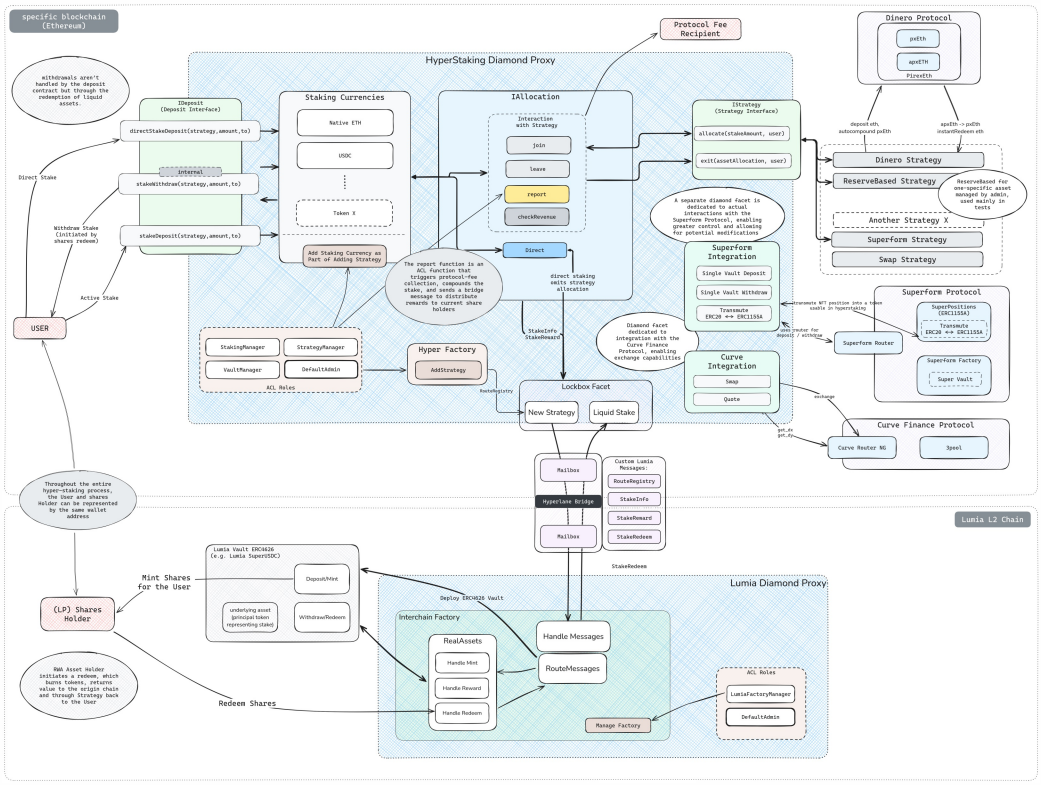
Lumia Smart Contracts

This specification describes the architecture and key features of the Lumia Smart Contracts, which manage staking pools, cross-chain asset handling, and contract upgrades.

The system follows a star architecture, centered on a Diamond Proxy contract deployed on the Lumia Chain to handle interchain communication, ERC-4626 shares minting, and revenue distribution, and multiple Diamond Proxy contracts deployed on various origin chains (e.g., Ethereum, Base, Arbitrum) to manage local deposits, staking pools, and revenue strategies.



0. Diamond Proxy Architecture

Lumia smart contracts is developed using the Diamond Proxy pattern:

- **Modularity and Upgradeability**: The architecture supports adding or replacing functionalities without affecting ongoing operations, ensuring flexible contract modularity.
- **Second Diamond Proxy on Lumia Chain**: The secondary Diamond Proxy operates on the Lumia Chain to manage cross-chain asset handling, effectively functioning as a bridge. It facilitates Hyperlane interchain messaging and ensures that liquid RWA tokens accurately represent real value.

Access Control List (ACL)

The **Access Control List (ACL)** defines roles like `StakingManager`, and `VaultManager`, each with specific permissions to manage different parts of the protocol. The **DefaultAdmin** role holds the authority to assign and revoke these roles, ensuring controlled access to critical functions. Additionally, the **DefaultAdmin** role is also responsible for managing **proxy upgrades**, allowing it to add, replace and delete specific contracts functionality.

Hyperlane Integration

Hyperlane provides secure cross-chain messaging between origin-chain Diamond Proxies and the central Lumia Chain Diamond Proxy. This ensures: +

- **Direct Staking**: Direct staking allows users to stake assets directly on the Lumia Chain without needing to pass through the revenue strategy. This enables faster processing and more flexible staking options, while still maintaining cross-chain compatibility through Hyperlane's messaging infrastructure.
- **Stake Synchronization**: When a user stakes on an origin chain, a Hyperlane message is sent to mint ERC-4626 shares on the Lumia Chain representing the stake plus accrued revenue.
- **Redemption Flow**: When a user redeems or exits shares on the Lumia Chain, Hyperlane relays a burn-and-withdraw instruction to the origin-chain proxy to burn the shares and release the underlying assets and rewards.

This design centralizes staking and redemption logic around Hyperlane messaging, enabling users to seamlessly access multi-chain revenue streams without interacting directly with multiple protocols or bridges.

1. Handling Multiple Deposit Currencies

The Lumia protocol supports staking with both native assets (e.g., ETH) and ERC-20 tokens. In some cases, even NFTs are indirectly supported through certain strategies (e.g., **Superform**). This allows for flexible staking operations across different asset types.

- **Native and ERC-20 Tokens:** The protocol distinguishes between native chain coins (such as ETH) and ERC-20 tokens using the `Currency` struct:

```
/**
 * The Currency struct represents a token
 * If `token` is address(0), it represents native coins (e.g. ETH)
 * @param token Address of the token, address(0) means native chain coin (e.g. ETH)
 */
struct Currency {
    address token;
}
```

- **Strategy Flexibility:** Each strategy can support different types of assets, including both native and tokenized assets. Strategies like **Superform** can even support NFTs indirectly by wrapping them in a compatible format.
- **Unified Management:** Despite supporting various asset types, the system maintains a unified interface for deposits, ensuring consistent behavior across different strategies and asset classes.

2. Revenue Strategies

Low-risk strategies are used to generate optimal income for stakers. All strategies implement a common interface, enabling seamless integration with staking pools and allowing new strategies to be added over time. The interface is defined in the IStrategy.sol contract and includes the following key functions:

```
    /**
     * @notice Allocates a specified amount of the stake to the strategy
     * @param stakeAmount_ The amount of stake received for allocation
     * @param user_ The address of the user making the allocation
     * @return allocation The amount successfully allocated
     */
    function allocate(
        uint256 stakeAmount_,
        address user_
    ) external payable returns (uint256 allocation);

    /**
     * @notice Exits a specified amount of the strategy shares to the vault
     * @param assetAllocation_ The amount of the strategy-specific asset (shares) to withdraw
     * @param user_ The address of the user requesting the exit
     * @return exitAmount The amount successfully exited
     */
    function exit(uint256 assetAllocation_, address user_) external returns (uint256 exitAmount);
```

This interface allows strategies to be deployed independently of the main upgradeable Proxy Diamond code and linked with new staking pools.

- **Shares:** Are minted on the Lumia Chain as ERC-4626 tokens, representing both users' stake and used to distribute revenue from strategies.

Reward Distribution via `report()` Function

A central `report()` function for each strategy aggregates yields from and updates share value.

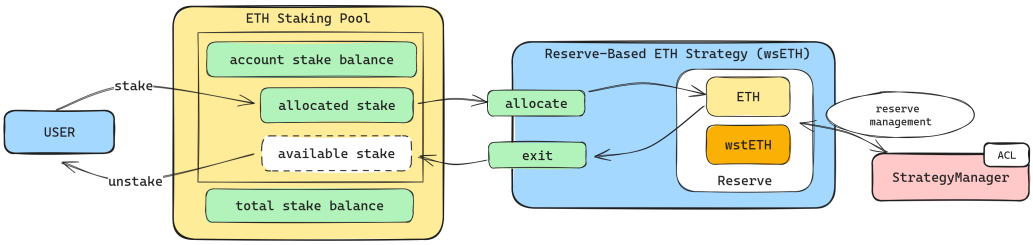
```
/**
 * @notice Harvests and compounds revenue for a given strategy
 */
function report(address strategy) external;
```

Flow:

1. **Trigger**: authorized manager calls `report()`.
2. **Accounting**: Increases total assets; recalculates `pricePerShare = totalAssets / totalShares`.
3. **Cross-chain Distribution**: Emits `RewardsReported(amount)` event; Hyperlane relays report to origin proxies.
4. **User Update**: Share price bump represents distributed rewards; user share holdings automatically reflect yield.

Example Strategy: Reserve-Based Strategy

One example of a strategy is a **reserve-based strategy** focused on yield generation through a specific defined asset (e.g., stETH from the Lido Protocol). This reserve is managed to ensure sufficient liquidity for staking and unstaking operations. When users stake ETH, the strategy allocates a portion of the available wstETH from the reserve to the user, allowing them to benefit from staking rewards generated by Lido.



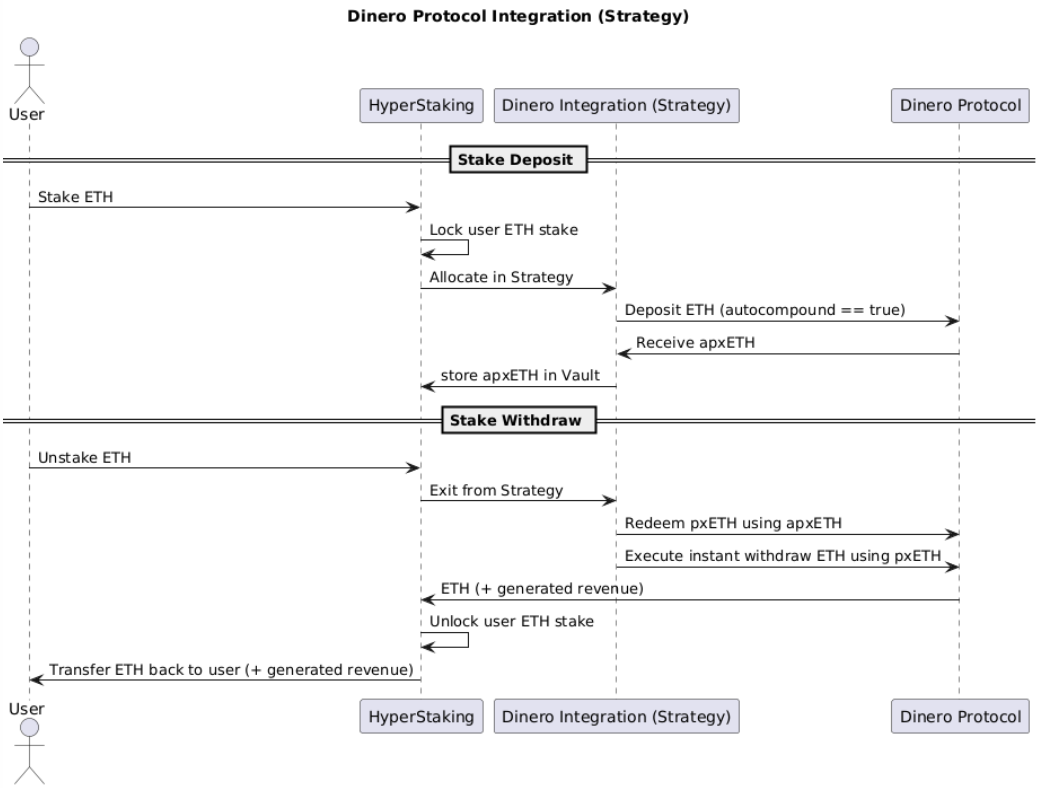
When users exit, the strategy returns their initial ETH plus the generated income, ensuring smooth exits without needing to interact with external protocols for each individual transaction. This approach minimizes transaction costs and optimizes the use of liquidity within the pool.

However, this solution has its limitations. It's possible that the strategy may not have full ETH coverage at certain times. In such cases, the user will still be able to perform a partial unstake. Additionally, the user will not lose any accrued revenue, as it is tracked within the contract, allowing them to claim their rewards once the reserve is replenished.

A simplified version of this strategy has been implemented and is currently being used for testing purposes.

Example Strategy: Dinero Protocol Integration

Another example strategy is the **Dinero Protocol Integration**, focused on yield generation through the **apxETH** token, emitted by the `PirexETH` contract from the Dinero Protocol. The strategy auto-compounds pxETH into apxETH to maximize returns, generating around 8% APY, and is stored in the Lumia `StrategyVault`.



When users stake ETH, the strategy interacts directly with the Dinero Protocol, converting ETH into pxETH, which is then auto-compounded into apxETH. This allows users to benefit from the compounding returns offered by the Dinero Protocol.

When users unstake, the Dinero Protocol is used to redeem pxETH from apxETH (an ERC-4626 vault). pxETH is then converted to ETH for withdrawal, plus accumulated interest, with a 0.5% fee applied.

In the future, the fee could be reduced by implementing a delayed unstake option, creating an unstake buffer for ongoing operations, similar to the model used in the Dinero Protocol.