



Technical White Paper

DaVinci Labs

Version 1.0

Disclaimer Please read the entirety of this "Disclaimer" section carefully. Nothing herein constitutes legal, financial, business or tax advice and you should consult your own legal, financial, tax or other professional advisor(s) before engaging in any development and activity in connection herewith. Neither any of the project team members (the CORE team) who have worked on the DaVinci blockchain (as defined herein) or project to develop the DaVinci blockchain in any way whatsoever, any distributor/vendor of \$coretokens, nor any service provider shall be liable for any kind of direct or indirect damage or loss whatsoever which you may suffer in connection with accessing this whitepaper, the website (the Website) or any other websites or materials published by the foundation and the team. This whitepaper is subject to change without any prior notice.

1. Introduction

Since Bitcoin's creation in 2008, blockchain technology has spread worldwide. However, Bitcoin's limited speed (~7 TPS) and high costs made it inefficient as a payment system. In 2014, Ethereum introduced smart contracts but still struggled with scalability (~15 TPS), limiting high-demand applications like gaming and decentralized exchanges.

To improve scalability, many blockchains replaced Bitcoin's Proof-of-Work (PoW) with Proof-of-Stake (PoS) or Delegated PoS (DPoS). Others, like IOTA, used a different structure (DAG) to speed up transactions. However, these solutions often sacrificed security or decentralization.

Sharding emerged as a promising solution, allowing transactions to be processed in parallel. Zilliqa was the first blockchain to adopt sharding but had limitations—it

DaVinci: The Next-Gen Scalable and Secure Blockchain

DaVinci is a cutting-edge blockchain that solves scalability, security, and efficiency issues in existing systems. It integrates top research and engineering to create an optimized, high-performance network. Key innovations include:

- **Full Scalability:** Unlike Zilliqa, DaVinci shards not just transactions but also block-chain storage, ensuring full scalability.
- **Secure Sharding:** Uses a provably secure randomness process and reshards continuously to prevent attacks.
- **Fast & Energy-Efficient Consensus:** Uses PoS instead of PoW, making it 100 times faster and more energy-efficient.
- **Adaptive PoS:** Prevents stake concentration in a single shard while allowing small participants to join and earn rewards.

Efficient Networking: Uses RaptorQ and Kademlia routing for fast cross-shard transactions that scale efficiently.

Reliable Cross-Shard Transactions: Direct shard communication with atomic locking ensures transaction consistency.

By optimizing both protocol and network layers, DaVinci enables high-speed decentralized applications like large-scale exchanges, fair gaming, fast payments, and IoT transactions—scaling trust for billions worldwide.

2. Consensus Mechanism

A blockchain’s consensus protocol ensures validators agree on the next block securely and efficiently.

Proof-of-Work (PoW) (used by Bitcoin) requires miners to solve cryptographic puzzles, with the longest chain becoming the valid one. However, it is slow and energy-intensive.

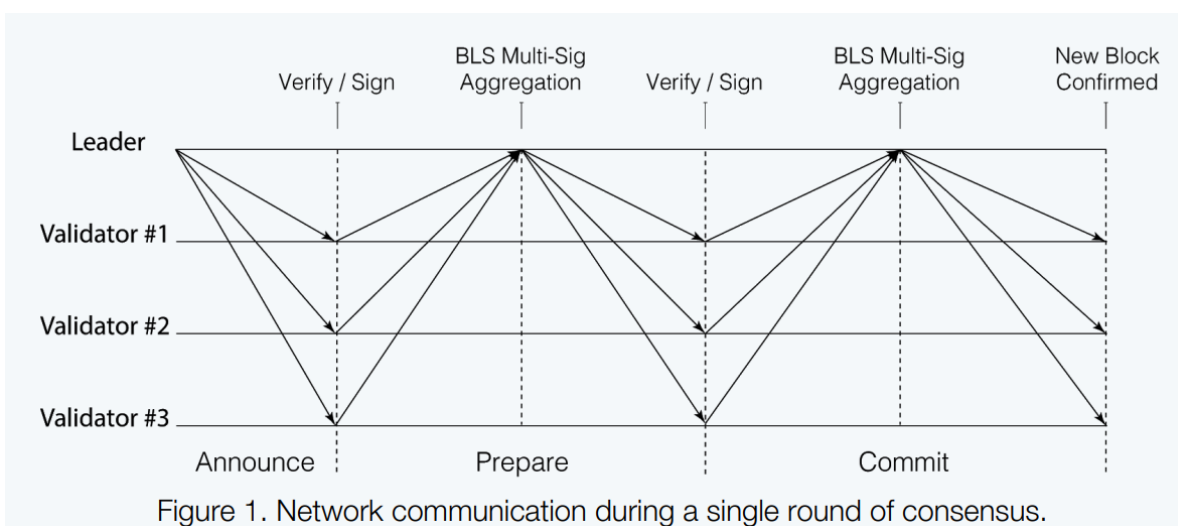
Practical Byzantine Fault Tolerance (PBFT) is another consensus method where a “leader” proposes a block, and validators vote in two phases (prepare and commit). However, PBFT requires heavy communication between validators, making it inefficient for large networks.

DaVinci’s Fast Byzantine Fault Tolerance (FBFT)

To improve scalability, DaVinci introduces FBFT, which reduces communication complexity using multi-signatures. Instead of all validators broadcasting votes, the leader collects them into a single compact signature and then broadcasts it.

- Inspired by ByzCoin’s BFT but improves efficiency using BLS multi-signatures, requiring only one round-trip instead of two.
- 50% faster than ByzCoin’s method.
- Uses RaptorQ fountain code for faster block propagation and improved security.

FBFT makes DaVinci’s consensus both fast and scalable, ensuring efficient blockchain performance.



3. Sharding

Sharding is a key solution for blockchain scalability. It divides the network into smaller groups (shards) that process transactions in parallel.

Existing Sharding Solutions

- **Zilliqa (industry):** Achieves 2,800 TPS but lacks state sharding, meaning every node must store the full blockchain.
- **Omniledger & RapidChain (academia):** Use state sharding, where each shard holds only part of the blockchain. These solutions prevent shard corruption by reshuffling nodes periodically.

DaVinci's Full Sharding Approach

DaVinci improves on previous designs by implementing a PoS-based, fully scalable, and secure sharding model:

- **Beacon Chain & Shard Chains:** The beacon chain generates secure randomness and manages identities, while shard chains store data and process transactions independently.
- **Efficient Randomness Generation:** Uses Verifiable Random Function (VRF) and Verifiable Delay Function (VDF) for secure and fair shard assignment.
- **PoS-Based Security:** Instead of relying on a minimum number of nodes, security is based on minimum voting shares to prevent attacks.

Key Sharding Features in DaVinci:

1. Distributed Randomness Generation – Ensures fair and unpredictable shard assignment.
2. Epochs – Periodic time intervals for managing network changes.
3. Staking-Based Sharding – Uses PoS to select validators for shards.
4. Resharding – Periodically reshuffles shards to enhance security.
5. Fast State Synchronization – Enables quick data updates between shards.
6. With these innovations, DaVinci ensures high-speed, secure, and scalable blockchain performance.

4. Shard Chain & Beacon Chain

4.1 Shard Chain

A shard chain is an independent blockchain that processes its own transactions and stores its own data. However, it still needs to communicate with other shards for cross-shard transactions.

Cross-Shard Communication

Cross-shard communication allows transactions between different shards. There are three methods:

1. **Main-chain-driven** (e.g., Zilliqa) – Relies on a central chain for cross-shard transactions.
2. **Client-driven** (e.g., Omniledger) – Clients handle cross-shard messaging, increasing their workload.
3. **Shard-driven** (e.g., RapidChain) – Nodes within shards communicate directly without external help.

DaVinci's Approach: Uses the shard-driven method for efficiency and ease of use. To reduce network costs, DaVinci implements:

- Kademlia routing, which lowers communication complexity from $O(N)$ to $O(\log(N))$.
- Erasure coding, ensuring reliable message delivery.

4.2 Beacon Chain

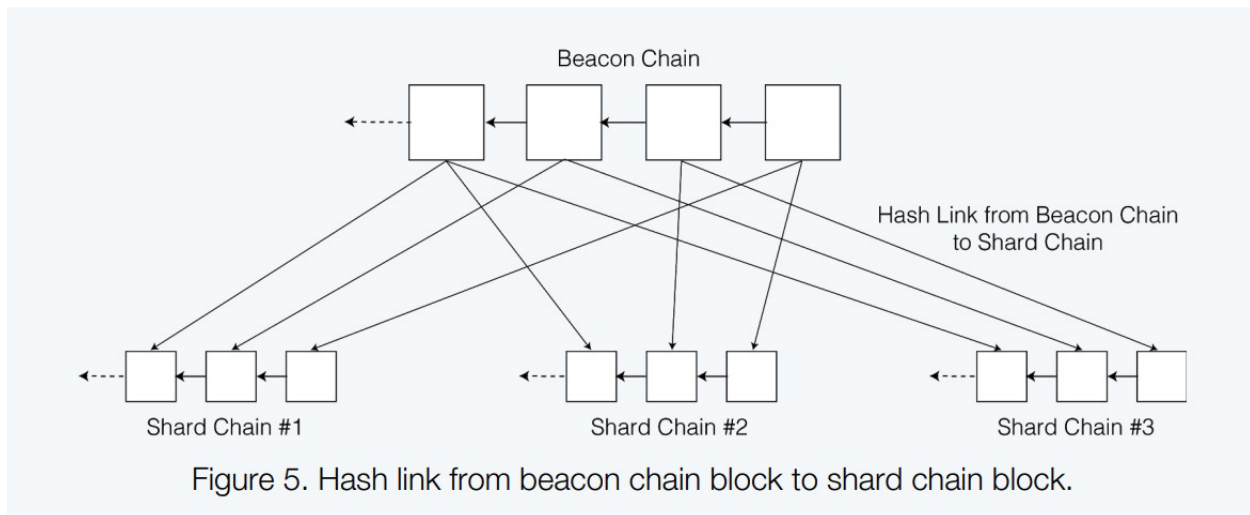
The beacon chain is a special shard responsible for:

- **Generating randomness** for secure shard assignments.
- **Managing staking**, where validators deposit tokens to participate.

Validators for the beacon chain are selected similarly to shard chains, with extra slots allocated for it during the sharding process.

By integrating efficient cross-shard communication and a secure beacon chain, DaVinci ensures smooth, scalable blockchain operations.

4.3 Hash Link from Shard Chain



The beacon chain helps strengthen the security and consistency of the shard chains' states by including the block header from each shard chain. Specifically, after a new block is committed to a shard chain, its block header will be sent (via Kademlia-based inter-shard communication) to the beacon chain. The beacon chain checks the validity of the block header by:

1. The hash of its previous block, which must have already been committed in the beacon chain;
2. The signers of the block's multi-signature, which must be the correct validators for that shard.

The committed block headers at the beacon chain will then be broadcasted to the whole network. Each shard will keep a chain of valid block headers for all other shards, which will be used to check the validity of transactions from other shards (i.e. simple payment verification). Adding the shard chains' block headers into the beacon chain serves two main purposes:

1. Increases the difficulty of attacking a single shard. Attackers have to corrupt both the shard chain and beacon chain in order to convince others that an alternative block in the shard chain is valid.
2. Reduce the network cost of broadcasting the block headers among shards. There will be a $O(N)$ network communication if we let each shard broadcast its headers separately. With the beacon chain as a central relay, the complexity is reduced to $O(N)$.

5. Blockchain State Sharding

Unlike other state-sharding blockchains [7,8] that adopted UTXO (Unspent Transaction Output) data model, DaVinci's state sharding is applied on account-based data model. Each shard chain contains its own account state, and all the tokens in existence are spread among all the shard.

We treat the user account and the smart contract account differently in sharding. An user account can have multiple balances at different shards (e.g. 100 tokens at Shard A and 50 tokens at Shard B). A user account can move its balance between shards by issuing a cross-shard transaction. A smart contract account is limited to the specific shard where the contract was created. However, for a decentralized application that requires more throughput than a single shard can handle, the Dapp (Decentralized Application) developer can instantiate multiple instances of the same smart contract in different shards and let each instance handle a subset of the incoming traffic. Note that the different instances of the same smart contract do not share the same state, but they can talk to each other via cross-shard communication.



6. Incentive Model

6.1 Consensus Rewards

After the successful commitment of a block, a protocol-defined number of new tokens will be rewarded to all validators who signed the block in proportion to their voting shares. The transaction fees are rewarded to validators similarly.

6.2 Stake Slashing

For any misbehavior detected by the network, a certain amount of staked tokens will be slashed. For example, if a leader failed to finish the consensus process and triggered the leader change process, P staked tokens will be slashed. If validators are proven to sign a dishonest block, all votes of their stake under the same shard will be slashed. This severe punishment is meant to strongly discourage any dishonest behavior and make the network as secure as possible. A proof of misbehavior can be two signed blocks that conflict with each other. Any validator can submit a transaction to prove the misbehavior of another validator and if verified, the slashed token will be rewarded to the prover(s).

6.3 Stake withdrawal

Long-range Attacks

Proof-of-stake blockchains, unlike proof-of-work blockchains, tend to suffer from long-range attacks. These are attacks that leverage the fact that proofs are based on signatures rather than on resource-intensive tasks. In a long-range attack, the private keys of honest validators are stolen long after they have been used, and the attacker is able to create a forked blockchain by signing fake blocks with those keys. When this happens, new validators joining the network have no way to distinguish between the original, legitimate chain and the attacker's simulated chain. Long-range attacks happen in the following two scenarios. Private keys can be compromised either by a lack of security on validators, or more commonly, by the fact, after a validator withdraws their token, he could financially benefit if an attacker which would be looking to buy its private key. Also, by design each set of validators is trusted to approve the block of transactions that also determines the next set of validators. After enough private keys (i.e. those that collectively hold more than voting shares in a shard) have been compromised, an attacker has total control on 3 2 who the subsequent validators are.

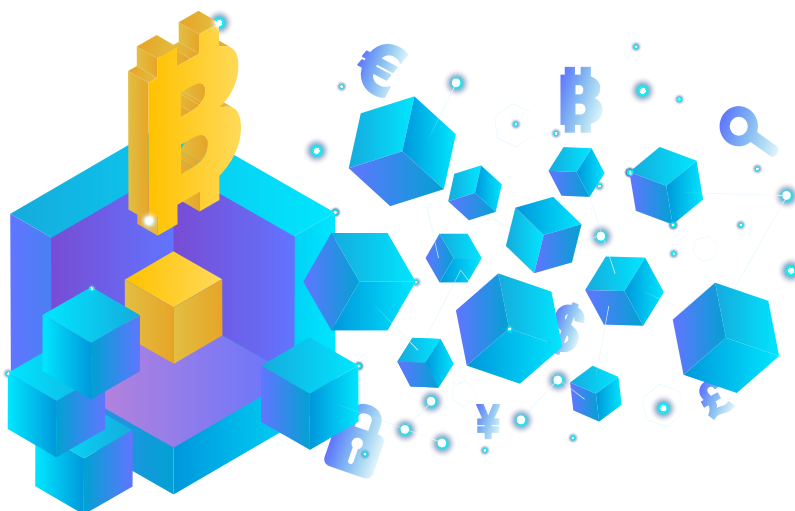
Long-range Defense:

Resonant Quorums

Proof-of-work blockchain protects against the above attacks by giving honest validators an objective method of fork choice. In a proof-of-work blockchain, the fork choice to select the canonical chain is the accumulated amount of work done in terms of hashes computed. In a proof-of-stake blockchain, the only objective measure that can be used to select between forks is the total weighting of signatures used to approve each block. If we use these weighted signatures to compare two different blocks, we come to the following equation to determine when a chain may be forked:

Safety = “Block approval key weight” - “Compromised key weight”

The “Block approval key weight” means the voting power of the keys that are signed on the block. If, by stake weight, more private keys are compromised than were used to approve a block, then the block can be forked. Until then, validators will always prefer the original, legitimate version of the block. DaVinci maximizes the safety of each block in its proof-of-stake blockchain by maximizing this equation. It is infeasible to disincentivize leaking private keys in the long term. DaVinci instead incentivizes validators to maximize the approval weight of each block after a quorum has been achieved. This is done by requiring validators to sign each quorum-approved block before allowing those validators to withdraw their stake. These new additional signatures only need to exist within the blockchain, and they do not need to be generated at consensus time for each block. Because of this, the new signatures can be added to subsequent blocks when validators decide to withdraw their stake, and so they may freely improve the safety of the chain without impacting its liveness.



7. Decentralized Apps (dApps)


Since DaVinci is using WebAssembly, it's opening a new corridor for Decentralized App developers, and DeFi applications by allowing them to write smart contracts with their favorite programming language. DaVinci is taking the initiative to help and grow the WASM smart contract developers community and as such has 10% of the total supply of CORE tokens designated for grants to developers.

8. Use Cases

DaVinci provides developers and financial institutions with a complete essential infrastructure to build any DeFi applications. Moreover, DaVinci will incentivize qualified developers to build intuitive dApps. Some of the proposed use-cases of DaVinci blockchain are:

- Tokenized Securities
- Liquidity Providers (LPs) and Market Makers
- Cross-border Payments, Banking and Remittance (e.g. Swift)
- Stablecoin Ecosystems (e.g. USDC, USDT, ...)
- Lending Platforms (e.g. Block, Nexo)
- Wrapped Cryptocurrencies (e.g. ERC20, BEP20)
- Decentralized Exchanges (e.g. Sologenic.org, UniSwap, ...)
- Metaverse Applications (e.g. Decentraland, The Sandbox, Meta)
- NFT Marketplaces (e.g. Sologenic.org, Oopensea.io, ...)
- Gaming and Play-to-earn apps (e.g. Axie Infinity)
- EGB (ISO 20022) compatibility layer

9. Coin/Token Economy

BlockChain Name	DaVinci Protocol
Coin/Token Symbol	DCoin
Initial Supply	100.000.000
Icon/Logo	

10. Coin/Token Allocation

Funds	Percentage	Allocation	Proportion	Vesting Period/ Distribution Plan
Community	70%	SOLO Community 20% 1 - 12 months	20%	1 - 12 months Distribution Schedule
		CORE Community 30% 12 - 36 months	30%	12 - 36 months Distribution Schedule
		Validators' 10% Unlocked Rewards	10%	Unlocked
		Community d'Ap 10% Unlocked Developers	10%	Unlocked
Operation	30%	DaVinci Maintenance, Operations, Vesting Period Developers, Teams and Investors	20%	1 - 24 months Vesting Period

