Dynamo Protocol - Liquidity Re-utilization and Yield Generation from Liquid Staking Tokens and Real World Assets

DynamoProtocol Team

contact@DynamoFinance.tech

GitHub: https://github.com/DynamoProtocol

Abstract

Dynamo Protocol is an innovative solution designed to unlock and re-utilize liquidity from assets traditionally considered idle within the blockchain and DeFi ecosystems. By tapping into both Liquid Staking Tokens (LSTs) and Real World Assets (RWAs), Dynamo aims to create a versatile and universal liquidity source. Here's a closer look at how Dynamo achieves this and its potential impact on the DeFi landscape:

Leveraging Liquid Staking Tokens (LSTs): In proof-of-stake blockchains, users lock their tokens to support network security and earn staking rewards. However, these staked tokens become illiquid as they are locked away and can't be used for other activities. Liquidity solutions have been created to create Liquid Staking Tokens (LSTs).

Re-utilizing Liquidity from LSTs: Through innovative mechanisms, Dynamo Protocol these LSTs to be utilized in DeFi activities like derivatives trading or yield farming, expanding the usefulness of LSTs beyond restaking other decentralized services.

Utilization of Real World Assets (RWAs): Dynamo extends its liquidity generation capabilities to real-world assets, such as tokenized gold ETFs. These assets, represented on the blockchain, provide a bridge between traditional finance and DeFi.

Enhancing DeFi Liquidity and Stability: By incorporating RWAs, Dynamo diversifies the sources of liquidity, making the protocol robust and resilient to market fluctuations that typically affect crypto assets.

Universal Liquidity Reservoir: Dynamo's approach creates a universal liquidity reservoir that various DeFi platforms can tap into. This liquidity can be used for a wide range of financial activities, from simple swaps to complex financial derivatives.

Dynamo Protocol enables the use of staked, idle, or otherwise illiquid assets across different DeFi platforms, enhancing their utility and generating additional yields for asset holders.

Yield Generation and Risk Management: Innovative Yield Opportunities: By unlocking the liquidity of staked tokens and RWAs, Dynamo offers new yield-generating opportunities. Users can earn from their assets without the need to unstake them, preserving their staking rewards while gaining additional income streams.

Enhanced Stability: The inclusion of RWAs also helps mitigate risks associated with crypto volatility, providing a more stable liquidity source for DeFi platforms.

Dynamo Protocol creates universal liquidity in the DeFi space by regenerating liquidity from staked tokens and real-world assets. This approach not only enhances the utility and yield potential of these assets but also contributes to a more robust and versatile liquidity framework for DeFi platforms. As it develops, Dynamo has the potential to significantly impact the way liquidity is sourced and utilized in decentralized finance.

Contents

List of Figures

1 Introduction

1.1 Background and Motivation

The decentralized finance (DeFi) sector has revolutionized traditional finance by enabling open, permissionless, and transparent financial services. Despite its rapid growth, DeFi faces significant liquidity challenges, primarily due to the illiquidity of staked assets and under-utilization of real-world assets (RWAs).

Proof-of-stake (PoS) blockchains, which secure networks through staking, often lock up a substantial amount of tokens. This staking mechanism, while essential for network security, limits the availability of these tokens for other financial activities. Similarly, RWAs such as tokenized gold ETFs represent a vast reservoir of value that remains largely untapped in DeFi.

Dynamo Protocol aims to address these challenges by regenerating liquidity from Liquid Staking Tokens (LSTs) and RWAs. By converting these assets into DynamoDollar (DYD), a stablecoin pegged to USD, Dynamo Protocol provides a stable and versatile liquidity source for DeFi platforms.

1.2 Objectives

The primary objectives of this paper are to: 1) Analyze the current state of liquidity in DeFi and the limitations of existing solutions. 2)Explore the potential of LSTs and RWAs as sources of liquidity. Describe the architecture and operational mechanisms of Dynamo Protocol. 3)Provide a technical discussion on tokenization, liquidity dynamics, and stablecoin mechanisms. 4)Discuss the benefits, use cases, and challenges associated with Dynamo Protocol.

Ultimately, this paper 1) proposes a novel liquidity solution that integrates staked and real-world assets into DeFi. 2) introduces DynamoDollar, a stablecoin that transforms illiquid assets into a versatile liquidity source. 3) details the Dynamo Rollup Blockchain's advanced security features and its role in supporting Dynamo Protocol. 4) presents models for the protocol's core mechanisms and evaluating their impact on liquidity and stability.

Figure 1: Dynamo Protocol = Liquidity Regeneration from LSTs and RWAs for DeFi applications

2 DeFi and Liquidity Challenges

2.1 Importance of Liquidity in DeFi

Liquidity is essential for the efficient functioning of DeFi platforms. High liquidity ensures that assets can be easily exchanged with minimal impact on their price, supporting smooth and efficient market operations. In DeFi, liquidity enables:

Trading: Low slippage and efficient price discovery on decentralized exchanges (DEXs).

Lending: Reliable collateralization and loan availability in lending protocols.

Yield Generation: Effective functioning of yield farming and staking mechanisms.

2.2 Current Liquidity Sources and Limitations

DeFi platforms primarily source liquidity from native cryptocurrencies and stablecoins. These sources, however, face limitations:

Volatility: Native cryptocurrencies are often volatile, leading to instability in liquidity pools.

Finite Supply: Stablecoins, while stable, are limited in supply and can become scarce during high demand.

Locked Assets: Staked tokens in PoS blockchains are locked and cannot contribute to liquidity.

2.3 Role of Liquid Staking and Real World Assets

Liquid Staking Tokens (LSTs) LSTs represent staked assets that continue to earn staking rewards while remaining liquid and usable within DeFi. They provide a solution to the liquidity constraints posed by traditional staking mechanisms.

Real World Assets (RWAs) RWAs, when tokenized, offer a stable and substantial source of liquidity. Tokenized assets such as gold ETFs provide a bridge between traditional finance and DeFi, offering stability and value to the otherwise volatile DeFi market.

3 Dynamo Protocol

Dynamo Protocol aims to create a universal liquidity reservoir that supports the DeFi ecosystem by unlocking the value of staked tokens and real-world assets. Its mission is to transform these traditionally illiquid assets into DynamoDollar (DYD), a stablecoin that enhances liquidity and stability across DeFi platforms. Dynamo Protocol is comprised of several key components:

1. DynamoDollar (DYD): A stablecoin pegged 1:1 to USD, backed by LSTs and RWAs.

2. Dynamo Rollup Blockchain: An application-specific EVM-compatible rollup providing secure and efficient transaction processing.

3. Homomorphic Encryption and Zero-Knowledge Proofs: Advanced data security features.

DynamoDollar (DYD) serves as the core of Dynamo Protocol's liquidity strategy:

1. Pegged to USD: Ensuring stability and predictability.

2. Backed by Diverse Assets: Supported by LSTs and RWAs, providing a diversified and resilient liquidity base.

3. Utilized Across DeFi: Used in various DeFi activities, including trading, lending, and derivatives, to generate yield and provide liquidity.

3.1 Dynamo Rollup Blockchain with Data Security

Dynamo services are powered by Dynamo Rollup Blockchain. Dynamo Rollup features a unique data security model based on homomorphic encryption and zero knowledge proofs. Homomorphic encryption based Data Security smart contract system offers Data Security of transaction data while preserving all the benefits of smart contract based financial transactions. Rollup blockchain offers high transaction speed and throughput suitable for real world financial transactions. [2]

Practical usage of DeFi requires user experiences similar to that of traditional financial transactions. To achieve such an end, Dynamo Protocol employs a rollup blockchain architecture compatible with the EVM smart contract ecosystem. The Dynamo Rollup Blockchain is built on top of established Rollup Stack, which has been proven in real world financial transactions. [1]

Armed with two key innovative features, Data Security smart contract based on homomorphic encryption and EVM equivalent rollup blockchain, Dynamo Protocol brings Data Security and speed to real world asset transactions, and provides user experiences meeting real world expectations. [1]

3.2 Homomorphic Encryption and Data Security Smart Contract

Ensuring data Data Security is a critical concern for smart contracts handling sensitive information. Dynamo Protocol adopts the ZeeStar system, a language and compiler that allows non-experts to create private smart contracts and perform operations on external data. The ZeeStar language enables developers to specify Data Security constraints conveniently using zkay's Data Security annotations. The ZeeStar compiler then guarantees the realization of these constraints by combining non-interactive zero-knowledge proofs and additively homomorphic encryption. ZeeStar is practical, as it prepares transactions for our contracts in at most 54.7 seconds, at an average cost of 339,000 gas.[2]

4 Technical Implementation

4.1 Dynamo Rollup Blockchain

The Dynamo Rollup Blockchain is a proprietary EVM-compatible rollup designed to support the Dynamo Protocol. Key features include:

1. Scalability: Efficient transaction processing to handle high volumes of DeFi activity.

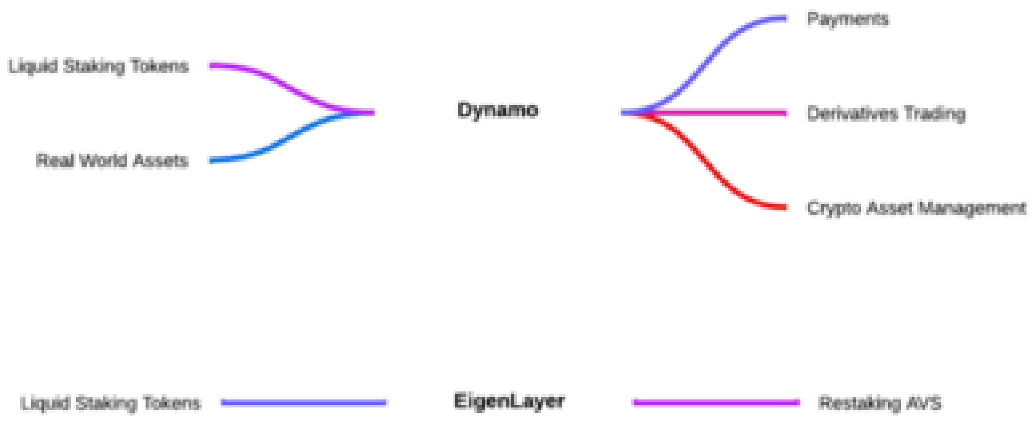

Figure 2: Dynamo Architecture

2. EVM Compatibility: Full compatibility with Solidity, allowing easy deployment and integration of existing smart contracts.

3. Data Security: Unique security features, including homomorphic encryption and zero-knowledge proofs, to ensure data privacy and integrity.

Scalability and Efficiency Dynamo Rollup uses zk-Rollups to batch transactions off-chain, reducing on-chain congestion and enhancing throughput. This allows for faster and cheaper transactions, essential for supporting high-frequency DeFi activities.

EVM Compatibility The rollup is fully compatible with Ethereum's Virtual Machine (EVM), enabling seamless integration with the vast ecosystem of Ethereum-based DeFi protocols. This compatibility allows developers to deploy and interact with smart contracts without modifications.

4.2 Dynamo Data Security Smart Contact with Homomorphic Encryption

Dynamo Protocol employs advanced cryptographic techniques to secure data and transactions:

A. Non-interactive Zero-knowledge Proofs A non-interactive zero-knowledge (NIZK) proof enables a prover to convince a verifier that she possesses a secret without disclosing the secret itself. Specifically, she can demonstrate knowledge of a secret witness w that satisfies a given predicate $\phi(w;x)$ for some public value x, without revealing any information about w other than the fact that $\phi(w;x)$ holds. Here, ϕ is referred to as the proof circuit, w is the private input, and x is the public input. [2]

For instance, in a cyclic group G with generator g and $h \in G$, one can prove knowledge of the discrete logarithm z of h with respect to base g using the proof circuit $\phi(z;h)$, which is satisfied if and only if $g^z = h$.

Zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs) are a type of generic NIZK proof construction that supports any arithmetic circuit ϕ and offers constant-cost proof verification proportional to the size of ϕ (plus a typically negligible linear cost in the size of x). Due to their efficient verification costs, zk-SNARKs are commonly utilized on the Ethereum blockchain.[2]

B. Additively Homomorphic Encryption An additively homomorphic encryption scheme enables the addition of plaintexts corresponding to a pair of ciphertexts without requiring knowledge of private keys. Formally, let pk_α and sk_α be the public and private keys of a party α, respectively, and $Enc(x, pk_\alpha, r)$ represent the encryption of plaintext x under pk_α using randomness r. This scheme is additively homomorphic if there exists a function $\oplus$ on ciphertexts such that for all x, y, α, r, r_0:

$$Enc(x, pk_\alpha, r) \oplus Enc(y, pk_\alpha, r_0) = Enc(x + y, pk_\alpha, r_{00})$$

for some r_{00}, where $\oplus$ can be efficiently evaluated without knowledge of sk_α. It's important to note that both arguments to $\oplus$ must be encrypted under the same public key. Typically, additively homomorphic schemes also allow the homomorphic evaluation of subtraction using a function defined analogously.[2]

For instance, the Paillier encryption scheme is additively homomorphic in Z_n (i.e., addition in Eq. (1) is modulo n) for an RSA modulus n, and exponential ElGamal encryption over a group G is additively homomorphic in $Z_{|G|}$, where $|G|$ is the order of G (see App. B).[2]

Data Security Annotations and Types. To facilitate precise and user-friendly specification of Data Security constraints, ZeeStar utilizes Data Security annotations inspired by zkay. These annotations track ownership of values within a Data Security type system: Data types τ (such as integers and booleans) are extended to types of the form $\tau@\alpha$, where α determines the owner of the expression. The value of an expression can only be accessed by its owner. The owner α may be ”all” (indicating the value is public) or an expression of type address. Expressions with owner ”me” are referred to as self-owned, while those with owner $\alpha \notin \{me, all\}$ are considered foreign.

To prevent implicit information leaks, private expressions with owner α cannot be directly assigned to variables with a different owner $\alpha_0 \neq \alpha$. Instead, developers can use the $reveal(e, a)$ function to explicitly disclose a self-owned expression e to another owner a.

It's important to note that the Data Security annotations entail minimal overhead compared to existing non-private smart contract languages such as Solidity. As discussed further, Data Security is automatically enforced by ZeeStar's compiler, eliminating the need for developers to manually instantiate cryptographic primitives.[2]

Compilation. ZeeStar compiles the input contract into an executable Ethereum contract that enforces the specified Data Security constraints.[2]

In the output contract, values with an owner $\alpha \neq$ ”all” are encrypted under the public key of α using an additively homomorphic encryption scheme. Private expressions are precomputed locally (off-chain) by the sender and only published on the blockchain (on-chain) in encrypted form. Expressions revealed to all are additionally published in plaintext.[2]

In essence, any expression involving only public and self-owned variables is computed by the sender as follows: First, decrypt any private input variables. Then, evaluate the expression using the plaintext arguments. Finally, if the expression is private, encrypt the result using the owner's public key.[2]

Leveraging Homomorphic Encryption. As the encryption scheme used by ZeeStar is additively homomorphic, it also permits the evaluation of expressions. First, the sender re-encrypts the plaintext value val under the public key of to to obtain a ciphertext c. Then, the sender computes bal. In the proof circuit ϕ, ZeeStar ensures that c is computed correctly. Interestingly, the operation $\oplus$ is also evaluated within the proof circuit. While not necessary for Data Security, this practice leads to reduced on-chain costs. Additionally, as we will discuss shortly, it allows for greater expressivity.[2]

After constructing ϕ, ZeeStar inserts a proof verification statement into the output contract. When calling the transfer function, the sender must generate and provide a NIZK proof for the circuit ϕ as a function argument proof. The public arguments of ϕ are provided as arguments to verify. If verification fails, the transaction is rejected, and the contract state is reverted.[2]

4.3 Dynamo Rollup Blockchain

The Dynamo Rollup Blockchain is constructed on proven rollup technology stack, which addresses limitations commonly found in layer 1 smart contract systems. One embodiment of the Dynamo Rollup is based on the Arbitrum Stack and introduces a novel approach to overcome common limitations of layer 1 blockchains.[1]

Arbitrum contracts are highly cost-effective for verifiers to handle. When participants act in line with incentives, Arbitrum verifiers only need to verify a small number of digital signatures for each contract. Even in cases where parties deviate from their incentives, Arbitrum verifiers can efficiently resolve disputes regarding contract behavior without needing to inspect more than a single instruction execution by the contract.[1]

Additionally, Arbitrum enables contracts to execute privately, disclosing only hashed versions of contract states. Dynamo utilizes the Arbitrum technology stack as the foundation of its Rollup Blockchain. Moreover, the Dynamo Protocol customizes and optimizes the Arbitrum stack for DeFi operations.[1]

Dynamo Rollup Blockchain is further optimized for DeFi by providing customized middle-ware modules and precompiled smart contacts, including sub-block time data API. [1]

Dynamo Rollup Blockchain may be implemented on various rollup architecture besides Arbitrum. In this paper, Arbitrum is used for illustrative purposes.

4.4 Dynamo Protocol - Data Security and High Performance for DeFi

Armed with Data Security smart contract and rollup blockchain, Dynamo Protocol solves two of the major hurdles hindering the wide adoption of decentralized finance transactions for real world tokenized assets, data Data Security and transaction speed and throughput.

Data Security smart contracts are usually computationally intensive. This translates into high gas consumption and fees in blockchain ecosystem. Dynamo Rollup Blockchain solves these drawbacks with a highly efficient and low cost EVM compatible ledger.

Together with Data Security smart contract and rollup blockchain, Dynamo Protocol offers community a viable and robust solution to bring tokenized real world assets and related DeFi transactions to an industry scale.

5 Dynamo DeFi Primitives

5.1 Tokenization and Collateralization

The tokenization process involves converting staked assets and RWAs into digital tokens. For Liquid Staking Tokens (LSTs), this is represented by:

$$\text{LST}_i = \text{Tokenize}(S_i) \tag{1}$$

where S_i represents the staked asset i.

For Real World Assets (RWAs), the tokenization is given by:

$$\text{RWA}_i = \text{Tokenize}(A_i) \tag{2}$$

where A_i denotes the real-world asset i.

These tokenized assets form the collateral backing for DynamoDollar:

$$\text{Collateral}_{\text{DYD}} = \sum_i (\text{LST}_i + \text{RWA}_i) \tag{3}$$

5.2 Liquidity Pool Dynamics

The dynamics of the liquidity pool, which aggregates LSTs and RWAs, can be modeled as:

$$L_t = \sum_{i=1}^{n} (\alpha_i \times \text{LST}_i(t) + \beta_i \times \text{RWA}_i(t)) \tag{4}$$

where L_t represents the total liquidity at time t, and α_i, β_i are the weightings of each asset type in the pool.

The yield generated from the liquidity pool is distributed according to:

$$Y_t = \lambda_t \times L_t \tag{5}$$

where Y_t is the yield at time t and λ_t is the yield rate.

5.3 Stablecoin Pegging Mechanism

Maintaining the peg of DynamoDollar to USD involves ensuring that the collateral value matches the circulating supply of DYD. This can be formulated as:

$$V_{\text{collateral}} \geq \sum_{i=1}^{m} \text{DYD}_i \tag{6}$$

where $V_{\text{collateral}}$ is the value of the collateral backing the stablecoin and DYD_i is the amount of DynamoDollar in circulation.

6 Dynamo Protocol Economic System and Governance

Dynamo Rollup Blockchain utilizes the ERC20 token, DynamoProtocol Token DYNAMO, as a payment method for transaction fees. A portion or all of the transaction fees may be distributed to verifiers of transactions. A portion of the transactions fees may also be burned based on community decision.

Community decisions are based on votes by community members. Community members may vote in proportion of the number of DynamoProtocolToken owned for a proposal.

7 Conclusion

Dynamo Protocol is a high performance, low cost, and Data Security DeFi system optimized for tokenized real world assets. Through innovative homomorphic encryption and rollup blockchain technologies, Dynamo Protocol provides the necessary technological foundation for the wide adoption of tokenized real world assets and related decentralized financial transactions.

7.1 Future Work

We plan to further strengthen the Dynamo system by focusing on the following areas:

- Production grade Data Security smart contract templates
- Decentralized spot and derivatives exchanges tailored towards RWA
- Real world data integration services

7.2 Acknowledgements

We would like to acknowledge 1) ETH and ZeeStar for providing the foundation for Dynamo Data Security Smart Contract system. 2) Arbitrum for providing the foundation for Dynamo Rollup Blockchain.

7.3 Whitepaper Versions

- Dynamo v. 1.0 – Jan. 2024, initial release

7.4 Code Base

Codebase: `https://github.com/DynamoProtocol`

References

[1] Xiaoqi Chen S. Matthew Weinberg Edward W. Felten Harry Kalodner, Steven Goldfeder. Arbitrum: Scalable, private smart contracts. https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-kalodner.pdf, 2018.

[2] ROGER BAUMGARTNER MARTIN VECHEV SAMUEL STEFFEN, BENJAMIN BICHSEL. Zeestar: Private smart contracts by homomorphic encryption and zero-knowledge proofs. https://www.sri.inf.ethz.ch/publications/steffen2022zeestar, 2022.