Cubics Blockchain Whitepaper

A lightweight & scalable blockchain for NFT, Gaming and Metaverse applications

Version 0.8.1
9th May, 2022
https://cubics.com

Cubics Blockchain

A Serverless Blockchain for Metaverse, Gaming, and NFT applications

Cubics is a serverless layer-1 blockchain that provides infinite scalability, high throughput, sub-second confirmation times, and fees at a tenth of a cent. Cubics achieves this by leveraging serverless compute and storage cloud services while innovating incentive structures and extending the Byzantine Fault Tolerance consensus mechanism for scalability.

Introduction

We have witnessed a rapid increase in Metaverse, Gaming, and NFT adoption during the recent year. The increased demand brought infrastructural challenges, considering these applications' throughput, fee, and latency requirements. Current solutions such as Bitcoin, Ethereum, and more recently, Binance Smart Chain have shown to be not viable enough as long-term candidates for these rapidly expanding industries: the transaction confirmation times are too slow, and the fees are too high. These issues originate from the inherently nonoptimal economic incentives layer, which is the most challenging part of adjusting once implemented. Misalignments in incentives are precisely why Bitcoin, Ethereum, or BSC nodes are not willing to charge lower fees or process transactions faster.

Cubics is a lightweight and serverless solution that solves the incentive problems by incentivizing validators to compete for speed and fees. Cubics builds on top of available serverless infrastructure services offered by all major cloud providers, leveraging the simplicity, predictability, and (theoretically) infinite scalability of storage and compute. Rather than spending significant portions of time, focus, and development resources on innovating distributed compute and storage, Cubics wants to build upon proven serverless offerings from major cloud providers and decide to focus on what matters most to Cubics's mission: the advancement of Metaverse, Gaming and NFT adoption at the interface level, with the goal to onboard 100 million users.

To do that, Cubics provides a lightweight but highly scalable blockchain with native pre-written and extensively tested "programs" that asset-pools can use. These programs currently include ten common decentralized finance and gaming functionalities such as auctions, fee delegation, launch pads/fundraising, locking/vesting, loot boxes/drop zones, lottery/airdrops, royalty/assets with native yield, staking, swapping/decentralized trading, and voting.

Serverless Architecture

We propose a new serverless blockchain architecture that focuses on user benefits such as fast transaction confirmation times, low fees (around $0.001), and theoretically infinite data scalability for the demands of Gaming, Metaverse, and Web 3.0 applications (easily in the TB/PB range) by leveraging serverless cloud services.

Cubics's node software can be executed on any machine in theory. However, we see an enormous opportunity in serverless offerings by major cloud providers, which allows tapping into proven, stable, and theoretically infinitely scalable compute and storage services while providing the most demanded benefits to end-users (fast confirmation times, low fees, and integrity/verifiability of transactions).

Serverless technologies by common cloud platforms such as AWS, Azure, GCP, Alibaba Cloud were early in 2019/2020. Today, however, we can label these as fully mature and production-grade choices. Furthermore, many content delivery networks such as Cloudflare and Fastly also provide edge-compute capabilities, which have improved to the point of supporting the demands of high-throughput financial transfers.

The level of maturity, the ease of access/setup, and linear cost models are reasons for Cubics to focus exclusively on the serverless stack through offerings by the cloud platforms mentioned above.

For compute functionality, this encompasses building on AWS Lambda, Azure Functions, Google Cloud Functions, Alibaba Function Compute, Cloudflare Workers, and Fastly Compute@Edge. Storage leverages Amazon Quantum Ledger DB, Amazon DynamoDB, Azure CosmosDB, Google Firestore, Alibaba Tablestore, and Cloudflare Workers KV.

Cubics aims to eventually allow anyone to run confirmation nodes in just two steps, 1. by providing programmatic API keys from either of these platforms and 2. to subscribe to the transactions mempool (a high throughput topic service responsible for ordering and broadcasting transactions). Hence, we hope to democratize opportunities to run nodes, to provide work for the Cubics network while getting compensated in a way that makes sense economically for individual entities.

Scalability

Serverless infrastructure can scale depending on the limits of underlying technologies as offered by cloud providers. In Cubics's case, we reached a peak of 80k TPS with our current setup. The storage and compute capabilities are theoretically limitless (practically, limited to node providers' account limits or global limits enforced by cloud providers due to hardware constraints).

Most serverless compute offerings can support hundreds of thousands of invocations per second, while storage offerings can store trillions of records at a petabyte-scale. To participate in the validation of transactions on Cubics, one only requires to know about the last confirmed transaction affecting the form/to balance, which means that the minimum requirement for joining as a node is extremely low (there is no need to store the entire ledger nor to keep the state of all balances).

Cubics Benefits

- Sub-second confirmation times for transactions
- Minimal fees (approx. $0.001 per transaction for simple transfers)
- Infinite scalability and storage (with linear cost)
- An adjusted Byzantine Fault Tolerance consensus mechanism
- Proof of Audit mechanism for integrity and verifiability of transactions
- Fee delegation (owners can choose to pay for fees for users' transactions)
- Pools & programs tailored and optimized for Gaming/Metaverse/Web 3.0 use-cases
- Native read-only interfaces (APIs, frontends) for clients

Permissionless Validation

Proof of Audit

"It is possible to verify payments without running a full network node." Cubics allows any client/node to validate transactions and balances (states) at any given time. With each transaction, Cubics nodes broadcast the resulting state (new to and from balances), representing their vote about the outcome.

The updated state offers clients a running balance which provides an easy way to audit transaction histories without knowing about all prior transactions or global balance states. If clients don't trust publishing nodes with transactions (or need to perform audits for their own reasons), they can openly confirm balances as many steps back as they require based on their desired confidence level.

The simplest audit that a client can perform is requesting the two prior transactions affecting from/to addresses of the transaction to be verified. Clients can confirm that a certain "UTXO" existed and was correctly reflected by the new transaction. This functionality can be performed through a public API and interface provided by Cubics read-only nodes so that anyone can participate and make use of this validation methodology when desired.

Blockless Ledger

Cubics nodes validate and process every transaction individually, rather than agreeing on the validity of a group of transactions within a block. In practice, this means Cubics is blockless or that every transaction can be treated as a block of one. This allows Cubics to produce a continuous data structure of transactions (public ledger) and states (public balances). The benefit for applications is the easy access of transaction history and state of balances at any given point in time while being able to verify the integrity of each transaction (proof of audit).

Cubics Consensus

Cubics Byzantine Fault Tolerance

Cubics builds on top of the original BFT mechanism proposed by Lamport, Shostak, and Pease and the Practical Byzantine Fault Tolerance (PBFT) approach proposed by Castro and Liskov. We propose a modified mechanism following four progressive rounds for every transaction: proposal, voting, quorum, synchronization.

1. Proposal

Any node automatically becomes a leader if it submits the first confirmation for a transaction (resulting balances and reflected state). N Nodes propose their solution to the coordinator client, which registers the first confirming node as leader/authority for the transaction to be confirmed.

2. Voting

All subsequent votes (within the acceptable voting window of 1500ms after the transaction has been broadcasted) confirm the leader's vote or object in case their proposed state differs from the leaders' solution. Once confirmations of all participating nodes arrive or the window of 1500ms expires, the voting is considered complete.

3.1 Quorum (total agreement)

Usually, transactions are confirmed without conflicts, meaning all nodes confirm the same states as the leader node. In this case, the transaction fee is distributed to all participating nodes. The leader receives at least 10%, while all other nodes share the fee reminder proportionally.

3.2 Quorum (partial agreement)

A transaction is also considered confirmed if the weight of confirmations vs. objections is at least 4/5. The weight of each vote is a function of the staked amount by voting nodes multiplied by a factor for length of operation. The node(s) who objected (with <1/5 weight) are deducted a penalty of 0.1-Objection Weight/500 of their staked amount, which is distributed at synchronization step among the then-majority.

3.3 Quorum (non-agreement)

If a vote does not reach 4/5 voting weight, it is considered unconfirmed and is rebroadcasted by the coordinator node so that all nodes can resubmit their votes. The same penalty of 0.1-Objection Weight/500 is applied on every round of rebroadcasting to all nodes who object until an agreement (4/5 majority) is reached. Voting nodes can decide to change their initial vote or drop out (by themselves or when staking requirements fall below minimum).

4. Synchronization

After a transaction has been confirmed, it is pushed to all nodes, which might update their ledgers based on the outcome (if they have not voted correctly already, or if they are nodes interested in the partition range of the transaction). If a transaction has amassed a penalty balance (due to objecting nodes), it is distributed to the majority (which could flip during voting/rebroadcasting, so penalties are not distributed at every voting round but only after final confirmation).

Validator Staking Requirements

Nodes who want to participate in the confirmation of transactions have to stake at least 1,000,000 CUBIC. Our current fee structure allows approximately 50 nodes to participate economically while financing the underlying infrastructure and making a profit.

Cubics BFT Conclusion

Cubics BFT incentivizes high throughput, low latency, and energy efficiency (unlike PoW mechanisms). Additionally, it guarantees the finality of transactions after the self-imposed maximum confirmation window of 1500ms closes. Lastly, this mechanism ensures that the network functions under a certain amount of faulty nodes (whether unintentionally by being out of sync with the current state or by their intention to harm the network).

Mempool & Nodes

Cubics Mempool

Double spending is the biggest problem for distributed networks like Bitcoin or Ethereum. A "double spend" occurs when transactions arrive in different orders on different nodes. Think about a transaction, sending 10 CUBIC from wallet A to B, and another transaction sending 10 CUBIC from wallet A to C. Assuming wallet A has only 10 CUBIC, one of these two transactions must fail by default. However, hypothetically, if both transactions are submitted to different nodes in different orders, a double spend occurs (assuming nodes don't coordinate between each other - which is precisely what consensus mechanisms are designed to do).

The Cubics Mempool is a global pub/sub system for all incoming transactions and is responsible for broadcasting these transactions to all subscribing pools. It provides an agreed-upon ordering for messages, which solves the double-spending problem. Additionally, the Cubics Mempool involves multiple fallback mechanisms to avoid a single point of failure.

The current limitations of our implementation are 80k TPS and that no prioritization of transactions is possible. Additionally, transactions cannot be reshuffled or moved ahead by paying a higher fee. These attributes are intentionally waived in favor of efficient one-way sorting with higher throughput.

TLDR: The Cubics Mempool is a high throughput pub/sub mechanism that ensures the sorting of transactions by order of arrival and broadcasts them to subscribing nodes that process transactions, vote on transaction outcomes, synchronize their state-ledgers, and submit confirmations.

Mempool Subscribers

The Cubics Mempool receives all incoming transactions and broadcasts them to subscribing nodes. Nodes can listen selectively based on their requirements and preferences (e.g., nodes might want to listen only to transactions with from-address ranges 1-9a-z and to-address ranges from A-Z). This is a purely illustrative example based on the base58 encoding range of addresses.

Nodes subscribe to the Mempool, which broadcasts all incoming transactions sequentially ordered by received-at-timestamp. Messages trigger serverless compute functions, which execute modules and programs (either simple transfer, or pre-programmed functions like auctions, staking, swapping, voting, etc.) and update states of balances. Once nodes complete the computation and state reflection internally, they broadcast the result to the coordinator node and are rewarded a portion of the transaction fee upon success.

Node Synchronization

Nodes subscribe to the Cubics Mempool to receive incoming transactions and update them locally. Depending on different node types (read-only API node or

confirmation node), nodes can participate in receiving all transactions or listen selectively and filter for specific segments. Nodes keep a local ledger and submit confirmations to balance updates they believe are correct (to participate in the transaction fee payout).

Additionally, nodes consume confirmed transactions, which allows them to update the resulting to & from states locally without having kept a prior ledger. Nodes keep track of transactions internally via an incrementing number, allowing them to stay in sync and detect missed transactions. In the case of discrepancies between a local index and the one of a last broadcasted transaction, nodes can resync and replay transactions they have missed.

TLDR: Cubics provides a single data structure of all records, including all states over time. This allows nodes to sync and validate the integrity of transactions at any point in time without knowledge about prior transactions or the state of balances.

Node Types

Cubics has four node types: coordinator nodes, validator nodes, storage nodes, audit nodes.

Coordinator nodes are mediators over the voting process, which receive votes from validator nodes and submit confirmations/rebroadcast transactions, dependent on different quorum outcomes. Every validator node can become a coordinator node if agreed upon by 4/5th of the majority.

Validator nodes subscribe to transactions and store a subset of transactions (depending on their preferences or limitations, e.g., different infrastructure limits imposed by certain cloud providers). Validator nodes then execute the transaction instructions and programs, reflect state locally, and submit the reflected state as their vote. The reward for validator nodes is either 0.9/(N-1)*fee if they confirm the authority/leader node or 10% if they are the first submitting node.

Storage nodes subscribe to all transactions and store them on serverless storage offerings for hypothetically infinite scalability. Storage nodes provide clients with the ability to access historical balances/transactions.

Audit nodes perform audits of transactions and balances by requesting historical data from storage nodes and confirming balances. Audits can be performed as many transactions back as necessary and required by applications (up to the genesis transaction of each asset). Clients can perform audits in their browsers by requesting the required data from storage nodes.

Risks & Attack Vectors

51% Attack

Attackers are guaranteed to be penalized a portion of their stake with every false vote they submit as confirmation. Confirming nodes have a staked amount (minimum 1,000,000 CUBIC) which is weighted by their length of operation. The network is secure as long as honest nodes control the majority of the total voting stake.

Sybil Attack

Voting weight is calculated based on the staked amount and adjusted by a factor for the length of operation. The length of operation serves as a form of reputation which is impossible to fake. Cubics accounts for vote weight rather than vote counts. Hence, the number of nodes and whether owned by the same entity or different entities do not matter.

Spam Attack

Attackers could spam the network with valid or invalid transactions, affecting bandwidth (best case) or bandwidth and compute resources of nodes. The Cubics Mempool runs sophisticated load balancing and protection software to ensure maximal throughput of honest transactions while throttling bad actors (or clients sending transactions with false markup leading to frequent invalid transactions).

Free-Riding

Transaction states are broadcasted only after the confirmation is closed and finalized. Therefore nodes can not rebroadcast existing confirmations to collect fees without providing work.

Economics

Fee Incentives

All confirming nodes receive portions of the transaction fee for every successfully confirmed transaction. The leader (proposer of a transaction) is rewarded 10%, and all other nodes are rewarded 0.9/(N-1). A certain number N exists, which is the number of nodes at which the current static transaction fees provide economic reasons to operate a node. This number is approximately 50. Hence up to 50 nodes could participate economically in the voting process of each transaction.

The authority/leader node (first node to confirm a transaction and propose the resulting state) receives at least 10% or 0.9/(N-1)*fee, whichever is larger. This simple but effective fee structure allows incentivization that is meaningful to the user by fostering competition between nodes (speed of confirmation, the truthfulness of confirmations).

The proposed reward mechanism solves the collective action problems of currently popular networks, mainly the free-rider problem, which originates from misaligned incentives between nodes and clients. Ethereum incentivization discourages providing infrastructure, hence such a dependence on the heavily subsidized and non-economic Infura. Bitcoin discourages confirmation speed and low fees since volunteers are paying for the network infrastructure.

TLDR: Cubics's fee structure incentivizes nodes to behave truthfully while contributing efficient infrastructure to the network. This solves collective action problems as experienced by Bitcoin and Ethereum.

Pools & Programs

Programs replacing EVM

Cubics prioritizes client benefits (most notably confirmation speed and low fees) rather than providing a virtualized compute platform that executes bytecode. Cubics covers major use-case programs such as locking, staking, trading, voting, etc., which find importance in applications for Metaverse, Gaming, and NFTs.

Focusing on a few programs rather than providing full EVM capabilities results in faster development and easier testing. Cubics provides a limited number of pre-written programs to creators who are building on top of Cubics. Creators often are not familiar with languages like Solidity and would rather spend time implementing features than writing and testing the underlying smart contracts.

Lastly, each of Cubics's pre-written programs goes through extensive testing and becomes more trustworthy with every transaction processed. This is the opposite of Ethereum/BSC, where every smart contract has to be audited with each new deployment due to the possibility of change or error.

TLDR: Cubics provides native programs instead of EVM functionality to improve development speed, guarantee ease of access to creators/users, and ultimately, focus on what matters most to Cubics's mission: the adoption of Metaverse/Gaming/NFT applications.

About

Cubics is a Metaverse company focused on bridging blockchain, vision AI, and 3D technology. Specifically, Cubics focuses on NFT, Gaming, and Metaverse applications to accelerate user adoption in the coming years ahead. Let's look at each of Cubics' focus areas to give you a better understanding of Cubics' goals and vision.

Cubics' first goal is to promote blockchain adoption by enabling high-throughput transfers at an extremely low cost. Cubics released a serverless blockchain in 2021, which enables up to 80000 TPS, network fees in the range of a tenth of a cent, and confirmation times below one second. With its innovation in incentive structures for nodes and an improved consensus mechanism, the Cubics blockchain is a crucial piece of technology to scale the next generation of Metaverse applications.

Cubics' second goal is innovation within vision AI and 3D technology. Specifically, Cubics works on avatar technology, 3D scanning, and virtualization apps, as well as real-time reconstruction of faces, bodies, clothing, and shapes of all types. The mission of Cubics is to provide tools to creators, builders, and users and impactfully shape the future of the Metaverse.

Mission

Cubics's mission is to onboard 100 million users to the Metaverse by developing intuitive applications & interfaces for mainstream adoption.

Focus Areas

The Cubics Blockchain focuses on promoting Metaverse, Gaming, and NFT adoption using blockchain (asset registry, virtualization, dApps) and vision AI (object detection/segmentation). The six core pillars of The Cubics Blockchain are:

Blockchain

Cubics Blockchain is a serverless blockchain enabling up to 80,000 TPS, transaction finality in under one second, and fees as low as a tenth of a cent. It provides a non-Turing complete language and 11 pre-built smart contracts, developed explicitly for NFTs, Gaming, and Metaverse use-cases such as Auctions, Lending, Staking, Decentralized Swaps, Lotteries and more.

Metaverse

Cubics focuses on developing mixed-reality spaces as interface-layer for popular blockchains and, of course, Cubics's own blockchain. Primarily, Cubics creates a Minecraft-like virtual world with a focus on avatars, virtual land ownership, tradeable in-game NFTs, virtual spaces and events, 3D auctions & exhibitions, and Multiverse interoperability.

Creators & Community

Cubics provides tools for creators and users to simplify token and NFT launches through launch pads and various DeFi mechanisms such as staking, lotteries, NFT royalties, and lending pools. Additionally, Cubics enables creators to collaboratively build, play and interact within the Metaverse using only a web browser.

Tokenomics

Currently, there are 1 billion CUBIC tokens in circulation. The total circulation will be 10 billion CUBIC tokens. The total token circulation of 10 billion CUBIC tokens will be reached approximately by 2027. 1-1.5 billion CUBIC tokens will be in circulation on Ethereum and Binance Smart Chain (BSC), while 8.5-9B will be in circulation on the Cubics native blockchain. Bridges will make it possible to swap between these blockchains seamlessly.

Allocations: 20% ecosystem & partnerships, 20% community enrichment, 15% foundation, 10% legal, marketing & biz dev, 10% future financing rounds & OTC, 10% team & development, 10% public liquidity, 5% advisors & contributors.

Roadmap & Milestones

2021 Q3/Q4

Blockchain Development: A lightweight modular blockchain to support asset tokenization and transactions with sub-second latencies.

Block explorer & smart contracts: A block explorer, a wallet web app, and a dashboard for interacting with the Cubics Blockchain. Simplification of minting NFTs & creating tokens - enabled by a no-code interface.

Testnet & 11 Prewritten Smart Contracts on Cubics: Launching the Cubics Blockchain Testnet and 11 pre-written programs using Cubics' own non-Turing complete smart contract language.

2022 Q1/Q2

Cubics Mainnet & Bridge: Launching the Cubics Blockchain Mainnet and developing a cross-chain bridge to enable swaps between CUBIC-ETH-BSC.

DEXs, NFTs, Staking & Lending Launch: Launching common pools such as decentralized swaps, staking pools, lending pools, auction pools, etc on the Cubics Blockchain. Enabling any project using Cubics Blockchain to create such pools for their communities. Enabling creators to easily create and upload NFTs to the Cubics Blockchain.

Metaverse Development & Land Sale: Developing an alpha version of the Metaverse. Bringing together Cubics Blockchain (land registry), 3D graphics, and our underlying APIs (storage) to virtualize and trade virtual land.

2022 Q3/Q4

Metaverse Usability & Scaling: Combining the Cubics Blockchain (and other chains) with the Metaverse, focusing on usability and interoperability between blockchains applied to the Metaverse.

Avatars & Virtual Spaces: In-game avatars, tradable skins, and NFT clothing. Focusing on growing use-cases for the Metaverse, towards virtual spaces, hangouts, gaming applications, events, and other virtual activities.

Gamification & Creator Tools: Providing tools for users and creators to increase adoption of the Metaverse. Tools might include simple gameplay creation tools and one-click setups of 3D spaces for various use-cases.

2023 Q1/Q2

Multiverse Interoperability: Focusing on interoperability between reality and various Metaverses. Working on AI technology to enable object detection, segmentation, and reconstruction enabling scanning and importing objects from the real world into the Metaverse.

Team Expansion & Growth: Hiring additional team members to promote the Cubics blockchain, to onboard users and creators, to raise awareness and educate around the Cubics Blockchain and Metaverse.

Cubics Ecosystem Development: Expanding business development to establish partnerships and to implement Cubics Blockchain & Cubics Metaverse technology.

Resources

https://www.theblockresearch.com/the-burden-of-infura-6899

https://bitcoin.org/bitcoin.pdf

https://www.microsoft.com/en-us/research/uploads/prod/2016/12/The-Byzantine-Generals-Problem.pdf

https://arxiv.org/pdf/1803.05069.pdf

http://pmg.csail.mit.edu/papers/osdi99.pdf

https://vldb2020.org/assets/files/tutorial1_part2_system.pdf

https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf