

# Migratoor



Migrate to Scroll L2 with Zero Overhead.  
Empower Your Project with vLayer's  
Time Travel and Scroll's L1 State  
Reading Features.

# Problems we are solving

## L2 Migration Challenges

Many projects launched on Layer 1 (L1) aim to move to Layer 2 (L2) for improved user experience and additional features.

One approach is complete migration, abandoning multi-chain operations.

## No Standard Migration Mechanism

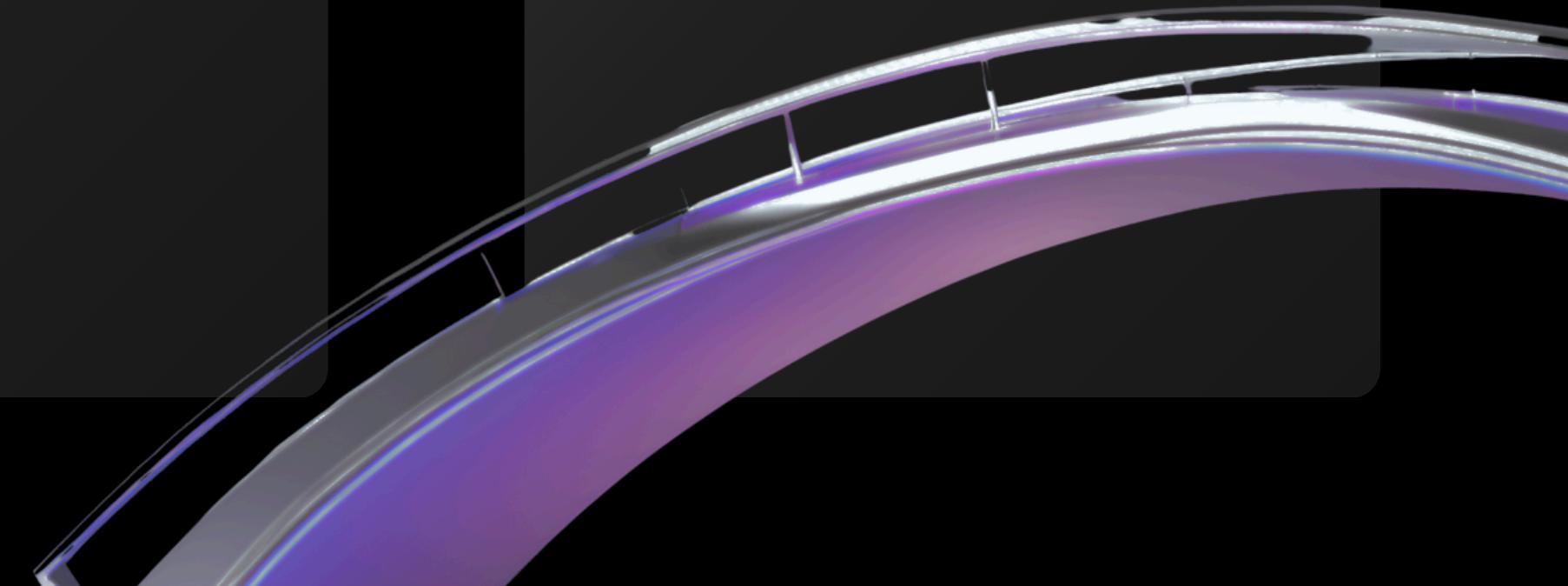
Migration currently requires projects to build their own solutions.

This creates a fragmented and non-seamless experience for users.

## Lack of User Alignment

Users often lack incentives to migrate.

Instead, they may sell tokens on L1, disrupting the migration process.



# Solution

- We have developed a universal trustless protocol that simplifies migration for any project, offering flexibility and ease of use.



# Key Features

## Flexible Migration Options

- **Snapshot-based migration:** Seamless and precise.
- **Time-Averaged Balance migration:** Ensures fair and balanced transitions.

## Customizable Time Ranges

Tailor the timing of snapshots to fit project-specific needs.

## User Incentivization Strategies

- **Time-based rewards:** Encourage users to act within specific timeframes.
- **Protocol loyalty bonuses:** Reward users for long-term commitment.
- **Early migration incentives:** Motivate early adopters with exclusive benefits.



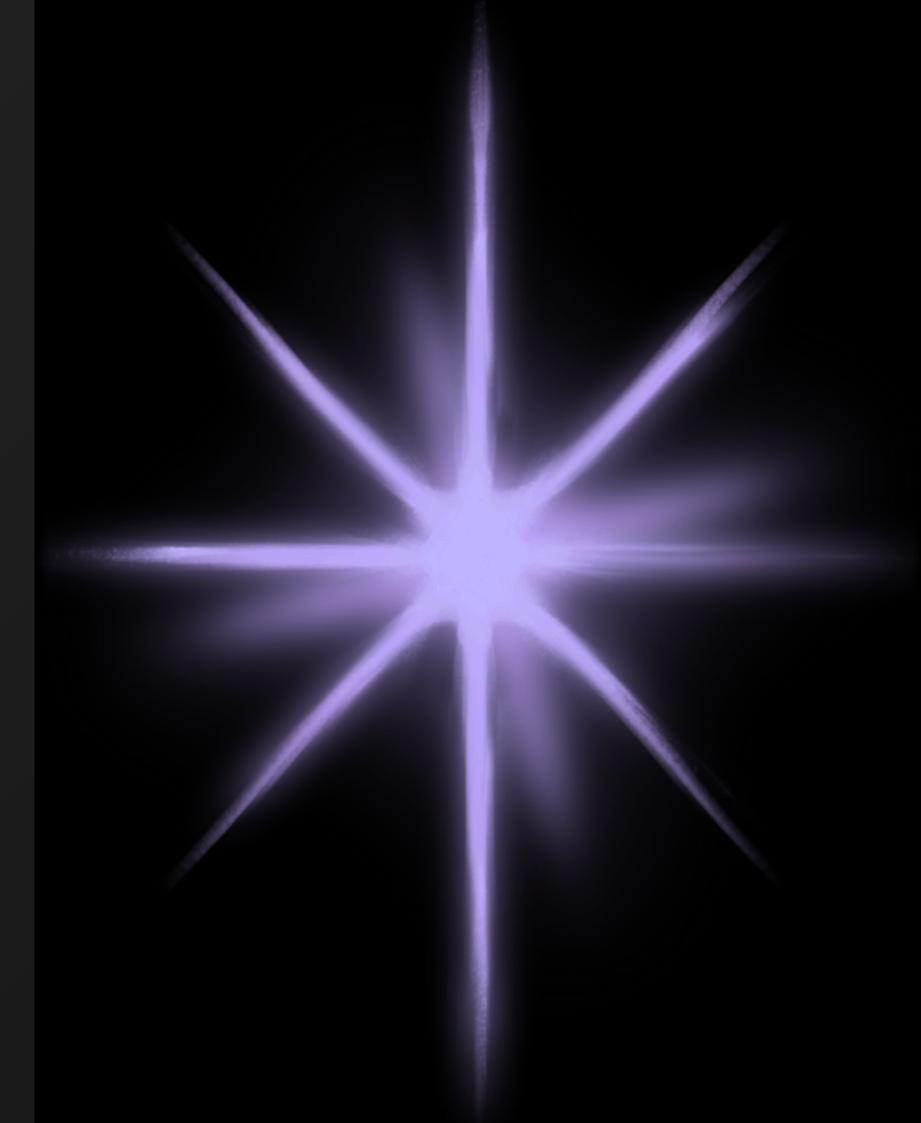
# How Did We Achieve This?



## vLayer's Time Travel Technology

Enables retrieval of historical data, particularly user balances.

Eliminates centralization risks (e.g., relayers) by leveraging: zk-Proof Soundness and Completeness: Secured through zkEVM.



## Scroll's Innovative L1SLOAD Precompile

Provides direct access to L1 storage for verifying that users have locked tokens on L1.

The background features several abstract, semi-transparent 3D geometric shapes in shades of blue, pink, and yellow, scattered across the dark purple surface.

Dive deeper  
into solution

# Project



Deploys L2-native token



Calls **Migratooor factory**  
(deploys migrator infrastructure for  
the protocol)



Escrows tokens for users migration in Migratooor

# User



Locks L1 tokens



Generate zk-Proof of  
average balance



Use Proof to Claim Tokens with  
Potential Bonuses on **Scroll**



# Features



Community Goods  
Infrastructural  
Protocol



Seamless UX for  
protocol and users



Incentive  
mechanism



Secured protocol by  
ZK proofs



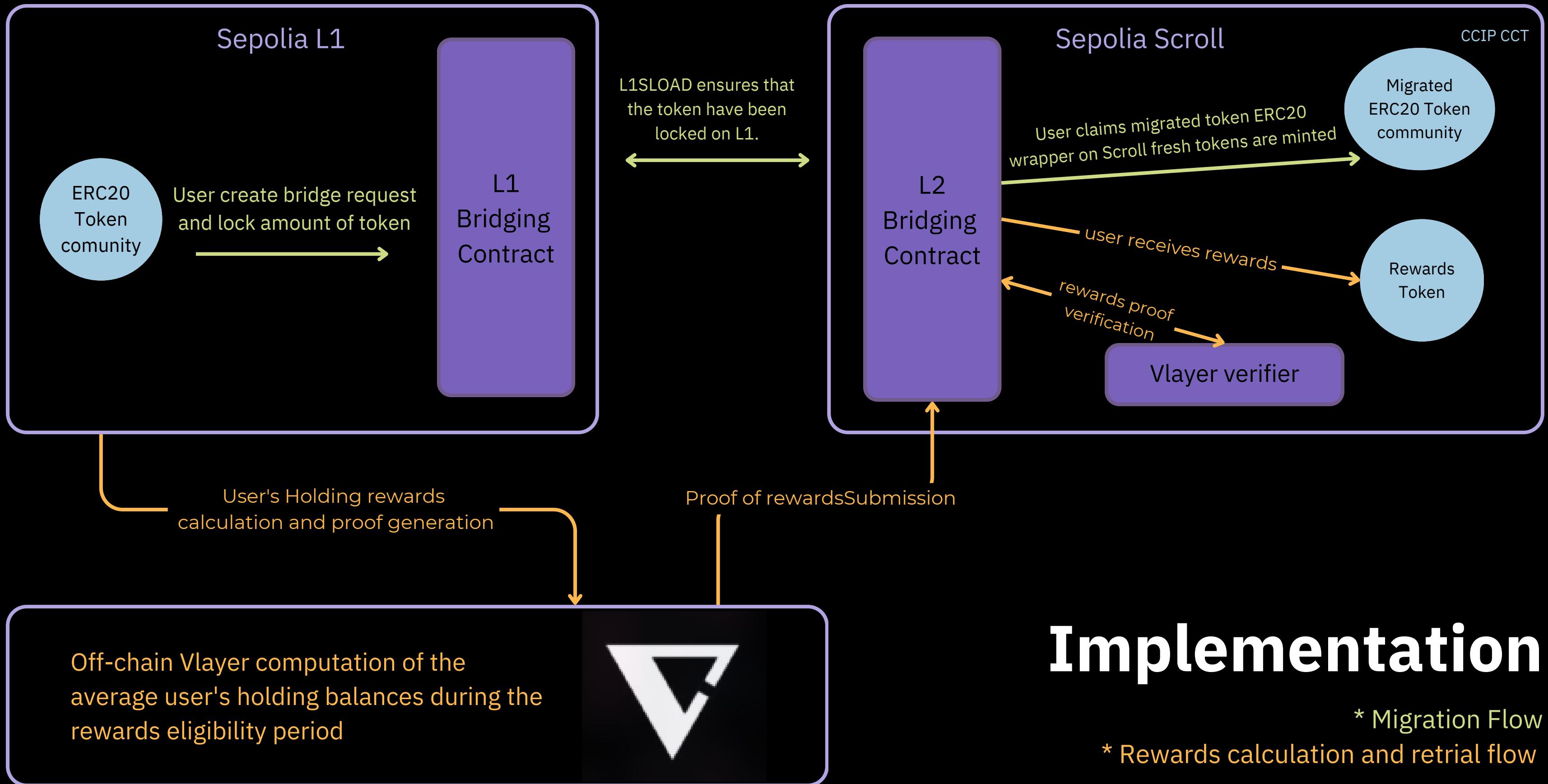
Scroll Tech-stack  
advantages!



Out of the box inter-chain  
L2-native token thanks to  
CCIP integration



# **Roll-up centric roadmap**



# Implementation

\* Migration Flow

\* Rewards calculation and retrial flow

# A little bit more about Rewards

Reward token can be the same as migrated one, in order to avoid migrated token inflation, this can be configured by the protocol itself.

Protocol owners should lock the rewards token into the bridge before starting claiming period. The amount of reward per reward unit is calculated as:

$\text{TotalRewardsTokenLocked} / \text{TotalRewardsUnitsGenerated}$

# Thank you!

See you on the winners stage