On Decentralizing Prediction Markets and Order Books

Jeremy Clark[1], Joseph Bonneau[2], Edward W. Felten[2], Joshua A. Kroll[2], Andrew Miller[3], and Arvind Narayanan[2]

[1] Concordia University
[2] Princeton University
[3] University of Maryland

Abstract. We propose techniques for decentralizing prediction markets and order books, utilizing Bitcoin's security model and consensus mechanism. Decentralization of prediction markets offers several key advantages over a centralized market: no single entity governs over the market, all transactions are transparent in the block chain, and anybody can participate pseudonymously to either open a new market or place bets in an existing one. We provide trust agility: each market has its own specified arbiter and users can choose to interact in markets that rely on the arbiters they trust. We also provide a transparent, decentralized order book that enables order execution on the block chain in the presence of potentially malicious miners.

1 Introductory Remarks

Bitcoin has demonstrated that achieving consensus in a decentralized network is practical. This has stimulated research on applying Bitcoin-esque consensus mechanisms to new applications (*e.g.,* DNS through Namecoin,[4] timestamping through CommitCoin [10], and smart contracts through Ethereum[5]). In this paper, we consider application of Bitcoin's principles to prediction markets.

A prediction market (PM) enables forecasts about uncertain future events to be forged into financial instruments that can be traded (bought, sold, shorted, *etc.*) until the uncertainty of the event is resolved. In several common forecasting scenarios, PMs have demonstrated lower error than polls, expert opinions, and statistical inference [2]. Thus an open and transparent PM not only serves its traders, it serves any stakeholder in the outcome by providing useful forecasting information through prices.

Whenever discussing the application of Bitcoin to a new technology or service, its important to distinguish exactly what is meant. For example, a "Bitcoin-based prediction market" could mean at least three different things: (1) adding Bitcoin-related contracts (*e.g.,* the future Bitcoin/USD exchange rate) to a traditional centralized PM, (2) converting the underlying currency of a centralized prediction market to Bitcoin, or (3) applying the design principles of Bitcoin to decentralize the functionality and governance of a PM.

Of the three interpretations, approach (1) is not a research contribution. Approach (2) inherits most of the properties of a traditional PM: Opening markets for new future events is subject to a commitment by the PM host to determine the outcome, virtually any trading rules can be implemented, and trade settlement and clearing can be automated if money is held in trading accounts. In addition, by denominating the PM in Bitcoin, approach (2) enables easy electronic deposits and withdrawals from trading accounts, and can add a level of anonymity. An example of approach (2) is Predictious.[6]

This set of properties is a desirable starting point but we see several ways it can be improved through approach (3). Thus, our contribution is a novel PM design that enables:

- **A Decentralized Clearing/Settlement Service.** Fully automated settlement and clearing of trades without escrowing funds to a trusted straight through processor (STP).
- **A Decentralized Order Matching Service.** Fully automated matching of orders in a built-in call market, plus full support for external centralized exchanges.

[4] `http://namecoin.info`
[5] `http://www.ethereum.org`
[6] `https://www.predictious.com`

- **Self-Organized Markets.** Any participant can solicit forecasts on any event by arranging for any entity (or group of entities) to arbitrate the final payout based on the event's outcome.
- **Agile Arbitration.** Anyone can serve as an arbiter, and arbiters only need to sign two transactions (an agreement to serve and a declaration of an outcome) keeping the barrier to entry low. Traders can choose to participate in markets with arbiters they trust. Our analogue of Bitcoin miners can also arbitrate.
- **Transparency by Design.** All trades, open markets, and arbitrated outcomes are reflected in a public ledger akin to Bitcoin's block chain.
- **Flexible Fees.** Fees paid to various parties can be configured on a per-market basis, with levels driven by market conditions (*e.g.,* the minimum to incentivize correct behavior).
- **Resilience.** Disruption to sets of participants will not disrupt the operations of the PM.
- **Theft Resistance.** Like Bitcoin, currency and PM shares are held by the traders, and no transfers are possible without the holder's digital signature. However like Bitcoin, users must protect their private keys and understand the risks of keeping money on an exchange service.
- **Pseudonymous Transactions.** Like Bitcoin, holders of funds and shares are only identified with a pseudonymous public key, and any individual can hold an arbitrary number of keys.

2 Preliminaries and Related Work

2.1 Prediction Markets

A PM enables participants to trade financial shares tied to the outcome of a specified future event. For example, if Alice, Bob, and Charlie are running for president, a share in 'the outcome of Alice winning' might entitle its holder to \$1 if Alice wins and \$0 if she does not. If the participants believed Alice to have a 60% chance of winning, the share would have an expected value of \$0.60. In the opposite direction, if Bob and Charlie are trading at \$0.30 and \$0.10 respectively, the market on aggregate believes their likelihood of winning to be 30% and 10%. One of the most useful innovations of PMs is the intuitiveness of this pricing function [24]. Amateur traders and market observers can quickly assess current market belief, as well as monitor how forecasts change over time.

The economic literature provides evidence that PMs can forecast certain types of events more accurately than methods that do not use financial incentives, such as polls (see [2] for an authoritative summary). They have been deployed internally by organizations such as the US Department of Defense, Microsoft, Google, IBM, and Intel, to forecast things like national security threats, natural disasters, and product development time and cost [2].

The literature on PMs tends to focus on topics orthogonal to how PMs are technically deployed, such as market scoring rules for market makers [13,9], accuracy of forecasts [23], and the relationship between share price and market belief [24].

Concurrently with the review of our paper, a decentralized PM called Truthcoin[7] was independently proposed. It is also a Bitcoin-based design, however it focuses on determining a voting mechanism that incentivizes currency holders to judge the outcome of all events. We argue for designated arbitration in Section 5.1. Additionally, our approach does not use a market maker and is based on asset trading through a decentralized order book.

2.2 Bitcoin

Bitcoin is a digital currency, minted and maintained by a decentralized peer-to-peer network [18]. Bitcoin transactions require the digital signature of the account holder to prevent theft, and every transaction is published in an append-only hash chain, called the block chain. The block chain can be extended with new transactions by any participant, and such participants may obtain newly minted Bitcoin currency (XBT or BTC) and/or tips from the transactions for performing this function. However extensions to the block chain require a proof-of-work that rate-limits the process (to approximately an extension every ten minutes) that enables a steady inflation rate, ample competition among participants to extend the block chain, and adequate time to obtain and verify the history of the block chain for new participants.

[7] `https://github.com/psztorc/Truthcoin`

Block Chain Consensus Protocol. Informally, the proof-of-work protocol in Bitcoin is intended to maintain the following two essential properties about the block chain:

- Every party eventually agrees on the order and correctness of transactions in the block chain.
- Any party can publish a transaction (for a fee), which will then be verified and, if valid, included in the block chain within a small bounded delay.

The Bitcoin consensus protocol makes use of a message-diffusion communication primitive, which is much weaker than the standard network model of point-to-point channels. Informally, the message-diffusion primitive guarantees that any honest party can publish a message, which will be delivered to every other honest party within some bounded time; however, the recipient of a message does not learn the identity of the sender (in fact, parties do not have any prearranged identifiers whatsoever), and dishonest parties might deliver a message to some parties but not others. The Bitcoin protocol additionally relies on an assumption about the distribution of computational resources among participants: specifically, that the adversary wields less computational power (specifically the rate at which the hash function SHA-256 can be computed) than the combined contribution of the honest participants.

The protocol is as follows: each party maintains a view of the *best* sequence of messages it has seen, where *best* (which we will define more clearly below) means that the sequence appears to have been generated using the largest amount of computational effort. Each party additionally maintains a list of messages that are not yet contained in the log, but are valid (*i.e.,* they carry an appropriate fee, have valid signatures, and do not conflict with or *double-spend* previous transactions). Each party attempts to find a solution to a hash-based proof-of-work puzzle [12,3,4], where the puzzle challenge consists of a commitment ($\mathsf{MerkleRoot}$) to the sequence of new valid transactions, as well as the last commitment of the *best* sequence ($\mathsf{PrevBest}$). A solution to this puzzle consists of any nonce value, such that the hash of $\mathcal{H}(\mathsf{Nonce}||\mathsf{PrevBest}||\mathsf{MerkleRoot}) < 2^{-d}$, where d is a difficulty parameter (other metadata is elided for clarity). If the hash function is modelled as a random oracle, then the best approach is to brute force by trying nonces at random. When a party finds a solution, the party publishes the solution (and the transactions) to every party; the *best* block chain is now extended by one block.

This protocol achieves consensus because the difficulty is set high enough, relative to the rate of message diffusion on the network, that eventually the honest parties converge to a single *best* chain. With high probability, the attacker has insufficient computational power to supplant the *best* chain after sufficient time has passed. Additionally, the attacker cannot prevent honest parties from contributing some blocks to the chain, and these blocks will contain all the valid published messages [15].

2.3 Altcoins and Bitcoin Enhancements

The term Bitcoin can refer to both the protocol and a single, widely-used, global instance of the protocol. While Bitcoin is far and away the most popular instance of this protocol, there are numerous other instances, "altcoins" such as Litecoin and Namecoin, which replicate large portions of the protocol while making certain tweaks, the details of which are not relevant here.

There are thus essentially three avenues for the deployment of any proposed modification, such as ours:

1. **As a modification to Bitcoin.** This would require a significant majority of the current participants of Bitcoin to agree that such a modification is desirable and safe, in which case everyone could update their software and switch to the modified system. There is a hierarchy to the synchronization required for such a change (*e.g.,* soft fork vs hard fork). Examples of modifications to the Bitcoin protocol deployed this way include bug fixes and a small number of new features (*e.g.,* M-of-N multi-signatures).
2. **As a separate altcoin.** An instance of the modified protocol can run as an entirely separate entity to Bitcoin. Such a separate instance may nonetheless compete with Bitcoin for users, resources, and attention, possibly leading to fragmentation. Since the resilience against an attacker is defined by the amount of computational power contributed by the users in an instance, fragmented systems are weaker than a cohesive system. Merged-mining (where computational work can be shared among two instances) is also possible, although this also reduces the cost of an equally-powerful attack.

3. **As an overlay using the current system.** This is the approach taken by colored coins, and we elaborate on it in Section 5.2.

2.4 Cryptographic Warranties

The public nature of the Bitcoin log enables the use of *warranties* in protocols in which at least one party has a public reputation and in which all necessary messages are transmitted using the public log. A warranty is a signed statement by an ostensibly reputable party specifying the exact terms of the protocol in a way which will be easily verifiable by any third party after the fact. If the warrantor fails to behave correctly, the warranty will exist as proof of misbehaviour and can be published by the aggrieved party to damage the warrantor's reputation (encapsulated in the key used to sign warranties). This technique was recently used in the design of Mixcoin to make it apparent if mixes steal funds from clients [5]. We can use the concept of warranties to build a number of protocols, assuming there will be parties with a legitimate business interest in honest behaviour.

2.5 Cryptographic Markets

Applying cryptographic techniques to financial markets has been examined in several papers [11,21,26,22]. The design goals and techniques are however quite different, focusing on how certain aspects of the market can be done under encryption (by computing on encrypted data) while generating proofs of correct behaviour [11]. One significant proposal enables continuous trading on an encrypted order book, with support for limit and market orders [21], while later work explored how more general trading instructions could be cryptographically executed on an exchange similar to a crossing network [22].

The focus of this related work is confidentiality: protecting trading information from a centralized exchange (and from other participants [26]). By contrast, our focus is decentralizing the exchange while having public trades (albeit by pseudonymous participants). This difference is akin to the difference between anonymous e-cash [8] and Bitcoin. An interesting area for future work would be combining both approaches, similar to how Zerocoin [17] combined them for digital currency.

Our work can be further distinguished from the cryptographic trading literature by our focus on prediction markets, not explicitly considered there, and by the fact that the instruments being traded in our work are digital, allowing us to address clearing and settlement in addition to trading.

3 Design Goals and Threat Model

All deployed PMs that we are aware of are run by a trusted central authority, who holds the money and the shares backing the market in escrow (*i.e.,* in street name) while providing an electronic communication network capable of matching orders and posting the bid/ask of unmatched orders. Orders that are executed are cleared and settled by the authority, and when the events being forecast are realized, the authority declares the outcome and settles the accounts.

The core component of our research is a decentralized clearing and settlement process (bottom of Figure 1). This component allows markets for specific forecasts to be created and closed, and allows trading to occur within the market. Once a trade is arranged (top of Figure 1), whether through our decentralized order matching service, an external exchange service, or privately between traders, the details of the transaction are broadcast to a P2P network. The network checks that the traders have the requisite digital money and digital shares to execute the trade (*clearing*), and then updates the block chain to turn over signing authority on the money and shares to their new owners (*settling*).

Design Goals. Bitcoin is highly decentralized with only two types of participants: users and miners. It would be ideal to design a PM to reach this same level of decentralization, however for certain tasks, its beneficial to designate external entities to fulfill critical, but limited, roles. In this way, our design is closer to

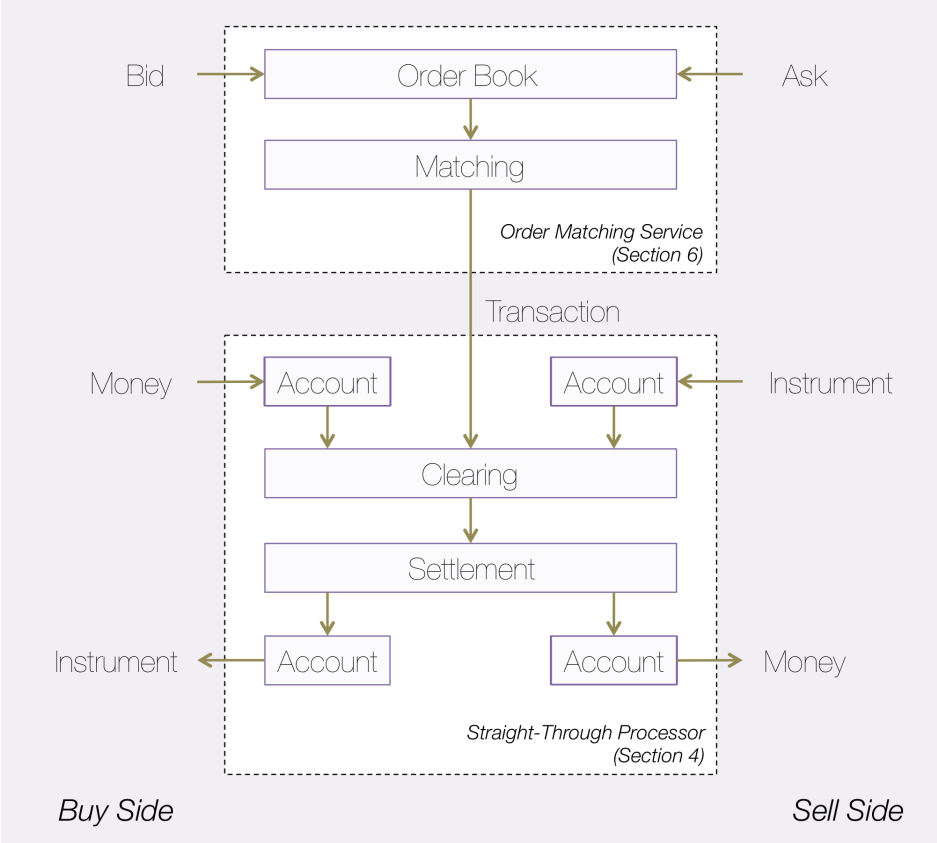


Fig. 1. PMs consist of two main components. An order-driven market matches buys and sellers that are willing to trade, and a processor that enables the execution of valid trades (regardless of how they are arranged) in a timely fashion.

Bittorrent where the users do the bulk of the work themselves, but trackers are required to match downloaders with uploaders, and indexing websites facilitate finding the trackers for specific files.

Our design *decentralizes* order matching, settling, and clearing to a community of miners, where authority is held in proportion to computational ability. We argue that trusted authorities, called arbiters, are better suited to declare the outcome of an event than through community consensus, however users can adaptively move between markets to ensure they only rely on arbiters they trust (*trust agility*). Arbitration is also designed to have a *minimal barrier to entry*: arbiters do not have to run an entire market, they only have to send one transaction to agree to govern over declaring the outcome of the market and, later, one transaction to endorse the outcome.

Recall that prediction markets are useful to non-participants as price changes communicate information about the forecast of future events. Like Bitcoin, all transactions are published in a public block chain providing *transparency* and ensuring the visibility of all transactional data for study. Finally, accounts are identified by public keys providing *pseudonymity* to participants. However, as with Bitcoin, the block chain is a rich dataset that may be useful for deanonymization [1,16,19,20].

Threat Model. We design for three main types of participants: traders, miners, and arbiters. Traders buy and sell shares in outcomes, miners maintain a transcript of all transactions as they do in Bitcoin, and arbiters are entities authorized to declare the outcome of an event.

We assume the participation of an honest majority (according to computational power) of miners at all times. We assume miners receive transactions from other participants over standard network communication and thus are privy to IP addresses. Participants seeking anonymity should communicate over, *e.g.,* Tor. Both of these assumptions underlie Bitcoin.

We assume that traders may be malicious and attempt to steal shares, double-spend currency to obtain shares, and redeem shares in outcomes other than the declared outcome. We design to prevent such behaviour. At a meta-protocol level, traders may profit from inside information: in prediction markets, this is considered a feature and not a bug as large orders move the price which indirectly conveys the private information publicly. Traders may also try to manipulate the price with large orders, to skew how observers interpret the probability of the outcome. Once again, prevention is outside the threat model, but we note that such strategies are costly to sustain and typically self-correcting through market forces.

Arbiters pose the most significant threat. A malicious arbiter can profit directly from misbehaviour, for example, by obtaining cheap shares in a low probability outcome and falsely declaring that outcome the winner. Arbiters may also be simply unreliable, and not declare outcomes in a timely fashion or at all. While these threats cannot be eliminated, we use trust agility and reputation to control and mitigate them.

4 Decentralized Clearing and Settlement

To convey the core functionality of our design, we first present a skeleton construction (Construction 1) of its clearing and settlement service. Since it is fully automated, it can be referred to as a straight-through processor (STP). In Section 5, we elaborate on a number of ways this basic construction can be extended to offer increased transparency, increased decentralization, and more sophisticated market structures.

The skeleton construction is presented as an altcoin. Altcoins replicate the functionality of Bitcoin while modifying and extending it in certain ways. Here, we replicate without modification the Bitcoin mechanisms for minting currency, transferring currency, and establishing a reliable block chain of transactions, while adding new functionality for implementing the PM. In Section 5.2, we consider alternative methods for deployment that do not require a full-fledged altcoin, either by using 'casino chip' tokens generated from XBT or by implementing it within Bitcoin itself.

For convenience in discussing our design, we label the altcoin's internal currency—we choose XFT, where FT abbreviates 'future events.'

4.1 Basic Protocol

We begin by extending an altcoin with five additional functions: OpenMarket, BuyCompleteSets, SellCompleteSets, Exchange, and CloseMarket, as specified in Construction 1.

While OpenMarket could be initiated by anyone, the specified arbiter must demonstrate her intention to serve as the arbiter for the forecast, which she does by signing the OpenMarket transaction. Multiple markets may be opened for the same event. Traders should choose markets to participate in based on the terms of the contracts and the use of reputable arbiters, who are trusted to adjudicate the outcome fairly and promptly.

Shares come into existence through BuyCompleteSets. Given that holding one share in each outcome will guarantee a payout invariant to which share is the actual outcome, we allow clients to divest without waiting for the outcome through SellCompleteSets. SellCompleteSets enables a direct method for short-selling shares, without the traditional involvement of a lender and borrower: the speculator buys a portfolio, sells the share she anticipates will drop in price, and later purchases it for a lower price, keeping the difference and re-completing the portfolio for conversion back into a Bitcoin. Note that while SellCompleteSets results in newly minted currency, the supply of currency can never be increased through the use of SellCompleteSets because SellCompleteSets can only occur if a BuyCompleteSets transaction occurred in the past, which eliminated the equivalent amount of currency.

Exchange allows shares to be moved between accounts the same way Bitcoins are transferred, plus it allows shares to be sold for a specified amount of Bitcoin. We provide the exact structure of Exchange in Section 4.3.

An altcoin is extended with the following functions:

- OpenMarket(MarketID, Contract, {ShareID$_i$}, Price, Arbiter)
  A transaction initiated and signed by an arbiter to create a new prediction market with a unique identifier MarketID. Contract contains a natural language description of the future event, the set of possible outcomes, and the criteria used to determine the outcome. Each outcome is assigned a ShareID that is unique within the MarketID ({ShareID$_i$} denotes the complete set). Price specifies the forward price or payout, in XFT, of a share in the determined outcome. Arbiter specifies the public signing key of the arbiter authorized to initiate CloseMarket (see below), and they sign OpenMarket.

- BuyCompleteSets(MarketID, Volume)
  A transaction initiated and signed by a user agent to obtain complete sets of shares for the market MarketID at the Price defined in its OpenMarket. Upon approval and insertion into the block chain, the transaction debits the user of Price $\times$ Volume XFT and issues Volume newly minted complete sets {ShareID$_i$} to the user's key. The debited XFT is eliminated from circulation.

- SellCompleteSets(MarketID, Volume)
  A transaction initiated and signed by a user agent to sell compete sets of shares for the market MarketID at Price. Upon approval and insertion into the block chain, the transaction debits the user Volume complete sets and issues Volume $\times$ Price of newly minted XFT. The debited shares are eliminated from circulation.

- Exchange(MarketID, ShareID, Sender, Volume, Receiver, Payment)
  A transaction initiated and signed by a Sender to transfer a Volume of the specified shares, ShareID in MarketID, to a Receiver for Payment in XFT. Payment may be 0; otherwise the receiver must also sign the transaction.

- CloseMarket(MarketID, ShareID$_w$)
  A transaction initiated and signed by an arbiter to declare the ID of the winning outcome: ShareID$_w$. The arbiter must be the one specified in the corresponding OpenMarket transaction for MarketID. Upon approval and insertion into the block chain, ShareID$_w$ will be converted to Price with newly minted XFT, while all other shares are eliminated from circulation.

Construction 1: Skeleton construction of a decentralized clearing and settling straight-through processor

With CloseMarket, the arbiter can close the market at any time with any outcome. There is no validation by the miners extending the block chain that the market was closed 'correctly.' Aside from a misbehaving arbiter, there is the possibility of the arbiter not closing the market at all, whether maliciously or for technical reasons like a lost signing key. We discuss design alternatives to mitigate these issues in Section 5.1.

Shares can be considered fungible assets that are always in zero net supply. Every share in existence originated from a BuyCompleteSets, which also created a single share in each other outcome and eliminated 1 XFT from circulation. For every share redeemed for 1 XFT through a CloseMarket, it can only be redeemed once and no other share created in the same BuyCompleteSets can be redeemed. Every share redeemed, as part of a complete set, through SellCompleteSets, can likewise be only redeemed once and the set is redeemed for exactly the 1 XFT eliminated to create it (or an equivalent set of fungible shares).

4.2 Working Examples

To illustrate how our asset-based PM functions, consider the following examples of actions taken by an investor who believes a share is undervalued or overvalued. To ease the explanation, we assume an order book exists for indexing the current bid and ask prices of each share. We have not yet discussed how to realize such an order book, but will do so in Section 6.

	Initial Investment	Potential Gain	Potential Loss
Buy and Hold a Stock	S	∞	S
Buy a Call Option	O	∞	O
Buy and Hold a PM Share	S	$P - S$	S
Buy PM Portfolio and Retain Share	P	$P - S$	S
Short Sell a Stock	none	S	∞
Buy a Put Option	O	K	O
Short Sell a PM Share	none	S	$P - S$
Buy PM Portfolio and Release Share	P	S	$P - S$

Table 1. Summary of initial investments, potential gains, and potential losses for various ways of going long (first half of table) and going short (bottom half of the table). Assume a complete portfolio sells for P, a given share sells for $0 \leq S \leq P$ (and for comparison, S denotes the price of a stock), a put option has a strike price $0 \leq K \leq S$, and either option sells for O. If S goes to ∞, this is the best case for a long position and the worst case for a short, and conversely for S going to zero.

Going Long. It is the group stage of the 2010 World Cup, and Alice believes the Netherlands are undervalued to win the cup at {`Bid: 0.08, Ask: 0.16, Last: 0.09`}. Alice places an order with a bid of 0.10 and it is filled. As Netherlands makes it further in the tournament without being eliminated, its share price increases. Before the World Cup final, it is not favored to win, trading at {`Bid: 0.39, Ask: 0.41, Last: 0.40`}. Alice believes this is a fair evaluation and wants to realize her earnings. She sells her share for 0.39 and Spain defeats the Netherlands. Assuming no transaction fees (see Section 5.3), Alice's initial investment of 0.10 XFT earned her 0.39 XFT.

Alternatively, it may be the case that Alice is an early investor in the market and there is not sufficient liquidity to buy a share in the Netherlands for a fair price, or even at all. In this case, Alice can use BuyCompleteSets to purchase a complete portfolio and then sell shares in every options except the Netherlands. If the sale of these shares nets her 0.90 on her 1 XFT investment, her position is identical to purchasing a share in the Netherlands for 0.10. However this does force her to place an evaluation on shares she has no interest in, which is a drawback relative to a centralized PM, where the PM can take on risk by selling to Alice without necessarily having the counter bets to offset this risk. However in both cases (centralized and decentralized), a market cannot form until some entity provides an initial offer price for each share. Any entity can serve this role, potentially structuring the initial prices to leave a profit margin (*i.e.,* a vigorish).

This example illustrates how an informed forecaster can enter the market when she believes prices do not reflect of their true value, and freely leave if the price adjusts in her favour. In particular, this means that buying a share in an outcome is only a statement that it is undervalued, not a statement that the forecaster believes that share will necessarily translate into the final outcome. Forecasters have an incentive to correct market prices for all shares, including low probability outcomes.

Going Short. Alice believes candidate Thomas Fredrick is over-valued at {`Bid: 0.24, Ask: 0.27, Last: 0.26`} for becoming his political party's presidential nominee. She purchases a portfolio using BuyCompleteSets for 1 XFT and immediate sells the share in Fredrick for 0.24 XFT. Two weeks later, Fredrick drops out of the race, and his share price plummets to an Ask of 0.001. As long as Fredrick stays out of the race, Alice owns a share in each other option and can expect 1 XFT when the primary finishes in two months. However she will earn nothing (not even interest) between now and then. Alice purchases a share in Fredrick for 0.001 to complete her portfolio and uses SellCompleteSets to receive 1 XFT immediately. Assuming no transaction fees, Alice initial investment of 1 XFT earned her $1 + 0.24 - 0.001 = 1.239$ XFT.

Alternatively, Alice can use the traditional method of short selling by selling a borrowed share in Fredrick, buying it back at a lower price, and returning it to the lender. However lenders are traditionally hesitant to

lend shares they know will be immediately sold, as this puts downward pressure on the price of what they own, so they typically charge large fees as compensation. The asset model (BuyCompleteSets and SellCompleteSets) also enables Alice to avoid the logistics and counterparty risks associated with borrowing and lending shares.

Arbitrage. Alice notices that the bid price for each share in a portfolio adds up to 1.05 XFT. She immediate buys a portfolio for 1 XFT using BuyCompleteSets and sells each share at the bid price, gaining 1.05 XFT for a 0.05 XFT profit. Similarly, if Alice notices that the ask price adds up to 0.98 XFT, she spends 0.98 XFT to purchase a set of shares and uses SellCompleteSets to sell them for 1 XFT netting 0.02 XFT. While we cannot guarantee that share prices (*i.e.,* last sale price) will add up to 1 XFT, BuyCompleteSets and SellCompleteSets ensure that in the long run, 1 XFT will remain within the the cumulative bid-ask spread of a complete portfolio. However the spread will not necessarily be narrow (as with other traded instruments, popularity, and thus liquidity, leads to tighter spreads).

Summary. Trading PM shares has certain differences from from trading stocks or derivatives. Table 1 shows the initial investment, potential gains, and potential losses of the PM trading methods along with traditional methods of buying a stock, short selling a stock, and buying a stock derivative. The key takeaway from Table 1 is that both gains and losses are bounded with PM shares. The second takeaway is that dealing with a full portfolio, as opposed to individual shares, requires a higher initial investment. For this reason, traders will always prefer to deal directly with shares, but may decide to deal with portfolios when trading is illiquid or if borrowing shares are hard to arrange.

4.3 Microstructure of Exchange

The central and most complex transaction is Exchange. By virtue of being an altcoin, we assumed a transaction existed for users to send XFT to other users. In fact, Exchange can subsume such a transaction plus allow shares to be exchanged for XFT and to allow winning shares to be converted into XFT. BuyCompleteSets and SellCompleteSets are also special cases of Exchange.

Figure 2 shows a simplified Exchange transaction. Structurally it is similar to a Bitcoin transaction. Exchange has a unique identifier (which in actuality is a truncated hash of the transaction), a set of inputs, and a set of outputs. Inputs refer to the output of a previous transaction. Inputs and outputs are one of four types: XFT, shares (denoted $M_{ID} : S_{ID}$ for MarketID and ShareID) in an active market, shares from a closed market, and portfolios.

As in Bitcoin, inputs must completely spend the output they refer to. However when the inputs are summed together, they can be split up into any number of outputs of any size as long as two conditions are met: (1) all outputs are larger than the smallest transactional unit, which is 10^{-8} XFT, 1 share, and 1 portfolio (preventing shares from being divisible is a design decision aimed at simplicity for end users and is not an inherent constraint) and (2) the sum of the inputs exceeds the sum of the outputs (after necessary conversions between types). If the sum of the inputs exceeds the outputs, the remainder is retained by the miner that adds the transaction in the block chain as a fee (in Figure 2, the miner receives a tip of 0.001 XFT). Fees are discussed in Section 5.3.

Every input of XFT eventually traces back to the creation of it through mining (in the altcoin model) or a conversion from a winning share. Every share eventually traces back to a previous BuyCompleteSets transaction. Shares from a market that has closed can be spent as if they were XFT, except that a reference to the CloseMarket transaction is included along with the previous transaction that obtained the share. The miners traverse from the current transaction to CloseMarket to check that the market has closed, and then from CloseMarket to OpenMarket to learn the Price of a winning share. Technically there is no limitation to trading shares as shares after the market has closed or including non-winning shares valued at 0 XFT as inputs to transactions.

Although not shown in Figure 2, portfolios can also be inputs and outputs with reference to the OpenMarket transaction that specifies their Price. Portfolios are simply a complete set of shares. If the miner detects an imbalance of shares between inputs and outputs, it will attempt to resolve the imbalance by converting complete sets of shares into XFT (on the input side for SellCompleteSets and on the output side for BuyCompleteSets).

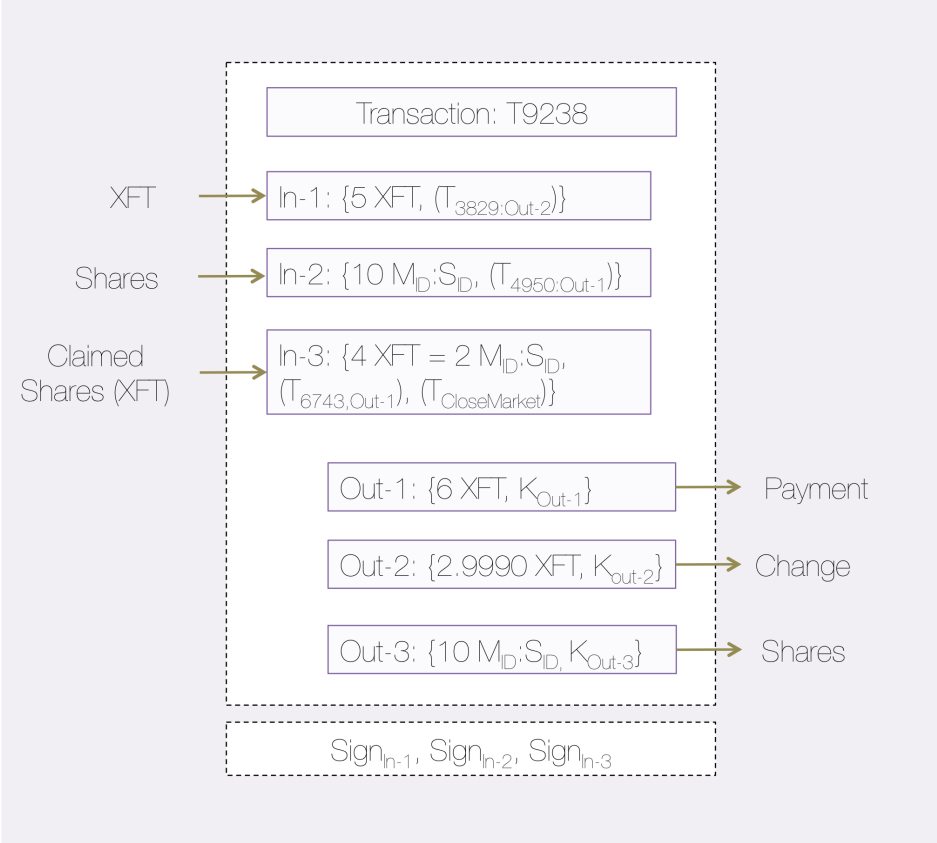


Fig. 2. A simplified example of an Exchange transaction to demonstrate how shares can be transacted in ways similar to Bitcoins. Here Alice sells 10 shares to Bob for 6 XFT. Bob obtains 9 XFT from previous transactions (including 4 XFT from previous winnings), outputs 6 to Alice, provides a 0.001 XFT tip to the miner, and retains 2.999 XFT as change.

We have not explicitly shown OpenMarket and CloseMarket, however they are relatively simple as OpenMarket is an atomic transaction and CloseMarket only refers to OpenMarket.

Scripting. Outputs are assigned to public keys (in actuality, fingerprints of public keys). Exchange must be signed with the signing keys associated with each input's output. In Bitcoin, this represents the simplest case and the nature of the majority of transaction. However, outputs can more generally be associated with a predicate using a scripting language, and only be redeemed by an input satisfying that predicate. The is no reason our design cannot adopt the same scripting capabilities as Bitcoin (or even expanded scripting capabilities seen in some altcoins), however it is not a central focus of this research.

5 Design Parameters

5.1 Arbitration.

Now we explore the design space for arbitration mechanisms. Arbitration is the process of reaching consensus on the outcome of an event. In the previous section, we posited an arbiter for each market, but this is far from the only possibility.

We emphasize that we do not deal here with events having an outcome that is genuinely unclear or not captured by the definition of the market. For example, American football fans have bet for years on which

colour Gatorade the winning team in the Super Bowl will dump on its head coach. In 2014, the Seattle Seahawks unexpectedly performed the tradition twice with two different colours, and no arbitration could fully square this reality with how the market was set up. Instead we focus on the vast majority of cases, such as the winner of the Super Bowl game itself, in which the outcome is obvious to all observers. Even in these cases, arbitration in the absence of a central trusted authority turns out to be a tricky problem.

To begin with, it is clear that for most markets arbitration requires one or more humans in the loop and cannot be automated, nor can the correctness of arbitration be defined or checked automatically. Nevertheless, the problem at least theoretically has a solution if the majority of participants are honest. This is complicated by three factors: first, participants may be pseudonymous, and anyone can create sybils. Second, some participants will have a monetary stake in subverting the vote because they hold shares in a losing outcome. Third, we would like to minimize the human effort required to reach consensus.

There are different ways to distribute *voting power* over an event's outcome to participants, keeping in mind that participants don't have identities. The first option is for each share in a market to confer one vote in the outcome of that market. This model inevitably results in a kind of cyclicality, as we discuss below. The second is to distribute voting power to miners based on proof of work, with the mechanics of voting integrated into block processing. Here distribution of voting power is the same for all markets. A third option is to construct a new type of entity for the purpose of voting, combining attributes of the first two.

Voters can be incentivized to vote *correctly* by carrots and/or sticks. The *carrot* approach involves reputation combined with payment for voting. Voters could either build up reputation through a history of accurate voting or 'import' reputation by identifying themselves as well-known real-world entities. This is what the single-adjudicator model uses. The *stick* approach involves voters putting up a bond that may be forfeited as a penalty based on the actions of other voters. The two can be combined, giving voters a financial instrument whose value is tied somehow to the correct arbitration of markets.

Based on this menu of choices, we describe three designs in addition to the simplest case of an arbiter per market. The first two are novel, whereas the third was communicated to us by Joshua Brown in a manuscript [7].

Miners as voters. In this approach, anyone broadcasting a block gets a vote in every market. Votes take the form of CloseMarket transactions, which require no special privileges. Each miner locally maintains a list of markets and their votes for some of those markets. A miner may refrain from voting on a market because the event in question has not yet happened, or because the miner simply does not have an interest in voting on that market.

Each miner is supposed to follow three rules: (1) a CloseMarket transaction is to be treated as invalid if it contradicts a vote that the miner has recorded locally; (2) miners are to ignore blocks that contain invalid CloseMarket transactions, but are allowed to build on such a block if it already has k confirmations; (3) when constructing a new block, miners are to include in it CloseMarket transactions for all open markets for which they have recorded votes locally. A market is to be considered closed if it has t consistent CloseMarket transactions on the longest valid chain, with t specified as a parameter of OpenMarket.

Given these rules, a miner who votes incorrectly on any market risks having their block ignored by other miners, forfeiting the mining reward. However, if the market is relatively obscure with most of the miners ignoring it, then an incorrect vote may pass uncorrected for the next several blocks, effectively becoming irreversible. We can conclude that the miners-as-voters approach will work better for more prominent markets where a significant fraction of miners are likely to record votes (*i.e.,* voting will conclude more quickly and will also be harder to game).

Voters as shareholders in a company. Imagine a virtual 'company' (PredCo) whose ownership is distributed among many participants. This can be implemented via operations to create and trade shares similar to the PM operations. The implementation details are secondary, however; all that we require is that share ownership be a cryptographic assertion that anyone can verify, and that payments made to the company accrue to its shareholders in direct proportion to ownership.

Now consider what happens if PredCo acts as an adjudicator of markets instead of an individual, with arbitration being the outcome of a shareholder vote. Due to trust agility, if PredCo arbitrates a market incorrectly, participants are unlikely to bid on markets arbitrated by PredCo in the future, decimating PredCo's expected future earnings. Since the value of a share in PredCo is proportional to the net present value of these earnings, those shares will lose their value.

There are two salient differences between this model and simply having a threshold of arbiters. First, trust is tied to this virtual entity instead of a set of individuals, decreasing the cognitive burden on prediction market participants and allowing voters the flexibility to enter or leave the voting business at any time simply by buying or selling shares in PredCo. Second, the effect of trustworthy or untrustworthy behavior by voters gets immediately translated into monetary terms (*i.e.,* the value of PredCo shares), disincentivizing voters from misbehaving even if they are irrational or have a heavy discounting rate for future earnings.

Voting as a Keynesian beauty contest. In this approach, each of N shares in a prediction market confers one vote for market arbitration [7]. In the common case, a supermajority of voters ($k > 2N/3$) vote the same way, this is considered a consensus and the winning shares will be paid out.

Of course, voters holding losing shares have a financial incentive to vote contrary to reality. To address this dilemma, all market participants are required to post a bond in addition to the price of the shares they purchase. Any voters who vote contrary to an outcome with reaches a $2N/3$ consensus forfeit their bond, disincentivizing voters to vote against the likely final outcome. The market might still fail to reach consensus if, for example, there are two possible outcomes and all participants holding shares in the losing outcome form a coalition and refuse to provide any votes necessary for the market to reach consensus. To disincentivize this, if the market fails to reach consensus after a certain time period then all participants forfeit their bonds.

The functioning of such a system in practice is unknown. In the case of markets with a genuinely unclear outcome, the system creates a "Keynesian beauty contest," with all participants incentivized to vote for the outcome they believe others will consider correct, rather than their own fundamental beliefs.

Ignoring such cases though and assuming a market with a clear outcome, this system produces an iterated game of chicken between coalitions of voters holding shares in each outcome. If either coalition is able to convince the other that they are absolutely going to spend their $N/2$ votes on their preferred outcome, the other side is incentivized to back down and concede to prevent losing their bond. It is only be repeated play, in which participants have a reputation to maintain that will be damaged if they vote for a patently incorrect outcome, that this game can be avoided. Thus we think this approach is less desirable as it requires tracking reputations for all participants in the market and not just a small number of adjudicators.

5.2 Bitcoin Integration Options.

Colored Coins. As an alternative to deploying a PM as an altcoin, we consider using the 'colored coins' technique[8] for transferable warranties to deploy the same design within Bitcoin. Essentially, a counterparty accepts deposits from users in exchange for transferable warranties, and promises to redeem the winning shares for the full price once the bet has ended. The advantage of this design is that it would be compatible with the existing Bitcoin system, rather than requiring a separate altcoin currency. The tradeoff is that this approach involves counterparty risk, although the function of the warranty system is meant to minimize it.

The colored coin technique makes use of the fact that Bitcoins are not truly fungible, but instead have a traceable transaction history that enables the association of arbitrary metadata (in this case, called a *color*) to any coin. To use this technique for implementing a PM, a counterparty/lender sets aside a small *carrier quantity* of Bitcoins, and announces (by publishing a signed statement) that these coins are intended to represent a specified number of shares in a specific bet. The colored coins can now be transferred, split, and merged, just like ordinary Bitcoins. Any other party can use the public transaction log to verify that a given

[8] Assia, Buterin, Hakim, and Rosenfeld, 2013. `https://docs.google.com/document/d/1AnkP_cVZTCMLIzw4DvsW6M8Q2JC0lIzrTLuoWu2z1BE`

quantity of coins bears the color. To do this, the verifier traces the transaction history of the given coins back to the original announcement.

More technically, the verifier applies a *coloring kernel* rule at each transaction.The coloring kernel must define what happens, for example, when a transaction contains both colored and uncolored coins as inputs. The coloring kernel must guarantee that a transaction cannot increase the supply of colored coins, but should be flexible enough to allow splitting and merging of values. A simple rule is to sum all of the input quantities, and apply them in sequence order to the outputs. If the sum of the inputs is not exactly utilized by the first transaction outputs in sequence, then any partial remainder is eliminated—such a transaction is considered a user error and effectively destroys a portion of the original colored coin.

A user purchases a portfolio by exchanging Bitcoins for the counterparty's colored coins. The counterparty is obligated, once the bet has been arbitrated, to exchange coins bearing the color of the winning share for the original portfolio purchase price.[9]

These colored coins can be considered transferable warranties, since if the counterparty defaults on its obligation to redeem the winning shares once the market closes, the evidence of this is publicly verifiable and the counterparty can be held accountable. For simplicity, an Arbiter may also act as the counterparty; however, this consolidates more trust in the Arbiter. In any case, regardless of whether the Arbiter and counterparty are separate entities or one in the same, this colored coins implementation does require additional reliance on third parties, compared to implementation as an altcoin. In addition, the order matching service in Section 6 is not directly supported.

Forecasting with Options. In our PM, a user can only purchase shares using money (XFT) that she already has. It is common in financial markets to use put or call options for forecasts. Options can be seen as a composition of two kinds of instruments: first, a bet, in the sense that a PM provides; and second, the option (but not obligation) for parties to exchange a fixed quantity of shares for a prespecified amount of money at a later date.

The latter can be implemented in a general way by using warranties to minimize counterparty risk. The key idea is that both parties sign a contract agreeing on a set of conditions under which one party will make a transaction (*e.g.,* to transfer XFT to the other party, or to sell off shares of a bet at the market rate). Since the condition (*i.e.,* the outcome of the bet, after the market closes) and the transaction obligations are public record, any failure to fulfill the obligation is publicly detectable.

Bitcoin Tokens. Another alternative to implementing a PM as a full-fledged altcoin, with an independent block chain and monetary base, is to issue tokens that are tied in value to Bitcoin—something akin to poker chips. The tokens are fully controlled within the PM, and can be created and destroyed with the consensus of its miners. Given that miners are still needed to perform the work of maintaining a separate block chain, and that they are not rewarded through new tokens, miners would likely only include transactions that included a fee. This is the same model Bitcoin will transition to once the complete supply of XBT has been created.[10]

The remaining question is how to distribute an initial supply of tokens fairly? Providing tokens to a central entity is antithetical to our design goal of a decentralized PM. Instead we propose that a new token is issued in the PM to the same public key as one used to demonstrably give up a Bitcoin to obtain it (thus we must use the same signature scheme as Bitcoin: i.e., ECDSA). This could be through burning it (sending it

[9] Note that that this purchase price is typically much larger than the face value of the carrier quantity, which should be as small as possible without triggering Bitcoin's DoS-prevention mechanisms. Bitcoin de-prioritizes transactions with extremely small value in order to increase the cost of DoS attacks.

[10] While new coins are minted—25 XBT every 10 minutes, on average, for the moment—the schedule of new coin minting is scheduled to gradually but predictably diminish. At some point, around the year 2024, no new coins will be minted, and the network reaches its stationary state (or deflationary, since some coins are lost due to forgotten private keys). Note that in the steady state, the 'mining' process continues, despite the fact that no new coins are created—at this point, miners will be incentivized by transaction fees alone, rather than coin minting bonuses.

to an address with no known private key)[11] or through donating it to one of a set of approved charities. The latter is effectively the same as giving the initial supply of tokens to the charities to sell, however without burdening the charities with the logistics of selling them. Once enough tokens are in circulation, they can be sold for Bitcoin through a traditional currency exchange. Tokens should never be significantly more valuable than XBT since XBT can be converted directly into tokens, with relative efficiency.

5.3 Other Design Parameters

Fee structures. The authorization given to an arbiter to determine the outcome of a market enables a malicious arbiter to profit from misbehaviour. If a portfolio is priced at 1 XFT, shares will generally trade for less than 1 XFT and approach zero when a different outcome is determined. A malicious arbiter could purchase shares in the cheapest outcome, declare it as the outcome in spite of the actual result, and profit from the difference. In this case, the arbiter would never be trusted again and would lose the ability to arbitrate future events.

In order to deter such outcomes, we allow arbiters to earn revenue through a commission fee. The hope is that the net present value of the future revenue stream will outweigh the profit earned by absconding (discounted by any loss to the arbiter's reputation outside of the PM). The fee can be applied in a few ways: (i) upon BuyCompleteSets which results in the cost of a portfolio being 1 XFT + fee, (ii) upon CloseMarket which results in shares in the outcome being awarded 1 XFT − fee, or (iii) upon Exchange which requires the sender to send a fee in XFT to the arbiter. The arbiter can propose any combination of these types of fees and the amount of each fee in a new Fee parameter added to OpenMarket. The distribution of the fees to the arbiter will be enforced by the mining community.

Arbiter fees are in addition to any fees paid to the miners in the underlying currency. Currently, miner fees in Bitcoin are non-mandatory (but may result in the miners prioritizing inclusion of the transaction) as they are offset by the miner receiving newly minted coins.

PKI Issues. The arbiter(s) are specified by public key(s) in OpenMarket. For arbiters with an identity outside of the block chain, any binding between this identity and their public key is external to the block chain, leaving a gap in authentication users must learn to fill. Arbiter key(s) are also subject to the standard threats of theft or loss, and the current mechanism provides no method for revocation or key rollover. Note that prediction markets could be held over many years, decades, or indefinitely.[12]

To address these issues, it may be desirable to hook into an existing PKI for identity management. For example, OpenMarket could specify a Distinguished Name and set of Certificate Authorities, and the CloseMarket signer would include a valid (*i.e.,* unrevoked and unexpired) X.509 certificate for their signature key, issued by one of the specified CAs. In particular, extended validation (EV) certificates enable revocation checks and are restricted to legally registered entities.

Payoff Structure. A limitation of the prediction market payoff structure is that share prices are designed to reflect the probability that an event will occur. This leads to an unnatural expression of certain outcomes. For example, on InTrade, it was not possible to directly predict the number of seats the Republicans or Democrats would win in the US House of Representatives. Instead, several markets would be launched with binary outcomes: the Democrats would win at least x_1 seats, x_2 seats, *etc.*, for increasing values of x_i. OpenMarket could allow the Arbiter to give a non-binary payout. For example, portfolios could be sold for 4.35 XFT (435 seats / 100) with shares for each major party (*e.g.,* Democrat, Republican, Independent, other) and payoffs would be the number of seats won / 100. In this case, if Republican shares traded at 2.32 XFT, the market forecast translates to a win of 232 seats. Recall that part of the power of prediction markets is usability: enabling information to be readily inferred from the share price.

[11] I Stewart. Proof of Burn. `https://en.bitcoin.it/wiki/Proof_of_burn#Coin-burning_as_a_tool_for_transition_between_cryptocurrencies`

[12] See Long Bets: `longbets.org`

6 Decentralized Order Matching

In the previous section, we outlined how the functions of a PM can be realized in a decentralized manner and how transactions can be cleared and settled. The remaining question is how buyers find sellers, and how traders agree on a price.

Our proposal is to build an integrated order matching system into the block chain, while also providing functions that enable external third-party exchanges to emerge. We cannot rely on the immediate emergence of such third party markets, because unlike arbitration, the barrier to entry for running an exchange is high. Thus, we anticipate that the integrated system will bootstrap trading, and will serve as an always available default for users who have not chosen an exchange market of their own.

The nature of an exchange is defined by its trading rules [14]. For example, in equity markets, the same stock may trade on multiple exchanges, each with its own unique rules. Since different traders have different preferences and needs, there is a demand for a variety of markets with a variety of rules.

The most common order-matching system seen in other prediction markets (such as InTrade) is a continuous two-sided auction where traders submit bid orders (the price they are willing to pay to buy a certain volume) and ask orders (the price they are willing to accept to sell a certain volume). An order will be executed if a trader is willing to match (or better) its conditions. Orders can only be executed in an sequence specified by precedence rules: *e.g.,* first sorted by best price, and then by first received. Rules also specify the price at which the order is executed, which is never any worst than the price specified in the order, but may in certain conditions be better. Finally rules govern the granularity with which the volume of an order can be partially filled.

6.1 Integrated Call Market

Recall that transactions are broadcast to a P2P network that includes the miners. The miners bundle transactions into blocks, which are integrated into the block chain at semi-regular time intervals. Now consider the design of a block chain-based decentralized continuous two-sided auction with price-time precedence, where orders are signed transactions (we detail the exact structure in Section 6.2 below). Several constraints make this market type difficult, if not impossible, to deploy with a block chain:

1. Reliably establishing time precedence is difficult in a decentralized network with varying propagation times between nodes, unsynchronized clocks, and the potential for manipulation (delays) by nodes;
2. Traders have an incentive to configure the nodes they control in the P2P network to not forward orders that higher trade precedence then their own;
3. Transactions are not finalized until added to the block chain, which does not occur continuously;
4. Miners in the P2P network are disincentivized from including orders that have higher trade precedence than their own in a block; and
5. Without reliable time precedence, miners can front run any orders that cross.

Trade Precedence Rules. By examining this set of constraints, we can see that an alternative design is necessary, and the trading rules of this alternative will be essentially prescribed by the narrow properties of a block chain structure. Constraint (1) leads us to believe time-precedence is unachievable, so we give precedence to price, and then execute orders at the same price relative to their volume. This is called 'pro rata allocation' and means if Alice and Bob respectively bid for 100 and 200 units at the same price, and this price matches to an offer of 75 units, then Alice will fill 25/100 and Bob will fill 50/200.

Constraint (2) is not possible to entirely eliminate, however it can be mitigated by broadcasting the message to as many peers as possible. To function correctly, we must assume that it is always possible to reach a well-connected peer that does not have a standing order in the market that the user is trading in (*i.e.,* the market for the particular outcome of the particular event).

Call Market. Constraint (3) forces us to consider a call market instead of a continuous market. In a call market, orders are placed over an interval of time and then executed as a batch. While less common than continuous markets, they are sometimes used on low-volume stocks, to open and close an otherwise continuous market, or to resume after a halt. In a call market, orders are sorted by precedence, and the best bids proceed to be matched with the best asks until no more orders can be executed.

To illustrate the difference, consider the following example. Assume the following four orders arrive, each with the same volume: \$100 bid, \$100 ask, \$99 bid, and \$99 ask. In a continuous price-time market, the first two orders would execute and then the final two would execute. In a call based market, the best bid of \$100 would execute against the best ask of \$99. A 'price discovery' rule dictates the price. For now, assume its half the spread: \$99.5. The remaining \$99 bid and \$100 ask would not execute. Neither system is better than the other, they merely have different properties. The continuous market allowed more trades to execute ('quantity') and faster ('immediacy'), while the call market allowed the traders to achieve an improved price ('trader surplus').

Front-Running & Manipulation. To illustrated constraint (4) and a variant of (5), assume a miner places a \$99.5 ask in the example above. By excluding the \$99 ask from her block, she can be assured that her order will be matched to the \$100 bid for a surplus of \$0.5. This manipulation gains her two things: (i) the price improvement and (ii) the ability to prioritize her trades over others. Mitigating such manipulation is difficult, however, perhaps surprisingly, we can build a case for embracing it.

Miners typically earn fees. If we allow the miner to keep any surplus from matched orders, we can fulfill this function without requiring additional fees, plus we remove their ability to gain from (i), the price improvements accrued through dropping orders. In other financial markets, brokers are allowed (ii), the ability to fill orders themselves, but must do it at the best market price. Thus we modify the trade precedence rule to be price then miner[13] then pro rata by volume.

With the miner keeping the surplus, they no longer are incentivized to do anything but fulfill at the best price. In the example above, if the miner lets the \$99 bid stand, she will make \$1 on the surplus. If she offers to fill the order herself at the best price, \$99, she will also make \$1 on the surplus. If she submits a worst bid and drops the best \$99 bid, she will not gain anything. For example, at \$99.5 she improves her price by \$0.5 and but decreases her gains from the surplus by an equivalent amount, netting only \$0.5 for a total of \$1.

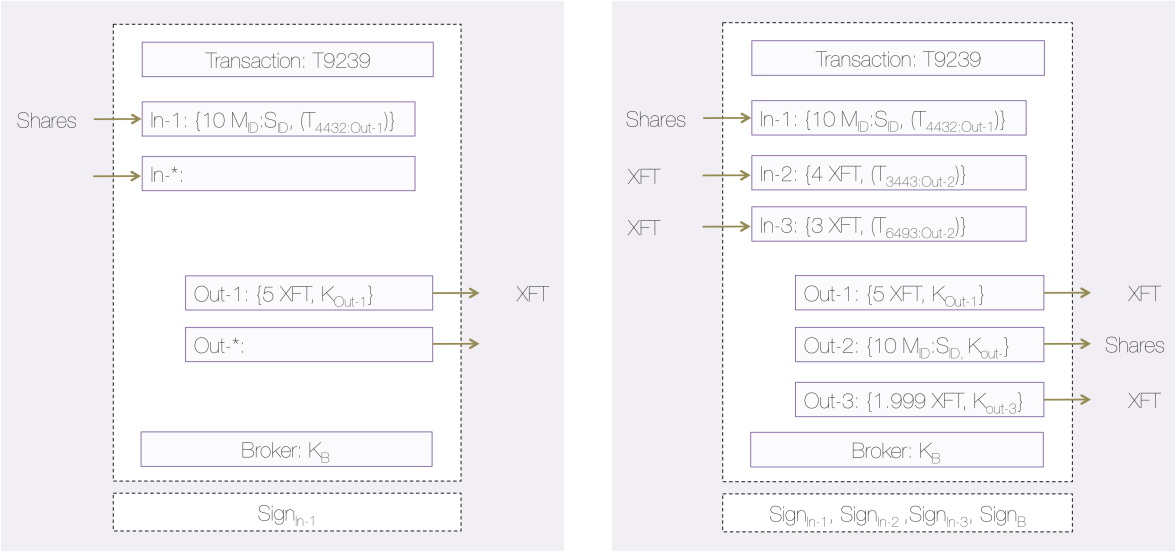
Price Discovery. The miner must execute the batch of orders at a price that maximizes the number of orders executed, according to the order precedence rules, however there may be an interval of prices that enable market clearing. Given the trading surplus is retained by the miner, it is no longer relevant to the traders what price the trade actually executes at. From their perspective, it will always be the limit price in their order.[14] Thus the price is only useful for reporting purposes. We suggest that the midpoint of the interval is reported, but note that since the executed and standing orders are completely transparent, market observers are free to interpret the price using any method of their choosing.

Call Time Uncertainty. A natural question is why we expect there to ever be a surplus for the miner to claim. Since orders are broadcast to the P2P network, it is a lit market where every trader knows the status of the order book and can compute the going market clearing price. By waiting until immediately before the call, traders can decide to execute at that price or hold off.

A similar issue arises in crossing networks, which are call markets that are typically dark (*i.e.,* a private order book) but determine the market clearing price from prices on an external lit exchange. Crossing networks have additional concerns, such as manipulation in the external network, that do not apply here. The primary mitigation used by one such network, POSIT, is to execute the trades a random time during a short interval (7 minutes) after the call [14]. Surprisingly, the block chain incidentally gives us the equivalent

[13] As identified by public key in the coin base transaction.

[14] For this reason, we do not support market orders.



(a) Order (b) Executed Order

Fig. 3. Modification to Exchange transaction (see Figure 2 above) to facilitate orders. Alice submits (a) a signed ask of 5 XFT for 10 shares. Bob accepts the order and (b) augments it with his transaction details, signs it, and gives it to his external exchange, broker, who finalizes it by signing.

functionality. Block creation is a random event paramaterized to exhibit a certain frequency on average (*e.g.*, every 10 minutes in Bitcoin) but the timing for a particular block cannot be predicted.

For this reason, traders who prioritize immediacy over price will make orders early in the time leading up to a block creation to ensure the order is considered during the current call, while miners who prioritize price over immediacy may gamble by waiting for further price clarity before submitting their orders (with a chance of missing the current call). Note that due to the nature of the proof of work executed by the miner, orders cannot be added to a solved block without ruining the solution. Thus miners cannot stuff late orders (their own or others) into a solved block.

Order Presentation. User clients will listen to orders as they are broadcast on the P2P network. The user client can then display the order book to the user and offer to execute orders. As blocks are solved, the user client will update a history of the market clearing prices.

6.2 Order Microstructure

The design of Exchange could be modified to allow one party to sign a transaction with all the details specified except the identity of the corresponding party inputting the specified Volume of shares or the Payment of XFT, and the output to receive the complimentary input. Such 'half' transfers would enable orders: an ask would not specify a Receiver, and a bid would not specific a Sender. Figure 3 shows this (the way Broker is used in the figure will be specified below).

To handle a number of different trading scenarios, including orders to be executed in the integrated call market, we introduce a new optional field in Exchange called Broker. A special flag would be inserted into this field to indicate that the order may be arranged by whichever miner happens to solve the next block. We also add a time to live (TTL) field to specify the block number at which the order will expire. This is essential as there is no easy mechanism to allow orders to be canceled (cancelling can be done indirectly, by transferring the funds/shares out of the account specified in the transaction, which requires at least one

block creation for insertion). It would be up to user clients to choose a default setting for the TTL, however it would be natural to set orders to expire immediately if not included in next block (*i.e.,* as IOC orders) by default, while allowing longer lived orders for more advanced use cases.

Privately Arranged Trades. The openness of Exchange enables any two parties to arrange a private ('over the counter') exchange without using the internal call market. It also enables a traditional type of external exchange service, where parties sign their shares and XFT over to the exchange and the exchange signs on behalf of the parties.

Enabling External Exchanges. Orders could also be left open for external trading, where any interested party could co-sign the order and broadcast it. This would allow an external exchange to only provide order matching, without requiring it to hold user shares and XFT. The difficulty with this approach is that many people may co-sign the same order, including the miner that solves the block that ultimately includes it, who will have possession of the order last. Even if the order is kept private until co-signed, anyone can strip the second signature off the transfer to create an uncosigned order. For this reason, we can use the Broker field to specify the public key of an additional party that must sign the completed transfer in order for it to be valid.

In this scenario, Alice could create a bid for a share, provide it to the matching service specified in Broker and go offline. Later if Bob is willing to accept the conditions of the bid, Bob can sign the share and return it to the matching service and go offline. The matching service signs the completed transaction, ensuring no one except Bob can accept Alice's bid, and submits it to the P2P network for inclusion in the block chain. The matching service could earn a commission by providing this service.

A matching service has a few advantages over a traditional exchange. Since Alice and Bob never give up signing authority over their shares and XFT, they do not have to trust the service with their holdings (only that it executes a fair matching process). The service is relieved of the security and liability considerations of holding others' shares and XFT. By being external, the service can implement any market structure it wants (continuous or call markets, any trading or order precedence rules, *etc.*).

The only limitation to a matching service is that clearing and settlement requires block chain integration, and the timing of integration depends on the average block creation time. With a 6 block confirmation delay at 10 minutes a block (as per Bitcoin), we are left with a reasonable period of one hour. By comparison, equities often take one or three days ('T+1' or 'T+3') to clear and settle. However an external exchange that holds the shares and XFT for its traders can clear and settle immediately, at least in terms of the traders' accounts with the exchange (while actually withdrawing the shares or XFT from the exchange is subject to the same block chain delay). Nearly every altcoin with significant use reduces the block confirmation time, and we could also use a shorter confirmation (for example, 2 minutes).

6.3 Market Transparency

In terms of transparency, the Broker field enables market observers to track all orders, standing and executed, in the internal call market. It also shows all the trades executed by each matching service, as identified by the key of the matching service (which may be kept pseudonymous). Trades arranged off-exchange are also evident in the block chain. The only trades that are not visible are trades made within an external exchange that holds its traders' shares and XFT. These exchanges may chose to publish their own order book and executed trades, or may choose to operate a dark pool. Both offer certain advantages to certain types of traders. Our design is general enough to support either.

7 Discussion

7.1 Prediction Market Issues

In 'bets off' scenarios, an open market cannot be closed due to unforeseen circumstances (*e.g.,* a cancelled sports match). Ideally, the market would be opened with shares in a catch-all 'other' outcome, however this is not typically done.

A complete portfolio can always be redeemed for currency, and so in the worst-case, there is always a market for shares that can be used to complete a portfolio. However the price of the shares at this point will be invariant to the value that the shares held before the cancellation of the market. Non-binary outcomes, discussed in Section 5.3, would allow an arbiter to issue a pay-off proportional to share prices immediately preceding the cancellation, assuming the cancellation happens at a discrete point in time. In either event, users should only invest in shares with a very clear contract that considers the possibility of cancellations and specify an acceptable policy.

Prediction markets must also clearly specify the conditions under which an outcome is considered finalized. For example, the winner of the 2012 Iowa Republican caucus (an electoral event in the US) was initially declared and reported in newspapers to be Mitt Romney and InTrade settled the market based on the fact this candidate was declared the winner. However the official count released two weeks later revealed Rick Santorum to be the winner. Once again, clear criteria for setting markets is integral to the contract.

Finally, prediction markets (and most forms of betting) suffer from the threat of the arbiter or market maker stealing the money. Our design is actually an improvement as arbiters do not hold the money in the market as a deposit, and any clearly malicious decision can be overridden by the community as discussed.

7.2 Legal and Regulatory Issues

Our decentralized PM is strictly a design. We are not currently pursuing any level of actual deployment. If it were to be launched, it is not clear that it would be legal in all jurisdictions to participate, whether as a user, arbiter, or miner. For example, in the US, the CFTC filed a civil complaint against InTrade for off-exchange options trading, and InTrade closed to US users as a result. In certain ways, we anticipate that the regulation issues parallel those of Bitcoin. Regulatory issues of Bitcoin, prediction markets, and Bitcoin-based prediction markets are explored by Brito *et al.* [6].

7.3 Ethical Issues

In centralized PMs, the decision to allow any market is discretionary. In decentralized systems, there is little to no control over what types of markets will be opened. Markets on assassinations or terrorist attacks can be considered unethical, and were cited as reasons for cancelling at least one prediction market, developed in the US by DARPA for predicting foreign political developments [25]. In our design, for particularly abhorrent markets, a consensus formed by a majority of miners to not include transactions opening, trading, or closing a given market will effectively stifle it. However markets with merely questionable ethics will likely proceed, and it will be left to individual users to decide to participate or not.

7.4 Limits of Extension to Derivatives

Since a prediction market allows forecasting about anything, including the future price of financial instruments, it can function as a crude derivatives market. The payoff structure of a prediction market share is, however, different than that of an option. Further, prediction market payoffs are highly unusual in the special case of forecasts about the future exchange rate of the currency underlying the market. Consider in a Bitcoin-denominated market, the value of a share in the forecast that the exchange rate of XBT will plummet past a certain strike price by a certain date. If the exchange rate approaches the strike price with time to spare, the probability of the forecast being true increases, making the nominal value of the share increase. However since the value of the share is denominated in a currency that is decreasing in value, the

value of the share may not actually increase in real terms. This limits the PM's ability to serve as a hedge against holding XFT, and in the integration options where XFT is fixed to XBT or implemented as colored coins, it also cannot serve as a hedge to holding XBT.

8 Concluding Remarks

The success of Bitcoin, in the face of many past attempts at e-cash, is intriguing. Among its novel contributions, the block chain stands out as a useful component for forming a consensus within a decentralized network about efficiently decidable events. Repurposing this consensus mechanism for new uses, both related and non-related to finance, is a promising research direction. In our present work, we show it can be repurposed to deploy a predication market—a useful tool for forecasting future events. However at a higher level, our work demonstrates how the block chain can be used as an independent module for providing new functionalities—a template that can be emulated for the decentralization of many different services.

9 Acknowledgments

We thank Ron Bernstein (InTrade), Gordon Mohr, the participants of the CITP Bitcoin and Cryptocurrency Research Conference at Princeton, and the anonymous WEIS reviewers for useful feedback that improved the paper. Joshua Kroll was supported by the National Science Foundation Graduate Research Fellowship Program under grant number DGE-1148900.

References

1. E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun. Evaluating User Privacy in Bitcoin. In *Financial Cryptography*, 2013.
2. K. J. Arrow, R. Forsythe, M. Gorham, R. Hahn, R. Hanson, J. O. Ledyard, S. Levmore, R. Litan, P. Milgrom, F. D. Nelson, G. R. Neumann, M. Ottaviani, T. C. Schelling, R. J. Shiller, V. L. Smith, E. Snowberg, C. R. Sunstein, P. C. Tetlock, P. E. Tetlock, H. R. Varian, J. Wolfers, and E. Zitzewitz. The promise of prediction markets. *Science*, 320(5878), 2008.
3. T. Aura, P. Nikander, and J. Leiwo. DoS-resistant authentication with client puzzles. In *Security Protocols*, 2000.
4. A. Back. Hashcash: a denial of service counter-measure, 2002.
5. J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *Financial Cryptography*, 2014.
6. J. Brito, H. B. Shadab, and A. Castillo. Bitcoin financial regulation: Securities, derivatives, prediction markets, & gambling. Discussion Draft (Cited with Permission), 2014.
7. J. Brown. Betting Markets with Decentralized Resolution and Persistent Reputation using Electronic Cash. private communication, Dec. 2013.
8. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, 1982.
9. Y. Chen and D. M. Pennock. Designing markets for prediction. *AI Magazine*, 2010.
10. J. Clark and A. Essex. Commitcoin: Carbon dating commitments with bitcoin. In *Financial Cryptography*, 2012.
11. G. Di Crescenzo. Privacy for the stock market. In *Financial Cryptography*, 2001.
12. E. Gabber, M. Jakobsson, Y. Matias, and A. Mayer. Curbing junk e-mail via secure classification. In *Financial Cryptography*, 1998.
13. R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1), 2003.
14. L. Harris. *Trading and exchanges: market microstructure for practitioners*. Oxford, 2003.
15. J. A. Kroll, I. C. Davey, and E. W. Felten. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. In *WEIS*, June 2013.
16. S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: characterizing payments among men with no names. In *IMC*, 2013.
17. I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In *IEEE Symposium on Security and Privacy*, 2013.
18. S. Nakamoto. Bitcoin: A peer-to-peer electionic cash system. Unpublished, 2008.

19. F. Reid and M. Harrigan. An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*, 2013.
20. D. Ron and A. Shamir. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography*, 2013.
21. C. Thorpe and D. C. Parkes. Cryptographic securities exchanges. In *Financial Cryptography*, 2007.
22. C. Thorpe and S. R. Willis. Cryptographic rule-based trading. In *Financial Cryptography*, 2012.
23. J. Wolfers and E. Zitzewitz. Prediction markets. *Journal of Economic Perspective*, 2004.
24. J. Wolfers and E. Zitzewitz. Interpreting prediction market prices as probabilities. Technical Report 12200, NBER Working Paper, 2006.
25. P. F. Yeh. Using prediction markets to enhance us intelligence capabilities: A "standard & poors 500 index" for intelligence. *Studies in Intelligence*, 50(4), 2006.
26. W. Yuen, P. Syverson, Z. Liu, and C. Thorpe. Intention-disguised algorithmic trading. In *Financial Cryptography*, 2010.