RUBY

Advanced Decentralised Blockchain Platform

Based on

Proof of Authority (POA)

“This Logo is also used in various Platform”

White Paper Version 2.0

Ruby Protocol Version 1.0

Ruby Community

September 26 2023, Dubai, U.A.E.

Introduction

1. **Vision : Building a Transparent and Efficient Blockchain Ecosystem**
2. **Background: Advancing Blockchain Technology**
3. **Terminology**
    - **3.1. Address/Wallet**
    - **3.2. Application Binary Interface (ABI)**
    - **3.3. Application Programming Interface (API)**
    - **3.4. Asset**
    - **3.5 Block**
    - **3.6 Block Reward**
    - **3.7 Block Header**
    - **3.8 Cold Wallet**
    - **3.9 Decentralised Application (DApp)**
    - **3.10 gRPC (gRPC Remote Procedure Calls)**
    - **3.11 Hot Wallet**
    - **3.12 Markle Root**
    - **3.13 Public Testnet (Ruby Testnet)**
    - **3.14 RPC (Remote Procedure Call)**
    - **3.15 Scalability**
    - **3.16 RUBY (RBC)**
    - **3.17 Throughput**
    - **3.18 Timestamp**
    - **3.19 Token Standards (Ruby Standards)**
4. **Architecture**
    - **4.1 Storage Layer**
    - **4.2 Core Layer**
    - **4.3 Application Layer**
    - **4.4 Ruby Protocol & Architecture**
        - **4.4.1 Fast Blocking Time**
        - **4.4.2 Rapid Finality**
        - **4.4.3 Fee Based Reward**
        - **4.4.4 Ethereum Compatibility**
        - **4.4.5 Efficient Governance**
5. **Key Features of Ruby Blockchain**
    - **5.1 Validator based Consensus**
    - **5.2 Efficiency and Scalability**
    - **5.3 Validator Accountability**
    - **5.4 Network security**
    - **5.5 No Inflation**
6. **Advantages of Ruby Blockchain**
    - **6.1 High Performance**

- 6.2 Energy Efficiency
- 6.3 Simple Governance
- 6.4 Compatibility with Ethereum
- 6.5 Scalability
- 6.6 Developer Friendly
- 6.7 Reliability and Security

7. Validator Set in Ruby Blockchain
   - 7.1 Active Validator
   - 7.2 Validator Selection
   - 7.3 Accountability
   - 7.4 Flexibility
8. Validator Role and Election Process
   - 8.1 Active Validator (Cabinet)
   - 8.2 Standby Validator (Candidates)
   - 8.3 Inactive Validators
9. Election Timing
10. System Contracts in Ruby Blockchain
    - 10.1 Validator Set Contract
      - 10.1.1 Purpose
      - 10.1.2 Key Feature
    - 10.2 System Reward Contract
      - 10.2.1 Purpose
      - 10.2.2 Key Feature
    - 10.3 Slash Contract
      - 10.3.1 Unviability
      - 10.3.2 Double Signing
      - 10.3.3 Malicious Voting
    - 10.4 Stake Hub Contract
      - 10.4.1 Purpose
      - 10.4.2 Key Feature
    - 10.5 Benefit of System Contract
      - 10.5.1 Automation
      - 10.5.2 Transparency
      - 10.5.3 Security
      - 10.5.4 Flexibility
11. Reward Distribution in Ruby Blockchain
    - 11.1 Source of Rewards
    - 11.2 Daily Reward Distribution
    - 11.3 Undelegation and Claiming Rewards
    - 11.4. Key Benefits of Ruby Blockchain's Reward Distribution
      - 11.4.1 Fair Incentives
      - 11.4.2 Transparency
      - 11.4.3. Flexibility

12. Security and Finality of Ruby Blockchain
13. Probabilistic and Fast Finality
    13.1 Block Proposal
    13.2 Validator Voting
    13.3 Vote Collection
    13.4 Vote Aggregation
    13.5 Justification of Blocks
    13.6 Finalization
14. Key Benefits of Security and Finality in Ruby Blockchain
    14.1. Enhanced Security
    14.2. Immutable Finality
    14.3. Efficient Validation
15. Governance in the Ruby Blockchain
    15.1 Proposal and Voting Rights
    15.2 Continuous Rewards:
    15.3 Flexible Delegation
    15.4 Secure Exceution
16. Governance Process
    16.1 Temperature Check
    16.2 Final Decision Voting
17. Benefits of Ruby Blockchain Governance
    17.1 Decentralised Decision Making
    17.2 Incentivized Participation
    17.3 Flexibility and Representation
    17.4 Transparency and Security
18. Slashing and Penalties in Ruby Blockchain
    18.1 Double Sign
    18.2 Consequences
    18.3 Slash submission Process
    18.4 Malicious Fast Finality Vote
        18.4.1 Description
        18.4.2 Consequences
        18.4.3 Slash submission Process
19. Unavailability
    19.1 Description
    19.2 Consequences
        19.2.1 First Threshold
        19.2..2 Higher Threshold

1.Vision:

Building a Transparent and Efficient Blockchain Ecosystem

Ruby is an ambitious and forward-thinking initiative designed to revolutionize the blockchain ecosystem by leveraging the Proof of Authority (PoA) consensus mechanism. The project's vision centres on creating a transparent, inclusive, and highly efficient platform that serves as the foundation for decentralized applications (DApps) and services. Ruby's approach integrates cutting-edge technology with robust governance to establish a scalable and reliable infrastructure capable of empowering enterprises, developers, and individuals.

At the core of Ruby's vision is the adoption of the PoA consensus mechanism, a strategic choice that offers a host of advantages. Unlike traditional Proof of Work (PoW) or Proof of Stake (PoS) models, PoA ensures swift transaction processing, reduced energy consumption, and unparalleled security. This combination makes Ruby an ideal platform for enterprises and individuals seeking trustless and efficient solutions. By addressing the challenges of scalability and energy efficiency, Ruby aligns with global priorities of sustainable and secure digital transformation.

Ruby Protocol exemplifies innovation in blockchain design. Its high throughput and robust scalability empower developers to seamlessly create and deploy groundbreaking DApps. These applications can serve a wide array of industries, from finance and supply chain to gaming and beyond. Ruby's framework not only simplifies development but also fosters collaboration within a decentralized and community-driven environment. By enabling seamless interaction between users and developers, Ruby cultivates an ecosystem where innovation thrives and the potential of blockchain is fully realized.

A distinguishing aspect of Ruby's vision is its commitment to governance and accountability through the PoA mechanism. Unlike traditional blockchains that may prioritize either decentralization or efficiency, Ruby strikes a balance between the two. Validators in the PoA system are pre-selected and identifiable, ensuring transparency and fostering trust within the ecosystem. This governance model bridges the gap between decentralization and operational efficiency, creating a platform that is both reliable and equitable.

Beyond technological advancement, Ruby aspires to democratize access to blockchain technology. The project is deeply rooted in the belief that

decentralized infrastructure should be accessible to all, regardless of geographical or socio-economic barriers. By prioritizing inclusivity, Ruby enables communities worldwide to harness the transformative potential of blockchain innovation. Whether it's enabling financial inclusion, supporting supply chain transparency, or driving the next wave of decentralized applications, Ruby's vision extends far beyond the boundaries of technology.

Ultimately, Ruby is more than a blockchain platform—it is a catalyst for widespread adoption and empowerment. By providing a transparent, scalable, and efficient ecosystem, Ruby seeks to unlock new possibilities for individuals, businesses, and communities. The project's commitment to sustainability, inclusivity, and innovation positions it as a leader in the next era of blockchain evolution. Ruby's vision is to transform the way we interact with technology, fostering a decentralized future where opportunities are limitless and accessible to all.

2. Background: Advancing Blockchain Technology

The launch of Bitcoin in 2009 revolutionized the way society perceives financial systems, especially in the wake of the Great Recession (2007–2008). As traditional financial institutions like centralized banks and hedge funds crumbled under the weight of speculative investments in opaque financial instruments, blockchain technology emerged as a beacon of transparency and decentralization. Bitcoin introduced a groundbreaking system where transactions were securely validated through the Proof of Work (PoW) consensus mechanism, mitigating risks such as double spending and establishing trust without intermediaries.

In 2013, Ethereum expanded the blockchain landscape with its visionary white paper, introducing a platform that allowed developers to create decentralized applications (DApps) powered by smart contracts and the Turing-complete Ethereum Virtual Machine (EVM). This innovation opened up a world of possibilities, enabling blockchain to extend beyond digital currency. However, by 2017, the rapid growth of Bitcoin and Ethereum exposed critical limitations. Scalability issues such as low transaction throughput and soaring fees made these systems less practical for widespread use, highlighting the need for more efficient solutions.

Ruby Currency is a remarkable initiative that began its journey on 29th December 2020. Launched as an ERC-20 token on the Ethereum blockchain, it

set itself apart with its unique approach to distribution and community involvement. At the time of launch, Ruby Currency had a total token supply of 278.22356 million. However, what made it special was the decision by the Ruby Community to release only 40 million tokens for sale. The remaining tokens were reserved for airdrops, ensuring a fair and inclusive distribution over the next 18 years. Each year, 13 million Ruby tokens are airdropped to all holders, making the 29th of December an eagerly awaited date in the Ruby community calendar.

This deliberate approach to token allocation not only fostered a sense of belonging among users but also encouraged widespread participation. By incentivizing long-term holding and engagement, the Ruby Community created a solid foundation for growth. Within just three years of its inception, Ruby Currency had attracted a staggering 250,000 users. This impressive milestone was a testament to the community's vision and the effectiveness of its strategy.

The journey didn't stop there. On 26th September 2023, Ruby Currency took a significant leap forward by launching its own blockchain. Moving away from Ethereum, the new blockchain operates on the Proof of Authority (POA) consensus algorithm. This transition marked a pivotal moment in Ruby's evolution, providing it with the autonomy and scalability required to support its rapidly growing user base. The launch of the Ruby Blockchain was met with enthusiasm and resulted in an exponential increase in adoption. The number of holders skyrocketed, crossing the 2.5 million mark. This incredible growth highlighted the community's trust in the platform and its potential to shape the future of decentralized finance.

The Ruby Blockchain is a truly decentralized platform, putting the power in the hands of its users. The community plays a central role in decision-making, program development, and the overall growth of the ecosystem. Various initiatives have been introduced to encourage participation and expand the community further. Educational programs, promotional campaigns, and strategic partnerships have all contributed to Ruby's success. The focus has always been on inclusivity, ensuring that anyone can join and benefit from the ecosystem.

One of the standout features of Ruby Currency is its airdrop mechanism. By rewarding loyal holders annually, it creates a consistent flow of value and fosters long-term commitment. This unique approach not only sustains the existing user base but also attracts new participants eager to be part of the community. The

airdrop distribution system is a masterstroke in ensuring fairness and building trust, both of which are essential for any decentralized platform's success.

Ruby's transition to its own blockchain has also opened up new possibilities for innovation. The POA consensus algorithm ensures efficiency and security, making it well-suited for large-scale adoption. The decentralized nature of the Ruby Blockchain allows for seamless integration with various applications, paving the way for an ecosystem that extends beyond mere transactions. From decentralized finance (DeFi) solutions to tokenized assets and beyond, the potential use cases are vast and varied.

The Ruby Community's commitment to growth is evident in its proactive approach to user engagement. By running numerous programs and initiatives, the community ensures that holders remain invested and motivated. These programs are not just about increasing numbers but also about building a sense of belonging and shared purpose. The emphasis on decentralization means that every holder's voice matters, creating a truly democratic platform.

Looking ahead, Ruby Currency is poised for even greater achievements. The combination of a robust blockchain infrastructure, an engaged community, and a clear vision for the future sets the stage for sustained growth. As more users join the ecosystem, the value of Ruby Currency is expected to rise, benefiting all stakeholders. The annual airdrops will continue to serve as a powerful incentive, ensuring that the community remains vibrant and active.

In conclusion, Ruby Currency's journey from an ERC-20 token to a fully-fledged blockchain platform is a story of vision, innovation, and community spirit. By prioritizing fairness and inclusivity, it has created a thriving ecosystem that benefits all participants. With its decentralized approach and commitment to growth, Ruby Currency is not just a digital asset but a movement that embodies the true spirit of blockchain technology. As it continues to evolve, it promises to remain a beacon of innovation and a model for others to follow.

Ruby was conceived to address these pressing challenges and propel blockchain technology into its next phase of evolution. By leveraging the Proof of Authority (PoA) consensus mechanism, Ruby offers a transformative approach to blockchain design. Unlike PoW, which demands immense computational power and energy, PoA relies on a network of trusted and verified validators to secure transactions. This ensures faster transaction processing, enhanced scalability, and

significantly lower energy consumption while maintaining the integrity and reliability that blockchain systems demand.

Ruby's PoA-driven architecture provides the robust foundation needed for modern DApps, enabling high throughput and seamless scalability. It bridges the gaps left by earlier blockchain platforms, creating a system that is not only efficient but also sustainable and accessible. By addressing the inherent limitations of its predecessors, Ruby positions itself as a catalyst for mainstream blockchain adoption.

Ruby represents the next chapter in blockchain innovation, offering a scalable and secure ecosystem tailored to the needs of contemporary decentralized applications. By overcoming the bottlenecks of earlier platforms, Ruby lays the groundwork for a future where blockchain technology drives transparency, efficiency, and inclusivity on a global scale.

3. Terminology:

3.1. Address/Wallet

An address or wallet on the Ruby network is a fundamental component enabling secure interactions and transactions. Each wallet is generated through a cryptographic key pair, consisting of a private key and a public key. The private key is the cornerstone of security, acting as the exclusive means to access and manage the account. The public key, derived from the private key through advanced cryptographic algorithms, facilitates essential tasks such as encrypting session keys, verifying digital signatures, and securing data transmissions.

The public key plays a crucial role in ensuring the integrity and confidentiality of communications within the Ruby network. It encrypts data that can only be decrypted using the corresponding private key, creating a robust mechanism to safeguard sensitive information. This asymmetric encryption system underpins the trustless and decentralized nature of the Ruby ecosystem.

Wallets serve as the user's gateway to the Ruby blockchain, empowering them to perform transactions, manage assets, and interact with decentralized applications (DApps) securely. By leveraging advanced cryptographic techniques, Ruby

wallets ensure that users maintain full control over their accounts and private keys, reducing the risk of unauthorized access.

Create Wallet

Endpoint: /createwallet

Method: GET

Description: Create a new Wallet.

Request Example:

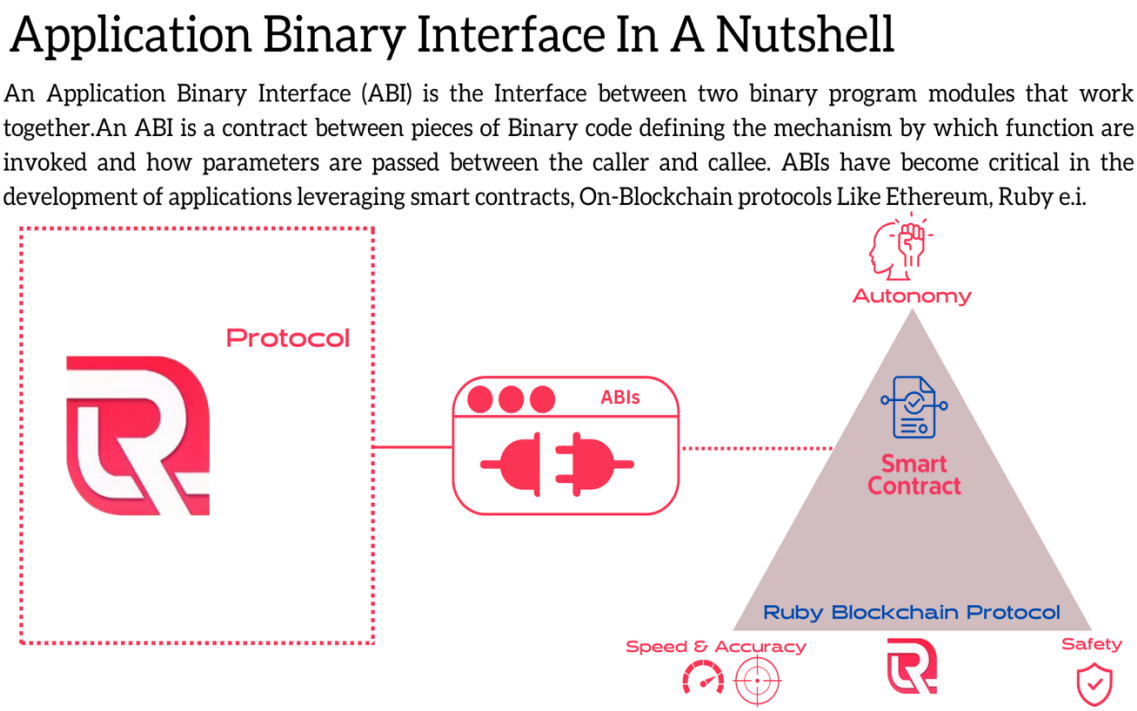
```
fetch('/createwallet', {
  method: 'GET'
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error('Error:', error));
```

Response:

```
{
  "status": true,
  "result": {
    "address": "rf76a80b19e40ae7fb4ad59bc44b031a9faab711c",
    "privateKey": "2baf481b36c8bbfa38e4f1ee22610af395d3be502c64361b4
  }
}
```

In summary, addresses and wallets are integral to Ruby's secure and efficient blockchain ecosystem. They combine cutting-edge cryptographic methods with user-centric functionality, enabling seamless and trustworthy participation within the decentralized framework.

3.2. Application Binary Interface (ABI)



An Application Binary Interface (ABI) is a critical component that facilitates communication between two binary program modules, such as a user-run program and a library or operating system facility. It acts as a bridge, ensuring seamless interaction and compatibility between these components.

Within the Ruby blockchain, ABIs play a pivotal role in the deployment and execution of smart contracts. They define the rules and structures that enable efficient communication between the blockchain's runtime environment and the smart contracts, ensuring precise execution of functions and data exchange.

By standardizing interactions, ABIs streamline the integration of software components, allowing developers to focus on building innovative decentralized applications (DApps). In the Ruby ecosystem, ABIs are indispensable for maintaining the reliability and efficiency of smart contract operations, fostering a robust and interoperable blockchain platform.

3.3. Application Programming Interface (API)

An Application Programming Interface (API) is a vital tool that simplifies the development of user-facing applications by providing predefined methods for interacting with a system. In the Ruby blockchain ecosystem, APIs enable

developers to create custom tools, user interfaces, and token issuance platforms while ensuring smooth and efficient communication with the network.

Ruby's APIs are designed to enhance accessibility and usability, empowering developers to integrate blockchain functionality into their applications seamlessly. They provide the building blocks necessary to interact with the network's core features, such as executing transactions, querying data, and managing assets.

Get Balance

Endpoint: /getBalance/:address

Method: GET

Description: Get the balance of a specific address.

Request Example:

```
fetch('/getBalance/rc85781fb5c5346e542ea87cc2518723e8e975d92', {
  method: 'GET'
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error('Error:', error));
```

Response:

```
{
  "status": true,
  "result": "11111249.31899"
}
```

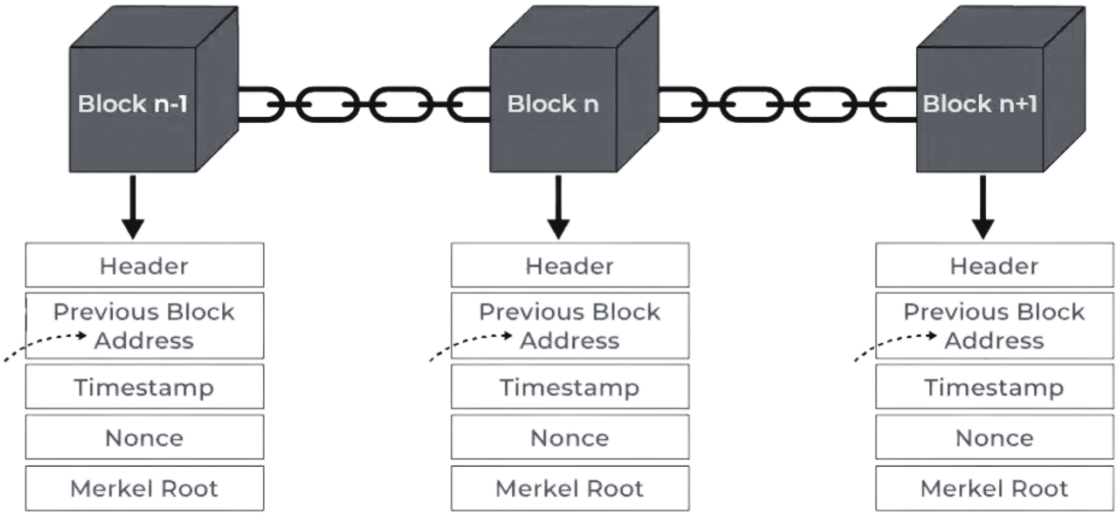
By streamlining these processes, Ruby's APIs not only accelerate development but also improve the overall experience for end-users. They play a crucial role in expanding the ecosystem's reach, fostering innovation, and driving the adoption of blockchain technology across various industries.

3.4. Asset

In the Ruby ecosystem, an asset refers to a token created and managed according to the network's standards. These tokens are designed for security, efficiency, and seamless integration into decentralized applications (DApps) and various use

cases. Assets play a vital role in powering transactions, fostering innovation, and enabling diverse applications within the Ruby blockchain.

3.5 Block



Blocks are the fundamental building blocks of the Ruby blockchain, serving as digital containers that securely store transaction records. Each block comprises several key components: a magic number, block size, block header, transaction counter, and transaction data. Together, these elements ensure the integrity, structure, and reliability of the blockchain.

Get Latest Blocks

Endpoint: /getLatestBlocks

Method: GET

Description: Retrieve the latest blocks on the blockchain.

Request Example:

```
fetch('/getLatestBlocks', {
  method: 'GET'
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error('Error:', error));
```

Response:


```
{
  "status": true,
  "result": [
    {
      "blockNumber": 123456,
      "timestamp": "2024-07-08T12:34:56Z",
      "transactions": 20,
      "miner": "0xMinerAddress"
    },
    {
      "blockNumber": 123455,
      "timestamp": "2024-07-08T12:30:00Z",
      "transactions": 15,
      "miner": "0xAnotherMinerAddress"
    }
  ]
}
```

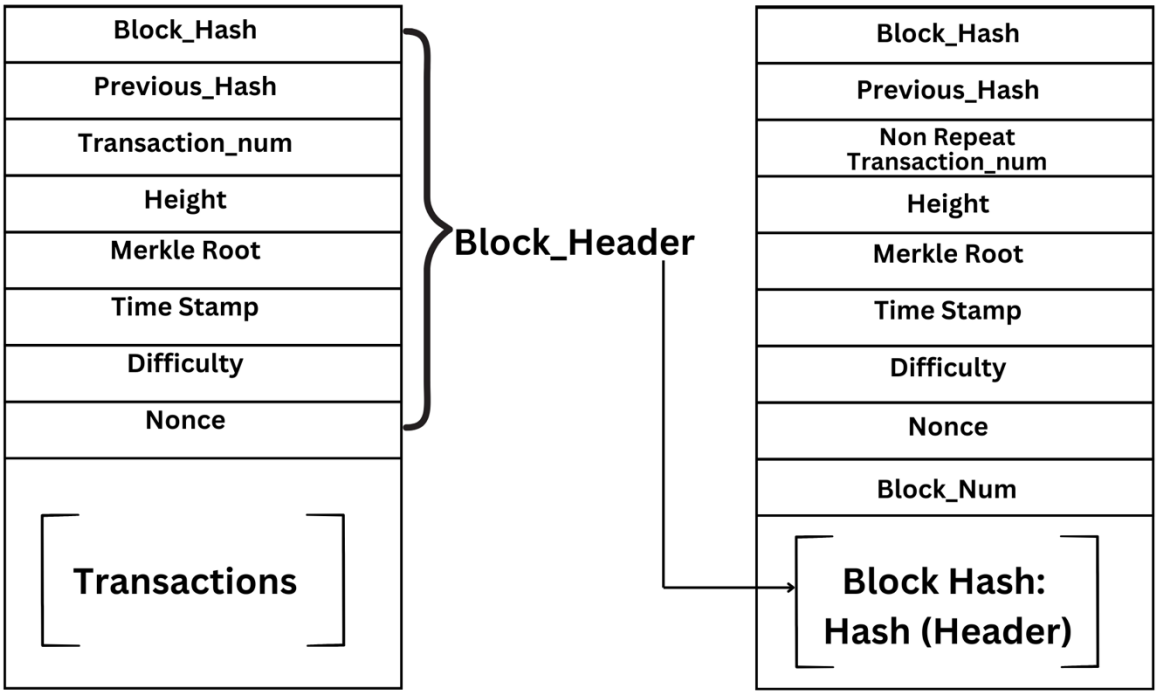

The blocks are cryptographically linked in sequential order, creating a continuous, immutable ledger. This structure not only preserves the transparency and security of the network but also ensures that every transaction is permanently recorded and verifiable. Blocks are central to maintaining Ruby's efficient and tamper-proof blockchain system.

3.6. Block Reward

Block rewards in the Ruby blockchain are a key incentive mechanism, issued to designated validators (authorized nodes) for producing new blocks and maintaining the network's integrity. These rewards are directly sent to the validator's account or wallet, reinforcing the security and efficiency of the ecosystem.

Validators can easily claim their rewards through Ruby's governance interfaces or by leveraging APIs, ensuring a streamlined and transparent process. The reward structure is integral to Ruby's Proof of Authority (PoA) consensus model, encouraging active participation, fostering accountability, and sustaining the network's stability and reliability.

3.7. Block Header



The block header is an essential part of a block in the Ruby blockchain, encapsulating vital metadata to ensure validation and integrity. Each block header includes the hash of the previous block, the Merkle root summarizing the current block's transactions, the timestamp marking when the block was created, the version number, and the validator's address responsible for producing the block.

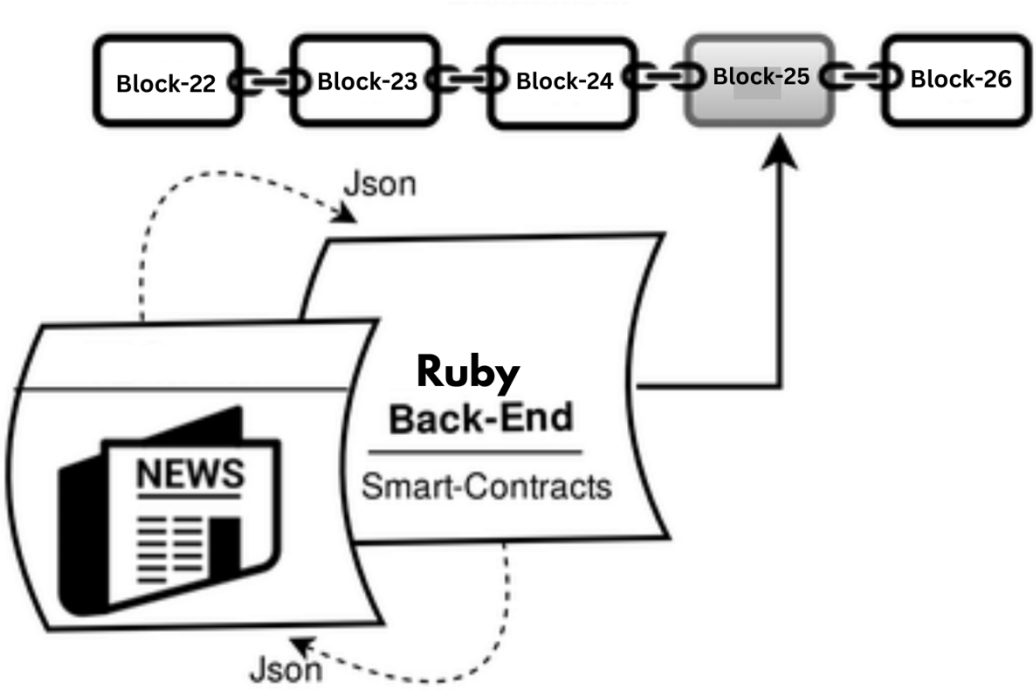
This meticulously structured information is critical for maintaining the blockchain's security and immutability. By securely linking each block to its predecessor, the block header plays a pivotal role in preserving the integrity and trustworthiness of the Ruby blockchain.

3.8. Cold Wallet

A cold wallet, or offline wallet, is a highly secure method for safeguarding private keys by keeping them entirely disconnected from any network. On the Ruby (RBC) network, cold wallets are installed on offline devices such as computers or mobile phones that never connect to the internet.

This offline storage approach significantly reduces the risk of unauthorized access, hacking, or malware attacks, ensuring the utmost protection for Ruby private keys. Cold wallets are an essential tool for users who prioritize security, offering a reliable and resilient way to protect their assets within the Ruby blockchain ecosystem.

3.9. Decentralized Application (DApp)

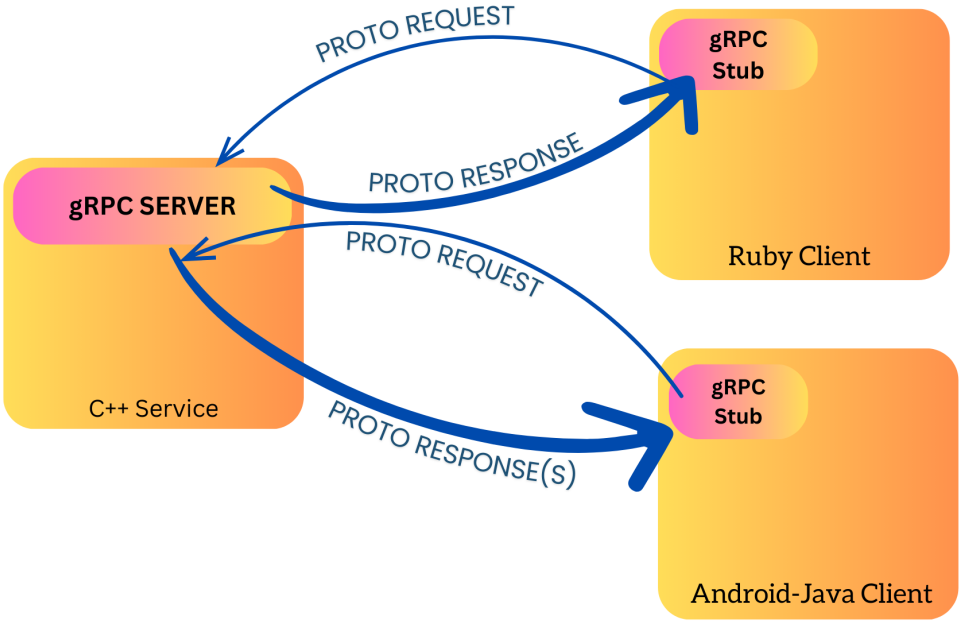


A Decentralized Application (DApp) is a software application that operates independently of any centralized authority, leveraging the Ruby blockchain to enable direct interactions between users and resources. By removing intermediaries, DApps foster a trustless environment where transactions, agreements, and communications are handled transparently and securely.

This decentralized structure ensures that data is immutable and processes are resistant to tampering, enhancing overall security and efficiency. DApps on the Ruby blockchain empower users with greater control over their interactions, making them ideal for various use cases, from financial services and gaming to supply chain management and beyond.

Through their innovative design, DApps transform traditional systems by promoting transparency, reducing operational inefficiencies, and encouraging user collaboration. Ruby's robust blockchain framework supports seamless DApp development and operation, driving widespread adoption and unlocking the full potential of decentralized technology.

3.10. gRPC (gRPC Remote Procedure Calls)



gRPC (gRPC Remote Procedure Calls) is an open-source communication framework initially developed by Google, designed to enable efficient and versatile interactions between applications. It utilizes HTTP/2 for transport and Protocol Buffers as its interface description language, offering features like authentication, bidirectional streaming, flow control, and support for both blocking and non-blocking bindings. Additionally, gRPC supports cancellation and timeouts, enhancing its adaptability for various use cases.

Within the Ruby blockchain, gRPC serves as a critical tool for seamless communication between services, such as in microservices architectures or connecting mobile and browser-based clients with backend systems. Its cross-platform compatibility ensures smooth integration across diverse environments, while its high-performance capabilities enhance the scalability and responsiveness of the Ruby ecosystem.

By leveraging gRPC, Ruby facilitates robust and efficient interactions, promoting a connected and accessible blockchain environment that supports a wide range of decentralized applications and services.

3.11. Hot Wallet

A hot wallet, or online wallet, is a digital wallet connected to the internet, enabling users to manage their private keys and conduct real-time transactions on the Ruby

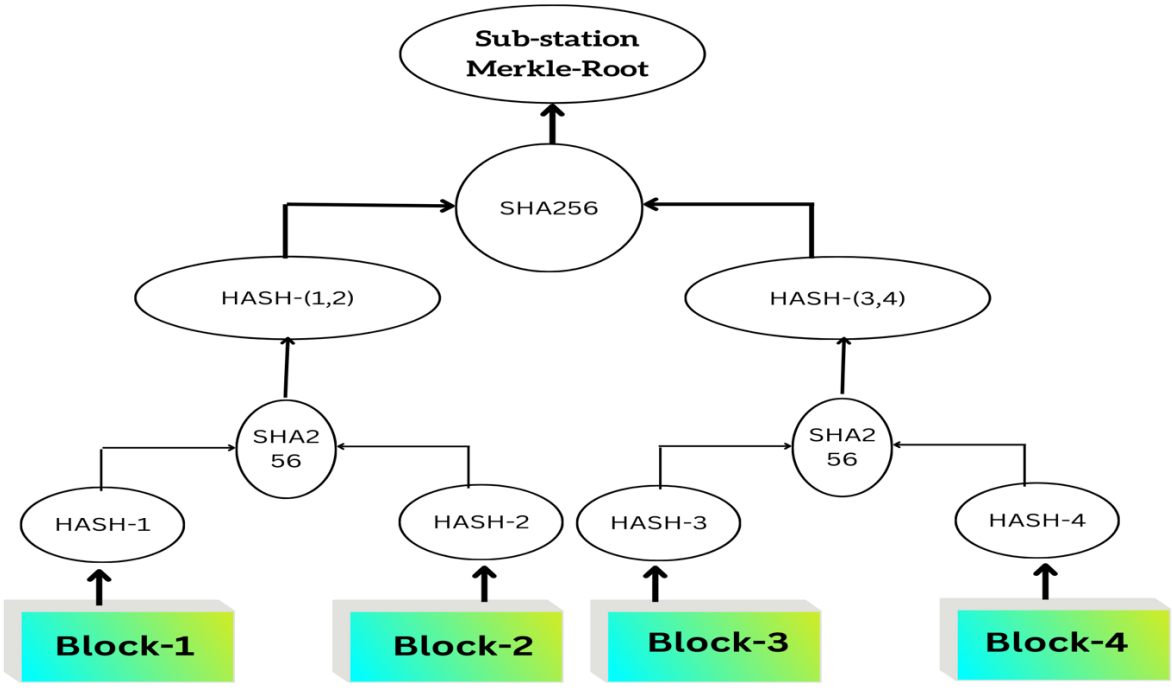
blockchain. This type of wallet offers convenience for active use, allowing seamless access to RUBY assets for trading, transfers, and other activities.

However, the online nature of hot wallets makes them more vulnerable to cybersecurity threats, such as unauthorized access or malicious attacks. To safeguard their assets, users are strongly encouraged to implement stringent security practices, including encryption, strong passwords, and multi-factor authentication.

Hot wallets are ideal for those requiring immediate access to their RUBY tokens but must be used cautiously to balance accessibility with security. By adopting best practices, users can minimize risks and enjoy the convenience of managing their assets in the dynamic and secure Ruby ecosystem.

3.12 Merkle Root

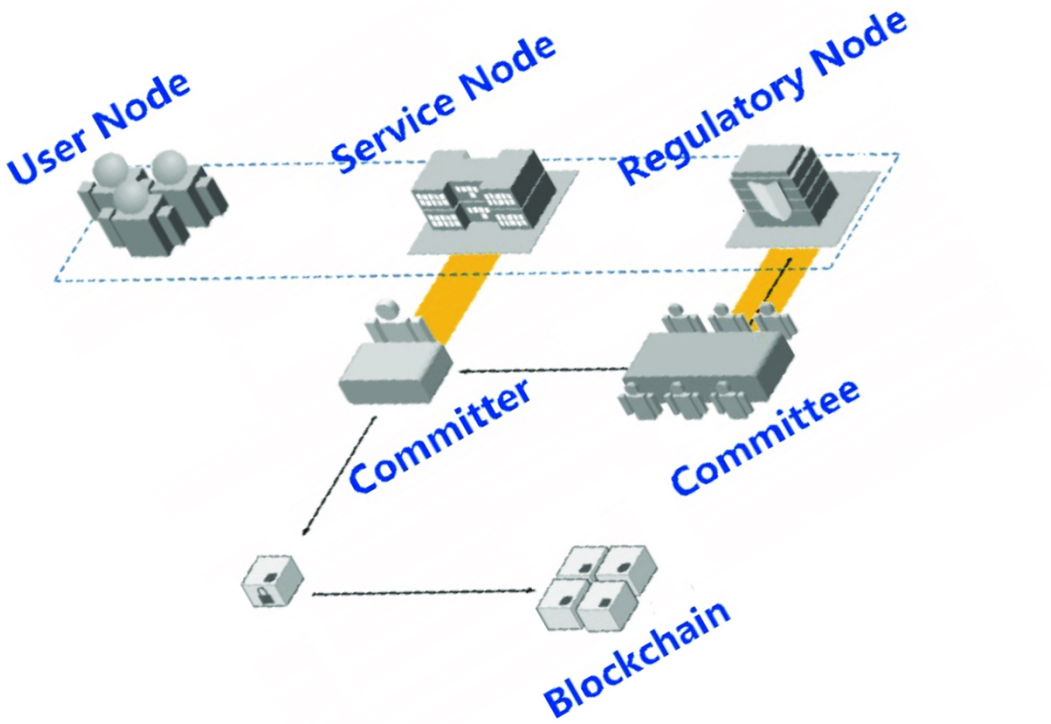
The Merkle root is a cryptographic hash that summarizes all transaction hashes within a block on the Ruby blockchain. Acting as a compact and secure representation, it ensures data integrity and enables swift verification of transactions.



In the Ruby blockchain, which utilizes the Proof of Authority (PoA) consensus mechanism, the Merkle root plays a vital role in maintaining security and reliability. Validators rely on the Merkle root to efficiently confirm the integrity

of block contents, ensuring the blockchain remains trustworthy and tamper-proof while enabling high-performance transaction processing.

3.13 Public Testnet (Ruby Testnet)



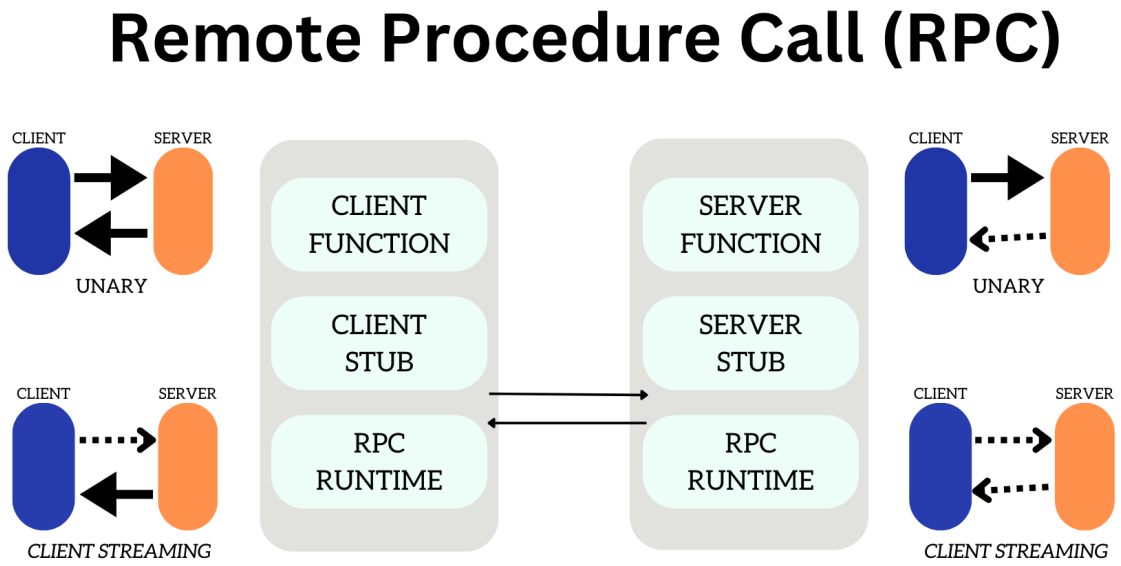
The Ruby Public Testnet is a dedicated testing environment designed to simulate the Ruby blockchain in a controlled, single-node configuration. This setup allows developers to experiment with features, functionalities, and applications in a risk-free manner.

In the Ruby Testnet, transactions and activities involve testnet tokens that hold no intrinsic value, ensuring that developers can freely test and refine their applications without any financial implications. These tokens can be obtained easily from a public faucet, enabling developers to conduct experiments, validate their code, and troubleshoot issues effectively.

By providing a secure and cost-free environment, the Ruby Public Testnet fosters innovation and accelerates development within the ecosystem. It serves as a critical resource for ensuring the robustness and functionality of decentralized applications (DApps) before they are deployed on the mainnet, contributing to the overall reliability and scalability of the Ruby blockchain.

3.14. RPC (Remote Procedure Call)

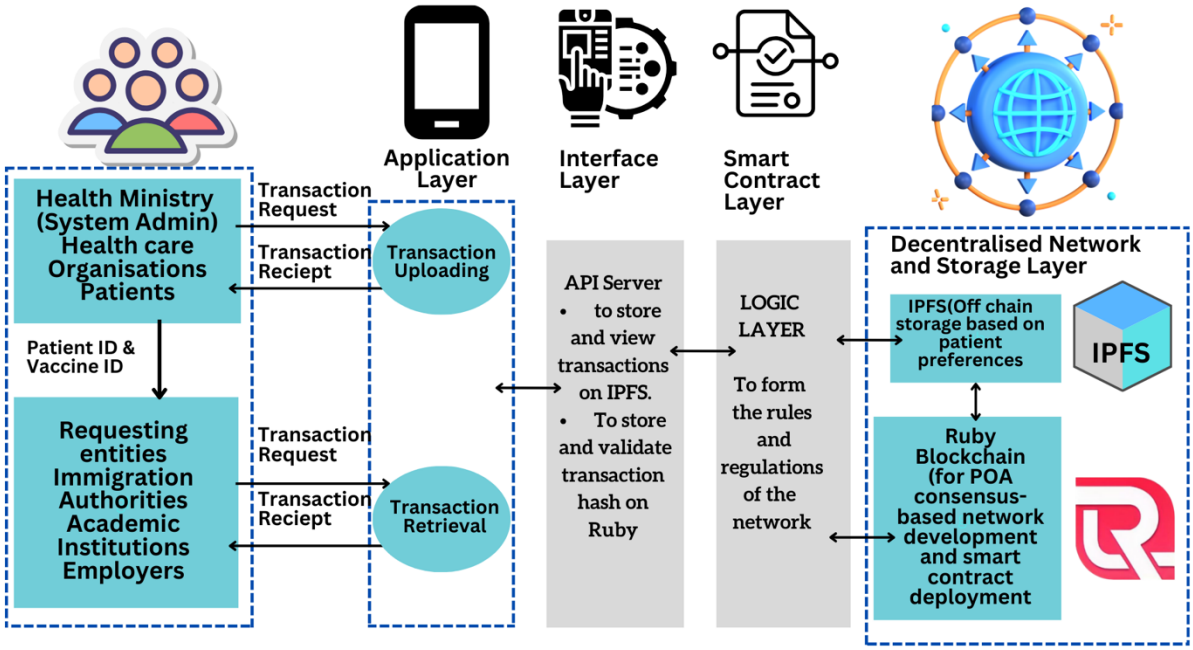
In distributed computing, a Remote Procedure Call (RPC) enables a program to execute a procedure or subroutine located in a different address space, often on another computer within the same network. This process is designed to function as seamlessly as a local procedure call, effectively abstracting the complexities of remote interactions and underlying communication protocols.



Within the Ruby blockchain, RPC plays an essential role in facilitating efficient communication between nodes, wallets, and decentralized applications (DApps). It provides a standardized interface for exchanging data and executing commands across the network, ensuring smooth interaction between various components. By leveraging RPC, developers can easily integrate and interact with the Ruby blockchain, enabling streamlined operations and enhanced user experiences. This capability makes RPC a cornerstone of Ruby's decentralized ecosystem, supporting its scalability, interoperability, and overall functionality.

3.15. Scalability

Scalability is a core strength of the Ruby Protocol, showcasing its capacity to accommodate a growing volume of transactions and operations seamlessly. Powered by the Proof of Authority (PoA) consensus mechanism, the Ruby blockchain delivers high throughput and low latency, enabling efficient and reliable performance even as network demand increases. This robust scalability



positions Ruby as an ideal platform for widespread adoption, supporting the needs of developers and users in a rapidly evolving blockchain ecosystem.

3.16. Ruby (RBC)

Ruby Currency (RBC) is the driving force behind the Ruby blockchain ecosystem, serving as the backbone for all on-chain operations. As the native cryptocurrency, RBC is indispensable for facilitating transactions, deploying smart contracts, and enabling various interactions within the network. It ensures the smooth functioning of the Ruby blockchain by covering transaction fees and supporting decentralized activities.

RUBY is more than just a transactional token; it is a vital component that sustains the efficiency and reliability of the Ruby platform. Every interaction within the ecosystem relies on RUBY, from transferring value to executing advanced blockchain-based applications. Its role is central to maintaining a dynamic and scalable decentralized environment, making RUBY an essential asset for users and developers alike.

The utility of RUBY extends beyond basic operations. By powering the network, it fosters innovation and engagement within the community, encouraging the development of new solutions and applications on the Ruby blockchain. This strategic integration of RUBY into every layer of the ecosystem underscores its importance as a foundational element, ensuring the long-term growth and success of the Ruby blockchain.

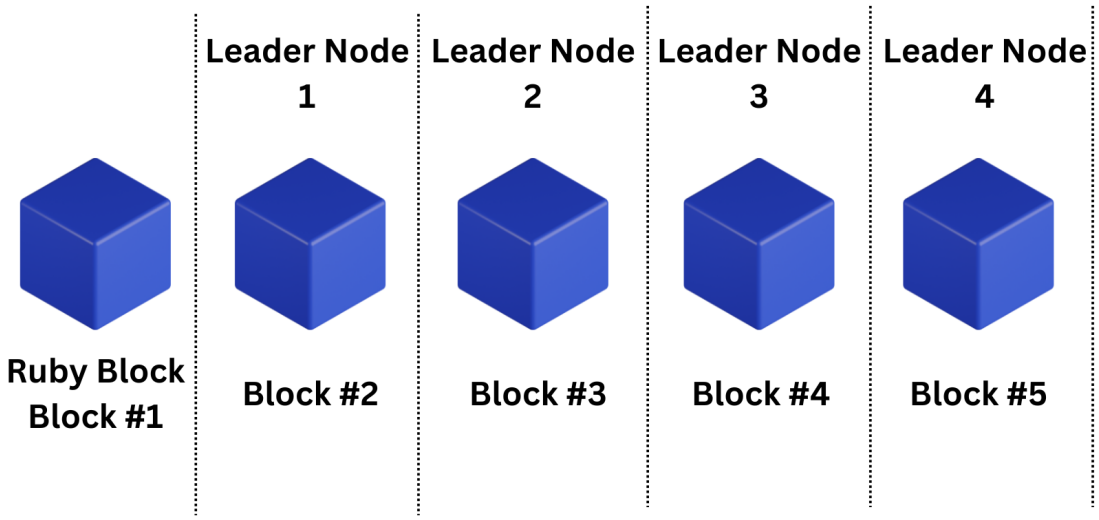
3.17. Throughput

The Ruby blockchain's throughput highlights its ability to process a significant number of transactions efficiently, measured in Transactions Per Second (TPS). This capability underlines the network's capacity to support heavy on-chain activity without sacrificing performance. The high throughput ensures scalability and positions Ruby as a robust platform ready for widespread adoption, offering users a seamless and dependable experience.

By driving the network, RUBY fosters community engagement and innovation, encouraging the creation of new solutions and applications on the Ruby blockchain. Its integration into every aspect of the ecosystem solidifies its role as a cornerstone for the platform's sustained growth and success.

3.18. Timestamp

Every block on the Ruby blockchain features a Unix timestamp, marking the approximate time of its creation. This timestamp is recorded as the number of milliseconds elapsed since 11:59:59, December 29, 2020, ICT. It serves as a crucial element for ensuring accurate sequencing and verification of transactions within the blockchain, maintaining the network's integrity and transparency.



3.19. Token Standards (Ruby Standard)

The Ruby blockchain provides standardized token configurations that define the rules and interfaces for creating and managing tokens. These standards enable developers to seamlessly issue tokens, implement functionalities, and support token-based activities within the network. By adhering to these well-defined protocols, Ruby ensures interoperability, efficiency, and ease of integration for

developers and users alike. This streamlined framework empowers innovation and simplifies token management, fostering a robust and versatile blockchain ecosystem.

4.Architecture

The Ruby blockchain uses the Proof of Authority (POA) system to make decentralized applications more efficient, secure, and scalable. Unlike traditional blockchains that need a lot of energy for mining or use anonymous participants, Ruby relies on a network where trusted and verified people, called validators, keep everything running smoothly.

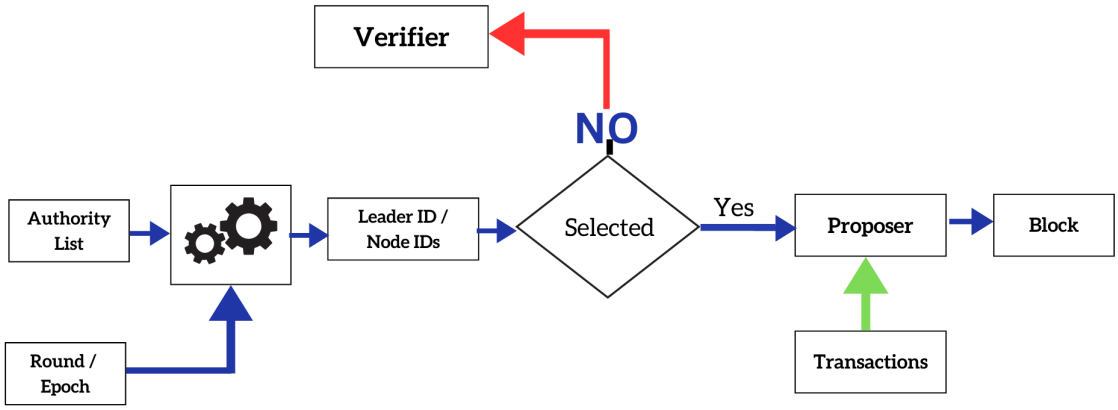
In Ruby's setup, validators are chosen based on their reputation and identity, which are publicly known. This keeps things fair and reduces the chances of bad behaviour since validators' reputations are at stake. The system is also designed to be cost-effective and fast, so it doesn't waste resources or time.

Ruby is compatible with Ethereum tools, meaning developers can easily create and run smart contracts and decentralized apps (dApps) on it. It also has special bridge technology to connect with other blockchains, making it easy to transfer tokens and data between networks.

The way Ruby is managed is a mix of decentralized and community-driven approaches. Validators and network rules are chosen and updated openly, so the system can evolve without losing trust.

Ruby stands out for being fast, secure, and affordable, making it perfect for things like tracking supply chains, managing finances, and verifying identities. Its practical design shows how POA-based systems can improve blockchain technology in real-world scenarios.

The Ruby blockchain is built on a robust three-layer architecture, carefully designed to ensure efficiency, scalability, and flexibility. Each layer plays a distinct role in maintaining the network's functionality and supporting seamless operations. This structured design enables the efficient processing of transactions, supports high scalability for growing demands, and offers the flexibility needed to accommodate diverse use cases and applications within the ecosystem. Ruby's three-layer architecture is foundational to its ability to deliver a reliable and adaptable blockchain platform.



4.1. Storage Layer

The Storage Layer is crucial for securely managing and storing blockchain data, including account states, transactions, and smart contract details. It employs advanced storage mechanisms to ensure high performance, data reliability, and efficient retrieval processes. This layer is fundamental to maintaining the integrity and seamless operation of the Ruby blockchain ecosystem.

4.2. Core Layer

The Core Layer is integral to the Ruby blockchain, managing transaction processing, smart contract execution, and the smooth operation of the consensus mechanism. At its core, Ruby employs a Proof of Authority (PoA) consensus model, relying on trusted validators to handle block production with efficiency and precision. This approach ensures high throughput, low latency, and a secure, reliable network infrastructure, making the Core Layer a vital component for delivering seamless and scalable blockchain performance.

4.3. Application Layer

The Application Layer acts as the gateway between developers and the Ruby blockchain, providing the tools and interfaces needed to deploy and execute decentralized applications (DApps) and smart contracts. This layer empowers developers to innovate while offering users seamless interaction with blockchain-based solutions. By supporting robust functionality and fostering user engagement, the Application Layer is a cornerstone of the Ruby ecosystem,

driving adoption and enabling the creation of cutting-edge decentralized technologies.

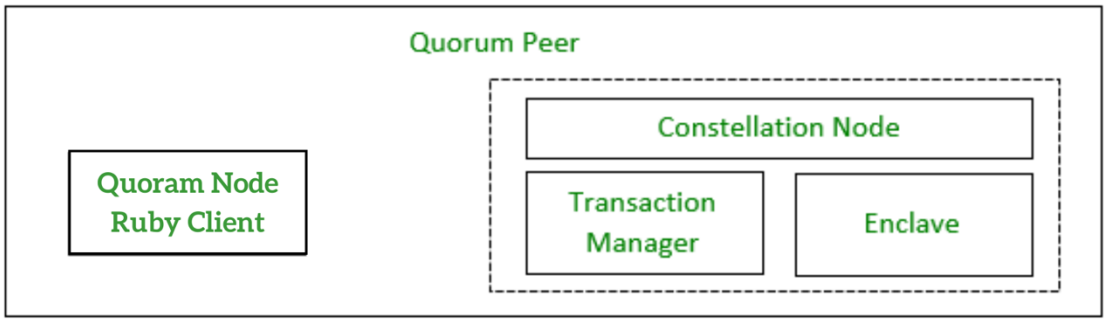
4.4. Ruby Protocol and Architecture

The Ruby protocol is built on Google Protocol Buffers (Protobuf), a powerful mechanism for efficient serialization and deserialization of structured data. By leveraging Protobuf, Ruby ensures compatibility across multiple programming languages, enabling developers to seamlessly integrate Ruby into diverse applications and development environments. This approach simplifies cross-platform development and enhances flexibility for creating innovative solutions.

Ruby's architecture is meticulously designed to support a secure, scalable, and robust blockchain ecosystem. It caters to enterprise needs, decentralized applications (DApps), and large-scale adoption by providing high performance, reliability, and seamless functionality. This design ensures the Ruby blockchain is well-suited for handling the complexities of modern decentralized technologies while meeting the demands of developers and users alike.

By combining the efficiency of Protobuf with its scalable architecture, Ruby empowers developers to build cutting-edge solutions with ease. Whether for enterprise-grade applications or individual DApps, Ruby delivers a platform optimized for security, interoperability, and innovation. This forward-thinking approach makes Ruby a leading choice for fostering widespread blockchain adoption and supporting a vibrant ecosystem of decentralized technologies.

Consensus and Validator Quorum in a Ruby Blockchain with PoA Design



Quorum Architecture

In designing a Ruby blockchain that uses Proof of Authority (PoA), the consensus protocol adheres to the following principles:

4.4.1 Fast Blocking Time: Blocks are generated every 3 seconds, ensuring a high transaction throughput.

4.4.2 Rapid Finality: Transactions achieve finality within 10 seconds or less, providing users with a seamless experience.

4.4.3 Fee-Based Rewards: The blockchain operates without inflation of its native token. Validator rewards are collected from transaction fees and distributed in the native token.

4.4.4 Ethereum Compatibility: The blockchain maintains compatibility with the Ethereum ecosystem to support the development of decentralized applications (dApps) and tools.

4.4.5 Efficient Governance: PoA enables a straightforward governance model with trusted validators to ensure security and efficiency.

Proof of Authority (PoA) for the Ruby Blockchain

Proof of Authority (PoA) is the cornerstone of the Ruby blockchain, delivering a cutting-edge consensus mechanism that prioritizes trust, speed, and efficiency. Unlike older blockchain models like Proof of Work (PoW), which rely on energy-intensive mining, PoA entrusts network security to a select group of verified validators. These validators aren't random or anonymous—they're trusted individuals or organizations chosen for their reputation and reliability. This makes PoA a game-changer for networks like Ruby, where trust and performance go hand in hand.

One of PoA's standout features is its energy efficiency. Without the need for heavy computational work, PoA slashes energy consumption, making it eco-friendly and cost-effective. But it doesn't stop there—PoA also excels in speed. Transactions are processed swiftly, and blocks are produced efficiently, enabling Ruby to handle high transaction volumes effortlessly. This makes it perfect for large-scale, enterprise-level applications that demand fast and reliable performance.

Security and accountability are baked into the PoA model. Validators' identities are publicly known, and their reputations are directly tied to their roles. This ensures they act responsibly, as any malicious behavior could damage their credibility. With this system, the Ruby blockchain achieves a rare balance of transparency and efficiency, creating a trustworthy environment for users and developers alike.

The PoA mechanism elevates Ruby to a league of its own, providing a secure, high-performance platform for decentralized applications and services.

Whether it’s tracking supply chains, managing financial transactions, or handling sensitive data, Ruby powered by PoA ensures unparalleled reliability, scalability, and sustainability. This innovative approach redefines what’s possible in modern blockchain networks, setting a new standard for trust and efficiency.

5. Key Features of Ruby Blockchain:

5.1. Validator-Based Consensus:

A small, pre-approved set of validators is responsible for producing blocks.

Validators take turns producing blocks in a predictable and deterministic manner.

5.2. Efficiency and Scalability:

Blocks are generated quickly, enabling the Ruby blockchain to handle high transaction volumes efficiently.

The deterministic nature of block production reduces the computational overhead compared to Proof of Work (PoW).

5.3. Validator Accountability:

Validators are publicly known entities with reputations at stake, such as organizations or trusted individuals.

This model ensures a high level of accountability, as malicious behaviour could result in loss of trust and removal from the validator set.

5.4. Network Security:

PoA provides resilience to 51% attacks due to its controlled validator set.

The network maintains security through Byzantine fault tolerance, enabling it to function reliably even if some validators act maliciously.

5.5. No Inflation:

Transaction fees, instead of block rewards, serve as incentives for validators, ensuring their economic interests are directly tied to the network's activity and encouraging efficient and sustainable operation.

6. Advantages of Ruby Blockchain

6.1. *High Performance*

Ruby's deterministic block production and limited validator set enable rapid transaction processing and significantly reduce network latency, ensuring a seamless user experience and scalable performance.

6.2. *Energy Efficiency*

By eliminating the resource-intensive mining processes required in Proof of Work (PoW) systems, the Ruby blockchain minimizes energy consumption, making it an environmentally sustainable blockchain solution.

6.3. *Simple Governance*

Ruby features streamlined governance mechanisms, allowing for straightforward management of validators. This simplicity ensures the network remains secure, efficient, and adaptable to future needs.

6.4. *Compatibility with Ethereum*

Ruby is fully compatible with the Ethereum ecosystem, enabling developers to leverage existing Ethereum tools, libraries, and smart contracts. This compatibility accelerates development and fosters innovation.

6.5. *Scalability*

Ruby's architecture is optimized for scalability, ensuring the network can handle increasing transaction volumes and support extensive decentralized applications (DApps) without compromising performance.

6.6. *Developer-Friendly*

With robust documentation, APIs, and standardized token protocols, Ruby offers a developer-friendly environment that simplifies the creation and deployment of DApps and smart contracts.

6.7. *Reliability and Security*

Ruby's Proof of Authority (PoA) consensus mechanism ensures a reliable and secure network by relying on trusted validators, offering consistent block production and protection against malicious activities.

The Ruby blockchain, leveraging the Proof of Authority consensus mechanism, is tailored for scenarios demanding high efficiency, minimal costs, and rapid transaction finality. It is ideally suited for private and consortium blockchains, enterprise-grade applications, and environments where consistent performance and a network of trusted participants are essential. This design ensures that Ruby meets the specific needs of use cases

requiring reliability, scalability, and secure operations, making it a robust choice for modern blockchain solutions.

7. Validator Set in Ruby Blockchain

The validator set consists of nodes responsible for validating transactions and producing blocks on the Ruby blockchain. Unlike staking-based systems, where validator selection depends on token holdings, the PoA model relies on a predetermined set of trusted validators. These validators are selected based on their reputation, authority, and commitment to the network's stability.

Key Features of the Validator Set:

7.1. Active Validators:

A limited number of trusted nodes serve as active validators.

Active validators take turns proposing and validating blocks in a predictable sequence, ensuring efficiency and low latency.

7.2. Validator Selection:

Initially, a trusted group of validators is set during the genesis stage of the blockchain.

Over time, new validators can join by meeting predefined eligibility criteria, such as community approval or governance decisions.

Validators can step down voluntarily or be replaced through governance if they fail to meet the network's expectations.

7.3. Accountability:

Misbehaving validators (e.g., those that remain offline or engage in malicious activities) can face penalties, such as being removed from the validator set or other sanctions, depending on the governance rules.

7.4. Flexibility:

Any trusted organization or individual can become a validator by applying through the network's governance process.

Validators can withdraw and leave the network at any time if they decide to opt out.

8. Validator Roles and Election Process

Validators on the Ruby blockchain are categorized into distinct roles based on their participation and contribution:

8.1. Active Validators (Cabinet):

The top K validators serve as the active validator set.

They are responsible for block production and validation during their tenure.

8.2. Standby Validators (Candidates):

Validators ranked between K+1 and K+N serve as standby nodes.

While they are not actively producing blocks, they are eligible to replace active validators when necessary.

8.3. Inactive Validators:

Validators outside the active and standby sets are considered inactive.

They do not participate in block production but can reapply to join the active or standby sets by meeting the network's criteria.

9. Election Timing:

The validator set is reviewed and updated at regular intervals.

The system uses the latest governance and validator rankings to refresh the active and standby validator lists, ensuring transparency and adaptability.

The Ruby blockchain ensures efficiency, reliability, and security by maintaining a well-defined validator set, balancing authority and accountability, and enabling transparent governance for network participants.

10. System Contracts in Ruby Blockchain

To support the functionality of staking, validator management, and other network operations, the Ruby-based blockchain implements several built-in system

contracts. These contracts enable automated, efficient, and transparent handling of core processes.

10.1. Validator Set Contract

10.1.1. *Purpose:*

Periodically elects and updates the active validator set based on the governance rules.

Maintains a vault for temporarily storing validator rewards.

10.1.2. *Key Features:*

Ensures the validator set is updated regularly to reflect network conditions.

Distributes validator rewards based on their performance and contributions.

10.2. System Reward Contract

10.2.1. *Purpose:*

Collects a portion of transaction fees and allocates funds for public purposes, such as maintaining network infrastructure or rewarding validators for fast finality.

10.2.2. *Key Features:*

Serves as a central repository for transaction fee allocations. Supports the distribution of incentives for validators and other key contributors to the network.

10.3. Slash Contract

Purpose:

Tracks validator misbehaviour and enforces penalties when predefined thresholds are reached. Handles specific slashable events, such as:

10.3.1. Unavailability: When a validator frequently goes offline.

10.3.2. Double Signing: When a validator signs conflicting blocks.

10.3.3. Malicious Voting: When a validator participates in harmful activities during finality voting.

Key Features:

Automatically imposes penalties, such as reducing a validator's rewards or removing them from the active validator set.

Protects the network's integrity by deterring and penalizing malicious behaviour.

10.4. Stake Hub Contract

10.4.1. *Purpose:*

Serves as the entry point for managing validators and delegation operations. Implements slashing logic for validators that violate network rules.

10.4.2. *Key Features:*

- Manages the lifecycle of validator operations, including:
- Delegation: Allowing users to delegate their tokens to validators.
- Undelegation: Enabling users to withdraw their stake.
- Redelegation: Allowing users to shift their delegation from one validator to another.
- Interacts with individual validator contracts to handle user stakes efficiently.

10.5. Benefits of System Contracts in the Ruby Blockchain

10.5.1. Automation: System contracts streamline complex operations like validator elections, rewards distribution, and slashing without requiring manual intervention.

10.5.2. Transparency: All contract operations are recorded on-chain, ensuring full transparency for participants.

10.5.3. Security: Slashing mechanisms and reward distribution are enforced through immutable smart contracts, reducing the risk of manipulation.

10.5.4. Flexibility: Delegators and validators benefit from a simple and efficient staking management system.

These system contracts form the backbone of the Ruby blockchain, enabling robust governance, efficient staking, and secure network operations.

11. Reward Distribution in Ruby Blockchain

The reward mechanism in the Ruby blockchain is designed to incentivize validators and delegators by distributing transaction fees efficiently and transparently.

11.1 Source of Rewards

- *Transaction Fees:*

  When a block is produced, the majority of the transaction fees collected within that block are allocated as rewards.

  The validator responsible for proposing the block receives the block's transaction fees as part of their reward.

11.2. Daily Reward Distribution

- *Validator Commission:*

  Each day, a portion of the collected rewards is automatically sent to the validator's operator account as a commission for their services.

- *Validator Credit Contract:*

  The remaining rewards are deposited into the validator's credit contract. These funds accumulate and are used to reward delegators based on their stake and participation.

11.3. Undelegation and Claiming Rewards

When a user chooses to undelegated their stake:

- The accumulated rewards are calculated and sent to the user.
- The original staked amount is returned to the user's account.

This process ensures that both validators and delegators are fairly rewarded based on their contributions to the network, while maintaining transparency and simplicity.

11.4. Key Benefits of Ruby Blockchain's Reward Distribution

11.4.1. Fair Incentives: Rewards are distributed proportionally, ensuring that validators and delegators are rewarded for their efforts.

11.4.2. Transparency: All reward transactions are recorded on-chain, allowing users to track their earnings and validate the process.

11.4.3. Flexibility: Delegators can easily claim their rewards and original stakes when they undelegate, ensuring liquidity and accessibility.

This reward distribution model aligns the interests of validators, delegators, and the network, fostering long-term participation and growth.

12. Security and Finality in Ruby Blockchain

In Ruby Blockchain network, the Ruby blockchain operates securely as long as more than (N/2 + 1) validators remain honest. This ensures proper functionality under normal conditions. However, certain Byzantine validators may still attempt to attack the network, such as through techniques like the "Clone Attack."

To enhance security, Ruby blockchain users are advised to wait for blocks sealed by at least (2/3N + 1) different validators. This threshold ensures that the blockchain achieves a security level comparable to highly secure networks and can tolerate up to 1/3N Byzantine validators without compromising integrity.

13. Probabilistic and Fast Finality

While the Ruby blockchain provides probabilistic finality, it also introduces a fast finality mechanism to permanently finalize blocks. Once a block is finalized, it becomes immutable and cannot be reverted. This mechanism is inspired by Casper FFG and follows a multi-step process:

13.1. Block Proposal:

A validator proposes a new block and broadcasts it to other validators in the network.

13.2. Validator Voting:

Validators use their BLS (Boneh-Lynn-Shacham) private keys to sign the block, generating a vote message to confirm the block’s validity.

13.3. Vote Collection:

Votes from validators are gathered into a pool for further processing.

13.4. Vote Aggregation:

If the direct parent block of the proposed block has received enough votes, the validator proposing the new block aggregates the BLS signatures.

The aggregated vote attestation is embedded into the extra field of the new block's header.

13.5. Justification of Blocks:

Validators and full nodes that receive the new block, along with its parent block's attestation, can justify the direct parent block.

13.6. Finalization:

When two consecutive blocks are justified, the first block is finalized and becomes immutable.

14. Key Benefits of Security and Finality in Ruby Blockchain

14.1. Enhanced Security:

By waiting for attestation from at least (2/3N + 1) validators, the network ensures high resilience against Byzantine faults.

14.2. Immutable Finality:

Fast finality ensures finalized blocks are permanent, preventing any possibility of rollback.

14.3. Efficient Validation:

The use of BLS signatures streamlines the voting and aggregation process, reducing overhead and improving network performance.

By combining probabilistic and fast finality, the Ruby blockchain achieves a high level of security, trust, and efficiency, making it a reliable platform for decentralized applications and services.

15. Governance in the Ruby Blockchain

The Ruby blockchain introduces a native governance module inspired by established frameworks like Open Zeppelin Governor, designed to enable decentralized decision-making and community participation.

Key Features of Ruby Blockchain Governance

15.1. Proposal and Voting Rights:

Staking credit holders can create and vote on governance proposals, ensuring stakeholders have a voice in network decisions.

15.2. Continuous Rewards:

Participants who vote on proposals continue to earn staking rewards during the voting period, incentivizing active participation in governance.

15.3. Flexible Delegation:

Users can delegate their voting rights to trusted representatives, allowing broader community involvement and representation in governance matters.

15.4. Secure Execution:

Passed proposals are subject to a time-lock period before execution, ensuring transparency and allowing time for potential disputes to be addressed.

16. Governance Process

Ruby blockchain governance operates through a two-step process, combining off-chain sentiment gathering and on-chain decision-making:

16.1. Temperature Check:

Any Ruby coin holder can submit a proposal for an initial community sentiment check.

This step is conducted off-chain using platforms like Snapshot, allowing the community to discuss and signal support for the proposal.

Proposals that receive sufficient backing proceed to the next phase.

16.2. Final Decision Voting:

Proposals approved during the temperature check move to on-chain voting.

Validators or participants with staked Ruby coins cast their votes to accept or reject the proposal.

The outcome of the vote determines whether the proposal is implemented or discarded.

17. Benefits of Ruby Blockchain Governance

17.1. Decentralized Decision-Making:

Enables stakeholders to actively participate in shaping the network's future.

17.2. Incentivized Participation:

Voters are rewarded for participating, encouraging higher engagement from the community.

17.3. Flexibility and Representation:

Delegation allows users to empower others to vote on their behalf, ensuring inclusivity.

17.4. Transparency and Security:

The use of a time lock ensures that changes are implemented securely and transparently, minimizing risks.

The governance model of the Ruby blockchain aligns the interests of validators, delegators, and the wider community, fostering a collaborative and secure ecosystem for long-term success.

18. Slashing and Penalties in Ruby Blockchain

Slashing is an essential component of the Ruby blockchain's governance framework, designed to penalize malicious or negligent actions that threaten the network's integrity. Anyone can submit a slash transaction on the Ruby blockchain by providing evidence and paying the associated transaction fees. Successful submissions are rewarded. Currently, there are three main types of slashable offenses.

18.1. Double Sign

Description:
A validator commits a severe violation when they sign more than one block with

the same height and parent block. This behaviour is considered deliberate and malicious since the protocol's implementation inherently prevents such errors.

18.2. Consequences:

- Immediate removal of the validator from the active validator set.
- A predefined amount of Ruby Coin is slashed from the validator's self-delegated stake.

18.3. Slash Submission Process:

- Anyone can submit a slash transaction by providing evidence of the offense to the Ruby Slash Contract.
- Evidence must include two block headers with the same height and parent block, signed by the offending validator.
- Upon verification, penalties are enforced, and the submitter receives a reward from the system contract.

18.4 Malicious Fast Finality Vote

18.4.1. **Description**:

This violation occurs when a validator submits two conflicting fast finality votes for the same target height or submits overlapping votes. Such behaviour is treated as a deliberate attempt to disrupt the network.

18.4.2. Consequences:

- Immediate removal of the validator from the active validator set.
- A predefined amount of Ruby coin is slashed from the validator's self-delegated stake.

18.4.3. Slash Submission Process:

- Evidence of malicious voting, including two conflicting votes and the validator's signing key, must be submitted to the Ruby Slash Contract.
- Once the evidence is verified, penalties are enforced, and the submitter receives a reward.

19. Unavailability

19.1. Description:

Validators are responsible for producing blocks during their assigned turn.

Failure to do so—caused by hardware issues, software malfunctions, configuration errors, or network problems—affects the network's performance and reliability.

19.2. Consequences:

An internal smart contract tracks each validator's missed block metrics.

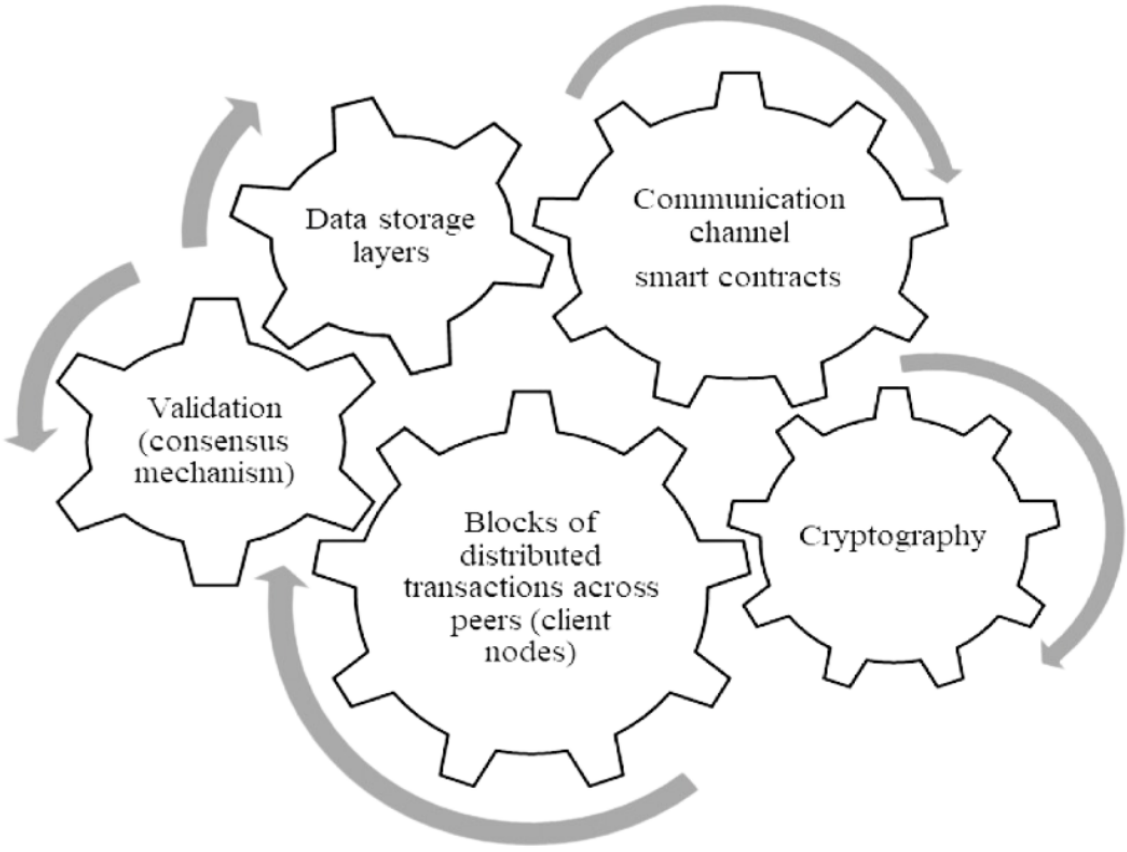
19.2.1. First Threshold:

Validators exceeding this threshold will forfeit their block rewards. These rewards are redistributed among more reliable validators.

19.2.2. Higher Threshold:

- Validators consistently failing to meet their responsibilities will be removed from the active set.
- A predefined amount of Ruby tokens will be deducted from their self-delegated stake.

Both the validator and their delegators lose staking rewards.



Building the Future with Ruby Blockchain.

Ruby envisions a blockchain ecosystem that is transparent, efficient, and scalable. Powered by the innovative Proof of Authority (PoA) consensus mechanism, Ruby combines fast transaction speeds, low energy consumption, and secure operations. Our protocol supports developers in creating decentralized applications (DApps) effortlessly while fostering community-driven collaboration.

With a focus on governance, accountability, and user empowerment, Ruby bridges the gap between decentralization and operational efficiency. By addressing scalability challenges and driving mainstream adoption, Ruby aims to democratize access to blockchain technology and unlock its transformative potential for communities worldwide.

Let's redefine blockchain innovation together.

Best regards,

Ruby Community