IS4302
Blockchain and Distributed Ledger Technology

Lecture 8

10 March, 2023

NUS
National University
of Singapore

Group project

- **Deliverables**
  - Demo presentation (8-15 min recorded video)
    - **No frontend required**
  - Architecture and design document
    - **Business logic and technical implementation**
  - Code
    - **As a GitHub link in the document**
  - Survey on peer evaluation
- **Deadline**
  - Document and presentation both week 12?
  - Document and presentation both week 13?
  - Document week 12, presentation week 13?

Overview

- **The oracle problem**
  - Reference: Caldarelli, Giulio. "Understanding the blockchain oracle problem: A call for action." Information 11.11 (2020): 509.

- **Oracles in the current ecosystem**
  - Reference: https://chain.link/

- **Example of a decentralized oracle**
  - Reference: Adler, John, et al. "Astraea: A decentralized blockchain oracle." 2018 IEEE international conference on internet of things (IThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). IEEE, 2018.

What Are Oracles?

- **The term “oracle” comes from Greek mythology**
  - refers to someone able to communicate directly with god and see the future
- **In ancient stories, people did not have enough information to make decisions, and turned to oracles for knowledge beyond their understanding.**
- **In the blockchain environment, oracles are systems that provide blockchain with information coming from the real world.**

What Are Oracles?

- **In a blockchain, extrinsic information cannot be provided along with transaction data, since other nodes would detect information coming from an “untrusted” source.**
- **Information coming from the real world should come from a third-party univocal source**
- **Reliability is undisputed for all nodes**

- **Oracles (on the blockchain) do not predict the future but retrieve information from the past**

What Are Oracles

- **Oracles are not specific programs or devices, but "concepts".**
  - Anything providing external data to the blockchain can be classified as an oracle.
- **Oracles, in general, do not insert information into the blockchain directly; conversely, they gather and store data from the real world.**
- **When a smart contract concerning extrinsic data is executed, the code then calls for the right information from a trusted oracle.**

Examples of oracles

- **IoT systems such as probes and sensors**
- **Platforms such as ERP**
- **In the case of private data, the very human that operates directly on the blockchain**

Examples of oracles

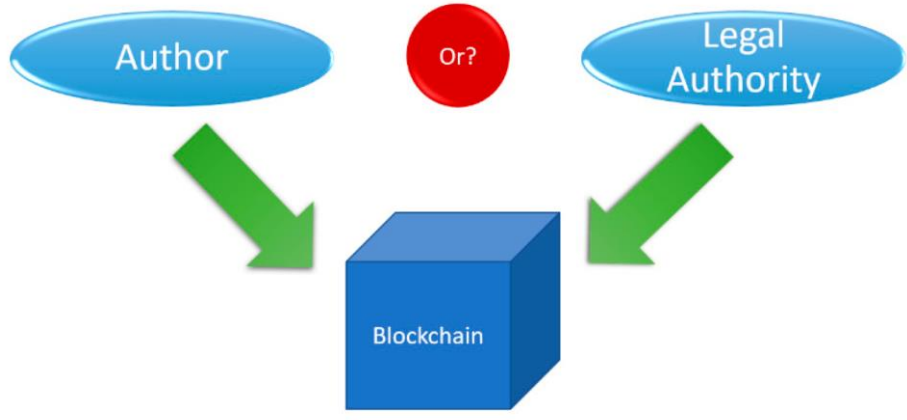
- **Examples of data gathered by oracles**
  - Events/data in other blockchains
  - Randomness
  - Lottery winners
  - Natural disasters along with risk measurements
  - Price and exchange rate of real/crypto assets
  - Static data (e.g., country codes)
  - Dynamic data (e.g., time measurements)
  - Weather conditions
  - Political events
  - Sporting events
  - …

The oracle problem

- **The security, authenticity, and trust conflict between third-party oracles and the trustless execution of smart contracts**
  - Reintroduce the single-point-of-failure
  - Needs to be trusted, removing trustless peer-to-peer interaction
  - “two step-back from decentralization”

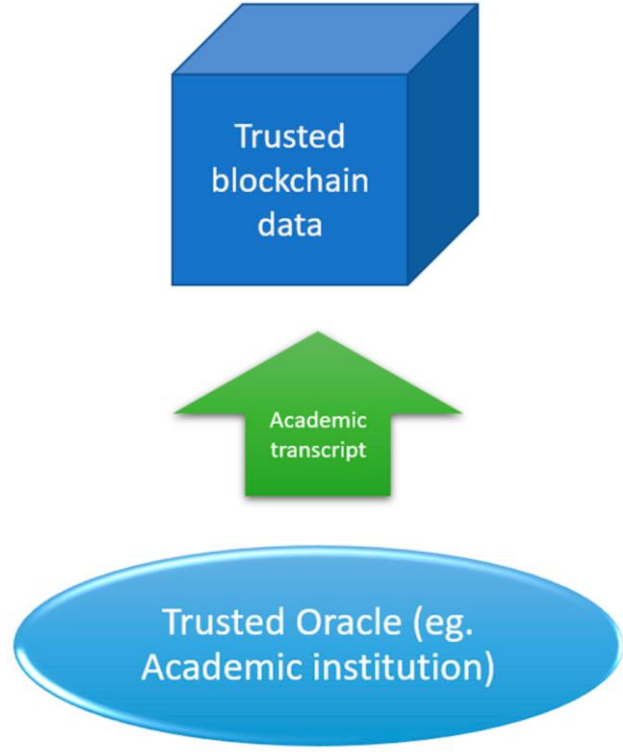
Example – IPRS protection

- **The original purpose of the blockchain, before the crypto 'era', was to "register" intellectual property rights**
  - Protect against tampering
  - Efficiently share revenues
  - Eliminate intermediaries?
- **Who decides ownership? Author or legal authorization?**
  - What if someone recognizing his or her work as registered by someone else?



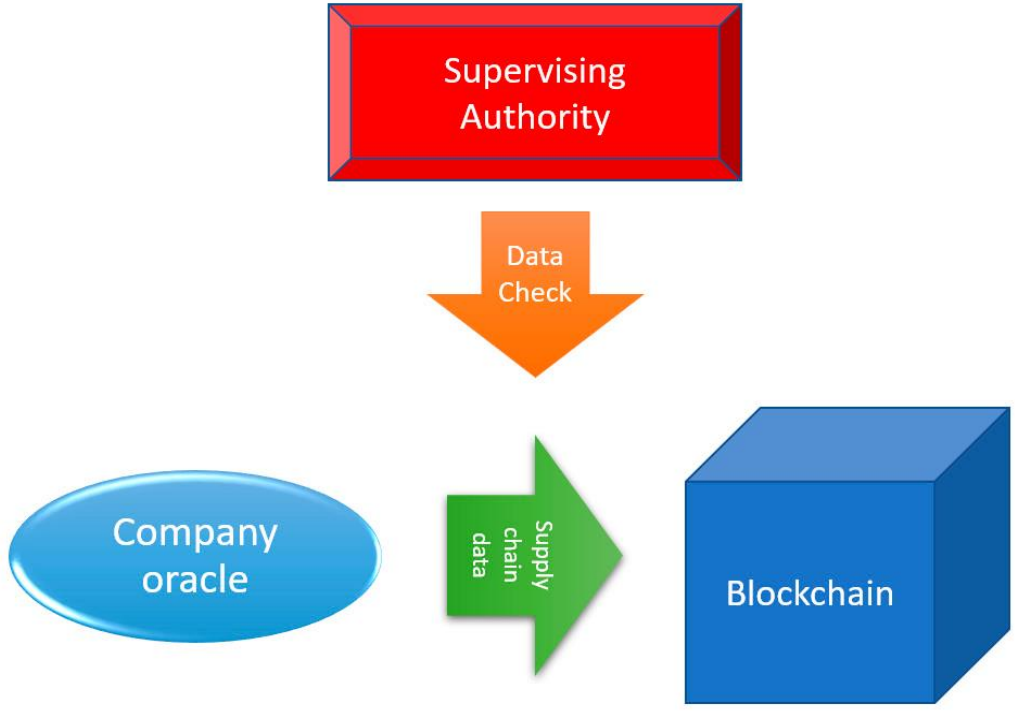
Example – Academic Transcript

- **The universities are themselves Oracles**
  - Universities, especially reputable ones, are generally trustworthy



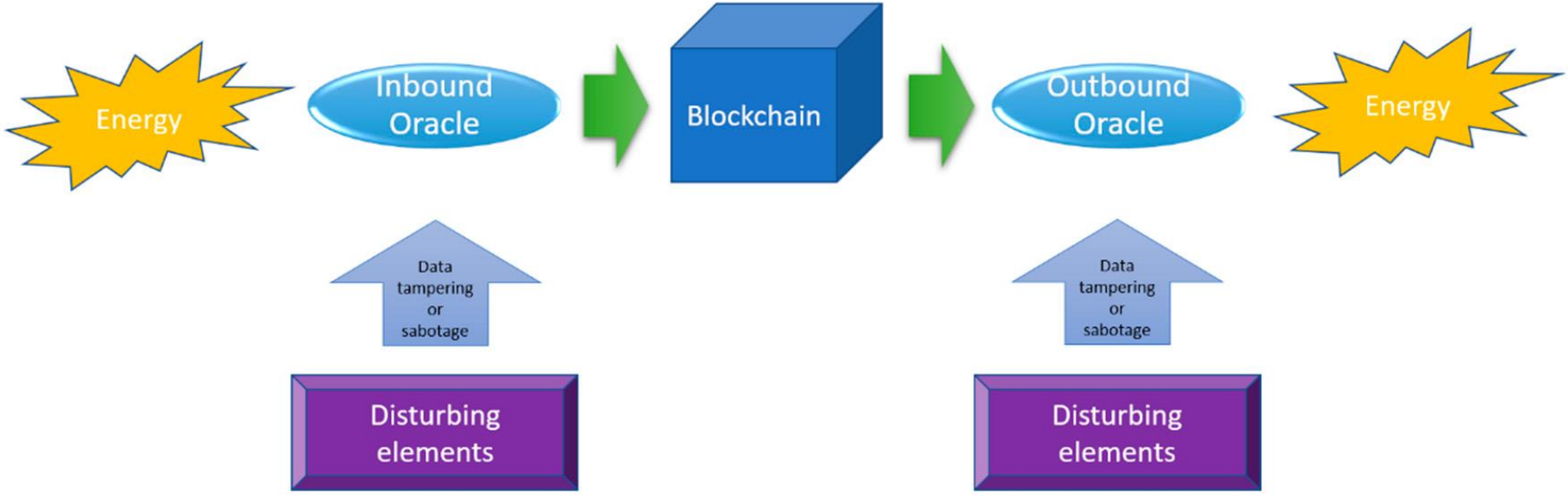
Example – Supply chain management

- **Oracles belong to the company producing goods that are being tracked**
  - Companies decide what information to upload on the blockchain
- **An auditor?**
  - Who audits the auditor?
  - Does benefits outweighs costs?



Example - Energy

- **Introducing blockchain within the energy sector**
  - reduction in costs for the marketplace
  - increased transparency and decentralization



Example - Energy

- **The platform could be decentralized and independent from a central authority**
- **Oracles are unlikely to also be decentralized and autonomous**
  - Local infrastructure
  - Monitoring and maintenance
  - …

Overview

- **The oracle problem**
  - Reference: Caldarelli, Giulio. "Understanding the blockchain oracle problem: A call for action." Information 11.11 (2020): 509.

- **Oracles in the current ecosystem**
  - Reference: https://chain.link/

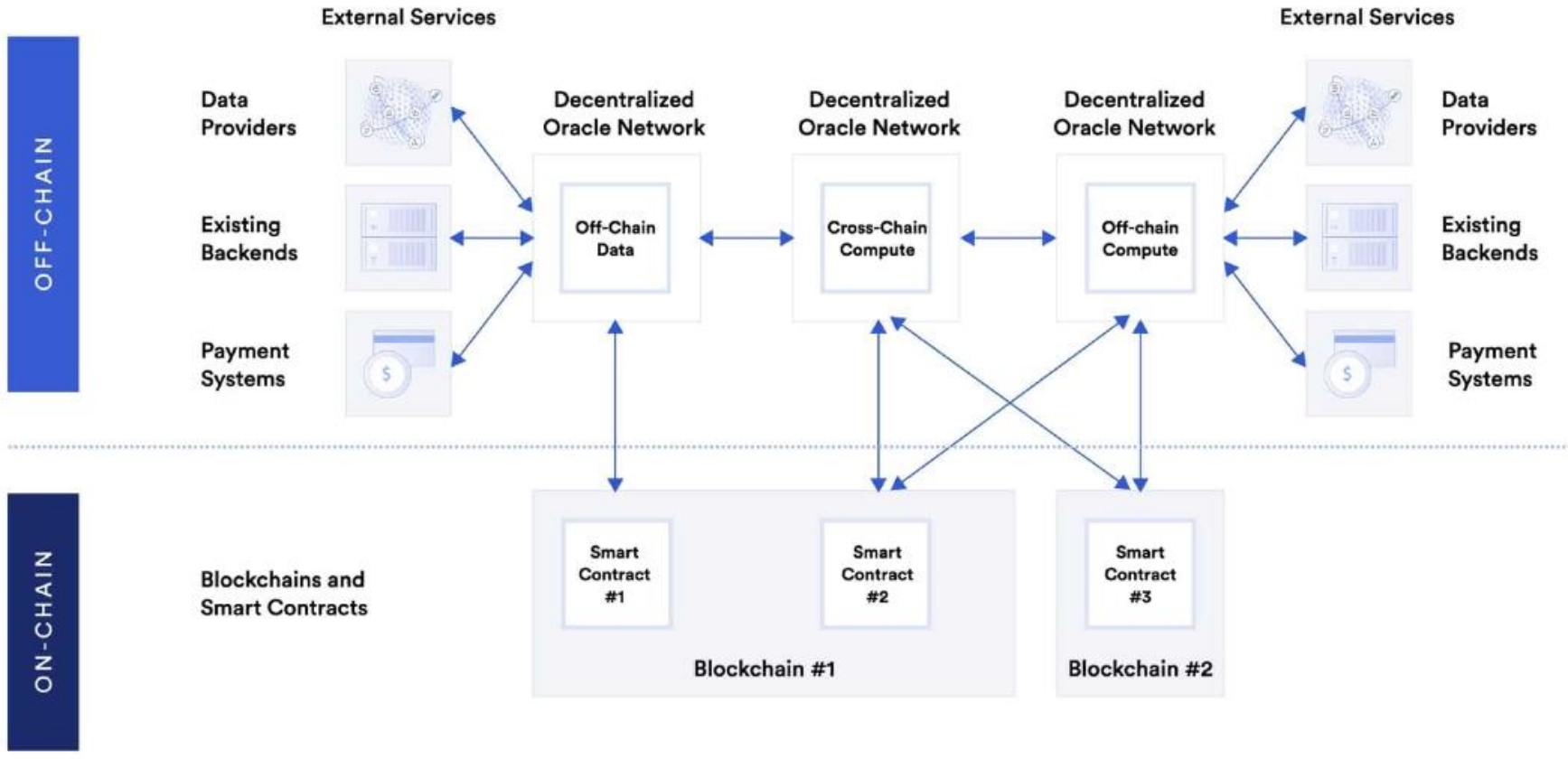
- **Example of a decentralized oracle**
  - Reference: Adler, John, et al. "Astraea: A decentralized blockchain oracle." 2018 IEEE international conference on internet of things (IThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). IEEE, 2018.

Types of Blockchain Oracles

- **Input Oracles**
  - most widely recognized type of oracle
  - fetches data from the real-world (off-chain) and delivers it onto a blockchain network for smart contract consumption
- **Output Oracles**
  - allow smart contracts to send commands to off-chain systems that trigger them to execute certain actions
  - E.g., informing a banking network to make a payment
  - telling a storage provider to store the supplied data
  - pinging an IoT system to unlock a car door once the on-chain rental payment is made

Types of Blockchain Oracles

- **Cross-Chain Oracles**
  - read and write information between different blockchains
  - enable interoperability for moving both data and assets between blockchains
  - E.g., using data on one blockchain to trigger an action on another or bridging assets cross-chain
- **Compute-Enabled Oracles**
  - secure off-chain computation to provide services that are impractical to do on-chain due to technical, legal, or financial constraints
  - E.g., computing zero-knowledge proofs to generate data privacy
  - running a verifiable randomness function



Different types of oracles enable the creation of hybrid smart contracts

Blockchain Oracle Use Cases

- **DeFi**
  - decentralized money markets use price oracles to determine users' borrowing capacity and check if users' positions are undercollateralized and subject to liquidation
  - automated market makers (AMMs) use price oracles to help concentrate liquidity at the current market price to improve capital efficiency
  - …

Blockchain Oracle Use Cases

- **Dynamic NFTs and Gaming**
  - Dynamic NFTs—Non-Fungible Tokens that can change in appearance, value, or distribution based on external events like the time of day or the weather
  - compute oracles are used to generate verifiable randomness that projects then use to assign randomized traits to NFTs or to select random lucky winners in high-demand NFT drops.
  - On-chain gaming applications also use verifiable randomness

Blockchain Oracle Use Cases

- **Insurance**
  - Insurance smart contracts use input oracles to verify the occurrence of insurable events during claims processing, opening up access to physical sensors, web APIs, satellite imagery, and legal data.
- …

Overview

- **The oracle problem**
  - Reference: Caldarelli, Giulio. "Understanding the blockchain oracle problem: A call for action." Information 11.11 (2020): 509.

- **Oracles in the current ecosystem**
  - Reference: https://chain.link/

- **Example of a decentralized oracle**
  - Reference: Adler, John, et al. "Astraea: A decentralized blockchain oracle." 2018 IEEE international conference on internet of things (IThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). IEEE, 2018.

System overview

- **Name: ASTRAEA**
  - goddess of justice, innocence, purity and precision
- **An oracle that decides the truth value of Boolean propositions**
- **Leverages human intelligence through a voting-based game**

System overview

- **Users of ASTRAEA participate in one (or more) of the following three roles: submitters, voters, and certifiers**

- **Submitters**
  - choose which propositions enter into the system
  - allocate money to fund (in part) the subsequent effort to validate the Boolean propositions.

System overview

- **Voters**
  - play a low-risk/low-reward role
  - Once such a player submits a deposit (stake), she is given the chance to vote on a proposition chosen uniformly at random from the ones available
  - The stake is placed before the voter is notified of which proposition she will be voting on.
  - The outcome of the voting process is a function of the sum of the votes weighted by the deposits.
  - The maximum voting deposit is a parameter of the system.

System overview

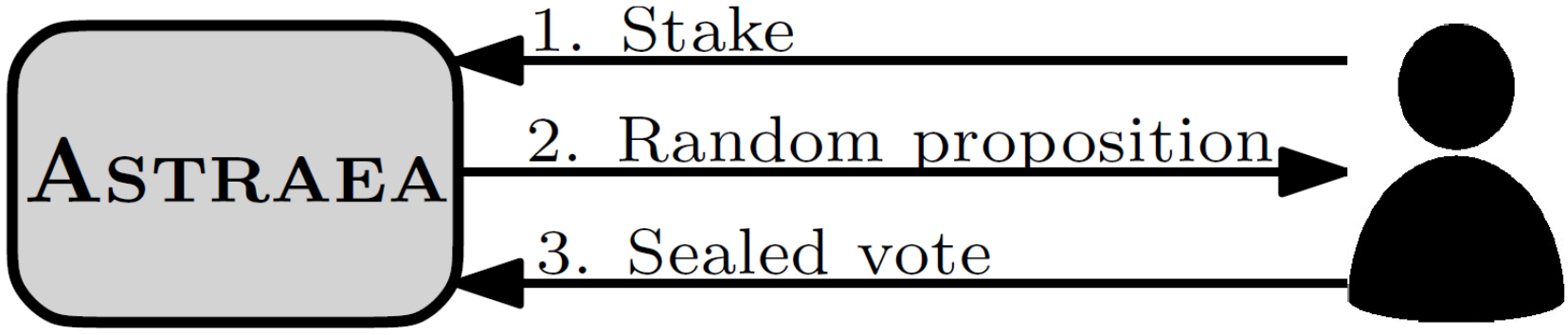


Fig. 1. Player voting in ASTRAEA

System overview

- **Certifiers**
  - play a high-risk/high-reward role
  - choose an available proposition and place a large deposit in order to certify it as either true or false
  - The certification outcome is a function of the sum of certifications weighted by the deposits.

System overview

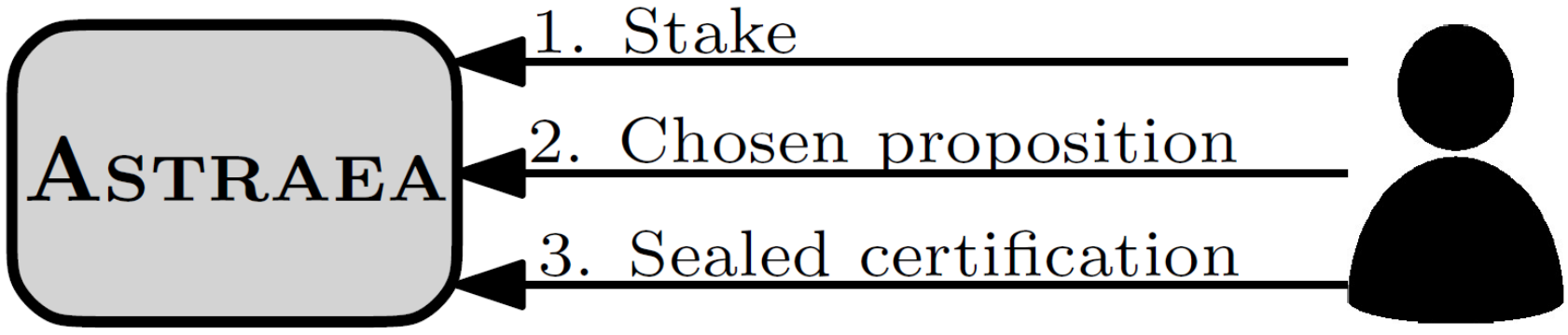


Fig. 2. Player certifying in ASTRAEA

Parameters

- **Setting the parameters appropriately is important for the functionality of the system.**
- **The maximum voting deposit size should be small relative to the total voting stake on each proposition.**
  - If a single vote can account for 100% of a proposition's total voting stake, an adversary can have total control over the outcome of a randomly drawn proposition.
  - Conversely, if it is 1%, the adversary would somehow need to draw the same proposition repeatedly to control its validity outcome.

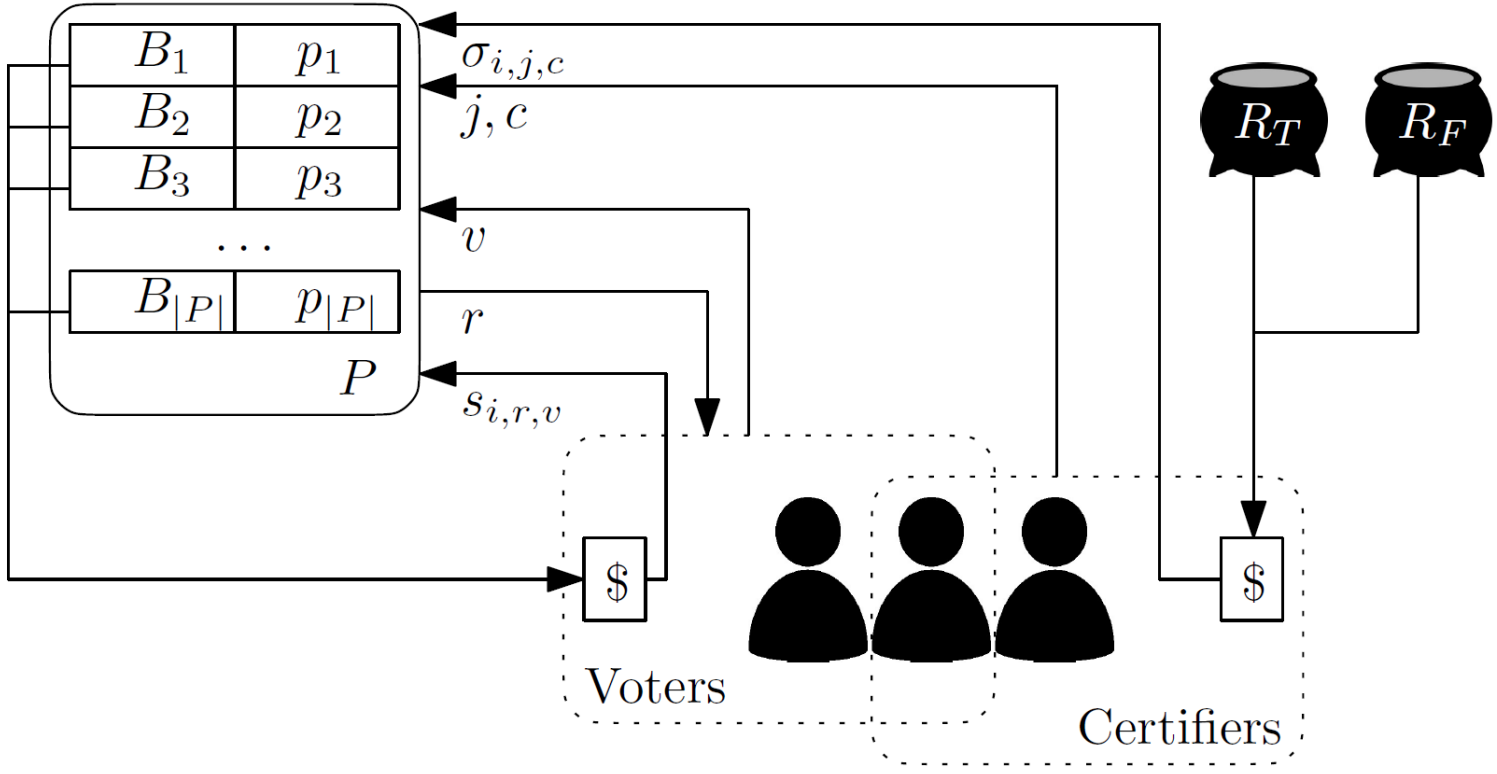
Parameters

- **The minimum certification deposit should be large enough that certifiers incur sufficient risk.**
  - Otherwise, they may be tempted to abuse their certifications for griefing purposes as a malicious certification can impact reward payouts for other players.
  - individual certifiers have enormous influence on the process for individual propositions
  - the certifiers alone cannot force the oracle to produce an incorrect value and they are encouraged to behave honestly by the incentive structure; otherwise they face large penalties.

The proposition list

- **There exists a set of propositions constructed outside of ASTRAEA called the proposition list, which is denoted by P and is of fixed size |P|.**
- **Each proposition $p_i \in P$ has a hidden truth value t_i and is associated with a bounty amount B_i**
- **The voting game is played over all propositions in P simultaneously.**
- **Two certifier reward pools containing R_T and R_F monetary units, intended to reward certifiers for true and false outcomes, respectively.**

Overview of ASTRAEA



Voting

- **Players may place a deposit in order to vote on a randomly-chosen proposition.**
- **First, the voter i indicates a desire to stake an amount of money $s_{i,r,v} \leq s_{max}$ on a vote for proposition p_r, where r is not yet known to the voter.**
- **Subsequently, the system chooses the value for r and sends it to the voter.**
  - r is chosen uniformly at random from [1, |P|]
  - a voter may be given the opportunity to vote on the same proposition more than once.
  - Finally, the voter submits a sealed vote $v \in \{T, F\}$. This can be accomplished using a commit-reveal scheme in which the voter commits to a hash of their vote concatenated with a nonce and later reveals the vote and nonce.

Certifying, termination, and decision

- **Players may place a large deposit in order to certify propositions of their choosing.**
- **The certifier simply submits a monetary stake $\sigma_{i,j,c} \geq \sigma_{max}$ and a sealed certification $c \in \{T, F\}$ for proposition p_j.**
- **Once a proposition has accumulated sufficient voting stake, it is decided.**

VOTING OUTCOMES

Outcome	Condition
T	$s_{TOT,j,T} > s_{TOT,j,F}$
F	$s_{TOT,j,F} > s_{TOT,j,T}$
Ø	$s_{TOT,j,T} = s_{TOT,j,F}$

CERTIFYING OUTCOMES

Outcome	Condition
T	$\sigma_{TOT,j,T} > \sigma_{TOT,j,F}$
F	$\sigma_{TOT,j,F} > \sigma_{TOT,j,T}$
Ø	$\sigma_{TOT,j,T} = \sigma_{TOT,j,F}$

- **Maybe supermajority better?**

Outcomes of the game

(a)	GAME		
V \ C	T	F	Ø
T	T	Ø	Ø
F	Ø	F	Ø
Ø	Ø	Ø	Ø

- **Determine rewards**
- **Oracle output not necessarily same with game outcome**
  - The oracle is not restricted to an output of {T, F,Ø}, but could instead have an output in the range [0, 1] indicating confidence in the truth or falsity of the proposition.

Rewards and penalties

Reward

Outcome	Voters	Certifiers
T	$\left(\frac{s_{i,j,T}}{s_{TOT,j,T}}\right) \times B_j$	$\left(\frac{\sigma_{i,j,T}}{\sigma_{TOT,j,T}}\right) \times \left(R_T \times \frac{1}{\tau}\right)$
F	$\left(\frac{s_{i,j,F}}{s_{TOT,j,F}}\right) \times B_j$	$\left(\frac{\sigma_{i,j,F}}{\sigma_{TOT,j,F}}\right) \times \left(R_F \times \frac{1}{\tau}\right)$
Ø	0	0

Penalty

Outcome	Voters	Certifiers
T	$s_{i,j,F}$	$\sigma_{i,j,F}$
F	$s_{i,j,T}$	$\sigma_{i,j,T}$
Ø	0	$\sigma_{i,j,F} + \sigma_{i,j,T}$

Monetary flows

- **Submitter fees are used to fund bounties, while the certifier reward pools are initially left empty.**
- **Certifier reward pools are funded by unclaimed bounties and penalties.**
- **It is expected that this method will approach equilibrium where reward pools and bounties are balanced to provide reasonable incentives for both voters and certifiers.**
- **It is adaptable as it can approach a new equilibrium automatically if external conditions change.**

Monetary flows

- **Each time a proposition is decided with voting outcome T, an amount R_T/τ is subtracted from R_T.**
- **If the certification outcome is also T , the funds are used to pay out certifier rewards. Otherwise, the funds are added to R_F.**
- **A symmetric case applies for false outcomes.**
- **It ensures that the reward pools encourage certifiers to certify equal numbers of true and false propositions**
- **Discourage voters from voting with constant T or F values in order to maximize profit without considering the actual propositions.**

Voting Outcomes and Manipulation

D_v	$\lvert P\rvert$	q	$\frac{n}{\lvert P\rvert \cdot D_v}$	P(Specific)	P(Any)
$20 \cdot s_{max}$	100	0.8	0	0.0006	0.0548
$20 \cdot s_{max}$	100	0.8	0.05	0.0028	0.2438
$20 \cdot s_{max}$	100	0.8	0.25	0.1275	≈ 1
$20 \cdot s_{max}$	100	0.95	0	$< 1 \times 10^{-9}$	$< 1 \times 10^{-7}$
$20 \cdot s_{max}$	100	0.95	0.05	$< 1 \times 10^{-6}$	$< 1 \times 10^{-4}$
$20 \cdot s_{max}$	100	0.95	0.25	0.0123	0.7100
$100 \cdot s_{max}$	100	0.8	0	$< 1 \times 10^{-11}$	$< 1 \times 10^{-9}$
$100 \cdot s_{max}$	100	0.8	0.05	$< 1 \times 10^{-8}$	$< 1 \times 10^{-6}$
$100 \cdot s_{max}$	100	0.8	0.25	0.0168	0.8156
$100 \cdot s_{max}$	100	0.95	0	≈ 0	≈ 0
$100 \cdot s_{max}$	100	0.95	0.05	≈ 0	≈ 0
$100 \cdot s_{max}$	100	0.95	0.25	$< 1 \times 10^{-5}$	0.0002

- **D_v: amount of votes needed for a decision**
- **|P|: number of propositions in the list**
- **q: accuracy of honest voters**
- **n/(|P|*D_v): proportion of votes from adversary**

Potential applications

- **Data Availability**
- **Machine Learning and Data Annotation**
- **Adjudication Mechanisms**

My comments

- **Important potential applications**

- **Interesting mechanism**
  - Randomness to reduce likelihood/increase cost for manipulation
  - Certifier and two separate pools to avoid constant voting for same outcome.
  - Some parameters are dynamically determined to adapt external conditions change

My comments

- **Still many concerns are open**
- **How to determine to bounty for each proposition?**
  - Maybe input a new parameter indicating the value of the proposition
  - Bounty needs to be high for high value propositions to attract more honest votes, so that it is more costly to manipulate
  - Bounty cannot be too high so that low accuracy voters are not attracted.
- **How to ensure the honest voters are independent of each other?**
  - Collusion
  - Same information source

Thank you!

Reminder:
Writing assignment 1 is due 1:59pm next Friday