A decentralized data market place as a source of truth

The Kylin Team

August 31, 2022

The Qilin was a magical beast native to China. They were deer like creatures with the ability to look into a person's soul and unveil if they were truely pure of heart.

Abstract

As a parachain to Polkadot, a layer-0 protocol and multichain network laying the foundation for the future of Web3, Kylin is seeking to fill in the much needed area of blockchain oracles and the democratization of data.

1 Introduction

Martin Heidegger in his essay "*On the Essence of Truth*" states that the freedom to chose from a varied horizon of possibilities is an essential condition for the possibility of truth and falsehood. He continues by saying truth is not only freedom, but also the uncloaking of secrets, the revealing of information. In essence, freedom lies in the openness of things... This is what drives the spirit of open source and the ability to know how systems that we use are made.

He aslo stated that the world is not simply given. It is a constantly developing world opening new fields of possibilities, and that to think of reaching some truth about the world statically is a mistake. To him, thinking about the world dynamically in the search for truth is essential... The world of dynamic NFTs reflect this by allowing us to record data about the world in a certified, updated and decentralized manner.

He continued by stating that, the search for truth is a dynamic process of unconcealment and disclosure. In the context of a blockchain oracle, constantly including information disclosing the source of the information along with a NFT's metadata at every block satisfies this condition.

The blockchain oracle problem is one of the most important barriers to overcome in the world of decentralized applications. It refers to the inability of confirming the veracity of the data collected by oracles, the blockchain's way of communicating with the outside world. How do we connect a deterministic world to an exponentially growing world of data? Although alot of projects claim to have solved this problem, blockchains cannot interact with external data with intrinsic built-in functionality. A blockchain is an isolated network and it is exactly this isolation which makes it extremely secure and reliable. Advancements on resolving this problem were mostly fueled by the need for accurate price feeds. Since the advent of third generation blockchain the ability for smart contracts to address human litigious matters has manifested itself.

Oracles perform the following:

- **Listen**
  Poll for data requests
- **Extract**
  Gather information from external APIs
- **Format**
  Make blockchain data compatible
- **Validate**
  Ensure that data is impartial and correct
- **Compute**
  Perform computation too demanding for blockchain ressources
- **Broadcast**
  Make data available to entities requiring it

Oracles should deliver data to the most:

- **Accuracy**
  Should be more precise and less approximate
- **Validity** Should correspond to the real world
- **Reliability** Should always be available
- **Timeliness** Should not be out of date
- **Relevance** Should be pertinent and useful
- **Completeness** Should be whole and useful

Systemic entities and systemic actions [9]

- **Ground Truth**:
  The goal of the oracle system is to relay the ground truth (i.e., the real true data) to the requester of the data.

- **Data Sources**
  Data Sources are entities that store or measure a representation of the ground truth. There are a diverse set of data sources: databases, hardware sensors, humans, other smart contracts, etc.

- **Data feeders**
  report off-chain data sources to an on-chain oracle system. In order to incentivize truthful data reporting, an oracle system can introduce a mechanism to select data feeders from a collection of available data providers. An incentive mechanism can be collateral-based, such as staking, or reputation-based to find a reliable set of data feeders for each round of selection.

- **Selection of Data Feeders:**
  The process of determining which data feeders should be used in an oracle system can be categorized into two main types: centralized and decentralized selection.

- **Aggregation**
  When data is submitted by multiple data feeders, the final representation of the data is an aggregation of each data feeder's input. The aggregation method can be random selection or algorithmic rule-based, such as using weighted average (the mean) or majority opinion. The design of the aggregation method is one of the most important aspects of an oracle system, as intentional manipulation or unintentional errors during the aggregation process can result in untruthful data reporting by the oracle system.

- **Dispute Phase:**
  Most oracle designs allow for a dispute phase as a countermeasure to oracle manipulation. The dispute phase might correct submitted data or punish untruthful data feeders. The dispute phase will also introduce further latency unless streamlined procedures are put into place.

Kylin Network vision:

- **Configurable framework for democratically creating and managing oracle system entities**

- **Service level agreement (SLA) between the providers and consumers of data**

- **Multi-level oracle data agglomeration to balance security, authenticity, persistency and speed**

- **Transparency and traceability of data provenance**
- **Democratization of the data management process**
- **Multi purpose dispute resolution mechanism/ truth machine**
- **Extra voting rules which can help to assess and strenghten the validity of a vote**
- **Data and feed ownership management with NFT and metadata**
- **Free data market**

2 Previous work

The Kylin Oracle Pallet is a substrate runtime module which pulls external data by means of offchain workers which reside on collator nodes. This data can either be stored on chain or on a database provided by the Kylin API's private network. The data can be pulled from urls directly built into the Kylin API or by specifying a custom API from which to pull from. This can be done by calling the collator's provided extrinsic or by having a completey different parachain (ex. Acala) calling the collator's Kylin Oracle Pallet extrinsic. The Kylin network resides on a parachain which ensures that the security and network performance is guaranteed by Polkadot and Kusama networks.

Entities that enable the current network:

- **Data provider**
  Consumers may contribute to a feed or query this feed for use in their application.
  - Parchains
    - contribute to federated feed through xcm
  - Kylin collator ocw
    - contribute to feed residing on OCW TBD
  - External apis: ex. RapidAPI/API3
  - Existing Kylin API API which allows to save on to a postgres database the results of calls to built-in or custom enpoints.
- **Kylin collator node**
  The Kylin Oracle nodes are collators to the Kylin parachain which validated by the relay chain to the Polkadot network.
- **Data Consumer**
  Query feed for use in their application.
  - Parchains

    * consume feed through xcm
- Dapps

3 Current research and projected work

The previous iterations consisted mostly of deploying substrate offchain workers to store data in a private database using an API. While researching similar projects and current litterature we have distilled best practices in order to upgrade the Kylin oracle's functionality.

3.0.1 Configurable oracle

Although steps to minimize risks litigious scenarios regarding data are taken, a situation where a dispute arises on data veracity could manifest itself at some point. The ability for a user/organization to configure the components of the oracle system could give a sense of assurance and responsability to the consumer about selected choices. The creators of a feed can specify what their oracles are, which algorithms they are usin to the consumers of this feed could so they could freely choose if they want to consume the data with full knowledge of their choice. Systemization of knowledge and dividing oracles into modules can be beneficial for enhancing security but also to manage user expectancies.

There are many cases where the ability to configure the components of an oracle might be advantageous:

- **Type of Data**
  The first use of blockchain oracles was for price feeds. As third generation blockchains emerged the need for a source of truth other than numerical is manifesting itself. *The consumers would want to subsribe to a variety of feed types.*

- **Selection of various trusted APIs**
  In the case of a price feed for example, using a single spot price is highly unreliable because it represents the price at which buyers and sellers are willing to accept on the spot on a certain exchange. The system using this feed will be disfunctional if the API goes down or is victim of manipulation. It is discouraged to use this value alone and proper selection of several reliable APIs will ensure an accurate and persistent feed. *The creators should be able to select from a list of approved APIs*, passing down this trust to consumers who can choose use the data because they know where it came from and how it was aggregated or treated with each SLA they are signing with each feed. [4]

- **Trusted algorithms**
  Much investigation still needs to be done on the most efficient manner to obtain the most representative value out of a running queues or static buffer. The ability to choose this will allow users to do their own research and either *select the most efficient configuration according to their needs* (speed vs. security vs. authenticity) or choose a default setting with conscious knowledge of the advantages or consequences of using that feed.

- **Signed SLA**
  A signed agreement between reporter, consumer and host chain can be required when aquiring a feed to promote a speedier and more harmonious resolution.

- **Private chains**
  In the case where data remains within the enterprise context, *the company should be able to create custom feeds containing pivate datasets* it wants to work with for their consumers or applications.

- **Push oracles**
  In the case where data needs to be obtained from more varied sources, the contribution of any community member wanting to *contribute to a feed by pushing data onto the chain for a reward* could bring a dimension of variety appealing to the creator of a feed.

3.1 Oracle system manifest

Keeping track of who owns the data, where the data comes from, when it was collected, and which algorithm was used to treat it can be recorded in a manifest that can published with each block. This information can be used as a guarantee for the expectated data for the next block.

3.2 Service level agreement

A service-level agreement (SLA) is a contract between a service provider and its customers that documents what services the provider will furnish and defines the service standards the provider is obligated to meet. More specifically, in the context of the Kylin configurable oracle system, it is an agreement that the user accepts the configuration the provider selected when creating the data feed. This agreement can be renewed with each block and can take the form of a Ricardian contract or json object. The user can opt out of this contract by choosing to subscribe to a different feed. A Ricardian contract is a digital contract that functions as a legally binding agreement between two parties based on agreed-upon terms and conditions. The contract is cryptographically signed and verified using the blockchain, but is readable by both people and machines.

3.3 Adherence to standards

Although this effort is in the early stages, efforts from a community of oracle builders have been made to promote best practices, collaboration and interoperability by putting forward standards through EIP-2362.

ttps://github.com/adoracles/EIPs/blob/851003d89e255492aa6c3cabb3dc8520f2e71d45/EIPS/eip-2362.md

Best practices will be found here:

ttps://github.com/adoracles/OracleBestPractices

3.4 Aggregation and validation of oracle results

3.4.1 Input Data types

As mentionned in section 3.1, the system should provide for data types other than numerical. The basic needs for running the iteration of this paper are

- **Numeric**
  The type specified in EIP-2362 is uint256
- **Small string** Initially 256 Bytes
- **Light bytecode** Initially 256 Bytes

Data is kept small in size for the onchain buffer for storage and computation considerations. Limitations may be disregarded for the buffers residing on the OCW. The data must be conform to the format specified in the SLA.

3.4.2 Output Data types

EIP-2362 contains specification for return in numerical values as:
returns(int256 value,uint256 timestamp,uint256 statusCode);
In the case of as string in the rust environment the output will be of type:
let _: [&str; 256]

3.4.3 FIFO / circular buffer sampling for numeric type

Aggregation buffers in the form of a FIFO queue is the traditional data structure used to evaluate data coming from multiple sources. Calculations of one sample of the buffer per block can give various results according to the algorithm that is used. In any case, the user should be able to configure which algorithms he/she wants to use and this algorithm should be specified in the config/SLA. Along with the type of algorithm used are the data structures parameters and rules applying to the reporters. Factors which will govern our research for the aggregation buffers:

- Various inputs: push and OCW pull reporter values
- Variable queue size
- Random sample rate
- Weighted/Moving average
  https://arxiv.org/pdf/2106.09349.pdf https://arxiv.org/pdf/2004.07140.pdf

- Mean
- Median
- Median filters(Neighboring pixel value)
- Quantization
- Time weighted median price
- Median Absolute Deviation
- Accepted same value over 2/3 of the buffer.
  [7]
- Ban period If value of OCW or push reporter is over or below the value of an accepted spread
- No 2 OCW or pull based reporter should be allowed submit twice in same sample period.
- Select storage [12]
- Binary/Quick sort for median
- Time weighed average price

3.4.4 Multiple oracles

The oracle system will use multiple buffers. These are dynamic or static and resid on the OCW or onchain. The buffers on the OCWs will be collecting data from various APIs. A manifest or snapshot of the system components can be created and sent to the user in time for the next block into the the main Kylin runtime Oracle.

Kylin Runtime Oracle This oracle is centered on security and democratisation. Mirroring attacks are when the data feeders are willing to freeload another data feeder's response to minimize their cost of data provision. The first step to address these attacks is to clearly identify the chain of reporting in the SLA, which is in turn signed by the querier. Furthermore, mechanisms should be put in place that ensure the confidentiality of the data sent by the data feeders. Confidentiality is achieved by a commitment scheme as each data feeder sends a commitment of the plain data as an encrypted message to the receiver (Kylin Oracle Runtime Buffer). Later, the sender can reveal the original plain data and verify its authenticity using the commitment sheme.

Kylin OCW pull Oracle The offchain workers are part of the kylin collator without being part of the runtime. They have direct access to storage. They can run intensive and demanding computations without affecting the runtime. Their work , verified by ZK proof, can either be used right away(speed) or sent to the Kylin runtime oracle buffer for subsequent treatment(security) as specified for the consumer in the SLA.

Kylin push reporters oracle TBD

3.5 Create a pool of APIs for inbound pull of data feed

As mentionned in section 3.1, it is highly recommended to aggregate the results given by multiple APIs in order to get the most accurate data. When a smart contract requests an off-chain state/data, the pull-based inbound oracle receives the request from the caller smart contract, gathers the state from off-chain components, and sends the state back to the caller using a transaction. This process is transparent as it is implemented using smart contracts that communicate via delegated calls. A pull-based inbound oracle's performance is constrained by the transaction rate of the blockchain network and the time needed to collect external data once the request is made to the oracle. Response time depends on the network speeds, can causes a bottleneck. *https://arxiv.org/pdf/2106.09349.pdf* This can be mediated by the immediate access to the runtime storage by the OCW.

3.6 Create extrinsic and pool for inbound push of data feed

A push-based inbound oracle uses off-chain components to monitor external state/data changes and proactively injects any updates into the blockchain. This enhances the performance as the calling entity does not need to wait for the oracle to collect data. In traditional systems, if the use case needs to resolve temporal conflicts, the history oracle pattern could be applied to complement push-based inbound oracle to provide historical values of off-chain data. In these patterns, how the data came into existence is invisible this cannot be mediated but because of a manifest generated by the system can provided to the user and because of the rules of a strict and discriminatory buffer queue, the effects of data manipulation will be lessened. In the case of inbound push-based reporters, participants could need to stake the amount of the transaction fee several times before gaining rewards obtaine in fees from the buffer snapshot within that timeframe. This can serve as protection against DDOS attacks and also prevent collusion because the cost of manipulation is greater than the rewards obtained.

3.6.1 Zero Knowledge proofs

The calculations on the OCW occur offchain and will need to be verified. This is a perfect usecase for zero knowledge proofs. ZKP is method by which one party

(the prover) can prove to another party (the verifier) that a given statement is true while the prover avoids conveying any additional information apart from the fact that the statement is indeed true, in this case that the operation was done according to specific conditions.

3.6.2 Filtering arbitration and triggering

Data should be vetted before it reaches the blockchain for completely irrelevant/illegal data. A keeper or arbitrator can initially filter the data for integrity and pertinence while watching the state in order to trigger pertinent actions on sybil other blockchains. [8] This entity can also act as an *automated custodian* that can allow for the outsourcing of regular maintenance tasks in a trust minimized and decentralized manner.

3.7 Use of Democracy pallet

Blockchain oracles are dependant on effective DAO governance. All instances will be created through the Kylin democratic process. Following the procedure, the community will propose, disscuss, endorse and submit questions/operations to be put through a vote by referendum. The system depends mainly on this process for:

- Creation and modification of feeds
- Creation and modification of reporters
- Creation and modification of curators
- Creation and modification of arbitrators

but also for interacting with or modifying the system with the usual commands which previously required sudo.

3.7.1 Interface

- **propose**
  - Submits a sensitive action, represented as a hash. Requires a deposit.
- **second**
  - Signals agreement with a proposal, moves it higher on the proposal queue, and requires a matching deposit to the original.
- **vote**
  - Votes in a referendum, either the vote is "Aye" to enact the proposal or "Nay" to keep the status quo.
- **unvote**
  - Cancel a previous vote, this must be done by the voter before the vote ends.

- **delegate**
  - Delegates the voting power (tokens * conviction) to another account.
- **undelegate**
  - Stops the delegation of voting power to another account.

3.7.2 Example uses of current democracy pallet

1. **Register data feed**
   (a) **Submit as proposal**
   (b) **Obtain community endorsements**
   After proposal period ends, data feed (Dynamic NFT) is created and is put up for sale on a decentralized data market. **Possibility of this step financing the vote. i.e. Anyone statisfying criteria for vote could actually get paid to vote. See democracy pallet enhancements.*
   (c) **Select APIs**
   (d) **Select OCW aggregation buffer algorithm**
   (e) **Select reporters aggregation buffer algorithm**
   (f) **Select Kylin Oracle runtime aggregation buffer algorithm**
   (g) **Create Service level agreement**
   Include APIs and algorithms, test runs and data shema validation as part of this step
   (h) **Community vote on data feed creation** Publication of config upon succeful vote. Users can select config for their feed according to their preferences.
   (i) **User/dapps request to query feed**
   (j) **Users sign SLA agreeing to configuration, terms and conditions**
   (k) **Possibility to delete data Feed**
   Service level agreement resiliation
2. **Register API**
   (a) **Submitted as proposal**
   (b) **Obtain community endorsements**
   (c) **Create Service level agreement**
   (d) **Vote on API provider** Vote is based on reputation or any other pertinent info.
   (e) **API inserts terms, conditions and liabilities in SLA to be submitted to consumer**

3. **Register keeper/arbitrator**

   (a) **Submitted as proposal**

   (b) **Obtain community endorsements**

   (c) **Create Service level agreement**

   (d) **Vote on keeper/arbitrator** Vote is based on reputation or any other pertinent info.

   (e) **Keeper sign SLA agreement with the system, terms and conditions**

   (f) **Permissions**

      i. Blacklisting reporters

      ii. Emergency stop of feed

3.8 Enhancements to the democracy pallet

3.8.1 Substrate governance v2

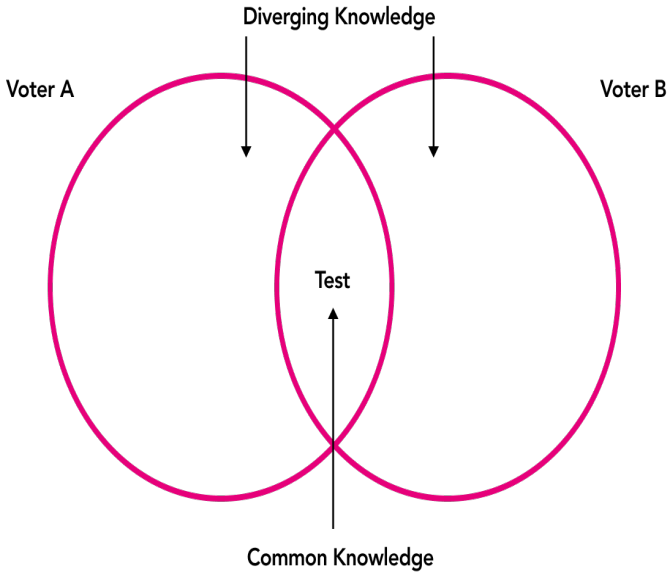
Being developed and Kusama only at this time.

3.8.2 Augmented democracy

Human oracles and dispute resolution Individuals with specialized knowledge/skills in a particular field can also serve as oracles. They can research and verify the authenticity of information from various sources and translate that information to blockchain rules. Since human oracles can verify their identity using cryptography, the possibility of a fraudster faking their identity and providing corrupted data is relatively very low. In Voting-based strategy arise issues for incentivized platform. There is a term called lazy equilibrium- a form of verifier's dilemma- in which voters always return the same answer to questions to secure profits without performing works for correctness. This is one scenario that could be addressed by a mechanism, ensuring that the body of voters agree on a set of facts before the vote, thus cerifying that the problem does not reside in the nature of the vote but in the bad will of some of the voters. The community could then propose to redo the vote or cast out bad actors. This concept is suggested as an enhancement to the voting process and not a solution. When used in combination with other methods, a clean vote could eventually be distilled, in the same way that renewable clean energy has to be gathered from different sources. [5]

Augmented voting Augmented voting is a vote where the particpants are required to take a test to enable their right to vote on a particular question/issue/dispute. This concepts seeks to abstract the diverging beliefs in a group of voters in favor of outlining their common credence.

- Diminishes amount of diverging knowledge
- Educates voters
- Promotes votes for the truth and common good
- Helps determine if the outcome of a vote is emotional or irrational
- Solidifies the value of the outcome of a vote
- Promotes unity



How do we ensure the tests to be accurate and based on facts? The solution lies in a modified application of token curated registries. [2] The Kilt protocol (Polkadot-network) has built a succeful approach that bonifies this method and brings it even closer to our usecase, token curated attesters. For this to work, the questions and information gathered have to abide by the following criteria:

- There is an objective answer to the particular question.
- The answer is publicly observable.
- The answer is very cheap to observe.

From the kilt protocol white paper: *The expert must be careful in her decisions since doing a bad job results in being dismissed (expelled from the list by the curators), damaging her reputation, and finally losing her income. Therefore, this system disincentivises bribed or bad decisions.* [1].. In the same frame of

mind we want to adopt this method to validate the pertinence and authenticity of the questions for the test that will enable community membres to vote.

Token curated tests grids Token curated test grids are decentrally-curated grids with intrinsic economic incentives for token holders to curate contents judicioulsy. Again from Kilt: *Token holders have a tactical incentive to challenge and reject every candidate to their registry in the interest of increasing their holdings, but this is at odds with their strategic interest of increasing the value of their holdings. An empty list is of no interest to consumers, so candidates would not bother applying to it. Candidates drive fundamental demand for a registry's intrinsic token and so by behaving tactically rather than strategically, token holders go against their own interest and incur a potentially severe financial loss.* Generally, it is in the interest of economically rational token holders to behave strategically and curate a high quality list.

The augemented democracy protocol suggests using the same tripartite proving structure as Kilt: Verifier, claimer and attester, but the later one being responsible for curating the list of contributers to the test grid: gatherers and curators. Gatherers simply gather facts about the question and curators conveniently arrange the information into the simplest test possible.

Assuming that this a good way for tests to be based on verified facts, could there be a case, in certain scenarios, where there would be voter accountability against proven facts? Could this also be a way for rendering elected officials accountable for their decisions when participating in high level votes?

3.8.3 Deliberative democracy

On the flip side, shining the light on someone's disagreement within the consensus could help the individual and the collective. In north american tribes, "Unanimity requires that everyone involved agrees." https://www.ictinc.ca/blog/what-does-traditional-consensus-decision-making-mean This more exhaustive approach could be applicable in certain cases and could be aided by using several round of augmented voting.

3.8.4 Staking and quadratic voting

The concept of *"putting your money where your mouth is"* has been used succesfully by proof of stake. Originally a good idea, whales have succesfully tampered with these networks. Whales have a proof-of-stake advantage, the more coin you have to offer, the higher the chance you have to be selected as a validator. With the existing democracy pallet, proposals are already created with a staked amount. If this could be extended to voting using this same formulae cost to the voter = (number of votes)^2 To even out the voting power. The cost of each vote to a single project from a single contributor will increase, encouraging community contributors to donate to more projects. *https://vitalik.ca/general/2019/12/07/quadratic.html*

3.8.5 customizable decentralized cryptoeconomic incentives

3.8.6 Node Reputation / Performance History

Leveraging substrate reputation mechanism for feeding Token curated test-grids. WIP [11]

3.9 Datafeed as a Dynamic NFT

The main difference between a static and dynamic NFT is the ability of the token's metadata to change with time. Subsequent to a favorable vote from the procedure enforced by democracy pallet, a dynamic nft is minted and its ownership is recorded on the blockchain. The data and its characteristics(metadata) are recorded in an ipfs partition created amongst kylin-collator ocws or parachain coalition.

3.9.1 Metadata

This is information that will be stored on each api's database or other persistent storage such as IPFS. It consists of the data itself, if is too large to reside on chain, and information regarding the data, metadata. Initially, at creation of the NFT, the service level agreement or terms and conditions waiver, should be stored there in the form of a Ricardian contract along with the expected data schema. The metadata from a data feed is constantly updated by the Kylin Data Oracle.

3.9.2 Semi Fungibility/ SFT

One of the token's metadata can serve as a holder of value, commonly labeled as shares. These shares can be bought or given as a incentive for work which needs to be done to satisfy the wellness of the feed. With every sale of the token a portion would go to paying out royalties on these shares.

Dynamic Data-Driven Strategy By assigning our data stream to a dynamic NFT's metadata, the data tied to this NFT can be managed and sold to the highest bidder. The price and the nature of the data offering should be driven by the need of this data from users and the value it generates. The system provides that if the data should need to be verified and curated by a comittee, the NFTvalue should serve to cover these costs. The data's ownership should then be constantly revised by its owners for profit or for preserving the integrity of the data itself.

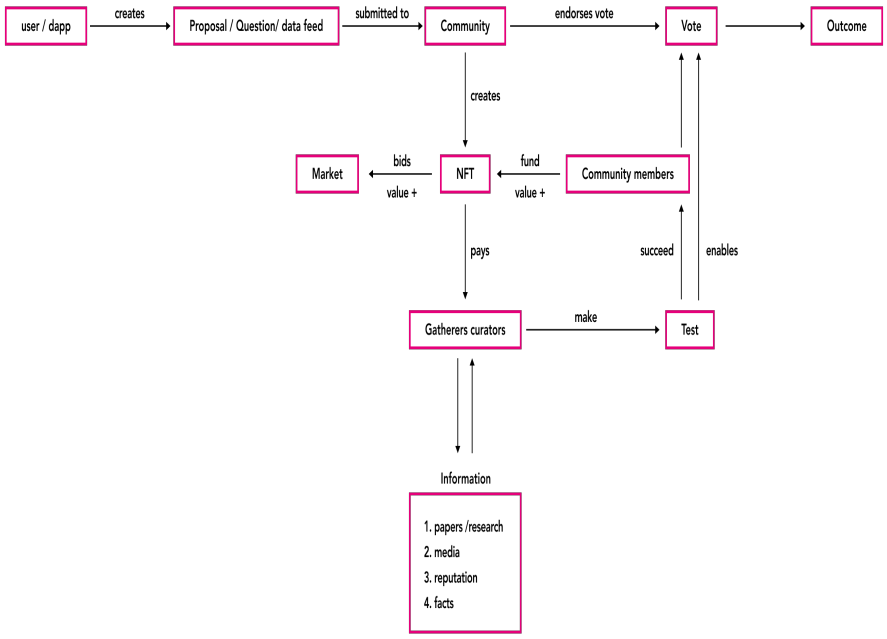
3.9.3 DeData Market

NFTs offered on the DeData market can serve for "boostrapping" data feeds. NFTs offered by the DeData market are such that *real* finanacial value will

be attributed and dynamically be modified according to real life results, rendering the appeal for data NFTs greater than speculative and empty (*"bored ape"*)NFTs which are more likely to be subject to the greater fool theory.

NFT Market making

- The system can offer to buy successful and sustainable feeds at market cost.
- The system can offer to buy successful questions that have accomplished a "common good" purpose.



3.10 Interchain operation

3.10.1

3.10.2

3.11 Security analysis

The following are security considerations which will have to be considered as we build the Kylin oracle system:

- **Goldfinger attack**
  Decentralized selection through Voting. The kylin Oracle system registration process aims to democratically register entities. By decentralizing the

selection process, the goal is to distribute the trust from a single entity to a collective decentralized governance. Voting distributes trust and provides a degree of robustness against entities failing to participate, however it adds latency and introduces the threat that an actor can accumulate voting rights to sway the vote. [10]

- **Sybil attack/off-chain freeloading**
  Currently, it is not possible to determine who is operating an oracle. A single individual could then operate multiple oracles without anyone else's knowledge. Multiple nodes run by a single individual can share API answers off-chain. By serving a request with two of their oracles, an attacker can halve their API call costs and be able to underbid the market while degrading the security of the aggregated answer. [3]..

  Types of Sybil attacks:

  1. Oracle botnets
     Node operators are going to be prime targets for hackers. OPSEC (operations security) is a security and risk management process and strategy that classifies information, then determines what is required to protect sensitive information and prevent it from getting into the wrong hands. A node set up and operated with poor OPSEC practices is a Sybil attack risk.
  2. Node-as-a-service (NaaS)
     NaaS is a business model where a third party sets up/configures/manages a node operator's node. The client here is typically a LINK investor or an API provider who does not want to run an independent node. Since the NaaS provider will likely be serving multiple node operators, this constitutes a Sybil attack risk.
  3. Smurfing
     Associating each oracle with a real identity is not enough to eliminate the Sybil attack risk. The attacker may set up multiple oracles to control themselves and have others get their identities associated with these oracles.

- **Insertion attack**
  An attacker will observe an unconfirmed oracle transaction in the mempool, craft a transaction that profits from knowing what the oracle data will be, and attempt to have this transaction confirmed before the oracle transaction itself.

- **Mirorring attack/freeloading**
  There is a risk on the oracle system that a data feeder uses another data feeder's information to self-report to the system. This a form of collusion.

  The data feeders are willing to freeload another data feeder's response to minimize their cost of data provision. They will also be confident that their data will not be an outlier and be penalized.

- Front-running
  Formalized security around cost of corrupting oracle and oracle extractable value. Solution-decentralized network of oracle nodes are used to order transactions on a per-contract basis using a custom policy, such as ordering transactions by the relative time each hits the mempool. The core advantage of FSS is that not only does it prevent miners and arbitrage bots from manipulating transaction orderings, but it also can be implemented in a backwards-compatible manner, requiring no changes to the layer 1 blockchain. Additionally, it can be implemented in different ways such as monitoring transactions in the blockchain mempool itself (meaning user experience would be the same as today) or allowing users to send their transaction directly to the oracle network, providing a meta-transaction type service which can additionally save on gas costs by batching transactions. Such a solution can also be used to ensure oracle updates are prioritized before any user trades take place, preventing oracle frontrunning all together. This implementation demonstrates how oracles go beyond secure data delivery and ensure the fairness of the application itself. [6]

- Sandwich transctions
  https://repository.tudelft.nl/islandora/object/uuid:06ddf1dd-7dba-495b-b7f3-31b8d2bb966c/datastream/OBJ/download

- Lack of node diversity(software and hardware)

- Lack of source authenitcity

- Use of spot prices

- Lack of code reviews, penetration tests

- Lack of standards

- KYC for all instances of the Kylin Oracle system/network

3.12 Solutions for mitigating security risks

1. **Random selection within the data feeders**

2. **Real identities**
   A strong registration process and quadratic voting scheme can dissuade the accumulation of disproportional voting rights. This is unfortunately done at the expense of privacy but protocols suck as Kilt could help find a middle ground solution.

3. **Arbitrators** [8] These entities are constanatly observing the system and can blacklist wrong doers.

4. **Incentive for calling out bad reporters**

5. **Mirroring attacks**
   The oracle designer should consider mechanisms that ensure the confidentiality of the data sent by the data feeders. A popular technique to achieve confidentiality is to use a commitment scheme In a commitment scheme, each data feeder should send a commitment of the plain data as an encrypted message to the receiver. Later, the sender can reveal the original plain data and verify its authenticity using the commitment

6. **Avoid a public mempool**

4 Tokenomics

Token: $KYL

- **Max Suppy:** $KYL1,000,000,000
- **Type:** Utility-token
- **Token standard**: ERC20

Token: $PCHU

- **Max Suppy** $PCHU 100,000,000
- **Crowdloan(s)** — 40%
- **Parachain crowdloan stimulus** 10%
- **PLO Reserve (Used for future auctions)** 30%
- **Ecosystem** — 60%
- **Community rewards** 9%
- **Lock drop rewards** 5%
- **Holders reward** 6%
- **Marketing fund (Promotions and educational content)** 20%
- **Liquidity fund (Liquidity for $KSM swaps)** 10%
- **Developer fund (Infrastructure costs and validator rewards)** 10%

4.1 Pichiu Airdrop

Pichiu is to be released on the Kusama Network at block 13,219,200. After this event an undisclosed amount of snapshots will be taken of $KYL token holders.

4.1.1 To be eligible for the $PCHU airdrop

To host the $PCHU airdrop the team will build a dApp that $KYL token holders can use to link their ERC20 wallet in which they store their $KYL tokens to a native Polkadot js wallet. When we join those two datasets it will identify which polkadot js holders are entitled to the $PCHU airdrop. This means only people who link their ERC20 wallet via the dApp to a native Polkadot js wallet will be eligible for the airdrop.

4.1.2 Holder Rewards

Total rewards are 6%.
1% $PCHU will airdrop to $KYL holders immediately and the remaining tokens will be distributed each month (0.5% per month)
Based on holder rewards of 6%, 6m / 173m equates to 0.034 $PCHU per KYL
The values will be different for lock drop rewards as it is 5% instead of 6%

4.1.3 Lock Drop Rewards

Total lock drop rewards amounts to 5%.
0.5% $PCHU will airdrop to $KYL holders who lock their $KYL per month by staking This will last 10 months. All airdrop rewards are given to Kylin addresses who are hodling their $KYL tokens in non-custodial wallets or staking from the time of the snapshots.

4.2 Fees WIP

- Queries from dapps
- Queries from other polkadot parachains
- Queries from other chains
- Dispute resolution
- Elections
- Enquiries

4.3 Staking

In order to secure their position in the Kylin ecosystem, entities will have to comply with the SLA requirements but also stake an amount that can be slashed in case of bad behavior.

- Push based oracles

- Pull based OCW
- Pull based API
- Collator node
- Parachain node

4.4 Data market/NFT/SNFT

- Bootstraping data feeds/ questions/ dispute resolution
- Financing(crowd source) feeds/ questions/ dispute resolution through SFT
- Decentralized Data Market Offering

4.5 Rewards

- Push based oracles KYC, the use of random and decentralized fair sequencing will act as a barrier for attacks and allow reporters to earn rewards equitably. The rporters will have to pay a transaction fee for their chance to be included in the aggregation buffer. If they do and if this contribution is used in the poverall process of answering a paid query, a perecentage of the fees will go to this reporter as well as a reimbursement for all previous transaction fees incured.
- Pull based OCW Because of they are integrated into a collator, OCW benefit from guaranteed access to the Kylin Oracle runtime aggregation buffer
- Pull based API
- Collator node
- Parachain node
- Attesters
- Curators
- Gatherers

References

[1] Kilt whitepaper. BOTLabs GmbH(Berlin, Germany), January 2020.

[2] Aditya Asgaonkar and Bhaskar Krishnamachari. Token curated registries - a game theoretic approach. Center for Cyber-Physical Systems and the Internet of Things(Viterbi School of Engineering, University of Southern California), September 2018.

[3] Young Choon Lee2 Amirmohammad Pasdar, Zhongli Dong. Blockchain oracle design patterns. June 2021.

[4] Abdeljalil Beniiche. A study of blockchain oracles. INRS(Montreal, Qc), July 2020.

[5] Sylvain Cormier. A machine based societal model for curbing citizen cynicism. https://github.com/sylvaincormier/augmented-democracy/blob/master/doc/augmented-democracy.pdf, August 2017.

[6] Kevin Tjiam Kaitai Liang. Investigating arbitrageurs and oracle manipulators in ethereum. Delft University of Technology, 2021.

[7] Jae Kwon. Tendermint: Consensus without mining. https://v1.cosmos.network/resources/whitepaper, 2014.

[8] Alex Coventry Steve Ellis Ari Juels Benedict Chan Farinaz Koushanfar Daniel Moroz Florian Tram Andrew Miller Sergey Nazarov Brendan Magauran Alexandru Topliceanu Fan Zhang Lorenz Breidenbach, Christian Cachin. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. Chainlink Labs, April 2021.

[9] Mehdi Salehi. An analysis of upgradeability, oracles, and stablecoins in the ethereum blockchain. The Concordia Institute for Information Systems Engineering(Concordia University Montréal, Québec, Canada).

[10] Wanyun Catherine Gu Shayan Eskandari, Mehdi Salehi and Jeremy Clark. Sok: Oracles from the ground truth to market manipulation. Concordia University(Montreal, Qc), September 2021.

[11] Dr. Gavin Wood. Polkadot: Vision for a heterogeneous multi-chain framework. https://www.parity.io/(Berlin, Germany).

[12] Yijing Lin Lanlan Rui Yang Yang Chen Zhao Zijia Mo Zhipeng Gao1, Zijian Zhuang. Select-storage: A new oracle design pattern on blockchain. State Key Laboratory of Networking and Switching Technology(Beijing University of Posts and Telecommunications, China).