KrawlCat Data Feeder Whitepaper

V 1.1
2019.03.22

Problems:

There have been many attempts to integrate blockchain technology with existing industries like supply chain, cloud storage and fin-tech. However, all of these attempts have come short because there is no way to reliably bring off-chain data onto a blockchain. While a blockchain system is safe, the data vendors responsible for providing real-world intelligence to these blockchains are not. With centralized companies responsible for the curation and distribution of this data, there has yet to be a decentralized and trust-less approach to providing off-chain data to a blockchain.

Besides the issue of centralization, these data vendors have a host of problems that affect the fundamentals of any blockchain business that needs off-chain data. Most notably, current data contributors are not compensated for their efforts, and the inefficient uploading speeds of these centralized vendors cannot meet a blockchain's transaction requirements.

As the development of decentralized applications continues, these projects need a trust-less way to bring-off chain data onto their applications. Otherwise, the public blockchains that are hosting these applications will never be able to disrupt the industries that centralized businesses are dominating.

Let's start with an example that contextualizes the value of a decentralized data feed:

- A Decentralized Application (Dapp) needs to quote the bitcoin price. But companies that list the Bitcoin price like Coinmarketcap are off-chain. So now, the company will need to check the price from an off-chain resource like the aforementioned Coinmarketcap or an exchange. However this poses several risks to the entity taking this information because:
    - The programmer could fake the price.
    - That exchange may have lower volume relative to the rest of the market; leading to inaccurate pricing.
    - The Dapp must use a server to keep updating the price which could break down.

1.2 Competitive Analysis:

The need for a decentralized data feed was not discovered recently. Not being able to reliably bring off-chain data to blockchains has always been one of the strongest inhibitors to adoption. As such, there have been several attempts to create a decentralized data feed. We will be analyzing some of this competition and highlighting their flaws as a truly decentralized data source:

- Oraclize: Oraclize's decentralized solution works in three steps:
    - You send a query to the Oraclize smart contract
    - Oraclize receives your query and makes the relevant request

        - Once they receive the data, they'll send a callback function to your smart contract where you'll be able to access the requested data.
    - The main issue with this process is, although they can provide off-chain data to a blockchain, a centralized entity is still performing this task. Relying on centralized entities to provide the data goes against the ethos of blockchain and decentralization; you are still putting your trust in an external entity. These vendors can compromise the data before it is even uploaded onto a blockchain.

- StreamR: StreamR is an off-chain solution for providing real-world data to blockchain projects. It works similarly to a stock-exchange or advertising-exchange in that anyone can purchase data from data vendors in a real-time marketplace.
    - The issue with StreamR is that, although their solution offers variety, it lacks the democratic voting mechanism that a truly decentralized solution requires. For a system like StreamR, it is still up to the data-receiver to do their due diligence and make sure they're purchasing the data from a reliable vendor.

Solution:

The KrawlCat Data Feeder (Feeder) uses a decentralized approach to acquiring internet statistics, providing a decentralized and trust-less data-gateway for blockchain applications to access real-world intelligence.

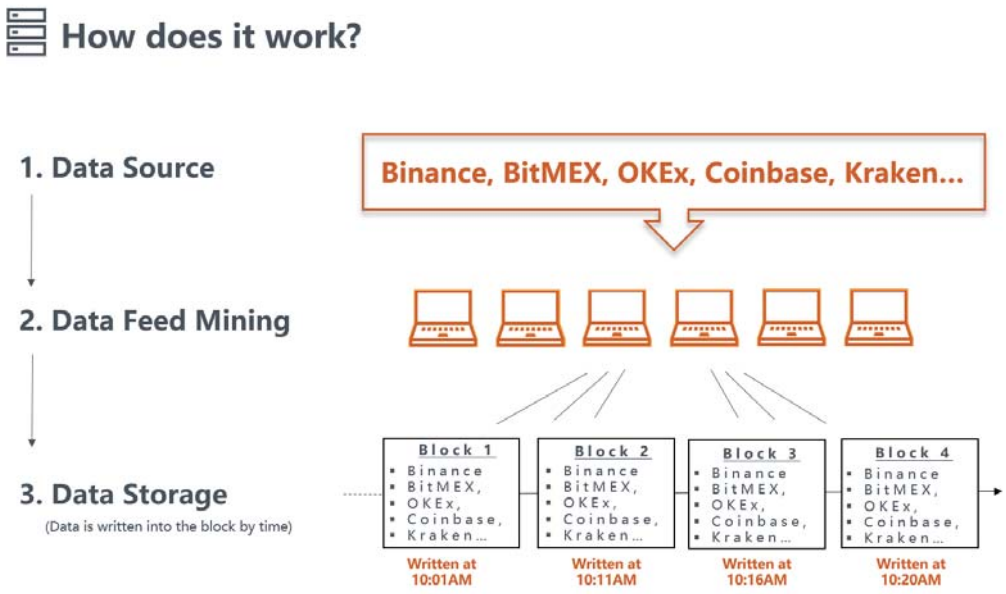
The Data Feeder does not provide hashing power to blockchains. Instead, Feeders scrape data from credible data vendors in finance, weather forecasting, aviation, exchanges, and any other industries that have numerical or categorical data. Together the Data Feeders make sure the data input is decentralized and can't be manipulated.

Product

KrawlCat Data Feeder is a scraping device for gathering data on the internet, and uploading it to blockchains in real-time. The more Feeders that make up this data feeding network, the more reliable the data would be. Once the network has a sufficient amount of Feeders in the ecosystem, then the Data Feeding process would be considered decentralized and trust-less since the majority approves the truth.

All data feeders are supported by KrawlCat Data Feeding Smart Contract, which acts like a Data Feed Marketplace (marketplace contract), where Feeders choose what data to feed into the blockchain.

Back to the scenario in Section Problem, if that DApp can't found a reliable Bitcoin price data source, then let's see what the process will look using KrawlCat Data Feeder. Every Feeder will get the price info from multiple exchanges' APIs, each feeder device will upload the price info to the Marketplace contract to calculate the "most moderate bitcoin price" among thousands of data-submissions from the Feeders. Now the DApp can get this decentralized Bitcoin price info from blockchain.



Besides retrieving the Bitcoin price from cryptocurrency exchanges, Data Feeder can also get numerical data or categorical data from Bloomberg, NASDAQ Index, Google Flight, and Yahoo Weather…

What are users mining?

This process is similar yet different from Bitcoin or any other Cryptocurrency mining. Data Feeder uses computing power to scrape data from internet and upload the data to the Data Feed Marketplace Contract, which aggregates all the

data and calculates the most moderate data using a Proof of Data consensus. There will be thousands of Feeders working in the network, and acquiring data independently. The Feeder also do hash validations, but the validation only happens when the Feeder tries to upload data to the blockchain, so there is little need for massive computing power.

Decentralized Scraper Network

Looking at each Feeder as a singular entity, none of these devices could be considered as a “trust-less data provider”. However, when there are 1,000 Feeders continuously working together, we can see the decentralized system take hold. Now we have a comprehensive network of devices all working together to prove one thing, determine what the most reliable data-set is for a given-task.

Within the context of Feeder, the idea of “decentralization” is that every Feeder is trying to scrape off-chain data to the blockchain independently, and the “acceptable range” represents the Feeders most widely accepted data-set.

Data Feeder Device

Data Feeder is a Linux-based microcomputer. Linux is safe against malware, requires little maintenance, and requires no installation; just connect the device to a power source and you are good to go.

- Easy to use
  KrawlCat engineers have hardcoded the entire scraping program into the device, so all you need to begin using the device is a power source and an internet connection.

  Data Feeder comes equipped with a friendly UI design and intuitive touch screen. So no prior technical skills are needed to use the device. You can view and change the device's settings by simply tapping the screen.

  More details could be found in the ***Feeder User Guide***.

- Built-in Wallet
  Each Feeder has a built-in wallet: Far too often are people leaving their digital assets on centralized exchanges, and thinking they they have custody of whatever is in their balance. In reality, this just shows you your balance, but this is not the same of having custody of that balance. The Data Feeder has a built-in wallet, so users actually possess a hardware wallet solution for their tokens and can send and receive funds anytime. The wallet is based on Linux, and comes with anti-virus software.

- Auto Updates
  Networks can undergo system updates at anytime. If you are not online at the time of this update, this can be very problematic. Data Feeder has an

auto-update function so users never have to worry about running obsolete software, ensuring these devices are always connected to the blockchain. Currently the Feeder only supports the Ethereum network, but we are working diligently to launch on the EOS network. KrawlCat Engineers will keep updating the software to ensure the Feeder's firmware is always up-to-date.

- Low Energy Consumption
  According to the Proof of Data consensus, Data Feeder doesn't require a substantial amount of computing power to scrape data. Each Feeder is just collecting internet statistics and uploading the statistics to Data Feeder Smart Contract. The general energy consumption for this task is 3-5W, compared to a regular mining devices energy consumption of 1350W.

  In addition, the fan inside the Feeder generates only 20dB in noise, the device is almost silent, even when it is running at full capacity.

Consensus and Implementation

Data Feeder uses Proof of Data and Proof of Stake for the reliable distribution of this off-chain data.

Proof of Stake

5,000 tokens will be the enrolment fee for joining the mining pool.

As a security measure, the Data Feeder Smart Contract has been coded to check the balance of each Feeder. Only if the balance is more than 5000 tokens can the Feeder be accepted to the network. This high upfront cost was designed to dissuade hackers from trying to compromise the mining network.

Note that the ownership of token doesn't necessarily generate any revenue for users. Any users must contribute valid data for the data buyer to claim reward.

Proof of Data

During every processing cycle, two data selections are made based on the mining procedure highlighted in section xxxx.

1. Each Feeder will get the same data-set provided my multiple vendors. Looking at the price of Bitcoin, we would scrape data from 6 exchanges. The Feeder will then send this price info to smart contract.
2. Secondly, the smart contract will generate a "moderate range", so that Feeder contributions are only valid if their submitted data-set falls within this "moderate range". Thousands of Feeders are working simultaneously and getting fed statistics from our list of data vendors. So once the smart contract collects these numbers from the mining pool, the smart contract

will filter out submissions that are outside of this reasonable range, and pick a median value from the values that are within this “moderate range” If the Feeder provides valid data then they will be rewarded, while Feeders who submit invalid data will be penalized.

Harmonic Series

The Feeder network is decentralized, each feeder can join or leave the network anytime they want, and each user can send request to smart contract anytime. In this way, it brings huge difficulty if we try to track the decentralized system using “time” variable.

Let’s think in this way: buses are scheduled by time. If bus company want to know how the buses pick passengers at a station, it is time that how bus company know when and which bus picks the passenger.

8:00am bus arrives, picks up 5 people
8:10am bus arrives, picks up 3 people
8:21am bus arrives, picks up 7 people (1-minute delay)
8:31am bus arrives. picks up 4 people

What if bus comes randomly? In decentralized system, buses can arrive at the station whenever they want, some drives can even choose to quit the job. So, it might act like this:

8:00am bus arrives, picks up 5 people
8:27am bus arrives, pick up 10 people
9:02am 3 buses arrive, pick up 8 people, 5 people. 7 people respectively
11:59am 2 buses arrive, pick up 21 people, 32 people respectively.

The bus company still want to know how the buses pick up passengers, but time is no longer a good dependence because bus can come whenever they want, the variable time useless for counting buses. It seems bus number is easier to count buses,

Bus1 picks up 5 people
Bus2 picks up 10 people
Bus3 picks up 8 people
Bus4 picks up 5 people
Bus5 picks up 7 people
Bus6 picks up 21 people
Bus7 picks up 32 people

In KrawlCat smart contract, the concept of Harmonic series is used for managing upcoming requests and calculate the balance for each user.

Harmonic series looks like below:

$$\sum_{k=1}^{\infty} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots$$

For each transaction in the smart contract, we don't manage them by "time" since all requests in the decentralized system happens unpredictably. Now we use "state" to record events, "state 7" means there are 7 events from the very beginning of this chain, and thus there are fractions in the formula, "state 8" mean there comes the 8^{th} event, for example, a Feeder joined the network, and a new fraction is added to the end of "state 7". Each event is a "trigger" of the state and will add a new fraction to the previous state.

The mission is to calculate the balance for each user. So, whenever an event happens, it will trigger a balance change in the entire smart contract. Now we need to keep track of it. Let's define the numerator as token added to the miners, the denominator means the number of miners, it's clear that $\frac{\text{token for miners to claim}}{\text{number of miners}}$ means the reward for each miner within that "state". We call the sum of $\frac{\text{token for miners to claim}}{\text{number of miners}}$ Round Mask, and we use this to calculate the user balance.

Each entering and leaving the system will cause a change to the token number of feeders in the system, so we record changes after each trigger.

$$\text{RM} = \text{Previous RM} + \frac{\text{token to claim 1}}{\text{number of miners 1}} + \frac{\text{token to claim 2}}{\text{number of miners 2}} + \frac{\text{token to claim 3}}{\text{number of miners 3}}$$

Trigger1	Trigger2	Trigger3
Feeder 207 enter the network	Feeder 792 enter the network	Miner 207 left the network

If Feeder 207 joined the network at some time, we call this event Trigger1. Before the next Feeder join or leave the system, Feeder 207 is continuously getting rewards. After a period of time Feeder 792 joined the network, we call this event Trigger2. Because the total number of Feeder has changed, the average reward will change accordingly. At event 3 Feeder 207 left the network, so the token rewarded to Feeder 207 should be between Tigger 1~2 and Tigger 2~3, which is $\frac{\text{token to claim 1}}{\text{number of miners 1}} + \frac{\text{token to claim 2}}{\text{number of miners 2}}$.

Round Mask and Player Mask:
We use Round Mask (RM)to record the entire transaction state, and Player Mask (PM) to record one single user's transaction state.

kn represents the number of keys that the player purchases in the n-th transaction of the game, including the kn keys that the player buys.

The formula to calculate the number of Ethereum in the system with the ktotal number of keys.

$$eth(k_{tot}) = TotalEth = \frac{A \times k_{tot}^2 + \frac{B \times C \times k_{tot}}{2}}{C^2}$$

Then we would define the price to purchase k2 number of keys at once, when there are k1 number of keys already in the system before the purchase:

$$\text{price} = \text{eth}(k1 + k2) - \text{eth}(k1) = (k22 + 2k1k2)\frac{A}{C} + k2\,B\frac{B}{2C}$$

Round Mask is the series that describes the historical states of the Profit Per Key, until the N-th round .We define that the round is ended after N transactions, then the final Round Mask of the this round is

$$R_M = \frac{d_1}{k_1} + \frac{d_1}{k_1 + k_2} + \frac{d_3}{k_1 + k_2 + k_3} + ... + \frac{d_N}{k_1 + k_2 + ... + k_N}$$
$$= \sum_{i=1}^{N} \frac{d_i}{\sum_{j=1}^{i} k_j}$$

Profit Per Key

Now for an arbitrary transaction n, with $1 \le n \le N$, the Profit Per Key is the fraction of money from this transaction that went into the dividend pool that the player receives as his/her dividend for himself for each keys. For the n-th transaction, it is calculated as:

$$PPK_n = \frac{d_n}{k_1 + k_2 + ... + k_n} = \frac{d_n}{k_{TOT}}$$

Smart Contract Structures

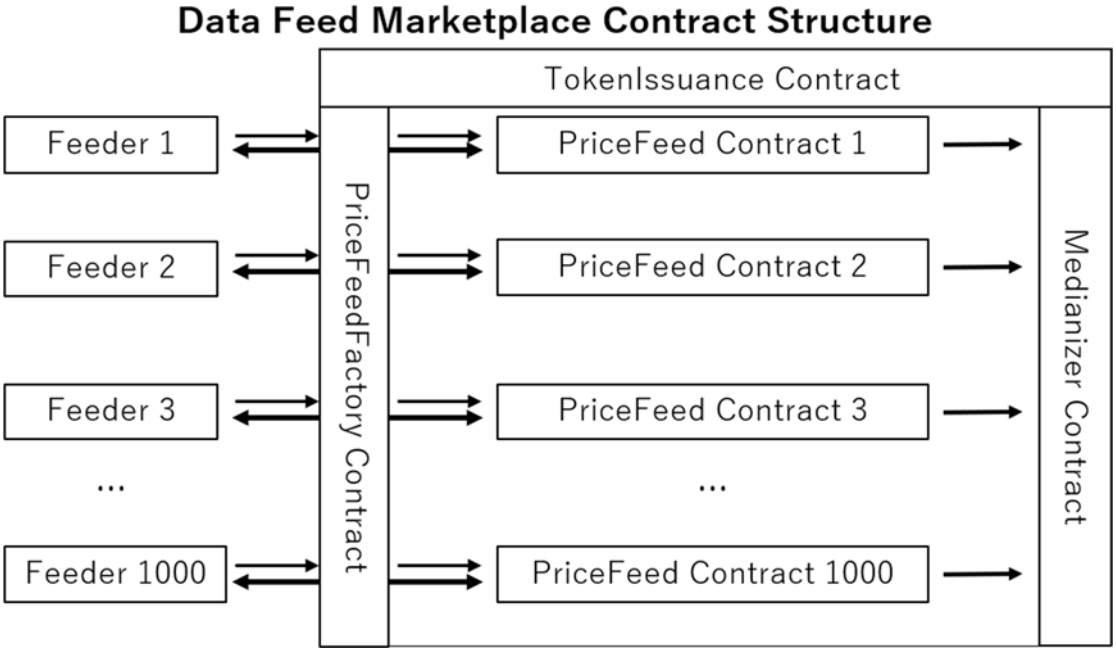
KrawlCat Data Feed Marketplace Smart Contract consist of 4 major contracts:

1. **TokenIssuance Contract:**
   The Token-Issuance contract defines the token distribution amongst Feeders. The Token-Issuance contract acts like a bank in that the contract tracts the balance of all users and is responsible for the distribution of incoming mining profits.

2. **Medianizer Contract:**
   The Medianizer possess a list of wallet addresses that have been approved to join the mining pool. The on-boarding criteria is whether the Feeder has 5,000 tokens in their Feeder-wallet. Medianizer also sorts the networks incoming data-submissions, and then finds the median based on these submissions.

3. **PriceFeedFactory Contract:**
   To start mining, new Feeders need to call the Price-Feed-Factory contract to receive a designated PriceFeed contract that corresponds to their specific device. Each mining device will always have a corresponding unique Price-Feed contract. To gain access to the Price-Feed-Factory Contract, Feeders will have to stake 5000 tokens.

4. **PriceFeed Contract:**
   Each Feeder uses their own PriceFeed contract to store the data collected on their device, and then upload the results to the Medianizer. To gain access to the Price-Feed Contract, Feeders will have to continue staking the 5000 tokens that they used to gain access to the Price-Feed-Factory contract.



Design of Marketplace Smart Contract

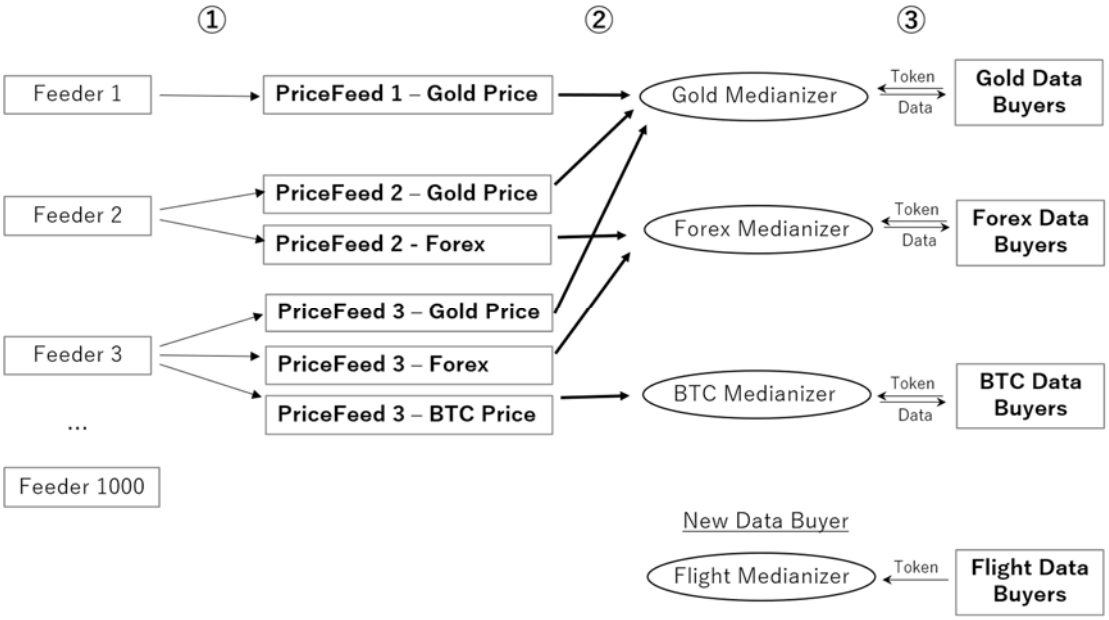
KrawlCat Data Feed Smart Contract acts like a free marketplace for all data feeders. The Marketplace contract holds a list for data vendors that each feeder can choose to mine, as well as a portal for data buyer to create new data orders.

Data Feeder \ Data Buyer	Gold	FX	BTC	Flight (New data Buyer)
Feeder 1	✓			
Feeder 2	✓	✓		
Feeder 3	✓	✓	✓	
...	...	...	...	
Feeder 1000	...	...	...	

In this example above, currently there are 3 data buyers in the network for Gold price, Foreign Exchange, and Bitcoin price respectively.

- Feeder 1 choose to only provide data for Gold price
- Feeder 2 provide data for Gold price and Foreign Exchange
- Feeder 3 provide for all of the 3 data sources
- A new data buyer just set up an order for Flight Info

From the perspective of contract flow, this example can be expressed as below:



- Feeder 1 only have PriceFeed Contract for Gold price, and will contribute to

Gold Medianizer. So the Feeder will receive token reward from the Gold Data buyers.

- Feeder 2 have two PriceFeed contracts for both Gold and Foreign Exchange, so Feeder 2 will upload data to the two Medianizer contracts for Gold and Foreign Exchange,
- Feeder 3 will have three PriceFeed Contract and contribute to three Medianizer contracts.
- Flight Information buyers pay token into the Flight contract to create the Flight data order, soon some feeders will start providing data.

Mining Procedure

1. When you start up the Feeder, the device will automatically perform an update-check to install the latest version of Feeder's firmware.

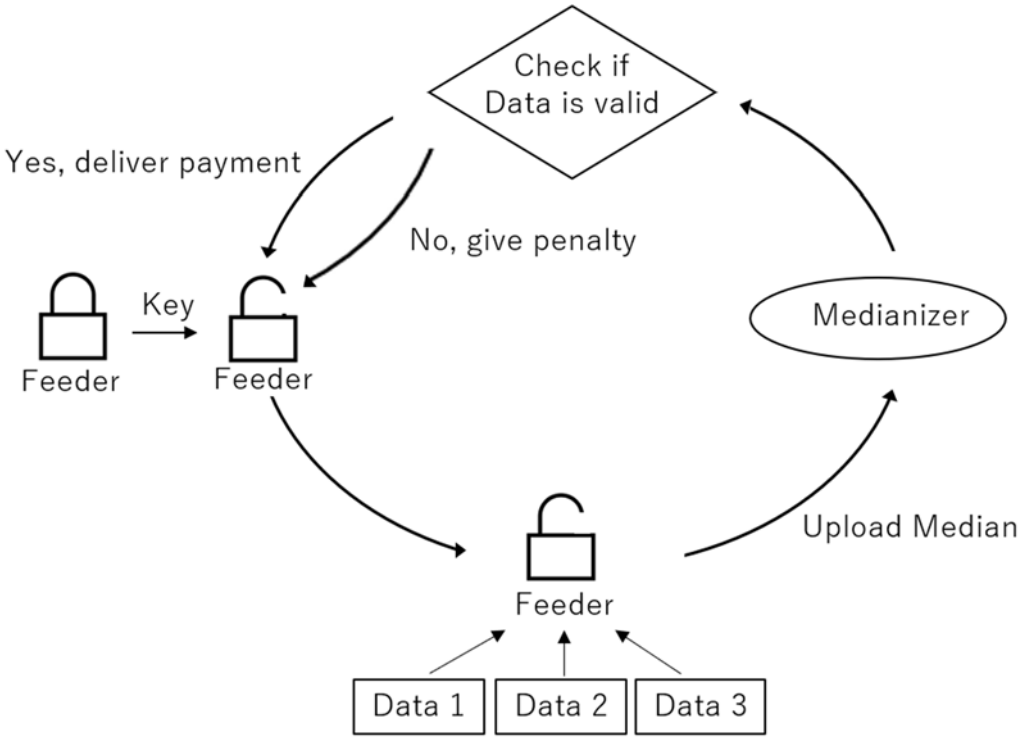
2. Users will register a private key or input an existing key to start mining. During this procedure, the Feeder will create a smart contract to store data, every Feeder has a corresponding smart contract. Right after this process, the Feeder will call the smart contract to determine if the user has staked the necessary capital and is therefore eligible to join the network.

3. To begin mining, Feeders will automatically get data from data vendors' APIs. Each Feeder will send all of their data to the medianizer where it will calculate the median.

   Please note: The calculation of this median value occurs on the back-end of each mining device. But the Feeder only will upload the median value to the smart contract when:
   a. There is a price discrepancy of more then 1% between the most recent price submission, and the current prices provided by the Vendor's API's.
   b. The last price update was more than 6 hours ago.

4. The Marketplace Contract will collect data from all Feeders in the network. The Smart contract will filter out values that are not "within the moderate range" and calculate the median based off the remaining data that is "within the moderate range". The contract will then store the median number on the blockchain. Since the majority of the Feeders are providing the same data-point, this final result will be the data that is actually sent to businesses that are using our Decentralized Data Feed.

5. In the meantime, the Marketplace Contract will decide which Feeders provided data within the moderate range, and outside of the moderate range. Feeders within the range will be rewarded, and Feeders outside of the range will be penalized.


Check if
Data is valid
Yes, deliver payment
No, give penalty
Key
Feeder
Feeder
Medianizer
Upload Median
Feeder
Data 1
Data 2
Data 3

Global Strategy

First Stage:
Sell 100 Beta Feeders to a pre-vetted group to establish a functioning network. At this stage, feeders are functioning, yet frequent updates will be made to ensure the stability of Feeder network.

Second Stage:
Open Feeder sales channel and sell 1000 Feeders globally, and start the multi-chain support like EOS network. 1000 Feeders are sufficient at this stage to be considered decentralized, and the Marketplace contract will start engaging commercialized data buying orders.

Third Stage:
KrawlCat team will stop selling Feeders, gradually reduce its role on the Data Feeding Marketplace. The team will opensource the Feeding software to all community, start accepting every third-party Feeder to join the network, and finally let the feeders in the network reach autonomy.

Circulation:

There are two major means of circulation of the token.

1. Since the Feeders are used to inputting this trust-less data into the blockchain, those who pay for the data will transfer the token into the Marketplace contract. The Marketplace contract will then distribute the earnings based off each Feeders contribution.
2. 1 million token will be minted each year, and the distribution of tokens will be delivered to the Feeders, proportionate to their contributions.

For the first distribution method, the revenue for Feeders comes from the buyer of data source. All data buyers will have to purchase tokens to pay for the data buying order, and they can only purchase these tokens from the secondary market, where Feeder could cash out their token earning. This process is the major value exchange, and keeps the Feeder ecosystem self-sustainable.

For the second distribution method, the annual token liquidation also goes directly to the Feeders. Feeder Devices are the fundamental supporters of the Data Marketplace, and their existence is crucial to maintain the network and its decentralization. The annual token addition will reward their effort.

In addition to the second point, the network will start accepting third-party Feeders, they also need to deposit 5000 tokens to use the Feeding software. The annual token addition will help keep the token price stable, which makes the token liquid (and affordable for upcoming Feeders.)