Whitepaper

SwissLedger: An Institutional Decentralized Ledger

Abstract

SwissLedger is a blockchain-based distributed ledger system designed for institutional applications. It is compatible with the Ethereum Virtual Machine (EVM) and uses Proof of Authority (PoA) consensus. SwissLedger combines efficiency, security, and regulatory compliance. This document describes the technical features of SwissLedger, its consensus model and potential use cases.

Introduction

Blockchain technologies have revolutionised the way digital data and transactions are managed. However, institutional adoption of these technologies requires platforms that balance security, scalability, and regulatory compliance. SwissLedger addresses these challenges by providing a DLT solution designed to meet the needs of government and corporate entities.

System Description

Ethereum Virtual Machine (EVM)

SwissLedger is based on the Ethereum Virtual Machine (EVM), a Turing-complete execution environment designed to execute smart contracts in a secure and deterministic manner. The EVM operates as an isolated virtual machine, executing bytecode code with mathematical precision and without external dependencies. Each smart contract has a dedicated permanent storage space, ensuring data persistence on the blockchain. Thanks to the sandboxing model, smart contracts are completely isolated, preventing interference or compromise by other contracts.

The EVM environment supports the compilation of languages such as Solidity, ensuring a flexible development process that is integrated with Ethereum's vast ecosystem of tools. This compatibility simplifies the writing and debugging of smart contracts. In addition, EVM enables the generation of logs through indexed events, allowing decentralised applications to monitor specific activities without burdening blockchain storage.

Proof of Authority (PoA) Consensus

SwissLedger uses a PoA model, in which a limited number of validators, authorised by trusted entities such as the City of Lugano, guarantee the integrity of the network. This model reduces energy consumption and improves efficiency compared to traditional Proof of Work (PoW) systems.

Security and Resilience

SwissLedger's PoA model incorporates a number of advanced mechanisms to ensure security and resilience. To prevent malicious behaviour, it limits the number of consecutive blocks a validator can sign, reducing the risk of power concentration and attacks by compromised validators. The system also includes a continuous monitoring process of validators based on an internal consensus mechanism, which allows validators to be added or removed based on predetermined parameters, such as reliability and behaviour during consensus.

A key element of the model is the adoption of penalties for non-compliant behaviour. These penalties may include the temporary or permanent removal of non-compliant validators, ensuring that only nodes that act correctly can continue to participate in the system.

From a technical point of view, the process of authorising a block is based on an algorithm that verifies the authenticity of the validator's signature before accepting a block into the chain. Each validator has a unique cryptographic key used to sign blocks, and the system verifies that the signature corresponds to an authorised validator. This mechanism prevents unauthorised or compromised nodes from entering invalid data into the blockchain. The generation time of each block is set to 5 seconds, ensuring rapid transaction propagation and high operational efficiency.

In addition, the model implements a validator system in which active validators can vote to add new validators or remove existing ones. Each proposed change must obtain a qualified majority to be approved, thus ensuring that decisions on the composition of the validator set are the result of a collective consensus. This process enhances network security and ensures that the system remains dynamic and adaptable to operational needs.

Finally, SwissLedger's architecture ensures redundancy and synchronisation between nodes through the use of regular checkpoints. These checkpoints contain key information, such as the current list of validators and the current state of the network, and are used to ensure data consistency between nodes. In addition, the protocol provides predetermined block times and an algorithm that dynamically adjusts the consensus to maintain high efficiency even in the event of changes in the availability of validators.

Token Standards

SwissLedger supports the main Ethereum token standards:

- **ERC-20**: For fungible tokens such as cryptocurrencies and utility tokens.
- **ERC-721**: For non-fungible tokens (NFT), which can be used in artistic and digital applications.
- **ERC-2980**: Specifically designed to represent financial assets compliant with Swiss regulations.

These standards are adapted to ensure regulatory compliance, efficiency, and security.

Security Mechanisms

Transaction Limiting

Rate limiting in the mempool is an advanced measure to ensure the security and efficiency of blockchains, especially those without transaction fees. Currently, the limit of pending transactions per single address is set at 8, based on an initial configuration defined by SwissLedger. However, as the system is completely decentralised, this setting is neither rigid nor definitive.

Each node in the network has the possibility to change this limit independently, deciding how many transactions per address it can accept. In a decentralised environment, such changes are possible at any time, since each node operates independently. However, network consistency depends on majority consensus: if a significant number of nodes adopt a different configuration, it may prevail as the new operating standard.

This flexible approach allows the network to adapt dynamically to user needs, while maintaining a balance between security and performance. In summary, rate limiting, although initially configured with a predefined limit, can be modified and adapted over time due to the decentralized nature of the network, ensuring optimal shared resource management.

Node Management

Access to the network is restricted to approved nodes, which must meet strict security criteria and adhere to the rules defined by smart contracts. This approval process ensures that only authorised, verified and secure nodes can participate in the network. Each node must provide specific credentials and meet security criteria before being considered eligible.

Account Management

Accounts are associated with specific permissions determined by smart contracts. These permissions are defined according to a hierarchical system that allows specific access and privileges to be established for each account. Access levels include:

- **ReadOnly (0):** Read-only access, which allows data to be viewed without making changes.
- **Transact (1):** Permission to perform transactions, allowing interaction with contracts and network data.
- **ContractDeploy (2):** Permission to deploy smart contracts, enabling the creation of new functionality in the network.
- **FullAccess (3):** Full access, which includes the ability to manage permissions, approve nodes and modify network configurations.

Each account must be approved before being activated, and changes to permissions must follow strict procedures to avoid errors or abuse.

Smart Contract Based Implementation

All node and account management operations are based on smart contracts. This approach enables automated and transparent configuration, reducing the possibility of human error. Nodes dynamically configure themselves according to the network administration smart contracts, providing distributed and secure control. These smart contracts provide an auditable and immutable system to monitor and manage activities, ensuring that network rules are strictly followed. The use of smart contracts enhances security and ensures that any changes in the network are traceable and comply with established policies.

Anchoring System to Bitcoin

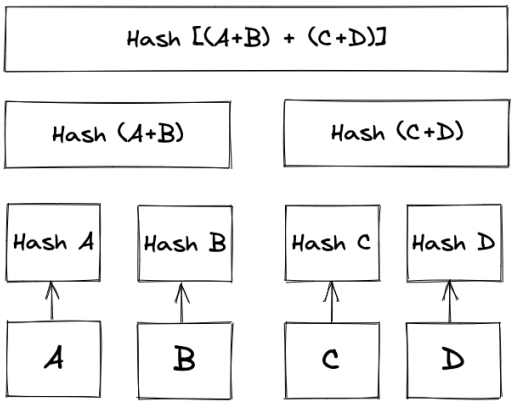
The anchoring system of an EVM blockchain to Bitcoin is crucial to ensure the immutable notarisation of the EVM blockchain state, leveraging the robustness and security of the Bitcoin network. This process allows a representative hash of the EVM blockchain state to be recorded within a Bitcoin transaction, providing an independently verifiable cryptographic proof associated with a precise instant in time.

The need for such a system stems from the need to increase trust in the information recorded on the EVM blockchain by incorporating a higher level of security via Bitcoin, the most established and attack-resistant blockchain. Through the use of OpenTimestamps and a smart contract on EVM, a fully decentralised process for storing and verifying notarised information is achieved, eliminating dependence on centralised infrastructure. This system allows anyone to transparently verify that the data on EVM blockchain was recorded at a specific point in time, ensuring integrity and immutability.

Technical Description of Timestamping with OpenTimestamps

The OpenTimestamps protocol allows digital documents to be anchored to the Bitcoin blockchain to ensure immutability and time verification. OpenTimestamps servers collect hashes of documents generated by users (OpenTimestamps clients). Periodically, these hashes are organised in a Merkle Tree, generating a Root Hash that is inserted into a Bitcoin transaction (OP_RETURN field). This process performs the timestamping operation.

To verify that a document has been recorded correctly, the process proceeds by calculating the hash of the document, combining it with intermediate hashes until the Root Hash is reconstructed. If the latter matches the one recorded in the blockchain, the operation is confirmed.



Anchoring process

1. **Time division of the SwissLedger blockchain**

   The SwissLedger blockchain is divided into regular intervals [t; t+T], where T represents the duration of the interval to be anchored to Bitcoin.

2. **Calculation of the blockhash hash**

   For each interval, the blockhashes of the blocks between t and t+T are concatenated in order of block number and the overall hash is calculated:

   `Hash(Block(t) + Block(t+1) + ... + Block(t+N))`.

3. **Creation of information files**

   The calculated hash is saved in a `.txt` file containing alphanumeric data. Then, with the OpenTimestamps client, an `.ots` file is generated that includes all the information required for timestamping and verification.

4. **Uploading and managing files**

   The `.txt` and `.ots` files are uploaded to an OpenTimestamps server, which takes care of updating the .ots file with partial hashes to reconstruct the Root Hash. This is then inserted into a Bitcoin transaction, ensuring time anchoring.

5. **Decentralised storage**

   The hashes and related files are recorded on a smart contract deployed on the SwissLedger blockchain. This smart contract keeps track of the correspondence between block number and concatenated hash, allowing users to access the files via a URL provided by the contract itself.

6. **Independent verification**

   Users can verify the integrity of the `.txt` file using a blockchain explorer to retrieve the necessary blockhash. In addition, it is possible to verify the `.ots` file through third-party services, as provided by the OpenTimestamps protocol. All the information to perform these verifications is publicly available and guarantees a transparent process.

Advantages of the system

- **Immutability and transparency**

  The use of SwissLedger blockchain and Bitcoin ensures that data cannot be changed or deleted.

- **Decentralised verification**

  Management via smart contracts eliminates the need for centralized platforms, allowing direct and secure access to data by users.

- **Flexible storage**

  Files can be stored on distributed repositories (such as IPFS) or centralised, depending on the needs of the system.

Use Cases *(non-exhaustive list)*

Digital Identity

SwissLedger enables the creation of secure, verifiable digital identities that comply with data protection regulations.

Supply Chain Traceability

The platform provides tools to track products along the entire supply chain, ensuring transparency and authentication of assets.

Tokenisation of Financial Assets

Thanks to Ethereum-compatible standards, SwissLedger facilitates the digital representation of stocks, bonds and derivatives.

Gaming and Entertainment Applications

NFT support enables the creation of unique digital assets, ideal for gaming platforms and digital collectibles.

Conclusion

SwissLedger is an institutional blockchain platform designed to address security, efficiency and regulatory compliance challenges. Its PoA system, combined with advanced functionality and a focus on innovation, is an ideal solution for government agencies and corporations wishing to adopt blockchain technology in a secure and scalable manner.