Truthcoin

Peer-to-Peer Oracle System and Prediction Marketplace

Paul Sztorc[1]
truthcoin@gmail.com
https://github.com/psztorc/Truthcoin
1M5tVTtynuqiS7Goq8hbh5UBcxLaa5XQb8
Version 1.4 – 4/29/2015

Abstract. Bitcoin can support financial derivatives and smart contracts, but the main benefits are lost if a trusted third party is required to inform these contracts. Instead, I propose a proof-of-work sidechain which collects information on the creation and state of Prediction Markets (PMs). An "oracle corporation" model attempts to guarantee that a group of self-selected, anonymous, greedy users will always resolve contract-outcomes honestly. Users [1] bear all of the economic costs and benefits of any PMs they create (ensuring efficiency), [2] can create PMs on any subject, and [3] can trade anonymously in any PM. All PMs enjoy low fees, permanent market liquidity, automatic token-issuance, and fair, high-speed trading through a LMSR market maker. Scalability and customizability are achieved via 'branching' (controlled-fork of the VoteCoin set).

PLEASE send Typos/Confusions to truthcoin@gmail.com or pull request into the following link:
https://github.com/psztorc/Truthcoin/tree/master/docs#addendum--errata
(And check the link for a preview of later version)

[1] Dedicated to Robin Hanson, for taking the high road.

Article I. Overview

(a) General Strategy

(i) Central Facts

1) Blockchains allow for the programmable, censorship-resistant exchange of value-tokens.

2) While most marketplaces require physical infrastructure to facilitate the trade of physical goods[2], a Prediction Marketplace requires only the trade of digitizable information, and can therefore exist entirely in a software environment.

3) As time progresses, questions which were previously mystifying become easier and easier to answer. For example, the question "Will Candidate X win the November 2016 US presidential election?" may be impossible to answer in 2015, easier to answer in late October 2016, and very easy to answer in December 2015.

(ii) Assumptions

1) Truthcoin inherits all of the assumptions of Bitcoin. For example: No malicious entity (an individual or perfectly-coordinated group) controls a large percentage of hashing power.

2) Users are greedy (prefer having more money to having less money), and lazy (prefer putting in low effort to high effort).

3) Miners might care (ie, "care with some difficult-to-measure but nonzero probability") enough about the transaction fees derived from the Truthcoin sidechain to (as a very rare last-resort) act purposefully on behalf of the coin. This may be, for example, to cast a single "Veto" for Ballots which contain large sets of mis-resolved Decisions.

(iii) Scope

1) Initially, this project aims to address public topics which are *very* easy to verify: no more than two minutes of web searching to resolve each "Decision" ("Decisions" partition the Prediction Markets (PMs)). Over time, Truthcoin is designed to be able to address topics of ever-increasing obscurity (see Article III "Branching").

(b) Brief Overview of Components

(i) The Two Coin-Types

1) The Truthcoin blockchain is a Satoshian proof-of-work blockchain[3] with different block-creation/validation rules. Most distinctively, Truthcoin uses two types of coin: "CashCoins" (CSH) and "VoteCoins" (VTC).

[2] https://www.lme.com/trading/warehousing-and-brands/warehousing/approved-warehouses/
[3] https://bitcoin.org/bitcoin.pdf

a) CashCoins are most relevant to the user: They are redeemable 1:1 for Bitcoin[4], and represent value. CSH-owners can make use of the protocol's features (create PMs, and buy/sell/transfer PM-shares), without owning/worrying-about VoteCoins at all.

b) VoteCoins are a new type of coin unique to Truthcoin, and represent[5] equity in the "oracle corporation". The introduction of this second coin-type achieves many things. First, it provides a sharp definition of the fuzzy concept of "reputation". Second, it makes "reputation" tradable, such that users can buy and sell this resource in precisely the same way that they might purchase shares of a real world corporation. This fixed amount of "tradable reputation" has many benefits, the chief of which are [1] Sybil-attack immunity, and [2] the alignment of ownership and control (economic value added/destroyed translates directly to an economic reward/penalty). Third, it gives the system a way to penalize agents (by withdrawing their VTC) for laziness. Fourth, it eliminates the temporal dimension from all incentive calculations (payments can be compared with each other, regardless of *when* they take place); if not eliminated, this dimension would have presented catastrophic risks (namely, the "exit scam"[6]).

i) Truthcoin can host many "oracle corporations", which are called "Branches", and each Branch has its own set of VoteCoins. A higher percentage of VoteCoins owned implies a greater degree of voting influence within the Branch, and a larger share in the Branch's revenues.

ii) VoteCoins do not interfere with the digital-scarcity of Bitcoin/CashCoins. As a store of value, VoteCoins are inferior to CashCoins, because VTC-owners are obligated to "vote" on a certain number of Outcomes, making a VoteCoin address more "employee ID" and less "checking account number".

iii) VoteCoins are used to perform labor in the "oracle corporation", in a weighted-voting system.

a. VoteCoins allow ("require") Owners to vote on the Yes/No/Scalar/Unknown status of 'Decisions' (questions whose answers are eventually "measureable at low cost").

b. VoteCoins allow one to proportionally collect (in CashCoin) [1] Listing Fees, and [2] half of the marketplace Trading Fees.

iv) Ownership of VoteCoins may change, solely based on voting activity.

a. VoteCoins are lost if Owners refuse to vote, or if Owners cast votes differing from the multivariate plurality (usually, the "majority").

b. VoteCoins are gained if Owners vote on neglected Decisions (those with few votes), or if Owners vote with the multivariate plurality on disputed Decisions (those where the Outcome was not unanimous).

[4] http://www.blockstream.com/sidechains.pdf
[5] The analogy is imperfect. The chief difference is that, in this "corporation", owners are employees and vice-versa.
[6] http://motherboard.vice.com/read/darknet-slang-watch-exit-scam

(ii) Automated Market-Maker

1) Literally, a protocol for updating market prices based on trading activity,[7] which aims to replace the complicated technical matching algorithms (and supporting infrastructure) of modern financial markets with a single sequence of atomic state-updates.

2) Figuratively, a "trader" who:

   a) Utilizes LMSR technology[8][9] to take the other side of any and all PM-trades, ensuring market liquidity (such that markets have a tradable market price at all times [even when volume and open interest fall to zero]). Low liquidity has been a problem on many PM-implementations, cryptocurrency or otherwise, and may have prevented the formation of crucial network-effects.

   b) Has blockchain-properties (constant mining, P2P network) which allow for an always-on, high-speed, censorship-resistant trading environment.

   c) Understands the creation (pre-trading, pre-event), maturation (post-event) and closure (post-event, post-trading) of Markets.

   d) Collects, stores, and pays out funds, without human-error or mismanagement.

(iii) (Claims about the) Incentive Mechanism

1) Authors

   a) Any user can create a prediction market ("Author a Market") about anything.

   b) Authors only have an incentive to write Decisions whose outcome (they believe) will be, by a certain date, confidently known to Voters.

   c) Authors only have an incentive to create Markets if they anticipate sufficiently-high trading volume (i.e. the contentious issues which would most-benefit from a prediction market).

   d) In all Markets where liquidity would be valued by Traders, Authors have an incentive to endow new Markets with an optimal amount of initial liquidity.

   e) Authors completely avoid the (prohibitive) cost of convincing Traders of their trustworthiness.

2) VoteCoin Owners ("Voters")

   a) Voters have an incentive to maximize the long-run trading volume of future PMs on their Branch, which encourages them to establish and maintain a reputable network.

[7] https://github.com/psztorc/Truthcoin/raw/master/docs/LogMSR_Demo.xlsx
[8] Original Publication: http://mason.gmu.edu/~rhanson/mktscore.pdf
[9] Clarifying Excel Spreadsheet: http://www.truthcoin.info/papers/LogMSR_Demo.xlsx

b) Voters have an incentive to participate in the resolution of all Decisions.

c) Voters have an incentive to vote "the way they believe other Voters will vote", which itself is contrived to be "an accurate description of reality" (see 'Voting Strategy').

3) Traders

a) Any CashCoin user can trade on any PM without directly interfacing with VoteCoins at all. VoteCoins are the "employee layer", not the "customer layer".

b) Traders have an incentive to set the market price to "their personal expectation of the probability of the event taking place", revealing that information to the public. For Scaled Decisions (on financial asset prices, for example), this trading activity produces an accurate and robust price feed.

c) Traders enjoy an absence of counterparty risk (but instead must endure the technical risk inherent to new Blockchain technology).

4) Bitcoin Miners

a) Miners always have an incentive to mine blocks, as the marginal cost for doing so is zero (merged mining allows reuse of Bitcoin hashes). Were Bitcoin to disappear, the marginal cost/benefit of Truthcoin-mining would equal that of Bitcoin-mining (and mining would therefore continue).

b) Miners have an incentive to include every trade and transaction into a block, as this maximizes not only transaction fees, but also dividend revenue and therefore market capitalization of the coins. Miners cannot even read Markets or Votes until they have already been included in blocks, making this process censorship-resistant.

(c) Extensible (Scalable, Customizable) Design

1) Accompanying software[10] is open source.

2) Truthcoin allows for the creation of controlled forks of the VoteCoin set ('Branches') enabling growth of the scope and quantity of Markets, specialized judging, choice of different fee and timing parameters, etc.

[10] https://github.com/psztorc/Truthcoin

Article II. How it Works

(a) Truthcoin Blockchain and Coin Types

1) The Truthcoin blockchain is a Bitcoin-inspired proof-of-work blockchain which aims to impose different block validation rules on the original Satoshian cryptosystem. While the Bitcoin blockchain is designed only to hold information about the ownership and transfer of a single coin-type, the Truthcoin blockchain is designed to contain information about the transfer of three coin-types (CashCoins, VoteCoins, and Shares[11]), as well as the existence and state of Prediction Markets.

2) The Truthcoin blockchain contains a new type of coin called a "VoteCoin":

	CashCoins ("Bitcoin") **User-Layer**	**VoteCoins ("Reputation")** **Employee-Layer**
1	Coin values are analogous to a saved quantity of gold.	Coin values are analogous to reputation, influence, or shares of a corporation.
2	Without user input, account balances do not change.	Accounts may either gain or lose unspent coins (based on voting activity). With no user input, coin balances would decrease.
3	Private keys sign messages that [1] transfer value, [2] create prediction markets, and [3] trade in those markets.	Private keys only sign Votes (which influence the Outcomes of Decisions) or messages which transfer VoteCoins.
4	To mimic the experience of gold and provide an objective initial distribution of coins, new coins are periodically introduced in each block by miners, asymptotically approaching 21 million total coins[12].	To mimic the experience of reputations (peer-relative, Sybil-immunity) and fulfill the requirements of voting, the total quantity of coins exists immediately, and this fixed quantity is constantly redistributed based on voting behavior.
5	Expectation of huge number of addresses, one per value-transaction.	Expectation of a maximum of 10,000 addresses, all of which will vote, but few of which will transact.

[11] That which *is* "worth $1 if Candidate X is elected", (the Arrow–Debreu securities themselves).
[12] I refer, of course, to the origin of these sidechained CashCoins: the Bitcoin Blockchain.

(b) (Decentralized, Incentive-Compatible) Calculation of Decision Outcomes

(i) Terminology

1) **CashCoins** – A cryptocurrency which is functionally equivalent to Bitcoin, yet with the ability to interface with Truthcoin prediction markets.

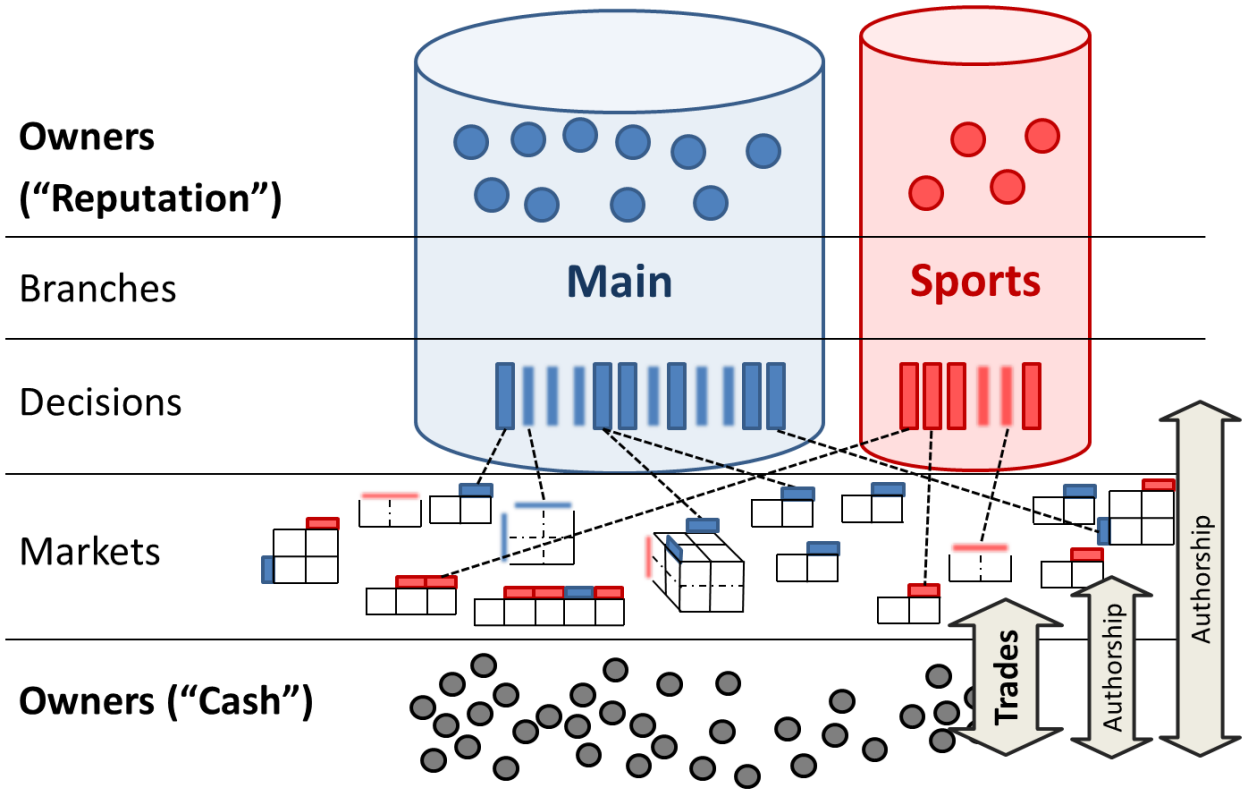


Figure 1. Graphical representation of Truthcoin's structure. Notice the two types of coin (circles), the VoteCoins representing reputation (top, colored) and the CashCoins representing money (bottom, grey). Decisions can either be Binary (bordered) or Scaled (blurred). When used in Markets, Scaled Decisions span an entire dimension, whereas Binaries only partition-from-null.

2) **Decisions** – Questions that must be resolved by Voters. These partition the State-space of a prediction market, and are defined by items such as 'event text', 'event date', 'tags', 'author', etc. (see Appendix IX).

a) Truthcoin supports two 'types' of Decision:

i) Binary (Boolean) Decisions: $x \in \{\{0,1\}, .5, NA\}$

a. Example 1: "Will Hillary Clinton be elected US President in 2016?"

b. Example 2: "Will the NYSE:DJIA closing price ever rise above 20,000 USD/Share in 2017?"

ii) Scaled (Scalar) Decisions: $x \in \{ [x_{min},\ x_{max}],\ .5,\ NA\}$

a. Example 1: "How many Electoral College votes will Hillary Clinton receive in the 2016 US Presidential election (if Hillary does not run, select 'zero')?" [x_{min} = 0, x_{max} = 538]

b. Example 2: "What will the NYSE:DJIA closing price be on January 1st, 2018 (USD/Share)?" [x_{min} = 8000, x_{max} = 24000]

c. x_{min} and x_{max} must be set in advance. Having a Decision expire at or near a bound has slightly adverse economic consequences for the Author of any Market using this Decision.

b) State ".5" denotes that a Decision is excessively confusing/irresolvable/unobservable (its veracity cannot easily be measured). This has adverse economic consequences for the Decision's Author.

c) At any given time, each Decision will have a 'status' of one of the following:

i) Active: The Decision has just been created. Decisions of the 'active' status would be likely to be used to create Markets, and those Markets would be actively traded.

ii) Matured: Decisions contain a 'date by which the information will become available'. After this date has passed, the Decision has a status of 'matured', and will enter the next Vote Matrix and be Voted on for resolution.

iii) Disputed: If Voters cannot sufficiently agree on the Decision's outcome, the Decision remains un-resolved and gains this status. From here it will be Audited.

iv) Vetoed: If Miners veto the Decision's Ballot, it gains this status (regardless of Voter-behavior).

v) Resolved: If Voters do sufficiently agree on the Decision's outcome, and the Ballot is not Vetoed, their agreed-upon value becomes the final value of the Decision, and the Decision's life-cycle is now over.

3) **Markets** – The lifeblood of the Truthcoin project, Prediction Markets allow anyone with CashCoin to buy and sell shares representing states of the world, and thereby speculate on and profit from selected events. This voluntary "win-win" speculation aggregates and summarizes information for use by the public.

a) States: Markets partition the world into 'states' or "mutually-exclusive possible descriptions of reality". When traders buy and sell shares, these shares are of a single Market State.

b) Status: Markets exist in one of two statuses:

i) Trading: In this status, a Market allows traders to buy and sell shares through an automated market-maker. A Market would be in this status from the moment it is created until all of its Decisions are voted on.

ii) Closed: When all of the Market's Decisions are successfully resolved, the Market can be "closed" with a special message, which disables buying and replaces selling with redeeming.

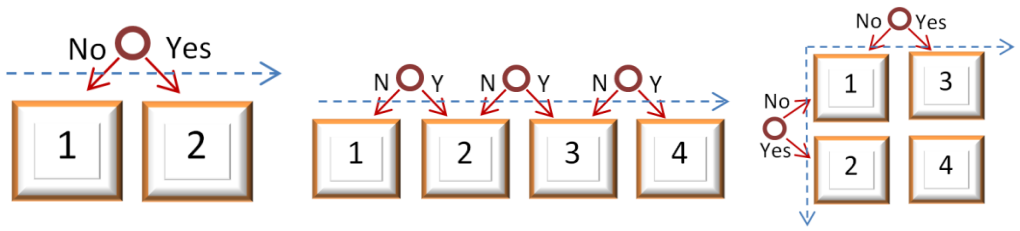


Figure 2. Graphical representation of three Prediction Markets, each with Binary Decisions. Left, the simplest form popularized by InTrade, with one dimension (blue dashed arrow), one Decision (red circle), and two states (yellow squares). Center, a Market with not two but four mutually exclusive states (for example, the winner of a 4-team tournament) and three Decisions. Right, a prediction market with two dimensions. Multidimensional prediction markets[13] allow users to trade not only on the probability of each state, but also the relationship between dimensions[14], such as the relationship between an election result and the achievement of an economic goal a year later[15].

4) **Branches** – Although all Markets are globally available to all users, Decisions are partitioned into clusters called 'Branches' based primarily on topic. Each Branch has its own set of VoteCoins (and therefore Voters), its own Decisions, and its own parameters (see Appendix IX).

5) **VoteCoins** – The second cryptocurrency type in Truthcoin. Unlike CashCoins, VoteCoins are a liability as well as an asset. Owners are expected to use their coins to vote honestly on the Outcome of each Decision (or lose them).

6) **Intervote Period ("Tau")** – The length of time between two consecutive votes on the same Branch.

7) **Vote** – The value which a Voter believes would match a given Decision to its real-world Outcome. The default value is "Missing", which indicates "No response from the Voter". A value of "1" would indicate "TRUE", "0" would indicate "FALSE", and ".5" would indicate "I can't easily tell" or "breaks the Branch rules").

8) **Ballot** – The set of all matured Decisions on a Branch. For each Decision in a Ballot, every Voter must cast a Vote with his report/opinion on the resolved value. Notice that Ballots are defined by the maturation time of their Decisions, not by their organization or use within Markets (and, crucial to the core design, Ballots contain the Decisions of many different Markets).

9) **Vote Matrix** – The matrix created by stacking the Ballots (of a particular voting cycle) by row. The columns of the matrix correspond to Decisions.

[13] http://www.truthcoin.info/papers/2_PM_Types.pdf
[14] http://www.overcomingbias.com/2008/07/intrades-condit.html
[15] http://www.overcomingbias.com/2008/01/presidential-de.html

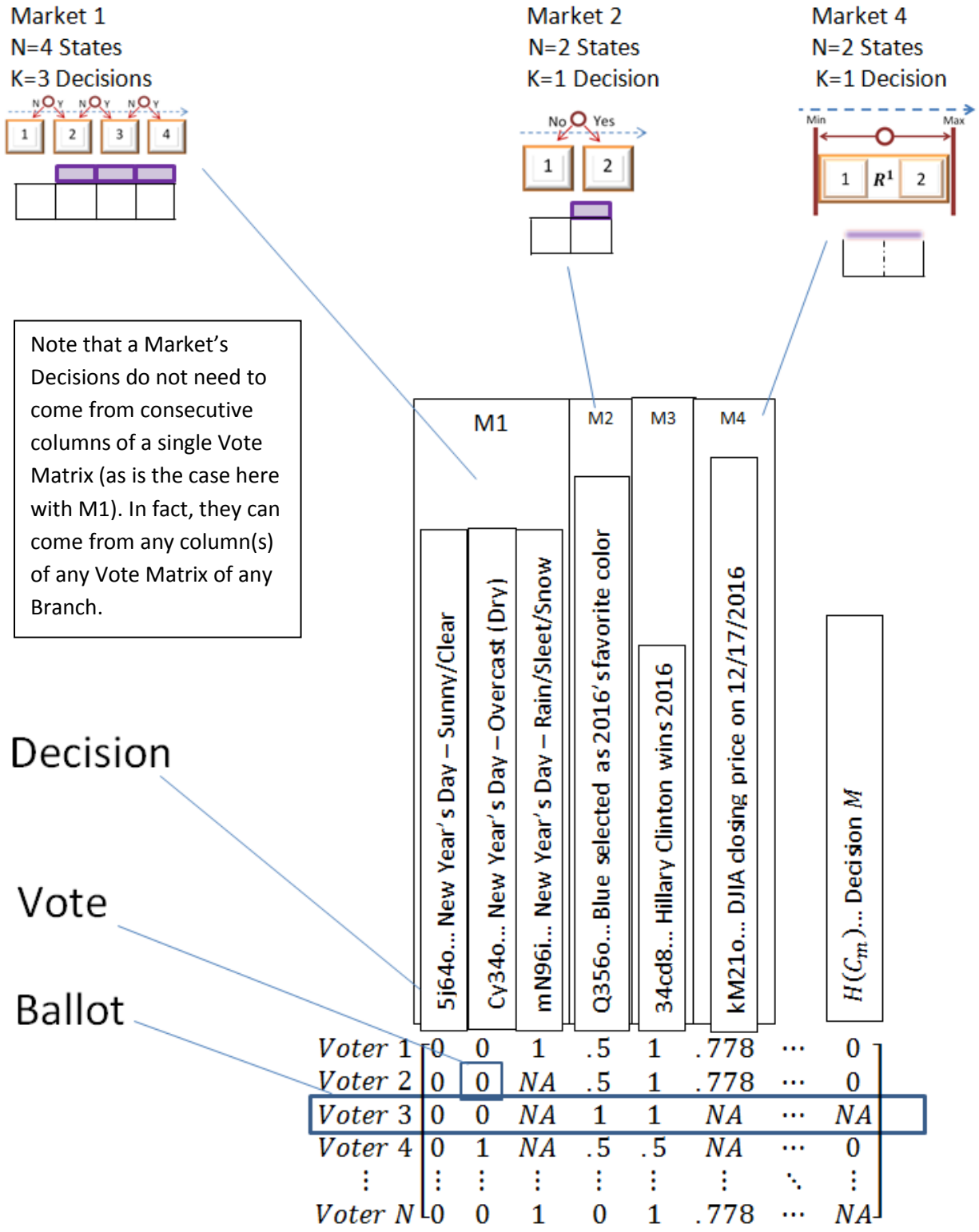

Market 1
N=4 States
K=3 Decisions
Market 2
N=2 States
K=1 Decision
Market 4
N=2 States
K=1 Decision
No
Yes
Min
Max
Note that a Market's Decisions do not need to come from consecutive columns of a single Vote Matrix (as is the case here with M1). In fact, they can come from any column(s) of any Vote Matrix of any Branch.
M1
M2
M3
M4
5j64o... New Year's Day – Sunny/Clear
Cy34o... New Year's Day – Overcast (Dry)
mN96i... New Year's Day – Rain/Sleet/Snow
Q356o... Blue selected as 2016's favorite color
34cd8... Hillary Clinton wins 2016
kM21o... DJIA closing price on 12/17/2016
$H(C_m)$... Decision M
Decision
Vote
Ballot
Voter 1 0 0 1 .5 1 .778 ⋯ 0
Voter 2 0 0 NA .5 1 .778 ⋯ 0
Voter 3 0 0 NA 1 1 NA ⋯ NA
Voter 4 0 1 NA .5 .5 NA ⋯ 0
⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋱ ⋮
Voter N 0 0 1 0 1 .778 ⋯ NA

Figure 3. A hypothetical January 2017 Vote Matrix, with annotations. This Vote Matrix would be for a Branch (at least) general enough to contain Decisions on US weather, politics, and financial indices.

10) **Outcome** – The final, calculated result for each Decision, as determined by the SVD-resolution algorithm.

(ii) Timeline

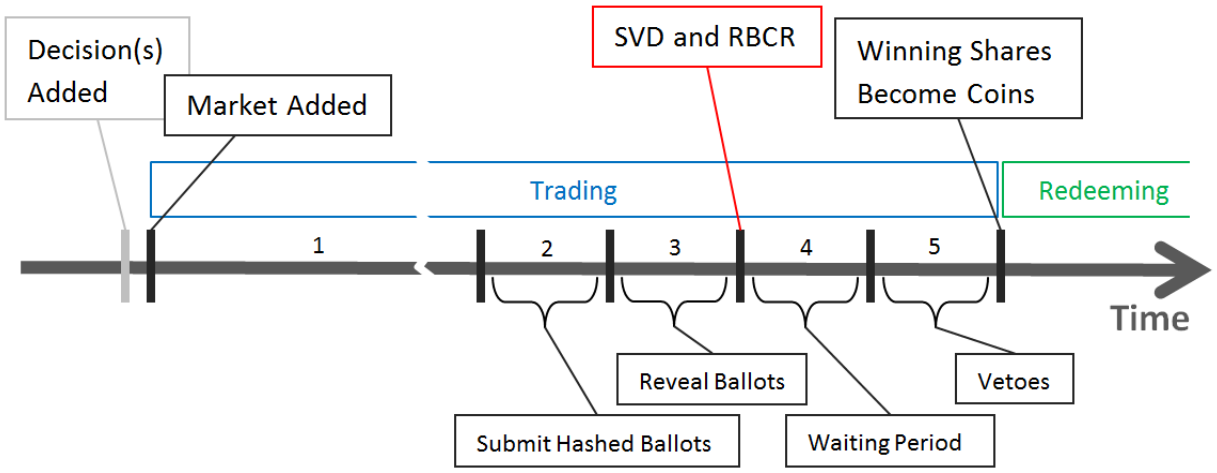


Figure 4. Timeline of a single Market (elaborated below). The horizontal axis is time, vertical lines represent points in time, brackets represent periods of time, and colored rectangles represent the status of the market. There is a break between the creation of the market and the submission of Ballots to emphasize the fact that period 1 will likely be (overwhelmingly) the longest in duration.

1) Decision Added – Markets require Decisions, making this the very first step. A "Decision Author" would select the topic-appropriate Branch, send a transaction adding the Decision, and wait for the Decision to be included in a block. Many fields of the Decision text can be submitted as a hash, and only need to be revealed later.

2) Market Added – With one or more Decisions added, a "Market Author" can create a new prediction market, by submitting some public information, the hash of some private information, and a payment, and waiting for the Market to be included in a block. The hashed data must include State dimensionality, but the component Decisions can remain hidden until after the Market has been included in a block.

3) Trading – With the Market built, it can now be advertised to traders, who buy and sell shares of the states of the Market (for example, "Buy 3.8 of State 2 of Market m16j9…").

4) Event(s) Occur – At this point in the timeline, the event(s) relevant to the Decision(s) of the Market occur and become observable.

5) Decisions Mature / Votes Cast – At this point, Voting begins on the Outcome of each of the Branch's Decisions which matured in this Intervote Period. Voters sign, and broadcast

the hash of Ballot which contains their Votes and a new public key (for which they have the corresponding private key). Critically, Voters have the option to change their Ballot (the hash) at any time and for any reason during this period (for example, to update the new public key). Only the latest included Ballot stands.

6) Votes Revealed – As this phase begins, the Votes have been included in the blockchain. No more voting can take place, and VoteCoins are now temporarily frozen. Voters reveal the message which hashes to their submission in (5), allowing these votes to be read into the consensus algorithm.

7) Decisions Resolve – Votes are run through the consensus algorithm to establish the Outcome of each Decision in the Ballot (each Decision that expired during this voting cycle). Simultaneously, the consensus algorithm allocates VTC to the new set of public keys according to RBCR (see below). The VoteCoins, which were frozen in (6), are now unfrozen and can be transferred or exchanged.

8) Audits – A Ballot may have one or more Decisions where Voters could not sufficiently agree on an Outcome. Failing to reach a consensus ultimately leads to an Audit of the "Disputed Ballot". Every Ω = 6 months, disputed Ballots accumulate in an audit-Vote-Matrix. The very same consensus algorithm of (6) is used, but with the original very-specific set of voters (owners of Branch VoteCoins) replaced with a more-general set (all CashCoin owners). See Appendix III for more details.

9) Vetoes – After a waiting period (the fourth period in the cycle, lasting τ_{review} =1 week), Miners can spend the fifth and final period accumulating vetoes for the "Resolved Ballot". If more than 50% of the blocks of this period veto the Ballot, then all of its Decisions must be re-voted on next period.

   a) Notice that software can easily remember previously-submitted Ballots, and even automatically re-submit them, so the cost to the (honest) Voter is negligible.

   b) Recall that, by the law of one price, market prices will constantly approach the present value of their expected final value. If the "expected final value" of a share is 1, some individuals (Wall St., investment-banker types) should *always* be willing to purchase such a share for $\frac{1}{1+r} - \varepsilon_{fee}$ (where r is the time value of money, and epsilon is a service fee). So (given –and this is the crucial point- that the final value *will* be accurate) the cost to the (honest) Trader is also negligible.

10) Redemptions – After a Market closes, the market-maker stops determining/broadcasting market prices, and instead uses the resolved-Outcome of Decisions to actively fix shares to their final prices. Instead of sell, Traders "redeem" these shares for CashCoin.

(iii) Consensus Puzzle Piece 1: Singular Value Decomposition

1) The mathematical process underlying the calculation of Outcomes is the matrix factorization known as singular value decomposition (SVD). Our application performs

(among other things) SVD on the Vote Matrix, which has dimension n Voters (VoteCoin Owners) by m Decisions.

2) The role played by SVD here resembles its role in the statistical technique of principal components analysis (PCA). It may be conceptually helpful to think of the RBCR function as a weighted PCA.

3) One purpose of SVD is to examine a matrix and reveal and sort its effects by influence. From SVD on the covariance of the Vote Matrix we will extract the first (most informative) component. In parallel, (for those things we cannot observe ourselves), what we decide to be 'true' is the figurative 'common denominator' among many opinions, each of which could be (and certainly is) biased, incorrect, deceptive, or otherwise non-representative. We extract "the story we believe to be most generally consistent" from the multiple eyewitness accounts we experience throughout our lives; our supportive friends, deceitful enemies, propagandist politicians, sensationalist news anchors, impractical professors, overcautious parents, reckless children, leftist Left-Party-Members, and right-leaning Right-Party-Members, together co-author our version of "the story most consistent with their combined points of view".

(iv) Consensus Puzzle Piece 2: Coordination Games

1) Imagine a game in which you have been teleported to a random location in a random city; another (randomly selected) individual has also been teleported to a random location in the same city. The object of the game is the same for both of you: to win, you must find each other (be at the same location) within 24 hours.

2) What factors would influence your behavior?

   a) Search Costs: You would like to minimize the search costs of Player Two, who is looking for you. Many places would have costly accessibility, such as night clubs (only open at night), or hotel rooms (which cost money). More importantly, a basement, or a forest, would increase the search burden of Player Two prohibitively. Ideally you'd find a news crew, or call emergency services (who are open 24/7, and already serve the function of 'coordinators'), early in the game. Making a gigantic sign that says 'Are you also looking for someone?' is costly but potentially very beneficial. Densely populated centers are better than empty, windowless rooms.

   b) Salience: The concept of salience refers to a kind of psychological perception cost. A [1] single dent in a smooth wall, a [2] bright orange vest against a grey background, or [3] the largest words of a brand label, are examples of 'salient' perceptions for which the mental costs are low. Especially salient perceptions can even have a negative cost (one must exert effort to ignore the message), as in advertising. In our game, locations would acquire salience by being uniquely functional or definitive. Economist Thomas Schelling found that the most common verbal response for the NYC version of this

game would be "noon at the information booth at Grand Central Station"[16] for the simple reason that (out of all locations in NYC) it most functions as a meeting place. Reportedly, the distant second was the (then) tallest building in NYC (in terrain there are usually many lowest points but only a unique highest point, and height has always been useful for vision [reduced search costs]), and the third most frequent response was the Statue of Liberty (a unique, large, visible, iconic place).

3) In general, humans play (and win) these games every day of their lives, by using awareness of shared human psychology to minimize shared mental costs.

(v) Operationalized Coordination Using Singular Value Decomposition (SVD)

1) SVD does not handle missing values, so if any are present (despite a Voter incentive to attend to each Decision), they are temporarily filled by reweighting the votes of everyone who did vote and forcing the missing values to adopt this as their vote (see Appendix I). This produces $V_{n\times m}^{filled}$, the completed Vote Matrix.

2) To measure coordination, we use the first score from a weighted principal components analysis. This column represents the degree to which *each voter* varied his or her votes with those of *a theoretical voter maximally representative of the covariance across all votes and Voters* (PCA automatically ranks the columns by influence, hence the choice of column 1 below).

   a) Firstly, we extract the first column of U matrix from singular value decomposition on a (weighted, using the current period's VTC balance as the weights) covariance matrix of the Vote Matrix. This is the first "loading" of the PCA.

      i) $Y_{m\times m} = Weighted.COV(\, V_{n\times m}^{filled}\,,\, r_t)\,.$

      ii) $SVD(Y) = U_{m\times m} \times \Sigma_{m\times n} \times V_{n\times n}^{*}$

      iii) $d_{m\times 1} = U_{,1}$

   b) Secondly, we build a normalized Vote Matrix, by subtracting the column-average from each column (which ensures that each of the columns would sum to zero). Finally, we multiply the normalized Vote Matrix by the first loading to get the first score.

      i) $\mu_{n\times m} = J_{n,m} \times diag\big(\, column.mean_{1\times m}\big(V_{n\times m}^{filled}\big)\,\big)_{m\times m}$

      ii) $V_{n\times m}^{norm} = (\, V_{n\times m}^{filled} - \mu_{n\times m}\,)$

      iii) $c_{n\times 1} = V_{n\times m}^{norm} \times d_{m\times 1}$

   c) Column *c* is then adjusted via scalar addition, such that the most deviant observation becomes zero. This is done either by *addition of minimum* or *subtraction of maximum*,

[16] http://en.wikipedia.org/wiki/Focal_point_%28game_theory%29

as determined by a simple rule: whichever of the two options produces outcome-ranks which minimize the difference from ranks calculated using the weights of the previous period (whichever of the two c^{adj} produces a $rank(\ c^{adj} \times V^{filled}\ )$ which most resembles $(\ r_t \times V^{filled}\ )$). When the ranks tie, the raw (un-ranked) outcomes for each option are compete on a common statistical metric: minimized sum of squared errors.

3) $c_{n\times1}^{adj} = (c_{n\times1} + (\mathrm{J}_{n\times1} \times a^*))$

4) This vector is then normalized such that all values are positive and sum to 1. The result is called the 'reputation vector'. However, before normalization a correction is applied: multiplication by previous period reputation vector over its mean. This simple correction ensures that reputation-use is additive (making it impossible to increase or decrease one's influence by separating or pooling the same amount of VTC among several accounts).

   a) $N(x) = \frac{|x|}{\sum|x|}$

   b) $r_{t+1,n\times1} = N(\ c^{adj} \times \frac{r_{t,n\times1}}{mean(r_{t,n\times1})}\ )$

5) Using the new reputation vector, Outcomes for each Decision are finally calculated. Binary Outcomes are "caught", or calculated with [1] a weighted average which is then [2] fitted to a weighted median across three choices: 0, .5, and 1. The weights for each choice depend on a Branch tolerance-parameter called 'Catch' (when Catch=0.1, the {0, .5, 1} weights are {0.4, 0.1, 0.4} respectfully). Scaled Outcomes simply use a weighted median directly.

   a) $o_{t,1\times m} = M(\ (r_{t+1,n\times1})^T \times V_{t,n\times m}^{filled}\ )$

(vi) Reputation Based Coin Redistribution (RBCR)

1) After a round of voting, Branch VoteCoins are redistributed amongst all of the VoteCoin accounts. We know where to send the redistributed VoteCoins, as each Ballot contains a new public key (destination address).

2) For each account, smooth (weighted average) the value of the previous reputation vector (r_t) with the value represented by the new reputation vector (r_{t+1}). For example, I suggest α=.20 (weighing the new value 20% and old value 80%). This parameter represents the dynamism of the voting environment: too low and an entrenched oligarchy can coast on inertia without punishment, too high and the network becomes volatile and neurotic. Recall that the most-deviant agent (even if only by a single careless error) has a zero-reputation for the current round, so they would lose the full α.

3) If (and only if) the votes are 100% unanimous, reputation-values do not change.

(vii) Temporal Economics of RBCR

1) RBCR ensures that, even in one single voting round, each Voter has one incentive to vote realistically: minimal effort. Information search costs and psychological effort will be lowest for the Realistic Ballot.

2) However, the economics of multiple voting rounds adds a second (and more important) incentive to vote realistically: revenue maximization.

3) Fees and dividends:

   a) Authors pay, in CashCoin, Listing Fees when creating a new Market.

   b) Traders pay Trading Fees (in CashCoin) while making trades on Markets.

4) These fees accumulate and are gradually paid out to VTC Owners.

5) The gradual payout:

   a) Rewards past conformity and provides an incentive to get and keep a high reputation.

   b) Offsets the constantly-present incentive to be dishonest today (and defraud traders by manipulating the Outcomes).

      i) Notably, because reputation is a tradable asset, this offset applies equally to all Voters, regardless of their personal discount rate. Individuals who wish to "retire" (as they lose interest, develop a terminal disease, etc.), would always prefer to "sell" the pristine reputation they've maintained over the years (by selling their VoteCoins).[17]

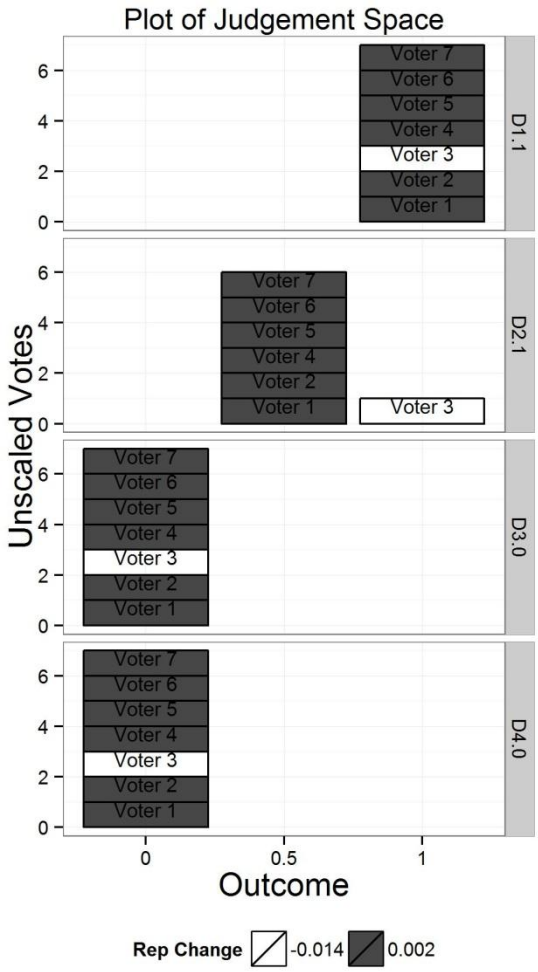
      ii) As will be shown, while the gradual payouts can be withdrawn (see "Tau Range", Appendices IV and V) as a result of Voter misconduct, the VoteCoin market capitalization is likely to be the present value of *all* gradual payouts (including those which would be withdrawn), as well as the present value of growth opportunities (as one Branch may later become two [see Article III. "Branching"], or become otherwise more desirable [see Appendix IV]). Owners must buy high (as if they intended to behave themselves), regardless of what they plan to do, yet can only sell high if they actually do behave honestly.

   c) Encourages other behaviors which maximize the future expected trading volume (good judgment, entrepreneurship).

[17] This is a major difference between Truthcoin and other, more poorly thought out ideas such as SchellingCoin and Counterparty.

$$\begin{array}{c} \\ Voter\ 1 \\ Voter\ 2 \\ Voter\ 3 \\ Voter\ 4 \\ Voter\ 5 \\ Voter\ 6 \\ Voter\ 7 \end{array} \begin{array}{c} \begin{array}{cccc} D1 & D2 & D3 & D4 \end{array} \\ \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 1 & 0.5 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0.5 & 0 & 0 \\ 1 & 0.5 & 0 & 0 \\ 1 & 0.5 & 0 & 0 \\ 1 & 0.5 & 0 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{c} \\ Voter\ 1 \\ Voter\ 2 \\ Voter\ 3 \\ Voter\ 4 \\ Voter\ 5 \\ Voter\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} D1 & D2 & D3 & D4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$



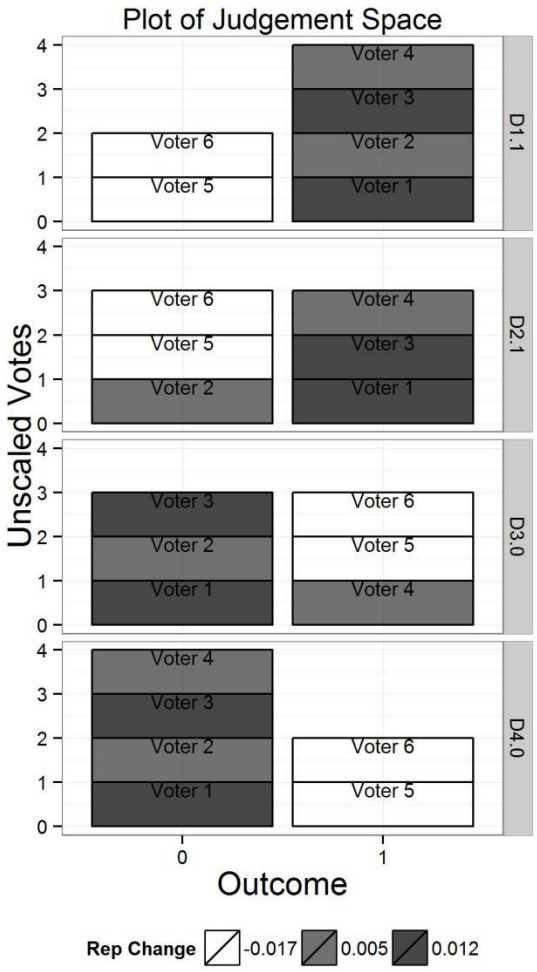


Figure 5. Two Vote Matrices and their corresponding SVD-Outcomes, represented graphically. Left, 7 Voters (matrix row, graph text-label), and right, 6 Voters. 4 Binary Decisions (matrix column, graph panel-row [right axis, above period, "D1"]), vote count (graph left axis), No/Yes outcomes (matrix cells, graph bottom-horizontal axes), consensus score (how well voters agreed with each other, graph darkness), and "correct" Outcome (graph panel-row [right axis, below period]).

Left: nearly-perfect agreement. One Voter, (#3), left the group once (Voting "1" for D2), and so his VoteCoin ownership, voting "weight", and CashCoin dividend payout all decrease (opacity). Right: notice D2 and D3, despite an apparent 3-3 tie in the quantity of votes cast for each outcome, the fact that Voters 5 and 6 were less-conformist than other voters removes enough of their vote-influence to shift the outcomes of D2 and D3 (to "1" and "0", respectively).

(viii) Voting Strategy

1) A coordination game is only a good model for the incentive scheme behind Truthcoin if malicious coalitions cannot communicate with each other in advance (and conspire to vote on the same false answers). To prevent such communications, Truthcoin provides a strong incentive for Voters to lie (to each other) about what they plan to do (the "double-agent incentive"), and votes must be kept private (or they can be "stolen"). Combined, these features prevent any voting-commitment-talk from being credible.

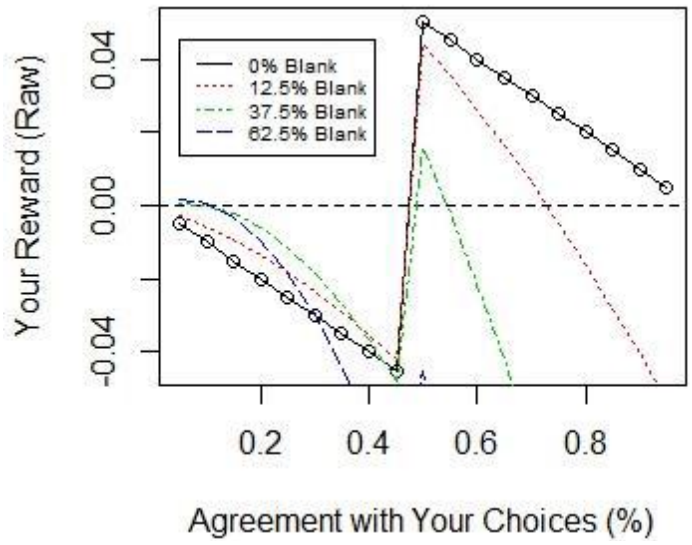


Figure 6. The "double-agent incentive", where Voters prefer to *falsely* claim that they intend to attack. As long as >50% of the Voters are honest overall, each Voter wants (reward, vertical axis) to minimize the number of honest Voters (rival-Voters who agree with your true information, x axis). This discourages cartels and "voting pools". The reward is indeed highest at 51% agreement; of course, this is perilous, as the 51% is just a few percentage points away from holding a bare minority view, which would net the lowest reward. This graphic includes information about how this reward function reacts to a Voter's failure to provide any data at all (see graph legend).

2) Notice that the SVD-procedure directly penalizes those who avoid coordinating (including those who are tricked into not-coordinating. Explicitly by design, the search costs to accurately resolve a Decision are always very low (lower than the cost of active coordination), because Voters have an incentive to flag excessively confusing Decisions by assigning them the value ".5". This incentive produces an interesting scenario: by definition, the "right answers" to the Ballot questions are always known to all Voters, and, therefore, any "wrong answers" must have been inserted deliberately. So long as a multivariate plurality of Votes are for the "right answers", SVD will directly measure "intentional deviation from those right answers", (ie, "intentional non-coordination", or "lying").

3) With the coordination game established, notice that (excluding Ballots of the same repeated answer, ie all "0", or all "1") the Ballot consisting only of "right answers" (RAs)

will be the cheapest (salience-wise) for Voters to cast. As all fully-coordinated Ballots provide the same benefit, the RA-Ballot has the optimal (lowest cost)/(same benefit) ratio. Because some agents will be most likely to select the cheapest (RA) Ballot, all agents will converge to that Ballot simply to achieve coordination. The RA-Ballot thus becomes the coordination point.

4) The availability of the VoteCoins on the open marketplace ensures that they are allocated efficiently (in other words, those who most-believe-in and are-most-dedicated-to the project will be VoteCoin owners and therefore Voters). Nonbelievers are likely to also be non-owners. Those who lose the faith would neither neglect nor interfere with the project, as their welfare-maximizing strategy is simply to sell all of their VTC.

5) While any attacker with an extremely high proportion of a Branch's VoteCoins could attempt to alter the judgment process of a Decision for personal gain, any attack with less than $(1-\Phi)=35\%$ of the voting power will fail outright, exposing the liars to huge VoteCoin losses. Recall that, because of the double-agent incentive, a coalition of attackers can never truly be sure of how many voters the truly control.

6) Optional.[18] On Decisions where fewer than Φ of the Voters agreed on an outcome, the Decisions are Audited, and Voters as a group are no long responsible for (or paid for) resolving the Decision. A dissenting individual with between $(1-\Phi)=35\%$ and $\Phi=65\%$ of Branch's VoteCoins has the ability to trigger an Audit of any Decision(s) in the Ballot.

   a) During the Audit, the set of distinct Ballots are separated into 5 maximally representative Ballots. Each of these five Ballot-options become "Decisions" in a bi-annual "Audit Resolution", in which voting is done with CashCoin instead of VoteCoin, and at-risk dividends are redistributed instead of voting-tokens (details in Appendix III and Appendix VI).

   b) The five Ballots constituting the Audit choices are [1] few in number and [2] maximally-representative, guaranteeing that, even among many widely disparate Ballots, an honest minority of 20% is guaranteed to be very easily identifiable. The remaining four options are likely to be easily rejected, as each of the four would contain many readily-noticeable examples of egregiously mis-resolved Decisions. Even when the honest minority is very small, say 10%, it will substantially influence one of the five Ballot-options (if the honest-Ballot is not itself an option).

   c) The dividend payments which would ordinarily go to Voters are instead split evenly between [1] the group of Auditors (the Auditor-group always receives half, no matter how they vote) and [2] the Voters with whom the Auditors agreed (so, only one cluster of Voters will be paid; if this cluster amounts to <50% of the total outstanding VTC, then members of this cluster actually profit as a result of the Audit). Auditor-payments

[18] The Audit is not a core requirement to the protocol design, and there are many interesting arguments against it (see Appendices III and VI).

are reweighted by SVD-consensus with each other, rewarding Auditors who cross-coordinate with each other.

d) A third point to raise, is that CashCoin which has been invested in Markets has been converted (from CashCoin) to Market-shares. As CashCoin itself (not shares) is required to audit-vote, attackers face a trade off with their money: the more CashCoin invested in a Market, the more the attacker can profit when the Decisions are mis-resolved, but the less voting influence the attacker has in the audit-process. Although ownership of CashCoins is private, in an accounting sense ("dollar for dollar") all auditing is done by third parties only – by definition.

e) Finally, note that the Audit does not change the ubiquity of the double-agent incentive. A conspiracy to push Decisions into an Audit can be profitably backstabbed, in exactly the same way that a conspiracy to attack Decisions can be profitably backstabbed.

7) A dissenting group with greater than Φ=65% of a Branch's VoteCoins would be able to successfully alter the state of all Decisions on that Branch as he or she chooses (and increase their share of the VTC through RBCR). Indeed, it is because this is the case that the project is capable of determining anything about reality at all. However, a ">Φ Group Attack" is unlikely for three reasons: stake, trust, and coordination.

a) Stake – As VoteCoins cannot be simultaneously spent (transferred) and used to vote, an 'Ownership attack' would collapse the market price of VoteCoin/USD before anyone could liquidate. As a Branch operates, adding Decisions and collecting trading fees, the market capitalization of the VoteCoins of that Branch (a function of trading fees) increases to reflect this, making a >Φ attack incur a higher and higher opportunity cost (as an attacker forgoes the money he could acquire by instead selling his VoteCoins).

b) Trust – Even an attacker-coalition which believes it has, say, 75% of the votes faces almost certain failure from a cascading fear of double-agents. A lying coalition involves coordinated deception to make a quick buck, and yet, by (costlessly) deceiving the coalition and returning to the truth, hypothetical "double-agents" can not only employ deception for a quick profit (against the attackers) but also retain the long run value of their coins. Even the leader of the 75% coalition has an incentive to betray his own strategy to scam his own coalition. It is paradoxical to require a coalition of liars to communicate truthfully, in what amounts to a massive prisoner's dilemma.

c) Coordination – Most importantly, a >Φ% coalition may fail to coordinate perfectly: members may have different priorities on which Decision they would most like to distort, and this difference of priorities provides incentives that unwind the entire distortion strategy.

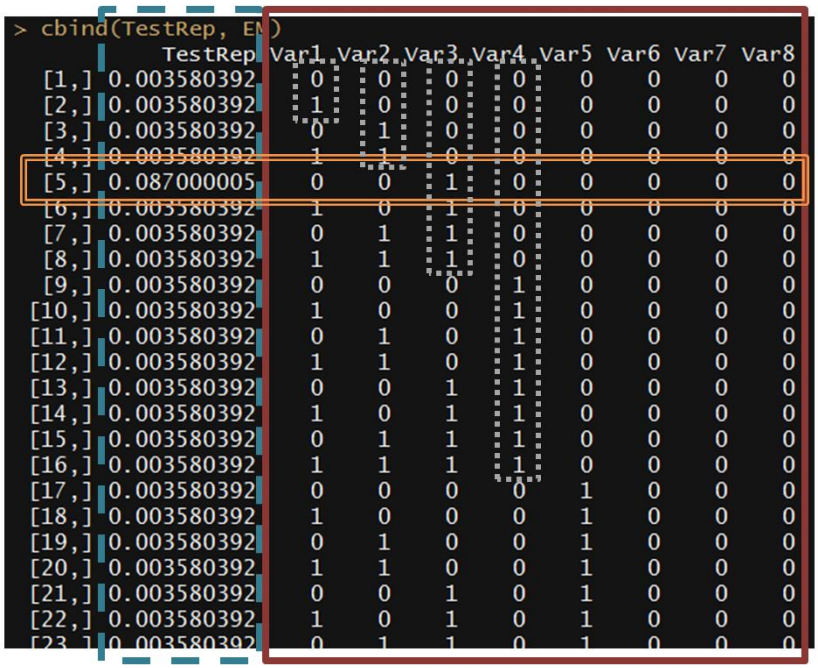
i) For an attack to be profitable, it must generate tremendous revenues during the attack-timeframe, to offset the lost VoteCoin value. To achieve a great profit quickly, the attack must distort many Markets (as each Market has a finite loss). Operationally,

this entails the purchase of cheap shares (of the realistically unlikely states) which will later be expensive after the attacker-coalition re-writes history.

ii) To succeed, the coalition must agree on the Market(s) to distort, and the False Outcome(s) they would like to use to replace the Realistic Outcome(s). Ideally, they would also agree on the total amount of money they expected to take in, and the allocation of those revenues to each participant. However, it will not be possible to manage the allocation of the revenues from the attack, because as the target Markets and Outcomes become known, participants have an incentive to buy shares of those Outcome-States until they are priced at the attack's target value. Each trade changes the price, making it practically impossible for the coalition to end up with a coordinated payout. Absent a credible commitment to reimburse (which is unlikely to exist among a coalition of liars), the coalition will have different priorities for which Decisions to distort.

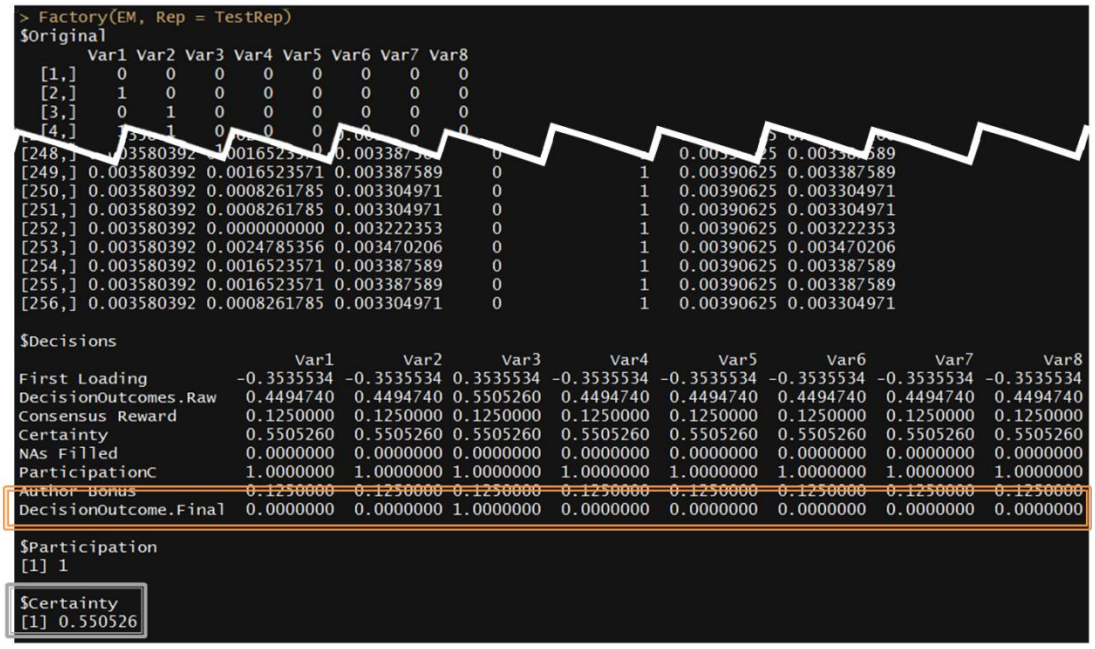
iii) The incentive mechanism pays Voters to coordinate with each other as much as possible. Therefore, those set on attacking a certain Decision would want to play realistically for all the other Decisions (that they are less-interested in), absent any convincing evidence that these uninteresting Decisions would be successfully distorted (which, again, is unlikely to exist). In other words, because RBCR considers the entire Ballot, not just the votes on one Decision, a lying coalition must be extremely complete in its coordination, even though they have every incentive to only partially-coordinate.

iv) As a clarifying example, assume [1] that Φ={0%} (audit disabled) and [2] a Vote Matrix with even as few as 10 Decisions. If there are two Voter-groups, one realistic and one whose members vote completely at random (zero coordination), the honest group needs only to control a tiny plurality, around 5%, of the votes in order to ensure that every single Decision is resolved accurately (and that they profit handsomely from RBCR). *The double-agent incentive strikes across the entire space of rival-voter-expectations.*



```
> cbind(TestRep, EM)
          TestRep Var1 Var2 Var3 Var4 Var5 Var6 Var7 Var8
 [1,] 0.003580392    0    0    0    0    0    0    0    0
 [2,] 0.003580392    1    0    0    0    0    0    0    0
 [3,] 0.003580392    0    1    0    0    0    0    0    0
 [4,] 0.003580392    1    1    0    0    0    0    0    0
 [5,] 0.087000005    0    0    1    0    0    0    0    0
 [6,] 0.003580392    1    0    1    0    0    0    0    0
 [7,] 0.003580392    0    1    1    0    0    0    0    0
 [8,] 0.003580392    1    1    1    0    0    0    0    0
 [9,] 0.003580392    0    0    0    1    0    0    0    0
[10,] 0.003580392    1    0    0    1    0    0    0    0
[11,] 0.003580392    0    1    0    1    0    0    0    0
[12,] 0.003580392    1    1    0    1    0    0    0    0
[13,] 0.003580392    0    0    1    1    0    0    0    0
[14,] 0.003580392    1    0    1    1    0    0    0    0
[15,] 0.003580392    0    1    1    1    0    0    0    0
[16,] 0.003580392    1    1    1    1    0    0    0    0
[17,] 0.003580392    0    0    0    0    1    0    0    0
[18,] 0.003580392    1    0    0    0    1    0    0    0
[19,] 0.003580392    0    1    0    0    1    0    0    0
[20,] 0.003580392    1    1    0    0    1    0    0    0
[21,] 0.003580392    0    0    1    0    1    0    0    0
[22,] 0.003580392    1    0    1    0    1    0    0    0
[23,] 0.003580392    0    1    1    0    1    0    0    0
```

Figure 7. The ability of SVD to detect coordination. Two inputs to the resolution-algorithm: Vote Matrix (red, solid, right) alongside the Reputations for each row (blue, dashed, left). This particular Vote Matrix is an exhaustive list (grey, dotted) of all possible 8-length Ballots, assuming –for simplicity– that the only Vote options are the two extremes "0" or "1". Reputation (VoteCoin balance) was split evenly amongst all 256 rows, but ultimately one row, #5 (yellow double), was nudged upward until it controlled 8.7% of the vote.



```
> Factory(EM, Rep = TestRep)
$Original
      Var1 Var2 Var3 Var4 Var5 Var6 Var7 Var8
 [1,]    0    0    0    0    0    0    0    0
 [2,]    1    0    0    0    0    0    0    0
 [3,]    0    1    0    0    0    0    0    0
 [4,]    1    1    0    0    0    0    0    0
 [5,]    0    0    1    0    0    0    0    0
 [6,]    1    0    1    0    0    0    0    0
 [7,]    0    1    1    0    0    0    0    0
 [8,]    1    1    1    0    0    0    0    0
 [9,]    0    0    0    1    0    0    0    0
[10,]    1    0    0    1    0    0    0    0
[11,]    0    1    0    1    0    0    0    0
[12,]    1    1    0    1    0    0    0    0
[13,]    0    0    1    1    0    0    0    0
[14,]    1    0    1    1    0    0    0    0
[15,]    0    1    1    1    0    0    0    0
[16,]    1    1    1    1    0    0    0    0
[17,]    0    0    0    0    1    0    0    0
[18,]    1    0    0    0    1    0    0    0
[19,]    0    1    0    0    1    0    0    0
[20,]    1    1    0    0    1    0    0    0
[21,]    0    0    1    0    1    0    0    0
[22,]    1    0    1    0    1    0    0    0
[23,]    0    1    1    0    1    0    0    0
[248,] 0.003580392 0.0016523571 0.003387589     0            1   0.00390625 0.003387589
[249,] 0.003580392 0.0016523571 0.003387589     0            1   0.00390625 0.003387589
[250,] 0.003580392 0.0008261785 0.003304971     0            1   0.00390625 0.003304971
[251,] 0.003580392 0.0008261785 0.003304971     0            1   0.00390625 0.003304971
[252,] 0.003580392 0.0000000000 0.003222353     0            1   0.00390625 0.003222353
[253,] 0.003580392 0.0024785356 0.003470206     0            1   0.00390625 0.003470206
[254,] 0.003580392 0.0016523571 0.003387589     0            1   0.00390625 0.003387589
[255,] 0.003580392 0.0016523571 0.003387589     0            1   0.00390625 0.003387589
[256,] 0.003580392 0.0008261785 0.003304971     0            1   0.00390625 0.003304971

$Decisions
                         Var1       Var2      Var3       Var4       Var5       Var6       Var7       Var8
First Loading      -0.3535534 -0.3535534 0.3535534 -0.3535534 -0.3535534 -0.3535534 -0.3535534 -0.3535534
DecisionOutcomes.Raw  0.4494740  0.4494740 0.5505260  0.4494740  0.4494740  0.4494740  0.4494740  0.4494740
Consensus Reward      0.1250000  0.1250000 0.1250000  0.1250000  0.1250000  0.1250000  0.1250000  0.1250000
Certainty             0.5505260  0.5505260 0.5505260  0.5505260  0.5505260  0.5505260  0.5505260  0.5505260
NAs Filled            0.0000000  0.0000000 0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
ParticipationC        1.0000000  1.0000000 1.0000000  1.0000000  1.0000000  1.0000000  1.0000000  1.0000000
Author Bonus          0.1250000  0.1250000 0.1250000  0.1250000  0.1250000  0.1250000  0.1250000  0.1250000
DecisionOutcome.Final 0.0000000  0.0000000 1.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000

$Participation
[1] 1

$Certainty
[1] 0.550526
```

Figure 8. SVD gets the right answer (yellow double). In fact, as far as SVD is concerned, the right answer actually got over 55% of the vote, not a mere 8.7%.

8) We now turn to the final scenario: that of a single agent (or perfectly-coordinated group) purchasing > Φ of a Branch's VTC and attacking the resolution process.

   a) While slightly discouraged by a collapse in the value of the VTC, this attack, and the potentially monumental revenues that it might generate[19], is primarily offset by the threat of a Miner Involvement (see Appendices III, VII, and VIII), which may block the attack, or, eventually, fully unmake the attack (resulting in a double-blow to the attacker: [1] failure to earn any money attacking, and [2] a collapse in the value of the attacker's VTC).

   b) Other, less effective solutions exist: For example, it is possible to force the attack to require additional scale by employing "Branch-insurance" Decisions (Decisions which state "Will anything on rival Branch X mis-resolve?" [see V. (e) (iii) 2. "Risk-Free Continuances"], and might each be located on a Branch specifically designed for this purpose –preferably one with a low Tau-Range). Furthermore, it may be possible to altogether prevent the purchase of > Φ VTC, by having at least (1–Φ) VTC (provably) owned by publically-identified parties who have signed legal (non-blockchain) contracts to report honestly.

   c) Readers who are significantly concerned with this possibility may be particularly interested in Appendix VIII ("Miners as Voters").

[19] Such a super-attacker can extract money from every other trader by virtue of his unique knowledge of each Market's final state.

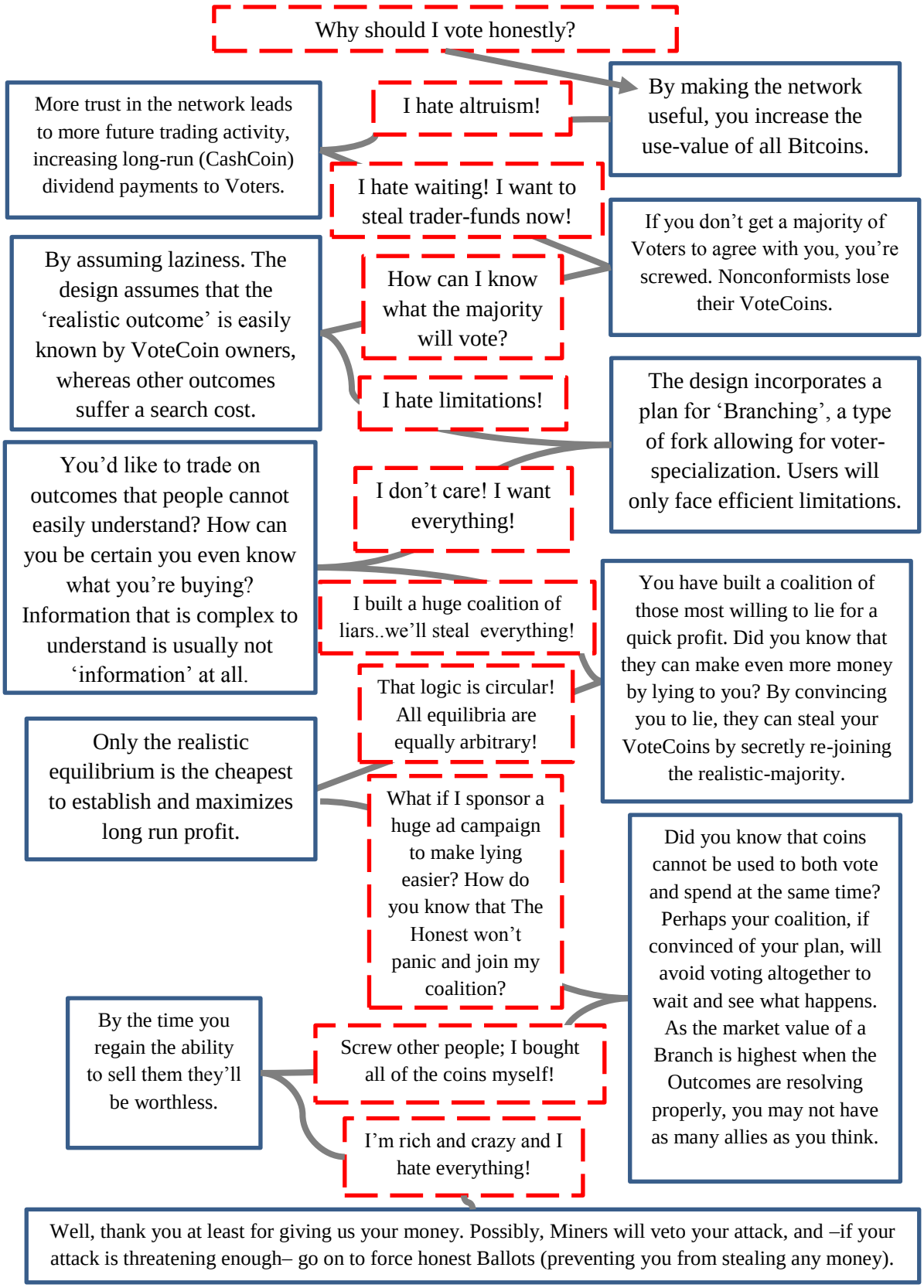


Figure 9. A hypothetical flowchart-conversation with a skeptical VoteCoin owner (red, dashed).

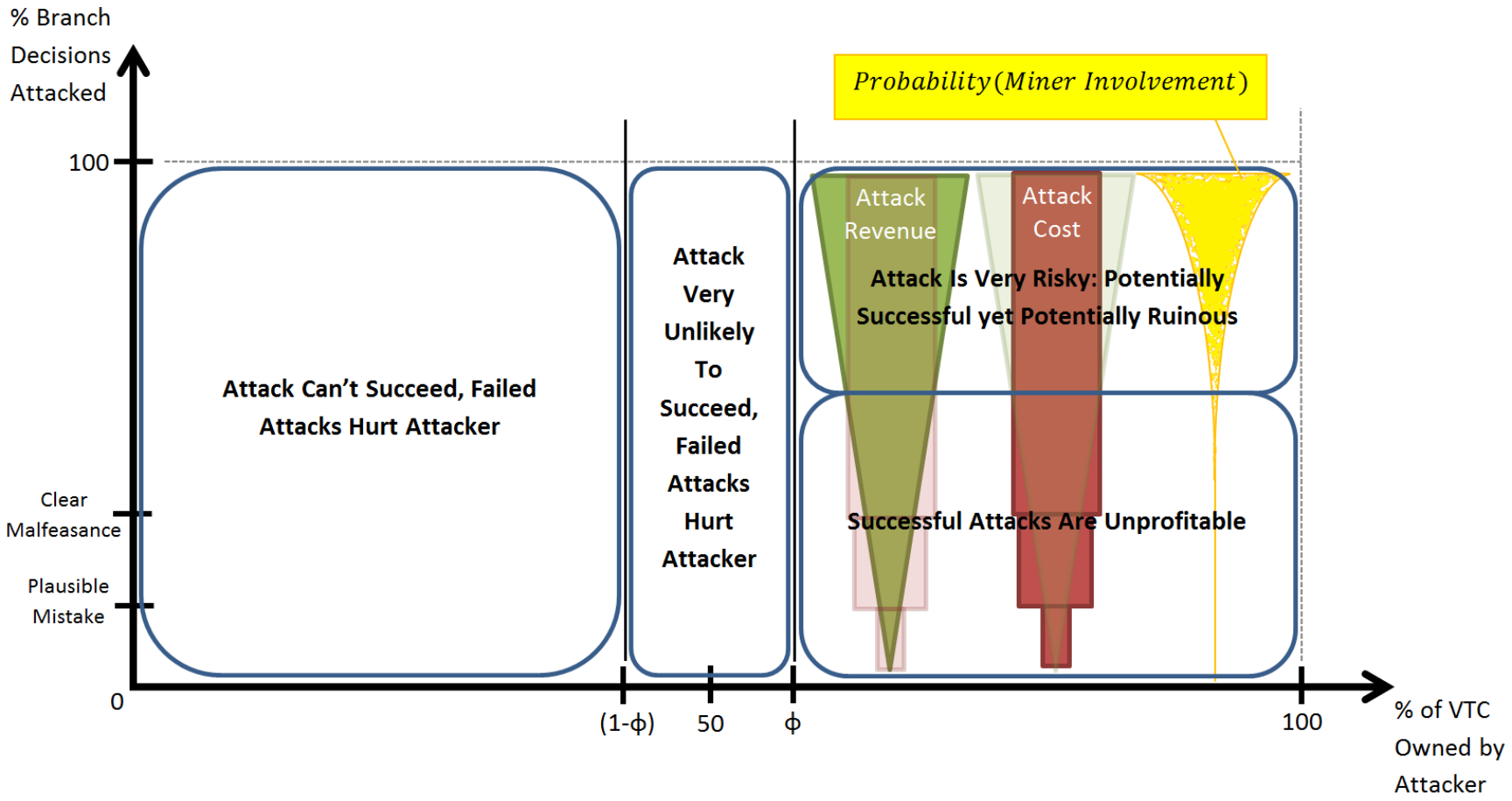


Figure 10. The more Decisions attacked, the easier it is to identify malfesance (a few obvious mistakes gives away any "purposefully mistaken Ballot"). The Audit process sorts deviant Ballots into 5 maximally-representative clusters, meaning that attacking Ballots are very easy to identify. If Miners choose to involve themselves significantly, they will get the last word, meaning that there is some liklihood of the attacker failing disastrously: not only destroying the market capitalization of the VoteCoins of the attacked Branch (and increasing the value of VoteCoins of rival Branches), but also failing to mis-resolve *any* Decisions and thus granting the attacker zero revenue.

Note that the attack cost is limited to the market capitalization of each Branch's VTC (to be precise, Φ% of the market capitalization). However, the *attack revenue is nearly unlimited*, as an attacker can *himself* place as many malicious trades as he wishes.

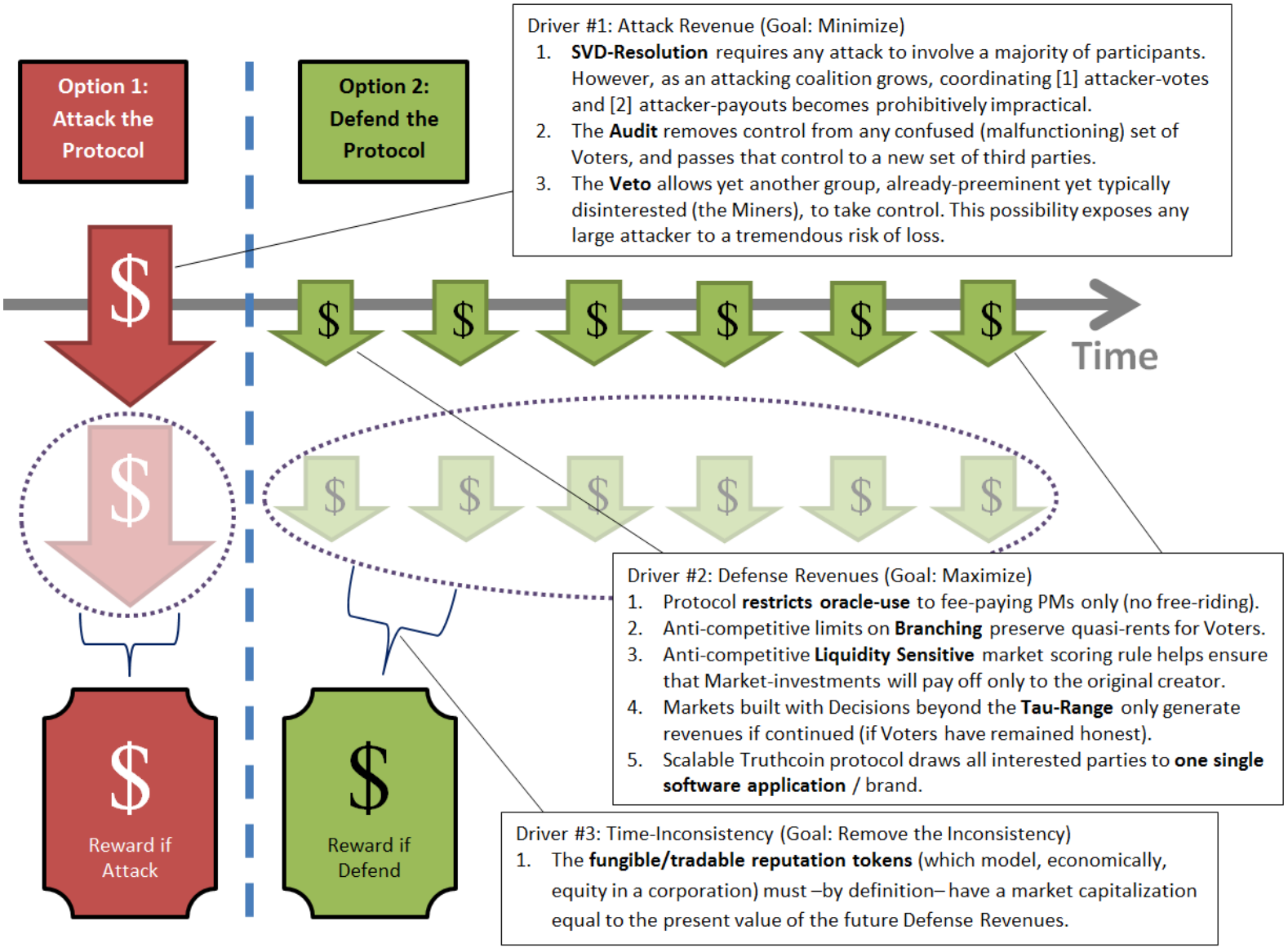


Figure 11. The three drivers of the grave "attack vs. defend dilemma". Payments over time are expressed with arrows, and payments across multiple time periods are aggregated into a single "reward". Text boxes note those Truthcoin-specific features which "drive the drivers". It seems likely that any successful decentralized-oracle-system will need to use the framework outlined here (as well as many of the ideas).

(ix) Oracle Risk: Measurement and Insurance

1) The efficacy of these protections is actually measureable and tradable, which [1] makes "oracle risk" insurable, and [2] enables skeptics and researchers to understand (and improve upon) the risks.

2) Observe this elaboration of a section of the Truthcoin timeline:

Phase	Trading	Ex-Post Trading	Voting	Redeeming
Begins When	Decision/Market Authored (Trading Begins)	Event Occurs	Decisions "Mature" (Voting Begins)	Market "Resolves" (Voting Ends, Traders Sell)
Plausible Duration	6 Months	2 weeks	2 weeks	N/A (Forever)

3) For example, a Market authored in January 2014 predicting Hillary Clinton to win the 2016 US presidential election (on November 8th) may begin its judging activities on December 1st, 2016 and not conclude them until Dec 15th. Each phase would respectfully last 34 months, 23 days, and 2 weeks.

4) Note the duration of "Ex-Post Trading" (23 days, Nov 8th to Dec 1st), during which the real-world event has occurred but no outcome-resolution activity has yet taken place.

5) Temporarily assuming a) no time value of money and b) absolute certainty that the Voters will rule correctly, post-event prices would converge quickly to their post-judgment price (for example, a "1$ if Hillary wins in 2016" contract would converge to about 1 dollar at more or less the exact moment Hillary's opponent conceded defeat).

6) If assumption (b) were violated, and there were some risk of unrealistic voting, the holdouts refusing to sell failed shares would produce a residual nonzero price, the interpretation of which would be the probability of misjudgment (or 'oracle risk'). During the "Ex Post Trading" phase, Traders can literally trade-off this specific risk amongst themselves (VoteCoin owners, or Miners, may be especially likely to make these trades), and we might use this metric to calibrate improvements.

(c) Mining Activity

1) The CryptoCoin Ecosystem

a) Merged mining allows the Truthcoin protocol unlimited use of the existing Bitcoin infrastructure. Once configured, Bitcoin Miners can advance the Truthcoin blockchain for free, and Miners are very likely to configure: they are paid transaction fees (in CashCoin), to advance the Truthcoin blockchain, and these CashCoins are 1:1 redeemable for Bitcoins (via sidechain).

b) Miners are also unlikely to allow any "parasite sidechain" (a sidechain which attempts to steal the Truthcoin oracle results) as this would result in an endless streak of parasites, each competing on price until they had "killed" (robbed of business) the Truthcoin host (and, therefore, themselves).

c) Merged-mining is free, so Truthcoin can run for free (ie, with no coinbase rewards with which to subsidize the proof-of-work clock). Non-merged-mining competitors will require their own dedicated miners, and therefore their own mining subsidy, and will therefore be more expensive than free, and will therefore be at a permanent competitive disadvantage.

2) Censorship

a) Miners cannot censor the creation of prediction markets. Adding a new Decision or Market requires only obscure details (for example hash, date / block number, and payment); the literal content of either may be withheld for several blocks.

b) Miners cannot censor votes, as they will be unsealed over a thousand block period ($\tau_{unsealing}$ = 1 week = 1008 blocks). Optionally, we could introduce anti-vote-censorship measure and ask blocks with relatively low cumulative participation to be rejected by nodes.

i) Each node can calculate (for each Branch, or in aggregate) a scalar called 'participation', which is essentially the proportion of the total network of Voters that submitted (or unsealed) their votes during the relevant period.

ii) Each node can also easily calculate the cumulative participation, the sum of participation over the previous, say, 4 blocks.

iii) Blocks can be discouraged (ignored) if there exists another orphan chain with: [1] all valid blocks, [2] similar total proof of work, and [3] Significantly higher cumulative participation.

iv) This provides censorship-resistance, because someone wishing to exclude certain votes would have to do so consistently across several blocks, which would substantially lower the cumulative participation on that chain.

v) Miners who innocently overlook a vote can simply include it in the very next block, which would only slightly lower the cumulative participation of that chain. This rule only discourages exclusion of votes across several consecutive blocks.

vi) Orphaned blocks present a perfect opportunity to 'break' a "voter-exclusion attack", as the Miner of an orphaned block, by including all censored votes in the block following his orphan, has a good change of un-orphaning that block.

vii) Large holders of VoteCoin cannot reliably execute selfish mining[20] (by withholding their own votes in an attempt to boost their block's participation) unless they also control a substantial quantity of hashpower, because cumulative participation is not only a function of the votes included in each block (VTC balance) but also of total number of blocks found (hashpower).

c) Miners are unlikely to block trades, as they collect transaction fees for every tx (including trade-txns). Moreover, VTC owners collect trading fees off of each trade, and CashCoins are most valuable when their trade-abilities are least obstructed. So, both coin types are most valuable when trading is unrestricted (and Miners have every incentive to make each coin-system they mine as valuable as possible).

(d) Authoring Activity

(i) This process is fully censorship-resistant. Any user can create a prediction market about anything, provided (s)he is willing to pay for it.

1) Prediction markets are created in two phase(s)

2) Phase 1 – Authoring the Decision(s)

a) Fee 1: $K * Fee_{\mathrm{d}}$

b) Each Decision (K) is added to the blockchain separately, at the cost of one Listing Fee. There are many options for determining the Listing Fee (see Appendix IV).

3) Phase 2 – Adding the Market

a) Fee 2: $b \log(N)$

i) Seed capital required to 'make the market'.[21] Anyone can use Decisions to create a Market for trading, but without some cost to doing so, there will be spam, waste, and needless redundancy. We therefore require all Authors to provide the small amount of seed capital required to ensure initial market liquidity.

ii) N is the number of states of the Market.

iii) b is a user-chosen market liquidity parameter (see Article IV, section (i) "Beta Amplification" for more details).

a. Low b, and this upfront cost is low, but the Market price is cheaply knocked around by Traders.

[20] http://bitcoinmagazine.com/7953/selfish-mining-a-25-attack-against-the-bitcoin-network/
[21] http://icmlmarketstutorial.pbworks.com/f/tutorial_combined_shortened.pdf

b. High b, and this upfront cost is high, but the price is more expensive to adjust. This can [1] reduce market sensitivity to large trades and [2] encourage trading. As trading fees are a percentage of trading volume (not price activity), a higher b would translate to more trading fees (if price movements were similar).

c. Authors will likely profit by selecting b based on the expected number of traders in the market (popular markets can get away with a low b [as they are already robust to large trades], unpopular markets may benefit from a higher b, as a small trader pool would imply that these traders are less likely to find each other [in a grand coincidence of market-topic and timing] and would each therefore be more reliant on the market maker).

iv) This value determines the initial account value of the Market. Although most of the funds required to ultimately pay the winning Traders post-resolution come from other Traders, this seed capital is required to make a liquid market.

b) Optional – Fee 3: $Fee_s * N^2$

i) N can potentially be very large, maximally 2^k, and each state requires the software to set aside a digital slot to count the outstanding shares, and use this data to calculate the market price. I anticipate this to be very cheap, but not free.

a. Fee_s is arbitrarily small, collected only to discourage Markets with more than N=256 states (such Markets would tend to be completely incomprehensible to most humans).

b. $f_3(N) = Fee_s * N^2 = f_2(N, b) = b\,log(N)\ @\ N = 8,\ b = 1$.

c. Therefore, $Fee_s = \frac{log(8)}{8^2}$.

ii) Alternatively, we could simply ban Markets with N>256 states.

(ii) Authors are entrepreneurial: they bear the costs of Market-creation, and benefit from the Market's use.

1) Authors cash out when their Market is closed (after all of the Market's Decisions have been resolved) and trading has ceased. If the Market had multiple Decisions, the Decision Authors split their share (25% of the Market's Trading Fees) equally.

a) The Market Author is the individual who sets the "Trading Fee Rate" (at, for example "0.1%"), which is the percentage by which Traders are overcharged. Authors (Decision Authors and Market Authors, collectively) get half of all trading fees (recall that Voters receive the other half).

b) For Markets with Scaled Decisions, Market Authors also receive a refund on their unused seed capital (Fee2). When the Market resolves to an outcome at a bound (minimum or maximum, as all Binary Decisions would), all the seed capital is used, and so the refund is zero, but otherwise this refund may be sizable. This encourages Market Authors to only use (or create) Decisions which have an appropriate range.

c) Authors therefore act as entrepreneurs:

i) Authors bear the total lifetime economic costs of a Market, by paying upfront fees for [1] the human judging activity required, [2] the working capital required to make a liquid market, and [3] the technical resources required to administer the market system.

ii) Authors bear also [4] the cost of enforcing the Market. By splitting trading fees with Voters, Authors transfer that judgment to an impartial third party, and eliminate the (crippling) requirement that Traders trust Authors.

iii) Conversely, Authors receive a payout proportional to the popularity and usefulness of the Market. Highly traded Markets serviced more trades, aggregated more information, and were more economically useful. Correspondingly, these Decisions/Markets reward their Authors with a larger pool of trading fees.

iv) The total lifetime volume of the InTrade.com Barack 2012 Market was 4.1 million shares, expiring at nearly 2.5 million shares at $10 per share.[22] Although the sum of all marginal updates to the market price, under a hypothetical LMSR market maker, is unknown, the trading fees for this Market would likely have been substantial.

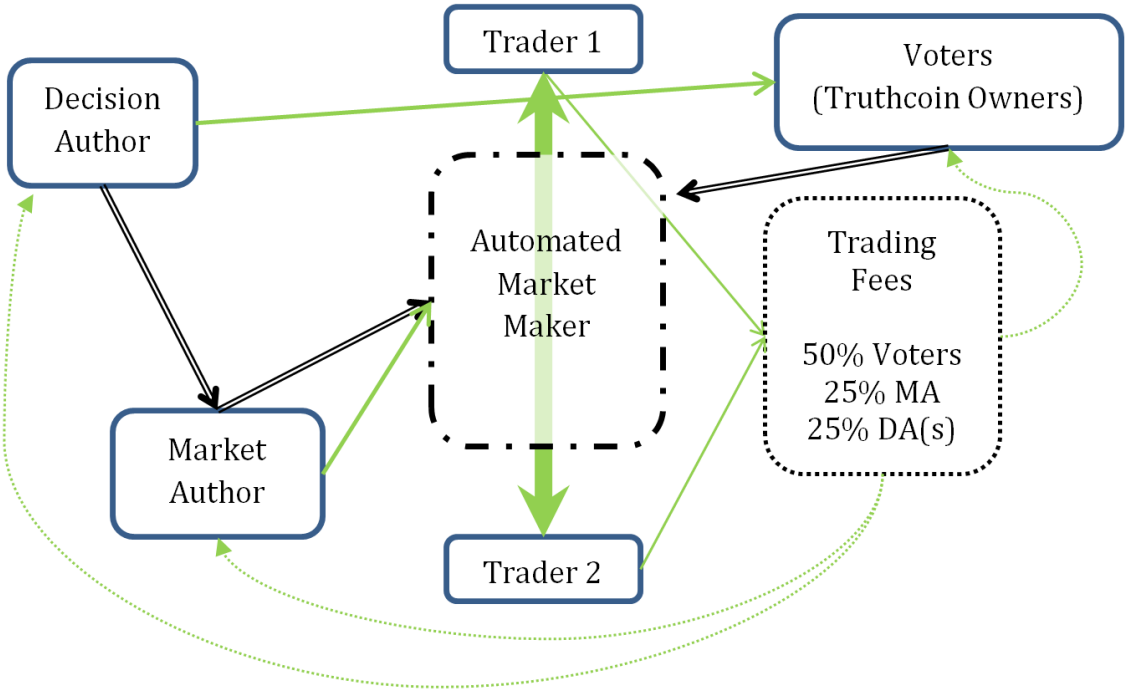


Figure 12. The flow of costs (green solid), revenues (green dashed), and information (black double) among various agents (blue solid) and accounts (black dashed). The horizontal axis corresponds to time, and line widths correspond to expected magnitudes, with the exception of revenues (whose magnitudes are a function of trading volume).

[22] https://www.intrade.com/v4/markets/contract/?contractId=743474

2) Ensuring Measurable Market States

a) Recall that the Branch votes (reports signed by VoteCoins) are scored on Consensus – i.e. how well one Voter's votes agreed with those of other Voters. Consensus relied on the assumption that reality was measurable at low search cost: all false information submitted by Voters was inserted deliberately (and not as the result of confusion).

b) For Binary Decisions, recall that it is possible to coordinate on any of three values: 0, 1, or .5 ("No", "Yes", and "Unknown"). Coordination on the value of .5 indicates that Voters { "believe that other Voters believe" }$^{\infty}$ that the True/False status of the given Decision is ultimately non-resolvable. This could indicate that the Decision text is blank, illogical, confusing, relies on inaccessible information or is otherwise unreasonable in its info/search demands. The .5 option provides a fail-safe which guarantees that search costs are low: lazy Voters will simply vote ".5" (and, because of RBCR, non-lazy Voters will mirror them) if a Decision is too confusing.

c) For Scaled Decisions, recall that the Author receives some of his market subsidy (Fee 2) back. This refund is highest at ".5" and lowest at the bounds ("0" and "1.00"). Although the central value (".5") offers the highest refund, notice that the Author is getting only his own money back (and is not netting any profits). Thus, the refund does not provide Authors with a marginal incentive to create un-measureable Decisions.

d) Unclear Decisions are unprofitable. We do not immediately know the answer to every question; sometimes, we must wait for more information (as is clear with "Will H. Clinton be elected U.S. president in 2016?", which cannot be answered until late 2016). However, we do immediately know the clarity of every question (as with "How will Clinton do in the next election?", which is too unclear). It is therefore obvious –today– what the final outcome (and final prices) will be for any unclear Decision (".5" and ".5"). This absence of disagreement about final prices (in Markets containing unclear Decisions) implies an absence also of [1] trading in those Markets, [2] trading fees in those Markets, and [3] payouts to Authors (of both kinds) from such Markets. Thus, Decision Authors have a strong incentive to only write easily-measureable Decisions.

(e) Trading Activity

1) The central goal of a prediction market is to have Traders pay for shares which they either a) sell at a future market price, or b) upon maturation of the Market, redeem at a (non-market) price which is instead a function solely of the prediction's accuracy (for example, a single share of "worth $1 if Hillary Clinton is elected" being redeemed for $1). Theoretically, efficient markets will converge "trader's expectations of likelihood of our reality matching the described state" to "the market price of that state". The automated market maker facilitates this goal by accepting 'buy' and 'sell' orders at the market price (pre-voting) and paying out at the resolved price per share (post-voting).

2) However, Traders also pay fees in the form of a small percentage (for example 1%) of each trade's cost. Competition among Market Authors will ensure that these fees are as

low as possible (likely much smaller than the implied and actual fees for existing financial/betting institutions).

3) Trading is censorship-resistant and confidential; anyone can make pseudonymous trades via CashCoin (recall that these are sidechained to Bitcoin). Each trade increases the trading fees collected, and the subsequent dividend payments to VoteCoin owners.

4) Shares themselves can be 'transferred', or passed from one keypair to another (as with Bitcoin transactions). This could be for efficiency or (optionally) even to offload trading-infrastructure to third parties. Instead of [1] selling for CashCoin, [2] transferring CashCoin, and then [3] re-buying (a cost of 2 trading fees, 3 transaction fees, price risk, and a time delay), a 'transfer' function can simply move shares among keypairs in one transaction. However, to remain incentive-compatible, this function would need to require an explicit payment to the Market of 2 trading fees.

Article III. Scalability and Customizability via 'Branching'

(a) Money Supply vs. Franchising

1) In Bitcoin, a fork occurs when the network cannot agree on a single reality. The fork results in two separate chains, each with nearly the same transaction history. All users who held 10 BTC before the fork would have two separate 'versions' of 10 'BTC' on two different forks.

2) This is spectacularly undesirable in a system designed to store value (i.e. a system of money, for example, Bitcoin/CashCoin), for several critical reasons, the chief of which is that we suffer either [1] the instantaneous and unexpected doubling of the money supply (if the chains remain separated) or [2] a full reversal of transaction history for an arbitrary subset of the currency system (if the chains successfully re-merge).

3) However, for VoteCoins, the values held by each account represent reputation and relative influence. Forking the blockchain by disagreeing on reality, or on the location of CashCoins, would indeed be as frustrating as a Bitcoin fork. However, as VoteCoin-sets ("Branches") all use the same CashCoins, and Markets exist independently of Branches, there is no way of doubling the money supply or double-spending by forking only the VoteCoins (ie copying one Branch into two, or "Branching").

4) What is possible, however, is double the supply of VoteCoins in order to half the future voting activity required on each of the two new "Branches". This could be done for several reasons: [1] because Voters are fatigued at the number of Decisions they are asked to vote on, [2] for the sake of increased competition, or [3] to change parameters (for fees, timing, etc). More interestingly, forking could eventually change the quality of the Decisions accepted for those VoteCoins ("listed on that Branch"), for example to create a "Sports Branch" or a "Finance Branch". By forking off a new Branch, all previous Owners would maintain their old VoteCoins (and with them the voting influence of their established reputation), which means that the established trust of the system would be upheld in both the new and old Branch. Eventually, some Owners would sell, or simply not use, their VoteCoins of a disliked Branch, and the Sports Branch would eventually be owned by individuals especially interested in sports. When "Sports" later splits itself into "Sports:Basketball" and "Sports:NonBasketball" (because, for example, there were just so many basketball Decisions on the Sports Branch), only the reputable sports-fanatics owning VoteCoins of the Sports Branch (and no other VoteCoin Owners) will have their voting power transferred to the two new Branches. Therefore the network grows organically, branching in the same way that a healthy tree splits new branches when the environment can support them.

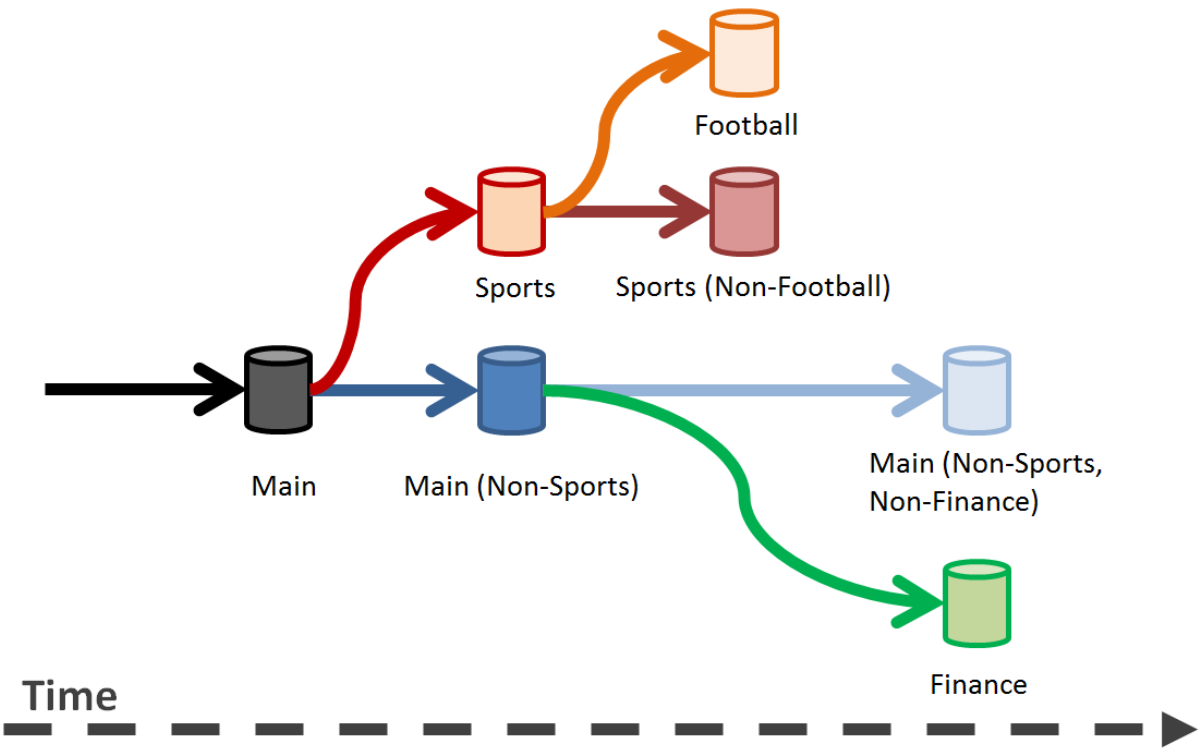


Figure 13. Holders of any Branch (cylinders) have a "free option" to own future branches. For example, anyone who owns 3 VTC on "Sport" will, when "Sport" splits, own 3 VTC on "Football" and 3 VTC on "Sports (Non-Football)".

(b) Quality Control / Digital Scarcity – Intelligent Splitting of Branches

1) An unlimited number of Branches is undesirable. First, limits create both technical and economic stability. Second, "unlimited Branches" implies perfect competition, which prevents Votecoin owners (most crucially, the earliest VoteCoin owners) from extracting Quasi-Rents (despite the fact that these rents are not only deserved, but also crucial to attack resistance and early network growth[23]). Third, the option for users with nothing at stake to alter the state of the network is highly undesirable ("any choice you give the user can and will be used against you"). Fourth, we would like to minimize blockchain storage and computation costs.

2) It is therefore ideal to have the Branching process controlled by the current set of Votecoin owners (in a sense, by the current set of Branches). If current Branch owners behave sub-optimally, they can be (profitably) bought out by new owners who will maximize the economic value (of the VTC-pseudoshares). This set of users (current Branch owners) is also, by definition, the group of users who are most able to provide vote-reports.

[23] Indeed, Bitcoin survives and thrives *only* because Bitcoin investors agree to enforce the norm of "one blockchain–currency", which endows early-adopters with *huge* early-discovery-rents.

3) If the conditions for Splitting a Branch are to be driven by Votecoin owners, we must turn our attention to an effective way of capturing Voter-preferences.

a) Formally, to ensure that a significant majority of Voters actually want to split, and have wanted to split for a significant amount of time:

i) Allow Voters to include a vote on x, a split-"decision". (Obviously, this would not at all be a "Decision" in the usual Truthcoin sense, as it would never go through resolution or be used in RBCR.) Voters would select 1 to encourage a Split, 0 to discourage a Split, and 0.5 to abstain.

ii) Define Ballot Net Growth Signal $g(b_{t,j}) = \frac{\sum x_{split}}{x_{total}} - .5$, where b_t represents the most recent Ballot to block *t*, on Branch *j*.

iii) Define Accumulated Branch Growth Signal $\sigma_{t,j} = \sum_{i=(t-25{,}000)}^{t} g(b_{i,j})$. Note that 25,000 blocks is approximately 6 months, so this translates as: "Over the last 6 months, how strong a majority desired to Split this Branch into two?" Sigma obviously ranges from (+0.5 * 25000) to (-0.5 * 25000). When Sigma exceeds some threshold, say 5000, the Branch Splits off into two Branches.

b) An alternate explanation: Voters vote on whether to Split**,** with a **cumulative** trigger of 5,000,000 %-blocks[24].

i) For example, the Branch would Split…

a. …if the Intervote Period ("Tau") is once every 100 blocks[25], and 100% of VTC-owners signal 'split=yes' 500 times in a row.

b. …if the Intervote Period ("Tau") is once every 1,000 blocks, and 100% of VTC-owners signal 'split=yes' 50 times in a row.

c. …if the Intervote Period ("Tau") is once every 5,000 blocks, and 50% of VTC-owners signal 'split=yes' 20 times in a row.

d. …if the Intervote Period ("Tau") is once every 7,500 blocks and 34% of VTC-owners signal 'split=yes' 34 times in a row.

c) Note that splitting is independent of Tau, being instead a function of the 6 month 25,000 parameter.

4) Technical/ Transaction Details

[24] A "%-block" would be the number of VTC percentage points which Voted for something, times the number of blocks over which it was voted for. Therefore, it would, take a certain amount of time (say, 6 months) to Split, regardless of the number of Intervote Periods which pass during that amount of time.

[25] Obviously, 100 blocks is implausibly brief; this is only an example to help explain the drivers of the calculation.

a) The most-straightforward way to capture this "Split Branch" vote would be simple to include it (and the parameters –see below) in every submitted Ballot.

b) After the Branch is triggered, the newborn Branch has an opportunity to take on new parameters.

i) Each Ballot may contain a set of "parameter-multipliers". These scalars will alter the parameter of the child Branch.

ii) These parameter-rescalars would be constantly updated by a weighted-median calculation (weights being across VTC balances [while equally-weighting Ballots across time]).

5) Branch Death

a) If there exists a way for Branches to be added, there should also exist a way for unneeded Branches to be removed.

b) I propose a simple rule: If there is no authoring at all for 3 consecutive Intervote Periods then the Branch and all of its VTC are removed from the blockchain database.

i) The requirement of zero may seem to be unnecessarily conservative. However, an unpopular Branch is likely to quickly become completely undesirable (much in the way that the last few guests at a poorly-attended party tend to all leave at the same time). I give more attention to the opposite risk, that the price of BTC has recently skyrocketed, making Decisions temporarily over-expensive and depressing Authorship.

ii) Malicious Voters are unlikely to artificially keep a useless Branch alive (by continuously Authoring a single Decision on the Branch to keep it from dying). Note that [1] the Listing Fees would dilute to all VTC owners (not just malicious Voters), and [2] every message would cost tx fees (which would go to Miners). This would be strange (comparable to a store owner buying his own products), but there wouldn't really be anything wrong with it, as long as this activity produces net benefits (to someone) that offset the (constantly-paid) tx-fees.

6) Branch Policy Changes

a) It would be technically straightforward for Voters to not only direct the creation of a new Branch with new parameters, but also direct changes to the parameters of their own Branch (without creating a second Branch in the process). This could easily be done with a second "governance decision": instead of a "split-decision" it would be a "policy-decision".

Article IV. Implementation Details

(a) Basic Aspects (Block Structure / Chain Validation Rules)

1) To the latest C++ implementation of the Bitcoin codebase, it should be relatively straightforward to add [1] scalar parameters (for fees, cumulative participation, etc), as well as [2] new data sets (such as VoteCoin sets, Branches, Decisions, Markets, etc).

2) Writing a new blockchain with different fields and block validation rules has already been done so many times that there are currently about 450 tradable, useable (if not useful) "Altcoins"[26].

3) There should be nothing fundamentally problematic about new transaction types. Nodes can validate any operation, be that message signing or signature verification, or the complex SVD-resolution algorithm.

4) By using a market scoring rule, there is no need for Bids or Asks, or other order book artifacts. Markets are updated instantly with a single signed message.

(b) Computational work for SVD

1) Recall that Voters select the True/False/Scalar/Unknown status of each Decision. The vote matrix is [Voters, Decisions], meaning that at 10,000 users and 500 Decisions (a realistic upper limit), the matrix becomes quite large. My testing of such a matrix on an average computer indicated that, in Python, the algorithm completed instantaneously, but, in R, the consensus algorithm ran for a couple minutes.

2) We may have to limit the total number of Voters (but not Owners) on a single Branch to 100,000 (or similar), involving a sort and filter to remove the smallest values. Those with a small amount would probably neither collect dividends nor participate in RBCD at all (in practice, they would be unable to submit their Ballots at all). If this limit is a problem (which I highly doubt), individuals can privately form (actual) corporations and jointly-control a unit of > 1/100000th VoteCoin (the minimal un-removable amount). This limit could also be increased as computers become faster.

(c) Market Maker – Near-Instant Transaction Speeds

1) Bitcoin transactions currently occur at 1 per 10 minute, with a 1 hour confirmation time. This would be acceptable, but unfortunate for a competitive trading environment. It is possible that GHOST[27] or something similar[28] will greatly improve Bitcoin transaction speeds.

[26] http://coinmarketcap.com/
[27] https://eprint.iacr.org/2013/881.pdf
[28] http://roamingaroundatrandom.wordpress.com/2013/11/30/bitcoin-idea-temporary-notarized-wallets-secure-zero-confirmation-payments-using-temporary-notarized-p2sh-multisignature-wallets/

2) Fast Sequential Intra-Block (SIB) Trading

a) The Market Maker algorithm implies an ordered transaction history (because the market price always changes after every trade). Signed messages 'trade X for Y', with nodes accepting the first received trade as valid, and allowing more trades to be built on top of this ("unconfirmed") trade, with ultimately only one timestamp ("confirmation") landing on all of these trades once every 10 minutes would still work, because double-trades will not make it far enough to steal (let alone withdraw) funds.

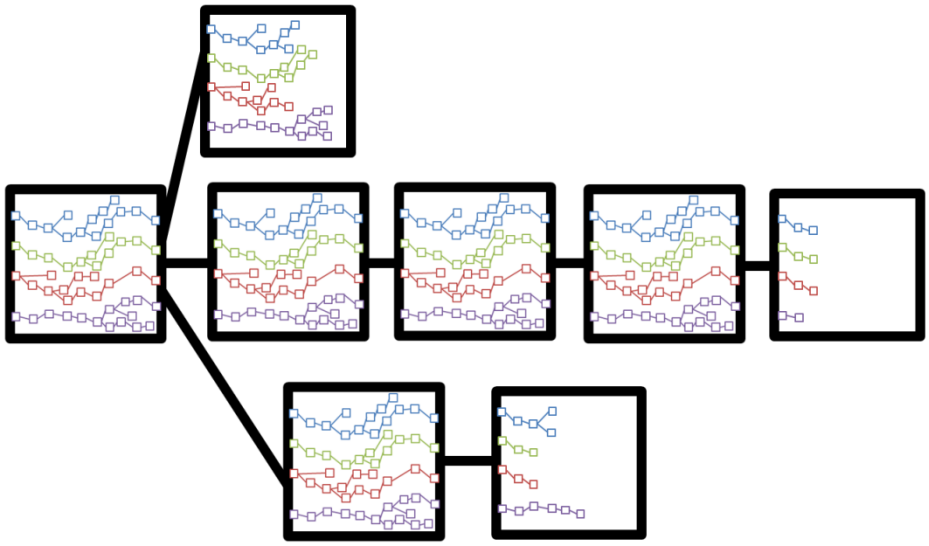


Figure 14. Blocks (large black squares) containing trades (small colored squares). Notice one orphaned block above the main blockchain and two below, but more importantly notice that each block contains many trade sequences for many Markets (colors). Some within-block sequences are themselves orphaned, but the trade-orphans are likely to be especially harmless (see below).

b) The Double-Spend 'Problem' Within-Blockchain

i) Double Spending is overwhelmingly less of a problem in within-blockchain transactions, and especially non-problematic in these within-blockchain portfolio trades. In the prediction markets described here, double-spend attempts have no adverse effects, and may actually increase overall market efficiency.

ii) The most important aspect of within-blockchain double-spends is that, as the double-spent transaction unwinds, the trade also unwinds. A traditional double-spend involves [1] the sale of a good or service for money, and [2] the attacker making off with the good while redirecting his first payment to himself. Here, however, the exchange of CashCoin for shares and back ('double-trade') takes place within the same transaction, so the double-spender ends up unwinding both the payment transaction and the trade. The net result is that nothing happens.

iii) Moreover, in building an asset portfolio, all users (attacker or otherwise) have a preferred asset allocation. As users have almost no control over which double spend goes through, any double-trade is just a pointless financial risk (to the double-trader).

iv) Although it is impossible to steal, it may be possible to confuse by making random trades and temporarily distorting prices. This is sometimes phrased 'market manipulation' with a supposed[29] psychological advantage to a trader in a subsequent trade. Although this may work in traditional markets for a variety of reasons, it has been shown that, in prediction markets, so-called manipulators actually increase market efficiency and on average improve the bottom line of non-manipulators[30]. Whatever the case may be, the protocol collects transaction and trading fees for these transactions.

c) Let us examine some hypothetical fields of a SIB transaction:

Field	Example	Description
Account	Tloxo4R…	A funded CashCoin address.
Market	Mjq11qc…	The hash of the Market.
State	2	Purchasing shares of State 2.
Amount	.03465	Total cost of this trade.
[1] Price Limit	.70	This trade is only valid if the market price of state 2 is <.70.
[2] Sequence Limit	73	This trade is only valid if there have been 72 or fewer trades on this Market since the last block.

i) Note the ability to "Limit" the trade from going through if [1] the price moves too far against you, or [2] too many trades jump in front of yours.

ii) Miners would build on the intra-block Market-tx-chains which maximized their transaction and trading fees, which would almost certainly be the longest chain.

(d) Front-Running

1) In a decentralized blockchain system, "front-running" (where one trader makes a trade, a second trader observes this trade, and 'runs in front' of the trade by copying it and attempting to have his copy included in the ledger first) may be a problem.

2) However, notice that front-running *profitably* is quite difficult: front-runners must not only be confident that they can reliably game the block-inclusion rules (which is easy), but also be confident that they are copying an informed and valuable trade (which is almost impossible).

3) We might discourage front-running by allowing Market Authors to force all trades in their Market to show some proof-of-work.

[29] http://ideas.repec.org/p/chu/wpaper/08-01.html
[30] http://dmac.rutgers.edu/Workshops/Markets/hanson.pdf

a) The PoW requirement exploits the discriminating fact that original trader can always build the transaction message (the trade) before an attacker can.

b) Professional tx-miners would put a great deal of money at risk, in an environment where they can't control the trade quality ("Half of all trades are 'losers', how was the one I front-ran?") or quantity ("Will anyone *make* a trade for me *to* front-run?"), and, ultimately, exist under (permanently unprofitable) perfect competition.

c) The Market Author can set the "difficulty" (work requirement), and even the hash function (or combination of functions), potentially leading to a completely new, responsive, tx-PoW per market.[31]

d) With block-mining, all individuals have an incentive to mine, and all miners have an incentive to improve. However, with trade-hashing, there is no existential requirement for "tx-mining", let alone that tx-mining ever be "profitable" (whatever that would mean). We might, then, enter an equilibrium where no front-running is ever even attempted (and wouldn't be successful if attempted), and remain in that equilibrium permanently.

(e) Will algorithmic ("high frequency") trading extract rents?

1) This environment has extremely competitive features (unlike those of a traditional asset exchange), and in general barriers to entry are much lower. Traders who invent creative rent-extraction methods will see those rents destroyed by perfect competition. Algo-traders may attempt to fake-out each other with fake trades, pre-trades, and other techniques, in what would ultimately be a large waste of effort impacting actually-informed traders minimally.

2) One noteworthy feature of market scoring rules (in this case, our LMSR market maker) is that orders are automatically either [1] immediately filled or [2] immediately cancelled. The lack of counterparty makes certain suspicious activities, such as constant barrage of limit orders which are later cancelled[32], literally impossible.

3) Moreover, this exchange does not employ leverage (which creates fragility and momentum), does not necessarily operate with the approval of a regulatory environment (which can allow the dishonest to operate comfortably under the illusion of consumer protection[33]), is not bound to a specific tax/fee/legal structure (which can allow 'outsiders' to be fleeced), etc.

[31] The author's humble opinion is that ASICs are (contrary to public opinion) completely harmless. Regardless of anyone's opinion on the matter, the sheer quantity of hash-function-combinations would likely prevent ASIC-use in this case.

[32] http://www.institutionalinvestor.com/Article/2617564/Markets-Exchanges/Whats-All-the-Fuss-About-High-Frequency-Trading-Cancellation-Rates.html

[33] http://en.wikipedia.org/wiki/Madoff_investment_scandal#Red_flags

4) The nature of Bitcoin Mining discourages any targeted hardware-software conspiracy (as the physical "locations" of the exchange is unknown –and changes constantly).[34]

5) It is possible that some exchange activities will privatize centrally, in a sort of 'brokerage firm'.

   a) Imagine a 'BitStamp for trading', or some website, which aggregates trades and then submits large updates to the Truthcoin network.

   b) Such an aggregation would certainly save on total transaction fees. As many trades offset each other, such a pooling of trades may also save on trading fees, yet because of the delay between trade and block-inclusions there is potentially serious basis risk on the part of the website.

   c) These privatized entities would compete on cost and quality, and would be accountable to their customers (with regard to front-running, for example).

(f) Sealed Voting: Preventing Active Coordination

1) "Sealed Votes" (where no Voter can learn the contents of a rival vote until after all voter have been cast) assist us in discouraging malicious voting by requiring all credible coordination to be tacit. Votes could be sealed in two ways.

   a) Hash Method

      i) Consider the following schedule: hash(signed Ballot, NewPublicKey), sign hash, broadcast hash, (last signed hash counts), voting deadline passes, broadcast hash contents, SVD-resolution. Finally, post-resolution, the new VTC are allocated to the NewPublicKey keypair.

      ii) Introduce a new transaction: 'StealFromLoudVoters(VictimHash, VictimNewPublicKey, TheifNewPublicKey)' which checks to see if a "VictimHash" ultimately corresponds to a broadcast Ballot that contains a matching "VotersNewPublicKey". If it does, that Ballot changes to a Ballot consisting entirely of missing votes (NA's) and the post-RBCR VoteCoins go to the "thief". The first StealFromLoudVoters to be included in a block wins. This transaction would be submitted in the same manner as the Ballots (hash-reveal style, and at the same times – [recall: pre-hash, the sealed Vote has not been cast, and so VictimNewPublicKey does not exist, and, post-reveal, the VictimNewPublicKey is known to everyone]).

      iii) Note that there is [1] no incentive to prematurely reveal the contents of one's hash, [2] no way to provably reveal only part of the hashed data, [3] no incentive to reveal the second half (votes can and will be stolen), and [4] no incentive to steal from yourself (Votes become missing). Therefore, Voters are encouraged to keep votes

[34] http://www.extremetech.com/extreme/154977-high-frequency-stock-traders-turn-to-laser-networks-to-make-more-money

private.

b) Encryption Method

i) Consider the following schedule: encrypt vote[35], sign vote, broadcast vote, voting deadline passes, reveal private key, decrypt vote. Sharing one's key before voting deadline could allow someone to change your vote (potentially in a malicious way) or outright steal your coins, so no one could reasonably ask to know your key or vote. However, votes would contain a transaction (a new keypair/address controlling next period's vote) which becomes valid after the voting deadline passes.

c) Both schemes prevent Voters from 'spending' their coins and voting with them at the same time.

(g) Floating Point Math / Decimal Precision

1) Consensus under continuous math can be a problem because computers occasionally disagree on the number of decimal places to keep, or how to truncate/round. I assume that it will be easy to implement some rule, such as truncation, significant digits, or precision requirement, so that all nodes reach the same answer and hash.

(h) Initial Allocation of Coins

1) One of Bitcoin's most successful implementation details was its distribution strategy (gradually introducing the initially worthless coins to existing users [miners] at a geometrically decreasing rate). This distribution can be easily replicated with the CashCoins (via sidechain or hard fork), but there are at least two problems with doing this for the VoteCoins.

a) Labor Problem

i) In Truthcoin, Miners only do some of the labor, unlike in Bitcoin where they do almost all of the labor. With Truthcoin much of the labor is really done through voting.

ii) The Labor Problem prevents a Bootstrap Mining Scheme as done with several Altcoins (a 'fast release' for Miners before reaching a steady state of some kind).

b) Trust Problem

i) Initial coin Owners must be trustworthy to vote, yet they will not have established a reputation. They may have "bought in" to the coin, but not bought in to the costs and benefits of Voting activity.

ii) The Trust Problem favors some kind of cost or sale, for example a Dutch Auction, donation address (Mastercoin), or burn address (Counterparty).

[35] https://bitcointalk.org/index.php?topic=196378.0

2) It may be useful to distribute the VoteCoins to developers or investors who contribute to an initial release of the software. This makes some economic sense: these individuals bore the marginal cost of adding this functionality to cryptocurrencies, so they should also own the marginal reward (use of PM infrastructure as measured by Trading Fees). This also solves the Trust Problem above: the first developers and investors sacrificed the most to construct the network, and would therefore have the most trustworthy reputation.

(i) Beta Amplification / Modification

1) All LMSR market-makers are created with a level of initial liquidity. If the chosen level is not working, it would be advantageous to be able to alter it.

2) To increase $\boldsymbol{b_1}$ to $\boldsymbol{b_2}$ at inception, the additional cost would have been $(\boldsymbol{b_2} - \boldsymbol{b_1}) * \boldsymbol{log}(\boldsymbol{N})$. Testing confirms that this cost can actually be paid mid-trading with no adverse impact upon existing Traders (and does not produce "strange" behavior, allow the market maker to run out of money, etc). Instead it adds liquidity to the markets by making the price harder to move, and, during the Amplification-transaction, moves each state's price closer to the uniform distribution (50%-50% for a binary market). It would be convenient if interested parties could donate to a Market (in the hopes of increasing its liquidity, trading activity, and accuracy).

3) Some research[36] suggests that a continuously varying $\boldsymbol{b}$ would achieve more desirable combinations of cost, profit, and liquidity. This may be helpful if, for example, Traders are sufficiently more likely to trade today in markets which will become more liquid tomorrow, or if some Traders will only trade after a personal liquidity threshold has been crossed.

   a) A Liquidity-Sensitive MSR overcharges all traders, and provides liquidity which scales with open-interest. It therefore imposes (ceterus paribus) a net cost on early traders and a net reward to later traders.

      i) First, this feels backward: the earliest trades are the most important, as they may generate 'buzz' which draws in later traders. Therefore, we would ideally (somehow) subsidize early traders at a net cost to later traders.

      ii) However, the "overcharge" is not as bad as it might seem. While traders are overcharged in the LS case, they are also proportionally over-compensated when they sell, if sales are at similar prices. The overcharge is the greatest at central (uniform) values (such as 50% in a two-state market), and approaches zero at edges (0% or 100%).

   b) The LS-MSR is more realistic: in the real world, the traders who buy into a market earliest [1] suffer by participating in a relatively illiquid market (where their trades are

[36] http://www.cs.cmu.edu/~aothman/flex.pdf

restricted), and [2] add to the market's existing liquidity. (Sellers, with MSRs, need to have already bought shares in order to sell them, so only the buy-comparison is relevant).

c) A very interesting benefit to the LS-MSR is that it discourages Market Authors from copying already-existing Markets. If several similar markets existed, some LS and others non-LS, and traders placed a buy-premium on liquidity, arbitrage would still force some trading volume to diffuse across all Markets (such a diffusion would be redundant, disorganized, and undesirable). Over a long enough time, at least one LS-market is likely to reach a state where it has surpassed all non-LS markets in liquidity. At this point, the most liquid LS-market of a topic would begin to attract a highly disproportionate amount of trading (all non-arbitrage trading), as it would be the most-liquid overall.

d) If traders place a buy-premium on liquidity, and all markets are LS, we can offer some guarantee that the first Author to introduce a (liquid-enough) Market will be the unique benefactor of the resulting trading volume (as the most liquid Market would get all of the trades and trading fees). This may help prevent a hold-up problem where Authors are afraid to invest in great Market concepts, for fear that their entrepreneurship will be stolen by copy-cat rivals.

(j) Trading on Events with Bounds

1) Scaled Decisions cannot be unbounded (ie, it cannot be possible to trade on [or vote on], any number at all for "What is the price of gold on Date d?"), as this is intractable for both trading and SVD. Scaled Decisions must have some upper and lower bound.

2) Nonzero Minimums Introduce Unwanted Leverage

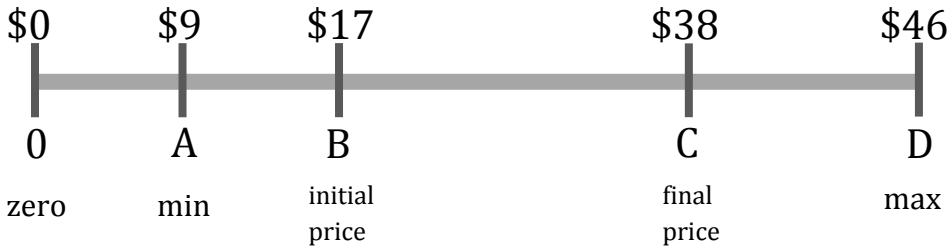
a) Decisions can easily be 'scaled' up and down by a scalar without consequence (in finance, only percentage-returns matter). This simply displays the Decision in different units. However, more complex is the 'shift' of a Decision's range by a minimum value. One design choice would be to leave the technical-minimum at zero, knowing that trading below the true minimum would be irrational and never take place. The most obvious disadvantage to this is that the Market pre-allocates liquidity which is never used, potentially a great deal of liquidity (failing to take full advantage of the LMSR market-maker). This would also complicate the Author incentive to set appropriate bounds (as ".5" would not be the point at which the market-maker has the maximum amount of refundable cash), as well as Voter incentives (as ".5" is no longer the information-less prior).

b) A more desirable option is to simply mix the over-levered portfolio with cash, to proportionally de-lever it. Assume a single-Decision Market ranging from $9 to $46. Also assume that the current price is $17, and lastly that the present value of the final price is $38. A purchase of one share today should ultimately produce a return of (38-17)/17 = + 123.53%. However, with a min of 9, the return produced would be

unknowingly shifted by 9 units, to ((38-9)-(17-9))/(17-9) = + 262.5%, which is substantially higher.

c) As the relevant range over which one can trade is inversely related to the minimum value (which is always ≥0), the range will always be "too small" and therefore the actual return will always "too big".

d) In fact, the return is always "too large" by a calculable proportion . If one only purchases a certain fraction ("h") of the over-levered "actual" portfolio, the total (cash + shares) portfolio will produce a return identical to that of the "expected" portfolio. This fraction is a function of the current market price, and therefore must be calculated per trade. Once set up properly, the algebra is easy:



$$You\ Got: \quad \frac{C-B}{B-A} = Ratio_1 = r_1$$

$$You\ Expected: \quad \frac{C-B}{B} = Ratio_2 = r_2$$

$$h * r_1 = r_2$$

$$h\left(\frac{C-B}{B-A}\right) = \left(\frac{C-B}{B}\right)$$

$$h = \left(\frac{B-A}{B}\right)$$

e) So, the solution is, for each purchase of 1 share, to instead purchase h (which is between zero and one) shares, and set aside (1-h) shares worth of cash. This might involve the purchase of BitUSD or actual USD (or local fiat currency), if the user does their accounting in a non-Bitcoin currency.

Article V. Appendices

(a) Appendix I – Calculation of Missing Values

1) SVD cannot be performed on a matrix with missing values.

2) To fill any missing values, a simple procedure is used:

   a) The Decision Outcomes are calculated using all available data (ie, for all votes that were cast for a Decision). The previous period reputations are used (as the present period reputations do not yet exist) and they are renormalized by dividing by their sum.

   b) For Binary Decisions, the calculated values are then binned, according to the Catch parameter, into one of three values: 0, .5, and 1, as these represent vote-format. Scaled Decisions are calculated using the weighted-median, as always.

   c) Each Decision has all of its missing values replaced with the calculated outcome.

3) Non-Voters are, later, penalized according to the following scheme:

   a) Calculate "Participation" for each Voter i as $P_i = \frac{\sum \#VotesCast_i}{\sum \#VotesExpected_i}$, and in total as $P_{total} = \frac{\sum \#VotesCast}{\sum \#VotesExpected}$.

   b) Calculate each voter's "Relative Participation" as $P_i^{rel} = \frac{P_i}{\sum_{i=1}^{n} P_i}$.

   c) Finally, merge the new smoothed RBCR value with this Missing Values value, in direct proportion to the total (1- Participation).

$$r_{final} = (r_{t,n\times 1}) * (Participation) + P_i^{rel} * (1 - Participation)$$

4) This ensures that the penalty for missing a vote is small when few votes are missed, but severe when many votes are missed. If more than 50% of the votes are missing, this "penalty" actually becomes a more important determinant of future coin values than agreement with other voters (as it should, because the "other voters" are not actually voting).

(b) Appendix II – How Resolved-Outcomes Translate to Share Prices

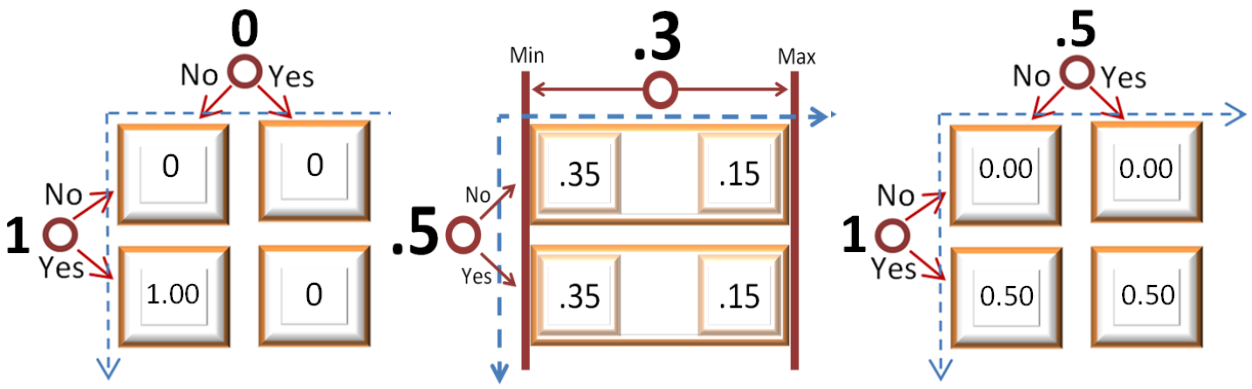


Figure 15. Three Markets, each with 2 Decisions and 4 States. The left Market had Outcomes of 1 and 0, the center Market had Outcomes of .5 and .3, and the right Market had Outcomes of 1 and .5. The final sale price is given inside each State-box, constructed by multiplication (precisely as joint probabilities are constructed from marginal probabilities).

The leftmost market is most straightforward: a Binary variable where the row-event happened but the column-event did not. Owners of the appropriate share can earn 1 unit each, owners of other shares get nothing. The center market involved at least one scaled Decision (the column-event), which resolved to ".3".

If a Binary Decision is ruled unresolvable, the winning State of any Market built with this Decision cannot be determined. However, we can preserve the utility of any Market built with an unresolvable Decision by causing that Outcome to take on the equally-spaced value of ".5".

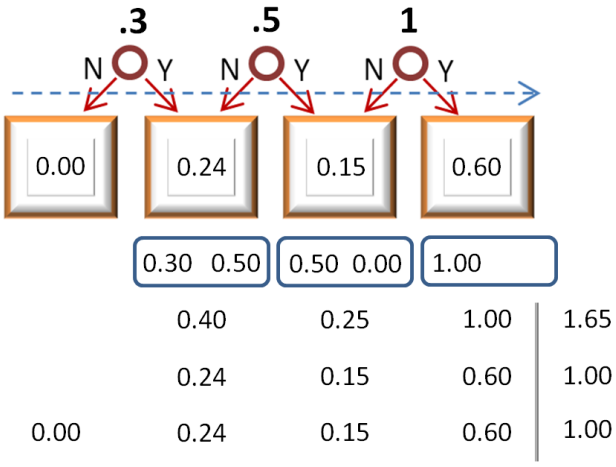


Figure 16. Mutually-incompatible reports are handled by simple averaging, and then renormalization (division by sum). Finally, State 1 is determined by subtraction. In the example above, it may seem strange that State 1, which would seemingly deserve 0.7 (1.0-0.3=0.7), instead ends up with zero. This is the result of the logical interpretation of Binary Decision question text (and deliberate mis-use of Scaled Decisions in this example). In practice, multiple Scaled Decisions would never be used along the same dimension, as a single Scaled Decision would already span the entire dimension (albeit on a restricted range along that dimension).

(c) Appendix III – Extra Truth-Layers (Auditors and Miners)

1) Recall that, typically, half of the total accumulated Trading Fees would go proportionally to Voters. However, in the case of an Audit, the Voters have failed to fully earn these fees, and so half of the Voter-half (25% of the total) goes instead to the auditors, and the other half-of-half goes proportionally to the Voters with whom the auditors agreed. Therefore, even a minority (<50%) of Honest Voters will profit disproportionally (as if they owned 50% of a completely honest Branch) by sticking it out with honest answers. The disputed Ballots are resolved via the same SVD-Consensus, but using all CashCoins instead of one set of VTC. As only unspent CashCoin can be used in Audit-voting, these voters are necessarily third-parties. A single Audit-SVD-resolution can include all disputes from all past Vote Cycles of all Branches.

Case (Agent)	What is being Voted on?	SVD-Consensus Weights	Laziness Policy	Non-Coordination Penalty
Normal (Voters)	Individual Decisions of the current Vote Matrix of a certain Branch.	One Vote per each VoteCoin on the relevant Branch.	Agents must report (on Standard Decisions), or be penalized.	Agents lose ownership of the VoteCoins that they purchased, and the associated dividend revenue.
Audit (Auditors)	1 of 5 Representative Ballots (these Ballots are constructed from a failed Normal Case vote).	One Vote per each "free" CashCoin a user is willing to lock up with a Vote ("Audit-Ballot").	Agents have no direct obligation to report, can ignore whole process.	Agents don't receive as large a stake-adjusted portion of the Voter Transaction Fee Pool as they otherwise could have.
Override (Miners)	1 Ballot of those submitted in the failed Normal Case.	One Vote per block found.		

2) Miner Veto: For all Ballots, including Audit Ballots, Miners may set their own "Miner Ballot", and Veto Ballots which do not match it. If >50% of Miners Veto a Ballot, it has no effect; Voters must try again during the next Voting Cycle.

3) Miner Override: Miners can also do their own SVD-vote, forcing one Ballot to be the "correct Ballot" (see Appendix VIII).

Source of Forecast-Correction	Cost	Network Capacity	Expected Throughput (Usage)
Traders	One LMSR trade	Very High	Very High
Voters	n Votes, one SVD proc, one Intervote Period	High	High
Auditors	effort from CashCoin owners (optional, easy, but unexpected), considerable delay	Very Low	Very Low
Miners	unexpected need for a coordinated effort from disinterested Miners, potential network-instability	Extremely Low	Extremely Low

Figure 17. The cost-of-truth has been matched (triangles) with the realistically-expected usage: more expensive truth-sources are rarer.

(d) Appendix IV – Regulating the Cost and Supply of Decision Slots

(i) Motivation

1) Protocols should operate with a minimal degree of user-input: while the protocol is established permanently, awareness of protocol-weaknesses grows over time. Hence, any choice given to the user "can and will be used against you". Already, the protocol is flexible to a hazardous degree.

2) Our challenge is to clear the market for "human attention" (the Decisions represent a service provided, and the production of this service consumes human labor, specifically: attention). To do this, we will need to simulate an actual market: supply and demand intersecting at a quantity and price.

3) Each Branch has a Maximum Quantity of Decision-Obligations ("Decision Slots"), which is a Branch parameter (and determines the amount of human-work the Voters obligate themselves to do in a given Tau). The supply, Q_{max} , is therefore given.

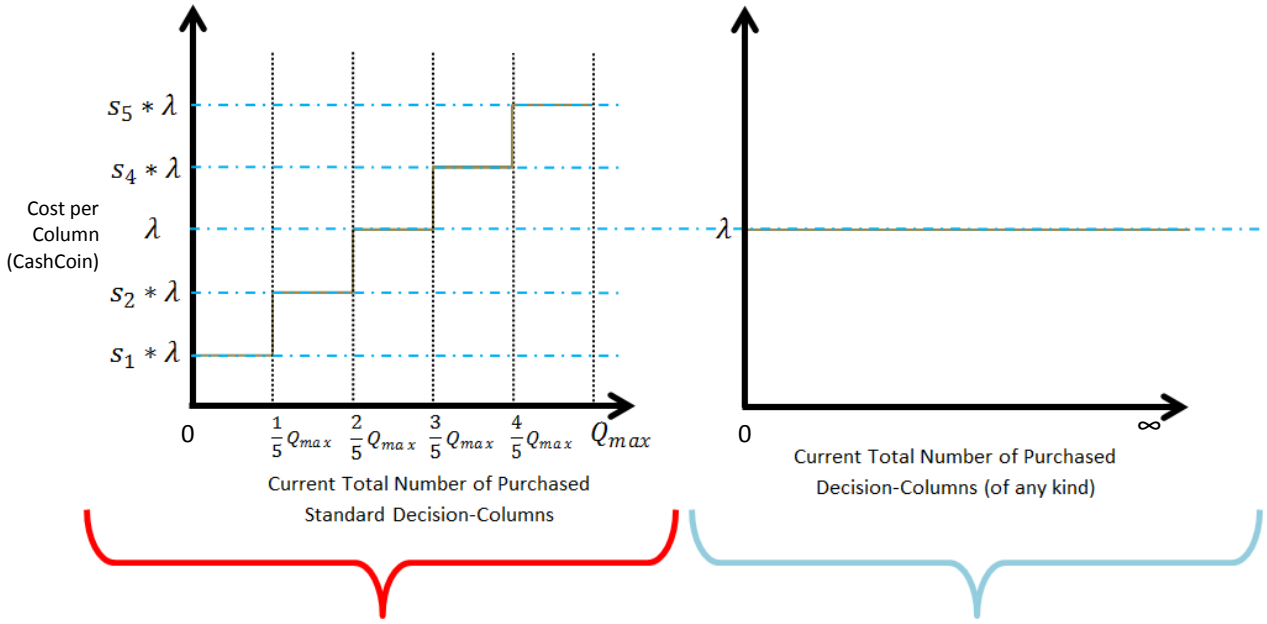
4) The price of a Decision-Slot will ultimately be expressed in the units (BTC / Decision-Slot), so, fundamentally the price will need to be driven by the supply-of/demand-for both [1] BTC and [2] Decision-Slots. The supply of BTC is fixed, as is the supply of Decision-Slots (over the time horizon relevant to this price-calculation[37]), but the demand for BTC alone is extraordinarily volatile: the price-calculation will need to be very sensitive to accommodate this.

5) Demand for Decision-Slots can be measured using revealed preference theory, where deductions are made by observing marginal purchases. The final (and least tractable) driver, the Demand for BTC, will need to be inferred from sales of Decision-slots at fixed prices.

(ii) Cost of the Decision-Slots

1) To simulate a supply curve, I use a simple shape which depends on just a single λ parameter. Feeling that a smooth supply curve would be too psychologically burdensome to users, I instead partitioned the curve into 5 flat levels.

[37] By this I mean that changes in Branch quantity (the Splitting of a Branch into two, or death of unused Branches), would –over a longer time horizon than is considered here– change the maximum global quantity of Decision-slots.



Standard Column-Slots	Overflow Column-Slots
Cost between $\frac{1}{2}\boldsymbol{\lambda}$ and $\boldsymbol{2\lambda}$.	Cost λ (always).
Voters must vote on these (or suffer a penalty).	No penalty for not-Voting.
Votes cast here are used for RBCR.	Votes cast here are not used for RBCR.
Here, Voters earn Trading Fees on all the Decisions in this set.	Here, Voters only earn Trading Fees/Listing Fees[38] for the Decision(s) on which they voted.

Figure 18. Standard Column slots (for which an answer must be provided) become more expensive as the quantity sold increases. Overflow Slots (buyer beware, no guarantee that any answers –valid or otherwise- will be provided) are a constant price. As Standard slots are superior, no one would consider purchasing an Overflow slot until at least 3/5ths of the Standard slots were purchased. The price feedback mechanism is therefore doing two things: [1] providing an incentive to get Standard ("must be answered") Decisions into the system (for a large, healthy, SVD), and [2] only measuring the psychic costs of a *requirement* to vote (as one would only pay greater than $\boldsymbol{\lambda}$ if this *requirement* were desired).

2) The s_i (scaling factors) along the vertical axis in the graph above (W=2) are as follows:

Section:	s_1	s_2	s_3	s_4	s_5
Raw:	$\frac{1}{W}$	$mean(\frac{1}{W}, 1)$	1	$mean(1, \frac{W}{1})$	$\frac{W}{1}$
Log:	$\frac{1}{W}$	$\frac{1}{e^{(\log(W)/2)}}$	1	$\frac{e^{(\log(W)/2)}}{1}$	$\frac{W}{1}$

[38] Listing fees are never zero, so –even with no trading whatsoever– there always exists *some* incentive for Voters to vote on all Decisions.

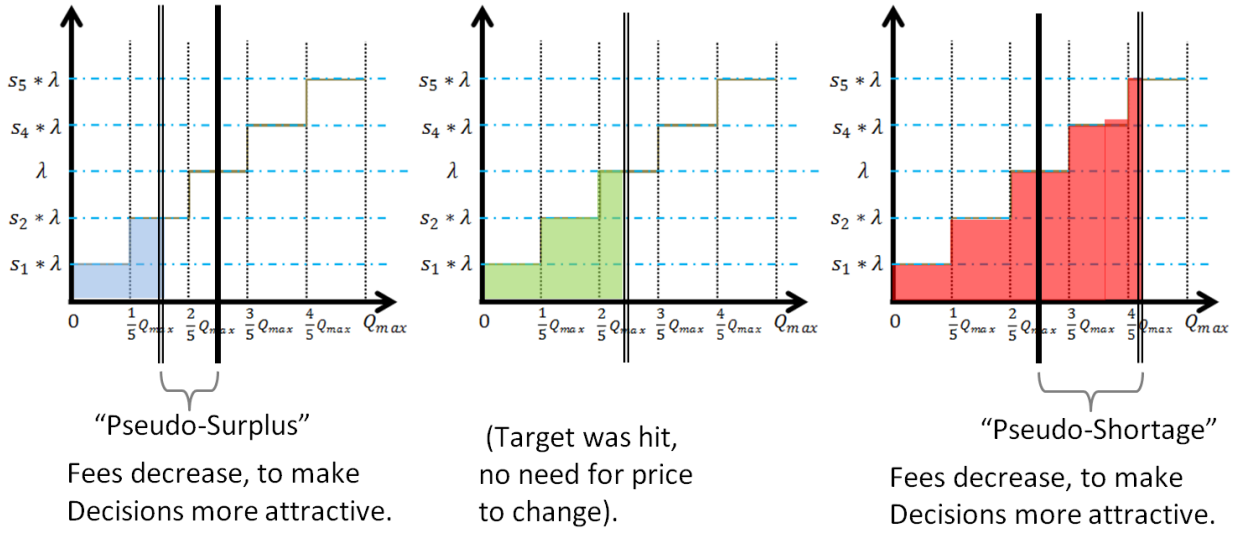


Figure 19. The effect of sales on Listing Fees. The central vertical black line intersects the horizontal axis at the target sale-quantity, and the double vertical black line represents the actual sales-quantity. When the target is hit exactly (center), $\boldsymbol{\lambda_{t+1}}$ does not change. When sales are under target (left), $\boldsymbol{\lambda_{t+1}}$ decreases, and when sales are above target (right), $\boldsymbol{\lambda_{t+1}}$ increases.

(iii) Computation of Resolved Overflow-Decisions

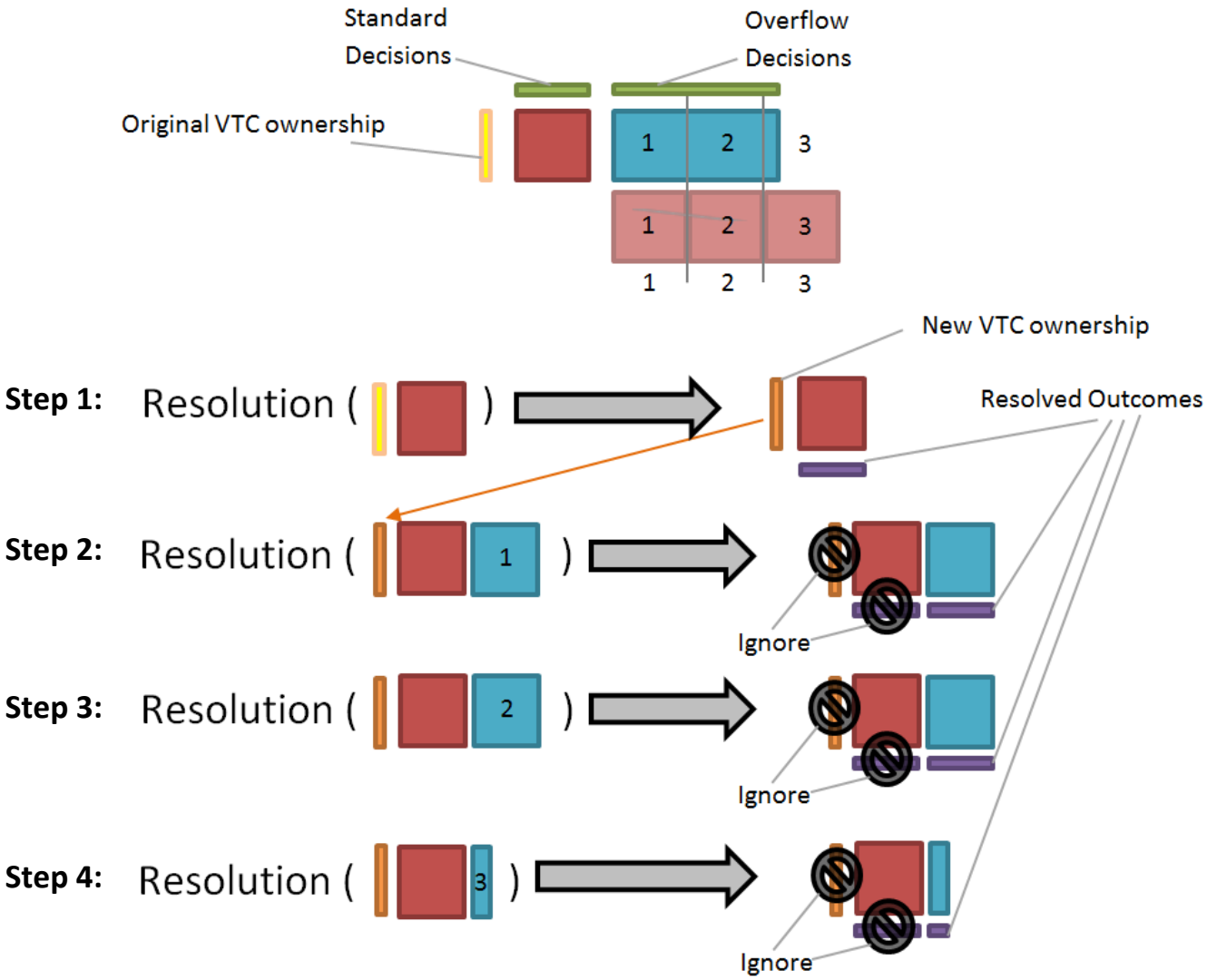


Figure 20. The resolution operation has approximate computational cost $\boldsymbol{O(m^3)}$, where m is the number of Decisions.[39] To prevent the resolution-algorithm from ever needing to do computations on an overly large matrix, we can simply batch the process, which moves the computational cost closer to $\boldsymbol{O(m * Q_{max}{}^2)}$, where $\boldsymbol{Q_{max}}$ can always be expected to be ≅500. In this example, the task has been split into 4 batches.

[39] The major operation is SVD, at cost $O(mn^2)$, where m and $\boldsymbol{n}$ are dimensions of an $m \times m$ covariance matrix.

(iv) Listing Fees and Branch Quantity

1) For every identical set of VTC-owners, Authors will always buy the cheapest Decision-slot available. The splitting of a Branch (which produces two identical sets of VTC-owners) results in a realistic "flooding" of the market with cheap Decisions: total revenue from Listing Fees (recall that Listing Fees are to offset the psychic costs of vote-labor, and Trading Fees are to offset the temptation to lie) will *only* increase if each Branch meets a threshold for Decision-sales, otherwise, revenue *decreases*.[40]

2) The split-threshold is at approximately 272% of the target of a single Branch, or 136% of the target for two Branches (where one Branch would be completely full, and a hypothetical second Branch would be at least 36% full). Importantly, this threshold is a gamble: by splitting in a case where the threshold wouldn't be sustainably met, toal revenues are *lower* than the revenues from a single un-split Branch. Therefore, risk aversion on the part of VTC-owners would further discourage a Split of one Branch into two.

3) We therefore have a case where the creation of new Branches is highly discouraged, being profitable only when there is substantial user demand for more Standard Decision-slots. This severe discouragement to Branch-creation (which tightly limits the number of standard Decision-slots) is offset by the flexibility of the user to add a potentially unlimited number of overflow Decisions-slots.

4) This analysis ignores a great deal (demand-side effects, price-elasticity). Specifically, it might be most profitable to never Branch, and constantly allow the price-feedback system to greatly increase the price. To discourage this, note that, if there are at least two Branches, the assumptions for Cournot Competition[41] are all met (as Decision slots are mostly-identical, production is deterministic and immutable, and sales cannot be refused). Cournot Competition can be maintained if the protocol always has two Branches (a simple way to do this would be to start with two 'Alpha' and 'Omega' Branches, and make them un-killable).

[40] Notwithstanding a lower edge case where "total revenue is the same, at any number of Branches" (encountered where all sales remain in the first pentile, seen below in the leftmost two points). By improving the analysis to consider the (uncertain) future sales across several future Voting Cycles (as one should), the edge case disappears.
[41] http://en.wikipedia.org/wiki/Cournot_competition

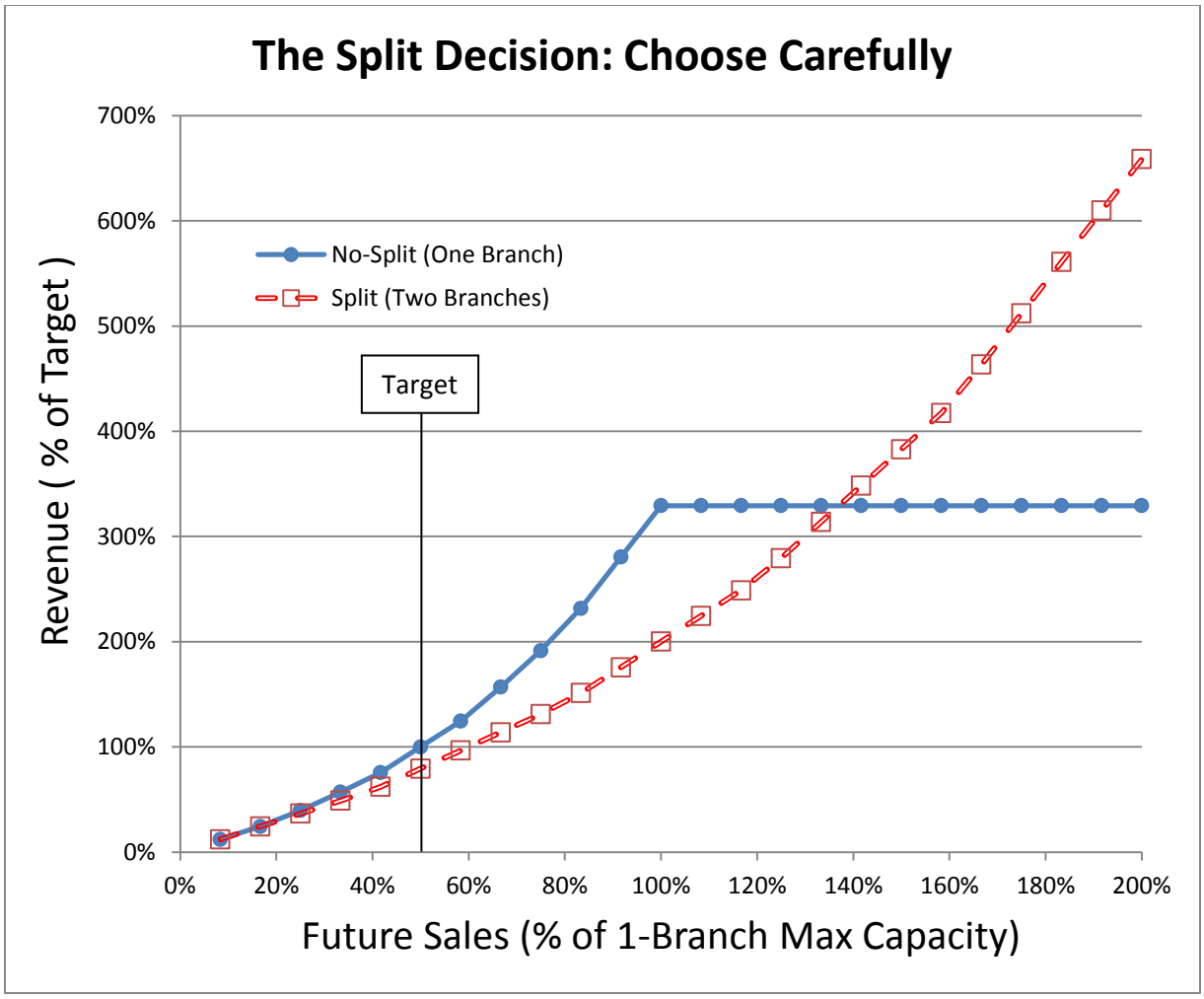
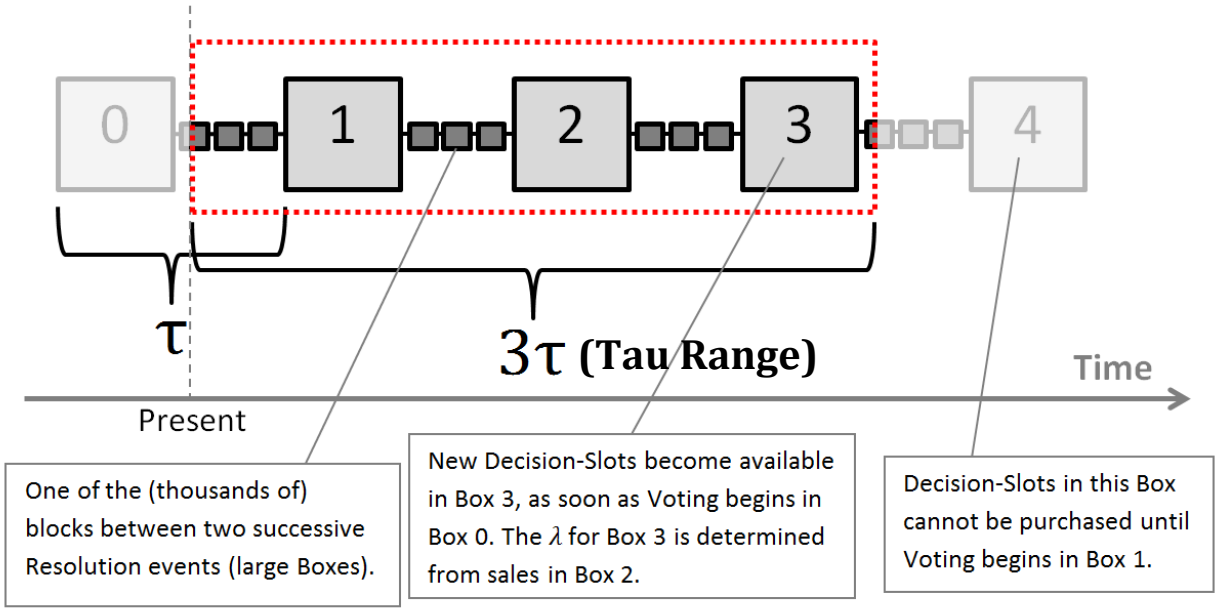


Figure 21. Listing Fees under two scenarios: No-Split (solid line, circles) and Split (dashed line, squares). The horizontal axis represents the percentage of Decision-slots which were sold: 50% would correspond to the target sales of a single Branch (or, 25% [half the target] on each of two Branches), 100% to a single full Branch (or, 50% on each of two Branches), and 200% would be two full Branches (which is impossible to achieve with only a single Branch). Resources are almost always highest with a single un-split Branch; a second Branch only maximizes revenue when expected future sales are very high.

(v) The Tau-Range

1) λ won't become known until sales figures are observed. This implies that sales can only take place a certain distance (call this the "Tau Range") from the present day (as Decision-slots far in the future will not have any basis for calculating their price). The limitations this creates are addressed in the next section (Appendix V).



(e) Appendix V – Betting on Events Beyond the Tau-Range (and Gratis Decisions)

(i) Motivation

1) We've previously established that, for fee-regulation, we may prefer to make it impossible to buy Decisions in the far future (for example, a Decision on stock prices in the year 2025).

2) Banning far-off Decision-sales also increases attack-resistance: recall that it is trading fees in the future which keep Voters in line today.

   a) Voters pay costs upfront (by purchasing pseudo-shares [VoteCoins] in the market-resolving-corporation), and periodically receive payments (half of the accumulated Trading Fees, as pseudo-dividends). Therefore, we can encourage good Voter-behavior by [1] increasing the price of the pseudo-shares (this gives Voters more to lose) and/or [2] reducing the Trading Fees collected in the event of an attack.

   b) The sooner the Trading Fees can be withdrawn, the more motivated Voters will be to *prevent* such a withdrawal. It is therefore desirable to, in general, have a short listing-turnover, to keep Voters on their toes. If Voters are well-behaved, they can *expect* future listings and future trading fees, and these expectations should be built into the current pseudo-share price. The result is highly desirable: current VTC owners have an incentive to behave the way which prospective Authors would like them to behave (in a way that maximizes trading volumes: by reporting honestly).

3) However, this implies that we will be unable to make Markets on events which take place far into the future ("beyond the Tau-Range"). Can we circumvent this limitation?

(ii) Continuous Price Feeds: Not a Problem.

1) Firstly, Decisions on values which have a continuously available price (such as a currency exchange rate [USD, Euro] or durable commodity [gold]), remain as accessible as before.

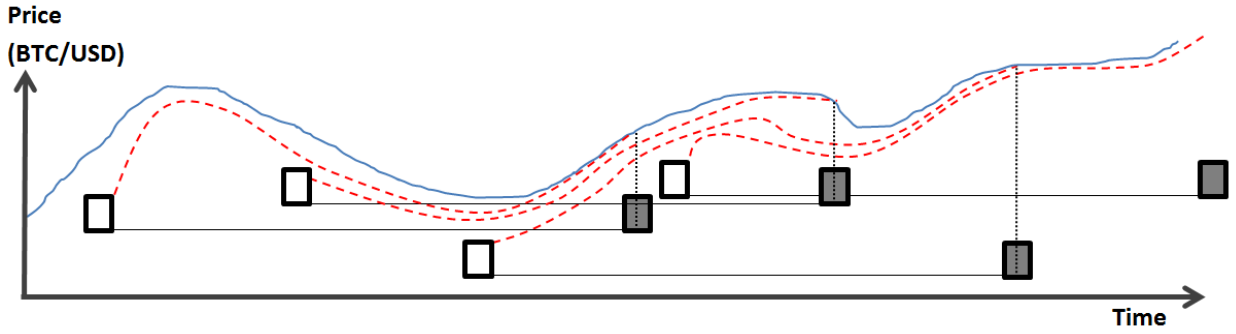


Figure 22. Decisions are created (clear rectangle) and then resolve (shaded rectangle). Even if each Decision has a limited lifespan, it is trivial to establish overlapping Decisions. Note: For a variety of reasons (chiefly, technical risk, and lack of convenience yield), I expect PM prices (dashed red line) to constantly be slightly-cheaper than their real-world target (solid blue line).

2) Software can easily monitor new Decisions/Markets for those which are identical (but for a later date) and automatically trade, or prompt the user to trade.

(iii) (Far Off) Events without a Continuous Price

1) For events lacking a reference-price, (for example, "Will Hillary Clinton win the U.S. presidential election in 2016?") the solution is to allow users to buy a "pre-Decision" today, which works as a Decision for the creation of Markets, but is only actually resolved (voted on) if the Author later buys a real Decision-slot at the appropriate future time (if the Author "continues the Decision").

   a) The transaction which "continues the Decision" can happen at any time in which the Decision is in the Tau Range, so the whole Decision-Authorship process is still mostly censorship-resistant (requires between one and six "consecutive honest blocks" to be found at least once [for every several-month period]).

   b) If Voters misbehave, Authors will not continue their Decisions. As a result, Authors' Decision fees are distributed as usual, but Voters' are unrecoverable (note that Voters never provided a service).

   c) As with VoteCoins, control over Decisions must be non-outsourceable (exclusively directed by only one private key). Otherwise Voters might buy control over Decisions, in order to keep them alive for the accumulated trading fees. Of course, if a high percentage of VTC were owned by one person, they may find this buy-up to be profitable anyway. On the other hand, if Voters misbehave, traders will likely have already stopped trading, so Authors not only resent Voters for damaging their investment, but also lack any non-bribe incentive to continue the Decisions (as Authors are bringing in the same payment, whether the Decision continues or not).

2) Making Continuances Risk-Free for Traders

   a) The goal of Truthcoin is, of course, to guarantee to traders that they will get exactly what they paid for. An obvious concern would be that, if Traders buy shares in a Market built with pre-Decisions (for an event happening in the far future), but the Author of the Market's Decisions fails to 'continue' the Decision, and it is therefore never resolved, can we still guarantee that Traders get the return they were entitled to? By combining two very powerful features of Truthcoin, the surprising answer is: yes.

   b) Puzzle Piece 1: Preeminence of the Oracle

      i) Fundamentally, the core requirement for meeting our general goal (to give traders what they paid for) is reliable external data. As 'continuances' already happen within-blockchain, the Truthcoin protocol will know –with certainty– the status of the Decision (continued or otherwise). There is no need for an oracle at all, decentralized, selfless, or otherwise.

c) Puzzle Piece 2: Market-Dimensionality:

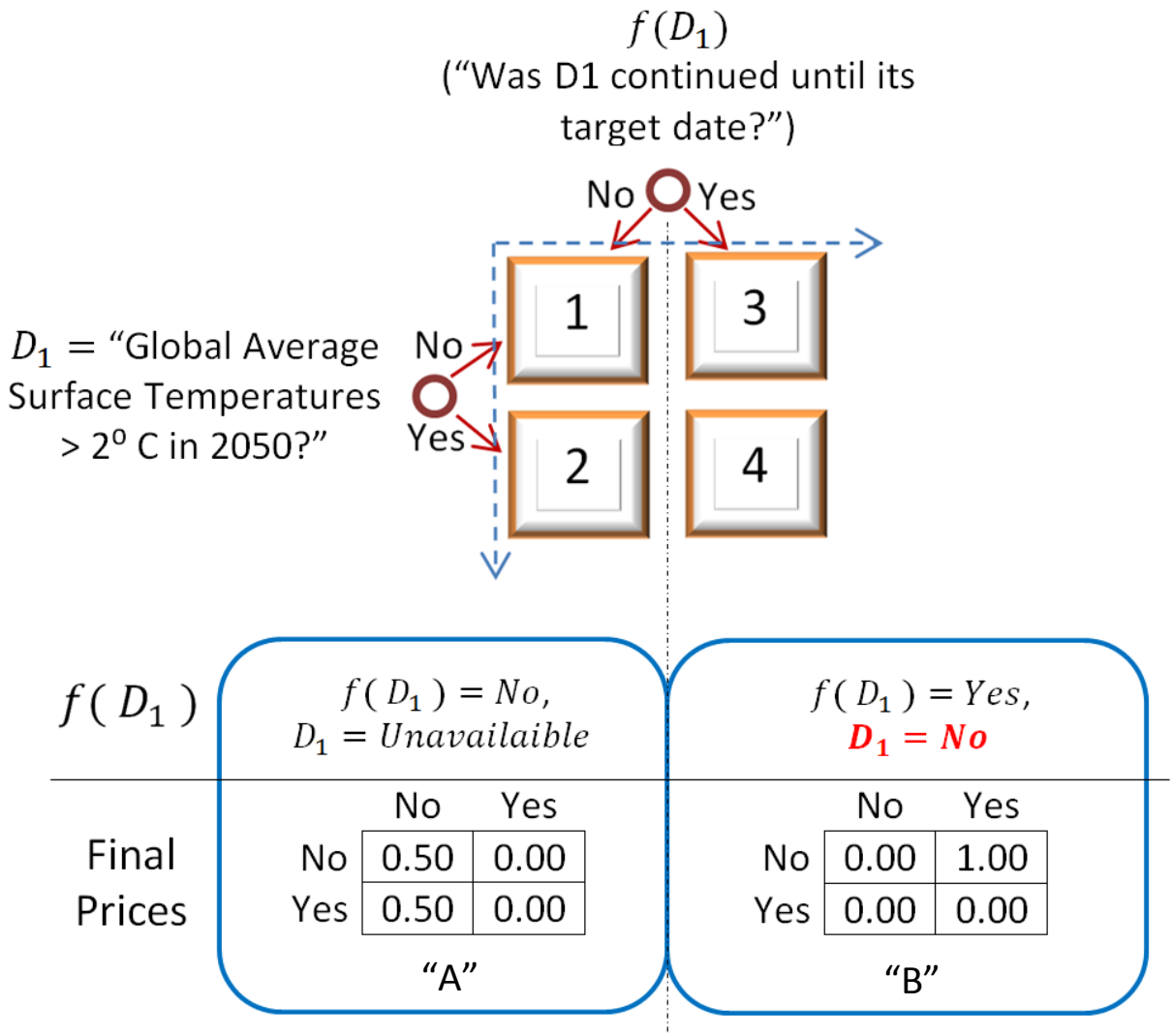


Figure 23. Before explaining further, it should be clear to the reader that a trader would purchase $\boldsymbol{s_3}$ if (s)he believed two things: firstly, that global surface temperatures would increase by fewer than 2 degrees ($\boldsymbol{D_1}$ = No, vertical axis), and secondly, that the relevant Decision to that belief would in fact be continued ($\boldsymbol{f(D_1)}$ = Yes, horizontal axis). What may not be clear, however, is how a trader should react if he or she wishes to ignore or be protected from one of those beliefs. For example, if one desires to speculate on $\boldsymbol{D_1}$ (temperature increase) yet does not know (or care to know) about the likelihood of $\boldsymbol{f(D_1)}$ (that $\boldsymbol{D_1}$, the temperature increase Decision, is continued), one would, instead of buying one share of $\boldsymbol{s_3}$, buy one share each of $\{\boldsymbol{s_1}, \boldsymbol{s_2}, \boldsymbol{s_3}\}$.

Note: the "extra cost" $(\boldsymbol{p_1} + \boldsymbol{p_2})$ is actually discounted by $\left(\frac{\boldsymbol{p_4}}{\boldsymbol{p_2}+\boldsymbol{p_4}}\right)$, the marginal cost of $\boldsymbol{s_4}$ (the single undesired state). As share prices must always sum to one, increases in $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ must decrease $\boldsymbol{p_3}$.

d) In the above figure, note that, in the left case ("A", where the Decision has failed to continue), no human-oracle reporting work needs to be done whatsoever, yet the total value of the purchased $\{s_1, s_2, s_3\}$ portfolio is 1, meaning that, regardless of the original bet, any speculator is treated as though (s)he were correct.

i) Market forces –on the actual likelihood of the Decision continuing– will force the sum $(p_1 + p_2)$ to equal the actuarially fair likelihood of the Decision continuing. Because the Decision's Author has complete control over whether the Decision continues or not, and because –(only) while Voters are reliable– he always has an incentive to continue the Decision (at a cost of a small marginal listing fee, and a benefit of a relatively complete period of trading fees), it is overwhelmingly likely that the Author himself would extract a return by buying $\{s_3,\ s_4\}$ for cost <1, and then selling the pair for 1 after continuing (ie, the Author can "sell continuance insurance").

ii) Of course, Authors would only sell continuance insurance if they knew they would choose to continue in the future, which they would only do if they expected future Voters to be honest. In this sense, it may be Voters who sell the continuance insurance, and drive the price of $\{s_3,\ s_4\}$. It is difficult to understand the factors or agents who would –in combination– drive the price of this insurance, but the free market environment in which it is sold ensures that it will always exist at the lowest possible price (…of course, the lowest possible price might be a high one).

e) Market Authors (in those rare instances when they are separate from Decision Authors) can buy continuance insurance (buy, not sell, as Market Authors cannot control the continuances, and may wish to hedge their loss of trading fees by buying $\{s_1,\ s_2\}$, without speculating on the Decision-topic at all).

f) There is never a reason to purchase either s_1 or s_2 alone: in all cases where either has any value, both will be worth ".5". These shares would always be purchased as a group.

g) Note, finally, that it is highly desirable to force the continuance to be purchased as soon as possible. This is because the last Tau-Range has the most accumulated-share-liabilities, and is likely to contain the highest volume of trading activity. It is likely that, once a Decision is in range, traders will avoid trading on it until it is continued.

(iv) Mitigating Voter-Author Collusion

1) The problem with Voter-Author collusion is that only one group is stable: recall that Voters are not able to refuse a Decision so long as the fees are paid. Anyone who wants to be an Author, can be an Author.

(v) Other Gratis Decisions

1) No marginal human labor is consumed in assessing whether or not a Decision has been continued, and so the meta-Decision $f(D_1)$ can be said to be "free", or "gratis". This "gratis" concept is not limited to continuances, and could instead subject D_1 to squaring, cubing, natural log, exponentiation, or logical operators (AND/OR) with other Decisions. This costs Truthcoin almost nothing, yet provides great benefits: the simultaneous measurement of $\{x, x^2, x^3 \ldots\}$ can sometimes be used to derive higher statistical moments (and, in turn, the standard deviation, skewness, etc.), and the natural log may be particularly useful in finance, where prices are measured in \$, but change in $\log_e(\$)$.

(f) Appendix VI – Why Add Layers (The Audit, Miner-Veto and Miner-Override)

1) The inclusion the 'Audit' possibility, in the outcome-resolution process, may seem to be an unnecessary complication.

2) The audit is justified in two ways: realism and

   a) Typically, it would not be said that "a group endorses" an opinion expressed by a slight 51% majority of group members. Instead the group would be said to "agree that the matter has not truly been resolved to anyone's satisfaction".

   b) The second justification is a small concern about unstable strategic reasoning with respect to copying another vote. Importantly, *any* additional layers beyond the Audit, such as the Miner Veto/Override, would also alleviate this concern.

Exhaustive Table of Outcomes		
Row No.	**Outcome***	**Notes**
1	<1, 1, +1>	Impossible to claim <x,:,:> and <:,x,:>.
2	<1, 1, +0>	Impossible.
3	<1, 1, –1>	Impossible.
4 # 1	<1, 0, +1>	(#1) Attacker's ideal: steal trader's money.
5 # 2	<1, 0, +0>	(#2) Not much worse than Attacker's ideal (#1).
6 # 3	<1, 0, –1>	(#3) Not much worse than (#2).
7 # 4	<0, 1, +1>	(#4) Defended against the attack, and rewarded.
8 # 5	<0, 1, +0>	(#5) Baseline: status quo prevails.**
9 # 6	<0, 1, –1>	(#6) You attacked and failed.
10	<0, 0, +1>	Impossible.
11	<0, 0, +0>	Impossible.
12 # 7	<0, 0, –1>	(#7) Someone else attacked, you failed to defend.

Figure 24. Table of possible sub-outcomes of the voting process, how these sub-outcomes contributed to one of seven outcomes, and the rank-desirability of each outcome to a voter who is experiencing it. Notice that some combinations of sub-outcome are impossible.

*Outcomes are decomposed into three sub-outcomes: first, if the Voter was able to mis-resolve outcomes and extract a large amount of money from traders, second, if the Branch VTC retained their market value, and, third, if the Voter's quantity of owned VTC changed.

**Of the acceptable outcomes, Row No. 8 (#5) is the most ideal from an accuracy perspective: 100% of the Ballots submitted were accurate.

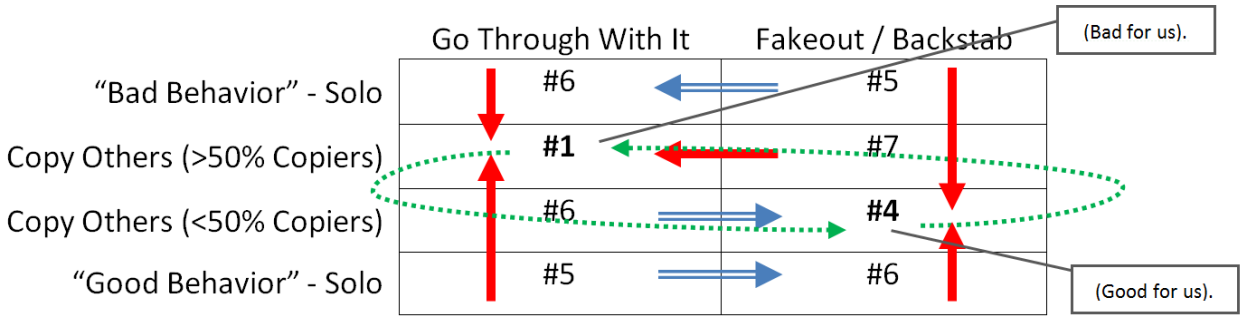


Figure 25. The copy-decision in a pseudo-normal-form game. "Move 1" would be on the row-axis, this is the 'net talk' (how a Voter claims to intends to behave). In "Move 2", on the column-axis, the Voter privately chooses his submitted action, as either one consistent with his talk in Move 1 or as an opposite action. Arrows indicate strategic incentives. Red arrows happen to also indicate an increase in copying, Blue-Double arrows indicate a decrease in copying. Starting with a prior of completely unknown copying (50% likelihood to copy for all players [including oneself, as one –by definition- has not yet decided whether to copy or not]), we cycle (green dot-dash arrows) between the two equilibria. Critically: the cycle contains one Red and one Blue-Double arrow. The status quo has no way to eliminate "talk" of copying.

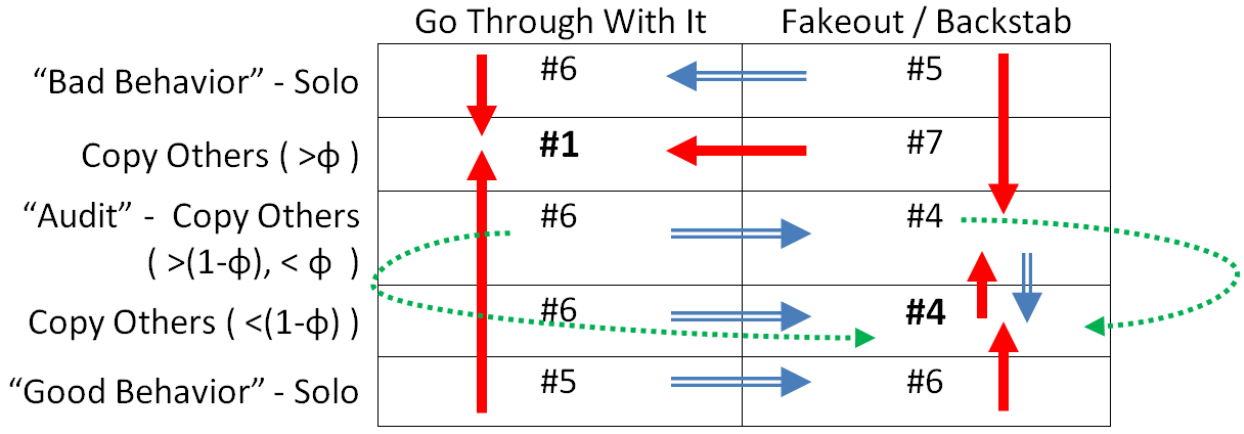


Figure 26. The same figure as before, but with the audit system added. A little friction halts the copy-inertia: The new cycle (green dashed loop) encourages individuals to lie about their claim to be copying (there is only one arrow: a Blue-Double). This de-emphasizes copying (blue arrows become stronger, red weaker), pushing the strategies toward the "Solo" rows, in this case holding them against the bolded cells. In particular, if we start with the information-less estimate of 50% copiers, the strategic logic carries us to the ("good") equilibrium.

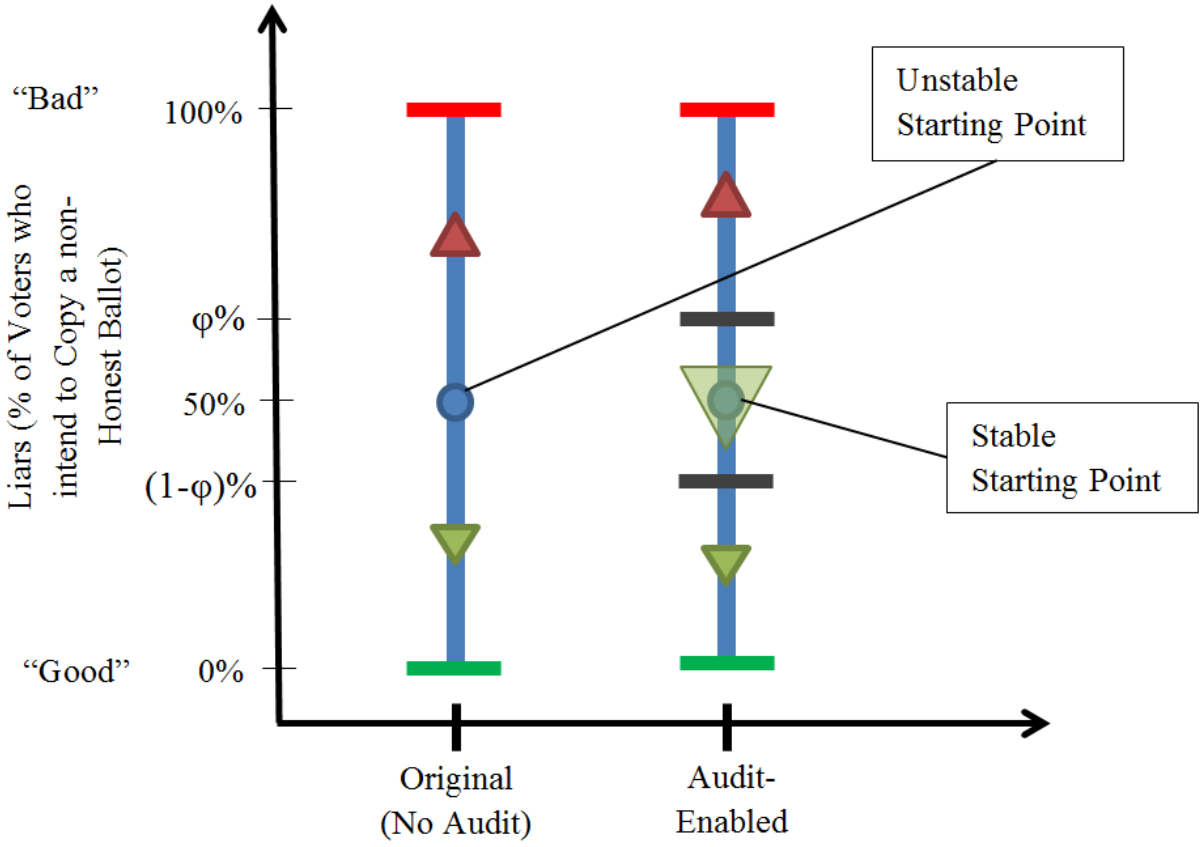


Figure 27. Phase lines of Voter-choice between the Realistic Ballot and a proposed Attack Ballot for which there is some credible talk. In both cases, a Voter starts with no information (blue circle) about the copy-beliefs of all Voters. In the left case (no Audit), momentum for the Attack Ballot (red line), might pull all voters to that critical point (instead of the more desirable critical point at 0% (green line)).

3) Arguments against the Audit:

a) The Audit is very complex, would require a number of additional lines of code, additional user-education and additional strategic complexity.

b) Recall that SVD can extract a multivariate plurality from the Vote Matrix (even when this plurality is not a majority), and so the protocol can resolve *all* Decisions successfully even if the honest Ballot receives *less* than 50% of the vote (potentially much less). The Audit would prevent us from taking advantage of this helpful property. This is because the Audit is designed primarily to protect against the actions of a well-coordinated ("single") attacker, and the properties of SVD are most advantageous when attackers are not coordinated.

c) If Miner involvement-options (Vetoes and Overrides, see next sections) are inevitable, and will also serve similar purposes (resistance to large-attackers, resistance to credible copying-talk), then the marginal benefit of the Audit (even if it always worked as intended and did not introduce technical complexity) might be low.

(g) Appendix VII –Sidechains and Miner-Vetoes

(i) The miner-veto may seem like another "overly-complex", unnecessary addition to Truthcoin. In reality it is likely to become a staple feature of all sidechains.

(ii) Review of Mining

1) Satoshi's concept of "mining" seems secure. It has a 6 year track record of success, in the real world under a variety of stressful conditions.

2) Miners can only steal if they coordinate a 51% attack. "Steal" in this case means "a long blockchain reorganization" (rewriting old transaction history).

3) Miner-cartels are discouraged through [1] wasted work ([on a not-longest chain] if the cartel fails to coordinate) and [2] a collapse in the value of the miner-payout-asset (if the cartel does coordinate).[42]

(iii) Sidechains as "Modular Hard-Forks"

1) For new Bitcoin-features, is there a difference between [1] the hard-fork case ("Original Blockchain" + "New Features") and [2] the sidechain case ("Original Blockchain" + "Sidechain with New Features")?

   a) Imagine a blockchain where only some of the token-features can be 51% attacked.

   b) Secondly, imagine a "valuable feature", which, if operating, increases the market value of the blockchain token (by making the token more useful).

   c) Let's apply the logic established above to any Bitcoin-sidechain with "valuable features". Sidechain-miners would not attempt to steal the accumulated funds on their sidechain, because this would result in [1] wasted work and [2] a reduction in the value of the mined-token.

   d) Therefore, no valuable feature will be attacked. Note that Bitcoin (which offers new token-features, which compete favorably with the tokens of PayPal or SWIFT) has not yet been attacked in this way.

2) Non-valuable features will (and should be) attacked, because such an attack would result neither in wasted work (the attack succeeds, so no competing chains) nor a collapse in value (the feature wasn't valuable).

(iv) Veto as "Modular Mini-Sidechains"

1) For a protocol, systemic risk is disastrous; it is better to "fail safe". For example, consider Bitcoin's privacy and security model: when "Bitcoin is hacked", the victims are individuals, not the protocol as a whole.

[42] Sound familiar? This is also how Truthcoin's SVD works.

a) Local failures can create global success, This is the so-called fractal anti-fragility, which powers evolutionary biology.

b) Failures of today make future failures less likely: [1] wallet bugs today, lead to an increased *future* emphasis on wallet security, [2] exchange failures today, lead to an increased *future* skepticism of exchange-reliability, etc.

(v) Sidechain-Theft implies Vetoes are Harmless (and are therefore Net Beneficial)

1) Sidechains imply the possibility that Miners may collude and steal from users. While distressing, this possibility has an interesting implication: It is impossible to believe that Miners will exploit the Miner-Veto.

   a) To expect and exploit of the Miner-Veto, one would need to hold the belief that "(enough) miners are conspiring against me". However, if one held that belief, the true threat would be outright 51%-sidechain-theft, not some inconvenient little Veto.

   b) So, the sidechain-threat is always more threatening than a misuse-of-veto threat.

2) The Veto is helpful for the network as a whole.

   a) Anything is better than a blockchain-reorganization.

   b) Mining is the only global activity that [1] is Sybil-proof, and [2] has users who are guaranteed to have some interest in the conflict at hand.

   c) Miners ultimately decide everything anyway, so the protocol might as well allow the expression of that power to have the appropriate degree of influence.

(h) Appendix VIII – The Voter Override (and "Miners as Voters")

1) How can we strengthen the weakest parts of the protocol? First, let us identify the weaknesses:

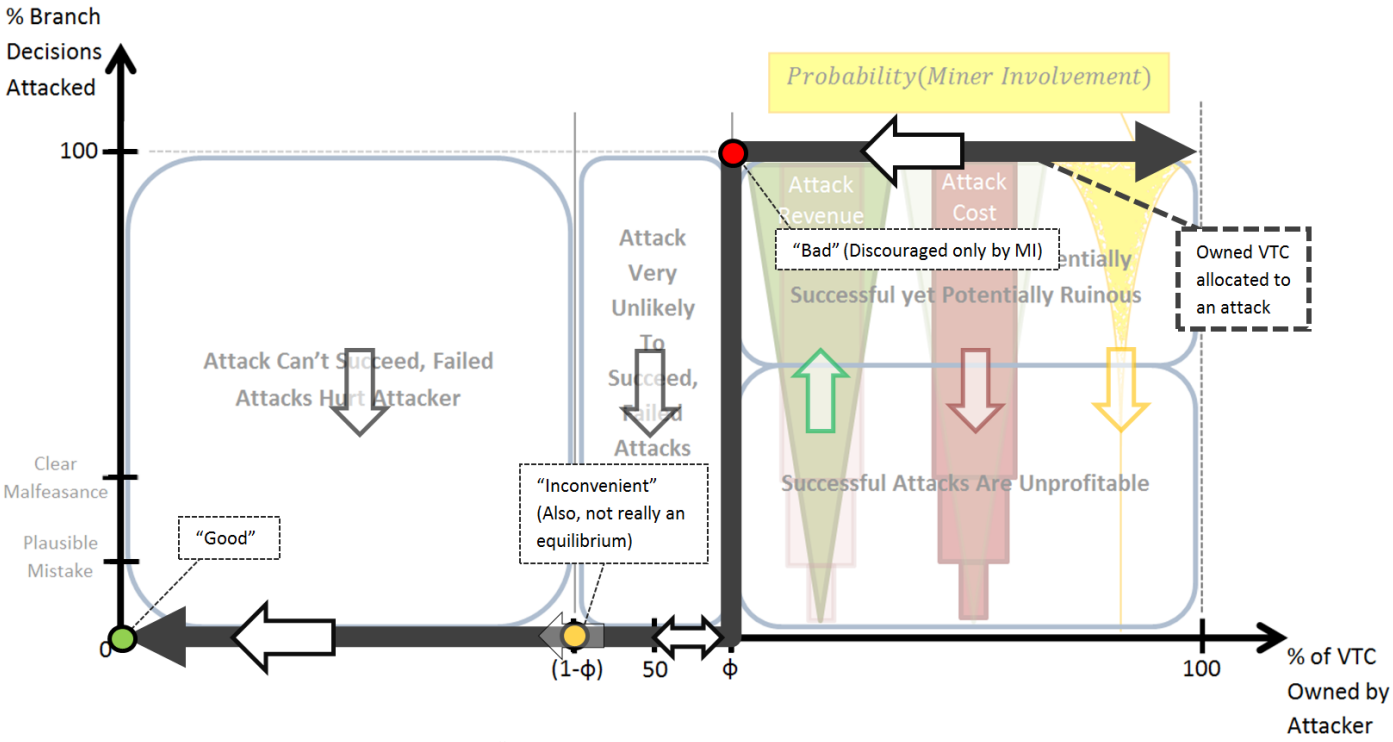


Figure 28. An earlier figure (see "Voting Strategy"), with new annotations. Strategic pressures (arrows) and equilibria (colored points) are given for a hypothetical individual who has the ability to buy or sell any percentage of the VoteCoins at any time (double-arrow). The central (orange) equilibrium is not particularly troublesome (and would only exists if Audits are used), and the left (green) equilibrium is very healthy. The right equilibrium (red) poses the relevant problem to us.

a) Fundamentally, the outcome-resolution process is at risk as a *direct* result of an attacker's ability to accumulate an "influential quantity" of VTC. In the simpler, no-Audit case, this "influential quantity" would be 51% for Scaled Decisions, and $\left(50 - \frac{Catch}{2}\right)$% for Binary Decisions.

b) This vulnerability bears an overwhelming resemblance to the 51% attack in Bitcoin mining. If we are already assuming that a malicious attacker is unable to control 51% of the hashing power, then by allowing Miners to vote we can eliminate the last of the protocol's weaknesses. In essence, the entire horizontal dimension of the graph above has been flattened.

2) Implementation Details

a) Votes are submitted precisely the same way as before, as special messages which contain Ballots, public keys (which receives the CashCoin trading-fee dividend payments instead of VoteCoins), (and, optionally, information about Splitting). In this

case, there were would be only one Vote per block (or per N blocks). Votes are cast over a long timeframe (Voting Cycles would likely have a longer vote-submission "period 2") and then unsealed ("period 3") over a subsequent 1000-block period, so as to attempt to remain comparably censorship-resistant.

3) To clearly explain this layer, let us examine the implication of a system which completely replaced all VoteCoins with Miners-as-Voters for all outcome-resolution:

	Individuals As Voters (VoteCoins)	**Miners as Voters**
Role Stability	Assumes that Miners generally "don't care" about anything except their narrow function of plugging in hardware and running it. Directly implementable without any Role changes, other than an additional merge-mined chain.	Assumes that Miners do care actively about the network Miners must now "change/expand their role". Miners can now attack the network in new ways.
Scalability	Highly Scalable with Branching	Not very Scalable, although some scalability could be achieved with multiple sidechains.
Complexity	More complexity (several interacting agent-types)	More instability … agents affect the blockchain itself at lower levels.
Centralization	No marginal impact on miner-centralization.	Marginal pressure for more mining-centralization.
Reputation-Rents / Security	Extra value created by trading fees can be owned and enjoyed by users, who have an incentive to act entrepreneurially (maximize this extra value).	Extra value will initially drive up Mining returns, but eventually *all* extra value will be competed away ("destroyed") under perfectly competitive, homogenous Mining. As a result of this, the security of the Bitcoin network would increase.
Temporal Depth	Reputation can slowly accumulate (can continue to accumulate more and more VTC over time).	Reputation can only accumulate instantly, or near-instantly (across one Voting Cycle). Overlap between trading-fee-rewards and coinbase-rewards makes accumulation-incentive (of mining equipment, not coins) less straightforward.

1) Discussion

a) This concept is theoretically stronger (fewer weaknesses), but much more difficult to roll out in practice (requires role changes). Moreover, it provides Miners with an incentive to form large and larger cartels. As it interferes with Miner incentives directly, and is complex, there is some chance that this modification would destroy Truthcoin, and possibly Bitcoin with it (or, at least, the ability to Bitcoin-sidechain).

b) There are no Votecoins, or concept or durably-"owned" reputation, and therefore no RBCR; instead, SVD-reweights determine (one time only) how Miners split this period's accumulated Trading Fees with each other. The market value of all mining equipment would logically increase by (what would have been, in expectation) the market capitalization of all VoteCoins.

c) As all Miners must vote on everything, this concept does not scale very far. Possibly, an implementation of this with a new sidechain per Branch might scale slightly better, but this dramatically complicates the reliability of the PM-service. On top of that, it may be prohibitively difficult to form Markets with Decisions from multiple "Branches" (multiple sidechains), which would be very disappointing.

d) The Miners-as-Voters concept of "reputation" feels to be slightly more appropriate. As all reputation is transient, lasting only a single Voting Cycle before the network "forgets", this model makes it impossible for anyone to be "very reputable". Voters count more or less equally. This implies more decentralization, at the cost of reduced overall "wisdom" (the ability to count on an older, wiser person to resolve a complicated question), yet, because all Decisions should be "easy" to resolve, there should be no need for great "wisdom" at all.

2) Hybrid Model

a) It is perhaps most ideal of all to invoke Miners-As-Voters as an ultimate last resort (high cost, high accuracy, see Figure 17 in Appendix III).

A (Convenient) Coincidence of Wants		
	Want	**Have**
Voters	…to know that they won't be screwed by a large rival Voter (which would result in the destruction of their VTC-investment).	…the ability, and the desire, to arbitrate all of the mature Decisions on the network.
Miners	…to only compute hashes in peace, without constantly needing to step in and micromanage the network.	…the ability to step in and micromanage the network.

(i) Appendix IX – Justification of Chosen Parameter Values

Each Branch is defined by the following parameters. Although separate Branches might compete over different parameter-families, it may be advantageous for the blockchain itself to impose "Reasonable Bounds" on possible choices for parameters. Branches themselves may impose "Reasonable Bounds" on Market-specific parameters, (*b*, content-tags, trading/audit fees).

Parameter	Parameter Represents…	Reasonable Bounds	Reasoning Favoring Low	Reasoning: Favoring High	Reasoning Behind Choice
"Retention" **$\alpha = .80$**	1] Forgiveness of RBCR to Voter-disagreement. 2] Neuroticism in assuming new ownership. 3] Penalty for least-coordinated Voter (loses (1- α) of VoteCoins).	(0,1) Zero and one would remove all long-term reasoning.	1] Want network to adapt quickly. 2] Want attackers (mis-voters) to suffer.	1] VoteCoins should more safely store-value. 2] Individuals may make mistakes, past history should count most.	Past history should count the most, but in general VoteCoin owners are responsible for proper voting. 20% for being the unanimity-failure seems not unreasonable.
"Intervote Period" **$\tau = \{\tau_{idle}=6w, \tau_{voting}=1w, \tau_{unsealing}=1w, \tau_{review}=1w, \tau_{veto}=1w\}$** $\sum \tau$ = 10 weeks	1] Pulse for network to "check in with reality". 2] Scale-economies of Voter-Time (setup costs/total costs). 3] Loss of info-salience over time ("memorability of events").	$T_{unsealing}$ involves revealing private keys, implying ~1000 blocks. Others depend on reliance-on-human-input.	1] Want to decrease basis risk for Traders. 2] Believe info decays too rapidly to remain available at low-search cost.	1] It is most important to contain many Decisions in a Vote Matrix to make attacks less practical. 2] Believe in minimizing setup cost.	Attacks must be avoided at all costs, but basis risk is also an important factor. With human involvement on the Main branch, a period of 8 weeks seems a helpful balance.
"Audit Accumulation Period" **Ω = 6 months**	1] Time between audits. 2] Minimum time one would have to prepare their Audit-Ballot.	[1 month, 3 years]	1] Believe audit will be easy (info has diffused).	1] Want to give Miners time to set veto. 2] Want 'punitively slow' audit (should never have reached	The Miner-Veto, which should be easy to set and sufficiently infrequent to be nonburdensome. The Audit should be painfully slow.

				this point). 3] Believe audit should span many different Intervote Periods.	
"Certainty/ Audit Threshold(s)" **Φ = {.65}**	1] Insistence on certainty, "supermajority", minimum proportion of Voters to declare something "undisputed".	{ 0, (.5, .9) } Attacker with a majority can ignore audit. Do not want audit-triggering spam.	1] Want to reduce strategic complexity. 2] Want failed attackers to be punished immediately.	1] Want attacker-individuals to need to buy more VoteCoins. 2] Want to rely more on the Threat of Audit.	2/3rds is a standard democratic threshold.
Tau Range **ρ = 3**	1] How far into the future the system looks. 2] Average length of time between present and prediction market event-dates.	(1, 8) A high value (>8) largely disables the feature (partially depending on Tau).	1] Want Fee1 to adjust very quickly (worried about BTC-demand volatility). 2] Want greater ability to punish Voters for misbehaving.	1] Want Authors to be able to buy standard decision-slots many months or years into the future. 2] Worried about the practicality or user-friendliness of 'continuance insurance'.	My opinion is that (certainly at first), [1] the price of Bitcoin will be extremely volatile, and [2] few users will be interested in using Truthcoin for bets on far-off events. I also expect continuance insurance to cost very little, or at least to have a reasonable cost.

(j) Appendix X – Data Structures and Messages *(Author's Note: As software v0.1 has not yet been completed, this section has been neglected. It should not be treated as authoritative until after v0.1 is finished).*

1) What follows is descriptive list of the messages which might be relayed across the Truthcoin network, as well as the data structures which would be created as a result.

(i) Blocks

1) High level overview of the most global data structure: the blocks.

Global Parameters – Not Specific to a single Branch	
Field	**Description**
Magic no	value always 0xD9B4BEF9*
ParamMutation	limits on how strongly a new Branch's parameters can differ from its parent Branch (ListingFee = c(0.8, 1.2), MaxDecisions= c(0.8, 1.2), …, ConsensusThreshold= c(0.8, 1.2))
…	…

Block Header			
Field	**Description**	**Updated When**	**Size (Bytes)**
Version	Block version number	You upgrade the software and it specifies a new version	4
hashPrevBlock	256-bit hash of the previous block header	A new block comes in	32
hashMerkleRoot	256-bit hash based on all of the transactions in the block	A transaction is accepted	32
Time	Current timestamp as seconds since 1970-01-01T00:00 UTC	Every few seconds	4
Bits	Current target in compact format	The difficulty is adjusted	4
Nonce	32-bit number (starts at 0)	A hash is tried (increments)	4

Block		
Field	**Description**	**Size (Bytes)**
Magic no	value always 0xD9B4BEF9*	4
Blocksize	number of bytes following up to end of block	4
Blockheader	consists of 6 items (see above)	80
Message Counter	positive integer VI = VarInt	1 – 9 *
Branch Messages	messages that add, remove, or edit … Branches	<variable>
Decision Messages	… Decisions	<variable>
Market Messages	… Markets	<variable>
Trade Messages	Buy, Sell, or Redeem operations on Market-Shares	<variable>
Tx Messages	Messages that transfer ownership of shares or	<variable>

	funds	
Branch Vetoes	counter on Resolution objects, if sufficiently high, the vote must be repeated	2 per Branch
Branch Overrides	see Appendix VIII "miners as voters", alternatively counter on a specific Ballot submitted by someone (if counter is high, and/or other Miners don't disagree, this Ballot wins).	2 per Branch

(ii) Messages

1) Broadcast peer to peer, and stored permanently in the blockchain.

Add Decision			
Field	**Public**	**Description**	**Size (Bytes)**
ID	Yes	Hash of all data fields (except this one)	32
PrivHash	Yes	Hash of all non-Public data fields	32
Branch	Yes	ID of this Decision's Branch	32
OwnerAd	Yes	the hash of the public key of the creator ("author") of this Decision	32
TauFromNow	Yes	the earliest Intervote Period by which the <Prompt> information will have become widely available	1
Standard	Yes	standard (mandatory to answer) or overflow (answering is optional)	1
Scaled	Yes	FALSE corresponds to a binary Decision, TRUE to a scaled Decision in range(<Min>,<Max>)	1
Prompt	No	human readable question for Voters to answer, with instructions on units and backup sources	1000
Min	No	only applies to scaled Decisions	2
Max	No	only applies to scaled Decisions	2

Create Market			
Field	**Public**	**Description**	**Size (Bytes)**
ID	Yes	Hash of all data fields (except this one)	32
PrivHash	Yes	Hash of all non-Public data fields	32
B	Yes	the liquidity parameter 'beta' (in LS case, the 'initial beta')	4
TradingFee	Yes	percentage which traders are overcharged	4
MaxCommission	Yes	the maximum possible price spread in LS market maker, determines alpha and initial liquidity	2
OwnerAd	Yes	hash of the public key of the creator ("author") of this Market	32
TX-PoW-h(x)	Yes	chosen hash functions for tx-proof-of-work	4
TX-PoW-Difficulty	Yes	proof of work difficulty, to discourage front-running (will have intelligent defaults)	4

DAC	Yes	is this for funding public goods (selling disabled)?	1
D_Dimens	Yes	length and number of state dimensions	4
Title	No	human-readable title, not required to be unique	32
Description	No	for finding the market in search	5000
Tags	No	for finding, and organizing the market in search	500
D_State	No	list of Decisions, defining the axes and dimensionality of the Market	32 per Decision
D_Functions	No	list of functions applied to these Decisions (by default, would be the identity function)	1 per Decision

- New User (Bitcoin deposited from Sidechain)
- Transfer (Cashcoin to Votecoin, one private key each)

Buy / Sell / Redeem		
Field	**Description**	**Size (Bytes)**
uID	address controlling unspent outputs	20
ID	Market id	32
State	of the Market's shares ("Yes", "No"), which to buy	2
P	price target ("buy until price reaches P"), either P or S can be provided	2
S	share target ("buy S shares"), either P or S can be provided	2
PriceLimit	trade is only valid if Price(<ID>, <State>) < <PriceLimit>	2
SequenceLimit	trade is only valid if there have been <SequenceLimit> or fewer trades on this Market since the last block	2
Nonce	for the tx-PoW	6

Continue Decision		
Field	**Description**	**Size (Bytes)**
ID	ID of Decision to be continued (all else the same, incl. OwnerID)	32

Submit Vote / Submit Steal		
Field	**Description**	**Size (Bytes)**
OwnerAd	hash of Votecoin public key	20
BallotHash	hash of (Ballot, NewPublicKey)	32

Reveal Vote		
Field	**Description**	**Size (Bytes)**
OwnerAd	hash of Votecoin public key	20
Ballot	each Vote for each outcome	<variable>
SplitParams	should branch prepare to split into two Branches? if so, how should parameters change (ListingFee = 1, MaxDecisions=1, …, ConsensusThreshold=1)	?
NewAd	the address (hash of public key) to which the new post-	20

	resolution Votecoins are assigned	

Reveal Steal		
Field	**Description**	**Size (Bytes)**
VictimHash	hash of the submitted Vote, for which one plans to steal	32
NewAd	the <NewAd> of the Voter who carelessly revealed the contents of his/her Vote-hash prematurely	20
ThiefAd	the address (hash of public key) to which some stolen Votecoins are assigned post-resolution	20

- The SVD-Resolution operation, performed once per Intervote Period ("Tau"), would produce a lot of unspent outputs (from the trading fees: some claimable by the Voters, others claimable by Authors).

Close Market		
Field	**Description**	**Size (Bytes)**
Market ID	Unique ID (from this, one can look up Market's PrivHash)	32
PrivData	The hash of this data must match PrivHash (this data includes Decisions and state-dimensionality)	<variable>
Final Price	Final Prices for the Market (this data must match the Outcomes of each Decision as determined by SVD-resolution).	20

(iii) Datasets

1) Constructed internally by software applications as they process blocks. May contain redundancy to accommodate human-readability.

Branch Set		
Field	**Description**	**Size (Bytes)***
ID	unique hash of the Branch data (below)	32
Name	unique human-readable title of the Branch	32
Exodus	information for identifying Branch Votecoins	32
Description	lengthy human-readable description of the guidelines for acceptable Decisions on this Branch	1000
Base Listing Fee	the target marginal cost to list a single Decision	2
Max Decisions	the highest quantity of Decisions that each Branch owner would commit to voting on in each Intervote Period, twice the target quantity of Decisions	2
Minimum Trading Fee	mainly to prevent free Decisions	2
Listing Fee Balance	total listing fees paid on mature Decisions from this Branch (may be one of these per Tau-Range)	6
Trade Balance	total fees paid on Markets made with mature Decisions from this Branch	6

Intervote Period ("Tau")	the time (blocks) between required ballot-submissions	2
Ballot Time	time (blocks) between 'start of voting' and 'voting deadline', vote-hashes are submitted during this time	2
Unseal Time	time between 'voting deadline' and 'unseal deadline', vote-content is revealed during this time	2
Tau Range	the maximum amount of time from present, measured in Intervote Periods, in which Decision-slots are available for purchase	1
Catch	for Binary Decision, width of the "bin" that corresponds to ".5"…a Catch=0.1 would map the values {.46, .47, .53} to ".50" but {.20, .43, .44} to "0" and {.56, .57, .93} to "1"	2
Consensus Threshold ("Phi")	proportion of total votes which would constitute a supermajority	2
Split Threshold	sigma=5000 how much split-signal must accumulate before branch splits	6
Current Split	how far we have progressed toward the split threshold	6
Split Memory	=25000 blocks, length of time votes to split are "remembered"	6
ParamRescalers	the current weighted median of the rescalers for each rescalable parameter, these values determine the parameters for a newly split Branch	32?
FeePerKb	fee for consistent pricing in CreateBranch and CreateDecision messages	2
MaxStates	Markets with more than 256 states are too computationally burdensome to be allowed anywhere.	2
Starvation	how long (measured in Intervote Periods) a Branch remains in the system after Authors have stopped buying Decision-slots on it.	2

Decision Set		
Field	**Description**	**Size (Bytes)***
ID	hash of all fields, excluding <ID>, <State>, <ResolvedOutcome>, and <Size>	32
State	status (trading, resolved, auditing), possibly redundant with <ResolvedOutcome>	1
ResolvedOutcome	the post-resolution result	1, or 4
Size	size (in bytes) of all fields, excluding <ID>, <State>, <ResolvedOutcome>, and <Size>	2
Branch	id of this Decision's Branch	32
Prompt	human readable question for Voters to answer, with instructions on units and backup sources	1000
OwnerAd	the hash of the public key of the creator ("author") of this Decision	32
TauFromNow	the earliest Intervote Period by which the <Prompt> information will have become widely available	1
Pre	TRUE if the Decision needs to be purchased	1

	again, later (when in Tau-Range) to actually be voted on	
Scaled	FALSE corresponds to a binary Decision, TRUE to a scaled Decision in range(<Min>,<Max>)	1
Min	only applies to scaled Decisions	2
Max	only applies to scaled Decisions	2
Standard	standard (mandatory to answer) or overflow (answering is optional)	1

Market Set		
Field	**Description**	**Size (Bytes)***
ID	hash of permanent features (all fields, excluding 1 through 7)	32
Size	size in bytes of permanent features (all fields, excluding 1 through 7)	2
Shares	array of outstanding shares of each state	4 per state
Balance	the cash accumulated in this market, depleted when traders sell or redeem shares	8
FeeBalance	total trading fees collected	8
State	is this market 'tradeable' or 'redeemable'	1
B	the liquidity parameter 'beta' (in LS case, the initial beta)	4
TradingFee	percentage which traders are overcharged	4
MaxCommission	the maximum possible price spread in LS market maker, determines alpha and initial liquidity	2
OwnerAd	hash of the public key of the creator ("author") of this Market	32
Title	human-readable title, not required to be unique	32
Description	for finding the market in search	5000
Tags	for finding, and organizing the market in search	500
MaturationTime	the point in time at which the resolved outcome-axis would be known	2
DAC	is Market part of a Dominant Assurance Contract?	1
D_State	list of Decisions, defining the axes and dimensionality of the Market	32 per Decision
D_Functions	list of functions applied to these Decisions (by default, would be the identity function)	1 per Decision
TX-PoW-h(x)	chosen hash functions for tx-PoW	32
TX-PoW-Difficulty	proof of work difficulty, to discourage front-running	4

Article VI. Document History

(a) Version 1.0 – January 31, 2014

(b) Version 1.1 – March 28, 2014

1) Substantially edited Article IV "Implementation Details" based on feedback from expert cryptographers, senior bitcointalk.org members, and developers.

2) Added Appendices describing the handling of Missing Values and Partial Incoherence (which were always part of the original design, I had simply forgotten to write about them in version 1).

3) Fixed several typos.

4) Changed wording on LMSR from "infinite" to "permanently nonzero". The previous wording was incorrect (I don't know what I was thinking).

5) De-emphasized demurrage as it is unnecessary and confusing.

(c) Version 1.2 – May 21, 2014

1) Added and documented functionality for Scaled Decisions, which take on a Scalar Outcome (not a Boolean).

2) Edited substantially for clarity, removing a few paragraphs which were outdated or otherwise confusing. Caught numerous typos and formatting errors.

(d) Version 1.3 – August 25, 2014

1) Reworked paper to present the idea not as a Bitcoin addon, but instead as a new blockchain and Bitcoin replacement. This included a change in terminology: Bitcoins became "CashCoins" and Truthcoins became "VoteCoins".

2) Improved the assumptions section, by removing implicit and redundant text, and adding a few (previously overlooked) assumptions.

3) Added appendix section "Justification of Chosen Parameter Values".

4) Added the concept of a Transfer (moving shares as one would move Bitcoins).

5) Added the Audit Process, which makes the Outcome-Resolution process more realistic (more time to resolve disagreements), and strengthens incentives to realistically-coordinate by adding a Wealth-layer (audit) and Miner-Layer.

6) Changed the way Binary Outcomes are resolved as "unresolvable", as the previous way has been superseded by the Audit Process.

7) Added protection against Dying Branches (they now "stall" instead, see $\Lambda = 200$).

8) Mentioned tx-PoW requirement to prevent front-running of trades.

9) Edited Implementation Details substantially for clarity and updated-relevance.

10) Edited for clarity generally, and added a few helpful graphics.

(e) Version 1.4 – April 29, 2015

1) There was an error in the description of RBCR. The updated description notes that the first column of the U matrix is transformed, as in the statistical technique of PCA, into the first score and loading. It is the first Score which becomes the new reputation vector.

2) Changed terminology: "Voting Period" implied that what-was-being-discussed was "the time it takes to vote". Instead, it has been changed to "Intervote Period" (time between two successive votes), and each individual period is referred to as an “Intervote Cycle” or “Vote Cycle”.

3) Emphasized that Tau-'Voting' and Tau-'Unsealing' really do not vary anywhere near as much as Tau-'Idle'. In fact, Voting/Unsealing need to be more or less fixed at 1000 or so blocks.

4) Added the Hash-Publish method of voting, as described on the forum.

5) Emphasize that Auditors fight over Pooled Trading Fees: Half to the Auditors, Half to the Ballot which agreed with the Auditors (this is what encourages minority voters to 'stick it out'), and erased the 'Audit Fee parameter' as it was a terrible way of explaining this.

6) Update Figure of the Attacker-flowchart, to include two additional pages.

7) Explained the Miner-Veto Concept better
    1. Princeton idea (Miners do everything) isn't great - consensus failure, laziness, instability.
    2. A rare, 'fail-safe', would be better (more like mining an empty block)
        1. Easily implemented Nonce + Vetos have no signature.
        2. Explain that it will probably never be used (and this is a good thing).

8) Added Appendix where Listing Fee (Fee1) is set algorithmically/autonomously (without relying on user input).

9) Added Appendix sections on “far off” Decisions – why they exist and why they won’t be a problem.

10) Explained that it is extremely easy to have Markets not just reference Decisions, but also LOG, LN, ()^2, ()^3, etc of Decisions. This includes NotSequenced() of Decision. These are now called 'Gratis Decisions'.

11) Added Appendixes to provide greater detail on the reasoning and motivation behind “extra truth layers”.

12) Updated the Vote-Outcome plot using the new PlotJ().

13) Move figures to the body of the document – to help poor readers fight through an 80+ page document.

14) Added the concept of "non-outsourceable" VTC buying/selling (one VTC private key per transfer), to prevent an obscure but critical assurance-contract attack (which would place attacker a world where he only had to purchase when his attack would succeed).

15) Improved Appendix II, by highlight the (1-p) case, and by adding an example that isn't 2 by 2, to highlight the (p + (1-p)) / 2 aggregation in the code.

16) Greatly simplified "statuses" of Decisions/Markets.

17) Removed "planting"/private PMs from the paper and the project, such ideas are now considered out of scope.

18) Modified the PCA directional index to use ranks, as I found this to work slightly better.

19) Added Data Structures/Messages Appendix (which clarifies, among other things, how Decisions/Markets are kept private and un-censorable).

20) Corrected numerous typos and unclear sections.