TYKORA

WHITEPAPER

Save together. Win together. On Rootstock

V1.0.0

Table of Contents

1. Abstract 2
2. Motivation & Problem Statement 2
3. Protocol Overview 2
   3.1 Network and Assets 2
   3.2 Core Components 2
   3.3 Non-Transferable Shares 3
4. User Experience and Core Flows 3
   4.1 Connect 3
   4.2 Approve & Deposit 3
   4.3 Withdraw 3
5. Vault Cycle Lifecycle (Defined) 3
6. Economics, Fees, and Prize Distribution 4
   6.1 Yield Computation 4
   6.2 Fees and Prize 4
   6.3 Multi-Winner Prize Split 4
7. Randomness and Winner Selection (BTC-Anchored, Deterministic) 4
   7.1 BTC Target Height Fixing 4
   7.2 BTC Header Retrieval and Seed Derivation 4
   7.3 Tickets and Weighted Selection 5
   7.4 Manual Award Mode (Owner) 5
8. Draw State Machine and Operational Roles 5
9. Security Model (High-Level) 6
   9.1 Non-Transferable Shares 6
   9.2 Reentrancy Protection 6
   9.3 Locking Semantics 6
   9.4 Emergency Cancellation 6
   9.5 Administrative Assumptions 6
10. Transparency and Verification 6
11. Risks and Disclaimers 6
12. Terms of use 7

Appendix A — Glossary 7

https://jxlabs.xyz – ping@jxlabs.xyz

1. Abstract

TYKORA is a prize-linked savings protocol deployed on **Rootstock**, designed to make saving in **DOC** (MoneyOnChain's USD-pegged asset) and **USDRIF** (a Rootstock-native stablecoin) simple, transparent, and verifiable. Users deposit DOC/USDRIF into a vault, receive non-transferable vault shares, and earn yield generated through an external strategy (currently designed for **Tropykus**). Over each draw period, accrued yield is allocated to a prize pool (after protocol fees). At draw finalization, the protocol selects **three distinct winners** (under the protocol's minimum-participation policy) using a deterministic process anchored to Bitcoin-derived randomness via the Rootstock bridge.

TYKORA's design goals are:

- **Capital preservation first:** depositors maintain a share-based claim on vault underlying, with withdrawals available when the vault is unlocked.
- **Verifiable prize selection:** winners are determined through on-chain logic derived from a BTC-referenced seed.
- **Simplicity:** minimal user actions (connect → approve/deposit → optionally withdraw → claim if owed).

TYKORA is developed by **JXLabs**.

2. Motivation & Problem Statement

Traditional saving products face friction, low engagement, or reliance on centralized intermediaries. In DeFi, yield opportunities exist but can be intimidating or operationally complex for users.

Prize-linked savings aligns incentives by converting a portion of yield into a prize—creating excitement and engagement—while keeping the saving experience straightforward. TYKORA aims to provide:

- A single vault experience for depositing DOC and USDRIF
- Transparent yield and prize accounting
- A draw lifecycle that is deterministic and auditable
- A BTC-anchored mechanism for randomness to reduce reliance on subjective or off-chain selection

3. Protocol Overview

3.1 Network and Assets

- **Network:** Rootstock (EVM-compatible, Bitcoin-secured sidechain)
- **Underlying asset: DOC** (MoneyOnChain), USDRID (Rootstock)
- **Yield source:** an external strategy contract implementing IStrategy (designed for allocation to **Tropykus**)

3.2 Core Components

1. PrizeVault (Vault Contract)
   - ERC-20 share token with a critical modification: shares are non-transferable (mint/burn only).

- Manages deposits, withdrawals, draw lifecycle, winner selection, and payout accounting.

2. Strategy (IStrategy implementation)
   - Receives underlying DOC/USDRIF from the vault and deploys it to generate yield.
   - Exposes deposit, withdrawUnderlying, totalUnderlying, and accrue used by the vault.
3. Rootstock Bridge (BTC Header Source)
   - Used to fetch Bitcoin block headers by height to derive an unpredictable BTC-anchored seed.
   - The seed is then used to deterministically select winners on-chain.

3.3 Non-Transferable Shares

Deposits mint vault shares **1:1** with underlying deposited. Withdrawals burn shares **1:1** with the underlying withdrawn. Shares cannot be transferred between users; they solely represent a depositor's position.

This improves accounting clarity and supports winner selection weighted by each depositor's share balance.

4. User Experience and Core Flows

4.1 Connect

Users connect an EVM wallet configured for Rootstock. The UI displays:

- Current draw information (status, time remaining, prize/yield)
- User's DOC/USDRIF balance
- User's share balance ("tickets" weight)
- Allowance and claimable prize (if any)

4.2 Approve & Deposit

If allowance is insufficient, the UI prompts the user to approve DOC/USDRIF spending, then deposit. On deposit:

- DOC/USDRIF is transferred to the vault and forwarded to the strategy.
- Shares are minted to the user (1:1 with DOC/USDRIF deposited).

4.3 Withdraw

Withdrawals burn shares and request underlying from the strategy to the user.
Note: Withdrawals are disabled while the vault is **locked**.

5. Vault Cycle Lifecycle (Defined)

The vault operates under a **fixed, deterministic lifecycle** enforced by the deployed contracts:

- Cycle Opens
  - Deposits are accepted.
  - Yield begins accruing.
- Cycle Progresses
  - Prize amount grows as yield accumulates (per yield split).
- Vault Locks

    - Certain actions (commonly withdrawals) become restricted.
- Cycle Finalizes
    - The draw ends, and the winners are determined **via on-chain awarding logic**.
- Prize Becomes Claimable
    - The winning address receive automatically the prize.
- Next Cycle Begins
    - A new draw ID/status appears and the process repeats.

6. Economics, Fees, and Prize Distribution

6.1 Yield Computation

At draw close, the vault calls strategy.accrue() to update strategy accounting and computes:

- totalAssets = idleUnderlyingInVault + strategy.totalUnderlying()
- yield = max(totalAssets - totalPrincipal, 0)

Where:

- totalPrincipal is the cumulative deposited principal tracked by the vault.

6.2 Fees and Prize

From computed yield:

- treasuryFee = yield * treasuryBps / 10,000
- keeperTip = yield * keeperBps / 10,000
- prize = yield - treasuryFee - keeperTip

Fees are parameterized in basis points (bps). The sum of treasury + keeper bps is constrained to ≤ 10,000 bps.

6.3 Multi-Winner Prize Split

TYKORA awards **exactly three distinct winners** on every draw. The total prize is distributed using a fixed split:

- **Winner #1:** 50%
- **Winner #2:** 30%
- **Winner #3:** 20%

Any remainder resulting from integer division is assigned to **Winner #1** to ensure exact conservation:

- winnerPrizes[0] + winnerPrizes[1] + winnerPrizes[2] == prize

Operational policy**:** TYKORA runs prize draws only when there are at least **three distinct participating wallets**, ensuring the 50/30/20 distribution is applied.

7. Randomness and Winner Selection (BTC-Anchored, Deterministic)

7.1 BTC Target Height Fixing

When a draw is closed, the vault calculates:

- btcTargetHeight = bestBtcHeight + btcConfirmations

and stores it in the draw state. This creates a predictable *future* target, where the BTC header is not yet known at close time.

7.2 BTC Header Retrieval and Seed Derivation

To award a draw using BTC-derived randomness, the vault:

1. Fetches the BTC header at btcTargetHeight from the Rootstock bridge.
2. Computes:
    - btcHash = sha256(sha256(header)) (double-SHA256)
    - seed = keccak256(abi.encodePacked(btcHash, vaultAddress, drawId))

The seed is stored on-chain and emitted so that anyone can verify it.

7.3 Tickets and Weighted Selection

Users' "tickets" are proportional to their vault shares. TYKORA maintains a **Fenwick tree (sum tree)** keyed by user indexes, with weights equal to user share balances.
At draw close:

- tickets = totalTickets() is snapshotted into the draw.

Winner selection chooses **three distinct winners** (subject to the minimum-participation policy):

- For each winner slot i = 0..2:
    - s_i = keccak256(abi.encodePacked(seed, i))
    - r = uint256(s_i) % remainingTickets
    - Resolve r into a winner index via Fenwick prefix sums
    - Temporarily remove the winner's weight to avoid selecting the same user again, then restore at the end

This yields:

- **Probability proportional to shares**
- **No duplicates within a draw**
- **Fully deterministic given the seed and ticket distribution**

7.4 Manual Award Mode (Owner)

For local testing or deployments where BTC randomness is not available, the owner can award with a manually provided seed. This mode is explicitly restricted and should be treated as an administrative/testing path, not the default production flow.

8. Draw State Machine and Operational Roles

TYKORA defines draw states:

- **OPEN → CLOSED → AWARDED → CLAIMED**

Key operations:

- closeDraw()
  Computes yield, fees, prize, snapshots tickets, sets BTC target height, and If there is **no yield** or **no participants (tickets = 0)**, the draw is **auto-finalized** and the next draw starts immediately (no lock, no award/claim phase).
- awardDrawFromBtc(drawId)
  Requires the BTC header to be available; computes seed and selects winners.
- claimDraw(drawId)
  Pays:
    - keeper tip to caller
    - treasury fee to treasury
    - prizes to winners (or records owed on transfer failure)
      Then unlocks the vault and starts the next draw.

Keeper Tip

The keeper tip incentivizes a third party to execute claimDraw promptly when a draw is awarded.

9. Security Model (High-Level)

9.1 Non-Transferable Shares

Shares are non-transferable, reducing secondary-market complexity and keeping the vault's accounting and ticketing model direct.

9.2 Reentrancy Protection

User flows and draw flows are protected by reentrancy guards.

9.3 Locking Semantics

When a draw has distributable yield and participants, TYKORA locks the vault at close. This prevents withdrawals during the critical award/claim window and ensures payout funds are reserved.

9.4 Emergency Cancellation

If a draw is closed and locked but cannot be completed, an emergency cancellation mechanism can be executed after a configured delay. The reserved funds are redeposited into the strategy so yield carries into the next draw, then the vault unlocks and a new draw opens.

9.5 Administrative Assumptions

- The strategy is set **once** by the owner.
- Treasury address can be updated by the owner.
- Manual award is owner-restricted.

These assumptions should be clearly disclosed to users and can be reduced over time via governance/multisig if desired.

10. Transparency and Verification

TYKORA is designed so that external observers can verify:

- Draw status transitions (OPEN/CLOSED/AWARDED/CLAIMED)
- Yield, fees, and prize amounts computed at close
- BTC target height, BTC hash, and seed used for awarding
- Winner addresses and prize splits
- Claim execution and transfer outcomes (including owed fallback)

Verification does not require trust in the UI; all critical parameters are anchored on-chain and emitted in events.

11. Risks and Disclaimers

TYKORA involves multiple risk layers common to DeFi systems:

- Smart contract risk: vulnerabilities, exploits, or unforeseen edge cases.
- Strategy risk (Tropykus): lending-market and integration risks, including liquidity constraints, insolvency events, pauses, or exploit risk.

- Keeper / automation risk: draw close/award/settlement may be delayed or fail due to operational conditions, incentives, or network issues.
- Bridge dependency: BTC header availability via the Rootstock bridge may be delayed, reorganized, or disrupted.
- Stablecoin risk (DOC/USDRIF): peg risk, market risk, and systemic risk in the underlying asset.
- Administrative risk: owner privileges (strategy setup, treasury updates, emergency parameters, manual award/testing paths).

Users should only deposit funds they can afford to risk. This document is informational and does not constitute financial advice. TYKORA is provided “as is”, without warranties of any kind, and use of the protocol is at the user’s sole risk.

12. Terms of use

By interacting with **TYKORA**, users acknowledge that they have read, understood, and accepted the terms, risks, and legal disclaimers described in this Whitepaper. Users assume full responsibility for the management of their digital assets, private keys, and wallet addresses, as well as any interaction with third-party protocols and infrastructure used by **TYKORA**.
To the maximum extent permitted by law, **JXLabs, contributors, and affiliates shall not be liable** for any direct or indirect losses or damages arising from the use of, or inability to use, **TYKORA**, including loss of principal, yield, prizes.

Appendix A — Glossary

- **DOC:** USD-pegged asset from MoneyOnChain on Rootstock.
- **USDRIF:** Stablecoin asset from Rootstock.
- **Strategy:** contract that deploys underlying to earn yield (e.g., Tropykus integration).
- **Shares:** non-transferable vault token representing depositor position (mint/burn only).
- **Tickets:** effective weight used for winner selection; proportional to shares.
- **Draw/Cycle:** time-bounded period with yield accrual and prize distribution.
- **Keeper:** actor incentivized to execute claim actions via keeper tip.
- **BTC seed:** deterministic randomness derived from BTC block header hash via Rootstock bridge.