LAYER ONE X

Enabling Scalability and Interoperability with Mobile Computing

Whitepaper

V3.0

1. Contents

2. Abstract

Scalability and interoperability challenges hinder the vast adoption and sustainability of blockchain-oriented solutions. This paper describes Layer One X (L1X): an Interoperable, Decentralized, Secure and Scalable Layer One Smart Contract Protocol.

To provide **Interoperability**, this paper proposes to change the terminology to **X-Talk**; L1X offers two Virtual Machines, one is termed as X-Talk Virtual Machine and the other is the core Virtual Machine which is called L1X Virtual Machine. X-Talk is a subset of the L1X VM. X-Talk leverages the ability to allow native (Within L1X) and Multi-Chain (Cross Chain and Cross Blockchain) transactions, interact, authenticate, validate and pass smart contract invocation and implementation which paves the way for multiple use cases for example, cross chain tokenization, staking which results in situations where an asset is tokenized on chain A and a borrow is done on Chain B. The difference between X-Talk and Bridges are many, but at the core is the ability to modify the payload to be sent to the destination chain based on third party conditions and to check the destination payload compatibility which is only done by the user at the core by optionally choosing the L1X MultiSig functionality.

The **Consensus Mechanism** has been renamed from Proof of Participation to Proof of X (PoX). The Consensus Mechanism has Full Validator Nodes (FVNs) and Mobile Enabled or Low CPU Powered Devices that act as Validators on the network. To Provide and maintain an inflationary **growth rate of Decentralization** as the protocol scales; it is important to consider the following points.

1. Staking and a dynamic Ceiling allocation on the Staked Value
2. Randomization Algorithm for Block Proposing for the Nodes
3. Cluster Allocation and Randomization rotation of the Cluster
4. The ability to allow mobile enabled devices or low powered chipsets to be the validators on the network. The Validators role is to maintain an encrypted format of cluster registry, cluster registry checkpoint and other Randomization related features. This combined with the ability of the L1X VM which can compile

WASM bytecode to eBPF bytecode will pave the way for Decentralized Cloud Computing.

These features of L1X work together to provide an incremental growth rate of Decentralization as the network scales and adds more clusters and validators to the protocol.

To provide **Security** in a Decentralized Ledger, Smart Contract Platform (Flash Code / FC), there are areas such as block proposing, stake registry maintenance, accidental forking and others that are being investigated. From L1X perspective, the protocol must redefine the security components as it includes features built from the ground up including its Virtual Machine, X-Talk, Consensus Mechanism and Multi-Sig Native On-chain Collection. The Pentagon Framework is used where the goal is to remove the problems of

1. Updating the State,
2. Flash Code Logic Validation,
3. Deterministic behavior of the Virtual Machine,
4. Randomized Selection of Block Proposer,
5. Randomizing the Cluster Registry with the Validators,

To solve the Pentagon problem there are certain technology components that need to come together in the form of innovation in a Layer One Smart Contract platform. Those components are:

1. WASM to eBPF Compiler
2. X-Talk Architecture and System Workflow with its Implementation
3. Native MultiSig Contract that is Native (L1X) and Multi-Chain compatible with the involved user(s)

To provide a scalable and high performing system that delivers a high throughput initially and can have an elasticity behavior to its growth rate of requirements; L1X has positioned itself to initially provide 100,000 transactions per second (TPS) (On Launch) and have a two-three digit growth rate; L1X and its capability to interact with the various technologies that it provides, including the Virtual Machine, Cluster Based Node Architecture and State

Management, Hybrid Consensus Mechanism and X-Talk synchronous transaction processing provides a framework for a scalable protocol. The focus of the protocol is to provide a faster Seconds per Transaction (SPT) and using Zero Knowledge Proof Trailing system provides a framework for a protocol that does not only provide high Transactions per Second but also a faster timeframe for Seconds per Transaction.

The L1X Consensus Mechanism also delivers a capability to host Subnets providing Governments, Larger Enterprises and Businesses (Public, Private or Hybrid Networks) requiring smart contract privacy, custom consensus mechanism rule set and FVN to validator node ratio settlement, providing a mechanism to tap into the existing infrastructure without needing a layer two solution. These subnets allow to; after a certain interval through time and block; roll information to the main chain if the involved entity decides to extend its security.

Finally, transaction and block statistics reveal that L1X achieves a compact transaction size of 100 bytes, 500 milliseconds short block time, quick transaction finality in 1.5 seconds, and 5 MB manageable block size, paving the way for efficient and scalable transaction processing.

3. Preface

It is with great excitement that we present this whitepaper, which addresses the fundamental challenges faced by blockchain technology in achieving optimal decentralization, security, scalability, and interoperability. This document introduces groundbreaking concepts and innovative solutions that pave the way for a new era of blockchain advancements.

In the realm of blockchain technology, the pursuit of decentralization, security, scalability, and interoperability has always been of paramount importance. However, achieving a balance among these four crucial elements has remained an elusive goal. This whitepaper aims to revolutionize the blockchain landscape by proposing a novel approach that aims to connect all blockchains and unlock their full potential.

Our solution centers around the introduction of X-Talk, a revolutionary concept that redefines interoperability across blockchain networks. X-Talk allows for seamless communication and interaction between native and multi-chain environments. By leveraging X-Talk, our proposed blockchain ecosystem aims to break down the barriers hindering efficient interoperability.

Furthermore, this whitepaper introduces a state-of-the-art Virtual Machine architecture that serves as the backbone of our solution. This Virtual Machine, known as the X-Talk Virtual Machine, works in conjunction with the core Virtual Machine, the L1X Virtual Machine, to deliver unparalleled scalability, security, and flexibility. Together, these virtual machines empower developers and users alike to explore new frontiers of blockchain applications, transcending the limitations of existing systems.

To ensure the robustness and resilience of our proposed blockchain network, we introduce a cutting-edge consensus mechanism named Proof of X (PoX). PoX leverages a distributed network of validators, with a unique twist: it allows mobile devices, even those with low computational power, to participate as validators. This inclusive approach not only enhances decentralization but also opens doors for broader participation, making blockchain technology more accessible and democratic.

This whitepaper presents a comprehensive framework that addresses the quadrant of decentralization, security, scalability, and interoperability. By incorporating novel concepts, such as X-Talk, our Virtual Machine architecture, and the PoX consensus mechanism, we believe that L1X can transcend the limitations of existing blockchain solutions.

We invite readers, researchers, and industry professionals to delve into the intricacies of our proposed framework. Together, let us forge a path toward a new era of blockchain technology, where the power of L1X innovative architecture is harnessed to unite and thus maximize the potential of different blockchain networks.

4. Introduction

Blockchain technology has garnered significant attention from industry and academia alike, offering immense potential for transformative applications. However, the adoption of blockchain faces notable challenges in scalability, interoperability, and privacy, while maintaining the essential principles of decentralization and security. This whitepaper presents the Layer One X framework, a comprehensive solution designed to address these challenges and propel the blockchain ecosystem forward.

Achieving scalability, interoperability, security, and decentralization simultaneously in blockchain networks (referred to as quadrilemma) has remained a complex endeavor. Many blockchain professionals have observed the limitations that impede the widespread adoption and advancement of blockchain solutions. This whitepaper aims to present a groundbreaking framework that integrates innovative concepts and novel technologies, providing a holistic approach to overcome existing limitations.

The framework's core objective is to revolutionize the way blockchains interact with each other. The aim is to unite all blockchain networks and unleash their maximum potential through the provision of decentralized multichain interoperability. To enable seamless communication and collaboration between blockchains, a truly decentralized, permissionless, and secure blockchain ecosystem is established that supports micro validation and tokenization. By thoroughly exploring the intricate interplay of quadrilemma, Layer One X framework offers a unified synergistic solution.

The innovative architecture of the Layer One X framework introduces the concept of X-Talk, which fundamentally reshapes the notion of interoperability within blockchain networks. By seamlessly connecting native and multi-chain environments, X-Talk revolutionizes smart contract invocation, validation, and execution, unlocking a plethora of cross-chain interaction possibilities. This notion is reinforced by the L1X Virtual Machine, which exhibits accelerated transaction processing capabilities.

To ensure a resilient and inclusive blockchain network, the Proof-of-X (PoX) consensus mechanism is introduced. This approach harnesses a distributed network of validators, including low-powered mobile devices, facilitating broader participation and enhancing true decentralization. PoX embodies a democratic and accessible blockchain ecosystem, fostering increased security and robustness.

This comprehensive framework effectively addresses the fundamental challenges of quadrilemma by synergistically converging the innovative concepts such as X-Talk, L1X Virtual Machine, the PoX consensus mechanism, MultiSig Native On-Chain Collection and State Management to establish a collaborative L1X architecture.

The remainder of this paper is structured as follows. The next section of this paper details scalability and interoperability limitations in existing blockchain solutions. Section 6 provides briefs about the design principles, prominent features, and benefits of the L1X network. Mobile nodes and their significance in the L1X network are described in section 7. Section 8 details the overview architecture of L1X. In section 9, the L1X transaction lifecycle is explained followed by Section 10 detailing Transaction Types in L1X. Section 11 details the X-TALK architecture for cross-blockchain interoperability and L1X Virtual Machine is detailed in Section 12. Section 13 emphasizes the need for AVDR and describes the workflow. DB diversity is achieved in L1X through DB Driver abstraction described in Section 14. The possibility to interconnect diverse blockchains is made feasible through X-Talk as the L1X Oracle System, explained in Section 15. Similarly, to broader the L1X ecosystem, Section 16 allows EVM instance creation and deployment in L1X blockchain network. The L1X rewards mechanism is explained in Section 19 along with L1X Tokenomics briefed in section 20 of this paper. After detailing the technical components till Section 20, L1X applications are detailed in Section 21. Use cases of the L1X system are discussed in section 22. The paper is concluded in section 23.

5. Scalability and Interoperability Limitations of Existing Blockchain Networks

Scalability and interoperability hinder vast adoption and sustainability of blockchain-oriented solutions. Sharding being one of the most practical solutions, to reduce overhead of duplicate communication, storage and computation, is widely employed. As the size of the network increases, high latency, low throughput and weak robustness are observed. Directed Acyclic Graph (DAG) allows to position many blocks at once where transactions are linked to one another in the form of graph. This approach resolves speed and scalability issues but at the expense of security and centralization. Performance of the blockchain can also be improved using Off-chain solutions. Off-chain stores and data processing occur on a separate publicly inaccessible blockchain and broadcasts a summary of batch of off-chain transactions to the main blockchain. Privacy and security issues have been identified in the off-chain solutions. Similar to off-chain are cross-chain solutions with a difference that they facilitate interaction among multiple cross blockchain networks. In such solutions either trade-off between scalability and centralization is observed or privacy and locked funds are the concern. Another method to offload computation and increase scalability is through mobile computing. Resource orchestration is the critical aspect of mobile computing which can be managed through optimized resource allocation algorithms. Out of the very limited solution being proposed under this category, trade-offs between latency and security or support for multiple heterogeneous applications have been identified as issues. Table 1 lists the scalability projects under varied scalability solutions.

Scalability Solutions	**Scalability Projects**
Sharding	Elastico, Zilliqa, Omniledger, Monoxide, RapidChain, Chainspace, Ethereum 2.0, EOS
DAG	BlockClique, Byteball, DagCoin, Nano, IOTA
Off-chain	Lightning Network, Raiden Network, Sprites, Plasma
Cross-Chain	Multi-centre witness, Relay, Hash Locking, Distributed Private Key Control
Mobile Computing	EdgeChain, EdgeAI

Table 1 Scalability Solutions and Projects

Interoperability solutions can be categorized into three generation solutions as depicted in Table 2.

The first generation of blockchain interoperability solutions identifies and describes different interoperability strategies across blockchains including sidechain and relay approaches, notary schemes, and hash time lock contracts. Sidechains are an extension of the main chain where the communication protocol facilitates asset transfer between the two. It does not allow horizontal communication among the sidechains resulting in inconsistent state in the closure phase. In relay solutions, relayers keep track of the block headers of the mainchain and input them to the relay smart contract hosted on another blockchain. These block headers can be used for verification of transactions or information. Bad actors can then take control of a potentially small set of validators. Increasing validator set can result in slow transaction settlement. Notary schemes are fundamentally intermediaries between blockchains as they execute actions on behalf of the end- users or facilitate matching for the end-users with trade offers. Such solutions are preferred for cryptocurrency exchanges where an order book is managed, and buyers and sellers are matched. Hashed Time-Lock Contracts (HTLCs) enable atomic operations between different blockchains using hashlocks and timelocks allowing asset exchange without a pre-existing trust relationship between the two different blockchains. The protocol and governance in HTLC implementation are different across existing solutions which result in blockchains becoming siloed.

Interoperability Solutions	**Interoperability Techniques**	**Interoperability Projects**
First Generation	Sidechain, Relay, Notary Schemes, HTLC	Lightning Network, BTC Relay, Peace Relay Binance, Coinbase, Liquid, Zendoo, Wanchain, Fusion, Xclaim, Xchain, DeXTT, Blocknet, RSK, Loom Network
Second Generation	Bridge, Hub-and-spoke	Polkadot, COSMOS, Ark, Aion, Komodo
Third Generation	Hybrid Connectors	Trust, relays, Interledger, Protocol, Hyperledger Quilt

Table 2 Interoperability Solutions and Projects

The second generation of blockchain interoperability solutions provides the ability to create application specific blockchains that can interoperate between the customized blockchains. This generation of blockchain interoperability solutions use strategies of bridge or hub-and- spoke which also introduces limitations. A few among these solutions are highly experimental products. The third generation of blockchain interoperability solutions known as hybrid connectors attempt to deliver a blockchain abstraction layer in which a set of uniform operations can be exposed to allow an interaction between blockchains without the need of using different APIs. Under this category, some solutions are not entirely decentralized, and a few rely on trust on connector.

L1X provides a scalable and interoperable solution without compromising on security and decentralization as detailed in subsequent sections.

6. L1X Network

Layer One X (L1X) is a state-of-art blockchain network with an Interoperable, Decentralized, Secure, and Scalable Layer One Smart Contract Protocol. By offering interoperability through the X-Talk architecture, redefining security components, emphasizing scalability, and incorporating mobile nodes as validators, L1X strives to create a blockchain network that is inclusive, efficient, secure, and capable of supporting a wide range of applications. The following design principles and features lay the foundation for the L1X protocol, presenting a promising path towards a more interconnected and sustainable blockchain ecosystem.

6.1 Design Principles

With the objective of establishing a collaborative architecture, the following design principles are framed to achieve this aim. These principles aim to address the scalability and interoperability challenges that hinder the widespread adoption and sustainability of blockchain-oriented solutions.

1. **Interoperability**: emphasizes the need for a solution that enables both native and multi- chain communications, ensuring seamless authentication, validation, and invocation of smart contracts across diverse chains. This capability fosters cross-chain tokenization and facilitates the realization of various use cases within the blockchain ecosystem.

2. **Consensus Mechanism**: highlights the importance of achieving true decentralization and an inflationary growth rate through a consensus mechanism that includes Full Validator Nodes (FVN) and mobile-enabled devices in transaction validation. To enhance security, the selection of block proposers for staked participants will be completely randomized. Additionally, to ensure scalability, the consensus mechanism should incorporate the active participation of mobile nodes. These measures collectively contribute to the robustness and scalability of the blockchain network.

3. **Security:** aims to address key challenges in the network, including consistent state updates, flash code logic validation, deterministic behavior of the virtual machine,

randomized selection of block proposers, and randomization of the cluster registry with validators. By effectively resolving these challenges, ensures the network is able to provide reliable state management, secure execution of smart contracts, deterministic outcomes, fair block proposer selection, and a randomized distribution of validators within the cluster registry. This comprehensive approach contributes to the overall robustness and stability of the blockchain ecosystem.

4. **Scalability and Performance**: sets forth an ambitious goal of achieving a high initial throughput of 100,000 transactions per second (TPS). Moreover, it emphasizes the network's ability to dynamically and seamlessly adapt to evolving demands and requirements. This adaptability should enable the network to scale its capacity, throughput, and computational resources in real-time, ensuring efficient handling of growing workloads, increased transaction volumes, and an expanding user base. By incorporating these capabilities, the blockchain network can effectively address the challenges of scalability and cater to the needs of a rapidly evolving ecosystem.

5. **Subnets for Privacy and Customization**: Leveraging the power of subnets, L1X aims to offer customizable solutions tailored to specific privacy requirements and custom consensus rules. These subnets should enable the flexibility to establish desired ratios between FVNs and validator nodes, ensuring optimal network performance. Furthermore, L1X will support seamless integration with existing infrastructure, allowing entities to leverage their current systems without the need for additional layers. By providing these capabilities, L1X can empower users to tailor their blockchain environment according to their unique needs while maximizing efficiency and compatibility.

6. **Transaction and Block Statistics**: With a focus on efficiency, L1X aims to achieve impressive block creation timeframes of just 500 milliseconds, enabling rapid block generation within the network. Furthermore, L1X strives for transaction finality in a mere 1.5 seconds, ensuring quick confirmation and settlement of transactions. Significantly, these timeframes are coupled with block sizes of approximately 5 MB, highlighting the network's capacity to handle a substantial volume of transactions. This combination of fast block creation, swift transaction finality, and sizable block

sizes holds immense significance in enhancing the overall performance and responsiveness of the L1X blockchain network, fostering a seamless and efficient user experience.

These design principles collectively form the foundation of the Layer One X (L1X) protocol, driving interoperability, scalability, security, and performance in the real-world.

6.2 Prominent Features

Based on the aforementioned design principles, L1X encompasses a range of notable features that contribute to its value proposition. These features include seamless interoperability between native and multi-chain communications, a dynamic consensus mechanism that fosters true decentralization and inflationary growth, the Pentagon Framework to ensure security and deterministic behavior, and a focus on scalability and performance. Through the integration of these prominent features, L1X presents a comprehensive solution that addresses critical challenges in the blockchain landscape, ultimately paving the way for widespread adoption and sustainable blockchain-oriented solutions.

1. **Interoperability (X-Talk)**
    - X-Talk enables seamless interaction, authentication, and validation of smart contracts across multiple chains.
    - Native and multi-chain compatibility allows for cross-chain tokenization and borrowing, unlocking new use cases.
    - X-Talk distinguishes itself from a traditional bridge by modifying payload based on third-party conditions and ensuring destination payload compatibility through optional L1X MultiSig functionality.

2. **Consensus Mechanism (Proof of X)**
    - Proof of X (POX) consensus mechanism ensures decentralization and inflationary growth as the protocol scales.
    - Validators include FVNs and mobile-enabled/low-CPU-powered devices,

promoting inclusivity and accessibility.

- Staking and dynamic ceiling allocation on staked value encourage active participation and network security.
- Randomization algorithms for block proposing and cluster allocation enhance security and fairness.

3. **Security (Pentagon Framework)**

- The Pentagon Framework addresses critical security components from the ground up, ensuring a secure network.
- Updatable state, flash code logic validation, and deterministic behavior of the virtual machine are safeguarded.
- Randomized selection of block proposers and cluster registry with validators enhances security measures.
- Multi-Sig Native On-chain Collection provides additional security and compatibility features.

4. **Scalability and Performance**

- L1X offers a scalable and performant network, starting with an impressive initial throughput of 100,000 transactions per second (TPS) and the potential for substantial growth.
- The L1X virtual machine, cluster-based node architecture, hybrid consensus mechanism, and synchronous transaction processing in X-Talk enable scalability.
- Faster Seconds per Transaction (SPT) and Zero Knowledge Proof Trailing ensure high TPS and reduced transaction timeframes.
- L1X supports Decentralized Cloud Computing through its mobile-enabled devices and low-powered chipsets as validators.

5. **Subnets for Privacy and Customization**

- L1X allows the creation of subnets, catering to entities requiring smart contract privacy and customized consensus mechanism rules.
- Subnets provide a flexible FVN to validator node ratio and enable the rolling of information to the main chain when needed.
- Governments, larger enterprises, and businesses benefit from enhanced security and tap into existing infrastructure without requiring a layer two solution.

6. **Transaction and Block Statistics**

- L1X boasts efficient transaction and block statistics.
- The average transaction size is approximately 100 bytes.
- Block time is set at 500 milliseconds, ensuring fast confirmation times.
- Transaction finality time is targeted to be 1.5 seconds, providing quick settlement and responsiveness.
- Each block can accommodate up to 50,000 transactions, resulting in a block size of 5,000,000 bytes or 5 MB.

The Layer One X (L1X) network incorporates these features to offer a truly interoperable, decentralized, secure, and scalable Layer One Smart Contract Protocol. By combining innovative technologies, such as X-Talk, the Pentagon Framework, and the PoX consensus mechanism, L1X provides a robust foundation for building blockchain-oriented solutions that overcome the challenges of scalability, interoperability, security, and performance.

In the subsequent section, we detail the four cornerstones of L1X and the techniques employed to achieve them without compromising any among them.

6.2.1 Decentralization

Decentralization is a fundamental feature of the Layer One X (L1X) network, aligning with the design principles and enhancing the overall functionality and sustainability of the protocol. Below are the rationale highlighting the inclusion of decentralization as a key feature in L1X:

1. **Consensus Mechanism (Proof of X)**: L1X implements the Proof of X (PoX) consensus mechanism, which distributes the validation and decision-making process among a diverse set of validators. By allowing FVNs and mobile-enabled/low-CPU-powered devices to act as validators, L1X ensures a decentralized network where multiple participants have an active role in maintaining the integrity and security of the system. This distributed approach prevents any single entity from monopolizing control, promoting a more democratic and inclusive network.

2. **Validators and Cluster Allocation**: L1X employs a cluster-based node architecture, where validators play a crucial role in maintaining an encrypted format of the cluster registry, cluster registry checkpoint, and other randomization-related features. By rotating and allocating clusters randomly, L1X ensures that no single entity or group of validators can gain undue influence or manipulate the network's operation. This decentralized cluster allocation mechanism ensures a fair distribution of power and resources across the network.

3. **Subnets for Customization**: L1X's support for subnets enables governments, larger enterprises, and businesses to create their own private networks with customized consensus mechanisms and rules. By providing the ability to establish subnets within the L1X network, the protocol allows entities to maintain control over their specific requirements while still benefiting from the overall security and interoperability provided by the L1X protocol. This feature empowers different entities to operate within their own decentralized environments, tailored to their specific needs.

4. **Inclusion of Mobile-Enabled Devices**: L1X recognizes the importance of accessibility and inclusivity in a decentralized network. By allowing mobile-enabled devices and low- powered chipsets to participate as validators, L1X expands the pool of participants beyond traditional high-powered computing systems. This inclusion ensures that individuals with diverse resources and capabilities can contribute to the network's decentralization, fostering a more distributed and resilient ecosystem.

Decentralization in L1X is not only a feature but a core principle that underpins the entire protocol. By distributing power, decision-making, and participation across a wide range of validators, clusters, and subnets, L1X establishes a robust and resilient network that is less susceptible to single points of failure, censorship, or control. This decentralized nature enhances trust, security, and the overall sustainability of the Layer One X network, making it a reliable choice for blockchain-oriented solutions.

6.2.2 Interoperability

Interoperability is a crucial feature of the Layer One X (L1X) network, addressing the challenges posed by the lack of seamless interaction and compatibility between different blockchains and chains. Interoperability is emphasized as a key feature, especially through the X-Talk architecture, for the following reasons:

1. **Eliminating Silos and Fragmentation**: L1X recognizes that the blockchain ecosystem is highly fragmented, with different chains and blockchains operating in isolation. By introducing the X-Talk architecture, L1X aims to break down these silos and enable seamless interaction, communication, and data transfer between different chains. This interoperability promotes a cohesive and connected blockchain ecosystem, fostering collaboration, innovation, and the development of new use cases.

2. **Native and Multi-Chain Compatibility**: X-Talk enables both native compatibility within the L1X network and multi-chain compatibility across different blockchains. This means that smart contracts and assets deployed within the L1X network can seamlessly interact and operate with other L1X contracts, as well as with external chains and blockchains. This compatibility allows for cross-chain tokenization, staking, and other functionalities, expanding the possibilities for decentralized applications and creating new opportunities for users and developers.

3. **Enhancing Use Cases and Functionality**: The X-Talk architecture unlocks multiple use cases by enabling native and multi-chain interactions. For example, assets can be tokenized on one chain (Chain A) and used for borrowing on another chain (Chain B). This interoperability expands the scope of decentralized applications, making it possible to create complex, interconnected ecosystems that leverage the strengths and

capabilities of different chains and blockchains.

4. **Payload Modification and Compatibility**: X-Talk differentiates itself from traditional bridge solutions by offering the ability to modify the payload sent to the destination chain based on third-party conditions. This allows users to adapt and customize the transaction payload to meet specific requirements or conditions on the destination chain. The optional use of L1X MultiSig functionality provides an additional layer of security and validation for payload compatibility, ensuring a smooth and secure interoperability process.

5. **Enabling Cross-Chain Communication and Validation**: X-Talk not only facilitates cross-chain data transfer but also enables the authentication, validation, and passing of smart contract invocations and implementations across different chains. This capability ensures that transactions and interactions between chains are secure, reliable, and consistent, enhancing trust and confidence in cross-chain operations.

By incorporating the X-Talk architecture and emphasizing interoperability, L1X breaks down barriers between different chains and blockchains, fostering a connected and collaborative ecosystem. This feature promotes innovation, expands the functionality and use cases of decentralized applications, and empowers users and developers to leverage the strengths of various chains to create interconnected and interoperable solutions. The interoperability of L1X through X-Talk is a significant step toward achieving widespread adoption and sustainable growth in the blockchain industry.

6.2.3 Security

Security is a critical feature in the design of the Layer One X (L1X) network, ensuring the integrity, confidentiality, and resilience of the protocol. Emphasizing security as a key feature is supported by the following reasons:

1. **Pentagon Framework**: L1X incorporates the Pentagon Framework, which addresses key security components from the ground up. By focusing on updating the state, flash code logic validation, deterministic behavior of the virtual machine, randomization of block proposers, and cluster registry with validators, L1X establishes a strong

foundation for a secure smart contract platform. The Pentagon Framework tackles vulnerabilities and potential attack vectors, mitigating risks and enhancing the overall security of the network.

2. **Multi-Sig Native On-chain Collection**: L1X introduces the concept of Multi-Sig Native On-chain Collection, providing additional security features. By leveraging this functionality, users can enhance security through multi-signature transactions that involve multiple parties in the validation process. This capability ensures the integrity and authenticity of transactions, reducing the risk of unauthorized or fraudulent activities within the network.

3. **Interoperable Security Components**: L1X redefines security components, incorporating them into its virtual machine, X-Talk architecture, consensus mechanism, and multi-sig native on-chain collection. This approach ensures that security measures are integrated seamlessly across different layers of the protocol, providing end-to-end security for smart contract execution and interaction.

4. **Subnets for Privacy and Customization**: L1X's support for subnets enables entities requiring smart contract privacy and customized consensus mechanism rules to establish secure and private environments. By allowing entities to create subnets within the L1X network, sensitive operations can be shielded from public scrutiny, and specific security requirements can be tailored to meet the needs of different entities. This feature provides an additional layer of security for enterprises, governments, and organizations operating within the L1X network.

5. **Focus on Auditing and Code Verification**: L1X emphasizes the importance of auditing and code verification processes to ensure the security of smart contracts deployed on the network. By promoting best practices in code review, third-party audits, and rigorous testing, L1X reduces the likelihood of vulnerabilities, bugs, or malicious code being introduced into the system. This proactive approach enhances the overall security posture of the network.

By incorporating these robust security measures and design principles, L1X ensures the protection of user assets, data privacy, and the overall stability of the network. The focus on the Pentagon Framework, multi-sig native on-chain collection, interoperable security components, subnets for privacy, and a strong emphasis on auditing and code verification collectively contribute to a secure environment for deploying and executing smart contracts. These security features instill confidence in users and foster trust in the Layer One X network as a secure and reliable blockchain solution.

6.2.4 Scalability

Scalability is a key feature of the Layer One X (L1X) network, addressing the challenges that hinder the widespread adoption and sustainability of blockchain-oriented solutions. Below arguments support the notion that scalability is an important component.:

1. **High Throughput**: L1X aims to provide a high throughput network, starting with an initial capacity of 100,000 transactions per second (TPS) and the potential for significant growth. This high transaction processing capability allows the network to handle a large volume of transactions efficiently, meeting the demands of applications with high transactional requirements.

2. **Incremental Growth Rate of Decentralization**: L1X's design principles focus on maintaining an incremental growth rate of decentralization as the network scales. The staking and dynamic ceiling allocation on staked value incentivize participation and ensures a decentralized network. As the network expands and adds more clusters and validators, the decentralization of the network increases, enhancing its resilience and security.

3. **Cluster-Based Node Architecture**: L1X employs a cluster-based node architecture, which allows for the efficient distribution of network resources and workload across multiple clusters. This architecture enables horizontal scalability, as new clusters can be added to accommodate growing demand and increase the network's capacity. The randomization of cluster allocation further enhances fairness and prevents concentration of power within specific clusters.

4. **Hybrid Consensus Mechanism**: L1X utilizes a hybrid consensus mechanism,

combining the strengths of different approaches to achieve scalability. By striking a balance between security and scalability, L1X ensures that the network can handle a large number of transactions without compromising on decentralization or consensus robustness. This hybrid approach optimizes the trade-off between scalability and security, enabling the network to scale effectively.

5. **X-Talk Architecture**: The X-Talk architecture within L1X enables interoperability and interaction across multiple chains, paving the way for innovative use cases and expanding the network's scalability potential. By leveraging native and multi-chain compatibility, X- Talk allows for cross-chain tokenization, staking, and other interactions, increasing the scalability of decentralized applications that require seamless cross-chain functionality.

6. **Decentralized Cloud Computing**: L1X's support for mobile-enabled devices and low-powered chipsets as validators opens the possibility for decentralized cloud computing. By leveraging the computing power of these devices, L1X can harness their collective resources to provide scalable and performant decentralized cloud computing solutions. This approach not only enhances scalability but also promotes inclusivity and accessibility in the network.

The scalability features in L1X, including high throughput, incremental growth rate of decentralization, cluster-based node architecture, hybrid consensus mechanism, X-Talk architecture, and support for decentralized cloud computing, collectively contribute to the network's ability to handle increased transactional demands, accommodate growth, and provide a scalable infrastructure for blockchain-oriented solutions. By addressing scalability challenges, L1X offers a sustainable and efficient platform for a wide range of applications, paving the way for the widespread adoption of blockchain technology.

6.3 L1X Benefits

The Layer One X (L1X) network offers a range of features that collectively provide numerous benefits for users and developers in the blockchain ecosystem:

1. Handles a large volume of transactions efficiently, accommodating growth and increasing scalability. This scalability enables the network to support diverse applications with high transactional requirements.

2. Enables seamless interaction, communication, and data transfer between different chains and blockchains. This interoperability eliminates silos and fragmentation, expands use cases, and enhances functionality by enabling native and multi-chain compatibility. Users can tokenize assets, engage in cross-chain operations, and create interconnected ecosystems, fostering collaboration, innovation, and the development of new applications.

3. Ensures the integrity, confidentiality, and resilience of the network, providing a secure environment for smart contract execution and interaction.

4. Promotes a decentralized network, incentivize participation, distribute resources efficiently, and increase the network's resilience and security.

5. Provides a scalable and performant system with high throughput and elasticity. The network initially supports 100,000 transactions per second (TPS) with the potential for two-three-digit growth.

6. Enables fast transaction speeds and efficient resource utilization, providing a framework for a protocol with high performance and faster seconds per transaction (SPT).

7. Allows governments, larger enterprises, and businesses to create secure and private environments within the network. Users can extend security and customize their operations while leveraging the existing infrastructure of the L1X network.

6.4 Key Benefits for Developers

As web3 dApp developers embrace L1X, they gain access to a multitude of benefits that empower their solution development.

1. **Enhanced Interoperability**: L1X's X-Talk architecture enables seamless communication and interaction between different chains, providing dApp developers with a broader ecosystem to integrate and collaborate with, expanding the potential user base and utility of their applications.

2. **True Decentralization**: The inclusion of mobile-enabled devices as validators in the consensus mechanism promotes a more decentralized network, empowering dApp developers to build applications on a secure and trustless infrastructure that is not reliant on a central authority.

3. **Robust Security**: The Pentagon Framework ensures a high level of security for dApp development, mitigating risks related to state updates, flash code logic validation, and ensuring deterministic behavior of the virtual machine. This instills confidence in developers and users, making the platform more attractive for deploying mission-critical and sensitive applications.

4. **Scalability and Performance**: L1X's scalability features and high transaction throughput allow dApp developers to build applications that can handle a large number of users and process transactions quickly, providing a smooth and responsive user experience even during peak usage periods.

5. **Customizability and Privacy**: The subnets feature of L1X allows dApp developers to create private or custom consensus subnetworks, offering privacy for sensitive applications and the flexibility to define specific consensus rules that align with their unique requirements.

6. **Fast Transaction Confirmation**: L1X's fast block creation timeframes and transaction finality ensure quick confirmation and settlement of transactions, reducing latency and enabling dApp developers to deliver real-time interactions and responsiveness to their users.

The combination of these features in the L1X network creates a robust and versatile blockchain ecosystem. Users and developers benefit from collaboration, and trust within the blockchain community a scalable, interoperable, secure, decentralized, and high-performing platform that supports a wide range of applications and fosters innovation.

7. Mobile Nodes

In the Layer One X (L1X) network, mobile-enabled devices and low-powered chipsets have a significant role as validators, contributing to the decentralization and security of the network. These mobile nodes bring unique advantages and play a crucial part in the validation process. Here, we delve into the significance of mobile nodes as validators in the L1X network.

1. **Inclusive Participation**: By allowing mobile devices to act as validators, L1X promotes inclusivity and broad network participation. Unlike traditional blockchain networks that primarily rely on high-powered nodes, L1X extends the opportunity to a wider range of devices. Mobile devices are ubiquitous, accessible, and owned by individuals worldwide. Enabling these devices to participate as validators empowers users who may not have access to high-performance computing equipment but still wish to contribute to the network's operation and consensus process.

2. **Efficient Resource Utilization**: Mobile nodes offer an opportunity for efficient resource utilization within the L1X network. While high-powered nodes play a critical role in providing robust computational resources, mobile devices can contribute their computing power during idle or low-usage periods. By leveraging the idle resources of these devices, L1X optimizes resource allocation and increases the overall efficiency of the network. This efficient utilization helps distribute the validation workload across a diverse set of devices, ensuring scalability and mitigating the concentration of power.

3. **Decentralization and Resilience**: Mobile nodes enhance the decentralization and resilience of the L1X network. By allowing a wide array of devices to participate in the validation process, L1X reduces the dependency on a small number of powerful nodes, thus mitigating the risk of centralization. The distributed nature of mobile nodes enhances the network's resistance to attacks

and ensures that no single point of failure can compromise the entire network. This decentralization strengthens the overall security and robustness of the L1X ecosystem.

4. **Encrypted Cluster Registry and Checkpoint**: As validators, mobile nodes fulfill critical responsibilities in maintaining an encrypted format of the cluster registry and cluster registry checkpoint. These components play a vital role in the randomization of cluster allocation and rotation, which further enhances the fairness, security, and integrity of the L1X network. Mobile nodes actively contribute to the distribution and management of these essential elements, ensuring a reliable and trusted validation process.

5. **Decentralized Cloud Computing Possibilities**: The inclusion of mobile nodes as validators in the L1X network opens up possibilities for decentralized cloud computing. By leveraging the collective computing power of mobile devices, L1X can tap into a vast network of distributed resources. This enables the development of decentralized cloud computing solutions, where mobile devices collectively contribute to tasks such as data storage, processing, and execution. Decentralized cloud computing aligns with the ethos of blockchain technology by leveraging existing resources rather than relying on centralized cloud infrastructure, thereby enhancing the overall scalability and efficiency of the network.

In conclusion, mobile nodes serve as valuable validators in the Layer One X network, contributing to decentralization, resilience, and the efficient utilization of resources. By incorporating mobile devices as validators, L1X promotes inclusivity, broadens participation, and enhances the security and scalability of the network. Mobile nodes play a vital role in the distributed validation process, ensuring the integrity of the cluster registry and checkpoint, and paving the way for innovative decentralized cloud computing solutions. The incorporation of mobile nodes as validators demonstrates L1X's commitment to fostering a diverse, decentralized, and accessible blockchain ecosystem.

8. L1X Architecture

This section provides a comprehensive architectural overview of the Layer One X (L1X) network, delving into the underlying components and implementation details that drive its functionality.

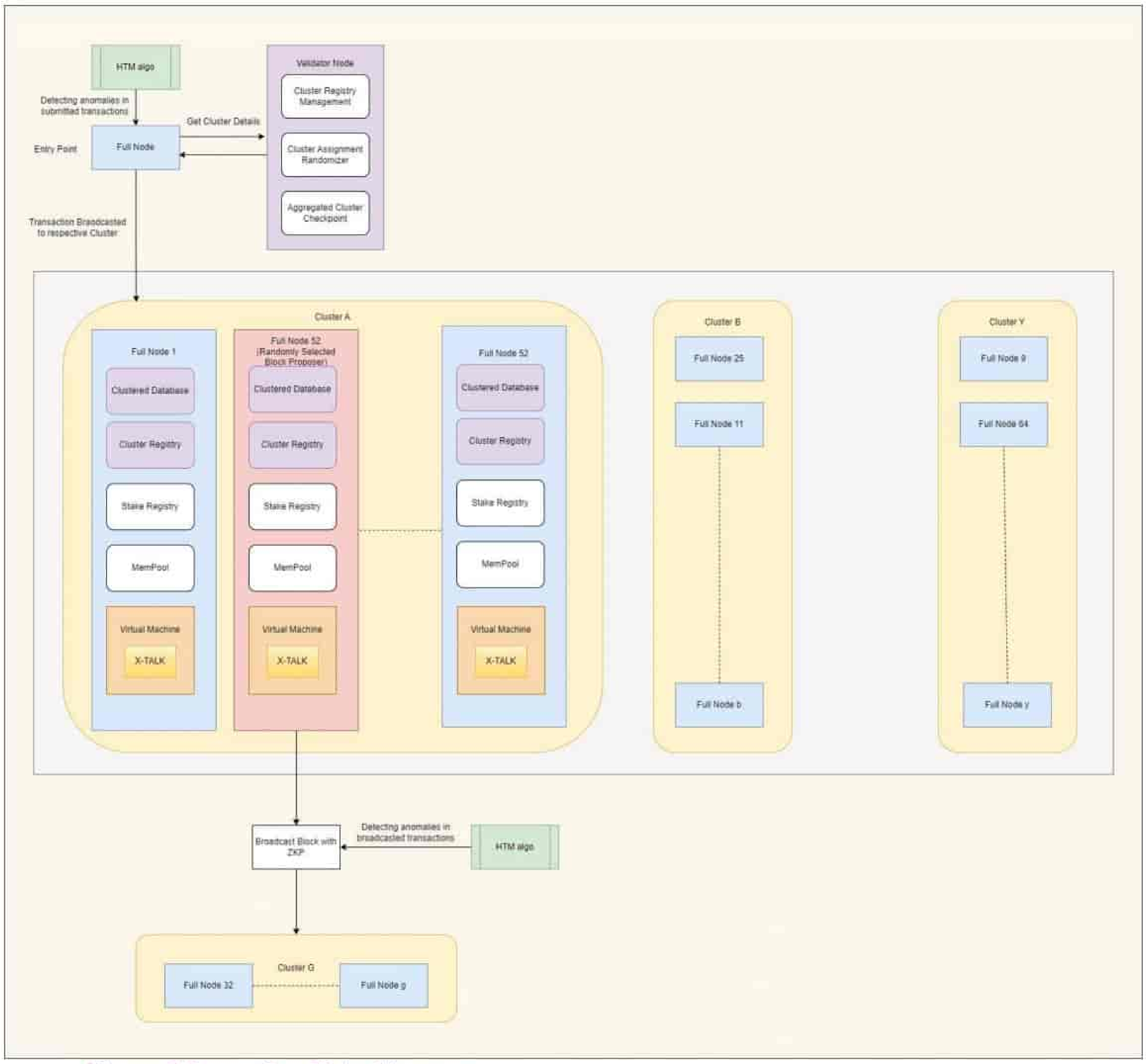


Figure 1 Layer One X Architecture

8.1 Components

The L1X network is comprised of several key components that work together to enable its interoperability, scalability, and security. These components form the foundation of the network and ensure the smooth functioning of the Layer One Smart Contract Protocol. The components of the L1X network include:

8.1.1 Virtual Machine

The L1X Virtual Machine (L1X VM) is a powerful and efficient environment specifically designed for executing smart contracts on the Layer One X blockchain platform. L1X innovates with a WASM to eBPF Compiler that improves the performance of L1X blockchain network at large scale. The L1X VM utilizes the eBPF bytecode interpreter and leverages the LLVM compiler infrastructure to optimize and compile smart contracts written in Rust. The L1X VM consists of several key components, including the L1X Smart Contract Software Development Kit (SDK), L1X System Calls, L1X-SDK-Macros, L1X SDK, on-chain collections, and the core of the L1X VM itself.

The L1X Smart Contract SDK provides a comprehensive set of tools and libraries for developers, offering a higher-level abstraction and simplifying the development process. It includes system calls for register manipulation, storage operations, accessing execution context, and other essential functionalities. The SDK also supports on-chain collections, such as LookupMap, LookupSet, and Vector, which provide efficient storage options for large-scale data management within smart contracts. The L1X SDK Macros further enhance the development experience by providing pre- defined functionality and convenient macros. The #[program] macro, for example, translates conventional Rust code into code that utilizes the primitives from the l1x-sys crate.

The L1X VM core serves as the foundation for executing eBPF bytecode in a secure and isolated environment. It performs two crucial steps: relocation and interpretation. During relocation, the VM analyzes the ELF sections of the eBPF file, including text instructions, data bytes, and function calls, and adjusts them accordingly. This process allows for efficient access to global variables and constant values. The interpretation stage utilizes a modified version of rBPF to handle non-standard extensions, enabling the execution of internal functions and ensuring compatibility with WASM.

Thus, the L1X Virtual Machine provides a powerful and optimized execution environment for smart contracts on the Layer One X blockchain. Its carefully designed components and robust infrastructure contribute to the efficiency, security, and scalability of the platform, empowering developers to build complex and innovative decentralized applications. A more detailed description of the L1X Virtual Machine can be found in VIRTUAL MACHINE.

8.1.2 X-Talk

X-TALK is a revolutionary cross-chain communication architecture introduced by L1X to enable secure exchange of smart contracts and facilitate interoperability across various blockchain networks. This architecture is designed with a decentralized approach, ensuring seamless asset and logic exchange across chains. By incorporating a blockchain-agnostic framework, X-TALK allows for efficient communication and interaction among multiple chains throughout the entire process. It leverages various components to handle tasks such as transaction identification, event validation, execution strategies, and transaction record-keeping, ensuring the integrity and security of the system. With the integration of the L1X MultiSig Smart Contract, users can securely authorize cross-chain token and asset exchange, adding an additional layer of trust and reliability to the architecture. Overall, X-TALK represents a significant advancement in cross-chain communication, enabling a scalable and interoperable blockchain ecosystem. The topic has been further detailed in X-TALK.

8.1.3 Full Validator Nodes (FVN)

Full Validator Nodes (FVN) play a crucial role in the L1X network by serving as the backbone for transaction processing and maintaining the integrity of the blockchain. These nodes are responsible for introducing transactions into the L1X network and performing essential technical validations to ensure the accuracy and validity of transaction details.

Beyond technical validations, FVNs also undertake non-technical validations to verify the logical correctness of the transactions. This step ensures that the transactions adhere to the predefined rules and conditions set by the L1X protocol. By performing these validations, FVNs contribute to the overall security and trustworthiness of the network.

In addition to transaction validation, FVNs also act as Block Proposers. Through a random selection process, FVNs are assigned the responsibility of proposing new blocks to be added to the blockchain. This randomization mechanism ensures a fair and decentralized block creation process, enhancing the network's resistance to malicious attacks and maintaining the decentralization principle.

Furthermore, FVNs actively participate in the consensus mechanism of the L1X network. They validate and confirm the transactions that are proposed by other nodes, ensuring consensus on the state of the blockchain. This collective validation process helps achieve agreement among network participants and establishes the definitive state of the blockchain.

The FVNs' contributions extend beyond transaction processing and consensus. They also play a crucial role in the overall maintenance and synchronization of the global state. By actively participating in block creation and verifying transactions, FVNs help ensure the consistency and integrity of the distributed ledger.

8.1.4 Validator Nodes

Validator nodes in the L1X network are integral to the transaction validation process, especially in the presence of mobile devices with limited computational capacity. These mobile nodes, serving as validators, contribute to the efficiency of transaction validation in L1X. The validator node assumes the responsibility of managing key components such as Cluster Register Management, Cluster Assignment Randomizer, and Aggregated Cluster Checkpoints.

In the L1X network, validator nodes play a crucial role within the Proof of X (PoX) consensus mechanism. They establish synchronization with FVNs at specific intervals, ensuring the dissemination of up-to-date information regarding nodes in the cluster and similar clusters. This information is vital for FVNs, as it allows them to determine which nodes belong to the same cluster before broadcasting blocks. By enforcing this mechanism, L1X mitigates the risk of malicious FVNs gaining control over the blockchain, enhancing the security and integrity of the network.

The validator nodes' involvement in managing the Cluster Register, which maintains information about participating nodes, ensures the proper organization and coordination of cluster assignments. Additionally, the Cluster Assignment Randomizer algorithm contributes to the fairness and randomness in the selection of block proposers, further enhancing the decentralization aspect of the network.

Another crucial responsibility of the validator nodes is the maintenance of Aggregated Cluster Checkpoints. These checkpoints serve as reference points for validating the blockchain's state and verifying the integrity of transactions. Validator nodes contribute to the creation and verification of these checkpoints, ensuring the consistency and reliability of the blockchain.

By actively participating in the transaction validation process and maintaining essential components, validator nodes play a vital role in the overall functioning and security of the L1X network. Their involvement helps expedite transaction validation, leverages the capabilities of mobile devices, and strengthens the decentralization and integrity of the blockchain.

8.1.5 Event Listener Nodes (ELN)

The Event Listener Node (ELN) stands as a cornerstone within the infrastructure of the L1X network, facilitating the seamless flow of information and transactions across diverse blockchain platforms. Operating at the heart of the X-Talk process, ELNs serve as the conduits through which events are listened to, processed, and relayed to the broader L1X network for subsequent action. This critical function ensures that disparate blockchain networks can effectively communicate and interact with one another, fostering greater interoperability and synergy within the decentralized ecosystem. The reliability and trustworthiness of ELNs are upheld through rigorous evaluation mechanisms which establish their reputation for dependable cross-chain communication. Leveraging dynamic parameter weighting and adaptability, ELNs possess the agility to evolve alongside the evolving landscape of blockchain technology, further enhancing their capacity to facilitate seamless connectivity. Moreover, ELNs that exceed predefined XTalkScore thresholds ascend to leadership positions, actively contributing to the advancement and evolution of interconnected blockchain ecosystems, thereby solidifying their pivotal role in driving innovation and progress within the industry.

8.1.6 Clustered Databases

In the L1X blockchain network, clustered databases play a vital role in ensuring high availability, fault tolerance, and scalability to accommodate a large number of users. By utilizing multiple servers connected in a cluster, L1X effectively distributes data storage, enabling the database to handle a significant volume of transactions.

The primary responsibility of clustered databases in L1X is to store user accounts and maintain a comprehensive record of all transactions. Each FVN within the cluster synchronizes and maintains user accounts, facilitating data replication and ensuring data access even in the event of a few FVNs being unavailable. This replication technique enhances the network's fault tolerance and availability, safeguarding against potential single points of failure.

One of the notable advantages of clustered databases is their ability to enable parallel processing, improving the overall performance of the L1X network. By leveraging parallelism, the network can process large datasets more efficiently, reducing latency and enhancing transaction processing speeds. This capability is crucial in meeting the scalability requirements of the blockchain, ensuring smooth and rapid transaction execution.

By serving as the core component of the L1X architecture, clustered databases contribute to the robustness, reliability, and scalability of the blockchain network. They enable efficient data storage, replication, and parallel processing, ensuring high availability, fault tolerance, and optimal performance for a growing user base.

8.1.7 Cluster Registry

The cluster registry stores and maintains essential information about the FVNs within a cluster. It serves as a repository for storing details such as the current state of each node, IP addresses, metadata, and other relevant information necessary for efficient communication and coordination among the FVNs.

One of the primary responsibilities of the cluster registry is to ensure correct data storage and processing within the L1X network. By maintaining up-to-date information about the nodes in the cluster, the registry enables efficient routing of data and transactions, facilitating seamless communication and collaboration among the participating nodes.

By providing a vital source of information, it streamlines the process of data synchronization and facilitates efficient decision-making within the network. The cluster registry enables FVNs to access the latest information about other nodes in the cluster, ensuring smooth coordination and consensus during block creation and transaction validation.

The cluster registry also enhances the fault tolerance of the L1X network. In the event of a node failure or disruption, the registry can facilitate the redistribution of tasks and responsibilities among the remaining nodes in the cluster, ensuring continuity of operations and mitigating the impact of potential single points of failure.

8.1.8 Cluster Register Management

Cluster register management ensures the seamless integration and coordination of nodes within the cluster. Its primary role is to maintain accurate and up-to-date information about the FVNs in the cluster, including their status, availability, and other relevant details. By monitoring the cluster and updating the register accordingly, cluster register management enables efficient cluster operations and contributes to the high availability and robustness of the L1X blockchain.

The responsibilities of cluster register management encompass two key aspects. Firstly, it involves continuous monitoring of the nodes within the cluster to track their status and detect any changes, such as node additions, departures, or failures. This real-time monitoring ensures that the cluster register remains synchronized with the dynamic nature of the network, allowing for accurate representation of the active nodes at any given time.

Secondly, cluster register management is responsible for updating the cluster register based on the detected changes in the cluster. When a new node joins the cluster, the management component adds its information to the register, ensuring its inclusion in the network operations. Conversely, if a node leaves the cluster or encounters any issues, the management component removes or updates its entry in the register accordingly. These updates guarantee that the cluster register accurately reflects the current state of the cluster and enables effective communication and coordination among the nodes.

Cluster register management contributes to the overall efficiency, reliability, and availability of the L1X blockchain network. By maintaining an up-to-date and accurate register of the FVNs, it enables seamless communication and collaboration within the cluster, facilitating efficient transaction processing and consensus mechanisms. Additionally, cluster register management ensures fault tolerance by promptly detecting and responding to node failures or departures, allowing the network to adapt and redistribute tasks as needed.

8.1.9 Cluster Assignment Randomizer

Cluster Assignment Randomizer ensures proper load balancing and even distribution of nodes across clusters. Its primary responsibility is to randomly assign nodes to different clusters based on factors such as the total number of nodes, total number of clusters, and the nodes already present in each cluster.

By distributing the nodes evenly, the Cluster Assignment Randomizer helps maintain optimal performance and prevents any single cluster from becoming overloaded, which could potentially degrade the network's efficiency. This load balancing mechanism ensures that the processing power and resources are effectively utilized across the clusters, enhancing the overall scalability and performance of the L1X blockchain.

One significant advantage of the Cluster Assignment Randomizer is its contribution to the security of the L1X blockchain. By randomly assigning nodes to clusters, it becomes challenging for attackers to target a specific cluster and disrupt its operations. The dynamic creation of clusters after certain intervals with a random combination of available FVNs further adds an additional layer of security. This randomization makes it difficult for attackers to predict the composition and structure of clusters, thereby bolstering the network's resilience against malicious activities.

Cluster Assignment Randomizer optimizes the network's performance, ensures fair distribution of nodes, and enhances the overall security of the L1X blockchain. By intelligently assigning nodes to clusters, it enables efficient resource utilization, load balancing, and reduces the vulnerability to targeted attacks. This mechanism reinforces the scalability, robustness, and integrity of the L1X blockchain network, contributing to its overall reliability and trustworthiness.

8.1.10 Cross Cluster Messaging

Cross Cluster Messaging facilitates seamless communication and coordination among nodes located in different clusters. Its primary purpose is to enable the exchange of data and transactions involving different Clustered Databases within the L1X network.

In the context of L1X, Cross Cluster Messaging ensures that transactions related to different Clustered Databases can be securely transmitted between clusters without compromising the integrity of the data. This is achieved by applying digital signatures to the transactions, guaranteeing their authenticity and protecting them from unauthorized modifications during transit.

To maintain optimal network performance as the volume of cross-cluster transactions increases, L1X implements efficient routing and processing algorithms. These algorithms ensure that messages are routed accurately and processed swiftly, minimizing latency and maximizing throughput. By optimizing the routing and processing of cross-cluster transactions, L1X achieves scalability without compromising the network's overall performance.

By enabling seamless communication between clusters and facilitating the exchange of transactions involving different Clustered Databases, it promotes interoperability and expands the scope of use cases that can be supported by the L1X blockchain. Moreover, Cross Cluster Messaging ensures that the L1X blockchain can handle increasing transaction volumes without sacrificing network performance or introducing significant delays. This feature is particularly crucial in scenarios where multiple clusters are involved, allowing for the parallel processing of cross-cluster transactions and minimizing the impact on overall transaction throughput.

8.1.11 Aggregated Cluster Checkpoint

The Aggregated Cluster Checkpoint ensures the synchronization and integrity of the blockchain data among numerous FVNs within a cluster. It addresses the challenge of maintaining consistency and data integrity in the presence of node failures or network issues.

In L1X, each FVN in a cluster may operate in a different state, and to ensure synchronization across all nodes, the concept of cluster checkpoints is introduced. The aggregated cluster checkpoint serves as a consolidated record of the last updated timestamp for each FVN within the cluster. It collects and combines individual checkpoints from all FVNs, creating a single, unified checkpoint that represents the latest state of the blockchain data within the cluster.

The primary responsibility of the Aggregated Cluster Checkpoint is to safeguard against data loss and maintain the consistency of the L1X blockchain in the event of node failures or network disruptions. By consolidating the individual checkpoints, it provides a reliable reference point for verifying the integrity and consistency of the blockchain data across different FVNs within the cluster.

Aggregated Cluster Checkpoint ensures the resilience and reliability of the L1X blockchain network. In the event of a FVN failure or network issue, the aggregated checkpoint serves as a recovery mechanism, allowing the network to recover and restore data integrity efficiently. By tracking and maintaining the last updated timestamp for each FVN, it enables the network to identify any discrepancies and initiate the necessary measures for data synchronization.

Furthermore, the Aggregated Cluster Checkpoint enhances the overall security of the L1X blockchain by ensuring that no single FVN can unilaterally manipulate the blockchain data within the cluster. It acts as a consensus mechanism that ensures the consistency of the blockchain state across different nodes, reducing the risk of malicious activities and ensuring the trustworthiness of the network. Thus, the Aggregated Cluster Checkpoint maintains the synchronization, integrity, and consistency of the blockchain data among multiple FVNs within a cluster.

8.1.12 Stake Registry

The Stake Registry serves as a record-keeping mechanism for staking activities and token ownership. It plays a significant role in facilitating the PoX consensus mechanism and ensuring the integrity and transparency of the staking process. The dynamic ceiling allocation mechanism in L1X adjusts the maximum amount of tokens that can be staked by participants based on various factors, such as network demand, token supply, and overall network conditions. This dynamic adjustment allows for optimal utilization of network resources while maintaining the stability and security of the blockchain.

In L1X, FVNs participate in the consensus by staking L1X tokens. The Stake Registry maintains detailed information about the staking activities performed by FVNs. It includes essential details such as the number of tokens staked, the time of staking, rewards earned on the staked tokens, and other relevant information associated with the staking FVNs. One of the primary responsibilities of the Stake Registry is to provide an accurate and up-to-date record of token ownership within the L1X blockchain. By tracking the staked tokens and their corresponding ownership details, it enables the network to establish the rightful stakeholders and their influence on the consensus process. This promotes a fair and decentralized approach to governing the blockchain network.

Moreover, the Stake Registry serves as a foundational element for implementing slashing and rewarding mechanisms in L1X. It enables the network to enforce penalties, such as slashing a portion of staked tokens, in case of malicious behavior or protocol violations. Conversely, it also facilitates the rewarding of staked nodes for their active participation and contribution to the consensus. This dynamic allocation of staked value ensures a balanced distribution of tokens across the network, preventing centralization and promoting decentralization and security.

Ensuring the security, transparency, and accountability of the staking process within the L1X blockchain network is the key responsibility of the Stake Registry. By utilizing a distributed database system replicated across all FVNs, it enhances data availability, resilience, and tamper resistance. This ensures that the ownership records and staking details remain consistent and verifiable across the network. Additionally, the Stake Registry promotes the economic incentives and governance aspects of the L1X blockchain. It encourages active participation from stakeholders by providing a mechanism to stake their tokens, earn rewards, and influence the consensus process. This fosters a healthy ecosystem where participants are incentivized to act in the best interest of the network's growth and stability.

8.1.13 Mobile Node Syncing

Mobile node syncing is the process where mobile devices synchronize with the cluster registry and maintain an updated state of the cluster. It ensures the efficiency, coordination, and robustness of the L1X blockchain ecosystem. This process facilitates the coordination among FVNs for consensus and maintains correct data storage and processing within the L1X network. By synchronizing with the cluster registry, mobile nodes gain access to the latest information about the cluster, including node states, IP addresses, metadata, and other essential details required for effective communication among FVNs.

In the L1X blockchain, the incremental syncing mechanism is employed to optimize the syncing process for mobile nodes. Instead of updating the entire cluster registry, the incremental syncing mechanism focuses on transmitting only the modified entries or updates. This approach is particularly beneficial for mobile devices with limited processing power, as it minimizes the computational burden while ensuring that mobile nodes stay updated with the latest state of the cluster.

The role of mobile node syncing extends beyond mere information retrieval and storage. It contributes to the overall stability and resilience of the L1X blockchain network. By actively participating in the syncing process, mobile nodes strengthen the network's robustness, especially in adverse scenarios where stable connections may be challenging to maintain. The involvement of a large number of mobile nodes enhances the network's ability to withstand potential disruptions and ensures a more distributed and resilient consensus mechanism.

Furthermore, mobile node syncing aligns with the core design principles of the L1X blockchain, which emphasize decentralization, efficiency, and adaptability. By allowing mobile devices to synchronize with the cluster registry and participate in the consensus process, L1X promotes inclusivity and leverages the collective power of a diverse range of devices. This approach enhances the network's efficiency, scalability, and ability to handle a large volume of transactions.

8.2 Consensus Mechanism

The L1X blockchain utilizes a two-tiered hybrid consensus mechanism, known as Proof of X (PoX), to achieve consensus among decentralized nodes regarding transaction validity and chronological order. This innovative approach incentivizes both active participation and stakeholding, striking a balance between the interests of users and stakeholders. It represents a forward-thinking and dynamic method of consensus building within the L1X network.

In the L1X PoX consensus, mobile devices with limited computation and memory capacity are empowered to actively participate in the consensus process by continuously tracking and sharing the latest state of the blockchain. This inclusive approach allows even mobile devices to contribute to the consensus, enhancing the overall decentralization and resilience of the network.

Meanwhile, FVNs within the L1X blockchain play a critical role in consensus by holding and staking L1X coins. By staking their tokens, these nodes demonstrate their commitment to the network and earn the right to validate transactions and propose new blocks. This stakeholding mechanism ensures that participants with a vested interest in the network have influence over the consensus process.

The design principles of the L1X blockchain, coupled with the PoX consensus mechanism, offer several advantages. Firstly, it incentivizes active participation from both mobile devices and FVNs, fostering a more inclusive and diverse network. Secondly, it balances the interests of users and stakeholders, promoting fairness and transparency in consensus building. Lastly, this dynamic approach allows for adaptability and scalability as the network grows, ensuring its long-term viability and effectiveness.

8.2.1 L1X Consensus Mechanism Working

Transactions in byte format, such as RLP (Recursive Length Prefix), are received by L1X FVNs through the RPC (Remote Procedure Call). The FVNs append a Receival Timestamp to the transactions before initiating transaction authentication. During this authentication process, various aspects of the transaction, including its structure, balance sufficiency, addresses, and digital signature, are meticulously verified. Once authenticated, an Authentication Timestamp is added, and the transaction is forwarded to the MemPool.

The MemPool, or Memory Pool, serves as a storage mechanism for unconfirmed transactions. In L1X, user accounts are partitioned into Clustered Databases to enhance efficiency. Each Clustered Database holds the account states for a specific subset of users. Mobile nodes maintain the Cluster Registry, which maps account records to the corresponding FVNs. By periodically synchronizing the Cluster Registry from mobile nodes, the FVNs ensure that up-to-date information is available.

Leveraging the Cluster Registry, the MemPool identifies the relevant clusters and the associated FVNs that hold the Clustered Databases containing the account records relevant to the transaction. The transaction is broadcasted to all the pertinent cluster nodes. Subsequently, these nodes perform Non-Technical Validation, an essential step involving the verification of smart contracts, adherence to predefined rules, testing edge cases, and confirming transaction accuracy. This validation process ensures the overall integrity of the blockchain.

Beyond the Clustered Databases and Cluster Registry, FVNs also maintain the Stake Registry. The Stake Registry records which FVNs have staked tokens to participate in consensus and propose blocks. This information is stored locally within each FVN. Random selection of a Block Proposer ensures true randomization outcomes while maintaining privacy by applying homomorphic encryption to node identities. The chosen Block Proposer constructs the block and applies Zero Knowledge Proof (ZKP) to enable succinct and fast verification of block. The block, along with the ZKP proof, is exclusively broadcasted to the relevant cluster nodes. FVNs validate the integrity or correctness of block with much smaller proof sizes and transmit it for further validation.

In cases where a transaction involves accounts across different Clustered Databases, Cross Cluster Messaging is employed to facilitate data sharing. The Aggregated Cluster Checkpoint plays a pivotal role in this process by maintaining the last updated timestamp for each FVN's Clustered Database. Regular synchronization between FVNs and mobile nodes ensures that the Clustered Databases and Aggregated Cluster Checkpoint remain consistently updated.

The implementation of Hierarchical Temporal Memory enables real-time detection of malicious nodes by identifying abnormal transaction patterns. This robust security mechanism is applied both during the transaction submission phase and when blocks are broadcasted across the network, actively safeguarding the L1X blockchain network against malicious activities.

The consensus mechanism of the L1X blockchain, thus, employs multiple stages, including transaction authentication, Non-Technical Validation, block proposal, and data synchronization. These stages, combined with techniques like homomorphic encryption and Hierarchical Temporal Memory, ensure the integrity, security, and privacy of the L1X blockchain network, providing a robust and trustworthy decentralized ecosystem.

8.2.2 Block Proposer Selection: True Randomization and Homomorphic Encryption

In the L1X blockchain, the process of selecting a block proposer plays a critical role in maintaining the security and fairness of the network. The block proposer is responsible for creating and adding new blocks to the blockchain, contributing to the overall consensus mechanism. To ensure the integrity and robustness of the blockchain, a randomized algorithm is employed for the selection of the block proposer.

8.2.2.1 PoX Consensus Metrics

In the quest for achieving consensus and decentralization, the participation of network participants holds the utmost importance. Trust in the L1X blockchain network and the availability of trusted participants are crucial factors that drive the success and sustainability of the blockchain ecosystem. To address this, a mechanism that efficiently considers both old and new participants in the PoX consensus is proposed. By incorporating metrics such as StakeScore, KinScore, and XScore, we aim to evaluate participants' stake holdings, active involvement, and adherence to security measures. These metrics play a vital role in achieving consensus, promoting decentralization, and ensuring the integrity and security of the network. In the subsequent sections, we will delve into the significance of each metric, highlighting their importance in maintaining a robust and reliable PoX consensus mechanism.

- **StakeScore**: This is a measure of a node's commitment to the network, based on the amount of L1X coins they have staked, the length of time they have been staking, and the length of time they have agreed to lock up their coins. A high StakeScore indicates that a node is more likely to behave honestly, as they have more to lose if they are caught cheating.
- **KinScore**: This is a measure of a node's reliability and trustworthiness, based on their uptime, participation history, response time, and security measures. A high KinScore indicates that a node is more likely to be able to participate in the consensus process reliably and without disruption.
- **XScore**: This is a combined measure of StakeScore and KinScore, which is used to determine which nodes are eligible to participate in the PoX consensus. A higher XScore indicates that a node is more likely to be a reliable and trustworthy participant in the consensus process.

The PoX consensus metrics are designed to achieve consensus and decentralization by rewarding nodes that are committed to the network, reliable, and trustworthy. By ensuring that the network is sufficiently decentralized, the PoX consensus helps to protect the network from attack and ensure that it is fair and transparent.

8.2.2.2 PoX Consensus Process

XScore, StakeScore, and KinScore are integral components of the sophisticated block proposer selection process in the L1X blockchain ecosystem.

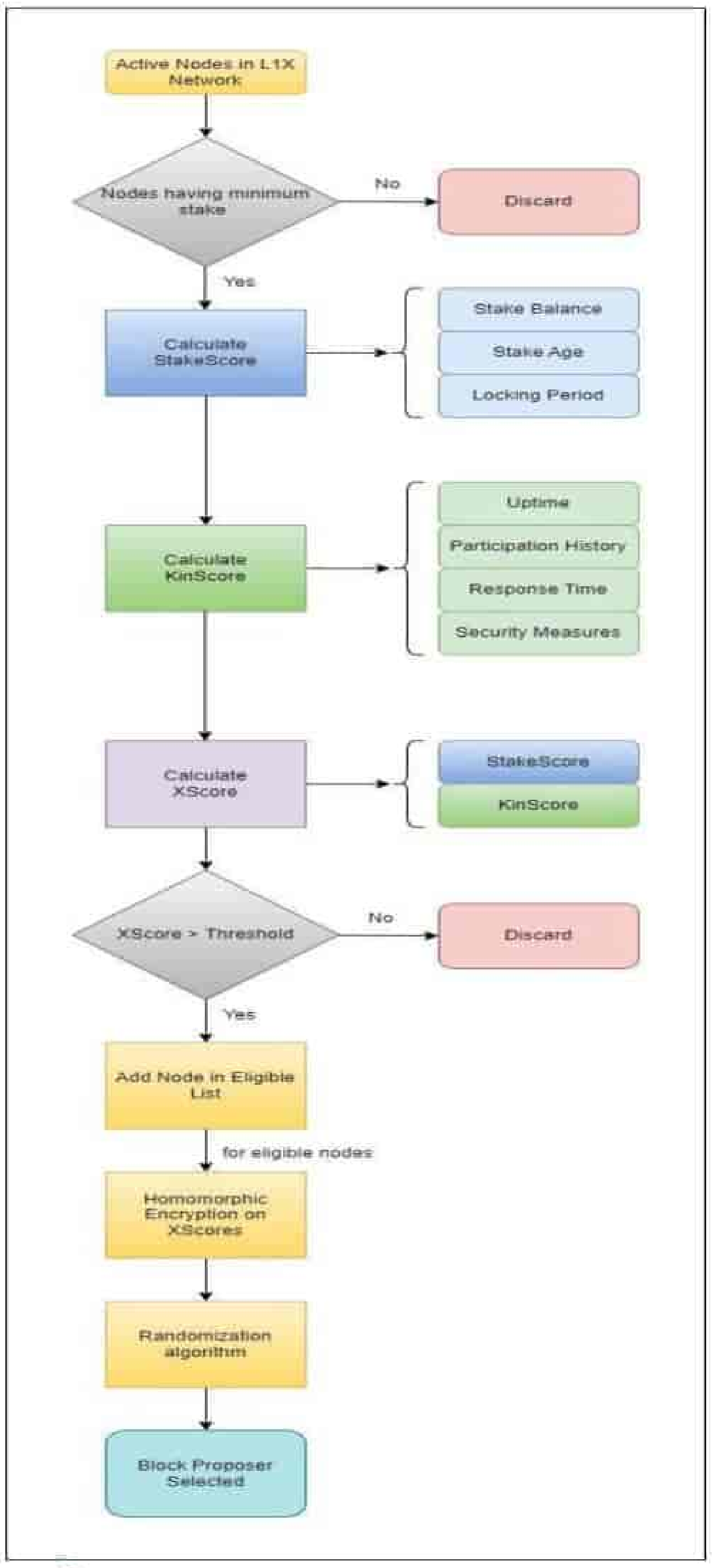


Figure 2 PoX Consensus Process

To calculate XScore, the process considers all nodes that have staked a minimum balance and are actively available within the network. These nodes undergo evaluation to determine their XScore, which subsequently plays a significant role in determining their eligibility for the next epoch of the consensus process. Nodes with an XScore exceeding the defined XScore Threshold (that varies based on network dynamics) are deemed eligible for participation.

To ensure data privacy and security, homomorphic encryption is applied to XScore. This cryptographic technique enables computations to be performed on encrypted data without compromising its confidentiality. By leveraging homomorphic encryption, the privacy of

XScore calculations is preserved, allowing for a secure evaluation process.

Furthermore, a randomized algorithm is applied to the homomorphically encrypted XScores. This algorithm introduces an element of randomness in the selection of the block proposer. By employing a randomized approach, the consensus protocol mitigates potential biases and ensures a fair and decentralized block proposer selection process.

Overall, the intricate interplay between XScore, StakeScore, KinScore, homomorphic encryption, and randomized algorithms forms a robust framework that enables accurate evaluation, privacy preservation, and fair selection within the L1X blockchain network. The randomness injected into the selection process prevents any undue advantage or bias towards specific nodes, fostering a level playing field for all participants. Furthermore, the use of homomorphic encryption ensures that the privacy of the nodes is preserved, enhancing the overall security posture of the L1X blockchain.

8.2.3 PoX Mathematical Model

Multi-objective optimization is a fundamental concept in mathematical optimization that involves optimizing multiple objective functions while considering a set of constraints. In the context of blockchain consensus, the StakeScore, KinScore, and XScore calculation problem can be viewed as a multi-objective optimization problem. This problem aims to simultaneously optimize groups of conflicting objectives in order to achieve an optimal solution.

Stakescore Multi-Objective Model

Let U_{SS} represent the Universal set for Stake Score. A general Stake Score metric combines Stake Balance, Stake Age and Locking Period. To consider the relative importance of objectives, the selection model is formulated by the weighted arithmetic mean operator.

Stake Balance Membership Function

Let *sb* represent the fuzzy set of Stake Balance. Since, a large Stake Balance indicates more trust in the network and willingness to invest, its membership function for a node *i* can be defined as:

$U_{SS}(sb_i) = sb_i - sb_{min} / sb_{max} - sb_{min}$ if $sb_{min} <= sb_i <= sb_{max}$

$U_{SS}(sb_i) = 0,$ if $sb_{min} > sb_i$

$U_{SS}(sb_i)=1,$ if $sb_{max} < sb_i$

A value of Stake Balance sb_i close to maximum stake balance sb_{max} indicates that the node is committed to network success and hence, can be considered for consensus process.

Stake Age Membership Function

Let fuzzy set of Stake Age be represented by *sa*. Since, a longer Stake Age indicates commitment to the network, its membership function for a node *i* can be defined as:

$U_{SS}(sa_i) = sa_i - sa_{min} / sa_{max} - sa_{min}$ if $sa_{min} <= sa_i <= sa_{max}$

$U_{SS}(sa_i) = 0,$ if $sa_{min} > sa_i$

$U_{SS}(sa_i) = 1,$ if $sa_{max} < sa_i$

A value of Stake Age sa_i close to maximum stake age sa_{max} indicates that the node has trust in the network and is therefore, committed to network by investing tokens since a longer period of time and hence, is a suitable candidate for consensus process.

Locking Period Membership Function

Let *lp* represent the fuzzy set of Locking Period. A longer Locking Period indicates sense of responsibility and dedication. Therefore, its membership function for a node *i* can be defined as:

$U_{SS}(lp_i) = lp_i - lp_{min} / lp_{max} - lp_{min}$ if $lp_{min} <= lp_i <= lp_{max}$

$U_{SS}(lp_i) = 0,$ if $lp_{min} > lp_i$

$U_{SS}(lp_i) = 1,$ if $lp_{max} < lp_i$

A value of Locking Period lp_i close to maximum locking period lp_{max} indicates that the node doesn't have opportunistic behavior and is dedicated as it is willing to stake coins for a considerably longer period. It is a good candidate for consensus.

StakeScore Weighted Arithmetic Model

To consider the relative importance of objectives, the selection model is formulated by the weighted arithmetic mean operator i.e., Maximize U*WT

$$w_{sb} + w_{sa} + w_{lp} = 1, \text{ for } w_{sb}, w_{sa}, w_{lp} \in [0,1]$$

where W_{SS} is the weight vector containing w_{sb},

w_{sa}, w_{lp} w_{sb} means the weight of stake balance,

w_{sa} means the weight of stake age,

w_{lp} means the weight of locking period.

Thus, weight factors will help the model to be flexible and can be used to evaluate significance of each parameter to maximize StakeScore. The degree of overall satisfaction is the sum of all membership values. The fuzzy decision may be considered as the choice that satisfies all of the objectives.

$$\lambda_{SS}(i) = w_{sb}*U_{SS}(sb_i) + w_{sa}*U_{SS}(sa_i) + w_{lp}*U_{SS}(lp_i)$$

Kinscore Multi-Objective Model

Let U_{KS} represent the Universal set for KinScore. A general KinScore metric combines Uptime, Active Participation History, Response Time and Security Measure. To consider the relative importance of objectives, the selection model is formulated by the weighted arithmetic mean operator.

Uptime Membership Function

Let *ut* represent the fuzzy set of Uptime for a node. A node *i* with larger uptime contributes to maintaining stable network, therefore its membership function can be defined as:

$U_{KS}(ut_i) = ut_i - ut_{min} / ut_{max} - ut_{min}$ if $ut_{min} <= ut_i <= ut_{max}$

$U_{KS}(ut_i) = 0,$ if $ut_{min} > ut_i$

$U_{KS}(ut_i) = 1,$ if $ut_{max} < ut_i$

A value of Uptime ut_i close to maximum uptime ut_{max} indicates that the node is committed to network success and hence, can be considered for consensus process.

Active Participation History Membership Function

Let *ph* represent the fuzzy set of active participation history. A node *i* actively involved in transactions validation and block proposal is a significant contributor to the network consensus. Its membership function can be defined as:

$U_{KS}(ph_i) = ph_i - ph_{min} / ph_{max} - ph_{min}$ if $ph_{min} <= ph_i <= ph_{max}$

$U_{KS}(ph_i) = 0,$ if $ph_{min} > ph_i$

$U_{KS}(ph_i) =1,$ if $ph_{max} < ph_i$

A value of phi close to maximum active participation history ph_{max} indicates that the node is actively engaged in efficient and reliable PoX consensus, making it a good candidate for further consensus process.

Response Time Membership Function

Let rt represent the fuzzy set of Response Time for a node. A node i with smaller response time is quickly validating the transactions and therefore its membership function can be defined as: $U_{KS}(rt_i) = rt_{max} - rt_i / rt_{max} - rt_{min}$ if $rt_{min} <= rt_i <= rt_{max}$

$U_{KS}(rt_i) = 1,$ if $rt_{min} > rt_i$

$U_{KS}(rt_i) = 0,$ if $rt_{max} < rt_i$

A value of Response Time rti close to minimum response time $rtmin$ indicates that the node has a low latency and is making the L1X network more responsive and faster. Such a node is suitable for consensus process.

Security Measures Membership Function

Let sm represent the fuzzy set of Security Measure for a node. A node i with more security measures is making the L1X network secure and reliable, therefore its membership function can be defined as:

$U_{KS}(sm_i) = sm_i - sm_{min} / sm_{max} - sm_{min}$ if $sm_{min} <= sm_i <= sm_{max}$

$U_{KS}(sm_i) = 0,$ if $sm_{min} > sm_i$

$U_{KS}(sm_i) = 1,$ if $sm_{max} < sm_i$

A value of Security Measure sm_i close to maximum Security Measure sm_{max} indicates that the node is genuine and concerned about the overall security of the L1X network. It is, therefore, a suitable candidate for consensus process.

KinScore Weighted Arithmetic Model

To consider the relative importance of objectives, the selection model is formulated by the weighted arithmetic mean operator. Maximize U*WT

$W_{KS} = w_{ut} + w_{ph} + w_{rt} + w_{sm} = 1$, for $w_{ut}, w_{ph}, w_{rt}, w_{sm} \in [0,1]$

where W_{KS} is the weight vector containing $w_{ut}, w_{ph}, w_{rt}, w_{sm}$

w_{ut} means the weight of uptime,

w_{ph} means the weight of active participation history,

w_{rt} means the weight of response time,

w_{sm} means the weight of security measures.

Thus, weight factors will help the model to be flexible and can be used to evaluate significance of each parameter to maximize KinScore. The degree of overall satisfaction is the sum of all membership values. The fuzzy decision may be considered as the choice that satisfies all of the objectives.

$$\lambda_{KS}(i) = w_{ut}*U_{KS}(ut_i) + w_{ph}*U_{KS}(ph_i) + w_{rt}*U_{KS}(rt_i) + w_{sm}*U_{KS}(sm_i)$$

XScore Weighted Arithmetic Model

XScore is a combination of StakeScore and KinScore. Weighted Arithmetic Model is also employed for XScore so that flexibility in the PoX consensus can be easily provided. This feature will help to customize consensus as per business requirements.

$$W_{XS} = W_{SS} + W_{KS} = 1, \text{ for } W_{SS}, W_{KS} \in [0,1]$$

where WXS is the weight vector containing wss and wks wss means the weight of StakeScore,

wks means the weight of KinScore.

The fuzzy decision may be considered as the choice that satisfies all of the objectives.

$$\lambda_{XS}(i) = W_{SS}*\lambda_{SS} + W_{KS}*\lambda_{KS}$$

where $\lambda_{XS}(i)$ is the XScore of the node *i*.

Based on the XScore Threshold (T_{XS}), all the nodes having XScore greater than the threshold are suitable candidates for the consensus process in the next epoch. For nodes n in the network, this can be represented as:

$$\forall\, i,\ 1 \le i \le n:$$

if $\lambda_{XS}(i) > T_{XS}$, then Node *i* is a suitable candidate for the consensus process in the next epoch.

8.2.4 Zero Knowledge Proof

The Block Proposer applies Zero Knowledge Proof (ZKP) on the block created and then broadcasts it in the network. Applying ZKP to the block before broadcasting it to the network serves multiple purposes. It ensures the integrity and correctness of the block by verifying that it adheres to the predefined rules and structure of the blockchain. This prevents the inclusion of invalid or malicious transactions in the block, enhancing the overall security of the network.

The application of ZKP in the L1X blockchain also leads to improved network efficiency. Since the block undergoes ZKP before broadcasting, the size of the block is significantly reduced. This reduction in block size enables faster transmission and lower latency within the network. Consequently, the overall performance and scalability of the L1X blockchain are enhanced, allowing for a smoother and more efficient consensus process.

8.2.5 Malicious Node Detection – Hierarchical Temporal Memory

Hierarchical Temporal Memory (HTM) is a powerful machine learning algorithm renowned for its ability to identify anomalies in temporal patterns. In the context of blockchain technology, where maintaining security is of utmost importance, HTM serves as a valuable tool for detecting and mitigating malicious activities within the network.

Malicious nodes in a blockchain network often exhibit abnormal behavior over time, such as initiating fraudulent transactions or generating an unusually high volume of transactions. By leveraging the capabilities of HTM, these behavioral patterns can be analyzed to identify deviations from the norm. HTM excels at processing large volumes of transactions in real time, making it an ideal solution for monitoring the activities of all nodes within the network.

One of the significant advantages of HTM is its ability to learn and adapt to changing behavioral patterns, including newly emerging attack techniques. As the blockchain landscape evolves, attackers continuously devise new strategies to exploit vulnerabilities. HTM's adaptability enables it to recognize and respond to these evolving threats, thereby bolstering the security and resilience of the L1X blockchain.

The integration of HTM within the L1X blockchain plays a crucial role in identifying malicious nodes by promptly detecting anomalies in their behavior. By applying HTM at two key stages—the transaction entry point and the block broadcasting stage—L1X ensures comprehensive coverage and robust security throughout the blockchain network.

At the transaction entry point, HTM analyzes incoming transactions, allowing for the immediate detection of suspicious or fraudulent activities. This early detection capability prevents potentially harmful transactions from entering the network, mitigating the risk of security breaches.

Similarly, HTM is employed when blocks are broadcasted to nodes within the network. By analyzing the temporal patterns of block propagation and processing, HTM can identify any irregularities or deviations from expected behaviors. This enables the system to take appropriate measures to address potential threats and maintain the integrity of the blockchain. As a result, HTM serves as a critical component in upholding the security and trustworthiness of the L1X blockchain ecosystem.

8.2.6 Benefits

The PoX consensus mechanism offers a multitude of advantages in achieving genuine decentralization while ensuring a secure and scalable blockchain infrastructure capable of high throughput as listed below:

1. **Efficient transaction validation**: The PoX consensus mechanism in L1X reduces energy consumption by eliminating the need for high computational power, resulting in a more sustainable blockchain network.

2. **Scalability through active microdevice participation**: With PoX, the network's scalability is maintained even as the number of validators increases. The active engagement of microdevices ensures consistent network performance, accommodating a growing user base.

3. **Pure decentralization**: PoX enables a purely decentralized network by actively involving microdevices in the consensus process. The random selection of block proposer ensures fairness and prevents any single or group of nodes from gaining unfair advantages.

4. **Enhanced security and privacy through ZKP**: L1X applies ZKP on blocks before broadcasting them, ensuring the correctness of blocks while preserving sensitive transaction information. ZKP enhances security by reducing the block size, enabling faster transmission, and minimizing the risk of data exposure.

5. **Enhanced privacy through Homomorphic Encryption:** Utilizing homomorphic encryption to protect the privacy of FVN identities during consensus brings significant advantages to the blockchain ecosystem, as it ensures that the selection process remains secure and unbiased while maintaining the confidentiality of node participants. By encrypting identities and applying a randomization algorithm, the blockchain network enhances privacy and prevents malicious entities from manipulating the selection process.

6. **Proactive detection of malicious nodes**: The utilization of HTM in L1X allows for the real- time identification of anomalies and abnormal transaction patterns associated with malicious nodes. HTM's ability to handle large transaction volumes and adapt to emerging attack patterns strengthens network security.

7. **Protection against targeted attacks**: Random selection of block proposers makes it difficult for attackers to target specific nodes. This additional layer of security ensures the integrity and resilience of the L1X blockchain network.

8. **Robust network integrity**: By combining transaction authentication, Non-Technical Validation, data synchronization, and security measures, the PoX consensus mechanism in L1X ensures the overall integrity and reliability of the blockchain network.

These benefits collectively contribute to a more efficient, scalable, secure, and decentralized blockchain ecosystem in L1X. By reducing energy consumption, promoting scalability, preserving privacy, and detecting and preventing malicious activities, the PoX consensus mechanism enables the L1X blockchain to meet the demands of a growing network while maintaining a high level of security and integrity.

9. L1X Transaction Lifecycle

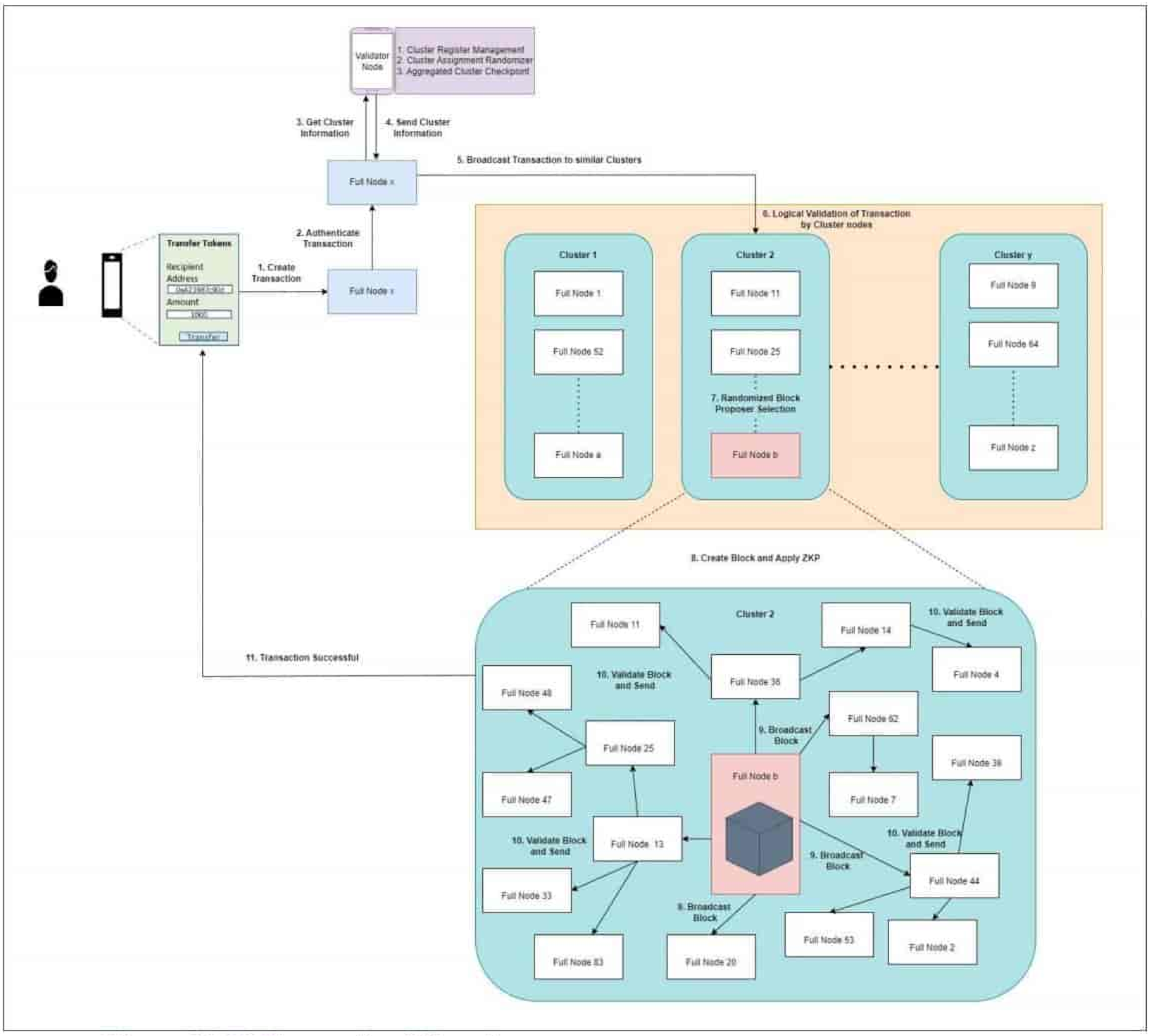


Figure 3 L1X Transaction Lifecycle

1. User digitally signs the transaction using ECDSA and submits the transaction to the L1X network.
2. The FVN in the L1X network verifies the authenticity of the transaction and the authenticated transaction is added to the MemPool.
3. The FVN queries validator nodes i.e., mobile devices to get cluster information.
4. The validator node sends the details of similar clusters and the FVNs in the cluster.
5. Now, the FVN broadcasts the transaction to the cluster nodes.
6. All the cluster nodes perform logical validation for the transaction.
7. At this stage, the Block Proposer is randomly selected by the PoX consensus mechanism.

8. The Block Proposer creates a block containing multiple validated transactions and applies Zero Knowledge Proof on the block.
9. The Block Proposer broadcasts the block to the connected nodes in similar clusters.
10. The FVNs in the network perform block validation and send it to peers.
11. The user is intimated about the successful transaction completion.

10. Transaction Types

In the L1X blockchain, transaction types represent the innovative and versatile ways users can interact with digital assets, including tokens and Non-Fungible Tokens (NFTs), across diverse blockchain networks. These transaction types enable cross-chain functionality, allowing assets to seamlessly move, be utilized, and integrated with different ecosystems. From token swaps to NFT transactions and beyond, L1X transaction types provide users with powerful tools to enhance liquidity, utility, and engagement within the blockchain space. Let us have a quick overview of different transaction types on the L1X blockchain network.

10.1 Native Token Transfer

A Native Token Transfer is a transaction type that involves transferring the native cryptocurrency L1X coin or token of the L1X blockchain from one account to another.

10.1.1 Purpose and Requirement

The primary purpose of executing an L1X Native Token Transfer is to move units of the native cryptocurrency between accounts within the L1X blockchain network. When such a transfer occurs, it's essential to ensure that both the sender and the receiver have active accounts on the L1X blockchain. The creation of an account on-chain for the sender should be executed first, before the transfer, to establish the sender as an active participant in the L1X blockchain network. This is crucial because the L1X blockchain needs to keep track of account balances and ownership of native tokens accurately.

10.1.2 Benefits

1. **Account Activation:** By requiring the creation of an account for the sender on-chain before the transfer, the system ensures that both the sender and receiver have valid and active accounts on the L1X blockchain network. This helps maintain accurate accounting and ownership of tokens.
2. **Security and Accountability:** Native Token Transfers are recorded on the L1X blockchain's public ledger. By verifying the sender's identity and account status,

the L1X blockchain enhances security and accountability, reducing the likelihood of fraudulent or unauthorized transactions.

3. **Consistency:** Requiring an on-chain account creation before a transfer ensures consistency within the L1X blockchain's state. It avoids the scenario where transfers occur without corresponding active accounts, which could lead to inconsistencies in token ownership records.
4. **Blockchain Health:** Implementing such a mechanism contributes to the overall health and reliability of the L1X blockchain network. It helps prevent situations where inactive or improperly created accounts might disrupt the functioning of the network.
5. **User Experience:** By having clear guidelines for sender account activation before transfers, the user experience is improved. Users will be aware of the steps they need to follow, reducing confusion and the likelihood of failed transactions due to account issues.

In summary, Native Token Transfer involves moving the native cryptocurrency i.e. L1X coins of the L1X blockchain between accounts. Requiring sender account creation on-chain before the transfer ensures the sender's status as an active participant in the L1X blockchain network and provides benefits in terms of security, consistency, and user experience. This practice contributes to maintaining the integrity of the L1X blockchain and its token ecosystem.

10.2 Smart Contract Deployment

The term "SmartContractDeployment" refers to a transaction type in the L1X blockchain ecosystem where smart contracts are created and deployed onto the L1X blockchain. Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They automatically execute predefined actions when certain conditions are met.

10.2.1 Supported Contract Types

Smart Contract Deployment transaction type supports several contract types:

1. L1XVM
2. EVM
3. XTALK

10.2.2 Purpose and Requirement

The primary purpose of transaction-type Smart Contract Deployment is to introduce new functionality or business logic onto the L1X blockchain network through smart contract creation and deployment. Smart contracts enable automation and programmability of various processes, allowing parties to interact and transact without intermediaries. Smart Contract Deployments are required whenever a new smart contract needs to be added to the blockchain to facilitate specific operations, agreements, or decentralized applications (DApps).

10.2.3 Benefits

1. **Decentralization:** Smart contracts enable decentralized execution of agreements and processes. This reduces the need for intermediaries, enhancing trust and transparency among participants.
2. **Automation:** Smart contracts automatically execute predefined actions when conditions are met. This reduces manual intervention, streamlines processes, and reduces the likelihood of errors.
3. **Immutability:** Once a smart contract is deployed, its code is usually immutable. This means that the terms and logic of the contract cannot be altered without creating a new version, ensuring the integrity of agreements.
4. **Transparency:** The code and logic of smart contracts are visible on the L1X blockchain, providing transparency to all parties involved.
5. **Efficiency:** By automating processes, smart contracts can improve efficiency and reduce the time and resources needed to execute agreements.
6. **Global Accessibility:** Smart contracts on the L1X blockchain are accessible from anywhere, allowing participants from different parts of the world to engage in transactions without geographical barriers.

7. **Security:** Smart contracts are executed in a tamper-resistant and secure environment, reducing the risk of fraud or manipulation.

Thus, a Smart Contract Deployment transaction type facilitates the creation and deployment of smart contracts onto the L1X blockchain network. It introduces automation, transparency, and efficiency to various processes, offering benefits such as decentralization, immutability, and security. Deploying new smart contracts enables the L1X blockchain ecosystem to evolve and provide solutions for a wide range of use cases.

10.3 Smart Contract Initiation

A Smart Contract Initialization Transaction (SmartContractInit) is a specific type of transaction that is executed to initialize or configure a smart contract after it has been deployed onto the L1X blockchain. This transaction is separate from the initial deployment transaction and is used to set initial parameters, values, or states within the smart contract.

10.3.1 Purpose and Requirement

The primary purpose of executing a SmartContractInit transaction is to provide the smart contract with the necessary initial data or configuration to start functioning as intended. Some smart contracts might require specific parameters or states to be set before they can be used effectively. The initiation process can involve setting default values, assigning roles or permissions, configuring variables, or performing any necessary setup for the smart contract to begin its operations.

10.3.2 Benefits

1. **Customization:** SmartContractInit transactions allow for the customization of smart contracts based on specific use cases and requirements. Each deployment of the contract can be tailored to the needs of the application or business logic.

2. **Flexibility:** By separating the deployment of a smart contract from its initialization, developers and users have the flexibility to deploy a contract with basic functionality and later customize it without redeploying the entire contract.
3. **Upgradability:** Separating the contract deployment and initialization phases can facilitate future upgrades. If changes are needed in the initialization process, they can be made without affecting the contract's core logic.
4. **Efficiency:** Initialization transactions are more focused and can be executed quickly since they only involve setting up initial parameters. This can save resources and reduce execution time compared to deploying an entirely new contract.
5. **Modularity:** The modular approach of separating deployment and initialization allows for a cleaner code structure and easier maintenance of smart contracts.
6. **Testing and Development:** Developers can deploy a basic version of the smart contract for testing and debugging purposes and later initialize it with more complex parameters once testing is successful.
7. **Reduced Costs:** By only deploying the contract once and initializing it as needed, users can save on deployment fees and associated costs.

In short, a SmartContractInit transaction involves initializing a previously deployed smart contract with specific parameters, values, or states. It offers customization, flexibility, and efficiency benefits, allowing for tailored smart contract deployments and facilitating future upgrades without the need to redeploy the entire contract.

10.4 Smart Contract Function Call

A **SmartContractFunctionCall** transaction refers to a transaction in the L1X blockchain network that involves interacting with a deployed smart contract by invoking one of its functions. It is an action taken by a user or application to execute a specific function that is defined within a deployed smart contract on the L1X blockchain. Smart contracts consist of functions that perform predefined actions when called with appropriate inputs. A SmartContractFunctionCall transaction triggers the execution of one of these functions and can include input parameters that determine the behaviour of the function.

10.4.1 Purpose and Requirement

The primary purpose of a **SmartContractFunctionCall** transaction is to interact with, and utilize the functionality provided by a smart contract in the L1X blockchain network. Smart contracts are designed to automate and execute single or multiple actions (in one transaction) based on predefined logic. By invoking specific functions within a smart contract, users can trigger these actions and access the features or services that the contract provides.

10.4.2 Benefits

1. **Decentralized Functionality:** Smart contracts enable the execution of predefined functions in a decentralized and trustless manner. Users can utilize the services provided by the contract without relying on a central authority.
2. **Automation:** SmartContractFunctionCall transactions automate processes that are encoded within the smart contract's functions. This can streamline various interactions, such as transferring tokens, updating data, or performing calculations.
3. **Transparency:** All function calls and their outcomes are recorded on the L1X blockchain's public ledger, ensuring transparency and auditability of actions taken.
4. **Immutability:** The outcomes of function calls are immutable once recorded on the L1X blockchain. This ensures that actions and results cannot be altered retrospectively.
5. **Programmability:** SmartContractFunctionCall transactions allow for programmable interactions with the L1X blockchain. Developers can create applications that interact with smart contracts to build complex decentralized applications (DApps).
6. **Customization:** Smart contracts can offer various functions, allowing users to customize their interactions based on their specific needs.
7. **Reduced Intermediaries:** Users can directly interact with smart contracts without intermediaries, reducing costs and enhancing efficiency.
8. **Global Accessibility:** SmartContractFunctionCall transactions can be executed from anywhere in the world, providing access to the L1X blockchain-based services regardless of geographical location.

In nutshell, a **SmartContractFunctionCall** transaction enables users and applications to interact with the functionality offered by a deployed smart contract. By invoking specific functions, users can automate actions, access services, and utilize decentralized features provided by the contract. This interaction is a fundamental component of the L1X blockchain based applications and decentralized ecosystems.

10.5 Create Staking Pool Transaction

Create Staking Pool refers to the establishment of the staking pool in the L1X blockchain network. A **Create Staking Pool Transaction** is a transaction executed on the L1X blockchain network to create a staking pool. A staking pool is a collective investment mechanism in which participants pool their cryptocurrency holdings to collectively participate in the staking process. Staking involves actively participating in the PoX consensus mechanism of the L1X blockchain network by holding and "staking" a certain amount of the L1X blockchain network's native cryptocurrency i.e., L1X coins.

10.5.1 Purpose and Requirement

The primary purpose of executing a **Create Staking Pool Transaction** is to establish a staking pool that allows multiple participants to contribute their tokens and collectively participate in staking in the L1X blockchain network. Staking pools are particularly useful for individuals who may not have enough tokens to independently stake and participate in the L1X blockchain network's consensus mechanism.

10.5.2 Benefits

1. **Increased Accessibility:** Staking pools enable users with smaller amounts of cryptocurrency to participate in staking activities that would otherwise require a larger stake.
2. **Diversification:** Participants in a staking pool can benefit from a more diversified staking portfolio since the pool's collective stake is likely to be spread across multiple participants.

3. **Consolidated Rewards:** Rewards earned from staking activities are distributed among pool participants based on their contributed stake. This consolidation simplifies the reward distribution process.
4. **Reduced Technical Knowledge:** Staking pool participants don't need to possess extensive technical knowledge about the L1X staking and blockchain protocols, as the pool's operator manages the technical aspects.
5. **Cost-Efficiency:** Operating a staking node often requires continuous uptime and technical maintenance. In a staking pool, these responsibilities are managed by the pool operator, saving individual participants time and resources.
6. **Community Engagement:** Staking pools can foster a sense of community among participants who share an interest in the success of the L1X blockchain network.
7. **Incentives:** Staking pools may offer incentives or bonuses to attract participants and encourage contributions.
8. **Economies of Scale:** Larger staking pools can achieve economies of scale, potentially leading to more efficient staking processes and increased rewards.
9. **Network Security:** Staking pools contribute to the security of the L1X blockchain network by collectively participating in the PoX consensus mechanism, helping maintain network stability.

In summary, a **Create Staking Pool Transaction** involves establishing a staking pool on the L1X blockchain network. Staking pools enable smaller token holders to participate in staking, benefit from diversification, and share rewards while delegating technical responsibilities to the pool operator. Staking pools contribute to the network's security and foster a more inclusive staking ecosystem.

10.6 Stake Transaction

A **Stake Transaction** is a transaction executed on the L1X blockchain network to participate in the staking process. Staking involves locking up a certain amount of L1X coins in the L1X wallet or smart contract to actively contribute to the L1X blockchain network's operations, security, and PoX consensus mechanism.

10.6.1 Purpose and Requirement

The primary purpose of executing a **Stake Transaction** is to actively participate in the PoX consensus mechanism of the L1X blockchain network. Staking helps secure the network, validate transactions, and maintain network integrity. It's required as part of PoX consensus mechanism where stakers are rewarded for their participation.

10.6.2 Benefits

1. **Network Security:** Staking transactions contribute to the security and decentralization of the L1X blockchain network by ensuring that tokens are locked up as collateral for validating transactions.
2. **Incentives:** Stakers are often rewarded with additional tokens for participating in the staking process, incentivizing token holders to actively engage in the L1X blockchain network operations.
3. **Passive Income:** Staking can provide participants with a source of passive income, as they earn rewards for their participation in securing the L1X network.
4. **Reduced Inflation:** Staking mechanisms can help reduce the overall inflation rate of the L1X coin by encouraging holders to lock up their tokens rather than selling them.
5. **Token Value:** Staking can influence the demand and value of the L1X coins, as tokens are locked up, potentially reducing the circulating supply and impacting supply-demand dynamics.
6. **Network Consensus:** Staking transactions play a role in the L1X blockchain network's PoX consensus mechanism, ensuring that transactions are verified and validated by participants who have a stake in the L1X network's success.
7. **Decentralization:** Staking encourages a decentralized distribution of network validation, as participants are geographically diverse and hold varying amounts of tokens.
8. **Economic Participation:** Staking allows individuals to actively participate in the economic aspect of the L1X blockchain network beyond just holding tokens.

In summary, a **Stake Transaction** involves actively participating in the staking process of the blockchain network. It contributes to the L1X network security, decentralization, and consensus while offering incentives in the form of rewards and potential passive income for participants. Staking is an integral part of the L1X blockchain ecosystem and aligns the interests of token holders with the network's success.

10.7 Unstake Transaction

An **Unstake Transaction** is a transaction executed on the L1X blockchain network to withdraw tokens that were previously staked. Staking involves locking up a certain amount of L1X coins in order to participate in the L1X network's operations, security, and PoX consensus mechanism. Unstaking, on the other hand, is the process of unlocking and withdrawing these previously staked tokens.

10.7.1 Purpose and Requirement

The primary purpose of executing an **Unstake Transaction** is to regain access to tokens that were previously locked up for staking. Unstaking may be required for various reasons, such as the need for liquidity, changes in investment strategy, or the desire to move tokens to a different network. Additionally, unstaking might be necessary if a participant wishes to participate in a different staking pool or adjust their staking parameters.

10.7.2 Benefits

1. **Liquidity:** Unstaking provides participants with access to their staked tokens, allowing them to use these tokens for trading, investing, or other purposes requiring liquidity.
2. **Flexibility:** The ability to unstake tokens offers flexibility for token holders to adjust their strategies based on changing market conditions or personal preferences.
3. **Portfolio Management:** Participants can manage their token holdings more effectively by unstaking tokens and reallocating them to other investments or opportunities.
4. **Participation Changes:** Users might want to unstake tokens if they wish to switch staking pools, move tokens to a different network, or change their staking strategy.

5. **Reduced Risk:** Unstaking provides a way for participants to mitigate risks by withdrawing tokens from the staking process during periods of uncertainty.
6. **Participation in Other Activities:** Unstaking allows participants to use their tokens for other activities like trading, lending, or providing liquidity in decentralized finance (DeFi) protocols.
7. **Network Engagement:** Unstaking can encourage participants to actively engage with the network's staking mechanisms, contributing to a dynamic ecosystem.

In summary, an **Unstake Transaction** involves withdrawing tokens that were previously staked in the L1X blockchain network. It offers benefits such as increased liquidity, flexibility, and portfolio management. Unstaking provides participants with the ability to adjust their strategies, engage in other activities, and respond to changing market conditions while managing their token holdings.

10.8 L1X_Swap: X-Talk Token Swap

L1X_Swap is a cutting-edge cross-chain transaction type that enables the seamless exchange of fungible tokens between two distinct blockchain networks, referred to as Chain A and Chain B. This innovation closes the gap between EVM (Ethereum Virtual Machine) and Non-EVM compatible blockchain networks, offering users the ability to effortlessly trade fungible tokens across different platforms.

10.8.1 How It Works

When users initiate an L1X_Swap, they are essentially swapping a fungible token on Chain A, for its equivalent counterpart on Chain B. This transaction is executed automatically, with the source token being burnt on Chain A, ensuring its removal from the source chain. Simultaneously, the equivalent destination token is minted on Chain B, making it instantly available for use on the destination chain. This versatile functionality empowers users to leverage the swapped tokens on a variety of platforms for purposes such as transaction fees, staking, governance participation, and trading.

10.8.2 Use Cases

L1X_Swap is employed to facilitate the exchange of fungible tokens between disparate, previously incompatible blockchain networks, for example between the Binance Smart Chain (BSC) and Solana. This enables users to access and utilize their tokens seamlessly on both chains. The tokens acquired through L1X_Swap can be used for:

1. **Transaction Fees:** Users can utilize the tokens to pay for transaction fees on the destination chain, making it convenient to conduct activities without constantly switching between tokens.
2. **Staking:** Swapped tokens can be staked on the destination chain's staking platforms, allowing users to earn rewards and contribute to the network's security and operations.
3. **Governance Participation:** Token holders can participate in governance processes on the destination chain, influencing decisions and policies of the network.
4. **Trading:** The tokens can be freely traded on both chains, increasing liquidity and trading opportunities for users.

10.8.3 Benefits

1. **Interoperability:** L1X_Swap breaks down the barriers between different blockchain networks, allowing fungible tokens to flow seamlessly across chains, regardless of their compatibility.
2. **Efficiency:** The automated process of token burning and minting ensures a swift and hassle-free exchange, reducing the need for manual interventions.
3. **Versatility:** Users gain the flexibility to utilize their tokens for various purposes on different platforms, enhancing their engagement within the blockchain ecosystems.
4. **Enhanced Liquidity:** By enabling token trading on multiple chains, L1X_Swap increases liquidity and trading opportunities for users, contributing to healthier markets.
5. **Accessibility:** L1X_Swap democratizes access to different blockchain networks, enabling users to benefit from the unique features and opportunities each chain offers.

In summary, L1X_Swap revolutionizes the way users interact with and utilize fungible tokens across disparate blockchain networks. Its automated and efficient token exchange mechanism, combined with the versatility of token usage, empowers users to seamlessly engage in activities such as staking, governance, and trading on multiple platforms.

10.9 L1X_Stake: X-Talk Token Staking

L1X_Stake introduces a ground-breaking cross-chain transaction type that empowers users to mint and own native Fungible Tokens (FT) on Chain A and seamlessly stake them as destination native FT on Chain B. This innovative solution closes the gap between EVM and Non-EVM compatible blockchain networks, enabling users to lock tokens on one platform and stake them as destination native tokens on another.

10.9.1 How It Works

With L1X_Stake, users can mint and own a native Fungible Token on Chain A. This source token is then automatically locked on Chain A. Simultaneously, an equivalent destination native FT is staked on Chain B, utilizing a Smart Contract on the L1X network. This process ensures that the user's source token remains locked and secure on its original platform, while the staked equivalent on Chain B contributes to consensus, network security, and governance activities.

10.9.2 Use Cases

L1X_Stake serves as a powerful tool for seamless participation in cross-chain staking, enhancing token utility, and ecosystem engagement. Key use cases include:

1. **Consensus Participation:** The staked tokens play a vital role in maintaining consensus and network security on the destination chain. Participants contribute to validating transactions and securing the network's integrity.

2. **Network Security:** By staking tokens, users actively participate in securing the destination chain's operations, discouraging malicious activities and ensuring the overall health of the network.
3. **Governance Involvement:** Staked token holders gain the ability to participate in governance processes on the destination chain, influencing decisions related to protocol upgrades, parameter adjustments, and more.

10.9.3 Benefits

1. **Interoperability:** L1X_Stake facilitates the seamless movement of tokens between disparate blockchain networks, allowing users to participate in staking activities on platforms beyond their original chain.
2. **Token Utility Expansion:** Users can unlock new opportunities by staking their source tokens as destination native FT, engaging in consensus, network security, and governance activities.
3. **Risk Management:** By locking the source tokens on Chain A and staking the equivalent on Chain B, users mitigate risk exposure and diversify their participation in multiple ecosystems.
4. **Economic Incentives:** Participants are rewarded for their staking contributions with incentives such as token rewards, transaction fee sharing, and governance tokens, encouraging active engagement.
5. **Ecosystem Growth:** L1X_Stake promotes collaboration between blockchain networks, fostering a dynamic ecosystem where users can seamlessly participate in diverse activities across chains.

In conclusion, L1X_Stake revolutionizes the cross-chain staking landscape by allowing users to mint, lock, and stake native Fungible Tokens on different blockchain networks. This innovative transaction type enhances token utility, network security, and governance participation while providing users with a seamless and efficient way to engage with multiple ecosystems simultaneously.

10.10 L1X_LendBorrow: X-Talk Fungible Token Lending and

Borrowing

L1X_LendBorrow introduces a revolutionary cross-chain transaction type that empowers users to lend a Fungible Token (FT) on Chain A and simultaneously borrow the equivalent FT on Chain B. This ground-breaking solution closes the gap between EVM and Non-EVM compatible blockchain networks, offering users the ability to leverage their locked tokens as collateral for borrowing purposes on a different platform.

10.10.1 How It Works

L1X_LendBorrow enables users to lock and lend a FT on Chain A. This locked source token serves as collateral for the creation of an equivalent borrowed FT on Chain B. This borrowed token is minted automatically, granting users access to liquidity without having to liquidate their holdings on the source chain. The borrowed FT on Chain B can be utilized for various purposes, such as transaction fees, staking, governance participation, and trading.

10.10.2 Use Cases

L1X_LendBorrow presents users with a multitude of use cases that enhance liquidity, unlock new opportunities, and promote engagement across different blockchain ecosystems:

1. **Liquidity Provision:** Users can leverage their locked tokens as collateral to borrow funds, ensuring access to liquidity for activities on the destination chain.
2. **Arbitrage and Trading:** Borrowed tokens can be used for arbitrage and trading opportunities, enabling users to capitalize on price discrepancies between different platforms.
3. **Transaction Fees:** Borrowed tokens can be utilized to pay for transaction fees the destination chain, streamlining user experiences.
4. **Staking and Yield Farming:** Borrowed tokens can be staked or used in yield farming programs on the destination chain, allowing users to earn rewards and participate in ecosystem growth.

5. **Governance Participation:** Token holders can use borrowed tokens to participate in governance processes on the destination chain, influencing decision-making within the ecosystem.

10.10.3 Benefits

1. **Seamless Cross-Chain Usage:** L1X_LendBorrow closes the gap between different blockchain networks, enabling users to leverage their locked tokens on one chain for various activities on another.
2. **Preservation of Holdings:** Users can access liquidity without selling their locked tokens, ensuring they continue to benefit from potential price appreciation and long-term value.
3. **Opportunity Diversification:** Borrowed tokens open up new avenues for users to engage in activities such as trading, staking, and governance participation on the destination chain.
4. **Efficient Collateral Utilization:** L1X_LendBorrow maximizes the utility of locked tokens by allowing users to simultaneously lend and borrow, making the most of their holdings.
5. **Increased Flexibility:** Users gain flexibility in managing their assets, harnessing the power of lending and borrowing to achieve their financial goals.

In summary, L1X_LendBorrow redefines token utility by enabling users to lock and lend tokens on one blockchain and borrow their equivalent on another. This innovative cross-chain transaction type enhances liquidity, promotes engagement, and offers users a seamless way to access the benefits of multiple ecosystems without sacrificing the potential of their locked holdings.

10.11 L1X_Advertisement: X-Talk NFT Liquidity Provision

L1X_Advertisement introduces a pioneering cross-chain transaction type that enables users to mint and own Non-Fungible Tokens (NFTs) on Chain A and seamlessly advertise them on various platforms, such as OpenSea, Magic Eden, and Ethereum Marketplace. This innovative solution connects ecosystems, allowing NFTs to be effectively utilized and promoted while maintaining their utility and permissions.

10.11.1 How It Works

With L1X_Advertisement, users can mint and possess NFTs on Chain A. These NFTs can be advertised for sale on platforms like OpenSea, Chain B, and Ethereum Marketplace. When the NFT changes ownership on any of these platforms or chains, the NFT is automatically burned or custom-actioned, ensuring that the NFT's utility and permissions are preserved. This enables users to leverage the same NFT on multiple platforms while synchronizing its status across different ecosystems.

10.11.2 Use Cases

L1X_Advertisement opens up new horizons for NFT owners and creators, offering a range of applications and possibilities:

1. **Multi-Platform Sales:** Users can list their NFTs on various marketplaces, increasing exposure and opportunities for sales without creating multiple instances of the same NFT.
2. **Interoperable Utility:** NFTs can be used for specific purposes on different chains or platforms, such as being part of a game on Chain A while being advertised on Ethereum's OpenSea Marketplace.
3. **Cross-Chain Collectability:** Collectors can have the same NFT on different chains, preserving its rarity and uniqueness while being part of diverse ecosystems.

10.11.3 Benefits

1. **Efficient NFT Management:** L1X_Advertisement streamlines NFT promotion by allowing users to list and advertise their NFTs on multiple platforms without duplicating NFT instances.
2. **Wider Exposure:** NFTs can be showcased and sold on various marketplaces, reaching a broader audience and increasing the chances of finding interested buyers.

3. **Maintained Rarity:** The automatic burn or custom action upon ownership change ensures that the NFT's rarity and uniqueness are preserved, regardless of the platform.
4. **Versatile Utility:** NFTs can have diverse applications across different platforms, making them a versatile asset in multiple ecosystems.
5. **Simplified Cross-Chain Integration:** L1X_Advertisement makes it easy for NFT owners to engage with different blockchain networks and marketplaces without the need for complex technical integrations.

In summary, L1X_Advertisement revolutionizes the NFT space by offering a seamless way to mint, own, and advertise NFTs across various platforms and chains. This cross-chain transaction type enhances NFT accessibility, reach, and utility, allowing users to effectively utilize their NFTs while preserving their rarity and value across diverse ecosystems.

10.12 L1X_Lease: X-Talk NFT Leasing Provision

L1X_Lease is a cross-chain transaction type that empowers users to lease Non-Fungible Tokens (NFTs) on Chain A and enables lessees to utilize these NFTs seamlessly on Chain B. This innovative solution, known as X-Talk, revolutionizes NFT leasing across diverse blockchain networks, promoting collaboration and flexibility.

10.12.1 How It Works

With L1X_Lease, lessors can lease out NFTs on Chain A, while lessees utilize these NFTs on Chain B. This cross-chain interaction is facilitated through the X-Talk mechanism. The NFT on Chain A will be locked during the leasing period, ensuring the lessee's exclusive use. Upon the agreed leasing term completion, the NFT is automatically custom-actioned according to the predefined terms, effectively transferring control back to the lessor.

10.12.2 Use Cases

L1X_Lease brings forth a wide array of use cases and opportunities for both lessors and lessees:

1. **Collaborative Ventures:** Creators and collectors can collaborate by leasing NFTs, fostering cross-chain partnerships and engaging audiences on multiple platforms.
2. **Monetization for Lessor:** Lessor can monetize their NFTs by leasing them out, generating income during periods of non-use.
3. **Temporary Access:** Lessees gain access to NFTs on different chains for various purposes such as virtual events, gaming, or showcasing without the need for outright ownership.

10.12.3 Benefits

1. **Cross-Chain Accessibility:** L1X_Lease enables NFTs to be utilized and enjoyed across different chains, broadening their reach and utility.
2. **Monetization Opportunities:** NFT owners can earn income by leasing out their assets, creating an additional revenue stream from their collections.
3. **Temporary Ownership:** Lessees can use and benefit from NFTs for a specific period without having to commit to long-term ownership.
4. **Platform Integration:** L1X_Lease simplifies the process of utilizing NFTs across platforms, eliminating complex technical integrations.
5. **Diverse Use Cases:** Lessees can leverage the NFTs for purposes ranging from virtual exhibitions, events, gaming, and more.

In conclusion, L1X_Lease redefines NFT utility by allowing users to lease and utilize NFTs across different blockchain networks. This cross-chain transaction type fosters collaboration, monetization, and accessibility, creating a versatile ecosystem where NFTs can be temporarily leased and effectively used by lessees on various platforms.

11. X-Talk

Blockchain interoperability refers to cross-chain communication of smart contracts. L1X provides interoperability at the base layer of the blockchain.

11.1 X-Talk Architecture

X-TALK is the cross-chain communication architecture provided by L1X for the secure exchange of smart contracts. It has a decentralized architecture and has a blockchain-agnostic framework enabling seamless asset and logic exchange across chains. X-TALK ensures communication throughout the process of interacting with multiple chains.

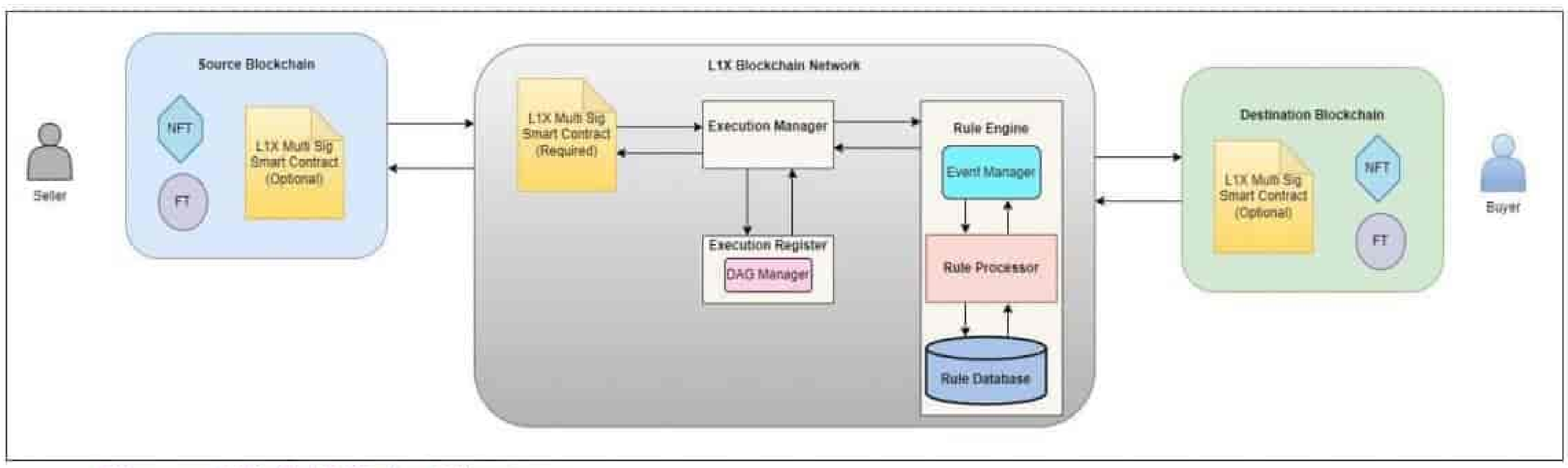


Figure 4 X-TALK Architecture

1. **Execution Manager**

The Execution Manager provides a layer of abstraction between the underlying cross-chains for seamless interaction among them. It deals with the identification, classification, verification, and execution strategy for the received event. It endows the blockchain-agnostic behavior of L1.X-TALK. Below listed are the major functions of the Execution Manager.

i. Identify the transaction type.
ii. Listen to the incoming events from the Rule Engine.
iii. Verify the received event.
iv. Perform logical validation for the received event.
v. Create a structure for the responsive action.
vi. Connect with other components such as Execution Register and smart contracts.

2. Rule Engine

The Rule Engine is the core component of X-TALK architecture, being responsible for processing and executing the rules that initiate and verify the communication cross-chain. It receives events coming from the Event Manager and triggers appropriate action based on predefined rules. It provides an empty struct to Execution Manager with different functions and rules such as Mint, Transfer or Collect Fees and its relevant variable/function parameter details.

a. Event Manager

The Event Manager listens to events from smart contracts or other functions that have the feature of emitting events. It monitors the events from these sources, validates them, and conveys the events to the Rule Engine, ensuring data integrity throughout the process.

b. Rule Processor

The Rule Processor filters the event details received from the Event Manager and sends only relevant information to the Rule Database. The Rule Processor acts as an interface between the Event Manager and the Rule Database to get a response with the associated function call.

c. Rule Database

This component contains the execution rules and instructions to create appropriate events/transactions for the target blockchain in response to the request received from the relevant blockchain. It is responsible for mapping the events received from the relevant blockchain to the appropriate transactions that need to be executed on the target blockchain.

3. Execution Register

The Execution Register is responsible for maintaining the state and integrity of the blockchain. It updates the state with the transaction level information as and when received from the Execution Manager. It keeps track and maintains a complete record of all the executed transactions by the system. The Execution Register is a crucial component in ensuring the security and integrity of the X-TALK system, as it provides a complete and tamper-proof record of all transactions.

a. **DAG Manager**

The Execution register employs a Directed Acyclic Graph to orchestrate the transaction execution tasks, to ensure appropriate execution order while avoiding conflicts (if any).

4. **L1X MultiSig Smart Contract**

L1X MultiSig Smart Contract is deployed on both, the source and destination blockchains. Its job is to act as a secure and trusted component between the user and the L1X platform for NFT transactions. It enables seamless interaction and validation across multiple chains, facilitating cross-chain transactions with ease. It allows the user to give complete authority to the L1X blockchain to carry cross-chain tokens and asset exchange. It adds an extra layer of security by requiring multiple signatures from authorized participants, mitigating the risk of unauthorized or fraudulent activities.

11.2 X-Talk TechStack

The X-Talk Tech Stack is a robust and visionary infrastructure designed to revolutionize the world of blockchain technology. Comprising two main layers, Business Logic and X-Talk Interface, it empowers developers and users to decentralize their business logic and state while enabling seamless cross-chain communication. Let's explore the stack layer by layer in sub-sections.

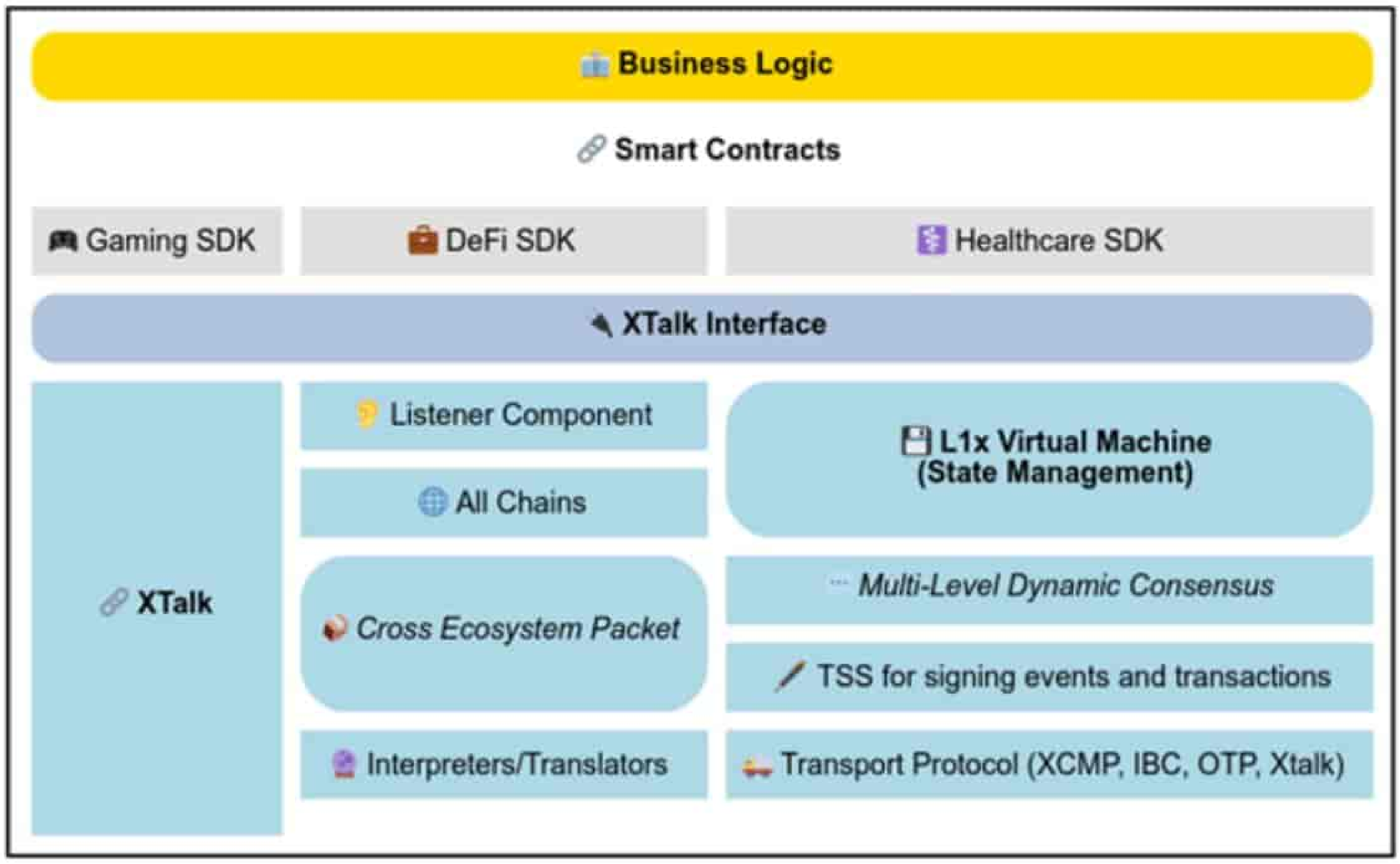


Figure 5 X-Talk Tech Stack

11.2.1 Business Logic Layer

At the pinnacle of the X-Talk Tech Stack is the Business Logic layer, which represents the core of X-Talk's functionality. It is essential for defining and executing complex logic and operations on the blockchain. It's designed to enable trustless, automated, and decentralized business processes. It consists of several key components that collectively drive its transformative capabilities.

1. **Smart Contracts**: Smart contracts are the workhorses of the blockchain ecosystem. They encode predefined rules and logic, allowing for automated execution of agreements and transactions. They remove the need for intermediaries, reducing costs and increasing transparency.
2. **Gaming SDK**: Catering to the gaming industry, this Software Development Kit empowers developers to create decentralized games and applications, ensuring fairness, transparency, and trustlessness, eliminating the potential for fraud and cheating.
3. **DeFi SDK**: The DeFi SDK empowers developers to build decentralized financial applications. It enables automated lending, borrowing, trading, and other financial activities, reducing reliance on traditional financial intermediaries and promoting financial inclusion.

4. **Healthcare SDK**: Tailored for the healthcare sector, this SDK promotes interoperability and secure data sharing among healthcare applications. It enhances patient data privacy, streamlines healthcare processes, and improves patient care.

11.2.2 X-Talk Interface Layer

Beneath the Business Logic layer lies the X-Talk Interface, which serves as the layer connecting various blockchain networks and components, facilitating cross-chain communication and logic validation. The XTalk Interface also makes it possible for the underlying XTalk protocol and the application-specific business logic to communicate with one another.

1. **XTalk**: At the heart of this layer is XTalk itself, the communication and logic validation infrastructure that drives interoperability between blockchains. It facilitates the flow of information and transactions across disparate chains. It is a Smart Contract Format with pre-defined contract rules that provides an interface to process, store, and execute types such as event, data, and payload with rules associated with every event object. It provides composability to apply rules on types with determinism with X-Talk Flow Contract.
2. **Listener Component (Listening to All Chains)**: This component serves as the attentive ear, listening to events and transactions across all participating chains. It ensures that no information is lost and that all chains communicate effectively. It takes a proactive role in verifying the incidents reported. Additionally, it performs the role of passive observers, identifying and documenting modifications to the blockchain brought about by other blockchain networks. These modifications may include new transactions, interactions between smart contracts, or state alterations.
3. **Cross-Ecosystem Packet:** It represents the fundamental building block of communication, embarking on a journey across the intricate X-Talk protocol to closes the gap between disparate blockchain networks. These packets serve as the vessels of valuable information and precise instructions, encapsulating the very essence of interoperability. The L1XVM (Layer 1 X Virtual Machine) packet, with its intricate fields such as interpreter, user_origin, salt, program, and assets, acts as the initial messenger. As this packet enters the capable hands of the executor residing within the virtual machine, a remarkable transformation unfolds. The executor's role is akin to that of a skilled translator, taking the L1XVM packet and expertly crafting it into the language

understood by the destination chain. Whether it manifests as an XCMP Packet for seamless message passing, an IBC Packet facilitating cross-blockchain communication, or an XTalk Cross Ecosystem packet tailored to the specific format of the destination chain, the cross-ecosystem packet stands as a beacon of unity amidst the diversity of blockchain networks. It's the linchpin that harmonizes these ecosystems, fostering a new era of collaboration, efficiency, and endless possibilities in the world of blockchain technology.

4. **Interpreters/Translators**: These components facilitate the translation and interpretation of data and transactions between different blockchain ecosystems, ensuring seamless communication and compatibility. Interpreters particularly deal with Data Format Conversion, Protocol Conversion, and Semantic Understanding whereas Language Translation, Functionality Mapping, and Security and Validation are handled by Translators. They allow different blockchains with varying data formats, protocols, and programming languages to understand and work with each other's data, logic, and functionalities. This enables the seamless exchange of assets, information, and services between blockchain ecosystems, ultimately realizing the vision of cross-chain communication and collaboration that X-Talk seeks to achieve.
5. **L1X Virtual Machine**: This virtual machine manages the state of X-Talk across chains, ensuring consistency and reliability in data management. Without forcing users to commit their assets to outside parties, it enables direct digital asset exchange and brings cross-chain smart contract execution. Assuring the maximum flexibility and versatility, this real native interoperability enables users to connect to any chain, whether it is private or public, EVM-based or not.
6. **Multi-Level Dynamic Consensus**: Dynamic consensus mechanisms help maintain agreement and consistency across disparate chains, even as they evolve and adapt.
7. **TSS (Threshold Signature Scheme) for Signing Events and Transactions**: TSS enhances security by enabling multiple parties to collaboratively sign events and transactions, safeguarding against unauthorized access.
8. **Transport Protocol (XCMP, IBC, OTP, XTalk)**: At the foundation of the last vertical are transport protocols like XCMP (Cross-Chain Message Passing), IBC (Inter-Blockchain Communication), OTP (Off-Chain Transaction Protocol), and XTalk's proprietary protocol. These protocols facilitate the secure exchange of data and

transactions across chains, forming the backbone of X-Talk's cross-chain communication infrastructure.

11.3 X-Talk Flow Contract

The X-Talk Flow Contract serves as the central orchestrator of the entire X-Talk infrastructure. It facilitates the processing, storage, and execution of event-driven data and payloads based on predefined rules specific to each event type. This contract ensures deterministic application of rules across different types of events, enhancing interoperability and consistency within the L1X blockchain ecosystem. The X-Talk Flow Contract handles cross-chain communication seamlessly, enabling developers to define and execute logic for various event types efficiently.

X-Talk Flow Contract includes X-Talk Port Contract, Event Configuration Contract and Signing-Broadcasting Contract as a subset of the flow contract. These contracts process instructions based on X-Talk messaging standard or instruction sets. It comprises several key components, each with its own unique purpose and significance. Let's delve into each of these components in more detail.

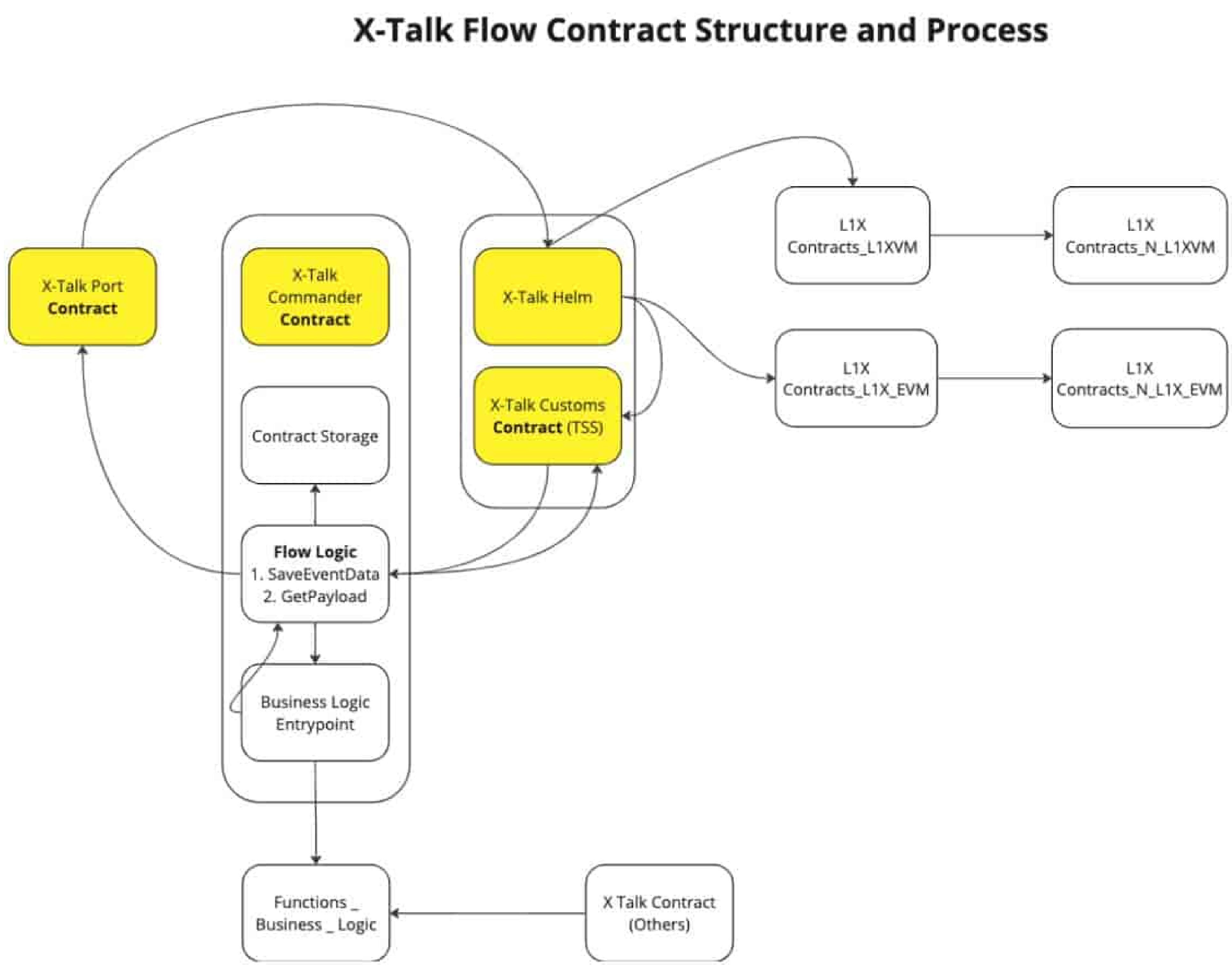


Figure 6 X-Talk Flow Contract

11.3.1 X-Talk Port Contract

The X-Talk Port Contract is a crucial component residing on the Listener Node within the X-Talk infrastructure. Its primary responsibility is to serve as a connector between the external world and the L1X Protocol, facilitating the seamless flow of data into the blockchain. This data encompasses various sources:

- **Real-World Data (Oracle)**: The contract is designed to fetch and integrate real-world data, ensuring that blockchain applications can access and utilize up-to-date information from external sources.
- **Event Data from Other Protocols**: It can collect event data generated by other blockchain protocols, enabling cross-chain communication and interoperability.
- **Custom Data Insertion Scripts**: Custom data insertion scripts empower the contract to acquire data from diverse applications or configured endpoints through APIs. This versatility allows for the integration of specific data sources as needed.

Upon receiving data, the X-Talk Port Contract standardizes it into a serializable format known as the "X-Talk Message." This standardized format ensures compatibility and consistency in data representation. Subsequently, the data is forwarded for validation.

Event Listening Nodes, specialized components within the X-Talk infrastructure, validate the incoming data, adding a unique identifier for tracking purposes. Once validated, the data is then handed over to the Scheduler for further processing.

The Scheduler plays a pivotal role in organizing and dispatching the serialized data based on predefined filters and rules. These filters encompass criteria such as data type and event types, ensuring that the data reaches the appropriate destination cluster within the blockchain network. In essence, the X-Talk Port Contract serves as the gateway for external data to enter the L1X Protocol, maintaining data integrity and routing it efficiently within the blockchain ecosystem.

11.3.2 X-Talk Commander Contract

The X-Talk Commander Contract serves as the central orchestrator in defining and controlling the flow of interactions and logic within the X-Talk ecosystem. Developers leverage this contract to build modular logic for cross-chain applications, employing interfaces that encompass:

- **Contract Storage**: This component allows developers to manage and manipulate various states of the contract, including events, default and transformed payloads, business logic, and other predefined states. It ensures that critical contract data is stored and accessible as needed.
- **Flow Logic**: The X-Talk Commander Contract incorporates essential methods that enable the contract to interact with and process data effectively.
    - **save_event_data(...)**: This method stores event data within the contract, categorizing it based on the source ID and event type. It also handles the setup of payloads for execution based on the nature of the event.
    - **entrypoint_Business()**: This function is where core business logic is implemented, allowing for the manipulation of default event data and the

application of transformations before payload creation.

- **get_payload_to_sign(...)**: This method retrieves payloads associated with a given global transaction ID, facilitating their signing for cross-chain verification.

Additionally, Business Logic Entry points enable the transformation of payloads within the contract. These entry points facilitate both intrasystem transformations and cross-contract calls, extending the functionality of the contract by incorporating the capabilities of the L1X VM and L1X EVM. This seamless integration ensures that smart contracts within the X-Talk ecosystem can interact, share data, and execute complex operations efficiently.

11.3.3 X-Talk Helm

X-Talk Helm handles routing, data retrieval, and instruction provision within the X-Talk ecosystem. Its primary functions include:

- **Routing**: X-Talk Helm determines the appropriate path for data and instructions, ensuring that they reach their intended destinations. It plays a pivotal role in directing the flow of information within the ecosystem.
- **Data Retrieval**: The component is responsible for fetching data from relevant sources, making it available for processing and further interactions within the X-Talk ecosystem.
- **Instruction Provision**: X-Talk Helm interfaces with the X-Talk Customs contract, providing instructions for signing and broadcasting payloads to internal contracts or cross-chain contracts. This functionality ensures that transactions and interactions are securely executed across the blockchain network.

Overall, X-Talk Helm introduces modularity into the ecosystem, enabling dynamic routing and the selection of appropriate signing and broadcasting nodes. This flexibility ensures efficient communication and interaction with default or transformed payloads, enhancing the versatility and adaptability of the X-Talk infrastructure.

11.3.4 X-Talk Customs Contract

The X-Talk Customs Contract plays a pivotal role within the X-Talk ecosystem, responsible for ensuring the security and integrity of data transmissions. Its primary responsibility lies in providing the necessary signing mechanism for payloads before broadcasting them to their designated destinations.

Here's a detailed breakdown of the X-Talk Customs Contract's role and significance:

- **Payload Signing**: The core responsibility of the X-Talk Customs Contract is to securely sign payloads. Before any data is broadcasted to its intended destination, it is essential to validate and authenticate the data to prevent tampering or unauthorized access. This contract ensures that payloads are equipped with the appropriate digital signatures, enhancing data security within the X-Talk ecosystem.
- **Data Broadcasting**: Once payloads are appropriately signed, the X-Talk Customs Contract takes on the task of broadcasting these data packets to their designated destinations. This process ensures that the data reaches the target clusters or contracts reliably and efficiently.

The X-Talk Customs Contract acts as a crucial security checkpoint, adding an additional layer of trust and verification to the data flow within the L1X ecosystem. By providing the right signing mechanisms, it helps guarantee the authenticity and integrity of the data being transmitted, a critical aspect in the decentralized and trustless environment of blockchain technology.

In essence, the X-Talk Customs Contract serves as the guardian of data integrity, ensuring that only verified and signed payloads are transmitted to their intended recipients, thus upholding the security and reliability of the X-Talk infrastructure.

11.4 Use Case: Advertise BSC NFT Sale in OpenSea Ethereum Marketplace

Generally, a seller having an NFT on a blockchain, usually prefers to advertise selling the NFT in a marketplace on the same blockchain. This limits the advertisements' visibility to a larger audience of potential buyers. To provide greater liquidity and a larger user base, a seller may wish to advertise the NFT sale in another marketplace on the Ethereum blockchain. For this purpose, bridges are utilized. But these bridges are not decentralized or safe.

X-TALK allows any user on an EVM-compatible blockchain to publish an advertisement for an NFT sale in the OpenSea marketplace on the Ethereum blockchain network in a completely decentralized way. This section describes the X-TALK workflow for this scenario.

Scenario: The seller has an NFT on the Binance Smart Chain.

Aim: To advertise NFT on OpenSea Ethereum Marketplace

11.5 X-Talk Workflow

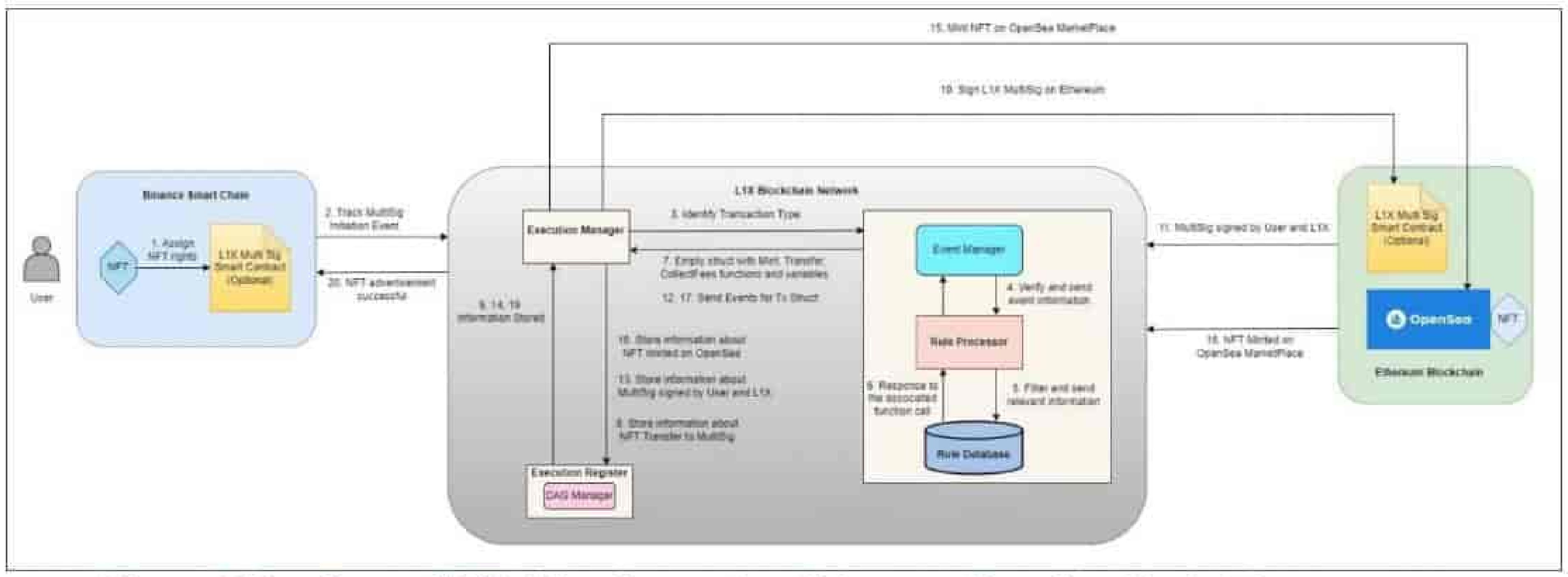


Figure 7 Use Case – NFT Advertisement on Ethereum OpenSea Marketplace

Workflow:

1. User assigns NFT rights to the L1X MultiSig smart contract on the Binance Smart Chain.
2. User initiates the Smart Contract on the L1X Chain that makes the Execution Manager ready in terms of listening to the MultiSig contract initiation events.
3. Execution Manager informs the Rule Engine about Transaction Type. In this use case, it's 'L1X_Advertisement'.
4. Rule Engine verifies the event and forwards it to the Rule Processor.
5. Rule Processor filters and sends relevant information to the Rule Database.
6. Rule Database gives the response for the associated function call.
7. Rule Engine sends an event for an empty struct to the Execution Manager with function/rules such as Mint, Transfer, Collect Fees, and its relevant variable/function parameter details.

8. Execution Manager requests the Execution Register to store the information about NFT transfer to the MultiSig.
9. Execution Register stores the event information and informs Execution Manager.
10. Execution Manager invokes the user and L1X to sign the MultiSig on the Ethereum blockchain network.
11. L1X MultiSig Smart Contract triggers an event once the MultiSig is signed by the user and L1X.
12. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
13. Execution Manager updates the Execution Register to store information that L1X MultiSig on Ethereum blockchain network is signed by the user and L1X.
14. Execution Registers informs Execution Manager about information storage.
15. Execution Manager invokes L1X MultiSig Smart Contract on Ethereum blockchain network to mint NFT on OpenSea Marketplace.
16. Event is triggered once NFT is successfully minted on OpenSea Marketplace.
17. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
18. Execution Manager provides the information about successfully NFT minting on OpenSea Marketplace to Execution Register.
19. Execution Register stores the information and updates Execution Manager.
20. User is informed about successful NFT advertisement and listing on Ethereum OpenSea marketplace.

11.6 Benefits

Benefits of the above discussed use case are as listed below:

1. **Decentralized Architecture:** Reliance on a central authority for asset transfer across the blockchain networks poses potential risks related to censorship and control. L1X provides a completely decentralized architecture.

2. **Increased User Base:** Publishing an advertisement in the marketplace on the Ethereum blockchain network increases the user base as Ethereum has a robust ecosystem of many NFT marketplaces.

3. **Greater Liquidity:** The availability of active traders and a well-developed ecosystem allows Ethereum to provide greater liquidity for NFTs.

11.7 X-Talk vs. Bridges: A Comprehensive Comparison for Blockchain Interoperability

In the realm of blockchain interoperability, choosing the right approach is pivotal for enabling seamless communication between disparate blockchain networks. This section delves into a detailed comparison of X-Talk vs. bridges. X-Talk, characterized by its single-contract architecture and native cross-chain capabilities, is pitted against traditional bridges, which rely on multiple contracts and external connectors to achieve blockchain interoperability. Let's explore the nuances of these two approaches, spanning contract complexity, state management, event handling, and more, to better understand their respective strengths and applications.

11.7.1 Number of Contracts

- **X-Talk**: X-Talk operates with a single smart contract for facilitating cross-chain communication. It simplifies the contract structure, making it easier to manage and deploy.

- **Bridge**: Bridges typically involve multiple contracts to establish connections between different blockchains. This multi-contract approach introduces complexity and coordination challenges.

11.7.2 Contract Complexity

- **X-Talk**: X-Talk contracts are designed to be decentralized, often serving as the primary point of interaction for cross-chain applications.

- **Bridge**: Bridge contracts are distributed across multiple chains and require coordination. They are inherently more complex due to the need to synchronize activities across various contracts.

11.7.3 State Management

- **X-Talk**: State management primarily occurs within the single contract, simplifying data handling and ensuring data consistency.

- **Bridge**: Bridge solutions often involve state management across multiple contracts on different chains. This distributed approach can lead to increased complexity and the potential for data inconsistencies.

11.7.4 Event Handling

- **X-Talk**: X-Talk contracts handle events directly within the contract itself, simplifying event management.

- **Bridge**: Bridge solutions may require individual specialized contracts to handle events on each participating blockchain, potentially leading to a fragmented event-handling structure.

11.7.5 Storage Interaction

- **X-Talk**: X-Talk contracts typically interact directly with storage through load and save methods, streamlining data operations.

- **Bridge**: Bridge solutions might involve storage interactions spread across different contracts on various chains. This can result in increased gas costs and complexity.

11.7.6 Interoperability

- **X-Talk**: X-Talk is purpose-built for cross-chain applications, offering native interoperability features without the need for external connectors or oracles for each chain involved.

- **Bridge**: Bridges often rely on external connectors or oracles to enable communication between different blockchains. This external dependency introduces additional points of potential failure and complexity.

11.7.7 Security Concerns

- **X-Talk**: X-Talk's single-contract approach simplifies the system's architecture, reducing the risk of a decentralized point of failure.

- **Bridge**: Bridge solutions involve multiple contracts and chains, potentially increasing the number of points of failure. However, they also isolate risks to some extent, enhancing security in specific scenarios.

11.7.8 Development Complexity

- **X-Talk**: Development complexity is lower when working with X-Talk due to the simplicity of single contract development.

- **Bridge**: Bridge development is more complex as it necessitates coordination between multiple contracts on different chains, requiring careful planning and synchronization.

11.7.9 Execution Efficiency

- **X-Talk**: X-Talk contracts are more efficient in terms of execution because they encapsulate logic within a single decentralized contract, reducing overhead.

- **Bridge**: Bridge solutions can introduce high overhead and complexity due to interactions between multiple contracts. This may impact execution efficiency, especially for cross-chain operations.

In the ever-evolving landscape of blockchain interoperability, X-Talk stands out as a superior solution due to its streamlined, single-contract architecture, native cross-chain capabilities, and efficient execution. X-Talk simplifies the complex task of cross-chain communication, offering a secure and efficient approach without the need for multiple contracts and external dependencies. Its seamless integration and reduced development complexity make it a compelling choice for those seeking a robust and user-friendly solution for blockchain interoperability.

12. Virtual Machine

L1X Virtual Machine (L1X VM) provides a secure and deterministic environment for the execution of smart contracts on the L1X blockchain. It ensures determinism in the smart contract execution lifecycle through the process of transaction and block creation. It limits the boundaries of smart contract execution by providing an isolated environment that restricts its interference with the blockchain network.

12.1 Technical Stack

The entire smart contract development flow goes through three phases:

- **Frontend**. A developer writing a smart contract on L1X. They will be using Rust programming language which provides a safe and efficient environment for contract development. L1X Smart Contract SDK provides them with a safe API that they can use to build such contracts.
- **Middle End**. The code goes through various transformations to reach its final form.
    - ***Macro pre-processing.*** By leveraging Rust's powerful metaprogramming capabilities, high-level abstractions for the ease of development on L1X are created.
    - ***WASM.*** The LLVM compiler infrastructure is used for compiling and optimizing Rust code into WASM.
    - ***WASM -> eBPF.*** L1X compiler takes the generated WASM file and compiles it to eBPF bytecode.
- **Backend**. The resulting ELF file with eBPF bytecode is the contract binary that the developer publishes on-chain.
    - ***L1X VM.*** The Layer One X Virtual Machine is responsible for executing the smart contracts in a secure and efficient manner, integrating with the L1X blockchain and handling state updates.

12.2 Virtual Machine Architecture

The L1X VM is designed to provide a robust and efficient environment for executing smart contracts on the Layer One X blockchain platform. It is built on top of the eBPF bytecode interpreter and leverages the LLVM compiler infrastructure to compile and optimize smart contracts written in the Rust programming language.

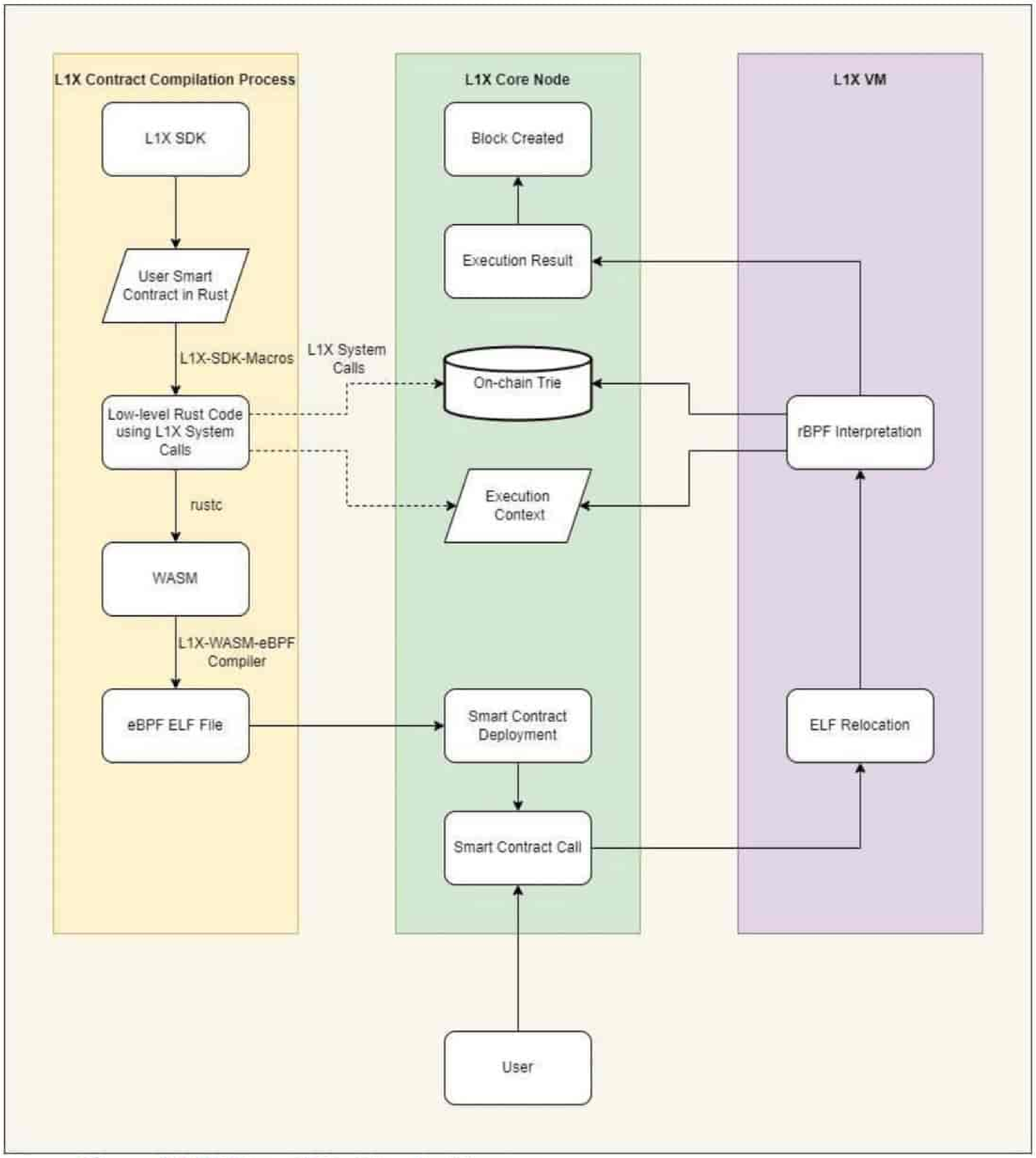


Figure 8 L1X Virtual Machine Architecture

The following comprise the components of the L1X VM:

12.2.1 L1X Smart Contract Software Development Kit (SDK)

Contract SDK is a collection of three crates (going from lowest level to highest level):

a. **l1x-sys**: A header file specifying all of the available system calls using their native signatures. A developer is unlikely to use any of these directly unless they want to build something very low-level without involving l1x-sdk.
b. **l1x-sdk-macros**: A collection of macros that create an abstraction over some of the fine details in VM. For example, it enables developers to accept arguments and return values as if this was just a regular Rust program.
c. **l1x-sdk**: Contains two main things right now: higher-level system calls (e.g., a low-level system call deals with raw pointers and data length, while a high-level system call just accepts &str) and on-chain collections.

12.2.2 L1X System Calls

L1X System Calls can be categorized as listed below:

a. **Register API:** Provides a register abstraction between guest and host machines. These system calls enable guests to write into registers, read from registers, and access the length of data contained within. Please note that these registers are different from the actual WASM or eBPF registers.
b. **Storage API**: Provides guest machine with capabilities to store data on the trie (i.e. a key- value storage that is a part of the L1X consensus). The system calls enable guests to read, write or remove data from the trie.
c. **Context API**: Provides guest machine access to the context in which the current call is being executed. Input writes the user-supplied arguments in the specified register. Output registers a certain byte slice as the result of this call's execution.
d. **Misc. API**: Other system calls are covered under this category. *panic* interrupts the execution immediately. *msg* emits an on-chain message using the supplied byte slice as a UTF-8 string.

12.2.3 L1X-SDK-Macros

L1X-SDK-Macros provide pre-defined functionality at a higher level of abstraction. They list any constants, functions, or data types that are required for development on the L1X platform. Currently, this crate supplies one macro *#[program]* that translates conventional Rust code to something that employs l1x-sys primitives.

12.2.4 L1X SDK

The L1X SDK is designed to provide all of the above plus high-level abstractions on top as a single dependency package. Building upon the foundation laid by the L1X SDK, developers can harness its comprehensive capabilities as a single, cohesive dependency package. The L1X SDK is meticulously crafted to offer not only the aforementioned functionalities but also high-level abstractions that significantly enhance the development experience. This holistic approach is geared towards delivering a seamless development experience, promoting code maintainability, and facilitating collaboration within the blockchain development community.

The L1X SDK comes equipped with essential functionalities that significantly contribute to its ease of use and efficiency.

a. **Storage Management**: Developers gain seamless access to the L1X blockchain's storage layer through three key functions. The *storage_write* function facilitates the insertion of data into the L1X blockchain storage, associating it with a specified key. On the other hand, the *storage_remove* function empowers developers to effortlessly delete data from the L1X blockchain storage, targeting entries linked to a designated key. Complementing these, the *storage_read* function enables the retrieval of data from the L1X blockchain storage by providing a specific key, returning the information in an optional vector format. This trio of functions not only simplifies storage interactions but also enhances the overall versatility and user-friendliness of the L1X SDK in handling on-chain data.

b. **Input Output Handling**: In addressing the pivotal components of any blockchain operation, the L1X SDK adeptly streamlines the intricate process of input and output management. At the core of this capability are two essential functions. Firstly, the *input* function is designed to seamlessly retrieve input data, presenting developers with a straightforward method to access relevant information vital for the execution of L1X blockchain operations. Complementing this, the *output* function serves as a powerful tool for sending data to the appropriate channels, ensuring a cohesive and well-orchestrated communication flow within the L1X blockchain environment. These functions collectively empower developers to navigate the intricacies of input and output in a manner that is both intuitive and efficient, underscoring the L1X SDK's commitment to simplifying key operational aspects for blockchain developers.

c. **Account Information**: In the dynamic realm of the L1X blockchain development, the L1X SDK distinguishes itself by seamlessly granting developers access to crucial account-related information. Anchored by two integral functions, the L1X SDK simplifies the intricacies of account management. The *current_account_id* function takes center stage, offering developers a streamlined means to retrieve the current account ID, providing essential context for ongoing operations. Complementing this, the *caller_account_id* function proves equally instrumental, enabling developers to access the account ID of the entity initiating the operation. These functions collectively furnish developers with a clear lens into the current and calling account identities, facilitating nuanced decision-making and enhancing the precision of the L1X blockchain interactions. As a testament to the L1X SDK's commitment to user-friendly development, these functionalities contribute to a more intuitive and comprehensive approach to managing account information within the L1X blockchain ecosystem.

d. **Error Handling**: Acknowledging the inherent complexity of blockchain operations, the L1X SDK stands out for its robust error handling mechanisms. Two pivotal functions encapsulate this capability. Firstly, the *abort* function takes center stage by allowing developers to gracefully halt the execution of the current operation, providing a controlled and systematic response to unforeseen circumstances. Complementing this, the *panic* function proves to be a decisive tool for developers, enabling the intentional triggering of a panic state with a customized error message. This dual-functionality not only fortifies the reliability of the L1X blockchain applications by preemptively addressing potential issues but also empowers developers with the tools needed to communicate specific error conditions effectively. In navigating the unpredictable landscape of the L1X blockchain development, these error handling mechanisms within the L1X SDK emerge as invaluable assets, contributing to the creation of resilient and dependable L1X blockchain solutions.

12.2.5 On-Chain Collection

Although a smart contract developer can choose to store data in the native Rust collections, sometimes storing them in an on-chain collection is a much smarter choice. Consider a situation where you have 100K different balance holders in a single fungible token contract. It is unreasonable to load the entire 100K-sized map when you only need to access one of the key-value pairs. This is where on-chain collections come in.

Currently, the following collection types are supported by the L1X SDK:

a. **LookupMap**: The LookupMap introduces a highly efficient map data structure tailored for on-chain utilization. Resembling hashmaps or dictionaries in other programming languages, this feature facilitates the storage of key-value pairs while optimizing memory consumption. Crucially, the LookupMap avoids loading the entire map into memory, opting for on-demand access to individual key-value pairs. This deliberate design minimizes resource requirements and results in significant efficiency improvements, particularly when managing extensive datasets within the L1X blockchain context.

b. **LookupSet**: The LookupSet functionality within the L1X SDK presents a specialized set data structure optimized for on-chain operations. Comparable to sets in various programming languages, this feature stores unique values and offers rapid verification of a value's inclusion in the set. Much like the LookupMap, the LookupSet's optimization for on-chain usage ensures that individual values are accessed as needed, circumventing the need to load the entire set into memory. This approach enhances the efficiency of set-related operations within blockchain scenarios, contributing to streamlined and resource-conscious implementations.

c. **Vector**: An integral component of the L1X SDK, the Vector constitutes a dynamic array data structure designed explicitly for on-chain scenarios. Analogous to lists or dynamic arrays in conventional programming languages, the Vector facilitates the storage of multiple values within an ordered collection. Notably, it prioritizes on-chain efficiency by optimizing operations such as appending items or accessing elements based on their position in the collection. The Vector's distinctive feature lies in its ability to avoid loading the complete array into memory, opting for on-demand access to individual elements. This strategic design choice ensures optimal resource utilization, making the Vector a valuable asset for scenarios demanding efficient array management within the blockchain environment.

12.2.6 L1X VM Core

L1X VM is designed to be able to run ELF files containing eBPF bytecode in a secure and self- contained environment. This is done in two steps:

a. Relocation: Any eBPF file is a relocatable object file, meaning it has specific ELF sections that can be used to relocate .text instructions, .data bytes, and function calls. This is done by inspecting the .rel sections in the object file and following specific logic implemented in l1x-ebpf-runtime crate as follows:

    i. .text: Whenever the eBPF program tries to access a global variable, there is a matching .rel entry that can be used to locate the ld*/st* instruction and change it. All global variables are relocated to the metadata buffer.

    ii. .data: Const values are written to a .rodata.* subsection that is also relocated to the metadata buffer (but with a different offset to the global variables from text).

    iii. Calls: Usually, eBPF does not support calling internal functions, but our variant of eBPF needs to have this capability to be on par with WASM. Calls instructions have src and imm sections. src is used to distinguish between internal and external calls (0 means external, everything else means internal). If it is an internal call, imm points to the callee's offset the .text section. Otherwise, imm has a MurmurHash3-hashed value of the external function's name.

b. Interpretation: rBPF is used as the basis for L1X interpretation logic, yet it has been modified to handle some non-standard extensions such as internal functions.

12.3 Contract Lifecycle in L1X VM

Smart Contract lifecycle on the L1X blockchain follows a structured process comprising three distinct phases. These phases are crucial in ensuring the secure and efficient execution of smart contracts on the L1X blockchain.

12.3.1 Smart Contract Development

The initial phase of contract lifecycle is smart contract development that involves the developers writing the actual smart contract code. Since Rust programming language provides a safe and efficient environment for contract development, it is an excellent choice for developers to write smart contracts on L1X. Resulting smart contracts are robust and less prone to vulnerabilities thus enhancing security of the L1X ecosystem. To simplify the development process and enforce best practices, L1X Smart Contract SDK provides a safe API custom tailored for building smart contracts on L1X.

12.3.2 Building the Smart Contract

After writing the initial code, the smart contract undergoes a series of transformations to optimize and prepare it for deployment on the L1X blockchain.

- **Macro Preprocessing**: Rust's metaprogramming capabilities are leveraged to create high-level abstractions, simplifying development for the L1X platform. Macros help streamline code and improve readability.

- **LLVM Compilation**: The LLVM compiler infrastructure is utilized to compile and optimize the Rust code. This step generates the LLVM Intermediate Representation (IR), which is platform-agnostic and optimized for efficiency.

- **LLVM IR to WASM**: The LLVM IR is further processed to convert it into WebAssembly (WASM), a format widely supported by blockchain platforms, including L1X.

- **WASM to eBPF**: A custom translator takes the generated WASM file and translates it into eBPF bytecode, a format suitable for execution within the L1X ecosystem.

12.3.3 Running on L1X VM

The final product of the development and transformation phases is the ELF (Executable and Linkable Format) file containing eBPF bytecode. This ELF file represents the contract binary that the developer publishes on the L1X blockchain.

- **L1X VM**: The Layer One X Virtual Machine is responsible for executing these smart contracts securely and efficiently. It seamlessly integrates with the L1X blockchain, ensuring that contracts operate as intended. Additionally, the L1X VM handles state updates, ensuring that changes to the blockchain's state are appropriately managed and maintained. Thus, The L1X VM plays a critical role in executing smart contracts, providing a secure and reliable environment in which developers can deploy their contracts and interact with the blockchain.

12.4 L1X VM Cross Contract Calls

The eBPF runtime in the L1X blockchain introduces a powerful capability for smart contracts to interact with one another by calling each other's functions. This inter-contract communication allows for the execution of complex workflows and the sharing of data across contracts. When Smart-Contract A invokes a function in Smart-Contract B, it initiates a process where Smart-Contract B is executed with specific contextual information as detailed below.

12.4.1 Process

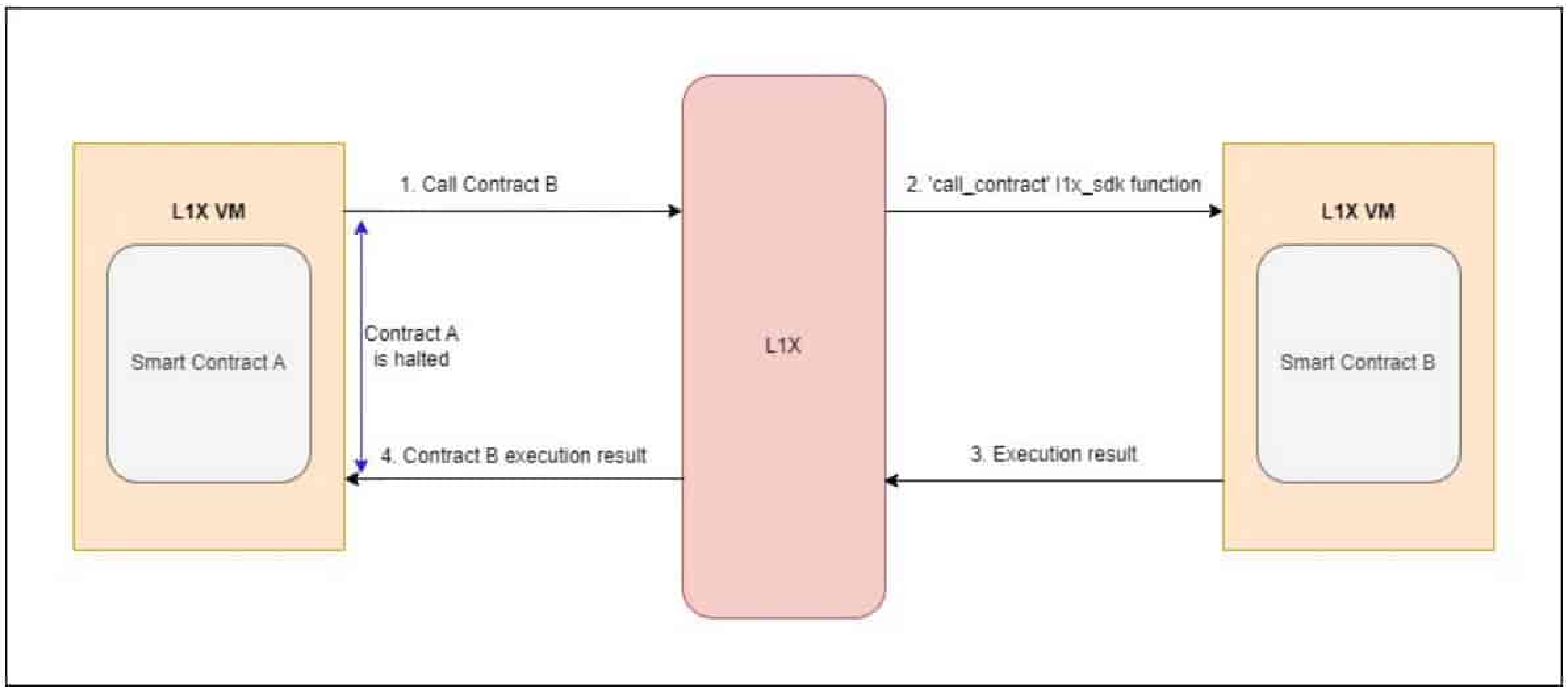


Figure 9 L1X VM Cross Contract Calls

The L1X VM Cross-Contract Call Process involves a series of steps for one smart contract (Contract A) to call and interact with another smart contract (Contract B) within the L1X blockchain ecosystem. Here's a detailed description of each of the four steps in this process:

1. **Request Contract interaction**: In this initial step, Contract A's L1X VM initiates a cross-contract call by invoking the **call_contract** function such as call_contract(contract_B, arguments). It serializes the method name and arguments intended for Contract B into a JSON format, preparing the necessary data for the call.

2. **Contract B Execution**: The serialized call request is then passed from L1X Core to the L1X VM of Contract B. Contract B's L1X VM processes the request and invokes the **call_contract** L1X SDK (Software Development Kit) function, effectively executing the desired method of Contract B with the provided arguments.

3. **Result from Contract B:** Upon completing the execution of the requested function, Contract B's L1X VM generates an execution result. This result, which may include return values or outcomes of the function call, is then transmitted back to L1X Core.

4. **Result Handling in Contract A:** Finally, L1X Core receives the execution result from Contract B's L1X VM in JSON format. L1X Core deserializes this JSON data to extract the execution result, converting it into a usable format that Contract A's

L1X VM can interpret and utilize. Contract A's L1X VM can then process the response, potentially using the result for further actions or logic within the smart contract.

In summary, these four steps in the L1X VM Cross-Contract Call Process enable smart contracts within the L1X blockchain ecosystem to communicate and collaborate effectively. This process ensures that one contract can call functions in another contract, share data, and coordinate actions, enhancing the functionality and versatility of decentralized applications built on the L1X blockchain.

12.4.2 Execution Context

- **Caller's Address:** The **caller_address()** API function within Smart-Contract B's execution context returns the contract instance address of Smart-Contract A. This address identifies the calling contract, enabling Smart-Contract B to track the origin of the invocation.

- **Token Transfer:** The **transfer_from_caller()** API function facilitates the transfer of tokens from Smart-Contract A's balance to Smart-Contract B's balance. This feature allows for the exchange of value between the two contracts, enabling various economic interactions and transactions.

- **Contract Instance Address:** Within Smart-Contract B's execution context, the **contract_instance_address()** API function provides access to Smart-Contract B's contract instance address. This address is essential for addressing and interacting with Smart-Contract B's specific instance, especially in scenarios involving multiple instances of the same contract.

- **Contract Owner Address:** The **contract_owner_address()** API function returns the address of Smart-Contract B's contract owner. This information is valuable for Smart-Contract B to determine ownership and access control rights.

It's important to note that the execution context of Smart-Contract B depends on whether

Smart-Contract A was called in a read-only or read-write context:

- **Read-Only Context:** If Smart-Contract A was invoked in a read-only context, meaning it did not modify its own persistent storage, then Smart-Contract B will also be executed in the same read-only context. In this scenario, Smart-Contract B will not have the ability to modify its own persistent storage either. This restriction ensures data consistency and integrity when interacting with contracts in a read-only manner.

12.4.3 Benefits

The purpose of L1X VM (Layer One X Virtual Machine) cross-contract calls is to enable interaction and communication between different smart contracts within the L1X blockchain ecosystem. These cross-contract calls serve several important purposes:

1. **Decentralized Functionality**: L1X VM cross-contract calls allow smart contracts to call functions in other contracts, decentralizing the execution of code and enabling the creation of complex, distributed application.

2. **Modular Design**: Developers can create modular smart contracts that perform specific functions and then call these contracts from other contracts as needed. This modular design enhances code reusability and maintainability.

3. **Data Sharing**: Cross-contract calls enable smart contracts to share data with each other. This data sharing can be used for a wide range of purposes, including aggregating information, conducting computations, and updating contract states based on external events.

4. **Complex Workflows**: Smart contracts can coordinate with each other to execute complex workflows that involve multiple steps or interactions. For example, one contract might trigger an action in another contract when certain conditions are met

5. **Token Transfers**: Cross-contract calls are essential for transferring tokens or assets between different contracts. This functionality underpins various decentralized finance (DeFi) applications and token ecosystems.

6. **Interoperability**: L1X VM cross-contract calls promote interoperability within the L1X blockchain ecosystem. Different contracts can interact seamlessly, allowing developers to build diverse and interconnected applications.

7. **Resource Efficiency**: Instead of duplicating code and functionality in multiple contracts, cross-contract calls enable contracts to leverage the capabilities of other contracts. This resource-efficient approach reduces code redundancy and saves gas fees.

8. **Enhanced Flexibility**: Developers can create specialized contracts for specific purposes and then combine them in different ways through cross-contract calls. This flexibility fosters innovation and experimentation in the development of decentralized applications.

Thus, L1X VM cross-contract calls play a crucial role in enabling smart contracts to work together, share data, and execute complex operations within the L1X blockchain ecosystem. They empower developers to build sophisticated, decentralized applications that leverage the collective functionality of multiple contracts, enhancing the versatility and capabilities of the blockchain network.

12.5 L1X VM-EVM Cross Contract Calls

L1X VM-EVM (Ethereum Virtual Machine) cross-contract calls enable interoperability and interaction between smart contracts residing on the L1X blockchain and those deployed on the Ethereum network. The process flow for L1X VM-EVM cross-contract calls is a critical aspect of the L1X blockchain ecosystem, allowing for interoperability between different types of smart contracts. This process involves several stages, from client submission to feedback from the Ethereum Virtual Machine (EVM). Let's explore each step in detail.

12.5.1 Process

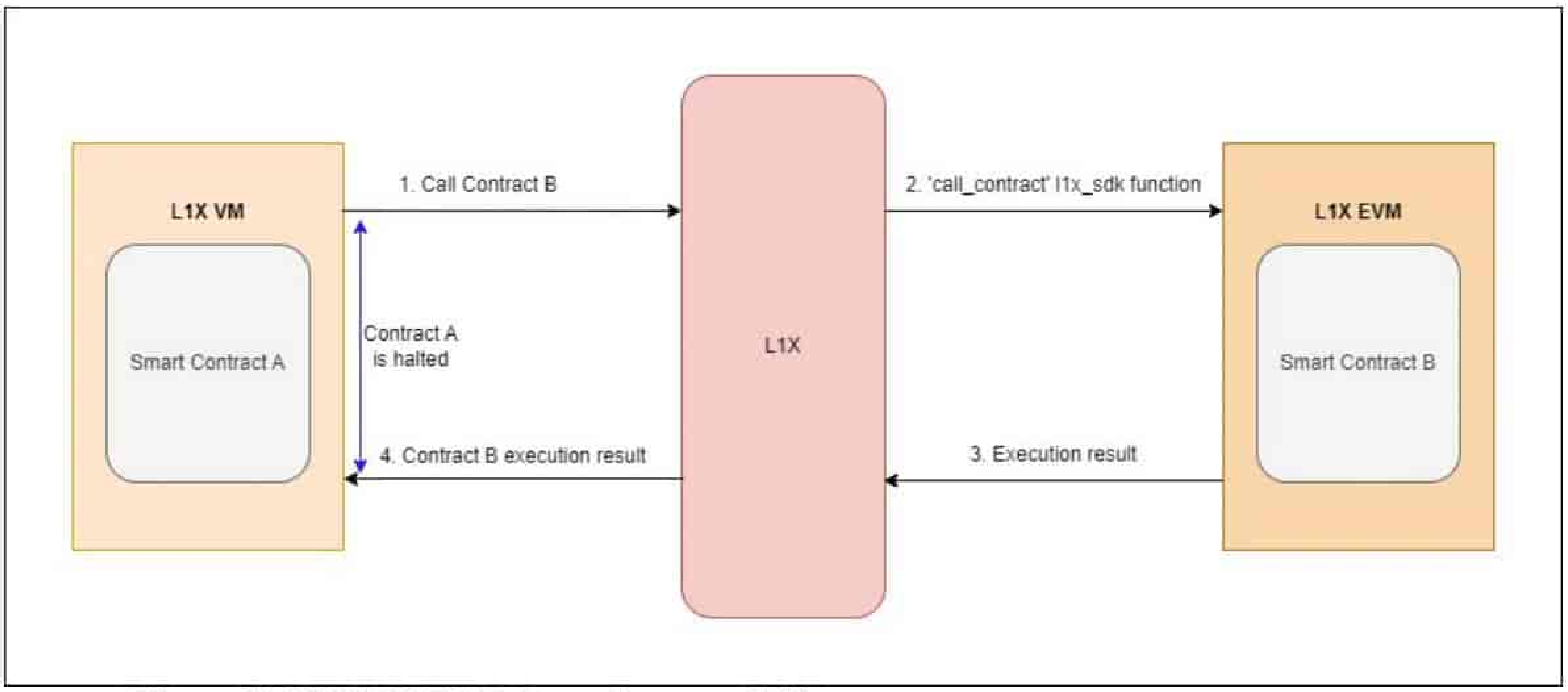


Figure 10 L1X VM-EVM Cross Contract Calls

The L1X VM EVM Cross-Contract Call Process involves a series of steps for one smart contract (Contract A) to call and interact with another smart contract (Contract B) within the L1X blockchain ecosystem, particularly when Contract B is an Ethereum Virtual Machine (EVM) contract. Here's a detailed description of each of the four steps in this process:

1. **Initiate Cross-Contract Call:** In this initial step, Contract A's L1X VM initiates a cross-contract call by invoking the call_contract function such as call_contract(contract_B, arguments). Contract A's L1X VM serializes the method name and arguments intended for Contract B according to the Ethereum Virtual Machine (EVM) Application Binary Interface (ABI) standard. This ensures compatibility with EVM contracts.

2. **Execute Contract B via L1X EVM:** The serialized call request is then passed from L1X Core to the L1X EVM of Contract B. Contract B's L1X EVM processes the request and invokes the call_contract L1X SDK function, effectively executing the desired EVM contract method with the provided arguments, adhering to the EVM ABI standard.

3. **Response from L1X EVM of Smart Contract B:** Upon completing the execution of the requested EVM contract function, Contract B's L1X EVM generates an execution result. This result, which conforms to the EVM ABI standard, is then transmitted back to L1X Core as a response to the cross-contract call.

4. **Result Handling in L1X VM of Smart Contract A:** L1X Core receives the EVM execution result from Contract B's L1X EVM. L1X Core then deserializes the EVM result, converting it into a usable format that Contract A's L1X VM can interpret and utilize for further actions or logic within the smart contract.

The L1X VM EVM Cross-Contract Call Process, thus, facilitates the interaction between smart contracts in the L1X blockchain ecosystem, particularly when one of the contracts is an Ethereum Virtual Machine (EVM) contract. This process ensures that smart contracts can call EVM contract functions, share data, and coordinate actions while adhering to the EVM ABI standard, enhancing the versatility and compatibility of decentralized applications within the L1X ecosystem.

12.5.2 Execution Context

1. **Client Submission**

Client submission initiates the deployment or interaction of smart contracts on the L1X blockchain, serving as the entry point for users to engage with the blockchain and its interconnected contract ecosystem.

- **Initiate Contract Deployment:** The process begins when a client initiates the deployment of a smart contract on the L1X blockchain. This typically involves deploying a new contract or initializing an existing one.

- **Initialize the Contract:** Initialization of the contract involves setting up the initial parameters and state of a smart contract on the L1X blockchain, preparing it for execution and interaction with other contracts in the network.

- **Make a function call to the L1X Core:** The client interacts with the L1X core through an L1X node. This node acts as an interface between the client and the L1X blockchain, facilitating contract deployment and function calls.

2. **Contract Identification by L1X Core**

The L1X core examines the contract being deployed or invoked to determine its type. It categorizes contracts into one of three possible options: L1XVM, EVM, or XTALK. This categorization is crucial for routing and executing the contract appropriately.

3. **Execution via the EVMContractCallTrait**

If the L1X core identifies the contract as an Ethereum Virtual Machine (EVM) contract, it leverages the EVMContractCallTrait to manage its interaction with the EVM for execution.

- **Contract Deployment:** The EVMContractCallTrait provides the necessary functions for contract deployment. The **execute_evm_contract_deployment(args...)** function is responsible for deploying EVM contracts onto the Ethereum Virtual Machine.

- **Contract Call:** For executing functions within EVM contracts, the EVMContractCallTrait offers the **execute_evm_contract_call(args...)** function. This function allows L1X VM to invoke EVM contract methods.

4. **Interaction with the EVMSyncApi**

The EVMSyncApi, a component responsible for synchronizing L1X VM with the Ethereum Virtual Machine, invokes the functions provided by the EVMContractCallTrait, including **execute_evm_contract_deployment** and **execute_evm_contract_call**.

5. **Feedback from EVM**

As the EVM executes the requested contract deployment or function call, it generates results, events, or outcomes. These outcomes and results are then relayed back to the L1X VM, allowing the L1X blockchain to process and handle the results as needed. This feedback loop ensures that the execution of EVM contracts is synchronized and integrated seamlessly into the L1X ecosystem.

12.5.3 Deployment Flow

The deployment flow illustrates the seamless interaction between the Client (Hardhat), RPC Endpoints, and the L1X EVM within the L1X VM-EVM Cross Contract Calls architecture. The diagram showcases the essential steps involved in deploying smart contracts, emphasizing the transmission of critical information through transaction, event, addresses and bytecode.

1. **Client: Hardhat**: The deployment process begins with the client, in this case, Hardhat, which is a popular development environment. Hardhat allows developers to compile, test, and deploy smart contracts efficiently. Developers use Hardhat for writing, compiling, and testing smart contracts. After successful compilation, Hardhat generates artifacts, including ABI (Application Binary Interface) and bytecode, which are essential for interacting with and deploying contracts.

2. **RPC Endpoints**: The RPC (Remote Procedure Call) endpoints act as the connector between the client and the L1X EVM. They facilitate communication and data transfer between the client application and the L1X nodes. Hardhat connects to the L1X network via RPC endpoints. Upon completion of contract development and artifact generation, Hardhat uses RPC to broadcast transactions to the L1X network.

3. **L1X EVM:** The L1X EVM is the runtime environment for executing smart EVM compatible smart contracts on the L1X network. It interprets and executes bytecode, making it one of the core components of the L1X blockchain. If the transaction involves deploying an EVM-compatible smart contract, the L1X EVM executes the contract creation process. The L1X EVM executes the bytecode of the smart contract, which defines the logic and behavior of the contract. After a successful contract deployment, the L1X EVM generates an address for the newly deployed contract, and this address is communicated back to the client. The client needs the contract address to interact with the deployed contract for subsequent transactions and calls.

12.5.4 Benefits

The purpose of L1X VM-EVM (Ethereum Virtual Machine) cross-contract calls is to enable interoperability and interaction between smart contracts residing on the L1X blockchain and those deployed on the Ethereum network. This feature provides several important benefits as listed below.

1. **Interoperability**: L1X VM-EVM cross-contract calls allow smart contracts on the L1X blockchain to communicate with and utilize the functionality of smart contracts on the Ethereum network. This interoperability closes the gap between two distinct blockchain platforms, facilitating seamless data and asset transfer between them.

2. **Access to Ethereum Ecosystem**: By enabling cross-contract calls to Ethereum-based contracts, L1X blockchain users gain access to the vast and mature Ethereum ecosystem. They can leverage existing Ethereum dApps, DeFi protocols, tokens, and services without needing to migrate assets or create separate accounts on the Ethereum network.

3. **Cross-Chain Functionality**: Cross-contract calls enable cross-chain functionality, allowing L1X contracts to trigger actions on the Ethereum network and vice versa. This opens up possibilities for decentralized applications that span multiple blockchains, enhancing the overall functionality of decentralized systems.

4. **Asset Transfers**: L1X VM-EVM cross-contract calls are particularly useful for transferring digital assets (tokens) between L1X and Ethereum. Users can move assets between the two chains securely and transparently, facilitating various financial transactions and interactions.

5. **Diversified Use Cases**: The capability to interact with Ethereum-based smart

contracts broadens the scope of use cases for L1X blockchain applications. It enables L1X-based projects to participate in decentralized finance (DeFi), NFT (Non-Fungible Token) ecosystems, oracle services, and other Ethereum-centric applications.

6. **Enhanced Security and Trust**: Cross-contract calls provide a secure and trusted way for L1X VM to interact with Ethereum. Smart contracts can confidently invoke functions on the Ethereum network, knowing that the execution will be carried out reliably and transparently.

7. **Scalability and Resource Efficiency**: Instead of attempting to replicate the entire Ethereum network on L1X, cross-contract calls leverage the existing Ethereum infrastructure. This approach is more resource-efficient and scalable, as it offloads complex operations to the established Ethereum ecosystem.

8. **Blockchain Connectivity**: Cross-contract calls play a role in traditional blockchain bridge solutions, where tokens and data can be transferred between two or more blockchains. L1X provides users with the ability to transact cross-chain without a bridge is essential for projects that aim to connect and utilize multiple blockchain networks.

In summary, L1X VM-EVM cross-contract calls fosters interoperability, enhances functionality, and provides access to the broader blockchain ecosystem. It allows for seamless integration between L1X and Ethereum, enabling a wide range of use cases, asset transfers, and cross-chain interactions while maintaining security and efficiency.

12.6 Transaction Compilation and Execution Process

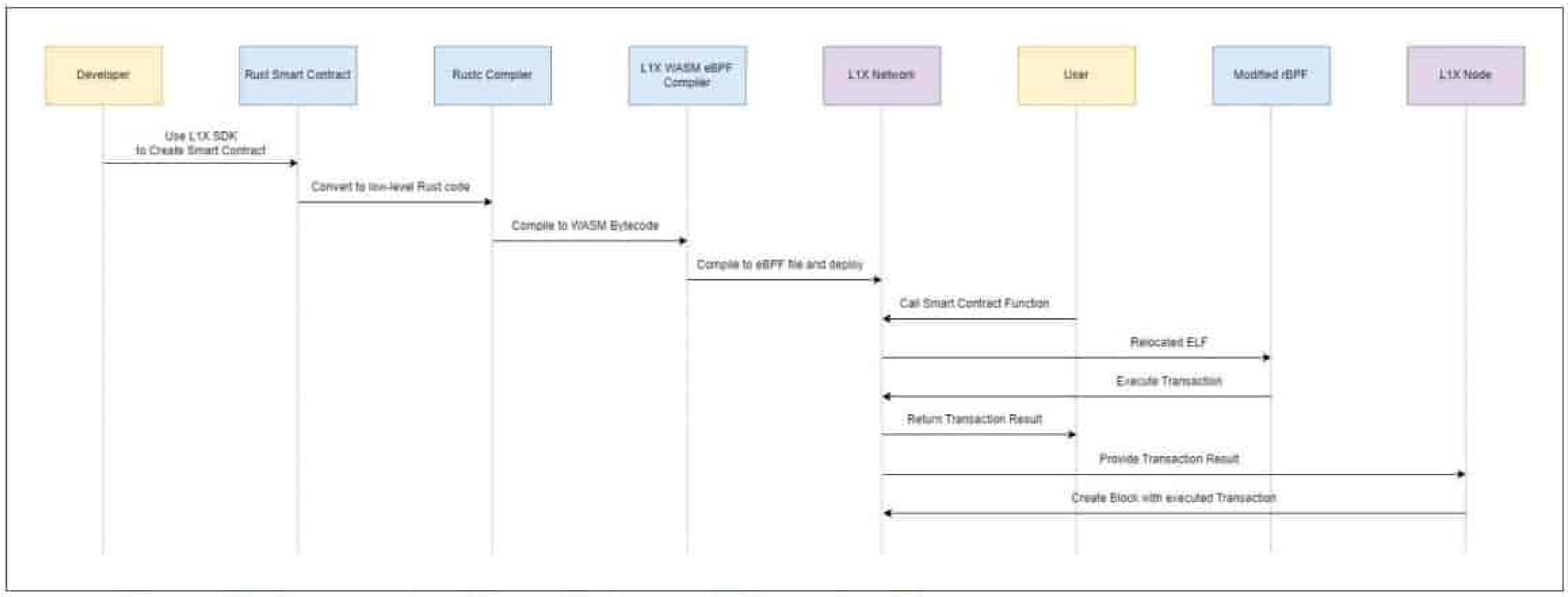


Figure 11 Transaction Compilation and Execution Process

1. L1X SDK is available to the developer to create smart contracts in the RUST programming language.
2. The Rust smart contracts contain L1X SDK macros that are converted to low-level Rust code using L1X System calls.
3. Rustc compiler compiles this low-level rust code into WASM bytecode.
4. L1X WASM eBPF Compiler compiles the WASM bytecode to eBPF Executable and Linkable Format i.e., eBPF ELF file.
5. Thus, the smart contract is deployed on the L1X blockchain as an eBPF ELF file.
6. Whenever a user calls any function of this deployed smart contract, ELF is relocated.
7. Modified rBPF acts as the interpreter which takes required details from on-chain trie and execution context to execute the transaction.
8. The result of the transaction execution is produced by the rBPF.
9. L1X core node creates the block containing the executed transaction.

12.7 VM Fundamental Components

The L1X Virtual Machine encompasses several fundamental components that contribute to its robustness and efficiency. These components can be categorized into libraries, custom optimization passes, and system calls.

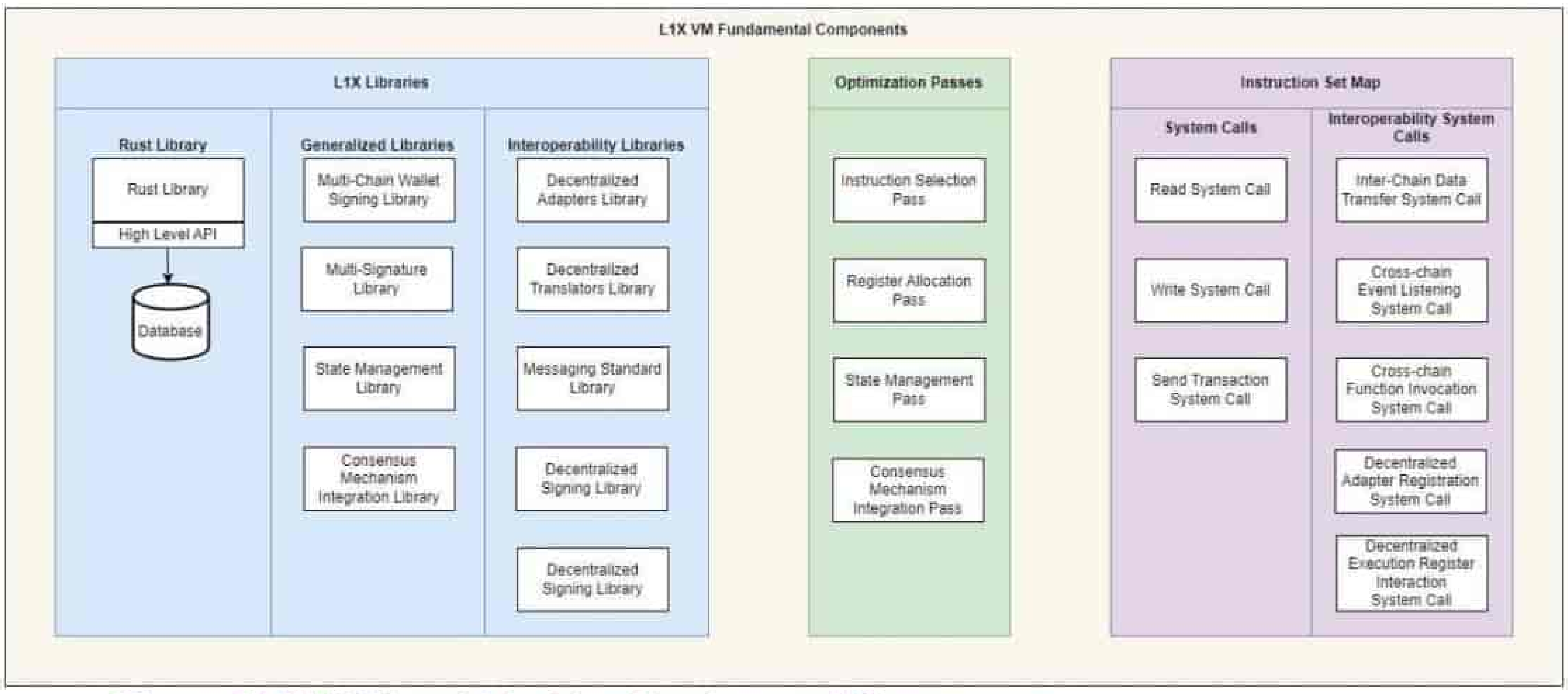


Figure 12 L1X Virtual Machine Fundamental Components

12.7.1 Libraries

The libraries play a crucial role in supporting various functionalities within the L1X Virtual Machine. The multi-chain wallet signing library enables secure signing of cross-chain transactions, facilitating seamless interaction between L1X and other chains. The multi-signature library is utilized for implementing multi-signature transactions, enhancing security and accommodating cross-chain transaction requirements. The state management library efficiently handles the storage, retrieval, and modification of data on the L1X blockchain. Additionally, the consensus mechanism integration library establishes a clear interface between the L1X VM and the underlying consensus process, ensuring accurate transaction execution and updates to the blockchain state.

12.7.2 Interoperability Libraries

Interoperability libraries further enhance the L1X Virtual Machine's capabilities by enabling secure communication and interaction with other connected blockchains. The decentralized adapters library provides functions and interfaces for implementing and managing decentralized adapters, facilitating seamless connectivity. The decentralized translator's library assists in translating messages between different blockchains or applications within the same blockchain. The messaging standard library implements secure and efficient communication standards for the Interoperability Stack. The decentralized signing library ensures secure and authenticated communication between components. Lastly, the decentralized execution register library guarantees deterministic execution of transactions and prevents double-spending across multiple blockchains.

12.7.3 Custom Fundamental Optimization Passes

Custom fundamental optimization passes optimize various aspects of the L1X Virtual Machine. The instruction selection pass enhances performance and compatibility by optimizing the selection of eBPF instructions. The register allocation pass minimizes resource usage by optimizing register allocation within the eBPF bytecode. The state management pass optimizes the handling of state- related operations, improving efficiency and mitigating potential issues. The consensus mechanism integration pass ensures compliance with the requirements and constraints of the L1X consensus mechanism.

12.7.4 System Calls

System calls provide essential functionalities for smart contracts running on the L1X Virtual Machine. The read system call allows contracts to retrieve data from the L1X blockchain state, while the write system call enables writing data to the blockchain. The send transaction system call initiates transactions on the L1X blockchain or other connected chains.

12.7.5 Interoperability System Calls

Interoperability system calls facilitate seamless communication and interaction with other blockchains, including inter-chain data transfer, cross-chain event listening, cross-chain function invocation, decentralized adapter registration, and decentralized execution register interaction.

These fundamental components collectively contribute to the robustness, efficiency, and interoperability of the L1X Virtual Machine, empowering developers to build complex and innovative decentralized applications on the Layer One X blockchain platform.

12.8 Performance

L1X Virtual Machine is a critical component of the L1X blockchain, enabling the execution of Rust smart contracts on the blockchain. As a result, creating the right virtual machine is essential to ensure optimal performance, efficient memory usage, and ease of use. This section focuses on a comparative study of two approaches, the WebAssembly (WASM) interpreter and the eBPF runtime in user space, and their performance in running smart contracts.

Here, the performance of WASM and eBPF for running smart contracts is compared. The metrics for comparison include the time taken for the contract to execute, the amount of memory used, and the lines of code required for the execution.

eBPF outperforms WASM in both execution time and memory usage. The eBPF programs execute faster than the WASM programs, with an average speed up of 2.5x. The eBPF programs also used less memory than the WASM programs, with an average memory usage reduction of 1.7x. The number of lines of code required was also significantly less for eBPF, averaging 20 compared to 60 for WASM.

Metrics	eBPF	WASM
Execution time	0.003 seconds	0.05 seconds
Memory Usage	6 MB	20 MB
Lines of code	20	60

Table 3 WASM vs eBPF

The study's results indicate that eBPF offers superior performance, memory usage, and ease of use compared to WASM when it comes to running smart contracts. The primary reason for this is eBPF's kernel-based nature, which allows it to leverage the full power of the kernel and use the entire kernel's resources. In contrast, WASM is a low-level bytecode format designed for web browsers and has limitations when it comes to memory usage and performance. The interpreter for WASM is also slower than the eBPF runtime.

eBPF's design provides several advantages over WASM. First, the eBPF runtime is more efficient in memory usage, enabling it to execute smart contracts using significantly less memory than WASM. This is crucial in environments where memory usage is a concern. Second, the eBPF runtime executes smart contracts significantly faster than the WASM interpreter, allowing the smart contracts to execute more quickly.

Additionally, eBPF's design allows it to be more user-friendly than WASM. It requires fewer lines of code to execute smart contracts, which reduces development time and cost. This can be especially important for companies looking to develop smart contracts for the blockchain, as it allows them to focus on the smart contract's functionality rather than the implementation details.

12.8.1 Stack-based vs Register-based Virtual Machine

Stack-based execution involves the use of a stack data structure to manage program execution. The stack contains a sequence of instructions and data, and the program execution involves manipulating the stack through push and pop operations. Stack-based execution is relatively easy to implement and is therefore popular. On the other hand, register-based execution involves the use of registers to store and manipulate data during program execution. Registers are temporary storage locations within the CPU, and their use can result in faster program execution due to the reduced need to access memory.

The research study shows that eBPF has a higher performance than WASM, primarily due to its register-based architecture. The eBPF's use of registers provides faster access to data during program execution, resulting in improved performance. Here is a comparison of stack-based WASM and register-based eBPF virtual machines.

Metrics	**Stack-based WASM**	**Register-based eBPF**
Memory	High	Low
Instruction Execution Speed	Low	High
Data Access	Slow	Fast
Instruction Set Complexity	Simple	Complex

Table 4 Stack-based vs Register-based Virtual Machines

As shown in the table, register-based eBPF virtual machines have several advantages over stack-based WASM, including faster instruction execution speed, faster access to data, and lower memory usage. However, eBPF's instruction set is more complex than WASM, which can make it more difficult to implement.

In conclusion, eBPF is a more suitable technology for running smart contracts on the blockchain than WASM. eBPF provides superior performance, lower memory usage, and ease of use when it comes to executing smart contracts. Owing to these reasons L1X Virtual Machine harnesses the benefits of eBPF to provide high-speed performance.

13. AVDR

This whitepaper introduces the AVDR (Access Version Data Retention) framework, a cutting-edge solution designed to address the intricate needs of secure content deployment, version control, and data retention within the L1X blockchain ecosystems. This framework presents a meticulous approach to managing digital assets, specifically tailored to the evolving landscapes of NFTs and gaming.

13.1 AVDR Fundamental Concepts

While the core principles of blockchain are rooted in decentralization and security, ensuring proper access, version, and data control is essential for maintaining the integrity of the system. Let's us have a quick overlook at these core concepts of AVDR.

13.1.1 Access Control

Access control refers to the management of permissions and restrictions on who can access specific resources within the L1X blockchain network. In a decentralized environment, it becomes crucial to define and enforce access policies to prevent unauthorized users from altering or accessing sensitive data. Access control in the context of NFTs and gaming involves defining and enforcing permissions for various actions, such as creating, transferring, or interacting with digital assets. Smart contracts play a pivotal role, ensuring that only authorized entities can participate in the creation, sale, or utilization of NFTs and in-game items.

13.1.2 Version Control

Version control ensures the consistency and uniqueness of digital assets within the L1X blockchain. In NFTs and gaming, this is crucial for maintaining the scarcity and value of virtual items, as well as for tracking the evolution of in-game content.

13.1.3 Data Control

Data control focuses on securing the storage and retrieval of data related to NFTs and gaming assets within the L1X blockchain. Cryptographic techniques play a pivotal role in ensuring the integrity and transparency of virtual asset information.

As the L1X blockchain network aims to reshape the landscape of NFTs and gaming, the implementation of robust access control, version control, and data control mechanisms is crucial.

13.2 Need for AVDR

In the rapidly evolving landscape of blockchain technology, the imperative for robust solutions has given rise to the AVDR (Access Version Data Retention) framework. This pressing need arises from the intricate challenges of securing access, managing versions, and retaining data integrity, ushering in an era of enhanced reliability and innovation within decentralized systems. Below content gives a glimpse of the need for AVDR.

1. **Asset Protection**: Safeguarding NFTs and in-game assets from unauthorized access or manipulation.
2. **Monetary Transactions**: Regulating access to transactions involving valuable virtual assets.
3. **User Privacy**: Protecting user data and identity within the virtual environment.
4. **Asset Authenticity**: Confirming the uniqueness and origin of NFTs and in-game items.
5. **Content Updates**: Coordinating and validating changes to virtual assets and game content.
6. **Historical Tracking**: Preserving the chronological order of asset creation and ownership.
7. **Tamper-Proof Records**: Protecting against unauthorized modifications or tampering of virtual asset data.
8. **Transparency and Trust**: Maintaining a transparent and trustworthy record of asset ownership and transactions.
9. **Interoperability**: Facilitating secure data exchange between different gaming platforms and NFT marketplaces.

By addressing the specific needs and challenges of these domains, L1X paves the way for a secure, transparent, and innovative virtual future.

13.3 AVDR Workflow

In this section, we delve into the intricacies of AVDR workflow.

13.3.1 Content Deployment

When an owner deploys new content within the AVDR framework, a seamless process is initiated:

a. **Initializing Data Storage**: A new instance of Data Storage is created, dedicated to storing the deployed content securely.
b. **Initializing Version Control**: Simultaneously, a new instance of Version Control is initialized, utilizing the address Content Identifier (CID) of the Data Storage instance. This ensures an intrinsic link between the content and its versioning metadata.

13.3.2 Version Submission

When the owner submits a new version of the content, the AVDR framework ensures a structured approach:

a. **Initializing Data Storage**: A new instance of Data Storage is created to accommodate the updated content securely.
b. **Version Control Update**: If the owner possesses write permissions for access control, the address of the newly created Data Storage instance is submitted for Version Control, maintaining a meticulous record of version history.

13.3.3 Instance Structure

For deployed instances within the AVDR framework, a streamlined structure is maintained:

a. **Single Access Control Instance**: Each deployed content has a single Access Control instance, managing permissions and access policies.
b. **Version Control and Data Storage Instances**: For each content, there is a singular Version Control instance, documenting the versioning history, alongside multiple Data Storage instances based on the different content versions.

13.3.4 Content Sharing

On the request to share content from a user, a structured approach is adopted by the AVDR framework:

a. **Access Request Initialization**: Users commence the content sharing journey by submitting a request to the data owner, providing their public key and the Content Identifier (CID) of the desired data. The Access Control Instance (ACI) takes charge, appending the request to the pending list and assigning a unique Request IDX to the user.

b. **Owner-Driven Approval Sequence**: Navigating the ACI, the data owner retrieves pending requests, eliciting a vector of pending requests in response. Armed with the content corresponding to the requested CID, the data owner encrypts it and solicits Data Storage to initiate a new instance. Data Storage executes the required tasks and communicates the CID instance address.

c. **Seamless Access Control Approval Process**: The data owner proceeds to the ACI to approve the pending request, subject to approval permission verification. The ACI orchestrates a series of actions, transitioning the request from the pending list to the shared list and updating the status to approved. A success response is conveyed back to the data owner.

d. **Version Control Update for Shared Content**: Post-approval, the data owner updates the Version Control Instance by submitting the new version of the content with the new CID, providing a comprehensive view of the shared content's versioning.

e. **User-Centric Request Status Tracking**: Users gain insights into their request's status by utilizing the Request IDX. Upon approval, users effortlessly retrieve the CID for the shared content from the ACI.

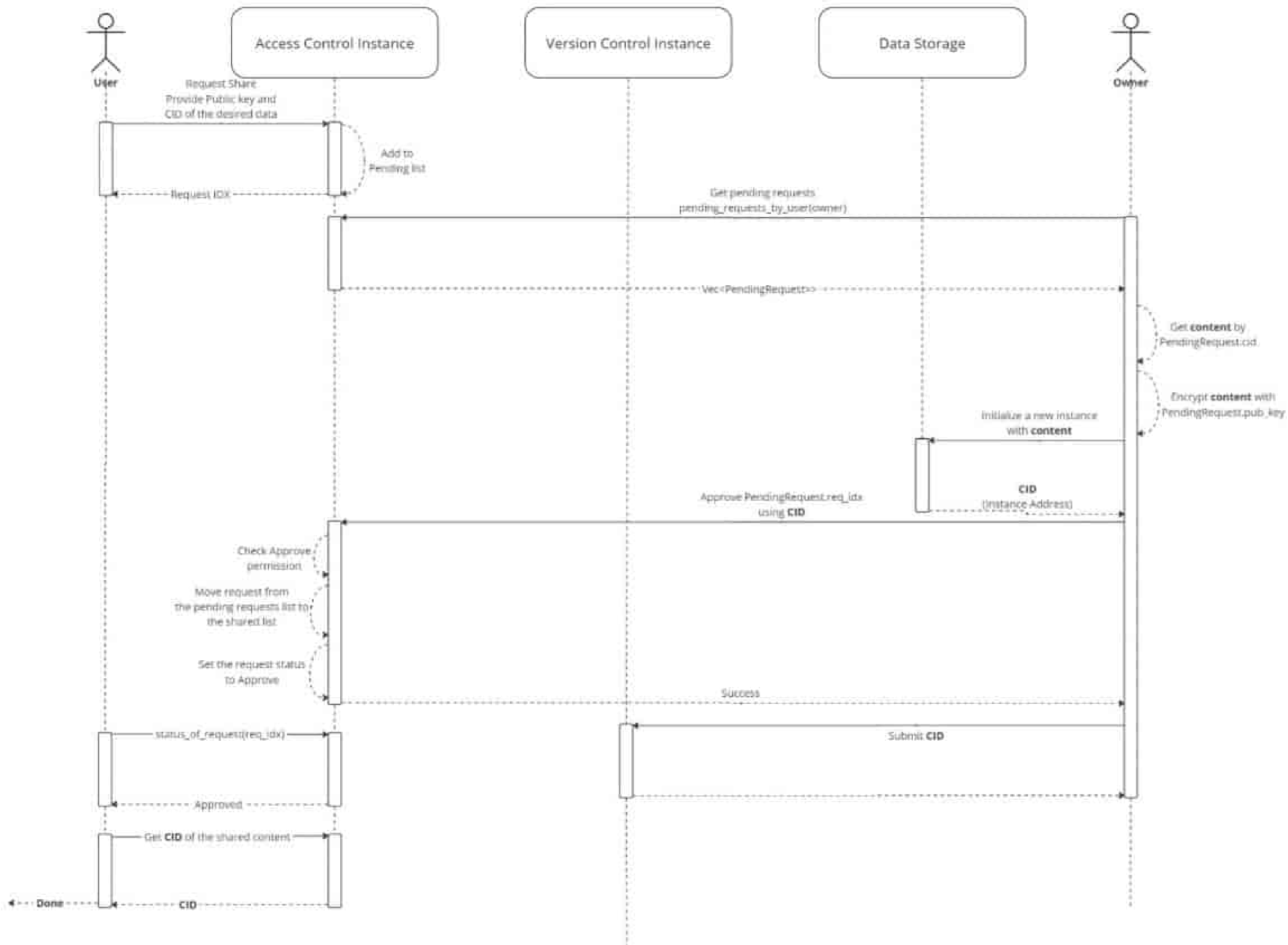


Figure 13 AVDR Content Sharing Workflow

13.3.5 Content Removal

When the owner decides to remove a version of the content, AVDR ensures a secure process. If the owner possesses the necessary permissions, the framework checks and validates whether the owner has the authority to remove a specific version of the content.

13.4 Benefits

The AVDR framework brings forth a myriad of technical advantages tailored specifically for NFTs and gaming. From bolstering security and seamless version integration to providing efficient data organization and dynamic content curation, these benefits collectively underscore the framework's pivotal role in shaping the secure, collaborative, and evolving landscape of the L1X blockchain-powered virtual assets and gaming experiences. The subsequent section highlights the benefits of the AVDR framework.

1. **Enhanced Security and Isolation**: Each deployed content item enjoys a dedicated Data Storage instance, ensuring secure and isolated storage. This feature minimizes the risk of unauthorized access or tampering, enhancing the overall security of digital assets.

2. **Seamless Version Integration**: The AVDR framework facilitates the submission of new versions by initializing separate Data Storage instances for each iteration. By linking these instances to the associated Version Control, it ensures a smooth integration of updated content, preserving the history and allowing for iterative improvements.

3. **Granular Access Control Management**: Deployed instances are governed by a single Access Control instance, streamlining permission management. This feature simplifies access control while providing granular control over user permissions for NFTs and gaming assets.

4. **Efficient Data Organization**: The AVDR framework organizes deployed instances efficiently with a single Access Control instance, a single Version Control instance per content, and multiple Data Storage instances for various versions. This structure optimizes data organization, reducing redundancy and ensuring a clear and organized representation of content.

5. **Iterative Versioning for Collaboration**: AVDR supports iterative versioning by allowing creators to submit new versions seamlessly. This encourages collaboration in gaming scenarios and enables the continuous improvement of NFTs, fostering a dynamic and evolving virtual environment.

6. **Comprehensive Version History:** The framework maintains a comprehensive version history by associating each content item with a single Version Control instance. This feature ensures that creators have a clear and accessible record of all versions, facilitating traceability and transparency in the evolution of digital assets.

7. **Controlled Content Removal**: AVDR ensures controlled content management by

conducting permission checks when an owner wants to remove a version. This feature prevents unauthorized removal, preserving content integrity and allowing creators to manage their assets with precision.

8. **Dynamic Content Curation**: The ability to remove versions based on permissions empowers content creators to dynamically curate their digital assets. This feature supports the evolving nature of NFTs and gaming content by allowing controlled pruning of versions.

In summary, the AVDR framework brings forth a multitude of technical benefits, ranging from enhanced security and efficient data organization to facilitating collaboration and dynamic content curation. By addressing access, version, and data retention in a structured manner, it represents a powerful tool for blockchain professionals navigating the intricacies of NFTs and gaming ecosystems.

13.5 Use Cases

As AVDR framework emerges as a cornerstone solution, addressing multifaceted challenges across diverse blockchain applications, AVDR's use cases permeate various sectors, showcasing its adaptability and significance in shaping the future of decentralized systems.

1. **NFT Marketplaces**: Controlling access to minting, selling, and transferring NFTs.
2. **Gaming Ecosystems**: Regulating in-game purchases, trades, and interactions based on user roles.
3. **Virtual Real Estate**: Managing access to virtual spaces and assets within decentralized virtual worlds.
4. **Limited Edition NFTs**: Ensuring the scarcity and uniqueness of limited edition digital collectibles.
5. **Game Asset Updates**: Managing updates to in-game items and characters while maintaining backward compatibility.
6. **Digital Art Evolution**: Tracking and proving the evolution of digital artworks over time.
7. **NFT Ownership Records**: Ensuring the integrity of ownership records for NFTs.
8. **Decentralized Gaming Platforms**: Securing user and asset data in decentralized gaming

ecosystems.

9. **Cross-Platform Compatibility**: Enabling secure data exchange for assets used across multiple gaming platforms.

As the realms of NFTs and gaming evolve, the symbiotic relationship between access control, version control, and data control proves paramount. AVDR collectively define the integrity and innovation within decentralized virtual spaces. In essence, AVDR not only regulate and secure but also propel the dynamic and interconnected future of the L1X blockchain-powered NFTs and gaming ecosystems.

14. DB Driver Abstraction

The L1X blockchain network offers a novel approach to database integration by providing DB Driver Abstraction. This abstraction layer allows developers to seamlessly integrate one of three database systems: RocksDB, PostgreSQL, or Cassandra. By providing this flexibility, L1X empowers developers to tailor their blockchain solutions according to specific use cases and scalability requirements. This paper delves into the significance of DB Driver Abstraction, justifies the selection of supported databases, and outlines the advantages of this approach. Furthermore, it explores scenarios where each database might be preferable, catering to the diverse needs of developers and investors in the blockchain ecosystem.

14.1 DB Driver Fundamental Concept

DB Driver Abstraction serves as a crucial component within the architecture of the L1X network. It abstracts the underlying database implementation from the application layer, enabling developers to interact with the database through a unified interface. This decoupling facilitates portability, as applications remain agnostic to the specific database being utilized. Consequently, developers can seamlessly switch between supported databases without necessitating significant code changes or disruptions to the blockchain network.

14.2 Need for DB Driver Abstraction

In blockchain networks, efficient and scalable database management is crucial for ensuring high performance, data integrity, and resilience. Traditional blockchain networks often lock developers into using specific database systems, limiting flexibility and adaptability. L1X aims to break this mold by introducing DB Driver Abstraction, a feature that empowers developers with the flexibility to choose between RocksDB, PostgreSQL, and Cassandra for database integration.

14.3 Database Diversity

The diversity of database options in the L1X network, including RocksDB, Postgres, and Cassandra, is driven by their unique strengths and their aptness for various use cases in blockchain applications.

14.3.1 Postgres

With its robust relational model and extensive feature set, Postgres offers versatility for a wide range of blockchain applications. Its support for complex queries, ACID compliance, and proven scalability make it an attractive choice for applications requiring sophisticated data management, such as identity verification systems or decentralized social networks.

14.3.2 RocksDB

Renowned for its high-performance, low-latency characteristics, RocksDB is particularly well-suited for scenarios demanding rapid data ingestion and retrieval. Its efficient storage engine excels in applications requiring real-time processing, such as high-frequency trading platforms or decentralized exchanges.

14.3.3 Cassandra

Designed for distributed, fault-tolerant environments, Cassandra excels in scenarios demanding scalability and resilience. Its decentralized architecture and eventual consistency model make it ideal for applications requiring massive scalability, such as decentralized storage networks or IoT data management platforms.

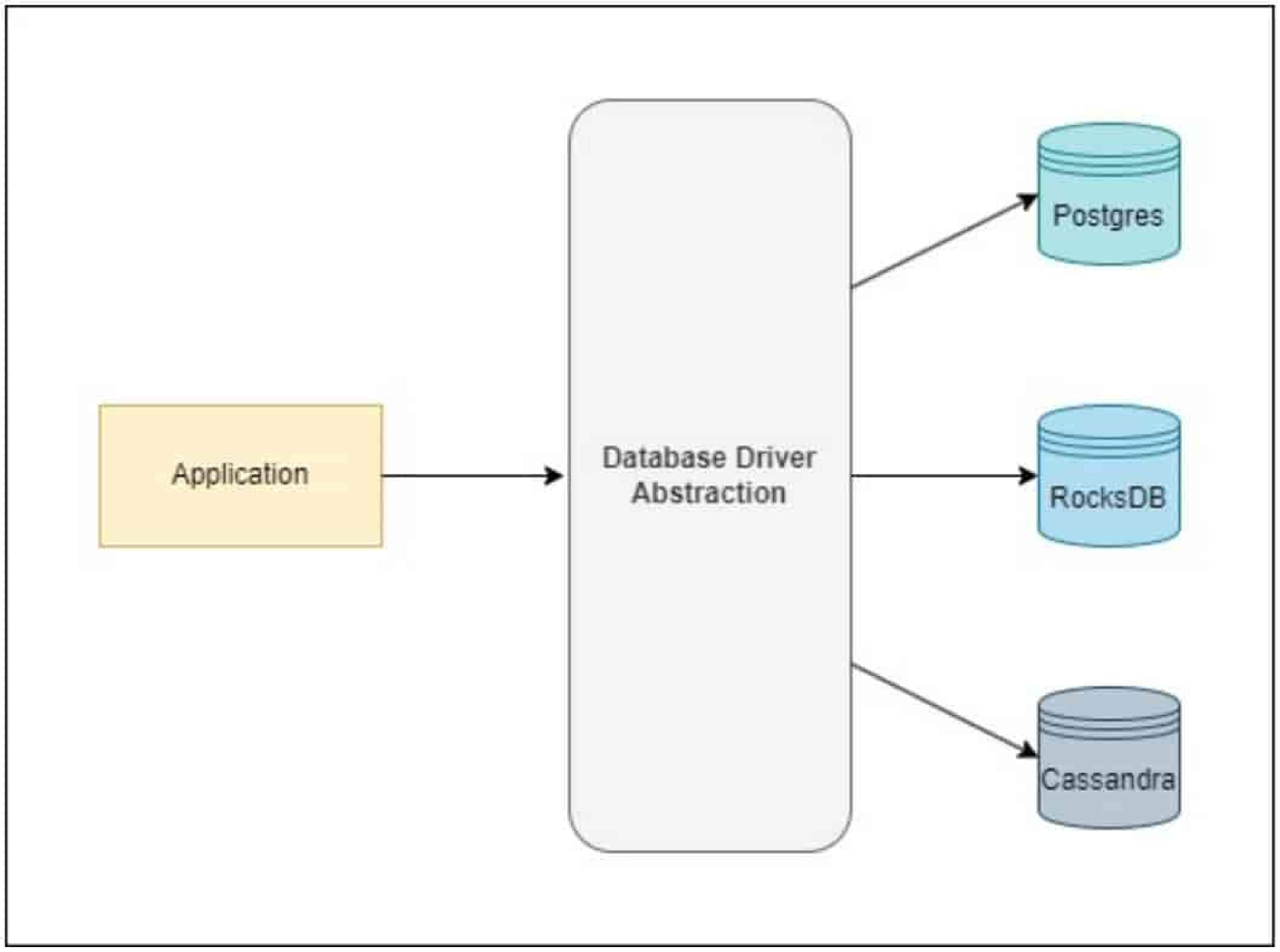


Figure 14 DB Driver Abstraction

14.4 Benefits

The integration of DB Driver Abstraction within the L1X network confers several noteworthy advantages:

1. **Flexibility**: Developers can choose the database best suited to their application's requirements, fostering innovation and adaptability in blockchain development.

2. **Portability**: Applications remain agnostic to the underlying database, facilitating seamless migration and interoperability across different database implementations.

3. **Scalability**: By supporting multiple databases, L1X enables scalability at both the application and database layers, accommodating diverse use cases without compromising performance.

4. **Future-Proofing**: As the landscape of blockchain technology evolves, the flexibility

offered by DB Driver Abstraction ensures that the L1X network remains resilient to emerging challenges and technological advancements.

14.5 Use cases

The suitability of each database for specific scenarios within blockchain applications can be delineated as follows:

1. **High Throughput Transactions**: RocksDB emerges as the preferred choice for applications necessitating rapid transaction processing, such as cryptocurrency exchanges or payment gateways.

2. **Complex Data Queries**: Postgres is well-suited for applications requiring complex data queries and structured data management, including identity management systems or regulatory compliance platforms.

3. **Massive Scalability**: Cassandra offers unparalleled scalability for applications dealing with vast amounts of data, such as decentralized storage solutions or IoT networks.

The incorporation of DB Driver Abstraction within the L1X network represents a paradigm shift in blockchain architecture, offering unprecedented flexibility and scalability to developers and investors alike. By embracing a diverse array of databases, L1X paves the way for innovative blockchain applications tailored to specific use cases. As the blockchain ecosystem continues to evolve, DB Driver Abstraction stands as a testament to L1X's commitment to fostering innovation, interoperability, and scalability in decentralized systems.

15. Oracle System

In the realm of the L1X blockchain network, X-Talk functions as an Oracle System, facilitating the seamless integration of Web2 websites with smart contracts. Through X-Talk, data can be transferred from off-chain to on-chain environments, expanding the horizons of blockchain applications. This integration underscores a strategic move towards bolstering interoperability, data accessibility, and scalability within the L1X blockchain ecosystem.

15.1 Oracle System: A Gateway to Integration

The X-Talk serves as a pivotal component within the L1X network, acting as an oracle system between off-chain data sources and on-chain smart contracts. This integration facilitates real-time data retrieval and communication between web2 platforms and blockchain applications, enhancing interoperability and expanding the utility of decentralized solutions. The incorporation of X-Talk as an oracle system within the L1X network is driven by several compelling reasons:

1. **Data Accessibility**: By enabling the transfer of data from web2 websites to smart contracts, L1X fosters a richer and more dynamic interaction between users and blockchain applications.

2. **Interoperability**: The X-Talk enhances interoperability by allowing smart contracts to interact with external data sources, enabling complex functionalities and use cases that transcend the limitations of blockchain data alone.

3. **Scalability**: By offloading certain data processing tasks to web2 platforms, the Oracle System contributes to scalability, ensuring optimal performance and efficiency within the L1X network.

15.2 Need for Oracle Systems

While blockchain technology offers unparalleled security, transparency, and decentralization, it often operates within a closed environment that has limited access to external data. This limitation hinders the functionality, versatility, and real-world applicability of blockchain applications. X-Talk, as as oracle system, serves as a conduit for integrating external data feeds from web2 websites, enabling smart contracts to access and interact with real-time information. This integration enhances the interoperability of blockchain applications, expands their utility across diverse use cases, and fosters a richer and more dynamic user experience. Moreover, the Oracle System facilitates the transfer of data from traditional web platforms to the blockchain, paving the way for innovative solutions in decentralized finance, supply chain management, identity verification, and more. Thus, the need for an Oracle System in the L1X network is pivotal in unlocking the full potential of blockchain technology by closing the divide between decentralized systems and the broader digital landscape.

15.3 Oracle System Workflow

Integrating web2 website data with smart contracts in the L1X network via the X-Talk Oracle System involves a structured two-step process. Below, we delineate each step, detailing the necessary actions developers must take to execute this integration seamlessly.

15.3.1 Smart Contract Implementation

Developers initiate the process by implementing the smart contract, say for example OracleStorage, tailored to store data fetched from web2 websites. This entails the following key components:

i. **Define Smart Contract Struct**

A struct is declared within the OracleStorage smart contract to encapsulate the data fetched from web2 websites. This struct typically comprises a lookup map consisting of a string as a key (Global Transaction ID) and a corresponding value representing the generic data format obtained from web2.

ii. **Implement Functions**

Several functions are incorporated into the smart contract to manage data storage and retrieval:

- **Load Function**: Initializes the smart contract and loads its content.
- **Save Function**: Saves the state of the smart contract.
- **New Function**: Initiates the smart contract.
- **Save_data Function**: Called by X-Talk to store event data in the smart contract.
- **Getter Function**: Allows users to retrieve data associated with a specific Global Transaction ID.

15.3.2 Register Web2 Website on Source Registry

Following the successful implementation of the smart contract, developers proceed to register their web2 websites on the source registry, enabling the X-Talk Oracle System to fetch data from these websites. This registration process involves the following steps:

i. **Configure Website Details**

Developers configure essential information about their web2 websites, including the website address (URL), the number of arguments in the URL, and the possible values for each argument.

ii. **Register the Smart Contract on X-Talk**

To enable X-Talk to listen to the specified web2 website and fetch data accordingly, developers register the OracleStorage smart contract on X-Talk. This registration typically includes the following parameters:

- **Flow Contract Address**: The address of the user's smart contract where the fetched data will be stored.
- **Time Interval**: Specifies the periodicity of the Oracle System in fetching data from the web2 website.

- **Oracle ID**: An identifier for the API.
- **API URL**: The URL of the web2 website, with placeholders denoted by curly braces ({}) indicating where arguments will be inserted.
- **Function Arguments**: Defines the possible values for the arguments in the URL.

By diligently following this two-step process, developers can seamlessly integrate web2 website data with smart contracts in the L1X network using the Oracle System.

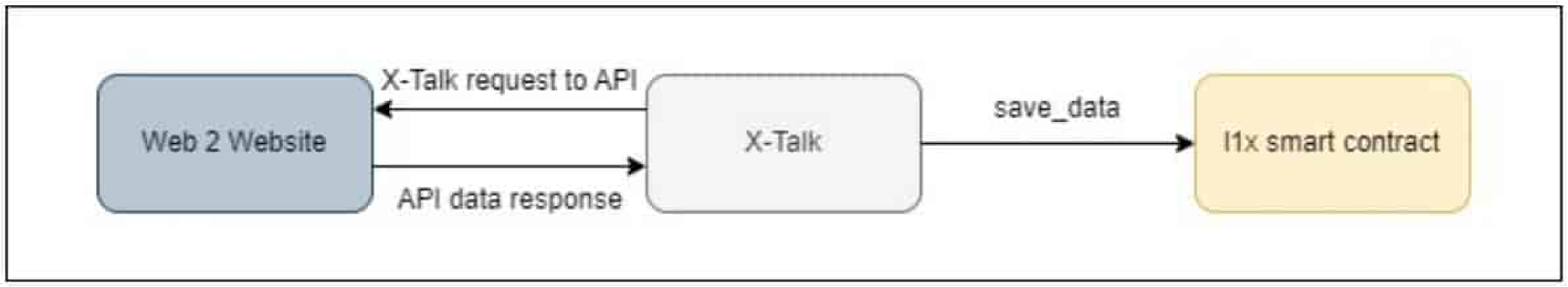


Figure 15 Oracle System

15.4 Benefits

The integration of Oracle Systems within the L1X network offers numerous advantages:

1. **Data Versatility**: Oracle Systems support the integration of diverse data types and sources, including APIs, IoT devices, and external databases, providing developers with unparalleled flexibility in designing blockchain applications.
2. **Real-Time Updates**: By facilitating real-time data feeds to smart contracts, Oracle Systems enable blockchain applications to respond swiftly to changing conditions and events, enhancing their responsiveness and utility.
3. **Interoperability**: Oracle Systems promote interoperability between Web2 and blockchain platforms, fostering collaboration and synergy between traditional web applications and decentralized systems.
4. **Scalability**: By offloading data processing tasks to external Oracle Systems, blockchain networks can achieve greater scalability and efficiency, mitigating congestion and

latency issues associated with on-chain data processing.

15.5 Use Cases

The Oracle System's versatility opens up a myriad of applications and use cases within the L1X network:

1. **Decentralized Finance (DeFi)**: Integration with financial data feeds from web2 platforms can enhance the functionality of DeFi applications, such as decentralized exchanges or lending protocols.

2. **Supply Chain Management**: Real-time data from web2 logistics platforms can be utilized to track and verify supply chain transactions on the blockchain, improving transparency and accountability.

3. **Decentralized Identity**: Web2 identity verification systems can be integrated with blockchain-based identity solutions, enhancing user authentication and privacy mechanisms.

The integration of an Oracle System within the L1X network represents a significant stride towards closing the gap between traditional web platforms and blockchain technology. By facilitating seamless data transfer and interoperability, the X-Talk enriches the functionality, scalability, and versatility of decentralized applications. As the blockchain ecosystem continues to evolve, the X-Talk driven Oracle System stands as a testament to L1X's commitment to innovation, interoperability, and the broader adoption of decentralized technologies.

16. EVM Instance

EVM Instance deployment within the L1X blockchain is realized by leveraging Sputnik VM's compatibility with Ethereum's ecosystem. This integration enables developers to instantiate EVM Instances effortlessly, empowering them to deploy Ethereum-compatible smart contracts and dApps within the L1X network.

16.1 Sputnik VM- Empowering EVM Compatibility

Sputnik VM stands as a beacon of innovation in the realm of EVM implementations. Designed to deliver a faster, more secure experience for Ethereum network users, Sputnik VM is the cornerstone for deploying EVM Instances within the L1X blockchain.

16.2 Why Sputnik VM?

Below listed Sputnik VM features justify the rationale behind selection of Sputnik VM for EVM instance deployment.

1. **Feature-Rich and Secure:** Sputnik VM is a Rust implementation of the Ethereum Virtual Machine (EVM), boasting a comprehensive feature set aimed at enhancing efficiency and security. Developed by seasoned blockchain experts, Sputnik VM prioritizes performance and reliability, ensuring a robust execution environment for smart contracts within the L1X network.

2. **Stateless Design for Scalability:** A distinguishing feature of Sputnik VM is its stateless design, which streamlines execution by separating the computing engine from state storage. This design choice not only enhances scalability but also simplifies the development of lightweight and resource-efficient decentralized applications (dApps) on the L1X blockchain.

3. **High Performance and Efficiency**: Sputnik VM's focus on performance is evident in its architecture, optimized for high transaction throughput and smart contract execution speeds. By leveraging innovative techniques and optimizations, Sputnik VM offers a fast and efficient execution environment, ensuring optimal user experience and system responsiveness.

16.3 EVM Instance Implementation & Integration

The developers can instantiate EVM instances within the L1X network effortlessly. This capability enables the deployment and execution of Ethereum-compatible smart contracts, enabling developers to tap into the vast ecosystem of Ethereum tools, libraries, and frameworks.

The integration of EVM instances within the L1X network ensures seamless interoperability with Ethereum-based dApps and protocols. This integration empowers developers to leverage their existing Ethereum expertise and infrastructure, accelerating development cycles and fostering innovation within the L1X ecosystem.

16.4 Benefits

Deploying EVM instances within the L1X blockchain network offers a range of benefits tailored to support Ethereum-compatible applications and protocols.

1. **Seamless Compatibility**: EVM instance deployment ensures effortless integration of Ethereum-compatible smart contracts within the L1X network, facilitating smooth interoperability and compatibility with existing Ethereum-based solutions.

2. **Optimized Performance**: Efficient execution engines ensure rapid and reliable deployment of EVM instances, contributing to high transaction throughput and low latency within the L1X blockchain, thereby enhancing overall network performance.

3. **Enhanced Security**: Robust architectures and stringent testing protocols employed in EVM instance deployment bolster security measures, ensuring secure and reliable smart contract execution within the L1X network.

4. **Proven Reliability**: With a track record of supporting Ethereum-compatible applications, EVM instance deployment instills confidence in the reliability and stability of smart contract operations within the L1X blockchain, minimizing risks and uncertainties for developers and users alike.

16.5 Use Cases

Below listed are few use cases of EVM instance deployment in L1X blockchain network:

1. **Decentralized Finance (DeFi):** EVM Instances can facilitate the deployment of popular DeFi protocols, such as decentralized exchanges, lending platforms, and yield farming applications within the L1X network.

2. **Tokenization and NFTs**: EVM Instances support the creation and management of tokens and Non-Fungible Tokens (NFTs), enabling the development of EVM-compatible digital collectibles, asset management solutions, and tokenization platforms on the L1X blockchain.

3. **Blockchain Gaming**: With efficient and scalable architecture, EVM Instances can power blockchain-based gaming platforms, virtual realities, and augmented reality applications on the L1X network.

The integration of EVM instances within the L1X blockchain marks a significant milestone in enhancing interoperability, scalability, and developer experience. By adopting EVM standards, the L1X network seamlessly integrates with the broader Ethereum ecosystem, fostering innovation and growth. As EVM instances gain traction within the L1X network, they serve as a reliable and robust foundation, driving the adoption and success of Ethereum-compatible applications and protocols.

17. L1X Time Units

17.1 Time Units

L1X blockchain employs a structured approach to time measurement, with slots as the smallest unit. One block is created within each

Time Unit	Relative Units	Time	Blocks
1 Slot	Base Unit	0.5 secs	1 block
1 Chunk	40 slots	20 secs	40 blocks
1 Epoch	3 chunks	60 secs	120 blocks
1 Cube	3 epochs	180 secs	360 blocks

Table 5 L1X Time Units

17.2 Slots

Slots represent the smallest time unit. Each slot has a fixed duration of 0.5 seconds. Within each slot, a single block is created, encapsulating a batch of validated transactions. The utilization of slots ensures a regular and predictable block creation schedule, contributing to the network's stability and transaction throughput.

17.3 Chunks

To organize blocks within larger time intervals, the L1X blockchain introduces the concept of chunks. Each chunk consists of 40 slots, resulting in a 20-second time interval. By grouping blocks into chunks, the L1X blockchain enhances efficiency in block management and maintains a consistent rhythm for network operations.

17.4 Epochs

The L1X blockchain further aggregates chunks into epochs. An epoch comprises three consecutive chunks, resulting in a 60-second time period or 1 minute. This larger time unit allows for a broader evaluation of network performance, consensus, and synchronization. Epochs provide a useful reference point for analyzing blockchain metrics and facilitating various network operations.

17.5 Cubes

Cubes represent an extended time unit within the L1X blockchain. Each cube consists of three consecutive epochs, resulting in a time period of 3 minutes. The use of cubes enables long-term analysis, trend identification, and strategic planning within the blockchain network. Cubes serve as key milestones for tracking the progression of the L1X blockchain's performance and growth.

By implementing slots, chunks, epochs, and cubes, the L1X blockchain optimizes time management, ensuring precise block creation, consistency, and effective network operation. This systematic approach to time units enhances the blockchain's scalability, performance, and overall operational efficiency.

18. Full Validator Nodes (FVN)

In the L1X blockchain network, FVNs play a crucial role in maintaining the integrity and security of the network. These nodes become FVNs by staking a certain amount of L1X coins, which grants them the responsibility to validate transactions and contribute to the consensus process. Additionally, FVNs have the chance to be selected as a Block Proposer through a pseudo-random algorithm. This content elaborates on the functions and selection process of FVNs within the framework.

1. **Staking L1X Coins**: To become a FVN, participants are required to stake a specific amount of L1X coins. The staking process involves locking a predetermined number of coins in a designated wallet or smart contract. By staking L1X coins, nodes demonstrate their commitment to the network and gain the privilege to participate in the consensus mechanism.

2. **Transaction Validation**: FVNs have the responsibility to validate transactions within the L1X blockchain network. When a transaction is submitted, FVNs verify its correctness, ensuring that it adheres to the predefined protocol rules and meets the required criteria. This validation process involves confirming the transaction's digital signatures, checking for sufficient funds, validating the transaction's integrity, and ensuring that the block is proposed by the node selected as the Slot Head.

3. **Consensus Participation**: FVNs actively participate in the consensus process to agree on the order and validity of transactions. They contribute to reaching a consensus by proposing and validating blocks that contain verified transactions. FVNs collectively validate and agree upon the state of the L1X blockchain, ensuring consistency and preventing malicious activities.

4. **Block Proposer Selection**: The L1X blockchain network employs a pseudo-random algorithm to select a FVN as a block proposer. This algorithm takes into account various factors, such as the amount of L1X coins staked, etc. The selection process is designed to be fair and unbiased, ensuring that each FVN has a chance to propose a block.

5. **Block Proposal**: The selected FVN, acting as the block proposer, creates a new block by bundling verified transactions for the slots it is selected. The proposed block includes a cryptographic hash of the previous block, establishing the chain of blocks within the

blockchain. The block proposer broadcasts the proposed block to the nodes in the respective cluster in the L1X network for further validation and consensus.

6. **Consensus Confirmation**: FVNs collectively evaluate the proposed block to reach a consensus on its validity. They verify the included transactions, perform additional checks to prevent double-spending or other malicious activities and validate the block's integrity. If the majority of FVNs of a cluster agree on the proposed block's validity, the block is broadcasted to the other selected clusters in the Cube. The block becomes part of the L1X blockchain when consensus is achieved from 51% of clusters, advancing the network's state.

By staking L1X coins, validating transactions, and participating in the consensus process, FVNs contribute to the overall security, integrity, and decentralization of the blockchain network. The selection of a block proposer through a pseudo-random algorithm ensures a fair distribution of responsibilities, allowing every FVN to have an opportunity to contribute to block creation.

18.1 FVN LifeSPAN

In the L1X blockchain network, FVNs play a crucial role in maintaining the integrity and security of the system. The FVN lifecycle encompasses the various stages that FVNs go through, from their initial selection to their ongoing participation in block validation. This process ensures that only trustworthy and competent participants are chosen to validate transactions and create new blocks. The FVN lifecycle involves steps such as staking, selection, validation, and potential penalties or rewards. By understanding the intricacies of the FVN lifecycle, stakeholders can gain insight into the inner workings of a PoX blockchain network and appreciate the robustness of its consensus mechanism.

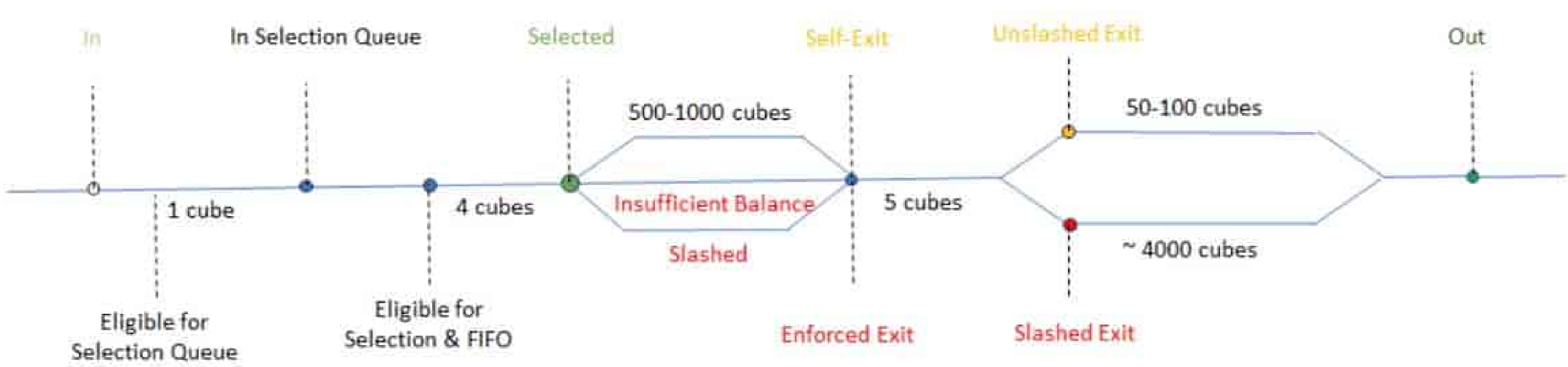


Figure 16 FVN Lifespan

18.1.1 Staking and Selection

During the "In" stage, a FVN stakes L1X coins to become an active participant in the network. After one cube, the FVN enters the selection queue, awaiting its turn to be eligible for selection as a FVN. The time required for selection depends on the network dynamics and the number of nodes ahead in the selection queue, with a minimum waiting period of four cubes.

18.1.2 Active FVN and Block Proposal

As an active FVN, the node can validate transactions, contributing to the consensus process and network security. Additionally, an FVN may have the opportunity to propose blocks if chosen through a pseudo-random algorithm, further participating in the block creation process. An FVN can be active for a maximum of 1000 cubes. At this stage, there are two more possibilities for a FVN. A FVN may have an insufficient balance or may get slashed due to malicious activity.

18.1.3 Exit

A self-exit occurs when a FVN voluntarily chooses to withdraw their staked amount prior to the completion of the active period i.e., 1000 cubes, signaling their decision to no longer participate in transaction validation. In contrast, an enforced exit can occur for these reasons: completion of the FVN's lifespan, insufficient balance, or being slashed for malicious activity. Moreover, to maintain network dynamics, a FVN may be enforced random exit anytime between 500-1000 cubes.

18.1.4 Exit Waiting Period

Upon initiating the exit process, a FVN must wait for five cubes before proceeding. For an unslashed exit, where a FVN exits without any penalties, the waiting period is typically between 50 and 100 cubes. However, in the case of a slashed exit, where the FVN is penalized for malicious behavior, the waiting period extends to 4000 cubes.

18.1.5 Withdrawal

The "Out" stage marks the final phase of a FVN's lifespan. At this stage, the FVN can withdraw their staked amount from the network, concluding their participation in the L1X blockchain as a FVN.

Through the above-discussed process, the lifespan of a FVN in the L1X blockchain network is defined, offering flexibility for voluntary exits and ensuring network integrity through enforced exits when necessary.

18.2 FVN Lifespan Algorithm

This algorithm provides a high-level overview of the FVN lifespan in the L1X blockchain network.

1. FVN enters the "In" stage by staking L1X coins.
2. After one cube, the FVN joins the selection queue after being eligible.
3. Wait for a minimum of 4 cubes based on network dynamics and nodes ahead in the selection queue.
4. FVN becomes eligible for selection as an active FVN.
5. If selected:
    a. FVN enters the "Active FVN and Block Proposal" stage.
    b. Validate transactions and contribute to consensus.
    c. Possibility to propose blocks through a pseudo-random algorithm.
    d. Repeat steps 5-9 until exit conditions are met.
6. For Self-Exit:
    a. Wait for 5 cubes.
    b. Initiate an unslashed exit.
7. For Enforced Exit:
    a. Identify the exit condition (lifespan completion, random exit, insufficient balance, or being slashed).
    b. Wait for 5 cubes.
    c. Initiate a slashed exit if penalized else initiate Unslashed Exit.
8. For Unslashed Exit:
    a. Wait for 50-100 cubes for withdrawal.
    b. Proceed to the "Out" stage for withdrawal.
9. For Slashed Exit:
    a. Wait for 4000 cubes for withdrawal.
    b. Proceed to the "Out" stage for withdrawal.
10. FVN completes the lifespan and proceeds to the "Out" stage.
11. Withdraw the staked amount from the network.

18.3 FVN SELECTION

At the beginning of each cube in the L1X blockchain network, the FVN selection process takes place to determine the FVNs in the selection queue for the next cube. The selection process ensures that the network has an adequate number of FVNs to maintain security, consensus, and transaction validation in adverse scenarios. The following steps outline the FVN selection process:

18.3.1 Minimum FVNs in a Cluster

Based on network dynamics, a threshold is set for the minimum FVNs required in the cluster. This value, known as the FVN threshold, is a proportion of total nodes. It ensures that there are a sufficient number of FVNs within a cluster to achieve consensus and prevent centralization.

18.3.2 FVNs Required

The next step involves calculating the total number of FVNs needed for the next cube based on active clusters and the FVN threshold.

$$\text{FVNs Required} = (1.5 * \text{Clusters}) * \text{FVNs Threshold}.$$

By considering 1.5 times the number of clusters, the L1X network ensures that there are enough FVNs to maintain a decentralized and secure network in adverse scenarios as well.

18.3.3 FVNs SELECTED

To fulfill the required number of FVNs, the difference between the FVNs Required and the existing Active FVNs is determined.

$$\text{FVNs Selected} = \text{Difference}(\text{FVNs Required, Active FVNs})$$

This value represents the additional FVNs that need to be added to the selection queue for the next cube.

The FVN selection process aims to strike a balance between network security, decentralization, and efficient consensus. By ensuring that each cluster has a minimum number of FVNs and calculating the required number of FVNs based on the number of clusters, the L1X blockchain network maintains a robust FVN ecosystem.

18.4 FVN Selection Algorithm

The algorithm provided is a simplified representation of the FVN selection process and can be adapted based on specific network requirements, selection criteria, and validation rules.

1. Set the FVN Threshold.
2. Calculate the total number of clusters in the network (*TotalClusters*).
3. Calculate the required number of FVNs for the next cube:
   *FVNsRequired = (1.5 * TotalClusters) * MinimumFVNs*
4. Get the count of active FVNs in the network (*ActiveFVNs*).
5. Calculate the number of FVNs needed to meet the requirement:

FVNsSelected = FVNsRequired - ActiveFVNs

6. Add *FVNsSelected* number of FVNs from the available nodes in the selection queue.
7. Repeat the process at the start of each new cube.

18.5 Vigilant Nodes

In a blockchain network, vigilant nodes play a crucial role in observing transactions, detecting any malicious activity performed by nodes, and promptly notifying the block proposer about such behavior. These nodes are incentivized with negligible rewards to discourage the promotion of malicious activities. This content provides an overview of the functions and purpose of vigilant nodes within the blockchain network.

1. **Monitoring Transactions**: Vigilant nodes actively monitor the transactions occurring within the blockchain network. They continuously analyze the transaction flow, observing the behavior and actions of participating nodes. Through their vigilant stance, these nodes strive to maintain the integrity, security, and fairness of the network.

2. **Detection of Malicious Activity**: With their keen observation, vigilant nodes possess the ability to identify and detect any instances of malicious activity perpetrated by nodes within the network. This includes actions such as double-spending, unauthorized modifications to the blockchain, or any other behavior that violates the established L1X protocol rules.

3. **Notification to Block Proposer**: Upon detecting malicious activity, vigilant nodes promptly notify the block proposer. By providing timely notifications, vigilant nodes ensure that the block proposer is made aware of the identified malicious behavior, enabling them to take appropriate actions.

4. **Negligible Rewards**: To discourage the promotion of malicious activities, vigilant nodes receive negligible rewards for their watchful role within the L1X network. The aim is to avoid creating incentives for nodes to engage in or tolerate malicious behavior. Instead, the focus is on maintaining a trustworthy and secure blockchain environment.

The presence of vigilant nodes reinforces the network's integrity and security by actively monitoring transactions, detecting malicious activity, and notifying the block proposer. By incentivizing vigilant behavior with negligible rewards, the L1X blockchain network strives to create a robust and transparent ecosystem, discouraging nodes from engaging in activities detrimental to the network's health and well-being.

18.6 Radix Nodes

The radix node is responsible for facilitating the entry of new nodes into the network. It goes beyond traditional boot nodes by incorporating additional functionalities, including cluster assignment, cluster maintenance, node shuffling, and threshold management. This comprehensive description outlines the significant aspects of the radix node's role within the L1X framework.

1. **Root Node Functionality:** The radix node serves as the initial point of contact for nodes seeking to join the L1X blockchain network. Its primary responsibilities include providing crucial information and assistance to new nodes and guiding them through the integration process. The radix node's functionalities encompass:
2. **Network State Information**: The radix node supplies new nodes with up-to-date details about the network's current state, including the latest blockchain state, network topology, and active nodes participating in the L2X blockchain.
3. **Connection Establishment**: The radix node plays a pivotal role in facilitating connections between new nodes and the existing L1X network. It offers a curated list

of active nodes and their IP addresses, allowing new nodes to establish direct connections and participate in the L1X blockchain network.

18.7 Cluster Assignment

In addition to the above functionality, the radix node assigns new nodes to specific clusters within the network. Clusters are logical subdivisions that group nodes based on shared characteristics or roles. The cluster assignment process involves the following steps:

1. **Network Topology Analysis**: Leveraging its understanding of the network's topology and node capabilities, the radix node evaluates new nodes to determine the most suitable cluster for each. This evaluation ensures optimal communication and collaboration among nodes with complementary features.

2. **Cluster Assignment**: The radix node assigns new nodes to appropriate clusters based on the evaluation results. This strategic allocation enhances the overall performance, scalability, and efficiency of the L1X blockchain network.

18.7.1 Cluster Maintenance and Node Shuffling

The radix node assumes the responsibility of maintaining clusters within the L1X network, ensuring their ongoing stability and effectiveness. This involves periodic node shuffling, wherein nodes are reorganized within clusters in regular intervals called Cubes. The cluster maintenance and node shuffling process encompass the following steps:

1. **Cluster Monitoring**: The radix node continuously monitors the clusters' composition and performance, keeping track of participating nodes and their respective roles within the cluster.

2. **Epoch Initialization**: At the beginning of each cube, the radix node initiates the node shuffling process by creating a randomized order of nodes within each cluster.

3. **Node Shuffling**: The radix node reassigns nodes within the cluster based on the randomized order, promoting fairness and preventing centralization. This shuffling enhances network resilience and discourages potential attacks.

4. **Cluster Update**: The radix node updates the cluster configurations and disseminates the revised information to all participating nodes, ensuring they are aware of the newly shuffled cluster arrangement.

18.7.2 Threshold Management

The threshold management aspect of the radix node evolves around maintaining the network's threshold by tracking node records and ensuring clusters have a minimum number of nodes to meet the threshold requirement. This process encompasses the following actions:

1. **Node Record Maintenance**: The radix node maintains records of nodes that have exited or been slashed from the network due to various reasons such as misconduct, insufficient resources, or voluntary withdrawal. These records help in tracking node history and determining their eligibility for future participation.

2. **Minimum Cluster Size Threshold**: The L1X framework defines a minimum number of nodes required to be present in each cluster to meet the threshold. The radix node ensures that each cluster maintains this minimum cluster size to preserve the L1X network integrity and security.

3. **Cluster Redistribution**: If a cluster falls below the minimum cluster size threshold, the radix node initiates the redistribution process. It identifies surplus nodes from other clusters or new nodes attempting to join the network and assigns them to the affected cluster, thereby maintaining the minimum number of nodes required.

4. **Node Eligibility Assessment**: When assigning nodes to clusters, the radix node evaluates the eligibility of new nodes based on the predefined threshold criteria. This assessment ensures that only qualified nodes, meeting the required computational power, reputation, stake, or other factors, are included in the cluster.

5. **Cluster Update and Communication**: Once the redistribution is completed, the radix node updates the cluster configurations and communicates the revised information to all nodes in the network. This enables nodes to stay informed about the updated cluster composition and maintain a synchronized view of the L1X network.

Thus, the radix node reinforces the network's stability, security, and adherence to predefined criteria. It guarantees that the L1X blockchain network operates with reliable and qualified nodes, enhancing the overall trustworthiness and effectiveness of the system.

18.8 Core Concepts

This section explaining core concepts in L1X, provides a foundational understanding of the fundamental principles and key elements that underpin the L1X blockchain ecosystem, enabling them to navigate and engage with the technology effectively.

18.8.1 Account

In the L1X blockchain ecosystem, each account maintains an associated state, where crucial metadata is stored. While this metadata is publicly readable, only the respective account owner can modify it. Accounts can possess multiple smart contracts, and the account's state is pivotal for tracking essential information, encompassing:

- **Balance**: This field records the account's balance in L1X tokens, indicating the financial resources available to the account.
- **Nonce**: The nonce is a unique number associated with each transaction from the account. It aids in preventing replay attacks and ensuring transaction order.
- **Account Type**: Accounts in L1X are categorized into two types:
    1. **User Account**: Representing a user's personal account, used for various transactions and interactions.
    2. **System Account**: Functioning as service accounts that serve specific purposes within the blockchain network.

Account Creation in L1X is contingent on the account type:

- **User Account Creation**: User accounts are created by transferring L1X tokens to a newly generated account address. These accounts cater to individual users' needs within the network.
- **System Account Creation**: System accounts come into existence through specific actions:

1. **Staking L1X Tokens**: When a user stakes L1X tokens, a new system account is automatically created, serving as an integral component of the staking mechanism.

2. **Deploying Smart Contracts**: Contract deployment leads to the creation of a new system account. This account initiates with a zero balance, and any tokens transferred to it are irretrievably lost. It primarily acts as a repository for contract-related transactions.

3. **Initializing Smart Contracts**: Smart contract initialization results in the generation of a new system account. Initially, this account starts with a zero balance but can receive deposits. It retains the state of the associated smart contract. Multiple initializations of the same smart contract create distinct system accounts, each dedicated to a specific instance of the contract.

In essence, L1X account states play a pivotal role in tracking metadata, balances, and transaction history, while the differentiation between user and system accounts provides versatility and functionality within the blockchain ecosystem. Account creation methods depend on the desired type, ensuring that user and system accounts fulfil their respective roles efficiently.

18.8.2 Address

The concept of addresses in the L1X blockchain is a crucial aspect of the system's architecture. These addresses serve as unique identifiers for various accounts and entities within the blockchain network. Below, we'll delve into the details of how addresses are derived based on the account type and the associated payload:

Address Derivation in L1X: Address generation in L1X is determined by applying the **keccak256::hash** function to a specific payload, which varies depending on the type of account or entity. The resulting hash is then truncated to the last 20 bytes to create the final address. Let's explore the address derivation process for different types of accounts:

1. **User Account Address:**

- **Payload**: For a user account, the payload used in address generation is the user's public key.
- **Address Derivation**: The keccak256 hash of the public key is calculated, and the last 20 bytes of this hash represent the unique address of the user account.

2. **Contract Code Address:**
   - **Payload**: The payload for generating a contract code address consists of three components: Account Address, Cluster Address, and Nonce.
   - **Address Derivation**: These components are concatenated to form the payload. The keccak256 hash of this payload is calculated, and the last 20 bytes of the hash become the contract code address. This address is generated when a new contract is deployed on the L1X blockchain.

3. **Contract Instance Address:**
   - **Payload**: Similar to the contract code address, the payload for a contract instance address includes Account Address, Contract Code Address, Cluster Address, and Nonce.
   - **Address Derivation**: These components are combined into a single payload, and the keccak256 hash of this payload is computed. The last 20 bytes of this hash constitute the unique address of the contract instance. This address is generated when a contract is initialized, and it is distinct for each initialization.

4. **Pool Address:**
   - **Payload**: The payload for generating a pool address comprises Account Address, Cluster Address, and Nonce.
   - **Address Derivation**: These components are concatenated, and the keccak256 hash of the resulting payload is calculated. The last 20 bytes of this hash serve as the pool address. This address is generated when a user stakes L1X tokens in the network's staking mechanism.

In summary, addresses in the L1X blockchain are generated based on a specific payload that varies according to the account type or entity. The keccak256 hash of the payload is used to derive the address, with the last 20 bytes of the hash serving as a unique identifier. This address derivation process ensures that different types of accounts and entities within the blockchain ecosystem possess distinct and recognizable addresses.

18.8.3 Smart Contract

L1X smart contracts are integral to the blockchain's operation as they execute the instructions embedded within each transaction on the network. These contracts are implemented using L1X eBPF (Extended Berkeley Packet Filter) machine bytecode, making them capable of autonomously processing and responding to transactional commands.

- **Ownership of Smart Contracts**: Ownership plays a vital role in the L1X smart contract ecosystem, distinguishing between the owner of the contract code and the owner of the contract instance:

    1. **Owner of Contract Code**: The owner of the smart contract code is the individual or entity responsible for deploying the smart contract onto the L1X blockchain. This owner has the authority to initialize and manage the contract but does not necessarily own the individual instances of the contract.

    2. **Owner of Contract Instance**: The owner of a smart contract instance is the individual or entity that initializes a specific instance of the smart contract. This owner has control over the contract's persistent storage and any state changes within that instance. It's important to note that multiple instances of the same contract can exist, each with its own owner.

- **Smart Contract Accounts:** Smart-contract accounts are a fundamental concept in blockchain technology, serving as the building blocks of decentralized applications (DApps). In this context, we'll explore the distinct roles and functionalities of two essential types of smart-contract accounts: the contract code account and the contract instance account.

1. **Contract Code Account**: When a smart contract is deployed on the L1X blockchain, it results in the creation of a new system account known as the "contract code account." This account is dedicated to housing the bytecode and logic of the smart contract. It acts as a repository for the contract's executable code, allowing it to be executed and accessed by the network.

2. **Contract Instance Account**: To fully utilize a smart contract, it must first be initialized. When a contract is initialized, a new system account, referred to as the "contract instance account," is generated. This account serves a specific instance of the contract and is responsible for managing the contract's persistent storage and state changes. Notably, a single smart contract can have multiple instances, each with its own instance account, allowing for parallel and distinct use cases of the same contract.

In summary, L1X smart contracts are essential components of the blockchain, executing eBPF bytecode to perform various tasks. Ownership is divided into the owner of the contract code (the deployer) and the owner of individual contract instances (the initializers). Smart contract accounts include the contract code account, responsible for storing the contract's code, and the contract instance accounts, created upon initialization to manage specific instances of the contract, with the possibility of multiple instances for a single contract.

18.9 Lexicon

- Slot: The smallest time unit in the L1X blockchain, where one block is created every 0.5 seconds.
- Chunk: A 20-second time interval consisting of 40 slots in the L1X blockchain, organizing blocks for efficient management and synchronization.
- Epoch: A period in the L1X blockchain comprising consecutive 3 chunks, typically lasting for one minute, during which clusters propose blocks and facilitate network consensus.
- Cube: A unit of time in the L1X blockchain that encompasses multiple epochs, typically lasting for three minutes, incorporating cluster shuffling and providing milestones for network analysis and planning.

- Radix: a node that serves as an entry point for new nodes, providing network information, cluster assignment, and maintaining thresholds for participation.
- FVN: a node that validates transactions, secures the network, and can be selected as block proposers through a pseudo-random algorithm based on the stake and other factors.
- Block Proposer: a node responsible for creating new blocks and proposing them to the network for validation.
- Vigilant Node: a node in a blockchain network that monitors transactions, detects and reports malicious activity to the block proposer.

19. L1X Reward Mechanism

Blockchain-based token economy design requires a rigorous engineering approach to be able to sustain and lead to a virtuous cycle of growth driven by network effects. Token engineering can be defined as the theory, practice, and tools to analyze, design, and verify tokenized ecosystems. Token engineering encompasses theoretical concepts being applied to create and deploy a token economy. Defining the main goals of the token system is a critical step in token engineering leading to rigorous analysis, design, and verification of ecosystems and desired behavior.

Layer-One.Coin (Layer-One.C) measures its stability and facilitation of transactions by engaging its resource accessibility model to tend toward the right balance between the supply and demand intersection to be determined by time t and resources r on the network.

The goal to reduce the friction to an optimal level based on the correlation takes the following sequence:

- Decrease in friction/transaction cost leads to,
- Increase in token transaction leads to,
- Increase in token velocity leads to,
- Increase in Supply of tokens then leads to the,
- Decrease in the Dollar Value of the token.

The factor of t and r on the network are directly correlated to Demand D and Supply S on the network. Here, the friction of the token with the number of Groups and the Consensus Mechanism of relatable transactions are correlated.

19.1 Friction on the Network

Friction F on the network determines the excess the transaction initiator is willing to bear to conduct a transaction on the network. The goal of the F is to be at its optimal. Three situations emerge out of Friction as follows.

19.1.1 When $F < 0$ (not ideally possible)

Where the number of transactions must be recorded on the network in a time interval, the number of transactions should not have latency time and cost associated. Where F

< 0, validation compensation will be less than zero which is not economically feasible as there is a positive correlation in between validation compensation and the time required to record a transaction.

If the network uses its resources to conduct transactions, the market cap of the total volume will go down, which will drain resources on the network. This will not provide a scalable mechanism of conducting transactions.

19.1.2 When $F = 0$ (no growth in the token)

This will lead to no growth in the value of the tokens as the friction cost will increase on a constant basis if the supply and demand increase on a constant basis. If there is an imbalance between the supply shift, the impact will be on the friction and, hence, this model is not suitable for an imperfect marketplace for transactions.

19.1.3 When $F > 0$ (Inflation, Mint and Burn)

Where the friction for $F > 0$, PoS and PoW based consensus mechanism will create ways for an accumulation problem. Velocity will be lower. This has a direct correlation with the transaction cost increasing.
When the friction is set by the network based on the Proof of Participation, it will adjust the friction curve based on the supply and demand of the transactions on the network. The goal is to keep the velocity of the token at an optimal network acceptance level.

Where F: Friction S: Supply D: Demand n : Number of nodes G : Groups of (i) smart contract accounts, (ii) transactions, (iii) receipts, and (iv) others Tr : Transactions t: Time r: Resources
v : Velocity

The friction is paid by the initiator and the allocation of the friction is put back into the pool, which is distributed to adjust the supply curve into the system. The more transactions that are processed on the network by an initiator, the more resources the initiator will pass on to the network to maintain the pool.

19.2 Consensus Mechanism and Incentive Scheme

The Proof of Participation consensus mechanism proposed is based on a resource allocation module where the participants in transaction or value transfer constitute the nodes and the network. The user u is the initiator and the nodes n will be used as the verifier on the network. Hence,

$$F=0 \rightarrow V=\{G1,G2,G3,G4\} \quad (29)$$

Where F: Friction V : Validator G : Groups of (i) smart contract accounts, (ii) transactions, (iii) receipts, and (iv) others

19.3 Resource Costing Model

The cost of the resource allocated must be based on the equilibrium of initiation and validation. For the total number of transactions, there is an equal amount of demand based on the network. Hence, Supply and Demand will be 1:1 with an error margin of 1(10%):1(10%) which will be corrected by the network.

19.4 Token Distribution and Pricing Strategy

The distribution of the Layer-One tokens is based on building network effects and bootstrapping the network from the initiation phase, including participants such as layer two application developers, network growth facilitators, and users. A pricing strategy that is optimized for growth will be adopted. There are various correlations with the factors that comprehend the token distribution model.

19.5 Consensus Mechanism and Token Pricing with Respect to Token Inflation

Layer One X and its innovative Proof of Participation based consensus mechanism provides an acceptance state of validation. When the participants are validators, they are a part of the same pool of the network. The resource utilized to achieve a consensus state can be directed towards an optimal resource utilized and incentive mechanism. When the incentive mechanism is optimal, the velocity of the token for rewards is optimal.

If there is an imbalance between the incentive and token velocity, the price of the token will incur more inflation. L1X has inflationary measures taken into account.

For example, FLOW tokens (the native currency of the Flow network) distribute the inflation to stakeholders and this results in a decrease in the value of the token. Evidently, the control of inflation must be done at the consensus mechanism stage.

19.6 Token Participant Distribution and Pricing

The participant token distribution strategy incorporated at L1X provides for growth in the network and incentivizes participants to get on board with the mission.

At L1X, the resource model is based on the usage of the computing power that an account requests and provides on the network where the participants and the validators are mutually exclusive but form a part of the same pool. In such instances, the participants that provide resource utilization and availability are the developers and the users on the network.

19.7 Ongoing Distribution Rules, Governance, and Token Pricing

The ongoing distribution falls under two major components and its relation to token pricing is also considered.

Firstly, the technical module considers the resources available in the network and the cost to facilitate state changes. The value of the resource is based on optimally setting the value of the token to avoid considerable friction that will bottleneck the network.

Secondly, network and service level issuance consider different components such as ongoing mainly ancillary allocation of resources such as threading, spills, bundles, and others.

19.8 Layer Two Tokenomics Vs Layer One Tokenomics

The distribution of tokens to the developers enables the distribution and utilization of the resources on the network, which enables ongoing coin distribution to increase coin decentralization as the users acting as the validators get rewarded.

To overcome the problem of increasing the cost of resources on the network, resulting in coin inflation, the ongoing coin distribution model needs to correlate with the growth of the network.

19.9 L1X Coins

Layer-One coins (L1X), are required in the following cases:

- **Resources:** To conduct a transaction on the network or to be able to execute scripts on the network. Resources on the network are the primary driver and L1X coins are required.

- **Exchange:** L1X coins provide a medium of exchange amongst its participants on the network, which can be used to transact on the network.

- **Liquidity for secondary coins:** To gain liquidity on the platform, L1X coins will provide access to decentralized pools that will allow for coin exchange. They will also be able to act as collateral for secondary tokens.

- **Rules and governance participation:** For the ongoing rules and governance changes, it is important for the network to be able to sustain changes and updates. To validate updates, L1X coins will allow users to participate in governance and rule updates.

20. Tokenomics

Layer One X (L1X) employs a meticulously crafted combination of mathematical modelling, economic theory, and advanced data analysis to design a robust and trust less platform. Strategic Partnership with Prysm Group: Building a resilient Tokenomics Framework for L1X

Layer One X (L1X) had partnered with Prysm Group, a globally renowned consulting firm specializing in the economics and governance of blockchain-based ecosystems. Known for their leadership in economic modeling and innovative tokenomics, Prysm Group has a stellar reputation for creating sustainable and impactful economic frameworks for decentralized platforms. Prysm Group collaborates with academic and industry experts from leading institutions such as MIT, Stanford, and the University of Chicago. These advisors provide cutting-edge insights into crypto economics and decentralized governance.

L1X Tokenomics: A Game Theory-Inspired Design

L1X's tokenomics framework is deeply rooted in foundational principles of game theory, market design, contract theory, monetary economics, and social choice theory. These disciplines guide the architecture of governance mechanisms, enabling stakeholders to actively participate in platform updates, resource allocation, and decision-making processes. By embedding these elements into the protocol, L1X ensures transparency, fairness, and economic sustainability.

Governance and Economic Mechanisms

The governance framework integrates a systematic approach to proposals and decisions, authorizing stakeholders to contribute to the platform's evolution. The mechanisms include:

• Monetary Policies: Establishing the coin supply dynamics to manage inflation, deflation, and market stability.

• Platform Subsidies and Rewards: Incentivizing participants to enhance liquidity, security, and ecosystem growth.

• Rights and Obligations: Defining contractual terms, including escrow services and a dispute resolution process, to enforce trust and compliance.

Core Economic Parameters

L1X models its ecosystem by analyzing and optimizing critical economic parameters:

1. Economic Activity: Tracking and promoting healthy transaction volumes and ecosystem engagement.

2. Price Stability: Mitigating price shocks and ensuring consistent coin valuation.

3. Coin Release Schedule: Implementing a controlled emission rate for long-term sustainability.

4. Reward Distribution: Structuring incentives for validators, developers, and community members to drive participation.

5. Lock-Up and Vesting Periods: Introducing mechanisms to reduce coin velocity and maintain market stability.

6. Governance Value: Quantifying and enhancing the value of governance rights for token holders.

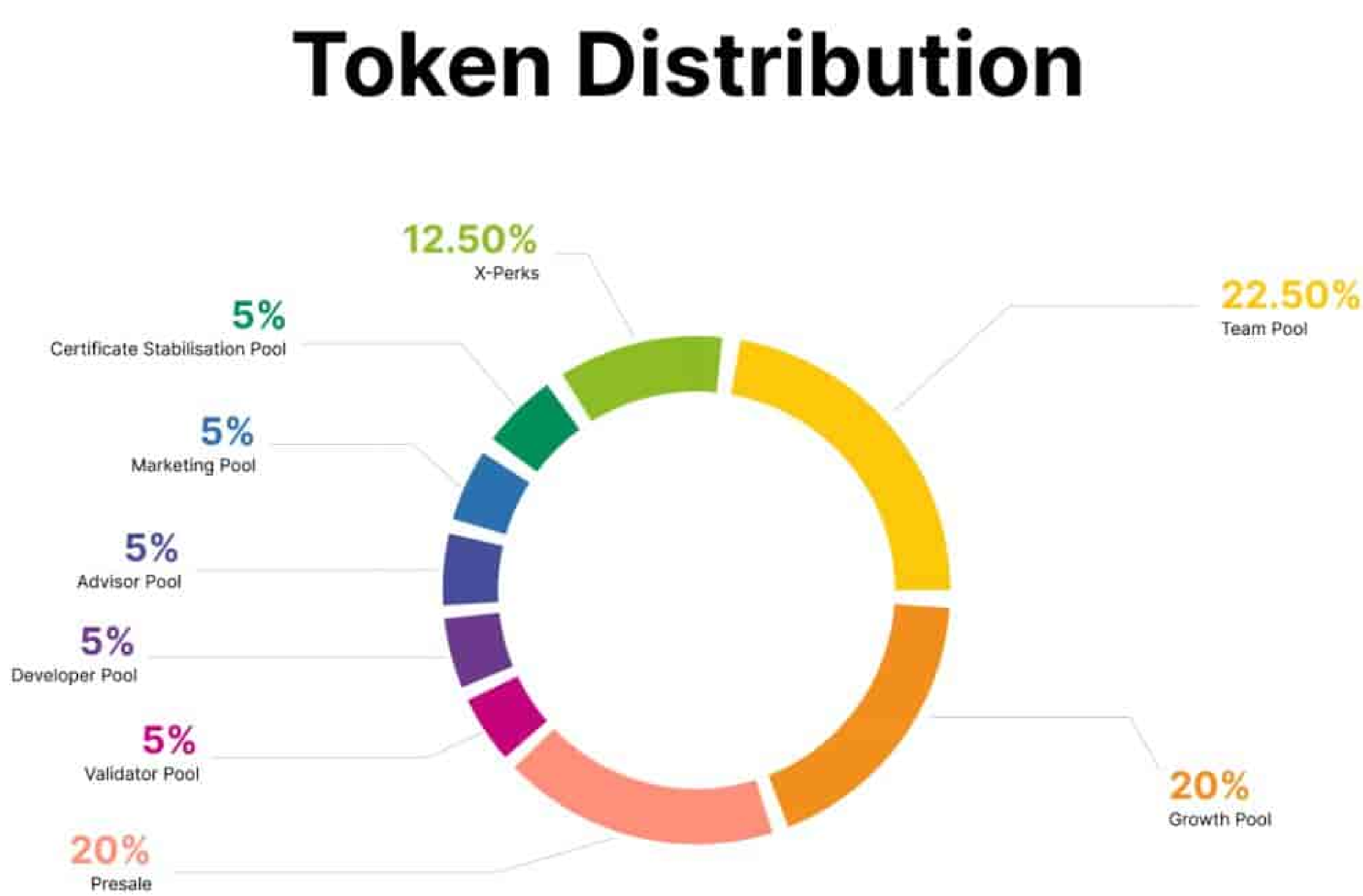
Hard Cap and Supply Dynamics

To maintain economic integrity and prevent inflationary risks, the L1X token supply is capped at 1 billion coins. This hard cap aligns with the platform's overarching goals of balancing supply with demand, incentivizing participation, and ensuring long-term ecosystem health.

By integrating these comprehensive economic and governance models, L1X achieves a tokenomics framework that is not only efficient and scalable but also resilient to market fluctuations, fostering a thriving ecosystem for cross-chain interoperability.

Token Distribution

Layer One X (L1X) has strategically designed its token distribution model to ensure a balanced allocation across all critical components of its ecosystem. The distribution strategy reflects L1X's commitment to fostering sustainable growth, incentivizing key stakeholders, and maintaining long-term stability within the network. Below is a detailed breakdown of the L1X token distribution.



Team Pool (22.50%)

The Team Pool allocation underscores L1X's dedication to rewarding the contributions of its core team members. This portion ensures that the team remains incentivized to drive innovation and achieve long-term goals. The vesting schedule for this allocation reflects a commitment to aligning the team's interests with the platform's growth. The Team Pool has the longest vesting schedule among all stakeholders on the platform, featuring a 12-month cliff followed by a 36-month vesting period.

Growth Pool (20%)

The Growth Pool is designated to fuel ecosystem expansion, partnerships, and technological advancements. This allocation is key to maintaining a competitive edge in the evolving blockchain landscape by enabling strategic investments and ecosystem development.

Presale to DeX (20%)

Tokens allocated for the Presale are intended to onboard supporters and strategic investors who believe in L1X's vision. This distribution ensures that the platform has the necessary funding to scale operations and enhance its infrastructure.

X-Perks (12.50%)

The X-Perks allocation is specifically reserved for a group of early investors who have demonstrated their confidence in the L1X vision and its long-term potential. This pool holds coins for their early investments and is designed to incentivize their continued engagement with the platform. Additionally, X-Perks helps foster user participation in governance, liquidity provision, and other essential network activities, ensuring alignment with the broader goals of the ecosystem.

Certificate Stabilization Pool (5%)

This pool supports Certificate Stabilization which is in addition to rewarding nodes including L1X Full Validator Nodes and X-Talk Nodes, ensuring the platform maintains high standards of reliability, trust, and accuracy across its decentralized services. This allocation strengthens the security and trustworthiness of the network.

Marketing Pool (5%)

The Marketing Pool is dedicated to driving awareness and adoption of L1X. These tokens fund promotional activities, campaigns, and educational content to expand the platform's reach and grow its user base.

Advisor Pool (5%)

The Advisor Pool compensates industry experts and strategic advisors who contribute their expertise to the platform's development. This allocation ensures that L1X benefits from cutting-edge guidance in blockchain and economics.

Developer & Bug Bounty Pool (5%)

The Developer & Bug Bounty Pool is dedicated to incentivizing developers to contribute to the growth and innovation of the L1X ecosystem. This allocation supports the development of tools, dApps, and technological advancements while reinforcing L1X's commitment to fostering an open-source developer community. Additionally, it funds bug bounty programs to ensure the platform's security and reliability, encouraging collaborative contributions and continuous improvement.

Validator Pool (5%)

The Validator Pool rewards participants who secure the network by validating transactions and maintaining its integrity. This allocation ensures decentralized governance and platform security.

Vesting Schedules & an Innovative Anti-Dump Release Pool

Vesting of the L1X Coins is used to maintain the project's overall integrity while managing the technology's adoption phase. The vesting schedule is designed to ensure a balanced distribution of tokens, aligning the interests of various stakeholders and securing long-term project sustainability. Below is a breakdown of each pool, its allocation, cliff period, and vesting period.

L1X Vesting Schedule

Pools	Allocation %	Total Allocation	Cliff from TGE	Linear Vesting Period
Team Pool	22.5%	225,000,000	12 months	36 months
Growth Pool	20%	200,000,000	No Cliff	120 months
Presale (DeX)	20%	200,000,000	No Cliff	-
Validator Pool	5%	50,000,000	No Cliff	60 months
Developer Pool	5%	50,000,000	No Cliff	60 months
Advisor Pool	5%	50,000,000	12 months	36 months
Marketing Pool	5%	50,000,000	No Cliff	120 months
Certificate Stabilisation Pool	5%	50,000,000	24 months	60 months
Xperks	12.5%	125,000,000	As per X Perk Tiers	

20.1.1 Release Pool and Anti-Dump Mechanism

The Release Pool plays a critical role in Layer One X's (L1X) tokenomics, functioning as a key mechanism to ensure price stability and mitigate the risk of sudden market dumps. This strategic feature is designed to regulate the automated sale of coins, aligning with L1X's broader commitment to fostering a sustainable and healthy token ecosystem.

20.1.2 How the Release Pool Works (More Details)

L1X coin holders have the choice to stake their coins in the Release Staking Pool (Decentralised Contract), which operates in tandem with a sophisticated performance algorithm. This algorithm interacts seamlessly with the trading pools where L1X is listed, ensuring controlled and strategic token sales. The system is designed to prevent large-scale sell-offs from any token distribution pool, thereby maintaining market stability and preventing price volatility. For 2025, this strategy is aimed at preserving a healthy and sustainable L1X coin price, while simultaneously providing coin holders with the flexibility to sell their tokens in a steady and controlled manner, achieving optimal market conditions for all stakeholders.

20.1.3 Key highlights of the Release Pool mechanism

• Automated Sales: Tokens are released systematically to maintain liquidity without overwhelming the market.

• Controlled Supply: The gradual release prevents oversupply, ensuring a healthy balance between supply and demand.

• Dynamic Adjustments: The system can adapt to market conditions, further reducing the likelihood of price volatility.

20.1.4 Anti-Dump Strategy

The Release Pool is central to L1X's anti-dump strategy, specifically designed to protect the token's value and promote long-term growth. By automating and controlling token sales, the pool minimizes the potential for large-scale sell-offs by stakeholders or malicious actors, which could destabilize the market.

20.1.5 Benefits of the Release Pool

1. Market Stability: Protects the token price from drastic fluctuations caused by high-volume token releases.

2. Investor Confidence: Demonstrates a commitment to sustainable market practices, building trust among investors and stakeholders.

3. Strategic Liquidity Management: Ensures sufficient liquidity in decentralized exchanges (DEXs) while avoiding excess supply.

By incorporating the Release Pool into its tokenomics model, L1X takes a proactive approach to address the challenges of token price stability. This mechanism not only supports the platform's long-term vision but also enhances its appeal to both investors and users by fostering a fair and balanced ecosystem.

Open-Source and Transparent Release Pool Algorithm

To promote transparency and build trust within the community, the Release Pool Algorithm has been designed as an open-source solution, allowing stakeholders to audit and understand its functionality. This sophisticated algorithm governs the controlled release and sale of L1X coins, ensuring market stability and mitigating the risk of price volatility.

Key Features of the Algorithm

1. Performance-Based Optimization: The algorithm dynamically adjusts token release rates based on market conditions, such as trading volume, liquidity, and price trends, to ensure steady sales without overwhelming the market.

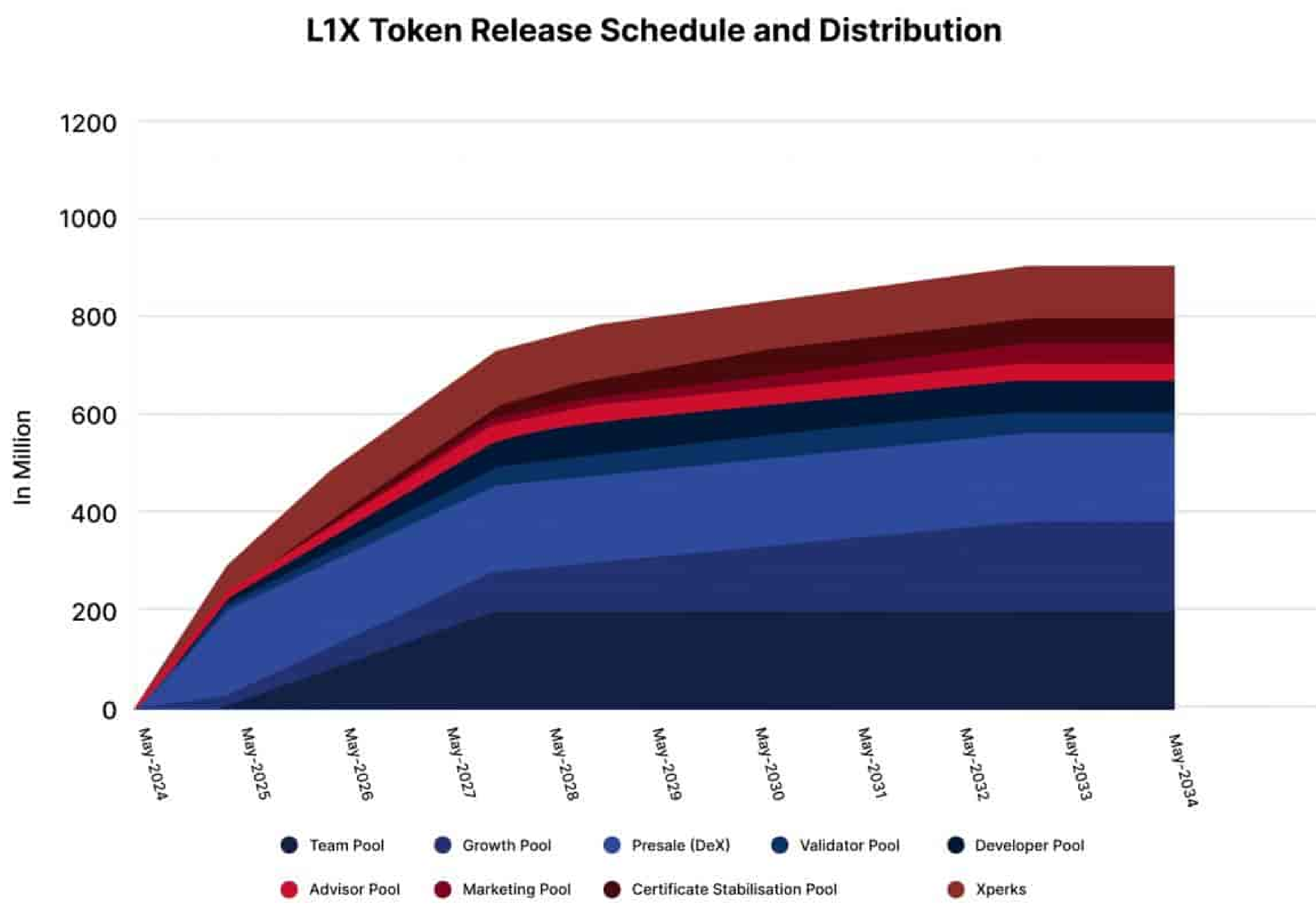
2. Anti-Dump Safeguards: Built-in mechanisms prevent large-scale sell-offs, ensuring that tokens from distribution pools enter the market in a controlled and predictable manner.

3. Market Synchronization: The algorithm interacts directly with trading pools, analyzing real-time data to determine the best conditions for releasing tokens, thereby maintaining healthy market activity.

4. Stakeholder Flexibility: Coin holders staking in the Release Staking Pool benefit from a system that prioritizes steady sales, allowing them to sell their coins at optimal prices while contributing to the overall stability of the ecosystem.

Token Release of L1X Coin Supply

The L1X coin release schedule is strategically crafted to manage token supply over the long term, promoting sustainable growth, market stability, and investor trust as we progress toward the hard cap of 1 billion tokens. This approach ensures that all categories can stake their coins in the release pool while encouraging new buyers on exchanges to invest in L1X, supported by robust anti-dump measures.



21. L1X Applications

The robust and efficient L1X network design facilitates multiple applications across various industries including the financial market, logistics, supply chain management, and healthcare, to list just a few. Following are some scalable and interoperable use cases that can be built on top of the L1X network.

21.1 Staking

The Layer One X solution will permit developers to use the abstraction layer that will enable the crypto holders such as Ethereum to make use of decentralized pools. It will allow for cross- cryptocurrency staking, reducing the staking cost through centralized exchanges that are currently charging between 10-30%.

21.2 Exchange

The exchange of tokenized values will permit the industrial use case to transfer the tokenized assets cross-blockchain which will permit value, data, and transactional exchange. Whilst there are a few solutions that allow for heterogeneous blockchain transactions (e.g., Wanchain), it converts the native cryptocurrency into a subset token called WanX (for example WanETH), which increases the transaction cost and time to exchange values. The Layer-One.I will focus on decentralized pool exchange using the wallet integrated mechanism that will allow users to authenticate transactions without transferring and changing the native global state.

21.3 Ledger Tokenization

With the advocation of Decentralized Ledger systems and its access to be able to securely store confidential information and share it amongst third parties with a commercial standpoint, it is imperative to establish a standardized ledger tokenization system. L1X provides the same use case in Intellectual Property (IP) and its functionality such as: (i) tokenizing information and agreements through smart contracts, (ii) proof of ownership and version control, and (iii) unifying the IP system.

Regarding IP ownership rights and sharing the rights with other parties, the modules or the rights to use certain parts of it can be tokenized using L1X smart contracts. This will allow for the information to be shared securely with resource and time factors attached to it. The tokenization process can be applied to the complete spectrum of parties such as suppliers, distributors and manufacturers that will permit data tokenization and commercialization.

IP ownership can be securely communicated into a hash table with its version control. The different versions can be linked allowing for the data to be sharded and stored securely on the nodes on the blockchain. L1X will provide a use case for IP tokenization with a different version control system that will allow the data to be hashed and shared with security measures. A confidential ledger is maintained and shared with parties on custom terms and conditions allowing for trading on the marketplace.

With the interoperable nature of the LayerOne.I technology, it is possible to tokenize the data and store it in a single ledger to unify the system of sharing and accessing IP systems.

21.4 Micropayment Transactions

As the consensus mechanism does not rely heavily on rewarding the miners or staked pool holders, it will be possible to accommodate micropayment transactions cross-chain and cross- blockchain, which is not possible today, even after ETH 2.0 activates. The staked pool will have to be tipped, which will increase the cost of the transactions. On 1 March 2024, the average gas fee for a swap transaction on Ethereum was around $79, and some instances saw fees climbing as high as $400 in late February. This is not suitable for micro-transactions.

21.5 Micro Trading

L1X solution could decentralize a trustless P2P network to connect buyers and sellers without a middleman, unnecessary fees, or restrictions. Given the micropayment features enabled in the L1X solution, opportunities and efficiencies to the retail industry can be opened up. For example, a renewable energy marketplace could utilize micro trading on a retail L1X solution to transact rooftop solar energy trading, down to 5-minute intervals, instead of 30-minute intervals through Power Ledger's blockchain platform or 30/60-day billing cycle through Synergy (Western Australia's government electricity supplier).

21.6 Gaming

L1X blockchain revolutionizes Web3 gaming with its native X-Talk cross-chain interoperability and cutting-edge features. Game developers can leverage L1X to create secure and transparent games using smart contracts in WebAssembly (WASM) format. The powerful L1X Compiler optimizes graphics and elevates gameplay performance, immersing players in a seamless and visually stunning gaming experience. The cross-chain interoperability offered by L1X enables collaboration and shared experiences among players across different platforms, breaking down the barriers that currently confine Web3 game developers and players to separate blockchains. By connecting these siloed blockchains, L1X unlocks the true potential for network effects within the Web3 gaming ecosystem, fostering growth, innovation, and sustainability.

With L1X, epic gaming competitions become a reality as players can directly influence the game's Blockchain state through smart contracts. This empowers gamers to actively participate and shape the outcome of their gaming experiences, while ensuring scalability and transparency. L1X's robust infrastructure boasts an impressive capacity of handling up to 100,000 transactions per second (TPS) at a minimal cost of only $0.01 per transaction, providing a seamless and cost-effective gaming environment. By leveraging the power of L1X blockchain, Web3 gaming enters a new era of limitless possibilities, where developers and players can thrive and create immersive gaming worlds. The innovative features and scalability of L1X elevate the gaming experience, making it highly engaging, secure, and transparent for all participants involved.

21.7 Metaverse

L1X allows for seamless collaboration and interaction within the Metaverse through its X-Talk cross-chain interoperability. Users can effortlessly navigate between different virtual worlds, fostering a vibrant and interconnected ecosystem where experiences can be shared and a diverse community can thrive. Furthermore, L1X's smart contract capabilities, coupled with the WebAssembly (WASM) format, empower Metaverse developers to create decentralized environments. These smart contracts ensure transparency and security within virtual transactions and interactions, enabling users to engage with virtual assets, participate in virtual economies, and even influence the state of the Metaverse itself. The addition of the L1X Compiler further enhances the Metaverse experience by providing real-time graphics and performance optimizations, delivering visually stunning and immersive virtual environments. With L1X's scalability and high transaction speeds, the Metaverse can accommodate a large user base and guarantee a seamless and enjoyable experience for all participants.

22. L1X Components used in example Use-Cases

22.1 L1X Fungible Token Smart Contract

Example-based Description:

L1X Fungible Token smart contracts empower businesses to reimagine loyalty programs, creating a win-win scenario for both customers and the organization. The implementation of these contracts not only enhances customer loyalty but also offers numerous benefits such as cost efficiency, interoperability, transparency, and data-driven insights. Recognizing the potential of L1X Fungible Token smart contracts in transforming traditional business models can lead to substantial long-term returns.

Example:

Let's consider the scenario of a loyal WYD Mart customer who frequently shops at their stores. With the implementation of L1X Fungible Token Smart Contracts, the customer now receives loyalty tokens every time he makes a purchase. These tokens are stored securely on the L1X blockchain, providing transparency and trust to all stakeholders involved.

Technical Workflow

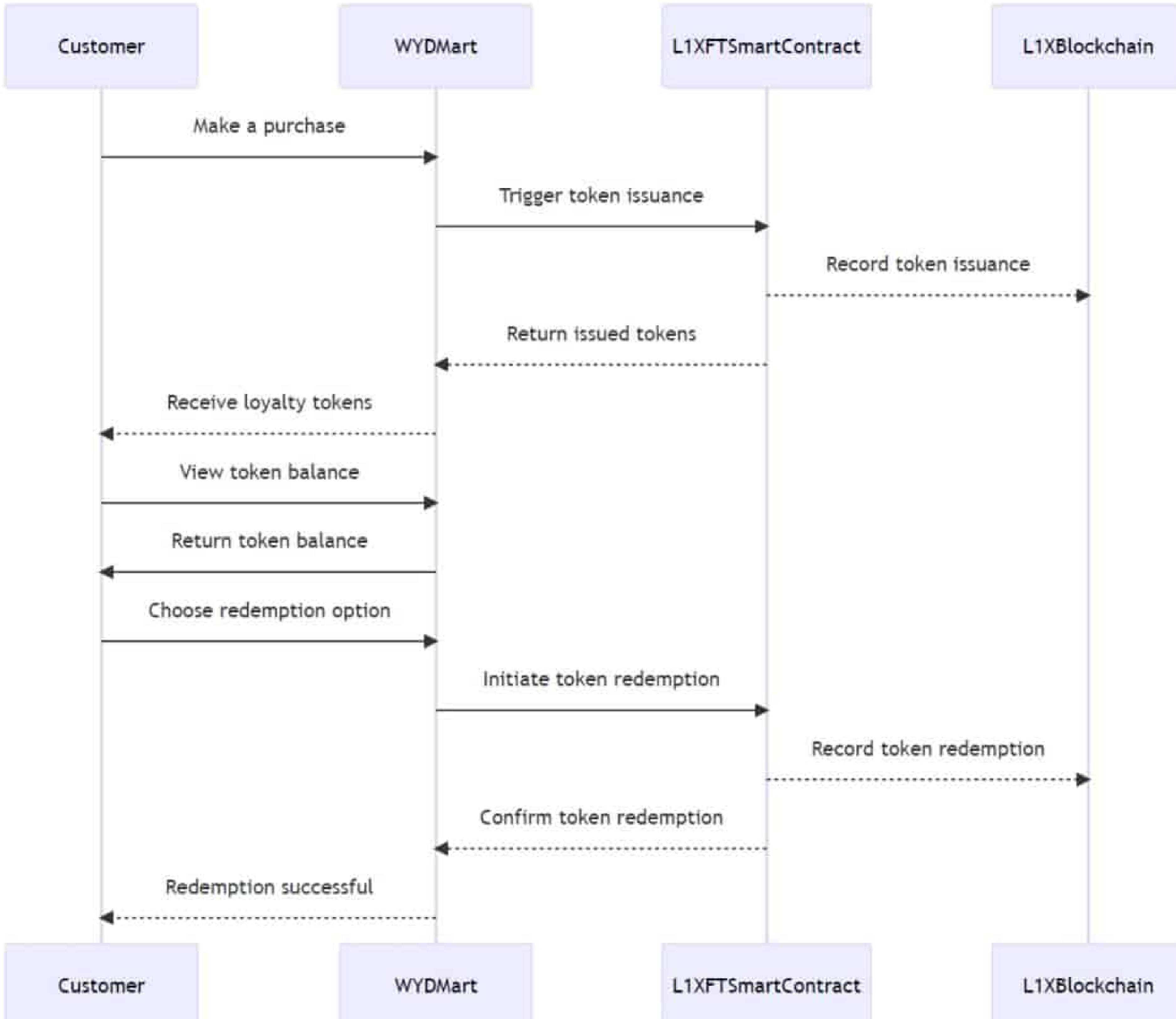


Figure 20 L1X Fungible Token Smart Contract Workflow

1. Customer purchases trigger the issuance of loyalty tokens based on a predetermined algorithm, such as a percentage of the transaction value or specific criteria.
2. The loyalty tokens are automatically assigned to Customer's digital wallet, which can be accessed via a mobile application or web interface.
3. The customer can view his accumulated tokens, check their current value, and explore the available redemption options.
4. WYD Mart's loyalty program integrates partnerships with other businesses, allowing the customer to use his tokens not only within WYD Mart but also at partner establishments.

5. When the customer decides to redeem his tokens, he can choose from a variety of options, including discounts on future purchases, exclusive products, or even converting tokens into other digital assets.
6. The redemption process is executed seamlessly through the L1X Fungible Smart Contract, ensuring instant and transparent settlement.
7. WYD Mart can analyze the transaction data recorded on the L1X blockchain to gain insights into customer behavior, preferences, and overall program effectiveness.

22.2 L1X Non-Fungible Token Smart Contract

Example-based Description:

L1X Non-fungible token (NFT) smart contracts provide a groundbreaking solution for virtual real estate ownership and trading within metaverses. By leveraging L1X NFTs, users can acquire, develop, and monetize virtual properties, creating a vibrant digital economy and immersive experiences. L1X NFT-enabled virtual real estate has the potential to position businesses at the forefront of the evolving metaverse landscape, reaping long-term benefits and opportunities.

Example:

Imagine a metaverse called "VirtuCity," a virtual world where users can explore, interact, and build their digital experiences. Within VirtuCity, there is a limited supply of virtual land parcels that can be owned and developed. The implementation of L1X NFT Smart Contracts enables users to acquire and trade these virtual properties, unlocking a multitude of possibilities.

Technical Workflow

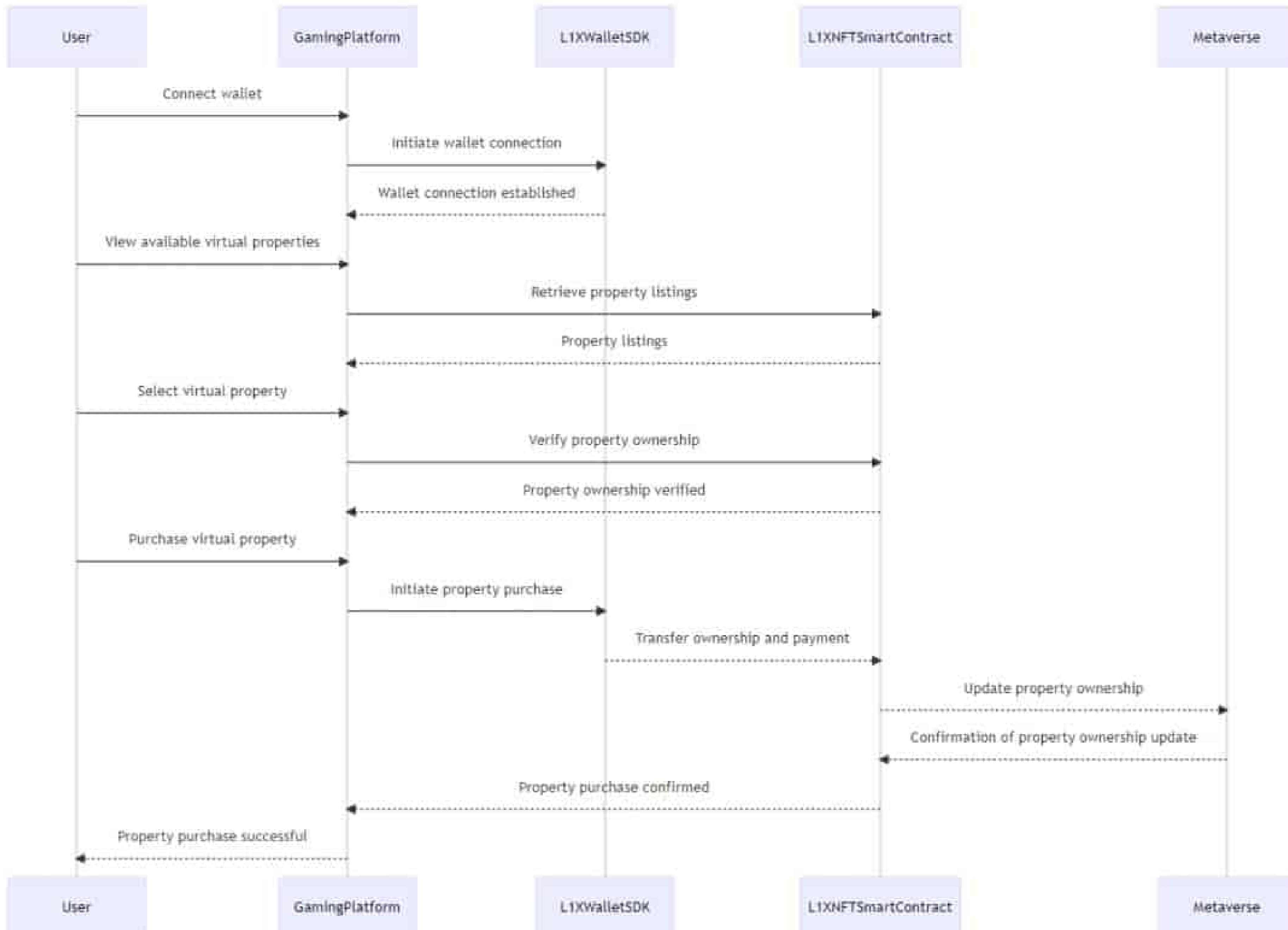


Figure 21 L1X Non-Fungible Token Smart Contract Workflow

1. **Land Tokenization:** Each virtual land parcel within VirtuCity is represented as a unique L1X NFT. L1X NFT smart contracts tokenize the properties, assigning them distinct characteristics, such as location, size, and development potential.

2. **Land Auctions:** The platform conducts auctions where users can bid on available virtual land parcels. L1X NFT smart contracts facilitate the bidding process, recording transparent and immutable transaction details on the L1X blockchain.

3. **Property Development:** Once a user becomes the owner of a virtual land parcel, they can use L1X NFT smart contracts to develop and customize their property within the metaverse. This can include constructing buildings, creating virtual businesses, or hosting events.

4. **Virtual Economy and Trade:** L1X NFT smart contracts enable users to monetize their virtual properties by engaging in commerce within VirtuCity. Users can sell or lease their developed virtual properties, providing opportunities for virtual businesses, events, and experiences.

5. **Land Ownership Rights:** The ownership of virtual land is securely recorded on the L1X blockchain through L1X NFT smart contracts, ensuring provable and transferable ownership rights. This allows for seamless and transparent trading of virtual properties.

6. **Governance and Decentralization:** L1X NFT smart contracts can be utilized for decentralized governance within VirtuCity, allowing landowners to participate in decision-making processes that affect the metaverse's development, rules, and policies.

22.3 X-Talk Token Swap

Example-based Description:

The ability to swap up a Fungible Token on Chain A with the Fungible Token on Chain B. The two chains may be EVM or Non-EVM compatible blockchain networks. The source token will automatically be burnt on Chain A, and the equivalent destination token will be minted on Chain B. This allows the Fungible Token to be swapped and used on any platform for various purposes.

Example:

The ability to swap a fungible token on Binance Smart Chain with a fungible token on Solana. The source fungible token will be automatically swapped with the destination fungible token. The token will be usable on the destination chain for transaction fees, staking, governance, and trading.

Technical Workflow

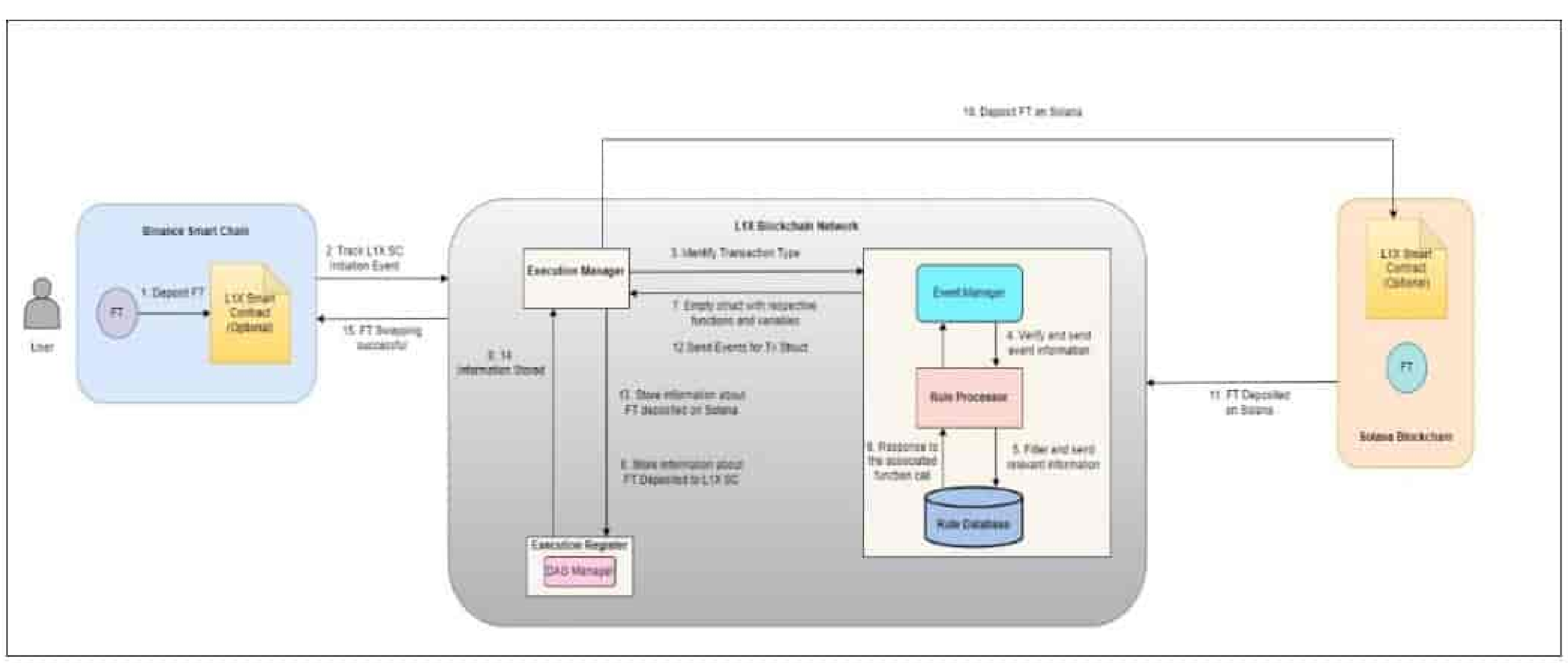


Figure 22 X-Talk Token Swap Workflows

Workflow:

1. User deposits Fungible Token to the L1X Smart Contract on the Binance Smart Chain. L1X Smart Contract sends these tokens to the L1X Liquidity Pool.
2. User initiates the Smart Contract on the L1X Chain that makes the Execution Manager ready in terms of listening to the L1X Smart Contract initiation events.
3. Execution Manager informs the Rule Engine about Transaction Type. In this use case, it's 'L1X_Swap'.
4. Rule Engine verifies the event and forwards it to the Rule Processor.
5. Rule Processor filters and sends relevant information to the Rule Database.
6. Rule Database gives the response for the associated function call.
7. Rule Engine sends an event for an empty struct to the Execution Manager with function/rules and its relevant variable/function parameter details.
8. Execution Manager requests the Execution Register to store the information about Fungible Token deposited to the L1X Smart Contract.
9. Execution Register stores the event information and informs Execution Manager.
10. Execution Manager invokes L1X Smart Contract on the Solana blockchain network to send Fungible Token from L1X Liquidity Pool to the destination address.
11. Event is triggered once Fungible Token is successfully transferred to the Solana destination address.
12. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
13. Execution Manager provides the information about 'the successful Fungible Token swapping to the Solana network', to Execution Register.
14. Execution Register stores the information and updates the Execution Manager.

15. User is informed about successful FT swapping and availability of the Fungible Token on the Solana Blockchain network.

22.4 X-Talk Token Staking

Example based Description:

The ability to mint and own a native FT on Chain A and stake it as a destination native FT on Chain B. Both chains may be EVM or Non-EVM compatible blockchain networks. The source token will automatically be locked on Chain A and the equivalent destination token will be staked on Chain B. This allows the Fungible Token to be locked on one platform and staked as the destination native Fungible Tokens on any other platform.

Example:

The ability to mint a DAI on Ethereum blockchain and stake it as RAY on Solana blockchain network with a Smart Contract on L1X. DAI, the source fungible token will be locked and equivalent RAY, the destination fungible token would be staked. The token will be usable on the destination chain for consensus, network security and governance.

Technical Workflow

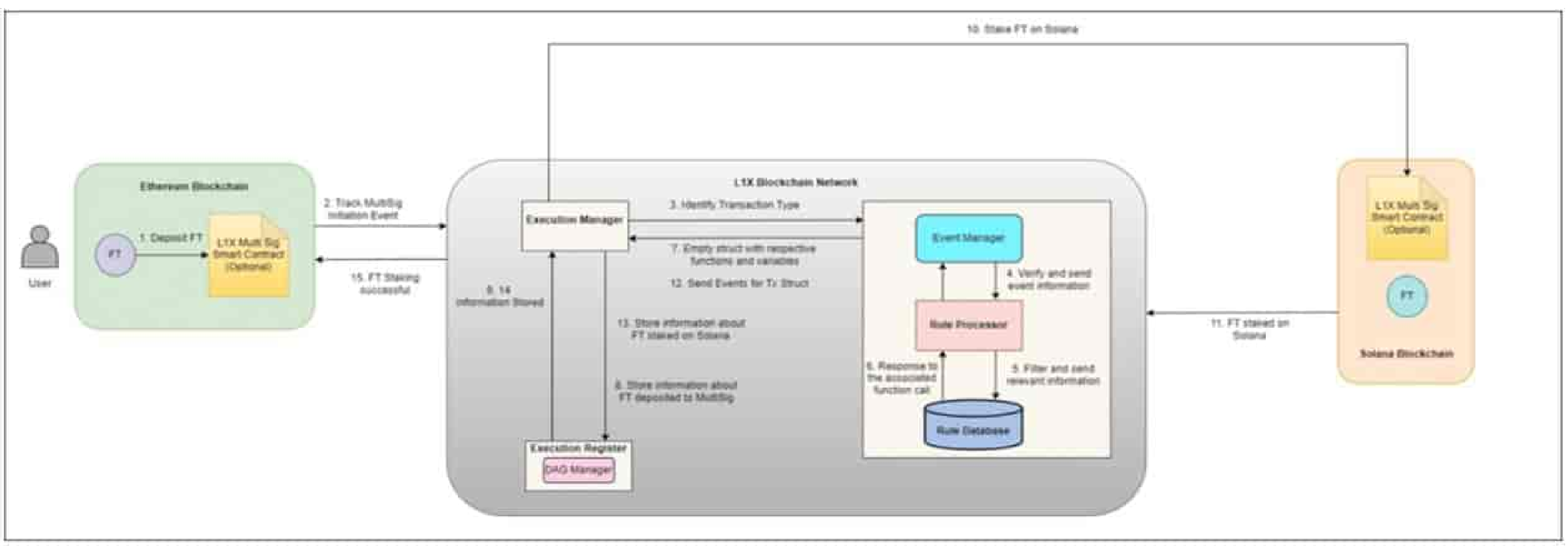


Figure 23 X-Talk Token Staking Workflow

Workflow:

1. User deposits FT to the L1X MultiSig smart contract on the Ethereum. L1X MultiSig Smart Contract transfers these FT to L1X DEX.

2. User initiates the Smart Contract on the L1X Chain that makes the Execution Manager ready in terms of listening to the MultiSig contract initiation events.
3. Execution Manager informs the Rule Engine about Transaction Type. In this use case, it's **'L1X_Stake'.**
4. Rule Engine verifies the event and forwards it to the Rule Processor.
5. Rule Processor filters and sends relevant information to the Rule Database.
6. Rule Database gives the response for the associated function call.
7. Rule Engine sends an event for an empty struct to the Execution Manager with function/rules and its relevant variable/function parameter details.
8. Execution Manager requests the Execution Register to store the information about FT deposited to the MultiSig.
9. Execution Register stores the event information and informs Execution Manager.
10. Execution Manager invokes L1X MultiSig Smart Contract on Solana blockchain network to stake FT through L1X DEX.
11. Event is triggered once FT is successfully staked on Solana.
12. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
13. Execution Manager provides the information about successfully FT staking on Solana to Execution Register.
14. Execution Register stores the information and updates Execution Manager.
15. User is informed about successful FT staking on Solana.

22.5 X-Talk Fungible Token Lending and Borrowing

Example-based Description:

The ability to lend a Fungible Token on Chain A and borrow the Fungible Token on Chain B. Both chains may be EVM or Non-EVM compatible blockchain networks. The source token will automatically be locked on Chain A and the equivalent destination token will be minted on Chain B. This allows the Fungible Token to be lent on one platform and used as collateral to borrow Fungible Tokens on any other platform for various purposes.

Example:

The ability to lend a fungible token on Avalanche and borrow Fungible Tokens on Solana. The source fungible token will be lent, and the destination fungible token would be borrowed. The token will be usable on the destination chain for transaction fees, staking, governance, and trading.

Technical Workflow

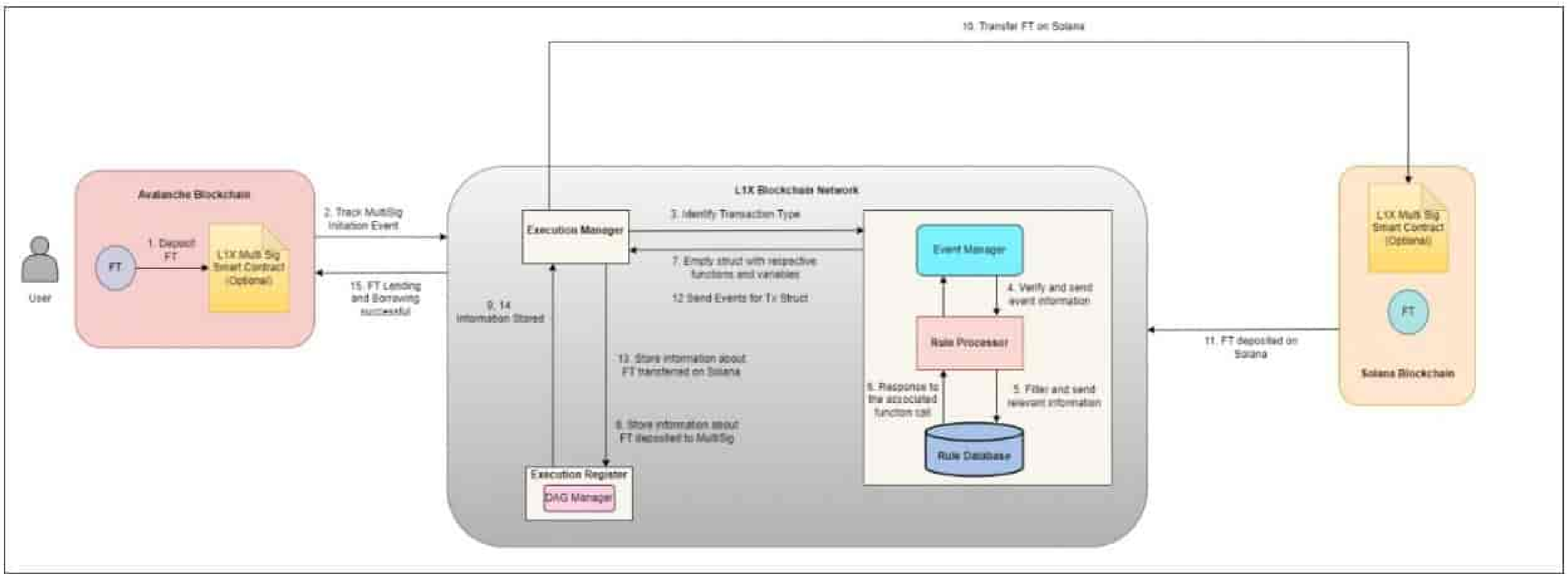


Figure 24 X-Talk Fungible Token Lending and Borrowing Workflow

Workflow:

1. User deposits Fungible Token to the L1X MultiSig smart contract on the Avalanche Blockchain. L1X MultiSig Smart Contract sends these tokens to the L1X Liquidity Pool.
2. User initiates the Smart Contract on the L1X Chain that makes the Execution Manager ready in terms of listening to the MultiSig contract initiation events.
3. Execution Manager informs the Rule Engine about Transaction Type. In this use case, it's '**L1X_LendBorrow**'.
4. Rule Engine verifies the event and forwards it to the Rule Processor.
5. Rule Processor filters and sends relevant information to the Rule Database.
6. Rule Database gives the response for the associated function call.
7. Rule Engine sends an event for an empty struct to the Execution Manager with function/rules and its relevant variable/function parameter details.
8. The Execution Manager requests the Execution Register to store the information about Fungible Token deposited to the MultiSig.
9. Execution Register stores the event information and informs Execution Manager.

10. Execution Manager invokes L1X MultiSig Smart Contract on the Solana blockchain network for the L1X Liquidity Pool to transfer FT to the destination address.
11. Event is triggered once FT is successfully transferred to the destination address on the Solana blockchain.
12. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
13. The Execution Manager provides information about successful FT borrowing on the Solana network to Execution Register.
14. Execution Register stores the information and updates the Execution Manager.
15. User is informed about successful FT lending on Avalanche and borrowing on Solana.

22.6 X-Talk NFT Liquidity Provision

Example based Description:

The ability to mint and own an NFT on Chain A and selling the NFT on OpenSea and/or Chain B and/or Ethereum Marketplace (More Marketplaces to be added). The NFT will automatically be burned / custom actioned when the NFT changes ownership on the chain(s) or marketplace(s). This allows the similar NFT to be used on various platforms while preserving the utility and permissions.

Example:

The ability to mint an NFT on Binance Smart Chain, list it on OpenSea Ethereum Marketplace with a Smart Contract on L1X. The NFT will be automatically burned when it changes ownership on the Binance Smart Chain or Ethereum OpenSea Marketplace. The NFT will be usable on the Game on the Binance Smart Chain while this is listed on OpenSea Ethereum Marketplace.

Technical Workflow

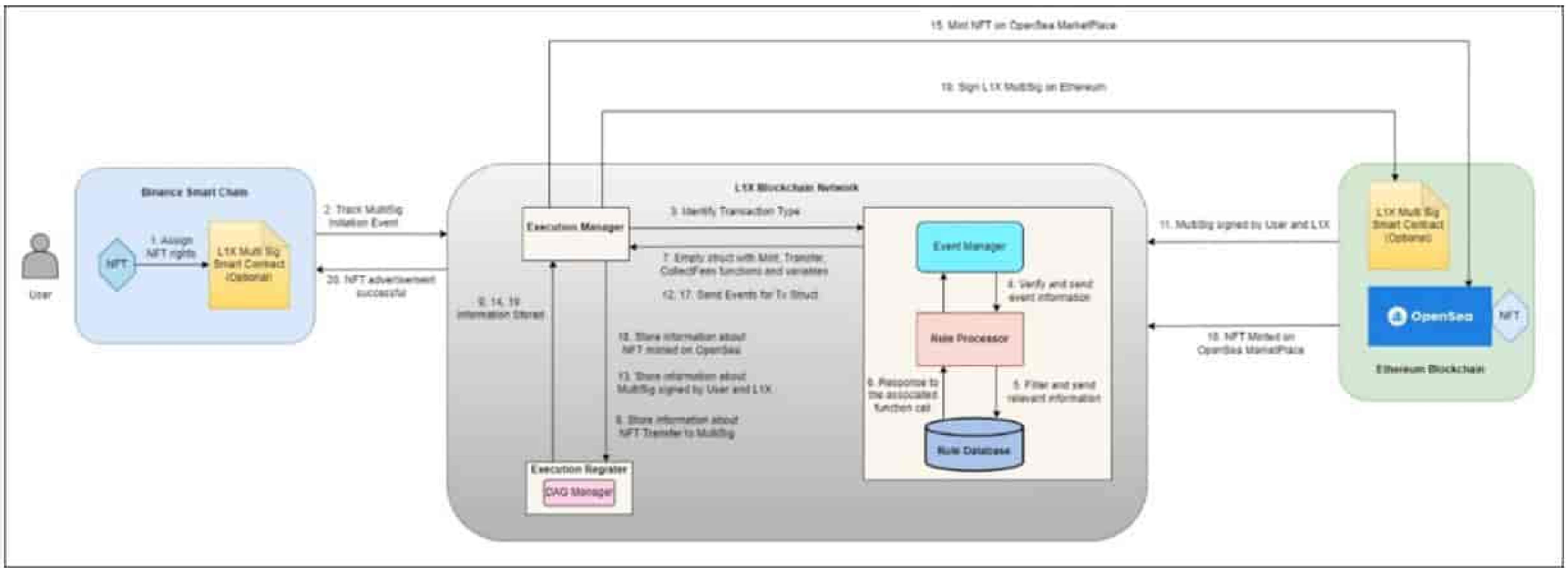


Figure 25 X-Talk NFT Liquidity Provision Workflow

Workflow:

1. User assigns NFT rights to the L1X MultiSig smart contract on the Binance Smart Chain.
2. User initiates the Smart Contract on the L1X Chain that makes the Execution Manager ready in terms of listening to the MultiSig contract initiation events.
3. Execution Manager informs the Rule Engine about Transaction Type. In this use case, it's 'L1X_Advertisement'.
4. Rule Engine verifies the event and forwards it to the Rule Processor.
5. Rule Processor filters and sends relevant information to the Rule Database.
6. Rule Database gives the response for the associated function call.
7. Rule Engine sends an event for an empty struct to the Execution Manager with function/rules such as Mint, Transfer, Collect Fees, and its relevant variable/function parameter details.
8. Execution Manager requests Execution Register to store the information about NFT transfer to the MultiSig.
9. Execution Register stores the event information and informs Execution Manager.
10. Execution Manager invokes the user and L1X to sign the MultiSig on the Ethereum blockchain network.
11. L1X MultiSig Smart Contract triggers an event once the MultiSig is signed by the user and L1X.

12. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
13. Execution Manager updates the Execution Register to store information that L1X MultiSig on Ethereum blockchain network is signed by the user and L1X.
14. Execution Registers informs Execution Manager about information storage.
15. Execution Manager invokes L1X MultiSig Smart Contract on Ethereum blockchain network to mint NFT on OpenSea Marketplace.
16. Event is triggered once NFT is successfully minted on OpenSea Marketplace.
17. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
18. Execution Manager provides the information about successfully NFT minting on OpenSea Marketplace to Execution Register.
19. Execution Register stores the information and updates Execution Manager.
20. User is informed about successful NFT advertisement and listing on Ethereum OpenSea marketplace.

22.7 X-Talk NFT Leasing Provision

Example-based Description:

The ability to lease an NFT on Chain A by the lessor and the lessee utilizing it on Chain B. X-Talk allows leasing NFT across any blockchain network. The NFT will automatically be custom actioned when the agreed leasing term completes. This allows the NFT to be leased from other platforms and utilized for various purposes.

Example:

The ability to mint and own an NFT on Solana and lease it on Polygon. The NFT on Solana will be locked, and the lessee will be able to use it on Polygon. The NFT on the destination chain can be used, displayed, and monetized in accordance with the agreed terms.

Technical Workflow

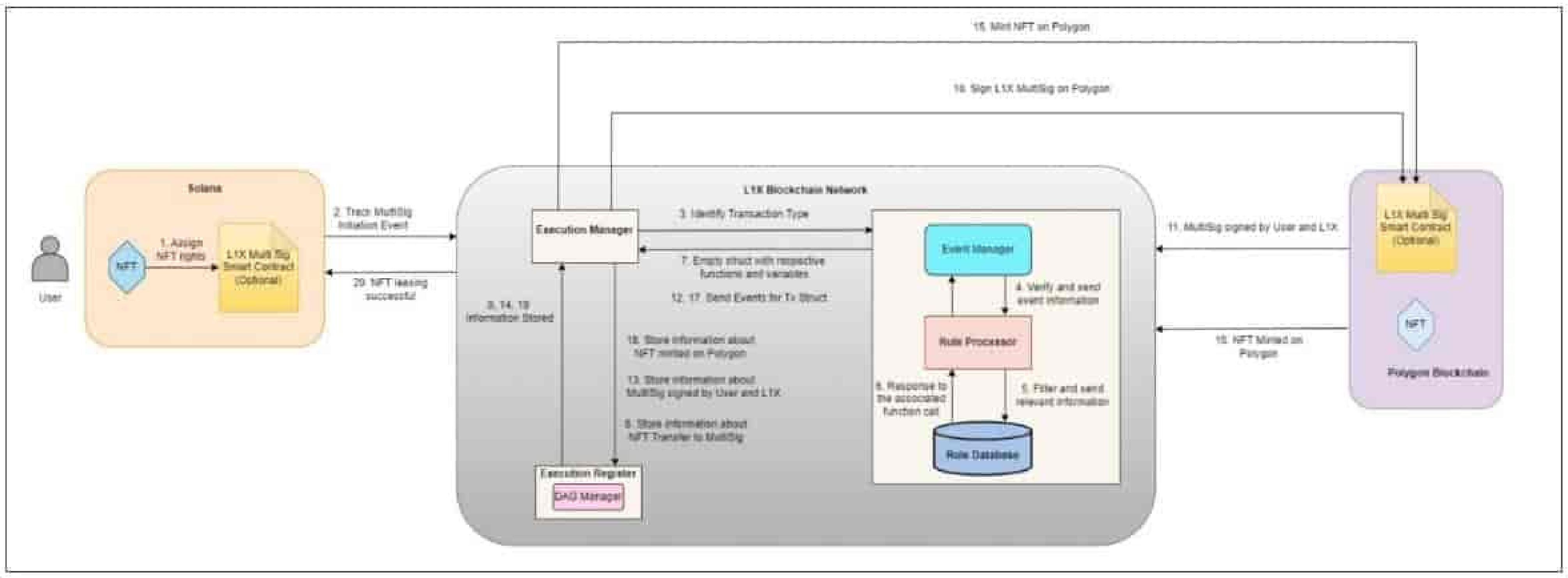


Figure 26 X-Talk NFT Leasing Provision Workflow

Workflow:

1. User assigns NFT rights to the L1X MultiSig smart contract on the Solana.
2. User initiates the Smart Contract on the L1X Chain that makes the Execution Manager ready in terms of listening to the MultiSig contract initiation events.
3. Execution Manager informs the Rule Engine about Transaction Type. In this use case, it's **'L1X_Lease'.**
4. Rule Engine verifies the event and forwards it to the Rule Processor.
5. Rule Processor filters and sends relevant information to the Rule Database.
6. Rule Database gives the response for the associated function call.
7. Rule Engine sends an event for an empty struct to the Execution Manager with function/rules and its relevant variable/function parameter details.
8. Execution Manager requests the Execution Register to store the information about NFT transfer to the MultiSig.
9. Execution Register stores the event information and informs Execution Manager.
10. Execution Manager invokes the user and L1X to sign the MultiSig on the Polygon blockchain network.
11. L1X MultiSig Smart Contract triggers an event once the MultiSig is signed by the user and L1X.
12. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
13. Execution Manager updates the Execution Register to store information that L1X MultiSig on the Polygon blockchain network is signed by the user and L1X.

14. Execution Registers informs the Execution Manager about information storage.
15. Execution Manager invokes L1X MultiSig Smart Contract on Polygon blockchain network to mint NFT.
16. Event is triggered once NFT is successfully minted on Polygon.
17. Rule Engine sends the corresponding event for the transaction struct to the Execution Manager.
18. Execution Manager provides the information about successfully NFT minting on Polygon to Execution Register.
19. Execution Register stores the information and updates Execution Manager.
20. User is informed about successful NFT leasing on Polygon.

22.8 Health Smart Contract

Example-based Description:

L1X Health Smart Contract usage for maintaining patient health records and facilitating appointment booking brings numerous benefits in terms of data security, patient empowerment, and streamlined processes. Investors recognizing the potential of the L1X Health Smart Contract standard can position themselves at the forefront of the evolving healthcare industry, capitalizing on the transformative power of blockchain technology.

Example:

Consider a decentralized healthcare platform, which aims to provide patients with ownership and control over their health records while facilitating seamless appointment booking with doctors. Through L1X Health Smart Contract, the HealthCare platform ensures the privacy, integrity, and accessibility of patient records while simplifying the appointment booking process.

Technical Workflow

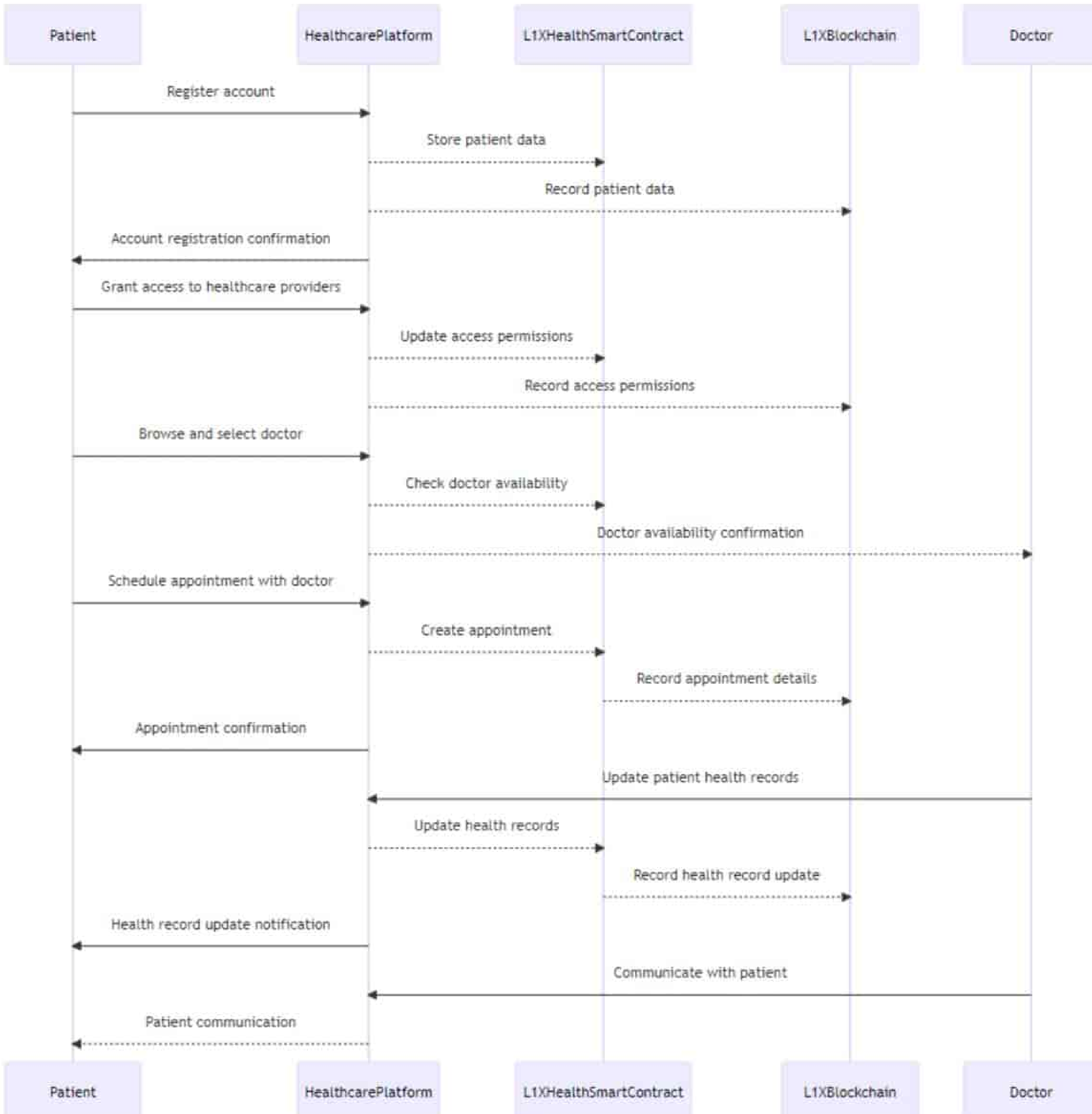


Figure 27 Health Smart Contract Workflow

1. **Patient Registration**: Patients create an account on the HealthCare platform, providing necessary personal information and medical history.
2. **Health Record Storage**: Patient health records, including medical history, lab results, and prescriptions, are securely stored on the L1X blockchain, ensuring immutability and tamper-proof data integrity.

3. **Access Control**: Patients have sole ownership and control over their health records. They can grant permission to healthcare providers, such as doctors, clinics, or hospitals, to access their records for treatment purposes.
4. **Appointment Booking**: Patients can browse through the profiles and availability of doctors on the HealthCare platform. They can select a suitable doctor and schedule an appointment directly through the L1X Health Smart Contract.
5. **Smart Contract Verification**: The L1X Health Smart Contract verifies the availability of the selected doctor based on their schedule, ensuring appointment accuracy and avoiding conflicts.
6. **Appointment Confirmation**: Once the appointment is successfully scheduled, the L1X Health Smart Contract records the appointment details, including date, time, and doctor's information, on the L1X blockchain.
7. **Real-Time Updates**: Patients and doctors can access real-time updates on their appointments, including any rescheduling or cancellations, through the HealthCare platform, enhancing communication and reducing no-shows.
8. **Secure Communication**: Through the platform, patients and doctors can securely communicate regarding pre-appointment instructions, post-appointment follow-ups, and other healthcare-related discussions.
9. **Health Record Updates**: During the appointment, doctors can update the patient's health records with diagnosis, treatment information, and prescribed medications, ensuring comprehensive and up-to-date medical records.
10. **Data Privacy and Consent**: Patient data remains private and is shared only with authorized healthcare providers based on patient consent and specified access permissions.

22.9 L1X Wallet SDK

Example-based Description:

The integration of L1X Wallet SDK in a tokenized gaming ecosystem empowers players to securely manage, trade, and showcase their in-game currencies, virtual assets, and NFTs. By leveraging L1X Wallet SDK, gaming platforms can create engaging and player-driven economies, revolutionizing the way players interact with and derive value from their digital gaming experiences.

Example:

Imagine a multiplayer online game where players embark on quests, collect virtual assets, and engage in player-to-player trading. Through the integration of the L1X Wallet SDK, players can conveniently manage their in-game currencies, virtual items, and NFTs within the game's ecosystem.

Technical Workflow

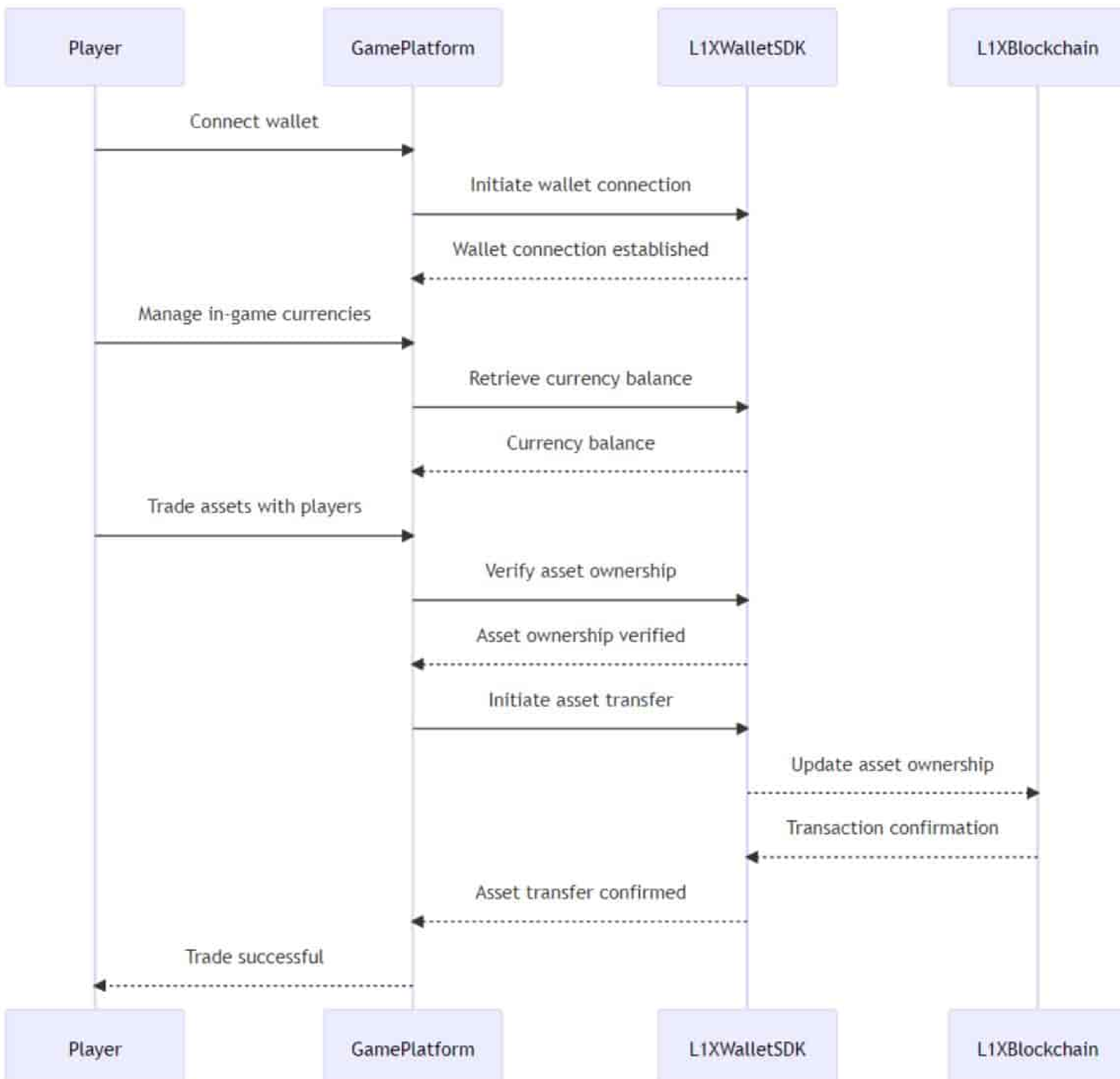


Figure 28 L1X Wallet SDK Workflow

1. **Wallet Integration**: The XYZ game integrates the L1X Wallet SDK into its platform, allowing players to connect their wallets securely.
2. **Wallet Connection**: The user initiates the wallet connection process within the game interface and selects her L1X cryptocurrency wallet. The L1X Wallet SDK facilitates secure authentication and establishes a connection between the game and the user's wallet.
3. **In-Game Currency Management**: The user can view and manage her in-game currencies, such as gold coins or special tokens, directly from her connected L1X Wallet. She can track her balance, initiate transactions, and participate in the game's economy seamlessly.
4. **Virtual Asset Ownership**: The L1X Wallet SDK securely records and verifies the user's ownership of virtual assets and NFTs within her L1X wallet. She can easily view her collection, transfer assets to other players, or showcase her rare and valuable items.
5. **Peer-to-Peer Trading**: The L1X Wallet SDK enables the user to engage in player-to-player trading within the game. She can initiate trade offers, negotiate terms, and securely exchange assets with other players, creating a dynamic and player-driven economy.
6. **Secure Asset Storage**: The user's virtual assets, including in-game currencies and NFTs, are securely stored within her L1X Wallet. The L1X Wallet SDK ensures the protection of her assets from unauthorized access or fraudulent activities.
7. **CrossChain-CrossGame Compatibility**: If the user plays other games across different blockchain networks, the L1X Wallet SDK allows her to seamlessly transfer and utilize her assets across different games, promoting interoperability and enhancing the overall gaming experience.

22.10 Multichain Balancer Pool

In the ever-evolving landscape of decentralized finance (DeFi), the need for interoperability between blockchain networks is paramount. The current limitations of a Balancer Pool, which restricts token hosting to a single chain, necessitate a pioneering solution. This section presents a comprehensive proposal for the implementation of a Multichain Balancer Pool, leveraging the capabilities of Layer One X (L1X) blockchain network.

L1X being equipped with its own Virtual Machine, decentralized consensus mechanisms, and state management system provides remarkable cross-chain interoperability features such as:

- **Interoperability with Both EVM and Non-EVM Chains:** L1X effortlessly connects the chasm between Ethereum Virtual Machine (EVM) and non-EVM-based blockchains.
- **Secure Cross-Chain Transactions:** L1X's architecture ensures secure and trustless cross-chain value transfers and data exchanges.
- **Native Smart Contract Support:** L1X's native smart contract capabilities are pivotal for creating a versatile and dynamic Multichain Balancer Pool.
- **Efficient Transactions:** With swift transaction processing and minimal fees, L1X is engineered for the rapid pace of DeFi operations.

Additionally, X-Talk empowers the Multichain Balancer Pool with the ability to communicate across distinct blockchains, ensuring a harmonious coexistence of tokens from various ecosystems. The Multichain Balancer Pool, driven by X-Talk, allows the inclusion of tokens from multiple chains, resulting in unparalleled diversification opportunities. Tokens such as USDC (Ethereum), USDT (Avalanche), USDC (Binance Smart Chain), and beyond can coalesce within a single, dynamic pool.

22.10.1 Functionalities

In this section, we delve into the multifaceted functionalities of the proposed Multichain Balancer Pool. Explore how it enhances accessibility, liquidity, and engagement in the decentralized financial realm.

1. **Swaps Across Various Protocols**

   In the realm of decentralized finance, the ability to seamlessly exchange assets across diverse blockchain protocols is paramount. The Multichain Balancer Pool introduces groundbreaking "Swaps Across Various Protocols," facilitating cross-chain token swaps and liquidity provision, all while minimizing price slippage, thus redefining the DeFi trading experience.

Features

- **Cross-Chain Swaps:** At the heart of the Multichain Balancer Pool is the ability to facilitate seamless token swaps between a myriad of blockchains, regardless of their underlying technology.

- **Liquidity Provision Across Chains:** Liquidity providers can contribute assets to the pool from different chains, creating a robust ecosystem that caters to users from diverse blockchain communities.

- **Slippage Control:** Advanced measures are in place to mitigate price slippage during substantial trades, bolstering trading strategies and user experience.

Benefits

- **Unparalleled Accessibility:** The Multichain Balancer Pool opens doors to a broader array of assets without the need for users to navigate multiple decentralized exchanges.

- **Enhanced Liquidity:** Consolidating liquidity across different chains ensures ample trading volume and, consequently, better market conditions.

- **Optimized Efficiency:** By minimizing the necessity for complex multi-step transactions, users enjoy a more streamlined trading experience.

Web3 Market Opportunities

- **Expanded Market Reach:** This innovation unlocks fresh opportunities for traders and investors from diverse blockchain networks, fostering a more interconnected DeFi ecosystem.

- **Catalyst for Innovation:** The Multichain Balancer Pool's cross-chain capabilities inspire the development of novel financial products and services that leverage this newfound interoperability.

2. Distribution of Rewards to Balancer Pool Contributors

Distribution of rewards to Balancer Pool contributors" is a pivotal element of the Multichain Balancer Pool, ensuring fairness and engagement within the ecosystem. This section unveils a sophisticated system of automated reward distribution, multi-asset staking, and customizable reward structures, promoting a vibrant and inclusive DeFi community while enabling users to diversify their portfolios.

Features

- **Automated Reward Distribution:** Smart contracts underpin an automated reward distribution system, ensuring equitable compensation based on each contributor's participation.
- **Multi-Asset Staking:** The Multichain Balancer Pool empowers users to stake a multitude of tokens from various chains, allowing for diversified, cross-chain exposure.
- **Customizable Reward Structures:** Pool creators have the freedom to craft unique reward structures tailored to their pool's specific objectives.

Benefits

- **Fair Incentivization:** Contributors are rewarded proportionally to their contributions, promoting a fair and inclusive DeFi landscape.

- **Increased Engagement:** Customizable rewards foster a sense of community and dedication among liquidity providers.

- **Diversified Portfolios:** Multichain exposure through staking leads to a diversified portfolio, spreading risk intelligently.

Web3 Market Opportunities

- **Community Strengthening:** The Multichain Balancer Pool encourages the growth of a robust, decentralized community of contributors spanning various blockchain ecosystems, enhancing the resilience of DeFi.

- **Innovative Investment Horizons:** By offering novel investment avenues catering to diverse investor preferences and risk profiles, this solution drives innovation within decentralized finance.

The Multichain Balancer Pool, thus, hinging on the cross-chain capabilities of Layer One X and the transformative potential of the X-Talk component, heralds a new era in DeFi. This visionary proposal dismantles the barriers between blockchain networks, empowering users with unprecedented access to assets and investment opportunities while nurturing a more interconnected and vibrant decentralized finance ecosystem.

23. Conclusion

Layer One X (L1X) presents a compelling solution to the challenges of scalability and interoperability in blockchain systems. Through its innovative design principles and advanced features, L1X aims to establish itself as an interoperable, decentralized, secure, and scalable Layer One Smart Contract Protocol.

The L1X Virtual Machine (VM) plays a crucial role in enabling efficient execution of smart contracts. The L1X VM provides a robust and optimized environment for executing smart contracts written in the Rust programming language by building innovative WASM to EBPF compiler. Additionally, the X-Talk Virtual Machine enhances interoperability by facilitating native and multi- chain interactions, opening up new possibilities for cross-chain tokenization and staking.

The consensus mechanism in L1X, known as Proof of X (PoX), introduces a novel approach to achieving decentralization and maintaining network security. By utilizing a combination of FVNs and Mobile Enabled or Low CPU Powered Devices as Validators, L1X ensures a diverse and distributed set of participants in the consensus process. Various mechanisms such as staking, randomization algorithms for block proposing, cluster allocation, and the inclusion of mobile devices as validators contribute to maintaining an inflationary growth rate of decentralization as the network scales.

The MultiSig Native On-chain Collection and the Pentagon Framework for security address critical concerns within the protocol. The MultiSig contract allows for secure authentication and validation of smart contract invocations across chains, enabling seamless cross-chain operations. The Pentagon Framework tackles challenges related to state updates, flash code logic validation, deterministic behavior of the virtual machine, randomization of block proposers and cluster registry, ensuring a secure and reliable smart contract platform.

With its focus on scalability and performance, L1X demonstrates its capability to handle high transaction volumes and adapt to evolving requirements. The Virtual Machine, Cluster Based Node Architecture, State Management, Hybrid Consensus Mechanism, and X-Talk synchronous transaction processing collectively contribute to a scalable protocol. By targeting faster Seconds per Transaction (SPT) and utilizing Zero Knowledge Proof Trailing, L1X aims to achieve both high Transactions per Second (TPS) and shorter timeframes for transaction completion.

Moreover, L1X recognizes the need for privacy and custom consensus mechanisms in specific use cases. The ability to host subnets allows governments, larger enterprises, and businesses to maintain smart contract privacy and customize consensus mechanisms while benefiting from the L1X infrastructure. The option to roll information to the main chain provides an additional layer of security and flexibility.

In summary, Layer One X (L1X) presents a comprehensive and innovative approach to address the scalability and interoperability challenges faced by blockchain systems. Through its advanced virtual machine, consensus mechanism, X-Talk interoperability, MultiSig contract, and focus on security and performance, L1X paves the way for the connecting different blockchains and unleashing their true potential through widespread adoption and sustainable growth of blockchain- oriented solutions in diverse domains.

24. Documentation Links

The following links are provided to the L1X Developer Tool Kit and the L1X Source Code Quick Links.