


Kalima

Blockchain & IoT

Collect.Protect.Monetize

TECHNICAL WP

Summary

I. Abstract

The purpose of this document is to define Kalima blockchain and highlight its characteristics and market positioning. This paper should also serve as a resource for technical blockchain developers and managers interested in using its architecture and set of tools available. While we appreciate that there may be discrete definitions for the terms “blockchain” and “Distributed Ledger Technology (DLT),” for the purposes of this paper we will treat both terms synonymously and use the term “blockchain” throughout.

Visit our website: **www.kalima.io**

This document was created with the goal of positioning Kalima Blockchain within the current blockchain community landscape and highlighting its specificities.

While blockchains may appear similar to distributed databases, they are typically implemented without a central authority and central repository. Therefore, blockchains provide some unique differences from everything that has come before. A blockchain survives faults and attacks by using redundant checking at multiple nodes. This resiliency goes far beyond replication, since it happens across the network without any "central coordinator".

This paper describes industry use cases that drive the principles behind a new blockchain platform, a distributed ledger dedicated to Enterprise et capable to manage Industrial Internet of Things and new mobile devices. It outlines the basic requirements and high-level architecture based on those use cases.

The principal requirements of Industrial Internet of Things use cases are:

- data transport integrity
- data storage immutability
- the data must be stored in a convincing way
- security
- fast response time
- verified identities
- private and confidential transactions
- low cost of transactions and low total cost of ownership
- scalability through interoperability

The first central element which emerges from these requirements is that every member/participant in a given trust ecosystem needs to maintain its own ledger system which, all together, build a distributed ledger made up of mutually distrusting nodes synchronized with each other. This distributed and resilient ledger records the state of all transactions, deals and obligations between people, machines and services.

The second crucial element is that such kind of distributed architecture must be events driven to prevent inefficient latencies and costly pooling type mechanism.

Moreover, it must be capable of automatically launch actions (through smart contracts) on events receptions.

The last characteristic is that the core elements of such system is the transaction itself and that no transaction should be lost, modified or added during the transport and storage. Hence the

embedded Blockchain from Edge to the operator in mobility that ensure end-to-end the integrity and immutability of these transactions.

Response time and very low latency while scaling the systems are critical factors that Kalima blockchain addresses through its architecture.

In some other networked systems like web tiers, adding nodes serves to divide work among more resources and increase performance. This is often the reverse in blockchains, where more nodes increase the resilience of the system in terms of integrity and availability, at some expense to performance generally.

Even this simple notion can be complicated when we look across the breadth of blockchains.

Some blockchains specialize the roles of nodes. In those systems a “node” may no longer imply a unique participant (such as a company) with a share of the resiliency burden. To help precisely and consistently evaluate the unique performance attributes of Kalima blockchains, this paper will refer to many relevant terms and metrics, and discusses some complex issues mentioned in other white paper quoted in appendix to this document.

II. Background

Blockchain is a peer-to-peer distributed ledger technology that first gained traction in the financial industry because of its capacity to issue, trade, manage, and service assets efficiently and securely. As the shared ledger concept gains traction in the business world, blockchain smart contracts are also getting a lot of attention from industry [Eth].

Bitcoin and other cryptocurrencies based Blockchain were designed to be completely open, decentralized, and permissionless: anyone can participate without establishing an identity; one only has to contribute by spending computation cycles. Under the Bitcoin model of blockchain, there is no central authority that controls admission; these networks have been called permissionless. Bitcoin is costly to operate because it requires innumerable proof-of-work computations [N09].

Ethereum added functionalities compared to Bitcoin by enabling a function that Is today a must have for blockchains, smart contracts. This new feature turned out to be a key moment in Blockchain history enabling Ethereum to expand functionalities of a blockchain from a cryptocurrency to a platform for decentralized applications.

After Ethereum, Tezos was designed from the ground up as a completely new blockchain that was different from Bitcoin or Ethereum design. A common goal at this time was to improve Ethereum transaction throughput, and the solution found was to use a Proof of Stake (PoS) consensus instead of a Proof of Work (PoW). Tezos was one of the first to implement an efficient delegated proof of stake consensus (DPoS) mechanism, to solve this issue. Ethereum Is still today transitioning to a PoS consensus.

This last few years blockchain interoperability has become a major challenge that has not fully be Implemented yet. Cosmos, Polkadot have emerged as leading examples for blockchain interoperability the inter Blockchain Communication protocol of Cosmos and the Polkadot protocol ease inter-blockchain communication. Nevertheless blockchain interoperability is still in tis development phase and does not take into account physical data from the real world.

These permissionless Blockchain like Bitcoin, Ethereum, Comos, Polkadot... have never been designed to meet the requirements of the Industrial Internet Of Things.

On the other end Permissioned Blockchain as Hyperledger Consortium consider that existing blockchain implementations have fallen short of meeting the multitude of requirements inherent in the complex world of business transactions. To meet the varied demands of the modern marketplace, Hyperledger projects have been designed for a broad array of industry-focused use cases, thereby extending the work of the pioneers in the field by addressing the existing shortcomings. [HPL02]

Kalima Blockchain is a network of permissioned blockchain, something that most cryptocurrencies do not directly support. With no need for Proof Of Work in a permissioned network, Kalima Blockchain instead offers a unique delegate Proof Of Stake consensus mechanisms that is more appropriate for consortium chains and has a special focus on Enterprise.

This unique solution, which use a Raft [Raft] variant for votes gives Kalima Blockchain potential to save computation cycles, scale efficiently, and respond to the multitude of enterprise use case requirements by providing a secure, robust model for identity, auditability, and privacy.

III. Why a new blockchain?

Existing Public blockchain implementations have fallen short of meeting the multitude of requirements inherent in the complex world of Enterprise, Industrial Internet of Things and mobile devices. Technically and economically

Real time challenges, and the lack of support for confidential and private transactions, among other limitations, make their use unworkable for many industry mission-critical applications.

Trusted permissioned architecture are often coupled with Open-source elements that requires complex and costly implementations. Industry 4.0 requires trusted architecture with low Investment threshold, low Total Cost of Ownership and legacy systems compatibility. These are real challenges in Industries that have heavily invested in their legacy systems.

In order for the distributed ledger platform to be resilient to time and support requirements across industries, it needs to be lightweight, fast, open, energy-efficient and support artificial intelligence at the edge. Most of all, these characteristics should be available and operational within a short timeframe and scalable within a very reasonable investment.

Kalima Blockchain has been designed for a broad array of enterprise use cases, supply chain, smart manufacturing, smart grid, smart building and the Sharing Economy with the ambition to meet the varied demands of Industrial Internet Of Things.

IV. Our vision

Through Kalima Blockchain, we implement our vision of the future challenges facing blockchain technology.

We believe that the future will involve many interconnected distributed databases and blockchains, each of them contextually specialized in order to suit its own mission.

That is why Kalima Blockchain is a network of permissioned blockchains, each with its own governance organized in a MainChain and many PrivaChain.

Kalima Blockchain contains a rich, easy-to-use API along with numerous core modules in order to promote easy development and interoperability.

Kalima Blockchain will not address all use cases, but will provide the necessary interfaces through these API, to interconnect with other Blockchains, Middlewares, industrial field networks, and Bank backends.

The other important facet of modularity is that it facilitates outside development. If companies or individuals need specific functionalities, it should be able to build and distribute as they wish one or an entire collections of distributed applications that fit in or interact with Kalima Blockchain with the help of Kalima Blockchain API.

This emerging market will take years to mature. Blockchain economics based on cryptocurrency will struggle to adjust the requirement of a B2B market and on the other hand most “Open-source” Blockchain foundations will foster a complexity compatible with its service-based economy.

V. Our positioning

We have developed, a Blockchain, a distributed ledger [Ledger] secure by hashchain [Hash] (technology originally used to secure passwords and first used in Blockchain technologies by BitCoin) with Open Source Tools and API that matches our vision for the future of blockchain technology.

Kalima Blockchain positioning is based on the Ready to Service concept at the Core, whether the Nodes are hosted on the Cloud, on the Edge or in mobility with a very affordable Entry ticket and a long term unmatchable TCO...

The Open-Source tools, smart contracts, SDK, API technologies insure the modularity of its approach as well as the plasticity of its Ecosystems Business model.

In order to reach this promise, Kalima Blockchain contains:

1- At the Core technology level, a robust and operational Core platform with a unique delegate proof of stake consensus mechanism with a vote solution, which is a Raft variant which gives Kalima Blockchain the potential to save computation cycles, scale efficiently, and respond to the multitude of enterprise use case requirements by providing a secure, robust model for identity, auditability, and privacy. To have a complete vision of Raft consensus look at [Raft]. This document offers a good presentation of its properties and competitive advantages.
2- At the interface and modular level thanks to a rich, easy-to-use API, Kalima has developed numerous core modules on current use case, module that allow for easy development and interoperability on other projects. In Kalima Blockchain modularity is based on smart contract openness. Openness is provided by the use of standard languages, java, c#, c, NodeJS, JavaScript and python, for smart contract and API.

3- On performance point of view Latency in a single Kalima Blockchain is lower than 1s and a Kalima Blockchain allow a maximum of 1000 blocks per second in each Kalima Blockchain.

While we designed the core Kalima Blockchain modules to be able to address wide array of use cases, we understand that Kalima Blockchain core should focus on a limited number of industry use case. As stated in our vision, the interconnectivity of Kalima platform with other specialized Blockchain or federative platform is in our development priorities.

VI. Partnership & industrial cluster ecosystems

The other important facet of modularity is that it facilitates Open-innovation development.

If a Partner company improve some module of Kalima Blockchain, it should be possible for that company to build a business model on its development and to distribute their solution with Kalima inside as they wish.

Indeed, companies or individuals should be able to build entire collections of modules (that could be required to be used together, or “plug and play” with other Kalima Blockchain components) that fit in or interact with Kalima Blockchain. Essentially, it is possible to build a blockchain that uses none of the Kalima Blockchain core components yet still resides in the Kalima Blockchain framework.

In addition, the simplicity of Kalima Blockchain API and smart contract programming should enable as many people as possible to work with Kalima Blockchain. We hope that this simplicity allows people who invent or develop new technologies relevant to the blockchain to find it easy and painless to incorporate them into or use them with Kalima Blockchain.

These underlying technical primitives are configurable to support elements that are important to business transactions, such as varying degrees of guaranteed transaction finality and auditability.

In summary, we designed Kalima Blockchain to be an easy-to-use, highly functional, and robust platform that anyone who is interested in building blockchain software can use as core code. While Kalima Blockchain may fall short of this ideal functionality for every possible user and every possible use case due to practical considerations, it is our goal to make Kalima Blockchain come as close to this ideal as possible.

VII. Industry Use Cases

We have compiled a set of initial blockchain requirements that are considered essential for supporting the following abstract use cases. These use cases are not meant to represent the entire set of use cases for Kalima Blockchain, but instead compose a representative sample that demonstrates some of the capabilities and features of Kalima Blockchain.

Supply chain and smart manufacturing

The blockchain platform must provide a means to allow every participant on a supply chain or manufacturing network to input and track sourcing of raw materials, record parts manufacturing telemetry, humans and machines tasks, track provenance of goods through shipping, and maintain immutable records of all aspects of the production and storage of a finished good through to sale and afterwards. This case emphasizes the need to provide deep searchability, backwards in time through many transaction layers. This requirement is at the core of establishing provenance for any manufactured good that is built from other component goods.

Sharing Economy and Internet of Things

The Sharing Economy will generate new types of revenues in many industries, including smart cities, smart buildings, smart grids, automotive, transportation, healthcare, retailing and construction.

While transacting, however, individuals, organizations and regulators will not always trust each other. Properly implemented, distributed blockchain ledger technology will help resolve many of the trust issues that exist between various parties.

Many transactions should be settled, and status of assets should be accessible in near real time. Simplicity, openness, smart contracts using standard languages, machine learning at the edge will be important for many deployments of Kalima Blockchain.

Featured requirements

We next describe some of the featured requirements of Kalima Blockchain. While these requirements allow for many of the proposed use cases and business applications of Kalima Blockchain, we expect that Kalima Blockchain will evolve to have many more features than what we describe here.

The first, and perhaps most important, requirements of Kalima Blockchain is fast response time and low TCO. As we have repeatedly stated, response time must be compatible with alarm management and TCO must be compatible with Industry requirements. However, with this in mind, we detail some more specific requirements that will be useful for many common applications.

Private Transactions and Confidential Contracts

Kalima Blockchain should eventually support confidentiality requirements. Kalima Blockchain is a permissioned blockchain. Users and devices must have the correct level of authorization to (accede or Access) specific set of data. Each user and each machine shall have granular or limited access to specific set of data and therefore they can execute smart contracts associated with those data.

Identity and Auditability

In addition to the existence of private transactions and confidential transactions, the well-vetted concepts of identity and auditability, completes cryptographic algorithms and allows fully-realized confidentiality on Kalima Blockchain.

Besides the pure existence of an authentication process providing access to user's machines and services on the respective blockchain elements, Kalima Blockchain is also required to provide the possibility to support a comprehensible, immutable documentation/historization of these identities - including all requirements on confidentiality around them. This is necessary in order to be able to implement any use case around change of ownership, audit trails on document changes, etc.

Interoperability

In the loosely coupled world of many networks, separate networks don't need to know the details of how they each work. These separate networks, however, do need to have enough common ground to reliably exchange messages without error or misunderstanding. Especially with the expected future widespread use of blockchain technology, the parallel existence of a variety of blockchains needs to be taken into account. It is very likely that many use cases will span across several blockchains.

The differences in implementation of various blockchain networks, and their evolving and dynamic nature may result in a variety of highly specialized implementations. Standardized specification for inter-ledger communication will go a long way towards creating this as a common language across many networks.

Interoperability thus truly occurs when services can interact with each other despite the likely differences in design and implementation of blockchain technology. It is defined by the ability of two or more systems, or components, to exchange information, and to use the information that has been exchanged. In order to allow for envisioned broad usability of Kalima Blockchain across industries and use cases a functionality/ protocol allowing for interoperability between two or more blockchains is available.

Portability

The Kalima Blockchain Project achieves portability by abstracting the value-added systems from the interfaces of its core components. For instance, smart contracts could be moved from one deployment to another without having to make any other changes.

Portability of the value-added systems, such as API libraries and GUIs for developing applications, extensions, will ultimately ensure application of such value-added systems across the many versions, implementations and deployments of the Kalima Blockchain Project.

Portability on the infrastructure level will ultimately ensure that the Kalima Blockchain Project functions in the same way across many heterogeneous computing platforms and network environments, which is essential to running large blockchain networks in practice.

Architecture

Kalima Blockchain reference architecture aligned in four class of services: Identity services, Policy services, Blockchain and Smart contracts. These categories are a logical structure, not a physical depiction of partitioning of components into separate processes, address spaces or (virtual) machines.

Virtual machine

The virtual machine we have selected for smart contracts execution and validation, for Validation Nodes and for Master Nodes is the OpenJDK Java Virtual Machine, as it has a wealth of existing libraries and a large skill base and reusing an industry standard makes it easier for companies to reuse their existing code inside contracts. C# virtual machines can be also used. In clients nodes created from C SDK, you don't have any Virtual machine provided. As Smart Contracts are run from client side, they don't create a risk for the blockchain itself but for your client side implementation.

Identity Services

Identity services manages identities of entities, participants and ledger objects such as assets and smart contracts.

Policy Services

Policy services manages access control, privacy, consortium rules, etc.

Blockchain

Blockchain services manage the distributed ledger through a peer-to-peer communication protocol. The data structures are optimized to provide efficient schemes for maintaining the world state replicated at many participants.

Blockchain services consists of three key components: Peer-to-Peer (P2P) Protocol, Distributed Ledger and Consensus Manager.

Smart contracts

Smart-contract are a secured and lightweight way to sandbox the smart-contract execution on MasterNodes.

Smart contracts are codded in Java, JavaScript and Python. It makes it easy to directly use contract code from internal applications, once that contract has been reviewed, which should simplify application development considerably.

Kalima blockchain enforces business logic through smart contract code, which is constructed as a pure function that either accepts or rejects a transaction, and which can be composed from simpler, reusable functions. The functions interpret transactions as taking states as inputs and producing output states through the application of (smart contract) commands and accept the transaction if

the proposed actions are valid. Contracts define part of the business logic of the ledger: nodes will run contracts inside a sandbox.

VIII. API: Application Programming Interface

An easy-to-use, flexible API is one of the cornerstones of Kalima Blockchain. Well-defined APIs are essential for Kalima Blockchain to support the many use cases that people have developed.

In addition, APIs are designed to be extremely usable, so that a relatively unskilled developer can write code on top of Kalima Blockchain without too much trouble.

Different SDKs are provided In C, C#, Java, Kotlin, NodeJS, JavaScript and Python which implement those API.

API are open source to warranty the openness of the project. Core of Kalima technology source code is available only to "Consortium Members" as a way to protect against uncontrolled forks which could complexify the governance and create security Issues, but its governance, "Kalima Blockchain Consortium", could change this in the future.

IX. Network Topology

The network topology of Kalima Blockchain can, in principle, be quite varied: in particular, participants can use cloud services to host all kinds of peer nodes, including MasterNodes, or they can run such nodes themselves. It is important to note, though, that Kalima Blockchain runs in an agnostic manner to the underlying network structure, so who is actually running the hardware behind nodes matters little. However, Kalima blockchain has been designed to run and execute smart contracts, as much as possible, standard nodes at the edge and typically inside edge gateways.

Some Kalima Blockchain deployments will experience a great degree of variability when it comes to latency of communication and fault tolerance. Network failures, node failure, overall network resiliency and recoverability should be factored into the requirements when planning deployments.

Kalima Network is a decentralized network of **permissioned blockchains** composed of **Kalima MainChain** and of independent blockchains called **Kalima Privachains**. It’s a third generation of blockchain like Cosmos and Polkadot proposing blockchain interconnection as a new paradigm to solve the decentralization objective and achieve scalability.

The interconnection of blockchains is for us the way to help developers and businesses to adopt blockchain technology at an industrial level. Blockchains on Kalima Network are permissioned blockchain, where only predetermined nodes can see the ledger and participate in the consensus.

MainChain

Kalima **MainChain** is composed of Channels.

The KLX tokens are stored in “Kalima MainChain”.

Kalima MainChain is managed by "Consortium Members".

For the beginning of the project there will be a maximum of 5 members within the "Kalima Consortium". Each "Consortium Members" will have the option of owning a master node. The "Kalima MainChain" requires a minimum of 50 validators with a least 5 master nodes

Each validator of the MainChain must have a validation node for each Channel.

Each MasterNode of the MainChain Kalima must have a MasterNode for each Channel.

Channels

Each channel is a blockchain with at least 50 validation nodes among them at least 5 Master Nodes.

A channel cannot be closed, MainChain governance and validators are in charge to keep running all channels and to secure all KLX hold in any Channel. Each Channel must have one "coordinator client" in charge to coordinate KLX transfers between Channels. Processus Is described in the "MutiChain Processus" chapter.

Channels are deployed to warranty scalability of the MainChain.

PrivaChains

Each **Kalima Privachain** is independent with **its own governance and can be interconnected, or not**, with another Kalima Privachain. PrivaChains are interconnected with Kalima MainChain. Each Privachains require 5 master nodes.

Nodes

Each blockchain Kalima is composed of nodes, Master nodes and client nodes which are organized as follow. They manage peer to peer message to message encrypted and secure communication. They manage hashchain [Hash], data synchronization, intermittency of network and in memory memcache.

Kalima Blockchain consists of 6 different types of nodes:

Master Nodes (full nodes)

Master nodes are the main element in charge of validating transactions, they ensure traceability, integrity and immutability of all transactions.

Master nodes participate in the consensus to elect the Leader Node in charge of timestamping and hashing of all transactions.

You can install as many master nodes as you need to set up a Kalima Privachain with a minimum of five of them. Master nodes store blockchain data and they publish them to the client's node after validation. Master nodes are the only nodes with administration nodes that are authorized to access all the data contained within the blockchain, including authorization data. Master Nodes are special validation and full nodes, they implement standard Kalima consensus and when higher security is required, add a second level of validation with simple validation nodes.

Validation Nodes

Validation nodes and master nodes are in charge of controlling transactions integrity and blockchain data immutability. Simple validation nodes do not have to be full nodes, they don’t have to store all the ledger. Validation Nodes elected to validate and timestamp transactions.

Each validator must validate all blocks and blocks must be validated in respect with them time arrival. The same reward is given to all validators for all blocks validations. The reason of this choice is that we don't want to allow the risk to have transaction delayed or blocked for economical reason. For example, an alarm or a business transaction cannot be delayed or blocked using any economical hack based on validation rewards.

In a situation where a validator can't validate a given block in the allotted time, the validator will receive a temporary participation penalty. His right to candidate as a validator will be suspended for a duration of 3 months as a mean of promoting the smooth functioning of the network.

In the case of purposeful harming of the network from a validator (network attack, lack of bounty conformity) the validator will see their 3 months suspension be extended depending on the severity of their action. Kalima does not apply slashing as Kalima Blockchain technology does not allow the governance to compromise KLX ownership.

Administration Nodes

Administration nodes are the only nodes allowed to give, suppress or change authorizations to the client nodes. Authorizations are themselves stored in the blockchain. All nodes benefit from strong device identification. Devices must be authorized before any connection is made. Authorizations can be limited to a subset of data. User interface of administration nodes is Kalima Admin explorer.

Voting Nodes

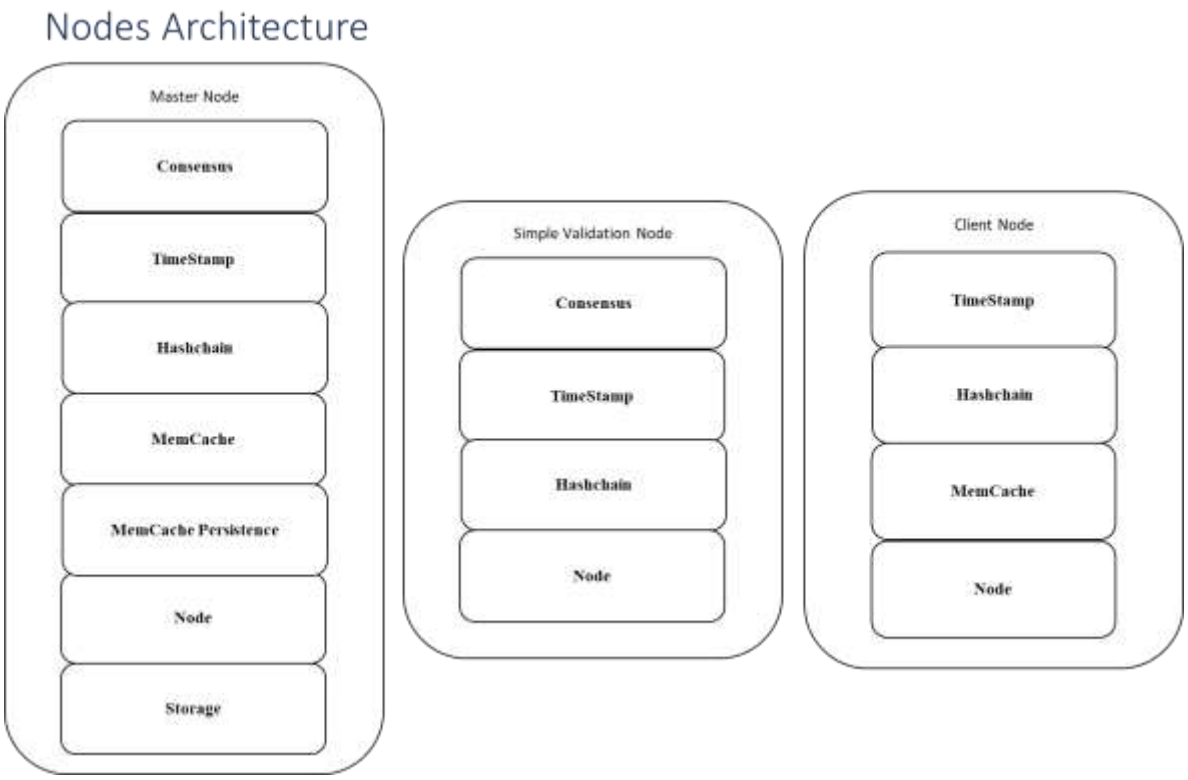
Voting nodes enable validators to vote for governance management choices and to confirm loging access to administration nodes in a multi signature way. They are also used by "Stakers" for "Staking" purposes. Each "Validator" owns one "Governance voting node". Each "Staker" owns one "Staking voting node". Votes weight are related to the number of KLX owned by Stakers or Validators.

Client Nodes

Client nodes are used to synchronize data to which they are authorized, create new transactions and execute smart contracts. Client nodes can add and receive data to and from the blockchain depending on their authorizations. Smart contracts are executed in the client nodes on data arrival. Smart Contracts must have been controlled and authorized by Validation nodes before being executed. Client nodes can be developed by our users and partners with the help of Kalima SDK. Kalima SDK provides tools to develop java, C#, C, NodeJS, Android and iOS client nodes.

Data lake and Data Safe Nodes

Data Lake and data safe nodes are Client nodes which collect data of one or multiple blockchain, depending on the authorizations, to compile and publish them to facilitate data search, machine learning, statistics, secure version management or to provide a highly secure storage.



1. Kalima Blockchain instead offers a unique delegate Proof Of Stake consensus mechanisms. This unique solution, which use a Raft variant for votes gives Kalima Blockchain the potential to save computation cycles, scale efficiently, and respond to the multitude of enterprise use case requirements by providing a secure, robust model for identity, auditability, and privacy.

2. TimeStamp register the date and time of block validation

3. Hash chain is the successive application of SHA256 hash function to blocks. Hash function is applied successively to blocks in order to record the chronology of block's existence. Block size is limited to 16 ko to limit latency and protect from buffer overflow attacks.

4. MemCache is an in memory replicated key value store in charge to maintain real time value of systems, control of data integrity and data resynchronization. For example, if you use Kalima to monitor a temperature gage in a Plant the address of this sensor in memcache will contain the current value. If you use Kalima for a token account, it will hold the current balance of your address. This solution ensure that a data is sent once to MasterNodes, validated, and then time stamped, hashed, stored and then published to all subscribers. Each subscriber executes, depending on rules, its own corresponding smart contract and then can read it locally without to have to get it back from blockchain and by this solution save time, bandwidth and energy.

5. MemCache persistence is a key value store database in charge to facilitate restart of nodes even in case of a full stop of the blockchain.

6. Node layer is a aes256 encrypted publish subscribe message to message peer to peer communication layer capable to manage network failure including half broken connections.

Kalima publish subscribe layer has been developed by Kalima Team from bottom up to depend on no dependency an to provide the best warranty of security and reliability. It includes algorithms managing network intermittent failures and data resynchronization when needed to warranty data integrity.

7. Storage is the permanent and immutable storage of blocks secure by hashchain. Kalima Blockchain storage is store in encrypted filesystems, in a hierarchical key value stores organized on time and addresses.

Cryptography

In response to attacks threatening networks in general, including wireless networks, that are present in current communication systems, we have put in place a solution combining symmetric and asymmetric algorithms to add their respective advantages and to produce robust encryption and strong identification of communicating nodes .

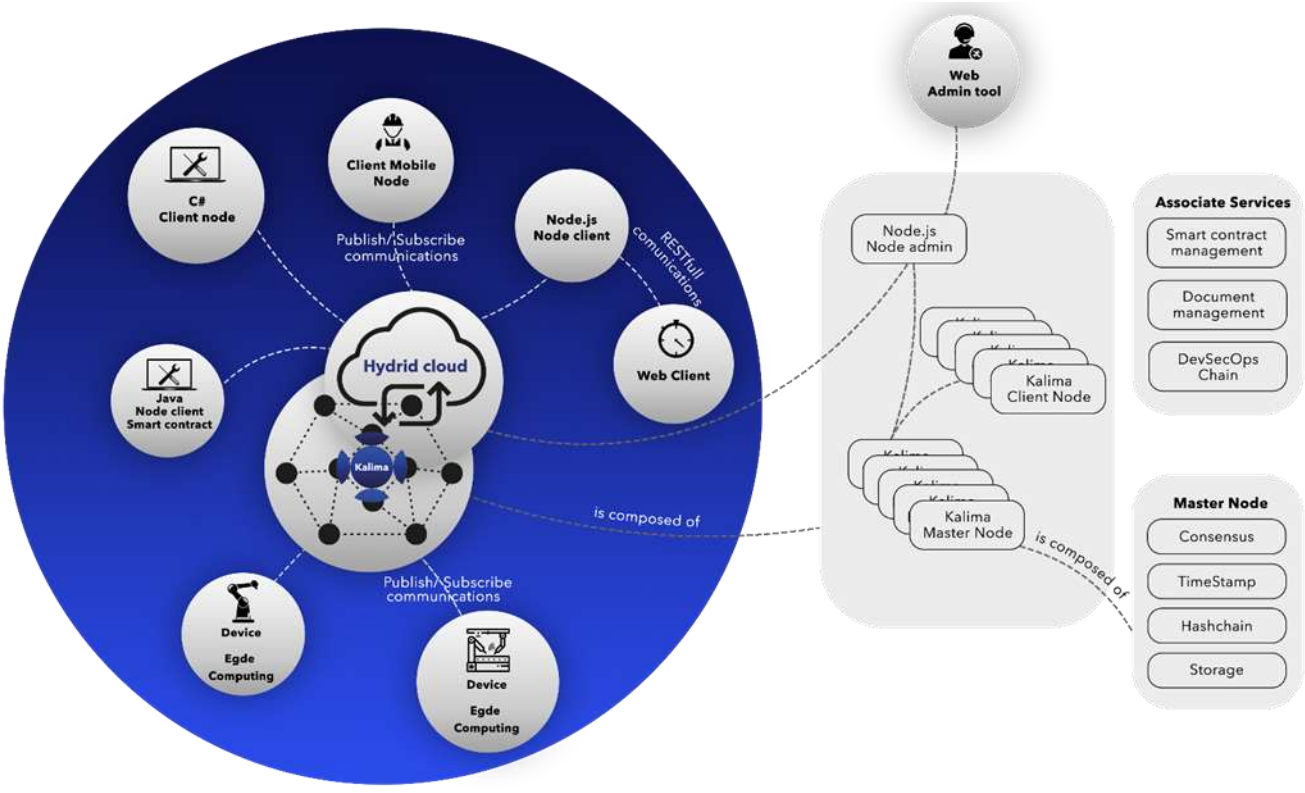
These algorithms are based on the implementation of asymmetric encryption (or a “challenge”) which will be used to generate/ exchange symmetric private keys.

Once the exchange has been made the symmetric encryption is set up.

The regular rotation of the keys allows to keep an optimal safety.

Blockchain Kalima

Each blockchain Kalima is composed of master nodes, validation nodes and client nodes but also of associate services



Associate Services

Smart Contract management include a full DevSecOps management of Smart Contract code. It includes:

1. A version control warranting the traceability and integrity of codes.
2. A quality and security control of Smart Contracts code
3. A quality and security of deployments including upgrades.

Document management includes a version control warranting the traceability and integrity of documents.

DevSecOps Chain include a full DevSecOps management of Kalima Blockchain code. It includes:

1. A version control warranting the traceability and integrity of codes.
2. A quality and security control of Kalima Blockchain code
3. A quality and security of deployments including upgrades.

Master Nodes (full nodes)

1. Kalima Blockchain instead offers a unique delegate Proof Of Stake consensus mechanisms. This unique solution, which use a Raft variant for votes gives Kalima Blockchain the potential to save computation cycles, scale efficiently, and respond to the multitude of enterprise use case requirements by providing a secure, robust model for identity, auditability, and privacy.

2. TimeStamp register the date and time of block validation

3. Hash chain is the successive application of SHA256 hash function to blocks. Hash function is applied successively to blocks in order to record the chronology of block's existence.

4. Core of the consensus is a distributed state machine. Raft Proof-of-Authority (PoA) consensus [Raft] alone would be sufficient for Kalima Blockchains. However for Channels, to be deployable as a network in a fully open and public situation, a double validation level mechanism is implemented to enforce Master Nodes validation and we select a set of validators with a Delegate-Proof-of-Stake (DPoS) based selection criteria [PoS].

5. Storage is the permanent and immutable storage of blocks. Kalima Blockchains storage is organized depending on time and addresses. Master Nodes, for security reasons, even in case of hardware maintenance has to be installed in a fully encrypted filesystem.

MultiChain processus

Kalima Blockchain allow a client to be connected to several Kalima Blockchain and thus the only condition Is that this client has the required authorizations in both Kalima Blockchain for the targeted purpose.

All KLX addresses are in the MainChain and can be in different channels.

Kalima Blockchain implements a "non-blocking two phase commit" and more precisely a "Presumed Commit (PrC)" (cf chapter 6.2 [ISI_98]) algorithm for KLX Inter Blockchain Transactions between

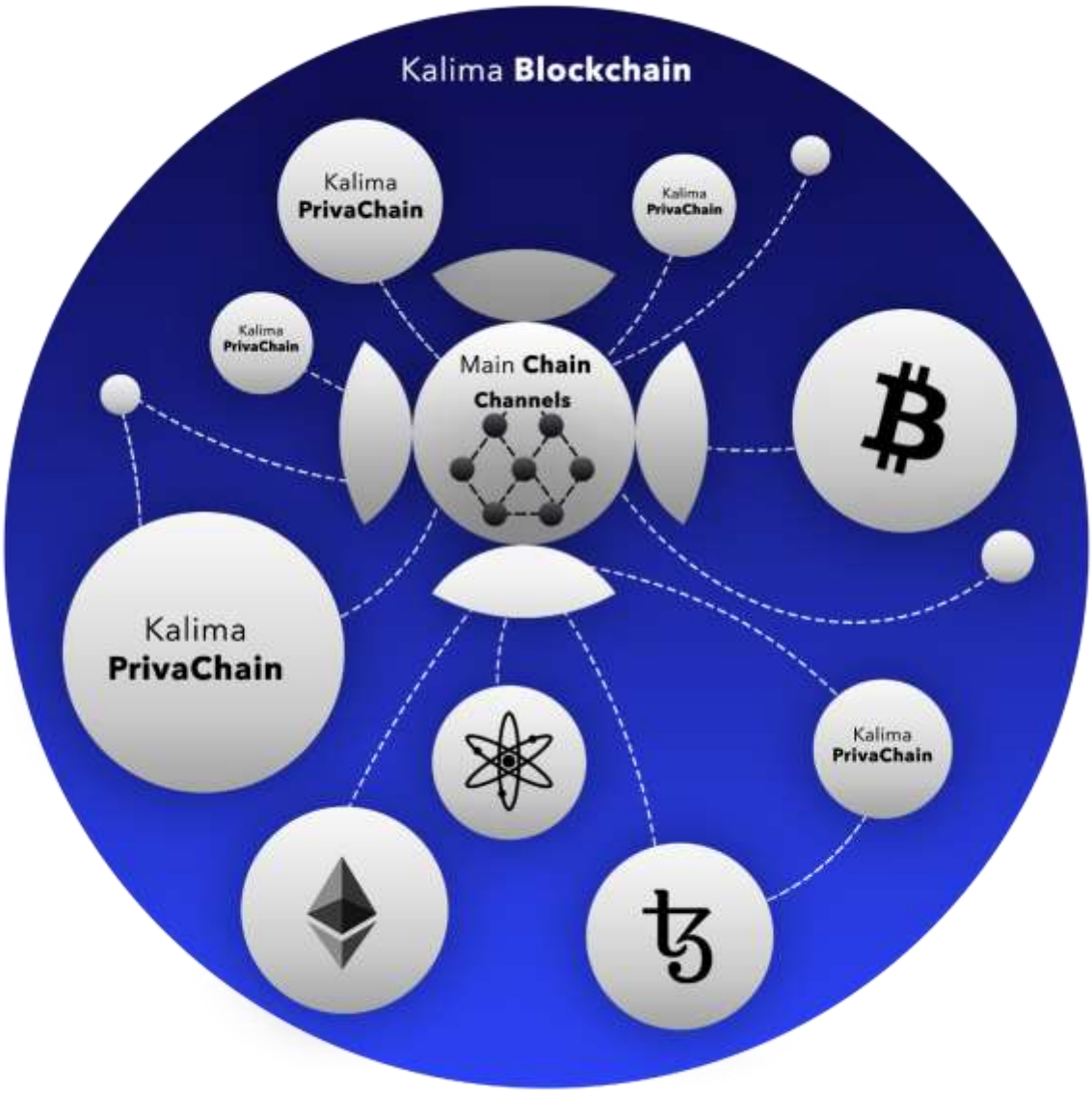
Channels and more generally for financial transactions. Blockchain integrity of communication, fault tolerance and immutability of transactions storage offer required properties to secure It.

Client connected to both transaction sides act as the "coordinator" for this process. The full immutable traceability of transaction is available on both sides. In short, the Initiator of the transaction will hold the amount in a temporary transfer address, then the "coordinator client" will inform the recipient, and request a vote from both parts. If both parts vote positively, amount will be transferred to the recipient if one vote negatively, amount will be released to the sender. This process is valid for multi devises. All this process Is transparent for users and transaction time stay short but higher than a mono-chain transaction.

Kalima Network

Kalima Network consists of one public MainChain and many PrivaChains.

"Kalima MainChain" is interconnected with Tezos, Bitcoin, Ethereum and Cosmos public Blockchains to offer a hybrid private/public blockchain possibility.



Staking pool

A « staking pool », is a voting nodes where votes weight are related to the number of KLX they stake.

Stakers

A « Staker », is a KLX holder willing to stake their KLX tokens.

Each Staker choses the pool in which they want to stake their KLX.

The staking lock-up period is defining the period between the staking action and the release of the KLX.

Delegates

Members of the Networks can candidate as a validator or can delegate their vote by pooling their tokens into a staking pool and linking those to a candidate. Holders do not physically transfer their tokens to another wallet, but instead stake the tokens in a staking pool.

RoadMap

Main technical elements remaining to develop are:

Double level validation, which is not deployed as not completed tested and validated.

External independent Security audits. Only internal security validation has been done.

MainChain deployment with KLX, only internals non-tradable tokens, KLX will be launched first as an ERC20 and then transfer to the MainChain.

Data Lakes and Data Safes are not already publicly available as non-tested and validated.

MultiChain process is not available as non-completed tested and validated.

Voting nodes development is in progress and will allow access to votes and multi signatures usages like for Kalima Explorer Admin access. It is not available as non-completed tested and validated.

Kalima Blockchain Explorer is the blockchain oriented explorer dedicated to end users is currently under development non completed tested and validated.

Conclusion

Kalima Blockchain's mission is to enable mainstream industry adoption of blockchain technology. After reviewing the available blockchain solutions and hearing use cases from both industry leaders and technology evangelists, we are convinced that blockchain will be an extremely important technology pattern that could revolutionize many industries and businesses.

We have observed that industry is urgently calling for a business-ready blockchain platform that is both efficient and scalable and offers enterprise-grade support for privacy and confidentiality. We have also discovered many different categories of use cases, each of which may require fast response time and low TCO.

We designed the Kalima Blockchain platform to fulfill those requirements.

Glossary

Types of Network

Permissioned vs. Non-permissioned

Permissioned Network	A blockchain network collectively owned and operated by a group of identifiable and verifiable business entities.
Non-permissioned Network	A blockchain network with no identifiable ownership structure and is operated by a community of participants that may or may not be identifiable.

Types of Chains

Standard Chain	A blockchain network with many participants; each chain operates one or multiple applications/solutions validated by a group of organizations/business entities.
Confidential Chain	A special purpose chain created to run confidential business logic that is only accessible by contract stakeholders.

Kalima Blockchain Entities

Smart Contract

MasterNode Smart contract	Smart contracts deployed by MasterNodes.
Node Smart contract	Smart contracts deployed standard (client) nodes.

Ledger

Real time state	In memory database maintaining the current state of all entities.
Historical chain	All processed transactions are kept in the ledger in their original form (with payload encrypted for confidential transactions), so that network participants can interrogate past transactions to which they have access permissions.
Ledger Hash	A hash that captures the present snapshot of the ledger. It is a product of all validated transactions processed by the network since the genesis transaction.

Node

Node	Node providing APIs to interact with their MasterNodes and chain network. These nodes are responsible to construct transactions.
MasterNode	MasterNode are responsible to control, certify, process transactions, deploy and execute smart contracts, maintain ledger data, and trigger the consensus process.

References

- [Ledger] S, Surbhi (26 Jul 2018) difference-between-journal-and-ledger
- [Hash] L. Lamport Hashchain Password Authentication with Insecure Communication
- [N09] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
- [Eth] Ethereum Whitepaper
- [Raft] Diego Ongaro, John Ousterhout Stanford University In Search of an Understandable Consensus Algorithm
- [HPL02] Miguel Castro and Barbara Liskov, Whitepaper_IntroductiontoHyperledger
- [ISI_98] Maha Abdallah, Philippe Pucheral Atomic Validation: state of the art
- [PoS] Sunny King, Scott Nadal, 2012 Peer-to-Peer Crypto-Currency with Proof-of-Stake

Follow The Future
IoT Blockchain Leader.

www.kalima.io

Linkedin

Kalima

Twitter - Instagram

@Kalima_KLX

Kalima
Blockchain & IoT
Collect.Protect.Monetize