BRINGING EQUITIES TO ON-CHAIN FINANCE

LIQUIDITY FOUNDATION

December 2024

Abstract. The Liquidity Equity Protocol is a system for representing traditional equities as on-chain tokens through a three-layer architecture: vaults in the form of segregated portfolios storing underlying securities, a token issuance engine managing the minting and burning of equity-backed ERC-20 tokens, and an equity operations engine handling corporate actions. The protocol maintains 1:1 backing through automated reconciliation while enabling near-permissionless transfers. Token holders can participate in corporate actions including dividends and voting through delegation, with all equity instruments sharing consistent behaviour through a proxy-based, computationally efficient token factory.

Contents

1. Introduction

Global financial markets face a critical gap: the absence of a high-performance, US-based exchange that bridges traditional equity markets with digital asset trading while maintaining institutional-grade execution capabilities. This paper sketches the first and principal step towards closing this gap: an equity protocol enabling traditional securities to be traded on-chain.

This paper introduces the Liquidity Equity Protocol, a system for representing traditional equities as on-chain tokens. Through a novel architecture combining robust vaults in the form of segregated portfolios, automated reconciliation, jurisdiction-specific transfer restrictions, and delegated corporate actions, the protocol enables near-permissionless issuance of equity-backed tokens that preserve beneficial ownership rights.

The paper is organized as follows. Section 2 describes the protocol's general architecture and core components. Section 3 details the token design and deployment engine. Section 4 introduces how the protocol handles equity operations including redemption, dividends, voting, and stock splits. Illustrations are provided in Appendix A.

2. General Architecture

The Liquidity Equity Protocol establishes a bridge between traditional equity markets and decentralised finance through a system of tokens representing beneficial ownership of underlying securities. The protocol consists of three core components. Equities are stored off-chain in *vaults*. The *token issuance layer* is the engine driving the issuance and redemption of on-chain claims representing beneficial ownership in the equities stored by a vault. The *equity operations layer* is the engine managing transfer restrictions, dividends, and voting rights. The architecture is illustrated in Figure 1.

2.1. Vaults.

The bridge between off-chain equities and on-chain tokens is provided by secure and computationally efficient vaults operating as segregated portfolio companies. A vault is a segregated portfolio within a segregated portfolio company; each portfolio corresponds to a specific equity instrument. This legally separates the assets and liabilities of different token types. The equity contracts deposited in the vault maintain a 1:1 relationship between issued tokens and underlying shares through automated reconciliation after each mint or burn operation, regular update calls, and independent audits. A registered broker-dealer, acting for the vault, executes the underlying securities transactions in an automated manner, while a qualified custodian holds the securities.

2.2. Token Issuance Layer.

The interface between vaults and on-chain trading is implemented through typed ERC-20 contracts with transfer restriction capabilities. Each token type corresponds to a specific portfolio in the segregated portfolio company, with minting and burning controlled by a two-phase commit process that ensures consistency with vault operations. The tokens encode shareholder rights like dividend claims while retaining voting power at the vault level. Transfer restrictions enforce jurisdiction-specific compliance rules.

2.3. **Equity Operations Layer.** The equity operations processes primary market operations through smart contracts that coordinate token minting and burning with the vaults' off-chain operations. This layer validates order parameters against regulatory requirements, manages the flow of funds between users and the vault system during minting and redemption, handles equity events such as dividend dispersion and vote management, obtains reference data from price oracles, and maintains relevant records.

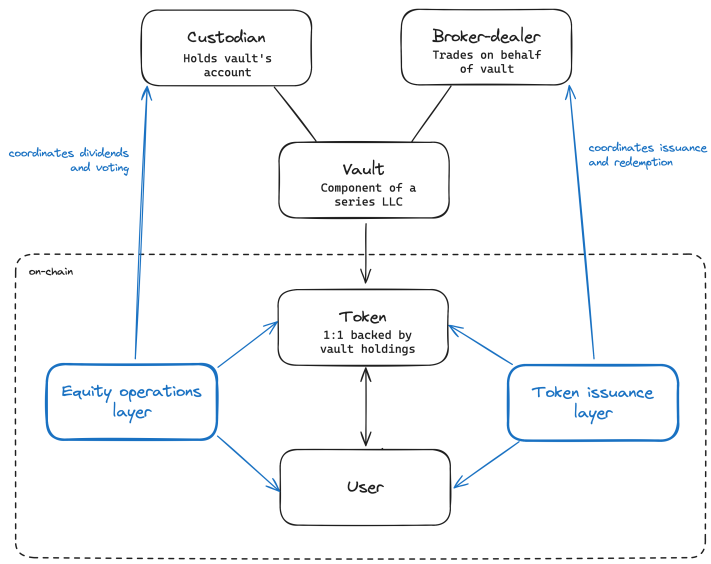

FIGURE 1. General architecture for a given vault

3. TOKEN DESIGN

Equity tokens are implemented through ERC-20 contracts, one for each equity instrument, enhanced with features that implement regulatory restrictions, beneficial ownership rights, and integration with other decentralised finance protocols. The token supply matches the number of shares deposited in the vault 1:1, with minting and burning controlled by the token issuance layer to maintain the correspondence.

3.1. **Token Factory.** New equity tokens need to be created whenever the protocol supports a new stock. We want to ensure that new tokens can be deployed efficiently while maintaining consistent behaviour across all equity instruments. Our solution leverages a proxy pattern that separates token logic from token state. In the background, there is an implementation contract containing the shared equity token logic (from basic ERC-20 operations to equity-specific functions such as dividend claims). Rather than redeploying this logic for each equity instrument, we deploy it once and create lightweight proxies that delegate to this shared implementation.

The engine creating these lightweight proxies is the *token factory.* The token factory ensures that each equity instrument has exactly one corresponding token by using the stock's ISIN to generate a deterministic contract address. Furthermore, token creation is standardised so that all tokens are deployed using consistent parameters, and a registry of all created tokens is maintained.

While the general structure is therefore standardised, each new token deployment still requires:

(1) Registration of the underlying equity details (ISIN, jurisdiction, etc.);
(2) Configuration of jurisdiction-specific transfer restrictions;
(3) Establishment of the corresponding off-chain segregated portfolio; and
(4) Deployment of the token factory with validated parameters.

From these parameters, the factory extracts a unique identifier to generate deterministic addresses for new tokens. This ensures that a given equity instrument will always resolve to the same token address, preventing duplicate tokenisation while enabling predictable address generation. During deployment, the factory temporarily stores initialisation parameters and the implementation address. The proxy retrieves these during its construction, then delegates all subsequent calls to the implementation. This separation of concerns allows us to upgrade the shared implementation contract when necessary without affecting existing tokens' state or functionality. This architecture achieves three objectives: gas efficiency through code reuse, consistent behaviour across all equity tokens through shared implementation, and flexibility to upgrade core logic while preserving token-specific state.

3.2. **Transfer Restrictions.** The protocol implements transfer restrictions for each token type. Each jurisdiction is associated with a `TransferRestrictor` contract that enforces jurisdiction-specific rules across all equity tokens issued from that jurisdiction.

Before any token transfer, the equity token contract queries the relevant transfer restrictor. The restrictor maintains a blacklist of addresses that are prohibited from sending or receiving tokens, typically used for enforcing sanctions compliance, blocking compromised accounts, or implementing emergency freezes.

The approach differs from more complex permissioning systems by defaulting to allowing transfers unless explicitly restricted, rather than requiring pre-approval of all transfers, thereby reducing complexity and gas costs. Additional restrictions can be added through contract upgrades if required by changing regulatory frameworks.

3.3. **DeFi Composability.** The equity tokens are designed to be maximally composable with existing DeFi protocols while upholding transfer restrictions. Each token implements the ERC-20 standard interface, enabling basic integration with other DeFi solutions. Unlike basic ERC-20 tokens, however, composability must account for transfer restrictions. To this end, protocol implements a standardised interface for checking transfer validity, allowing DeFi protocols to verify transferability before attempting transfers.

4. Equity Operations

The Liquidity Equity Protocol implements redemption orders, dividend disbursements, shareholder voting, and stock splits.

4.1. **Redemption.** Redemption flows through all three protocol layers. The equity operations layer receives redemption orders and validates them against the token's parameters—in particular, transfer and redemption restrictions. Upon validation, it initiates a two-phase commit process with the token issuance layer to escrow the tokens for redemption: that is, the token issuance layer locks the tokens in a redemption escrow contract, preventing transfers while the redemption is executed.

Off-chain, the vault processes the redemption through its broker-dealer, who executes a sell order for the underlying equity. The qualified custodian releases the corresponding securities, and settlement proceeds flow back through the vault structure to the redeeming token holder. Once settlement completes, the token issuance layer burns the escrowed tokens, maintaining the one-to-one relationship between tokens.

The process is illustrated in Appendix A, Figure 2.

4.2. **Dividends.** When a company announces a dividend distribution, the vault, as the registered owner of the underlying securities, receives the dividend payment. The equity operations layer captures a token holder snapshot at the announced distribution time and creates a dividend claim contract. The token issuance layer then enables token holders to claim their proportional share of the dividend through this contracts. Dividends are distributed in USDC to minimize complexity and ensure consistent settlement. The claim contract maintains withdrawal rights for a specified period, after which unclaimed dividends distributed may be distributed to other holders according to the terms and conditions of the protocol.

The process is illustrated in Appendix A, Figure 3.

4.3. **Voting.** Voting rights are retained at the vault level rather than passed through to token holders, since the vault is the registered shareholder. While token holders do not directly participate in shareholder votes, the protocol implements a delegation mechanism allowing token holders to influence voting decisions. When a company announces a shareholder vote, the equity operations layer creates a voting contract that enables token holders to signal their voting preference.

Token holders can submit their votes on-chain, with voting power proportional to their token holdings at the voting snapshot time. The vault considers the aggregated token holder preferences when exercising its voting rights as the registered shareholder. Vote outcomes and participation statistics are published through the equity operations layer.

The process is illustrated in Appendix A, Figure 4.

4.4. **Stock Splitting.** When a company announces a stock split, the protocol must adjust token supply to maintain the one-to-one relationship with underlying shares. The equity operations layer monitors for split announcements and initiates a supply adjustment process. The vault continues holding the post-split shares, while the token issuance layer executes a corresponding token rebase. The rebase adjusts all holder balances proportionally according to the split ratio. A similar process handles reverse stock splits, with token supply contracting to match the reduced share count. All token holder proportions remain unchanged through either operation.

The process is illustrated in Appendix A, Figure 5.

Appendix A. Illustrations of Equity Operations

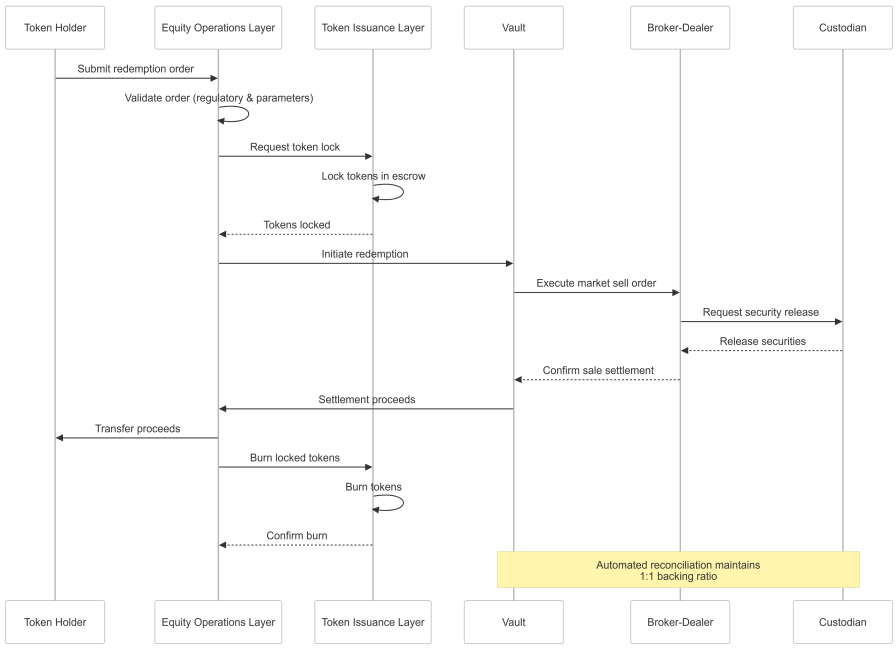

Figure 2. Redemption

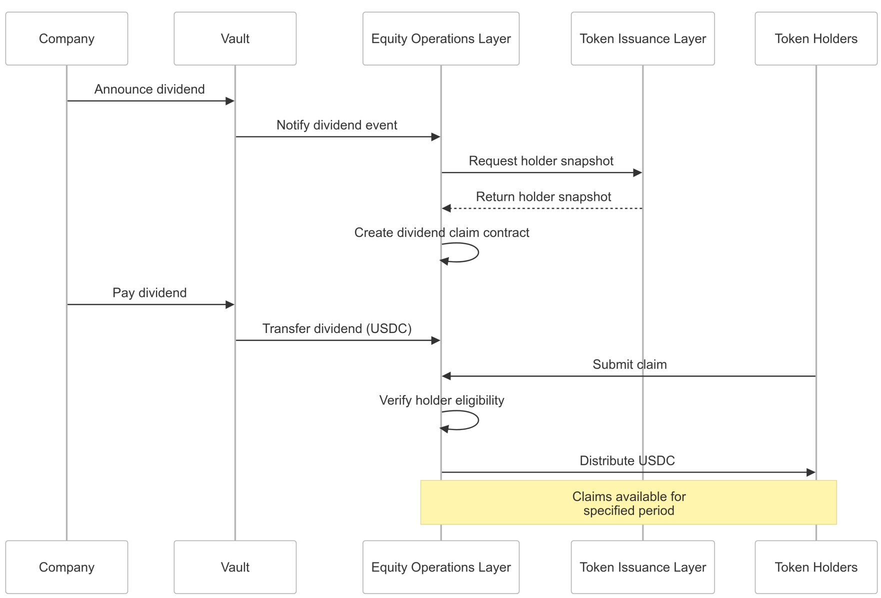

Figure 3. Dividend distribution

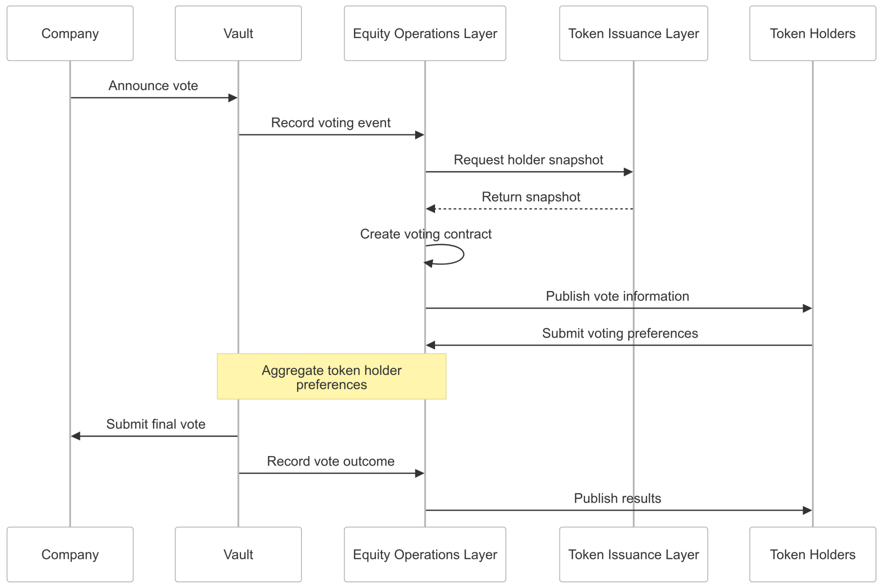

FIGURE 4. Vote delegation

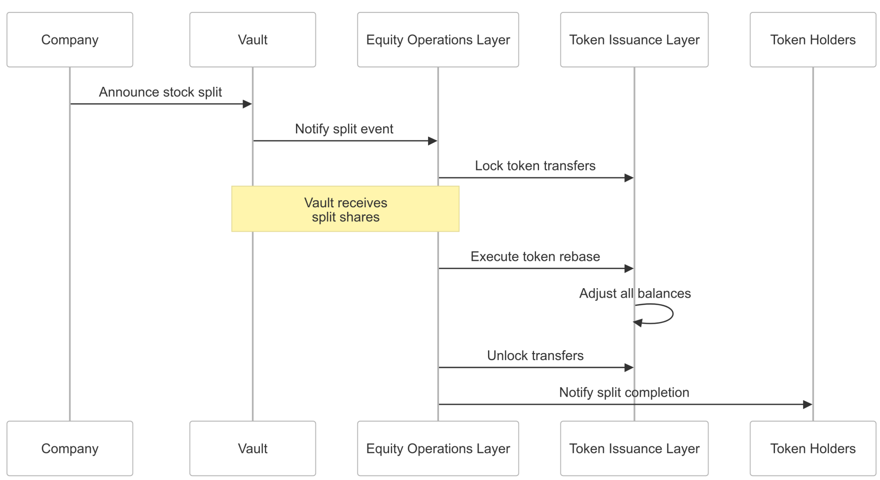

FIGURE 5. Supply adjustment following a stock split