Bedrock IOTX Liquid Staking Explained

Bedrock Team

September 30, 2023

Abstract

Staking in the crypto economy is gaining substantial momentum as Proof of Stake (POS) emerges as the prevailing consensus mechanism in the blockchain industry. In the IoTeX Network, token-holders can stake IOTX and vote for Delegates to support and expand the network. This not only strengthens the overall security of the IoTeX Network, but also provides token-holders with significant returns on their staked IOTX. Various staking methods are available, including Solo Staking, SaaS Staking, and staking via centralized exchanges. Each method has its own limitations. To improve the user experience on the IoTeX network, we have launched the Bedrock Liquid Staking (IoTeX) project.

1 Introduction

1.1 IoTeX Network

IoTeX is a blockchain network that employs Roll-DPoS consensus mechanism, which enhances the decentralization and security of the IoTeX Network without compromising performance by randomly selecting 24 out of the top 36 community-voted Delegates to mine every hour.

1.2 Staking Parameters

Before we start Staking on the IoTeX Network, several factors need to be taken into account. We will briefly discuss these parameters from the network's perspective.

- Delegate: More than 70 delegates each make a unique contribution to IoTeX and offer different reward amounts to voters. You have the option to change your vote to another delegate whenever you wish.
- Amount: This determines the amount of IOTX you wish to stake/vote. You can append IOTX to an existing bucket after your initial vote, provided the stake lock is activated. The more IOTX you stake, the greater your rewards will be.
- Stake Duration: This is the amount of time you wish to stake your coins (between 0-1050 days). The longer you stake, the more bonus votes or rewards you receive. The duration of your stake is essentially a countdown timer indicating when you can unstake your IOTX. Once the duration reaches zero, you have the option to unstake your deposit. Please be aware that after the stake duration has expired, the unstaking process requires an additional 3 days. Following this, you may withdraw your coins to your wallet.
- Stake-Lock(Auto-Stake): When you turn stake-lock ON, you pause your lock duration countdown until you decide to turn it OFF, and the countdown resumes. The advantage of stake-lock is that it allows you to earn additional bonus votes/rewards while activated. Once you deactivate the stake-lock, your stake duration will continue to count down to zero. Please note, if you reactivate the stake-lock at any point during your countdown, your stake duration will reset to the original period you had initially set.

1.3 Liquid Staking

IIP-13(IoTeX Improvement Proposal 13) proposes to natively support the representation of staking buckets as Non-fungible Tokens(NFT) and clears the way for Liquid Staking DApps and interest-earning derivatives to flourish in the IoTeX ecosystem.

SystemStaking is the contract implemented for IIP-13. It issues an NFT token for each bucket creation. Owner of the NFT token could update/transfer/unstake the corresponding bucket. The buckets created in this contract will be counted in the staking protocol in iotex-core.

1.4 Universal IOTX

To enhance the liquid staking experience on the IoTeX network, our protocol issues a **Universal IOTX** token, symbolized as **uniIOTX**, for staking services. Users will accrue uniIOTX when they deposit the native token, IOTX, into our staking pool. The value of uniIOTX will increase over time and provide liquidity, all without the need for intricate technical knowledge.

2 System Architecture

Figure 1 provides a depiction of the system's high-level architecture and context. This project is built on the IoTex network, involving various collaborative roles and modules. The smart contracts form the core component of the entire system. This is a concise overview of the responsibilities of subsystems:

- Exchange Service: This is an additional service designed to boost liquidity. Our protocol stipulates that users can only redeem IOTX in units of 1,000,000. However, we also provide an essential liquidity pool, enabling users to exchange uniIOTX for IOTX at any time through an eco-friendly trading platform such as a DEX. Please note that this service is currently in the planning stage and will be available soon.
- UniIOTX DApp: This is a Web 3.0 decentralized application. It mainly includes features that allow users to deposit, redeem, and claim IOTX for investment profits.
- Oracle System: This is an off-chain backend system. It primarily manages delegate and debt, synchronizes rewards, and facilitates compounding for re-staking.
- On-Chain Contracts: Overall, our liquid staking protocol is executed through the joint effort of four smart contracts: SystemStaking, UniIOTX, IOTXClear, and IOTXStaking. IOTXStaking relies on the other three contracts, while the IOTXClear contract solely depends on the SystemStaking contracts.
  - SystemStaking: This is the official IoTeX staking contract, a standard ERC-721 smart contract that facilitates bucket management. It inherently generates an NFT token for each bucket created and destroys the NFT token when its corresponding bucket is withdrawn.
  - UniIOTX: A standard ERC-20 smart contract which supports Batch Transfer. It generates uniIOTX when users deposit IOTX or when the manager fee is withdrawn, and eliminates uniIOTX when users redeem IOTX. Additionally, users can transfer or authorize the spending of uniIOTX to other users.
  - IOTXStaking: A smart contract designed to accept user-transferred value for an automatic compounding staking service. It involves the distribution of staking rewards and represents the initial stage of the business lifecycle. Furthermore, it transfers the relevant NFT to the IOTXClear contract for a new debt record upon receiving users' redemption requests.
  - IOTXClear: A smart contract designed to manage debt. It systematically organizes debt records based on users' redemption requests and adheres to a First-In-First-Out (FIFO) order for payment. Furthermore, it also keeps track of the yield rewards during the unstaking phase.

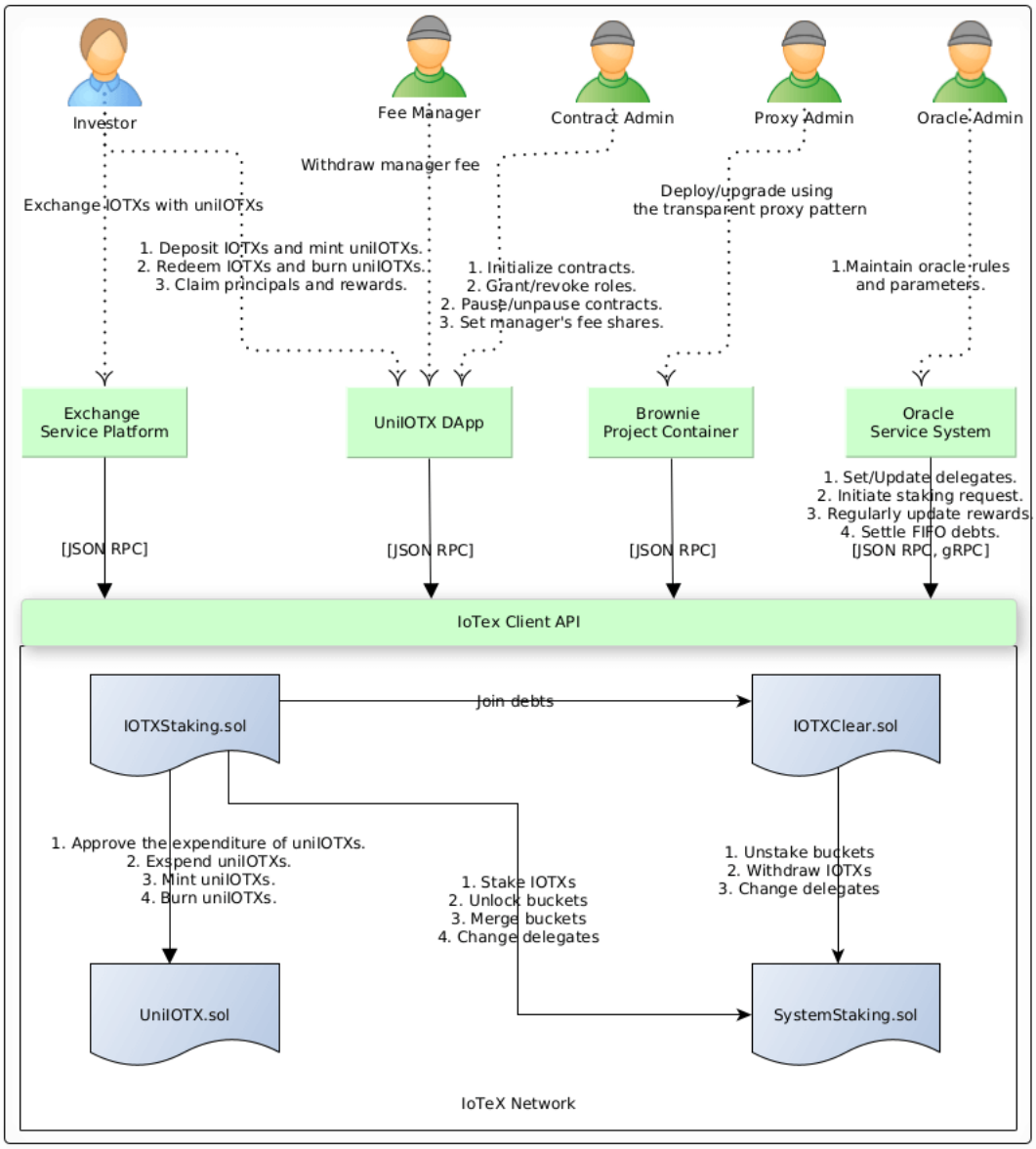

Figure 1: System Architecture of Bedrock Liquid Staking (IoTeX)

3 Business Lifecycle

Figure 2 illustrates the typical business lifecycle. The lifecycle starts with a user deposit and ends with a user claim, which corresponds to the staking and unstaking phases respectively.
From an investor's viewpoint, there are three primary business scenarios:

- Depositing IOTX: Deposit IOTX to mint uniIOTX and earn automatically compounded revenue.
- Redeeming IOTX: Redeem IOTX and burn uniIOTX, then claim the corresponding principal and rewards.
- Trading IOTX: Exchange uniIOTX for IOTX using an eco-friendly trading platform such as a DEX.

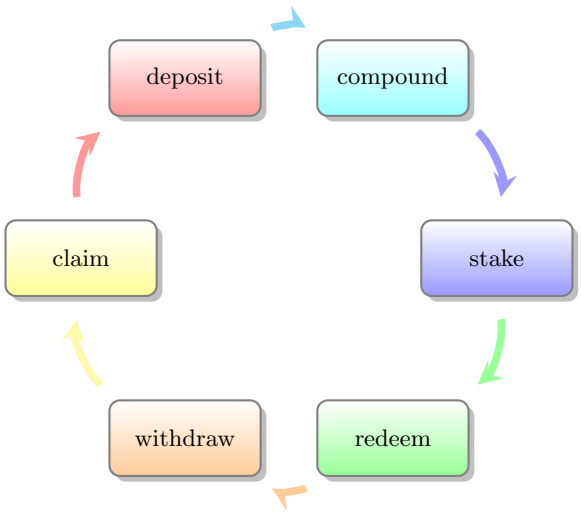

Figure 2: Business Lifecycle of Bedrock Liquid Staking (IoTeX)

4 Business Rules

We previously discussed the staking parameters in subsection 1.2, which must be considered. The following are the essential business rules tailored for our Bedrock Liquid Staking (IoTeX) project.

1. Staking Buckets: There is no limit to the amount of IOTX you can deposit. The available bucket types for staking, irrespective of the IOTX deposit amount, are as follows: {Amount: 10000/100000/1000000 IOTX, Duration: 91 Days}. During the staking process, an equivalent amount of uniIOTX will be generated based on the current exchange ratio.

2. Staking Order: The total pending IOTX will be staked in descending order, starting from the 1000000-bucket, down to the 100000-bucket, and finally to the 10000-bucket. A global map consisting of three queues is utilized to store all types of bucket/token IDs. These are represented by Queue-3, Queue-2, and Queue-1, separately.

3. Bucket Merging: When the bucket count of a lower queue reaches 10, its buckets are merged into a higher queue. Specifically, ten 10000-buckets in Queue-1 will be consolidated into a 100000-bucket in Queue-2, and subsequently, ten 100000-buckets in Queue-2 will be combined to form a 1000000-bucket in Queue-3. Thus the length of Queue-3 can dynamically increase, whereas the lengths of Queue-1 and Queue-2 always remain under 10.

4. Rewards Distribution: The Stake-Lock (Auto-Stake) feature is automatically activated for all staked buckets during the staking phase. When users redeem IOTX, the corresponding buckets transition to the unstaking phase for debt repayment. Users are rewarded in both phases, managed separately by the IOTXStaking and IOTXClear contracts. During the staking phase, rewards distributed to users from delegates are automatically reinvested into the total pending IOTX. However, rewards earned during the unstaking phase can be claimed by users.

5. Debt Management: Before initiating a redemption request, users are required to authorize the IOTXStaking contract to expend uniIOTX via the UniIOTX contract. Upon initiation of a redemption request, the user's uniIOTX will be immediately destroyed, and the corresponding debt recorded, which will subsequently be paid in a First-In-First-Out (FIFO) order. The principal of a debt can only be claimed after 94 days. However, any rewards distributed during this unstaking phase can be claimed at any time.

6. Redemption Limit: Users can redeem IOTX only in units of 1000000. Therefore, only the largest buckets containing 1000000 IOTX are eligible for debt payment withdrawal. However, users have the option to exchange uniIOTX for IOTX using an ecological trading platform like a DEX. This process also necessitates prior approval to expend uniIOTX via the UniIOTX contract.

7. Manager Fee: The manager's fee will be computed and accumulated solely from the rewards distributed during the staking phase. A fee manager will periodically withdraw this amount and reinvest it into the total pending IOTX, which is earmarked for future re-staking. When the manager's fee is withdrawn, an equivalent amount of uniIOTX will be created for the designated recipient, based on the current exchange rate.

8. Oracle Service: The Oracle system determines the most appropriate delegate, synchronizes rewards on a daily basis, and regularly initiates staking requests to re-stake pending user rewards. Furthermore, it manages the debt payment process during the unstaking phase, which lasts as long as 94 days.

9. NFT token: The SystemStaking contract will issue an NFT token for each bucket created during the staking process. Once a redemption request is triggered, the ownership of the NFT token will be transferred from the IOTXStaking contract to the IOTXClear contract. The SystemStaking contract will ultimately withdraw/burn the NFT for debt payment during the unstaking phase. Users remain unaware of the NFT tokens throughout the business lifecycle.

10. Exchange Ratio: All procedures concerning minting and burning uniIOTX must maintain the current exchange ratio. This approach aims to prevent past investment activities from affecting future ones unnecessarily, thereby deterring user arbitrage, and promoting fairness.

    Proof. To prove ρ invariant and irrelevant of IOTX to stake, for $CurrentReserve \in (0, +\infty)$:

    $$\rho = \frac{TotalSupply}{CurrentReserve} = \frac{TotalSupply' + \rho' \cdot IOTXsToStake}{CurrentReserve' + IOTXsToStake}$$

    as by definition:

    $$\rho' = \frac{TotalSupply'}{CurrentReserve'}$$

    we have:

    $$\rho = \frac{CurrentReserve' \cdot \rho' + \rho' \cdot IOTXsToStake}{CurrentReserve' + IOTXsToStake} = \frac{\rho' \cdot (CurrentReserve' + IOTXsToStake)}{CurrentReserve' + IOTXsToStake}$$

    finally:

    $$\rho = \rho'$$

    □

5 Contract Interaction

Figure 3 demonstrates how message calls are created between contracts. Here is a corresponding textual explanation:

- When an investor initiates **IOTXStaking.deposit()**, it activates **UniIOTX.mint()**, **SystemStaking.stake()**, and **SystemStaking.merge()**, provided the specified conditions are fulfilled.
- When an investor invokes **IOTXStaking.redeem()**, it sequentially triggers **UniIOTX.burnFrom()**, **SystemStaking.unlock()**, **SystemStaking.safeTransferFrom()**, and **IOTXClear.joinDebt()**.
- When the fee manager initiates triggers **IOTXStaking.withdrawManagerFee()**, it consequently triggers **UniIOTX.mint()**.
- When the Oracle invokes **IOTXStaking.stake()**, it subsequently triggers **SystemStaking.stake()**, and **SystemStaking.merge()**, provided that the specified conditions are met.

- When the Oracle initiates either **IOTXStaking.updateDelegates()** or **IOTXClear.updateDelegates()**, it subsequently triggers **SystemStaking.changeDelegates()**.
- When the Oracle calls either **IOTXClear.unstake()** or **IOTXClear.payDebts()**, it subsequently triggers **SystemStaking.unstake()** or **SystemStaking.withdraw()**, respectively.

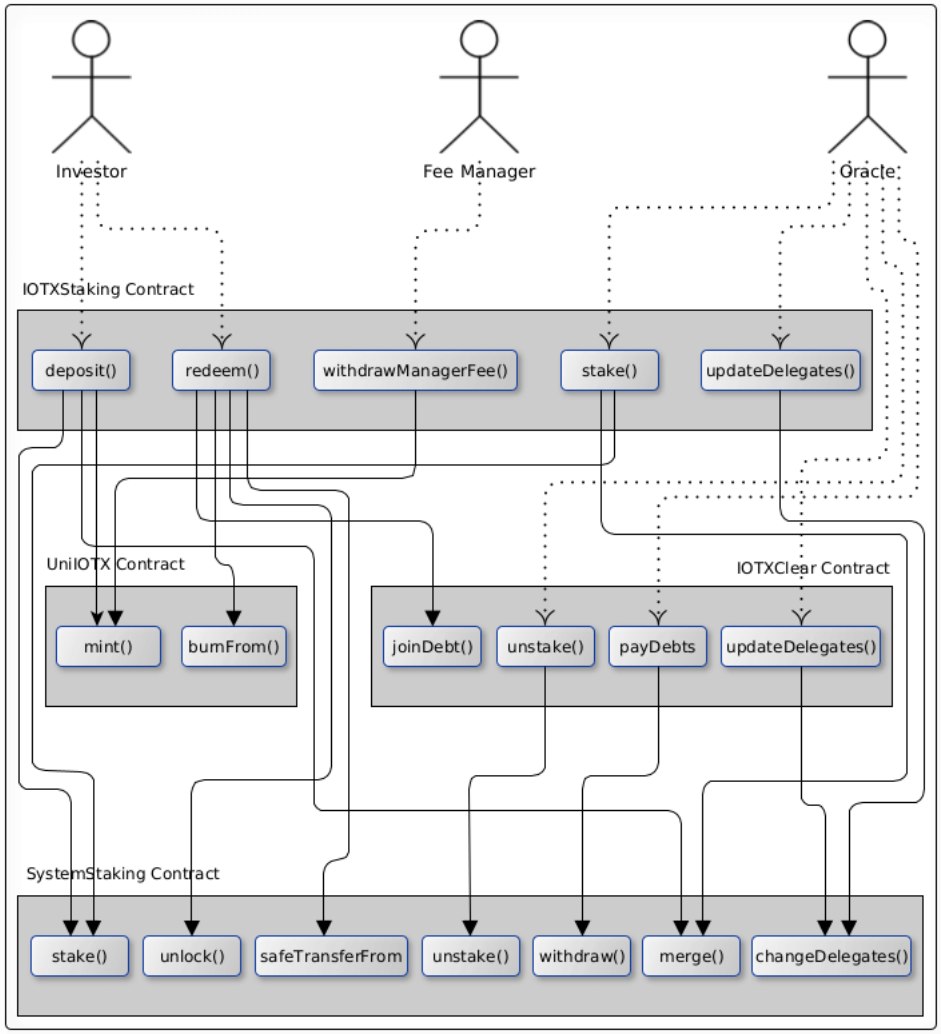

Figure 3: Contract Interaction of Bedrock Liquid Staking (IoTeX)

6 Conclusion

Bedrock Liquid Staking (IoTeX) provides all users the opportunity to earn rewards on any amount of IOTX, accruing benefits over time. This allows retail users to participate in maintaining the IoTeX network, functioning similarly to an inclusive financial system in the real world. Additionally, stakers have the ability to hedge their uniIOTX tokens to prevent financial loss.

The overall design of this liquid staking protocol prioritizes fund security when using the funds to earn rewards. The source code and architecture have been made publicly available. Furthermore, the source code has undergone an audit by PeckShield.

A Terminology

IOTX $1 \cdot IOTX \equiv 10^{18}$

TotalSupply The total amount of uniIOTX being supplied.

TotalPending The total amount of IOTX awaiting staking.

TotalStaked The total amount of IOTX being staked.

TotalDebts The total amount of IOTX awaiting debt repayment.

CurrentReserve The total amount of IOTX under management, given as:

$$CurrentReserve = TotalPending + TotalStaked$$

ExchangeRatio Defined as symbol ρ of uniIOTX to IOTX, given as:

$$\rho = \begin{cases} \frac{TotalSupply}{CurrentReserve} & CurrentReserve \in (0, +\infty) \\ 1 & CurrentReserve = 0 \end{cases} \tag{1}$$

normally: $\rho \leq 1.0$

ManagerFeeShare The share of the manager fee, represented as 1 in 1000, managerFeeShares $\in [0, 1000]$

GlobalDelegate The address for the global delegate for the upcoming staking requests.

StakeAmounts The geometric series of legal IOTX amounts for all staking requests, given as:

$$StakeAmounts \equiv \{10000, 100000, 1000000\}$$

RedeemAmountBase The unit amount of IOTX for redemption request, given as:

$$RedeemAmountBase \equiv 1000000$$

StakeDuration The staking duration of 91 days for all staking requests, given as:

$$StakeDuration \equiv 1572480$$

AccountedManagerReward The accumulated IOTX rewards distributed for manager fee during the staking phase.

AccountedUserReward The accumulated IOTX rewards distributed for users during the staking phase.

AccountedStakingReward The synchronized IOTX rewards produced by delegates during the staking phase, give as:

$$AccountedStakingReward = AccountedUserReward + AccountedManagerReward$$

CompoundedAmount The total amount of IOTX automatically compounded for future re-staking during the staking phase, given as:

$$CompoundedAmount = AccountedUserReward$$

DebtAmountBase The unit amount of IOTX for debt record, given as:

$$DebtAmountBase = RedeemAmountBase$$

ThisBalance The actual IOTX amount that is being controlled by the contract account.

AccountedBalance The synchronized IOTX amount assigned from **ThisBalance**, given as:

$$AccountedBalance = ThisBalance$$

IncrReward The total amount of increased IOTX reward during the unstaking phase, given as:

$$IncrReward = ThisBalance - AccountedBalance$$

RewardRate The monotonically increasing shared reward metric during the unstaking phase, given as:

$$RewardRate \texttt{+=} \begin{cases} \frac{IncrReward}{TotalDebts} & TotalDebts \in (0, +\infty) \\ 0 & TotalDebts = 0 \end{cases} \tag{2}$$

B References

- The IoTeX Network
- Roll-DPoS Consensus Mechanism
- Staking on IoTeX
- Delegates on IoTeX
- IIP-13: Represent Staking Buckets As Non-fungible Tokens
- SystemStaking Contract for IIP-13
- Liquid Staking Explained by Lido
- Design of Bedrock Liquid Staking Contracts on IoTeX
- Smart Contract Audit Report for Bedrock Liquid Staking (IoTeX)
- Bedrock UniIOTX Web 3.0 DApp
- SystemStaking Smart Contract
- UniIOTX Smart Contract
- IOTXStaking Smart Contract
- IOTXClear Smart Contract