Flux: an Open Market Protocol and Data-Driven Decentralized Oracle

Version 1.0

Jasper de Gooijer
@jaspergooijer

Wesley Graham
@wesatbab

Peter Mitchell
@realpetermitch

April 20, 2020

Contents

Glossary

- **APR** - annual percentage rate
- **Bond** - a predefined amount of Flux Token
- **cDAI** - an ERC20 token that represents a lending or supply balance and the interest it has accrued on Compound Protocol
- **DAI** - a decentralized cryptocurrency stabilized against the value of the US dollar
- **Dispute** - a rebuke of the initially proposed market resolution with financial weight attached
- **Dispute Bond** - an amount of Flux Token a Validator must stake to dispute a initially proposed market resolution
- **Dispute Round** - a process in which a Validator stakes an amount of Flux Token double the amount staked by previous Validators in favour of an outcome.
- **EPOCH** - a period of 24 hours on the NEAR blockchain
- **Flux Token** - native currency designed to secure the Flux Protocol
- **Maker CDP** - collateralized debt position on maker protocol
- **Market Creator Fee** - a predefined trading fee between 0-5%, a market creator may designate to collect a fee on volume traded in their market
- **Market Validity Bond** - a refundable bond of Flux Token a market creator pays to cover resolution cost if the market is deemed invalid
- **Slashing** - subtracting (a part of) a malicious actor's staked funds
- **Smart Contract** - a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract expressed in code
- **Stablecoin** - a crypto currency pegged to the price of a real world asset of currency e.g. the USD
- **Universe Fork** - a worst-case process in which the Flux Protocol migrates to one child state to rectify a market dispute, chosen by a majority of Flux Token holders.
- **USDC -** a 1:1 representation of a US dollar on the Ethereum blockchain.
- **Validator** - a token holder that provides resolution data for markets
- **Validator Stake** - a base amount of Flux Token a token holder must stake to run a active validator

Abstract

Prediction markets are exchange-traded markets that are created to trade on the outcome of, and gather real-time insights on events or assets. One approach towards developing such a service is to allow anyone to generate markets ("open market creation"), as well as to resolve the outcome of markets without the use of a potentially-corruptible central authority ("decentralized market resolution").

Such is the goal of Flux: to build a scalable and decentralized prediction market protocol.

To build such a service, however, two core challenges must be resolved. The first challenge is "I/O" - the supply and aggregation of input and output data entered into a prediction market. This I/O challenge is predominantly associated with data resolving a market's outcomes. The second challenge is that of "economic incentive" - designing a protocol that is profitable for honest actors to use while economically discouraging actions that damage the health of the protocol.

Our key offerings to resolve these challenges are a "lottery-based" resolution mechanism based on Random Number Generation (RNG), as well as the use of market validity bonds and dispute bonds. These foundations couple with a "worst-case" forking mechanism based on the work of the Forecast Foundation[1].

Layered on top of these offerings is "Flux Token (FT)," a NEAR protocol equivalent of the ERC-20 standard fungible token used to secure the protocol.

In this paper, we describe the Flux Resolution properties, and apply these properties to examples of valid, invalid, and disputed markets. We show that the resulting mechanisms are scalable, cost-effective, and Sybil-resistant.

[1] https://www.augur.net/whitepaper.pdf

1. How Flux Resolution Works

The Flux Resolution protocol broken down into five key stages:

1. Creation
2. Trading
3. Resolution
4. Dispute
5. Finality

In the creation stage, any actor (applications, clients, individuals, et cetera.) can create a market, posting a small bond to ensure market validity. In the trading stage, "traders" buy and sell shares of the market's outcomes. In the resolution stage, "validators" are tasked with identifying the outcome of the market. In the dispute stage, any individual that suspects foul play can post a bond to challenge a market's outcome. In the finality stage, rewards are paid out to successful traders, market creators, and honest validators/disputers, while the stake of malicious actors is slashed.

The operation of each of the creation, resolution, and dispute of markets involves the use of "Flux Tokens" (FT). FT secures Flux Resolution with mechanisms that make it prohibitively expensive to attack the Flux service. FT is used to create markets, pay for resolution, trigger disputes, and govern fees. FT is not used to wager on outcomes of events: all trading utilizes stablecoins.

1.1 Creation

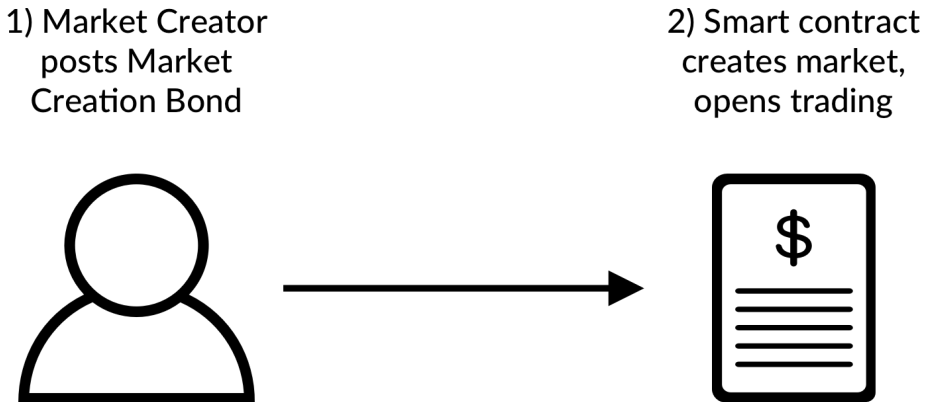
Any individual may create a market on top of the Flux Resolution protocol. A market validity bond requires the users to acquire a base amount of Flux Token.

1.1.1 Market Validity Bond

The market validity bond is a refundable bond that discourages the creation of markets that are "unhealthy" to the Flux service. "Unhealthy" markets

are markets that have no valid outcome (aka invalid markets). If a market resolves to the “invalid” state, this bond is forfeited and paid to market validators. Otherwise, this bond is paid back in full to the market creator.

Once a bond is posted, the market is made available for trading. As trading ensues, market creators profit off of "shares" issued on a market’s outcome. These profits are distributed upon market resolution.



1.2 Trading

The trading stage immediately follows market creation. In this stage, “traders” can acquire shares of a market outcome at a market-adjusted price. The following flow describes a standard trading engagement involving Alice, trading on outcome A, and Bob, trading on outcome B.

1. Alice posts an offer to buy $30 in shares of outcome A at 30% odds.

2. Bob sees this offer and places an order to buy $70 of outcome B at 70% odds.

3. Alice and Bob’s orders are automatically matched. Their funds are escrowed into a smart contract until the market resolves.

4. If the correct reported outcome is A, Alice receives Bob's $70 as profit, as well as her initial $30 in invested capital.

5. If the correct reported outcome is B, Bob receives Alice's $30 as profit, as well as his initial $70 in invested capital.

Note: Once there is an offer to buy or sell shares of an outcome, that offer may transact on secondary markets. For example, if news comes out that places the likelihood of outcome B is 50%, Bob may want to liquidate his position in advance of the market resolving. This liquidation happens via the same contract displayed above.

1.3 Resolution

Once a market has expired (trading has ceased, and the resolution event has taken place), the resolution stage is triggered. In this stage, “validators” are assigned to report the outcome of a specific market according to a lottery process. Markets follow a “most popular outcome wins” approach - whereby the outcome that receives the most votes from validators is the correct outcome.

1.3.1 Market Creator Fee

Market creators can denote a fee for publishing a market. This fee has an upper limit at 5% of the total open interest in the market.

1.3.1.2 Market Affiliate Fees

To bootstrap virality in markets, a market affiliate fee can be included by the market creator to allow actors who share or promote a market to receive a percentage of open interest, which generates through their affiliate link.

1.3.2 Market Creation Cost

Market creators pay a refundable bond which guarantees resolution services, if there is no organic traded volume . The bond is similar in cost to the overhead cost for the work required to report on the outcome of an event.

Note: If a market creator is willing to pay a higher market validity bond cost, they receive additional validators (and therefore additional security) on their market's resolution.

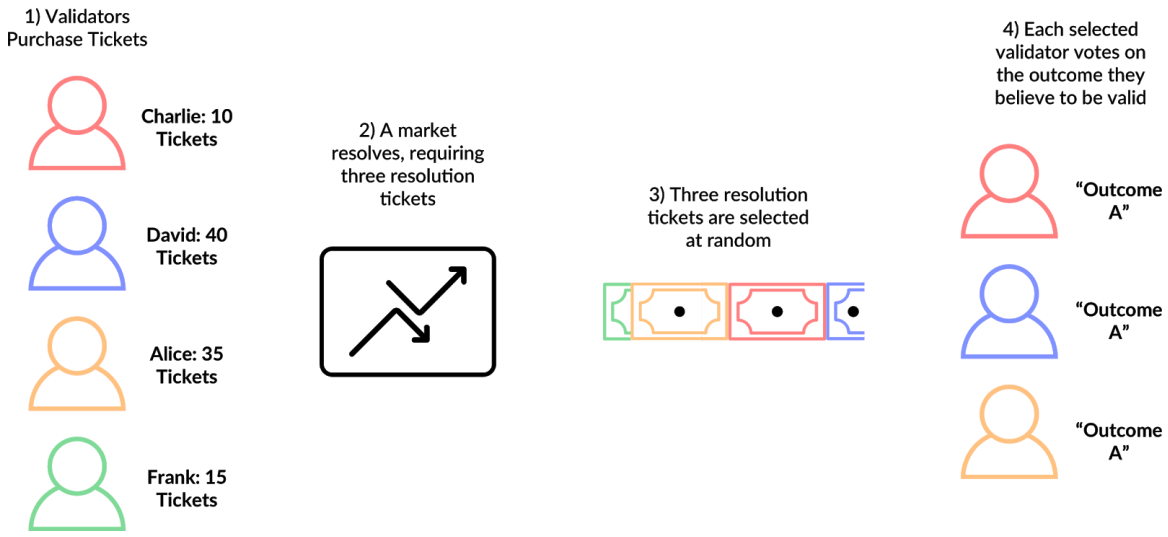
Overall, market creator profitability models as follows. Let *cf* represent market creation fee, let *cc* represent market creation cost, let *so* represent shares outstanding, let vb represent market validity bond.

$$(cf - cc) \cdot so - P(Invalid) \cdot (vb)$$

1.3.1 Validator Selection

The process of assigning validators to markets reflects that of a lottery. Every 24 hours (referred to as one epoch), validators may purchase "lottery tickets," which are acquired by locking up a predetermined amount of FT for the next EPOCH. Validators may continue the validation process indefinitely, subject to price changes in Flux Token. Once the EPOCH has begun, validators may not withdraw their stake until the relevant EPOCH has concluded. These lottery tickets are numbered 1,..., n.

Each time a market requires a resolution, a random number is generated, selecting a subset of the lottery tickets as "winners."



These selected validators have the task of reporting the outcome of the assigned market. They are then compensated for this action once the market is finalized.

1.3.2 Resolution Sources

Once a validator has been assigned to a market, the validator is ultimately responsible for reporting a correct outcome. Market creators can specify a "recommended API," which validators can use as a basis for their resolution. Validators are additionally able to seek data from alternative API sources or report outcomes manually.

1.3.2.1 Fallback Resolution for Data Driven Markets

The ability to seek data from alternative API's/report manually is what is referred to as an "escape hatch." This hatch is always made available - therefore, if a recommended API is invalid/crashes, reporters have an outlet to provide resolution services.

Markets that do not specify a public API are not supported in version 1 of the Flux Resolution protocol.

1.3.3 Resolution Compensation

In exchange for resolution services, validators who vote per the popular outcome split the validator fee as compensation. The validator fee is a function of open interest traded in markets. . As such, rewards may vary from *cc/n* to *o*cc/n*, where *cc* represents creation cost, *n* represents the number of validators, and *o* represents the number of outcomes.

$$MinReward: \frac{CC}{n}$$

$$MaxReward: O \cdot \frac{CC}{n}$$

1.3.4 Validator Payout

When the "big bang" occurs at the start of a universe, the Validator Payout is set at 1.00%, which is adjusted at each EPOCH to regulate the market cap to ensure protocol integrity. The Validator Payout is calculated by taking the current open interest to derive a target market cap with the formula shown in 1.1.2. A minimum Validator Payout is set at 0.01% to ensure Validators still have an incentive to provide resolutions to markets. A maximum Validator Payout is set at 33.33% of potential profits to ensure interaction with Flux remains attractive for traders while still retaining the security of the protocol during a black swan event. Let *t* represent the target market cap, let *c* represent the current market cap, let *v* represent the validator payout.

$$\max\left\{\min\left\{\frac{t}{c}v,\ \frac{333}{1000},\right\}\ \frac{1}{10{,}000}\right\}$$

1.3.5 Minimum Validators Threshold

The minimum amount of validators on any given market is three. This minimum ensures that there is a majority vote of ⅔ on every market. The minimum validator threshold adjusts as a function of open interest. The target is to ensure that the total value of the bond from validators assigned to a market meets a minimum threshold of 1% of the market's open interest. This function then denotes the number of validators per market, with the minimum always being three. If you want to increase assured security beyond this recommended threshold, a larger Validator Bond can be placed at the creation of the market, which ensures (n) amount of validators over the minimum threshold of three. The Validator Bond is calculated using the current Validator Payout fee average from the previous EPOCH. This bond allows a market to retain validators over the minimum amount. If the market experiences virality and the open interest supports more validators than the minimum threshold of three plus extra (n) amount of validators, the market creator can reclaim the validator bond, if the bootstrapped security is no longer viable. Let *vb* represent the validator bond, let *vp* represent the average validator payout, let *r* represent the previous EPOCH range.

$$vb = vp, r$$

1.3.6 Validators per Market

As open interest scales in markets, it's imperative to decrease the likelihood of disputes. We accomplish this decrease, by leaving the number of Validators per market open as a function, which is adjusted on a case by case basis. The function uses Open Interest in a market to determine the value which is at stake. To ensure a minimum quantity of validators are selected, 1.00% of open interest in the selected market is used to quantify the security level required, and thus the number of validators. Let *o* represent the open interest in a market, let *x* represent

1.00%, let *s* represent the stake per validator, let *n(v)* represent the number of validators selected per market.

$$n(v) = \frac{o(x)}{s}$$

1.3.7 Cost to be a Validator

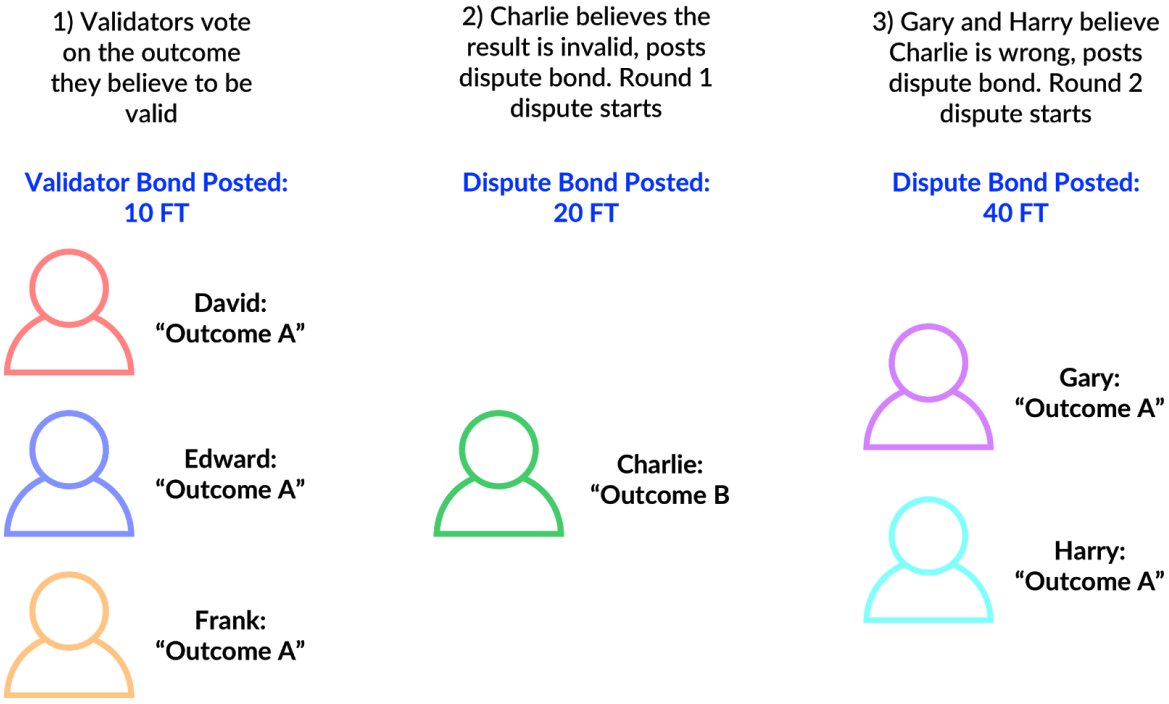
The Flux token is volatile as it's not directly pegged to a FIAT based asset, as with stable tokens like DAI. For this reason, a fixed amount cannot be hardcoded into the protocol as the market cap of the protocol naturally fluctuates based on market conditions and open interest locked in markets. For this reason, we have devised a function that is initially hard-coded to create a benchmark for Validator Cost. The protocol launches with a 0.01% hard-coded Validator Stake.

After an established benchmark is selected, the protocol will rely on a DAO structure, which gives token holders the ability to adjust the validator stake cost each EPOCH. This adjustment allows the community to ensure there is a diverse collective of validators operating on Flux. This function is done by calculating open interest and adjusting the percentage based on how many validators currently stake. If the number of validators is too low, then the function can be decreased, if the number of validators is too high and is reaching points of lower profitability for validators, the function increases. Let *o* represent the current open interest in all markets, let *x* represent the percentage for the increase and decrease, let *n(v)* represent the number of validators, let *vs* represent the validator stake.

$$vs = \frac{o(x)}{n(v)}$$

1.4 Dispute

If any individual (creator, validator, public citizen) believes that a market has been resolved incorrectly within a variable time between 30 minutes to 24 hours of reporting, they may "dispute" the market. This variable time is based on open interest in each market. To dispute a market, one must post a "dispute bond," which scales in cost proportional to the number of validators required for the market. Larger markets are more expensive to dispute, while smaller markets are less expensive.

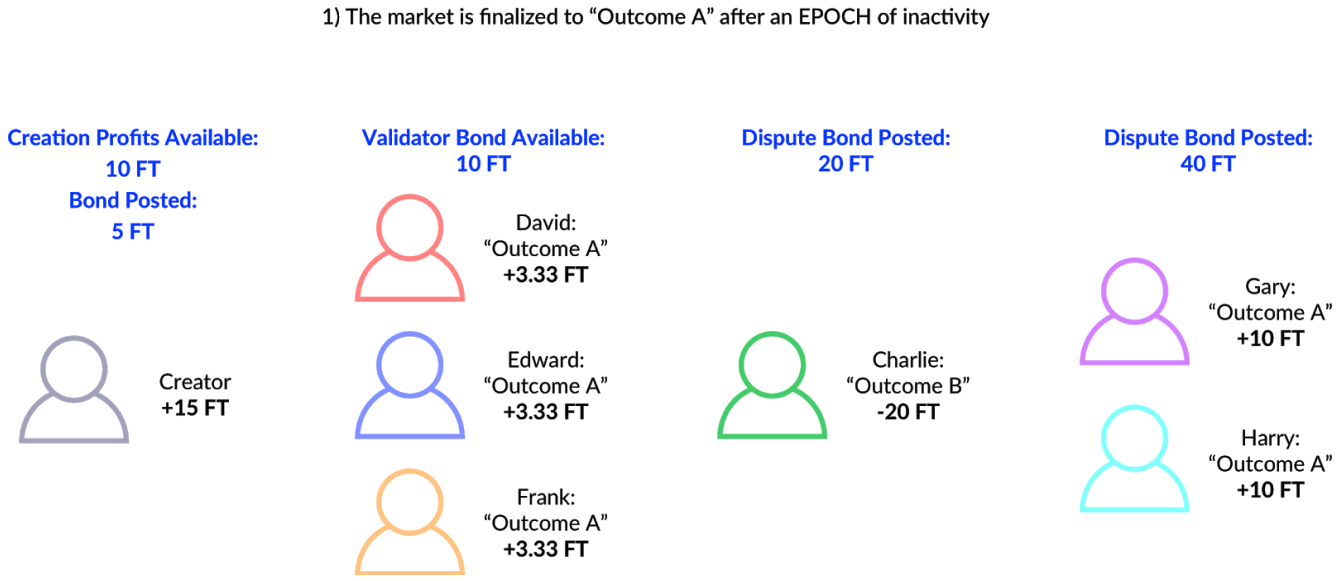


Once the dispute bond has been posted, the first dispute round is triggered. If the outcome generated in the first round is not satisfactory, the original disputers, or a collection of validators, can post an additional dispute bond to escalate the dispute to round two. This must be done within one EPOCH of resolution. This bond is always twice as expensive to insure disputes become exponentially more expensive each round, involving more token holders, drawing more awareness to the market. This process continues iteratively until twenty rounds have passed, or

2.5% of the global validator population is engaged in this market. At this point, a fork is triggered.

1.5 Finality

Once a market becomes finalized (either via original resolution or dispute), rewards are paid out to successful parties. The market validity bond is refunded to the market creator if the market does not resolve to "Invalid." Validators who provided the valid answer receive a function of open interest traded, and if applicable, the stake of incorrect validators. Dispute bonds staked on incorrect outcomes are split amongst disputing parties relative to their stake in the dispute.



2. Security Model

2.1 Validity Bond - *Invalid Markets*

Flux believes the cost of creating a market should be as close to free as possible. This belief is coupled with the understanding that malicious markets are damaging to user experience, and should thus be made expensive.

Such is the goal of the market validity bond - to disincentivize invalid markets.

Three values parameterize the market validity bond - the current Flux Market Cap, the frequency of invalid market creations, and the number of unresolved shares issued in a market (now referred to as “Open Interest”).

2.2 Resolution Cost

To secure markets, the number of validators requested to resolve a market (a metric of security) must scale with the open interest in a market. This scaling is necessary because more open interest allows for more profits resulting from dishonest actions.

Because validators expect income for their services, the market resolution cost must also scale with open interest.

2.3 Profitability

Market creators earn income via a user-specified market creator fee. For creation to be profitable, this fee must follow the below relationship:

$$0.05 \geq CreationFee \geq CreationCost + P(Invalid) \cdot Bond$$

2.4 Resolution Staking Requirements

Oracles that participate in the resolution process receive rewards or are held accountable for their reporting decisions. As such, to participate in the resolution process, oracles must escrow Flux Tokens. These escrowed FT’s grant “vote tickets,” which are selected at random to resolve specific markets.

If a validator resolves a market incorrectly, their Vote Ticket (and escrowed funds) is redistributed to honest protocol actors. These actors include i) the validators on a market who voted correctly, and ii) the individuals who posted bonds on the correct resolution (if a dispute arises).

On the flip side, those who resolve markets correctly continue to accumulate FT for their honest actions.

2.5 Resolution Selection Criteria

During resolution, validators are randomly assigned to markets to avoid collusion. The Near Random Number Generator[2] is used to generate a truly random number. Each EPOCH, a random number, is generated and made public. This number is then hashed repeatedly by the Flux protocol, with the last n/16 bits (n=# tickets) of the hash output mapping to a single ticket. This ticket is then assigned to a market. This process repeats until all markets have all required validators.

Eg. n = 2 markets, each requiring m = 1 validator out of k = 16 tickets issued.

H(Random Number) = "0xA1B2C3FD5" → market 1 validator 1 = ticket 5
H(H(Random Number)) = "0QR4V3C8FFA" → market 2 validator 1 = ticket 11

This function ensures that validators are assigned the expected number of markets in a given epoch. Let m represent the number of open markets, let t represent the number of tickets held by validator i, let T represent the total number of tickets issued.

$$m \cdot \frac{(t,i)}{T}$$

2.6 Collusion

Validators adhere to a commit reveal scheme on all resolutions to mitigate collusion and response bias. Before the conclusion of a market, validators must publish a "commit" message to the flux contracts, consisting of the hash of their

[2] https://nearprotocol.com/blog/randomness-threshold-signatures/

vote, a salt, and their address. Reveals are submitted to the flux contract after the market's epoch. If a resolution is revealed in advance of market conclusion the validator will be slashed. Reveals that are never made are treated similarly to malicious reveals, and any reporting entity may similarly claim associated FT.

2.7 Invalid Markets

Markets resolved "Invalid" should not be able to enrich any party financially. As such, all entities that have purchased shares in a market resolved invalid earn back their invested capital at their share's purchase price. Also, the market creator bond is forfeited and distributed to validators who marked the market invalid.

2.8 Market Cap Adjustment

Flux connects with an assortment of on- and off-chain third parties to calculate the current open interest in markets and the current fully diluted market cap of the protocol. This connection allows the protocol to adjust the target market cap at the start of each EPOCH. The minimum protocol security threshold uses the current open interest to target an appropriate market cap. The target market cap should always be five times the open interest in the current EPOCH to protect the integrity of the security mechanisms. This mechanism protects against a black swan event in which open interest surges 66.6% in a single EPOCH.

2.9 Forking Procedures

Upon conclusion of either i) 20 rounds of dispute or ii) the collection of 2.5% of global staked validator pool FT in dispute, a fork ensues. This forking mechanism is nearly identical to that of the Augur protocol (see here for a description of their diligent work). The resulting "fork state" is a unique state that lasts a number of epochs. This state is highly disruptive to the Flux protocol and should be actively avoided.

In this state, up to n “children” universes are created - where n is the number of potential valid outcomes of the disputed market in question. Each of these child universes exists to perpetuate a single one of these n outcomes.

For example, if a market's outcome is disputed as either being A or B, child universe #1 would exist with the assumption that A was the correct outcome, while child universe #2 would exist with B as the ultimate outcome of the market.

A fork in the FT would also exist across these universes, creating FT1 and FT2 (one representing FT on the market of child universe 1, the other representing FT on universe 2). A single voting event is held to ensure the fork resolves (majority stake on a universe over one EPOCH wins). Upon making this final decision, the value of the child tokens corresponding to the non-valid outcome is ignored (as the valid outcome token represents the correct outcome of all Flux markets to date).

During these days of a fork, all other markets that have not been finalized are put on hold. New markets may not be created in the parent universe but may be created in either of the child universes. It is worth considering that, if the child universe that the market is created on is ultimately ruled “incorrect,” the value of the FT’s being transacted or earned in these child universes is ultimately disregarded as well. To make a market in a child universe, FT must be migrated to that universe. This action is irreversible → if the child universe is rendered invalid, that FT is non-redeemable. All parent universe FT must eventually be migrated to a child universe - the safest way to make this transition is to wait for the final market outcome.

3. Flux Token Use

Flux Tokens will be the base currency governing and securing the Flux Oracle. FT will be used for the following purposes:

3.1 Resolution

Each epoch, all validators must stake Flux Tokens in exchange for "vote tickets." These Flux tokens are subject to slashing/redistribution to honest actors. The more FT validator stakes, the more tickets they receive for resolution (and the more capital they can expect to earn from successful resolutions).

3.2 Market Creation

Flux Tokens are needed to post the "market validity bond" discussed in the above sections. This bond can be recovered in the case of a valid market or are distributed to validators in the case of invalid market creation.

3.3 Dispute

Disputers must post dispute bonds in Flux Tokens.

3.4 Fee Setting

Market creator fees, creation costs, and validity bonds each include several constants that may be adjusted by Flux Token holders over time. These constants represent effective governance over the Flux protocol.

Addendum: Note that Flux Tokens are NOT used for the below purposes:

- Trading on market outcomes (done in DAI)
- Paying market creator fees (creators paid DAI)
- Paying validators for initial resolution (done in DAI)

3.5 Lending

Aside from validating and market disputes, Flux Token can also be used to capture value from the open interest, which is naturally locked in escrow. All markets denominate in nDAI (a wrapped version of DAI on NEAR). nDAI

allows the protocol to swap DAI into CDAI[3] or DAI Savings[4] automatically. This open interest can then generate interest income while it sits in escrow. Let r represent current (APR) Annual Percentage Rate, let t represent APR broken down into seconds, let o represent open interest on the protocol, let d duration of time in escrow, let f represent total flux token supply, let i represent interest earned per token.

$$i = \frac{\left\{\left\{\left(\frac{r}{t}\right) \cdot o\right\} \cdot d\right\}}{f}$$

3.6 No Loss Market Mechanics

With the addition of the lending format to the protocol. It opens up an entirely new mechanic for markets. A precise fit is for markets that merit base from signaling rather than winner takes all.

3.6.1 No Loss Payouts

Implied in the name "no loss market," all participating parties retain their principal at the resolution date of the market. Rather than a payout deriving from one party to another. The payout is in the form of a user's funds opportunity cost for interest via lending protocols. This mechanism allows users to create a forecast on-chain using real money, which ensures more accurate predictions. The user on the correct resolving side of the market receives a payout based on the total shares owned in the markets against the total outstanding shares.

[3] Via Compound Finance
[4] Via Maker

4. Known Attack Vectors/Future Research

4.1 Self-Referential

Self-referential markets are markets that are designed to encourage malicious action. For example, a question such as "Will Attack X on the Flux Protocol work by January 1" would be a self-referential market. While there is no perfect solution for this attack yet, the attack may be discouraged by ruling these markets "invalid," or by disputing the payouts of the attackers if an attack were to fail.

A satisfactory solution to these markets will be left as a future exercise.

4.2 Parasitic Markets

Parasitic markets are markets that rely on the Flux oracle for resolution but don't contribute to validator payment. For example, a market which is hosted on "examplemarket.com" could rely on the Flux oracle to provide a resolution to a market. Examplemarket.com then uses this resolution to finalize its market without contributing to the payment for validator services. As of today, there is no solution to this problem.

4.3 Mirroring

"Mirroring" is a standard attack in decentralized oracles whereby an attacker coordinates all validators to utilize the same resolution source, which produces an invalid outcome. This attack is mitigated via the "dispute" mechanism, as well as the commit-reveal scheme.