Pythia: a decentralized expert validation platform for crypto

Anton Kravchenko and Nikita Kravchenko

(Dated: Monday 20th February, 2023)

Pythia is a decentralized reputation platform that leverages prediction markets and cryptographic primitives to establish a reputation standard for crypto content creators. Pythia's markets are questions covering key industry trends and events, e.g., token price movements, protocol economic model flaws, presence of bugs in a protocol's codebase. Anyone can create a market and make predictions in it. Users predict anonymously and without having to put their money on the line. Pythia's markets are resolved in a decentralized fashion using Chainlink [6] oracles and Kleros [5] arbitration courts/optimistic oracles. After resolution, the user's predictions are revealed. Users who have made correct predictions are rewarded with non-transferable Reputation Token (RT), which tokenize their on-chain prediction track record and which they then can share on social media platforms (Twitter, Lens Ecosystem apps, LinkedIn, Telegram etc.).

I. INTRODUCTION

The lack of a reliable method to identify expertise is stumbling block for the crypto industry. In contrast to traditional finance or Web2, crypto does not and cannot have a centralized authority (extensions of the state such as the SEC in the US) to regulate the ecosystem and identify who to trust. As things stands, this power vacuum leaves the space vulnerable to scammers and grifters. And this is a problem that the crypto ecosystem has to solve if it is to on-board the next billion users in the coming years and become a real alternative to TradFi and Web2 more broadly. Incoming and existing crypto users deserve a standard that would easily allow them to identify who to trust and listen to rather.

Yet, for all the criticism surrounding it, the crypto industry is overflowing with talent. There are thousands of quality crypto content content creators, a lot of them quite small. Yet, their voice is drowned out by a few unqualified crypto celebrities and shills, since the former often lack huge marketing budgets or the necessary promotion skills. The status quo is hugely detrimental to the space undermining the industry's credibility and alienating potential users.

To address these implications, we need to ensure that the best people in the space get the most attention. In other words, there needs to be a fair and reliable way to validate crypto expertise.

A solution that has been touted for a long time now are decentralized prediction markets. Vitalik Buterin [1] has long been advocating them as a reputation-measuring mechanism for all kinds of events. Prediction markets quite naturally lend themselves to determining expertise, because they force people to make concrete directional statements about future outcomes, which is the most natural way to test the validity of someone's hypotheses in the absence of formal proofs. Hence why making predictions is foundational to the modern scientific method. Building decentralized prediction markets, meanwhile, enables people to avoid the common biases such as how to determine the correct answer and, more pertinently, allows anyone to see the full history of a person's predictions.

Consequently, a number of such applications cropped up in recent years including Augur[2], Polymarket[3], Gnosis's Olympia [4]. However, all these applications have suffered from three major drawbacks: (1) they require users to bet money, often in an obscure native token, exposes users to unlimited downside and substandard execution (due to the absence of liquidity resulting from the high cost of liquidity in DeFi) and hence only serves to attract gamblers (2) they are not crypto-focused and hence do not address the issue at hand (3) they lack integrations with the rest of the ecosystem and social media, essential for them to have the necessary impact.

In this paper we propose Pythia, a platform for expert verification in crypto that does not require users to bet money, is positive-sum in nature, crypto-focused, and tightly integrated into the surrounding ecosystem with with the help of its custom soulbound tokens (SBTs).

II. HIGH LEVEL OVERVIEW

A. What is Pythia?

Pythia is a decentralized reputation platform for crypto content creators powered by anonymous prediction markets. Pythia's users verify their expertise by making predictions on crypto-related topics, which are represented by markets.

B. Pythia's markets

Pythia's markets are questions that users have to answer to validate their crypto expertise. Pythia's markets are defined by three phases: 1) creation, 2) prediction, and 3) resolution. Following the resolution phase players who made correct predictions receive a certain amount of Reputation Token. See subsection IV G for more detail.

C. Market creation

Market creation likewise entails three distinct steps. First, a user has to create a question by filling out a special form. Second, the user has to pick the possible outcomes for the markets. Lastly, the user must submit the market on-chain. In essence, this means deploying a new contract. Crucially, at Pythia, market creation is permissionless meaning it does not require us or any other third party to verify the question before it is officially allowed on the platform. The only notable limitation of the questions Pythia's users can ask is the form itself. The form ensures that all questions are asked in an understandable format and are appropriate thematically.

D. Making prediction

The prediction phase is fully anonymous both for the Pythia team and other users. The prediction process consists of two steps. First, a user creates their prediction. They then hash their account address in their browser (this is all done automatically, the user does not have to perform any manual actions). They then forward us their hash and the prediction. In the final step, Pythia relays this information on-chain. Note that we still record the user's answer in our smart contract which enables us to calculate the metrics we need to determine the distribution of Reputation Tokens. However, no one knows who submitted that prediction because of the hashing.

E. Market resolution and pay off

Markets are resolved in a decentralized fashion using either Chainlink oracles or Kleros courts. Once the market has been resolved, a user can receive Reputation Tokens by revealing their prediction. The token amount the user receives depends on 1) difficulty (final market odds), 2) popularity (the number of people that participated in the market), and 3) how early prediction was made.

F. Social media integrations

Pythia's users can display their Pythia profile as a link on social media. The profile will contain information about the markets that a user has participated in and, crucially, their Reputation Token balance. Note that it is fairly easy to prove that a given user owns a given profile. As a profile is simply an NFT, one can provide a digital signature to verify their ownership.

III. PRELIMINARIES

Before we delve into a detailed overview of Pythia, let us first briefly discuss the important cryptographic primitives used by Pythia.

Definition 1 (Cryptographic hash function)**.** [7] A cryptographic hash function (CHF) is a hash algorithm (an algorithm mapping a map of an arbitrary binary string to a binary string with fixed size of n bits) that satisfies the following properties:

1. The probability of a particular n-bit output result (hash value) for a random input string (”message”) is 2^{-n}, so the hash value can be used as a representation of the message;

2. Restoring a message from its hash, generated by CHF should be computationally infeasible

3. Finding any pair of different messages that yield the same hash value (also known as a collision) is infeasible

Example 1. Examples of existing CHF include $SHA-0$, $SHA-1$, $SHA-2$, and $SHA-3$ $(Keccak)$ with which the reader might be familiar with their use cases in Etherium.

Definition 2 (Public key cryptographic system)**.** [8] A public key cryptographic system is a cryptographic system that uses pairs of related keys. The key pair consists of a public key and a corresponding private key. The key pair is generated using cryptographic algorithms based on mathematical problems, known as one-way functions. The public key is used for data encoding, while the private key is used for decoding data. The keys should satisfy the following properties

1. It should be infeasible to decode data without knowing the private key

2. It should be infeasible to determine the private key by possessing only the corresponding public key

Example 2. One algorithm used for key generation is RSA which utilises Euler's totient function and modular arithmetic.

Definition 3 (Digital signature)**.** A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents. It usually involves three steps:

1. key generation: the cryptographic algorithm generations a pair of public and private keys **pubKey**, **privKey**

2. signature: the signature algorithm creates a signature **s** and a message **m** with **privKey**

3. verification: anyone can verify the signature using **pubKey**, **m**, and **s**

The digital signature scheme should satisfy the following properties

1. It should be infeasible to restore the **privKey** given the signature

2. It should be infeasible to recreate a signature without knowing **privKey**

Example 3. Etherium uses the ECDSA [9] digital signature scheme to prove account ownership.

The most basic example of a zero-knowledge proof is a digital signature.

Definition 4 (Digital (Cryptographic) signature)**.** A digital signature is a mathematical scheme for verifying the authenticity of digital messages and documents. A valid digital signature, where the prerequisites are satisfied, gives the recipient very high confidence that the message was created by a known sender (authenticity), and that the message was not altered in transit (integrity). A digital signature usually consists of three steps

1. Generate a pair of keys **privateKey**, **publicKey** using a cryptographic algorithm G

2. Sign a message m with **privateKey** using an algorithm S

3. Verify the signature s with **publicKey** and the message m

IV. PYTHIA'S DECENTRALIZED APPLICATION

Pythia is a decentralised reputation platform for crypto content creators currently running on Polygon. It uses public key cryptography to enable anonymous prediction markets. In the following section, we shall present the main components of the application, i.e. user accounts, markets, Reputation Tokens from a technical point of view.

A. Pythia's contracts

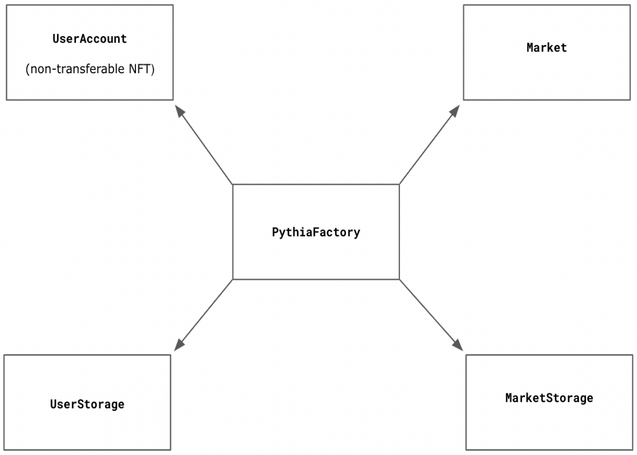

Figure 1. Pythia contracts

Pythia's application consists of five contracts **PythiaFactory**, **UserAccount**, **UserStorage**, **Market**, **MarketStorage**. **PythiaFactory** is a factory contract storing addresses of the deployed **MarketStorage** and **UserStorage** contracts. The latter two, in turn, store the created markets and registered users respectively.

B. User account

1. Creating an account

To create an account on Pythia, a user needs to have a a blockchain account (meaning have a key pair). From the user's perspective creating an account involves two steps

1. Connect a wallet

2. Create a username

After these actions have been executed, a new instance of **UserAccount**(non-transferable NFT) is deployed. The address of the deployed account is stored in **UserStorage** contract.

2. Account properties

The contract **UserAccount** representing the user's account has the following properties

Property	Type	Description
owner	address	address of the user
ownerUsername	string	username
markets	mapping	mapping storing markets in which user participated
isSubscribed	bool	flag indicated if user has subscription, see section IV H

Table I. User profile parameters

C. Market life cycle

Pythia's markets are questions covering various crypto topics. An example of a market could a statement as simple as: "Will the price of Bitcoin exceed $30,000 by the end of March 2023." All markets exist on-chain as instances of the **Market** contract. Pythia's market life cycle consists of three main phases: **creation**, **prediction**, **resolution**.

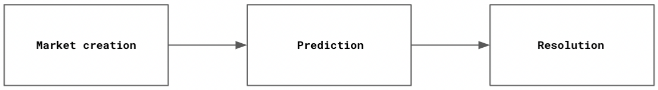

Figure 2. Market life cycle

D. Creating a market

At the onset, Pythia plans to support markets covering the following themes:

1. **price movements** - e.g., will the price of the pair exceed a certain threshold by a given date
2. **technical architecture** - e.g., will a certain protocol get hacked by this date
3. **economic models** - e.g., will a stable be able to maintain its peg by this date

Creating a market involves filling out a template form. The form includes the following fields: **question**, **predDeadline**, the last time people can predict for this market, and **resolutionDate**, the date when the market should be resolved. The questions that are allowed on Pythia should satisfy the rough template:

Will {event} happen before {timestamp}

After the form has been completed, a new instance of the market resolver contract is deployed. Depending on the market type it can be either a Chainlink priceFeeder or a custom Kleros contract. Shortly after that, a new instance of the **Market** contract is deployed. The address of this new market instance is stored in the **MarketStorage** contract. **Market** contract has the following properties

Field	Type	Description
question	string	question created by filling in the template
predDeadline	timestamp	the last date at which users can make predictions
resolutionDate	timestamp	date at which resolution process should start
creator	address	address of user account which created market
createData	timestamp	date when market was created
topic	int	topic identifier (price feeds or protocol hacks or economic model)
arbitrator	address	address of market resolver contract (Kleros contract or Chainlink contract)

Table II. Market parameters

1. Community moderation of Markets through DAO

The goal of Pythia is to establish a standard for crypto expertise, hence, trivial markets would not work. It is better to have a few quality markets a week then a litany of trivial markets. Thus, we plan to have Pythia's to moderate market creation, initially through a Discord channel and eventually via a DAO. In essence this means that when any user creates a market, it has to be validated by us/ community of Pythia's users.

E. Prediction

All user predictions in Pythia's markets are fully anonymous. This means that users hide their predictions while keeping their identity (address) open. At resolution phase in order to receive reward one needs to disclose his prediction.

1. Hiding user identity

The most trivial way to hide information using cryptography is hashing. So the first thing that comes to mind is hashing the user's prediction i.e. choice. These, however, leads to multiple addresses with same prediction to have the same hash which is not safe. Another option is generating a unique random nonce for every user. This though viable is not user friendly and requires users to store a list of random numbers if they participate in multiple markets. Thus we can formulate the following 3 criteria for prediction hiding scheme:

1. The scheme should be deterministic i.e. easily reproducible by user
2. It should be infeasible to disclose the prediction without some private information of which only the user possesses
3. At any point of time, a user should be able to proof that they have made a certain prediction without revealing any of their private information.

At Pythia we utilise the following scheme satisfying the above properties

$$predictionHash = keccak(ECDSA(marketID||prediction||userAddress)) \tag{1}$$

It is worth noting that we use a deterministic seed for the ECDSA algorithm here

$$k = keccak(privKey) \tag{2}$$

The user can disclose and prove their prediction by providing their public key, ECDSA signature, and market id along with their prediction. Others can verify the signature using this information.

Remark 1. *Note that in order for the scheme to be anonymous marketID has to be a verifiable random number.*

2. Making an off-chain prediction

The user hashes their account by using the algorithm the 1 and selects one of the market outcomes. All calculations happen on the user's device. The user's then sends the hash to our smart contract.

Field	Type	Description
userHash	int	hash of the user
marketId	int	id of the market
outcome	int	selected outcome
timestamp	timestamp	timestamp of prediction

Table III. prediction parameters

3. Relaying prediction on-chain

Remark 2. *Notice that during the prediction phase Pythia and other users do not know the identity of the user. Pythia only deals with the hash of the user's account and the prediction itself.*

4. On double spending

Pythia does not allow to make prediction multiple times for the same market. By default, the current system prevents double spending since the hash generated for user is deterministic. Thus it can be easily checked that the user has predicted previously.

F. Resolution

Pythia markets can cover a variety of crypto topics, as mentioned in the section IV D. Markets related to price movements are resolved with the help Chainlink price feeds. The remaining market types use Kleros' arbitration courts for resolution. In this section we will cover how each of these resolution methods will work in Pythia.

1. Triggering resolution

Recall that each market in Pythia has a designated resolution date, set by the market creator before the market contract was deployed. The resolution phase for the market is triggered off-chain by a keeper node once the resolution date has been reached.

2. Chainlink Pricefeeds

For markets which cover price-related topics are resolved using Chainlink pricefeeds [10]. A system or oracles gets the price of the asset pair from multiple sources (exchanges, aggregators) and aggregates it on-chain. Later the aggregated price is compared to strike price to finalize resolution.

3. Kleros resolution

Markets that do not cover price movements i.e. protocol hacks, economic models, use Kleros for their resolution. First, an the question transaction is send to the reality.eth oracle [11] Arbitrator contract. Then the staking process is initiated. Any users can come and post an answer along with a bond. Other users can challenge this users by doubling the bond.

In case no one has offered a higher bond, the answer with highest bond at the moment is identified as the correct answer to the market question. Additionally, at any stage of the bonding process, users can request Kleros court resolution. However, it makes sense to do so, when the bonds become very high, as resolution through a Kleros court is expensive.

The arbitration court will consist of a number of randomly selected judges which will vote on the correct answer. In that case, the resolution happens using the so-called Schelling point concept [12].

G. Pay-off

After the market resolved the pay-off stage can be split into two parts: 1) disclosing predictions 2) receiving reward.

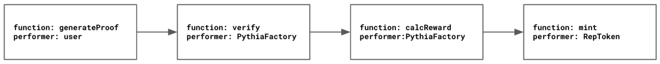

Figure 3. Pay-off life cycle

1. *Disclosing predictions*

After the market has resolved users have $24h$ to disclose their predictions. If they do not disclose their predictions they will receive 0 Reputation Tokens for this market and will be marked as losers. The main idea here is that once the market is resolved the winners are incentivised to disclose their predictions as they are likely to receive reward and vice versa.

2. *Reputation Tokens*

The number of Reputation Tokens that the user receives depends on 2 things, 1) correctness of the prediction, 2) how early the prediction was made. Define the following variables

$$\begin{aligned} &C - user\text{'}s\ prediction\ correctness\ flag \\ &P\_start - predictions\ start\ date\ for\ the\ market \\ &T - timestamp\ of\ prediction \\ &P\_end - predictions\ deadline\ for\ the\ market \end{aligned} \tag{3}$$

The amount of Reputation Tokens that user receives for the market is

$$C \times \frac{P\_end - T}{P\_end - P\_start} \tag{4}$$

Remark 3. *Notice that the formula consists of two logical parts. The first part is a flag C. When a user makes an incorrect prediction, they receive* 0 *of the Reputation Token for the market. The second part says that the earlier one makes predictions the more Reputation Token they will receive, provided they made the correct prediction.*

Remark 4. *Another important quality of the formula above is Sybil resistance, assuming, of course, that the markets created are of sufficient difficulty. If the user tries to cheat the system and create multiple accounts, the reward will be spread among all those accounts.*

Example 4. Let's say Bob has correctly predicted the price movement of Bitcoin, and he made the prediction when there was 3/4 of the time left before the prediction deadline. Then the amount of Reputation Tokens that the user is entitled to

$$1 \times 3/4 = 0.75\ \boldsymbol{Rep\_Token}$$

After the necessary reward amount in the Reputation Token has been computed, it is minted by the token contract to the Bob's NFT account.

H. Business model

Currently, all notable applications built on top of public blockchains use fee-based business models, in other words, charging users per action performed. A major reason for this is that it is very easy to implement this model on-chain. Smart contracts cannot perform actions by themselves. Usually, they need an off-chain instruction (i.e., a user calling a contract function) to perform the action. In a fee-based model, the relevant contract reacts to user actions instead of being called by a third party to collect the fee.

We believe that a subscription model is more suitable for Pythia for several reasons. Firstly, subscription-based models are inherently more sustainable, as they are less volume dependent. More importantly, they impose a hard cap on how much a user has to pay, thereby capping their potential downside. Having a per-prediction fee also does not make a lot of sense in our case since we do not require the user to bet with their money, so there would be nothing to apply the fee to.

Pythia uses the $ERC - 948$ standard to handle subscription payments. Thus, whenever a new user comes to our platform, he can create three markets and make predictions in five markets before they have to register the subscription in the **Subscription** contract. A keeper node will occasionally check user accounts and call the contract function to charge the subscription.

For our beta version, we will introduce a $7 monthly subscription which is similar to what question-and-answer platforms like Quora charge. Note that this is an nearly infinitesimal amount compared to how much people fork out to become top predictors on Polymarket and other prediction market platforms. As we transition to towards monetising content creator analytics and becoming a marketplace for experts, we will gradually reduce the subscription for predictions and eventually aim to make the cost basis equal zero for crypto content creators seeking validation.

It is also worth noting that a user can cancel their subscription at any time by calling the **cancelSubscription** function in the **Subscription** contract.

I. Social media integration

1. Linking profile

Users can share their Reputation Tokens on social media platforms like Twitter, Lens, or LinkedIn by linking their Pythia profile to them. Notably, social media users are not required to have a wallet to view the contents of the link. Once a person clicks on the link, they will see the profile with the following statistics provided

Field	**Type**	**Description**
accountAddress	address	address of the account (NFT)
ownerAddress	address	address of account's owner
balance	int	account balance of Reputation Token
balance30d	int	the amount of Reputation Token received for the last 30 days
numMarkets	int	the number of markets in which user participated in

Table IV. Shared profile properties

2. Proving ownership to outsiders

Any social media account can easily prove the ownership of a Pythia profile using a digital signature. To do so, they need to know the address of the owner of Pythia's profile NFT. For example, say Alice wants to make sure that Bob owns a certain Pythia account. To do so, she merely has ask him to sign the random message that she has chosen with his private key.

[1] Vitalik Buterin, "Tales from the Election" (2021) https://vitalik.ca/general/2021/02/18/election.html

[2] Augur: decentralized predictions marketplace, 2014, https://www.allcryptowhitepapers.com/wp-content/uploads/2018/05/Augur-white-paper.pdf

[3] Polymarket: information markets platform, 2018, https://polymarket.com/

[4] Gnosis Olympia, 2017, http://olympia.gnosis.pm

[5] Kleros:a decentralized arbitration service, 2017, https://kleros.io/whitepaper.pdf

[6] Chainlink: Blockchain Oracles for Hybrid Smart Contracts, 2017, https://research.chain.link/whitepaper-v1.pdf

[7] Aggarwal, Kirti; Verma, Harsh K. (March 19, 2015), Hash_RC6 — Variable length Hash algorithm using RC6, ieeexplore.ieee.org.

[8] Stallings, William (3 May 1990). Cryptography and Network Security: Principles and Practice. Prentice Hall. p. 165. ISBN 9780138690175.

[9] Johnson, Don; Menezes, Alfred (1999). "The Elliptic Curve Digital Signature Algorithm (ECDSA)"

[10] Chainlink pricefeeds, https://docs.chain.link/data-feeds/price-feeds/

[11] reality.eth optimistic oracle, https://reality.eth.link/

[12] Schelling, Thomas C. (1960). The strategy of conflict (First ed.). Cambridge: Harvard University Press.