by Harsh Patel, Ankit Malik, Kazım Rıfat Özyılmaz and Intrepid Ventures

White Paper

Metabase

A blockchain platform to build, scale, and monetize next-generation businesses

v1.0.8 – March 2018

Table of contents

Introduction

First-generation blockchains like Bitcoin and Ethereum have revolutionized philosophical notions of economic trust, political power, and social control. They have laid the foundations for how networked systems, web services and digital communities of the future are to be designed and built. However, as the ecosystem evolves and hits the next level of adoption with more use-cases being envisioned - these blockchains are hitting their technological and design limits.

Challenges with First-Generation Blockchains

We have identified that first generation blockchains have hit challenges across the following dimensions, constraining real-world adoption:

01 / **Scalability**

This is a real challenge for blockchains - both crypto-currencies and platforms have hit adoption limits because of scalability issues. In the case of Bitcoin, scalability issues have given rise to significant transaction fees, clogged mempools and long drawn debates resulting in forks and community splits. Ethereum's ambitions are currently constrained by the transactions per second (TPS) factor. For it to replace Visa, it should have a TPS of 45,000; however at its current TPS of 15 - one popular application like CryptoKitties suffices to make the network unusable.

02 / **Limited Programming Ecosystem**

As a first-generation, blockchains have established a base programming ecosystem. But with increasing evolution, this ecosystem appears to be constrained by limited built-in primitives. The burden of programming constructs is mostly on the application layer. Certain use-cases have hit a dead wall because identity constructs are not an intrinsic part of the blockchain design.

03 / **Security and Usability**

Due credit needs to be given to first-generation blockchains for their protocol layer security. Very few instances of the Bitcoin protocol or the Ethereum protocol being vulnerable have been reported - which bodes well for their security considering these networks have now been around for years. However, application layer security has been inadequate - because of which we have seen multiple incidents from the Parity Multi-Sig Wallet issue (which led to 500,000 Ether being stuck) to the infamous DAO attack - which caused an irrevocable forking of Ethereum into two different blockchains and communities.

Developer tools need a major upgrade for the next round of adoption and evolution. Existing developer tools are basic and often require hacky third-party tools or centralized platforms to fill the gaps. Also, light client implementations targeting mobile platforms are needed to enable massive deployments.

04 / **Governance Mechanisms**

First-generation blockchains - being the first-generation - didn't foresee governance challenges that a decentralized system, with no central party calling the shots, will face.

Hence, we had the Bitcoin scaling debates and the Ethereum forks which illustrate that governance mechanics should be part of the blockchain protocol.

To give credit where due, first-generation blockchains have been instrumental in establishing the philosophy and initiating a movement and have paved the way for second generation blockchains to drive the adoption.

Introducing Metabase

With Metabase, our vision is to build a blockchain platform to build, scale, and monetize next-generation businesses.

In this whitepaper, we describe the technical aspects of Metabase - providing technological solutions and innovations to overcome the challenges discussed above. As we mentioned the challenges of the first-generation blockchains, namely - Scalability, Limited Programming Ecosystem, Security and Usability and Governance. Here is an outline of how Metabase is planning to address these issues.

01 / **Scalability**

> Metabase - a scalable blockchain protocol and a scalable cryptocurrency inspired by real-world economics.

In the Challenging Scalability section, we explain why solving scalability requires redressal at two levels - a technical level and an economic policy level. In the Protocol section, we address scalability at a technical level by describing an innovative next-gen blockchain protocol that scales with adoption. In the Monetary Policy section, we address scalability at an economic policy level by introducing a new monetary policy - taking inspiration from real-world economies. Finally, both these levels incorporate parameters that will be subject to well-defined Governance mechanisms.

02 / **Limited Programming Ecosystem**

> Metabase features at its core - a next-generation virtual machine - expanding blockchain programming capabilities.

In the Virtual Machine section,

we describe the core of Metabase programming capabilities. A Virtual Machine 2.0 with an augmented OpCode Space - unleashing a rich programming ecosystem which enables programming on high-level programming languages for end users, low-level programming languages for organizations and identity systems within the core blockchain.

03 / Security and Usability

> Metabase is designed with security and usability in mind.

With a goal to improve application layer security, we are building Metabase from the ground-up to provide a secure platform for application layer programmers. In the Virtual Machine section, we describe how Metabase will incorporate standard security constructs into the OpCode space itself - reducing the burden of application programmers. An augmented OpCode space will also make blockchains easier to program.

The Tools section describes the software and tools that will be built for developers - keeping usability and security features in mind. Metabase aims to provide light client implementations both on mobile and browser platforms to create seamless integration for blockchain-based applications.

04 / Governance

> Metabase features a governance framework toenable a controlled evolution of the blockchain.

A blockchain should be like a living specimen that evolves over time. Under the Governance section, we describe a framework to enable a controlled evolution of Metabase - with the process guided by open consensus of the participants. Governance of Metabase encompasses areas from Metabase Improvement Protocols to controlling base axioms of the system.

Hence, with these features, Metabase will overcome limitations of first-generation blockchains and enable programmers and entrepreneurs to build their blockchain based solutions on a rich, scalable, secure, easy-to-use and evolving platform.

Challenging Scalability

Currency and its purposes

The current Bitcoin scaling crisis and high transaction fees have brought to limelight the economic intricacies of a currency. A currency can act as a medium of exchange, a store of value or both. For example, in modern times, the US Dollar has acted as a medium of exchange for international trade. It has also acted as a store of value, especially for people in volatile economies. Before the advent of paper currency, gold and silver coins acted as a 'store of value' while less precious metals like tin and copper coins provided for 'medium of exchange' coins.

As the financial ecosystem around a currency evolves, its primary purpose can toggle between 'store of value' and 'medium of exchange'. Bitcoin started as (quoting Satoshi's white paper title) an electronic peer-to-peer cash system - primarily suggesting a 'medium of exchange' purpose. However, because of its rising popularity and adoption combined with a deflationary supply - it keeps becoming more useful as a 'store of value' - and less useful as a 'medium of exchange'.

Bitcoin's woes

The economic dilemma of a cryptocurrency is best articulated by Bitcoin's tricky situation. For it to be useful as a store of value - its demand and adoption should keep rising. This along with its deflationary nature will ensure Bitcoin keeps rising in value (vis-a-vis other currencies like USD). However, this has a double negative impact on its purpose as a medium of exchange.

Impact I

Transaction fees are typically a function of the currency. If the value of the currency rises (say, with respect to USD), then so do the transaction fees.

Impact II

More adoption means more transactions competing for the same block, hence jacking up transaction fees in BTC terms.

Solving these impact points requires solutions across technical and economic dimensions.

Metabase and Challenging Scalability

Impact I needs redressal at an economic policy level - for Metabase, we address this in the Monetary Policy section.

Impact II requires addressing the technical issue of scalability or transaction throughput. In the Protocol section, we introduce how Metabase throughput would scale with increasing adoption to provide stable transaction fees.

Monetary Policy

Motivation

Scalability needs redressal at a monetary policy level as the purposes of 'store of value' and 'medium of exchange' have somewhat contradictory demands from a currency.

In the real world, this is addressed by having different 'currencies' for these two purposes.

For, eg. in a real world economy.

The native currency acts as the 'Medium of Exchange' currency.

Transaction fees are a function of the native currency. Supply of the native currency is typically inflationary and is regulated by a government body (with an eye on currency competitiveness, inflation, etc). The government maintains the currency's status as the 'medium of exchange' by enforcing taxation and transactions in this currency.

'Gold' acts as the 'Store of Value' currency.

By gold, we refer to the varied pool of assets that economic participants look to preserve their wealth in the face of the inflationary native currency. Typically these assets are deflationary or limited in supply. For example, gold, art pieces and more recently, Bitcoin and other crypto-assets. Using this currency for transactional purposes is economically and/or legally disincentivized by government regulations or extraneous taxation.

Hence, participants transact in an inflationary currency but store their wealth in a second deflationary currency. Taking inspiration from real-world economies, we have incorporated two currencies into the Metabase economy - Metacore and Metabit.

Metacore - Store Of Value

Metacore is the Store Of Value currency of Metabase. It is the primary token of Metabase and will be issued in ICOs and traded on exchanges. (For distribution statistics, please refer to the Token Sale Economics document). Metacore can't be used as a medium of exchange currency on the blockchain. To pay transaction fees on the blockchain, a Metabase node converts Metacore to Metabit tokens using the prevailing exchange rate and then pays the appropriate fees.

Metacore tokens issued via mining will have a deterministic reducing/deflationary supply - similar to Bitcoin but with different generation parameters. Generation parameters of Metacore tokens are hard-coded into the protocol and can't be modified by Governance Mechanics.

Metacore Tokens Issuance - as Block Rewards.

Total count of Metacore tokens issued via mining/block rewards will be determined after the Pre ICO event (refer Token Sale Economics document).

Details of Metacore tokens generated as block rewards are as follows:

01 / Block reward per block for the first 2 Million Blocks are fixed at a constant denoted by $Reward_{max}$

02 / Block reward per block halves after every 2 Million Blocks, hence block rewards for different block numbers look as follows:

Block Numbers	**Block Rewards For Each Of These Blocks**
0 to 1,999,999	$Reward_{max}$
2,000,000 to 3,999,999	$Reward_{max}$ / 2
4,000,000 to 5,999,999	$Reward_{max}$ / 4 ...
...	...

03 / Each duration where the block reward is constant is referred to as a 'Halving Epoch'

04 / The duration of a 'Halving Epoch' is denoted by $k_{halving}$ = 2 Million Blocks

05 / $Reward_{max}$ will be a constant value determined by token issuance in Pre Sale (refer Token Sale Economics document for a sample calculation).

Hence, block rewards per block for each of the halving epochs are as follows:

Halving Epoch	Block Numbers	Block Rewards For Each Of These Blocks
h=0	0 to 1,999,999	$Reward_{max}$
h=1	2,000,000 to 3,999,999	$Reward_{max}$ / 2
h=2	4,000,000 to 5,999,999	$Reward_{max}$ / 4
h=N	(2 Million * N) to ((2 Million * (N+1)) - 1)	$Reward_{max}$ / 2^N

Note: The block rewards pattern is similar to Bitcoin, but with different parameters

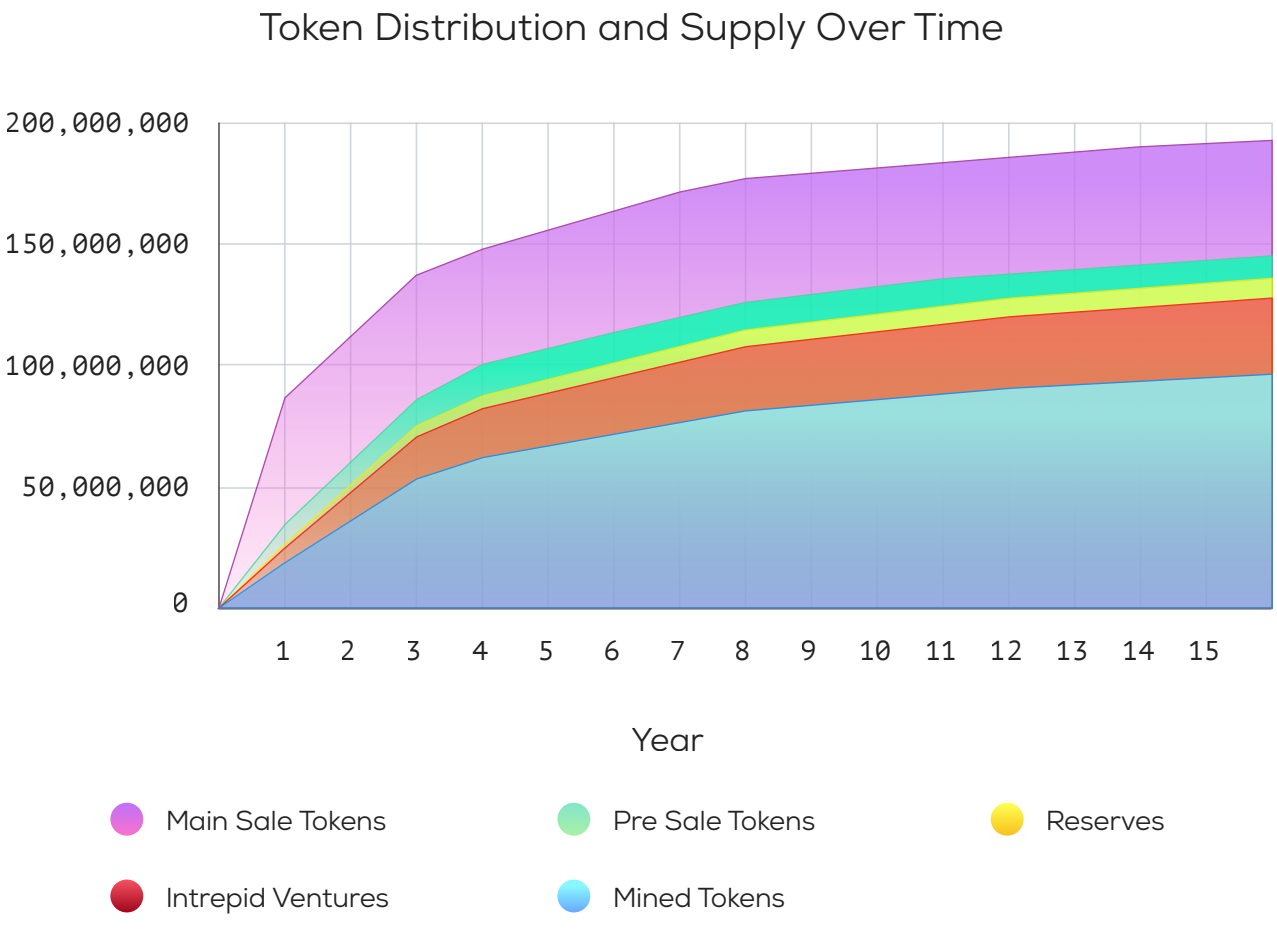
	Bitcoin	Metabase
Block Rewards in First Epoch	50 BTC	$Reward_{max}$ MBT*
Blocks in Halving Epoch	210,000 (~ 4 years)	$k_{halving}$ = 2,000,000 (~3.8 years)
Total Mined Supply	21 million	10x of Pre Sale*

The following graph represents token distribution and network supply over time for all participating groups, assuming

01 / All buyers (Pre Sale and Main Sale) have a 1 year vesting schedule

02 / 1 Year = 525, 600 Block Numbers

03 / 10 million tokens were sold in Pre Sale and hence, $Reward_{max} = 25$

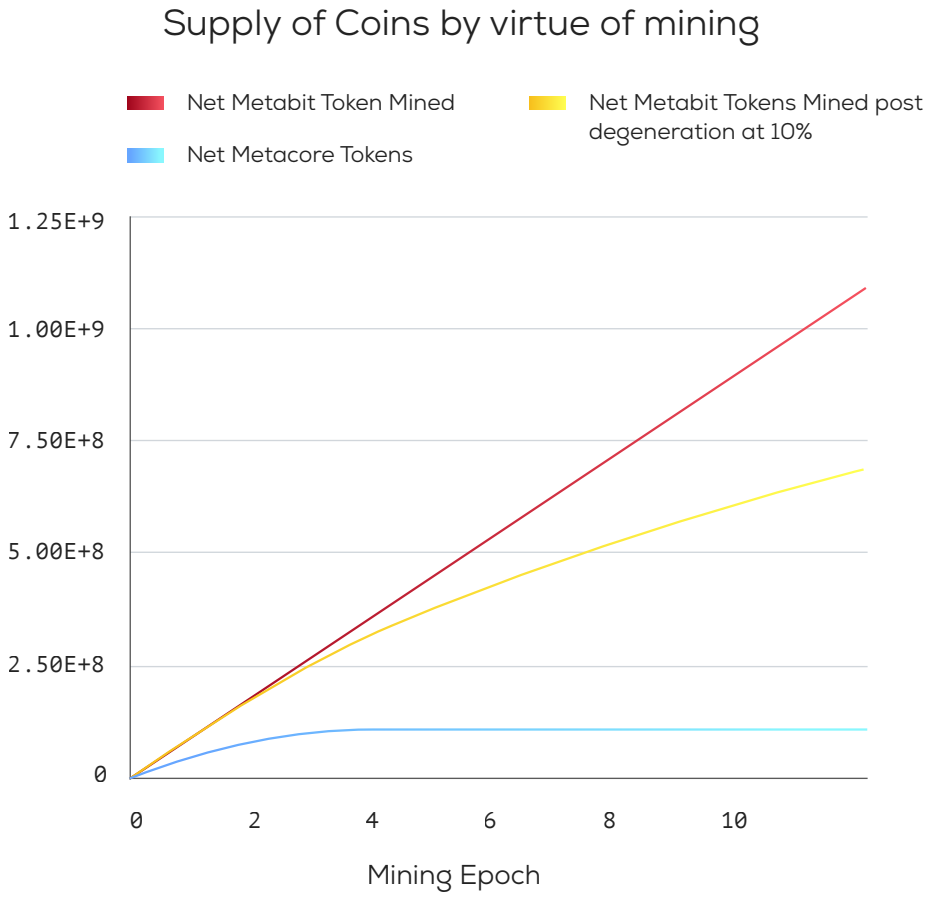


Metabit - Medium Of Exchange

Metabit is the Medium of Exchange currency of Metabase. It is the transactional token of Metabase and will be used to denominate transaction fees. Metabit as a currency is inflationary in supply therefore creating a decreasing exchange rate with Metacore over time. The inflation rate of Metabit is dynamic and is governed by the governance framework of Metabase.

Based on supply of Metabit and Metacore, there is a rate of exchange derived between both tokens. All transactions of Metabase have a static transaction cost in terms of Metabit, but when the rate of exchange is applied, the overall transaction cost is reducing in nature with respect to Metacore.

For the mathematical details of Metabit supply, the transaction costs and the prevailing exchange rate between the two currencies - refer to the detailed document on Monetary Policy of Metabase.



Graph denoting total number of currency in existence over time for both Metabit (with different inflation rates) and Metacore tokens.

Protocol

01 / Solving Technical Scalability With An Innovative Protocol and Blockchain Design

02 / Introducing Metabase - a scalable blockchain protocol

Introduction and Motivation

In this section, we introduce the core of Metabase - a scalable blockchain protocol.

Conventional blockchains are constrained in transactional throughput by the nature of the protocol and blockchain design. The primary design structure of most existing cryptosystems is a linear linked-list style blockchain. As adoption increases and more miners are attracted to the system, all mining capability is dedicated to mining the one next block in the linear blockchain. Therefore, the increased mining capability doesn't facilitate scalability at all.

We are still mining the same number of transactions per block / per block generation time.

We are proposing a blockchain structure that looks more like a directed binary tree than a linked list.

Initially, the Metabase blockchain is linear in nature. As adoption of the Metabase blockchain increases, the transactional load of the chain increases.

When the transactional load hits a certain high threshold consistently - we split the blockchain structure into two chains - mathematically and evenly distributing mining capability and transactional load across the two chains. So now we have two blocks being mined simultaneously - one for each of the chains.

When the transactional load becomes high in the blockchain (with n chains), we further split the blockchain into 2n chains. Hence, over an extended period, the blockchain resembles a directed binary tree with "chain-split block instances" as nodes and "chains between epochs" as the edges.

Transactional throughput in the multi-chained Metabase blockchain

Scaling Epoch	Number of Chains	Net Transactions across all chains at one Block Number	Net Transactions Per Second (TPS)
0	1	1024	17.06
1	2	2048	34.13
2	4	4096	68.26

Scaling Epoch	Number of Chains	Net Transactions across all chains at one Block Number	Net Transactions Per Second (TPS)
3	8	8192	136.53
4	16	16384	273.06
5	32	32768	546.13
6	64	65536	1092.26

A natural question follows - "How do you prevent double spend across chains".

As discussed in the details below, the proposed chain split mechanism creates a parallel mining structure operating on multiple chains. However, this is only a separation on write-activities on the blockchain - all nodes still get a full view of the blockchain.

Metabase nodes will store the complete copy of the blockchain at any given time, contained on the split chains. Combined with the complete view of the mempool, Metabase nodes will be able to detect and reject double-spend attempts just like a traditional blockchain.

In the following sections, we detail out the Metabase protocol.

Network Layer

Similar to existing blockchains, the Metabase blockchain is run and maintained by nodes running a Metabase client over the Internet. Each node within the network will have a Node ID.

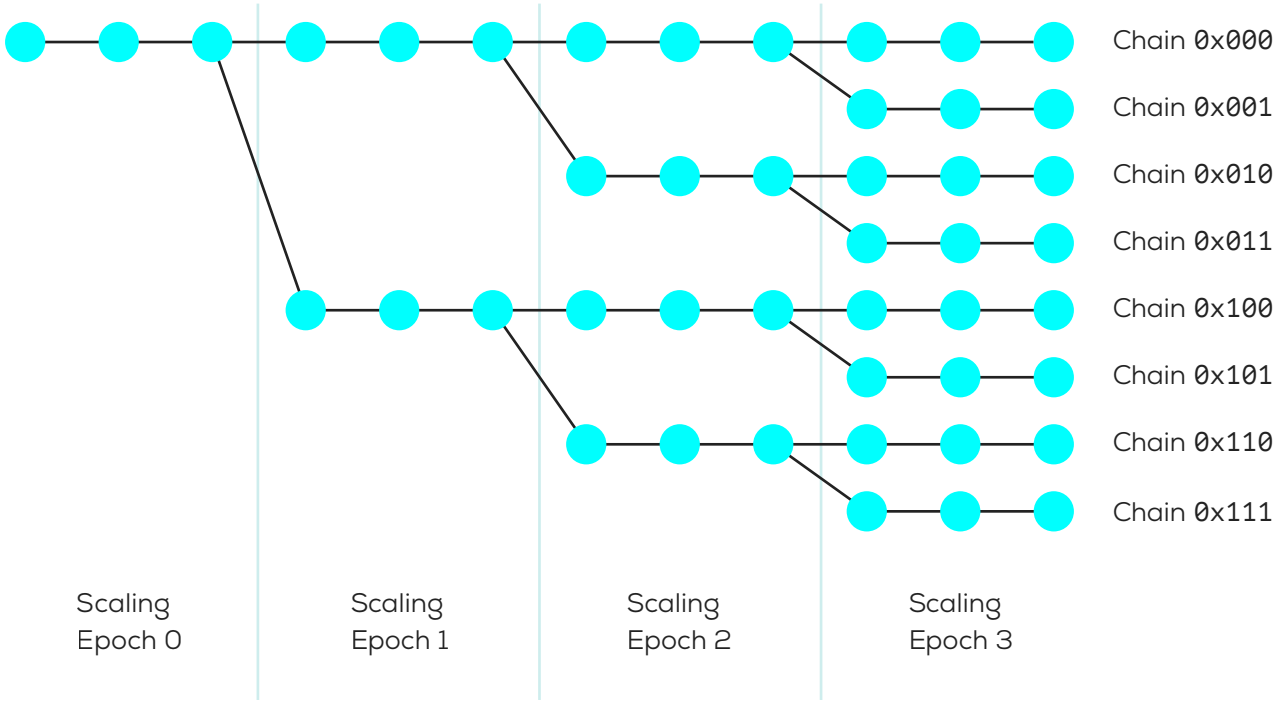
01 / When a node runs the Metabase client for the first time - a private key is generated by the client for the node along with a corresponding 256 bit address-like derivative. This 256 bit number acts as the Node ID for this node.

02 / The private key used for Node ID generation remains static until the node does a hard reset.

With a Node ID, a Metabase node identifies itself to the network and signs messages to the network. The Node ID is also used to determine which chain on the multi-chain Metabase blockchain the node can mine on.

Probabilistically speaking, the formula ensures even and random distribution of mining nodes on the different chains of the blockchain. Every node maintains a database which stores information about peer nodes and also which node belongs to which Chain ID in the current chain-split state of the blockchain.

Blockchain Layer



Block

The Node ID determines which chain the node would listen, based on scaling epoch. Based on Scaling epoch miners identifies hashes and club them into blocks of respective chains. (i.e. Transactions starting with 0x111 are part of chain ID 0x111 for epoch 3 and for epoch 2 these transaction would be in blocks of chain 0x110.)

Graphical representation of the chain.

Blocks

In a linked list style blockchain, each block is identified by the Block Number. In Metabase, each block has a Block Number, Epoch Number, and Chain Number.

Epoch Number

Starts at 0. Increments every time a chain splitting event happens. If a block has Epoch Number E, then it exists at the epoch level where the blockchain has 2^E chains.

Chain Number

If a block has Epoch Number E, then it exists at the epoch level where the blockchain has 2E chains, so that it will have a Chain Number (between 0 and 2^{E-1}) identifying its chain.

Block Number

Genesis block starts at 0 and increments as 1,2,3...N. Unlike traditional blockchains, block number doesn't uniquely identify a block. If a block number N is at Epoch Level with Epoch Number E, then there are 2^E chains at this Level, and hence 2^E blocks will

share the same lock number N.

Epoch Number	Number of Chains	Number of transactions in one block (with priority pools at 10)	Transaction capacity / Min	TPS
0	1	1024	1024	17.06666667
1	2	1024	2048	34.13333333
2	4	1024	4096	68.26666667
3	8	1024	8192	136.5333333
4	16	1024	16384	273.0666667
5	32	1024	32768	546.1333333
6	64	1024	65536	1092.266667
7	128	1024	131072	2184.533333
8	256	1024	262144	4369.066667
9	512	1024	524288	8738.133333
10	1024	1024	1048576	17476.26667

Block ID uniquely identifies a block with respect to its position and is represented as follows

Block ID					
Epoch Number		Chain Number		Block Number	
Max Size	4 Bits	Max Size	16 Bits	Max Size	Inf

Chain Splitting and Epochs

Each Metabase block has a defined cap on the number of transactions (i.e., 65536 transactions per block in case of 16 priority pools).

When the blocks are 95% full - considering the moving average of transaction count in last K block numbers (across chains) - the system is programmed to increment the epoch level from E to E+1 (the algorithm is similar to how the difficulty is adjusted across the bitcoin blockchain) and further split the blockchain from 2^{E} chains to 2^{E+1} chains.

When this new epoch starts, the mining nodes and transaction space are evenly re-distributed across the new set of chains. Node IDs and Transaction Hashes are critical identifiers in this context.

01 / Transaction hash for a transaction would dictate what chain would mine it in this epoch period.

02 / Node ID would dictate what chain a miner can mine on in this epoch period.

K above is an algorithmic parameter and miners can start announcing the next epoch level in the last 10000 blocks leading to the scaling event and others can validate it (hence consensus over what is going to be the next epoch level).

Blockchain Transactions and Validation

Achieving consensus across multiple chains that are handling transactions for a varied set of transactions is critical to the integrity of Metabase. The following concepts enable transaction validation and prevent double spend attacks across the blockchain.

As stated in the Network Layer section every Metabase node has a 256-bit address called Node ID derived from a private key created at node installation. As a result, miners in the Metabase system will also have a Node ID that determines on which chain they can mine.

Here is the flow of how mining and block validation will work:

01 / Miners will keep track of the mempool, so they see all the unconfirmed transactions.

02 / Miners pick a set of transactions depending on their "Node ID" e.g. transactions should belong to the same split chain as miners' Node ID.

03 / Miners check for double-spend attempts in the transactions (similar to checking "account nonce" values on Ethereum or UTXO presence on Bitcoin).

04 / Miners create a valid block by satisfying the Proof-of-Work (PoW) requirements which Metabase system will initially adopt. In later stages miners will participate in the voting/ witnessing process described by the Delegated Proof-of-Stake (DPoS).

05 / Miners (or witnesses for DPoS)

announce the newly created block to the network.

06 / Nodes pick up the newly proposed block and because all sub-chains (split chains) are stored in every node, block validity can be checked:

a. / Cryptographically using block header

b. / By parsing contained transactions and verifying that they belong to the same sub-chain (split chain)

c. / By ensuring that no double-spend is happening (similar to checking "account nonce" values on Ethereum or UTXO presence on Bitcoin)

07 / Nodes will add the proposed block to the respective sub-chain.

Transaction Costs and Priority Pools

In Metabase transaction capacity per block is defined in exponents of 2 which also defines the number of priority pools that need to exist for each block. Priority pool puts an upper bound on the transaction cost that can be given to a miner .

$$B_{tx} = 2^{p}$$

(Transaction capacity of a block is always in exponents of 2 and are in accordance to number of priority pools - p)

P Pool No	Tx Volume	Multiplication factor	Tx Cost in Metabits	Net Cost in Metabits	Metabit to Metacore Conversion rate*	Net Cost of Tx in Metacore
0	512	1	1000	512000	0.86	440320
1	256	2	1000	512000	0.86	440320
2	128	4	1000	512000	0.86	440320
3	64	8	1000	512000	0.86	440320
4	32	16	1000	512000	0.86	440320
5	16	32	1000	512000	0.86	440320
6	8	64	1000	512000	0.86	440320
7	4	128	1000	512000	0.86	440320

P Pool No	Tx Volume	Multiplication factor	Tx Cost in Metabits	Net Cost in Metabits	Metabit to Metacore Conversion rate*	Net Cost of Tx in Metacore
8	2	256	1000	512000	0.86	440320
9	1	512	1000	512000	0.86	440320
				Max Tx fee per block in Metacore		**3522560**

Ray Network & Spectrum Network

Lightning Network and Raiden Network are two protocols which implement instant P2P payments over existing blockchains. But in their current form, they are an externality to the system. Or in other words, they are not part of the base blockchain protocol - hence there are concerns about the security and stability of their payment channels.

Metabase will feature two P2P networks as part of the base protocol -

01 / Ray Network : this will enable users to invoke channels for instant P2P payments (similar to the Lightning Network)

02 / Spectrum Network : this will enable users to invoke channels for instant smart contract transitions.

Summary

Metabase will address scalability issues by introducing concepts of Split Chains, Priority Pools and Ray & Spectrum channels. By utilizing these mechanisms, the mining operation is parallelized, transaction capacity per block is increased and instant transfers become possible.

Metabase blockchain will split itself automatically if transaction capacity is heavily used. To start with, the blockchain would follow a PoW based consensus mechanism - similar to Bitcoin - where repetitive hashing is required to meet a difficulty target. Difficulty retargeting would be done based on averages of block generation times across all chains. The entire block reward would be alloted to the PoW mechanism.

Eventually, the protocol would be shifted to a DPoS + PoW based mechanism, and block reward would be split in 1:1 ratio between the two methods. The DPoS mechanism - similar to the one used by Bitshares - would enable stakers to delegate their mining roots to miners who can then sign the Merkle Root. Upon completion of DPoS based mining, the Merkle Root then goes through a PoW based mining process where a difficulty target is to be met.

On top of on-chain scaling mechanisms, Metabase will introduce a native, off-chain scaling solution to enable instant transfers between peers.

Essential parameters (Axioms) of Me-

tabase blockchain are as follows:

01 / Block Generation time ~ 60 Seconds

02 / Number of priority pools - 10 Pools (can be altered by Governance)

03 / Max Transaction handled by a block - 1024 (for 10 Pools)

04 / Type of Consensus - DPoS and PoW integrated

05 / DPoS Punishment ratio - 50%

Reward Split between DPoS and PoW - 1:1

Virtual Machine

Blockchain-based smart contracts have ushered in a new era of computational law whereby contracts are backed and agreed on by a blockchain which is unbiased and universally prevalent. Ethereum Virtual Machine was a significant step-up from the highly limited (by design) Bitcoin programming environment. However, with increasing adoption; the EVM has hit design limits and security pitfalls.

Metabase will feature an upgraded Ethereum style virtual machine. This virtual machine will feature an augmented OpCode space which will unleash a more vibrant ecosystem of programming on the blockchain.

Additionally, we will incorporate identity constructs into the Metabase virtual machine - making identity a first-class citizen on the blockchain.

Augmenting the OpCode Space

Traditional processors have a bus size limit - which is a physical limitation. For example, a 64 bit processor can process only 64 bits of information in one pulse cycle (one Hz). Hence, it can process at most a 64 bit long instruction set within one pulse. This limits the OpCode Space to 2^5 or 2^8 OpCodes (depending on processor), and the remaining space is allocated for the actual instruction set which is processed by the processor.

However, the "processor" of a blockchain is a "virtual machine". Hence this physical limitation does not exist as the physical processor is abstracted out.

Since the virtual machine is no longer bound by the physical limitation instruction set length, we can have variable instruction set length in the exponents of 2 (which can range from 1 bit to 65536 bits) and is denoted by Instruction set length. This can enable more flexible processing and advanced compiler designs that can process variable length OpCodes. For the Metabase Virtual Machine (MVM):

`01 /` we intend to have 2^{16} (I.e. 65536) OpCodes.

`02 /` each OpCode has a variable instruction set length which is denoted by the 4 bit field labelled "Instruction Set Length".

`03 /` The Instruction set length can vary from 1 bit to 65536 bits.

OpCode	Instruction set length	Instruction set
16 bit	4 bit	1 bit to 65536 bits

Graphical representation of an instruction in Metabase VM.

Unleashing a richer programming ecosystem

This augmentation of the OpCode space for a "blockchain processor" will be utilized in Metabase to build a richer computational law ecosystem.

Each OpCode along with associated instruction set forms a DAG (Directed Acyclic Graph) which represent all the tasks done to process the OpCode within one flow. Multiple such OpCodes forms the basis for assembly language. This structure is utilized by higher level programming language whereby assembly level code is made human readable.

For Metabase, we are not limited by the instruction set length. Hence we can define custom OpCodes which utilize existing OpCodes as DAG structures. Due to processor independent nature of the OPCodes and instruction set, we can use existing OpCodes, to define a new instruction set. The newly generated OpCode is subject to constraints defined for the stability of the system. This nature of MVM, requires us to develop a dynamic high-level programming which adjusts to the dynamicity of the low-level programming language.

Enabling Security By OpCodes - Security Constructs

Currently, security constructs that are widely used in smart contracts - eg. security constructs in standard token functions, multi-sig wallet functions - have to be defined by the programmer in the higher-level language (eg. Solidity in the case of Ethereum). This places the burden on the programmer to be more watchful of security bugs. With the MVM, we will convert security constructs and primitives into OpCodes and place them in the standard OpCode space. Hence, security constructs will be available to higher-level languages directly as inbuilt OpCodes - reducing the security burden on the programmers.

However, constructs of security need to be applied while defining the OPCodes and is determined by following rule sets.

01 / All custom OpCodes need to be accepted by governance protocol.

02 / All custom OpCode must use OpCodes from range 0x0000 to 0x9fff only to define its DAG structure.

OpCode Marketplace

Because the OpCode space is now expanded and open to customization, it enables Metabase to carve out a subspace for an OpCode marketplace.

On this subspace, organizations/ consortiums can buy and define customized OpCodes which can facilitate certain directed use cases for the ecosystem. The design of such OpCodes will be regulated by the governance frameworks of Metabase. Hence, for private players, it does not only entail buying out OPCodes, but also community approval of the OpCodes by a governance mechanism (for eg. DPoS). The following table elaborates on the division of the OpCode space.

0x0000	OpCodes for Backward compatibility of multiple VMs
0x1000	Base functions
0x2000	Advance Cryptographic functions and ancillary functions
0x3000	For defining standarts
0x4000	For Future use / Reserves
0x5000	
0x6000	Memory stack
0x7000	
0x8000	
0x9000	
0xa000	Publicly available Smart OpCode
0xb000	
0xc000	
0xd000	
0xe000	
0xf000	

OPCode Space Division

Since the publicly available OpCode Space uses existing OpCodes to derive functions, the newly derived OpCode would entail a cost - the sum of all the OpCodes used, which is transactional in nature. The OPCode marketplace is more like the marketplace for Internet IP addresses, TLDs or telecom spectrum. This market is kept open to the public, but ultimate utility would be for the public to build stable business models using Metabase.

Private OpCodes

A subspace of the OpCode space will be allocated for Private OpCodes. These OpCode slots can be bought by organizations/consortiums to place their OpCodes with the functionality abstracted out. This will enable such entities to incorporate private smart contracts on the public blockchain without losing confidentiality. Allocation of such slots would again be subject to Governance Mechanisms.

Inclusion of Identity within the VM

Identity systems in the current context are an externality to the blockchain/ cryptocurrency domain. They are not incorporated into the blockchain protocol or virtual machine - and identity constructs have to be built on the application layer by smart contract programmers. For Metabase, we intend to have a specific set of OpCodes reserved for identity constructs - enabling identity to be a first-class citizen on the blockchain.

Governance

Blockchain is a natural singularity for the Internet as the nature of blockchain is to facilitate immutability and transparency. A blockchain's rules - once defined - are very difficult to refute or alter once the system picks up momentum. So, it is important to embed governance into the structure of the blockchain which acts like valves and knobs to divert the flow of the blockchain.

Governance Aspects

The following governance areas would be embedded into the Metabase blockchain

01 / R & D Governance - Metabase Improvement Proposals

02 / Economic Governance

03 / Axiom Governance

04 / Virtual Machine Governance

All of these areas would be governed by the Metabase Governance Protocol.

R & D Governance - Metabase Improvement Proposals

Improvement proposals are submitted by developers to advance, optimize or improve the current functionality of the blockchain. Bitcoin lacks incentivization logic for core-developers hence the system stagnates over time, and the development dynamic becomes oligarchic over time. In Ethereum, the improvement protocols are open. But most of these proposals are funded by the Ethereum Foundation. This introduces a "centralization" in the development process of a system that strives towards decentralization in all aspects.

For Metabase, we will incorporate governance mechanics where the Foundation / Founding team would support the protocol for a duration of time - termed the Hand Holding Period. Over time, the improvement protocol governance would be decentralized - along the lines of the Decred project.

After the Hand Holding Period, 1% of transaction costs across all transactions would contribute towards a Development Fund - a fund for Improvement Proposals.

Proposals submitted by developers would be alloted funds by voting using DPoS based consensus mechanism. Upon positive agreement on an MIP (Metabase Improvement Proposal), funds allocated are dispersed to the developers in a linear vesting model

over the project duration. The span of voting for an MIP is 1 month. An MIP voting can be carried over 12 voting cycles. Beyond these cycles, the proposal would be rejected but archived for future reference.

Economic Governance

Economic Governance will enable stakeholders to steer the chain economics to the intended vision as the adoption of Metabase evolves over time. The following economic parameters are controllable under economic governance.

01 / **Inflation Rate**

The inflation rate determines the supply of the Metabit currency in the Metabase economy. The system starts with an initial inflation rate. As Metabase adoption evolves, there might be a need to adjust the rate (for, eg. if transaction costs are getting costly) and this can be controlled by governance mechanics.

02 / **Cost per OpCode**

In smart contracts, the estimated transaction cost is a (priority weighted) summation over the cost of each OpCode used in the transaction. Hence, individual OpCode cost is critical to determining transaction fees - and consequently economic accessibility. The cost per OpCode for each OpCode would be alterable by governance mechanics.

For example, Governance protocol could bring down the cost of an OpCode that is very frequently used.

03 / **Base Transaction Cost**

Governance mechanics can also increase or reduce base transaction cost (but bounded by a constant lower bound definition) to execute a transaction on the system (upon which the multiplier effect is set).

Axioms Governance

This concerns governance of base fundamental parameters of a blockchain. These are parameters which act like axioms for all nodes to work. For example, block generation time and number of priority pools.

All such parameters/axioms would be enlisted in an axiom sheet. The primary purpose of the axiom sheet is to ensure all nodes are operating on the same axioms. The chain would only sync up if there is a common agreed hash for the axiom sheet.

Governance mechanics can modify these axioms as defined in the Metabase Governance Protocol. All axiom changes have to mention an implementation block number in the future. At this block number, all nodes need to align to these axiomatic changes (by altering the axiom sheet and having the same axiom sheet hash).

OpCodes / Virtual Machine Governance

Metabase has an OpCode subspace reserved for the public to design custom OpCodes (see Virtual Machine section for more details). The allotment process for an empty OpCode slot is via an open auction for the public. But the acceptance of a custom OpCode on this slot would be decided by Governance mechanics (depending on factors like utility and completed development).

OpCode slots - for which no custom OpCode is built, or a custom OpCode is built but not accepted - within a defined amount of time - will be freed up and resold in another auction round.

Metabase Governance Protocol

Metabase Governance Protocol relies on on-chain "Delegated Proof-of-Stake" (DPoS) based voting mechanics. These mechanics ensure that changes are accepted by a significant fraction of stakeholders.

In each governance area, a minimum quorum of votes is required to achieve resolution of a proposal. Each proposal also gets several carry-forward cycles.

For, eg. a developer proposed a Metabase Improvement Proposal (MIP) and got 49% of the net vote in the first cycle. Hence, for the first cycle, the MIP is rejected. But the proposal can be carried forward over 11 more cycles of DPoS-based voting. The rationale of adding such a move is to allow the system to adjust to the behavioral economics of voting. (i.e., not all the time net population turns out to vote).

The following table describes various voting cycles involved in Metabase :

Areas of Governance	Type of Vote	Time Duration of Voting Cycle	Carry forward cycle	Min Vote
MIP's (Metabase Improvement Protocols)	DPoS	1 Month	12 Cycles	50%
eMIP's (Economics Metabase Improvement Protocols)	DPoS	1 Year	6 Cycles	75%

Areas of Governance	Type of Vote	Time Duration of Voting Cycle	Carry forward cycle	Min Vote
Axiom Governance	DPoS	4 Year	0 Cycles	90%
OpCode Governance	DPoS	1 Month	12 Cycle	50%

Smart Contract Security

Security has always been one of the controversial topics for smart contract platforms. Started with the DAO hack (~$60M) and followed by the recent Parity wallet freeze (~$300M), there is a consensus (No Pun intended) that blockchain platforms have room for improvement in that particular field.

Existing Solutions

Currently, smart contract security is addressed at various levels:

01 / **Code Analysers**

Using code analyzers to check for known exploitable code patterns at development stage (eg. Mythril, Oyente, Manticore, Dr. Y's Analyzer).

02 / **Formal Verification**

Using tools to create some form of a mathematical proof to ensure that smart contract will work as intended. This process is called formal verification and this technique is exercised using both at programming language and compiled bytecode level (eg. Securify).

Unfortunately compiler technology is also moving very fast so formal verification tools should catch up with it in order to create proofs.

03 / **Manual Validation**

Last one is using validation services involving real humans and code review processes (eg. Zeppelin) before launching real projects.

A multi-level approach is needed to tackle such a complex problem. There should be mechanisms implemented at the language and compiler level to ensure that a smart contract functions properly at OpCode level. In addition to that, developer tools should contain all kinds of code analyzers to spot and correct smart contract flaws in early stages. Therefore a dual solution with VM and Tools counterparts is needed.

Metabase and Smart Contract Security

With Metabase, we aim to attack the problem on both fronts.

Smart OpCodes

Metabase smart contract compilation and execution approach is much more inclined to modularization at OpCode level. As mentioned in the Virtual Machine section, Metabase will incentivize developers to create Smart OpCodes by using variable length Virtual Machine instructions. Instead of pushing developers in the direction of a multi-layer smart contract call hier-

archy, Metabase will encourage developers to use specialized OpCodes, however all OpCodes developed by developers are subject to base security rules mentioned in the Virtual Machine section.

This way, the OpCode marketplace will provide verified building blocks to create complex smart contracts:

01 / both by using less code during development and

02 / consuming less resources during execution.

This unique approach will also simplify formal verification of contracts.

Metabase IDE

The IDE will play a crucial role to spot and correct smart contract flaws. Currently, there are separate helper tools to detect and warn developers on common smart contract issues like overflows or reentrancy problems. Metabase IDE will integrate all kinds of helper tools via a simple plugin structure and enforce best security practices during development. This way security checks (known issues, weaknesses in specific compiler versions, etc.) are embedded right into the development process.

In the end, the Metabase system aims to create a secure smart contract platform by reducing the complexity at the development and compilation stages. Metabase will not only protect its users and developers funds, it also aims to create a new kind of income stream for developers by launching a verified OpCode Marketplace.

Tools

Blockchain-based systems have come a long way since Bitcoin in terms of innovation in smart contracts, consensus functions, and monetary policy. However, providing feature-rich tools for third-party developers has never been a priority. This approach has a negative impact on overall system security (detecting smart contract flaws) and limits widespread usage (lack of presence especially on mobile platforms).

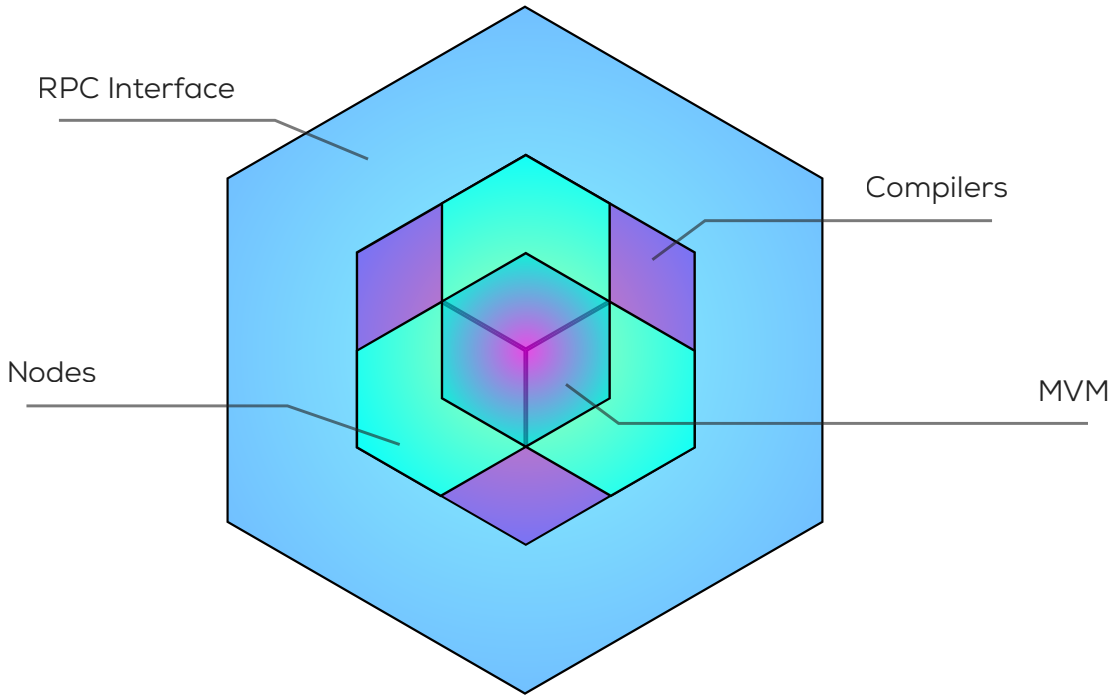
Currently, a parallel can be drawn between blockchain technology and TCP/IP (Transmission Control Protocol/Internet Protocol) in terms of adoption and disruptive potential. The reason for this is simple: blockchain technology is still considered to be in the early stages of development, and platforms are dealing with infrastructural challenges. Ethereum's Light Client has been integrated in an experimental stage, tools required for secure smart contract development are still in progress, and mobile applications interacting with blockchains are extremely low in numbers. Due to the lack of necessary frameworks, third-party companies stepped up and provided more centralized solutions for multi-platform availability.

If we follow the analogy, the development of the TCP/IP which started in 1970's and was adopted by ARPANET in 1983, has come a long way until the first versions of HTML and HTTP were introduced in the early 1990's. However, the real products that popularized the web are browsers: Mosaic in early 1993 and Netscape Navigator in late 1994. Similar to that, a modern blockchain platform should aim to create the 'browser equivalent for blockchain technology'. An easy to use financial system for its users and a rich platform for developers to implement next generation products using native, multi-platform frameworks.

At this point, our goal with Metabase is to create a Javascript API and the necessary development tools to be used to interact with Metabase even at the time of the launch. Unlike current solutions that involve third-party SDKs and centralized architecture, Metabase will provide frameworks that will be used to connect to the blockchain directly. The functionality will be implemented in the form of a library for mobile platforms or as a browser extension on browser platforms. Metabase system will embrace both browser and mobile platforms as first-class citizens and aims to encourage and support developers in search of easy-to-use blockchain applications on Metabase blockchain. This way Metabase aims to kickstart the second wave of innovation on blockchain which will be very similar to the advances in web services that followed the browser revolution.

Metabase core implementation will consist of Metabase Virtual Machine (MVM), the Metabase client (P2P, Consensus), Metabase Compiler and finally the interfaces that enable interactions with Metabase.

Base structure of a Metabase node



Metabase will provide the following applications and development frameworks for its users:

Metabase Mobile Light Client

Metabase will launch with a Light Client for mobile platforms, a type of blockchain client that does not need to download the entire blockchain to work. This way Metabase users will start using official mobile wallets right away instead of relying on third-party software. Besides wallet functionality and if the end-user so requires, Metabase Light Client will be able to store and process necessary KYC and identity information securely. This piece of information may be used for authentication in various types of interactions like trading in exchanges or sharing invoice details, therefore, embedding Metabase more and more into real life interactions.

Metabase Browser Extension

Metabase will be launched on modern browser platforms via a browser extension similar to Metamask. Metabase browser extension will enable users to interact with the next generation of products and services safely and securely right from their browser without downloading the entire blockchain.

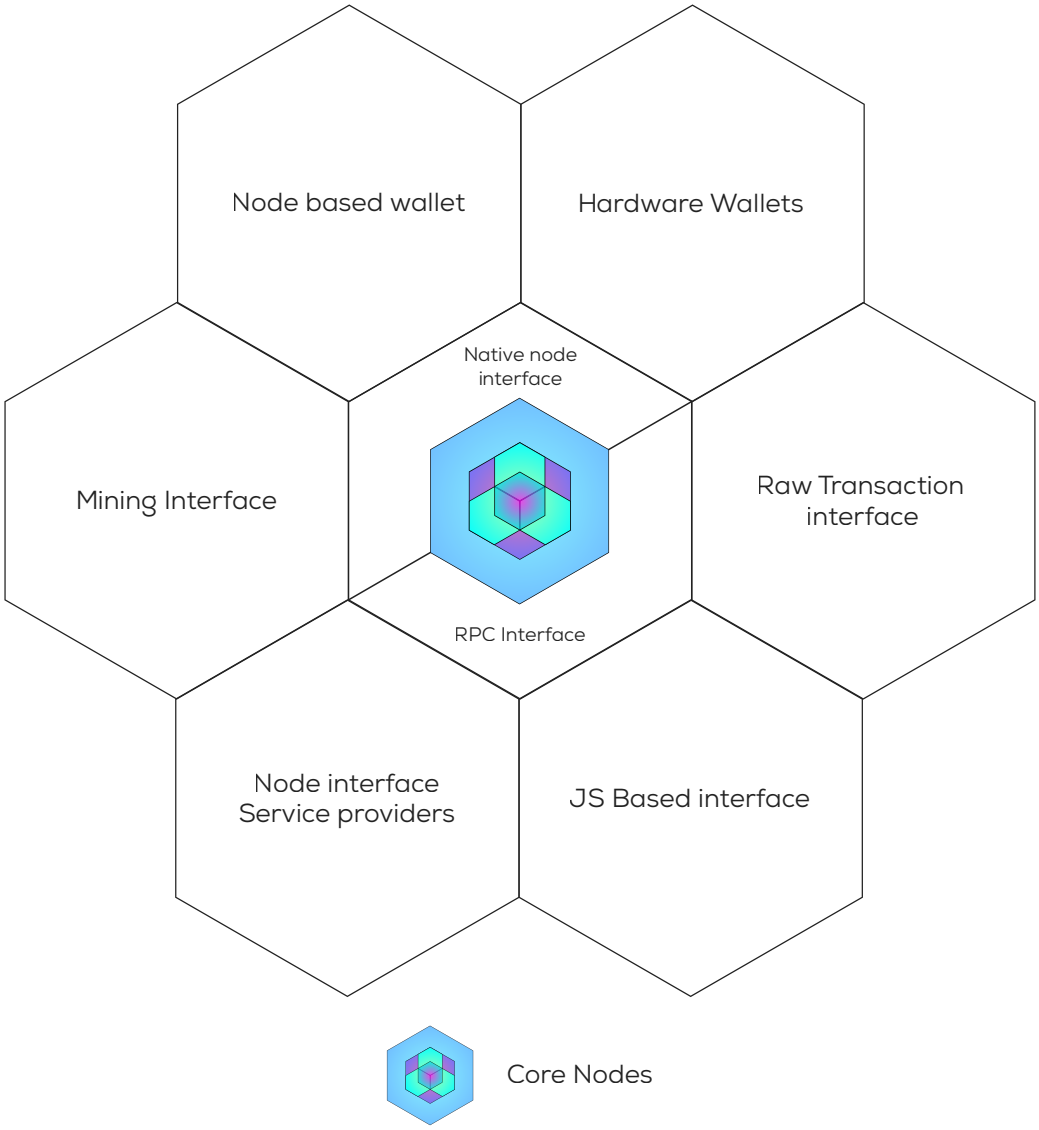
Metabase Developer SDK and Integrated Development Environment

Besides the official mobile and browser applications, Metabase will provide and maintain an SDK and an IDE for developers. Our intention for the Metabase platform is to create easy to use mobile and browser applications for end users and support developers with robust

tools and frameworks, enabling them to create the next generation of blockchain applications.

Developers will be able to create and deploy new products to Metabase blockchain directly by using the browser plugin and IDE. In terms of functionality, Metabase IDE will not be limited to compiling and deploying smart contracts but will play a critical role during development and guide developers to use best security practices at every step.

Furthermore, by using Metabase SDK, custom applications may connect to Metabase full nodes and interact with their smart contracts without the need for a proxy service. This way completely decentralized applications can be built without the need for third-party providers. Finally, Metabase developers will be equipped with rich APIs on KYC/identification, wallet operations, and blockchain query services which will provide a solid foundation for the next generation of blockchain applications.



Project Timeline

Initialization

01 / Conceptualization and Preliminary Research (June 2017- December 2017)

02 / Drafting Position Paper, White Paper And Deep Dives (December 2017-April 2018)

03 / Pre Sale & Main Sale (March 2018- May 2018)

Metabase Core Development

04 / Base Blockchain Utilities - Design & Development (May 2018- August 2018)

05 / Core Blockchain Development (August 2018- December 2018)

06 / Governance Framework Development (November 2018- January 2019)

07 / Launching Test-net (January 2019- February 2019)

MVM, Identity, Ray and Spectrum

08 / Metabase Virtual Machine (January 2019- April 2019)

09 / Augmented Code based Compiler and Dynamic High Level Programming Language (April 2019- August 2019)

10 / Ray Network and Spectrum Network (April 2019- September 2019)

11 / Identity Subsystem (September 2019- December 2019)

Metabase Ecosystem Development

12 / Tools Design and Development (June 2019- December 2019)

13 / Mining Ecosystem Development (June 2019- December 2019)

14 / Going Live (December 2019)