Omni Protocol

1 Introduction

1.1 Motivation

2 Protocol Architecture

2.1 Liquidity layer

2.1.1 Single coin pool (Pool)

2.1.2 Bridge

2.1.3 Single Coin Pool Manager (PoolManage)

2.1.4 AppManage

2.1.5 Governance

2.2 Application layer

2.2.1 Full-chain lending

2.2.2 PMM

2.2.3 NFT borrowing and lending

2.2.4 OmniSwap

2.2.5 Other Projects

2.3 Message Layer

Application messages

Liquidity pool messages

Cross-chain protocol messages

3 **Protocol Contract**

3.1 Single Coin Pool Contract

3.2 Messaging Protocol Bridge Contract

3.3 Protocol Core Contract

4 **Looking Ahead**

5 **Summary**

1 Introduction

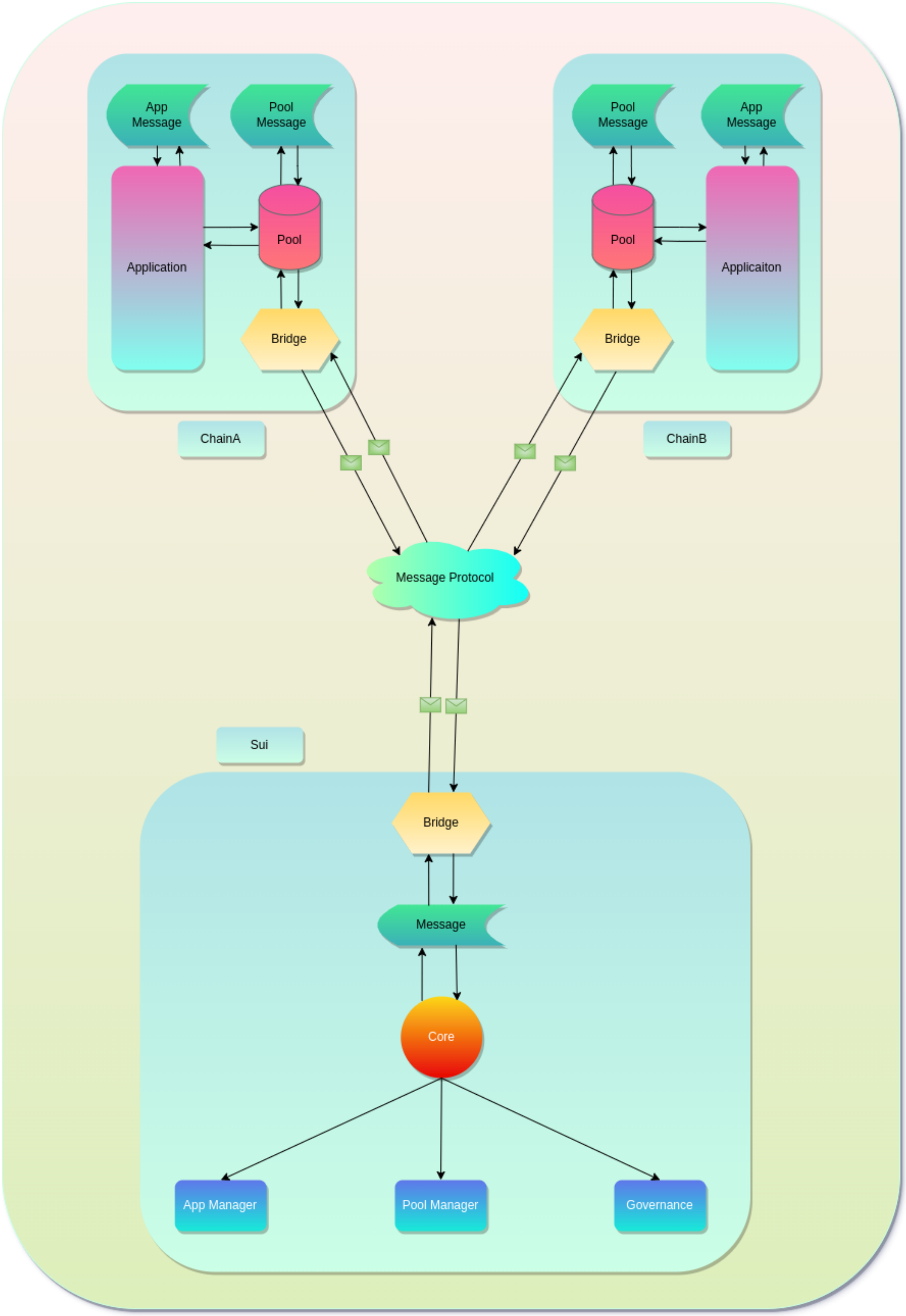
Omni Protocol is a chain-wide liquidity aggregation and settlement system with single coin pools of each public chain as the core, cross-chain messaging protocols such as Wormhole, Layerzero as the bridge, and Sui public chain as the settlement center. Each public chain's single coin pool allows users on any chain to provide liquidity to the protocol. The cross-chain messaging protocol is responsible for interconnecting the single coin pools, and Sui public chain acts as the clearing house, bringing security, concurrency, and low fees to the protocol. Based on the Omni Protocol, we will develop a variety of financial applications. We will develop a full-chain lending application based on the full-chain single coin pool, which can meet the deposit and borrowing needs of different chain users. Compared to the Aave Protocol, which provides liquidity for lending on a single chain and borrows third-party cross-chain bridges for cross-chain lending, all lending operations for full-chain lending can be performed on any chain, which greatly increases lending protocol capital utilization and enables users to perform lending operations without perception of the chain. We will develop a new Dex application that differs from traditional AMM by allowing users to provide unilateral liquidity for unilateral market making. The new Dex application allows for efficient cross-chain Swap, low slippage and reduced impermanent losses. Also compared to the Stargate stablecoin cross-chain bridge, the new Dex application allows non-stablecoin cross-chain swaps, further aggregating chain-wide asset liquidity. Based on single-coin pool full-chain lending, we can extend to NFT full-chain lending in one step to meet the demand for collateralized lending of non-homogeneous assets. We can also plug Omni Protocol into the already live OmniSwap and other Web3 applications as a portal for full-chain liquidity. the original intention of Omni Protocol is to build a full-chain liquidity aggregation and settlement system that is scalable enough to allow us to introduce full-chain liquidity to different Web3 applications.

1.1 Motivation

Since the rise of Web3 smart contract programming, various public chains have emerged. From the initial Evm-based public chains represented by Ether to the current Move-based public chains represented by Sui and Aptos. This phenomenon has led to a considerable degree of fragmentation of liquidity. Currently, 55% of the liquidity is concentrated in Ether, while the rest is scattered in the rest of the public chains. The fragmented liquidity makes it difficult for users of different public chains to access new public chains. Users are required to find Dex for tokens of the new public chain and use token bridges to swap across chains. Different token bridges have different user interfaces that users need to learn, navigate, and use and are not cheap. Users who own assets on different public chains at the same time have difficulty in finding the right project to earn excess revenue.

In order to solve the above problems, we propose Omni Protocol, where users with different public chain assets can provide liquidity to the protocol and easily earn higher excess returns. The emergence of new public chains, the easy scalability of Omni Protocol, provides time guarantee for the first access. Consistent user portal for better operational experience.

2 Protocol Architecture



2.1 Liquidity layer

2.1.1 Single coin pool (Pool)

The single coin pool is the core of the protocol. The liquidity provided by the user is hosted by the single coin pool. The single-coin pool itself is stateless, and the state of the user is managed by the token pool manager aggregation at the core of the protocol. This approach makes the token pool very simple, brings great scalability, and facilitates rapid access to the protocol by new public chains. At the same time, user operations can be performed separately on different chains, greatly increasing the flexibility of user operations. Users can earn revenue by topping up to the protocol on chain A and withdrawing revenue on chain B. Main features of single coin pool:

- Asset ID: Assets of different public chains have unique IDs in the protocol, and similar assets of different public chains use the same ID for unified management (e.g. USDT on Evm and Optimism)
- User recharge: Allow users to initiate recharge to single pool, which is responsible for encoding user address, asset address and asset quantity, and then completing the protocol load delivery through the bridge to realize user's deposit operation.
- User withdrawals: allows users to initiate withdrawals to the single coin pool, which is responsible for encoding the user address and then completing the delivery of the protocol load through the bridge to achieve the user's withdrawal operation
- User recharge and withdrawal: allow users to complete recharge and withdrawal operations in one transaction, the Swap operation in the Dex application can be regarded as a recharge of T1 assets and withdrawal of T2 assets

2.1.2 Bridge

Bridge is responsible for receiving messages from single coin pools and completing message transfer using Wormhole, Layerzero cross-chain messaging protocol. The bridge only allows calls from the single pool and the protocol core. The single coin pool receives user access operations and encodes addresses and quantities, providing the protocol with security for user access.

The bridge is an adapter for different cross-chain messaging protocols. In order to achieve the commonality of messages from different public chains, the protocol will develop a unified message specification. The type of the message attributes meet the unified conversion of numeric values to 64-bit integers, addresses to byte types, and other data to byte types. After the bridge receives the single coin pool message to complete the type conversion, it will call a different cross-chain message protocol interface to complete the message transmission.

2.1.3 Single Coin Pool Manager (PoolManage)

Sui Clearing House's Pool Manager is a unified manager of single coin pools on different chains, responsible for asset classification and asset liquidity management of single coin pools. The Pool Manager provides a global view of the asset distribution of the pools on different chains. The Single Coin Pool Manager has a dynamic cross-chain fee algorithm that incentivizes the automatic rebalancing of single coin pool popularity by the available and desired liquidity on different chains. If the available liquidity of a single coin pool is lower than the desired liquidity, the cross-chain fee becomes higher. If the available liquidity of a single coin pool is higher than the desired liquidity, the cross-chain fee becomes lower.

2.1.4 AppManage

The application manager of Sui Clearing House is responsible for managing the applications supported by the protocol.When Omni Protocol introduces a new application, it needs to obtain permission and assign an application ID through the application manager.Each application in the protocol has a unique ID.Users can select the application they want to use by the application ID to provide liquidity to the protocol, or they can select it automatically based on the protocol recommendation algorithm, allowing Users have low learning cost and good user experience. The application manager is responsible for freezing and taking down applications for emergency situations. The Application Manager is responsible for developing or updating the message repository used by different public chain application layer interfaces.

2.1.5 Governance

Governance is one of the most important modules for the sustainability of the protocol. Through decentralized on-chain governance, the protocol can be continuously upgraded and optimized, and bring more benefits to the protocol participants. The governance module adopts the idea of DAOStack, which is co-governed on-chain through governance tokens and reputation system, and the governance scope includes the whole protocol. Early on, the development team will govern by proxy, and after the governance token is issued, it will gradually transition to decentralized governance and eventually be governed entirely by the community. The rules for the issuance of governance tokens will be determined in the near future.

Permissions management involves all permissions for the entire protocol. Permissions related to the single pool manager, which requires the single pool manager to be governed in order to introduce new assets and change the parameters of the dynamic cross-chain fee algorithm. Permissions related to the application manager, which requires the application manager to introduce new applications and freeze applications through governance. Through on-chain governance, the governance token holders can jointly develop and improve the protocol ecology.

2.2 Application layer

2.2.1 Full-chain lending

The lending protocol, represented by Aave, provides the ability to lend to a few of the most liquid tokens. In a decentralized financial world, Aave offers the good features of no trust and no permission. While decentralized lending has proven to work well over time, there is a large unmet need for current lending protocols. aave only meets the ability to lend to the most liquid assets and needs to be further extended for other assets. aave is a single-chain model. Although cross-chain collateralized lending is achieved through a token bridge, it requires each public chain to deploy the Aave protocol, which is costly and complex for users to operate.

Interest Rate Model

Reserves: In lending agreements, in rare cases the value of a borrower's collateral may be less than the value of its liabilities, in which case the borrower's ability to repay is referred to as insolvency. When an insolvent borrower is liquidated, there will be remaining liabilities that are considered to be bad debt. If bad debts appear to accumulate, lenders may all come at once to withdraw funds to avoid becoming a bad debtor. To mitigate this risk, follow Compound's practice of agreeing to draw down a portion of the interest accumulated into a reserve. If bad debts arise and the reserve is used to make repayments, the lender can avoid becoming the bearer of bad debts as long as the reserve accumulates faster than the bad debts. The percentage of interest withdrawn to become a reserve is called the reserve factor and is noted as RF. Different assets have different RF**.** The RF takes on a value between 0 and 1 and can be adjusted for trade-offs through governance.

Funding Utilization: Funding Utilization is the ratio of current borrowed funds to supplied funds. U_c represents funding utilization, D_c represents borrowed funds, and L_c represents remaining funds.

$$U_c = \frac{D_c}{D_c + L_c}$$

Borrowing rate: Aave and Compound use a static linear interest rate model to determine the borrowing cost of the agreement. Simply put, when the demand for borrowing from the pool increases or the supply decreases, the interest rate increases, while when the supply increases or the demand for borrowing decreases, the interest rate decreases. BR_b represents the base borrowing rate. $U_{optimal}$ represents the optimal funding utilization rate. BR_{slope1} represents the constant of the ratio of interest rate to utilization rate when the current funding utilization rate is lower than the optimal funding utilization rate. BR_{slope2} represents the constant of the ratio of interest rate to utilization rate when the current funding utilization rate is higher than the optimal funding utilization rate. BR_c represents the current borrowing rate.

$$BR_c = \begin{cases} BR_b + U_c * BR_{slope1} \quad U_c < U_{optimal} \\ \\ BR_b + BR_{slope1} + \frac{U_c - U_{optimal}}{1 - U_{optimal}} * BR_{slope2} \;\; U_c >= U_{optimal} \end{cases}$$

Liquidity Rate: The liquidity rate is the interest rate at which a lender should receive interest for providing a loan, funded by the borrowing rate, expressed as LR_c.

$$LR_c = BR_c * U_c * (1 - RF)$$

Cumulative Debit Index: indicates the cumulative index of interest payable by the borrower as time accumulates. ΔT represents the interval between the current time and the last update, and T_{year} represents the time of a year in seconds.

$$BL_t = (\frac{BR_c}{T_{year}} + 1)^{\Delta T} * BL_{t-1},\ BL_0 = 1$$

Cumulative Liquidity Index: Indicates the cumulative index of interest due to the lender as time accumulates. ΔT represents the interval between the current time and the last update, and T_{year} represents the time of a year in seconds.

$$CL_t = (\frac{LR_c * \Delta T}{T_{year}} + 1) * CL_{t-1},\ CL_0 = 1$$

The interest rate model refers to the current established practice of Aave and Compound for interest rate modeling, with the difference that the interest earned on the reserve factor RF becomes a special borrower that is added to the accumulated liquidity index thereby automatically earning interest and further increasing the ability of the agreement to withstand bad debt risk.

Rights-based tokens (oToken)

The entitlement token oToken is a derivative token received by the depositor after making a deposit. The entitlement token oToken is automatically accumulated as CL_t increases, and the increase represents the interest received by the depositing user. SoT_t is used to indicate the number of scaled oToken the user has at moment t. m is used to represent the number of tokens with m deposited/withdrawn by the user. oT_t is used to denote the number of oTokens the user has at moment t. The number of entitled tokens oToken is determined by both SoT_t and CL_t, as follows.

User Deposit：

$$SoT_t = SoT_{t-1} + \frac{m}{CL_t}$$
$$oT_t = SoT_t * CL_t$$

User Withdraw：

$$SoT_t = SoT_{t-1} - \frac{m}{CL_t}$$
$$oT_t = SoT_t * CL_t$$

Debt-enabled token (dToken)

The debtification token dToken is the debt incurred by the borrower to borrow money. The debtification token dToken automatically accumulates as BL_t increases, and the increase represents the amount of interest the borrowing user will pay. SdT_t is used to indicate the number of scaled dTokens a user has at time t. m is used to denote the number of tokens the user borrowed/repaid in m. dT_t is used to denote the number of oToken the user has at moment . The number of debtified tokens dToken is determined by both SdT_t and BL_t, as follows.

User Borrow：

$$SdT_t = SdT_{t-1} + \frac{m}{BL_t}$$
$$dT_t = SdT_t * BL_t$$

User Repay：

$$SdT_t = SdT_{t-1} - \frac{m}{BL_t}$$
$$dT_t = SdT_t * BL_t$$

Liquidation

Liquidation is when the value of a user's liabilities and collateral are less than the agreed upon over-collateralization requirements, and the user's collateral is liquidated to pay off the user's liabilities.

Risk Adjustment: Unlike traditional lending, which only considers the risk of bad debts due to a decrease in the value of the user's collateral. The agreement also takes into account the risk associated with an increase in the value of the liability, which is offset by a BF(greater than 1) increase in the value of the liability. CF(less than 1) is used to reduce the value of the collateral. When the user's collateral value $TotalCollateral$ and liability value $TotalDebt$ do not satisfy the following formula, they will be liquidated.

$$TotalCollateralCF >= TotalDebtBF$$

Resistance to MEV: In Aave and Compound, the incentive for liquidation is to offer the borrower's collateral to the liquidator at a fixed percentage discount, usually between 5% and 10%. Liquidators are profitable, but not resistant to MEV because miners and front runners can steal transactions from the liquidators. To limit this form of MEV, the protocol allows liquidity providers to qualify for discounts, miners and others do not.

Clearing costs: In Aave and Compound, clearing typically requires the use of external liquidity to process. The determination of this approach results in the liquidator not being able to liquidate at the desired price. Reasons for this include slippage, price fluctuations, fees, etc. The protocol's full-chain single coin pool allows clearing to allow clearing through lightning credits. The liquidator only needs to pay the Gas cost.

Soft Liquidation: In Aave and Compound, the liquidator's one-time liquidated debt is fixed, currently at 0.5, meaning that the liquidator is allowed to liquidate half of the debt at one time. The disadvantage of this approach is that it is excessive and unfair to liquidate half of the debt if a smaller liquidation would restore the debt to health. Therefore, this agreement will use a soft liquidation model that allows no more debt to be liquidated at a time than is necessary to restore the defaulters to health (plus an additional safety margin). This means that less than half of the debt will be liquidated for borrowers in minor default and more than half for borrowers in serious default.

Asset Segregation

To meet the demand for lending on long-tail assets, for long-tail assets users are allowed to lend in segregated mode. Unlike regular assets, where users must enter segregated mode, long-tail assets can only be used as collateral, can only be borrowed against stablecoin assets and have a debt cap. The debt cap allows for good control of risk, thus avoiding a sharp increase in bad debt due to price volatility of long-tail assets.

Overall, the full-chain lending application builds on the proven practices of the Aave and Compuond protocols, while improving on their shortcomings. Compared with Aave and Compuond: for users, it enables one-click deposit on chain A and lending operation on chain B, reducing the learning cost for users. For the clearer, it reduces the clearing stale cost and reduces MEV. for the protocol, it improves the risk resistance.

2.2.2 PMM

PMM is the active market maker algorithm, derived from the DODO protocol. Compared to traditional AMM, PMM allows users to make unilateral withdrawals, which fits very well with the full-chain single-coin pool of our protocol. By building PMM on top of the full-chain single-coin pool to form a new Dex application, users will have lower slippage and market makers will have lower unearned losses. Inevitably, PMM introduces higher algorithmic complexity, but the algorithmic logic built on top of Sui will effectively reduce additional fee consumption.

Marginal Price

The marginal price P is the instantaneous price in the current state and is used to indicate how many quote tokens can buy a base token. B_0 denotes the total base token recharge of the market maker and Q_0 denotes the total quote token recharge of the market maker. B denotes the total number of base tokens in the current asset pool and Q denotes the total number of quote tokens in the current asset pool. i is the market price provided by the prediction machine and k is a parameter in the range of 0 to 1.

$$P = \begin{cases} i * (1 - k + k(\frac{B_0}{B})^2) \quad B < B_0 \\ \\ \frac{i}{(1-k+k(\frac{Q_0}{Q})^2)} \quad Q < Q_0 \end{cases}$$

As can be seen from the marginal price formula, when k is 0, the marginal price is constantly equal to the market price offered by the prophecy machine, with no slippage and high capital utilization. k is 1, degenerating to the traditional AMM, where both assets must be charged in proportion to the current price, with high slippage and low capital utilization. When the number of assets B and Q in the pool deviates from B_0 and Q_0, it causes the current price to be higher and lower than the external price, prompting arbitrageurs to arbitrage, causing B and Q to return to the target quantities B_0 and Q_0. It is worth noting that when the system is in disequilibrium, price changes in the prognosticator can lead to profits or losses. For example, when there is a shortage of base token and the price of the prophecy machine base token increases, the excess quote token value is lower than the value of the base token returning to equilibrium, and the market maker will incur a loss. Therefore, for base token and quote token with high marginal price fluctuations, a larger k needs to be set to reduce the risk of market makers incurring losses.

Average price

The average price P is obtained by integrating the marginal price with the following formula, which gives the number of tokens a trader needs to pay to buy or sell a certain number of base tokens and quote tokens.

$$P = \frac{Q_1 - Q_2}{B_2 - B_1} = i * (1 - k + k * \frac{B_0^2}{B_1 * B_2}) = \frac{i}{1 - k + k * \frac{Q_0^2}{Q_1 * Q_2}}$$

Regression target

B_0 and Q_0 are the regression targets, and substituting B_0 and Q_0 into the average price equation and solving the quadratic equation yields

$$B_0 = B_1 + B_1 * \frac{\sqrt{1 + \frac{4*k*\Delta Q}{B_1^2*i}} - 1}{2*k}$$
$$Q_0 = Q_1 + Q_1 * \frac{\sqrt{1 + \frac{4*k*\Delta B*i}{Q_1^2}} - 1}{2*k}$$

Market makers top up base token when B_1 rises b and B_0 rises even more, so once a market maker fills up, it will cause all market makers to make a profit and the protocol will provide a top up to reward the market maker, the reward is mainly paid by the trader who made the system deviate from equilibrium. Conversely, a withdrawal by a market maker will cause all market makers to suffer a loss, and therefore a fee will be paid for the withdrawal. The handling fee is equal to the sum of the losses of the market makers caused by this withdrawal and is distributed to the market makers who have not yet withdrawn their money.

Overall, the new Dex application uses the PMM algorithm to allow users to top up assets on any public chain for unilateral market making. Also the flexible parameter configuration leads to lower slippage, less impermanent losses, and a better user experience.

2.2.3 NFT borrowing and lending

NFT lending, refers to borrowers going to lend cryptocurrency by using their NFT as collateral on the platform, with the lender or platform providing liquidity for the borrower. there are currently three main models for NFT lending, which are peer-to-peer, peer-to-peer pool and collateralized debt position. In the peer-to-peer NFT lending model, borrowers and lenders are matched directly, with borrowers applying for loans for a certain period of time and lenders offering different interest rate terms, with the borrower ultimately choosing the interest rate terms to complete the lending; in the peer-to-peer pool NFT lending model, users can pledge their NFT portfolios and receive other tokens such as USDT at variable interest rates, which is similar to Compound lending; in the collateralized debt position NTF lending model, users can mint stable coins after pledging a class of NFT collateral to create a vault, this model is derived from MakerDAO and is implemented in basically the same way. Currently NFT lending, NFTfi, BendDAO and X2Y2 occupy the vast majority of the market. Among them, NFTfi and X2Y2 are peer-to-peer lending models, and BendDAO is a peer-to-pool lending model. Peer-to-peer and peer-to-pool lending are derived from homogenized token lending and have many similarities. The difference lies in the uniqueness of NFTs, where the value of NFTs of the same type is different and more complex usage scenarios need to be considered.

Peer-to-Peer Model

Peer-to-peer model is for the NFT listed on the platform, the borrower to set the amount, term, APR, etc. on their own, after the lender and borrower agree, to conclude the transaction. The advantage of the peer-to-peer model is that it supports low liquidity assets and long-tail assets, while the disadvantage is that it has low capital utilization.

Peer-to-Pool Model

The peer-to-peer pool model is where the borrower borrows from a pool and the lender deposits funds into the pool in advance to receive the proceeds. There are three main entities in the peer-to-peer pool model: the borrower, the platform and the lender, with the platform acting as a lending reservoir and as a counterparty to the borrower and the lender. As the borrower, the appropriate NFTs are bundled into a unique NFT through the platform as a single unit of collateral and unavailable for any interaction. The cryptocurrency is then lent from the lending pool based on the floor price of the underlying and the collateral ratio set by the platform; the lender earns interest by depositing the cryptocurrency into the lending pool to provide liquidity. Liquidation is triggered when the borrower's health factor falls below 1 due to a significant drop in the floor price of the underlying. The advantages of peer-to-peer pooling are time efficiency and risk diversification through pooling, while the disadvantages are that it is not conducive to low liquidity assets and long-tail assets, and also suffers from poor capital utilization.

Omni Protocol will support both peer-to-peer model and peer-to-pool model for full-chain NFT lending. By fusing the advantages of both models, users will have more options to support long-tail assets while bringing higher time efficiency.

2.2.4 OmniSwap

OmniSwap is a one-click cross-chain Swap by aggregating liquidity across public chains. OmniSwap brings users a better Swap experience by effectively combining Dex and token bridges from different public chains to find the best route. At present, OmniSwap is already online to support EVM public chain led by Ether and Aptos public chain, using Stargate and Wormhole token bridges to combine Dex of different public chains to achieve one-click Swap.

The current OmniSwap is an effective improvement to the single-chain Swap. However, fusing OmniSwap into Omni Protocol can have higher capital efficiency. By using the Dex application built with PMM, most of the cross-chain Swap is done within the protocol, and the advantage of PMM algorithm can be effectively used to bring lower slippage and lower gas operating experience to users.

2.2.5 Other Projects

The Web3 world is always evolving and changing day by day. Financial and non-financial Web3 applications are popping up all over the place. Omni Protocol will leave enough room for expansion to meet the interfacing needs of these Web3 projects. By introducing these Web3 applications into Omni Protocol, project parties can effectively access the user and capital liquidity of each public chain and promote the development of Web3 projects. Users have more choices and discover more value and fun.

2.3 Message Layer

Application messages

Application messages are independent for different applications and need to be encoded by the application layer to be passed to the liquidity pool. Application messages are decoded and encoded by different application-defined messages, and with Sui protocol core to make application messages go to their respective corresponding settlement logic.

Liquidity pool messages

Liquidity pool messages mainly contain asset-related user addresses, asset addresses, asset quantities, etc. The liquidity pool messages are encoded in a uniform format on top of the application messages and eventually go into the bridge. This approach is to unify the management of the protocol assets and secure the protocol funds.

Cross-chain protocol messages

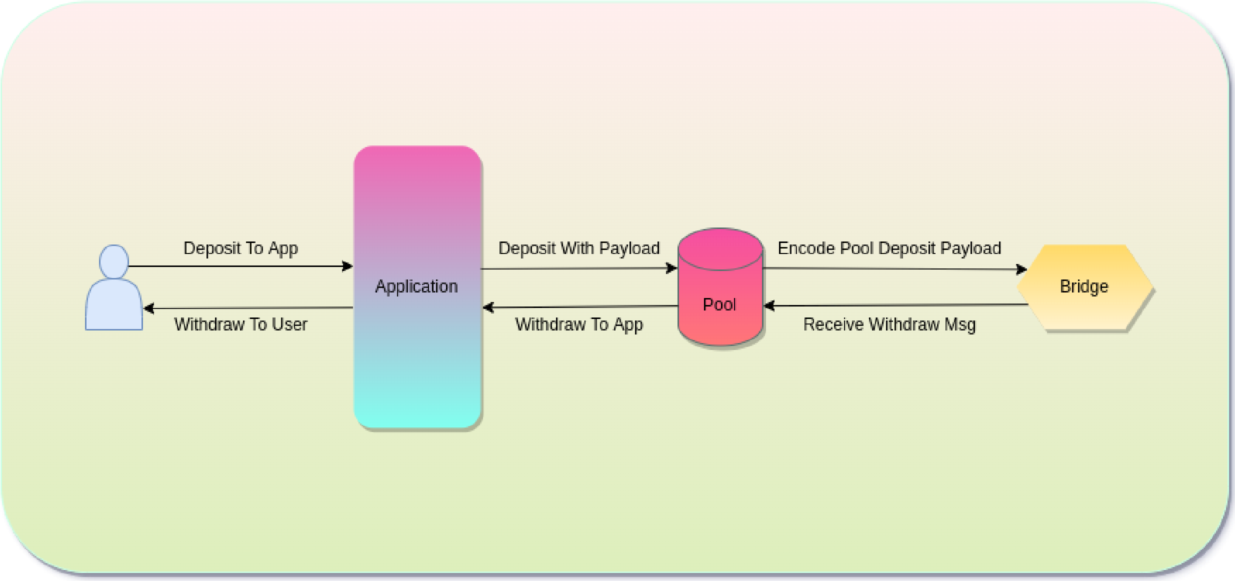
There are different cross-chain protocol messages for different cross-chain messaging protocols. Cross-chain protocol messages are the last layer of the messaging layer and contain application messages and liquidity pool messages, which are packaged and eventually propagated between bridges via cross-chain messaging protocols.

3 Protocol Contract

There are three parts to the protocol contract, which are the single coin pool, the message protocol bridge, and the protocol core.

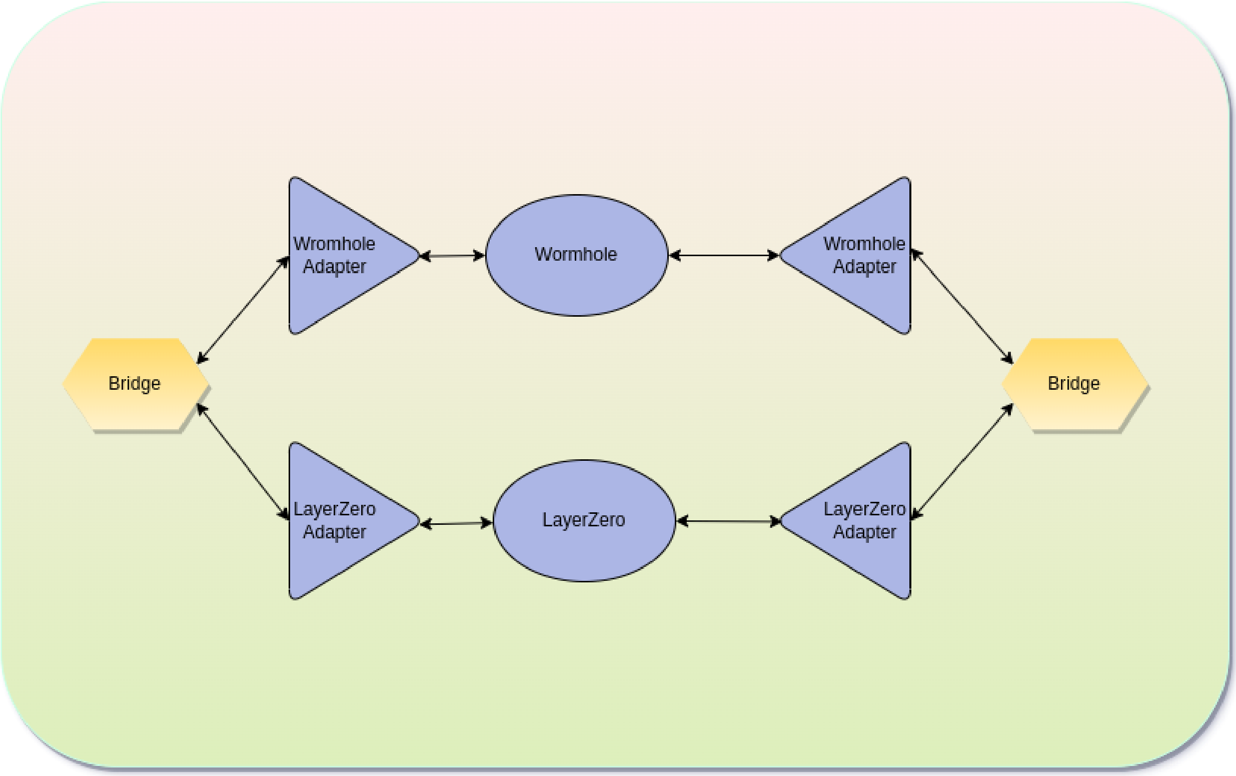
3.1 Single Coin Pool Contract

The single coin pool needs to provide an external interface for recharging, and an interface for withdrawals to the messaging protocol bridge, and to package messages from the application before sending them to the bridge.



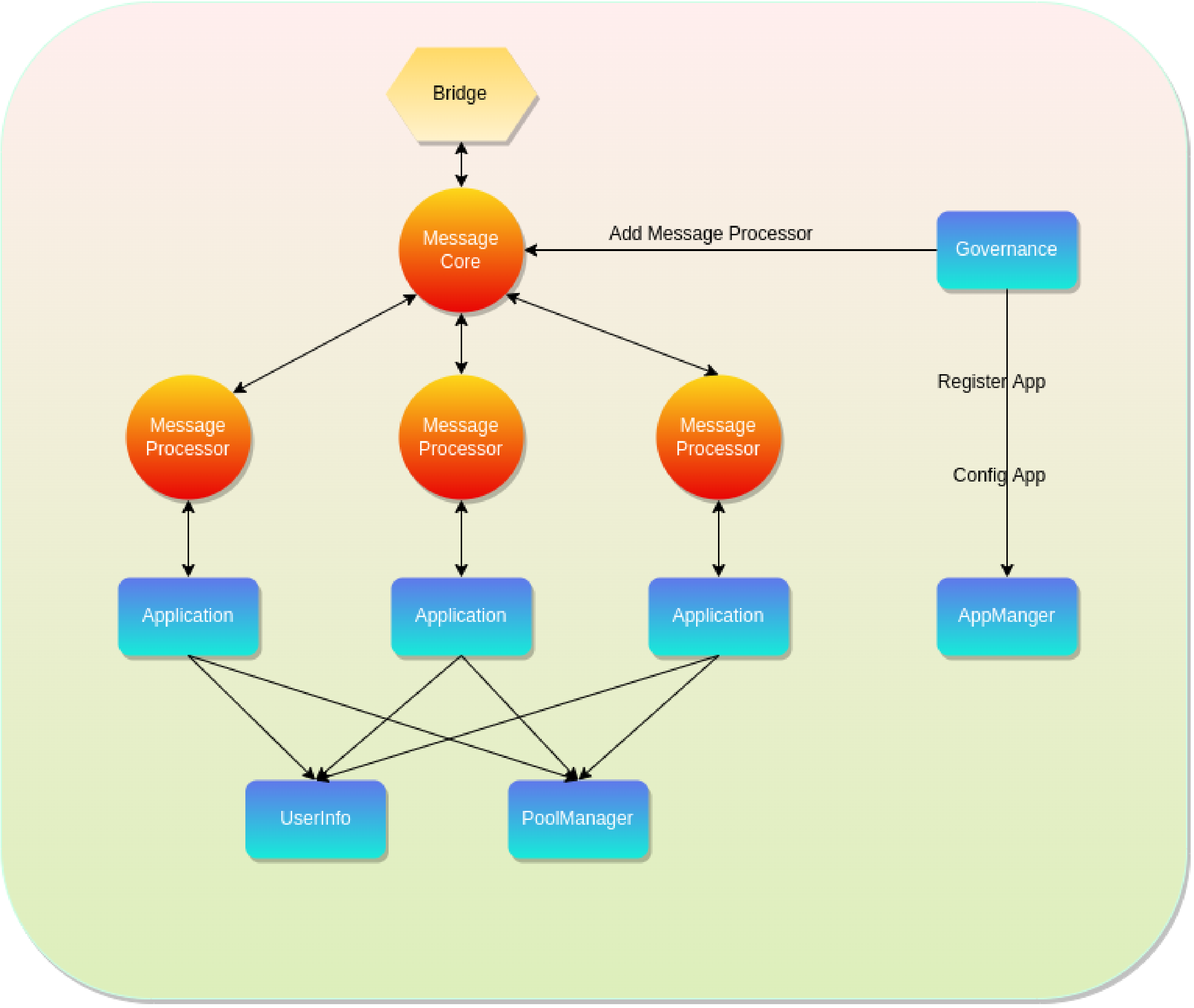
3.2 Messaging Protocol Bridge Contract

Message protocol bridges need to be compatible according to the accessed message protocols and are only responsible for the delivery of messages.



3.3 Protocol Core Contract

The protocol core is divided into 3 parts, which are message processing, application logic and governance. The first part is message processing, which is mainly responsible for decoding and distributing messages to the corresponding application logic contracts, and also responsible for coding messages from different applications to the message bridge; the second part is application logic, each application logic corresponds to the corresponding application messages, and updates the protocol user information and single coin pool accordingly; the third part is governance, which needs to control and manage the protocol global The third part is governance, which needs to control and manage the permissions of the protocol globally, including the authorization of Bridge, the addition of message types and the setting of application permissions.



4 Looking Ahead

The Chainwide Liquidity Protocol is an extensible protocol designed to aggregate liquidity from all chains and make it available to applications. Applications can use liquidity from any chain through the protocol. For developers, building applications on the upper layer of the protocol allows developers to spend more energy on application innovation and no longer need to focus on the chain and the lack of liquidity, which greatly reduces the threshold for developers and gives more opportunities for individual developers; for liquidity providers who want to earn coins, they have a unified portal for providing liquidity and no longer need to go to learn for each application, which greatly reduces This greatly reduces the learning cost for users and lowers the threshold for adding liquidity. For users, they have a chain-wide future because protocol-based applications are chain-wide and can be used to interact with applications on any chain without the need to exchange tokens on a chain to be able to use applications on that chain. I believe that the future of Web3 users must not be difficult to use Web3 applications, only need to be on an application should be able to operate most of the DeFi applications, and eventually do Web2 users are Web3 users. The future of the full-chain mobility protocol is the core of Web3, where most of Web3's money and applications are gathered, and is the foundation of the Web3 financial system.

5 Summary

The Full Chain Liquidity Protocol provides a way to aggregate liquidity for applications, while also giving applications the opportunity to use Full Chain Liquidity. Applications built on the protocol will naturally aggregate liquidity and will be able to use full-chain liquidity in a natural way. Current DeFi applications are still mainly developed on their own chains, and even if there is cross-chain, the effect is not very big, and the liquidity is mainly aggregated on the original chain. And it is difficult for each team to develop on other chains, both in terms of development and resources, because there is no unified protocol to support them. If we compare DeFi applications to companies and chains to countries, then the whole-chain liquidity protocol is the basis for economic globalization, and applications can better develop on the whole chain through our whole-chain liquidity protocol instead of being stuck on one chain. The Full-Chain Liquidity Protocol unifies the liquidity pool through an appropriate design, provides chain-wide operational convenience for various DeFi applications such as lending and borrowing, and solves the current dilemma of silos between chains. It has taken a solid step in the all-chain era and laid a good foundation for the development of the all-chain era.