# ProsperityChests: A Pirate-Themed Automated DCA Platform on Base

January 11, 2025

Clark Balan
*clark@prosperitypirate.com*

## ABSTRACT

***ProsperityChests*** is a fully on-chain DCA platform with a pirate-inspired gamification layer. Built on **Base**(https://base.org), it enables users to create, follow, and automate investment portfolios represented as NFT-based “Treasure Chests.” By leveraging low fees on Base and decentralized infrastructure, ProsperityChests fosters an accessible, trust-minimized environment for social trading, performance competitions, and dynamic portfolio management. This whitepaper outlines the platform’s architecture, tokenomics, security measures, and future roadmap.

## INTRODUCTION

Decentralized finance (DeFi) has fueled the emergence of on-chain, user-controlled investment strategies. Yet crafting reliable automated strategies—like dollar-cost averaging (DCA)—often involves fragmented liquidity, high gas costs, and complexity. Existing social trading solutions typically rely on centralized platforms or custom backends.

***ProsperityChests*** addresses these challenges with a pirate-themed, **fully on-chain** DeFi ecosystem. Users can create treasure chests (NFTs) that track allocations, performance, and fees, while others can follow top strategies and share in success. All logic and data is stored on Base, an Ethereum Layer 2 that offers low-cost, high-speed transactions.

**Vision and Goals** include:
- *On-Chain Social Trading*: Transparent ROI leaderboards.
- *Automated DCA*: Chainlink or community triggers for recurring trades.
- *Gamified Incentives*: Pirate ranks, achievements, protocol fees.
- *Decentralized & Trustless*: No single-party control—code is law.

## PROBLEM STATEMENT

Traditional DCA processes require manual bridging, repeated trades, and high gas for each action. Many novices are deterred by the complexity, while advanced users lack robust social features for discovering or monetizing their strategies. DeFi also suffers from fragmented liquidity, making frequent trades expensive on certain chains.

**Key Pain Points**:

- *High Complexity*: Scheduling, monitoring, bridging, and verifying trades.
- *Missing Social Transparency*: Few open, on-chain copy-trading solutions.
- *Gas Costs & Fragmented Liquidity*: Frequent rebalances are cost-prohibitive for smaller portfolios.

## OVERVIEW & ARCHITECTURE

## KEY COMPONENTS

***- Treasure Chest (NFT):***

Each user strategy becomes an ERC-721. Stores allocations, intervals, historical ROI, follower count, and fee distribution.

***- TreasureChestFactory:***
Creates new chests and mints NFTs to owners ("captains"). Manages the essential logic for chest initialization.

***- CrewManagement:***
Links chest owners to "crew" (followers). Distributes fees (80% to creator, 20% to protocol). Tracks follower states and copy-trade mechanics.

***- PerformanceTracker:***
Calculates ROI, logs trades, and updates scoreboard data for on-chain leaderboards.

***- ChainlinkAutomation:***
Triggers recurring trades at set intervals (premium) or uses manual / community execution to cut fees.

***- TreasureChestRegistry:***
Keeps a registry of all active chests for easy discovery and ranking.

## WORKFLOW

### 1. Chest Creation:
A user calls `createChest(...)`, specifying tokens, intervals, investment amounts, and automation type.

### 2. Copy Trading:
Others see the chest's ROI and choose to follow, replicating trades. Chest owners earn fees when followers profit or trade.

### 3. DCA Execution:
Either Chainlink Keepers or manual triggers perform periodic buys, updating chest value and broadcasting events.

### 4. Leaderboards & Gamification:
Ranking by ROI fosters competition, with "pirate ranks" and achievements. Higher ROI → more followers → more fees.

## TECHNICAL ARCHITECTURE

## BASE AS THE DEPLOYMENT LAYER

Base is a secure, low-cost L2 environment on Ethereum, minimizing gas overhead. All core contracts (Factory, CrewManagement, etc.) are deployed on Base. Users interact through wallets like MetaMask or Coinbase Wallet configured for Base.

## UNISWAP INTEGRATION

Periodic swaps rely on **Uniswap V4** on Base. A `SwapManager` handles slippage protections, stable liquidity pairs, and calls from Chainlink or user triggers. This allows seamless token rebalancing at scale.

## DATA & PERFORMANCE

All chest data is stored on-chain. PerformanceTracker calculates ROI from oracles (Chainlink Feeds or a Uniswap TWAP). Leaderboards can be rendered by any front end reading these on-chain metrics.

## AUTOMATION

- ***Chainlink:*** Fully automated, but requires LINK and keeper fees.
- ***Manual:*** Owner calls "executeTrade()" on their own schedule (no extra fee).
- **Community:** Anyone can trigger trades, optionally earning a tip.

## TOKENOMICS & FEE DISTRIBUTION

Followers pay a small performance or copy-trade fee. **80%** goes to the chest creator, **20%** to the protocol treasury. This structure rewards creators, supports the platform, and incentivizes quality strategies.

**Protocol Fees** fund:
1. Maintenance & audits.
2. A treasury that could support marketing, grants, or future expansions.

High ROI chests gain more followers, generating a positive cycle of popularity and fee revenue.

## SECURITY & AUDITS

## SMART CONTRACT SECURITY

- **OpenZeppelin:** Uses standard libraries (ERC-721, ReentrancyGuard, etc.).
- **Chainlink Oracles:** Minimizes price manipulation with reliable feeds.
- **Emergency Pause:** Allows disabling chest operations if a critical bug is found.

## AUDITS & TESTING

- Developed with Foundry or Hardhat, targeting near 100% coverage.
- Tested end-to-end on Base Sepolia.
- Third-party audits are planned for major releases.

## ROADMAP

- **Phase 1:** Basic chest creation, simple DCA, event-based performance updates.
- **Phase 2:** Uniswap V4 integration, advanced analytics, social features (follow, fees, ranks).
- **Phase 3:** Further gamification, cross-chain bridging (if needed), yield strategies, potential governance.

## FRONTEND & UX

A pirate-themed interface highlights treasure chests and crews. The user-friendly flow:

- **Dashboard:** Summaries of owned/followed chests and performance.
- **Explore:** Filter or search chests by token, ROI, or intervals.
- **Profile:** Track achievements, follower counts, and personalization.

No centralized backend is needed; data is fetched from on-chain or subgraph indexers.

## POTENTIAL RISKS & MITIGATIONS

**1. Smart Contract Bugs:**
Thorough tests, audits, bug bounties, and emergency pause.

**2. Price Manipulation:**
Use Chainlink or Uniswap TWAP oracles with robust slippage controls.

**3. Liquidity Constraints:**
Integrate major pools on Base or fallback to stable DEX pairs.

**4. Automation Reliability:**
Provide community triggers if Chainlink or manual calls fail.

## CONCLUSION

ProsperityChests aims to merge dollar-cost averaging with on-chain social trading in a fun, pirate-inspired ecosystem. By harnessing Base's low-fee environment and robust infrastructure, users can seamlessly create, share, and automate portfolios. The ultimate goal is a community-driven protocol enabling user-generated strategies and fair creator compensation, all in a trustless DeFi environment.

## REFERENCES

***Base Documentation***
*https://docs.base.org/*

***Chainlink Keepers & Oracles***
*https://chain.link/*

**Uniswap V4 Developer Docs**
*https://docs.uniswap.org/*

**OpenZeppelin Contracts**
*https://docs.openzeppelin.com/contracts/*

## DISCLAIMER

This whitepaper is for informational purposes only and does not constitute financial or legal advice. The ProsperityChests team does not guarantee specific financial outcomes; all trades and strategies carry risk. The platform may be subject to contract vulnerabilities, price manipulations, or regulatory changes. Proceed at your own discretion.

## CONTACT & COMMUNITY

- **Official Website:** *prosperitypirate.com*