Law, Science and Technology
MSCA ITN EJD n. 814177

RI E
Rights of Internet of Everything

Liuwen Yu[1,3], **Mirko Zichichi**[2,3], Réka Markovich[1], Amro Najjar[1]

[1]University of Luxembourg
[2]Universidad Politécnica de Madrid
[3]University of Bologna

Intelligent Human-input-based Blockchain Oracle (IHiBO)

Overview

1. Introduction
2. Conflict Resolution
3. Blockchain
4. IHiBO
5. Evaluation
6. Conclusion

Introduction

The Problem

General Problem → *Trust in decision-making process*

The Problem

General Problem → *Trust in decision-making process*

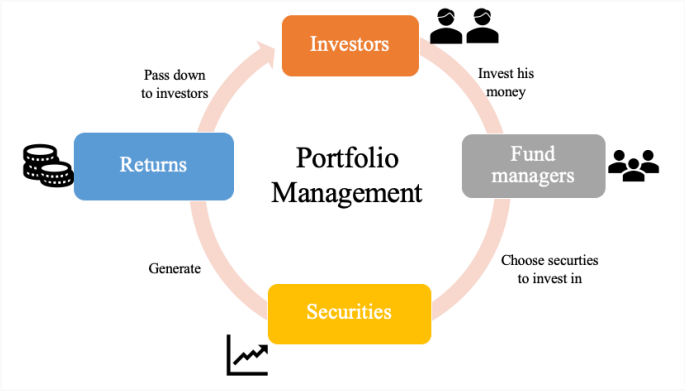
- **Trust service** ← persons or organization *acting on behalf of* another person to deal with the tasks involving finances.

The Problem

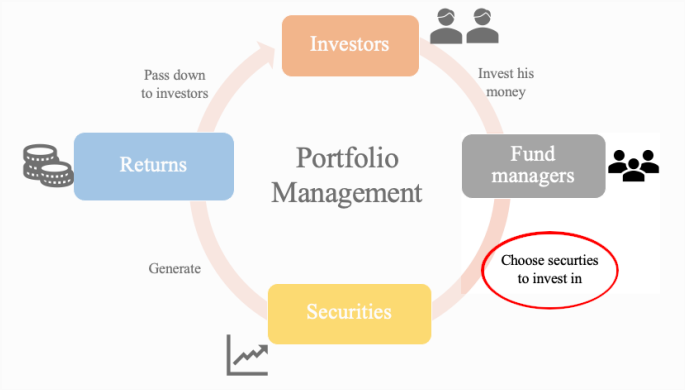
General Problem → *Trust in decision-making process*

- **Trust service** ← persons or organization *acting on behalf of* another person to deal with the tasks involving finances.
- **Fund management** ← fund managers *manage on behalf of* their investors a portfolio of securities (stock, bonds, etc.) and perform risk management.

Specific: Trust problem that emerges in the fund management



Specific: Trust problem that emerges in the fund management



- **reservation** and **lack of documentation** of the decision-making process of investments
- legislators declare investors **right to check the relevant activities** in order to give

Our aim

Perform the **decision-making processes of fund management** in a transparent and traceable way through a framework that integrates:

Our aim

Perform the **decision-making processes of fund management** in a transparent and traceable way through a framework that integrates:

- Formal Argumentation

Our aim

Perform the **decision-making processes of fund management** in a transparent and traceable way through a framework that integrates:

- Formal Argumentation
- Multi-Agent Negotiation

Our aim

Perform the **decision-making processes of fund management** in a transparent and traceable way through a framework that integrates:

- Formal Argumentation
- Multi-Agent Negotiation
- Blockchain and Smart Contracts

Motivation: Argumentation + Negotiation + Blockchain

- Is *formal argumentation* **suited for modelling the decision-making process** of fund management?

Motivation: Argumentation + Negotiation + Blockchain

- Is *formal argumentation* **suited for modelling the decision-making process** of fund management?
  - multi-lateral interaction and reasoning

Motivation: Argumentation + Negotiation + Blockchain

- Is *formal argumentation* **suited for modelling the decision-making process** of fund management?
  - multi-lateral interaction and reasoning
  - incomplete and inconsistent information

Motivation: Argumentation + Negotiation + Blockchain

- Is *formal argumentation* **suited for modelling the decision-making process** of fund management?
  - multi-lateral interaction and reasoning
  - incomplete and inconsistent information
  - $\rightarrow$ help to *explain why a claim or a decision is made*

Motivation: Argumentation + Negotiation + Blockchain

- Is *formal argumentation* **suited for modelling the decision-making process** of fund management?
  - multi-lateral interaction and reasoning
  - incomplete and inconsistent information
  - $\rightarrow$ help to *explain why a claim or a decision is made*
- *Multi-agent negotiation* is used to **determine the quantities and investment timing**

Motivation: Argumentation + Negotiation + Blockchain

- Is *formal argumentation* **suited for modelling the decision-making process** of fund management?
  - multi-lateral interaction and reasoning
  - incomplete and inconsistent information
  - $\rightarrow$ help to *explain why a claim or a decision is made*
- *Multi-agent negotiation* is used to **determine the quantities and investment timing**
- *Blockchain* used **not only** to trace the output of a decision-making process
  $\rightarrow$ **trace argumentation and negotiation and make it auditable**

Conflict Resolution

Portfolio Management Conflict Resolution

- portfolio management decisions → can be based on **arguments** and **counter-arguments**

Portfolio Management Conflict Resolution

- portfolio management decisions → can be based on **arguments** and **counter-arguments**
- **Different fund managers provide their arguments** from their own research

Portfolio Management Conflict Resolution

- portfolio management decisions → can be based on **arguments** and **counter-arguments**
- **Different fund managers provide their arguments** from their own research
- Argumentation Framework (AF) by *Dung (1995)*:
  - directed graphs
  - nodes are arguments
  - arrows are attack relations

Portfolio Management Conflict Resolution

- portfolio management decisions → can be based on **arguments** and **counter-arguments**
- **Different fund managers provide their arguments** from their own research
- Argumentation Framework (AF) by *Dung (1995)*:
  - **directed graphs**
  - **nodes are arguments**
  - **arrows are attack relations**
- **Agent Argumentation Framework(AAF)** ← argument belongs to one or more agents

Portfolio Management Example

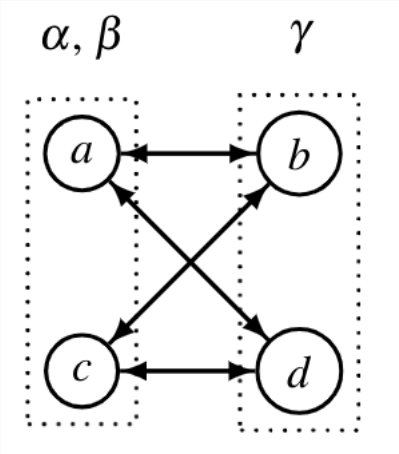


Figure 1: Agent Argumentation Framework

- $\{\alpha, \beta, \gamma\}$ = fund managers' agents
- *a*: ***Buy*** *the stocks, since the company just donated to* ***charities***
- *b*: ***Sell*** *the stocks, since the company has* ***poor sales*** *performance.*
- *c*: ***Buy*** *the stocks, since the company is going to adopt a* ***new technology*** *which will bring huge benefit.*
- *d*: ***Sell*** *the stocks, since there is evidence of charity* ***fraud***

Preference-based Argumentation Framework (PAF)

- **Social Semantics** $\rightarrow$ reduction of an Agent Argumentation Framework (AAF) to a **Preference-based Argumentation Framework (PAF)** by counting the number of agents that have an argument.

Preference-based Argumentation Framework (PAF)

- **Social Semantics** $\rightarrow$ reduction of an Agent Argumentation Framework (AAF) to a **Preference-based Argumentation Framework (PAF)** by counting the number of agents that have an argument.
- **[SAP] Social Reductions of AAF to PAF**
  intermediary step for social agent semantics, given an $AAF = \langle \mathscr{A}, \rightarrow, \mathscr{S}, \sqsubset \rangle$:
  - **SAP(AAF)** $= \langle \mathscr{A}, \rightarrow, \succ \rangle$, with
    $\succ = \{a \succ b || \mathscr{S}_a| > |\mathscr{S}_b|\}$.

Preference-based Argumentation Framework (PAF)

- **Social Semantics** $\rightarrow$ reduction of an Agent Argumentation Framework (AAF) to a **Preference-based Argumentation Framework (PAF)** by counting the number of agents that have an argument.
- **[SAP] Social Reductions of AAF to PAF**
  intermediary step for social agent semantics, given an $AAF = \langle \mathscr{A}, \rightarrow, \mathscr{S}, \sqsubset \rangle$:
  - **SAP(AAF)** $= \langle \mathscr{A}, \rightarrow, \succ \rangle$, with
    $\succ = \{a \succ b | |\mathscr{S}_a| > |\mathscr{S}_b|\}$.
- **[SR] Social Reductions of AAF to AF**
  $SR(AAF) = PR(SAP(AAF))$, where PR is a reduction PAF to AF such as:
  - **PR(PAF)** $= \langle \mathscr{A}, \rightarrow' \rangle$, where
    $\rightarrow' = \{a \rightarrow' b | a \rightarrow b, b \not\succ a, \text{ or } b \rightarrow a, \text{ not } a \rightarrow b, a \succ b, \text{ or } a \rightarrow b, \text{ not } b \rightarrow a\}$.

Portfolio Management Example Social Reduction

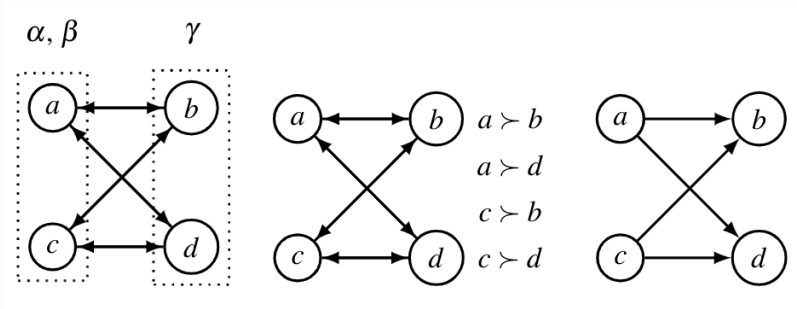


Figure 2: Social reduction

Then we can calculate the only acceptable set $\{a, c\}$.
The set tells the final decision is to buy the stocks.

Autonomous Agents and Negotiation

- Numbers of stocks to buy and the buy timing?

Autonomous Agents and Negotiation

- Numbers of stocks to buy and the buy timing?
- **Multi-agent system** to represent the subjectivity and nuances of different expert opinions and to help the different stakeholders finding agreements.

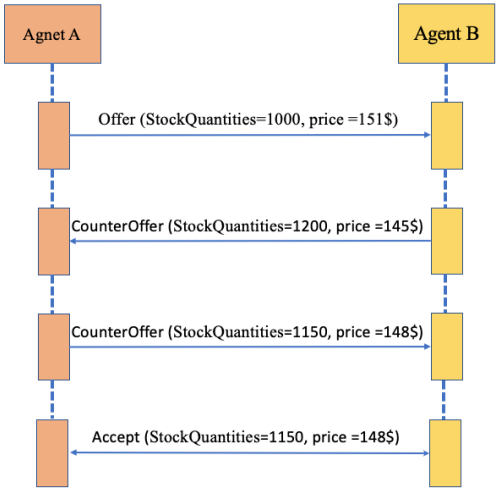
Autonomous Agents and Negotiation

- Numbers of stocks to buy and the buy timing?
- **Multi-agent system** to represent the subjectivity and nuances of different expert opinions and to help the different stakeholders finding agreements.
- **Automated negotiation** is one taking place among autonomous agents

Autonomous Agents and Negotiation

- Numbers of stocks to buy and the buy timing?
- **Multi-agent system** to represent the subjectivity and nuances of different expert opinions and to help the different stakeholders finding agreements.
- **Automated negotiation** is one taking place among autonomous agents
- The problem being negotiated can be described by one or more **issues** and the **priority** given to each issue can differ from one negotiator to another.

Portfolio Management Example Negotiation



Blockchain

Blockchain Trust

- **distributed ledger** and **immutability** $\rightarrow$ *enhances trust*

Blockchain Trust

- **distributed ledger** and **immutability** $\rightarrow$ *enhances trust*
- transactions stored in blockchains are **auditable** and **traceable**

Blockchain Trust

- **distributed ledger** and **immutability** → *enhances trust*
- transactions stored in blockchains are **auditable** and **traceable**
- **Distributed Ledger Technologies** → potential to revolutionize *financial agreements*.

Blockchain Trust

- **distributed ledger** and **immutability** → *enhances trust*
- transactions stored in blockchains are **auditable** and **traceable**
- **Distributed Ledger Technologies** → potential to revolutionize *financial agreements*.
- fund managers can trade securities on behalf of the investors.

Oracle

- **Oracle** $\rightarrow$ system that act as a bridge between DLTs and the “outside” world

Oracle

- **Oracle** $\rightarrow$ system that act as a bridge between DLTs and the “outside” world
- **Retrieve**, **verify** and **digest** data into distributed ledgers:

Oracle

- **Oracle** $\rightarrow$ system that act as a bridge between DLTs and the “outside” world
- **Retrieve**, **verify** and **digest** data into distributed ledgers:
  1. *software*: get data from online sources and insert these to DLT, e.g. stocks prices;

Oracle

- **Oracle** $\rightarrow$ system that act as a bridge between DLTs and the "outside" world
- **Retrieve**, **verify** and **digest** data into distributed ledgers:
  1. *software*: get data from online sources and insert these to DLT, e.g. stocks prices;
  2. *hardware*: retrieve data from the physical world through sensors, e.g. weather IoT;

Oracle

- **Oracle** $\rightarrow$ system that act as a bridge between DLTs and the "outside" world
- **Retrieve**, **verify** and **digest** data into distributed ledgers:
  1. *software*: get data from online sources and insert these to DLT, e.g. stocks prices;
  2. *hardware*: retrieve data from the physical world through sensors, e.g. weather IoT;
  3. *human*: individuals manually insert data to DLT, e.g. dispute resolution judge.

Smart Contracts

- Some DLTs enable is the possibility to execute **Smart Contracts**

Smart Contracts

- Some DLTs enable is the possibility to execute **Smart Contracts**
- **instructions** that, once deployed on the ledger, **cannot be altered**

Smart Contracts

- Some DLTs enable is the possibility to execute **Smart Contracts**
- **instructions** that, once deployed on the ledger, **cannot be altered**
- *each DLT node* execute these instructions and verify that the **output is equal** for all the other nodes in the network

Smart Contracts

- Some DLTs enable is the possibility to execute **Smart Contracts**
- **instructions** that, once deployed on the ledger, **cannot be altered**
- *each DLT node* execute these instructions and verify that the **output is equal** for all the other nodes in the network
- **handle data** coming from other smart contracts or from the user's inputs, e.g. smart contracts cannot fetch a webpage on the Internet,

Smart Contracts

- Some DLTs enable is the possibility to execute **Smart Contracts**
- **instructions** that, once deployed on the ledger, **cannot be altered**
- *each DLT node* execute these instructions and verify that the **output is equal** for all the other nodes in the network
- **handle data** coming from other smart contracts or from the user's inputs, e.g. smart contracts cannot fetch a webpage on the Internet,
- store data in the (immutable) ledger

IHiBO

Intelligent Human-input-based Blockchain Oracle (IHiBO)

- A cross-chain oracle that enables the *execution and traceability* of argumentation and negotiation processes

Intelligent Human-input-based Blockchain Oracle (IHiBO)

- A cross-chain oracle that enables the *execution and traceability* of argumentation and negotiation processes
- Combines

Intelligent Human-input-based Blockchain Oracle (IHiBO)

- A cross-chain oracle that enables the *execution and traceability* of argumentation and negotiation processes
- Combines
  - Formal Argumentation

Intelligent Human-input-based Blockchain Oracle (IHiBO)

- A cross-chain oracle that enables the *execution and traceability* of argumentation and negotiation processes
- Combines
  - Formal Argumentation
  - Automated Negotiation

Intelligent Human-input-based Blockchain Oracle (IHiBO)

- A cross-chain oracle that enables the *execution and traceability* of argumentation and negotiation processes
- Combines
  - Formal Argumentation
  - Automated Negotiation
  - DLT Framework

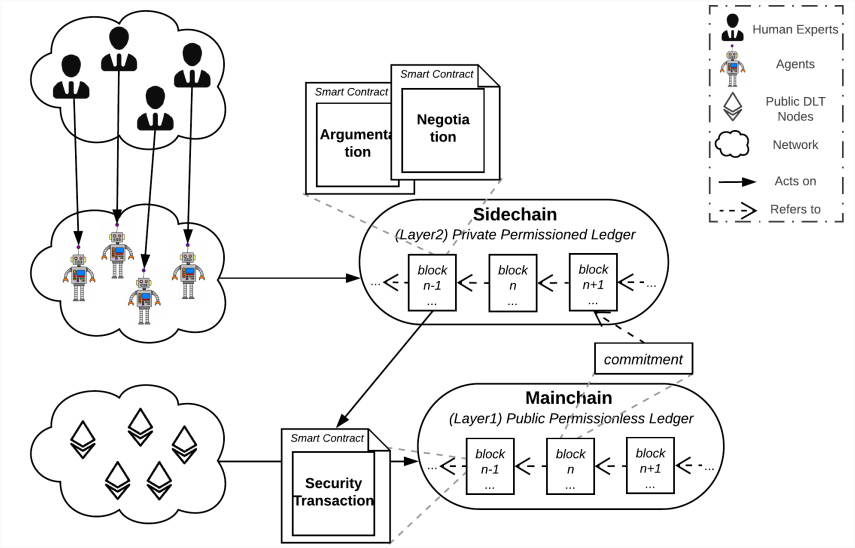
Intelligent Human-input-based Blockchain Oracle (IHiBO)

- A cross-chain oracle that enables the *execution and traceability* of argumentation and negotiation processes
- Combines
  - **Formal Argumentation**
  - **Automated Negotiation**
  - **DLT Framework**
- Provides *transparency and auditability*

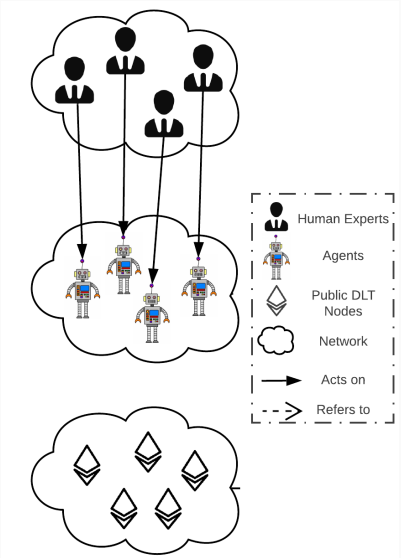
Intelligent Human-input-based Blockchain Oracle (IHiBO)

- A cross-chain oracle that enables the *execution and traceability* of argumentation and negotiation processes
- Combines
  - **Formal Argumentation**
  - **Automated Negotiation**
  - **DLT Framework**
- Provides *transparency and auditability*
- Involves the intervention of *human experts*

IHiBO Architecture

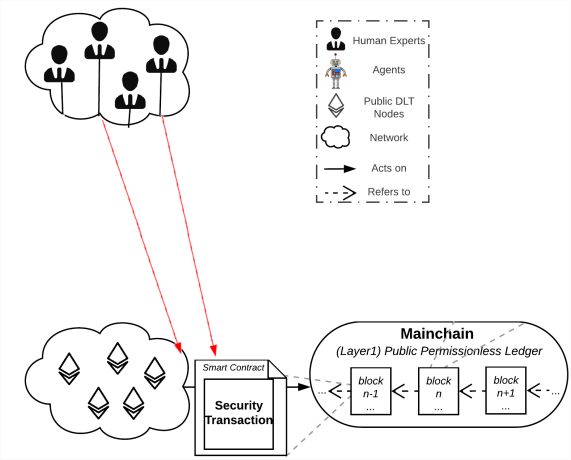


IHiBO Architecture



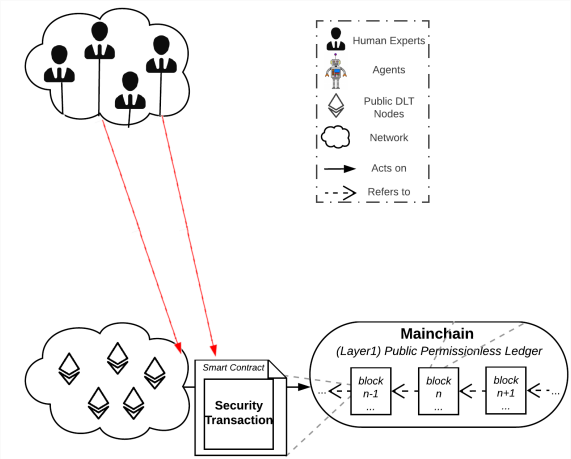
- **Human Expert**, the one who takes most of the decisions and that gives inputs to the agent;
- **Agent**, the one that can assist human experts and that directly interact with the DLT.
- **Public DLT Node**, a node that builds the network of a public DLT, such as the **Ethereum** blockchain

Decentralized Finance (DeFi)



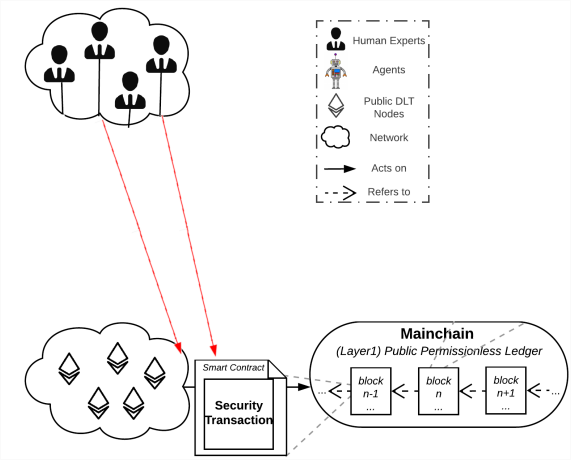
- **DeFi** $\rightarrow$ smart contract based financial infrastructures that are non-custodial, permissionless, openly verifiable and composable.

Decentralized Finance (DeFi)



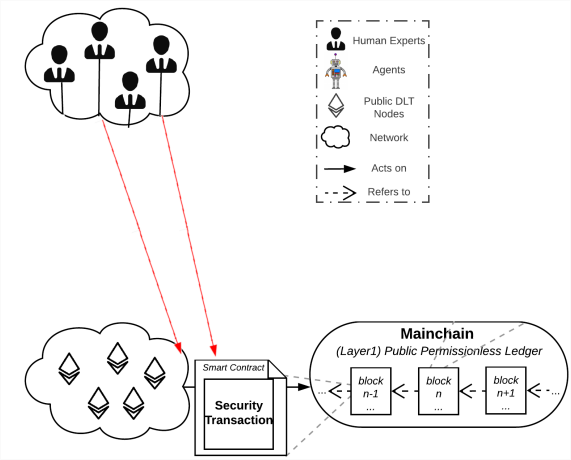
- **DeFi** → smart contract based financial infrastructures that are non-custodial, permissionless, openly verifiable and composable.
- **Decentralized Exchanges (DEX)** → non-custodial exchange of on-chain digital assets in the Ethereum blockchain

Decentralized Finance (DeFi)



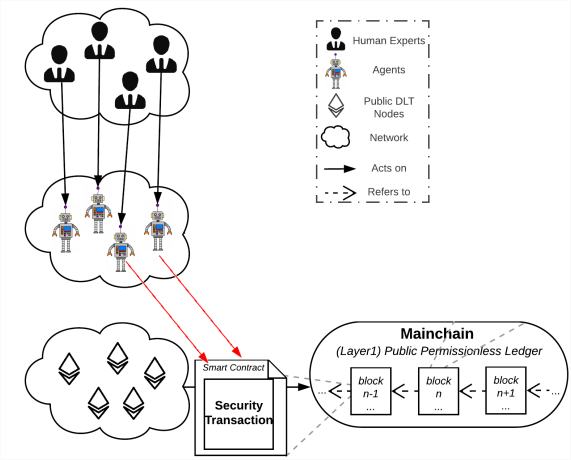
- **DeFi** → smart contract based financial infrastructures that are non-custodial, permissionless, openly verifiable and composable.
- **Decentralized Exchanges (DEX)** → non-custodial exchange of on-chain digital assets in the Ethereum blockchain
- *direct selling/buying of traditional stocks that have been "tokenized"*

Decentralized Finance (DeFi)



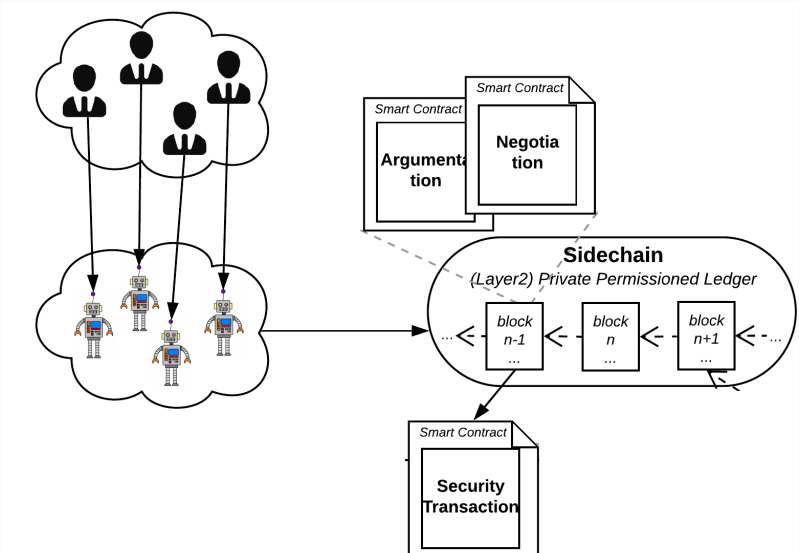
- **DeFi** → smart contract based financial infrastructures that are non-custodial, permissionless, openly verifiable and composable.
- **Decentralized Exchanges (DEX)** → non-custodial exchange of on-chain digital assets in the Ethereum blockchain
- *direct selling/buying of traditional stocks that have been "tokenized"*
- Smart contracts that implements swapping tokens and cryptocurr. → **SecurityTransaction**

Decentralized Finance (DeFi)



- **DeFi** → smart contract based financial infrastructures that are non-custodial, permissionless, openly verifiable and composable.
- **Decentralized Exchanges (DEX)** → non-custodial exchange of on-chain digital assets in the Ethereum blockchain
- *direct selling/buying of traditional stocks that have been "tokenized"*
- Smart contracts that implements swapping tokens and cryptocurr. → **SecurityTransaction**

IHiBO Sidechain



IHiBO Sidechain

- **second layer solution** on top of the mainchain (i.e. Ethereum public blockchain)
- for executing **conflict resolution** processes

IHiBO Sidechain

- **second layer solution** on top of the mainchain (i.e. Ethereum public blockchain)
- for executing **conflict resolution** processes
- to support execution **log** can be later **audited**

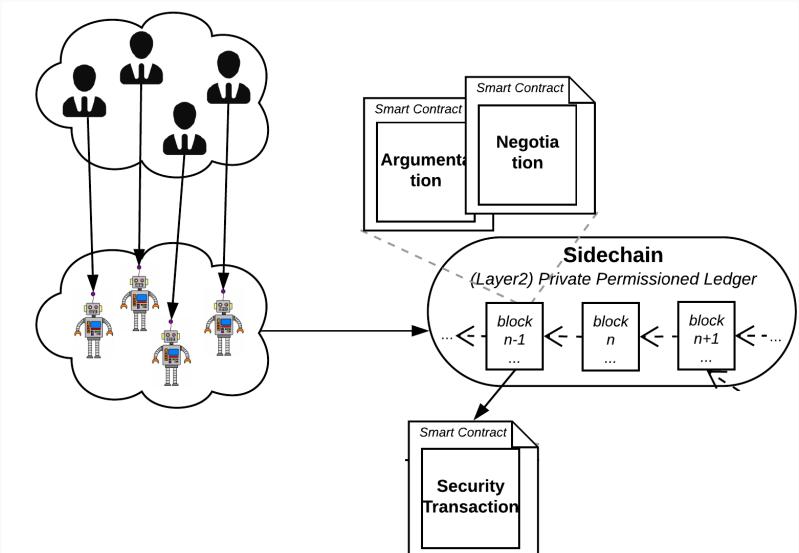
IHiBO Sidechain

- **second layer solution** on top of the mainchain (i.e. Ethereum public blockchain)
- for executing **conflict resolution** processes
- to support execution **log** can be later **audited**
- Ethereum protocol distributed among nodes in a **private permissioned network**

IHiBO Sidechain

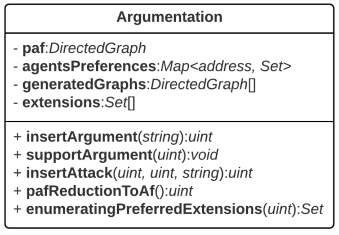
- **second layer solution** on top of the mainchain (i.e. Ethereum public blockchain)
- for executing **conflict resolution** processes
- to support execution **log** can be later **audited**
- Ethereum protocol distributed among nodes in a **private permissioned network**
- **Proof-of-Authority** (PoA) consensus algorithm

IHiBO Sidechain

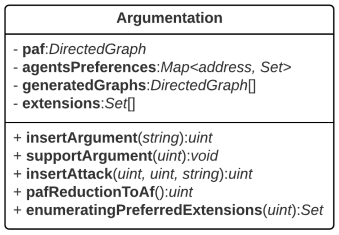


Argumentation Smart Contract

- A **directed graph** data structure, nodes are **arguments** and edges are **attacks**

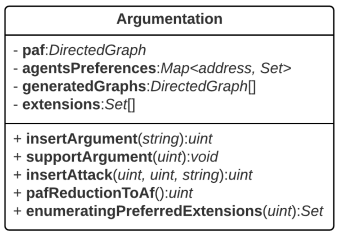


Argumentation Smart Contract



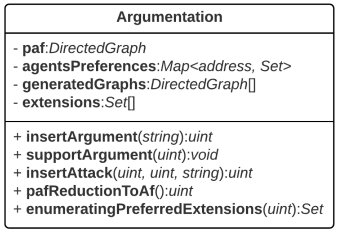
- A **directed graph** data structure, nodes are **arguments** and edges are **attacks**
- **Each agent** invokes *insertArgument()* and *insertAttacks()*

Argumentation Smart Contract



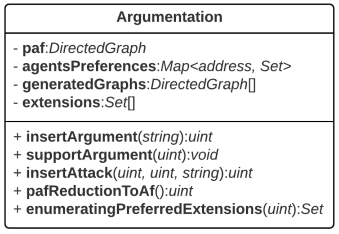
- A **directed graph** data structure, nodes are **arguments** and edges are **attacks**
- **Each agent** invokes *insertArgument()* and *insertAttacks()*
- Also set as "**preferred**" an existing argument (*supportArgument()*)

Argumentation Smart Contract



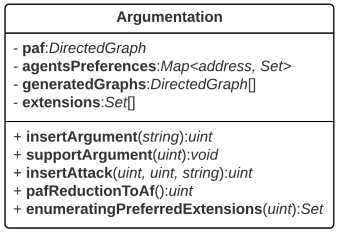
- A **directed graph** data structure, nodes are **arguments** and edges are **attacks**
- **Each agent** invokes *insertArgument()* and *insertAttacks()*
- Also set as "**preferred**" an existing argument (*supportArgument()*)
- **Reductions of PAF to AF** can be invoked and executed directly (*pafReductionToAfPr()*)

Argumentation Smart Contract



- A **directed graph** data structure, nodes are **arguments** and edges are **attacks**
- **Each agent** invokes *insertArgument()* and *insertAttacks()*
- Also set as "**preferred**" an existing argument (*supportArgument()*)
- **Reductions of PAF to AF** can be invoked and executed directly (*pafReductionToAfPr()*)
- Finally, an **extension** can be found for the previously obtained AF (*enumeratingPreferredExtensions()*)

Argumentation Smart Contract



- A **directed graph** data structure, nodes are **arguments** and edges are **attacks**
- **Each agent** invokes *insertArgument()* and *insertAttacks()*
- Also set as "**preferred**" an existing argument (*supportArgument()*)
- **Reductions of PAF to AF** can be invoked and executed directly (*pafReductionToAfPr()*)
- Finally, an **extension** can be found for the previously obtained AF (*enumeratingPreferredExtensions()*)
- This possibly provides a set of **arguments that lead to a final decision**.

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process
- A data structure holds the data needed during a negotiation thread

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process
- A data structure holds the data needed during a negotiation thread
- Each agent can **start a new negotiation** for a specific set of issues (*newNegotiation()*).

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process
- A data structure holds the data needed during a negotiation thread
- Each agent can **start a new negotiation** 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, e.g. *a time dependent tactic*.

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process
- A data structure holds the data needed during a negotiation thread
- Each agent can **start a new negotiation** 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, e.g. *a time dependent tactic*.
- Based on this evaluation, an agent:

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process
- A data structure holds the data needed during a negotiation thread
- Each agent can **start a new negotiation** 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, e.g. *a time dependent tactic*.
- Based on this evaluation, an agent:
  - **makes a new offer** (*newOffer()*) providing a new set of values related to the issues

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process
- A data structure holds the data needed during a negotiation thread
- Each agent can **start a new negotiation** 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, e.g. *a time dependent tactic*.
- Based on this evaluation, an agent:
  - **makes a new offer** (*newOffer()*) providing a new set of values related to the issues
  - **accepts** (*accept()*) the other agent's offer

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process
- A data structure holds the data needed during a negotiation thread
- Each agent can **start a new negotiation** 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, e.g. *a time dependent tactic*.
- Based on this evaluation, an agent:
  - **makes a new offer** (*newOffer()*) providing a new set of values related to the issues
  - **accepts** (*accept()*) the other agent's offer
  - **refuses** it (by not providing input to the smart contract).

Negotiation Smart Contract

- For automated negotiations on several issues on the arguments provided by the argumentation process
- A data structure holds the data needed during a negotiation thread
- Each agent can **start a new negotiation** 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, e.g. *a time dependent tactic*.
- Based on this evaluation, an agent:
  - **makes a new offer** (*newOffer()*) providing a new set of values related to the issues
  - **accepts** (*accept()*) the other agent's offer
  - **refuses** it (by not providing input to the smart contract).
- The *accept()* method **directly enact the process** of interaction with the SecurityTransaction smart contract on the **mainchain**.

Evaluation

Experiments

- Ethereum private network using PoA, with optimal configuration, can reach up to 1000 transactions per second

Experiments

- Ethereum private network using PoA, with optimal configuration, can reach up to 1000 transactions per second
- Our experiments focus on the scalability in terms of arguments and negotiation issues number

Experiments

- Ethereum private network using PoA, with optimal configuration, can reach up to 1000 transactions per second
- Our experiments focus on the scalability in terms of arguments and negotiation issues number
- We measure our experiments in terms of **gas cost**, a feature of the Ethereum protocol

Experiments

- Ethereum private network using PoA, with optimal configuration, can reach up to 1000 transactions per second
- Our experiments focus on the scalability in terms of arguments and negotiation issues number
- We measure our experiments in terms of **gas cost**, a feature of the Ethereum protocol
- Gas is a unit that measures the amount of computational effort that takes to execute operations in Ethereum smart contracts

Portfolio Management Example

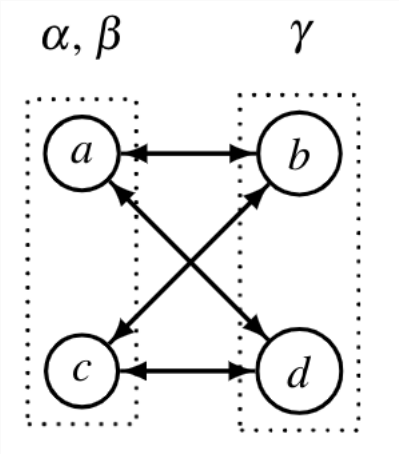


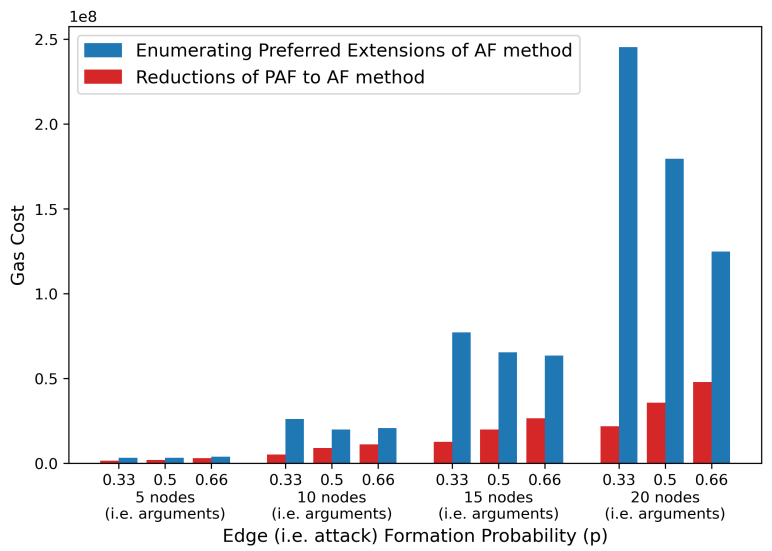
Figure 3: Agent Argumentation Framework

- $\{\alpha, \beta, \gamma\}$ = fund managers' agents
- *a*: **Buy** *the stocks, since the company just donated to* **charities**
- *b*: **Sell** *the stocks, since the company has* **poor sales** *performance.*
- *c*: **Buy** *the stocks, since the company is going to adopt a* **new technology** *which will bring huge benefit.*
- *d*: **Sell** *the stocks, since there is evidence of charity* **fraud**

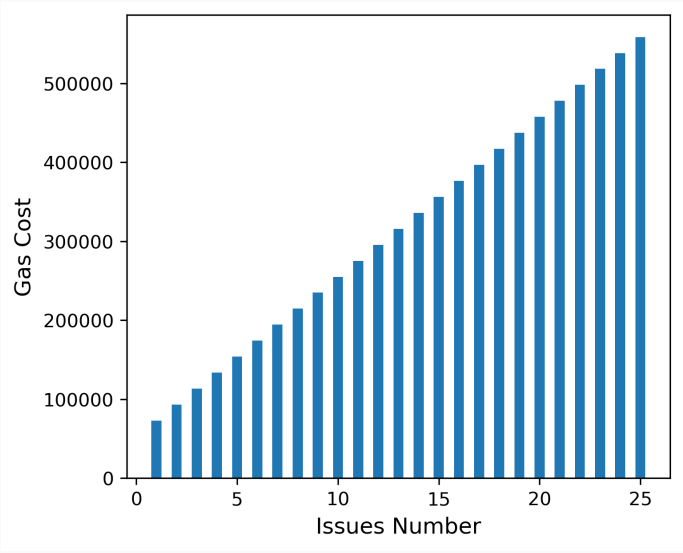
Portfolio Management Example Gas Cost

Smart Contract	Method	Occurrency	Gas Cost
Argumentation	insertArgument()	a	157 470
Argumentation	supportArgument()	$\leq a \times (n-1)$	80 491
Argumentation	insertAttack()	$\leq a \times (a-1)$	215 011
Argumentation	pafReductionToAfPr()	1	1 877 277
Argumentation	enumeratingPreferred Extensions()	1	1 412 065
Negotiation	newNegotiation()	1	104 961
Negotiation	newOffer()	t	52 438
Negotiation	accept()	1	64 211

Arguments Number



Negotiation Issues Number



Conclusion

Conclusion

- Proposal of an integrated framework incorporating formal argumentation and negotiation within a blockchain framework.

Conclusion

- Proposal of an integrated framework incorporating formal argumentation and negotiation within a blockchain framework.
- We explained our idea in a fund management scenario, but our proposal is not only bound to this domain.

Conclusion

- Proposal of an integrated framework incorporating formal argumentation and negotiation within a blockchain framework.
- We explained our idea in a fund management scenario, but our proposal is not only bound to this domain.
- The results of our experiments shows that the use of a second layer DLT, allows to securely operate without too many performance limitations *in bounded use cases*, while maintaining a high level of traceability.

Conclusion

- Proposal of an integrated framework incorporating formal argumentation and negotiation within a blockchain framework.
- We explained our idea in a fund management scenario, but our proposal is not only bound to this domain.
- The results of our experiments shows that the use of a second layer DLT, allows to securely operate without too many performance limitations *in bounded use cases*, while maintaining a high level of traceability.
- Future work:

Conclusion

- Proposal of an integrated framework incorporating formal argumentation and negotiation within a blockchain framework.
- We explained our idea in a fund management scenario, but our proposal is not only bound to this domain.
- The results of our experiments shows that the use of a second layer DLT, allows to securely operate without too many performance limitations *in bounded use cases*, while maintaining a high level of traceability.
- Future work:
  - Provide and adapt to a high level of adaptability in the decisions of the fund management (investors' preferences, attitude (aggressive or moderate) and the financial environment bull or bear)

Conclusion

- Proposal of an integrated framework incorporating formal argumentation and negotiation within a blockchain framework.
- We explained our idea in a fund management scenario, but our proposal is not only bound to this domain.
- The results of our experiments shows that the use of a second layer DLT, allows to securely operate without too many performance limitations *in bounded use cases*, while maintaining a high level of traceability.
- Future work:
  - Provide and adapt to a high level of adaptability in the decisions of the fund management (investors' preferences, attitude (aggressive or moderate) and the financial environment bull or bear)
  - Explainable AI, how we can make the decision-making process explainable for different types of users (experts, non-experts, etc.) and for different purposes (e.g. transparency, debugging, etc.).