# Charkha Protocol

Charkha is a proof-of-concept protocol that establishes money markets on the Chainweb blockchain. It is directly inspired by the Compound Protocol which does the same for the Ethereum blockchain. Charkha is a teaching project; the rest of the white paper is written as though this is a real project, but you should not use its code or ideas directly.

Money markets are markets for short-term debt that enable suppliers to earn interest on their cash savings and borrowers to access cash for short-term uses like funding operations. In Charkha these markets are structured as pools of assets with algorithmically-derived interest rates for supplying and borrowing based on the supply and demand for the asset. Suppliers and borrowers interact with the protocol directly, earning and paying a floating interest rate.

Each money market is unique to a Chainweb asset. Supported assets are the native Kadena token, KDA, and two KIP-0005 tokens: KETH (a wrapped version of ETH on Chainweb) and CHRK, the Charkha rewards and governance token. Each money market records activity on-chain, including transactions, historical interest rates, and changes to the protocol.

The protocol calculates interest payments, updates the interest rate, and allocates rewards every time someone transacts with the protocol (e.g. supplies assets, withdraws assets, borrows assets, etc.). However, interest earned, interest owed, and rewards balances are not distributed because doing so every block would result in unacceptable fees. Instead, suppliers receive their interest when they withdraw their funds; borrowers pay their interest when they repay their loan; and rewards can be claimed at any time they exceed the gas fee required to transfer them.

The protocol centers on four activities: supplying assets, borrowing assets, liquidation, and rewards. The community can also modify the protocol by creating and voting on governance proposals using their CHRK holdings.

This white paper will describe each of these activities and the Charkha governance model in depth. The code in the Charkha directory of Real World Pact implements this white paper.

## 1. Interest Rate Model

Money markets pair suppliers and borrowers of an asset. Suppliers earn interest on their deposits, and borrowers pay interest on their loans. The primary function of the market is to set the "price" of money by determining the interest rates suppliers will earn and borrowers will pay. The interest rate model is based on supply and demand; interest rates rise as demand rises, and as interest rates rise, suppliers are incentivized to enter the market.

In Charkha, interest rates are algorithmically determined based on the utilization ratio for a market, $U_a$. The utilization ratio is based on the total amount of the asset being borrowed, $Borrows_a$, and the amount of the asset available for borrowing, $Cash_a$.

$U_a = Borrows_a / (Cash_a + Borrows_a)$

## 1.1 Borrower Interest Rates

The borrowing interest rate is based on the utilization ratio of the market along with constant factors for the minimum borrowing rate, $BaseRate_a$, and the relative increase in the interest rate with respect to utilization, $Multiplier_a$, both of which can be changed by community governance proposals.

$BorrowInterestRate_a = BaseRate_a + U_a * Multiplier_a$

The Charkha protocol does not accrue interest by applying this interest rate to each borrower's balances, for efficiency's sake. Instead, the protocol records the effect of compounding the borrow interest rate over time in an interest rate index for each market. This index begins at 1. Every time the interest rate changes, the index compounds the interest since the prior index.

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

In this formula, the interest rate $r$ is the current borrow interest rate and the time period $t$ is the proportion of a year that has passed since the last update. On Chainweb, blocks process about once every 30 seconds on a given chain for about 1,051,920 blocks per year. Therefore, if the index was last updated in the previous block, then we should compound the borrow interest rate $r$ with the time period $1/1051920$.

To determine the total borrow balance owed by any particular user, we can use the interest rate index at the time they borrowed funds versus the current interest rate index:

$Balance_{current} = Balance_{previous} * (Index_{current} / Index_{previous})$

When the interest rate index updates it means interest has accrued. Therefore, we must update the $Borrows_a$ and $Reserves_a$ for the given market. The reserves for a market are funds held back from borrower interest for the platform, which can be used to cover bad debts or other issues. The $ReserveFactor_a$ is a constant factor that can be changed by a governance proposal.

$Borrows_{a,n} = Borrows_{a,(n-1)} * (1 + r * t)$
$Reserves_{a,n} = Reserves_{a,(n-1)} * (r * t * ReserveFactor_a)$

With interest applied to the total borrows for the market, the utilization ratio will change, and therefore the borrower interest rate will change, too. The new borrower interest rate will be used

to compound interest the next time the interest rate index updates. This cycle occurs every time a user interacts with the protocol.

### 1.2 Supplier Interest Rates

We can derive a supply interest rate from the borrower interest rate and the reserve factor $ReserveFactor_a$ as seen below:

$$SupplyInterestRate_a = BorrowInterestRate_a * U_a * (1 - ReserveFactor_a)$$

However, the Charkha protocol does not record interest accrued to individual suppliers for the same efficiency reasons it does not record interest owed by individual borrowers. We'll learn more about how suppliers earn interest in the next section.

## 2. Supplying Assets

Suppliers earn interest by depositing their assets into the protocol. Charkha aggregates the supply of each user into a pool; once supplied, the user's asset becomes a fungible resource and can be withdrawn, with interest, at any time. When a user supplies assets to the Charkha protocol they receive an equivalent KIP-0005 token balance called cTokens.

As the money market accrues interest via borrowers, cTokens become redeemable for an increasing amount of the underlying asset according to the $ExchangeRate_a$. Holding a cToken automatically earns interest, even though the user's cToken balance does not increase over time. When a market launches, the $ExchangeRate_a$ rate begins at 50:1 (50 KDA to 1 cKDA, for example). The exchange rate then increases at a rate equal to the compounding borrowing interest rate:

$$ExchangeRate_a = (Cash_a + Borrows_a - Reserves_a)/cTokenSupply_a$$

Users can redeem their cTokens at any time. The user transfers their cTokens to the protocol, which burns the tokens and transfers the equivalent amount of the underlying asset, according to the exchange rate, back to the user's address.

## 3. Borrowing Assets

Users can borrow assets from the protocol instantly, with no terms, using their cTokens as collateral. Each money market has a floating borrow interest rate set by market forces; the user's borrow will accrue interest every time someone interacts with the protocol according to the borrow interest rate for that market.

Each market has a *collateral factor* from 0 to 1 that represents what portion of the collateral can be borrowed against. The collateral factor is a measure of liquidity and market-cap; highly-

liquid, large-cap assets have high collateral factors. A user's *borrowing capacity* is the sum of their cToken balances multiplied by collateral factors:

$$BorrowingCapacity_a = cTokenBalance_a * CollateralFactor_a$$

Since borrowing capacity is aggregated across markets, we can normalize the value of the cTokens a user holds in any given market by converting them to the underlying asset using the exchange rate, and multiplying that against the price of that asset in USD using a price oracle:

$$NormalizedBorrowingCapacity_a = BorrowingCapacity_a * ExchangeRate_a * Price_a$$

The user's total borrowing capacity is the sum of their normalized borrowing capacity across all markets in which they possess cTokens:

$$TotalBorrowingCapacity_{a,b} = NormalizedBorrowingCapacity_a + NormalizedBorrowingCapacity_b$$

Users can borrow up to but not exceeding their borrowing capacity. An account cannot take an action that would raise the total value of borrowed assets above their capacity (e.g. borrow an asset, redeem cTokens, transfer cTokens, etc.). Once a user has borrowed an asset they begin to accrue interest owed on their loan according to the interest rate index for that asset.

Users can repay their loan at any time by transferring their borrowed balance, with interest, to the protocol.

# 4. Liquidation

Borrowers will sometimes exceed their borrowing capacity, such as when their collateral declines in value and their borrowed assets increase in value. In this case, a portion of the outstanding borrowing can be repaid in exchange for the user's cToken collateral, at the current market price minus a liquidation discount of 10%. Anyone can do this, which encourages an ecosystem of arbitrageurs to step in, reduce the borrower's exposure, and eliminate the protocol's risk (ie. that a borrower cannot repay, causing losses for the protocol).

The *close factor* is the portion of the borrowed asset that can be repaid, ranging from 0 to 1 depending on how much the borrower has exceeded their borrowing capacity.

$$CloseFactor = (UserBorrows - UserBorrowCapacity)/UserBorrowCapacity$$

For example, if the user has borrowed 1000 KDA but has only a 750 KDA borrow capacity, then their close factor is 33%. Up to 33% of their total borrows can be repaid in exchange for their cToken collateral for the repaid amount, plus an additional 10% liquidation discount. In this example, assuming a 1:1 KDA / cKDA exchange rate, a user can repay 330 KDA to receive 363 cKDA from the borrower.

# 5. Rewards

The Charkha protocol rewards suppliers and borrowers with the CHRK token. This token is distributed to suppliers and borrowers in each market; the more you supply and/or borrow, the more CHRK you receive. Each market distributes CHRK according to the relative interest payments of each market, measured as its $Utility_a$.

$Utility_a = Borrows_a * Price_a$
$UtilityShare_a = Utility_a / (Utility_a + \ldots + Utility_z)$

Each interactaction with the protocol generates new rewards until 10,000,000 total CHRK have been distributed. Each market generates 1 CHRK per block, which is divided among borrowers and suppliers in the markets according to their relative proportion of market activity:

$RewardShare_{a,user} = (Borrows_{a,user} + Cash_{a,user}) / (Borrows_a + Cash_a)$

For example, if a user represents 50% of the activity in a given market, then they will receive 0.5 CHRK per block. CHRK is an ordinary KIP-0005 token, which means it can be freely traded or loaned. In fact, CHRK is one of the tokens the Charkha protocol supports.

CHRK is not automatically distributed to users for efficiency reasons. However, users can claim their CHRK at any time from the controlling Charkha contract.

# 6. Governance

The Charkha protocol is governed by CHRK token holders. Token holders can propose, vote on, and implement changes through the admin functions of a market's contract or the overall Charkha contract. Most proposals tweak system parameters such as the base rate and multiplier in borrower interest rates, or the collateral ratio of various assets.

Anyone with at least 100 CHRK can create a governance proposal. Others can vote, with each CHRK they own comprising a single vote. Users can lend or borrow CHRK for the purposes of voting, but when voting closes they must still have the CHRK balance used to vote.

Governance proposals remain open for 2 minutes in the test instance of Charkha, so that users can see the results of tweaking system parameters right away. A more traditional governance period is 7 days.