This is the final peer-reviewed accepted manuscript of:

Intelligent Human-input-based Blockchain Oracle (IHiBO)

Proceedings: Special Session on Super Distributed and Multi-agent Intelligent Systems (SDMIS 2022) at the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022), 3 – 5 February 2022, Virtual Event

Author: Liuwen Yu; Mirko Zichichi; Réka Markovich; Amro Najjar

Publisher: SCITEPRESS

Rights / License:

© 2022 SCITEPRESS. This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. Use of this Accepted Version is subject to the publisher's terms of use:

https://portal.insticc.org/AuthorResources/Copyright

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

Intelligent Human-input-based Blockchain Oracle (IHiBO)

Liwen Yu[a][1,2][b], Mirko Zichichi[a][,2,3][c], Réka Markovich[1][d] and Amro Najjar[1][e]
[1]*University of Luxembourg, Luxembourg*
[2]*University of Bologna, Italy*
[3]*Universidad Politécnica de Madrid, Spain*
{*liuwen.yu, reka.markovich, amro.najjar*}*@uni.lu, mirko.zichichi@upm.es*

Keywords: Argumentation, Negotiation, Distributed Ledger Technologies, Blockchain, Smart contracts, Trust services

Abstract: The advent of Distributed Ledger Technologies (DLTs) has paved the way for a new paradigm of traceability in all information systems areas. In the context of decision-making processes, however, DLTs are generally used only to trace the end results. In this work we argue that a reasoning system can be put in place for making these decisions, in order to enhance auditability, transparency, and finally to provide explainability. We propose the Intelligent Human-input-based Blockchain Oracle (IHiBO), a cross-chain oracle that enables the execution and traceability of formal argumentation and negotiation processes, involving the intervention of human experts. We take as reference the decision-making processes of fund managements, as trust is of crucial importance in such "trust services". The architecture and implementation of IHiBO are based on leveraging two-layer DLTs, smart contracts, argumentation and negotiation in a multi-agent setup. Finally, we provide some experimental results that support our discussion, namely that in the use-case we have considered our methodology can increase trust from principals to trusted services.

1 INTRODUCTION

In situations where trust plays a significant role, the decision-making process might be considered as the pinnacle of the engagement between parties. In the case of funds management, for instance, investors choose managers based not only on forecasts of future performance but also on factors such as trust and reliability (Kostovetsky, 2016). Indeed, in these so called "trust services" the fund managers are in the position of a fiduciary acting on behalf of the principal, subject to the overall duty to act in the best interest of the client, i.e. the principal. Fund managers primarily research and determine the best stocks, bonds, or other securities to fit the strategy of the fund, then buy and sell them. The decisions taken by managers affect the principals directly, thus the legislator can and does declare the principal's right to check the fiduciary's relevant activities in order to give some weight to this duty by its intended controlability. However, this might not be so straightforward, as these activities, e.g. securities transactions, are increasingly executed as a collaborative process that involves not only a single fund manager but also other managers, analysts, and external entities that maintain business relationships. The beliefs and assumptions of this diverse group of participants can be influenced by a variety of different background knowledge and in turn shape the decision that leads to the execution of a fund activity. The fund management decision process is characterized by uncertain and changing information, dynamic opportunities, multiple goals and strategic considerations, interdependence among projects, and multiple decision-makers and locations (Cooper et al., 1997). This necessitates a collaborative process that is considered reliable and trustworthy by all participants, protects sensitive information at all times, enables traceability and auditability to maintain accountability, and supports distributed and iterative extending beyond the traditional boundaries of fund management.

With the advent of the use of Distributed Ledger Technologies (DLTs) in finance, some key concerns such as security, transparency and accountability have been addressed. DLTs and smart contracts seems to be able to break the stigma, only apparently immutable, of centrality and of central counterparties

---

[a] These authors contributed equally.
[b] https://orcid.org/0000-0002-7200-6001
[c] https://orcid.org/0000-0002-4159-4269
[d] https://orcid.org/0000-0002-2488-2293
[e] https://orcid.org/0000-0001-7784-6176

(CCPs) (Priem, 2020; Feenan et al., 2020). For instance, in some applications smart contracts can take on a role similar to that previously played CCPs, e.g. acting as a margin calculating agent and taking on the task of transferring collateral. Although in a different way, the smart contract can be used to resolve disputes in the event of non-compliance with payment (Morini, 2017). Decentralized Finance (DeFi), for instance, is a novel P2P financial infrastructure, based on smart contracts, that provides non-custodial, permissionless, openly verifiable and composable operations (Werner et al., 2021). The involvement of DLTs for the fund management, however, does not address the possible trust issues between the principal and the fiduciary: the principal still doesn't have access *why* the given transaction happened and whether it happened, indeed, in his best interest. DLTs are actually used only to trace the output of such a decision-making process. However, a reasoning system put in place for making these decisions could be featured to enhance auditability, transparency, traceability and to provide explainability.

To address these challenges we propose a novel system that leverages DLTs, smart contracts, formal argumentation and negotiation in a multi-agent setup. We argue that formal argumentation can help explain why a claim or a decision is made, in terms of justification, dialogue, and dispute trees (Čyras et al., 2016; Yu et al., 2022). Then, for enabling a conflict-resolution negotiation can be used to determine the quantities, investment timing or other activities. In (Yu et al., 2022), we first proposed the theory of our system, mainly discussing how the aggregation of argumentation and DLTs increases trust. Our contribution in this paper, on the other hand, consists of the implementation of our new system and the evaluation of the feasibility of our proposal. We demonstrate the practicability of the proposed system by implementing a proof-of-concept for the conflict resolution using a private Ethereum Blockchain. To the best of our knowledge, our contribution is the first one to include formal argumentation implemented using smart contracts together with the use of a multi-agent system. This leads to the impossibility of comparison with related works in terms of performance evaluation.

The remainder of the paper is organized as follows. Section 2 presents the background of formal argumentation, negotiation, blockchain and smart contracts which are significant ingredients of our system. Section 3 illustrates how to reach an investment decision with an example. Section 4 introduced IHiBO framework as well as its architecture. Section 5 shows the experiments and the results and Section 6 concludes.

2 BACKGROUND AND RELATED WORKS

In this section we introduce the background of formal argumentation, negotiation, DLTs.

2.1 Formal argumentation

Formal argumentation has achieved significant influence in artificial intelligence (AI), which has the capabilities of representing and reasoning with incomplete and inconsistent information. It can provide various ways for explaining why a decision is made, in terms of dialogue or proofs (Čyras et al., 2016). Dung illustrates an argumentation system consisting of a set of arguments and the relation (attacks) between them (Dung, 1995b). Argumentation semantics are defined later by Baroni and Giacomin for gathering acceptable arguments lying on different criteria (Baroni and Giacomin, 2007), in a way that somehow emulates the way humans tackle such a complex task. When regarding to providing explanation, one of the advantages of argumentation is that the decisions can be mapped to a graphical representation, with predefined attack properties that subsequently will lead to the winning decision and will show the steps that were followed in order to reach it. In the following, we provide the definitions needed for our agent argumentation framework for modeling the decision-making in fund management.

We first generalize argumentation frameworks studied by Dung (1995), which are directed graphs, where the nodes are arguments, and the arrows correspond to the attack relation.

Definition 2.1 (Argumentation framework (Dung, 1995a)). An *argumentation framework* (AF) is a pair $\langle \mathscr{A}, \rightarrow \rangle$ where $\mathscr{A}$ is a set called arguments, and $\rightarrow \subseteq \mathscr{A} \times \mathscr{A}$ is a binary relation over $\mathscr{A}$ called attack. For a set $S \subseteq \mathscr{A}$ and an argument $a \in \mathscr{A}$, we say that S attacks a if there exists $b \in S$ such that b attacks a, a attacks S if there exists $b \in S$ such that a attacks b, $a^- = \{b \in \mathscr{A} | b \text{ attacks a}\}$, $S^-_{out} = \{a \in \mathscr{A} \backslash S | \text{ a attacks S}\}$.

Dung's admissibility-based semantics is based on the concept of defense. A set of arguments defends another argument if they attack all its attackers.

Definition 2.2 (Admissible (Dung, 1995a)). Let $\langle \mathscr{A}, \rightarrow \rangle$ be an AF. $E \subseteq \mathscr{A}$ is *conflict-free* iff there are no arguments a and b in E such that a attacks b. $E \subseteq \mathscr{A}$ *defends c* iff for all arguments b attacking c, there is an argument a in E such that a attacks b. $E \subseteq \mathscr{A}$ is *admissible* iff it is conflict-free and defends all its elements.

Baroni and Giacomin then define semantics as a function from argumentation frameworks to sets of subsets of arguments.

Definition 2.3 (Dung semantics (Baroni and Giacomin, 2007))**.** A Dung semantics is a function σ that associates with an argumentation framework $AF = \langle \mathscr{A}, \rightharpoonup \rangle$, a set of subsets of $\mathscr{A}$, the elements of $\sigma(AF)$ are called extensions.

Dung distinguishes several definitions of extension.

Definition 2.4 (Extensions (Dung, 1995a))**.** Let $\langle \mathscr{A}, \rightharpoonup \rangle$ be an AF. $E \subseteq \mathscr{A}$ is a *complete extension* iff it is admissible and it contains all arguments it defends, i.e., $E = \{a | E \text{ defends } a\}$. $E \subseteq \mathscr{A}$ is a *grounded extension* iff it is the smallest (for set inclusion) complete extension. $E \subseteq \mathscr{A}$ is a *preferred extension* iff it is a largest (for set inclusion) complete extension. $E \subseteq \mathscr{A}$ is a *stable extension* iff it is conflict-free and it attacks each argument which does not belong to E.

An agent argumentation framework extends an argumentation framework with a set of agents and a relation associating arguments with agents. Note that an argument can belong to one agent or multiple agents.

Definition 2.5 (Agent argumentation framework (Yu and van der Torre, 2020))**.** An *agent argumentation framework* (AAF) is a 4-tuple $\langle \mathscr{A}, \rightharpoonup, \mathscr{S}, \sqsubset \rangle$ where $\mathscr{A}$ is a set of arguments, $\rightharpoonup \subseteq \mathscr{A} \times \mathscr{A}$ is a binary relation over $\mathscr{A}$ called attack, $\mathscr{S}$ is a set of agents or sources, $\sqsubset \subseteq \mathscr{A} \times \mathscr{S}$ is a binary relation associating arguments with agents. $\mathscr{A}_\alpha = \{a \in \mathscr{A} | a \sqsubset \alpha\}$ for all arguments that belong to agent α, $\mathscr{S}_a = \{\alpha | a \sqsubset \alpha\}$ for all agents that have argument a.

2.1.1 Social agent semantics

For the decision making of fund management, we introduce so-called social semantics, which is based on a reduction to preference-based argumentation by for each argument counting the number of agents that have the argument (Yu et al., 2020). It thus interprets agent argumentation as a kind of voting, as studied in social choice theory or judgment aggregation, this is also the most closed to fund management.

We next give the definition of a preference-based argumentation framework.

Definition 2.6 (Preference-based argumentation framework (Kaci and van der Torre, 2008))**.** A preference-based argumentation framework (PAF) is a 3-tuple $\langle \mathscr{A}, \rightharpoonup, \succ \rangle$ where $\mathscr{A}$ is a set of arguments, $\rightharpoonup \subseteq \mathscr{A} \times \mathscr{A}$ is a binary attack relation, $\succ$ is a partial order (irreflexive and transitive) over $\mathscr{A}$, called preference relation.

There are different reductions of preference have been introduced (Amgoud and Vesic, 2014; van der Torre and Vesic, 2017). We refer to those papers for an explanation and motivation, and we choose one of the reductions in our use case below which satisfies the essential conflict-free principle analyzed in (Yu et al., 2020).

Definition 2.7 (Reductions of PAF to AF (PR))**.** Given an $PAF = \langle \mathscr{A}, \rightharpoonup, \succ \rangle$: PR(PAF)=$\langle \mathscr{A}, \rightharpoonup' \rangle$, where $\rightharpoonup' = \{a \rightharpoonup' b | a \rightharpoonup b, b \not\succ a, \text{ or } b \rightharpoonup a, \text{ not } a \rightharpoonup b, a \succ b, \text{ or } a \rightharpoonup b, \text{ not } b \rightharpoonup a\}$.

In social agent semantics, an argument is preferred to another argument if it belongs to more agents. The reduction from AAF to PAF is used as an intermediary step for social agent semantics.

Definition 2.8 (Social Reductions of AAF to PAF (SAP))**.** Given an $AAF = \langle \mathscr{A}, \rightharpoonup, \mathscr{S}, \sqsubset \rangle$, SAP(AAF)=$\langle \mathscr{A}, \rightharpoonup, \succ \rangle$ with $\succ = \{a \succ b || \mathscr{S}_a| > |\mathscr{S}_b|\}$.

Definition 2.9 (Social Reductions of AAF to AF (SR))**.** Given an $AAF = \langle \mathscr{A}, \rightharpoonup, \mathscr{S}, \sqsubset \rangle$, SR(AAF)=PR(SAP(AAF)), PR is the reduction of PAF to AF, where the semantics $\delta(AAF) = \sigma$(SR(AAF))= σ(PR(SAP(AAF))).

2.2 Autonomous Agents and Negotiation

An agent is a software program that acts on behalf of another actor (often a human user) to perform a task or achieve a given goal (Wooldridge, 2009). Agents are designed to be bound to individual perspectives and this makes them good candidates to represent the subjectivity and nuances of different expert opinions. Multi-agent systems (Weiss, 2013) provide a distributed platform capable of implementing intelligence in decentralized ecosystems where agents are capable, using well-established conflict-resolution mechanisms (e.g. negotiation), of helping the different stakeholders finding agreements that satisfy their often conflicting interests.

Negotiation, in particular, is the process by which a joint decision is made by two or more parties, that firstly verbalize contradictory demands and then move towards agreement by a process of concession making or search for new alternatives (Pruitt, 2013). The problem being negotiated, or the topic under discussion (e.g. car purchase) can be usually divided into issues (also called attributes). Negotiators may not only disagree on the value assigned to each issue, the priority given to each issue can differ from one negotiator to another and hence this can be a source of both divergence and convergence (Pruitt, 2013).

Automated negotiation is one taking place among autonomous agents through a protocol. The latter is the set of rules that governs the interactions during a negotiation session (also called a thread). Whereas the negotiation protocol defines what is the set of possible actions that can be taken during a negotiation session, an agent has a decision model (Faratin et al., 1998) that allows the agent to (i) evaluate the value of an offer received from the opponent (e.g., using a utility function), (ii) decide whether it is acceptable , and (iii) determine what to do next (known as the negotiation strategy). Automated negotiation has been applied to solve conflicts and reach agreements in several domains including cloud and service provisioning (Najjar et al., 2013), smart grid and power distribution (Tom et al., 2020), and trading and stock market (Wellman et al., 2007). Compared with human negotiation, autonomous agent negotiation is efficient in contexts where the number of issues under negotiation is intractable for human users, or in one-to-many (Mansour and Kowalczyk, 2011) or many-to-many negotiation (An et al., 2009) settings in which the numbers of negotiators makes it difficult for humans to keep track of the evolution of the negotiation process. Therefore, autonomous agents can offload these tasks from the human expert shoulders, assist them in formulating their preferences, and help reach optimal solution that can be otherwise inaccessible to human negotiators with the agent assistance.

2.3 Blockchain and Smart contracts

With the launch of Bitcoin in 2008 (Nakamoto, 2008), the technology underpinning it is becoming increasingly popular, i.e. the blockchain, which is a part of realm of DLTs. DLTs consists of a network of nodes that maintain a distributed ledger by following the same protocol, and, in the case of the blockchain, the ledger is organized into chronologically ordered blocks where each block is sequentially linked to the previous one (Nakamoto, 2008). Thus, the blockchain is cryptographically guaranteed to be tamper-proof and unforgeable, enabling the creation of "trusted" mechanism exploitable by several users in a distributed environment and without the need for third party intermediaries. Smart contracts are instructions stored in blockchain and automatically triggered once the predefined condition is met (Buterin et al., 2013). Utilizing smart contracts allows us to employ blockchain far beyond monetary transactions (Kurt Peker et al., 2020; Zheng et al., 2020; Zichichi et al., 2020b). However, smart contracts cannot fetch data from off-chain themselves of whose the possibility usage is obviously limited, since the many smart contract applications would require real time information from the network external world. In this context, oracles emerge as a bridge that connects the blockchain network and the "outside" world, providing the ability to retrieve, verify and digest the data into smart contracts. Oracles can be implemented as software, hardware or human (Beniiche, 2020). In all cases their off-chain execution is either centralized, i.e. coming from a single source, or decentralized, consensus-based multitude of sources. The latter case can be also implemented as a cross-chain oracle, where a system in a blockchain, i.e. the main-chain, can validate and read events and/or state from another blockchain, i.e. a sidechain (Buterin, 2016).

2.4 Related Works

In the remainder of this paper we will discuss the implementation of a system for the decision-making based on formal argumentation, autonomous negotiation, blockchain, smart contracts, and oracles, thus leading to an overview of the related works from multiple perspectives. Indeed, to the best of our knowledge, there is no mature work on the adoption of argumentation in the financial world, nor the combination of argumentation and autonomous negotiation in blockchains using smart contracts. The only work we can find regarding the use of argumentation as a convincing tool in order to gain the stakeholders' support and trust is the one proposed by Palmieri (Palmieri, 2009). Focusing on formal argumentation only, several influential works discuss its role in providing trustworthy systems (Matt et al., 2010; Parsons et al., 2010; Tang et al., 2010). Parsons et al suggest argumentation might play a role which tracks the origin of information used in reasoning, thus it can provide provenance in trust (Parsons et al., 2010). Later the same authors develop a general system of argumentation that can represent trust information, and be used in combination with a trust network, using the trustworthiness of the information sources as a measure of the probability that information is true (Tang et al., 2010).

The adoption of blockchain and DLT has been under consideration and debating for several years both from economic and legal aspects (Priem, 2020) and many proposals on building a DLT-based securities has been conducted. However, most of them discuss the transaction process including how to use these technologies for clearing and settlement which are process after securities trading (Oprea et al., 2020; Wall and Malm, 2016). In our work, on the other hand, we pay attention to the pre-trading phase, particularly in fund management context, where the invest decisions made by the trust services are extremely crucial to investors. Nonetheless, we also refer to the oracle process seen in the previous subsection and many related works are built on this. Human oracles, i.e. the ones requiring an input which involves human intervention, are rarely applied (Damjan, 2018). The rare existing ones are deployed in applications with binary inputs, i.e., they only take input by one of two possibilities, typically "yes" or "no" (Nelaturu et al., 2020), such as ASTRAEA (Adler et al., 2018) that leverages human actions through a voting game. Augur (Beniiche, 2020; Peterson and Krug, 2015) is a decentralized oracle that needs specific human users obligated by *Reputation Tokens* to report outcomes at specific times, users who report incorrect results would be subject to a dispute process, and then through a consensus algorithm to calculate the results. A part from those, many services and projects have been already established for the implementation of oracles. Provable (Provable Things Limited, 2019), before known as Oraclize, is an oracle service that provides a data transport-layer for smart contracts to fetch external data from Web APIs. The peculiarity of such platform is that it is blockchain agnostic and that can serve requests coming from multiple DLT instances. Chainlink (Ellis et al., 2017) offers a general-purpose framework to build a decentralized oracle network on the Ethereum blockchain. Its main purpose is to provide reliable data tamper-proof input and output for smart contracts by accessing data resources. Gnosis (GnosisDAO, 2017) approach is different, but ultimately resorts oracles as well. Gnosis mainly derives information from centralized oracle services, but enables the users to challenge those results.

3 CONFLICT RESOLUTION USE CASE

In this section we use a simplified example to illustrate how we use agent abstract argumentation and autonomous negotiation for dealing with conflicting information raised by agents.

The process of decision-making in fund managements fits well with argumentation theory in artificial intelligence. The decision can be seen as being based on arguments and counter-arguments. Argumentation, as the result, can be useful for deriving decisions and explaining a choice already made. Managers provide their arguments from their own research to identify promising stocks with different level of accuracy and thereby make different portfolio choices which are likely to be incomplete and inconsistent. The fictitious simple example (the real life cases would be much more complex) is as follows. Manager α and β hold the arguments a: *To buy the stocks, since the company just donated to charities that is beneficial to good commercial reputation*, and argument c: *To buy the stocks, since the company has started to use a new promising technology which will develop the sale performance.* However, another manager γ at the same time is against buying the stocks, he holds the arguments b and d, b is *To sell the stocks, since there is evidence that the leader is under accusations of charity fraud*, and d is *To sell the stocks, since the company now has poor sale performance.*

Based on the above, we can build an agent argumentation framework on the left side of Figure 1, $AAF = \langle \mathscr{A}, \rightarrow, \mathscr{S}, \sqsubset \rangle$ where $\mathscr{A} = \{a, b, c, d\}$, $\rightarrow = \{$(a,b),(b,a), (a,d),(d,a),(b,c),(c,b),(c,d),(d,c)$\}$, $\mathscr{S} = \{\alpha, \beta, \gamma\}$, $\sqsubset = \{$(a,α),(a,β),(c,α),(c,β),(b,γ),(d,γ)$\}$. Since $|\mathscr{S}_a| > |\mathscr{S}_b|$, $|\mathscr{S}_a| > |\mathscr{S}_d|$, $|\mathscr{S}_c| > |\mathscr{S}_b|$, $|\mathscr{S}_c| > |\mathscr{S}_d|$, $a \succ b$, $a \succ d$, $c \succ b$ and $c \succ d$, we get the corresponding PAF showing in the middle of Figure 1, and giving the four reductions from PAF to AF, we have the AF on the right side of Figure 1. Then we can calculate the only acceptable set $\{a, c\}$ which is the only grounded, complete, preferred and stable extension. The set tells the final decision is to buy the stocks.

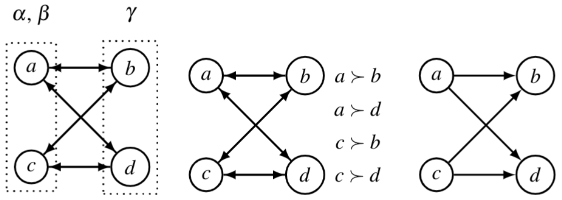

Figure 1: Social Reduction.

One thing needs to be noticed: argumentation does not always provide a unique outcome. People need to select the desired semantics based on various reasoning flavour (Baroni et al., 2011). On the other hand, depending on the decision making process, different protocols can be specified in advance for such cases: e.g. to roll back or to assign weights to the arguments and the relation among them. After reducing to AF and calculating the acceptable set, indeed, when the outcome results in the decision to buy the stocks, the next problem could become the numbers of stocks to buy and the buy timing. Here the computational automated negotiation comes into play. To illustrate how it works, we give an example of the negotiation sequence based on the quantities of stocks to buy. The negotiation process is based on the alternating offer protocol (Rubinstein, 1982). Agents can bid new offers to the opponent ($Offer()$ function). When receiving an offer, an agent can accept it using $accept()$ function or reject it and propose a counter-offer (with the $CounterOffer()$ function). In the example, we have a manager A, i.e., agent A, and manager B, i.e., agent B. Agent A proposes to buy 1000 stocks at the price of 151$, while agent B counteroffers to buy 1200 stocks at the price of 145$, then agent A proposes to buy 1150 stocks at the price of 148$. The final offer given by A is accepted by both parties which means they come to an agreement.

4 INTELLIGENT HUMAN-INPUT-BASED BLOCKCHAIN ORACLE (IHIBO) FRAMEWORK

In this section we present the details of the Intelligent Human-input-based Blockchain Oracle Framework. IHiBO is a cross-chain oracle that enables the execution and traceability of argumentation and negotiation processes, involving the intervention of human experts.

4.1 Architecture

The framework is centered around a layer two solution that moves the oracle's off-chain[1] processes to a sidechain. In particular, this sidechain consists of a chain where smart contracts are executed and data stored, and of a mainchain where commitments are periodically stored for the framework security and where the result of the conflict resolution is executed. Before going into the architecture details we describe the roles of the actors involved in the architecture, with reference to Figure 2:

- **Human Expert**, the one who takes most of the decisions and that gives inputs to the agent;
- **Agent**, the one that can assist human experts in formulating their preferences and to reach optimal solutions; these are also the ones that directly interact with the sidechain.
- **Public DLT Node**, the one that takes part to the mainchain consensus mechanism and that is external to the sidechain; this actor receives transactions to be stored in the mainchain, i.e. a DLT full-node (Nakamoto, 2008; Buterin et al., 2013).

For the architecture of the cross-chain oracle, we refer to a layer two solution because[2] : (i) the first layer includes a public permissionless DLT, i.e. mainchain, while (ii) the second layer consists of a private permissioned DLT, i.e. sidechain. The advantage of using a public permissionless DLT solution is that it usually offers a high level of security and decentralization (De Angelis, 2018), needed to completely trace and verify processes with trust, e.g. Ethereum (Buterin et al., 2013). The usual drawbacks are that storing large quantities of data on such a chain is expensive (Kurt Peker et al., 2020) and that scalability is often compromised for some features, such as smart contracts execution (Sedlmeir et al., 2021; Zichichi et al., 2020b) . Therefore, in the framework, a public permissionless DLT maintained by public DLT nodes is used as the mainchain solely to store "commitments" (explained later in this section) and to execute the business logic arising as a result of a conflict resolution, e.g., sell stock. On the other hand, the conflict resolution process is executed mainly on the second layer, thanks to the use of the sidechain.

In the second layer a network of agents and/or other nodes maintain the sidechain. We refer to a private permissioned DLT for the framework sidechain, where only some actors have the permission to read and write to the ledger, e.g. agents. Private permissioned DLTs solve the public permissionless issues of: (i) the publicity of information that would clash with trade secrets and privacy, as only allowed actors can read from the ledger; (ii) expensiveness and scalability, as permissioned DLTs protocols can be designed ad-hoc to specifically address these issues. However, the level of security of private permissioned solution decreases in respect to public permissionless ones, due to the fact that they generally are less decentralized and that usually use more efficient but less secure consensus mechanisms (Sedlmeir et al., 2021; De Angelis, 2018).

The mainchain and sidechain are tied together in the framework by the use of periodical commitments. A commitment consists of storing in the mainchain the result of an hash function applied to the state of the sidechain at a certain point in time. This would allow to store data that cannot be tampered in the mainchain and to allow its verification. At the same time, thanks to the hash function, the privacy of information stored in the sidechain is maintained, while assuring that any data corruption will be detected (Gudgeon et al., 2020), i.e. the hash result will change. Indeed, through the use of commitments, once the nodes operating the sidechain reveal part of (or all of) the in-

---

[1] The reference to "chain" will always be to the main chain thorough the text, opposed to the "sidechain" that will be always called as such.

[2] We investigated this aspect in a previous work (Yu et al., 2022), both from a practical and legal point of view

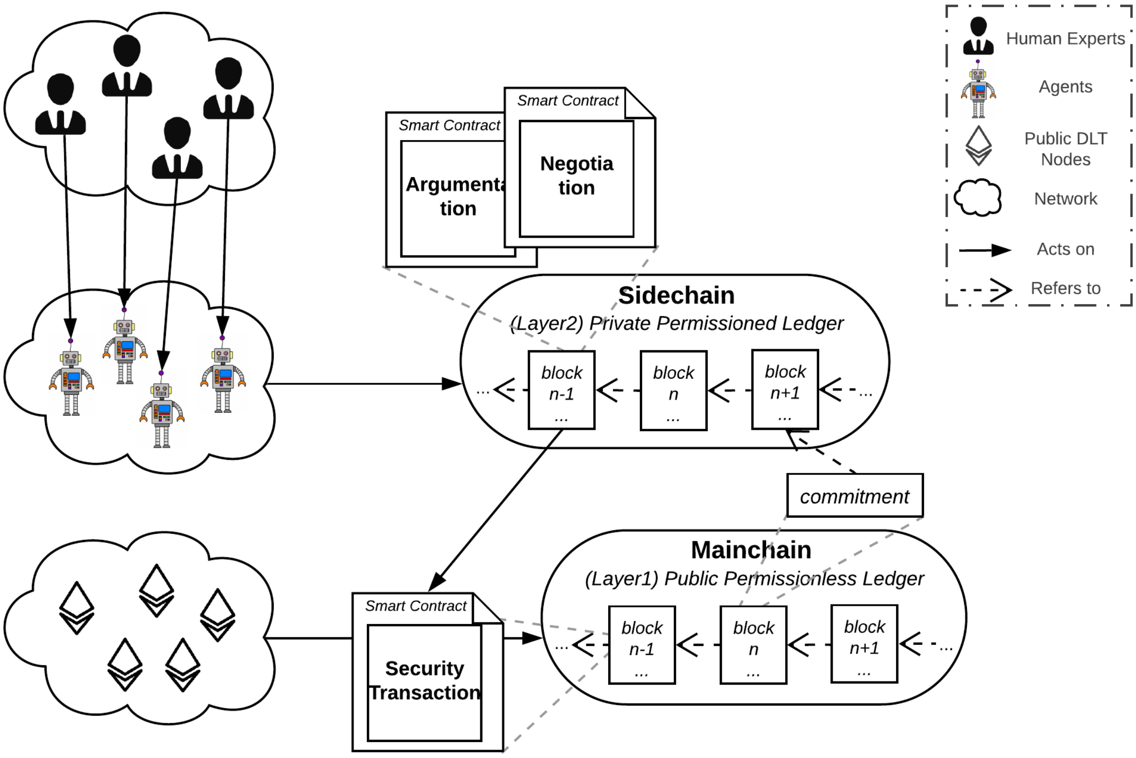

Figure 2: IHiBO Framework Architectures

formation stored in the sidechain to possible auditors, the latter can apply the hash function to the data received and check that the obtained result is equal to the hash stored in the mainchain (Singh et al., 2020).

4.2 Implementation

For the IHiBO Framework implementation we refer to Ethereum and to its smart contract specification (Buterin et al., 2013).

4.2.1 Mainchain

In particular for the mainchain, we leverage the Ethereum public blockchain and the functions exposed by its network nodes for creating and/or interacting with smart contracts. In the Ethereum blockchain some applications built through the use of smart contracts, i.e. decentralized applications (dApps), are already been developed for the execution of securities transactions (Pop et al., 2018). The Ethereum protocol allows smart contracts intercommunication, thus the framework we present is meant to include a dedicated smart contract that "bridges" the output of the execution in the sidechain, e.g. a conflict resolution, to a smart contract deployed to the mainchain[3]. We refer to this smart contract as the "SecurityTransaction", and its implementation mostly depends on the on-chain business process it interacts with. For instance, Decentralized Finance (DeFi)[4] protocols such as Decentralized Exchanges (DEX) are already been provided in the Etehreum blockchain for enabling anyone to engage in non-custodial exchange of on-chain digital assets, e.g. tokens (Werner et al., 2021). Smart contracts that implements such DEXes can be directly invoked for swapping tokens and cryptocurrencies depending on their value (International Token Standardization Association, 2021). An instance of a SecurityTransaction would be a smart contract that includes a method that directly invokes a DEX smart contract for executing a token swap. This can be seen as the direct selling/buying of traditional stocks that have been "tokenized" (International Token Standardization Association, 2021; Bhandarkar et al., 2019).

---

[3] Due to lack of space, we do not go into the details of this bridge implementation, but an instance would be the atomic execution of transactions across chains (Robinson and Ramesh, 2021).

[4] DeFi is a term that refers to smart contract based financial infrastructures that are non-custodial, permissionless, openly verifiable and composable (Werner et al., 2021).

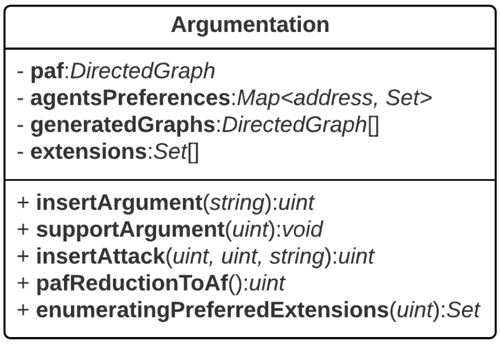

Figure 3: Argumentation smart contract class diagram

4.2.2 Sidechain

For what concerns the sidechain, any implementation of a permissioned smart contract enabled DLT is suitable for the framework we proposed. In our implementation, we make use of an Ethereum blockchain distributed among nodes in a private permissioned network. In this case, the consensus algorithm adopted by the network does not necessarily have to be the Proof-of-Work (Nakamoto, 2008; Buterin et al., 2013), but, in order to provide a faster service, used the Proof-of-Authority (PoA) consensus algorithm (Toyoda et al., 2020). PoA, indeed, does not depend on solving mathematical problems, and to issue a new block this one must be signed by the majority of the authorities, i.e. the nodes that are explicitly authorized to create new blocks and secure the blockchain.

The main purpose of this sidechain is to support smart contracts whose execution log can be later audited. Thus, we implemented two smart contract specifications for executing conflict resolution processes, however many others can be implemented following the Ethereum smart contracts specification (Buterin et al., 2013).

Argumentation Smart Contract We implemented a smart contract for providing a PAF (Section 2.1.1) to the agents that operates in the sidechain.

- A data structure within the smart contract allows to create and manage a directed graph, where nodes are arguments and edges are attack relations. Each agent can add an argument (*insertArgument()*) and its attacks (*insertAttacks()*) to the graph or set as "preferred" an already existing argument (*supportArgument()*).
- Arguments are handled through their id and the metadata associated to it, i.e. the actual argument text, can be stored directly on the ledger or outside and referenced through a hash pointer.
- After a predefined time period needed for completing the PAF, reductions of PAF to AF can be invoked and executed directly by the smart contract (*pafReductionToAfPr()*). The result of invoking this method is a new directed graph representing the AF.
- Finally, an extension can be found for the previously obtained AF, by invoking another method (*enumeratingPreferredExtensions()*). The implementation of this method is based on the algorithm found in (Nofal et al., 2014) for listing all preferred extensions of an AF (Algorithm 1). This possibly provides a set of arguments that lead to a final decision.

Negotiation Smart Contract We implemented a smart contract that concludes the conflict resolution (Section 2.1.1) with a negotiation on the arguments provided by the argumentation process.

- A data structure within the smart contract holds the data needed during a negotiation thread. A list of such structures enables agents to interact for automated negotiations on several issues. Each agent can start a new negotiation with another agent for a specific set of issues (*newNegotiation()*).
- Each agent has its own decision model executed off-chain, that allows this to evaluate the value of an offer received from another agent, e.g. a time dependent tactic (Faratin et al., 1998).
- Based on the evaluation, the agent can invoke the smart contract to make a new offer (*newOffer()*) providing a new set of values related to the issues, accepting (*accept()*) the other agent's offer, or refusing it (by not providing input to the smart contract).
- The invocation of the smart contract method for accepting the offer can directly enact the process of interaction with the SecurityTransaction smart contract on the mainchain (Robinson and Ramesh, 2021).

5 Experiments and Results

We developed a cross-chain oracle prototype to test the feasibility of the use of smart contracts for conflict resolution and in here we present the results of some experiments based on two assumptions. Firstly, we are not interested in testing out the performances in terms of transaction per seconds and scalability for public permissionless DLTs, since these

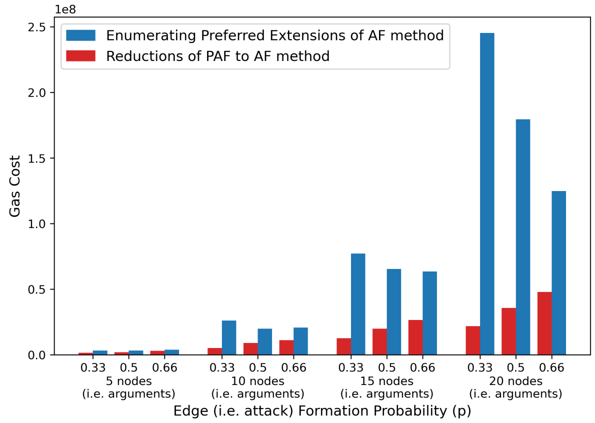

Figure 4: *pafReductionToAfPr()* and *enumeratingPreferredExtensions()* methods gas cost

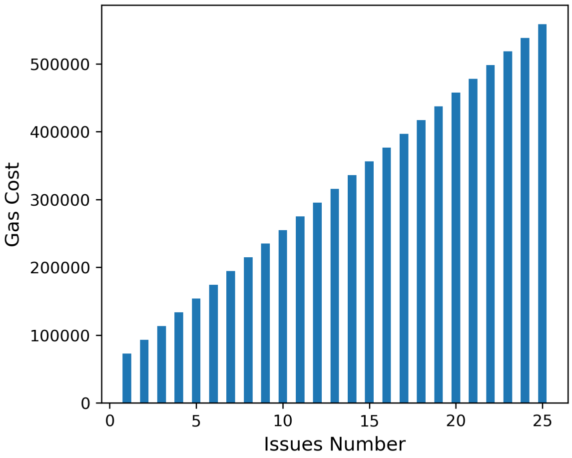

Figure 5: Negotiation newOffer() method gas cost

have already been studied in literature for similar use cases (Sedlmeir et al., 2021; De Angelis, 2018; Kurt Peker et al., 2020; Zichichi et al., 2020a). Indeed, these results have already impacted the IHiBO framework design by limiting the issuing of transaction to the mainchain only for periodic commitments (Yu et al., 2022). Secondly, for what regards the sidechain, performances depends on the specific implementation used by the actors in a specific use case. In our implementation we used an Ethereum private network using PoA and it has been shown that, with optimal configuration, it can reach up to 1000 transactions per second (Toyoda et al., 2020).

Therefore, our focus is on the execution of the smart contracts that we described in the implementation section (4.2), with regards to the argumentation and negotiation processes. We measure our experiments in terms of gas cost, following the Ethereum protocol (Buterin et al., 2013). Gas is a unit that measures the amount of computational effort that takes to execute operations in Ethereum smart contracts. Thus, the higher the gas cost for a method, the more intense the computation of a blockchain node to execute the method's instructions.

The complete experiments dataset and the reference software can be found in (Zichichi, 2021), following the FAIR data principles for access and reuse of models (Wilkinson et al., 2016).

5.1 Results

Table 1 shows the gas costs for the execution of the Argumentation and Negotiation smart contracts methods, taking as input the data of the example in Section 3. These results give an indicative idea relative to the the different methods executions, since their latency (i.e., the time between submitting a transaction that invoke such methods and the actual insertion to the blockchain) depends heavily on the blockchain's consensus mechanism. For instance, considering a as the arguments number and n as the agents number, the *supportArgument()* method is much less expensive than the *enumeratingPreferredExtensions()*, but it is executed up to $\leq a \times (n-1)$ times while the latter only 1 time.

In Figure 4 it is shown the increase of the gas cost while varying the AF. For each arguments number a taken into consideration, i.e. $5, 10, 15, 20$, some graphs representing a different AF have been created randomly. In these graphs, the edge connecting any two nodes, i.e. an attack in the AF, was firstly formed with a probability of 0.33, then 0.5 and finally 0.66. For each probability value, 20 random graph were created and the average of gas cost for invoking the methods *pafReductionToAfPr()* and *enumeratingPreferredExtensions()* was computed. For the latter method, results show that the gas cost depends heavily on the arguments number a, as with the increase of a the gas cost increments exponentially. At the same time, however, incrementing the edge formation probability p leads to a decrease of the gas cost. Results for the *pafReductionToAfPr()* method show a much less dramatic increment of gas cost with the increase of a, but here the increase of edges number leads to an increase of gas cost instead of a decrease. The minimum value for the *enumeratingPreferredExtensions()* method gas cost is $\sim$ 1.2 million gas units, while the maximum is $\sim$ 528 million gas units. For what concerns the other method, $\sim$ 0.6 million gas units is the minimum and $\sim$ 51 million the maximum.

Finally, we provide the results of the measurement of the gas cost for the *newOffer()* method of the Nego-

Table 1: Gas Cost

Smart Contract	**Method**	**Occur rency**	**Gas Cost**
Argumentation	**insertArgument()**	a	157470
Argumentation	**supportArgument()**	$\leq a \times (n-1)$	80491
Argumentation	**insertAttack()**	$\leq a \times (a-1)$	215011
Argumentation	**pafReductionToAfPr()**	1	1877277
Argumentation	**enumeratingPreferred Extensions()**	1	1412065
Negotiation	**newNegotiation()**	1	104961
Negotiation	**newOffer()**	t	52438
Negotiation	**accept()**	1	64211

tiation smart contract. In this case, we implemented two agents negotiating using a time dependent tactic, as in (Faratin et al., 1998), with two different set of starting conditions and maximum values. The number of new offers t proposed by each agent cannot be known a priori because it depends on the specific strategy of the agent. For this reason we measured the impact of the issues number j on the gas cost. Figure 5, indeed, shows that the latter increases linearly with the former, due to the increasingly storage demand.

5.2 Discussion

Generally speaking, we experienced a strong dependence on the arguments number for the increase of the gas cost. This was expected, as more arguments means a more complex argumentation framework to deal with. The use of a private Ethereum PoA network allows to limit the latency based on the results obtained in (Toyoda et al., 2020). Assuming one invocation per transaction, methods such as *insertArgument()* or *insertAttack()* easily fall into the 1000 transactions per second range. However, *pafReductionToAfPr()* and *enumeratingPreferredExtensions()* methods require more computation and might limit the transactions per second number. Regarding the Negotiation contract, the *newOffer()* method might highly influence performances when the number of issues is > 25.

The use of sidechain allows agents to operate without too many performance limitations, while maintaining a level of traceability that allows full auditing by an inspector. These results would not have been possible in a permissionless DLT. In fact, for example, in Etehreum the limit of gas cost per block is currently (at the time of writing this paper) 15 million gas units. This means that, not only some transactions could not be executed (e.g. *enumeratingPreferredExtensions()* with an AF with > 20 arguments), but also that the latency between operations would be very high because currently, in the Ethereum network, a block is created every 10/15 seconds on average.

6 Conclusions

In this paper, we have proposed an integrated framework which incorporates formal argumentation and negotiation within a blockchain environment. These techniques have distinctive features that complement each other. They together make the decision-making processes of fund management transparent and traceable. As a result, our methodology enhances trust from principals to trust services, which is grounded when knowing how the fund management make decisions sufficiently well so that the behavior of managers can be understood and predicted more accurately. Our motivation came from trust services, so we explained our idea in a fund management scenario, but our proposal is not bound to this domain. Also, the research on oracles is still in its infant stage, there are multiple pressing questions and challenges for the future. To the best of our knowledge, this is the first study where such a framework that incorporates argumentation and negotiation, i.e. IHiBO, is implemented using a cross-chain oracle and smart contracts. The results of our experiments shows that the use of a two layer blockchain architecture, allows to securely operate without too many performance limitations, while maintaining a high level of traceability that allows to audit trust services operations.

One follow up possible work is to provide a high level of adaptability in the decisions of the fund management, e.g. to define different investment scenarios according to the investors' preferences, attitude and the financial environment. Another possible work could be to investigate on the integration of consensus mechanisms for a layer two solution to the dispute resolution phase, in order to narrow the gap between blockchain and argumentation as well as negotiation, since there is no specialized blockchain yet that has a protocol that integrates reasoning.

Lastly, we also plan to rely on the recent advances of the domain of Explainable AI to explore how we can make the decision making process presented in this paper explainable for different types of users (experts, non-experts, etc.) and for different purposes (e.g. transparency, debugging, etc.).

ACKNOWLEDGEMENTS

This work has received funding from the EU H2020 research and innovation programme under the Marie Skłodowska-Curie Actions Innovative Training Networks European Joint Doctorate grant agreement No 814177 Law, Science and Technology Joint Doctorate - Rights of Internet of Everything.

REFERENCES

Adler, J., Berryhill, R., Veneris, A., Poulos, Z., Veira, N., and Kastania, A. (2018). Astraea: A decentralized blockchain oracle. In *2018 IEEE iThings and IEEE GreenCom and IEEE CPSCom and IEEE SmartData*, pages 1145–1152. IEEE.

Amgoud, L. and Vesic, S. (2014). Rich preference-based argumentation frameworks. *International Journal of Approximate Reasoning*, 55(2):585–606.

An, B., Gatti, N., and Lesser, V. (2009). Extending alternating-offers bargaining in one-to-many and many-to-many settings. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 423–426. IEEE.

Baroni, P., Caminada, M., and Giacomin, M. (2011). An introduction to argumentation semantics. *The knowledge engineering review*, 26(4):365–410.

Baroni, P. and Giacomin, M. (2007). On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700.

Beniiche, A. (2020). A study of blockchain oracles. *arXiv preprint arXiv:2004.07140*.

Bhandarkar, V. V., Bhandarkar, A. A., and Shiva, A. (2019). Digital stocks using blockchain technology the possible future of stocks? *International Journal of Management (IJM)*, 10(3).

Buterin, V. (2016). Chain interoperability. *R3 Research*.

Buterin, V. et al. (2013). Ethereum white paper.

Cooper, R. G., Edgett, S. J., and Kleinschmidt, E. J. (1997). Portfolio management in new product development: Lessons from the leaders—i. *Research-Technology Management*, 40(5):16–28.

Čyras, K., Satoh, K., and Toni, F. (2016). Explanation for case-based reasoning via abstract argumentation. In *Computational Models of Argument*, pages 243–254. IOS Press.

Damjan, M. (2018). The interface between blockchain and the real world. *Ragion pratica*, pages 379–406.

De Angelis, S. (2018). Assessing security and performances of consensus algorithms for permissioned blockchains. *arXiv preprint arXiv:1805.03490*.

Dung, P. M. (1995a). On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357.

Dung, P. M. (1995b). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357.

Ellis, S., Juels, A., and Nazarov, S. (2017). Chainlink a decentralized oracle network.

Faratin, P., Sierra, C., and Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182.

Feenan, S., Heller, D., Lipton, A., Morini, M., Ram, R., Sams, R., Swanson, T., Yong, S., and Zalles, D. B. (2020). Decentralized financial market infrastructures. *The Journal of FinTech, Forthcoming*.

GnosisDAO (2017). Gnosis whitepaper.

Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., and Gervais, A. (2020). Sok: Layer-two blockchain protocols. In *International Conference on Financial Cryptography and Data Security*, pages 201–226. Springer.

International Token Standardization Association (2021). List of synthetix derivative tokens.

Kaci, S. and van der Torre, L. (2008). Preference-based argumentation: Arguments supporting multiple values. *International Journal of Approximate Reasoning*, 48(3):730–751.

Kostovetsky, L. (2016). Whom do you trust?: Investor-advisor relationships and mutual fund flows. *The Review of Financial Studies*, 29(4):898–936.

Kurt Peker, Y., Rodriguez, X., Ericsson, J., Lee, S. J., and Perez, A. J. (2020). A cost analysis of internet of things sensor data storage on blockchain via smart contracts. *Electronics*, 9(2):244.

Mansour, K. and Kowalczyk, R. (2011). A meta-strategy for coordinating of one-to-many negotiation over multiple issues. In *Foundations of Intelligent Systems*, pages 343–353. Springer.

Matt, P.-A., Morge, M., and Toni, F. (2010). Combining statistics and arguments to compute trust. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 209–216. Citeseer.

Morini, M. (2017). Managing derivatives on a blockchain. a financial market professional implementation. *A Financial Market Professional Implementation (May 5, 2017)*.

Najjar, A., Serpaggi, X., Gravier, C., and Boissier, O. (2013). Multi-agent negotiation for user-centric elasticity management in the cloud. In *2013 IEEE/ACM*

6th International Conference on Utility and Cloud Computing, pages 357–362. IEEE.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Nelaturu, K., Adler, J., Merlini, M., Berryhill, R., Veira, N., Poulos, Z., and Veneris, A. (2020). On public crowdsource-based mechanisms for a decentralized blockchain oracle. *IEEE Transactions on Engineering Management*, 67(4):1444–1458.

Nofal, S., Atkinson, K., and Dunne, P. E. (2014). Algorithms for argumentation semantics: labeling attacks as a generalization of labeling arguments. *Journal of Artificial Intelligence Research*, 49:635–668.

Oprea, S.-V., Bâra, A., and Andreescu, A. I. (2020). Two novel blockchain-based market settlement mechanisms embedded into smart contracts for securely trading renewable energy. *IEEE Access*, 8:212548–212556.

Palmieri, R. (2009). Regaining trust through argumentation in the context of the current financial-economic crisis. *Studies in Communication Sciences*, 9(2):59–78.

Parsons, S., McBurney, P., and Sklar, E. (2010). Reasoning about trust using argumentation: A position paper. In *International Workshop on Argumentation in Multi-Agent Systems*, pages 159–170. Springer.

Peterson, J. and Krug, J. (2015). Augur: a decentralized, open-source platform for prediction markets. *arXiv preprint arXiv: 1501.01042*.

Pop, C., Pop, C., Marcel, A., Vesa, A., Petrican, T., Cioara, T., Anghel, I., and Salomie, I. (2018). Decentralizing the stock exchange using blockchain an ethereum-based implementation of the bucharest stock exchange. In *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 459–466. IEEE.

Priem, R. (2020). Distributed ledger technology for securities clearing and settlement: benefits, risks, and regulatory implications. *Financial Innovation*, 6(1):1–25.

Provable Things Limited (2019). Provable.

Pruitt, D. G. (2013). *Negotiation behavior*. Academic Press.

Robinson, P. and Ramesh, R. (2021). General purpose atomic crosschain transactions. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3. IEEE.

Rubinstein, A. (1982). Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109.

Sedlmeir, J., Ross, P., Luckow, A., Lockl, J., Miehle, D., and Fridgen, G. (2021). The dlps: a new framework for benchmarking blockchains.

Singh, A., Click, K., Parizi, R. M., Zhang, Q., Dehghantanha, A., and Choo, K.-K. R. (2020). Sidechain technologies in blockchain networks: An examination and state-of-the-art review. *Journal of Network and Computer Applications*, 149:102471.

Tang, Y., Cai, K., Sklar, E., McBurney, P., and Parsons, S. (2010). A system of argumentation for reasoning about trust. In *Proceedings of the 8th European Workshop on Multi-Agent Systems, Paris, France*.

Tom, R. J., Sankaranarayanan, S., and Rodrigues, J. J. (2020). Agent negotiation in an iot-fog based power distribution system for demand reduction. *Sustainable Energy Technologies and Assessments*, 38:100653.

Toyoda, K., Machi, K., Ohtake, Y., and Zhang, A. N. (2020). Function-level bottleneck analysis of private proof-of-authority ethereum blockchain. *IEEE Access*, 8:141611–141621.

van der Torre, L. and Vesic, S. (2017). The principle-based approach to abstract argumentation semantics. *FLAP*, 4(8).

Wall, E. and Malm, G. (2016). Using blockchain technology and smart contracts to create a distributed securities depository.

Weiss, G. (2013). *Multiagent Systems*. MIT Press.

Wellman, M. P., Greenwald, A., and Stone, P. (2007). *Autonomous bidding agents: Strategies and lessons from the trading agent competition*. Mit Press.

Werner, S. M., Perez, D., Gudgeon, L., Klages-Mundt, A., Harz, D., and Knottenbelt, W. J. (2021). Sok: Decentralized finance (defi). *arXiv preprint arXiv:2101.08778*.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1).

Wooldridge, M. (2009). *An introduction to multiagent systems*. John wiley & sons.

Yu, L., Markovich, R., and Van Der Torre, L. (2020). Interpretations of support among arguments. In *Legal Knowledge and Information Systems*, pages 194–203. IOS Press.

Yu, L. and van der Torre, L. (2020). A principle-based approach to bipolar argumentation. In *NMR 2020 Workshop Notes*, page 227.

Yu, L., Zichichi, M., Markovich, R., and Najjar, A. (2022). Enhancing trust in trust services: Towards an intelligent human-input-based blockchain oracle (ihibo). In *Hawaii International Conference on System Sciences 2022 (HICSS-55)*. To appear.

Zheng, Z., Xie, S., Dai, H.-N., Chen, W., Chen, X., Weng, J., and Imran, M. (2020). An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491.

Zichichi, M. (2021). miker83z/IHiBO: Intelligent Human-input-based Blockchain Oracle. https://github.com/miker83z/IHiBO.

Zichichi, M., Ferretti, S., and D'Angelo, G. (2020a). Are Distributed Ledger Technologies Ready for Intelligent Transportation Systems? In *Proc. of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock 2020), co-located with the 26th Annual International Conference on Mobile Computing and Networking (MobiCom 2020), ACM*, pages 1–6. ACM.

Zichichi, M., Ferretti, S., and D'Angelo, G. (2020b). A framework based on distributed ledger technologies for data management and services in intelligent transportation systems. *IEEE Access*, 8:100384–100402.