



IGNITE

INDEX

1. Ignite Chain Introduction
2. IGT Coin
3. Ignite Docker
4. Ignite Chain & EVM Mechanism
 - 4.1 Ethereum State Transition Function
 - 4.2 EVM Instructions
 - 4.3 EVM Implementation
5. Code Execution
6. Blockchain & Mining
7. State Transition Function
8. Masternodes
9. Consensus: Proof-of-Reward (PoR)
 - 9.1 Reward Distribution
10. Staking & Lending
11. Ignite Chain Applications
12. Governance
13. IGT Protocols
 - 13.1 Rebate Protocol
 - 13.2 GameX Protocol
 - 13.3 Privacy Protocol
14. Decentralized Exchange
15. Ignite Wallet
16. Ignite DevNet

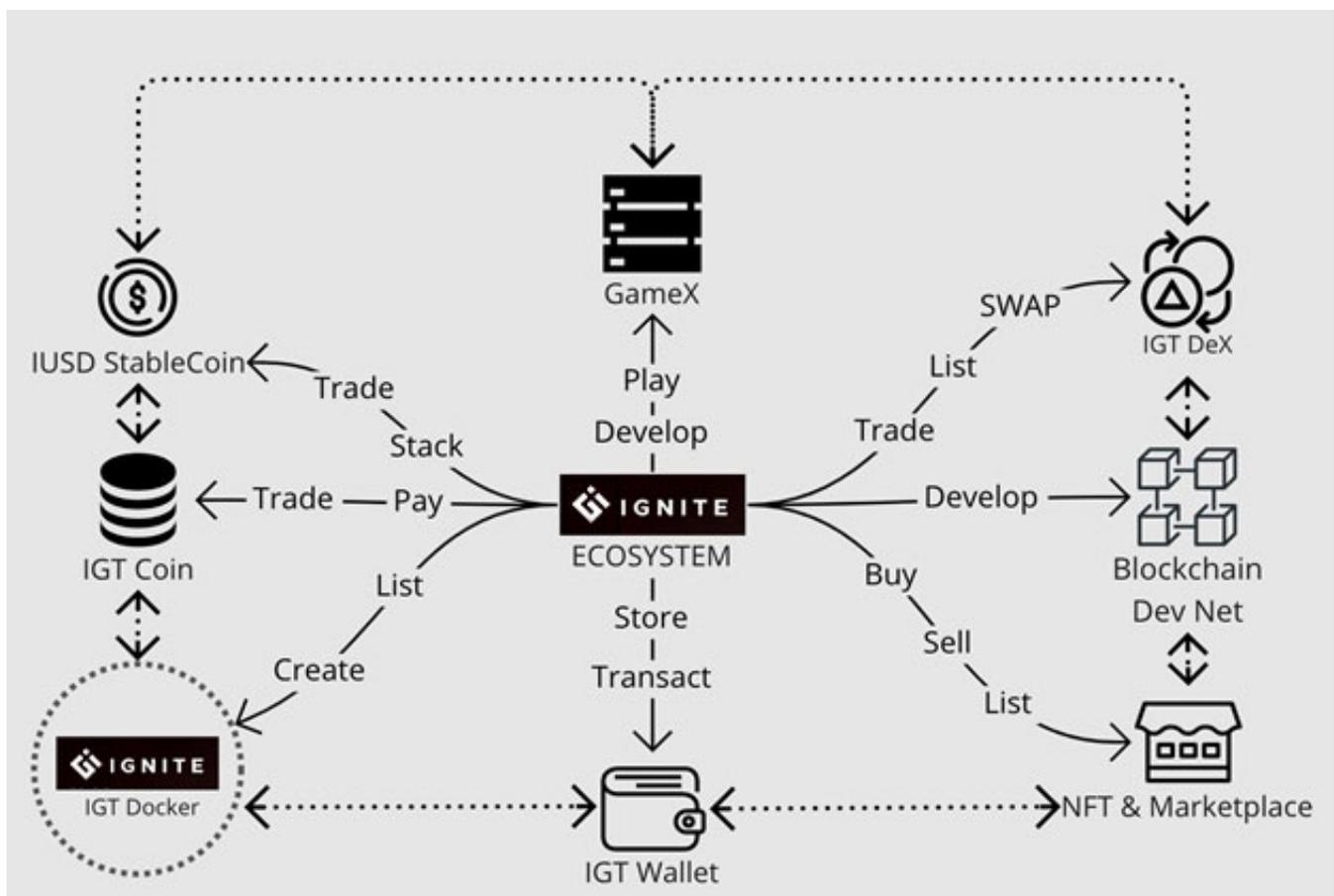
The next generation blockchain network

The world's First Decentralized and permissionless ecosystem provides several possibilities and innovations to the Blockchain stream, welcoming the developers, creators, artists, businesses, and several communities to use, build and grow with Ignite.

Ignite Aim & Principles

ignite chain is an economical, scalable, configurable, secure, auditable, and cross-chain interactive Blockchain Platform that provides several use cases:

- Building financial systems, which can provide faster confirmation and lower transaction fees for a scalable & economically viable transactional system.
- Developing Smart contracts and Non-Fungible Tokens & Assets which can be operated throughout the blockchain ecosystem.
- To build a Decentralized Ecosystem with integration to Swap Protocols and CrossChain interoperability to interconnect different blockchains and various applications.
- Providing a fluid user experience for decimating the complexity of blockchain-based operations and transactions so that users from a variety of age groups can access the Ignite platform and interact with the different facilities it has to provide.



The above statements refer to ongoing and advanced future developments by Ignite IT Solutions with the support of the open-source community and a set of developers from Ignite family to serve and provide state-of-the-art codebase and technological implementation for the improvement of the current technological scenario with the blockchain industry. According to the further timeline and development support scheduling, the elements of Ignite chain's eco-system may improvise.

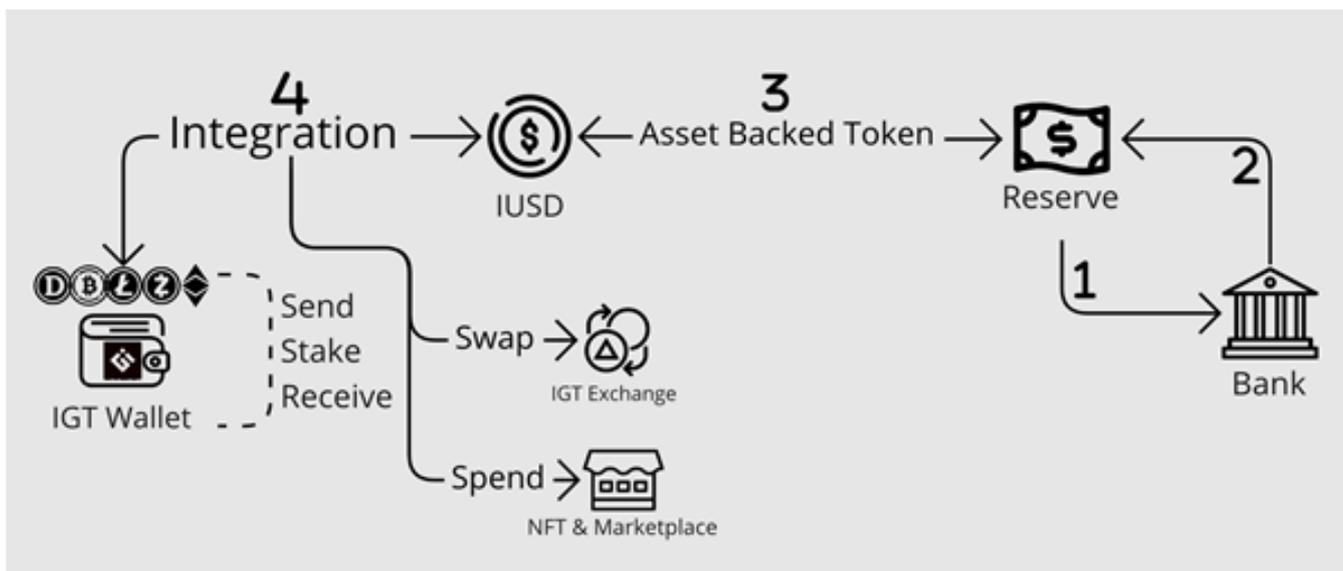
2. IGT Coin

The Native IGT Coin to Empower Ignite Chain Network & Ecosystem

Ignite aims to provide a conducive platform that can be used for transactional, creation, trading, and transferring digital assets. To govern the barter or transfer of value amongst parties for any of these applications, the IGT coin is introduced for ease of transfer and fluidity amongst the community to have secure, fast, and economically viable transactions unlike many other blockchain-based platforms, tokens, and coins which charges higher transaction fees and takes longer time compared to IGT for transactional confirmation. IGT coin is built to decimate such anomalies and provide a scalable

model, on top of which the ecosystem can be sustained by using IGT as a native token for transactions. IUSD is a stablecoin that is capped by US\$ and can be used as arbitrage for trading, proof of storage, or trading.

- IGT coin can be used to pay for transaction fees [gas] for every token, NFT, or smart contract, created on Ignite's platform.
- As a Native Token, IGT is capped supply with the cap set to 100,000,000 IGT coins, which are pre-mined or pre-minted, and the entire supply is available on main-net launch for audibility and transparency.
- For stable economics and transfer of a static value during various transactions, Ignite introduces the IUSD as their stable coin which can also be used as proof of storage for the assets they hold on the IGT blockchain.



IGT Coin

- IUSD minting and tokenization have proof of asset concept where every IUSD is backed by fiat under the institutional custody and the value of IUSD is directly proportional to the amount it holds under the custody so that IUSD can act as a bridge between fiat and crypto world and provides conducive experience while converting the value between fiat currency and cryptocurrency.

- IUSD coin will be introduced to provide the stability of funds backed by the value of US\$ for the users who want to withdraw their assets from the market volatility and secure their assets in US\$ with stable coins.

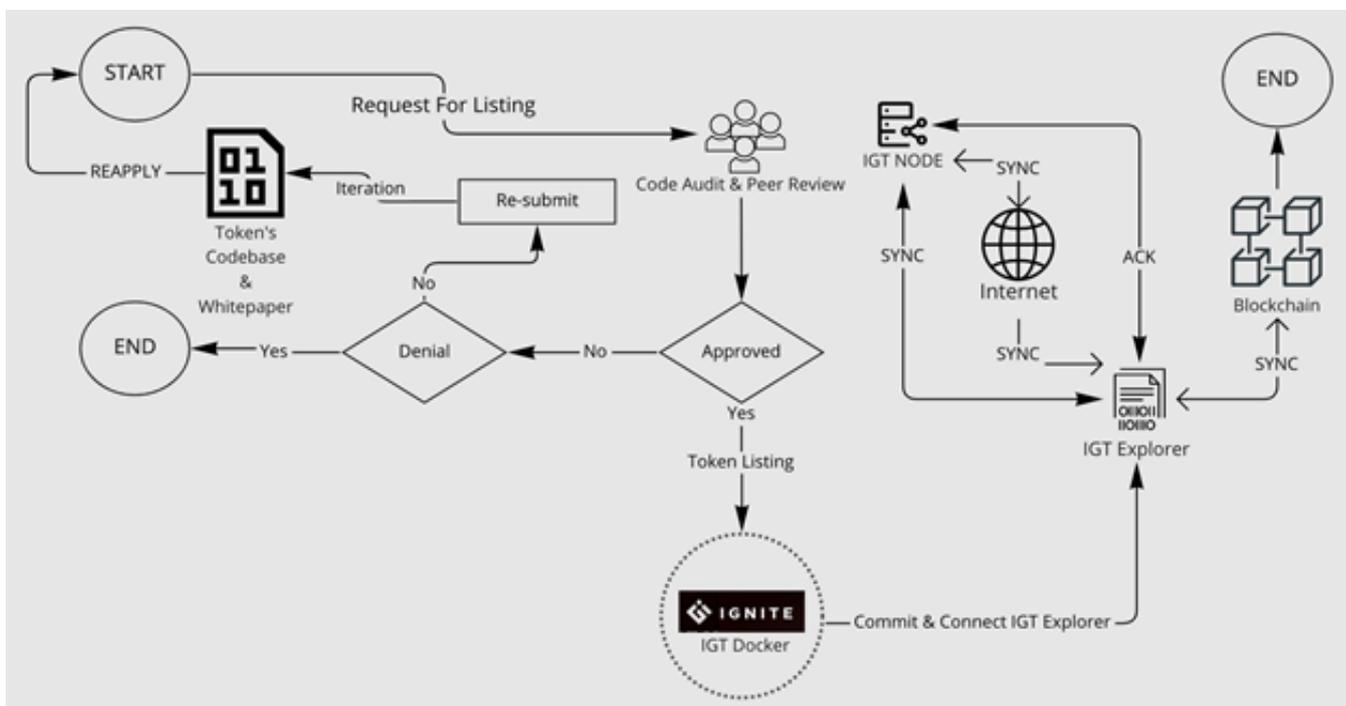
3. Ignite Docker

A Platform of Blockchain - for the Blockchain by the Blockchain

Ignite Docker signifies a platform where anyone can create their token and smart contracts using our IGT Docker ecosystem and get Financial and logistic sovereignty by being a part of the IGNITE blockchain family.

Ignite Docker is a Fully Transparent and Auditable Platform as it is developed and improvised on the bases of EVM [Ethereum Virtual Machine] compatible with every other blockchain which runs POS [Proof-Of-Stake] consensus. Docker doesn't limit to just tokenization and transactions, It can also accommodate various smart contracts to support a variety of applications.

The platform is also compatible with Creating and transacting NFTs [Non-Fungible Tokens] based on ERC-721 protocol to cherish the art industry and present future-ready facilities for the users. It is a highly adaptive platform that can be integrated with other applications using APIs and Customized protocols to give elasticity and more usability for crosschain integration of the entire Ignite Ecosystem.



Ignite Docker

Ignite's Docker is open for everyone to create their own Crypto tokens and NFT assets but to list the token in Ignite's ecosystem, the project along with the codebase has to be audited and reviewed by the peers and upon the approval from peers, the token can be listed in our exchange, wallet, explorer as well as all the peripheral elements of Ignite's ecosystem so the usability of the token can be extended.

In any case of doubt, the paperwork has to be resubmitted for peer review. To eliminate the risk of fraud, if the paperwork and codebase are found flawed, the application goes to denial and can not be listed in Ignite's ecosystem but an independent user can still utilize the token in their ecosystem by using APIs and integrating the nodes and block explorer from Ignite platform to ensure Ignite must remain open for all.

4. Ignite Chain & EVM Mechanism

The codebase and technological reference of Ignite's ecosystem is developed and improvised on top of the EVM [Ethereum Virtual Machine] infrastructure. Hence, we at Ignite welcome a whole community from the Ethereum network that can build, develop and integrate on Ignite's Platform.

Here is some information about EVM based on the references from ethereum.org

ETHEREUM VIRTUAL MACHINE (EVM)

The EVM's physical instantiation can't be described in the same way that one might point to a cloud or an ocean wave, but it does exist as one single entity maintained by thousands of connected computers running an Ethereum client.

The Ethereum protocol itself exists solely for the purpose of keeping the continuous, uninterrupted, and immutable operation of this special state machine; It's the environment in which all Ethereum accounts and smart contracts live. At any given block in the chain, Ethereum has one and only one 'canonical' state, and the EVM is what defines the rules for computing a new valid state from block to block.

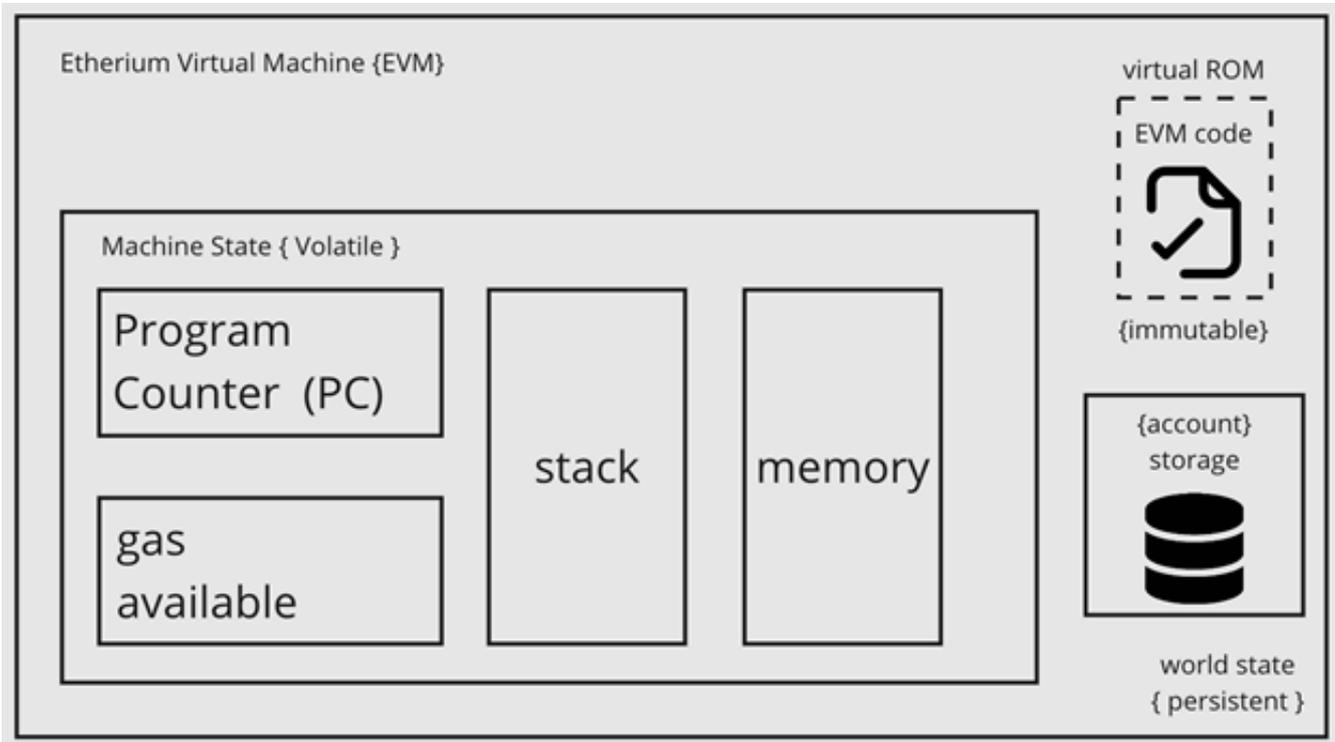
PREREQUISITES

Some basic familiarity with common terminology in computer science such as bytes, memory, and a stack is necessary to understand the EVM. It would be helpful to be comfortable with cryptography/blockchain concepts like hash function, proof of work and the Merkle tree.

FROM LEDGER TO STATE MACHINE

Analogy of 'distributed ledger' is often used to describe blockchain networks like Bitcoin, which enable a decentralized currency make use of fundamental cryptography tools. The cryptocurrency behaves like a 'normal' currency because of the rules which govern what one can and cannot do to modify the ledger. As in example, an address cannot spend more amount of Bitcoin than it has received before.

Ethereum has its own native currency ETH (Ether) that follows almost exactly the same intuitive rules but it also enables a more powerful function called smart contracts. This more complex feature require more sophisticated analogy. Instead of a distributed ledger, Ethereum is a distributed state machine. Ethereum's state is a large data structure which not only hold all accounts and balances, but a machine state, which can change from block to block according to a pre-defined set of rules, and which can execute arbitrary machine code. The specific rules of changing state from block to block are defined by the EVM.



Ignite Chain - EVM

4.1 Ethereum State Transition Function

The EVM behaves as a mathematical function would: Given an input, it produces a deterministic output. It, therefore, is quite helpful to more formally describe Ethereum as having a state transition function:

$$Y(S, T) = S'$$

■ Given an old valid state (S) and a new set of valid transactions (T), the Ethereum state transition function Y(S, T) produces a new valid output state S'

State

A state is an enormous data structure called a modified Merkle Patricia Tries, which keeps all accounts linked by hashes and reducible to a single root hash stored on the blockchain.

Transactions

Transactions are cryptographically signed instructions from accounts. There are mainly two types of transactions: 1. which result in message calls and 2. which result in contract creation on Ethereum.

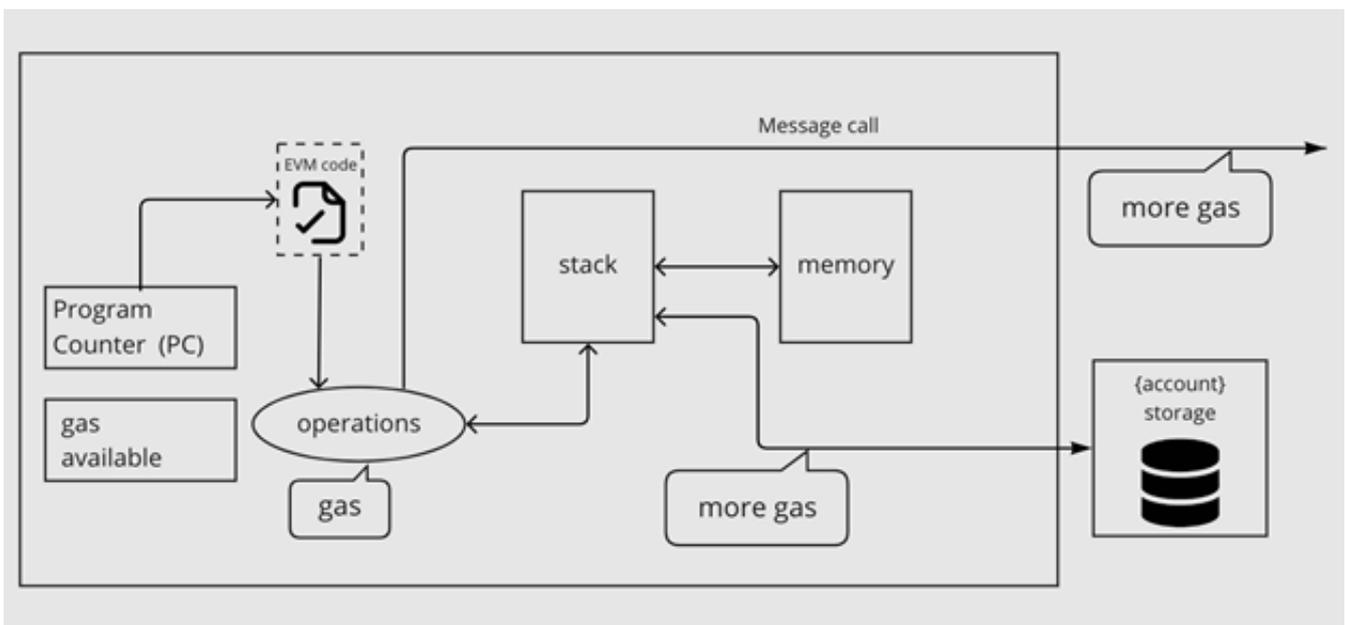
Contract creation results in the creation of a new contract account containing bytecode of compiled smart contract. Whenever another account makes a message call to that smart contract, it executes the bytecode.

4.2 EVM Instructions

The EVM executes as a stack machine with a depth of 1024 items. Each item is a 256bit word, which was chosen for ease of use with 256-bit cryptography (such as Keccak256 hashes or secp256k1 signatures). During execution, the EVM maintains a transient memory (as a word-addressed byte array), which does not persist between transactions.

Contracts, however, do contain a Merkle Patricia storage trie (as a word-addressable word array), associated with the account in question and part of the global state.

Compiled smart contract bytecode executes as a number of EVM opcodes, which perform standard stack operations like XOR , AND , ADD , SUB , etc. The EVM also implements a number of block-chain-specific stack operations, such as ADDRESS, BLANK, BLOCKHASH etc.



4.3 EVM Implementation

All implementations of the EVM must adhere to the specification described in the Ethereum Yellow-paper.

Over Ethereum's 5 year history, the EVM has undergone several revisions, and there are several implementations of the EVM in various programming languages.

All Ethereum clients include an EVM implementation. Additionally, there are multiple standalone implementations, including:

- Py-EVM- Python
- evmone - C++
- ethereumjs-vm - Javascript
- eEVM - C++
- Hyperledger Burrow - Go

Above Mentioned EVM function is purely mentioned from the linked references, based on which Ignite's Functionality is operating

5. Code Execution

The code in Ignite contracts is written in a low-level, stack-based bytecode language, referred to as "Ethereum virtual machine code" or "EVM code". The code consists of a series of bytes, where each byte represents an operation. In general, code execution is an infinite loop that consists of repeatedly carrying out the operation at the current program counter (which begins at zero) and then incrementing the program counter by one, until the end of the code is reached or an error or STOP or RETURN instruction is detected. The operations have access to three types of space in which to store data:

The stack, a last-in-first-out container to which values can be pushed and popped

Memory, an infinitely expandable byte array

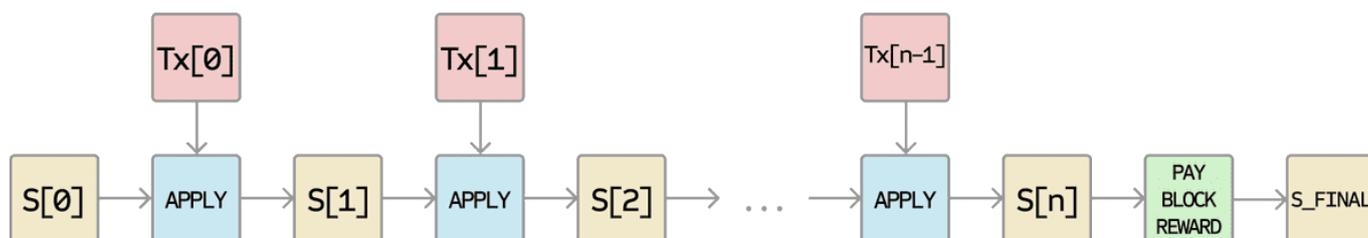
The contract's long-term storage, a key/value store. Unlike stack and memory, which reset after computation ends, storage persists for the long term.

The code can also access the value, sender and data of the incoming message, as well as block header data, and the code can also return a byte array of data as an output.

The formal execution model of EVM code is surprisingly simple. While the Ethereum virtual machine is running, its full computational state can be defined by the tuple (block_state, transaction, message, code, memory, stack, pc, gas), where block_state is the global state containing all accounts and includes balances and storage. At the start of every round of execution, the current instruction is found by taking the pc byte of code (or 0 if pc >= len(code)), and each instruction has its own definition in terms of how it affects the tuple.

For example, ADD pops two items off the stack and pushes their sum, reduces gas by 1 and increments pc by 1, and SSTORE pushes the top two items off the stack and inserts the second item into the contract's storage at the index specified by the first item. Although there are many ways to optimize Ethereum virtual machine execution via just-in-time compilation, a basic implementation of Ignite can be done in a few hundred lines of code.

6. Blockchain & Mining



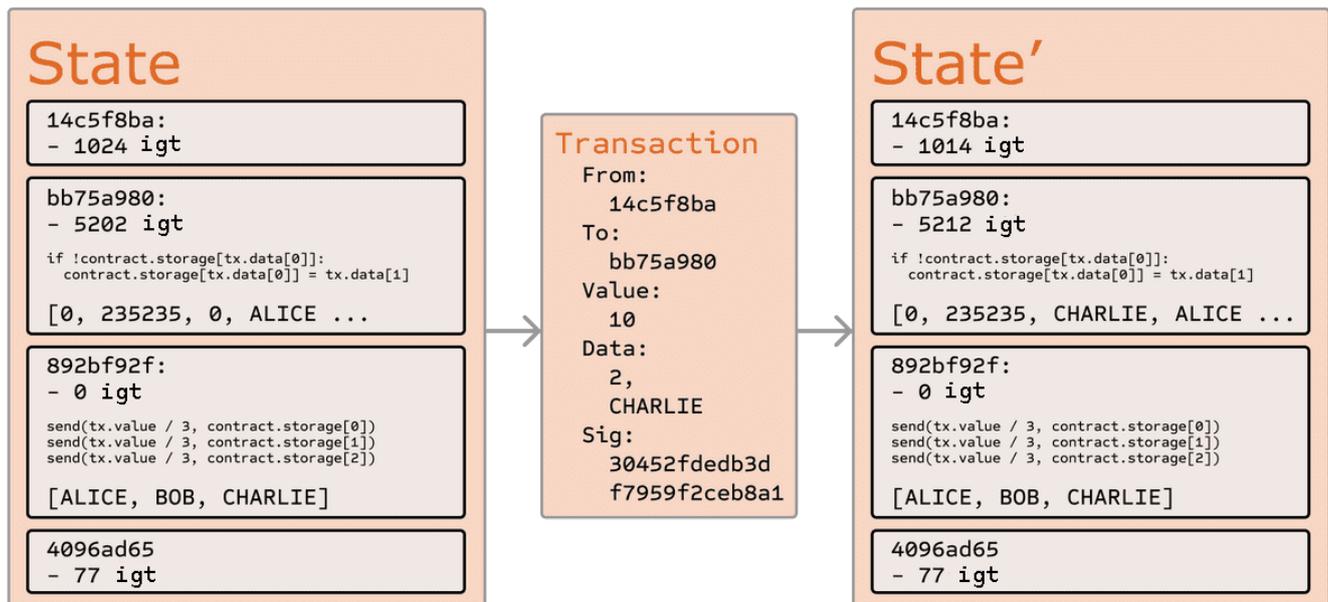
The Ignite blockchain is in many ways similar to the Bitcoin blockchain, although it does have some differences. The main difference between Ignite and Bitcoin with regard to the blockchain architecture is that, unlike Bitcoin, Ignite blocks contain a copy of both the transaction list and the most recent state. Aside from that, two other values, the block number and the difficulty, are also stored in the block. The basic block validation algorithm in Ignite is as follows:

- (1) Check if the previous block referenced exists and is valid.
- (2) Check that the timestamp of the block is greater than that of the referenced previous block and less than 15 minutes into the future
- (3) Check that the block number, difficulty, transaction root, uncle root and gas limit (various low-level Ignite-specific concepts) are valid.
- (4) Check that the proof-of-work on the block is valid.
- (5) Let $S[0]$ be the state at the end of the previous block.
- (6) Let TX be the block's transaction list, with n transactions. For all i in $0 \dots n-1$, set $S[i+1] = \text{APPLY}(S[i], \text{TX}[i])$. If any applications returns an error, or if the total gas consumed in the block up until this point exceeds the GASLIMIT, return an error.
- (7) Let S_FINAL be $S[n]$, but adding the block reward paid to the miner.
- (8) Check if the Merkle tree root of the state S_FINAL is equal to the final state root provided in the block header. If it is, the block is valid; otherwise, it is not valid.

The approach may seem highly inefficient at first glance, because it needs to store the entire state with each block, but in reality efficiency should be comparable to that of Bitcoin. The reason is that the state is stored in the tree structure, and after every block only a small part of the tree needs to be changed. Thus, in general, between two adjacent blocks the vast majority of the tree should be the same, and therefore the data can be stored once and referenced twice using pointers (ie. hashes of subtrees). A special kind of tree known as a "Patricia tree" is used to accomplish this, including a modification to the Merkle tree concept that allows for nodes to be inserted and deleted, and not just changed, efficiently. Additionally, because all of the state information is part of the last block, there is no need to store the entire blockchain history - a strategy which, if it could be applied to Bitcoin, can be calculated to provide 5-20x savings in space.

A commonly asked question is "where" contract code is executed, in terms of physical hardware. This has a simple answer: the process of executing contract code is part of the definition of the state transition function, which is part of the block validation algorithm, so if a transaction is added into block B the code execution spawned by that transaction will be executed by all nodes, now and in the future, that download and validate block B.

7. State Transition Function



Ignite Chain - State Transition

The Ignite state transition function, $APPLY(S, TX) \rightarrow S'$ can be defined as follows:

- (1) Check if the transaction is well-formed (ie. has the right number of values), the signature is valid, and the nonce matches the nonce in the sender's account. If not, return an error.
- (2) Calculate the transaction fee as $STARTGAS * GASPRICE$, and determine the sending address from the signature. Subtract the fee from the sender's account balance and increment the sender's nonce. If there is not enough balance to spend, return an error.
- (3) Initialize $GAS = STARTGAS$, and take off a certain quantity of gas per byte to pay for the bytes in the transaction.
- (4) Transfer the transaction value from the sender's account to the receiving account. If the receiving account does not yet exist, create it. If the receiving account is a contract, run the contract's code either to completion or until the execution runs out of gas.
- (5) If the value transfer failed because the sender did not have enough money, or the code execution ran out of gas, revert all state changes except the payment of the fees, and add the fees to the miner's account.
- (6) Otherwise, refund the fees for all remaining gas to the sender, and send the fees paid for gas consumed to the miner.

For example, suppose that the contract's code is:

```
if !self.storage[calldataload(0)]:  
  
self.storage[calldataload(0)] = calldataload(32)
```

Note that in reality the contract code is written in the low-level EVM code; this example is written in Serpent, one of our high-level languages, for clarity, and can be compiled down to EVM code. Suppose that the contract's storage starts off empty, and a transaction is sent with 10 ignite value, 2000 gas, 0.001 ignite gasprice, and 64 bytes of data, with bytes 0-31 representing the number 2 and bytes 32-63 representing the string CHARLIE. The process for the state transition function in this case is as follows:

- (1) Check that the transaction is valid and well formed.
- (2) Check that the transaction sender has at least $2000 * 0.001 = 2$ ignite. If it is, then subtract 2 ignite from the sender's account.
- (3) Initialize gas = 2000; assuming the transaction is 170 bytes long and the byte-fee is 5, subtract 850 so that there is 1150 gas left.
- (4) Subtract 10 more ignite from the sender's account, and add it to the contract's account.
- (5) Run the code. In this case, this is simple: it checks if the contract's storage at index 2 is used, notices that it is not, and so it sets the storage at index 2 to the value CHARLIE. Suppose this takes 187 gas, so the remaining amount of gas is $1150 - 187 = 963$
- (6) Add $963 * 0.001 = 0.963$ ignite back to the sender's account, and return the resulting state.

If there was no contract at the receiving end of the transaction, then the total transaction fee would simply be equal to the provided GASPRICE multiplied by the length of the transaction in bytes, and the data sent alongside the transaction would be irrelevant.

Note that messages work equivalently to transactions in terms of reverts: if a message execution runs out of gas, then that message's execution, and all other executions triggered by that execution, revert, but parent executions do not need to revert. This means that it is "safe" for a contract to call another contract, as if A calls B with G gas then A's execution is guaranteed to lose at most G gas. Finally, note that there is an opcode, CREATE, that creates a contract; its execution mechanics are generally similar to CALL, with the exception that the output of the execution determines the code of a newly created contract.

8. Masternodes

All about Ignite Chain node

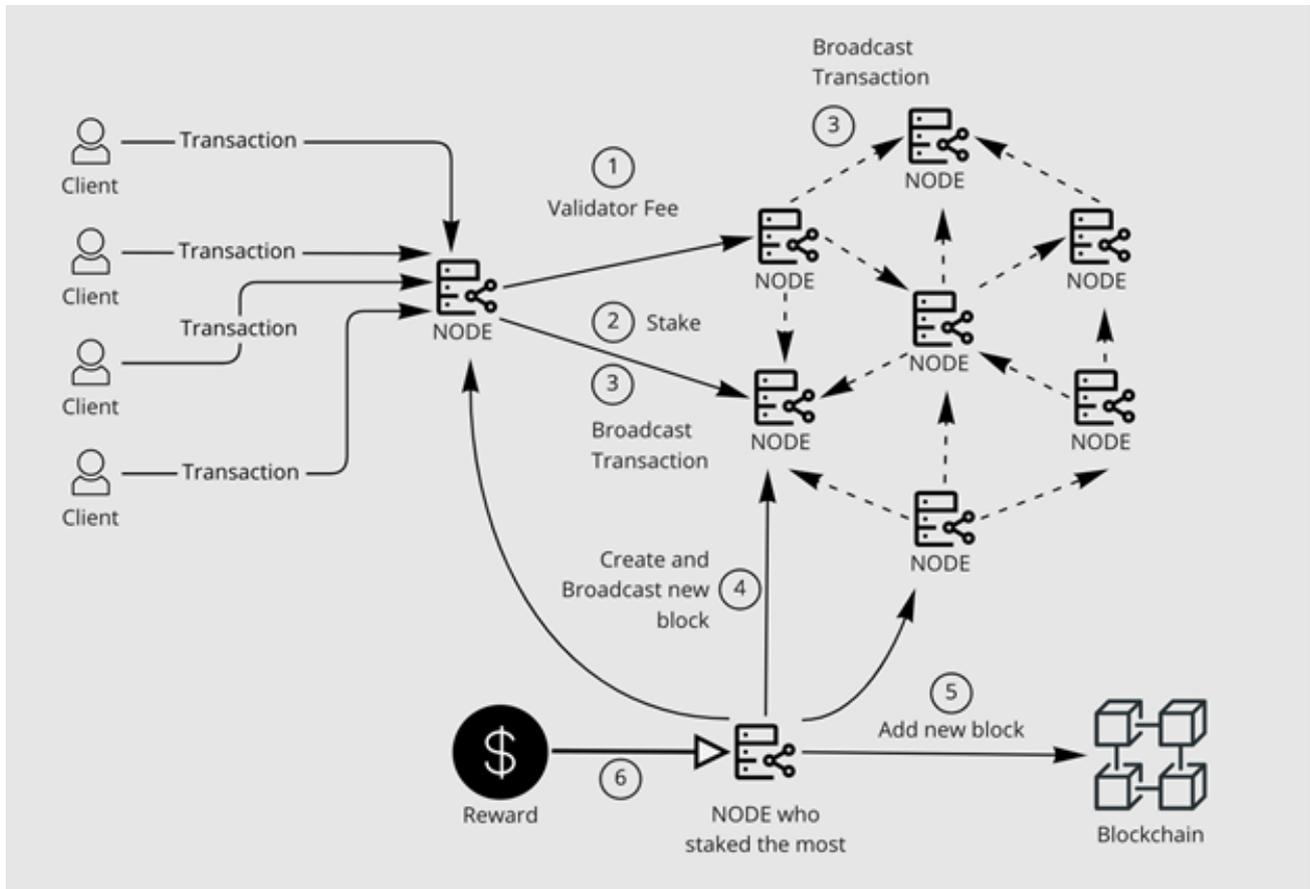
In Blockchain terminology, a computer system which connects to a blockchain network is known as a NODE. It saves a complete copy of the blockchain used by miners or validators for the verification, validation and relaying of transactions.

Nodes make sure that each block and transaction is validated before adding them to the blockchain. Running a node comes with its intangible benefits for both the Node operator and the blockchain network. It increases the security of the network and transactions conducted by users.

The algorithm chooses a random node to be the validator of the next block using an election process which is fully randomized. The node choosing process depends on several factors such as the age and size of the node.

The purpose of the node is to validate transactions on the blockchain network. When a node is selected for the next block, it will check if all the transactions in the block are valid. It will then sign the block and add it to the blockchain.

Ignite Chain Masternode



Ignite Chain - Masternode

Ignite Chain allows third party institutions or individuals to run a full node. Full node operator would be able to validate & add blocks to the ignite Chain. Running a full node on Ignite Chain need specific hardware requirements for better executions of node operations.

Setting up a node on a weaker machine might result in poor performance, significantly impacting the owner's rewards and the chain performance.

As being secured by proof-of-stake consensus, Ignite Chain will allow stakeholders & delegators to stake IGT coins to run a node once a sufficient amount of IGT coins are released to the market.

9. Consensus : Proof-of-Reward (PoR)

Consensus mechanism to power Ignite Chain network

Since every blockchain nodes require separate transaction validation, the blockchain network itself must use logical methods to make sure each node agrees on the transactions that are valid. The method the network uses to make sure that every node agrees is considered to be the blockchain consensus algorithm.

In proof-of-reward consensus, every node is rewarded for the transaction they sign right after the transaction is processed and broadcasted on the blockchain network. All the voters of the signing Masternode are incentivized in ratio of their share in total voters accordingly. This consensus system is also called Reward base validation. It is the latest consensus introduced by Ignite Chain core team members.

The core operational algorithm of Ignite Chain is called Yantra (YAN). It executes all the on-chain operations on Ignite Chain.

Proof-of-reward incentives node operators and stakers with Ignite Chain network's native IGT token. Ignite Chain is powered by proof-of-reward with using operational structure from proof-of-stake (PoS) consensus. Following this consensus, rewards are sent to eligible addresses on every state transition of the Ignite Chain network. The consensus is designed and developed to keep accountability on achieving high throughput. It is possible to execute up to 2,000 transactions per second (TPS) with and one second confirmation time on Ignite Chain.

Whenever a transaction is requested from user, the reward protocol executes the reward distributing smart contract. All the eligible parties (user addresses) are put in contract queue in order to execute rewards base validation.

9.1 Reward Distribution

How transaction fee is distributed using PoR consensus

The fee associated with the transaction is divided into 3 portion and it is sent to following sub-entities using reward distribution mechanism:

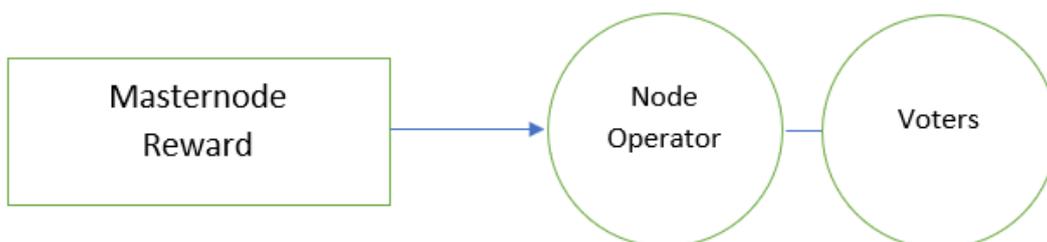
- Masternode
- Community
- Foundation

The fees are distributed in 5:4:1 by respective order.

Masternode

50% of the amount of transaction fee is sent to Masternode contract. The Masternode contract splits it in two portions:

1. 50% (25% of the transaction fee) goes to node operator
2. other half is shared among the voters to that particular node



Masternode Reward - Ignite Chain

The Masternode with the highest voters is prioritized and gets high chance to validate the transaction. From the total amount of Masternode, maximum of 251 gets to run the network and validating procedure. All node operators are required to stake 50,000 IGT tokens to be able to become Candidate Masternode.

Community

The second portion of the transaction fee, 40% goes to the community reward contract. It is shared among users and developers.

■ User

The user who send the transaction will receive up to 33% of their fee back as in form of transaction reward. The amount will depend on gwei used for in the transaction.

< 5 gwei : 25%

< 9 gwei : 29%

≥ 9 gwei : 33%

■ Developer

7% of the verified smart contract transaction fees is reserved for the developer with certain conditions. The smart contract needs to be verified by ignitescan and meet the following criteria to be able to receive developer rewards:

- 2000 Token Holders

- 10,000+ Transactions

■ Foundation

If there is any amount remaining after distributing community reward, it is sent to the Ignite Foundation. Such in case when transaction is processed with less than 9 gwei.

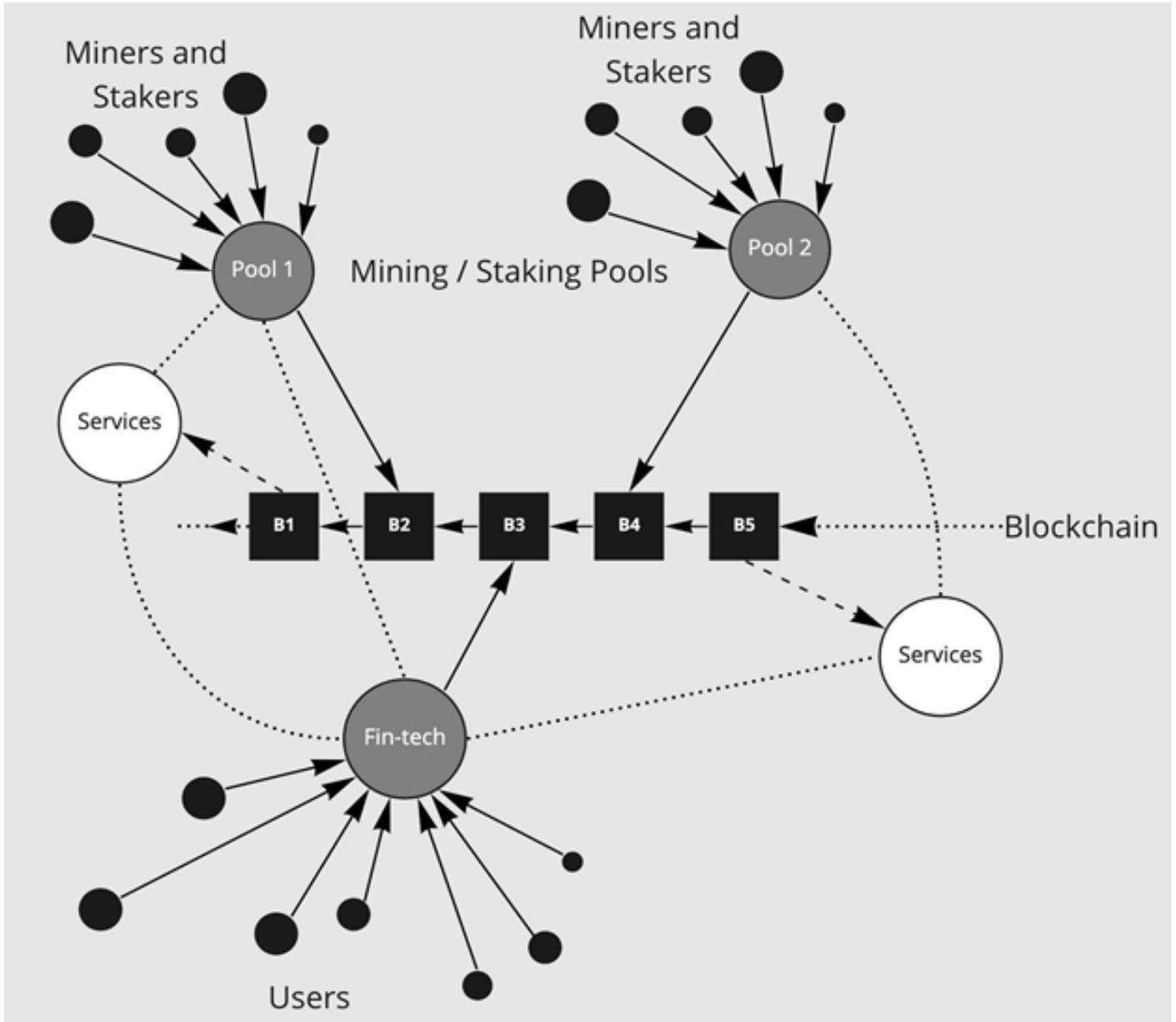
The foundation contract is entitled to use the funds in the development of Ignite ecosystem as well in such events like charity, donations and grants.

System

The third portion of gas fee, 10% will get into system and Ignite Governance. The fund is going to be used for maintenance and organizational use cases. Ignite Governance will decide where to use funds in order to grow and maintain stability of the organization.

10. Staking & Lending

Ignite provide staking & lending functionality where token holders can participate in making blockchain network more secure by staking their assets in return for being rewarded by the network with native tokens. Ignite Staking offers great rewards with safe asset custody programs.



Masternode Reward - Ignite Chain

11. Ignite Chain Applications

The use and adaptability of Ignite Chain network in real world use cases.

1. Digital Identity

Blockchain always stores unique entities meaning that the data is unalterable. Storing identities on Ignite Chain will make it easy for organizations and individuals to access data anywhere around the world. Identity verification would become less time consuming as it requires only one encrypted key to pass the verify the authority.

2. Healthcare & Pharma

Everything is turning to technology following the tech evolution, and so does the healthcare industry. Blockchain could be used to transact medical data, product verification & authenticity etc. Using the IGT Privacy protocol on Ignite Chain, it is possible to transfer healthcare and research data focusing on keeping private & encrypted.

3. Digital Payment

Ignite Chain offers ultra-low-cost, fast and borderless transactions of liquid & tradable assets. Fin-tech is already revolutionizing the whole finance sector & with the use of blockchain, it is going to make finance wide open and decentralized at the international level.

4. Real estate

Along with every traditional industry, the Real estate sector also has not been able to escape blockchain disruption. Property deals, removing intermediaries, proof of ownership etc. factors has made blockchain more useful in the real estate industry. There is also demand for virtual property ownership on popular blockchain protocols like Decentraland and other virtual world projects.

5. Logistics

Blockchain is used in the logistics industry to increase efficiency with functionality such as reduced paperwork and process automation. Ignite Chain would be providing a transparent yet traceable framework to make the workflow more structured and well managed.

6. Enterprise Solution

Ignite Chain will provide enterprise solutions to traditional as well as new generation businesses such as blockchain oracles, governance systems, data availability & off-chain transactions.

12. Governance

Ignite Chain Governance Model

In the Ignite Chain governance, there are participants or token holders that govern the blockchain more precisely. Whenever there is a need to amend or introduce new functionality into the blockchain's protocol, token holders would have to decide whether they want it on board or not. The decision is made following the majority of participants' vote. The voting event takes place over some time and only those users are allowed to participate who are tokens holders.

13. IGT Protocols

Ignite Chain custom protocols to enhance user experience in blockchain industry

Next-Generation Blockchain requires upgraded protocols and state-of-the-art technology implementation to serve the absolute amount of security, ease of transaction & distribution of tokens and digital assets like NFTs, easy adaptability, and interoperability when cross-chain implementation is executed.

13.1 Rebate Protocol

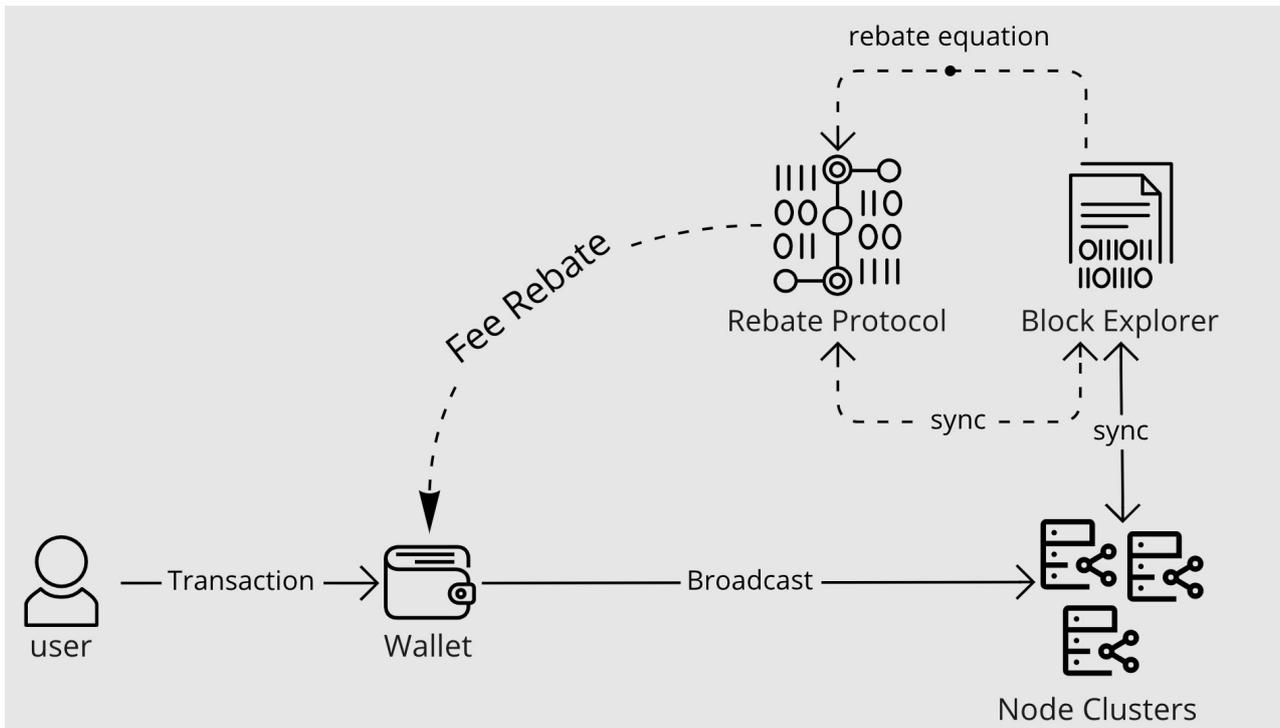
Rebate Protocol

IGNITE unfurls a novel "Rebate" protocol that can reimburse a fraction of transaction fees paid by the user after conducting N numbers of transactions. The Rebate protocol is built to make Ignite Chain's transactional system more economically viable and adaptive amongst the blockchain users and the entire industry at the other side, users are more encouraged to conduct transactions more frequently on Ignite Chain for the fee rebate. Ignite Chain's Wallet System provides flawless and instant transactions with their blockchain assets or tokens with partial reimbursement of fees [gas] paid by the users during the blockchain transaction.

Rebate Equation: The calculation based on which system reimburses the fees back to the user's wallet:

- Trnx: Transactions on Blockchain
- N: Number of transactions
- R: threshold trigger for rebate
- F: fees paid for the transaction
- Reb: reimbursement % of the transaction fee
- Total Reward = $(F * Reb)$ while $N \geq R$

[i.e. current rebate is up to 33% back on transaction fees and for rebate protocol to send back the fees to the respective wallet, it takes 1200 trnx with Ignite Chain, then $R = 1200$, so whenever N [number of trnx] will reach 1200 or larger, user can claim the rebate of 40% of the total fees F paid by the total transactions].



Transaction fees that will be sent back to the user will depend on the gwei used while processing the transaction. The prerequisite is as following:

- < 5 gwei : 25% fees back
- < 9 gwei : 29% fees back
- ≥ 9 gwei : 33% fees back

Gas Equation

Gas refers to the unit which measures the amount of computational effort required to execute specific operations like verifying the block, whispersync amongst nodes, or mining on the Ignite network. Gas fees are paid in Ignite's native token IGT.

Gas prices are denoted in gnit, which itself is a denomination of IGT- each gnit is equal to 0.000000001 IGT. For example, instead of saying that your gas costs 0.000000001 IGT, you can say your gas costs 1 gnit which is the smallest unit of IGT.

*This is just an example below of how Gas fees are calculated in the system :

Total Fees = Gas units (limit) * Gas price per unit

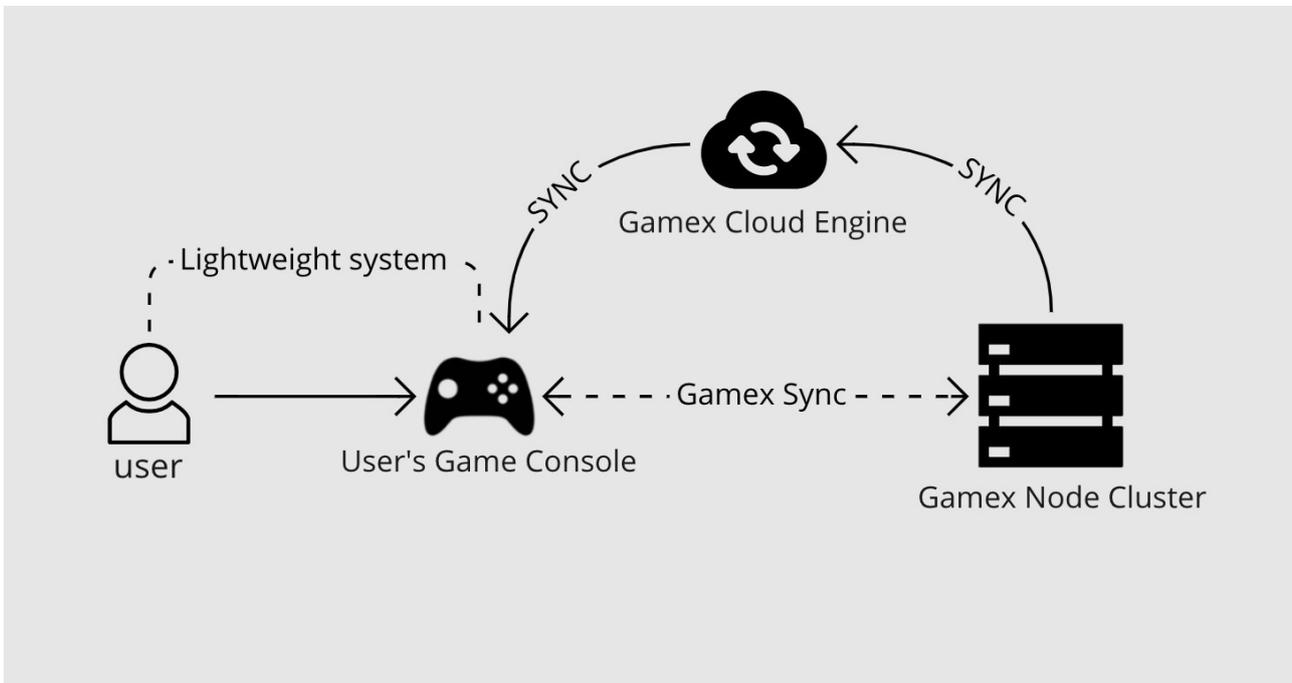
13.2 GameX Protocol

IGT GameX protocol focuses on an improved and immersive lightweight user experience by providing a cloud-hosted platform while interacting with the applications like Gaming, Graphical Editing & Rendering which require a significant amount of computing power from the native system.

IGT GameX Cloud can provide sufficient computational power to implement the heavily optimized application or computation so that users' native systems can be operated smoothly without any need to upgrade their current rig.

IGT GameX Protocol can be utilized and set up with the use of Ignite's SDK & Developer toolkit and can be easily implemented in DevNet to make a customized Dapp using our platform.

Apart from the Ignite Cloud, any user can dedicate their computing power to the GameX pool with the GameX Host application, participate in the decentralised system, and gain the rewards back to their wallet.



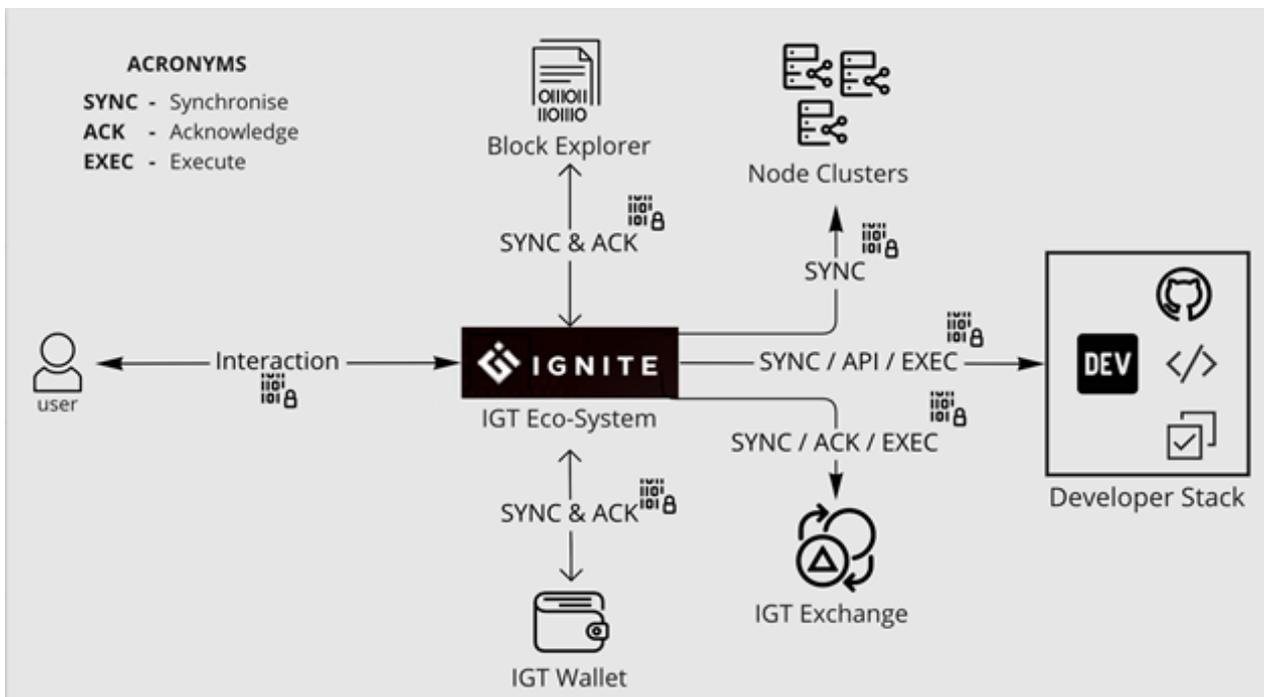
Masternode Reward - Ignite Chain

The above Graphical process, flow illustrates a simple communication between two endpoints, from the user's lightweight console heavy process can be easily executed with the integration of the Gamex endpoint with the conducive process and computations as Gamex can fulfil the computational requirement from its native endpoint where the user just need to synchronize with the game node and cloud engine for basic initial inputs and let the Gamex do the rest of the processing so that any lightweight system can execute higher processing without investing in powerful GPUs or CPUs.

13.3 Privacy Protocol

The key element of any blockchain is the Security and Privacy infrastructure, hence IGT Privacy protocol is developed to provide privacy and information obfuscating layers for retaining anonymity and decimating the risk of a data breach or malicious traceability.

- Throughout the IGT system from Block Explorer to Node implementation, we as Ignite do not keep any logs of trackers such as IP addresses, Geo Location, and other metadata and cookies which can disclose the identity of our users and their assets.
- The majority of our framework is community-driven and governed by a decentralized network, hence we are devoted to providing a private yet auditable and transparent network with no trackers or eavesdroppers.



Ignite Chain - Privacy Protocol

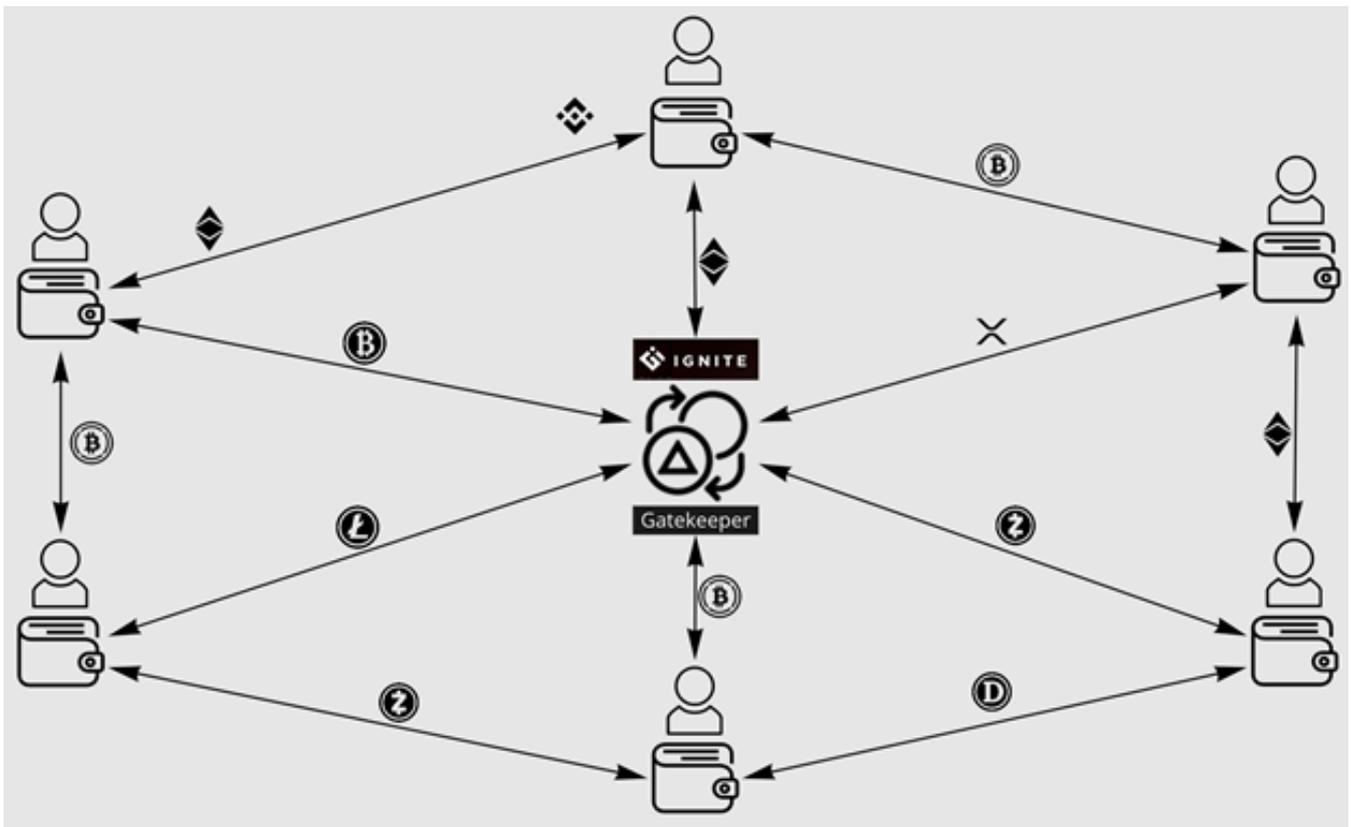
The diagram above illustrates the flow and interaction of the user and Ignite's ecosystem with our privacy measures making sure that every communication between the user and Ignite's ecosystem is kept discrete and private with an encrypted layer at every endpoint to obliterate online sniffers and trackers.

14. IGT - Decentralized Exchange

The most important factor for any digital asset is the ability to barter and trade with rapid and secure transactions amongst the network.

Decentralization of exchange has unfurled a whole new spectrum of barter with open and transparent trade and exchange of digital assets with no centralized governance, hence the access and control are distributed and driven by the support of the whole community.

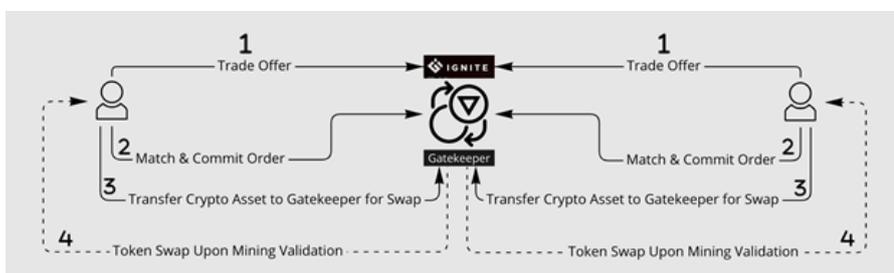
To keep the transparency and build the web of trust with our users, we are including the decentralized exchange in our ecosystem that can be integrated with other Dapps and wallet services to make a conducive and convenient system for all of our users.



Ignite Chain - Decentralized Exchange

Features & Functionalities:

- Our DeX platform is armed with the Igt Gatekeeper mechanism which ensures trade fulfilment with absolute security and verification while eliminating threats like double-spending or raised by value attack' as the gatekeeper act as an automatic concierge which releases the tokens for withdrawal only after the confirmation from both peers who committed swap of their crypto tokens in DeX.



- After confirming the trade on the exchange by users, crypto tokens from the peer's wallet get deposited to the gatekeeper for the swap and upon confirmation from both the peers, the Gatekeeper waits for the miners to verify the transaction upon successful verification, the token swap is committed and both peers receive their swapped tokens.
- The Gatekeeper validates and manages every swap by enforcing appropriate protocols so that a transaction can be fulfilled as per the pre-decided terms.

Decentralized Voting for Token Listing & Delisting :

- To list/delist any token or coin from Ignite's DeX, community voting is conducted and upon the agreement of majority voters, the token may get eligibility to list or delist.
- The community-based approach is required to keep the crypto community scam free from fraud tokens and hyped altcoins as any suspicious behavior or trade activity of listed tokens can be out-casted by the community which results in the delisting of that appropriate token.
- The same policy applies to listing the token as the community has to take a positive stand for the project that wants to be listed on IGT DeX so that only innovative and community-friendly projects can be onboarded to our DeX and all of our traders can openly trade without worrying much about the risk factors.

Lower Exchange fees / annual Subscription Based Mode

- One of the best features we can offer to the community is to minimize the fees for each trade or with an annual subscription-based model so that our users can take advantage of low cap trades as well as leverage the minor volatility into bigger accumulative hedge profits without worrying about high exchange fees.

Secured Layer with Privacy Protocol

- Ignite's Privacy Protocol is implemented throughout the ecosystem so that our DeX platform will be developed focusing on the security and privacy of our users without letting online trackers intrude on the communication or transaction Between our users and the DeX platform.

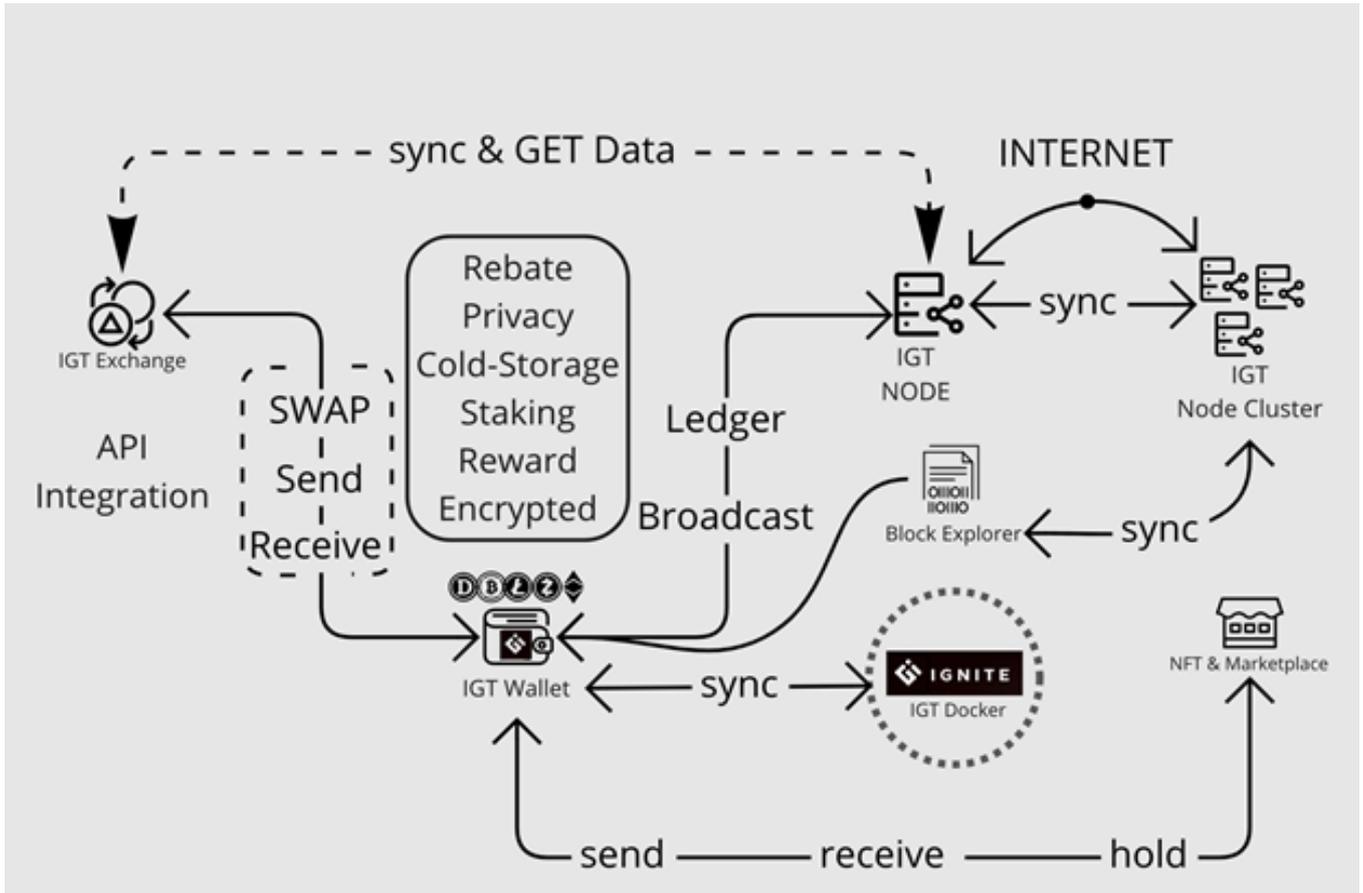
15. Ignite Wallet

The use and adaptability of Ignite Chain network in real world use cases.

We at Ignite Labs are developing Ignite's Crypto Wallet which can facilitate multiple tasks and provides a variety of functionality to our users with complete privacy and security with industry-standard RSA & ECDSA encryption algorithms.

- The user's Private Key is never exposed to the online environment and only the owner of the wallet has entire access to the key to mitigate hacking and fraudulent incidents and provide complete control of crypto assets to the user.

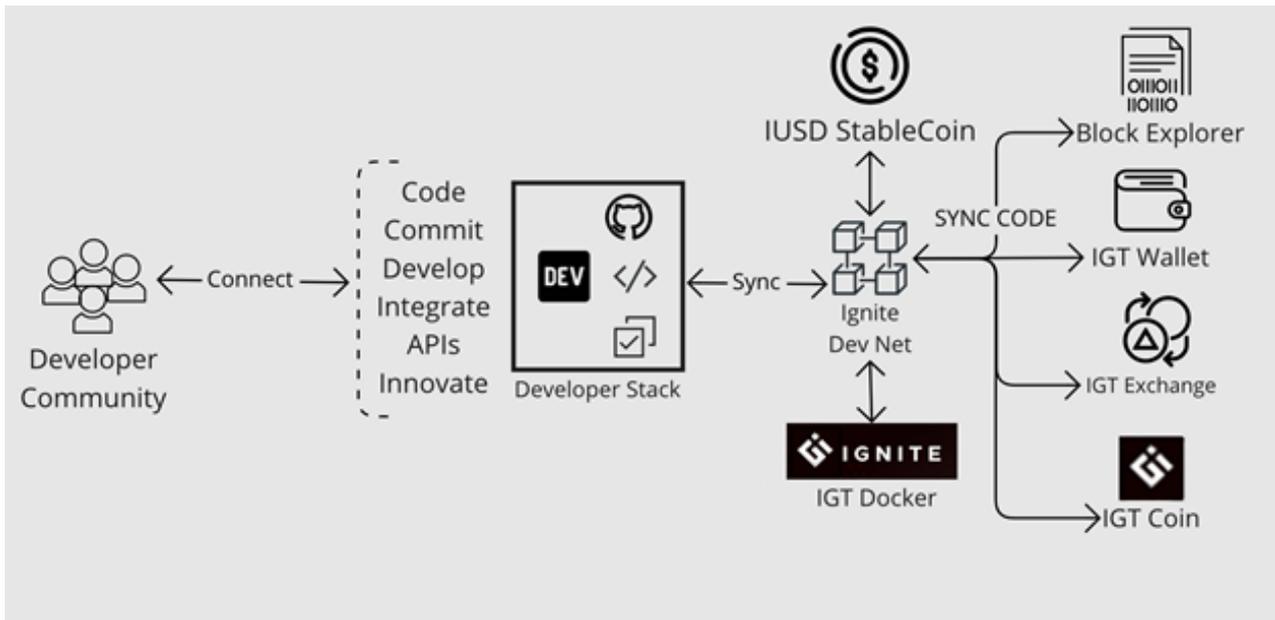
- The Wallet can be integrated with several services and third-party apps for expanding its usability as well as the wallet holder can connect the wallet to our DeX, marketplace, node and docker to send, receive stack and hold their crypto assets.



Ignite Chain - Wallet

16. Ignite DevNet

To Keep the Community thriving with innovation and upgrades in our technology, We at Ignite want to Introduce "IGT DevNet" which is a platform that can accommodate a huge library of different codebases and testing environments for coders and app developers so they can build and create novel concepts using our framework and contribute their skillset.



Ignite Chain - DevNet

Ignite DevNet is planned to be fully integrated into the ecosystem and also it can act as a medium to connect the developer community and compensate them by giving them an opportunity to be a core team member for development and coding and earn rewards for the innovation they bring in the ecosystem or existing problem they solve by submitting improved source code and commit to the DevNet.

Ignite's every element like Dex, Native Coins, Wallet, Block Explorer, and Docker, is open source and can be forked and connected to the DevNet for further improvements in source code.