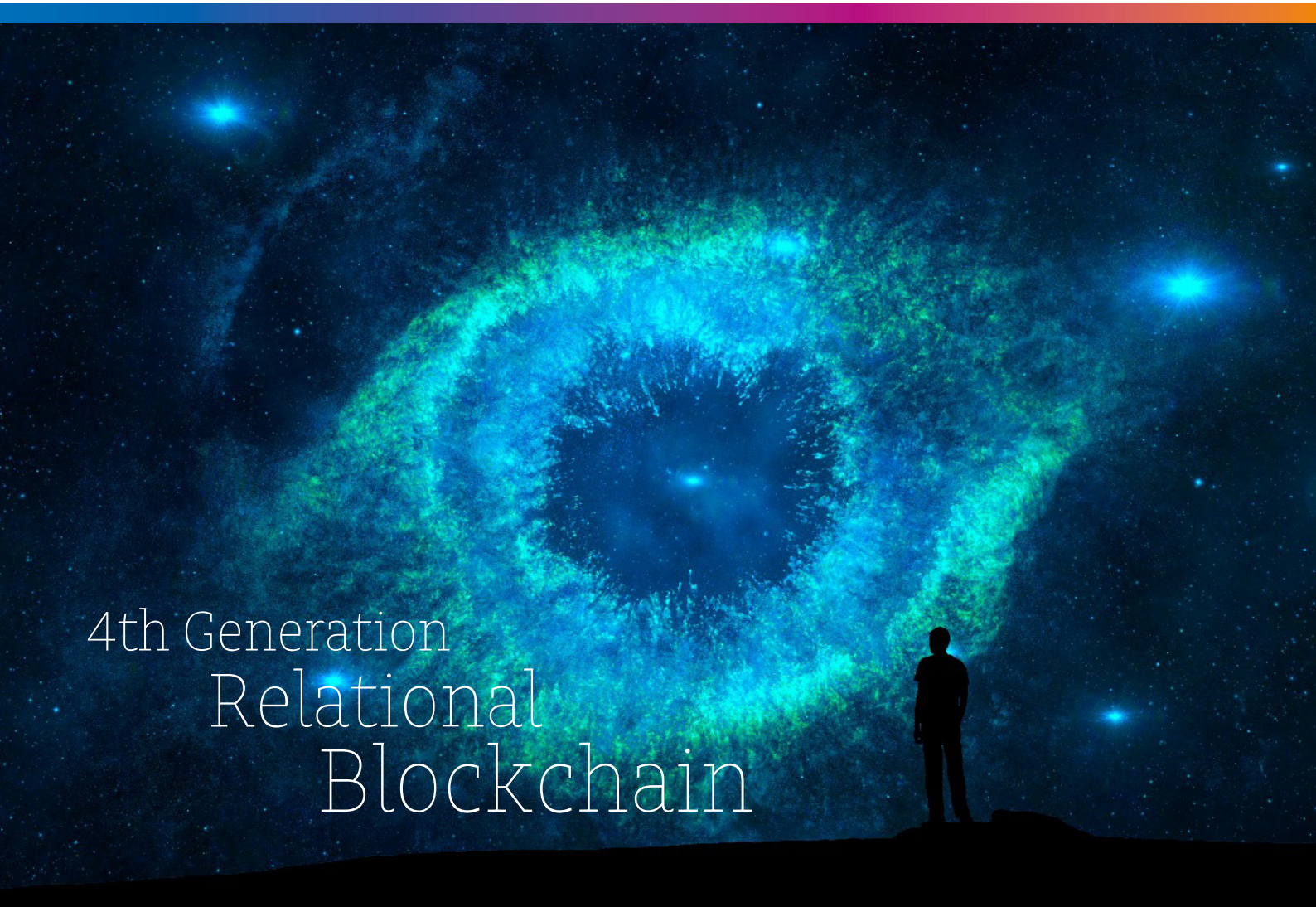
MULTIVERSUM
BLOCKCHAIN

WHITE PAPER v 2.0
Business | Technical
English

25.05.2018
Authors: Multiversum Team
www.multiversum.io



There are innumerable universes besides this one, and although they are unlimitedly large, they move about like atoms in You.

Bhagavata Purana 6.16.37

Multiversum Identity and Mission

The pioneering cryptocurrency, Bitcoin, together with all the various reinterations, modifications, clones and forks based on Proof of Work algorithm for transaction validation, are considered first-generation blockchains. The second generation, with Ethereum leading the charge of smart-contracts-enabled blockchains, is instead more heterogeneous, allowing easy tokenization of assets.

Both designs are extremely low energy efficiency with medium-low block validation speed and transactions per block. Solving the scalability, speed, and energy consumption issues is the goal of third-generation blockchain solutions, using different approaches and techniques like Proof-of-Stake validation algorithm, off-chain routing, graph-chains, and complete or partial centralization.

The fourth generation goes beyond, achieving faster and more scalable solutions while simultaneously becoming competitive from a business perspective. Simple chains of data are not flexible enough to fulfill corporate environment needs, in which complex data structures need to be organized in tables (as in relational databases). At the same time, those structures need to be validated and made immutable with blockchain-based techniques, increasing traceability and security.

In other words, fourth generation blockchain brings this technology to a complete primary production application, and expand current business-oriented offer in terms of data storage, application decentralization, auditing, security, reliability.

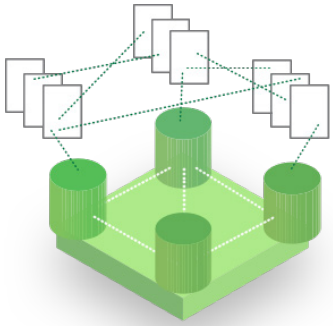
Multiversum offers complex data organization instead of data sequencing, chain splitting and rejoining to allow for greater scalability and parallelism. This includes concept of Proof of Integrity validation (i.e. cryptographic proof of server code) instead of existing Proof-of-Work or Proof-of-Stake solutions.

Always thinking about the future and future-proofing our development, Multiversum will feature ERC20/ERC23 integration, allowing coins and tokens from other solutions to be hosted on our chain and vice-versa, with notary services as an external confirmation method. Meanwhile, together with these innovations, we are certainly going to leverage several smart solutions that our colleagues have already implemented over time.

Multiversum

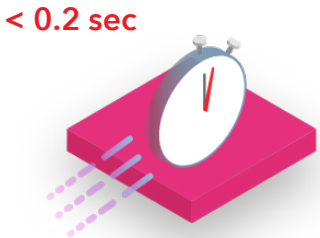
The 4th Generation Relational Blockchain

Why is Multiversum *the* 4.0 Blockchain?



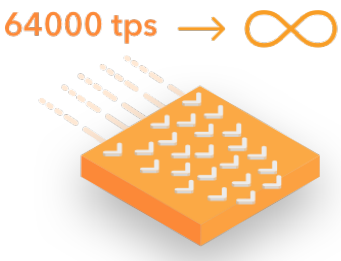
Relational Blockchain

A brand new blockchain which features different types of data, related in a multidimensional structure.



Transaction Speed

In less than 0.2 seconds funds are transferred across wallets, including secure validation of the transactions. Among the fastest in the world.



Transaction Throughput

Unmatched scalability: up to 64,000 TPS (1000 TPS/core) on a 64 cores server. Support to 64+ cores technologies.

Proof of Integrity

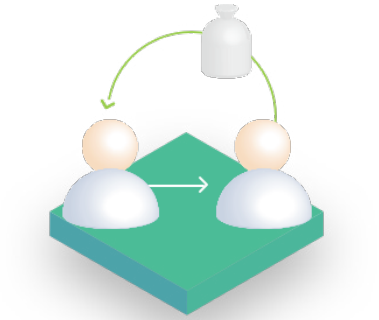
PoS (Proof of Stake) will be replaced by PoI (Proof of Integrity: cryptographic proof of server code)

Next Generation Wallet

Cutting edge security in access and funds transfers with biometric inputs.

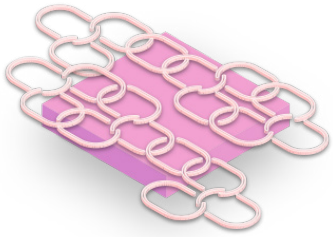
Eco-friendly

A Multiversum transaction will be low cost with next to zero environmental footprint.



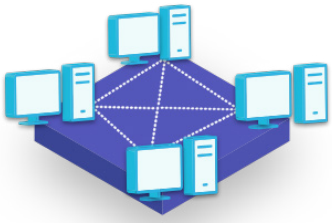
Rollback

Optional rollback can be activated on Multiversum-hosted tokens.



Divisible Chains

Resources optimization among nodes because of chain severability..



Recovery Nodes Allocation

MTV Nodes scattered all over the world for unmatched resilience, top reliability and global disaster recovery.

Public Presentation

Current Blockchain State of the Art

The primary innovative traits of Blockchain are remarkable safety and reliability. At the same time, we pay for this in terms of huge processing power, unacceptable pollution, high transaction costs and slow speed which can hardly represent current technological progress standards or give a reasonable technical answer to modern financial and commercial use cases.

This slowness is caused by the lack of horizontal scalability, i.e. the increase of computation capacity obtained by merely adding processors instead of replacing them with faster versions. Another cause is inherent to its current blockchain safety mechanism which is designed to prevent anyone from taking over the majority of the clusters by making it very expensive to achieve in terms of calculation power and/or cost (Proof-of-Work and Proof of Stake).

Furthermore, current blockchains are simple concatenations of single data entities state changes; reconstructing the actual states of these entities implies a whole chain scan, which causes an even greater system slowdown and resources usage. This simplification makes blockchains inadequate for scientific and industrial purposes, as requirements in terms of data structures can become extremely complex. Notably, these security measures stop at data level as they don't ensure user safety, making it impossible to recover lost or stolen coins and tokens even if they are located on the chain, or to block malicious accounts.

Lastly, we have the problem of fragmentation and inhomogeneity among cryptocurrencies that are unable to communicate with each other and live in unrelated universes.

A Generational Approach to Blockchain Technology

The first and most important concept to be introduced in order to fully understand the range of Multiversum impact in the IT world is "fourth generation blockchain".

Why fourth? What were the features of the other three generations?

It all started in 2009, with Satoshi Nakamoto's 9-paged technical white paper; "Bitcoin: A Peer-to-Peer Electronic Cash System". This paper introduced a new technology, or rather, a new application for the blockchain: a cryptocurrency, able to substitute modern electronic payment systems. This revolutionary payment system granted unparalleled security on a persistent distributed ledger, and authenticated all exchanges via timestamp. Transactions could be verified by anybody with an internet connection, while anonymity was guaranteed without compromising the transparency of all exchanges. Most of all, this system was decentralized. Power of handling and managing cash flow all over the world was no longer a privilege exclusively reserved for banks. Despite the absence of this technology from mainstream media (until late 2011), the revolution it kick-started was well and truly underway; the exchange of values could never be the same again.

The first generation of blockchain was the stick of a brand new spear. Bitcoin (BTC) was that stick to project Satoshi's vision of decentralization. However, the technology's upside was soon stymied by its downsides (huge power costs because of Proof of Work consensus model, (up to 1,8$ per transaction) and unsustainable confirmation times (up to 45 minutes). The lightweight Litecoin (LTC) and the hybrid Ripple (XRP) featuring extremely low cost and faster (3.3 seconds wallet-to-wallet) transactions (0,0004$ per transaction) and greater scalability (up to 1500 transactions per second), followed shortly, showcasing the tip of a very blunt spear.

Then, came a supernova.

A young and brilliant cryptoenthusiast, Vitalik Buterin, released his own blockchain: Ethereum. Ethereum was an outstanding achievement, allowing users to create smart contracts - basically the bridge between programming code and legal contracts. Almost any kind of business relation and work or transaction contract could be coded by the author to suit their needs and requirements that would automatically execute when the predefined conditions are met. Unfortunately, Ethereum features rather costly transactions (up to 0.46$/Tx) and a less than optimal confirmation speed (4+ minutes with maximum gas fee). Another problem that dogged the blockchain was chain saturation.

In 2016, Ethereum chain almost crashed due to the wild upsurge in traffic from ICOs. It ran gibberish for weeks, clearly showing the limits of this technology. It was time to rescue the situation with a newfangled blockchain dubbed "Ethereum-killer", NEO. NEO came out as the most evolved second generation blockchain, promising a cryptocurrency "that utilizes blockchain technology and digital identity to digitize assets, to automate the management of digital assets using smart contracts, and to achieve a "smart economy" with a distributed network". Ambitious as its claims, NEO set about filling the abstraction gap between crypto world, real assets, and real world economics. The blockchain also stressed privacy and a unique scam-proof digital identity, optimized smart contracts and extremely versatile programming protocols.

What more can you expect from a blockchain?

The third generation gave a prompt answer: Ark, Lisk, Bitshares, Steem and many more projects all featured unparalleled scalability and a new consensus model: Delegated Proof of Stake (DPoS). This new protocol worked as follow: given a set, usually small number of witnesses, they work to mine blocks and monitor the network. These witnesses are elected to this role by the chain users. Because all votes have the same weight, this can be seen as a decentralization of the voting process, i.e. it isn't limited to masternode holders like in Proof of Stake consensus.
Because there is a smaller group of individuals working to secure the network, needed changes can occur much more efficiently. Also, because witnesses are reliant of votes, it is in their best interest to pay attention to the concerns of the other participants on the network. This way we can achieve a community ruled blockchain with perfect decentralization and authentic "democracy".

So far, we've mentioned three generation of blockchain. We've seen their strengths and weaknesses, but why do we need a fourth blockchain? Let's start by examining the basic flaw of all these blockchains; data storage and access. No matter how beautiful a mansion looks, once the foundation is shaky, the whole structure is flawed. Hence the problems with the current blockchain generation. Multiversum as the fourth blockchain solves this problems. We are the perfect and best possible synthesis of all these features and a relational database.

What is a relational database? (source: en.wikipedia.org)

One well-known definition of what constitutes a relational database system is composed of Codd's 12 rules. However, many of the early implementations of the relational model

did not conform to all of Codd's rules, so the term gradually came to describe a broader class of database systems, which at a minimum:

- Present the data to the user as relations (a presentation in tabular form, i.e. as a collection of tables with each table consisting of a set of rows and columns);
- Provide relational operators to manipulate the data in tabular form.

The first systems that were relatively faithful implementations of the relational model were from:

- University of Michigan - Micro DBMS (1969)
- Massachusetts Institute of Technology (1971)[10]
- IBM UK Scientific Centre at Peterlee - IS1 (1970–72) and its successor, PRTV (1973–79)

The first system sold as an RDBMS was Multics Relational Data Store (1978). Ingres and IBM BS12 followed.
The most common definition of an RDBMS is a product that presents a view of data as a collection of rows and columns, even if it is not based strictly upon relational theory. By this definition, RDBMS products typically implement some but not all of Codd's 12 rules. A second school of thought argues that if a database does not implement all of Codd's rules (or the current understanding on the relational model, as expressed by Christopher J Date, Hugh Darwen and others), it is not relational. This view, shared by many theorists and other strict adherents to Codd's principles, would disqualify most DBMSs as not relational. For clarification, they often refer to some RDBMSs as truly-relational database management systems (TRDBMS), naming others pseudo-relational database management systems (PRDBMS).

Modern relational databases have several limitations: data storage, data safety, speed, scalability and versatility. Multiversum boasts of a brand new data link system, facilitating limitless connections between any kind of data, decentralized on-chain storage, data sharding, resilient network with recovery nodes, and paralleled throughput. Multiversum also supports autonomous hierarchical distribution of computation among cluster nodes, biometric input and a new consensus model based on cryptographic proof of server code (Proof of Integrity, PoI) that allows for unparalleled speed, next-to-zero power consumption and sci-fi scalability (64000TpS on a 64 core server with 64+ technology support). Our blockchain outperforms (cost-effectiveness-wise and performance wise) any existing blockchain and relational database (mySQL, noSQL etc.) available on the market.

Another incredible plus is its lightning fast implementation due to node decentralization

and data sharding of the database that does not require the setting up a new network each time a redundant, decentralized database is needed. While the versatile, scalable, multidimensional and lightweight database structure is easily implemented (both in corporate and institutional contexts), its design allows for unparalleled security, resilience in case of attack or catastrophic event and the most sought-after blockchain features (data sharding, persistence, biometric input, optional rollback and ultimate scalability).

Multiversum Blockchain is everything a company or institution could ever dream of: light, secure, fast, scalable, and versatile. A decentralized, next-to-zero impact and fee payment system, persistent and immutable crypto-relational database, resilient and extremely secure infrastructure for any IT, Big Data, archive, and database purpose.

Multiversum, being a blockchain as a service, offers simple solutions to complex issues.

	BITCOIN 1st gen	ETHEREUM 2nd gen	NEO 2nd gen	STELLAR 3rd gen	MULTIVERSUM 4th gen	VISA just as a comparison
Primary function	Digital currency	Programming	Universal	Payments	Universal	Payments
Funds cleared	10-60 minutes	3-5 minutes[1]	N/A	3-5 seconds	0.2 seconds	Over 24 hours
Transactions per second (tps)	7	25	1,000+[7]	1,000+	64,000+	3,526
Transaction cost	$55.16 Dec 2017[2] $2.38 Mar 2018[2]	up to $13.98[1] $0.42 Mar 2018[2]	N/A	< $0.00001	< $0.000005	1.43% - 2.4%[3]
Energy per transaction	892,000 Wh[4]	72,000Wh[5]	N/A	0.03 Wh[6]	0.0000056 Wh	1.69 Wh[4]
CO2 per transaction	437.25 kg	35.29 kg	N/A	0.000015 kg	0.000000003 kg	0.00083 kg

best performance | worst performance

Data checked: Mar 2018

1. ethgasstation.info
2. Average transaction cost: bitinfocharts.com/comparison/bitcoin-transactionfees.html#3m
3. valuepenguin.com/what-credit-card-processing-fees-costs
4. digiconomist.net/bitcoin-energy-consumption (31 Mar 2018)
5. digiconomist.net/ethereum-energy-consumption (31 Mar 2018)
6. Stellar Consensus Protocol: electricity of a high-end computer for 1 hour / 1,000 transactions per second / 3,600 seconds * number of validators
7. hackernoon.com/neo-versus-ethereum-why-neo-might-be-2018s-strongest-cryptocurrency-79956138bea3

Multiversum and Blockchain Global Adoption

Multiversum technology pushes traditional blockchain beyond its current limits, by enhancing the data layer through self-verifying and distributed structures of organized data entities, related one to another by symbolic links.

This technology set the foundations for a decentralized and distributed system of coherent self-verifying transactions: Multiversum blockchain.

Multiversum allows the creation of a Relational Crypto Database (an advanced and organized data storage solution) which can handle not just a single data-type, but a series of data grouped in graphs of complex data structures related one another. Relations are now first class citizens of the blockchain and reinforced by cryptographic methods.

Each one of them, when a state change is requested, will have its own sub-chain splitting from the original branch, which will rejoin after the operation, in order to be validated. This makes Multiversum an evolved blockchain technology, offering unique features to overcome the previously analyzed inconveniences, with a set a crypto-validation and distribution techniques fit for every environment: Administrative, Industrial, Financial and Governmental.

One of Multiversum main targets is to offer the market, in every moment, the most evolved product available: this will be possible adopting an AGILE4 software development methodology. AGILE methodology implies a drastic reduction in initial project design involvement, in favor of valorization of experiences encountered during project development, which show opportunities and threats hardly predictable ex ante, rewarding best practices and leaving behind the inadequate ones. AGILE is an established software development standard and urges developers, product owners and investors to consider project scope5 as flexible and readily adaptable to market needs. Furthermore, in such a rapidly evolving sector as software, releasing a product after six months of study and a year of implementation after it was conceived to match the market needs of eighteen months before, means offering an obsolete product which gives answers to outdated issues, that have maybe been solved by competitors and lacking of responses to just-created challenges.

AGILE, instead, gives the chance to offer to the market the most innovative product at time of delivery.

Speed and Technology

One of the strengths of this technology is indeed speed, thanks to its ability to run different transactions parallel to each other and the split-rejoin mechanism of our blockchain. These features allow for greater horizontal scalability, and increase processing transactions capacity adding additional computational power to the existing one, making each node count, performance-wise.

Horizontal Scalability

Multiversum benefits from two specific features to maximize system efficiency:
The main chain is able to optimize its structures by splitting autonomously in multiple subchains, according to requested resources and data streams, parallelizing the work across multiple threads and nodes. This chain-split process is executed until the normalization of workloads, when, still autonomously, the chain becomes whole again.

All of this is possible due to mechanisms allowing every block of the chain to validate two different sub-chains from two different incoming links. Data sharding, i.e. a technique that allows data distribution among multiple nodes. Given an ABC data series and three Cluster nodes, we will have a data distribution as follows:

- AB
- BC
- CA

This subdivision permits higher processing speed of transactions, as data queries will impact only subchain nodes, optimizing each step.

Another extremely important feature of our technology is High Availability: the chance of relying on a cluster type that ensures continuity of services even in case of shutdown of some nodes in the network.

Using the previous example (A, B and C nodes), should C go offline, A and B nodes would still remain completely operative, allowing continuity of service without any kind of data loss as far as 50%+1 of the nodes remain operative.

This way, in case of multiple node failure, the cluster will autonomously reorganize data distribution communicating with every node, until complete operational recovery.

Environment

Multiversum is also Eco-Friendly: one of our main goals is to lower the computational power needed for cryptographic validation therefore avoiding mining (Proof-of-Work), a huge waste of power and resources. Instead of this obsolete technique, we are implementing Proof of Integrity, a protocol that performs cryptographic validation by checking the authenticity of the software that resolves every persistence of the transaction.

Data Management

Multiversum, with its Crypto-Relational Database, can easily structure it without limits of data linking.

Every wallet will have a series of states and will be linked to a person (user); each new wallet state change will include two data fields: the previous state, to check for validation as well as a link to the last transaction (or to the last main chain link) so that the provenience of the new state change link will be known. After the change, the transaction modification will be added and its modified state link will rejoin the main chain.

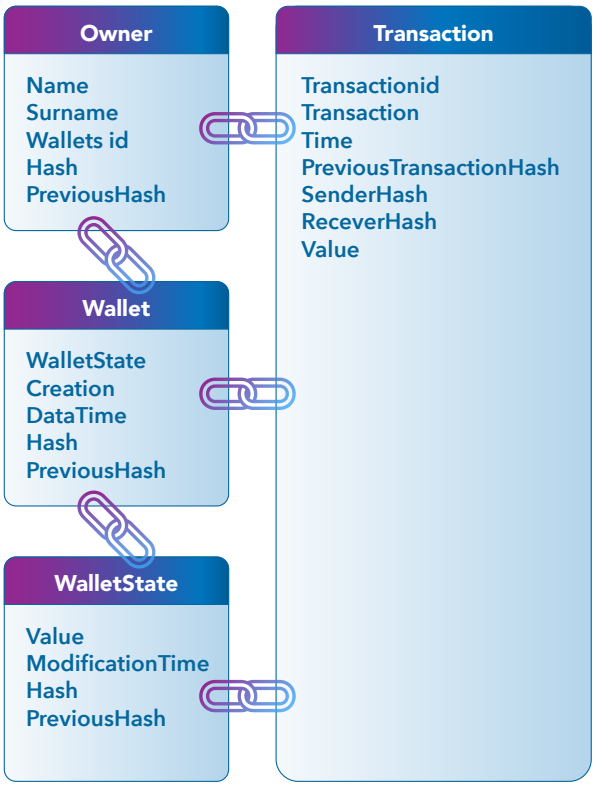
Using this reinforcement system, the new transaction will inherit two hashes: one from the state link, one from the previous transaction, and in this way, all operations will validate the previous ones related to the transaction itself.

This advanced solution, able to manage complex data scenarios, will allow people to implement any kind of application on our technology, ensuring worldwide institutional, governmental, financial and industrial diffusion, bringing the whole blockchain universe one step forward.

Multiversum Mission

Multiversum aims for a generational step-up in the blockchain world, and as Unique Selling Points, we are proposing the following objectives:

1. Achievement of a Crypto Relational DB with self-validating Complex Data Structures
2. Divisible / re-joinable chains based on current system workload (Parallel Work)
3. Data Sharding (Parallel Work)
4. Advanced API offering
5. Rollback (User Security)
6. Freezable wallets (User Security)
7. Integration of biometric data as seed for the Electronic Signature
8. ERC23 interface (Interoperability with other blockchains)
9. Native off-chain adaptors for its own ERC20/ERC23 (Interoperability with other blockchains)
10. Native off-chain adaptors for ERC20/ERC23 guests (Interoperability with other blockchains)
11. Double Access Lock (Structural Security)
12. Reverse Access Denial (Structural Security)
13. Reciprocal Chain Confirmation (Interoperability with other blockchains)
14. Integration for Java, Spring and Javascript
15. ACID model
16. Transactional Model
17. SQL-like language



1. Achievement of a Crypto Relational DB with Self-validating Complex Data Structures

Multiversum has a strong vocation towards industrial and institutional use, contexts in which we have data with complex structures, impossible to be represented in an efficient and normalized way with a simple chain. We aim to become the first relational crypto relational database on the market, decentralized or simply distributed if needed. This capability derives from chainable entities conceptualization: in our technology a primary chain is able to split into secondary chains, containing different sets of entities and records. These entities will rejoin again at their last persisted state and, after the required modifications, they will rejoin once more to the last link of the primary chain, becoming a whole again. The "chainable" interface presupposes a kind of record which includes two or more hashes of the previous records, validating not only one but more sub-chains.

In Multiversum standard implementation, that is used by Versum coins, the Chainable entities that coexist on the chain will belong to four tables: User, Wallet, Wallet State, Transaction, related one another and reciprocally confirming themselves.

2. Divisible / Re-joinable Chains Based on Current System Workload (Parallel Work)

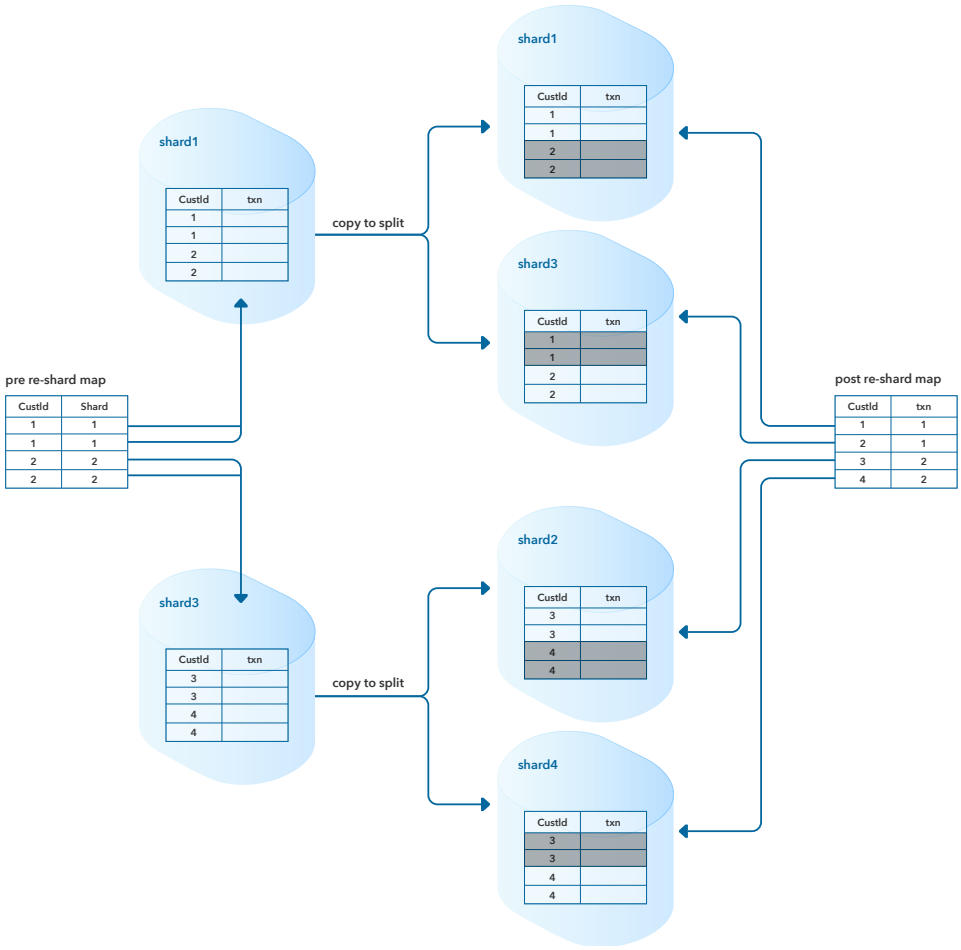
The same ability to derive multiple links from a given one and join them back allows the technology to use workload analyzers which will indicate the cluster the need of splitting the primary chain in two secondary chains (and possibly splitting themselves again indef-

initely) when a high request of transactions executions occurs. Once the workload has dropped again, multiple pre-existing sub-chains are allowed to link back and validated. This mechanism allows parallel work while maintaining safety to the transaction's record.

3. Data sharding (Parallel Work)

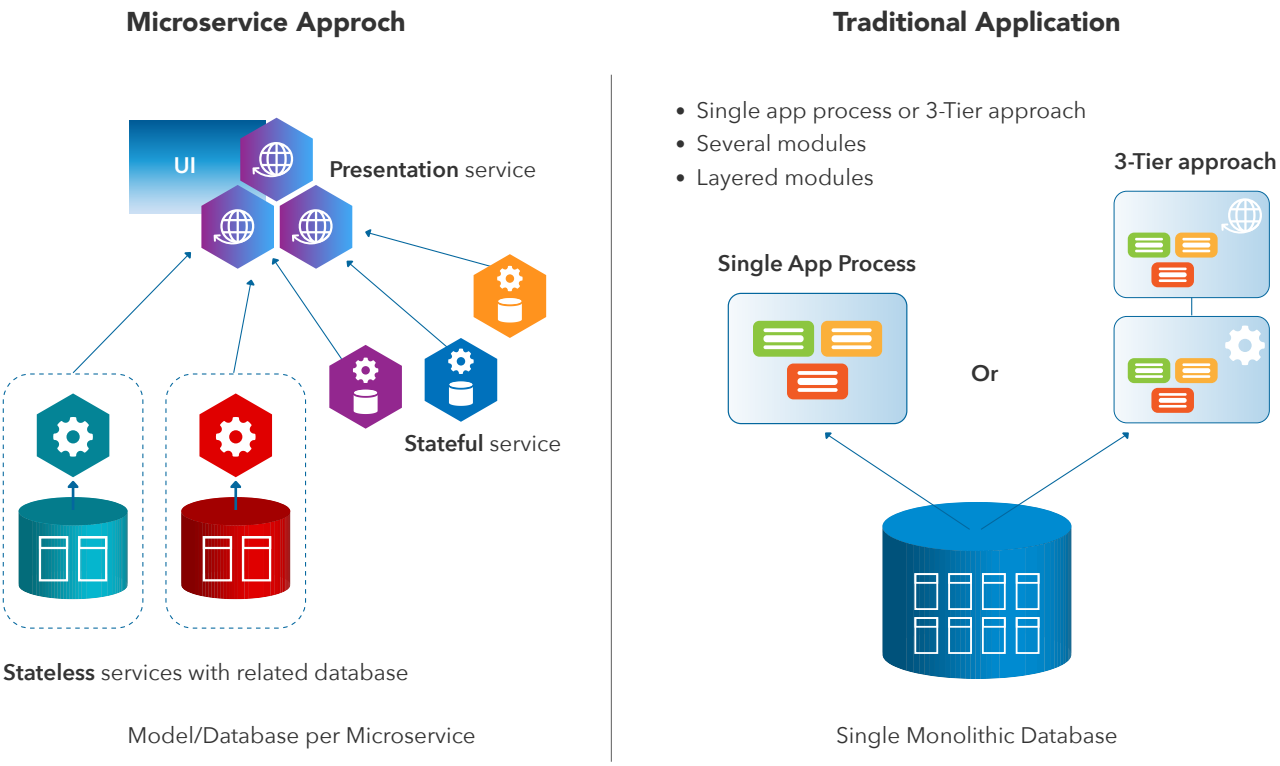
Each node will contain the whole chain data or just a part of the chain. When data sharding is needed, coordinator nodes will set specific data partition modes, in order to optimize their own distribution according to current workload. According to high availability techniques, reliability and persistence will always be ensured, even in case of sudden loss of part of the cluster, given that at least 50%+1 of the nodes survive.

These nodes, after a partial cluster crash, will be able to redistribute and reorganize the data structures to be able to confront another partial cluster crash as soon as possible. By means of techniques 2 and 3, Multiversum blockchain will have enhanced parallel work and data sharding capacity, which means horizontal scalability, increased security, high availability, system resilience, absence of a single point of failure and self-disaster recovery.



4. Microservice structure and Advanced API offer

Developed on a platform based on both Microservices9 and Serverless models10, Multiversum will be able to offer advanced secure and modern API functionalities and to adapt on both structures.

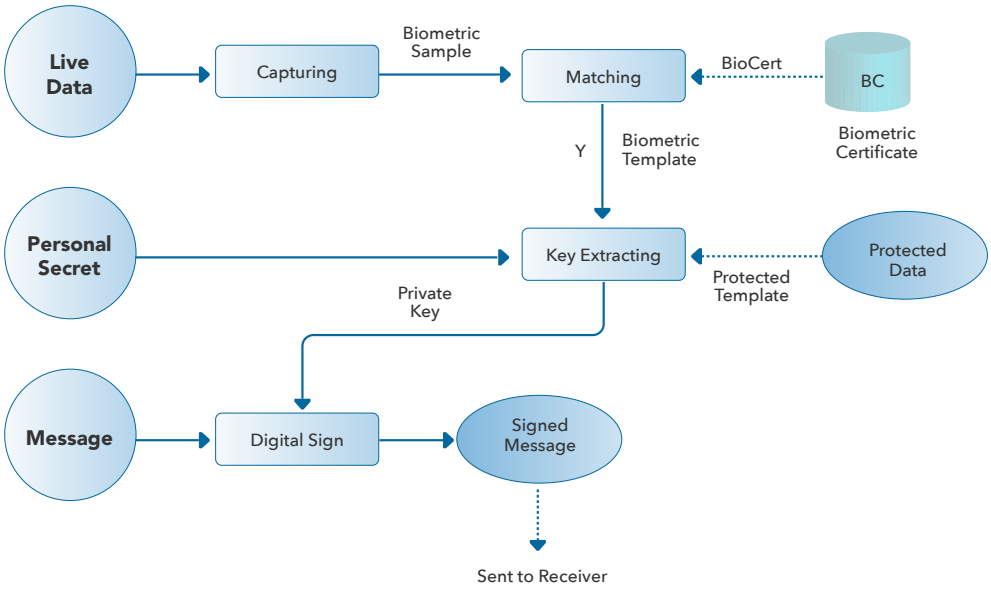


5. Rollback (User Security)

Our technology, in a transactional context, will allow for rollbacks of undesired operations, i.e. recover an early state without disrupting credibility of chain validation, by implementing a set of transaction recovery states. This feature can be enabled, optionally, on all tokens and applications hosted on the Multiversum blockchain.

6. Freezable wallets (User Security)

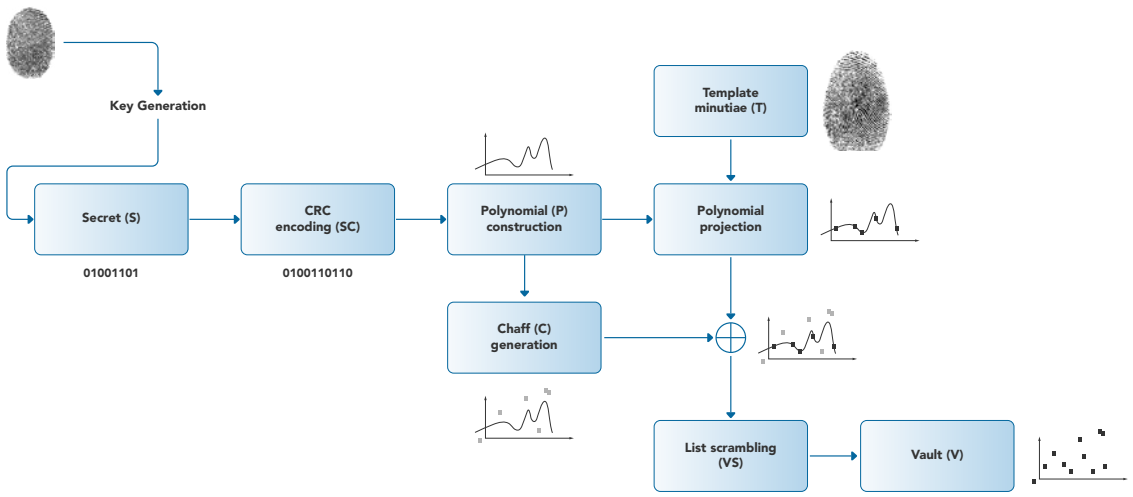
The chance of including a wallet-freezing feature in case of unlawful or suspect activities will be implemented after studying its feasibility on the Business Logic side. Proprietary applications, built on Multiversum blockchain, will have the option to implement this feature if desired.



Biometric Digital Key Generation Framework

7. Integration of biometric data as a seed for Electronic Signature

Starting from the research done by Je-Gyeong Jo, Jong-Won Seo and Hyung-Woo Lee's work, Multiversum team will assess the feasibility of biometric data like fingerprints, retina scan and graphometric signature as a source of asymmetric cryptographic key to guarantee authenticity of the signer's identity. Safety of encrypted data and their use as a validation in legal arguments will be evaluated. Furthermore, biometric data will be used on Android, IOS and other platforms applications to manage user security.



Fuzzy Vault Scheme for Biometric Digital Key Protection

8. ERC23 interface (Interoperability with other blockchains)

Multiversum coins will be developed implementing ERC23 interface, that is backwards compatible with ERC2012, to ensure interoperability with other chains.

9. Native off-chain adapter for proprietary ERC20/ERC23 (Interoperability with other blockchains)

Multiversum will develop a native adapter to allow inbound and outbound flow of its own coins and tokens to non-proprietary chains.

10. Native off-chain adapter for external ERC20/ERC23 (Interoperability with other blockchains)

Multiversum will develop a native adapter to allow inbound and outbound flow of coins and tokens from non-proprietary chain on its own chain.

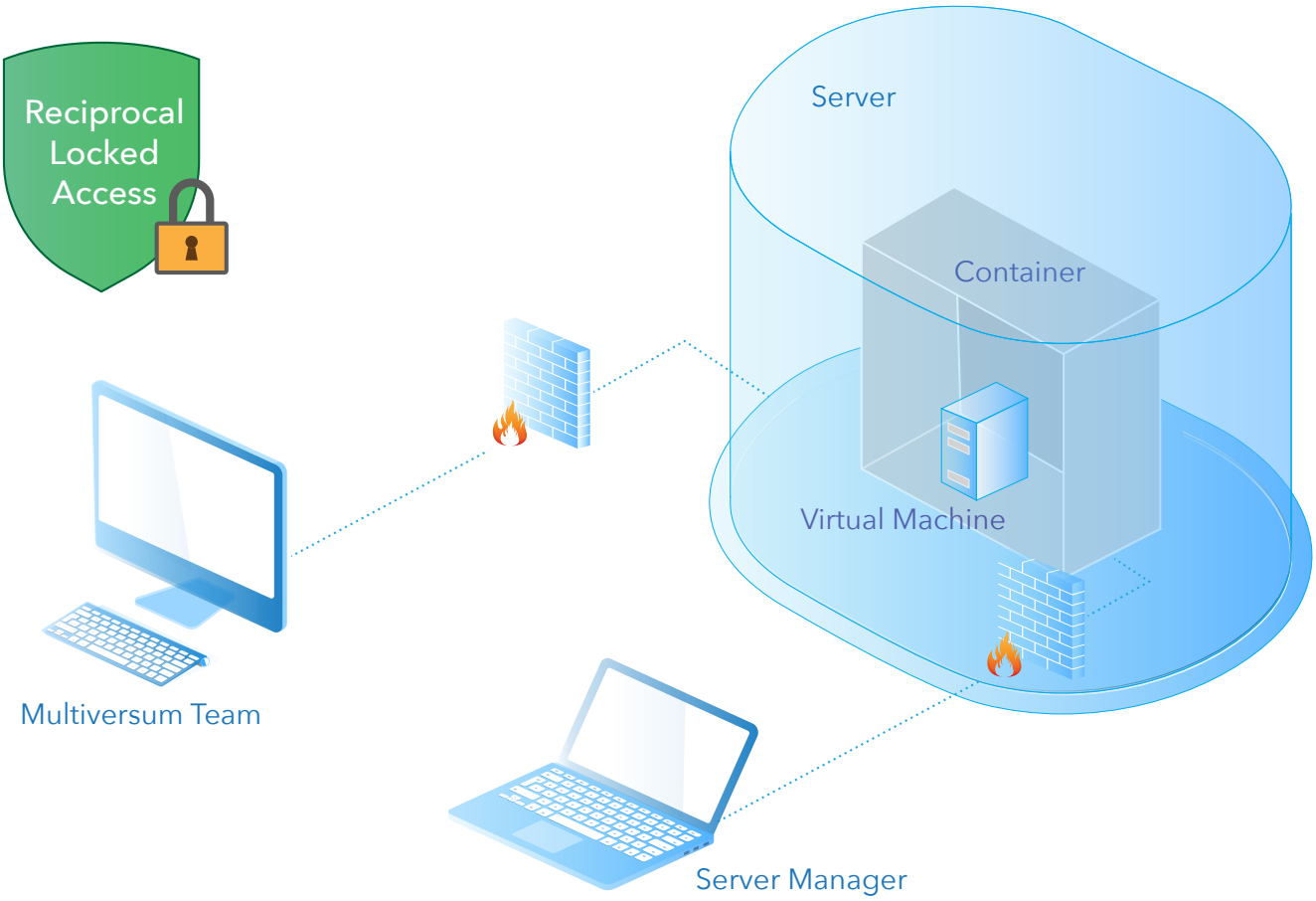
11. Double Access Lock (Structural Security)

Nodes will be distributed in secured Virtual Containers, with credentials not available to

the Host machine operator, precluding access; therefore, safety is referred to Linux Security Best Practices, as, for instance, SeLinux and/or other packages. Meanwhile, if someone had Guest machine credentials, he would still not be able to gain access to it, being unable to access the host machine running that node. The node is, as a matter of fact, secured by double access lock.

12. Reverse Access Denial (Structural Security)

The access lock described at point 12 entails a reciprocal preclusion of node access to both host machine operators and someone eventually possessing the node credentials; this ensures that every node not directly managed by Multiversum is authentic and inaccessible by anybody, basically autonomous and isolated from external human intervention. Three fundamental components will be distributed within container in addition to Operative System and Safety ones: Multiversum Server compiled code, a certificate with asymmetric key to authenticate to Multiversum cluster, a component already described by point 11) which is responsible for challenge computation based on server code hash, certificate, challenge seed and transaction data.



Additional optional security techniques might be implemented, such as automated update of container access credentials with a random password during its compiling phase to prevent anyone from accessing. This mechanism might be adopted for cluster access certificate.

13. Reciprocal chain confirmation (Interoperability with other blockchains)

Multiversum will study the feasibility of an external chain integration component, able to store states of other blockchains (eventually in exchange of tokens) providing additional validation and trust.

The very same technique can also be used to let Multiversum share its own state validation to other blockchains, "outsourcing" verification. A specific interface would be provided for this functionality, that would also need to be promoted among existing and future blockchain implementations. Such feature will rely on a serverless component that can be accessed also after container compilation, to allow the inclusion of adapters toward other chains.

14. Integration with Java, Spring and JavaScript

Multiversum will offer high end interfaces grouped in functional libraries for Java, Javascript and possibly other mainstream languages, allowing an easier adoption of our technology at enterprise and institutional level. Integration modules with frameworks like Spring15 will also be developed. These libraries will facilitate the integration of Multiversum in proprietary solutions, both in private chains and official MainNet.

15. ACID model

Multiversum will satisfy the ACID16 paradigm; this acronymous empathizes the logic properties
required by transactions. To ensure a secure transactional model, the technology implemented needs to fulfill the following properties:

Atomicity: A transaction is not divisible in its execution and its execution must be complete or null, partial executions are not allowed.

Consistency: Any transaction will bring the database from one valid state to another. Persisted data must be valid according to all defined rules.

Isolation: Every transaction must be executed in an isolated way: the eventual failure of a transaction shall not interfere with other concurring transactions.

Durability: Also named persistence, imposes that once a transaction is committed, the result cannot be lost for any reason (crashes, errors, power loss).

16. Transactional Model

Multiversum will persist transaction data in a transactional18 model, making sure that all of the data or none of it on the multiple sub-chains involved will be persisted, enforcing coherence of each executed transaction and data completeness.

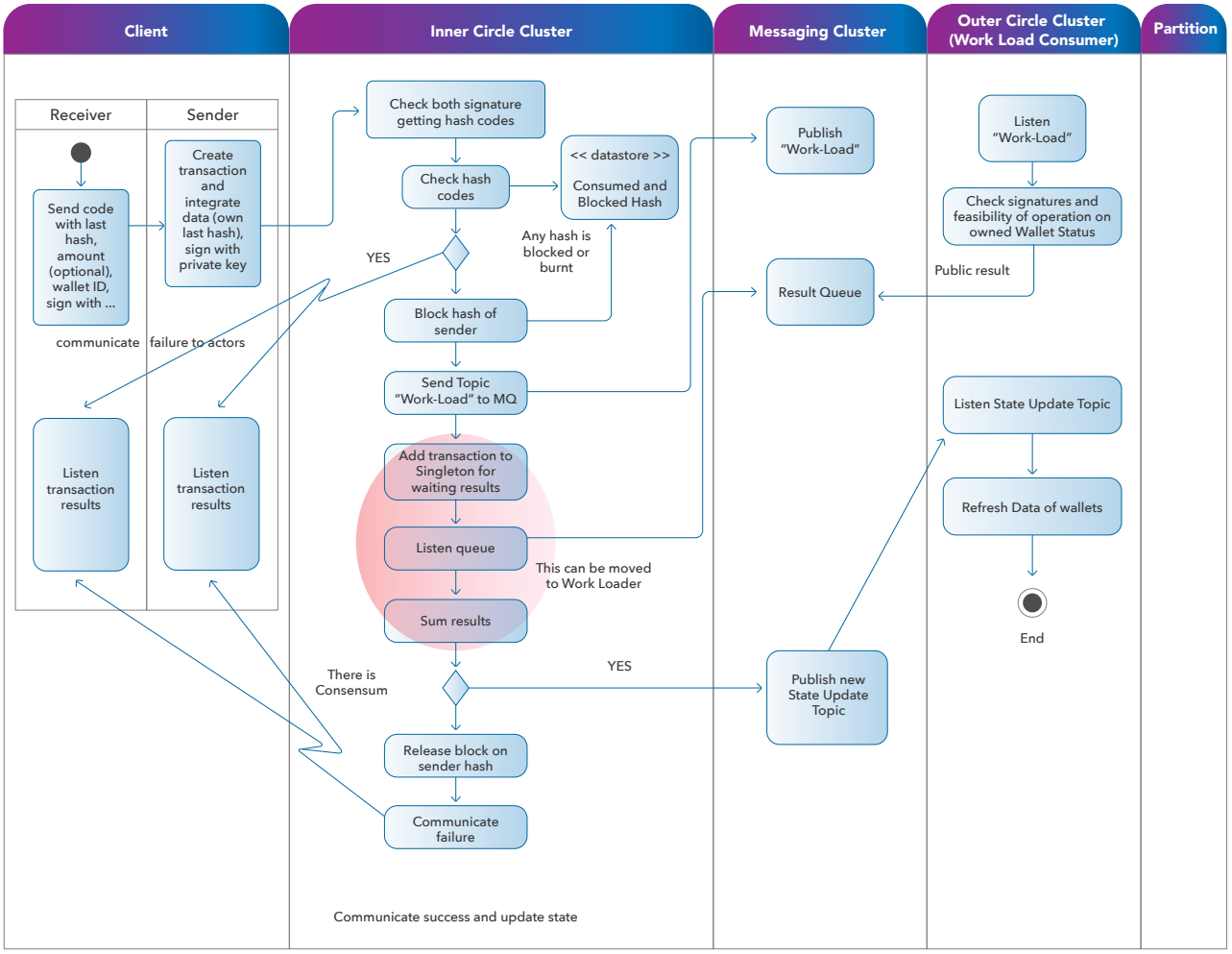
17. SQL-like Language

To simplify development of applications based on our Crypto-Relational Database technology and to soften the learning curve versus existing technologies, Multiversum will feature a SQL-based18 syntax to use standard persistent-storage functions (CRUD).

18. Full Route Data Flux

The processes of acceptance, control, validation and persistence of a transaction take place with the following schematized and simplified procedure:

- The transaction is sent to a REST client, with its necessary data, signed with private key;
- The REST client sends the transaction to a leader node of coordination clusters:
- It then splits the work across nodes with a proprietary coordination protocol.



These components will run an initial check of data completeness, signature, funds availability, already used hashes, in actual wallet states, blocked wallets or users; any additional operation from the sender ID is now locked in the volatile memory, while specific data fields are finalized (like previous transactions to link to, timestamp and previous hash).

The transaction is sent to a Topic Message Queue19 with a protocol that has to be defined (AMQP for the pilot, MQTT and others to be defined) and distributed in parallel to worker Nodes.

Worker nodes verify their interest in processing the request (they could be missing necessary data, be already busy and other conditions to be evaluated) and proceed to create the new wallet state, recovering correlated hashes of previous linked transactions and adding them to transaction record.

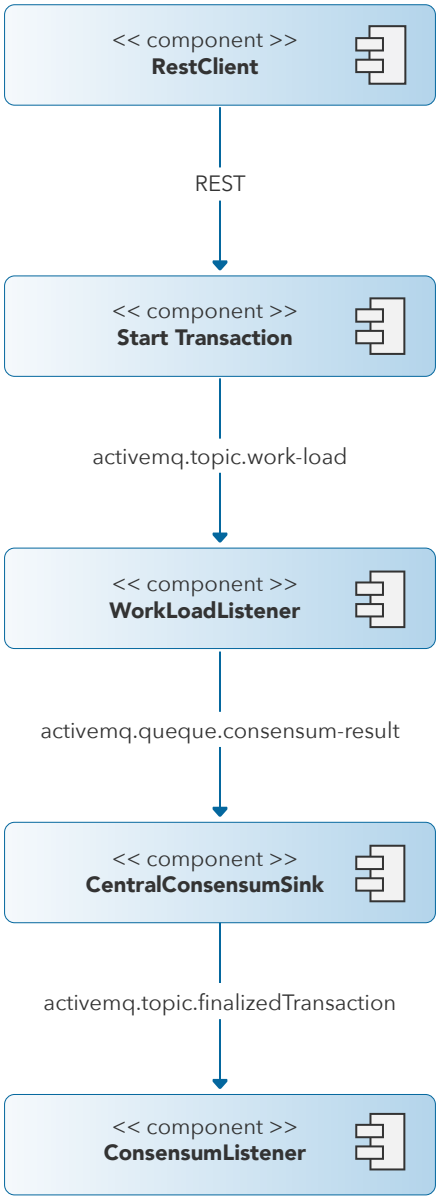
Proof of Integrity result is now added; Transaction Hash is calculated; worker nodes register the transaction in memory and send a vote to coordinator nodes through a Message Queue, collecting the results.

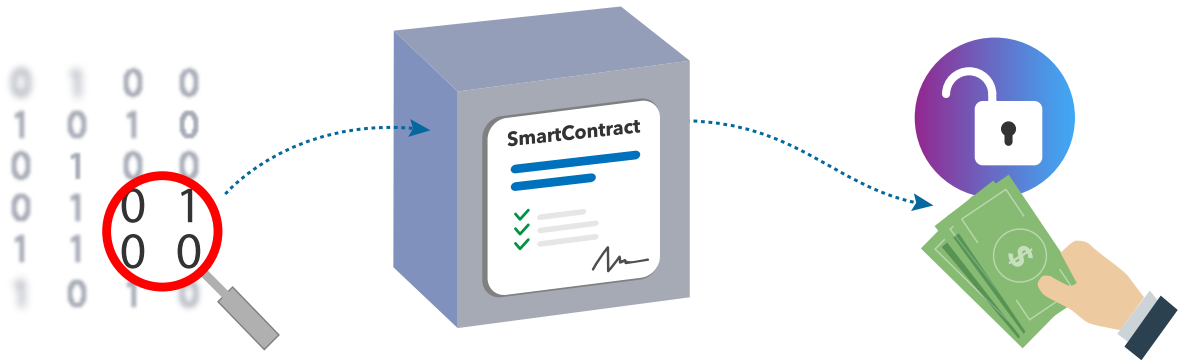
If votes and hashes are coherent, the coordination nodes will persist the transaction and any new states of wallets, burning any hashes from previous states and broadcasting vote validity with an additional Topic Message Queue system. Worker nodes will now also persist the transaction and wallet state changes.

This course illustrates the best case full route scenario.

Logic data flux

Detail of process flow





Smart Contracts

Multiversum believes in the importance of proposing improved Smart Contracts20 to the public, but at the time of writing, unless there is an adjustment in research scopes, has not decided to explore this possibility. Considering this, we are looking to include in Multiversum technology the Open Source solution that fits our needs at best, to be implemented as reference according with its licensing model.

Infrastructure

Multiversum infrastructure is designed to ensure resilience and reachability. This objective has been achieved developing node clusters that are able to self-elect their members for specific roles, according to each node technical specs, among which:

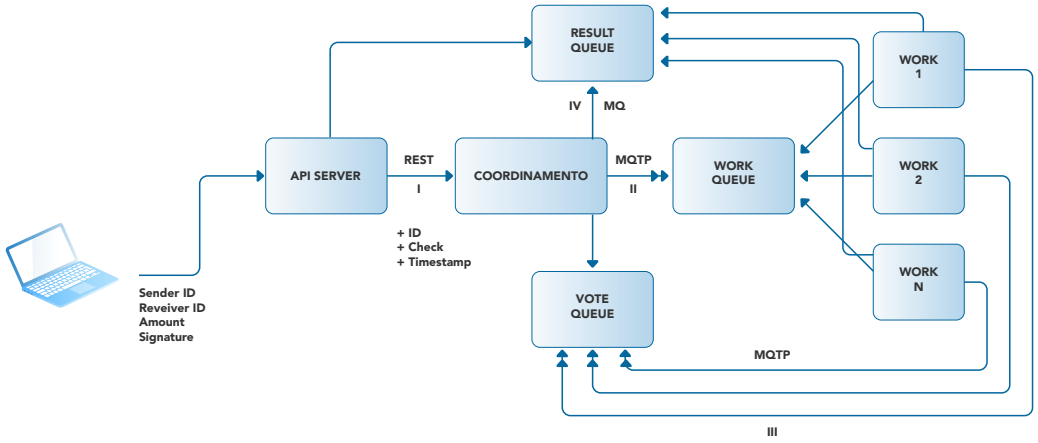
- Computation capacity
- Memory capacity
- Reciprocal Latency
- Chain data completeness
- Machine reliability
- Doubts on Proof of Integrity

Nodes will then have one or more roles:

- Client nodes
- Coordination nodes
- Messaging nodes
- Work nodes
- Persistence nodes
- Backup nodes

Every node that can provide a valid certificate will be able to register to the cluster and obtain a role.

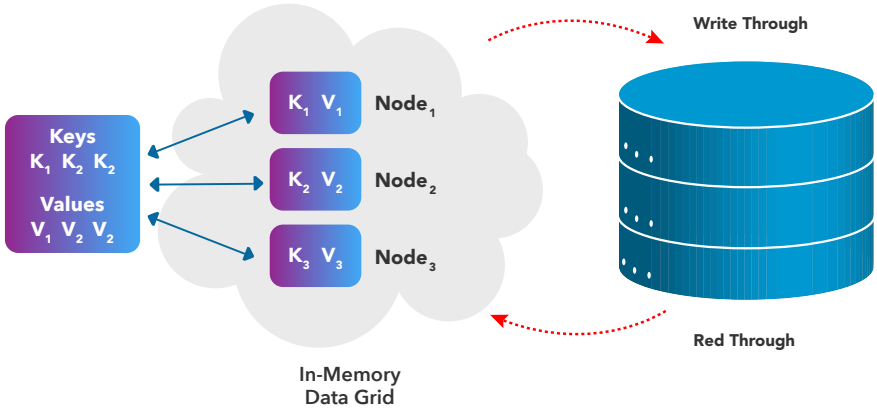
In case of crash of one or more nodes, the cluster will be able to autonomously redistrib-



ute tasks, optimizing roles.

Components of shared cache intra JVM22 will be there as memory database, allowing:

- **Read Through**, i.e. data-read queries directly executed in volatile memory before looking into physical memory.
- **Write Through**, loading data in volatile memory before executing a mass insertion to persist data, in order to optimize performance.



Notes on Security

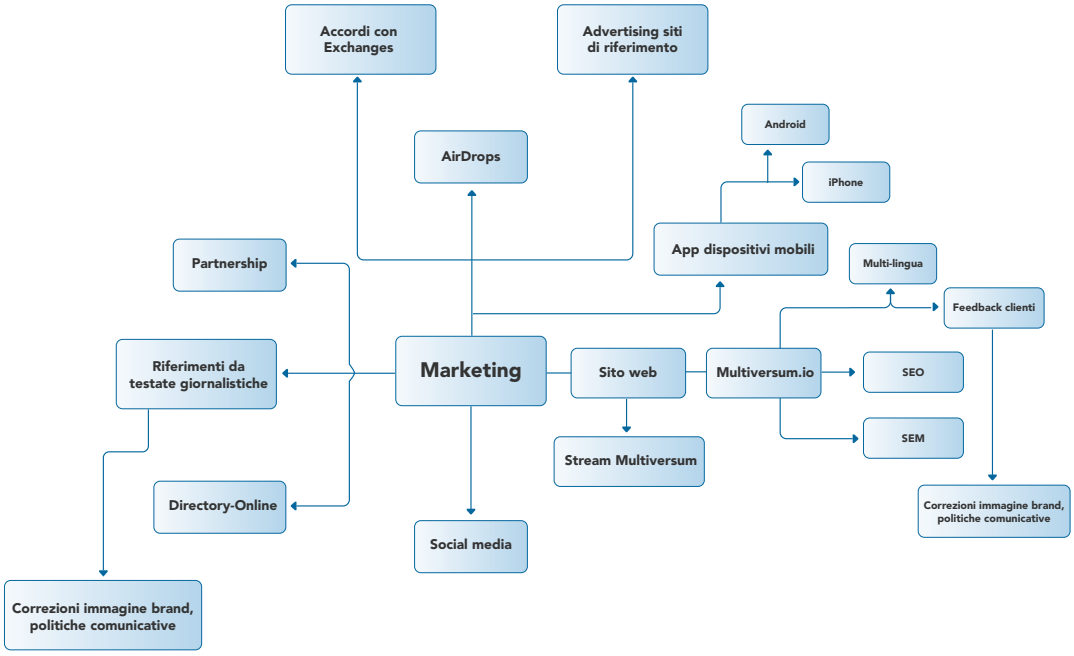
During development, "Hacker's bounties" will be offered to developers exposing vulnerabilities and able to suggest a valid fix.

Marketing Strategy

Operating in the ever-changing IT market, Multiversum will update our strategy, communication techniques and company mission accordingly, and focus on creating value for stakeholders, ensuring the appropriate balance between short and long-term management logic.

The key points of our plan are:

- Company mission
- Business objectives
- Business strategies
- Business activity portfolio



One of the main tools will be Social Media Marketing: campaigns conducted on social networks to increase brand awareness, identify potential consumers, generate contacts and build meaningful relationships with customers.

Our Social Media Strategists will carry out several actions that are part of a single strategic

plan, starting with the management and monitoring of channels using dedicated tools and community development, focusing on contents and interaction and tactics efficiency assessment based on obtained results.

Company Mission

Our mission is to achieve syntony between IT security and cryptocurrencies universes, ensuring that blockchain development is a simple as hassle-free as developing any normal applications with a shorter time to market, a lower cost, and so a lower risk. For developers, Multiversum will serve as medium of data persistence like any database and drastically reduces the complexity of developing blockchain solutions through our libraries.

Business Objectives

We are offering our technology to financial institutions, public and private organizations, government bodies, industries and individuals who want to achieve transparency and secure their data through the immutable and tamper-proof certification of the blockchain without the complexities inherent in developing for the technology

Business Strategies

Our first strategic decision was the headquartering of our operations in Belarus in order to take advantage of the country's openness to blockchain technology and very friendly and liberal legislations involving the technology. Belarus is not only tolerant but innovative, which has enabled us to be included in the High Tech Park; a tax-free zone.

Business Activity Portfolio

Multiversum blockchain is powered by a node system, enabling the storage, exchange, and transfer of data through peer-to-peer channels on the network. These nodes can be harnessed together to form a masternode - basically a server on a decentralized network - for private networks and privacy-based ecosystem.

While most organizations are eager to embrace blockchain technology despite the scalability and transaction challenges plaguing the existing solutions, the adoption rate - especially for private businesses and government - has been well below the enthusiasm displayed for the technology. This is due to the fact that most of these bodies prefer to have a private network where they can control private exchanges, protect their integrity, securely access and store data without broadcasting their content on the public ledger. Multiversum's masternode system enables them to do so cheaply at no infrastructural cost and rewards them for operating these nodes.

How Masternodes are Reward

Masternodes not only ensures the privacy of transactions, but also enables instant exchange of values, and facilitates participation in voting and governance, running 24/7 on the network. However, masternode owners must stake a certain amount of tokens for them to provide such service - 15,000 MTV in this case. Staked tokens are secured as collateral by the network and protected, however for rendering such an invaluable service, masternode owners must be incentivized in order to continue supporting the network.

Payment for running masternodes will be derived from block rewards on the network with up to 45% of the total accrued paid out in tokens to masternodes for their services. Daily amount payable may fluctuate based on the number of masternodes on the network. However, the fixed total amount of tokens in circulations (141,000,000 MTV) and masternode requirement (15,000 MTV), numerous use case for both the public and private sector, and partnerships will ensure the steady growth of the market value of the token and reward for masternode operators.

Token Economy

To begin understand and quantify the value proposition of Multiversum masternode system, one has to understand that Multiversum's ecosystem is underpinned by its native MTV tokens. MTV tokens are not only the vehicle for all exchanges, but also access to writing and querying records on the blockchain.

MTV tokens are not mere speculative assets with no intrinsic value or usage. MTV tokens affords all users the ability to write and query records on the blockchain depending on the amount of tokens owned by the users, the data size, and the processing power required. The fact that every operation on the blockchain requires these tokens in one capacity or the order embeds a quintessential supply-and-demand ecosystem for the token and the epitome of this is the Masternode system.

For writing records to the blockchain, the amount of tokens require is dependent on the data size, which for textual information could be very cheap, while visual contents will require slightly more (but still affordable and extremely secure on the blockchain). The cost of querying records on the blockchain will be dependant on the processor usage for the query, which in turn is dependent on the number of records being queried or involved in the query.

Masternode rewards owners with a share of the network fees from created and released blocks, creating a passive source of revenue for owners. With enterprises and cooperation requiring masternodes to run their private ledger, and public entities leveraging the public ledger for settlement of all exchanges, transparency, and auditability of records, Multiversum Masternodes promises passive income to owner, which in-turn increases demand and market value.

Masternodes enables an incentivized and always-on, dedicated second layer of nodes for trustless exchanges, hence the reason why the creation and ownership of nodes are collateralized with the same tokens. 15,000 MTV tokens is the required to collateral to run and own a masternode. These tokens cannot be moved nor can they be exchanged. However, owners can easily liquidate their collateral at anytime, shutting down the masternode.

Masternode reduces the total number of tokens in circulation, creating more demand for less, translating into increased cost of acquisition, and increasing the health of the network. The more nodes available, the more the resiliency of the network. These nodes can also be used for anonymizing transactions and voting on proposals

Multiversum can manage 1000 transactions per second per core: but it is able to do parallel work, so potentially infinite transactions per second. Need more power? Just add more nodes.

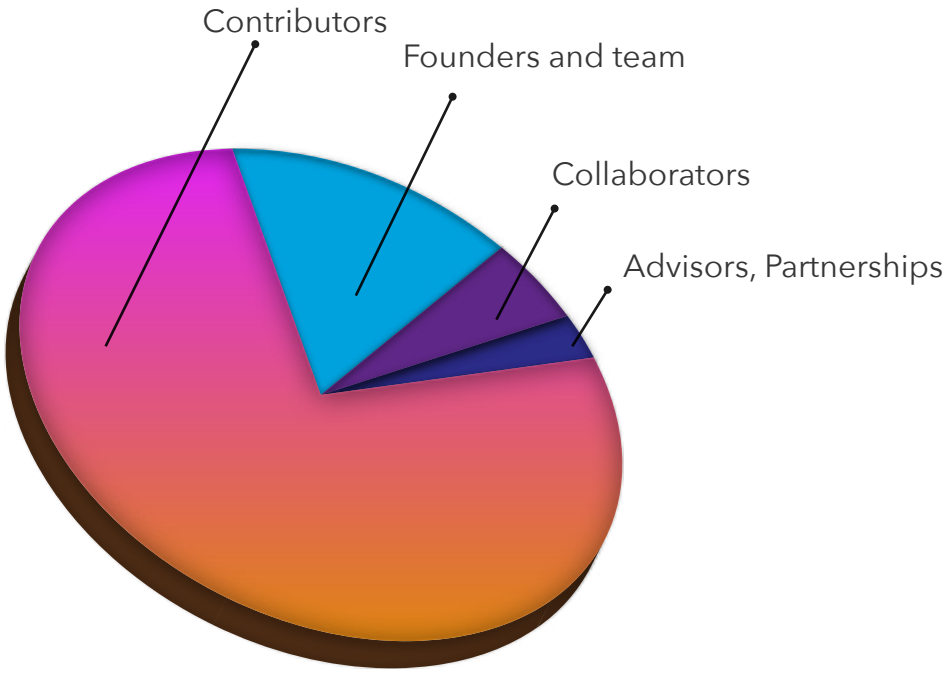
Token Offering

To further develop the platform, Multiversum will initial a Token generation Event offering 76% of the total token supply to the public. Swappable eMTV tokens will be minted and issued on Ethereum blockchain during the event. Contributors can acquire this tokens using ETH, BTC, LTC, and Credit Card payment gateway

- Token Name: eMTV
- Total Supply: 141, 000, 000
- Price: 1 eMTV = $1
- Soft Cap: $5,000,000
- Hard cap: $35,000,000

Token Allocation Structure

- 76% Contributors
- 15% Founders and team (blocked for 12 months)
- 6% Collaborators
- 3% Advisors, Partnership

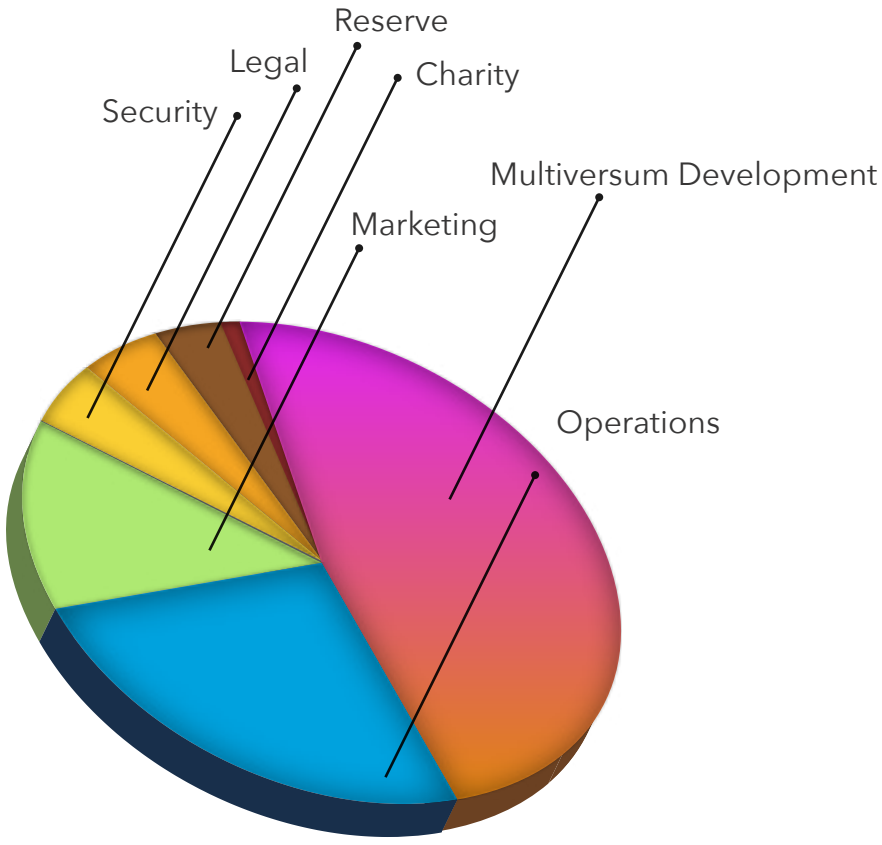


Multiversum will mint 141,000,000 ERC-20 compliant tokens, out of which 76% will be offered and distributed to contributors' ETH addresses no later than 10 days after the token sale event. After the event, no other eMTV tokens will ever be minted.

Unsold tokens will not be burned, rather, it will be rolled into the airdrop pool and distributed to token sale contributors who have not moved their tokens over a period of 18 months. Airdrop will occur once a week and will start 2 weeks after the end of the token sale event.

Funds Utilization Distribution

- 47% Multiversum Development
- 25% Operations
- 12% Marketing
- 5% Security
- 5% Legal
- 5% Reserve
- 1% Charity



Pre-ICO

Multiversum private sale will be offering eMTV tokens to early participants to facilitate main token sale launch and expedite development. Tokens will be offer as follows:

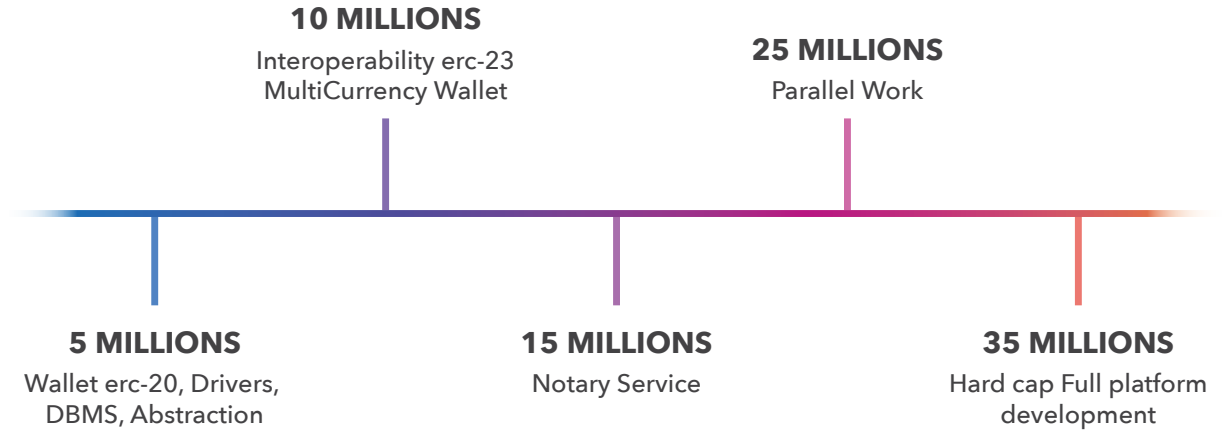
- Supply:
- Price:
- Bonus: 30%
- Duration:

Main ICO

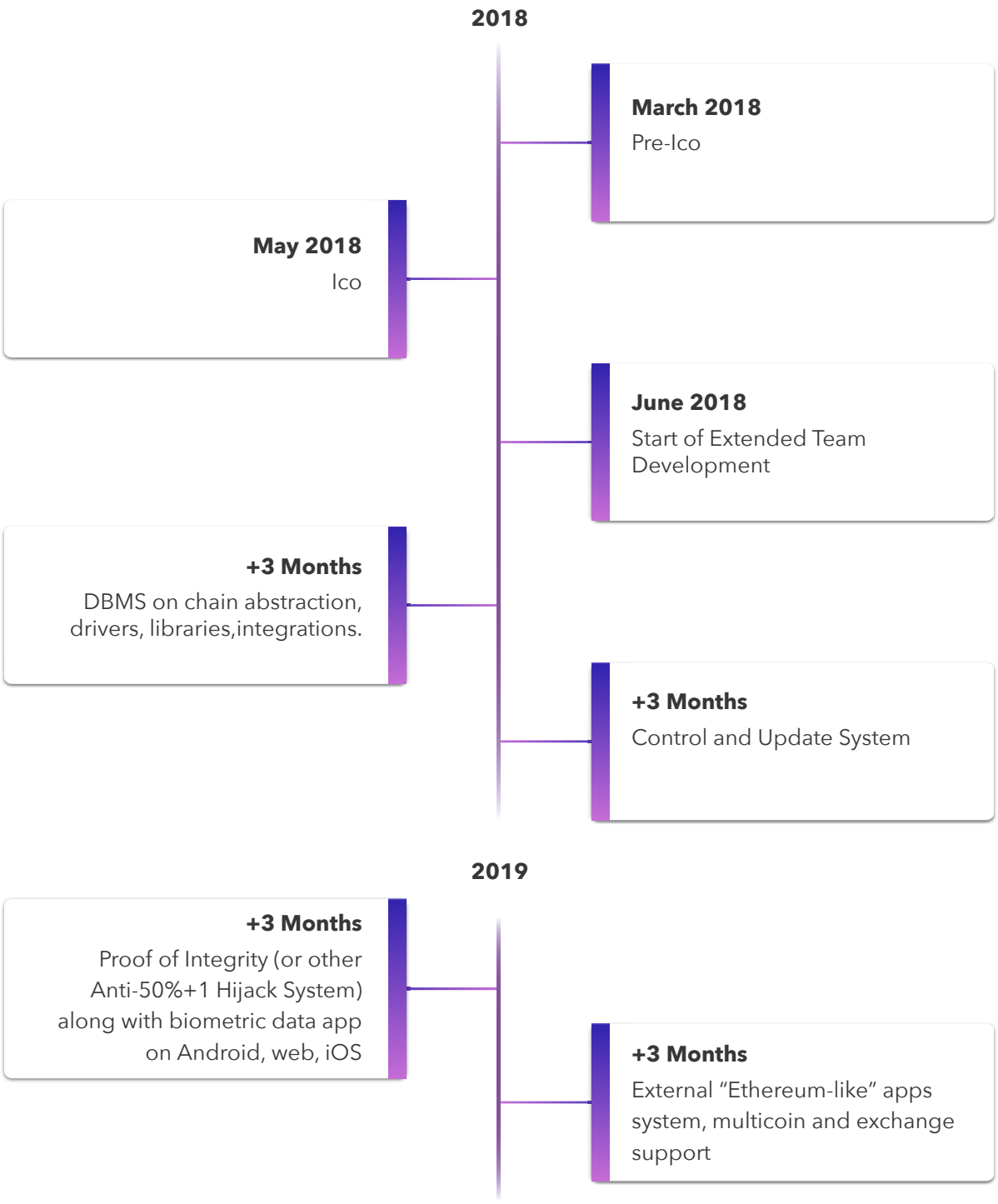
Multiversum main token offering kicks off on May 1st and runs through to June 10th. Hard cap for the event is $35,000,000

- Minimum Contribution: $1
- Bonus: Starting from 20%, bonus decreases by 5% every two weeks
- Duration: 40 Days
- Referral Commission:

Token Sale Targets



Developmental Road Map



Technical Section

The Relational Database in Blockchain
The challenge of the Relational Database in Blockchain

Creating a Relational Database in BlockChain means creating an archive of data where each record is validated by a signature inserted in the next record (hash) and for which the second record validates the first and every single record in the chain becomes immutable without breaking the chain.

This however is not sufficient for a corporate or government relational database, where information from citizens or customers must be constantly updated. This problem could have been solved by inserting different states of the record (without deleting old ones but adding new ones on top of one in a stack), however, even this would not have been enough since a Relational Database bases its importance on creating links between different records. A relational database on the Blockchain must be able to defend these links with immutability.

The development of this solution had one major challenge facing it and that was the fact that any record in a new state should have a different reference to the linked state of another record. In turn, this would lead to saving every new state in a new hash, eventually resulting in bloat on the blockchain, making it excessively cumbersome to manageable due to its computational requirement.

To solve this conundrum, Multiversum divides each record into two parts: a fixed part (for example: for acar the record identifier, the chassis number, the brand, etc) and a series of

states referencing the fixed part. The data that may vary the status of the record after other status must be protected by the BlockChain on the states links to the objects. However, they must also reference the fixed part of the record, which therefore does not change and does not cause a cascade update thanks to visibility with direction.

Multiversum employs a clearly definable, fundamental mechanism to store and record information. Every object on the BlockChain will be annotated with @Chainable. The Fixed class will be annotated with @DataRoot, the variable states with @DataState and each @Chainable will testify multiple hashes of records previously inserted in BlockChain.

Persisted objects, inserted into blockchain, will look for the hashes of the following records:

- In case the record is a @DataRoot or a Transaction registered in @MainChain, the object will look for the last transaction of the @MainChain
- If the record is a dataState, it will find the last @DataState referring to the same set of records connected to a @DataRoot
- In case it is the first @DataState, the hash of its @Dataroot
- If the records are inserted in more tables, each one will constitute its own sub-chain.

In order to verify the hash, hashes must be inserted within the same order and by obtaining the previous hashes in a multi-tasking environment, a sub-chains can reconnect to a transaction subsequent to the one immediately following.

This mechanism more accurately constitutes a Hypercubic Relational Database, i.e. a multidimensional Database (hence the name Multiversum) where the data have a knitted structure that binds them in a three-dimensional network. However, to this new structure is added a timeline where in every instant the status of each single data of the whole archive is defined. Through the BlockChain, this data can be certified at any time with a mathematical test thanks to the signature of the hashes in the chain.

This is an evolutionary concept of the Blockchain, hence the need to interpret Multiversum as a next generation solution, a new and innovative mutation, replacing the Neanderthal blockchain the same way Homo Sapien had.

Data Structure of the other Blockchains

Multiversum's enterprising solution clashes with the banality of the data structure of all previous generations of Blockchain. These generations of blockchains employ a sort of

rock painting-like proof of the existence for data, a way of notary certification of a state of a given, unable to be reworked into complex queries, and can only be searched by scrolling through the whole chain. These blockchains can only be used for very simple logics, serving as just proof of the existence of a hash of a given.

It's prudent to point out the difference between a written and a given. The written is a part of knowledge can be kept anywhere and whose reachability, accessibility, usability (ability to correlate, reuse, rework, etc) is difficult, expensive, inaccurate. A perfect analogy for this would be a magazine stored in a drawer. It's true that the magazine exists in the drawer, however, if there's more than one drawer then it may cost time and energy finding the right one. After finding the right drawer, we then have to contend with locating the right article - or worst, the right line - among unrelated items costing us more time and energy.

The given, on the other hand, is a part of knowledge that is correctly stored in an archive (database) where it can easily be found, reworked by looking for relations already made or derived from others, and transformed. Multiversum is the database.

The Structure of an Application on the Blockchain

A traditional elementary application is composed of three levels (tier).

- The graphical user interface (GUI), where information and data are entered by the user
- The Server with the Business Logic where the data (entered and retrieved from the database) are reprocessed according to the logic of the software to be then shown and / or saved
- The rescue device (Persistence) where the data is saved and then read when it is required.

Between Business Logic and Persistence, modern programming techniques views the presence of a component that is simply imported into the program and which contains software libraries that can be reused by the Logic to avoid having to do everything from scratch as an ORM library (Object Relational Mapping). ORM sole function is to transform a Java or .NET object - reworked by the Business Logic and present in the volatile memory - into an instruction that can be understood by a database to save that object in it.

This instruction is a string that constitutes an SQL Script (Query Language), the standard language used by relational databases to insert, modify, delete records on a Database.

The Multiversum Library

Multiversum Library is a super-set of ORM, a library with standard JPA interfaces powered by proprietary functions for transforming an object in the memory into a compatible structure with the values of the Blockchain. By breaking every record in two parts - a fixed one (if it already does not exist on the database) and a series of variable states, and managing the relationship between the records (from state to fixed part), Multiversum mediates all the complexity of the Blockchain for saving, readings, and updating data.

Any programmer developing l distributed applications does not require any indepth knowledge of how the BlockChain works when using our ORM libraries. The developer will be able to treat the BlockChain as a simple rescue device and decide where to persist their data: on normal database or on the Blockchain by simply changing the library in use. Multiversum not only facilitate the seamless integration of existing application into a decentralized ones, but also enables the creation of new ones from scratch at no additional charges. This drastically lowers the cost and time for developing, deploying, and marketing decentralized applications by more than 95%, reducing risk.

How to Use

The Multiversum library allows fast learning by leveraging the inherent, traditional knowledge of programmers. The blockchain employs two distinct subclasses - the fixed part and the variable part - and annotation to secure records.

Properties that are to be protected on through the BlockChain will be annotated with @ Chained, while those to be excluded from protection (for example to freely modify a part of the record for privacy reasons) will be annotated with @Unchained. Links from the DataState to the DataRoot will use the @OwnDataRoot annotation for association. Links to the DataRoots of the other records will employ @Chained if they are to be protected and @Unchained if they can be freely edited.

The rest of the operations on the records (queries, save, update) are managed invariably from the Multiversum library, creating scripts in SQL in the appropriate format and distributing them remotely. Here is an example of how little it is to change the data model on Traditional Database and how to Decentralized Database on BlockChain:

Per DB Tradizionale: Traditional DB	Per Db in BlockChain Decentralizzato = Decentralized Blockchain DB
DataRoot DataState	

With a simple textual configuration file, Multiversum libraries allows users to set whether they want to use a local database (easy to manage for the development and test phases), the distributed Global Ledger or on a private Ledger (for corporations and governments), or the private / global hybrid (to have a public certification of status hashes.

Creating distributed applications

Multiversum simplifies the creation of own spaces on Remote Relational Database Managers. The same database generation script is locally transformed in such a way as to manage records in BlockChain before passing it through an interface to the remote cluster. Afterwards, primary authentication checks are performed in order to send it to the nodes appointed to contain that particular database where each one then creates the application space inside it.

In this phase some aspects of the opening contract of the Data Space are still to be defined, such as minimum levels of distribution, minimum levels of consent required, priority, public key, etc. This information will determine the cost of inserting and reading data on the Ledger together with other parameters (use of the CPU for reading queries, space occupied by data in writing). Everything can be managed by a simple utility distributed with the Enterprise and Government versions or by hand for the Pay Per Use or BlockChain as a Service version.

The Pay-per-Use model

Small and medium-sized businesses, associations and individuals who want to save their data on the BlockChain are generally adverse to the challenges and cost of managing and maintaining a cluster of tens, hundreds or thousands of nodes. Hence, the reason why it is more convenient to use Multiversum's global public Ledger.

When their application does not require the blockchain, there is no cost to use and when it is time to chain data, the cost to use is very minimal. This provides huge savings and an ease of starting the business incomparably more accessible than making its own infrastructure.

Special features for corporations, financial institutions and governments

Due to restrictive legislation on the persistence of data imposed on government agencies, financial institutions and corporations, as well as having control over the ledger, some or-

ganizations may request a private ledger. This will be possible by acquiring a number of tokens to exchange for the license to use.
The corporate version will also have at its disposal the native functions of data encryption queryable on the blockchain.

These organizations will also have a support service with paid SLAs, consulting from private companies, and utilities that include tools for code migration, data migration, cluster analysis, hardware and software error forecasting, prediction of the growth of cluster resources, backups, tests, reports, native Identity and Access management, Data Access matrix, query and insert auditing, native PKI and many more to be defined.

The governance version, in addition to incorporating all aspects of the Corporate version, will have special functions related to the needs of Governments and Agencies, including the functions necessary to protect witnesses, anti-terrorism, privacy, accessibility, native voting infrastructure, etc.

Interoperability between databases on MTV

Each Database on Multiversum will be able to declare on the opening contract of the Data Space the level of access: public, private or paid. Through public interfaces it will (or not) make its data available and its structure to various titles.

Paid accessibility provides a native function for which the data collector will pay for a feed that covers the costs of querying or writing, rights for the application owner.

This opens up interesting scenarios of applications that independently sell data for the Internet of Things or other (collections of photographs, medical data, scientific data between universities and laboratories, online newspapers, etc) and reward the data owner.

Proof of Integrity

Proof of Integrity is based on two chains: the Mainchain with decentralized databases and a Side Chain that acts as a decentralized Public Key Infrastructure (dPKI).

This Side Chain dPKI will have a secure protocol for entering transactions (PoW, PoS, DaG or other, to be defined) derived from a fork without unnecessarily investing resources and energy.

This dPKI will contain the identifier of the application connected to the public key previ-

ously entered by the owner of the app. Each node will contact it to verify the authenticity of the data entered by the application connected to the database. Due to the fact that the public key usage model is write once, read many (write once for each application and read many) the side ledger's performance is not important. Instead, it will guarantee extremely high performances in the main chain, even superior to those of the original concept.

The Consensus on the Main Chain

This consensus system on the insertion of new data makes gossip (the exchange of information between the nodes for the verification of consent) useless because the verification is delegated to the SideChain through the signature of the owner and his public key: it does not therefore constitute a security risk.

The Multicoin Wallet

In order to guarantee the permeability between Multiversum mainnet and the other chains, a transaction buffer has been devised between these systems.
In this buffer, tokens from other blockchains (ETH, BTC, MONERO, any ERC20) will be deposited with the same amount of mToken placeholder of the same token/coin (always with a 1:1 exchange ratio) will be sent to the recipient's wallet in mETH format, mBTC, mMonero, mERC20 instantly.Tokens and coin placeholders can then move between Multiversum Multi-currency Wallets at a fixed cost of 1 cent instantaneously.

Should a payment be made from a Multiversum wallet to an external one, the mToken will be sent from the MTV wallet to the buffer, and the corresponding token will exit the wallet to the recipient's address. The same mechanism will be employed for exchanges between MTV and eMTV I'm any direction and always at a fixed exchange rate of 1: 1.

This system makes it possible to envisage the adoption of a financial institution license under Lithuanian law and the issue of a Multiversum SmartWallet credit card.

The layers of elements covering the universes are each ten times thicker than the one before, and all the universes clustered together appear like atoms in a huge combination

Bhagavata Purana 3.11.41

MULTIVERSUM

HERE TO STAY