Lilipad

A Trustless Fair-Launch Protocol for Transparent Token Distribution on Aptos

Whitepaper v1

November 29, 2025

Abstract

Abstract: Lilipad is a modular, self-serve infrastructure platform built on the Aptos network. It addresses the critical fragility of early-stage Web3 projects by providing tools for transparent token creation, fair launch sales, deterministic vesting, and liquidity security. By leveraging the high throughput and safety guarantees of the Aptos blockchain, Lilipad ensures that project launches are verifiable, trustless, and aligned with the ethos of fair distribution.

Contents

1 Introduction

The early stages of a Web3 project are the most fragile and the most defining. Yet, even within a high-performance ecosystem like Aptos, launching a new token responsibly remains a surprisingly difficult task. Most teams lack the technical depth to build secure vesting systems in Move, manage liquidity safely, or structure fair and transparent token sales. At the same time, communities struggle to trust new assets when insiders hold privileged access or when launch mechanics remain opaque. A single misstep, such as an unlocked treasury or an exploitable contract, can permanently erode trust.

Lilipad is built to address this gap on Aptos. It is a modular and self-service platform designed around one central idea: projects deserve a fair, transparent, and predictable way to launch, and communities deserve the confidence that comes with a fully trustless launch process. Lilipad offers a seamless set of tools that empower any builder to create a token, launch it publicly, distribute it responsibly, and secure its liquidity, all while keeping the project fully in control of the narrative.

2 Philosophy: A Home for Fair Launches

Fairness is the cornerstone of Lilipad. The platform is not just a collection of tools but a philosophy of how crypto projects should begin. The fair launch movement emerged to challenge an environment dominated by opaque token allocations and insider advantages. Lilipad builds on this philosophy by embedding fairness directly into the architecture of the application.

A fair launch on Lilipad means that the rules of participation are the same for everyone. It ensures that token distribution mechanics are verifiable and predictable, and that the early liquidity of a token cannot be quietly withdrawn or manipulated. Instead of giving power to privileged actors, Lilipad guarantees that every token path, from minting to vesting to liquidity provision, happens in a transparent and trustless manner secured by the Aptos blockchain. At the same time, the platform respects the autonomy of builders. Lilipad does not force a specific launch sequence. Projects are independent entities free to use only the Move modules they need in the order they choose.

3 The Role of Project Identity

Project identity on Lilipad is intentionally kept off-chain. Instead of storing metadata directly on the blockchain, Lilipad allows creators to establish project identity through signed messages. This proves wallet ownership without incurring unnecessary gas costs or bloating the network state. A project can define its name, brand, mission, and documentation off-chain, while Lilipad verifies authenticity through cryptographic signatures.

This approach is flexible, fast, and builder-friendly. Teams can update their project profiles organically, evolve their documentation, or migrate their metadata to decentralized storage without touching the blockchain. Developers can optionally anchor their project identity to the chain through a small on-chain hash commitment, but this is not required. Lilipad views identity as a social element of the launch process rather than a technical one, ensuring the tools remain lightweight and accessible to all Aptos users.

4 The Launch Lifecycle

A project journey on Lilipad follows a natural lifecycle but not a rigid sequence. Every component can stand alone yet connects elegantly with the next. Projects may create only a vesting schedule or only a liquidity lock. They may mint a token without running a sale or register a project identity without ever launching a token. The platform supports experimentation and partial usage.

For projects that wish to run a full fair launch, the lifecycle typically unfolds in four stages: **token creation**, **token sale**, **vesting distribution**, and **liquidity protection**. Each stage is powered by independent modules in the Lilipad Move contract suite. Tokens are kept in escrow during sales; APT raised through sales is locked until the sale ends; vesting mechanisms release tokens gradually according to strict on-chain rules; and liquidity locks ensure that neither developers nor malicious actors can withdraw liquidity prematurely. This lifecycle creates a transparent journey that community members can easily follow on the blockchain explorer. Fairness is not promised; it is provable.

5 Token Launch and Minting

At the earliest stage of development, Lilipad provides a simple interface for minting a new asset. This feature is optional, as some projects may bring their own token or use an existing asset. However, for teams that wish to start from scratch, Lilipad ensures that token minting is fully transparent with no backdoors or hidden supply manipulations.

The token factory allows creators to define essential parameters like total supply, decimals, and distribution destination compatible with the Aptos Token Standard. Tokens can be minted into the project wallet, audited by the community, and used immediately in a sale or vesting schedule. This simplicity helps teams move from idea to asset deployment in minutes while providing the community with clear and trustworthy data.

6 The Launchpad: Transparent Sales

The heart of Lilipad is its launchpad. Unlike traditional token sale platforms, the Lilipad launchpad is designed to enforce fairness through its architecture rather than through policy. When a project creates a sale, it reserves a specific amount of tokens in a secure on-chain escrow. These tokens cannot be touched, moved, or reclaimed until the Move logic dictates.

Participants purchase tokens using APT, and a precise calculation determines the number of tokens they receive. Instead of releasing tokens instantly, which historically creates problems like early dumps or price manipulation, Lilipad converts each purchase into a vesting stream. Buyers receive their tokens gradually over time, enforced entirely by the Move runtime, ensuring no one is able to unfairly manipulate early supply.

The sale continues until its defined end time. When the soft cap is reached, the system emits an event notifying the project and the community. At this stage, the project can decide to list or migrate tokens to Hyperion DEX and optionally lock the resulting liquidity tokens on Lilipad to prevent early rug pulls. When the sale ends, the project owner can withdraw the raised APT, again fully governed by transparent on-chain rules.

7 Vesting and Long Duration Distribution

Vesting is one of the most essential components of a fair launch, and Lilipad treats it as such. The platform includes a manual vesting engine that any project can use independently of sales. Whether distributing tokens to early investors, vesting tokens for contributors, or managing internal team allocations, the vesting module ensures accountability.

Vesting streams are simple, linear, and purely on-chain. Creators specify a beneficiary, a token, a total amount, and a start and end date. Lilipad holds the tokens in a dedicated escrow bucket and releases them gradually based on the Aptos blockchain timestamp. Beneficiaries can claim their unlocked portions at any time, and no one, not even the stream creator, can alter the vesting rules once the stream is created. This structure reinforces responsible tokenomics and avoids the pitfalls of manual multisig managed vesting.

8 Liquidity Locking and Hyperion Integration

Liquidity is the lifeblood of any token in the DeFi environment. It is also a frequent vector for abuse where developers pull liquidity after creating a pool. To counter this, Lilipad provides a locking mechanism that projects can use voluntarily but effectively.

After creating a liquidity pool on Hyperion, projects receive an LP position reference. They can then lock this reference on Lilipad by specifying an unlock date. The locking module prevents the LP position from being accessed until the specified time has passed. This public commitment establishes confidence that liquidity is stable and not at risk of withdrawal. The locking system also supports locking fungible tokens, enabling teams to secure treasury allocations or long duration incentive pools without relying on trust in a central authority.

9 Escrow Architecture and Security

Central to Lilipad is its escrow architecture which leverages the Aptos Move resource model. Each module regarding sales, vesting, and locking maintains its own independent escrow bucket. This ensures that tokens are always accounted for according to their intended purpose and cannot be mistakenly or maliciously intermingled.

During a sale, the project tokens are deposited into sale escrow, and buyer deposits of APT are held separately. When buyers purchase tokens, the system transfers the purchased amount from sale escrow into vesting escrow, ensuring that the future claim of the buyer is fully backed. For vesting, tokens remain locked in vesting escrow until beneficiaries claim them. For locks, tokens or LP positions remain held until the unlock date. This architecture prevents double spending, enforces strict ownership rules, and guarantees the integrity of the process.

10 Conclusion

Lilipad provides a new foundation for project launches on Aptos, a foundation grounded in transparency, fairness, and builder autonomy. While many platforms attempt to automate or centralize launches, Lilipad chooses a different path where projects remain in control but benefit from trustless enforcement. By providing modular tools that work independently, including token sales, vesting, locks, and optional token creation, Lilipad supports every type of early stage project on the network.

More importantly, Lilipad cultivates an environment where fairness is not an afterthought but a default. In a space where communities often feel uncertain or suspicious of new tokens, Lilipad offers a steady surface to stand on, a stable pad from which projects can launch and grow within the Aptos ecosystem

11 Developer Appendix: Architecture and Implementation

This appendix provides a deeper technical examination of the Lilipad architecture for developers who wish to understand how the system is constructed on Aptos, how its individual components interact, and how fairness guarantees are enforced at the smart contract layer. While the core whitepaper focuses on the philosophical and structural design, this section details the Move based safety guarantees, module interactions, and workflows for extending the protocol.

11.1 Move Native Safety and Resource Model

The foundation of Lilipad rests on the Move smart contract language, utilizing a resource oriented execution model naturally suited to custody centric applications.

- **Linear Logic:** At the lowest level, every token, escrow bucket, vesting stream, and lock entry is represented as a Move resource. These resources cannot be copied, implicitly destroyed, or reassigned. This fundamental guarantee prevents entire classes of vulnerabilities, such as double spending or accidental token burning.
- **Module Namespace Control:** Resources are stored within a carefully controlled module namespace, ensuring that only the defining module can modify the internal state of a sale or vesting object.

11.2 Scalable Data Structures

Lilipad is designed to scale horizontally using the Aptos Table standard, avoiding the congestion often associated with linear vector storage.

- **Table Implementation:** Each module operates within the framework of separate resource groups and explicitly typed Table structures. This allows the protocol to support an unlimited number of simultaneous projects without degrading gas performance.
- **Identifier Management:** Sale records, stream records, and lock records are indexed by unique identifiers (SaleId, StreamId) generated from global counter resources. These counters are protected by private access modifiers, ensuring no external actor can forge identifier values.

11.3 The Triple Escrow Architecture

To ensure absolute fairness, the escrow system is implemented as three distinct, non overlapping resource families: **Sale Escrow**, **Vesting Escrow**, and **Lock Escrow**.

- **State Isolation:** The contract never mixes tokens between these domains. Sale participants are isolated from vesting deposits, and locked liquidity is never commingled with sale supply.
- **Atomic Transitions:** When a buyer participates in a sale, tokens are transferred from Sale Escrow to Vesting Escrow in a single atomic transaction. Move guarantees that

this operation either succeeds completely or reverts entirely, eliminating the risk of inconsistent states where funds are accepted but vesting streams are not created.

11.4 Precision Arithmetic and Pricing

The sale logic utilizes rigorous math standards to prevent the rounding errors and overflow attacks common in DeFi.

- **Overflow Protection:** All calculations use unsigned 128 bit integers (u128). Since Move prohibits silent overflow, developers are forced to handle boundary conditions explicitly.
- **Fixed Point Precision:** The system uses a base precision factor to ensure price calculations remain stable without relying on floating point approximations. APT deposits are multiplied by a precision constant before division, ensuring that token distribution remains accurate regardless of the decimal scale.

11.5 Deterministic Vesting Logic

Vesting is calculated using a pure arithmetic formula based on blockchain timestamps, ensuring auditability and predictability.

- **Linear Unlock:** The system records the start time, end time, and total allocation. At the moment of a claim, the contract computes the unlocked amount as a linear proportion of the time elapsed.
- **State Independence:** The formula depends solely on the on chain timestamp and the immutable parameters of the stream, making it immune to external environmental manipulation.

11.6 Hyperion Integration and Opaque Locking

Liquidity locking is designed for flexibility and future compatibility with the Hyperion DEX.

- **Opaque Byte Vectors:** Lilipad records LP position references as opaque byte vectors rather than decoding them on chain. This allows the protocol to lock complex or evolving LP position types without requiring contract upgrades.
- **Time Based Enforcement:** Once an LP reference is recorded, the lock module enforces time based restrictions on the underlying fungible components, preventing withdrawal until the specified maturity date.

11.7 Event Emission and Indexing

The protocol emits structured events for every critical state transition, enabling easy integration for front end applications and analytics.

- **Comprehensive Coverage:** Events are emitted for sale creation, deposits, purchases, stream formation, claims, and locks.
- **Indexer Friendly:** Event schemas are designed to allow off chain indexers to reconstruct the full history of a project launch without performing complex state traversals.

11.8 Parallel Execution Optimization

Lilipad is optimized to leverage the Aptos Block-STM execution engine.

- **Non Overlapping State:** Because each sale entry and vesting stream is isolated within its own storage slot in the Table architecture, transactions for different projects or different buyers rarely touch the same memory location.
- **High Throughput:** This isolation allows the Aptos runtime to execute high volumes of purchase and claim transactions in parallel, resulting in faster settlement and lower gas contention during high demand launches.