Proof-of-Work Oracle

thefett
5 min read · Jul 30, 2018

by Nicholas A. Fett and Lucian Stroie

A Proof-of-Work Oracle solves the issue of how to get data onto the blockchain in a decentralized and trustless manner. This article will go over our implementation of an on-chain oracle that uses a mineable proof-of-work (PoW) competition to eliminate reliance on trusted third parties for access to off chain data.

To give a simple explanation, think mineable token, but with each solution submission, you get to put in some data (say BTC/USD price). The first n solutions are accepted and then the median is rewarded (neighboring answers get “uncle” rewards) in the form of a newly minted PoW oracle (PoWO) tokens. The median value is then timestamped and placed into a time series array which can be accessed through a getter function that charges parties a small amount of PoWO tokens.

The Oracle Problem

Smart contracts on Ethereum cannot access outside data. This means that if you have a contract that relies on data not on the Ethereum blockchain, you need to either manually enter data or rely on a third-party service. Oraclize.it[1] has been the standard, but it's a compromise away from decentralization.

Lucky for us (and the entire world), we have a way to agree on data with there being no central party….the blockchain. In the same way that Satoshi described using PoW to agree on what transactions should be included in the next block, we can use PoW to determine the next value in a time series.

Why Proof-of-Work?

I know what your thinking, "Ethereum is moving away from PoW (good!) and having another token that requires mining takes away from the hash rate of other systems and is vulnerable to 51% attacks when the hash rate is too low."

Well, yes. All valid points and they would disqualify the idea if there were better alternatives.

PoW is needed because it's a reliable way to get data into smart contracts *now*.

One of the big use cases for oracles are financial contracts such as those utilized by prediction markets. These markets, like Augur[2], are decentralized but they are not trustless. Most of these markets are using 'Proof-of-Stake'(PoS) style oracles where parties stake reputation (or Ether) on what they believe is the correct outcome. Although this setup does look promising, all research on completely trustless PoS is still being tested.

To be fair to Augur, it wasn't set up for time specific price data and that doesn't seem to be its intended purpose, but wouldn't it be useful if something else could?

How our Oracle Works

To summarize our solution, we use an expanded version of the mineable token on the Ethereum blockchain. Miners engage in a PoW competition to find a nonce which satisfies the requirement of the challenge. The miner who finds a nonce which correctly solves the PoW puzzle can now input data for the PoW Oracle contract and receive native tokens in exchange for their work. The oracle data submissions are stored in the smart contract for use by other on-chain operations. Figure 1 illustrates the flow for a token issuing oracle contract utilizing proof-of-work for data input and validation.

At the top of the diagram, a contract is created which specifies the data to be input into the contract (e.g. an API address), a difficulty (how hard our miners have to work), and a challenge (a random variable to be included in the hash of the solution).

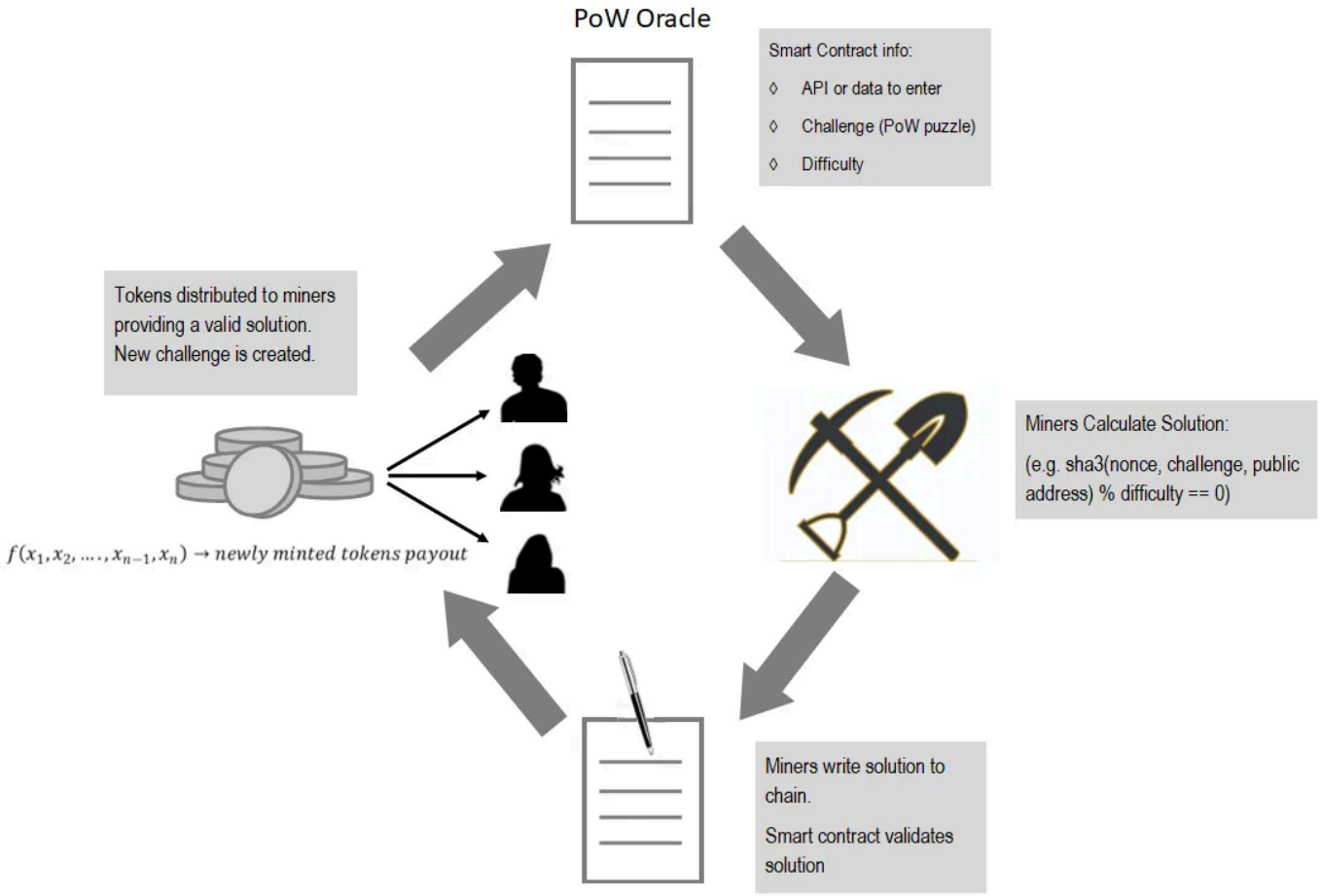


Figure 1. Proof-of-Work Oracle Diagram

On the right-hand side of the diagram, miners compete to find a solution to the challenge; the hash of the challenge, nonce (solution), and the public address of the miner will have a certain number of trailing zeros (the difficulty).

When a solution Is found, the miners will then input their solution along with the data requested into the smart contract (bottom of Figure 1). Parties will then be paid out an issued token based upon a formula for incentivizing honesty (e.g. median value is selected from n inputting miners).

Once the value is selected, it is stored, and a new challenge is created.

This simple system includes an additional game to incentivize a correct solution. Similar to the way Ethereum rewards 'Uncles' or miners who were close to winning, we plan to reward not the first party to solve the challenge, but the rather the party that submits the median value. For example, let's say we take the first five valid price submissions with valid proofs. We only take the median value and reward it fully, the next two entries that fall on either side are partly rewarded and the two submissions furthest from the median are rewarded even less.

$$f(x_1, x_2, \ldots., x_{n-1}, x_n) \rightarrow newly\ minted\ tokens\ payout$$

Positions, P:	1	2	3	4	5
	↓	↓	↓	↓	↓
Token Payout:	1	5	10	5	1

Figure 2- Sample PoWO payout

These tokens which are paid out, will then be valuable to parties who want to access data from our smart contract (we charge PoWO tokens for on-chain reads). This gives each token value, and more importantly, the value goes up as more smart contracts use our Oracle, thus creating a stronger incentive for miners.

With this solution, attacks on such a system now become relative to the value of the contracts that make use of the prices of the Oracle. At what point does it become worth it to burn compute cycles to manipulate the price? We are in the process of linking multiple Oracles to the value of our token, so the individual incentives from an oncoming loss due to an Oracle value will be diluted (e.g. a short on the BTC price would incentivize someone to report low values to our Oracle.)

Future Plans

We're planning on testing this out and making it more robust with plans including:

> *· GPU miner and miner not built in python*
>
> *· Standardized Oracle Token — Have one token and multiple oracle contracts (token is generalized for multiple Oracles)*
>
> *· Pay in Ether and/or token for data*
>
> *· Customize reward mechanism (not a hard fixed, median with uncles approach)*

Links

Check out our implementation (and miner): www.github.com/DecentralizedDerivatives/MineableOracle

Join our telegram: www.t.me/ddaorg

Become a member of DDA and come trade cryptocurrency derivatives: http://www.ddacoop.org/membership

[1] www.oraclize.it

[2] www.augur.net

Blockchain Ethereum Proof Of Work Token Smart Contracts

Written by thefett

Edit profile

183 Followers

CTO Tellor. Economics, crypto, regulation, and the revolution. twitter.com/@themandalore9

More from thefett

thefett

Social Flex #2- Dangers

LST's, restaking, systemic risks, parasites, greed, and a protocol's journey to death

8 min read · Dec 20, 2023

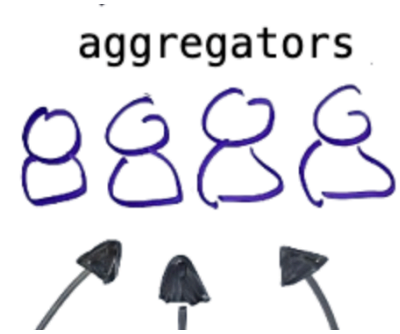
2

thefett

blue shell strategy—a path for removing centralization from the...

make a list of centralized people...ignore that list. the more overlap we have in our list, the...

7 min read · Apr 5, 2024



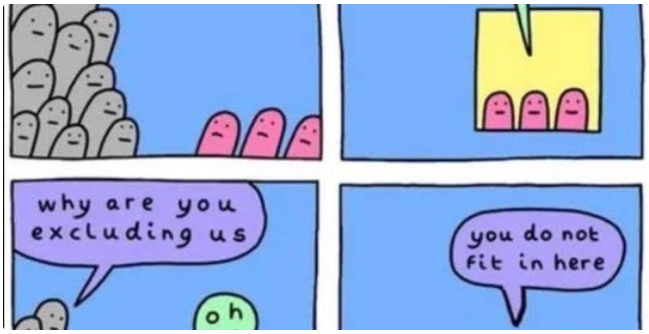
thefett

addressing systemic risks—discouragement attacks against...

Note: this is the fourth of a series of articles on social slashing of systemic risks. The first...

8 min read · Mar 22, 2024

2



thefett

social flex #1 — values

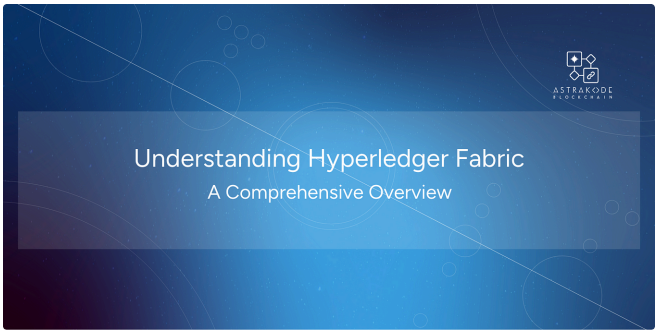
why values matter –social flexing in crypto

9 min read · Dec 7, 2023

31 1

See all from thefett

Recommended from Medium



AstraKode

Understanding Hyperledger Fabric: A Comprehensive Overview

When it comes to enterprise blockchain technology, hyperledger fabric stands out as...

11 min read · Apr 1, 2024

8

Yasint

Redstone

The integration of oracles into decentralized applications (dApps) and smart contracts is ...

3 min read · Jan 30, 2024

Lists

data science and AI

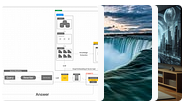
40 stories · 139 saves

My Kind Of Medium (All-Time Faves)

81 stories · 302 saves

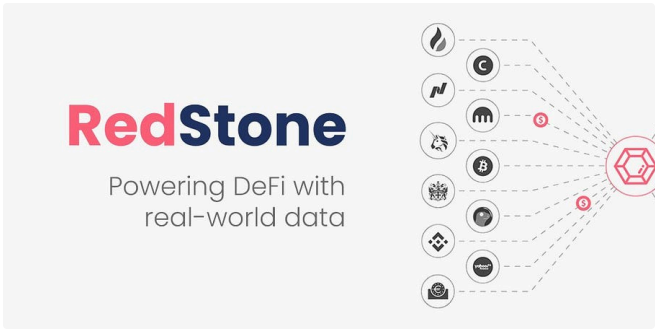
Modern Marketing

129 stories · 595 saves



Natural Language Processing

1411 stories · 910 saves

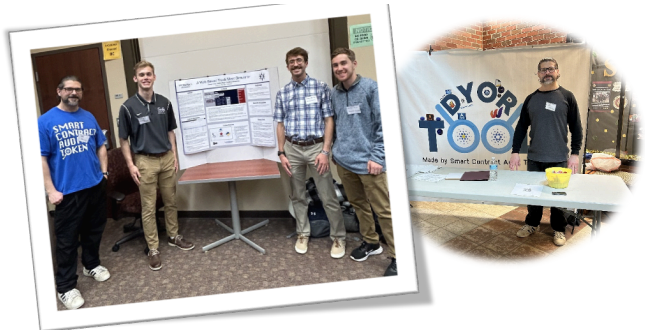


F1vot

RedStone Oracle. 3 Ways to integrate

Depending of the smart contract architecture and business demands we can deliver data...

3 min read · Jan 31, 2024



Smart Contract Audit Token

Fostering Blockchain Literacy in Academia

By Jed

3 min read · Apr 11, 2024

51

Kumzy

ALEO EXPLAINED

Aleo is a platform that offers private applications, achieving this by using...

3 min read · Dec 26, 2023

HEORHII YABLONSKYI

Securing Aleo: a deep dive into zkSecurity audits and resilient...

Hello, tech enthusiasts! I'm Heorhii, and I'm thrilled to delve into a riveting exploration of...

5 min read · Jan 5, 2024



See more recommendations