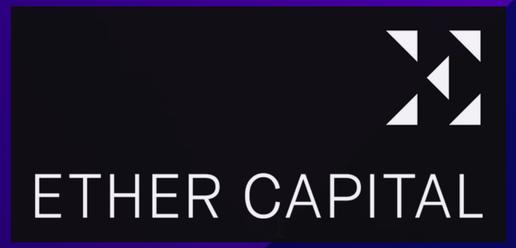
Oracles from the Ground Truth to Market Manipulation

ACM Advances in Financial Technologies
AFT 2021

Shayan Eskandari
CTO Ether Capital

CONSENSYS
Diligence



What is an oracle?

What is the context? How important it is to the rest of the system?

An oracle is a bridge or gateway that connects the off-chain real world knowledge and the on-chain blockchain network.

The Oracle Problem is real & still an open problem.

Oracle Problem

An interface from the blockchain environment to the external world is required for many applications.

If trust in this interface is required or if it is a centralized service, then why bother to use “trustless” decentralized smart contracts?

Oracle Use-cases

- Stablecoins and Synthetic assets
- Derivatives and prediction markets
- Provenance systems
- Identity
- Randomness
- Decentralized exchanges
- Dynamic non-fungible tokens (NFTs)

On the feasibility of decentralized derivatives markets

Shayan Eskandari[1] Jeremy Clark[2] Vignesh Sundaresan[1] Moe Adham[1]

1 Bitaccess
2 Concordia University

Abstract. In this paper, we present Velocity, a decentralized market deployed on Ethereum for trading a custom type of derivative option. To enable the smart contract to work, we also implement a price fetching tool called PriceGeth. We present this as a case study, noting challenges in development of the system that might be of independent interest to whose working on smart contract implementations. We also apply recent academic results on the security of the Solidity smart contract language in validating our code's security. Finally, we discuss more generally the use of smart contracts in modelling financial derivatives.

https://arxiv.org/abs/1802.04915

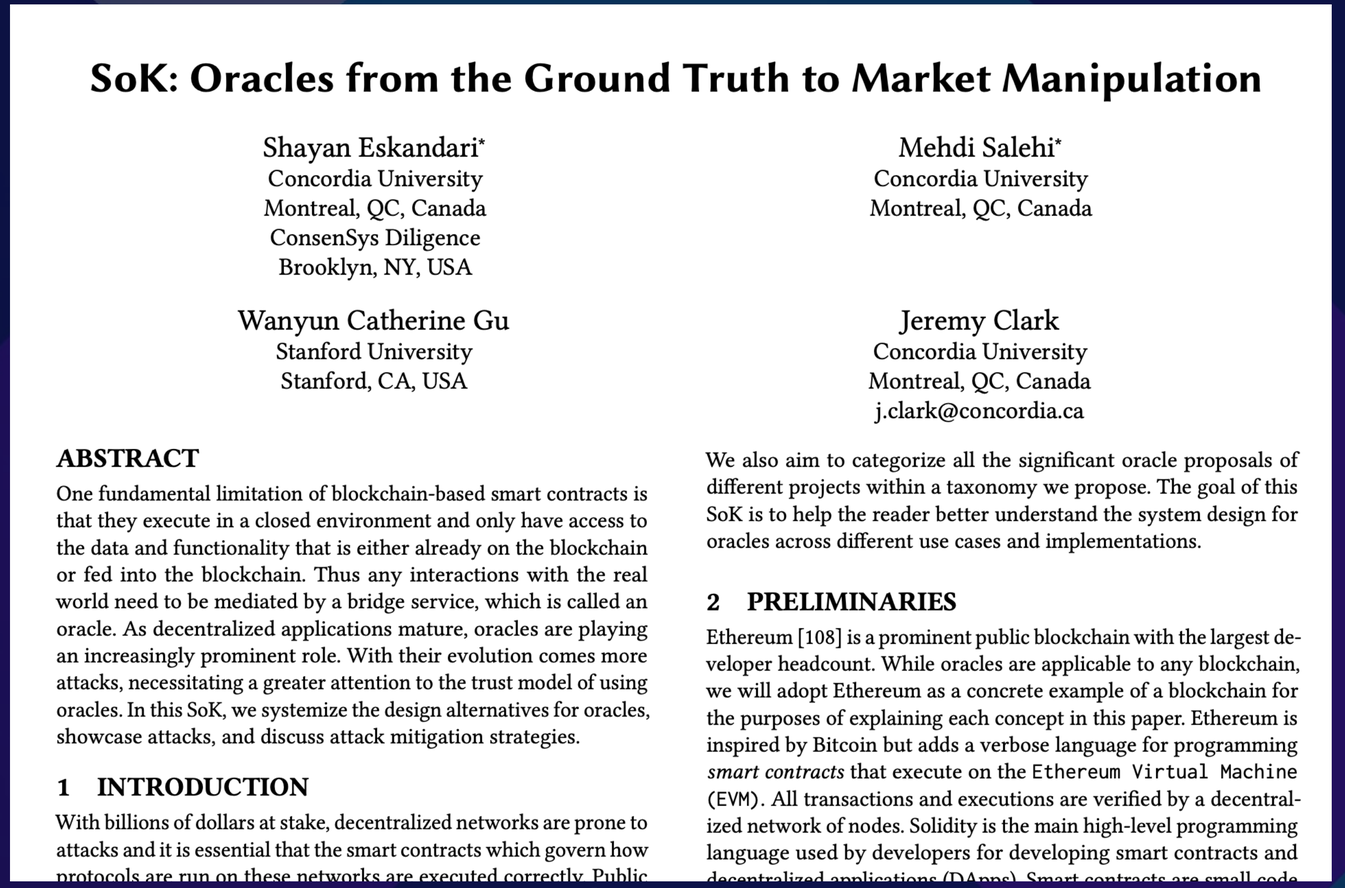
NFT: https://zora.co/shayan/2336

The prophecy of Decentralization

> *"The prophecy will be fulfilled soon, but before it can be, the Oracle must be consulted."*
>
> —— Morpheus

Systematization of Knowledge (SoK)

- A comprehensive research on the state of oracles
  - A specialized modular framework to analyze oracles
  - Systemize the design alternatives for oracles
  - Attacks and mitigation strategies
  - How to rethink oracle design



SoK: Oracles from the Ground Truth to Market Manipulation

Shayan Eskandari*
Concordia University
Montreal, QC, Canada
ConsenSys Diligence
Brooklyn, NY, USA

Mehdi Salehi*
Concordia University
Montreal, QC, Canada

Wanyun Catherine Gu
Stanford University
Stanford, CA, USA

Jeremy Clark
Concordia University
Montreal, QC, Canada
j.clark@concordia.ca

ABSTRACT

One fundamental limitation of blockchain-based smart contracts is that they execute in a closed environment and only have access to the data and functionality that is either already on the blockchain or fed into the blockchain. Thus any interactions with the real world need to be mediated by a bridge service, which is called an oracle. As decentralized applications mature, oracles are playing an increasingly prominent role. With their evolution comes more attacks, necessitating a greater attention to the trust model of using oracles. In this SoK, we systemize the design alternatives for oracles, showcase attacks, and discuss attack mitigation strategies.

1 INTRODUCTION

With billions of dollars at stake, decentralized networks are prone to attacks and it is essential that the smart contracts which govern how protocols are run on these networks are executed correctly. Public

We also aim to categorize all the significant oracle proposals of different projects within a taxonomy we propose. The goal of this SoK is to help the reader better understand the system design for oracles across different use cases and implementations.

2 PRELIMINARIES

Ethereum [108] is a prominent public blockchain with the largest developer headcount. While oracles are applicable to any blockchain, we will adopt Ethereum as a concrete example of a blockchain for the purposes of explaining each concept in this paper. Ethereum is inspired by Bitcoin but adds a verbose language for programming *smart contracts* that execute on the `Ethereum Virtual Machine (EVM)`. All transactions and executions are verified by a decentralized network of nodes. Solidity is the main high-level programming language used by developers for developing smart contracts and decentralized applications (DApps). Smart contracts are small code

https://arxiv.org/abs/2106.00667

CONSENSYS Diligence

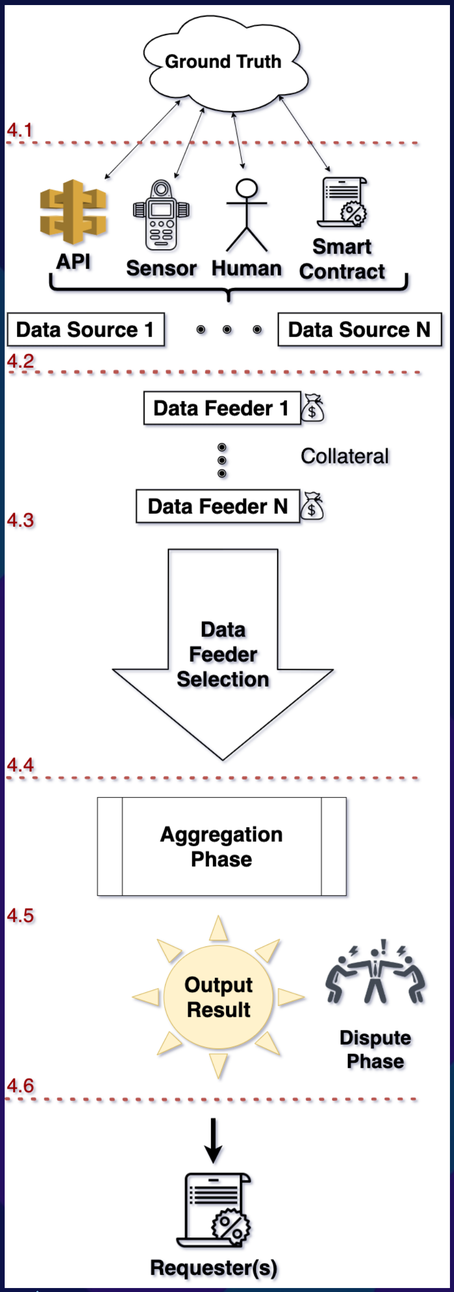
UNIVERSITÉ Concordia UNIVERSITY

acm Oracles from the Ground Truth to Market Manipulation

Oracle's Modular Workflow

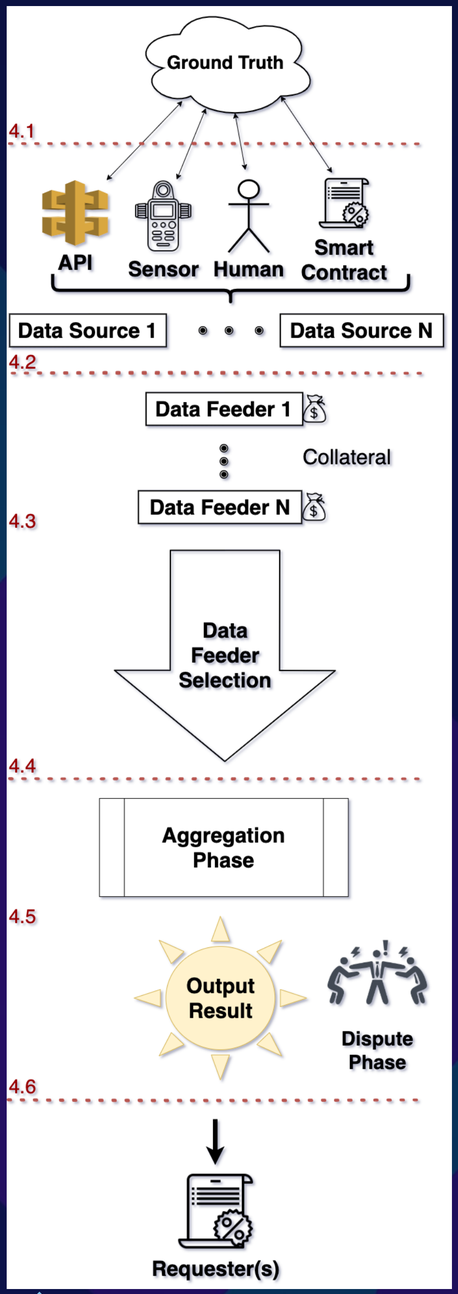
We break the design of an oracle system into modules

Each module can be on-chain or off-chain, or in different order. But all oracles can be defined using these modules



Modular Workflow overview

- Ground Truth
- Data Sources
- Data Feeders
- Selection of Data Feeders
- Aggregation
- Dispute Phase

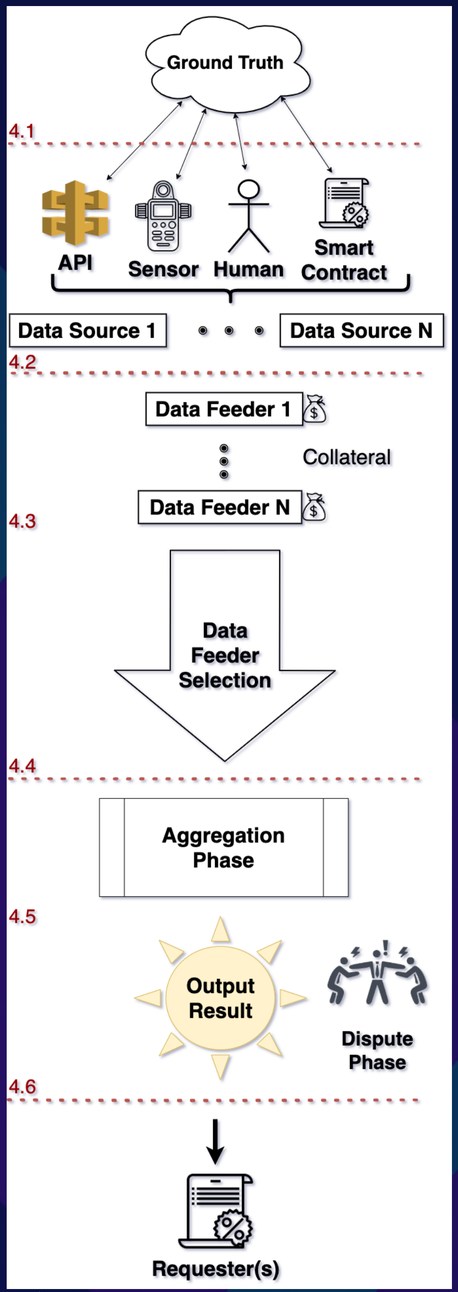


Modular Workflow overview

- Ground Truth
- Data Sources
- Data Feeders
- Selection of Data Feeders
- Aggregation
- Dispute Phase

Interacting with Blockchain

- Off-chain Infrastructure
- Blockchain Infrastructure
- Smart Contracts
  - ✳Oracle
  - ✳Data Consumer



Ground Truth

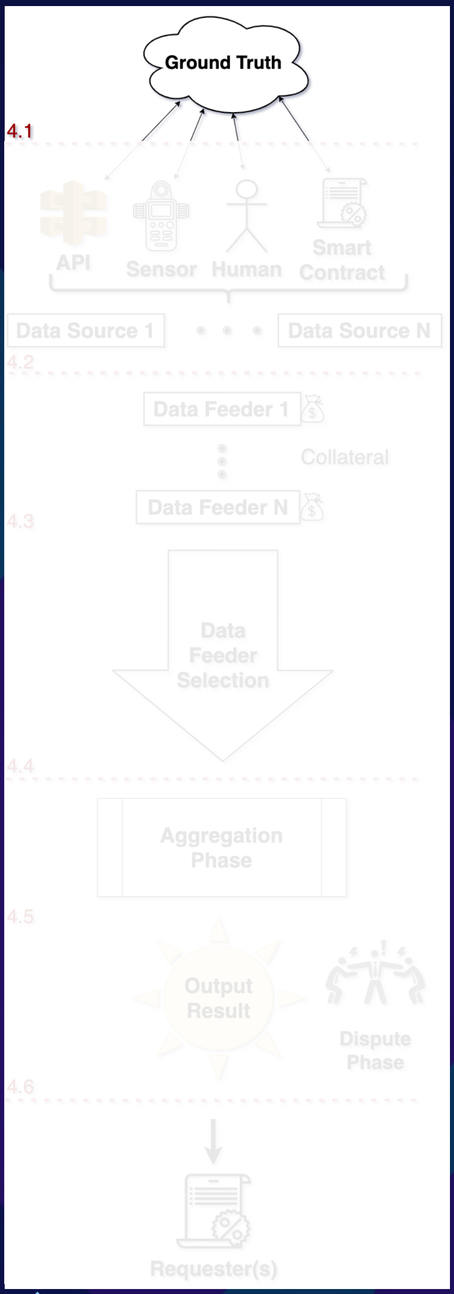
The first challenge of an oracle designer starts with a philosophical question:

- What is the truth?
- Is the truth objective or subjective?
- As an example, what is the real price of Ethereum to you?

– – – – – – – – – – – – – – – – – –

We try to solve:

- *What is the best way to collect raw data from sources and aggregate them to have a reasonable data quality?*

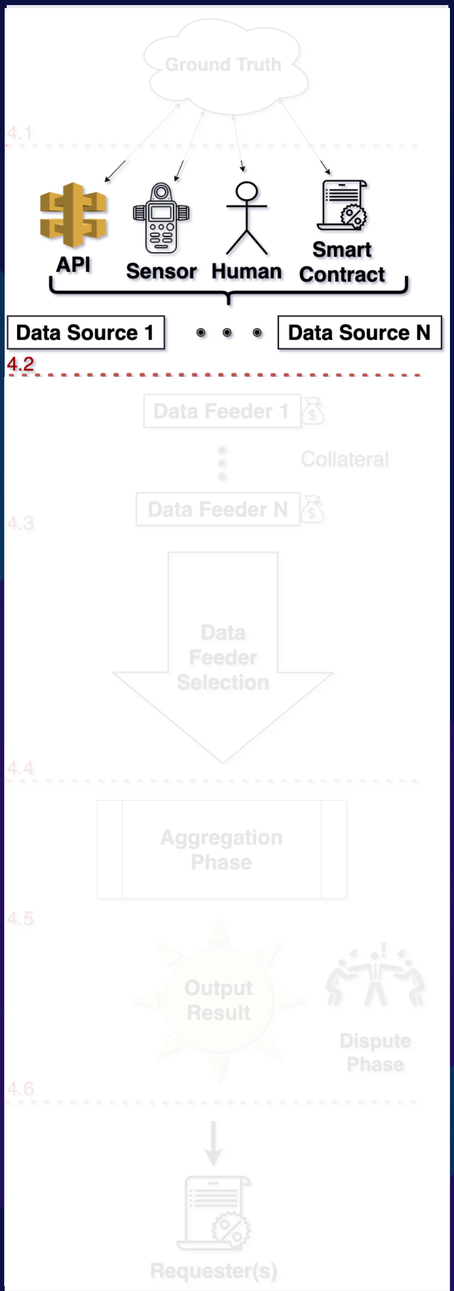


Data Sources

Passive entities that store and measure the representation of the ground truth.

Common types of data sources include:

- Databases & API
- Sensors
- Humans
- Smart contracts
  or a combination of them

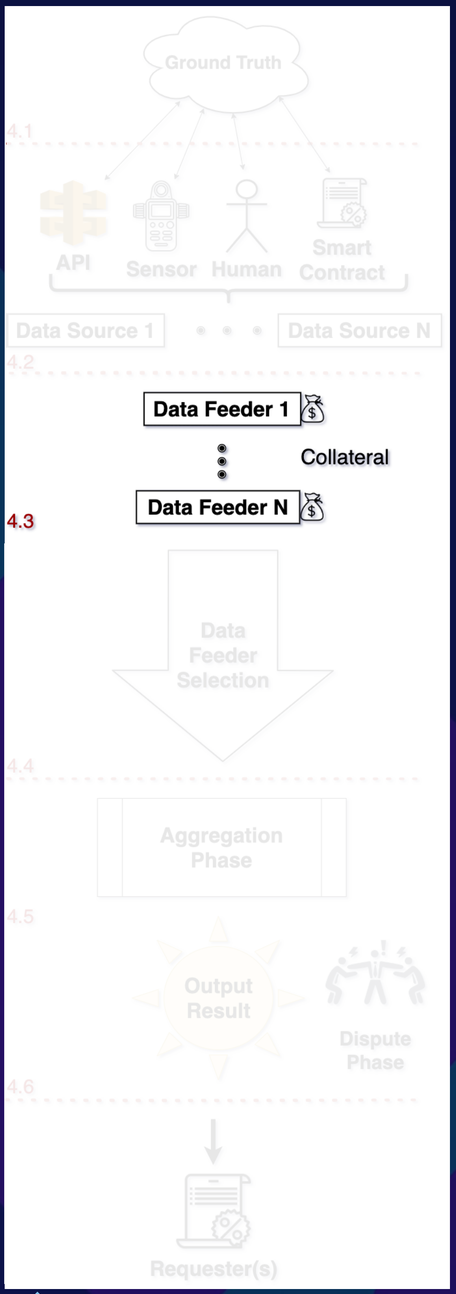


Data Feeders

Entities who gather and report the data from the source to the oracle system

Security Provisions:

- **Source Authentication**
  - **DECO, TLSNotary — HTTPS Schema**
  - **TownCrier — Intel SGX**
- Confidentiality
- Non-Repudiation



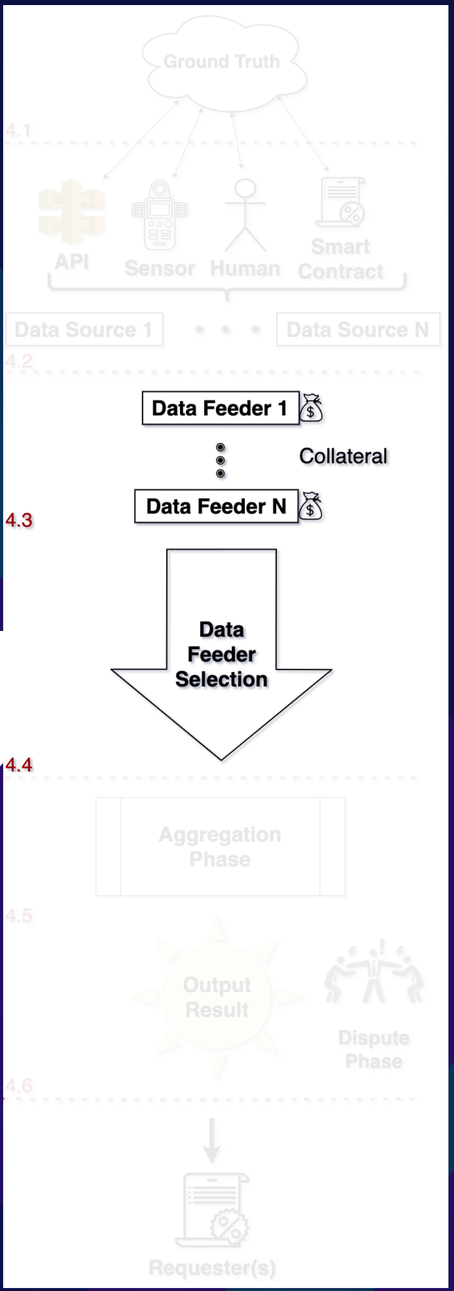
Selection of Data Feeders

Select legitimate data feeders and weed out the less qualified participants

- Centralized (Allowlist) Selection
- Decentralized (Allowlist) Selection
  - Voting
  - Staking
  - +Random

Category	*Example*	No Trusted Third Party	Low latency	Resilient to Sybil Attacks	Resilient to Targetted DoS Attacks	Incentives are Endogenous
Centralized	`Maker V1 Oracle`		•	•		
Voting	`Maker V2 Oracle`	•			•	
Staking	`Chainlink, ASTRAEA`	•	•	○	•	•

Table 1: Evaluation Framework on selection of data feeders. For details see Section 4.4.4

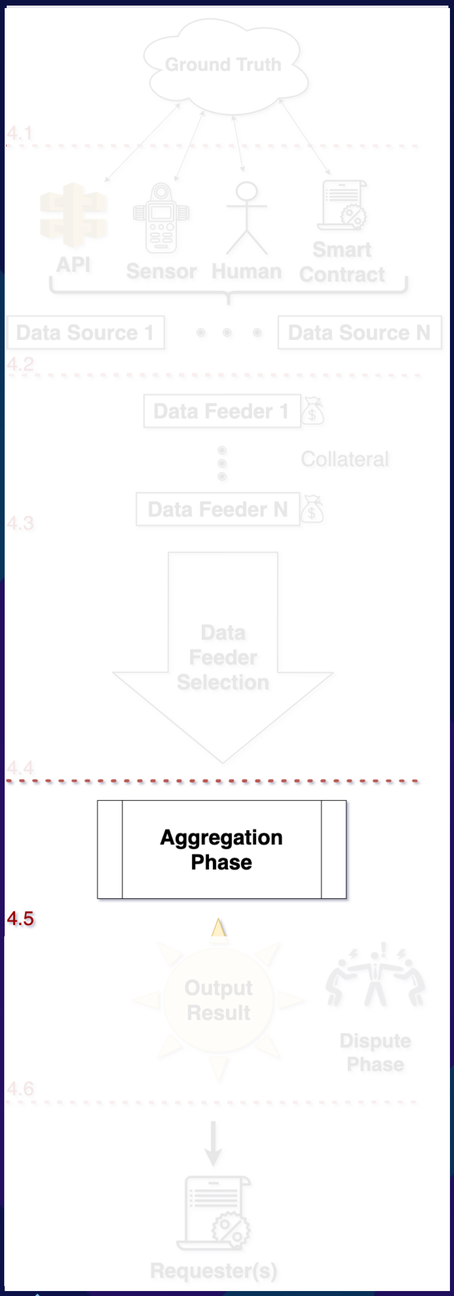


Aggregation Phase

The process of synthesizing the selected data feeds into one single output

- Statistical Measures
  - Mean / Median / etc
  - Weighting methods (Time, Volume, etc)
  - Random selection

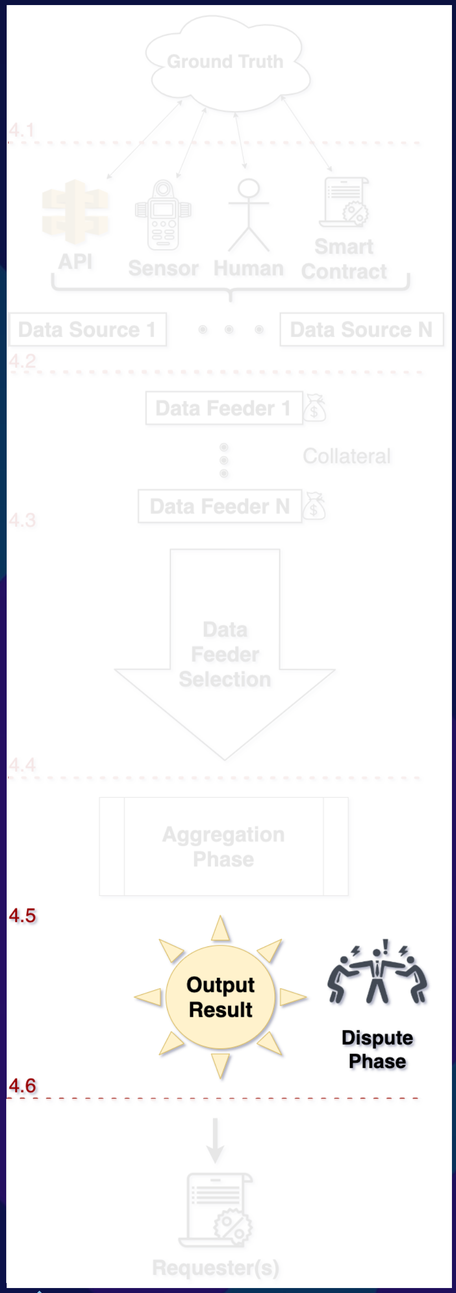
- Expiration Date and Stale Data



Dispute Phase

Safeguards the quality of the final output and give the stakeholders a chance to mitigate inclusion of wrong data

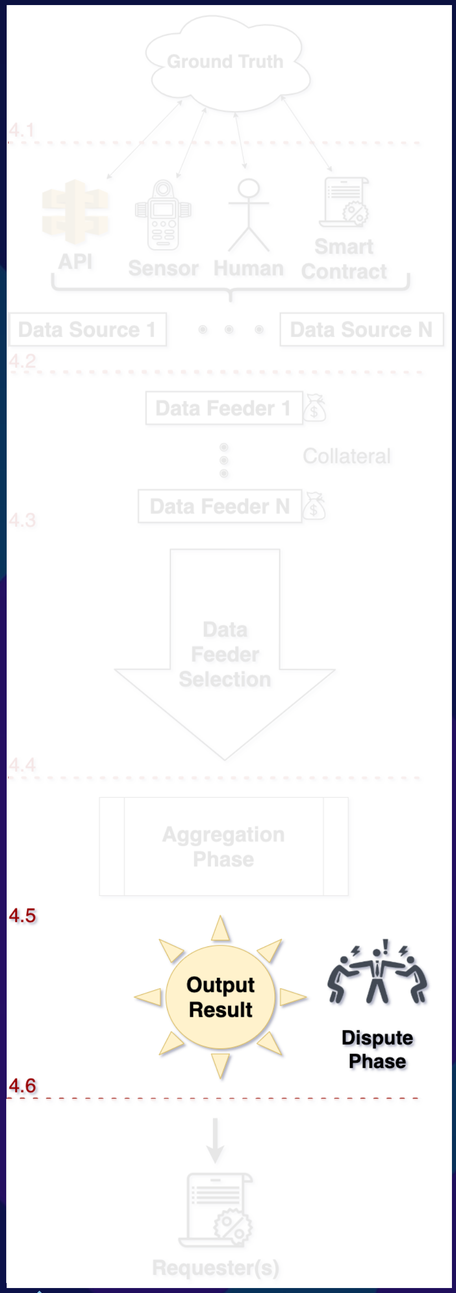
- Provider-level v.s. Data-level Vetting



Dispute Phase (continued)

Determining the Truth

- Statistical approach (+ Stake)
- Voting (+ Stake)
- Arbitrage (For exchange rates between two on-chain tokens)



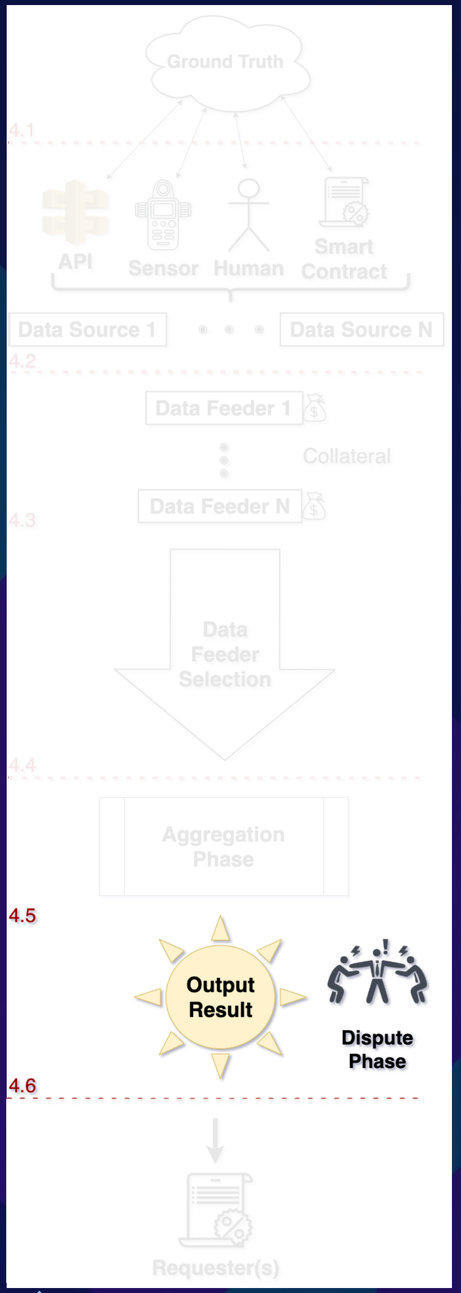
Dispute Phase (continued.)

Consequences for Incorrect Data

- Revert or Correct
  - Impacts finality of data

Consequences for Data Feeder

- banned, slashed, reputation loss, or no consequence



Classification of the current Oracles

Oracle	Data Source	Selection Mechanism: Data Feeder	Staking	Aggregation Mechanism	Provider/Data Vetting: Dispute	Determining the Truth: Dispute	Consequences (Slash, Ban, Loss): Dispute
ChainLink [41]	API	Reputation, Staking	•	Statistical Measure	P	Statistical Measure	S
UMA [101]	Human, API	FCFS†	•	×	D	Staking	S
Augur [85]	Human	Single Source★	•	×	D	Voting	S
Uniswap [102]	Smart Contract	×	×	TWAP	×	×	×
MakerDAO V1 [74]	Human, API	Centralized Allowlist	×	Median	×	×	×
MakerDAO V2 [74]	Human, API	Decentralized Allowlist	×	Median	P	Voting	B
NEST [80]	Human	×	•	×★★	D	Arbitrage	L
Band protocol [87]	API	Random Selection	•	Statistical Measure	P	Staking	S
Tellor [31]	Human, API	PoW	•	Median	P	Staking	S B
ASTRAEA [3] TruthCoin [96]	Human	Staking	•	Mode	D	Voting	S
Provable [10] PriceGeth [44]	API	×	×	×	×	×	×
DIA Oracle [38]	API, Smart Contract	×	×	×	D	Staking	B
DECO [113] TownCrier [112]	HTTPS	×	×	×	×	×	×
API3 [9] \w Kleros [68]	Oracles	Decentralized Allowlist	•	Statistical Measure	P	Voting	S B

Table 2: A classification of the existing oracle implementations using the modular framework described in Section 4.
• indicates the properties (columns) are implemented in the corresponding oracle (rows), and × indicates the property is not applicable.
† First Come First Serve ★*The Market Creator assigns the designated reporter* ★★ *The series of reported prices will be sent to requester without aggregation (See 4.6.1)*

Now the dark forest

Interacting with Blockchain

- Off-chain Infrastructure
- Blockchain Infrastructure
- Smart Contracts
  - ✳Oracle
  - ✳Data Consumer

Off-chain Infrastructure

- Monitoring the Blockchain (for request- response oracles)
- Connection to Data Source
- Data Feeders Network
- Transaction Creation
    - Secure private key & signing
    - Positive balance for gas
    - Dynamic gas fee

Blockchain Infrastructure

- Blockchain Node
  - Synced and available
- Block Creation
  - Front-running attacks
  - MEV
- Consensus
  - Uncle Blocks (EtherRoll reorg attack)

Oracle Interaction Models

- **Feed** (e.g. MakerDAO Oracle, Chainlink price feed)
  - Publishes the data for others to use
  - No additional transaction needed to fetch the data
  - Uses an interval to update data

- Request-Response
  - Similar to a client-server API request on web2
  - At least 2 transactions

- Subscribe-Response
  - Similar to Request-Response, but request doesn't need to be a transaction
  - Off-chain agreement (e.g. API3), event subscription

Oracle's SmartContract.

Can include:

- Data feeder selection
- Aggregation mechanism
- Dispute resolution

Additionally:

- Data feed storage
- Authenticate Oracle's response

Possible issues: Implementations Flaws & Governance attacks

Data Consumer Smart Contract

The final point in the oracle workflow

- Essential to use oracles with multiple data feeders and a proper aggregation methods
  - Compound, CoinbasePro incident
- Verify the request properties (SLA)
- Authenticate the Oracle
  - Nexus Mutual issue

Food for Thought

- Oracles use their governance token for Staking, two necessary conditions:
  - The market capitalization of the token stays material
  - The token is evenly distributed.

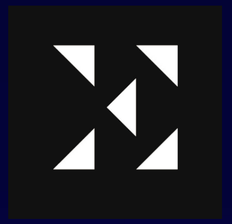
- How to incentivize on-chain oracles for use-cases with no direct revenue? (Weather data)

- Diversity in software promotes resilience in the system. If one project is the main Oracle, failure within that one project causes cascading failure across the ecosystem

- One can never capture the full extent of Profit from Corruption

CONSENSYS
Diligence

Thank You



Get in touch:

Twitter: @sbetamc

https://shayan.es

shayan@ethcap.co

acm

Raymond Chabot
Grant Thornton

NSERC
CRSNG

UNIVERSITÉ
Concordia
UNIVERSITY