Compound:
The Money Market Protocol

Version 0.4
June 2018

Authors
Robert Leshner, Geoffrey Hayes
https://compound.finance

Abstract
In this paper we introduce a decentralized protocol which establishes money markets with algorithmically set interest rates based on supply and demand, allowing users to frictionlessly exchange the time value of Ethereum assets.

Contents

1 Introduction

The market for cryptocurrencies and digital blockchain assets has developed into a vibrant ecosystem of investors, speculators, and traders, exchanging thousands [1] of blockchain assets. Unfortunately, the sophistication of financial markets hasn't followed: participants have little capability of trading the *time value* of assets.

Interest rates fill the gap between people with surplus assets they can't use, and people without assets (that have a productive or investment use); trading the time value of assets benefits both parties, and creates non-zero-sum wealth. For blockchain assets, two major flaws exist today:

- Borrowing mechanisms are extremely limited, which contributes to mispriced assets (e.g. "scamcoins" with unfathomable valuations, because there's no way to short them).
- Blockchain assets have negative yield, resulting from significant storage costs and risks (both on-exchange and off-exchange), without natural interest rates to offset those costs. This contributes to volatility, as holding is disincentivized.

Centralized exchanges (including Bitfinex, Poloniex...) allow customers to trade blockchain assets on margin, with "borrowing markets" built into the exchange. These are trust-based systems (you have to trust that the exchange won't get hacked, abscond with your assets, or incorrectly close out your position), are limited to certain customer groups, and limited to a small number of (the most mainstream) assets. Finally, balances and positions are virtual; you can't move a position on-chain, for example to use borrowed Ether or tokens in a smart contract or ICO, making these facilities inaccessible to dApps [2].

Peer to peer protocols (including ETHLend, Ripio, Lendroid, dYdX...) facilitate collateralized and uncollateralized loans between market participants directly. Unfortunately, decentralization forces significant costs and frictions onto users; in every protocol reviewed, lenders are required to post, manage, and (in the event of collateralized loans) supervise loan offers and active loans, and loan fulfillment is often slow & asynchronous (loans have to be funded, which takes time) [3-6].

In this paper, we introduce a decentralized system for the frictionless borrowing of Ethereum tokens without the flaws of existing approaches, enabling proper money markets to function, and creating a safe positive-yield approach to storing assets.

2 The Compound Protocol

Compound is a protocol on the Ethereum blockchain that establishes money markets, which are pools of tokens with algorithmically derived interest rates, based on the supply and demand for the token. Suppliers (and borrowers) of an asset interact directly with the protocol, earning (and paying) a floating interest rate, without having to negotiate terms such as maturity, interest rate, or collateral with a peer or counterparty.

Each money market is unique to an ERC-20 token (such as tokenized Ether, a stablecoin such as Dai, or a utility token such as Golem), and contains a transparent and publicly-inspectable balance sheet, with a record of all transactions and historical interest rates.

2.1 Supplying Tokens

Unlike an exchange or peer-to-peer platform, where a user's tokens are matched and lent to another user, the Compound protocol aggregates the supply of each user; when a user supplies a token, it becomes a fungible resource. This approach offers complete liquidity; users can withdraw their tokens at any time, without waiting for a specific loan to mature.

Balances in a money market accrue interest based on the supply interest rate unique to that asset. Users can view their balances (including accrued interest) in real-time; when the user makes a transaction that updates his or her balance (supplying, transferring, or withdrawing a token), accrued interest is converted into principal and paid to the user.

2.1.1 Primary Use Cases

Individuals with long-term investments in Ether and tokens ("HODLers") can use a Compound money market as a source of additional returns on their investment. For example, a user that owns OmiseGo can supply their tokens to the Compound protocol, and earn interest (denominated in OmiseGo) without having to manage their asset, fulfill loan requests or take speculative risks.

dApps, machines, and exchanges with token balances can use the Compound protocol as a source of monetization and incremental returns by "sweeping" balances; this has the potential to unlock entirely new business models for the Ethereum ecosystem.

2.2 Borrowing Tokens

Compound allows users to frictionlessly borrow from the protocol, using collateralized lines of credit, for use anywhere in the Ethereum ecosystem. Unlike peer-to-peer protocols, borrowing from Compound simply requires a user to specify a desired asset; there are no terms to negotiate, maturity dates, or funding periods; borrowing is instant and predictable. Similar to supplying an asset, each money market has a floating interest rate, set by market forces, which determines the borrowing cost for each asset.

We enforce a rule that each account must have a balance that more than covers the outstanding borrowed amount; we call this a ***collateral ratio***, and an account can take no action (e.g. borrow or withdraw) that would bring its value below our desired ratio; to increase (or reset) the collateral ratio, users are free to repay a borrowed asset in whole or in part, at any time. Balances held in Compound, even while being used as collateral, continue to accrue interest normally.

2.2.1 Risk & Liquidation

If the value of a user's supplied assets divided by the value of their outstanding borrowing declines below the collateral ratio, the user's collateral becomes available to purchase (with the borrowed asset) at the current market price minus a ***liquidation discount***; this incentivizes an ecosystem of arbitrageurs to quickly step in to reduce the borrower's exposure, and eliminate the protocol's risk.

Any Ethereum address that possesses the borrowed asset may invoke the liquidation function, in whole or in part, exchanging their asset for the borrower's collateral. As both users, both assets, and prices are all contained within the Compound protocol, liquidation is frictionless and does not rely on any outside systems or order-books.

2.2.2 Primary Use Cases

The ability to seamlessly hold new assets (without selling or rearranging a portfolio) gives new superpowers to dApp consumers, traders and developers:

- Without having to wait for an order to fill, or requiring off-chain behavior, dApps can borrow tokens to use in the Ethereum ecosystem, such as to purchase computing power on the Golem network
- Traders can finance new ICO investments by borrowing Ether, using their existing portfolio as collateral
- Traders looking to short a token can borrow it, send it to an exchange and sell the token, profiting from declines in overvalued tokens

2.3 Ledger System

Compound maintains a complete and auditable balance sheet and ledger, comprised of all transactions; for each money market ***a***:

$$Cash_a + Borrows_a = Supply_a + Equity_a$$

Each transaction generates two accounting entries (a debit and credit) using international accounting standards:

Event	Debit	Credit
Supply token	*Cash*	*Supply*
Withdraw token	*Supply*	*Cash*
Borrow token	*Borrows*	*Cash*
Repay borrow	*Cash*	*Borrows*

Liquidate (Borrower) Liquidate (Caller)	$Supply_{Collateral}$ $Cash_{Asset}$	$Borrows_{Asset}$ $Supply_{Collateral}$
Accrue Interest (Supply)	$Equity$	$Supply$
Accrue Interest (Loan)	$Borrows$	$Equity$

2.4 Interest Rate Model

Rather than individual suppliers or borrowers having to negotiate over terms and rates, the Compound protocol utilizes an interest rate model that achieves an efficient interest rate equilibrium, in each money market, based on the supply and demand of individual assets. The utilization ratio $\boldsymbol{U}$ for each money market $\boldsymbol{a}$ unifies supply and demand into a single variable:

$$U_a = Borrows_a \,/\, (Cash_a + Borrows_a)$$

Following economic theory, interest rates (the "price" of money) should increase as a function of demand; when demand is low, interest rates should be low, and vise versa when demand is high. The demand curve is codified through governance (and can be updated by the ***Chief Economist***) and is expressed as a function of utilization. As an example, borrowing interest rates may resemble the following:

$$Borrowing\ Interest\ Rate_a \;=\; 10\% + U_a * 30\%$$

For the protocol to be sustainable, and to withstand economic attacks (by both supplying and borrowing in the same money market), the total amount of interest earned by suppliers must be less than the total interest product by borrowers. The supply interest rate is a function of the borrowing interest rate, and includes a spread, $\boldsymbol{S}$ (such as 0.10), which represents the economic profit of the protocol:

$$Supply\ Interest\ Rate_a \;= Borrowing\ Interest\ Rate_a * U_a * (1 - S)$$

3 Implementation & Architecture

At its core, a Compound money market is a ledger that allows Ethereum accounts to supply or borrow tokens, while computing interest, a function of time. The protocol's smart contracts will be publicly accessible and completely free to use for machines, dApps and humans.

3.1 MoneyMarket Contract

A *MoneyMarket* contract stores each account's current balance. In its simplest form, the MoneyMarket acts similar to an ERC-20 token holding a balance for each account asset. Balances in the MoneyMarket accrue interest over time, and we describe a low-gas method to achieve these balance updates below.

As clients of the protocol supply or borrow assets, the MoneyMarket contract updates the balance sheet entries for the appropriate asset. Additionally, we use this balance sheet to calculate the current interest rates for supplying and borrowing tokens described in section 2.4 above.

3.2 Interest Rate Mechanics

Compound money markets are defined by a pair of prevailing interest rates (the supply and the borrowing rate), applied to all users uniformly, which adjust over time as the relationship between supply and demand changes.

The history of each interest rate, for each money market, is captured by an *Interest Rate Index*, which is calculated each time an interest rate changes, resulting from a user supplying, withdrawing, borrowing, repaying or liquidating the asset. A user's balance, including accrued interest, is simply the ratio of the current index divided by the index when the user's balance was last checkpointed.

The balance for each account address in the MoneyMarket is stored as an *account checkpoint*. An account checkpoint is a Solidity tuple *<uint256 balance, uint256 interestIndex>*. This tuple describes the balance at the time interest was last applied to that account. The current *Interest Rate Index* is also stored globally.

Each time a transaction occurs, both the supply and borrow Interest Rate Index for the asset are updated to compound the interest since the prior index, using the interest for the period, denominated by r * t, calculated using a per-block interest rate.

$$Index_{a,n} = Index_{a,(n-1)} * (1 + r * t)$$

3.3 Borrowing

A user who wishes to borrow and who has sufficient balances stored in Compound may call *Borrow(address asset, uint amount)* on the MoneyMarket contract. This function call checks the user's account value, and given sufficient collateral, will update the user's borrow balance, *transfer* the tokens to the user's Ethereum address, and update the money market's floating interest rates.

Borrows accrue interest in the exact same fashion as balance interest was calculated in section 3.2; a borrower has the right to repay an outstanding loan at any time, by calling `*repayBorrow(address asset, uint amount)*` which repays the borrow's principal and accrued interest.

If a user's overall account value ever falls below the *liquidation ratio* due to the value of his or her collateral changing (e.g. the user is holding ZRX as collateral to borrow Ether and ZRX significantly falls in value), then we have a public function `*liquidateMarketBorrow(address customer, address collateralAsset, address borrowAsset, uint collateralAmount)*` which exchanges the invoking user's asset for the borrower's collateral, at a slightly better than market price.

A *Price Oracle* maintains the current exchange rate of each supported asset; the Compound protocol delegates the ability to set the value of assets to a committee which pools prices from the top 10 exchanges. These exchange rates are used to determine borrowing capacity and collateral requirements, and for all functions which require calculating the value equivalent of an account.

3.4 Market Status

Each market (defined by an asset) can be in one of three states in the Compound money market: unsupported, supported or suspended. Every market will begin in unsupported and may transition (based on an action by the admin) to the supported state. Once a market is supported, any user may supply or borrow from the market as described above.

If for any reason, a market must be closed, the admin has the ability to move a supported market into the suspended state. The Compound governance committee will give users a significant lead time prior to suspending a market. A suspended market does not allow users to continue to supply or borrow assets (though users may still withdraw or close borrowed positions).

If a market is suspended indefinitely, it is important that all borrows are repaid; otherwise, the protocol will not have sufficient cash to repay all of the outstanding suppliers. For this reason, once a market is suspended, all borrows of that asset are available to be liquidated, at the standard discount, regardless of the borrower's collateral health.

3.5 Governance

Compound will begin with centralized control of the protocol (such as choosing the interest rate model per asset), and over time, will transition to the community and stakeholder control. The following rights in the protocol are control by the admin or governance committee:

- The ability to choose a new admin, such as a DAO
- The ability to set the interest rate model per market
- The ability to support, suspend or unsuspend a market
- The ability to delegate which entity may set oracle prices
- The ability to withdraw the equity (earned income) of the protocol

Over time, Compound expects to move these rights to a DAO controlled by the community. The DAO would be in the form of an Ethereum contract which would, by proposal and vote, call the functions associated with the abilities above. As the admin (or DAO) may choose a new admin, that DAO would have the ability to evolve over time based on the decisions of the stakeholders.

3.6 Contract Interface

Function ABI	Action
`supply(address asset, uint256 amount)`	Transfers an ERC-20 token into MoneyMarket and marks msg.sender's updated balance. *Updates Interest Rate Model*
`withdraw(address asset, uint256 amount)`	Transfers an ERC-20 to msg.sender if funds available. *Updates Interest Rate Model*
`borrow(address asset, uint amount)`	Checks msg.sender collateral value, and if sufficient, updates borrow balance and transfers an ERC-20 to msg.sender. *Updates Interest Rate Model*
`repayBorrow(address asset, uint amount)`	Transfers an ERC-20 token into MoneyMarket and reduces msg.sender's borrow balance. *Updates Interest Rate Model*
`liquidate(address customer, address collateralAsset, address borrowAsset, uint borrowAmount)`	Trues up borrower's interest, then transfers an ERC-20 token into MoneyMarket to repay the borrowed position. Msg.sender receives borrower's collateral at a discount. This function can only be called when a borrower breaches the collateral ratio. *Updates Interest Rate Model*
`setAssetPrice(address asset, uint256 value)`	Sets the price oracle price of a given asset. Called by the designated Oracle address.

Table 1. ABI and summary of select MoneyMarket functions on the Blockchain

4 Summary

- Compound creates properly functioning money markets for Ethereum assets
- Each money market has interest rates that are determined by the supply and demand of the underlying asset
- Users can supply tokens to a money market to earn interest, without trusting a central party
- Users can borrow a token (to use, sell, or re-lend) by using their balances in the protocol as collateral

References

[1] Cryptocurrency Market Capitalizations. https://coinmarketcap.com/
[2] Bitfixex Margin Funding Guide. https://support.bitfinex.com/
[3] ETHLend White Paper. https://github.com/ETHLend
[4] Ripio White Paper. https://ripiocredit.network/
[5] Lendroid White Paper. https://lendroid.com/
[6] dYdX White Paper. https://whitepaper.dydx.exchange/
[7] Fred Ehrsam: The Decentralized Business Model. https://blog.coinbase.com/