BLUEPRINT OF
BLOCKCHAIN PLATFORM
FOR BANKING SECTOR AND BEYOND

IDRBT
Explore, Enable, Excel

Institute for Development and Research in Banking Technology
(Established by Reserve Bank of India)

CONTENTS

FOREWORD

The Harvard Business Review says that Blockchain Technology is a foundational technology. A foundational technology is likely to have impact on human population across the geographies in several ways. However, it may take time for such a technology to get adopted and absorbed. It is, therefore, no surprise that Blockchain Technology is more discussed and debated than designed and deployed.

While discussing the relevance of any technology for a large-scale adoption, it is useful to ask two questions – (1) is it the right time to use the technology, and (2) is the technology likely to survive long enough to be useful. Recent history has witnessed instances of ideas prematurely used, dropped and reused later. Similarly, past few decades saw very quick exit of ideas within a short time of being considered great and futuristic. These two questions are raised in the context of Blockchain Technology too. For the time being at least, we can live in the comfort that the technology has been in practical use for over a decade.

Accepting that the Blockchain Technology, in view of its inherent strengths, is useful and that it is the right time to look at its usability, the usual questions like what, why, where, and how are to be addressed. IDRBT in its Whitepaper on Blockchain Technology attempted to give reasonable answers to what, why and where, at least in the context of banking sector.

That leaves the most important question – how. The Institute has been working closely with government, banks and industry in addressing the question – how to build a useful Blockchain that can serve as a common platform to launch varied applications. It virtually means, preparing a Blueprint for building Blockchain platform.

For drawing specifics to a generic platform, the Institute has leveraged the experience and expertise it has gained while working on Blockchain based projects for government as also from the interactions

it has been having with Blockchain Technology consortiums. Further, the Institute has constituted a team of experts from banks, industry, academia in addition to its own researchers.

The Blueprint is a result of the commitment of the members of the team. Let me thank and congratulate each and every member of the team.

The Blueprint is organized into three chapters. In the first chapter, it is attempted to present the current scenario of financial networks, further requirements and benefits that might arise out of adoption of Blockchain Technology. The second chapter is devoted to design aspects of three important constituents – architecture, governance and technology – that will help in successful deployment of Blockchain based services. The third and last chapter gives roadmaps for implementation, taking into account the requirements and designs discussed in earlier chapters.

I trust the Blueprint is likely to become a standard reference for any individual or institution or consortium that is working towards building a common Blockchain based platform. Though the focus of the publication is banking sector, in view of the generality of principles and practices, I am certain it would go beyond.

Date: January 22, 2019

Place: Hyderabad

(Dr. A. S. Ramasastri)

Director, IDRBT

IDRBT
Explore, Enable, Excel

• CHAPTER - I

PREAMBLE

Information and communication technologies (ICT) have become an integral part of many businesses to such an extent where business needs drive innovations in ICT. Blockchain Technology (BCT) has been hailed as a foundational technology that has the potential to disrupt many industries. BCT has existed for over a decade now, and the past few years have seen an explosion in both the quantity and variety of experiments. These experiments are from one of the two categories: (i) application of BCT to enhance existing business processes – benefits being improved efficiency, reduced fraud and costs, and (ii) introduction of new business processes made possible due to BCT – benefits being creation of a niche service that acts as a key differentiator among the competitors. Benefits of the first category of applications are easier to quantify – leading to a smooth path to adoption.

This chapter highlights the requirements of business networks, the extent to which they are supported by existing centralized systems, and how blockchain technology could be leveraged to complement the existing systems to enhance support for the business requirements.

Requirements of Business Networks

The following requirements of business networks are discussed in this section: (1) Business continuity and availability (2) Guarantees on message delivery (3) Guarantees on results of business processes (4) End-to-end security (5) Data integrity and finality (6) Privacy (7) Latency (8) Asynchronous processing (9) Scalability (10) Shared views (11) Business rule validation. The focus here is to clearly define the business requirements together with some high-level details on how they can be realized.

A few commonly used terms are defined below:

- **Entity** is a participant within the business network
- **Ledger** is where events that involve two or more entities are logged
- **Node** is a system comprising of hardware, software and network components, which is executing business logic. In case of deployments with blockchain, nodes are used to maintain ledger. It is possible that a single node may keep records of multiple entities so far as the records are clearly separated
- **Access points** provided by the network implement business APIs, which user applications invoke.

Business Continuity and Availability

Business continuity implies that even with failures (especially failures of single component), business network continues to function albeit with reduced capacity. This can be described as follows:

Failures within a node: It is expected that failure within individual nodes must not lead to any kind of data loss. This could be implemented in a variety of ways. For example:

- Conventional techniques could be used to ensure no data loss i.e. zero RPO based implementation of storage or database replication. For making RTO as close to zero as possible, redundant controllers and disks with active-active deployment at primary and secondary sites would help
- Blockchain based implementations may offer an interesting alternative – an entity could own, for instance, node1 and node2, which would stand-in for each other, i.e. each transaction is recorded by both the nodes but validation is carried out by either node. Network needs to treat node 1 and node 2 as equivalent. In typical blockchain based

implementations, it could be possible to reconstruct ledger (transactional storage) at a node by getting blocks from other nodes

- It could even be possible that end user applications can be provided multiple access points in order to ensure that the impact of node failure on end user application is minimized or eliminated. The downside is that, recovery of failed node takes longer and would put additional load on surviving nodes.

Failures within network: These are generally short-lived failures that could be addressed by retry logic at protocol level or they could require longer time to recover, in which case, usage of alternative network paths to reach required entities will be warranted.

Failures at site: These are catastrophic scenarios wherein an alternate site may need to be made operational within a short interval.

Guaranteed Message Delivery

This broadly means that if network indicates that a message has been taken up for delivery, no matter what, it will be eventually delivered to the intended entities. More specifically

- Delivery needs to take place irrespective of failure either within an individual node or of a link within underlying physical network. This can be attained as mentioned above under 'Business continuity'.
- For blockchain-based implementations, a transaction can be included within blockchain only after it is validated by a certain number of nodes as per the policy of the network. Guaranteed message delivery comes into play only after such a validation is done.
- For blockchain-based implementations, order in which blocks (which contain individual transactions) appear within individual ledgers is important since hash(chain of 'n' blocks) = hash(chain of 'n-1' blocks + contents of block 'n'). Even otherwise, it is possible that transaction T2 depends upon the effects of transaction T1 and hence T2 may need to be posted only after T1.
- If a network depends on a centrally deployed component (e.g. 'orderer' within Hyperledger Fabric or 'notary' within Corda network), that component must be deployed in a fault tolerant manner.

Guarantees on Results of Business Processes

Blockchain based business networks are natural candidates for implementation of pan enterprise business processes because:

- BPMN standard rigorously defines the way a business process can be described, in simple terms, business process involves transition from one state to another based on certain events. This event could be an outcome of an action taken by a human operator or it could be based on an automated action triggered by invocation of an API provided by the business application.
- Blockchain based systems provide a natural trigger whenever a transaction is included in the "ledger" stored in any node. This trigger can be used to queue up a work item within inbox of an individual user for manual action or to invoke an API provided by business application to implement an automated action.
- Blockchain based systems provide natural mechanism to validate each transaction which, as mentioned, leads to state changes within pan enterprise process.
  - Business logic used to validate the state transition is called as smart contract and blockchain implementations require strong governance to ensure that all validating nodes use the same version of business logic.
  - Validation logic needs to be repeatable e.g. it cannot have any constructs, which can lead to different results on repeated invocations e.g. random number generation or processing based on time of day, etc.
  - Outcome of business process is based on

series of state transitions, which takes the business process from 'started' to 'complete'. By virtue of each state transition being validated by multiple nodes, consistency of pan enterprise business process results can be guaranteed.

End to End Security

This covers various aspects like securing transmission from one end user application all the way to another end user application. It covers authentication and authorization of end user applications using the network. More specifically

- It is to be checked if entity invoking a particular transaction is (i) authenticated to use the network and (ii) authorized to invoke that particular transaction (role-based access control).
  - In case of blockchain implementations, above checks may need to be done as a part of logic that runs in multiple validating nodes.
- To secure data transmission, all network links through which the message passes need to use encryption like TLS. Usage of mutual TLS (client-side certificates in addition to server-side certificates) can ensure non-repudiation.
- Applications need to comply with OWASP and other geography and domain specific security guidelines. Examples could be prevention of injection attacks like SQL injection, OS command injection, cross site scripting, comprehensive server-side validations, etc.
- Deployment and operations need to follow Information Security guidelines like ISO 27000. This would cover encryption of data at rest and segregation of duties between various roles within the data centre.

Data Integrity and Finality

This means, an assurance that data either during transit or while at rest cannot be tampered. Finality specifically refers to assurance that transactions already recorded within the ledger cannot be changed. Note that some of these points overlap with end to end security discussed above. More specifically:

- Inclusion of hash of important fields within a message signed by originating entity using its private key allows final recipient(s) to ensure that message has not been tampered and has indeed been sent by the intended originator.
- Classic blockchain based implementations provide additional protection for data integrity since hash(chain of 'n' blocks) must equal hash(chain of 'n-1' blocks + contents of block 'n'). This means that a block that is already within the blockchain cannot be changed since that would affect hash calculation of all subsequent blocks and of the overall chain.

Privacy

This mainly covers dissemination of information to entities only on 'need to know' basis. To elaborate:

- If a buyer is sending a purchase order to a seller, only buyer, seller (and possibly their banks) need to know. If details of the PO document are revealed to other entities, business interests of buyer or seller could be hampered.
- Ideally, the transaction details of the above transaction should not be made available with other entities as a first choice. To achieve this, sensitive details may need to be encrypted in such a way that only respective parties would be able to decode the details.
- If a single node is keeping records of multiple entities, there needs to be clear separation between these records to ensure that an entity (including super-user of the node) can never see unencrypted records of another entity.

Latency

This mainly covers measurement of response time e.g., time it takes to invoke a business API provided

by the network or in other words, time it takes to receive a response message, once a request message is sent to the access point. Lower the latency, the better. Response time is dependent on several factors:

- In conventional implementations, response time depends on processing capacity of the central node(s), efficiency of business logic and considerations for scalability. Response time in such cases is more deterministic compared to blockchain based implementations, which involve multiple nodes for validation and processing.
- Response time depends on quality of network link between access point and the processing node. For blockchain based implementations, it also depends on speed of links between processing node and other validating/ processing nodes. Propagation delay and available bandwidth affect performance of a network link.
- For blockchain based networks, number of nodes that require validating a transaction as per network policy and time consumed for execution of validation logic on the slowest validating node determine the response time.
- Response time obviously depends on hardware commissioned e.g. CPU, memory, storage, etc.
- Design of the API e.g., whether API returns after minimal validations with a subsequent callback.

Asynchronous Processing

This mainly covers ability of network to provide APIs, which have asynchronous implementations:

- While asynchronous implementation of API is a powerful programming paradigm in general, it makes even more sense for blockchain based implementation of business APIs since these could produce slow response time due to involvement of multiple nodes for validation and processing.
- In blockchain parlance, asynchronous processing could comprise of an access point that performs minimum required validations and return back "pending" status for a business API.
- Subsequently access point could make a callback to end user application with updates like (a) validations across required number of nodes is done, and (b) transaction processing is complete and transaction is now a part of blockchain.

Scalability

Scalability of the system is measured in terms of ability to enhance processing capability of the system by adding hardware. For example, linear scalability would mean – doubling hardware leads to double the number of transactions processed per second (TPS). Most systems aspire to achieve near linear scalability. Typical impediments for linear scalability and ways to remove these will be as follows:

- Pieces of code that can execute on only one processing thread at a time cause non-scalable behavior e.g. synchronized blocks within Java or pieces of code, which lock a set of database records and prevent other concurrent executions.
- In classic blockchain implementations, validation of blocks by geographically separated nodes and adding these to their respective blockchains creates challenges for scalability. These are typically addressed by increasing number of transactions within a block. The permissible maximum block size depends on practical considerations so far as network bandwidth is concerned.

Shared Views

End user's expectation about transparency provided by bank's processing systems is increasing. One such expectation could be – bank and end customer share a view of the transaction data that pertains to

that end customer. Effectively, what customer sees is 'exactly' the same transaction data that bank user sees and not a derivative i.e. monthly statement mailer. The same logic can be extended to views of entities participating in a network.

Business Rule Validation

Expectation of design of applications used by business networks is to provide flexibility to users of the network. A technique used to provide required flexibility is to separate decision making from the application code. This has led to calling an externally defined business rule from within application code so that a business user within the bank can change the rule within specified contours, without expecting a new release of the application incorporating the required change.

In case of blockchain, each transaction needs to be validated by multiple nodes before it is added to the blockchain and as mentioned before, validation code needs to produce same results any time it is executed. These business rules are executed through smart contracts and this code is clearly separate from rest of the application code. Further, there is a need to standardize certain business processes across the entities participating in the blockchain network. These need to be clearly defined by the governance mechanism.

Features of Current Networks

Current business networks support some of the requirements listed in the previous section but fall short on other requirements. The table below provides an overview for the extent of support by current business networks:

Requirement of Business Network	Level of Support	Description
Guarantees on Message Delivery	High	Currently, financial messages are exchanged securely with non-repudiation capability and guarantees on message delivery.
Guarantees on Results of Business Processes	Low	Business processes are executed locally at each entity and the results are relayed to subsequent entities for further processing. Further, each entity develops its own processes based on their interpretation of the business agreement. This naturally may lead to disagreements, that are rectified through reconciliation. Current business networks do not automatically provide for guarantees on business processes.
Business Continuity	High	Business continuity is ensured by adding redundancy through disaster recovery procedures at each entity. The same measures also provide high availability in normal operating scenarios. Guaranteed message delivery mechanism further ensures near-complete synchronization of data across parties after system failures.
End-to-End Security	Low	In a multi-hop transaction, each entity has access and control only on the parties it is directly interacting with. Ad hoc mechanisms for end-to-end transparency can be implemented but may not be real-time.
Privacy	High	Information is stored in siloed manner within each entity. Only relevant information is shared with parties on need to know basis.

Requirement of Business Network	Level of Support	Description
Data Integrity	Medium	Information shared across multiple parties is processed locally at each entity potentially resulting in data discrepancy. Reconciliation addresses this issue to a certain extent.
Latency	High	Centralized systems are technically capable of low latency. In multi-hop transactions, there may be delays introduced at each party, thus reducing the overall latency of the business process.
Asynchronous Processing	High	Current interactions between entities through messaging middleware inherently support asynchronous processing.
Finality	Medium	In a multi-hop transaction, finality is reached for each party at different hops of the transaction. Due to this, there is a need to process transaction reversals/modifications if there are issues in end-to-end transaction finality.
Scalability	High	Current deployments have proven scalability utilizing both horizontal and vertical scaling.
Shared View	Low	Current business networks do not offer shared views. Each organization reconstructs the ledger based on the information/messages received from other parties. Consistency of the information is not automatically guaranteed by the business networks.
Business Rules Validation	Medium	Current systems are strong in validation of business rules before transactions are processed within the entity. In multi-party scenarios, these business rules are agreed centrally but implemented at each party separately.

Current business networks have proven capabilities in terms of key requirements like guaranteed message delivery, business continuity, privacy, latency and scalability. Current business networks have certain limitations in the context of transactions involving multi-party flows:

- Current business networks focus on guaranteed exchange of information/ messages between different parties but cannot guarantee on the processing and business outcomes of the messages.
- In multi-hop transactions, current business networks process each hop separately, which requires efforts to develop an end-to-end view of the transaction. Different hops of the transaction are finalized independently, thus creating risks and the need for rollback/ modifications
- Current business networks do not provide a guaranteed shared view of the information among the involved parties. This needs to be built by each party based on partial information it received form others
- Current business networks do not provide a mechanism for sharing business rules and processing logic across multiple parties. Trustworthy solutions for business logic sharing is the need of the hour.

Benefits of Blockchain Technology

Some of the limitations of current business networks for realizing business requirement have been discussed in the previous section. This section describes how blockchain technology can complement the existing business networks to overcome some of the limitations. A comprehensive discussion of blockchain technologies is beyond the scope of this document. A brief overview is presented below.

Distributed Ledger Technology (DLT), also commonly known as Blockchain, can be understood as a distributed database. DLT promises to preserve both consistency and availability using different consensus mechanisms. It uses hash algorithm to authenticate data on the Blockchain, allowing different nodes to replicate data among each other for high availability, without exposing the data to unauthorized access.

With these features, there are many applications of DLT especially in financial services sector. For instance, trade finance can utilize DLT to share the information about a trade from purchase order, bill of lading, open account financing, all the way to invoicing. It eliminates forged invoices with proof of authentic supporting documents on the DLT network.

Key features of the blockchain:

Distributed ledger: Identical copies of the information are shared on the blockchain. Participants independently validate the information without a centralized authority. Even if one node fails, remaining nodes continue to operate, ensuring no/low disruption to business. Furthermore, the decentralized storage in a blockchain is known to be failure-resistant. Even in the event of failure of a large number of network participants, the blockchain remains available, eliminating the single point of failure.

Near real-time updates: Based on deployment policies, the information on the blockchain nodes are updated in close to real-time. The transactions can be globally validated once they are part of the chain.

Chronological and time-stamped: Blockchain as the name suggests is a chain of blocks each being a repository that stores information pertaining to transactions and also link to the previous block. These connected blocks form a chronological chain providing a trail of the underlying transactions. Further, the blockchain can be designed to also keep information about transaction chains, that could demonstrate either (i) the source of inputs, or (ii) the linking between various hops in a business process across entities.

Cryptographically sealed: Blocks are cryptographically sealed in the chain. This means that it becomes impossible to delete, edit or copy already created blocks and put it on the network, thereby creating true digital assets. This ensures high level of robustness and trust. Data stored on blockchain are immutable, irreversible and auditable.

Programmable and enforceable contracts: A transaction on the blockchain can be executed only if all the concerned parties' consent – consensus rules can be designed to suit various business scenarios.

The features discussed above can enable the business network to exploit blockchain technology for:

- **Guaranteeing Results of Business Processes:** Distributed shared ledger along with programmable and enforceable contracts in blockchain provide this feature.
- **Improving Data Integrity and Finality:** Cryptographically sealed ledger with chronological and time-stamped transactions in blockchain provide this feature.
- **Providing a Shared View:** Near real-time updates coupled with distributed ledger provides shared view.
- **Validating Business Rules:** Programmable and enforceable contracts provide the mechanism to enforce and validate the shared business rules.

• CHAPTER - II

BLUEPRINT OF A BUSINESS BLOCKCHAIN

THE previous chapter, looked at requirements for business networks like business continuity and availability, scalability, latency, privacy, security, shared storage, data integrity, guarantees on message delivery and results of business processes and asynchronous processing. It also discussed how blockchain complements current business networks to better realize the requirements.

This chapter presents the blueprint for a business network leveraging blockchain. All the three aspects of a blueprint viz. Architecture, Governance and Technology are discussed.

Architecture

At a conceptual level, blockchain can be considered as a database shared between various entities, which are part of a business network. This database captures ownership of assets, which are allotted to an entity or shared between various entities and transfers from one entity to another on the network. Assets on blockchain could represent digitized physical assets or intangible assets like purchase order or invoice. For digitized physical assets e.g. land, blockchain transaction history would show transfer of land from one owner to another. For intangible assets, transactions would represent onboarding of an asset and/or sharing the asset with the required entities. For example, buyer sends a purchase order to seller who forwards it to the seller's bank. Transaction history can prove that purchase order is genuine. Sharing or movement of assets may be a part of a pan-enterprise process e.g. buyer could be sending a purchase order to a seller as part of 'Purchase order to invoice' process.

Broadly speaking, blockchain needs to meet two objectives: (a) describing ownership and (b) protecting ownership. As described above, individual transaction on blockchain describes transfer of ownership and transaction history needs to protect the current state of ownership.

Layers of the Architecture

End-user applications need to invoke multiple operations on the blockchain platform. They will have identities (probably driven by end-user identity) to interact with blockchain. The interface exposed by blockchain to interact with end-user applications is discussed as "Application" layer below. The alternatives for internal implementation of blockchain are discussed under "Implementation".

Application: Application layer provides an interface, which allows transfer of ownership of asset from one entity to another. From end user's perspective, while underlying implementation could be complex, application interface needs to be highly available and simple to use. Overall application architecture needs to be "flexible" to handle various kinds of assets. For example, application could deal with transfer of fiat currency from one entity to another, wherein it need not track individual units of fiat currency. On the other hand, it could also be used for transfer of an asset like "car" from one individual to another wherein each unit of the asset is identifiable by an engine chassis number, for instance. Note that the assets could be digitized physical assets or intangible assets.

Implementation: This mostly concerns "ownership logic", which is described in detail below. It is expected that "ownership logic" is implemented in a modular fashion. Multiple components may be independently plugged-in without impacting the design choices for the rest of the system. Transaction security algorithms should be "extensible" so that latest security algorithms could be used. Transaction processing and consensus

logic need to be designed in a secure manner. Design goal of storage logic (data at rest) is to ensure it is resilient to tampering. Finally, distributed peer-to-peer architecture needs to maintain integrity e.g. allow communication to take place only between registered entities.

Blockchain Reference Architecture

Considering that the primary purpose of blockchain is to provide a trustworthy mechanism to validate transfer of assets and record change of ownership from one entity to another, a reference architecture to cover all components required to implement ownership logic is presented below:

<table>
<tr><th colspan="4">Ownership Logic (validation of transfer and recording)</th></tr>
<tr><td colspan="2">Proof of Ownership</td><td colspan="2">Transfer of Ownership</td></tr>
<tr><td colspan="2">History of Transaction Data</td><td colspan="2">Individual Transaction Data</td></tr>
<tr><td>Transaction Processing Logic and Consensus</td><td>Transaction Security</td><td>Storage Logic</td><td>Distributed peer-to-peer Architecture</td></tr>
<tr><td>✶ Distributed consensus
✶ Validation of transaction data
✶ Validation of block headers (chain)</td><td>✶ Identification and authentication
✶ Authorization
✶ Digital signature
✶ Cryptographic hash values
✶ Asymmetric cryptography</td><td>✶ Immutable, append-only data store
✶ Asymmetric cryptography</td><td>✶ Doorman service to onboard peers
✶ Secure and resilient network
✶ Message passing and guaranteed delivery</td></tr>
</table>

Primary function of blockchain implementation can be considered as maintenance of state of the overall system that constitutes the state of various entities participating within the network. This state is either implicitly or explicitly maintained. For example, in case of bitcoin, state of the system is set of spendable outputs which are allotted to specific addresses. When bitcoins are transferred from one entity to another, bitcoins move from spendable output(s) belonging to transferor to spendable outputs that belong to transferee. Implementations like Ethereum represent these states more explicitly calling the representation as "world states". Similar representation is done by Hyperledger Fabric at the level of a channel. Corda does the same but a node stores only subsets of states that concern the particular node.

Thus, proof of ownership is recorded within the state database implemented through blockchain. Transfer of ownership is carried out through an individual transaction and history provides entire trace of movement of the ownership.

Transaction Processing Logic and Consensus

This logic needs to ensure that only valid transaction data are added to the collectively maintained history of transaction data. Its components are as follows:

Distributed consensus: Distributed consensus ensures that only valid transaction data are added to the data structure. Broadly, consensus in permissioned blockchain needs to address two aspects:

Validation of transaction data: As mentioned

above, all participants in a blockchain network share a state of the overall system. System is expected to move from state S1 to state S2 if a transaction T1 is performed as a part of which entity E1 transfers asset A to entity E2. States S1 and S2 differ in terms of ownership of asset A. Validation, which would be done by various entities, would ascertain that transaction T is valid in terms of its format and asset A is indeed owned by entity E1. This can be considered as "endorsing" activity in case of Hyperledger Fabric or "transaction signing flow" in case of Corda.

Validation of block headers (transaction sequence): Transaction validation mentioned above works only if transactions are processed in the right sequence. So, in the above example, if system state was S1 and it performed transactions T1 and T2 in that sequence, system would reach states S2 and S3 respectively. However, if it performed transactions in order of T2 followed by T1, there is no guarantee that system will reach state S3. "Ordering service" in Hyperledger Fabric and "notary" in Corda are responsible for this check. Notary in Corda performs the task of avoiding double spends, which is basically ensuring the right transaction sequence and Corda documentation calls this as "uniqueness consensus".

For entities to agree on how transactions are sequenced, there are multiple classes of algorithms:

- Competition, rewards and proof of work/stake: One alternative is that entities solve a hash puzzle, which is computationally expensive to earn a right to create a block, which can be independently verified by other participants. This is used by bitcoin where nodes are given an incentive (mining rewards) to maintain integrity of blockchain. Proof of stake, proof of authority, etc. are some of the other alternatives.
- Crash Fault Tolerance (CFT): This can be used when multiple sequencing components are run in a local cluster e.g. Hyperledger Fabric orderers implemented using Kafka. This provides fault tolerance against node failures but not against malicious nodes.
- Byzantine fault tolerance (BFT): This can be plugged in cases like Hyperledger Fabric and Corda (notaries). This would consider work done by majority of orderers and guard against malicious nodes and comes with a performance penalty.

Transaction Security

Transaction Security ensures that only the lawful owner can access and transfer ownership to another account. It can be achieved using the following components:

Identification and Authentication: A permissioned blockchain network needs to provide a unique identification for each entity that participates in the blockchain. This is typically carried out by issuing a digital certificate to each entity, which can be used for authentication of the entity.

Authorization: Another security aspect is authorization of users to invoke specific functions for operating blockchain platform. Specific users could be 'authorized' to perform a subset of functions, e.g. Corda has ability to configure which RPC operations can be invoked by which RPC users.

Digital Signature: Blockchain architecture needs to have digital signing component. SHA-256 (or similar) hash of transaction data is computed using private key of an entity. This technique is used to prove that contents of a transaction are not tampered and bear a verifiable signature of the entity.

Cryptographic hash values: Blockchain uses hash values to prevent alterations of contents of a transaction or of transaction sequence. Multiple transactions are stored in a block and to ensure that block is not tampered, a hash value of the block is computed using hash values of individual transactions and hash of the previous block. This makes it computationally impossible to change the history of transaction data.

Storage Logic

This is concerned with maintaining the whole history of transaction data and protecting it from manipulation/forgery. This is achieved by making changes to data prohibitively expensive:

- Transaction data is protected against changes by computation of hash like SHA-256 and this hash is encrypted with private keys of entities (digitally signed).
- Blocks themselves bear hash value and this is computed typically using a Merkle tree formation with transactions which form leaf nodes of the tree.
- Overall blockchain is protected by linking the blocks by including the hash of the previous block in a newly added block.

Distributed Peer-to-Peer Architecture

Doorman service to onboard independent nodes (peers): Blockchain, especially permissioned blockchain requires implementation of doorman service which is used to onboard independent nodes i.e. peers. Each entity, which is a part of blockchain network may 'own' nodes and would be responsible for those nodes. There is a possibility that an entity owns a node and allows other entities to share that node and that would require clear, entity-wise segregation of data. Node would be used for validating transactions and processing these to maintain the overall state of the system.

Secure and resilient network: For a typical permissioned network, it is expected that mutual TLS (client and server certificates) is used to ensure communication is between known nodes (entities) only. The deployment should have redundancies to ensure that availability and response time SLAs are met.

Message passing and guaranteed delivery: Blockchain requires exchange of information between requestor node and other nodes for validation and processing of blocks. For ledgers to be correctly distributed, system should ensure that all nodes eventually receive all information. This eventual consistency requires implementation of queuing, either internal to each processing node or deployed in a centralized fashion. Hyperledger Fabric uses Kafka based queues whereas Corda uses Apache Artemis and these would require high available deployment of message broker with persistent, high available storage for messages.

Governance

Businesses are looking for ways to improve messaging systems to bring in value propositions for the customers, and Blockchain Technology offers a new paradigm that can help improve efficiencies towards seamless processing of transactions. For any such solution to be considered effective, it must involve multiple parties across the industry. Multi-party involvement is necessary to align incentives for participation, outline roles and responsibilities and orchestrate and support the blockchain network. A well-defined governance model is required to be put in place for successful implementation and effective functioning of blockchain network.

Governance in itself will be one of the critical factors required for widespread adoption of this disruptive technology across the industry.

The governance model must define how an industry-wide solution can be implemented and managed. It must also define the rules and procedures about membership, management of permissions, transaction legitimacy, data security, dispute resolution, version and infrastructure updates, adherence to regulatory guidelines, and protection against cyber risks across ecosystem.

Approach and Structure

A collaborative approach amongst all the stakeholders in the network is recommended. A codified set of rules need to be set up for smooth operations and collaboration. Participation from

different functional teams across operations, business, technology, information security and PMO of the participating members can ensure that governance rules are codified after taking into account different perspectives.

Different working groups may be set up to look at different facets of governance. The model needs to cover governance across different stages of the lifecycle is listed below:

- Evaluation Stage
- Membership Policy & Guidelines
- Implementation Stage
- Member Onboarding
- Data Structure & Risk Management
- Maintenance and BAU Operations
- Change Management & Version Control
- Use case Process Flow Standardization
- Documentation & Manuals
- Training across Participant Groups
- Support Desk & Query Management
- Regulatory and Third Party Audit

Key aspects to be considered while putting a governance structure in place:

Network Ownership

- The network must be a 'members owned' permissioned business network.
- All participants in the network would be required to adhere to SLAs to be listed in a Key /Master Agreement to be created for the network. To begin with, the contents of this agreement can be modelled on existing B2B agreements.

B2B Relationships

- All participating businesses must be part of the network
- Every business in the network must have the right to decide who they want to do business with. They will be able to select any set of partners from the network
- The network must have the ability to on board various businesses on an ongoing basis
- Once a participant is provisioned into the network, existing members can have option to select transaction partners and enter into transactions with them
- All business liabilities are between the two transacting parties and they have the complete 'ownership' of the transactions that happen between them.

Participation of Stakeholders through Working Committees

Involvement of key stakeholders from various businesses is key to evolving a sustainable working model. For this purpose, a sample structure is proposed below. The Governing Council comprising of Steering Committee, Business Management Committee and Working Committees may be constituted from participating businesses and industry stakeholders:

- **Steering Committee**, comprising of participating businesses is the highest decision-making body having representatives from product, process, technology and compliance and is responsible for maintaining governance standards of blockchain solution
- **Business Management Committee**, comprising of participating businesses is responsible for codification of business rules, functional use cases, on boarding, etc.
- **Working Committee** comprising of participating businesses –
  - Technology SPOC
  - Product SPOC from member businesses
  - Business Analyst
  - Process Analyst
  - PMO

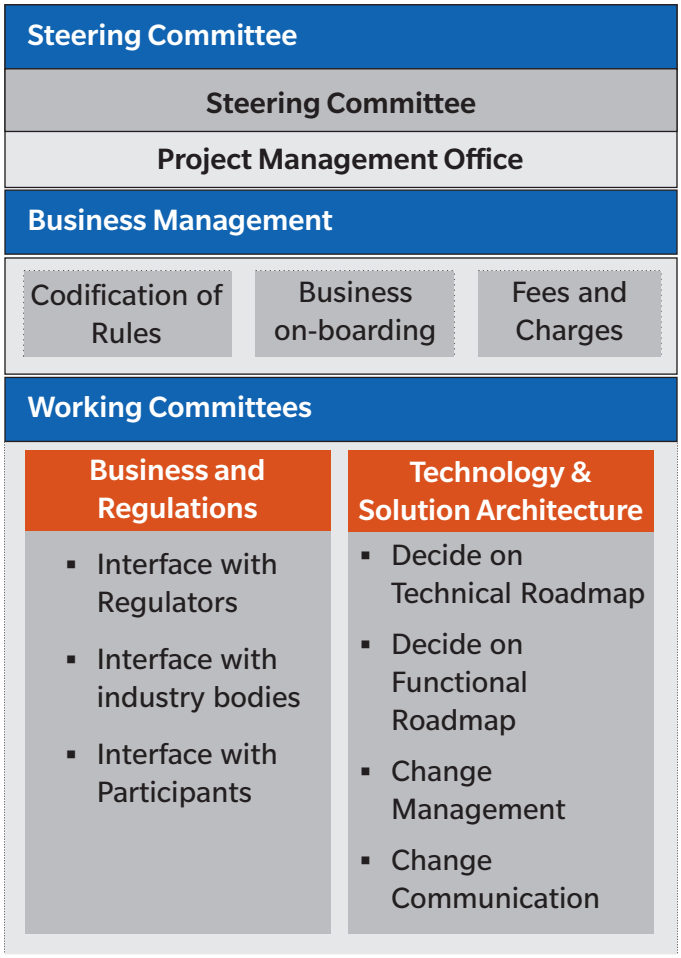
- Working Committee comprising of participating Infrastructure & Technology partner –
  - Business Analyst
  - Technology SPOC
  - Change Management SPOC
  - Account Manager for businesses
  - Delivery Manager
  - Key stakeholders from businesses

The Governing Council may meet at regular intervals to evaluate and approve critical decisions such as standards, book of work, systemic changes, risk controls, version consolidation/upgradation and adapting to changing regulatory, risk and compliance practices across the industry.

Dispute Resolution

- Dispute resolution may be bilateral in nature and primarily enforced based on the Key/Master agreement applicable for the network and will be done by the transacting parties.
- The software must also provide robust audit trail mechanism through which transactions can be recorded and tracked. This can serve as an input to any dispute resolution between transacting parties.

Sample Governance Structure



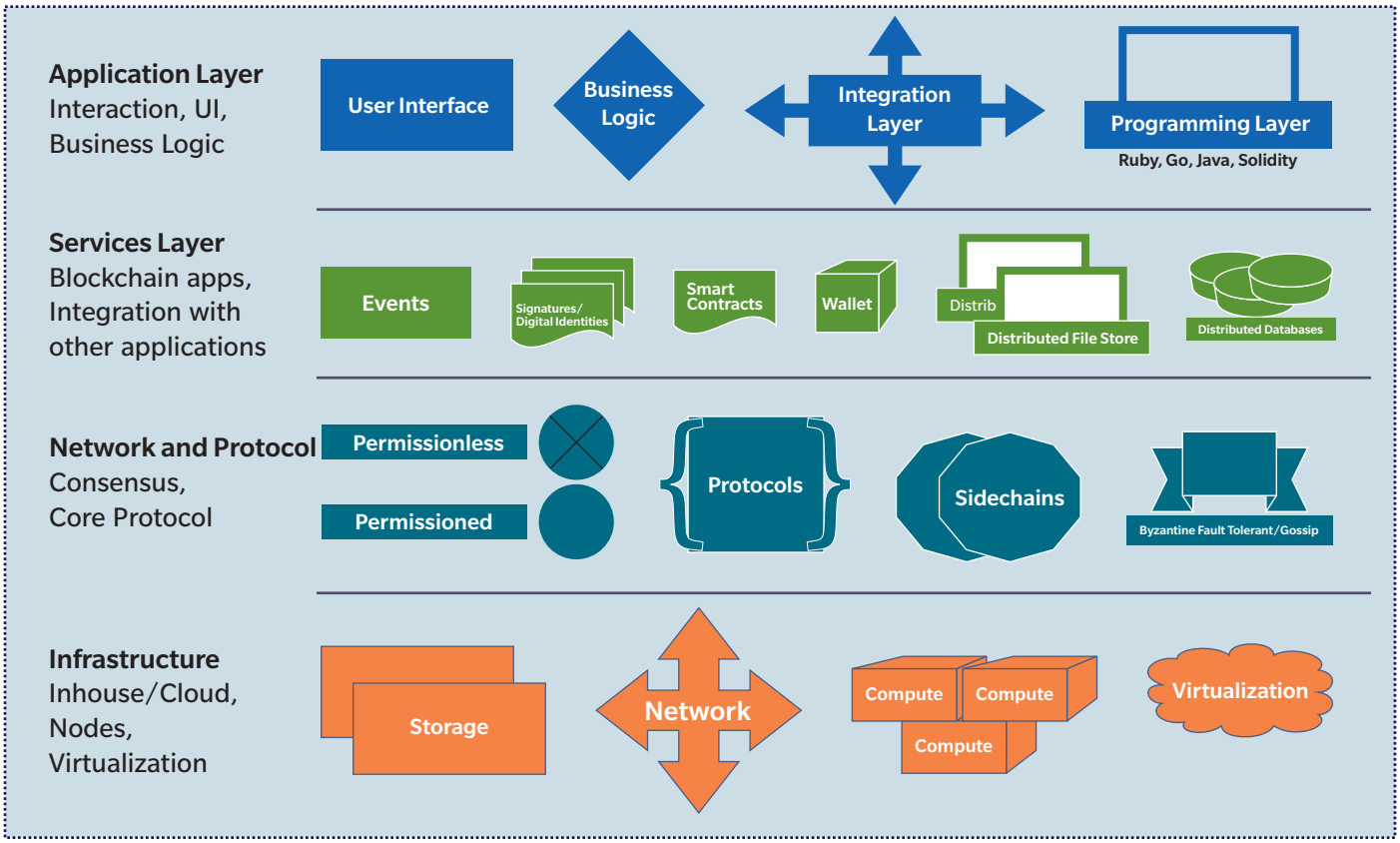
Key Takeaways

A codified set of rules need to be created with the consent of major stakeholders for smooth operations and collaboration. A collaborative approach involving multi-functional participation from all participants can help in quickly evolving standards and rules that are acceptable to all participants in the industry. Regulatory supervision can also be built in within the governance structure. These steps are expected to help drive increasing adoption of new technology among industry participants and continuous evaluation and change management will help in bringing in required stability and sustainability of the new framework.

Technology

THIS section provides a high-level diagram showcasing the components of the blockchain technology stack. Three alternate realizations are also discussed.

Blockchain Technology Stack



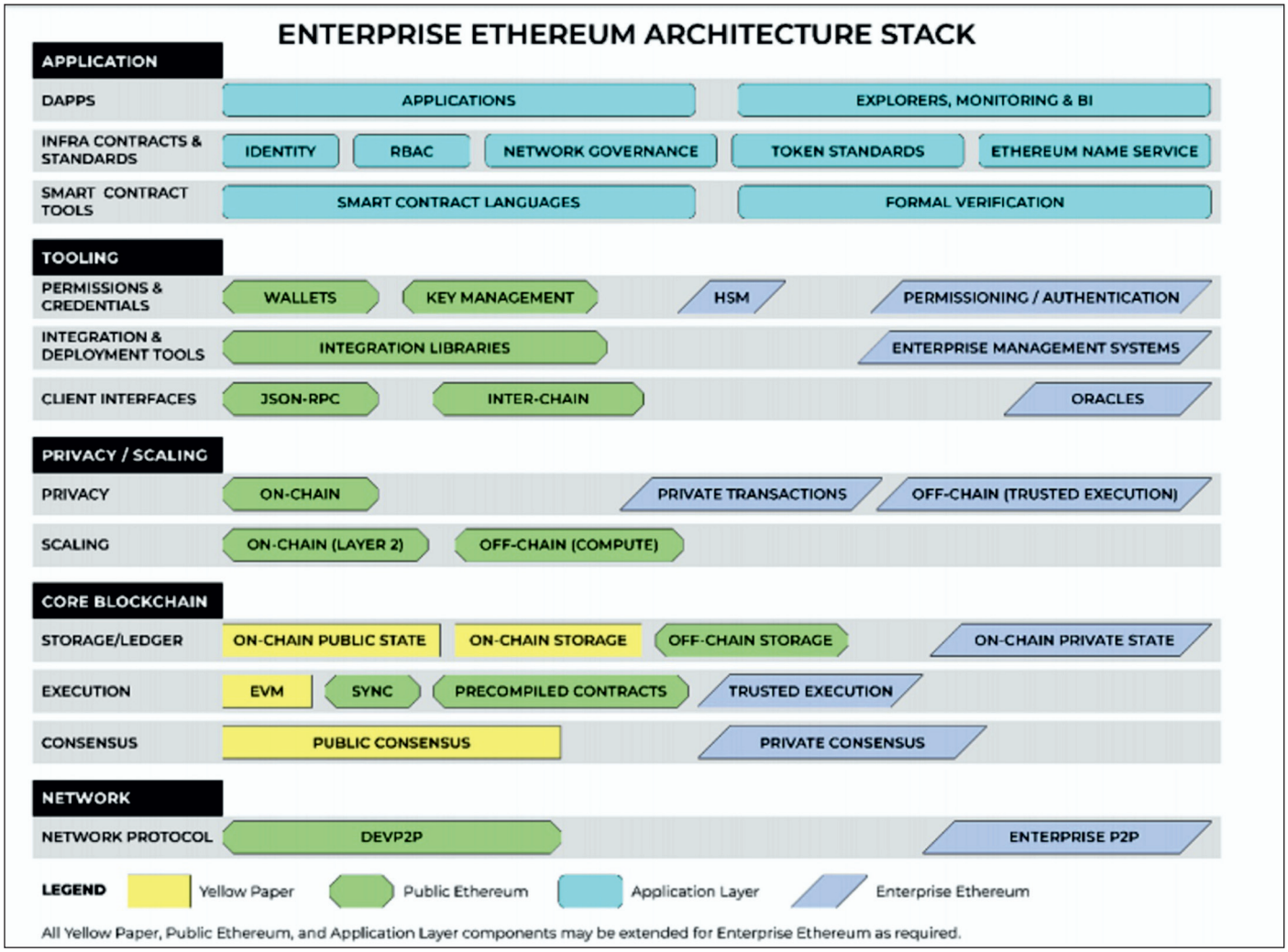
Blockchain Technology Stack covers:

- Modular Application Layer to enable blockchain-led application and application development environment
- Secure Application Services and Integration Layer to integrate with non-blockchain applications and environments
- Network and protocols for interoperability with permissioned and permissionless protocols
- Infrastructure to enable on-premise, virtualized, decentralized nodes.

Ethereum

Ethereum is an open blockchain platform that allows users to develop decentralized applications called DApps. A smart contract is a self-operating computer program that automatically executes when specific conditions are met. Because smart contracts run on the blockchain, they run exactly as programmed without any possibility of downtime, fraud or third-party interference.

The Enterprise Ethereum Architecture Stack provides details of the Core Blockchain platform for Ethereum (along with its components) and its tooling, application and privacy/scaling considerations.



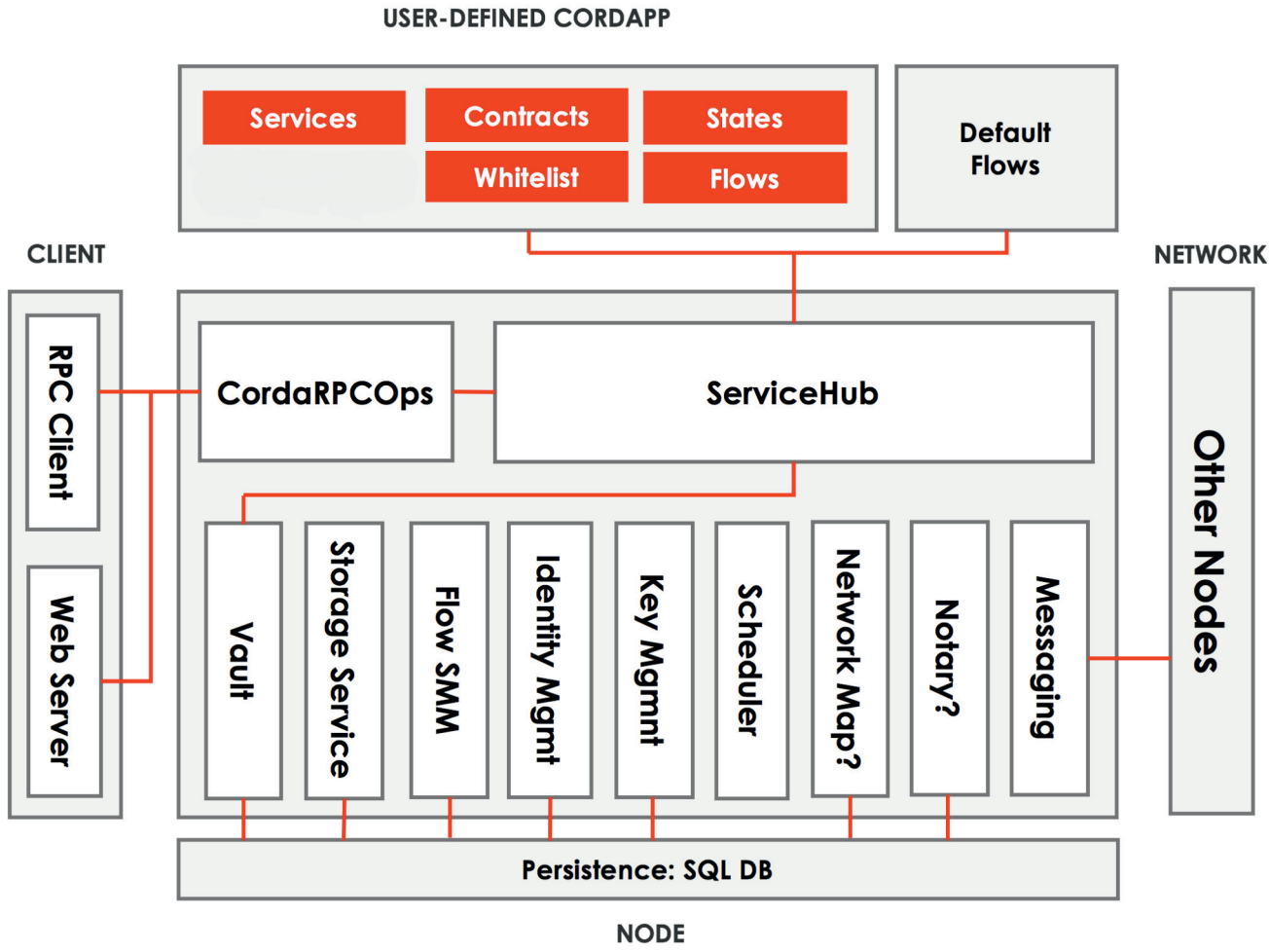
Source: "EEA Architecure Stack for Enterprise Ethereum Blockchain Platforms", https://entethalliance.org/wp-content/uploads/2018/11/EEA-Architecture-Stack-Spring-2018-Updated-1.pdf

Ethereum Core Blockchain Components:

- **Storage/Ledger** – This is used to store the actual blockchain
- **Execution** – Ethereum VM forms the Core Infrastructure for running the smart contracts
- **Consensus** – Enterprise Ethereum is an extensible platform that supports both public and private consensus mechanisms based on the deployment requirements.

Corda

Corda is a distributed ledger platform offered by R3. BFSI sector, especially banks have been at the forefront in early adoption of this technology. In the near future, one can envision a universal ledger with which all the participants will interact and which will allow any parties to record and manage agreements amongst themselves in a secure, consistent, reliable, private and authoritative manner.



Source: "Node Architecture",
https://docs.corda.net/head/key-concepts-node.html
https://docs.corda.net/head/_images/node-architecture.png

The core elements of the Corda architecture are:

- A persistence layer for storing data
- A network interface for interacting with other nodes
- An RPC interface for interacting with the node's owner
- A service hub for allowing the node's flows to call upon the node's other services
- A cordapp interface and provider for extending the node by installing CorDapps.

Persistence Layer

Corda offers developer community with the flexibility of exposing a contract to Object Relational Mapping (ORM) to a RDBMS. The ORM mapping is specified using the Java persistent API (JPI).

The persistence layer has got two components:

- The vault, where the node stores any relevant current and historic states
- The storage service, where it stores transactions, attachments and flow checkpoints.

Network Interface

A network interface component is responsible for all communication with other nodes on the network.

RPC Interface

The node's owner interacts with the node via remote procedure calls (RPC).

The Service Hub

Internally, the node has access to varied set of services that are used during transaction execution flow execution to manage ledger updates.

The CorDapp Provider

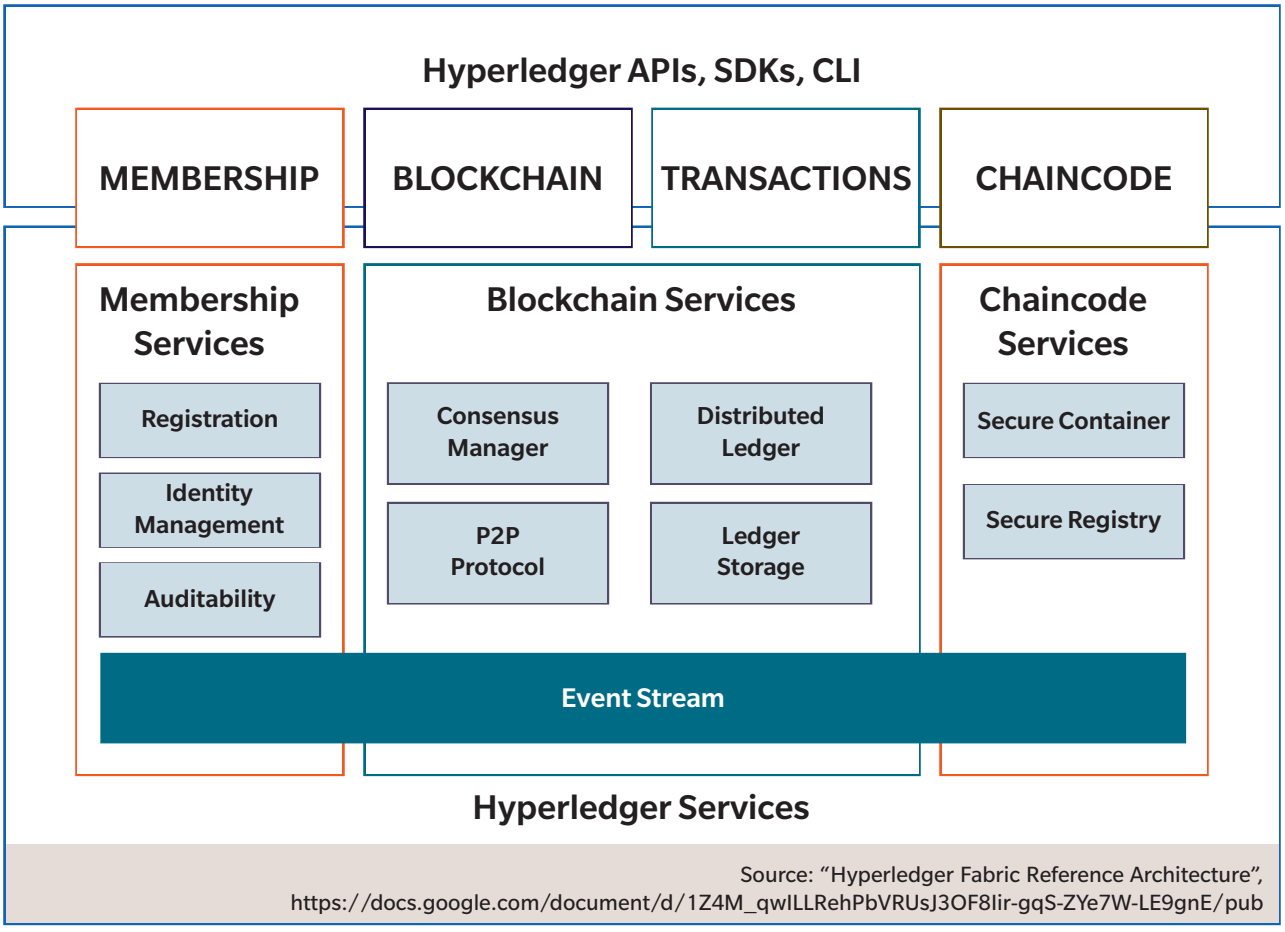
The CorDapp provider is where new CorDapps are installed to extend the behavior of the node.

Draining Mode

In order to mandate a clean shutdown of a node, it is important than no flows are in-process, meaning no checkpoints should be persisted.

Source: "Nodes", https://docs.corda.net/key-concepts-node.html

Hyperledger Fabric



Source: "Hyperledger Fabric Reference Architecture", https://docs.google.com/document/d/1Z4M_qwILLRehPbVRUsJ3OF8Iir-gqS-ZYe7W-LE9gnE/pub

Hyperledger Fabric is a platform for building distributed ledger solutions with a modular architecture that delivers a high degree of confidentiality, flexibility, resiliency, and scalability. This enables solutions developed with fabric to be adapted for any industry.

Hyperledger Fabric can be deployed (1) enabling fully disjoint network with separate endorser sets and ordering nodes to provide privacy and confidentiality, (2) restricting data replication only to permissioned parties and delivers the benefits of the blockchain for data integrity and non-repudiation of transactions, without compromising data security.

Fabric allows components, such as consensus and membership services, to be plug-and-play. It leverages container technology to host smart contracts called "chaincode" that contain the business rules of the system. And it is designed to support various pluggable components, and to accommodate the complexity that exists across the entire economy.

Fabric can also create channels, which enable a group of participants to create a separate ledger of transactions. This is especially important for networks where some participants might be competitors who don't want every transaction – such as a special price offered to some but not all – known to every participant in the network. If a group of participants form a channel, only those participants and no others have copies of the ledger for that channel.

Hyperledger Fabric Model

- **Assets** — Asset definitions enable the exchange of almost anything with monetary value over the network, from whole foods to antique cars to currency futures
- **Chaincode** — Chaincode execution is partitioned from transaction ordering, limiting the required levels of trust and verification across node types, and optimizing network scalability and performance
- **Ledger Features** — The immutable, shared ledger encodes the entire transaction history for each channel, and includes SQL-like query capability for efficient auditing and dispute resolution
- **Privacy** — Channels and private data collections enable private and confidential multi-lateral transactions that are usually required by competing businesses and regulated industries that exchange assets on a common network
- **Security & Membership Services** — Permissioned membership provides a trusted blockchain network, where participants know that all transactions can be detected and traced by authorized regulators and auditors
- **Consensus** — A unique approach to consensus enables the flexibility and scalability needed for the enterprise.

Source: "Hyperledger Fabric Model", https://hyperledger-fabric.readthedocs.io/en/release-1.3/fabric_model.html

References:

- https://medium.com/edchain/a-comparison-between-5-major-blockchain-protocols-b8a6a46f8b1f
- https://docs.corda.net/key-concepts.html
- https://www.corda.net/content/corda-platform-whitepaper.pdf
- https://hyperledger-fabric.readthedocs.io/en/release-1.3/arch-deep-dive.html
- https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf
- https://hyperledger-fabric.readthedocs.io/en/release-1.3/blockchain.html

IDRBT
Explore, Enable, Excel

• CHAPTER - III

REALIZING THE BLUEPRINT

The first part of Chapter II discussed how blockchain architecture can be used to develop superior business communication network that can address the gaps in the current networks. This was described in a generic manner, independent of the underlying implementation. The next part discussed the governance structure with key principles of open access to the network to all businesses and at the same time giving them the right to decide with whom they do business. Governance structure also foresees multi-industry participation in the form of corporations and other stakeholders. The final part of the Chapter discussed technology stacks i.e. popular implementations of the architecture described in the first part of Chapter II. This Chapter describes the practical considerations in adopting one of these technology stacks for business network. It also studies implications of the choice of a particular technology stack on the governance model.

Layered Approach

Key principles for realizing the blueprint are:

- Blockchain is a superior business communication network, which can address some of the gaps in the current systems. Setting up a blockchain network for one entity will not be a differentiator and needs large number of businesses to participate and simplify communication related to their transactions
- Interoperability vs. Common Infrastructure: Currently, interoperability between different blockchain platforms is not well-established. One option to realize such a business network is for different group of businesses to setup sub-networks with their respective platforms, standards and technology. These sub-networks may communicate with each using a potential future algorithm for interoperability between blockchain networks. However, such an approach cannot create a true ecosystems effect. Stakeholders like corporates have to hook to different blockchain networks depending on which party they are transacting with. Small institutions may not have deep pockets to invest in multiple platforms. Common infrastructure and technology is a far superior approach as it generates synergy across industry
- Future business communication networks developed using blockchain should be able to allow businesses to create innovative products and create a niche on top of common standardized infrastructure. This is also in alignment with governance principle of open access but leaving the creative freedom with each institution.

The diagram below depicts the blueprint for implementation of business blockchain using a layered architecture that is aligned to technology and governance principles described in the earlier chapter. Such a network may also have wider participation.

Specialized Services
- Specialized distributed applications
- Interoperable with basic services
- Space for Industry specific Innovation

Basic Services
- Basic Information Sharing/Digital Notary Service

Infrastructure Layer
- DLT Software
- Member Services (identity, on boarding)
- Integration/Smart Contract Standards
- Governance

The role of different stakeholders in each layer is discussed below:

Infrastructure Layer: This layer corresponds to the rails of the new business communication infrastructure using blockchain. This layer includes the technology components like DLT software, member services, technology standards for data communication and smart contracts. This layer also includes key governance functions like access provision, change management and future roadmap. To achieve maximum benefits from the technology, this layer has to be common across the industry. There is a significant role that can be played by large enterprises, regulators, trusted intermediaries and research organizations to design, build and operate such improved rails for business communication. Such an infrastructure may be built from scratch or one of existing infrastructure may be enhanced with blockchain capabilities.

Basic Services: These business applications are built on top of the infrastructure layer and are needed/available to all members. For example, regulatory information/notice sharing across members in a non-repudiation manner are basic services of such business communication network. Similarly, sharing of information related to cyber threats, or fraud, which can benefit all the members through speedy communication. Non-core services like KYC, digital notary are also potential candidates for basic services. Basic services developed on top of infrastructure layer provides a mechanism for all members to reap benefits from new and superior mode of communication among them.

These services may be launched by regulators, trusted intermediaries, large enterprises or third parties and can be subscribed by all the members.

Specialized Services: This layer is the space for innovation for members. Using the underlying infrastructure and basic services, members can forge new partnerships to offer innovative services. When developing new services, care must be taken to allow members to select the type of transaction and the entities participating in transactions. Improved processes for trade finance or supply chain finance across a few set of banks is a good example for this category of applications.

Such a layered realization of blueprint helps wider participation and cooperation among businesses to build the necessary infrastructure. Businesses can innovate on new services without worrying about interoperability, technology choices and onboarding/off-boarding of counter-parties. This architecture provides a mechanism for businesses to operate in a peer-to-peer fashion among select few in a broad network setup and operated by consensus.

Finally, there is a need to create an industry specific business value framework for analyzing suitability of business applications to be migrated to blockchain based business networks. Such a framework helps in prioritizing projects with high business value and also filter out the less impactful ones. Business case preparation for blockchain projects is complex due to the fact that both costs and benefits are shared across multiple parties and may not be evenly distributed among them. Cost of building and maintaining the ecosystem of large number of players has to be considered in the business case. Similarly, benefits have to be articulated in terms of efficiency improvements, reduction of fraud, reduction of manpower, reduction of capital and increased business due to digitization.

Path to Deployment

Business networks could vary in their compositions. They could include direct participants from banks and indirect participants like corporate customers or small to medium enterprises. Besides financial institutions, they could also include logistics providers such as transport and insurance companies as indirect participants. High on the agenda of these participants would be the privacy related considerations e.g. when buyer A places a purchase order with seller B, they may not want this transaction to be visible to everyone on the network (certainly not the details).

Besides, various participants may want to write smart contracts and it would be advantages if the platform allows them to leverage development and deployment skills available within the enterprise. If the technology stack requires a central function, it would be worth considering as to who would perform that function. Considerations for usage of different technology stacks with respect to (a) privacy, (b) fault tolerance, (c) reuse of skills, and (d) deployment of central functions are described below. These would help consortiums in taking the required decisions.

Corda based implementations

Corda is more of a distributed ledger platform than a conventional blockchain platform. This platform is created by a consortium of banks and is oriented towards implementations within financial industry.

Privacy: Transactions are visible only to concerned entities. If party X transfers asset A to party Y through transaction T1, party Z will not be aware of it. However, if party Y transfers the same asset to party Z, information on transaction T1 will need to be provided to Z to prove as to how Y acquired the asset. This can go up several levels.

Fault tolerance: It would be possible, though non-trivial to reconstruct database for a failing node by getting transactions from all other nodes. A simpler alternative would be to plan for business continuity of distributed ledger database like any other on-premise database.

Central function: Participants also need to take a decision as to who plays the role of a notary that provides uniqueness consensus. This could be performed either by blockchain infrastructure provider or a regulatory body.

Reuse of skills: "Flows" which are used to post a transaction to enable movement of assets from one entity to another, "states" which represent the assets and "smart contracts" which are pure Java functions, typically used to implement policies for checking the required signatures are all written in Java. States are stored in a conventional relationship database like PostgreSQL. This allows reuse of skills within IT organization for development and deployment.

Hyperledger Fabric based implementations

Hyperledger Fabric is a permissioned blockchain platform, enhanced to support privacy requirements.

Privacy: Hyperledger Fabric supports channels that are like virtual private networks. Summary information for all transactions is stored in the blockchain shared with all the participants in a channel, whereas private data is shared to limited number of parties through private data stores

(version 1.2 onwards). This comes fairly close to what R3/Corda provides.

Fault tolerance: Failed node can rejoin a channel it belongs to and recover all its data. Theoretically, it should be possible, for a failed node to contact all other nodes to get private data on transactions done with them. Alternatively, highly available deployment of CouchDB can be used – not yet well-documented and needs to be substantiated with tests.

Central function: "Orderer" decides the sequence in which transactions are added to blocks and is a central function. This could be performed either by blockchain infrastructure provider or a regulatory body.

Reuse of skills: Hyperledger Fabric smart contracts are written in "GO" language and database used for storing blockchain data is "CouchDB" (or "LevelDB"). It will require some training efforts for acquiring these skills. Using cloud based deployment would however simplify this further.

Quorum/Enterprise Ethereum based implementations

Ethereum is a general purpose blockchain platform. Quorum and Enterprise Ethereum projects are expected to enhance capabilities towards setting up a permissioned network with enhanced capabilities for "privacy" and "scalability". Quorum release is available while Enterprise Ethereum is a work in progress.

Privacy: Quorum allows storing of both public and private transactions on blockchain. State database at each node stores both public and private states. A component called "Enclave" (Intel SGX hardware trusted execution module) is used for encryption and decryption. Transaction hashes are stored within public states. This is conceptually similar to Hyperledger Fabric.

Fault tolerance: Failing node should be able to resync with network to rebuild the blockchain (public transactions). To retrieve data for private transactions, it should be theoretically possible to contact all other nodes to retrieve the same. It may also be possible to explore other options like storage level replication.

Central function: There is no need for a central function in case of Ethereum.

Reuse of skills: Smart contracts are written using "Solidity" and this will require significant training efforts for adopting the new technology. Database used could be "LevelDB" or "RocksDB".

Other Implementation Considerations

Onboarding and role assignment: Irrespective of the features offered by the underlying technology stack, services for onboarding entities to the business network will need to be built since this will have unique application related considerations. In case of a trade finance network, there would be considerations like type of entity being onboarded, whether that entity owns a blockchain node, etc. For pan enterprise process implementations using blockchain, roles that an entity can play is an important attribute of an entity.

Digitization of assets: To represent movement of physical assets on blockchain, first these assets need to be digitized. This may involve time-consuming, elaborate processes, e.g., it will require legal changes for bill of exchange or bill of lading on the blockchain to be considered as legally valid. In another context, digitization of an asset like say 'piece of land' on blockchain would first require unique way to identify that land.

Interoperability: Entities that are connecting to the business network could have their internal blockchain networks as well. Protocol interoperability between internal blockchain network and external one is an ongoing research topic. A practical approach would be to complete validation on the external blockchain network before adding a transaction to internal blockchain. This would reduce the probability that transaction is added to internal blockchain but cannot be added to external blockchain. If transaction on external network still fails, there would be a need to post compensating transaction onto internal network, in an assured manner.

From a governance point of view, decision of technology stack for external blockchain network can be independent of the choice of technology stack for internal blockchain network. For example, decision for say external blockchain network for remittances could be different than that for internal blockchain network for trade finance. However, using a single technology stack for both would result in significant cost savings.

Cloud Deployment: Deployment of blockchain components on cloud, especially if the cloud provider has presence in local geography is an appealing option. This would significantly reduce capital investment when a new network is being set up. Security compliance of the cloud provider (e.g. ISO 27000 compliance) need to be assessed on an ongoing basis. All data during transit and at rest need to be encrypted using database or storage level encryption. This would require a security audit from a competent third party.

Regulatory Compliance: If financial assets like fiat currencies are being maintained within blockchain ledger, permission from regulatory body is required. One possibility here would be to consider blockchain ledger only to process transactions while settlements continue through existing mechanisms. This would require an interface between blockchain ledger and core banking ledger. Consortium back office account within core banking books would then represent what entity owes to the consortium and vice-versa.

BLOCKCHAIN WORKING GROUP

Mentor

DR. A. S. RAMASASTRI
DIRECTOR, IDRBT

Members

- **Rishabh Gupta**
  State Bank of India
- **Ajjo John**
  ICICI Bank
- **Ajay Lande**
  Axis Bank
- **Shyamalesh Choudhury**
  Axis Bank
- **Yogesh Sawant**
  Yes Bank Ltd.
- **Pramod Kamath**
  Infosys
- **Vishwas Patil**
  IIT Bombay
- **Aayush Garg**
  ICICI Bank
- **Pramod Nair**
  HDFC Bank
- **Rajesh Nair**
  Axis Bank
- **Rishikesh Nagwekar**
  Yes Bank Ltd.
- **Deepak Hoshing**
  Ex-Infosys
- **Raghava Suresh**
  Tata Consultancy Services
- **N. V. Narendra Kumar**
  IDRBT

Published by:

Explore, Enable, Excel

Institute for Development and Research in Banking Technology

(Established by Reserve Bank of India)

Castle Hills, Road No. 1, Masab Tank, Hyderabad - 500 057, India.

EPABX: +91 - 40 - 2329 4999, Fax: +91 - 40 - 23535157

Web: www.idrbt.ac.in E-mail: publisher@idrbt.ac.in