KardiaChain - The First Decentralized Interoperable and Self-Optimised Blockchain Ecosystem

Huy Nguyen, Tri Pham, Thien Nguyen, Nam Do and Hung Nguyen

ABSTRACT—**Blockchain offers an unparalleled level of decentralisation and transparency, with a tradeoff in performance and inter-connectivity. In the foreseeable future, it is believed that blockchain solutions, especially smart contracts, will help people reach a trustless agreement at ease in all aspects of daily situations. However, the current approaches to achieve this either are built to make others obsolete, or requires a significant change on the part of the participating chains. KardiaChain follows an "integration without assimilation" approach, which focuses on simplicity and ease-to-use from the standpoint of both the end-user and the developer.. The KardiaChain team has developed a non-invasive solution, called Dual master node (or Dual node for short), to facilitate inter-chain operations among both existing and upcoming blockchain platforms. The ultimate goal of KardiaChain is to create a unified ecosystem where developers can easily create smart contracts that can run on multiple blockchains, in order to to optimise costs, avoid congestion, and allowcommunication with smart contracts on other chains in a trustless and secure manner.**

I. OUR PHILOSOPHY

The KardiaChain team strongly believes that mankind is soon reaching a stage at which blockchain solutions, especially smart contracts, will help people reach trustless agreement with ease in all aspects of daily situations.

Yet, unlike the current approach where many chains are built with sole purpose to make others obsolete and to become the one chain that fits all, the KardiaChain team expects situations to unfold differently in the real world. Similar to the internet (interconnected-networks) being a gigantic network of networks, expects an internet of blockchains that is a connection between numerous public/permissionless and private/consortium blockchains to form. Each of the individual blockchains will have a different design to serve specific purposes, and we believe they should be embraced for their own uniqueness.

Hence, there should be a smart ecosystem capable of utilising the collective strength of all existing blockchains and overcoming challenges relating to discoverability and isolation. The KardiaChain team would like think of this smart ecosystem as a single living organism, of which each individual blockchain is an organ or body part, and KardiaChain as the heart that facilitates blood flow (traffic/transactions) to every part of the body at a level depending on their specific need.

While there are several proposals for inter-chain communications, all of them currently require fundamental changes to existing blockchains and a massive wave of adoption of their protocol/framework from newly created blockchains. In contrast, the unique approach of KardiaChain is practical enough for an easy and fast implementation of inter-blockchain connections which require no change on participating chains. KardiaChain follows an *integration without assimilation* approach, with focuses on simplicity and ease-to-use from the perspectives of both end users and Dapp developers.

The KardiaChain team takes pride in introducing the KardiaChain smart inter-blockchain ecosystem, on which the KardiaChain team has been working hard, with the strong belief that this will facilitate mass adoption of blockchains.

H. Nguyen, N. Do, and T. Nguyen were with Google Inc. and KardiaChain. Email: {huy, nam, thien}@kardiachain.io

T. Pham and H. Nguyen were with KardiaChain. Email: {tri, hung}@kardiachain.io

II. BACKGROUND

A. Single chains

Blockchain offers an unparalleled level of decentralisation and transparency, with a tradeoff in performance and inter-connectivity. For most single chains, the top priority is scalability, or to increase their processing speed, with transactions per second (TPS) as the key metric. A popular technique being used is sharding [1], which is the division of the blockchain into smaller shards (sub-chains) and simultaneous processing of the transactions, promising TPS from thousands to millions. However, even based on theoretical estimations of existing projects, no single chain appears to be able to handle the calculating power needed for mass adoption. For instance, Facebook requires over 16k tps to facilitate major interactions: as every 60 seconds there are 510,000 comments, 293,000 statuses, and 136,000 photos uploaded [2]. On Instagram, 4 billion likes per day would result in over 46k tps [3]. In Youtube's case, 1 billion hours of video watched per day means nearly 7 million tps, assuming one piece of content is requested every 6 seconds per user session [4].

A prominent reason for this obsession over TPS is that single chains often suffer from lack of interoperability, meaning they cannot communicate with each other, and therefore, operations cannot be done on multi-chain. The lack of interoperability significantly reduces the chance of a finding appropriate blockchain-based solutions for many real-time problems in various businesses. For example, while there are many chains supporting smart contracts, each platform has a different computational capability. In an extreme case, Bitcoin-based blockchains do not even support smart contracts [5]. This means that there is virtually no way to build a complete solution using just one chain. As there is no one-size-fits-all solution for a blockchain application, this creates a steep learning curve if one desires to build smart contracts on different chains.

B. Inter-chain Blockchains

1) The scaling cube: If one thinks of a single chain as a single computer or a local network, it would be much more scalable to connect more computers to share the workload, instead of building an increasingly faster supercomputer. The following section employs a widely used visualization known as the scaling cube [7] to better map different approaches in blockchain scalability:

- X-axis = Horizontal duplication = Transaction routing: data is stored on multiple nodes to reduce traffic and avoid congestion.
- Y -axis = Function scaling = Different chains handle different tasks: this means solutions can be split into separated services that use separate databases. As a consequence, the high congestion in one operation does not affect the performance of the others.
- Z-axis = Data partitioning = Share workload between each part of a chain (such as Sharding): run identical copies of code on multiple nodes to share the workload.

Many single-chain solutions focus on Z scaling by implementing sharding. For example, Ethereum network sets sharding as one of the main focus of the next version [11]. The KardiaChain team aims to achieve full 3-dimensional scaling by putting emphasis on X/Y scaling while leveraging the sharding of existing and upcoming blockchains. Because of interoperability, similar tasks can be shared between blockchain platforms to achieve X scaling. For Y scaling, an on-chain operation can be divided into smaller tasks and run on different blockchains. This is the blessing of interoperability and interconnectivity on which KardiaChain is working.

2) Related works: While many attempts have been implemented to develop interoperability between blockchains, each of them have required some specific changes on the participating chains, which may have required major updates, even forks (soft and hard). An attempt in making changes to a specific blockchain will lead to two significant side effects: 1) a hard fork may be required to accommodate the new changes; 2) security issues may arise when one tries to change the core configurations of the blockchain as mentioned above. This section discusses some common cross-chain approaches.

a) Sidechains: Sidechain is a blockchain that runs parallel to the main blockchain, which extends functionality through interoperable blockchain networks, allowing a decentralised way of transferring/synchronising ones tokens between the two chains [12]. In other words, one can move one's cryptocurrency to the sidechain and then back to the main chain. The problem with sidechains is that assets transferred to sidechains (and back to mainchain) need to be locked for a certain duration, called the contestant period, before they become transferrable. Although the idea of sidechain dates back to 2014, so far there has been no provable implementation that is mature enough and production ready.

b) Message layer: Message layer is designed to be an intermediate layer that collects data generated from external blockchains [13]. Treating everything as messages creates a need for an extra layer to filter and order them messages. Simply speaking, the network need to sanitise the messages before putting them on the ledger. This process may actually require extra effort, which would lead to performance issues. In the opinion of the KardiaChain team, KardiaChain can extract the essential data from external transactions directly, thanks to the immutability of the blockchain, and thus KardiaChain can omit this extra, perhaps unnecessary, layer.

c) Hub/Connector/Adaptor: Hub (connector/adaptor) is designed to be the communication channel between participating blockchains. Through this kind of channel, blockchains can interact with each other to transact and exchange assets. One of the most famous implementations of this approach is Polkadot [14]. The critical problem of this approach is that participating chains must be compatible with the hub for cross chain communication, which means the current blockchains need to be changed (forked) to join the networks. In other words, the hub approach does not adapt to others but instead requires the participating chains to change. This is believed to be counter intuitive to the meaning of both interoperability and practicability.

3) KardiaChain approach - Integration Without Assimilation: In simple terms, KardiaChain adapts to others. The ultimate goal of KardiaChain is to offer a native and ready-to-market approach which satisfies several preferable requirements, notably:

- Fits in with participating blockchain as-is and thus allows simultaneous progress from solution to production.
- Keeps all the strengths , consensus mechanisms and most importantly the security of participating chains uncompromised.
- Being transparent and straightforward with Dapp developers, KardiaChain will handle the heavy workloads of inter-chain communication while still providing developers with the space to implement their own logic.

III. KardiaChain Solution (Pending Patent)

To tackle the interoperability challenge, and subsequently the scaling issue of the entire blockchain ecosystem, KardiaChain aims to provide a unified infrastructure that can develop solutions which are able to run on multiple platforms, by enabling the interoperability among them, and while leaving the implementation as simple as possible from the perspective of developers. One of the many unique features of KardiaChain is that a user can interact with KardiaChain to trigger an event on one chain that leads to a result on a different chain.

KardiaChain is a blockchain network that connects other networks and applications in the blockchain ecosystem. The goal is to create a unified ecosystem that combines the collective strength of all participants, and to lay the foundation for the upcoming blockchain mass adoption. The primary solution of KardiaChain is the *Dual Master Node* (or Dual Node for short) which has three prominent components: *Translator, Router*, and *Aggregator*. This solution provides a practical and non-invasive way to facilitate secure and decentralised inter-chain connection, reduceing costs and increaseing speed.

A. Technological advantages

The solution provides a number of technological advantages:

- Algorithmic approach in transaction routing to optimise costs and speed
- Unified smart contract language to enhance developability
- Non-invasive solution provides backward compatibility and zero-change is required on other chains for integration
- Practical implementation
- Secured and decentralised inter-chain data ingestion

B. *Important components*

- **Dual Master Nodes** have access to the ledger data of two chains simultaneously (KardiaChain and another chain of choice). They can receive transactions from external chains and safely ingest those updates into KardiaChain ledgers without compromising anything from both chains. *Dual Master Nodes* are decentralised because everyone can run a Dual Master Node (permissionless) and they have a consensus among them to verify the data from both chains. Dual Master Nodes are secure because interchain transaction data to/from KardiaChain is tamper-proof, protected by a multisignature scheme [8] such as Schnorr Signature Algorithm [9].
- **Translator** utilises *Kardia unified Smart contract Language* (KSML) to break the language barrier between different smart contract platforms, facilitating a "mutual understanding" of instructions in smart contracts between KardiaChain and external chains.
- **Router** determines the best chain to which the translated request can be directed, based on multiple inputs such as current perfor-mance, fee, waiting time, and capacity.
- **Aggregator** batches new updates from other chains to reduce strains on KardiaChain, potentially reduce one block of updates to one transaction on KardiaChain. The combination of the above concepts creates countless opportunities and lays the foundation for a mass adoption of blockchain.

The flow of a Dual node operation can be broken down into following steps:

- **Step 0:** Some users initiate $TX(r(i))$ with $i = 1, 2, 3$ targeting a smart contract SMC A.
- **Step 1 and 2:** Router and Translator detect and handle them.
- **Step 3:** Dual node makes $Func\ call(j)$ with $j = 1, 2, 3$ targeting External Chain for confirmation.
- **Step 4:** The result is $TX(u(l))$ with $l = 1, 2, 3$ coming from External Chain.
- **Step 5:** Dual node gets these $TX(u(l))$ back in external chain format.
- **Step 6 and 7:** Aggregator and Translator produce a single transaction $TX(k) = [value : (1, 2, 3)]$
- **Step 8:** Changes $(1, 2, 3)$ is executed on SMC A and then gets applied into ledger.

As one may notice, the new procedure allows KardiaChain to combine the processing capacity of participating chains, redefining the notion of TPS metric.

$$KardiaTPS = K_l + \sum_{l=0}^{N} MIN(R, E, K_l\, t) \times \frac{1}{1 - \bar{t}},$$

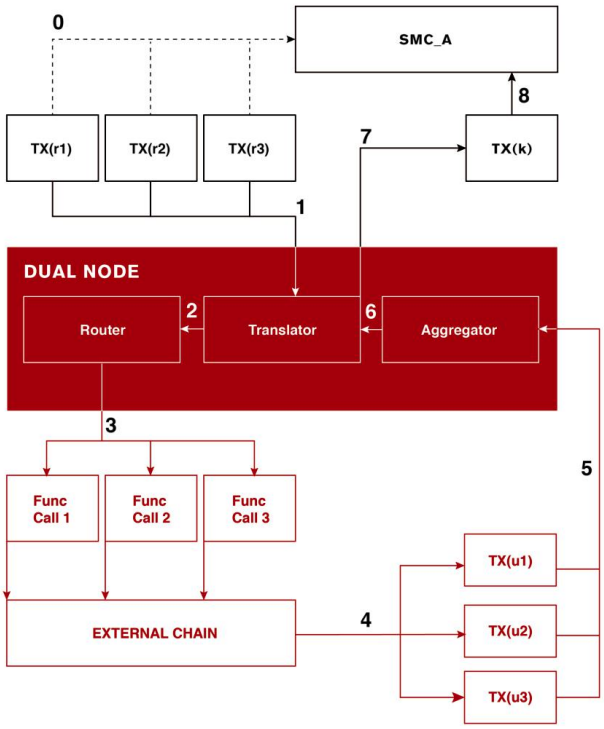

Fig. 1. Simplified Dual Node Operation

where K_l is the native TPS of Kardia and each Dual Node Groups, R is the total routing capacity of dual nodes, E is the total TPS of participating chains, and t is the batching ratio. Note that t varies as more chains supported by KardiaChain and depends on the Block Size as well as BlockTime of each chain. Typically, t ranges from 1 to a max value:

$$t_{MAX} = \frac{\sum_{l=1}^{N} TPS_l}{\sum_{l=1}^{N} \frac{Block\ Size_l}{Time_l}},$$

where n is the number of participating chains.[1]

IV. KardiaChain Technology

A. *Node components*

For the sake of visualisation, a node can be divided into three parts using a biological-analogy: The Heart, Arteries, and Veins. Dual nodes actively use all three to facilitate inter-chain connectivity. The Heart to stores and pumps transactions, through Arteries towards other chains, and come back to KardiaChain via Veins.

If for any reason, a node chooses not to contribute to the ecosystem interconnectivity, it will leave only the Heart active and become a Standard node that handles internal transactions.

1) The heart:

[1] $T = 1$ is a special case when only one transaction is routed, resulting in one update to Kardia, hence batching ratio is 1.

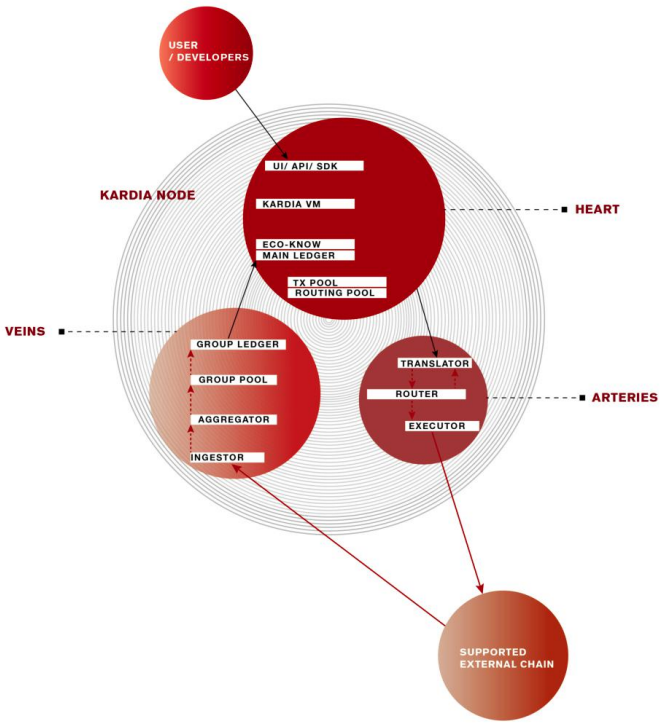

Fig. 2. Node Components

a) KardiaChain Development Toolkit: Since the goal of KardiaChain is to ease the steep learning curve of the smart contract framework of other chains, an intuitive set of *UI, API*, and *SDK* will be provided to enable the simple construction of a smart contract on KardiaChain. This set, known as the Conduction system keeps the Heart operating.

- **User-friendly UI** to provide template for creating simple smart contracts on KardiaChain. These templates help developers with little to no knowledge of smart contract language to further customise a number of popular contracts
- **Efficient API** to directly construct more complex smart contracts on KardiaChain. This feature aims for intermediate developers to deploy any custom logic that is executed on multiple chains
- **Powerful SDK** for Dapp developers to freely explore all the features of smart contract on KardiaChain. SDK provides developers with full control of the entire inter-chain flows of smart contracts on KardiaChain: how sub-contracts are created on external chains, how updates are gathered, everything is determined by the individual decision of developers.

b) Kardia Virtual Machine (KVM): KVM is an enhanced version of EVM [10] with added cross-chain functionalities. Smart contract running on KVM can process external events from other participating chains to reach the final state of the ledgers without compromising the consensus mechanism on KardiaChain.

KVM maintains two fee systems for internal and inter-chain transaction. A customised pricing will be applied for inter-chain transactions, to encourage participating nodes to carry out the inter-chain operations and ensure the system is working smoothly.

KVM also supports a special set of operations to help handle cross-chain logic relating to processing and aggregation of updates from external chains.

c) Ecosystem Knowledge (EcoKnow): EcoKnow keeps the statistics of the blockchains with which KardiaChain is actively engaging. This is one of the components that makes KardiaChain a superior solution, by providing a great deal of knowledge for the smart routing algorithm. Beside transactional updates, statistical updates are also obtained and analysed in a real-time manner. EcoKnow extracts and records this info to obtain comprehensive and actionable data pertaining to the ecosystem. EcoKnow serves as a knowledge base of the entire network, accessible to even newly-joined nodes, giving these new nodes instant knowledge of the whole network.

d) Main Ledger: Main Ledger is kept by all nodes (Standard and Dual) and consists of all Main Blocks (please refer to Block Structure). The main ledger is a tree-structured ledger, where each Main block keeps a reference to branch ledgers created by Dual nodes. Main Ledger contains the global state of the KVM, in which changes are made by transactions inside KardiaChain, as well as transactions run on external chains. The goal of Main ledger is to provide a consistent view of historical data regardless of which chain the transactions were executed on.

e) Master Wallet: Master Wallet is a safe way to store user credentials with a high level of security, and it allows users to create inter-chain transactions without micromanaging multiple sets of private/public keys for each chain.

2) Arteries: The Arteries allow transactions flow from KardiaChain to other chains while ensuring inter-chain operations work seamlessly on the destination chain. This is a multiple-step process which includes a two-pass flow between the Translator and the Router, and a smart contract submission via the Executor.

a) Translator: Translator uses an algorithm to translate a provided smart contract on KardiaChain into a comprehensive KardiaChain bytecode that includes smart contracts for the involved chains, logic to advance the smart contract state machine on KardiaChain, and the ability to handle failures on other chains.

b) Kardia Smart contract Markup Language (KSML): KSML is designed to be the ultimate tool for developers to develop smart contracts on the platform of KardiaChain without requiring significant learning curves or any prior experience. By introducing a huge set of code instruction in a human readable format (preferably JSON or YAML), developers can easily employ their popular logics without having to write a single line of real code for the targeted blockchains (eg. Solidity for Ethereum, C# , Java, Python for NEO and others).

Along with the KSML is the syntax checker, which the developer can use to quickly verify whether their KSML contracts are correct and parsable on the platform of KardiaChain. KSML and the tool will be packaged into the SDK to provide developers with a unified toolkit for implementing their own solutions on KardiaChain.

In the future, KardiaChain aims to not only support KSML contracts but also allow developers to embed the chain-specific code (eg. Solidity for Ethereum contract), providing them with unlimited capability to write any arbitrary logic beyond what is offered in KSML.

c) Inter-chain Machine-Learning Network Router (CMNR): CMNR uses the SON-based selection algorithm [16] to find the most suitable blockchains to participate in the interchain transaction. The decision is determined by many factors such as transaction fee, confirmation time and traffic load. CMNR applies dynamic scoring algorithm, which is inspired by the most advanced analytics framework and identifies the score of a specific blockchain X at the time t for a specific request R as follow:

ScoreXt = f (h, f, v, d)

h : the block height,

f : the recently recorded fee,

v : is the block time of X,

d : is the normalised difficulty calculated by a specific formula to find the respective difference in difficulty between smart contracts

With the live data continuouslybeing fetched from the EcoKnow described above, CMNR can make efficient decisions in smart contract routing, thus demonstrating the best performance at the most competitive cost. Thanks to the ability to self-organise and self-optimise, all modifications and enhancements on the core algorithm of SON can be applied automatically without any human actions.

Logically CMNR will have two interfaces: the internal interface, which is a JSON-RPC API, provides routing function for the SNode to route the transactions from KardiaChain, and the external interface, which is RESTful API for developers to invoke so that they can obtain the best routes for their smart contract calls.

d) Executor: Executor handles the least logic-intensive work of submitting translated smart contract bytecode, following detailed instructions from the CMNR to destination chains through their corresponding JSON-RPC.

3) Veins: Veins are responsible for getting updates from the ecosystem, efficiently processing the updates, and safely ingesting them to the Main Ledger. Veins has several key features as follows:

a) Ingestor: Ingestor gets new blocks from a specific chain to which the dual node is bound. Relevant updates are extracted and streamed to the Aggregator in real time.

b) Aggregator: Aggregator receives real-time updates from Ingestor and: (1) Matches the routed transactions with correct reference to external txID, and (2) Batches compatible transactions into new KardiaChain transactions and pushes these into the pool.

c) Group Pool and Group Ledger: Group Pool is where Dual nodes gather routed transactions by reference and process them in the next block. Group Ledger keeps a comprehensive record of these routed transactions. Refer to Block Structure part to see how Group Ledgers are linked to Main Ledger.

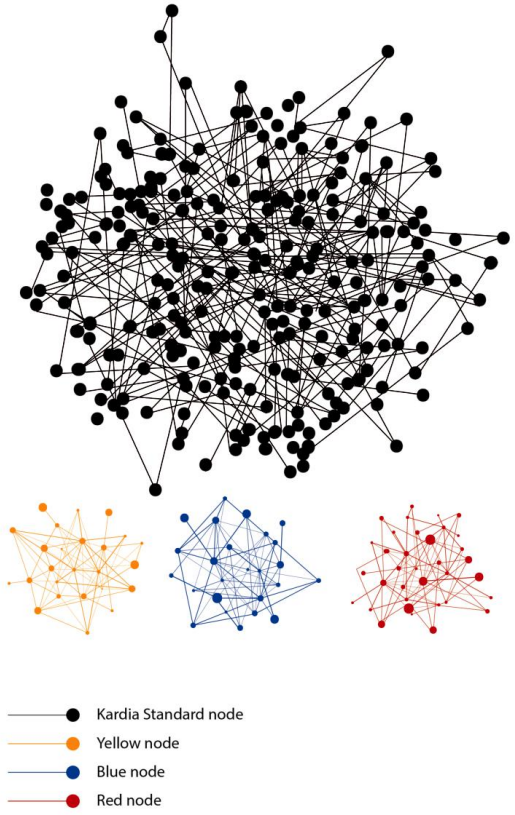

Fig. 3. Initial stage - all nodes are standardl

B. System Architecture

All newly joined nodes are Standard nodes by default. They can stake to become a Dual Node and choose an external chain to support. The process of staking follows the standard Proof-of-Stake consensus protocol [17]. Dual nodes operate in groups formed by the connection established among Dual nodes supporting the same chain. It is notable that each dual node only binds to one specific chain at a time. This is because, for a single node, processing transactions involving multiple external chains simultaneously is inefficient and impractical. Future research is required to explore the feasibility of Multi-nodes.

In the figure below, Dual nodes in the same group are scattered in terms of connection, since nodes chose their supported network by free will. The network periodically runs an optimisation algorithm (part of the Elastic Sharding with Incentive Mechanism) to ensure optimal connections between all nodes.

In Figure 5, Dual nodes in the same group are scattered in terms of connection, since nodes chose their supported network by free will. The network periodically runs an optimisation algorithm (part of the Elastic Sharding with Incentive Mechanism) to ensure optimal connections between all nodes, see Figure 6.

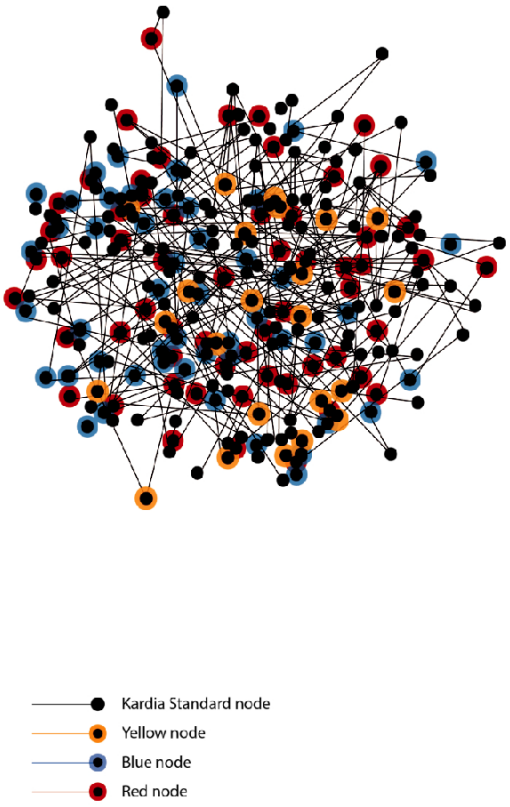

Fig. 4. Switching stage - nodes switch to support a network of choice

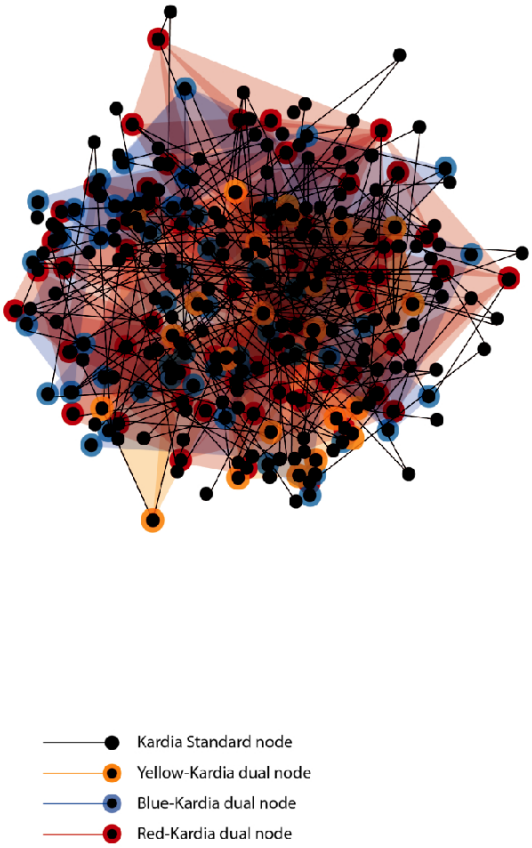

Fig. 5. Primitive distribution of Dual Groups

C. Elastic Sharding with Incentive Mechanism (ESWIM)

Dual nodes are incentivised by typically receiving a larger portion of block reward and total transaction fee (TX fee) in compensation for their complex function. The goal of ESWIM is to ensure both optimal network performance by maintaining an appropriate number of nodes in each group, and network security by adjusting staking power among Validators in groups and the main network.

ESWIM encourages nodes to switch between Dual/Standard mode and a demanding group using several instruments, such as adjusting the block reward distribution, TX fee com-mission, and modification of node requirements for specific chains. For example, increasing the minimum of staking requirement will shift some nodes out of a crowded group, while lowering this threshold can attract more nodes to join a group in need.

As illustrated above, the change in transaction fee in Blue chain and Red chain between time t1 and t2 leads to a transition in Dual groups on KardiaChain. As operations in Red chain become more expensive, the demand for routing to Red chain drops. The opposite occurs with Blue chain. ESWIM makes the decision to adjust the number of Dual nodes in group A and B accordingly in anticipation of the coming change in number of routing transactions. Note that transaction fee is only one of the criteria for ESWIM to act.

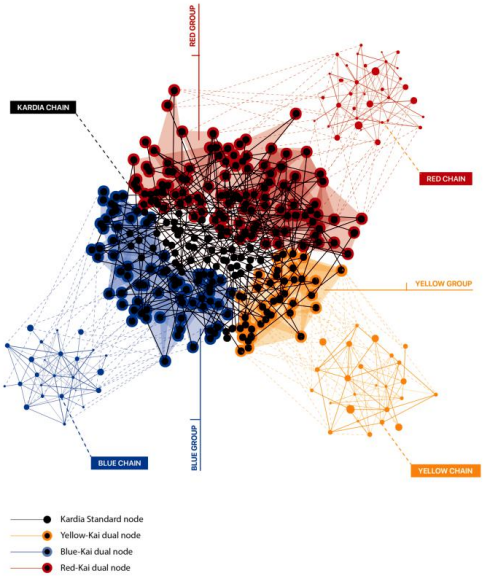

Fig. 6. Optimised distribution of Dual Groups

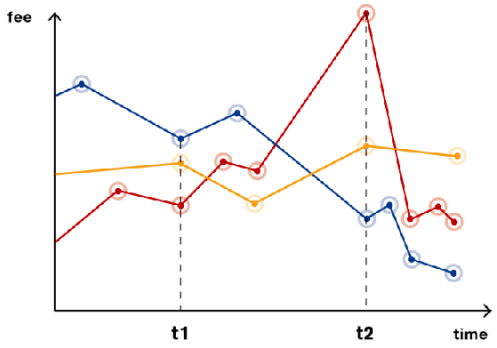

Fig. 7. Transaction fee on Blue and Red chain over time

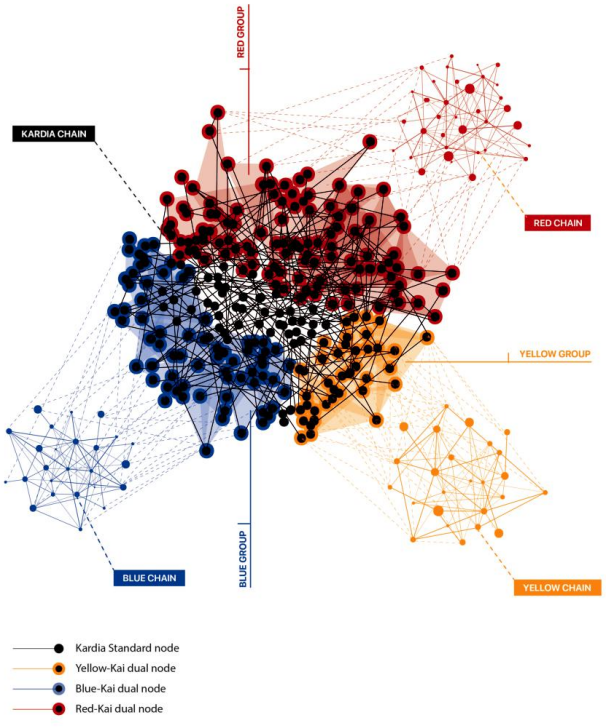

Fig. 8. Group distribution at time t1

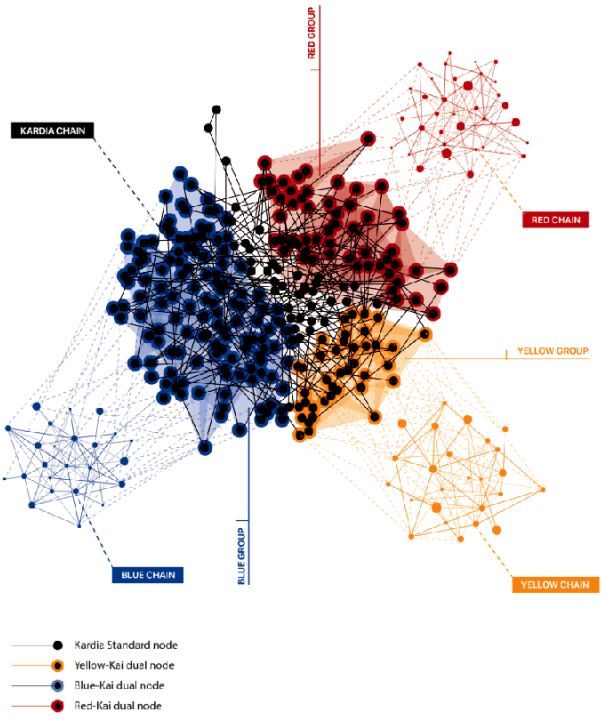

Fig. 9. Group distribution at time t2

D. Consensus Protocol

KardiaChain runs two Byzantine Fault Tolerance (BFT) [6] delegated-Proof-of-Stake (dPoS) consensuses: the Main Consensus (MCon) and the Group Consensus (GCon). MCon participants are called Main Validators (MVals), who are responsible for maintaining the ledger of KardiaChain. GCon participants are Group Validators (GVals), who validate interchain transactions and add to their respective Group ledgers.

1) Process: During GCon, a five-stage procedure takes place as follows:

- **Elect**: GVals pick a Group Proposer among them.
- **Propose**: The Group Proposer crafts a block and broadcasts to other nodes in its group. In case the Group Proposer fails to create a block in the given time, network enters next stage anyway. In addition, Group Proposer handles transaction forwarding (1).
- **Validate**: GVals receive the proposed block, start voting <Approval/Rejected/Nil> with respect to the block validity, and send their vote to others via the gossip protocol.
- **Commit**: When the Group Proposer receives 2/3 vote, it broadcasts the block to the Main Proposer (2). All dual nodes then wait for the Main block before committing.
- **Finalise**: If the block gets committed to the network, increase block height by 1 to show that the block has been finalised.

During MCon, a similar procedure takes place with the following different points:

- **Elect**: MVals pick a Main Proposer among them.

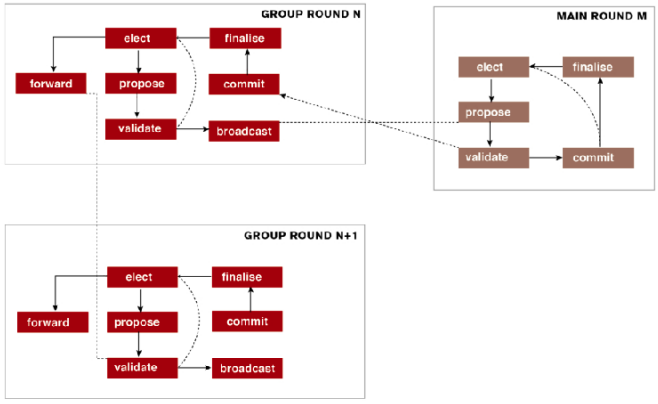

Fig. 10. Consensus Protocol

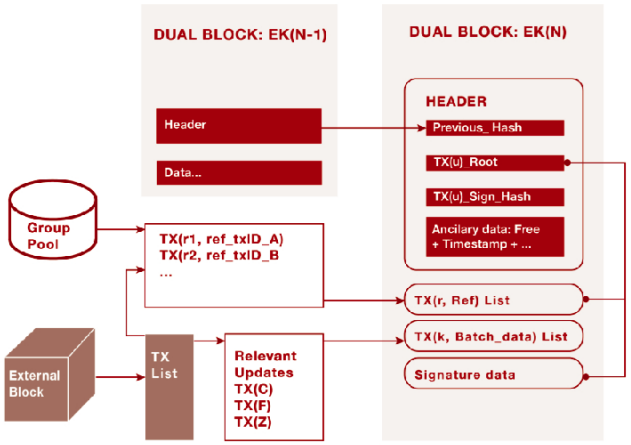

Fig. 11. Connection between Dual blocks and Main block

- **Propose**: The Main Proposer crafts a Main block from the Group blocks it received and validated.
- **Commit**: When the proposed block has received enough votes, all nodes start committing that block.

(1) Transaction forwarding: Group Proposers access their Routing TX Pool and execute pending Routing TXs that have matched the destination chain to their current supporting chain. After getting the receipts of forwarding, Proposers will move these RTXs to the Dual Pool, where Dual Nodes store Updating TXs for next block confirmation.

(2) Broadcast the committed block: After their proposed block getting 2/3 vote, Group Proposers broadcast that block to other nodes, targeting the Main Proposer, so the Main Proposer can include that block in its proposal.

2) Block Structure: KardiaChain uses a dual block structure, with 2 types of blocks: Main block and Dual block. Both types of block share common block fields such as:

- **Timestamp**: creation time of the block.
- **Height**: the length of the blockchain. Genesis block starts from 0.
- **Votes**: record of all the BFT votes for this block, including the signatures for the block.
- **PreviousHash**: hash of its parent block.
- **StateRoot**: hash of trie root, representing the global state after the block transactions are finalized.
- **GasLimit**: current gas limit for one block.
- **GasUsed**: total gas used by transactions in this block.
- **Data**: transactions data in the block. MainBlock keeps the transactions on the Kardia mainchain. DualBlock keeps the events happened on the dual group.

Dual block also maintains the state of the dual node group, such as the group of nodes that connect to Ethereum network. Meanwhile Main block keep references to each branch of dual blockchains, running on each dual node groups. The Dual block from different groups is marked with an identifier. For instance: EK for Eth-Kardia, NK for Neo-Kardia (the marking rule is not final).

Connection from Dual blocks serving one dual node group to Main block is displayed in Figure 11. Main Proposer combines the translated $TX(u)$ from all dual blocks with $TX(n)$ from standard nodes to form its TX data. Main Proposer validates TX list with Signature data gathered from dual blocks and standard nodes who submitted $TX(n)$. MP apply these TX to update its state root. The Main Block is now formed with other necessary data and MP signature.

E. Staking Model

1) Staking Model: A node operator needs to lock all Kardia Native Tokens (KAI) in their balance using *lockBalance* trans-action in order to become a Validator. Staking amount required for a Validator as Master node is $s(M)$, and as Dual Master node $s(M+D)$. KardiaChain applies the "coin age" concept to examine the effective staking amount, which means KAI token must be in an account for a number of blocks *(mature time)* before it is available for lockBalance.

KardiaChain is set up to optimise decentralisation without compromising on the security of the network. The model of KardiaChain has several lines of defence towards attackers. On one hand, mature_time helps lengthen the time required for an attack, as newly accumulated tokens cannot be used for staking. On the other hand, during mature time, the total staking amount may vary resulting in more staking token required to achieve majority. For example, the current total stake is 1,000,000 KAI, the attacker has to accumulate 2,000,001 KAI to reach $> 66\%$, due to the BFT consensus security model [15]. After mature time, some accounts join as validators and thus increase the stake by 100,000 KAI. The attacker then needs to add 200,000 KAI more and wait for another mature time to attack. Hence, nothing ensures that an attack on the network will be successful, making it less lucrative to malicious users.

An attack on dual node groups will also require control of the main network because external updates are verified by Main validators before added on Main ledger. As visualised in Figure 12, if an attacker wants to take control of a group that has 10% staking power, it requires only 7% total stake to, for example, fake an update from Yellow chain. However, for that update to take effect on the ledger of KardiaChain, it has

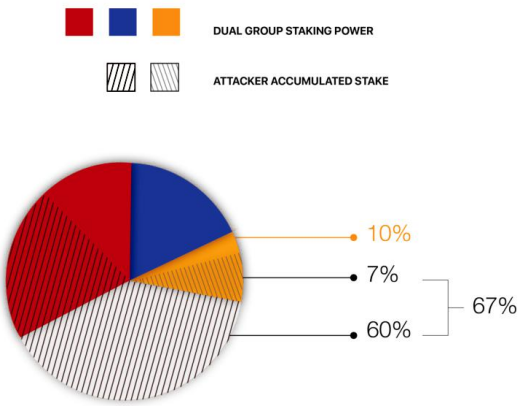

Fig. 12. Total network staking power required to take control over a small dual group

to pass the main validators' round. This means another 60% of network staking power is required to have a chance for a successful attack. In case of a failed attack, the stake amount of all nodes that voted for the malicious transactions/block is slashed (token burnt), making any attack on KardiaChain quite an expensive attempt.

2) Usage of token: The native digital cryptographically-secured utility token of KardiaChain (KAI) is a major component of the ecosystem on KardiaChain, and is designed to be used as the primary token on the network. KAI is a non-refundable functional token which will be used as the unit of exchange between participants on KardiaChain. The goal of introducing KAI is to provide a convenient and secure mode of payment and settlement between participants who interact within the ecosystem on KardiaChain. KAI does not in any way represent any shareholding, participation, right, title, or interest in the Foundation, the Distributor its affiliates, or any other company, enterprise or undertaking, nor will KAI entitle token holders to any promise of fees, dividends, revenue, profits or investment returns, and are not intended to constitute securities in Singapore or any relevant jurisdiction. Ownership of KAI carries no rights, express or implied, other than the right to use KAI as a means to enable usage of and interaction with KardiaChain, and to transfer KAI on third-party exchanges.

Computational resources are required for validation of additional blocks / information on the blockchain. Thus providers of these resources would require payment for the consumption of these resources (i.e. "mining" on KardiaChain) to maintain network integrity, and KAI will be used as the unit of exchange to quantify and pay the costs of the consumed computational resources. KardiaChain would require miners to stake a certain amount of KAI before being entitled to participate in mining. KAI is an integral and indispensable part of KardiaChain, because without KAI, there would be no incentive for users to expend resources to participate in activities or provide services for the benefit of the entire ecosystem on KardiaChain.

Users of KardiaChain and/or holders of KAI who did not actively participate will not receive any KAI incentives. In particular, you understand and accept that KAI:

- is non-refundable and cannot be exchanged for cash (or its equivalent value in any other virtual currency) or any payment obligation by the Foundation, the Distributor or any affiliate;
- does not represent or confer on the token holder any right of any form with respect to the Foundation, the Distributor (or any of its affiliates), or its revenues or assets, including without limitation any right to receive future dividends, revenue, shares, ownership right or stake, share or security, any voting, distribution, redemption, liquidation, proprietary (including all forms of intellectual property), or other financial or legal rights or equivalent rights, or intellectual property rights or any other form of participation in or relating to KardiaChain, the Founda-tion, the Distributor and/or their service providers;
- is not intended to represent any rights under a contract for differences or under any other contract the purpose or pretended purpose of which is to secure a profit or avoid a loss;
- is not intended to be a representation of money (including electronic money), security, commodity, bond, debt instrument or any other kind of financial instrument or investment;
- is not a loan to the Foundation, the Distributor or any of its affiliates, is not intended to represent a debt owed by the Foundation, the Distributor or any of its affiliates, and there is no expectation of profit; and
- does not provide the token holder with any ownership or other interest in the Foundation, the Distributor or any of its affiliates.

The contributions in the token sale will be held by the Distributor (or its affiliate) after the token sale, and contributors will have no economic or legal right over or beneficial interest in these contributions or the assets of that entity after the token sale.

The token standard underlying KAI will initially be the ERC-20 Token Standard which will then be swapped in KardiaChain native token. The ERC-20 Token Standard and the underlying token standard of the KardiaChain native token allow for KAI to be freely transferable on third-party exchanges, which might create a secondary market or exchange for trading KAI. Such third-party exchanges would be run and operated wholly independently of the Foundation, the Distributor, the sale of KAI and KardiaChain. Neither the Foundation nor the Distributor will create such secondary markets, nor will either entity act as an exchange for KAI.

V. KardiaChain Advantages

The KardiaChain team have set out the mission to alleviate an acute pain point of blockchain developers being tied to a single chain. KardiaChain allows developers to create Dapps running on different blockchains simultaneously without the need to know how to write smart contracts on those chains.

KardiaChain aims to create an inclusive network consists of private and public blockchains. In this network, a chain can interact with any other chains in the network without changing

each one's inner communication. Therefore, the scalability, interoperability and interchangeability of KardiaChain can be increased.

- **Scalability.** When a smart contract is submitted to the network of KardiaChain, it can be translated and deployed to other (specified) chains in the network. The best network will be elected, to make sure the performance achieved is the best possible at the time of submission. This is also known as offloading solution for blockchain to avoid a sudden congestion at a specific time or event.
- **Interoperability.** Via KardiaChain, all chains in the network are able to transact with data and assets on the others. Inter-chain functionality allows Dapps to specify different actions to be executed on different chains, which offload the congestion at a particular chain at specific time or event.
- **Developability.** Developers can easily switch to new blockchain technology using the smart contracts APIs on KardiaChain with their familiar tools and languages. KSML is designed to be a human-readable, ready to use language that supports most of the operations that a smart contract may involve. By composing smart contract using KSML, developers can easily deploy their solutions to any blockchain that is supported by the ecosystem of KardiaChain without any extra effort.
- **Adoptability.** Any organisation or individual can experience their very first blockchain application using the smart contracts APIs on KardiaChain, which provides a user-friendly toolkit with a comprehensive markup language to help one create, test and deploy one's very first contracts without hassle. All of the process can be done on a web-based interface without any prior environment setup required.
- **Cost.** When one sticks to a single platform, it means one has to pay fees at an unpredictable rate. On KardiaChain, users have CMNR to help select the best chain to use at any given time. Even considering the transaction fee on KardiaChain, overall cost is always lower than the average fee in a regular blockchain.

VI. The new horizon of Dapp feasibilities

KardiaChain envisions three unique Dapp features that are made possible by its native platform:

A. Simple and High traffic Dapp

Public voting should be a common example for this kind of Dapp. During a fixed period of time, the KardiaChain team expects a huge traffic volume depending on the size of a country. However, the setup process for this is quite simple. The organiser customises the voting smart contract template provided by KardiaChain and deploys on KardiaChain as Vote SMC ABC(K).

Unless otherwise specified, CMNR analyses and lists a number of suitable chains for Vote SMC ABC(K). It then deploys the translated smc to correspondent chains, such as Vote_SMC_ABC(Eth) to Ethereum.

The organiser now receives several addresses of Vote_SMC_ABC on multiple chains and can use them asthey see fit. The voters can start voting by interacting with any chains on which the Vote SMC ABC deployed. However, it is recommended to vote through KardiaChain to utilise the smart routing algorithm in order to avoid congestion.

Let's say conservatively, KardiaChain supports 2 chains (E and N) at the time of voting, each has a 2, 000tps/10s block time and 1,000tps/5s block time respectively. The native tps of KardiaChain is 100tps with block time of 20s. This provides a collective tps at 3100, which can accommodate a population of 50 million voters. Here it is assumed that all voters submit their vote within 8 hours in a single day, resulting in average usage at 1700tps.

B. Multi-function Dapp

A simple example should be Dapps like CryptoKitties [18], where every minor action (e.g. breeding, changing names) requires an onchain transaction, which is costly in terms of both time and money. It would be highly beneficial if all these minor actions were moved onto a chain with less traffic and lower fees, leaving the major actions (buy/sell kitties) on a more reliable chain. This approach also provides benefits in terms of utilising the difference in frequency: breeding may happen very often, while transferring kitties to each other is less frequent. So why do these operations have to run on the same infrastructure? In software engineering, it is considered good practice to use multiple databases with different characteristics (such as speed and consistency) for different functions.

C. Interconnected Dapps

KardiaChain provides the infrastructure for Dapps to exchange data and interact with each other securely and seamlessly. The following diagram showcases how a banking Dapp benefits from using a KYC data of another Dapp, while two are based on different blockchains (KardiaChain and ChainX). User A submitted KYC data to KYC_Dapp on KardiaChain. When user A engages with Banking Dapp on ChainX, user A can allow Banking Dapp to obtain the KYC A data from KYC Dapp_on KardiaChain. This process happens in a secure and decentralised manner, requiring reduced effort from developers and greatly enhancing user experience.

References

[1] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, 2016.

[2] D. Beaver, S. Kumar, H. C. Li, J. Sobel, P. Vajgel, "Finding a needle in Haystack: Facebook's photo storage," in *Proc. 9th USENIX Conf. Oper. Syst. Des. Implement. (OSDI),* 2010.

[3] Facebook Inc, "2018 Third Quarter Reports," 2018 [Online]. Available: https://investor.fb.com/investor-news/press-release-details/2018/Facebook-Reports-Third-Quarter-2018-Results

[4] https://www.omnicoreagency.com/youtube-statistics/

[5] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008 [Online]. Available: http://bitcoin.org/bitcoin.pdf.

[6] M. Castro, B. Liskov, "Practical Byzantine Fault Tolerance," in *Proceed-ings of the Third Symposium on Operating Systems Design and Imple-mentation*, ser. OSDI '99. Berkeley, CA, USA: USENIX Association, 1999.

[7] M.L. Abbott, M.T. Fisher, "The art of scalability: Scalable web architecture, processes, and organizations for the modern enterprise," Addison-Wesley, 2009.

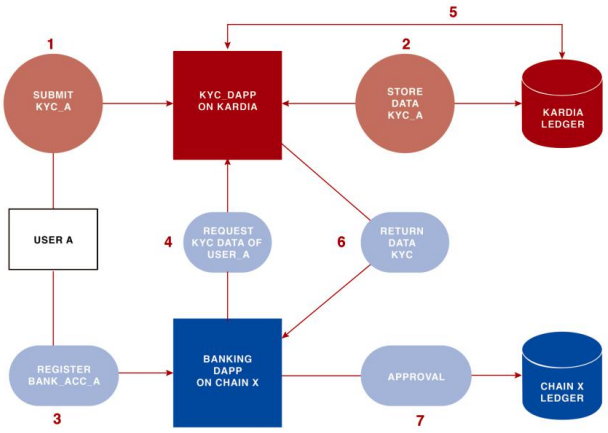

Fig. 13. Interactions between two Dapps on ChainX and KardiaChain

[8] S. Micali, K. Ohta, L. Reyzin, "Accountable-subgroup Multisigna-tures: Extended Abstract," in *Proceedings of the 8th ACM Conference on Computer and Communications Security* , ser. CCS '01. New York, USA, 2001.

[9] D. J. Bernstein, "Multi-user Schnorr Security, Revisited," in *IACR Cryptology ePrint Archive*, vol. 2015, 2015.

[10] G. Wood, "Ethereum: a Secure Decentralised Generalised Trans-action Ledger", 2014 [Onine]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[11] M. Al-Bassam, A. Sonnino, V. Buterin, "Fraud Proofs: Maximising Light Client Security and Scaling Blockchains with Dishonest Majori-ties," ArXiv:1809.09044. Available: https://arxiv.org/abs/1809.09044

[12] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timon,´ P. Wuille, "Enabling Blockchain Innovations with Pegged Sidechains," [Online]. Available: https://blockstream.com/sidechains.pdf

[13] G Verdian, P Tasca, C Paterson, G Mondelli, "Overledger White Paper," 2018. [Online]. Available https://www.quant.network/press/overledger-white-paper

[14] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework" 2017. [Online]. Available: https://polkadot.network/PolkaDotPaper.pdf

[15] N Chondros, K Kokordelis, M Roussopoulos, "On the practicality of practical Byzantine fault tolerance," in *Proceedings of the 13th International Middleware Conference*, Montreal, QC, Canada, Dec 2012.

[16] C. Brunner, D. Flore, "Generation of Pathloss and Interference Maps as SON Enabler in Deployed UMTS Networks," in *Proceedings of IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, 2009.

[17] W. LiSebastien,´ A. BohliGhassan, G. Karame, "Securing Proof-of-Stake Blockchain Protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, DPM 2017, CBT 2017.

[18] N. Bowles, "CryptoKitties, Explained ... Mostly," in *The New York Times*, Dec. 28, 2017. Available: https://www.nytimes.com/2017/12/28/style/cryptokitties-want-a-blockchain-snuggle.html