



The Current Landscape of DeFi Lending/Borrowing

The most popular collateralized lending/borrowing protocols today include Compound, Aave, and MakerDAO. These protocols offer a floating interest rate via a pool which users can supply or borrow from. While these protocols solve a variety of use cases, there is increasingly more demand both from suppliers and borrowers for more predictable, fixed interest rates.

We can make another nuanced observation. We can think of interest rate as a product that can be traded. After all, if your view is that DeFi interest rates will go up, you can borrow funds from one party at 2% and then turn around and lend those funds to another party at 3%, effectively locking in a profit of 1%. Typically, in the current generation of lending/borrowing protocols, the supply and borrow rates are set algorithmically as a function of the amount of liquidity in the pool. By allowing lenders and borrowers to instead set their own rates, can we promote the trading of fixed interest rates?

Enter Qoda: The Craigslist for borrowing/lending

Qoda aims to be a decentralized over-the-counter hub for users to post listings (called “Quotes”) for collateralized lending/borrowing at fixed interest rates for fixed periods.

Before moving on to explain the protocol in detail, some quick terminology first:

Quoters publish “Quotes” either on- or off-chain to indicate their intention to lend or borrow a sum of money at a fixed interest rate for a specific period of time.

Responders can search for published Quotes that they want to accept. Once accepted, the Quoter and Responder become opposite counterparties in a Fixed Rate Loan agreement.

Brokers are hubs for matching Quoters and Responders, and serve as intermediaries to Qoda’s smart contracts. They are free to list Quotes either on-chain or off-chain. Any user can serve as a Broker by implementing the necessary infrastructure, and be rewarded for every Fixed Rate Loan agreement they originate.

Liquidators scan the blockchain for existing fixed rate loan contracts. If any borrowers are in danger of being undercollateralized, their account can be liquidated for a reward to the liquidator.

At its core, Qoda’s smart contracts manage the loan execution and settlement, margining, liquidation, and interest accrual calculations. This is further supported by the Qoda Broker dApp, a UI which facilitates users trading with one another completely on-chain. All Quotes on the Broker dApp will be posted on-chain as events (as opposed to storage, saving gas) so that the platform remains provably decentralized, with no user’s data reliant on a centralized backend storage.

Importantly, Qoda implements the following key features:

- 1) No centralized custody of user funds
- 2) Off-Chain Quoting
- 3) On-Chain loan contract settlement
- 4) Portfolio Margining

Let's go through each of these features.

No Centralized Custody of User Funds

When a Responder has responded to a Quote, a Fixed Rate Loan agreement is instantiated. The smart contract will automatically handle transfers of the loan and collateral, as well as calculating the accrual of fixed interest over time. The Qoda smart contract takes the collateral as escrow, which can only be released back to the borrower when they've repaid the loan. Or released to Liquidators in the case of undercollateralized accounts.

Off-Chain Quoting

Suppose a lender wants to put out a listing to lend their assets. The basic structure of their quote looks like this:

```
Quote {  
    address principalTokenAddress,  
    address quoter,  
    uint8 side,  
    uint256 quoteExpiryBlock,  
    uint256 endBlock,  
    uint256 principal,  
    uint256 principalPlusInterest,  
    uint256 nonce,  
    bytes signature  
}
```

principalTokenAddress – The contract address of the token which the loan will be denominated in.

quoter – The public address of the Quoter

side – The side of the Quoter: 0 if Quoter is borrowing, 1 if Quoter is lending

quoteExpiryBlock – The block number after which a published Quote is no longer valid

endBlock – The block number at which the Fixed Rate Loan contract expires

principal – The initial size of the loan

principalPlusInterest – The final amount to be returned to the lender

nonce – Value to guarantee uniqueness of the quote

Finally, the most interesting field is the **signature**. The Quoter hashes all the other fields of the Quote and signs that hash, generating the signature. The signature trustlessly proves that the Quoter is in fact willing to enter the loan at the specified terms. Note that at this point, the Quote itself can be fully instantiated off-chain. Theoretically, a lender and borrower can come to an agreement on the terms of the loan completely off-chain whether via email, a Discord chat, some white-label exchange, or anywhere else. All they need to do is generate the signature, and reference that when submitting the trade. This feature provides gas savings for users who may not

want to post their Quotes on the blockchain, or for those who want to deal privately without informing the public about the loan details until the trade is confirmed.

On-chain Contract Settlement

When a Responder who is browsing through the list of Quotes, sees a loan deal which they want to enter, how do they enter the trade? The Responder inputs the details of the Quote along with the Quoter's signature, sending it as a transaction to the smart contract. The protocol will match the Quote details against the signature to verify that all the details of the loan are correct. If the hash of the inputs and the given signature do not match, the trade submission will fail and be reverted.

Portfolio Margining

In this section, we'll give a high-level overview of the portfolio margining system employed by Qoda. Before a user is allowed to borrow funds, they will need to first deposit assets as collateral into the Qoda smart contracts.

For every account, the margin is composed of three components:

depositValue: This is the sum of all the deposits of the account across all assets, expressed in USD via Chainlink oracles. For each asset, there will be a value from 0.0 to 1.0 called the risk factor. The risk factor serves to discount the value of the deposit of any particular asset and should be a function of the riskiness of that asset. Example: the deposit value for asset xyz would be computed as (amount in xyz) * (oracle price of usd/xyz) * (risk factor)

borrowValue: This is the sum of all the borrows of the account across all assets, expressed in USD via Chainlink oracles. There is no risk factor defined here, it is implicitly equal to 1.

lendCredits: Credits get applied to account if they are also lending out assets. Specifically, lendCredits can be non-zero if and only if the account has a borrow of that same asset at the exact same maturity, and it is capped at the amount of the borrow. For example: If an account borrows 100,000 USDC and lends 150,000 USDC, both maturing on 30th September, the user will receive 100,000 USDC of lendCredits. However, if the borrow matures on 30th September, but the lend matures on 31st December, the lendCredits would be zero.

These three components are used to calculate the **accountLiquidity**, an overall picture of the health of a particular account. It is computed as follows:

$$\text{accountLiquidity} = (\text{depositValue}) / (\text{borrowValue} - \text{lendCredits})$$

accountLiquidity must be above 1.0 at all times or else the account is in danger of being liquidated.

The advantage of the lendCredits system is that it allows the interest rate market to be tradable. Specifically, traders can speculate on interest rate rising or falling by entering and exiting borrows/lends without having to post new collateral every single time, eating up their capital. At the moment, this feature is possible only when the lends and borrows are on the same maturity date, but future iterations could potentially yield solutions for cross-curve netting of margin.

While in theory, Qoda allows the flexibility for loans to expire at any maturity date, in practice the frontend UI will encourage trading of “liquid dates”, such as quarterly expiries, to herd liquidity for maximum efficiency.

Roadmap

Qoda transforms the lending/borrowing scene into an open, decentralized fixed interest rate market, where users can set what rates they wish to lend and borrow at. The fixed interest rate allows users of the platform to generate more stable income compared to existing floating rate interest solutions. The platform can be extended in the future to offer floating rate loans referencing sources like Compound/Aave mid rates or perpetual futures funding rates.

The ultimate goal is to promote trading around interest rates, eventually encouraging traders to market-make the product, offering two-sided Quotes where they are willing to borrow at a lower rate and lend at a higher rate. Qoda also plants the seed for developing a tradable oracle for the interest rate curve, a key step in fostering the growth and maturity of the DeFi fixed income market as a whole.

Qoda’s mechanism for off-chain Quote generation also lends itself particularly well to over-the-counter derivative products, such as options or fixed-to-floating interest rate swaps. This can be an area for potential growth in the future.

Another interesting avenue for potential future growth is in origination of “corporate bonds”. Suppose a large crypto project wants to raise funds but does not want to give up further token equity. Instead of raising funds via selling equity, they can issue debt in the form of a decentralized corporate bond with enticing fixed-rate yields (perhaps sweetening the pot with minting NFTs to bondholders).