OraclΞSwap

Derivative Contracts for Long-Term Investors

Lorenzo Botticelli

V1.0
7/3/2019

Abstract. OracleSwap is a cryptoeconomically-optimized suite of contracts oriented towards long-term investors and ETH holders. The contract's oracle is also its administrator, which minimizes costs and complexity. Counterparties take long or short swap positions in ETH, BTC or the US equity market and transact at forward-starting 'market-on-close' prices. Weekly settlement and payoff caps at required margins minimize necessary interaction while capturing virtually all position performance. Liquidity Providers post two-sided offers, receive portfolio margining, and are paid via Funding Rates. It is presented to both use and emulate, as all code is open source without any restrictions on usage, modification or distribution.

Greater detail on issues mentioned in this paper are presented in the *OraclΞSwap Technical Appendix*.

Contents

1. Introduction

OracleSwap is a suite of Ethereum contracts facilitating long-term derivative positions with a dedicated Oracle. Users access an AssetSwap Contract to create long or short swap (CFD) positions, which pulls prices from a dedicated Oracle Contract that warehouses price. Initial contracts reference ETH/USD, BTC/USD, BTC/ETH, and the US stock market (S&P500). All users need is access to the blockchain and ETH; there are no tokens, off-chain processing, matching algorithms, email or IP monitoring. An emphasis on simplicity reduces risk and makes the contract as straightforward as a vending machine, one click for a new user to take liquidity, two clicks to post.

The Coase Theorem highlights how low transaction costs and secure property rights are the foundation of efficient economic outcomes.[1] Novel properties like anonymity and immutability have created unfamiliar transaction costs that have stifled contract development, but we should have confidence these are temporary obstacles given the fundamental importance of property rights in creating efficient outcomes in that blockchain technology creates the strongest property rights in human history. This contract uses a novel set of methods to minimize transaction costs and align incentives to solve the oracle problem.[2]

An honest and efficient oracle facilitates what is drastically needed on the Ethereum blockchain: a useful contract referencing off-chain data. Most of the focus has been on making oracles like the blockchain itself: decentralized with an almost infinite set of use cases. This focus underlies the failure to create a good blockchain oracle.

Positions are like forward-starting futures contracts in that a fill price is the next 4 PM spot price reported by the Oracle.[3] When a Taker takes, the expected value of this transaction is the same for long or short, zero. This avoids latency problems inherent to limit-order books, eliminates price impact, and simplifies oracle monitoring. Weekly settlement and cash flow caps at required margins make it easier to manage in that players need only attend to their margin once a week to avoid default, which orients it towards investors as opposed to day traders.

Liquidity Providers (LPs) are paid to provide exposure as opposed to exposure at a specific entry price. As only the initiator of a cancel pays the closing fee, and LPs are paid via a funding rate, they have an incentive to provide long-term exposure. LP books have separate contracts that are settled separately, allowing the LP to net their exposures and facilitates scaling. This results in attractive LP returns at competitive rates for investors.

The only way to cheat is for the Oracle to post fraudulent prices that help a conspiring long or short counterparty. The Oracle basically owns a non-transferable annuity of closing fees from OracleSwap contracts, and comparing the value of this annuity to potential cheat payoffs shows the Oracle's dominant strategy is to be honest. The Oracle's reputation is critical, and the Oracle's Price Contract has a fixed address that generates clear and concise event logs for its price reports, which makes it easy to monitor.

OracleSwap's Oracle is also its administrator, as these roles are complementary: a derivative contract needs a price, and an oracle needs a derivative contract. As OracleSwap's oracle and administrator need

[1] The Coase Theorem states sufficiently low transaction costs and clear property rights generate optimal capital allocations including externalities, regardless of the initial allocation of property.

[2] See Transaction Costs Economics, e.g., the works of Oliver Williamson, Douglas North and Elinor Ostrom. This field analyzes why and how various transactions are performed within firms rather than a market and why market structures vary across industries. It is based on the idea that efficient solutions minimize contextual transaction costs.

[3] All times referenced in this paper are USA Eastern Time (ET), New York City time.

to be pseudonymous to avoid censorship, users must presume oracle-administrator collusion in a worst-case scenario anyway. By having the Oracle administer the other contract features expenses are minimized, and collusion scenarios are simplified. The Oracle's revenue puts it into a specific repeated game that makes it straightforward to create an incentive compatible contract.

The contract is permissionless and there is no centralization point for outsiders to target; all relevant data is put directly on the blockchain. Events like oracle price reports or when a counterparty cancels are easily observed via blockchain queries provided in our open-source front end you can download from GitHub, but also available at www.Oracleswap.co.

Our open source code gives people the ability to create their own OracleSwap-like contracts at relatively low cost. Ideally, there will be many imitators with different attributes, giving consumers choice and making an outsider's attempt to squash a particular OracleSwap as pointless as removing a PC from a peer-to-peer network. While OracleSwap relies on a centralized pseudonymous oracle, a set of oracle/contracts competing on the blockchain is a free market, the ultimate decentralized system.

2. Two Keys to Solving the Smart Contract Derivative Problem

2.1 No Limit Order Book

One key is *not* to emulate a modern stock market. The standard is now the centralized limit order book, where almost anyone can costlessly post, take and cancel limit orders. These markets are run on centralized databases and allow direct market access at under ten milliseconds. In contrast, interacting with an Ethereum contract is at least 1000 times slower and more costly, creating a classic example of Akerlof's market for lemons.[4]

A standard central limit order book has three types of traders: noise, informed, and market makers. *Noise* traders do not affect the price because their order flow is random. *Market makers* provide instant liquidity to noise traders by posting two-sided resting limit orders where their bid is below their ask price. *Informed* traders have an information advantage that predicts future order flow, which can come from a subtle algorithm, low latency, or something as straightforward as knowing they have many more orders to fill on the same side. In equilibrium, the market maker needs to make enough revenue off the noise traders to cover the losses generated from informed traders.

Higher latency subjects limit orders to greater adverse selection, where the orders that tend to get filled are only the bids that are too high or the offers that are too low. This causes the market makers to post wider spreads compared to lower latency markets. Higher spreads discourage popular yet delusional short-term traders whose order flow is random. Fewer noise traders increases the proportion of informed traders who inflict predictable maker losses, causing the maker to increase spreads further to compensate for these losses. This positive-feedback loop causes markets to ultimately unravel, as the noise traders

[4] Akerlof's "Market for Lemons" (*Quarterly Journal of Economics*, 1970) paper analyzes markets where parties with asymmetric information separate, and so the only offers have obvious negative value to one party, and the market collapses (i.e., no trades).

essential for limit order books disappear. There is no chicken-and-egg problem facing current blockchain-based limit order books; high spreads and low volumes are intrinsic to any high latency system.[5]

Low latency exchanges off the Ethereum blockchain come at a price. The lightly-regulated exchanges have considerable credit and operational risk, and only plausible deniability limits their front-running activity. The highly-regulated exchanges have low credit risk and are essential for converting between fiat and crypto, but put users back on the traditional financial grid, removing anonymity, jeopardizing custody, and have pathetic leverage and shorting capabilities per regulatory dictates.

OracleSwap eliminates the limit order book by using forward-starting, oracle-supplied prices to avoid latency effects while staying on the blockchain. This solution is inspired by value-weighted-average-price (VWAP) transactions, a popular equity trading tactic among long-term institutional investors. By spreading the execution price over a future time window—often the next day's average price—VWAP overcame the problems that prevented market-on-close orders from scaling. Most relevant to this contract, the lag in trade execution does not inconvenience long-term investors, as a future 24-hour fluctuation is unforecastable noise to a long-term investor. OracleSwap uses market-on-close orders, which are conceptually similar to VWAP but simpler and more efficient given the liquid assets we are initially targeting.[6]

A fair, liquid, forward-starting price is all that is needed for investors to get efficient fill prices. Most VWAP trading is done on a *best effort* basis, where the broker merely targets VWAP, and charges this plus a fee (often 0.2 cents per share). This puts the brokers in the same position as an accountable oracle: the only thing preventing them from charging a dishonest price is the loss of future business. While some brokers are better than others, fraud in this market is unheard of because the brokers are playing a repeated game where the present value of honesty clearly dominates the value of a cheat. This sort of mechanism underlies the incentives for OracleSwap's oracle.

LPs are not getting paid for price discovery, just the incidental market risk they assume when their net position is long or short, which they have no control over. By avoiding the adverse selection risk in limit order books LPs neither endure this unnecessary expense nor do they need to invest time and money minimizing it. This also allows investors to avoid price impact in that they need not parcel out an order to avoid moving market bid-ask prices, so completing a trade at a fair price takes the same amount of time whatever the size. Lower indirect costs combined with portfolio margining make it possible for LPs to generate an attractive return at competitive rates for long-term investors.

2. 2 No Decentralized Oracle

The benefit of *market* decentralization is that by giving individuals property rights those with the most relevant information have the incentive to allocate resources towards their highest-valued use, the classic invisible hand of Adam Smith. While decentralization is necessary for a prosperous economy, this does not imply firms within such an economy must be decentralized, and very few are. In the same way, while a blockchain must be decentralized to preserve essential crypto principles, this does not mean oracles must be as well.

[5] 'High' and 'low' in this paragraph are all relative to an asset's alternative markets. For example, housing markets have high latency relative to stocks, but work well because everyone is equally slow.
[6] If OracleSwap lists a less liquid asset or becomes very popular, the solution would be to switch from market-on-close to a VWAP over some time window. This makes auditing more difficult, and initially unnecessary.

Blockchains need to be decentralized for several reasons, most prominently to protect them from outside attackers who may wish to censor various transactions if not shut them down entirely. A peer-to-peer system based on block producers that appreciate the importance of anonymity, permissionless access, and immutability can withstand attacks by ignoring compromised nodes, and for blockchains like Ethereum the profitability of mining under its founding principles gives its users confidence that there exists a significant reserve of such miners. In a worst-case scenario, a hard fork can occur if the attacking miners were clearly usurping foundational principles, and the mere possibility of this is sufficient to dissuade most outside attackers, as the attack would be both costly and futile.[7]

Self-interest by miners and potential miners protects against both internal collusion and external attacks, which are essential for giving any crypto currency value as a long-term store of wealth. A permissionless decentralized blockchain allows a pseudonymous oracle to inherit attack resistance merely by being on the blockchain, so it does not need decentralization to avoid censorship. The other benefits of decentralization are more efficiently managed by properly incenting the oracle as opposed to creating a decentralized governance structure like that of the blockchain itself. The idea that oracles must be decentralized is a *category error*, assuming the part must independently have the properties of the whole.

The game theoretic field of *mechanism design* highlights two necessary conditions for a good oracle. First, there is a *participation* constraint that motivates players to want to participate in the oracle's game. A platform may incent honest reporting but the costs—not the explicit fees so much as the delay, complexity and bid-ask spread—make 'not playing' the preferred choice.[8] Secondly there is an *incentive compatibility* constraint that the oracle can achieve its best outcome by reporting truthfully. Incentive compatibility is key to low-cost enforcement of contracts, and historically this mechanism centered on reputation, not contract law administered by the state.[9] When agents have incentives aligned with their counterparties, we minimize policing and enforcement costs, which makes it easier to satisfy the participation constraint.

Creating a cooperative or honest oracle is simpler and more efficient when there is only one oracle agent, making it easier to find rules that minimize the cheat payoff and maximize the value of developing an honest reputation. The 'wisdom of the crowd' effect is irrelevant for reporting liquid market prices, and mechanisms to filter out bad data are more efficiently done off the blockchain.

Time-consuming mechanisms that make it almost impossible for agents to collude are not just a distraction, but naïve. Systems that work on the blockchain—large blockchain miners, Infura—all *can* collude maliciously, but are constrained by their self-interest.[10] The key is making sure that whenever the oracle has a choice between dishonesty and honesty, the latter has greater value. More generally, an agent with sole non-transferable oracle rights and responsibilities reporting unambiguous prices to a specific contract has a stronger incentive to be honest than a fluid set of agents reporting on a wide variety of

[7] This is related to 'contestable market theory,' an economic concept that refers to a market in which there are only a few companies that, because of the threat of new entrants, behave in a competitive manner.

[8] Our current financial system, meanwhile, solves this problem by simply making alternatives illegal.

[9] For example, prior to civil commercial law there were courts along trade routes throughout Medieval Europe that enforced commercial laws (the *Lex mercatoria*), and its judgments were accepted not out of any legal authority granted by a state's monopoly on violence, but rather refusal would ruin one's business reputation and thus future revenue.

[10] There have been several cases where BTC and ETH miners have effectively colluded, but every significant case was an effort to protect the blockchain, not subvert it (e.g., Sep '18 Bitcoin DoS/inflation bug). 51% attacks for large blockchains are feasible but imprudent, which is sufficient.

events and contracts that no one can easily monitor. Consensus mechanisms are costly, and there is no reason to think that the key to creating an honest, accurate oracle is that they contain legions.

A trusted oracle needs to provide open-source code so that users can verify its logic and security. OracleSwap provides entrepreneurs a template to create their own versions of oracle-contracts, facilitating a market of competitors. Firm and individual reputation have always been important in markets because at some point in the transaction process this sort of accountability mechanism dominates decentralization in terms of efficiency. For an oracle, this is easiest to do when it is a single pseudonymous agent that reports on a focused set of outcomes tied to a contract where its payoff space generates proper incentives.

All OracleSwap needs to work efficiently is for the oracle to accurately report simple price data, and contract logic makes the honest reporting the Oracle's profit maximizing strategy at any time. All users need trust is that the Oracle *can* do what it selfishly *wants*, which comes from the permissionless access of the blockchain and a properly incented oracle.

3. The OraclΞSwap Solution

OracleSwap aligns incentives and minimizes costs—both direct and indirect—in the following ways:

- Only the oracle can cheat, and if a player sees a fraudulent Oracle reported price they have a 4-hour window where they can and should burn their PNL debit rather than send it to their counterparty, presumably an oracle sock-puppet. [11] Burning requires an extra cost to discourage griefing, but a cheated player is incented to burn when cheated, trivializing the exit scam potential for the oracle.
- A cheat is easily observed via the small standardized set of daily oracle price reports, and no rational person would use a contract serviced by an Oracle who has cheated, as it has complete responsibility for reporting an objectively verifiable event. Comparing an exit scam payoff versus the lost annuity of future closing fees shows that honesty is the Oracle's dominant strategy.[12]
- Market-on-close prices supplied by the Oracle eliminates latency problems, minimizes execution costs, and simplifies trading (no more cancel and replace). One can be a Liquidity Provider (LP) without specialized hardware or software as latency is not disadvantageous. Contract returns are based on business-day 4 PM cash prices assets that are easy to audit, and settlement occurs each Wednesday at 9 PM.
- LPs post ETH as margin, and takers then take long or short positions against the LP by also posting margin, where the initial fill price is the same market-on-close for both long and short positions. Each LP gets portfolio margining, netting long and short positions at each settlement, allowing them to support more takers.
- Its simplicity reduces cost and risk. OracleSwap needs an oracle, LPs, and investors, so these are the only parties that interact with the contract. There are only three contracts, the longest containing 500 lines of code.
- Forced liquidations are avoided by capping weekly PNL at the Required Margin (RM), which combined with Leverage Ratio (LR) and ETH price sets the notional exposure. This eliminates the need to monitor the market between weekly Wednesday settlements, and LRs are set so the expected value of truncated returns is economically irrelevant.
- RMs have a minimum size of 10 ETH, which corresponds to notional amounts of around $3k for cryptos, $15k for the SPX. Combined with the forward starting price at end-of-business day and Wednesday-only exit, this orients the contract towards rational long-term investors. Given the gas costs of managing a position a minimum size is needed to avoid economically unviable positions.

[11] Burn fees and burnt PNL are sent to the eth burn address (0xdead). Burning rather than contributing to some pool is related to Holmström's Theorem, which states that no incentive system for a team of agents can satisfy all the following properties: money in=money out, Nash equilibrium, and Pareto efficiency. See Bengt Holmström, "Moral Hazard in Teams" (*The Bell Journal of Economics*, 1982).

[12] Such outcomes are 'off the equilibrium path,' in that it is never reached in equilibrium, but its existence is necessary for the equilibrium.

3.1 Use cases

Easier, cheaper, and more transparent for investors. All one needs is access to the blockchain and some ETH, no 8-week onboarding process or tokens. With forward-starting prices and no forced liquidations, one merely attends to the contract once a week to sustain a long-term position. OracleSwap removes the costs involved in moving between ETH, BTC, stable coins, stocks, and USD: fees, time, crossing the spread. Transaction prices are the same regardless of size, side, maker or taker. LPs have a financial incentive to not initiate a close, giving takers exposure for as long as desired.

Diversification. There are many reasons people may not want to sell their ETH or BTC, yet diversification is the only free lunch in economics—creating value by lowering volatility without sacrificing expected returns—anyone with a significant amount of their wealth in BTC or ETH would be wise to allocate some of this capital to other assets. Those in countries that make it hard to get access to the world's largest equity market can find the SPX valuable diversification as well.

Leveraging ETH or BTC. An investor who wishes to remain in their crypto position often does so in part because they are bullish. OracleSwap provides a convenient way to increase one’s crypto exposure by offering 2:1 leverage on ETH, and 2.5:1 leverage for BTC and ETHBTC. Thus with 10 ETH, one can fund an additional 20 ETH notional long position, tripling one’s ETH exposure.

Shorting ETH. An ETH HODLer can create a short ETH position while staying within the ETH blockchain. 2:1 leverage implies that if you put 100 ETH in your margin, you generate a short position worth 200 ETH, so your net ETH position would be short 100 ETH.

Hedging. With OracleSwap one can completely eliminate their ETH portfolio's USD volatility while staying on the blockchain. The tactic here is merely matching one’s base long ETH to their short ETH long notional amount each week. As in many things we get most of the benefits with a fraction of the effort needed for perfection. For example, assuming ETH has an annualized volatility of 100%, if one were to invest *over three months* and applied a rule to withdraw or add to one’s margin if their ETH margin increased or decreased by 35%, one could reduce their net ETH’s USD fluctuation by 85% while only requiring an average of 1 margin adjustment.

Attractive ROE for Liquidity Providers (LPs). Attractive taker rates correspond to attractive returns for an LP. The marginal risk generated by being an LP comes from the random net exposure as long and short takers arrive, but as this is random it will be uncorrelated with a long ETH position regardless of the asset. The net effect of cross-margining and the diversification of uncorrelated marginal exposure generates Sharpe ratios above 1 over a variety of assumptions. As equity Sharpe ratios have averaged 0.33 over the 20th century, this is an attractive investment for a long-term investor.[13]

[13] A Sharpe ratio is the (return – Tbill rate)/volatility. Data are annualized. General US equity data suggest the ret=9%, risk-free rate=3%, vol=18%.

4.0 Contract Specs

4.1 PNL

A subcontract is a bilateral agreement between LP and Taker for a particular asset (e.g., SPX). It has a constant Required Margin (RM), which is converted to a notional amount via the Leverage Ratio (LR) and the ETH price.[14] LRs are set so that the RM represents a 3-standard deviation PNL, implying one should expect the next settlement to create a debit or credit of one-third their RM. Unlike futures accounts, there is only one RM, not an initial and maintenance margin; the RM in ETH is fixed for the duration of the subcontract. Those desiring lower leverage can simply overfund their margin without changing their notional exposure or margin at risk.

Each week the asset return and Funding Rate are applied to the start-of-period notional amount.

For reference assets priced in USD, the weekly cash flow, PNL (for **P**rofit a**N**d **L**oss), represents a investment in an asset, such that a notional amount is denominated in the same currency that the asset return is calculated in. Thus, in going long BTC priced in USD, implicitly the trade is as follows: at open buy USD with ETH and then buy BTC, at close sell BTC for USD and then buy ETH. This is the same as when a US investor invests in a Japanese stock: she needs to first buy yen to buy the stock, and eventually sell the stock for yen which she sells for USD. The contract does not actually do all of those transactions, but it prices them all to generate the weekly PNL, which is denominated in ETH.

$$PNL_t^{Taker} = \overbrace{RM \cdot LevRatio}^{\text{RM levered into notional exposure}} \cdot \overbrace{ETH_{t-1}}^{\text{translate to notional currency}} \cdot \left(\overbrace{+/-}^{\text{+ Long - Short Indicator}} \overbrace{\left({}^{A_t}\!/\!{}_{A_{t-1}} - 1 \right)}^{\text{asset return in notional currency}} \overbrace{- FundingRate}^{\text{LP Fee}} \right) \cdot \overbrace{\frac{1}{ETH_t}}^{\text{convert back to ETH}}$$

As each subcontract implies symmetric PNLs to its counterparties, Taker and LP, we need only calculate the PNL for one party: $PNL^{Taker} = -PNL^{LP}$.

Example:

RM=10 ETH, Leverage Ratio=2.5, Funding Rate =0.2%, Taker long, Asset BTC.
Start prices: ETH=\$145, BTC=\$5800. End prices: ETH=\$155, BTC=\$5600

$$PNL^{Taker}\left(long\right) = 10ETH \cdot 2.5 \cdot \frac{\$145}{1ETH} \cdot \left(+ \left({}^{\$5600}\!/\!{}_{\$5800} - 1 \right) - 0.002 \right) \cdot \frac{1ETH}{\$155}$$

Taker PNL = - 0.853 Ξ = – LP PNL
USD Notional at start of period= \$3625=10·2.5·145
Asset Return= - 3.45%
Taker USD Return on Notional= - 3.65% = - 0.853·155/3625
Note that the difference in the asset return and the contract return is 0.2%, the Funding Rate paid to the LP.

[14] Leverage Ratios are 2.5 for ETHBTC, 2.0 for ETH and BTC, 10 for the SPX.

For reference assets priced in ETH, currently the BTC, the calculation is simpler as we do not need to convert into and out of USD.

$$PNL_t^{Taker} = RM \cdot LevRatio \cdot \left(+/- \left({A_t}/{A_{t-1}} - 1 \right) - FundingRate \right)$$

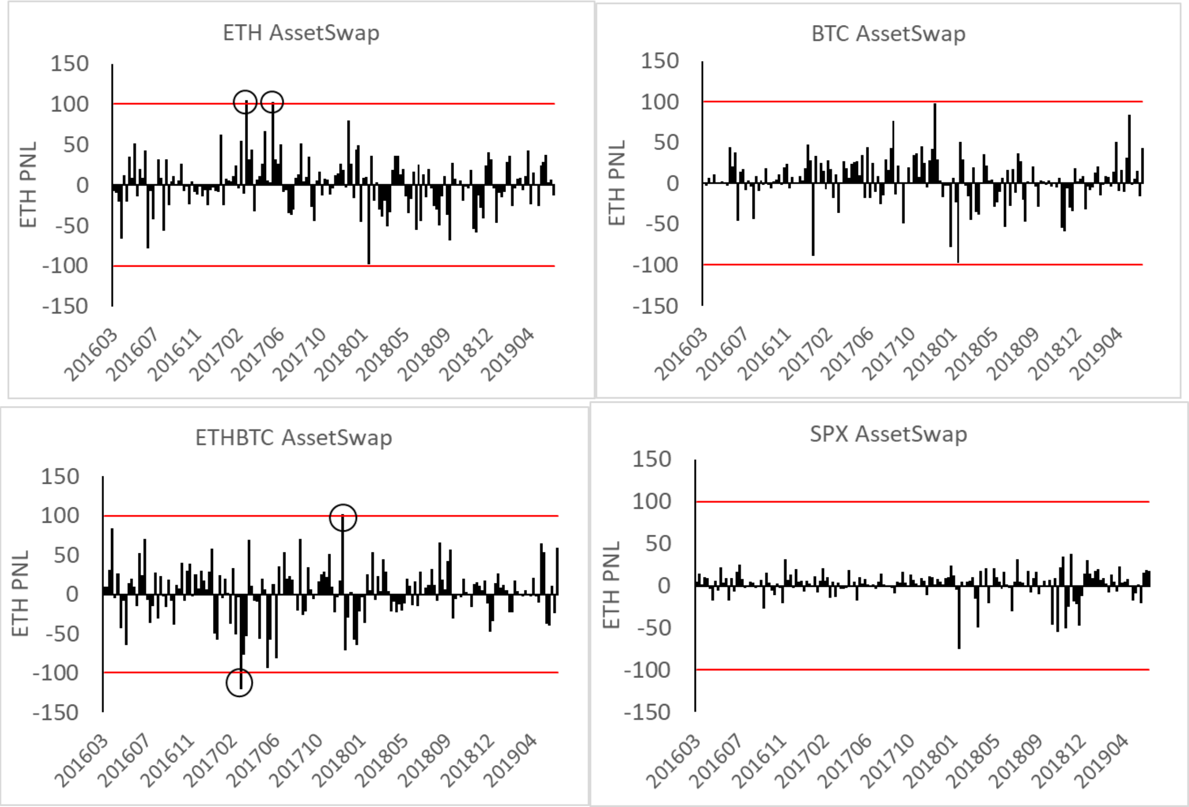
4.2 The Effect of the PNL Collar

The PNL caps make the long position like a long in the underlying, plus a portfolio of long put/short call that creates a *collar position*. The value of this option portfolio is dependent on whether the value of this 'long call/short put' is significantly different from zero. Unlike most collars this one has extreme out-of-the-money options.

We simulate the unconstrained PNL given a RM of 100 ETH for the four contracts over historical data. Only 4 out of 652 weeks generated PNL outside the RM, and the correlation of the unconstrained to constrained PNL is 99.9%. The only significant truncation was in May 2017 when the price of ETH doubled while bitcoin remain flat, generating a USD opportunity cost of 17% for the BTCETH contract (the long lost (short made) 17% less in USD than it would have without the cap). Those outlier weeks are shown below where they breach the RM bounds:

PNL Simulations

Weekly data from March 2016 through June 2019. All contracts have RM=100 ETH. PNL here is in ETH, not USD, as the RM cap applies to ETH.



Simulations assuming 100% annualized volatility generate economically irrelevant net effects (~ 0.4% annualized), orders of magnitude lower than the basis we see at BitMex. That is, to the degree participants wish to price this into the basis, the unobserved 'risk premium' will swamp this more objective adjustment.

4.3 Settlement

At 9 PM every Wednesday the Oracle runs the settlement function, which transfers the week's PNL between the counterparty margins. The counterparty margins are adjusted as follows at settlement:

$$TakerMargin_t = TakerMargin_{t-1} + \min(RM, \max(-RM, PNL_t))$$

$$LPMargin_t = LPMargin_{t-1} + \sum_i \min(RM, \max(-RM, PNL_t^i))$$

Here the min/max function is the collar on the PNL, in that RM is the maximum credited or debited for any subcontract.

The Oracle executes the settlement for each LP book in separate function calls, as first it runs a bockchain query to identify LP books associated with OracleSwap that have active subcontracts; processing all the LPs with active subcontracts implies processing all the active Takers. The settlement function first processes the LP's subcontracts that have positive PNL for the LP so that its gains plus RM will be enough to fund their losses.

The RM is both the initial and maintenance margin throughout the subcontract's life. For a taker, only a negative PNL can create a situation where a player's currently adequate margin will be deficient if not cured at Settlement, but a negative PNL does not necessarily imply an inadequate subsequent margin. For example, if the RM=10 and a player has 11 in their margin, an impending PNL= – 2 implies they need to add at least 1 ETH to their margin at settlement to avoid default. For an LP they could have a balanced book with RM=0, and so if their actual margin was 0 and one taker cancels and creates a positive RM, this would require a margin cure by the LP even if the LP's PNL was 0. In a default, the subcontract is terminated at settlement and the defaulting party's margin is debited a default fee of 12.5% of RM to make canceling dominate defaulting as a way of exiting subcontracts (cancelling is only 1.5%). A default does not cost the non-defaulting counterparty anything, just stops their contract; given no open fee, the counterparty can replace it at no cost.

4.4 Funding Rates and Basis

A swap or CFD referencing cash prices must adjust for the *basis* one sees in futures/forward market curves to account for the opportunity cost of money—the risk-free rate—and the costs and benefits of owning the asset outside of its price appreciation (e.g., storage costs, dividends). If there were no basis one could invest their cash in money markets then go long the futures and make the same return as investing in the asset plus the interest rate earned in money markets, which is not an equilibrium. A basis is charged symmetrically, subtracting from the long position and adding to the short.

In OracleSwap Funding Rates not only include a basis but also the fee that pays LPs to provide liquidity. Liquidity Providers in most markets are paid via the bid-ask spread or more subtly via price impact, both of which are not explicit. OracleSwap has no trading at maker-supplied prices, so our makers, the LPs, must be paid in a different way. Therefore, the Funding Rate consists of two components, a Target Rate to pay the LPs, and a basis rate to equilibrate long and short taker demand.

$$Long\ Funding\ Rate = Target + Basis$$
$$Short\ Funding\ Rate = Target - Basis$$

The basis on bitcoin futures at the CBOE or Deribit is often near zero, but the more comparable market is BitMex, where it is usually above 25% annually, suggesting that the central issue in blockchain Funding Rates is not the objective considerations above. The difference between this rate and that implied by mere interest rates is called a *risk premium* and occurs in many standard futures markets. One can explain a +25% risk premium just as easily as a -25% risk premium by noting those paying that basis are hedging their natural positions and those receiving are being paid to take risk. While it is useful for investors to understand the effects of interest rates and other factors on the basis, ultimately this differential is determined by supply and demand.

The Oracle will adjust Target and Basis rates to maximize outstanding balances, as this maximizes the Oracle's revenue. An equilibrium Basis rate that balances long and short demand increases gross notional for an LP, encouraging LP supply regardless of the Target Rate. Target Rates too high will discourage takers, while Target Rates too low will discourage LPs, and so balancing these objectives also maximizes the gross book size for the Oracle.

4.5 Becoming a Liquidity Provider

To become an LP one simply posts the following:

- Choose asset: SPX, ETH, BTC, BTC/ETH
- State a minimum RM. As each LP can have up to 90 takers, a minimum RM is needed so that those with large Margin do not max out their book with small subcontracts. This must be greater than or equal to 10 ETH.
- Post ETH margin.

At any time the LP can withdraw and then receive back whatever margin has not been taken by a Taker, and no fee is applied in such a case. Any amounts taken by a Taker are locked in until the next settlement. On the first settlement (the first Wednesday) the LP's positions are netted for calculating the LP's RM within any AssetSwap Contract (an LP can use the same address for BTC and SPX, but these would be independent books). [15]

LPs are motivated to extend liquidity for as long as takers want it in two ways. First, they are paid weekly on their outstanding balances via the Funding Rate, so the longer their positions exist the more money they make. Secondly, they maximize profits by adopting a strategy of rarely initiating a close, avoiding the closing fee.

There are economies of scale in being an LP, as the gas and time needed to tend the position are the same regardless of LP size. As LPs can only have up to 90 active subcontracts, an LP posting 1000 ETH would be poorly served if all its takers each took 10 RM. LPs can post minimum RMs to better manage their books.

[15] An LP who posts 200 ETH, and is soon taken such that she is long 100 and short 50, can only withdraw 50. At first Settlement her RM would then be 50, and so she would have an excess margin of 150 she can withdraw.

LP's are protected from getting too unbalanced in that takers cannot take positions that push an LP's book beyond 50% of her total offered amount. Specifically, the rule is:

Table 1

Maximum Taker Amounts as a Function on an LP's Book

Max Long Take = ½ ExcessMargin + Short – Long
Max Short Take = ½ Excess Margin - Short + Long

For example, the maximum amount a taker can take is shown below. Note that the Excess Margin is just the Total minus Required margin.

		Margin			**Max Take**	
Long	**Short**	**Required**	**Total**	**Excess**	**Long**	**Short**
0	0	0	100	100	50	50
0	100	100	100	0	100	0
100	0	100	100	0	0	100
0	0	0	250	250	125	125
0	100	100	250	150	175	0
100	0	100	250	150	0	175
100	100	0	250	250	125	125

An LP posts two-sided offers, and while the Oracle will try to set the basis to equilibrate long and short demand, invariably there will be periods where LPs will be unbalanced and exposed to market risk. This risk is the cost of generating an attractive LP return. For current ETH investors allocating 25% of their ETH to provide liquidity in OracleSwap generates a marginal Sharpe ratio above 1, which is exceptional (see Technical Appendix). This is because LP income is a function of their gross outstanding, yet their risk comes from their net outstanding. This net exposure is randomly long or short, implying a zero correlation with their current ETH position.

Real short-term government rates have averaged around 0.6% over the past century across 17 developed countries. For most savers without direct access to Treasury Bills, this implies an effective short-term riskless rate of zero, no different than when your ancestors buried their gold in the fields.[16] We should not expect this to be any different in crypto: zero risk generates a zero return. Platforms offering a 5%+ return for simply loaning their crypto implies an unacknowledged small probability of a very large loss. In contrast, OracleSwap LPs can generate sustainable attractive returns because they are taking a reasonable and understandable amount of risk from exposure generated by random net positions.

4.6 Taking a Position

To take a position, one does the following

- Find an LP book in the asset class desired
- Take a side (long or short) and an RM amount greater than the LP's minimum, but less than the Max Take for the side desired (Table 1 above). Must be integer amount.
- Send ETH ≥ chosen RM to their margin

[16] Currently money market rates are 2.2% lower than T-bill rates.

When a Taker *takes* the OracleSwap contract will generate an event log of the subcontract with the counterparties' addresses, informing both parties they have a live contract, and the players are committed to a contract to the next settlement. In the event of an LP closing the taker's position prior to a taker's investment horizon, a taker can replace the position at no cost because only the initiator of a close pays the close fee (and there is no open fee).

A subcontract uses the first 4 PM business-day close after initiation as its starting price, where business day is defined by the New York Stock Exchange (NYSE).[17] A fresh take is instantiated around 3:45 PM, before the close. For example, a contract taken Monday at 3 PM ET will have its first price taken on Monday close; a contract taken Monday at 4 PM will have its first price use the Tuesday close. After that, all subsequent PNL-relevant pricings use Wednesday closings.

4.7 OracleSwap's Oracle

The Oracle has four primary responsibilities:

- Updates prices daily to the Oracle Contract at 4:15 PM on NYSE business days[18]
- Executes the settlement at 9 PM ET Wednesday
- Initializes trades to set their initial price date around 3:45 PM each business day
- Adjusts the Basis and Target Rate to equilibrate long/short demand as needed

OracleSwap's Oracle will make a best effort to execute contract functions at stated times, but network traffic prevents precision. In a worst-case scenario where the Oracle is incapacitated and unable to update an active subcontract for nine days, all players can withdraw their entire margin. This involves redeeming their positions...

The Oracle has enough discretion to avoid absurd outcomes. For example, congestion can push back the timing of various functions being processed, but there will always be a 4-hour window between the Oracle Contract update and settlement. More mundane actions by the Oracle, restrictions on its actions, and implications for malicious behavior, are discussed in the Technical Appendix.

4.8 Fees

Close fee: The player who initiates a cancel pays a fee via a payable function. It is 1.5% of the RM, which corresponds 0.15% of notional for SPX deltas, and 0.6% of notional for ETH/BTC and BTC/USD, and 0.75% for ETH/USD. This fee goes to the Oracle.

Funding Rate: Funding Rates are quoted in weekly rates and are applied to the notional amount. These rates are updated each week just prior to settlement. Target Rates have a hard-coded range between 0% and 1%, while basis rates are constrained to be within 0% and 2%, so that the Funding Rate is constrained to within -2% and +3%. Target Rates initially will be 0.04%, 0.15%, and 0.19% weekly (2%, 8%, and 10% annualized). They are paid by the taker to the LP and differ by long and short for each asset.

[17] Stock exchange holidays are scheduled three years in advance, though occasionally there are *ad hoc* holidays, such as recently when ex-President Bush died and markets were closed on Wednesday, 12/5/18. If markets are closed Wednesday, we use the prior business day as the settlement Day.

[18] There are about 9 New York Stock Exchange holidays, and if they occur on a Wednesday, we will use the prior Tuesday as the Wednesday settlement date. Holiday-shortened days will use the closing stock market prices, but the 4:00 PM ET crypto prices that day.

Default fee: If a player defaults due to their Actual Margin<RM, they are charged 12.5% of their RM. As cancelling is not allowed during the Settlement Period (between the Oracle Price Update and and Weekly Settlement), defaulting is easily avoided and much more costly than cancelling (cancelling is only 1.5% of RM). A default is subtracted from the margin and sent to the Oracle. If the Excess Margin after the PNL is less than 0.125·RM, the default fee will simply take whatever is there.

Burn fee: This fee should never be invoked because the Oracle is incented to be honest, and a dishonest oracle price is the only rational reason for an investor to burn. However, a player who thinks the Oracle is cheating can pay a fee of ¼ of his RM to preclude his counterparty—presumably an agent of the Oracle—from receiving its PNL debit via fraudulent pricing. This fee is applied via a payable function, and like the burner's PNL (if <0) is sent to a burn address (0xdead). Burning money makes certain the fee or PNL cannot go to an Oracle sock-puppet address that would complicate OracleSwap's incentive structure.

4.9 Contract Code

OracleSwap consists of three Ethereum contracts: an Oracle Contract, an AssetSwap Contract, and the Book Contracts that pertain to specific LPs within a particular AssetSwap Contract. The Oracle Contract warehouses all the price data used by the subcontracts, generates event logs on its updates, and is set up for adding additional assets. Initially, we will have four AssetSwap Contracts for ETH, BTC, ETHBTC, and SPX. AssetSwap Contract LPs then have separate Book Contracts, so that all the margin for a particular LP's subcontracts is held there.

While one can use the website, a GitHub account provides a downloadable front-end to access the contract, and Takers and LPs both interact directly with the AssetSwap Contract. These contracts are relatively concise, only about 500 lines of code, and all open-source and fully readable. The security flaws discussed in the book *Mastering Ethereum* and others are all discussed in connection to OracleSwap on the GitHub repository.

5. The Oracle's Incentives

The only way OracleSwap ETH is transferred out of a user's account is to either have the private key of a subcontract account with ETH in its margin or balances, or via a debit applied at settlement per a negative PNL. The PNL is determined by the initial and ending prices reported by the Oracle to the Oracle Contract. An investor with insufficient margin precludes future exposure, but does not affect the liability from the prior week.

The only attack vector involves the Oracle, where an evil oracle conspiring with one of the players by posting fraudulent prices, such as reporting a true -3% return as a +7% return, effectively stealing 10%. Presumably, a cunning evil oracle would remove the middleman and be that counterparty using a different account it controlled; an evil oracle in practice is an evil oracle/counterparty sock-puppet.

Simplicity is essential for good contracts because analyzing a game becomes exponentially complex as participants are added. For example, to specify a general game in which *n* players each make a decision to either cheat, play honestly, or free-ride on whatever everyone else is doing, 3^n scenarios arise (free-riding is the majority strategy for most decentralized systems). With only the Oracle as a potential cheater (n=1), that means we must look at just its two options (it can't free ride).

The key to an honest oracle is that it plays a simple repeated game where dishonest play is easily observed.[19] Repeated payoffs are essential in moving the game-theoretic equilibrium from bad to good, as demonstrated by the different equilibrium for the prisoner's dilemma when it moves from a one-period game to a sequence of one-period games. With weekly settlement the Oracle's incentive structure is like that in the iterated prisoner's dilemma, where a multi-period game moves the dominant strategy from the suboptimal equilibrium where both parties maximize their one-period payoff by not cooperating, to the optimal equilibrium where parties maximize their payoffs by cooperating. In evolutionary biology, this shows up in *reciprocal altruism* and illustrates how cooperation emerges out of long-run self-interest, as cooperating players out-compete non-cooperators.

5.1 Cost-Benefit Analysis of an Oracle Cheat

The game theoretic concept of *common knowledge* is that two players both know some fact, they both know they both know that fact, and both know they both know they both know the fact, *ad infinitum*. Using this reasoning, most rational players will always burn when cheated, where a burn simply means that the burner's PNL debit will be sent to a burn address rather than its counterparty. By constraining players to either *burn*, *default* or *continue* in the settlement period (no cancel), we increase the probability of punishment without lowering the burn fee to trivial levels that might encourage whimsical burns. Encouraging cheated parties to burn their PNL lowers the cheat payoff to levels that make honesty obviously preferable to cheating without relying on exponential growth in contract use.

The reasoning is explained in full in the Technical Appendix, but the gist of why burning is the optimal response to a cheat is the following. If a player sees a fraudulent price posted they have 4 hours to either burn their PNL or continue. The cost of burning is the same regardless of their impending PNL credit/debit, ¼ of your RM that is sent to a burn address (and thus not to the evil oracle or its co-conspirators). If they do not burn, the evil oracle will rationally infer, via common knowledge reasoning, he can cheat them again next period by more than ¼ RM in expected value. Most players will have significant Excess Margin to avoid having to interrupt their activities to avoid default, as for example at MakerDAO 85% of CDP account value is over-collateralized by more than 100%.[20] For investors with more than ¼ RM in their margin after a fraudulent PNL is assessed to their margin, it will be cheaper to to burn their payoff and prevent the cheating oracle from receiving ETH as opposed to either continuing or defaulting.

Let us benchmark our payoffs via the Total OracleSwap RM, which is the sum of all the various subcontract RMs across all the AssetSwap Contracts. We will denote this RM*. An AssetSwap Contract with sufficient volume to hack will have many users, and many of those who might be co-conspirators will simply be accidentally aligned with the cheating Oracle's position side. This cuts down the RM hackable by at least ½. When combined with the fact that most players will burn rather than continue—say ¾–this cuts Oracle/Alice's final payoff to ½·(1- ¾) or ⅛ RM*.

There is another value to cheating via the fact that there is no need to report fake prices when the oracle is making money that week, and potentially seeing that burned. This strategy, played optimally, has an expected value of around a 1-standard deviation PNL, which is about ⅓ RM for each subcontract. As the

[19] For example, see Robert Axelrod's *Evolution of Cooperation* (1982), which highlights the value of repeated games in creating cooperative outcomes.
[20] MakerDAO on 5/11/19 had 85% of their CDP value collateralized over 100%.

Oracle will probably be only ½ of one side, this is ⅙·RM*. In total a cheating Oracle can expect to garner about ½ RM* (⅛+⅙).

If we assume subcontracts roll over every 2 months, then the Oracle expects 6 closes, and as each subcontract generates 1.5% RM in Oracle revenue, the annual dividend on RM* is 9%. Using the Gordon dividend discount model, a discount rate of 9% and a growth rate of 4.5% imply a present value of 2·RM*.[21] Thus under very modest assumptions, an evil oracle would find it rational to use their Oracle revenue to further other evil schemes, because even evil people prefer more money to less (2>½).

Initially, the Oracle could more easily dominate a side—long or short—which would increase the evil Oracle's cheat payoff. Yet initially the expected growth rate is much higher, and as a trusted oracle has many uses, we should expect it to be an order of magnitude above the stead-state growth rate of 4.5%. In the long-run, however, the decline in the expected growth rate is compensated by the fact that a successful Oracle would have a difficult time dominating a side, and very little ability to exercise the 'walk-away' option, as the positions it took would be uncorrelated. A seasoned OracleSwap oracle will have perfected the automated scripts that attend to its contracts, so its annuity stream would take virtually no effort or capital, a valuable asset worth preserving. This is a long-run incentive compatible mechanism.

5.2 Oracle Price Sources and Auditing

OracleSwap is based on transparent reporting so that casual consumers can easily see for themselves whether the Oracle is reporting honestly. Blockchain queries for burns and reported prices are provided by the OracleSwap.co front-end, reading from the immutable blockchain event logs generated at every Oracle Price update.

We will target 4 PM New York City time using data derived from US and European Bitwise-approved exchanges.[22] Auditors should be aware that historical crypto databases use different intraday closing times. Winkdex records 4 PM prices for ETH and BTC and so currently provides a good reference. Comparable crypto prices are usually within 0.1% of each other at any particular point in time. The SPX closing price refers to 4 PM, but as an index it is finalized around 4:10 PM. The Oracle will pull prices around 4:11 PM, but providers have been known to change their latency policy unexpectedly (e.g., from 1 to 10 minutes), which might create minor (~0.1%) difference with the historical close that day.

Burns show someone cared enough to pay to punish the Oracle. A truly fraudulent price report by the Oracle should ruin the Oracle, as no rational investor should invest in a contract relying on a cheating oracle. As such allegations are uncensorable, however, further investigation is recommended as the burner could be mistaken or mischievous.

5.3 Parasites and Mimics

The Oracle Contract stores closing prices that are accessible via a private function that only its OracleSwap Contract can access inside the Ethereum Virtual Machine (EVM). This prevents parasite contracts from using the Oracle Contract while not paying the Oracle.[23] While all state variable data in

[21] PV=dividend/(discount rate – growth rate).

[22] See the Bitwise Presentation to the US SEC, dated 3/20/19. Their approved exchanges are Binance, Bitfinex, Kraken, Bitstamp, Coinbase, BitFlyer, Gemini, itBit, Bittrex, and Poloniex

[23] While logically possible to back out our Oracle's returns on-chain, this would require the parasite to update its parameters frequently and create a very complicated contract with owner discretion.

this contract is visible outside the EVM, the data recorded by the Oracle Contract is trivial to discover elsewhere anyway. The oracle problem is not technical, it is strategic; data is only valuable in the context of a smart contract that is forced to use these prices. A parasite contract user would have to trust the parasite contract's oracle to copy this information truthfully outside the EVM, which is the main problem, and cost, of an oracle.

The contract's source code is fully readable and available for all to copy, modify and deploy as anyone sees fit. Simply copying the code is easy, but there is also the cost of creating automated scripts for updating the Oracle Contract, executing the daily and weekly functions, adjusting basis rates, error-correction, etc. Yet the cost of doing this is relatively low, making it feasible and affordable for many enterprising crypto programmers, and competition is the ultimate regulator.

There are downsides to the low cost of entry typical to all thriving markets. Without a significant fixed cost of entry, a hacker has little to lose by trying to abuse any misplaced trust in look-alike contracts. A few subtle code changes can give them a backdoor to contract ETH balances. Therefore, while we encourage imitators, we also advise users to be wary of them. Users of emulators should compare their code to the original OracleSwap code and understand each discrepancy.

6. Conclusion

Most blockchain development is oriented towards two extremes. On one hand you have protocols with broad generality and decentralized consensus or governance mechanisms like bitcoin itself, on the other hand you have firms emulating traditional institutions only they transact with crypto as well as fiat currency. The standard institutions invariably violate at least one of the non-negotiable crypto principles: transparency, immutability, anonymity, uncensorability, or custody. Generalized protocol Dapps create large consensus costs, and minimizing transaction costs is essential for creating viable markets.[24] Additional parties and tokens increase costs because they all need to be paid, create complicated collusion scenarios, and delay payouts.

OracleSwap's open source set of contracts should make it easy for individuals to create substitutes and extensions. Institutions like Goldman Sachs, Consensys, Facebook, or even Dapps run by firms like Kyber, Chainlink, or Maker cannot support these contracts as they have clear attack surfaces for regulators. OracleSwap's Oracle retains its anonymity on the blockchain making it immune to censorship, yet via cryptographic custody of the contracts it services can develop a reputation.

A set of competing OracleSwap contracts is more efficient than generalized oracles designed for unspecified contracts, or generalized trading protocols designed for unspecified oracles. Successful coinshifters have found that making 1% on a transaction is better than making 100% on a cheat because of the present value of future transactions. This is where reputation is difficult to measure as there are no clear connections between cross-chain transactions that can prove a cheat, so reputation is inferred via the noisy data one reads in chatrooms. Prices used to generate OracleSwap's PNLs are recorded in the Oracle contract's immutable, indexed event logs and need no parochial adjustment to compare to cash prices.

Funds cannot be frozen as an incapacitated oracle would allow users to get their margins back without a loss. Its reporting is easy to monitor, and the contract's design incents players to burn an evil Oracle's heist if it tries to cheat, trivializing an exit scam payoff. A player of this contract merely needs to trust the Oracle can and will act in its self-interest, which comes from the censorship-proof blockchain and an incentive compatible contract design.

Investors can create asset exposure that is difficult to get elsewhere while staying on the blockchain. By capping the weekly PNL at the required margin, investors need only attend to their positions once a week and maintain a long-term position. Forward starting prices help both the LP and taker, in that they remove any need to invest time and money into continuous market monitoring and low-latency access. Portfolio margining allows LPs to generate an attractive return while giving investors competitive rates.

There are significant gains from trade available in finance because our options are restricted by institutions protecting monopoly power under the pretext of protecting the public. Offering financial asset protection is a good business or government service; mandating protection is a racket. Ethereum financial contracts are alluring because many financial contracts are the application of straightforward rules to objective and widely disseminated asset prices. OracleSwap is a functioning contract, but the source code and documentation allow for people to inspect and extend.

[24] As Nick Szabo tweeted: blockchain governance generally comes in three varieties: Lord of the Flies, lawyers, or ruthlessly minimized.