12 http://www.coindesk.com/jpmorgan-juno-hyperledger-blockchain/

13 http://intelledger.github.io/introduction.html

14 https://bitcoinmagazine.com/articles/china-joins-the-blockchain-race-with-chinaledger-alliance-1462204569?q=&hPP=5&idx=articles&p=0&is_v=1

Figure 4: Illustrative distributed ledger technologies



Also in May, the Silicon Valley start-up chain announced[15] Chain Open Standard 1, which it built with the help of nine major banks and payments firms, including Capital One and Citigroup.[16] Open Standard 1 is an open-source technology, available to any financial companies that want to run high-scale financial applications on permissioned blockchain networks.

Chain claims that Open Standard 1 can finalize high volumes of transactions in less than a second. It also says the technology can encrypt data and then provide selective access to counterparties and regulators. It provides a smart contract framework that supports simple rule enforcement and key-value storage.

All these developments from different companies may soon add up to a digital ledger ecosystem. Some providers, including start-ups, will offer platforms developed for specific uses. Other providers will offer play boxes. There will be many open-standard collaboration groups. And more and more consensus mechanisms to address different and complex needs will emerge.

Figure 4 illustrates how several different combinations of blockchains, distributed ledger technologies, and other providers are working together to meet market demand for various use cases.

Figure 5, gives an idea of just how far the area has come already and how quickly it's still changing.

Consensus mechanisms for specific needs

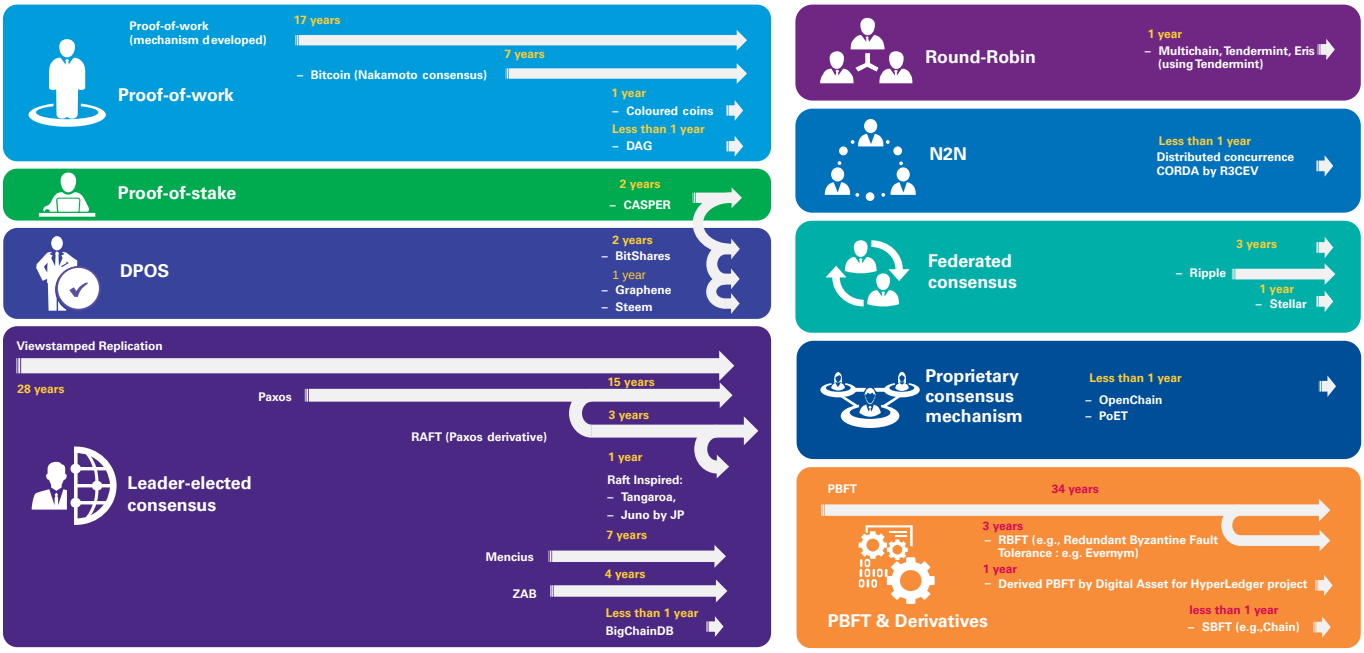
We believe that consensus mechanisms will evolve to target specific needs, whether those of a particular use case, of technical implementation possibilities, or of the regulatory environment. One example of the latter is MultiChain, whose permissions management system has seven types of permission that allow different participants to connect, send, receive, issue, mine, activate, or administer.

A node that only has permission to connect has read-only access. Another node may be able to read and write but not to validate, if it doesn't have permission to mine. There isn't much value in a node having permission to write but not to read, since it can't build transactions if it doesn't know where it's receiving assets from.

15 http://www.americanbanker.com/news/bank-technology/with-banks-help-startup-chain-rolls-out-open-source-blockchain-1080785-1.html?utm_content=socialflow&utm_campaign=amerbanker-tw&utm_source=twitter&utm_medium=social

16 Capital One Financial, Citigroup, Fidelity Investments, First Data, Fiserv, Mitsubishi UFJ Financial Group, Nasdaq, State Street and Visa

Figure 5: Illustrative historical comparison of consensus mechanisms



Key observations

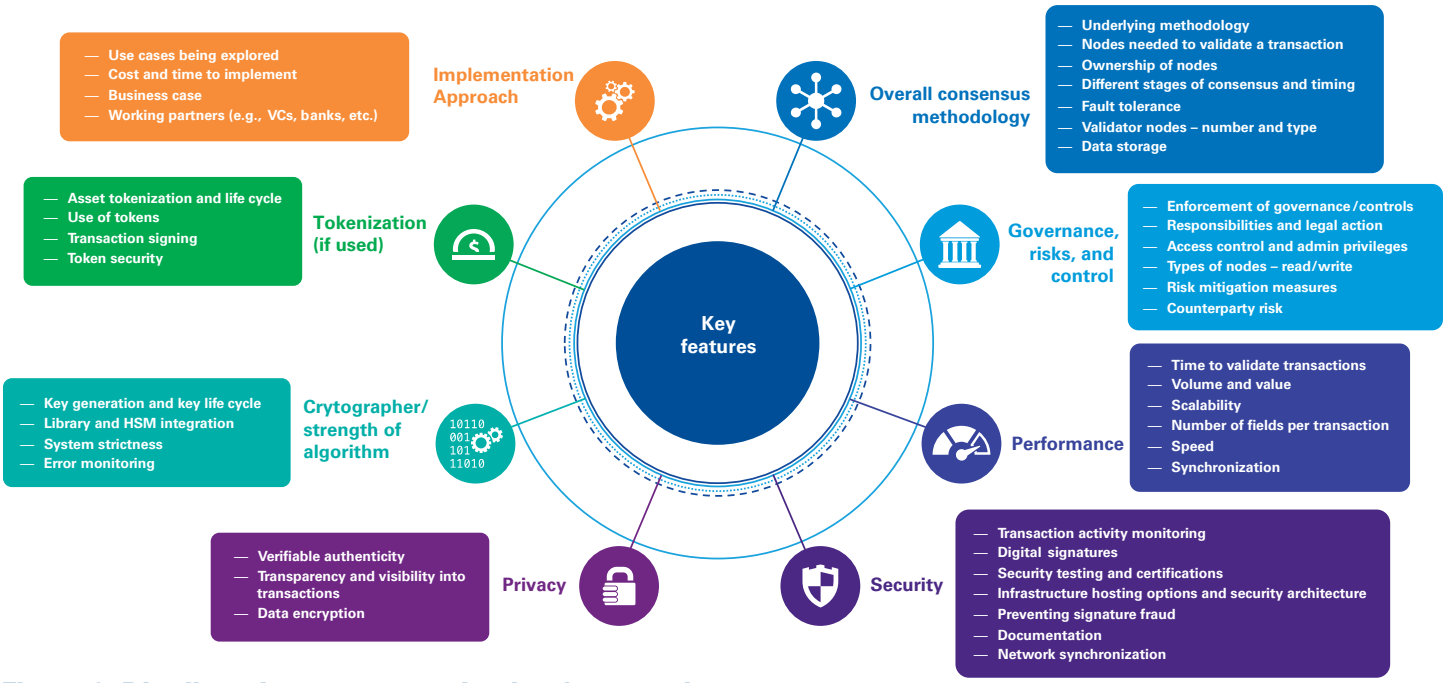


Figure 6: Distributed consensus evaluation framework

During our research for this paper, we surveyed more than 20 creators and corporate users of blockchains and other consensus mechanisms.

Figure 6, provides an overview of the framework and key topics we covered to evaluate some of the most important consensus mechanisms and distributed ledger technologies we are seeing in the market currently.

Note: See Appendix 2 for a detailed questionnaire that utilizes this evaluation framework.

The following represent our key observations after assessing their responses.

Overall consensus methodology

– Permissioned DLTs are proving popular with financial services institutions as participants are determined ahead of time. Figure 2 above provides a good overview of the different types of consensus mechanisms that are being implemented or tested for various use cases.

– Consensus mechanisms require parties to validate the transaction via an N2N communication. The number of nodes required to validate a transaction varies based on the distributed ledger technology. These range from one node (e.g., OpenChain) to a simple majority (e.g., Juno) to a super majority (e.g., Ripple) to requiring all nodes (e.g., Casper) or can be configurable; for example, Stellar can be configured to require 51% for a trusted node network or 67 percent for untrusted.

– While all providers appear to be resistant/fault-tolerant to an extent and do not require all nodes to be online, in most cases, a percentage of nodes will need to be online to make consensus progress. The percentage is dependent on the DLT and the underlying consensus mechanisms. Some DLTs, such as Casper, can function as long as one node is online, while some DLTs require a minimum of five nodes or a predefined majority to be online. N2N DLTs are the exception, as they require all parties to the transaction to be online.

– We see the emergence of various types of nodes roles and number of nodes. For example, MultiChain's permissions management system consists of seven types of permissions; connect, send, receive, issue, mine, activate, and admin. It is possible for a node to have read-only access if it has connect permissions only. It is possible for a node to have read/write but no validation access if it does not have mining permission. There

is less value in a node having write but no read access, because it cannot build transactions if it does not know where it is receiving its assets from. Being able to change the permissions seems a logical choice for distributed ledgers.

– Permission-less ledgers such as Graphene and Bitshares 2.0, which use DPOS, utilize a key concept that echoes what MultiChain has allowed but for an open blockchain: Flexibility. In this case, flexibility of blockchain parameters, e.g., fees, number of witnesses, block interval, block rewards, etc. are all configurable by the committee, which is a separate group of elected stakeholders from the witnesses, which do not receive any rewards, but the ability to manipulate the global blockchain parameters by vote and applied in a maintenance window.

– The number of nodes involved varies based on the concept. While proof-of-work has no limitation of nodes initially competing to validate (mining), there are the other extremes being developed using only two transaction parties nodes to verify the activity (Corda).

– Most consensus mechanisms have, in general, about three validation stages, but we see a number of variations, which is particularly true for the voting process.

– Juno allows every message to be encrypted in whatever method the user prefers, and Corda is allowing for N2N data encrypted services. This allows the counterparties to transact in a private, confidential manner without revealing the content to any other parties.

– Consensus mechanisms vary in how they consider a transaction as "committed," "safe," or "live." However, generally speaking, a majority of participants are required to accept a transaction for finality.

– The definition of incentives for the participating nodes within a permissioned system depends on the financial services use case. Usually, the nodes will be extrinsically incentivized through legal contracts, operational targets, etc., between participants. Some DLTs can still be configured to use proof-of-work incentives or proof-of-stake disincentives, and this is configurable.

Governance, risks, and control

– Network participants mostly own their nodes, but in the case of some DLTs, the consensus provider owns or governs the nodes (e.g., Evernym). In some instances, the providers may own a percentage of nodes, but the overall network remains open for other participants to provide nodes.

– While Ripple initially owned all validating nodes, the market sentiment within permissioned systems is shifting toward the view that nodes should be owned by participants of the network.

– The governance model varies across the various used ledger set-ups. However, external legal contracts, the use of supervision/regulatory/observer nodes, and the use of an integrated permission model are all commons examples of governance mechanisms utilized by DLTs.

– Most DLTs intend to continue to rely on the existing legal and regulatory framework for the identification of malicious actions and enforcement of legal action. In addition, some DLTs (e.g., MultiChain, Hyperledger, and Corda) are just platforms or services and are not responsible for the malicious actions of the participants.

– Several different techniques are utilized to restrict malicious activities. These techniques include blacklisting nodes, locking concurrence ledgers, protecting access control systems, disconnecting peer nodes behaving maliciously, and allowing clients to interrupt leadership (in the case of leader-based DLTs). Many of these techniques are similar to those employed in non-DLT applications.

– Public key infrastructure is utilized by most DLTs to ensure the trustworthiness of other participants. Most described consensus mechanisms are based on the underlying assumption that the keys utilized to post to the chain are secure.

– The ability to use administrator nodes was mixed based on the DLTs we reviewed. In the case of some DLTs, the use of administrator nodes was also indicated as configurable.

– The onboarding and offboarding of nodes to the (permissioned) network is handled differently by the various software solutions. Some defer the entire mandatory know your customer (KYC) and anti-money-laundering (AML) procedures back to the participants, while others consider covering various degrees of those responsibilities as part of onboarding the nodes.

– Having known actors for nodes allows for the access rights of malicious nodes to be restricted in a much more forward fashion. They can be voted off or deleted quickly.

– Counterparty risks will continue to be managed externally. However, most distributed ledger technologies implementations are targeted at mitigating counterparty risk through the use of real-time transaction finality, verifiable authenticity, and other DL features.

Key observations

Performance

– Throughput, latency, and number of nodes are the general measures for DLT scalability and performance. High throughput of transactions is only necessary for certain capital market operations. Many blockchain use cases can tolerate latency in transactions.

– Since the bitcoin implementation, DLTs have made significant progress on the performance front. Most DLTs now have the ability to provide transaction finality from milliseconds to seconds and volume ranging from 500 to 5,000 final transactions per second. Furthermore, transaction speed on individual ledgers (not distributed consensus) can be as high as 100,000 per second in the case of some DLTs (e.g., distributed concurrence).

– Most DLTs do not enforce a limit on the volume of data or the number of fields. However, some implementations are limited by the size of the payload or the metadata. Performance (primarily latency and throughput) of many DLTs were identified as being impacted negatively with increasing scale as more nodes are added to the network. However, a good percentage of DLTs that we reviewed have indicated that scale has no impact on the performance of the system.

– Scalability is important for financial services operations that depend on the high volume of transactions throughput. Most distributed ledger systems to be built follow industry-specific design rules to meet scalability and speed, as well as privacy of data.

Security

– The security aspects of the consensus mechanisms are in its early stages and is evolving. There is diversity among the security features across vendors, which is driven by their base architecture and used consensus mechanisms. In most cases, the security testing is in progressing but does not yet allow for security fortifications. Based on the responses, the focus on audit standpoint seems to have taken a back seat to consensus mechanisms, security, and other components. The current approach is to build a fully functional model and then tweak the product based on issues and roadblocks.

– Various risks and vulnerabilities with regard to attacks continue to remain. Most DLTs are actively identifying these risks and vulnerabilities and enhancing the technology to address them.

– Not all DLT providers have thought about extensive security testing and certification for their ledger solutions as implementation in a heavy regulated environment would require. However, we noted instances where customers have already started requesting for security testing and audits.

– Loss of private keys continues to remain as one of the key risk for distributed ledger solutions. Many mitigation measures are considered, such as the ability to reset/re-issue keys, on-disk encryption of keys, multisigning, blacklisting keys upon breach, etc. Services that provide private key management should have a place in the future.

– "Double spending" is a well-recognized risk, and most providers have designed mechanisms with varying degree of sophistication in place that inherently minimize or prevent this risk.

– Many ledger solutions have extensive system security documentation in place, with others looking to add in the future.

Cryptography/Strength of Algorithm

– Some DLTs such as Juno allow every message to be encrypted in whatever method the user prefers and CORDA is allowing for N2N data encrypted services. This allows the counterparties to transact in a private, confidential manner without revealing the content to any other parties. Other DLTs such as chain not only encrypts the metadata but also uses zero knowledge proofs to cryptographically conceal the assets and amounts in transaction. These come with known scaling trade-offs which are attempting to be addressed throughout the industry.

– Consensus mechanisms vary in how they consider a transaction as "committed," "safe," or "live." But generally speaking, a majority of participants are required to accept a transaction for finality.

– The definition of incentives for the participating nodes within a permissioned system depends on the use case within financial services. Usually the nodes will be extrinsically incentivized through legal contracts, operational targets, etc., between participants. Some DLTs can still be configured to use proof-of-work incentives or proof-of-stake disincentives, and this is configurable.

– DLTs and their providers offer key generation code and libraries that can be utilized to generate public and private keys. Keys can be generated when the node is being set up. Private keys can be stored locally and do not need to be exchanged with nodes, similar to most modern-day public key infrastructure implementations. In many of the cases observed, keys are generated and stored on existing hardware security module (HSM) infrastructure in order to maintain the appropriate control.

– DLTs predominantly utilize multiple-error monitoring through measuring fault rations, message handling, etc. Some DLTs specifically provide for error tracing and monitoring capabilities on-demand, while others have a change feed mechanism, where all nodes will hear about the behavior of other nodes.

– Many of the consensus methodologies implemented and designed that are derivatives of PBFT have allowed for many of the settings to be changed in order to work better for certain use cases.

Tokenization

– Across the various consensus mechanisms, there are various degrees of self-enforcing rules to ensure incentives that make nodes behave honestly and cooperatively. In the absence of those incentives, the creators went down the Ripple/ Stellar route or rely on reputation. Most DLTs are focused on providing a technology layer across various assets. Some DLTs utilize native crypto currency (e.g., Ether for Casper and XRPs for Ripple), while various others do not utilize native crypto currencies but can still provide the ability to tokenize different assets. Chain uses a master transmission node to speed up processing.

– Digital signatures (or equivalents) are used by almost all DLTs to sign transactions. We therefore conclude that this is one of the key parameters of DLTs that will help with its adoption.

Privacy

– DLTs are taking various measures to ensure privacy. These include:

- Not including customer data on the distributed ledger
- Pseudonymous addresses
- Encryption and permissioning models
- Zero knowledge proofs
- Ring signatures.

– Almost all distributed ledger technologies require the use of verifiable authenticity through digital signatures, etc.

– Nodes generally have a certain degree of transparency of all other transactions, except in the case of N2N DLTs.

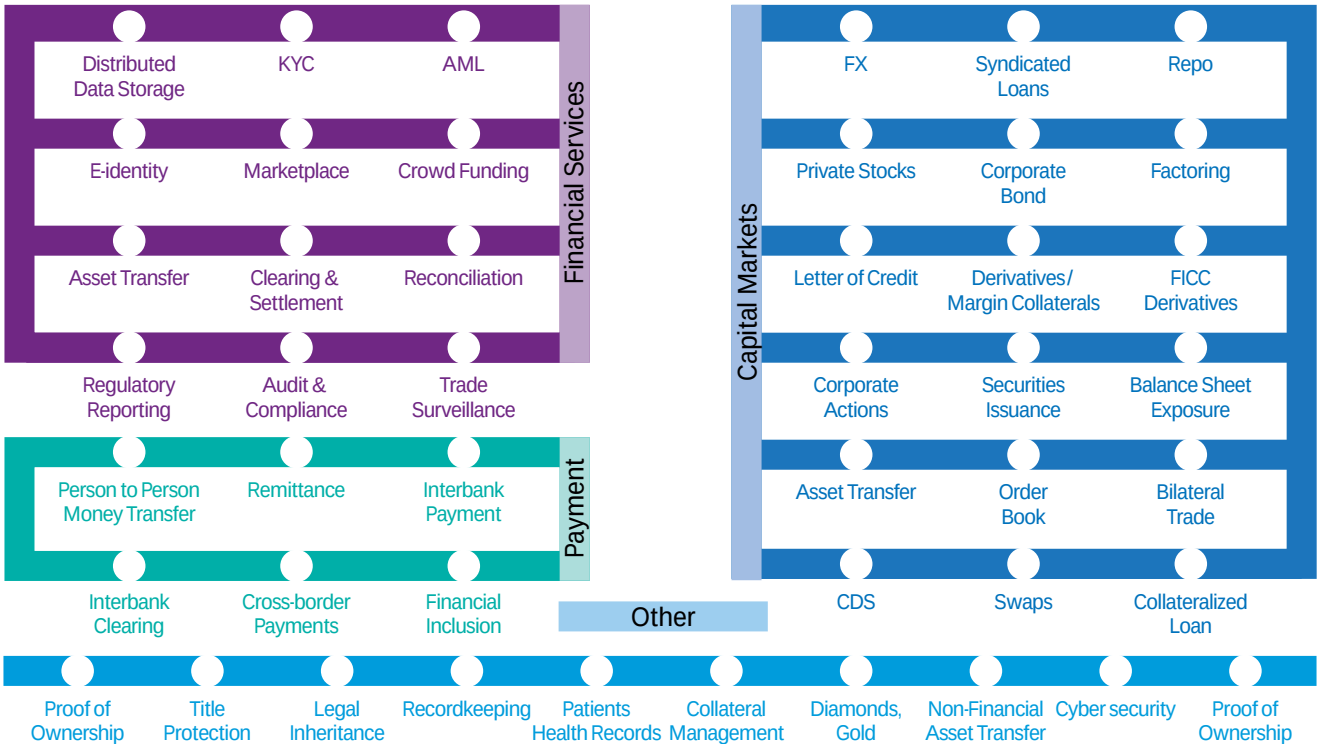
Implementation approach

– Implementation costs and time lines are mostly dependent on the specific use case, although the overall actual cost to license and deploy technology appears to be manageable. However, the actual total implementation cost of a distributed ledger solution for an entire asset class in the market is hard to judge.

– Various use cases include over-the-counter derivatives on fixed-income clearing corporation (FICC) asset classes and collateralized digital cash, international payments, crypto currency, loyalty points, and smart contracts usage for International Swaps and Derivative Association derivatives swaps.

Figure 7 below illustrates various examples for use cases being tested and partially implemented currently.

Figure 7: Use cases currently being tested and implemented



Is blockchain right for your organization?

Many financial institutions are working to take advantage of distributed consensus mechanisms, but there are many challenges. Regulations are heavy, cost is an issue, and the financial services industry as a whole is transforming quickly. Before making a big investment, institutions should consider some key questions:

– **Scope:** Which factors need to be considered?

– **Counterparties:** Which entities create and post transactions?

– **Process:** How is the process done today versus a DCL application? How is agreement/consensus reached on the business level/ data level? Who is allowed to validate transactions?

– **Data:** Which data needs to be shared with whom and when? What kind of assets will be transferred? Where should the data be stored? Does it need to be authenticated and notarized?

– **Technology:** What does the underlying existing technology landscape look like and in which way would it be impacted? What is the underlying technology cost?

– **People:** Which skills and organizational changes would be needed?

– **Regulatory:** Does that solution help to address my regulatory requirements in a more efficient way?

– **Industry:** Is there an industry-driven event requiring a refocus on current operations, to do things faster, with more trust.

– **Business case:** What is the overall business case, including the consideration of implementation cost? What is the return on investment? Is there sufficient scale effect?

– **Performance/security:** Can the solution scale to my needs? Are my security requirements met?

– Does the transaction record need to trigger further events (smart contract)?

Approach:

– What is the simplest way to solve for the problems in focus?

– Is a distributed ledger technology the right solution for the issue you are trying to solve for? Do you even need a blockchain, and if so, for which asset class and at which part of the life cycle?

– What are the specific issues you are trying to solve for?

– What are the design assumptions and goals for using distributed ledger technologies?

– Do you need immutability for your use case?

– Does the current business situation involve third parties?

– Are they known and trusted?

– Is there a central authority in place?

– Would the use case require a governance control framework?

– How do you agree with the immutable record of transactions (consensus)?

Implementation:

– How do you get started and how do you roll it out beyond the proof of concept?

– What is the right fit or technical tool kit to be used depending on requirements re scalability, security, performance, etc.?

– Private or public blockchain or off blockchain solution?

– Do you need more than one blockchain?

– What is your work flow?

– Where do you need which consensus mechanism? (Or none at all?)

– Which requirements does the consensus mechanism need to fulfill? (Our questionnaire in Appendix 3 may be helpful.)

– What are the various scenarios to be tested in an agile approach?

– How do you engage with regulators?

These steps may be hard to take successfully, given rigorous regulatory frameworks, tight investment budgets, and the fact that most companies already have highly complex technology landscapes. Just as importantly, most of the current distributed consensus ledger technologies are still works in progress. They can't simply be plugged in to provide solutions.

Any distributed ledger technology has to be both functional and acceptable to regulators. It has to be able to grow with changing needs and technologies—it has to be "future proof." Given IT departments' wariness about further increasing cost and technology, any new addition has to be more cost-efficient than the existing system.

For now, it's not possible to say that one consensus mechanism is clearly superior. One mechanism may be better for one use case, and another mechanism better for another use case. Companies will have to run case-by-case analyses, and these analyses will have to take into account that new mechanisms are still emerging.

Maneuvering the road ahead

DLTs and their underlying consensus mechanisms are already varied, and they're still changing rapidly as different use cases get mapped out. But in all this variety, some things are clear.

The most important question, as always, is how to best satisfy counterparties who may not trust one another but still want to hold safe transactions without a third party's involvement. Who are the participants or nodes, and what do they want/need to accomplish? These counterparties may want certain privacy features. They may want to control what other nodes are permitted to read or write. In some cases, the counterparties will want symmetry of information for all the nodes involved in a transaction, or they may want to determine different levels of information for different kinds of nodes.

It's important to focus on which consensus mechanisms are most relevant for individual companies, i.e., which can create real and scalable solutions that are also acceptable to regulators. For that, the question is less whether proof-of-work distributed ledgers or permissioned ones are better. It's more about whether or not, or when, consensus is needed for a particular use case. Transaction validation and nonfunctional requirements, such as performance and capacity, also play a key role in this decision-making process.

That means asking when distributed ledgers are really called for. The technology eliminates the need for a middleman, but in many cases, the cost is reduced confidentiality and privacy.

So, for many capital markets operations, when confidentiality and privacy is of the utmost importance, blockchains won't meet certain requirements. Closed systems, where only the involved counterparties interact, will remain the preferred option.

While the majority of financial services use cases prefer permissioned distributed ledger systems to open public systems for reasons mentioned throughout this paper, we see that very few examples of public blockchains are being tested, the Sydney Stock Exchange is the most recent announcement. It is not clear what the future may hold for open public systems in financial services as technical capabilities evolve to meet the strict industry requirements.

Most of the consensus models to date are based on theories from different industries and academia outside of what they are being applied to know and are having trouble scaling for financial services transactions which require hundreds of thousands transactions per second. One of the main problems involved is network stability. It's essential for the network to be running without stoppage even for a second. We expect performance and latency of distributed ledger technologies to continually

It's important to focus on which consensus mechanisms are most relevant for individual companies, i.e., which can create real and scalable solutions that are also acceptable to regulators.

Maneuvering the road

improve, but there may eventually be structural limitations with blockchain and distributed ledger technologies which might limit their adoption to use cases where there isn't a focus on very low latency and very high transaction volume.

Getting consensus right is really hard in a production setting, as most of the theory is academic in nature, and the real world has a way of proving theory wrong, particularly when it comes to complex systems like capital markets. It is still the early days.

Most distributed ledger technology is being used by banks for cost-cutting purposes to automate and streamline back-office operations. What is built on top of the blockchain in the application layer is what will be revenue generating.

There are always trade-offs when using one technology over another and for centralizing versing decentralizing ledgers. Privacy and confidentiality come at the expense of transparency and, with that, a different set of requirements which a blockchain may not be necessary for. Today's financial services industry, in particular capital markets, has been built on various standards over the years. Yet, although we are seeing consortiums being formed and regulators showing interest, the increasing proliferation of blockchain and DLTs has not shown true signs of standardization, which may become key for adoption and regulatory acceptance.

Data storage is another source of debate. Should every node on a blockchain store every record? In a decent-sized blockchain that requires nodes to store a lot of data about transactions in which they have no involvement. Is that a waste of effort and space? Will it be problematic to reconcile the different nodes and ensure consistency? What percent of data will be stored on the blockchain? What will be just hashed and stored locally? How data is being stored on and replicated to the blockchain is a key design feature.

Regulators, consortiums, and industry groups may end up having to dictate what kind of shared "write" databases are needed, based on protocols and standards.

If data storage isn't replicated in all the nodes on the ledger, only to certain nodes and to as little as N2N, then the system becomes more centralized and loses transparency, though it gains confidentiality and privacy. However, if data and transactions aren't distributed and consensus isn't always needed, then the question arises if a blockchain has any advantages over the status quo.

However, there are plenty of cases when consensus models and distributed ledgers, including blockchains, have fundamental advantages:

- When all parties need to know what data was transmitted to whom
- When the relevant parties need to view that information and can open the data to its own market
- When there is a clear and visible value chain that can be permissioned/securitized/quantified in risk/quantified in planning/factoring, as a state machine determines but can be permissioned to interested parties and ordinated (a node that has a business process to complete along that chain in a predetermined order)
- When immutable, interorganizational audit trails are needed
- When multiple parties need to directly write to a database without needing to trust each other and without the need for a middleman.

When is consensus needed? When should systems be centralized, and when should they be distributed? The debate is ongoing, but as we've demonstrated, there are valid reasons to use both systems, depending on the individual case's needs and the implementation possibilities.

And, it's clear the companies surveyed for this report agree. They're taking different approaches depending on their sectors and needs. But, in all cases, they're focusing closely on security, privacy, performance, and risk management.

Meanwhile, there are obvious signs that the excitement over the potential of distributed ledger technology is spreading outside of the financial sector and into the wider economy, where new business models based on blockchains are being developed. The excitement and buzz over blockchains and DLTs will pave the way for some fundamental transformation of certain otherwise inefficient processes within capital markets and financial services. Some of this transformation will extend beyond just the use of blockchain/DLTs but will be bucketed under the same umbrella.

In the end, the ability to operate in different sectors may be ultimately decided by determining which consensus mechanisms, blockchains or not, end up surviving and thriving.

Appendix 1
Key terminology

Authentication
The process of proving the counterparty identities and the existence of assets via private/public keys.

Blockchain:
A distributed database that maintains a continuously growing list of transaction records with various protections against tampering and revision

Consensus mechanism
A method to authenticate and validate a set of values or a transaction without the need to trust or rely on a centralized authority; can be constructed on and off a blockchain; a variety of approaches exist

Cryptography
The process of enforcing the authentication and cryptographic validation of transaction integrity via quorum structures and confirmation via code without the need to trust or rely on a centralized authority

Cryptographic signature
A method to mathematically validate the owner of a piece of data beyond any doubt if the user has kept the private key to sign the transaction safe

Delegated proof-of-stake
Delegated proof-of-stake stakeholders elect "witnesses," responsible for ordering and committing transactions, and "delegates," responsible for coordinating software updates and parameter changes.

Distributed ledger
A digital record of ownership that differs from traditional database technology, since there is no central administrator or central data storage; instead, the ledger is replicated among many different nodes in a peer-to-peer network virtual private network, and each transaction is uniquely signed with a private key

Fault Tolerance
The property that enables a system to continue operating properly even if some of its components fail

Federated consensus
A way to achieve Byzantine agreement (consensus), in which nodes can share another node and reach consensus without directly knowing all other nodes

Appendix 1: Key terminology

Governance
The establishment of a decentralized control—no central authority command whose approval is required for reaching consensus; some types of consensus mechanism use an elected leader who leads the validation and maintains the data which is been shared among the nodes. The governance aspect also includes the onboarding and offboarding of nodes within a permissioned network.

Hash functions
An application programming interface creates, through a process called hashing, a unique key or digital fingerprint for each file

Hierarchical deterministic keys
A deterministic wallet is a system of deriving keys from a single starting point known as a seed. The seed allows a user to easily backup and restore a wallet without needing any other information and can, in some cases, allow the creation of public addresses without the knowledge of the private key

Interledger protocol
Connects legacy ledgers of the past with the distributed ledgers of the future

Leader-based consensus
A type of consensus in which a leader is elected and stays in control until a vote decides on a new leader. In this model, it is the leader who validates transactions and sends data to the other nodes

Liveness
Refers to the transmission of data that is happening now and not a replay of a recording of data sent previously. Liveness is introduced into secure transmissions by mixing in a number that cannot be duplicated again. A node enjoys liveness if it can externalize new values without the participation of any failed nodes. Some nodes may fail, and as long as a majority of nodes are available, the network is still able to operate, can deal with latency (one or two slow servers will not impact overall consensus response times), and impact on the network bandwidth of ever-larger ledgers being distributed also has to be considered.

Merkle tree
multi-signature
An authentication function that allows a group of users to sign a single document with more than one private key.

Node
Members or systems of a consensus network; a server that holds a replicated copy of the ledger; can have varying roles: to issue, verify, receive, inform, etc. For all intents and purposes, a node can be a VM instance

Node-to-Node (N2N)
A mechanism in which only two nodes involved in a transaction take part; in effect, it eschews traditional consensus mechanism

Nonce number
A unique identifier used to get into a network just once

Permissioned
A private network in which users set rules about access, the consensus mechanism, governance, participation etc.

Practical Byzantine fault tolerance (PBFT)
A characteristic of a distributed computing system allowing for a certain amount of failures yet allows that system to continue operating and reach agreement. The traditional Byzantine consensus protocols today play a role in proof-of-concept settings where all nodes are known to each other (permissioned system, and authenticated and trusted validators within the network are chosen at random but always at a majority, which is resilient to Byzantine imposters and Sybil attacks.

Public blockchain
A network in which anyone can participate by reading data, submitting transactions, and participating in the validation process

Public key:
the public address where other wallets send transaction values

Private key
An encryption key uniquely linked to the owner and known only to the parties exchanged in a transaction; it is secretly held in a digital wallet.

Privacy:
Ensuring that only the receiver intended can read the message. The field of computing cryptography addresses many security and privacy issues of distributed consensus through the use of mathematical formulas for specific secure communication requirements within the context of any application-to-application communications

Proprietary consensus mechanism:
A consensus model that is unique in nature and may or may not be based off of any existing consensus algorithms

Quorum structures
The styles and stages used by nodes in a network to exchange messages asserting statements (can technically be differentiated by factors such as (nodes) leader election, types of leaders, the method of validating transactions, fault tolerance levels, utilization of tokens, strictness of algorithm, liveness guarantees, and permissions management)

Remote procedure call
a protocol that one program can use to request a service from a program located in another computer in a network without having to understand network details, also sometimes known as a function call or a subroutine call

Round-robin
A consensus mechanism in which nodes take turns at being the leader.

Scalability
The capability to cope and perform an increasing throughput and maintain or even increase its level of performance or efficiency when tested by larger operational demands. Latency is the delay in transaction processing

Security
Distributed ledger security is the process for protecting and safeguarding business and personal data, as well as transaction information. The validation of the results should be correct under non-Byzantine failures; also includes integrity (an assurance to the receiving node that a message received has not been altered in any way) and nonrepudiation (a mechanism to prove that the sending node really sent this message). Security can include digital signatures as a feature

Sidechain
The transfer of assets from one mechanism to a separate "pegged" mechanism; special-purpose ledger

Throughput
A measure of how many transactions can be processed in a given amount of time

Tokenization:
The process of replacing sensitive data with unique identification symbols that retain all essential information about the data without compromising its security

UTXO:
An unspent transaction model, in which assets are passed directly from one transaction's outputs to the next one's inputs and each output can only be spent once.

Appendix 2
Consensus mechanism evaluation questionnaire

Appendix 2: Consensus mechanism evaluation questionnaire

Framework category	Questionnaire
Overall Consensus Methodology	What is the underlying methodology used by the consensus mechanism?
	How many nodes are needed to validate a transaction? (percentage versus number)
	Do all nodes need to be online for a system to function?
	Does the algorithm have the underlying assumption that the participants in the network are known ahead of time?
	Who has ownership of the nodes (e.g., consensus provider or participants of network)?
	What are the different stages involved within the consensus mechanism?
	If applicable, what conditions are needed to be met to enter and exit each stage of the consensus mechanism?
	If applicable, what is the voting process after the "propose" stage?
	When is a transaction considered "safe" or "live"?
	Are there multiple rounds of vetting to decide which set of transactions are going to make it into the next round of consensus?
	How much time does a node need to reach a decision?
	How much time is actually needed to build the consensus until a new block is added?
	Does the system contain synchronous node decision-making functionality?
	What is the number of current and planned validators?
	What is the fault tolerance? How many nodes need to be compromised before everything is shut down?
	Is there a forking vulnerability?
	How are the incentives defined within a permissioned system for the participating nodes?
	What process does the system follow when it receives data?
	How is data currently stored?
	How does a party take ownership of an asset?

Framework category	Questionnaire
 **Governance, risks and controls**	How is governance/control enforced?
	Who is responsible and what are they responsible for in case of malicious actions within the network? How does legal action take place?
	Is there an intrinsic penalty mechanism in place for an attempted corruption of the consensus?
	How does the consensus mechanism allow access?
	How does the consensus mechanism restrict access concerning malicious activities?
	What is the permission management process? What is the process for adding or deleting nodes?
	How does the protocol assess the trustworthiness of other participants?
	Are there separate admin/administrator privileges? Who manages them?
	Are there restriction/privacy rights defined and enforced by a node?
	Can a node or a user have only "Read" or only "Write access?" Is specific node access required if performing only one functionality (e.g., back office outsourcing)?
	What are the measures in place to reduce risk?
	In case of permissioned systems, who manages the KYC/AML process and where is the data stored?
	How is counterparty risk settlement risk addressed?
 **Performance**	How long does it take for transactions to be validated and/or consensus to be achieved?
	What are some general measures of volume that the consensus mechanism can or will handle (e.g., number of trades)
	What are some general measures of the value that the consensus mechanism can or will handle (e.g., value of trades, in dollars)
	How do you measure scalability?
	Is there a limitation on the number of fields within a transaction?
	Is the speed of the system impacted if the system is made more scalable?
	Does synchronization have any impact on scalability?

Appendix 2: Consensus mechanism evaluation questionnaire

Framework category	Questionnaire
 Security	How is transaction activity monitored?
	Does the consensus mechanism utilize digital signatures?
	How does the consensus mechanism address an assumed industry standard?
	Which risk/security issues are currently being worked on?
	Are there any plans for getting the application/consensus mechanism certified (e.g., ISO, SOC, etc.)?
	What are the infrastructure hosting options? (e.g., cloud, hosted in a data center, etc.)?
	How would you describe the security testing performed to date (if any)?
	How are you planning to implement/integrate digital wallets? (including private key management)?
	In case of a breach, what data is at risk?
	How does the system prevent signature fraud (e.g., stolen keys)?
	Does the consensus mechanism have full documentation in place?
	How is the system expected to address general server issues?
	How does the consensus mechanism address the risk of "double spending"?
	How does system ensure network synchronization? What is the time needed for the nodes to sync up with the network?
	Do the nodes have access to an internal clock/time mechanism to stay sufficiently accurate?
	Under which conditions does a lock/unlock happen? (i.e., what is the proof safety?)
	What is the process for disaster recovery?
	What is the threat model being tested? What has been defined as 'normal'? How is fraud monitored?

Framework category	Questionnaire
Privacy	How does the system ensure privacy?
	Does the system require verifiable authenticity of the messages delivered between the nodes?
	Do all nodes have visibility into all other transactions?
	How is privacy defined and ensured between applications?
	How does the data encryption model work?
	If consensus happens in a permissioned network, are random public keys issued for every single transaction to increase the privacy, or does randomized CUSIP translation factors take place?
	Are participants' identities hidden from one another (e.g., Blackpool)?
Cryptography/ strength of algorithm	How are the keys generated?
	What does the key life cycle management look like?
	What is the library approach?
	What is the HSM integration approach?
	Does the consensus mechanism require a leader?
	How strict is the consensus mechanism? (Is the system strictness hard-coded or built with code flexibility?)
	Is node behavior currently measured for errors?
Tokenization	How are the assets tokenized (if applicable)? How would you briefly describe the tokenization concept and terminology?
	Which security mechanisms are assigned to the tokens?
	How would you briefly described the lifecycle management process for the tokens?
	Does the consensus mechanism utilize transaction signing?

Appendix 2: Consensus mechanism evaluation questionnaire

Framework category	Questionnaire
 **Implementation approach**	What are the current use cases being explored, tested, or implemented?
	What is the implementation cost?
	What is the time required to implement?
	Is there a reviewed business case to compare the implementation costs (including cost of the solution) to the current as-is process?
	Who are you currently working with? (e.g., venture capitalists, banks, credit card companies, etc.)?
	Are participants' identities hidden from one another (e.g., Blackpool)?

Appendix 3: Questionnaire response set

Please visit kpmg.com/us/blockchain-consensus-mechanism to download the questionnaire response set.

Acknowledgments

We'd like to acknowledge the contributions of numerous people in the blockchain network, many of whom reviewed and verified portions of this paper:

KPMG: Bob Hayward, Kiran Nagaraj, Walter Murphy, Mihai Liptak, Roshan Rao, Burak Karvan and Francis Sam Yesurathinam

BigChainDB: Trent McConaghy – gtrent@gmail.com

Bitshares 2.0: Ryan R. Fox – ryan@ryanRfox.com

Casper: Vlad Zamfir – vldzmfr@gmail.com

Directed Acyclic Graphs: Aviv Zohar – avivz@cs.huji.ac.il

Distributed Concurrence: Dan Conner – dan.conner@disledger.com

Evernym: Jason Law – jason@evernym.us
Timothy Ruff – timothy@evernym.us
Drummon Reed – drummond@respect.network

Graphene: Ryan R. Fox – ryan@ryanRfox.com

MultiChain: Gideon Greenspan – gideon@coinsciences.com
Maya Zehavi – mayazi@gmail.com

OpenChain: Flavien Charlon – flavien.charlon@coinprism.com

Ripple: Bob Way – bob@ripple.com

Steem: Ryan R. Fox – ryan@ryanRfox.com

Stellar: Jed McCaleb – jed@stellar.org;
Joyce Kim – joyce@stellar.org

Tendermint: Jae Kwon – jae@tendermint.com

Note: Ryan Fox does not represent Cryptonomex, Inc., Steemit, Inc., nor any other entity within the blockchain space. His responses are his own informed opinions based upon independent research.

Contact us:

Simon Gleave
Regional Head of Financial Services,
KPMG Asia Pacific
T: +86 10 8508 7007
E: simon.gleave@kpmg.com

Beijing
Raymond Cheong
Partner
T: +86 10 8508 5458
E: raymond.cheong@kpmg.com

Hong Kong
James McKeogh
Partner
T: +852 2847 5018
E: james.g.mckeogh@kpmg.com

Simon Phipps
Partner
T: +852 2143 8813
E: simon.phipps@kpmg.com

Shanghai
Longhua Zhang
Partner
T: +86 21 2212 3378
E: longhua.zhang@kpmg.com

kpmg.com/socialmedia