The XDC Protocol

TECHNICAL WHITEPAPER

By

XinFin

XinFin Organization

www.xinfin.org

Blockchain technology for global trade and finance

September 2017

Contact us: info@xinfin.org

Table of Contents

List of Figures:

List of Tables:

1. Blockchain Paradigm

History

Over the past few years, an important innovation colloquially known as the “Blockchain” has emerged as a potentially disruptive technology. The core of the innovation is built around the concept of a distributed cryptographic database. The database, also referred to as the ledger, is maintained by a network of computers.

The ledger makes it possible for the entire network to create, evolve and keep track of an immutable record of transactions. The most successful blockchain application thus far has been Satoshi Nakamoto’s cryptocurrency known as Bitcoin, which he outlined in his seminal paper, “Bitcoin: A Peer-to-Peer Electronic Cash System,” in 2008. This powerful technology has so far been implicated in the numerous cryptocurrencies that exist online. Financial institutions are only beginning to understand the potential applications of blockchain in conventionally regulated industries.

Understanding the Blockchain

At a fundamental level, the blockchain is a data structure that cryptographically links blocks of transactions or any potential transfers of value. As a paradigm, the blockchain allows for true privacy to exist between those involved in a transaction. This characteristic of the blockchain distinguishes it as a significant innovation in the financial industry. Its structure allows for automation, immutability, and decentralization. These characteristics were carefully chosen by Nakamoto in order to create a digital mechanism of trust.

For the financial world, this paradigm reduces reliance on third party trust mechanisms, enabling a direct contractual interface between two parties involved in any transfer of value. This technology has the power make such exchanges safer, faster, and cost-effective. Ownership of value stored in the blockchain is established through asymmetrical cryptography. Digital keys, wallet addresses, and digital signatures are all created cryptographically to ensure total privacy in transmission of transactional data.

Every transaction on the blockchain needs to be 'signed' using digital keys. Whoever owns these keys owns access to the value stored in the wallet. All keys generated by the wallet software come in pairs; a private key, which is kept secret, and a public key. The public key is akin to a bank account number, and the private key is like the secret pin used to control the account. The idea behind digital signatures is that a private and public key pair shares a mathematical relationship, whereby a message (transaction) signed by a private key can be verified by a public key — without revealing the private key.

Public Blockchain

The Bitcoin's blockchain network is structured as a peer-to-peer network architecture. In this network implementation, all nodes are equal or symmetric. There is no server, no centralized service, and no hierarchy in the Bitcoin network. Blockchain implementations for the financial industry, however, cannot allow for such an open and symmetric network structure. Bitcoin enthusiasts advocate, therefore, that the blockchain and Bitcoin are fundamentally linked. This is a short-sighted view, taking into consideration that the Bitcoin technology stack exists to solve the problem of an unregulated digital currency. The same stack can be evolved to solve similar transfers of value over an asymmetric network.

The development of the Ethereum blockchain was a major milestone in the blockchain ecosystem. The focus shifted onto system state and virtual machines that could run smart contracts across the network. With the availability of Turing complete smart contracts, the blockchain ecosystem overcame the limited functionality of the Bitcoin scripting language, and a number of different applications became possible.

Private Blockchain

Private blockchains came into prominence with the Hyperledger Project, which saw participation from IBM and Intel, in 2016. Another major private blockchain developed by R3, called Corda, also raised significant amounts of money. Since then, R3 has also joined the Hyperledger Project. Fully private blockchains make the case that centralized and trusted implementation of the core data structure of the blockchain, along with significant changes to the consensus mechanism, are meaningful value propositions for enterprise applications.

Hybrid Blockchain

Hybrid blockchains are fairly unexplored and only few implementations exist; even in development. Quorum developed by JP Morgan is designed to be the hybrid blockchain in a fully permissioned environment. Truly hybrid blockchains must necessarily be able to connect a public blockchain with a private blockchain implementation, running in a fully permissioned environment. The XDC hybrid blockchain aims to be exactly that, leveraging the power of both the public and private blockchain paradigms.

Table 1 illustrates the comparison between public, private and hybrid blockchain systems:

Parameters	**Public Blockchain**	**Private Blockchain**	**Hybrid Blockchain**
Transactional Visibility	Fully visible	Not visible	Both; subject to use case
Auditability	Low	Low to High	High
Network	Decentralized	Centralized	Hybrid
Security	Compromised	High	High
Throughput	Low (3-4 s^{-1} Bitcoin; 15 s^{-1} Ethereum)	Very high	Very High
Membership	Open to all	Restricted	Participation Open, Hosting Restricted
Interoperability	No	No	Yes

Table 1: Comparison of Blockchain Platforms

2. The XDC Blockchain

The public blockchain implementations, including Ethereum, serve an important purpose but are woefully inadequate for financial and other use cases. There are entire industries that would benefit greatly from blockchain technology but cannot implement an Ethereum based solution.

2.1 Why Not Vanilla Ethereum

All industries have stringent data security needs. A fully public blockchain that stores transaction data, even in encrypted form, can be compromised, as in the case of the Bitcoin blockchain. In addition to data security, institutions also have auditability requirements. This balance between data security and data auditability is an important consideration and the XDC protocol is equipped to satisfy both.

Data visibility: The nature of public blockchain is such that transactional data is visible to all participants in the network. This means almost anybody can track any transaction. While public key cryptography theoretically provides some level of anonymity, entire industries exist to de-anonymize the Bitcoin and Ethereum blockchain. This is wholly unacceptable for sensitive financial data and makes the public blockchain paradigm unsuitable for enterprise applications.

Data auditability: Data visibility concerns go hand in hand with auditability requirements. Different institutions, and different levels of participants within an institution, require ways of easily accessing data. Requisite participants' inability to access data, without compromising any benefits of public key cryptography, makes the application of public blockchains like Ethereum for financial use cases impossible.

The XDC blockchain runs its full and reference nodes on the infrastructure of its consortium members. The blockchain contents are not available on any public network such that it's possible to brute force the relationships between transactions and associated public keys. Consortium members will undergo stringent vetting and will be able to run 'gerent' nodes, which have the entire blockchain stored on their infrastructure. These nodes will also be 'bootnodes' in the Ethereum paradigm. The different kinds of nodes set to perform different functions are described in section 2.6.

2.2 The XDC Blockchain Network

The network architecture is a very important aspect of a distributed database. While plenty has been written about Public and Private blockchain architectures, consortium blockchains have been somewhat ignored. This is unfortunate because consortium blockchain architectures tread an important middle ground between the low trust ecosystem of public blockchains and the high trust member entities of private blockchains.

Fully public blockchains like Bitcoin and Ethereum rely on a certain number of long-term stable nodes. Consortium blockchains are similarly structured but with more selective inclusion clauses and centralization is a not an inherently undesirable quality in blockchain architectures. Enterprises depend heavily on centralization and it may be hard to make a blanket case for decentralization across regulated fiduciary institutions.

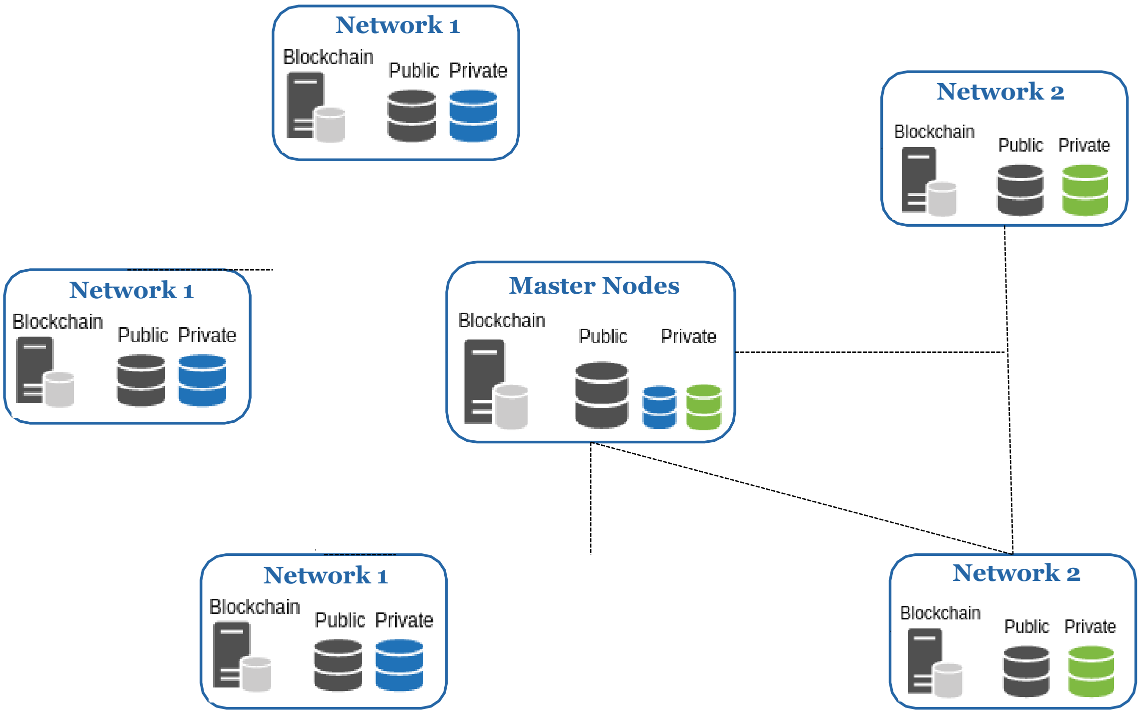


Figure 1: The XDC Network

2.3 Hybrid Network Architecture

2.3.1 Hybrid Network: Public *and* Private

The XDC blockchain is built upon the paradigm of consortium blockchains. The architecture differs from conventional private/permissioned blockchains as well as public blockchains. Built upon the Ethereum codebase, the XDC blockchain deals with the system state rather than blocks of transactions. There are two different kinds of networks that can exist within the XDC ecosystem. First exists the public network that all constituents are part of, and second exists a private/permissioned network that restricts participation. The private network state is maintained in its respective network, however, a record (hash) of transactions and smart contracts is stored on the public state of the blockchain. As depicted in Figure 1, various institutions will have different relationships with other participants. The public state of the XDC blockchain is shared by all participating nodes that are owned by different kinds of constituents. Groups of nodes can further form fully permissioned networks with their own private state that is only accessible to authorized members. For instance, let us assume that a private marketplace for goods and services is set up in Network 1. The specifics of the trade between parties are not accessible to Network 2. However, the record of individual trades is stored as hashes on the public state, which is shared, so that even in the private network there is an immutable record of transactions.

2.3.2 Hybrid Network: The XDC and Public Blockchains

The previous section outlines how private and public states on the XDC blockchain can co-exist and provide complimentary benefits. The 'hybrid' nature of the XDC blockchain extends to interoperability with public blockchains like Ethereum and Bitcoin. Transactions that are marked as hybrid on the XDC blockchain can be transmitted to and executed on the Ethereum public blockchain without the need of external wallets or exchanges. The XDC protocol seeks to create a truly decentralized cryptocurrency space through interoperability

2.3.3 Consortium Membership

Consortium membership here refers to the relationship that different institutions and individuals can have with the XDC blockchain.

The XDC blockchain has three kinds of membership. The first tier is the most accessible. If an individual or institution owns XDC tokens, they are part of the Tier 1 membership by default. These tokens can be bought by interested individuals or institutions through the planned crowdsale. Tier 2 and Tier 3 memberships are both obtained by holding a certain amount of XDC, subject to requisite vetting. These tiers allow institutions to host XDC nodes and participate in the consensus mechanism of the XDC blockchain.

2.3.4 Fork Prevention

Private/permissioned blockchains are usually very resistant to possible forks to their blockchain. The XDC blockchain is no exception. Only Tier 2 and Tier 3 consortium members are allowed to run the XDC nodes. There are asset forfeiture rules written into the protocol that seize the XDC holdings of unscrupulous consortium members looking to undermine the integrity of the XDC blockchain. Furthermore, each block is signed by the node that creates the block. Fraudulent additions to the XDC blockchain are, therefore, very easily prevented.

The asset forfeiture and close vigilance on additions to the XDC blockchain mimic the legal responsibilities that financial institutions have in the real world. Public blockchains have no mechanisms of exacting personal cost from ill-intentioned participants, which introduces a lawlessness that is not conducive for enterprise applications.

2.4 The XDC Blockchain Architecture

This section outlines the individual components of the XDC protocol. In the following sections, we have outlined the different components of the protocol that make up different type of nodes on the XDC blockchain. Certain nodes will require fewer components from the protocol than others, depending on the functionality deemed necessary.

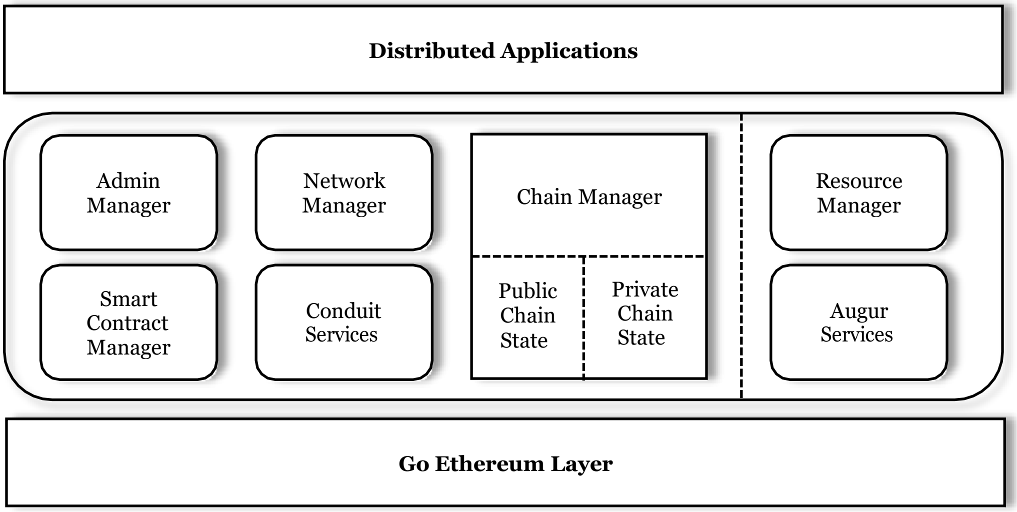
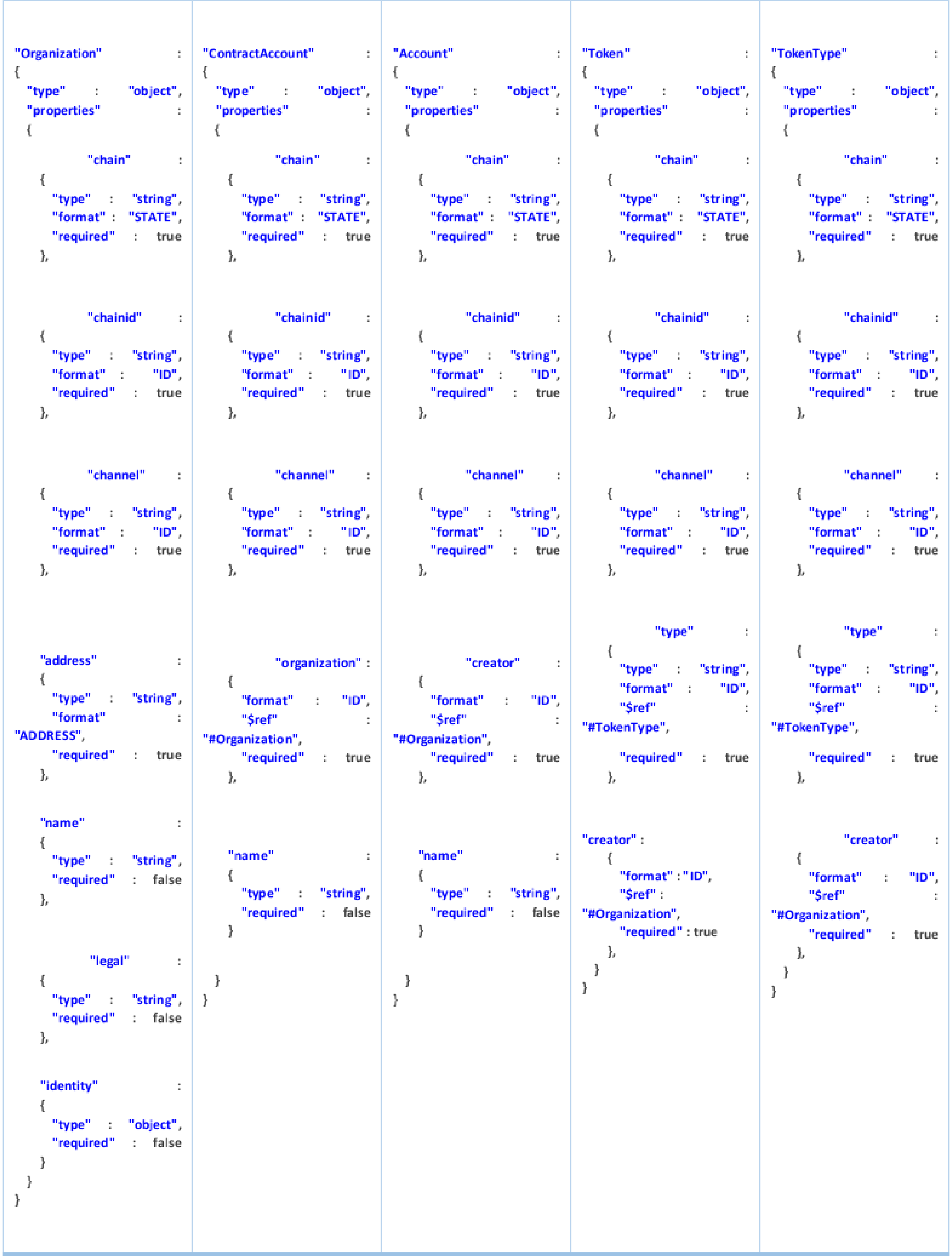


Figure 2: The XDC Blockchain architecture

2.4.1 Administration Manager

The Administration Manager lives in the heart of the XDC protocol. The activity of the Administration Manager can be divided conceptually into three sections. It is the location for the definition of the Data Model, Messaging, and carries out State manipulation. The Administration Manager is equipped with a Java script CLI that can be used to define Data Models, create Messaging frames, and alter the State of the system. The Administration Manager, in summary, allows for the definition of Data models, their initialization, instance creation, creating messaging frames, and the lifecycle of all State changes.



"Organization" :
{
"type" : "object",
"properties" :
{
"chain" :
{
"type" : "string",
"format" : "STATE",
"required" : true
},
"chainid" :
{
"type" : "string",
"format" : "ID",
"required" : true
},
"channel" :
{
"type" : "string",
"format" : "ID",
"required" : true
},
"address" :
{
"type" : "string",
"format" :
"ADDRESS",
"required" : true
},
"name" :
{
"type" : "string",
"required" : false
},
"legal" :
{
"type" : "string",
"required" : false
},
"identity" :
{
"type" : "object",
"required" : false
}
}
}
	"ContractAccount" :
	{
	"type" : "object",
	"properties" :
	{
	"chain" :
	{
	"type" : "string",
	"format" : "STATE",
	"required" : true
	},
	"chainid" :
	{
	"type" : "string",
	"format" : "ID",
	"required" : true
	},
	"channel" :
	{
	"type" : "string",
	"format" : "ID",
	"required" : true
	},
	"organization" :
	{
	"format" : "ID",
	"$ref" :
	"#Organization",
	"required" : true
	},
	"name" :
	{
	"type" : "string",
	"required" : false
	}
	}
	}
		"Account" :
		{
		"type" : "object",
		"properties" :
		{
		"chain" :
		{
		"type" : "string",
		"format" : "STATE",
		"required" : true
		},
		"chainid" :
		{
		"type" : "string",
		"format" : "ID",
		"required" : true
		},
		"channel" :
		{
		"type" : "string",
		"format" : "ID",
		"required" : true
		},
		"creator" :
		{
		"format" : "ID",
		"$ref" :
		"#Organization",
		"required" : true
		},
		"name" :
		{
		"type" : "string",
		"required" : false
		}
		}
		}
			"Token" :
			{
			"type" : "object",
			"properties" :
			{
			"chain" :
			{
			"type" : "string",
			"format" : "STATE",
			"required" : true
			},
			"chainid" :
			{
			"type" : "string",
			"format" : "ID",
			"required" : true
			},
			"channel" :
			{
			"type" : "string",
			"format" : "ID",
			"required" : true
			},
			"type" :
			{
			"type" : "string",
			"format" : "ID",
			"$ref" :
			"#TokenType",
			"required" : true
			},
			"creator" :
			{
			"format" : "ID",
			"$ref" :
			"#Organization",
			"required" : true
			},
			}
			}
				"TokenType" :
				{
				"type" : "object",
				"properties" :
				{
				"chain" :
				{
				"type" : "string",
				"format" : "STATE",
				"required" : true
				},
				"chainid" :
				{
				"type" : "string",
				"format" : "ID",
				"required" : true
				},
				"channel" :
				{
				"type" : "string",
				"format" : "ID",
				"required" : true
				},
				"type" :
				{
				"type" : "string",
				"format" : "ID",
				"$ref" :
				"#TokenType",
				"required" : true
				},
				"creator" :
				{
				"format" : "ID",
				"$ref" :
				"#Organization",
				"required" : true
				},
				}
				}

Table 2: Data Model

2.4.1.1 Data Model

The Data Model is the part of the Administration Manager, where it defines the structure of the data to be stored on the blockchain. The public state of blockchain has a predefined data model that is stored in here. The private state of the blockchain can inherit this extant data model, make changes to it, or create an entirely new structure according to the use case.

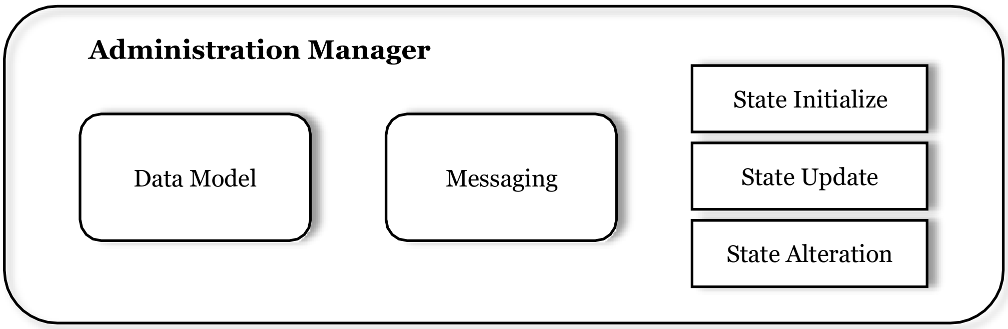


Figure 3: Administration Manager

2.4.1.2 Messaging

The Messaging section within the Administration Manager interfaces with the Conduit service to transfer the different kinds of messages across the constituent parts of the protocol. Using the JS CLI, the default messaging structure can also be altered for private state usecases.

2.4.1.3 State Transitions

The Administration Manager carries out state transitions across the private and public states of the blockchain. The Conduit Services and the Network Manager connect the Administration Manager to different parts of the protocol, local storage, and the XDC ecosystem in general, in order to define, initiate, and execute different classes of transactions.

2.4.2 Smart Contract Manager

The Smart Contract Manager executes the appropriate contracts at the right time. It is deeply connected with the Resource Manager and Augur Services in order to ensure that the conditions stipulated in the contracts are met. Highly audited smart contracts are available to be inherited by the public state participants with private networks having the ability to make further customizations according to the needs of the use case.

2.4.3 Message Conduit

The Message Conduit connects the various parts of the protocol with each other. It connects the Augur Service and the Resource Manager with the Smart Contract Manager to ensure external conditions critical to the execution of smart contracts are securely transmitted and are executed correctly. It also connects the Network Manager with the Administration Manager to carry out a number of state modification tasks including transactions.

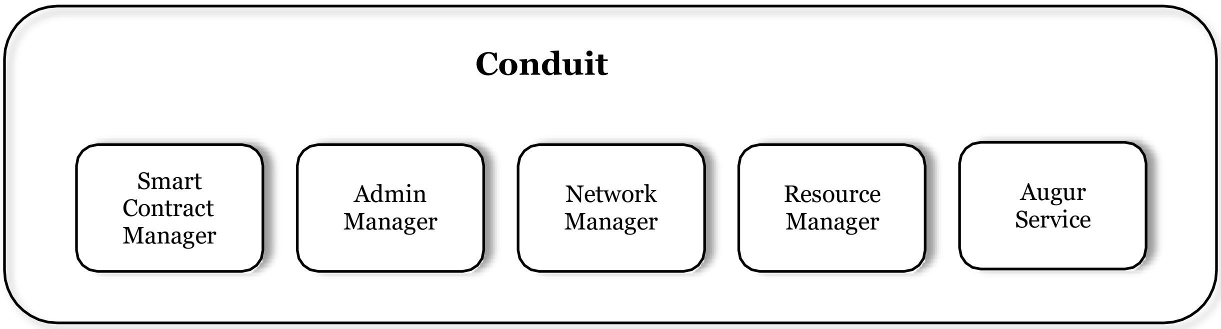


Figure 4: Conduit Services

2.4.4 Network Manager

The Network Manager connects the various nodes of the XDC ecosystem. It is responsible for creating a permissioned subnetwork that has its own private state and is responsible for connecting to other nodes to modify the public state.

2.4.5 Chain Manager

The Chain Manager is implicated in the consensus mechanism that resolves BFT (Byzantine Fault Tolerance) issues that can arise during state transitions. The Chain Manager maintains the private state for permissioned sub-networks in local storage and takes part in the voting mechanism to maintain the public state.

2.4.6 Resource Manager

The Resource Manager is the component of the protocol that connects to the equipment or resource monitoring ecosystem that is part of the fiduciary arrangements. For instance, an IoT based resource monitoring ecosystem might be set up to make sure equipment purchased through a loan is being utilized within the terms of the agreement. It is the Resource Manager's job to connect to IoT like devices that are tied to corresponding resources. The XDC protocol comes with inbuilt support for Arduino, Raspberry Pi, Intel Edison, and the ESP2866. More IoT device support is planned for as well.

2.4.7 Augur Service

The Augur Service communicates with the Smart Contract Manager as a highly secure oracle service. Smart contracts that depend on external variables, such as market conditions, get their data from the Augur Service. The Augur Service will contain highly vetted and controlled connections to the pertinent external sources. For smart contracts or transactions that deal solely with the private state of a permissioned network, oracle data from the Augur Services can be augmented with mutually agreed upon sources.

2.5. Forking Quorum

The XDC blockchain is built upon Quorum, a private/permissioned blockchain developed by J.P. Morgan. Quorum has been developed as a layer upon the Go implementation of the Ethereum protocol. There are few but significant changes made to the protocol. The consensus mechanism has been entirely reworked, replacing proof of work with a consensus mechanism called QuorumChain. This new consensus mechanism allows for new blocks be created in a two-step process. In the first step, the transactions to be included in the new block are voted upon by all participating nodes. In the second step, one node is selected as the leader or block maker randomly. The block maker node then creates the new block.

The XDC blockchain is forked from Quorum. There are a number of reasons behind this decision. Firstly, the powerful smart contract functionality that exists in the Ethereum protocol is easily accessible through Quorum. Secondly, the consensus mechanism is implemented as a smart contract in QuorumChain. Additional changes to this method of achieving consensus are easy to implement. Thirdly, the hybrid nature of the Quorum blockchain is ideal for a large number of enterprise use-cases. Fourthly, the fairly high throughput compared to public blockchains is essential for any scaling needs for high volume businesses. Finally, the ability to reuse the substantial development dedicated to the Ethereum protocol makes the choice of Quorum as our base implementation very appropriate.

In addition to the above, our fork includes a number of improvements to the Quorum protocol. The throughput of transactions is significantly increased in our test environment. We have developed a smart contract manager that allows for interoperability between the XDC blockchain and public blockchains. We have added punitive smart contracts that connect to the QuorumChain consensus smart contracts to ensure those who stake XDC tokens to run network infrastructure remain honest. We are also in the process of developing a light client built for the Quorum protocol that would connect natively with the XDC ecosystem.

2.6 The XDC Nodes

The XDC network hosts four different kinds of nodes. Many nodes don't require the functionality of or have the resources to run full nodes. Nodes that only write to the private state of a sub-network don't require all components of the XDC protocol. Furthermore, the ability to run full nodes will also be restricted to institutions that meet the criteria for it.

The consensus architecture for the XDC protocol is divided into two parts. The first is the XDC consortium membership requirements. In order to participate in the XDC network, institutions must belong to one of the three tiers. The tiers, in part, correspond to the XDC token holdings. Any fraudulent activity results in seizure of the XDC tokens that are staked to gain membership and host network infrastructure.

The second is the actual process of achieving consensus. Once the network topology is stable, nodes come together to vote on transactions that can be put on a new block. Then a new leader is selected from among the validator nodes at random. This leader creates a new block with the transactions that have been voted on by the other full nodes.

Described below are four different type of nodes that exist on the XDC network and their purpose.

2.6.1 Full Node

Full nodes will possess the entire XDC protocol. Certain full nodes will possess even the transaction history of the entire network in addition to the state. These nodes will be controlled by consortium members and come with a number of caveats. Full nodes will have to purchase (and hold) a fixed equity of the XDC to be able to host the full XDC protocol. One major advantage to this design choice is that no full node or groups of full nodes can gain control over the network. In order to possess more than 51% of the public network, full nodes must acquire more and more XDC. This drives up the price of the XDC and ensures that it is financially impossible for a full node cartel to control a majority of the network.

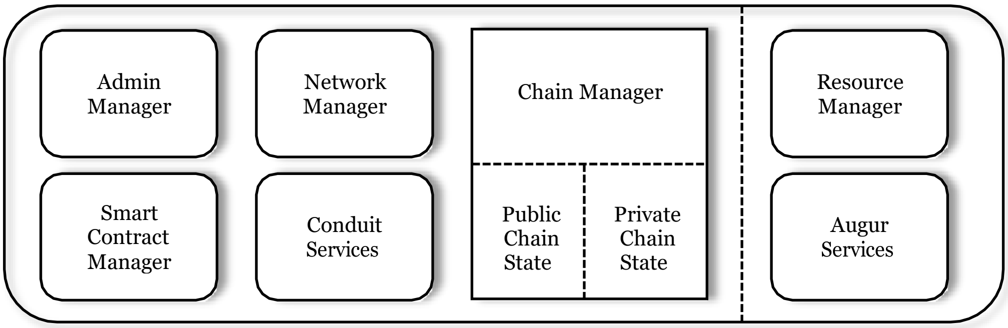


Figure 5: Full Node

2.6.2 Reference Node

The second kind of nodes are *reference nodes* which exist to transfer transactions to the transaction pool. These nodes do not have access to the entire blockchain and play only a minor verification role in that they ensure the structure of the transaction is according to protocol. The transactions are verified and added to the blockchain by the full nodes.

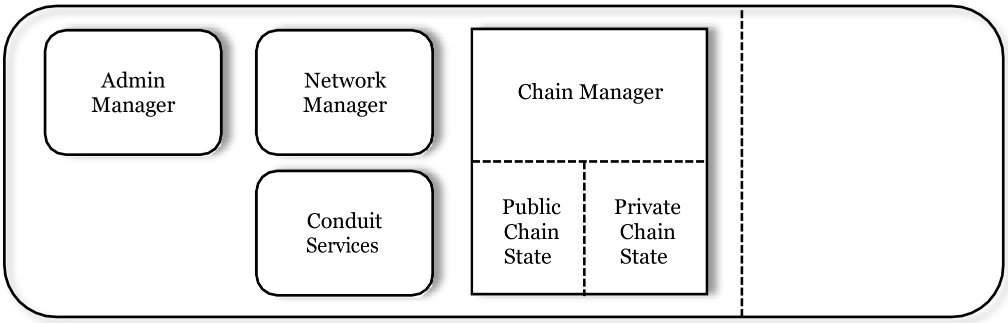


Figure 6: Reference Node

2.6.3 Light Node

The third kind of node is a *light node*. A light node is implemented in the private networks that don’t participate in achieving consensus through the voting process. They store the relevant private state in their local storage and can access the public state and the transaction history from full nodes.

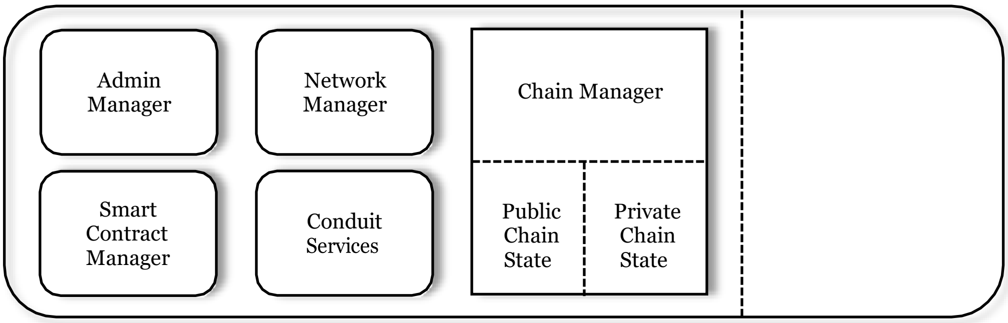


Figure 7: Light Node

2.6.4 Auditing Node

The fourth kind of node is an *auditing node*. This node has other parts of the protocol, besides the Chain Manager, but they are very limited in their functionality. The auditing node allows for the access to the public and private state of the blockchain for the purposes of regulation, auditing, or reconciliation with legacy systems.

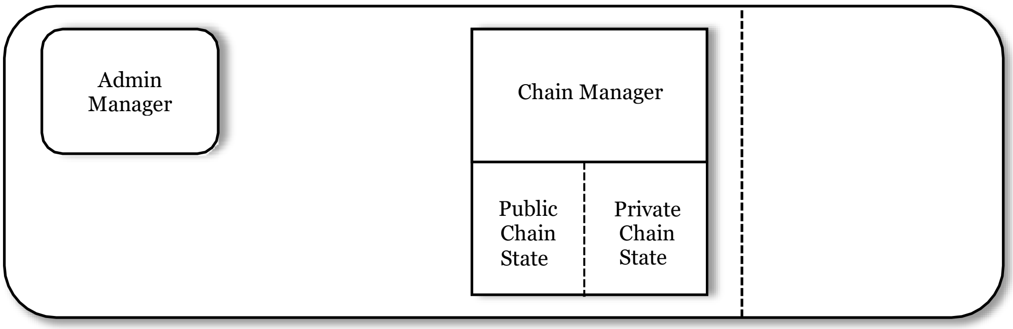


Figure 8: Auditing Node

2.7 Design Considerations

2.7.1 Tokens

The XDC token is built upon the ERC20 token standard. This design decision was taken to ensure a fundamental compatibility with the multitude of emerging Ethereum Dapps. With a view at future interoperability with the Ethereum blockchain, the choice of using the ERC20 standard was straightforward. This compatibility extends to smart contracts written for the Ethereum blockchain.

2.7.2 Centralized Exchanges

While we all claim to be proud participants in a movement for decentralization, we are tied to centralized exchanges. True decentralization will evade the blockchain ecosystem until there is meaningful interoperability. Our design choice was also taken with this perspective in mind. We envision a future where all kinds of tokens can be exchanged and smart contracts are not limited to the architecture of individual organizations.

2.7.3 Wallet Security

Online wallets often seem to be a convenient way of storing cryptocurrency tokens but could actually be very insecure. Certain implementations of online wallets are simply web applications with access to wallets running on a full node. It is extremely insecure to share private and public keys, something that is inevitable with most online wallet implementations. An accidental or planned shutting down of the node hosting the wallet is an added concern. In such a case, all tokens are lost or stolen. These incidents are not only financially damaging but also hurt the general reputation of the blockchain ecosystem.

The XDC ecosystem will provide easily accessible light wallets that will connect securely to full nodes. Each light wallet will have unique keys that are used to sign transactions from their associated account. Since the light wallet is a standalone application that serves only one account, the possibility of being hacked or losing the XDC tokens becomes highly unlikely.

2.7.4 Token Standard

One major problem with using the ECR20 standard in designing custom tokens is that transfers from Ethereum wallets will result in a loss of Ether. This is a design challenge that a lot of custom coins face. We at XinFin are going to be using specialized functions that call on the token's contract rather than sending them from a wallet, preventing a loss of tokens.

2.7.5 The XDC01 Standard

We plan to release a token standard of our own that will ensure users on the XDC blockchain cannot accidentally send their tokens to an incompatible ERC20 based blockchain. We plan to build the XDC01 standard on top of the ERC20 standard to ensure future compatibility when interoperability allows for such kinds of transfer.

3. A Real-World Use Case

There are 10 farmers in Ghana that want to procure a certain kind of harvester from an Indian manufacturer. Without delving too far into the financial relationships that need to be facilitated, the acquisition of the equipment, functioning, and repayment of the loan can be carried out over the XDC protocol.

Actors involved: Party A (the farmers), Party B (the equipment manufacturers), Party C (various institutions involved)

Party A accesses the marketplace where service provider, Party B, advertises its wares. This communication can happen on a Dapp marketplace or can be initiated through a more conventional method.

1. Dapp connects Party A, Party B, and Party C. After terms are agreed upon, smart contracts are written, initialized, and implemented through the Smart Contract Manager.
2. Dapp sends the transaction data frame to the Administration Manager.
3. The Administration Manager invokes the appropriate smart contracts through the Smart Contract Manager.
4. The Smart Contract Manager initializes a monitoring service via the Resource Manager and the Augur Service. The XDC that is being used to the purchase the equipment is transferred to Party B. In the case that a loan is required for the purchase, the required amount of XDC is transferred from Party C (the financiers) to Party B as depicted in Figure 9.

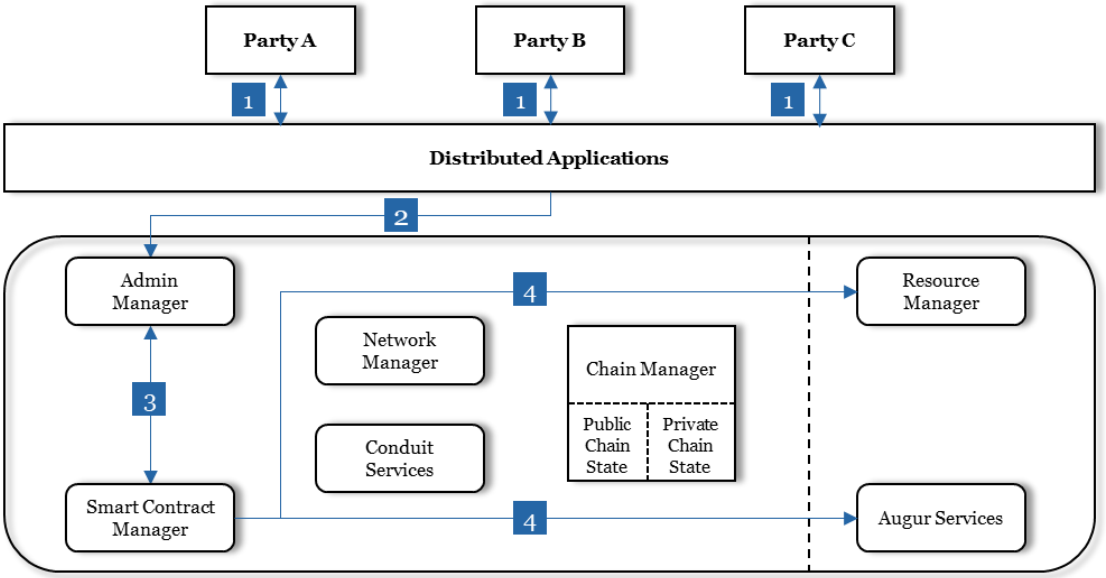


Figure 9: Steps 1 to 4

5. The state of the blockchain is updated through the Administration Manager by sending the transaction details through the Conduit Service to the Chain Manager and the Network Manager. If necessary, the Network Manager indicates to the remaining

network that the public state needs to be updated, or the Chain Manager updates the private state and updates the public state through the Network Manager.

6. The Network Manager broadcasts the pertinent information to the relevant actors across the network. An Escrow smart contract is initiated by the Smart Contract Manager and a percentage of XDC is stored in its account in order to underwrite the investment to protect against non-payment or term violations as depicted in Figure 10.

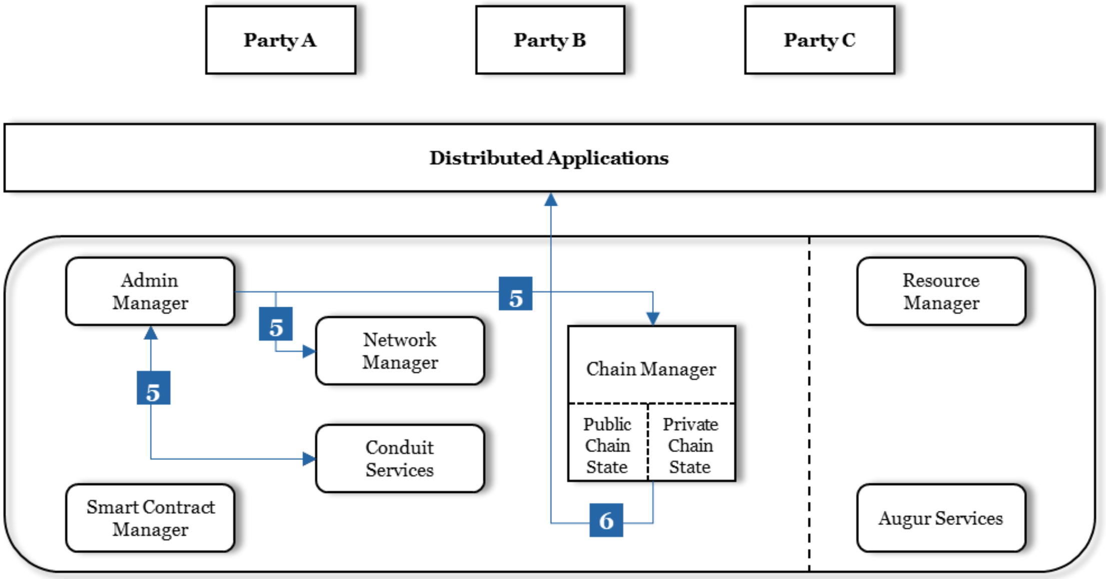


Figure 10: Steps 5 and 6

7. The Smart Contract Manager sends the escrow details to the Administration Manager.

8. The Administration Manager updates the private state, broadcasts the public state update through the Network Manager and to the actors involved in the private state transaction. A further service is started in the Resource Manager that monitors the physical location and activity of the equipment purchased. In the face of terms violation, preventive smart contracts, in this case the Escrow smart contract, initiate payout to the financiers according to the terms of the agreement as depicted in Figure 11.

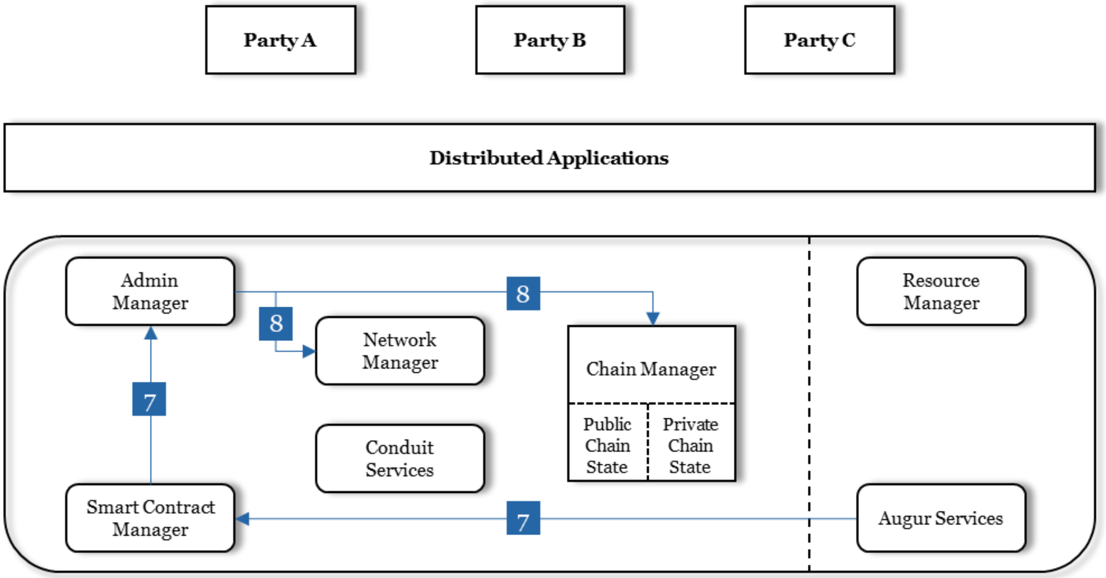


Figure 11: Steps 7 and 8

9. The Resource Manager updates the Smart Contract Manager with conditions being fulfilled or violated, which in turn updates the Administration Manager.
10. The value generated by the equipment is recorded on the XDC blockchain.
11. The loan amount is repaid and the Smart Contract terminates after all the payouts are distributed amongst the constituents as depicted in Figure 12.

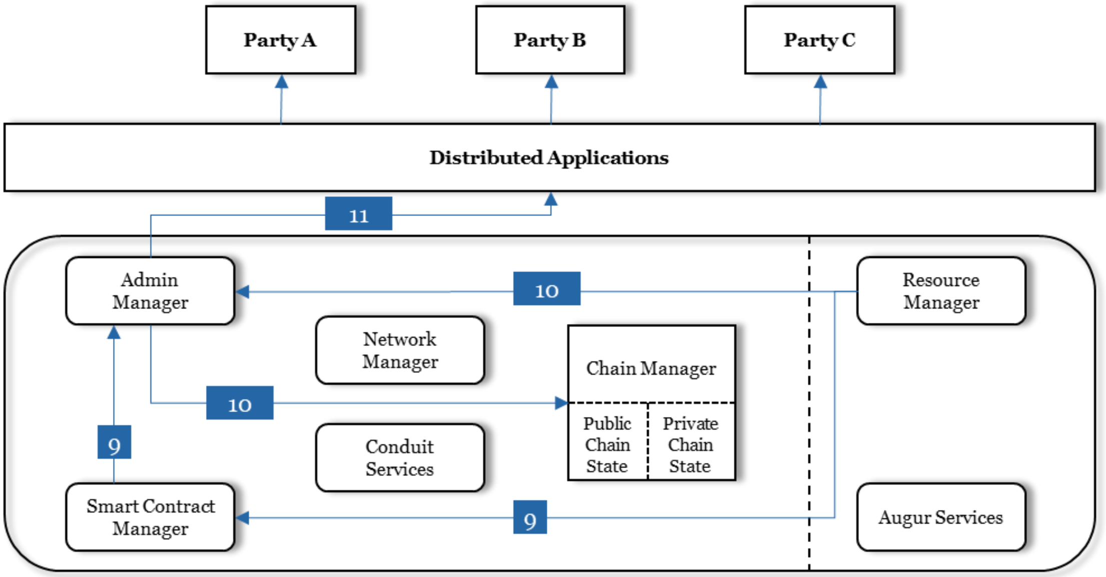


Figure 12: Steps 9 to 11

4. Blockchain Applications

4.1 Asset Valuation and Financing

Generally, asset valuation is a tedious process for financial institutions. Certain assets, like real estate, are easier to value than assets that depreciate in value, like airplanes. Instead of holding collateral to finance or refinance assets, it might make sense to constantly assess the asset in question itself. Unfortunately, because of opaque and outdated processes, asset valuation becomes impossible for financial institutions. The XDC blockchain will allow for asset valuation from procurement to financing at a later part of its lifecycle. Monitoring services will constantly evaluate the 'health' of an asset and make that information quickly available to relevant authorities.

Let us assume that an airline has financed a number of airplanes as part of its fleet. The asset valuation – in this case, of the airplane – will be recorded on the XDC blockchain. Standard formulas for value depreciation resulting from the use of the airplane, the age of the asset, repairs and maintenance, or traditional refinancing, can be coded in smart contracts to monitor the value of assets in a transparent and standardized format.

When the airline needs to trim part of the fleet, a cryptographically signed record of its history is available readily and in a standardized format. The aforementioned opacity no longer prevents thorough asset valuation.

4.2 Resource Monitoring

Section 3 outlines the procurement and monitoring of assets in a trade finance context. The innovation supported by the XDC protocol is the ability to finance and operate assets on value ascribed to the asset itself. In the example, farmers were able to procure farming equipment that was financed against the value that they would produce. In order to calculate the value produced and record on the XDC blockchain, mechanisms for resource monitoring must be put into place. IoT enabled resource monitors that connect securely to the XDC protocol allow for such monitoring.

4.3 Smart Contracts

Smart contracts are the truly innovative invention in the blockchain paradigm. We plan to build a large community of developers and enthusiasts who will design and make innovative smart contracts. We at XinFin are nonetheless cautiously optimistic about smart contracts. The DAO hack is undeniable proof that even with the Ethereum protocol having robust security, smart contracts can be compromised without the right kind of oversight.

4.4 Security, Highly Audited Standardized Contracts

Our solution to this problem is straightforward. Since the XDC ecosystem belongs to the consortium blockchain paradigm, we plan to allow only comprehensively audited smart contracts. This will ensure not just the security of the XDC ecosystem, but also create a standardization that has secondary benefits in a number of fiduciary use cases.

4.5 ‘Marketplace of Smart Contracts’

As the XDC blockchain gains maturity, we envision a regulated marketplace with a well-defined process of security auditability. Smart contract writing firms will undertake creation of standardized contracts for general financial use cases but will also cater to more customized applications. Entirely separate auditing firms will ensure the infallibility of these smart contracts. Our aim is to ensure hacks like the DAO don't happen on the XDC blockchain.

5. Conclusion

Enterprise application requirements are difficult to fulfill under normal circumstances. Implementing new technologies therefore comes with significant risk. Industry vetting of the popular public blockchains has not been favorable. The visibility of transactions, the rigidity of development paradigms (Bitcoin block size debate), the lack of permissioned infrastructure, and the energy intensive consensus are some of the gaps that make enterprise applications built on public blockchains difficult for real world applications.

The Bitcoin protocol brought with it grand visions of a decentralized and secure world to facilitate transactions. The cryptocurrency ecosystem that has since developed has ironically resulted in monolithic silos with no interoperability. Centralized exchanges are necessary to carry out conversions between cryptocurrency pairs that are susceptible to exactly the same kinds of security problems that the Bitcoin protocol set out to solve.

The XDC blockchain attempts to solve this problem of secluded ecosystems with highly secure and robust interoperability powered by the protocol described in this paper. Interoperability here refers to both within the XDC protocol and outside of it. Within the XDC ecosystem, public and private states interact with each other in specific ways. Outside of the XDC ecosystem, interoperability with public blockchains allows for the exchange of XDC tokens from within the protocol itself.

The XDC blockchain aims to bring together the significant advantages of public blockchains and the necessary restrictions of permissioned blockchains to create an ecosystem that is enterprise ready. The tiered membership rules for participation in the XDC network allows for robust mechanisms in maintaining network integrity. In addition, controlling network participation creates an environment of standardized relationships that allow for the focus to be on what is important: business.

References:

1. Nakamoto, Satoshi. Bitcoin: A Peer-to-Peer Electronic Cash System: Cryptography Mailing List, 2008.
2. Swan, Melanie. Blockchain: Blueprint For a New Economy: Sebastopol, O'Reilly, 2015.
3. Copeland, Christopher and Zhong, Hongxia. Tangaroa: a Byzantine Fault Tolerant Raft: Stanford, 2015.
4. Buterin, Vitalik. Ethereum: Platform Review, Opportunities and Challenges for Private and Consortium Blockchains. 2016.
5. Voell, David and Nielsen, Patrick: Quorum whitepaper v0.1: https://github.com/jpmorganchase, 2016