BitDEX: Building a Decentralized BitMEX using Priceless Finanical Contracts

DRAFT—v0.1

Regina Cai, Hart Lambur, Allison Lu
[regina, hart, allison]@umaproject.org

September 30, 2019

Abstract

In this paper, we present a decentralized levered contract-for-difference (CFD) trading platform called "BitDEX". The implementation utilizes "priceless financial contracts" to present a transparent, efficient, and scalable decentralized alternative to current fiat and blockchain-based margin trading platforms, such as BitMEX. The financial risk described is akin to a CFD or perpetual swap, although the priceless financial contract framework and underlying concepts can be applied to additional financial products, such as futures, swaps, and synthetic tokens.

Contents

1 Motivation

Levered financial transactions between counterparties require a trusted third-party operator (e.g. clearinghouse, exchange).

These operators perform some or all of the following useful functions for counterparties:

1. Aggregating liquidity among counterparties
2. Custodying margin on behalf of trading counterparties
3. Moving margin between counterparties ("clearing trades")
4. Ensuring that each counterparty has enough margin to pay for potential losses
5. Settling and closing counterparties' trades

In order to perform these functions, operators often charge counterparties fees, curate the list of counterparties permitted to trade, collect personal data, and confidentially record counterparty identities, positioning, and margin balances. As a counterparty, trading with a centralized operator means playing by their rules—ceding custody over your assets and control over who your counterparties are, how levered you can be, and how much money you can make.

By using a framework that makes counterparties' positioning and margining transparent (the "Priceless Financial Contracts" framework), we can create a high-performance levered CFD trading platform ("BitDEX") that provides counterparties with economic security and privacy while decentralizing the role of the operator to remove the financial and non-financial costs of centralization.

2 Priceless Financial Contracts

BitDEX is a single smart contract, per market, that custodies all counterparties' margin and publicly displays the net deposits of each counterparty. Counterparties are also responsible for monitoring other counterparties' positions to ensure

they are always properly collateralized. **Notably, there is no centralized price feed and no centralized arbiter of proper collateralization.**

This design is inspired by the observation that in a trade between Alice and Bob, only Alice and Bob need to agree on how much margin they each should have. If Alice and Bob agree that they are properly margined, they don't need a centralized operator to agree with them. They don't even need to explicitly state or agree on how much margin is required from each, so long as they are each satisfied with the other's margin levels. If Alice and Bob do not agree, though, they need a process in place to resolve their dispute. This can be extended to multiple pairs of counterparties, each with offsetting long and short risk.

We call financial contracts designed without a centralized price feed "priceless financial contracts". These contracts are securely margined when Alice and Bob each maintain margin that meets the margin requirements, knowing that if either of them does not, the other will initiate a dispute resolution process which will result in a penalty payment. By codifying the margin requirements in an objective form that anyone can understand and calculate, anyone can audit and verify that all counterparties are operating in line with the rules they claim to be following, without requiring an on-chain price.

This framework also introduces the features of being "quantum" and "optimistic":

- *Quantum*: The margin requirements at any point in time are not explicitly stated until "observed" after a dispute. The method or function of calculating the margin requirement does need to be agreed upon beforehand.
- *Optimistic*: If all counterparties are properly margined, they will never need to enter a dispute resolution process.

These features are also hallmarks of of the challenges and exit games of state channel approaches to layer 2 scaling solutions [1]. The design of this platform is influenced by Plasma Group's "optimistic virtual machine", in which participants' "local information" influences their "optimistic decisions" as to whether a proposed block is valid [2, 3].

3 BitDEX: Decentralized Levered CFD Trading

3.1 BitDEX Design Overview

In contrast to the "black box" design of most centralized operators, BitDEX is a single smart contract that transparently records counterparties' position and margin data on the blockchain. Because BitDEX is a publicly viewable smart contract, the system can easily be audited to verify that all long and short counterparties are matched off and that there is no residual risk left against the

smart contract. **Counterparties always trade against each other; no one trades "against" BitDEX.**

For simplicity, this paper will describe all BitDEX counterparties as having a single position defined as 1 unit of risk, long or short. Real-world designs where counterparties have multiple units of risk are left as an implementation decision.

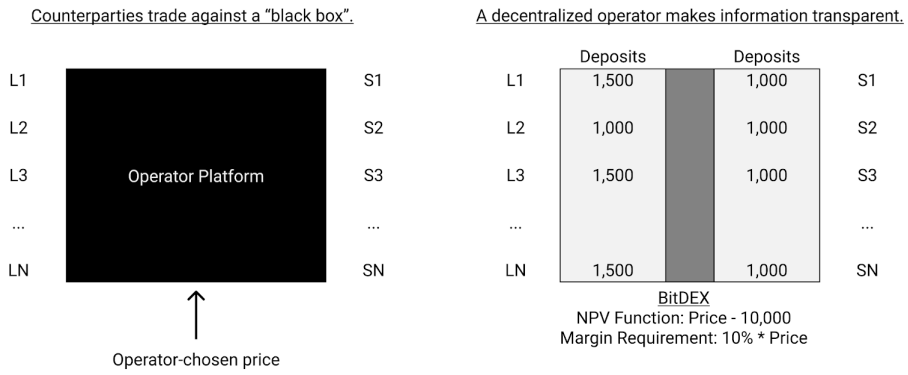

Figure 1: Traditional platforms are "black boxes", where counterparties' positioning and margining are known only to the operator, which relies on a price feed to move margin between counterparties. BitDEX publicly records each counterparty's margining and does not use a price feed.

Each BitDEX market is a single smart contract, whose parameters define the type of financial risk and rules of engagement for all counterparties. These parameters include an NPV function and margin requirement function that all counterparties agree to use and that are understandable by a dispute resolution process. BitDEX publicly stores the net amount of margin that each counterparty has deposited into it ("**deposits**"). By making all margining information publicly accessible, counterparties are able to view and dispute any position that they believe is undercollateralized. Rather than have a centralized operator "push" requests onto counterparties (making margin calls), counterparties must "pull" requests from the smart contract to deposit additional margin or withdraw excess margin. Counterparties are free to deposit and withdraw margin from BitDEX, although withdrawals are subject to a probation period, discussed below.

At any time, any long counterparty can dispute any short counterparty (or vice versa), a process outlined in a section below. This initiates a public dispute resolution process that will award a penalty from the disputed to the disputer if the dispute was valid. The only requirement is that the **disputer** and the **disputed** must have offsetting risk (one must be short and one must be long).

If the system behaves as expected, each counterparty should maintain sufficient margin to appease the most conservative counterparty with opposing risk. As a result, there should be very few on-chain calls to BitDEX to dispute.

Notably, while an NPV function and margin requirement function need to be written in such a way that all counterparties and the dispute resolution process can understand them, the functions themselves do not need to be calculated on-chain. Counterparties need only be capable of evaluating the functions off-chain and using the results of their off-chain calculations to determine whether a dispute needs to be called on-chain. This results in trades whose lifecycles are primarily devoid of any heavy computation on-chain.

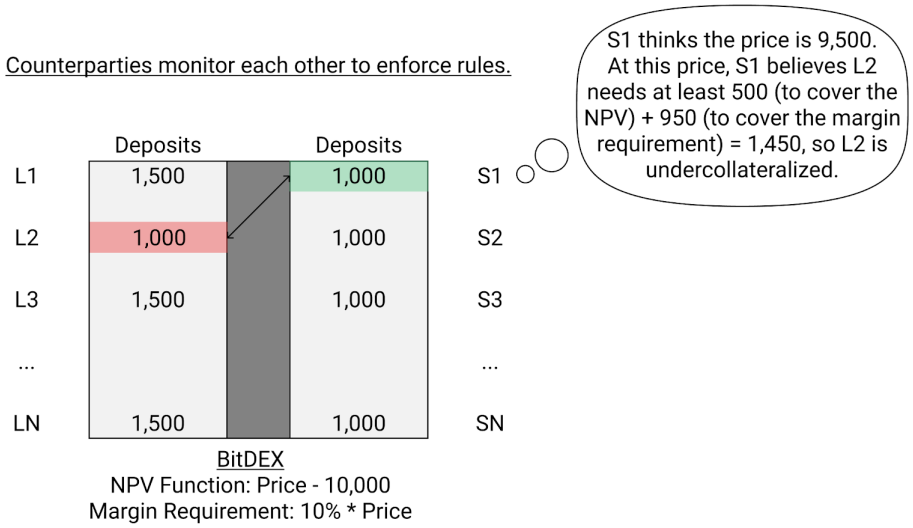

Figure 2: If short counterparty S1 believes that long counterparty L2 is improperly margined, based on off-chain calculations, S1 can dispute L2's position directly within BitDEX.

3.2 Trade Matching

Pooling multiple counterparties' margin in the same contract while ensuring that all counterparties maintain sufficient margin has historically been a difficult problem. As mentioned earlier, it is one of the biggest reasons that counterparties trust centralized operators to source a price feed. As such, when designing such a platform using Priceless Financial Contracts, where there is no centralized operator and no centralized price feed, special consideration must be given to the way that multiple counterparties enter and exit the platform.

BitDEX and the Priceless Financial Contract framework do not purport to solve any of the current problems of decentralized order matching. An orderbook

is still required to enable counterparties to coordinate and find each other to initiate new trades. Once those trades are matched, they can be recorded in BitDEX, where they will be securely margined throughout their lifetimes until unwound or otherwise settled.

3.2.1 Trade Initiation

For example, consider Alice, whose order to go "long" 1 unit of risk has been matched against Bob's order to go "short" 1 unit of risk via a separate mechanism. Based on the mutually agreeable price their orders matched at, they can calculate the current value of the NPV function and Margin Requirement function, which will influence the amount of margin they choose to initially deposit to begin their trade. All of this coordination happens outside of BitDEX, including how much of the margin that will be deposited into BitDEX to initialize the trade will come from Alice or Bob.

In line with the Priceless Financial Contract design philosophy, Alice and Bob should record their trade in BitDEX by depositing margin at levels acceptable relative to the parameters of the smart contract. However, because only other counterparties (not BitDEX itself) can check whether Alice and Bob are properly margined, BitDEX implements this safeguard by using existing counterparties' net deposits to infer whether Alice and Bob's initial deposits would be acceptable. So long as Alice and Bob's initial deposit amounts each meet or exceed the collateralization of other existing long and short positions, their positions should not to be disputed by the other existing counterparties.

3.2.2 Trade Collapse

Assume Bob, who has a short position, wants to close his trade. To do this, he enters into an off-setting long position. Bob can then collapse these two risk positions directly with BitDEX, withdrawing the sum of the long and short deposit balances while ensuring that all other counterparties remain securely margined.

4 BitDEX Contract Architecture

We divide the BitDEX architecture into four components: initialization parameters and global data structures; functions to manage margin for positions; functions to dispute undercollateralized positions; and functions to create or destroy positions.

4.1 Initialization Parameters and Data Structure

The following data structures and parameters are defined at the initialization of a BitDEX market:

- *longDeposits and shortDeposits*: data structures to record the deposits for all long and short positions, and all pending withdrawal requests.
- *livenessRequirement*: a global constant defining the delay required after a withdrawal request is filed before margin can be withdrawn.
- *calcNPV()*: a function to calculate the current NPV of any position
- *calcMarginRequirement()*: a function to determine if a position is currently margined correctly

The data structures *longDeposits* and *shortDeposits* publicly record the total margin deposited or withdrawn from any positions $(L_1...L_n; S_1...S_n)$, as well as any pending withdrawal requests. Anyone with access to the blockchain can view and verify the current deposits for any position, allowing any counterparty to initiate a dispute against an undercollateralized position.

A *calcNPV()* function is defined which, when evaluated, indicates the balance a counterparty is owed when their trade is terminated. Similarly, a *calcMarginRequirement()* function is defined which indicates the minimum amount of total margin a counterparty must have deposited at that point in time to not be in default. Importantly, these functions are not executed on-chain unless a dispute occurs. Rather, these functions are publicly recorded on the blockchain such that any counterparty and dispute resolution mechanism knows the exact logic that will be used to resolve disputes.

4.2 Margin Management Functions

To manage margin deposits and withdrawal, the following functions are defined:

- *deposit()*: a function to deposit additional margin
- *requestWithdrawal()*: a function to request the withdrawal of excess margin
- *withdraw()*: a function to withdraw approved margin

If any BitDEX counterparty believes their position is moving against them and may be liquidated for being undercollateralized, they can use *deposit()* to immediately add additional margin to that position.

Similarly, if any counterparty believes their position is overcollateralized, they can use *requestWithdrawal()* to signal to the system that they would like

to withdraw a specified amount of collateral. This request will be automatically approved after the time specified by the livenessRequirement has passed. This delay is inserted to give other counterparties time to observe the requested withdrawal and dispute it if it would leave the position undercollateralized. Any approved withdrawal requests can then be fulfilled using the *withdraw()* function.

4.3 Dispute Function

The core of the Priceless Financial Contract design is the *dispute()* function that maintains the solvency of the system by incentivizing counterparties to dispute positions they believe to be undercollateralized.

At any time, any long position can dispute any short position it believes to be undercollateralized, and vice versa. The only requirement is that the disputer and the disputed both are counterparties to BitDEX and have offsetting risk.

To initiate a dispute, a counterparty (the disputer) calls *dispute()* on another counterparty with offsetting risk that the disputer believes is undercollateralized. This immediately terminates the trade for both counterparties. BitDEX then initiates a dispute resolution process that tells it how much margin should be returned to the disputer and the disputed when the trade is terminated. This dispute resolution process could be implemented as follows:

- BitDEX logs a request with an oracle such as UMA's DVM for the accurate price at the time the dispute was initiated.
- When the oracle price is received, BitDEX runs the *calcMarginRequirements()* function to see if the dispute was raised accurately.
  - If the dispute was accurate, the disputer is paid the NPV (multiplied by their direction of risk) at the time of the dispute, plus a penalty paid by the disputed contract.
  - If the dispute was inaccurate, the disputer is paid the NPV (multiplied by their direction of risk) at the time of the dispute, less a penalty.

This both creates a strong disincentive for counterparties to dispute valid positions and rewards unfairly disputed positions for the inconvenience of having their trade terminated.

4.4 Trade Creation / Destruction Functions

Since any CFD platform must have an equal amount of long and short risk, all positions are created or destroyed in pairs. Creating new long/short pairs grows "open interest" in BitDEX; destroying existing long/short pairs decreasing "open interest". This is handled by two functions:

- *createPair()*: a function to create a new long/short pair.
- *destroyPair()*: a function to collapse a pair of risk.

Any two counterparties can create a new long/short pair by calling *createPair()* provided that sufficient margin is deposited in the new long and short positions. To prove that the margin balances are sufficient, the long must deposit margin greater than or equal to that of an existing, valid long position; similarly, the short must deposit margin greater than or equal to that of an existing, valid short position.

Checking that a new long/short pair is properly collateralized could be an expensive operation to perform on-chain. One simple and efficient implementation is to require counterparties to perform this calculation off-chain and then point to other existing long/short positions that have margin less than or equal to the new proposed positions when *createPair()* is called. If the existing position is considered properly collateralized by the BitDEX system, then by definition the new position will also be considered to be properly collateralized.

Any single counterparty that holds an offsetting long and short position can call *destroyPair()* and immediately be returned the sum of $long\_balance + short\_balance$. This is possible to do without any oracle or settlement process, since $long\_payout = long\_deposit + calcNPV()$ and $short\_payout = short\_deposit - calcNPV()$. Therefore, the holder of a valid long and short cancels out the *calcNPV()* and is entitled to an immediate payout of $long\_deposit + short\_deposit$.

5 Additional Financial Contract Designs

As described, we've shown how BitDEX can be used to decentralize existing CFD providers. This framework is extremely extensible, and using transparency to allow counterparties to do the work of the operators themselves can be applied to futures, synthetic tokens, and swaps.

Futures are another form of the CFD risk described above, but with an expiration timestamp (rather than perpetual risk). The same mechanisms can be used, while allowing for additional logic at the expiration timestamp to settle the relevant trades to a mutually agreed price, like one returned by a decentralized oracle such as UMA's DVM [4].

UMA has created a "Synthetic Token Builder" that looks like an overcollateralized lending facility, where a "Token Sponsor" creates and collateralizes a "Token Facility" [5]. The Token Sponsor then borrows "Synthetic Asset Tokens", backed by the collateral in his facility. The amount of collateral required to be maintained in the Token Facility is determined by a price feed.

To reduce the costs associated with maintaining a high frequency on-chain price feed, a Token Sponsor could do something similar using a slightly modified BitDEX smart contract. Each long counterparty could have long exposure

to the token, while each short counterparty could have short exposure to the token. A Token Sponsor would then define the NPV function of the BitDEX smart contract to be the price feed, and set the MR function to be 100% of the price for long counterparties and something less, say 10%, of the price for short counterparties. Long counterparties will never have to top up their margin, since they are, by definition, fully margined. Long counterparties can therefore tokenize their positions and trade them with others, allowing for potentially easier and broader adoption of the risk position.

Swaps are very similar to the CFDs described above, but with a different NPV function (and therefore, a different Margin Requirement function). The NPV function of a swap typically refers to the percentage performance of an asset relative to a strike level, multiplied by a notional amount, whereas the NPV function of the CFDs described above typically refer to the dollar performance (or USDC performance, etc.) of an asset relative to a strike level, multiplied by risk units (set to 1 in this example). Nevertheless, the same smart contract design can be used.

6 Conclusion

The Priceless Financial Contract framework, combined with a robust dispute resolution process, is a powerful and extensible infrastructure for building decentralized financial systems like BitDEX. By allowing counterparties to monitor each other's solvency, Priceless Financial Contracts enable high-performance, scalable decentralized financial products to be built. BitDEX uses these concepts to build a levered CFD trading platform that offers all the benefits of levered trading platforms without the downsides of a centralized operator. We look forward to seeing how this concept can be extended into other types of markets and financial contract designs and welcome further discussion.

Acknowledgments

The authors would like to thank Chris Burniske, Jill Carlson, Chase Coleman, Alex Evans, Yutaro Mori, Dan Robinson, and Chris Tonetti for their valuable feedback. The authors are particularly grateful to their colleagues–Varun Chirravuri, Matt Rice, and Prasad Tare–for their review and helpful comments.

References

[1] J. Poon and T. Dryja, The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments
`lightning.network/lightning-network-paper.pdf`

[2] Plasma Group, Introducing the OVM
medium.com/plasma-group/introducing-the-ovm-db253287af50

[3] J. Poon and V. Buterin, Plasma: Scalable Autonomous Smart Contracts
plasma.io/plasma.pdf

[4] UMA Data Verification Mechanism: Adding Economic Guarantees to Blockchain Oracles
github.com/UMAprotocol/whitepaper

[5] Announcing the UMA Synthetic Token Builder
medium.com/uma-project/announcing-the-uma-synthetic-token-builder-8bf37c645e94