Jury.Online

www.about.jury.online

Yellow Paper

Table of Contents

1- Introduction

A deal is the fundamental concept standing as the pillar supporting interactions between people. The fair and unbiased execution of any deal between the participants is the basis for establishing trust between individuals, groups and companies.

The advent of modern technologies has aggravated the issue of trust, as instruments have been developed for overcoming and subverting the trust factor, thus resulting in a tremendous amounts of fraud. The legal component is no longer capable of efficiently handling the vast amounts of complaints, and its abuse has been leading to an increase in dissatisfaction and disillusionment with the system as a whole.

We at **Jury.Online** consider the current means of execution and regulation of deals to be completely outdated and want to change them by introducing an interaction protocol for mediators and parties of

a deal, as well as a transparent, secure, and user-friendly platform for making deals using blockchain technology and modern cryptographic systems.

2- Project Overview

The Jury.Online platform enables its users to make and execute deals using smart contracts. In case of dissatisfaction of any of the parties with the fulfillment of the terms of the deal, the circumstances are considered by a panel of mediators delivering judgment in favor of one of the parties to the deal. The main actors of the platform are the counterparties of the deal and the mediators, which are picked from the pool. By calling on a Pool of mediators, Jury.Online gives any person with expertise in any certain field the opportunity to apply their experience and knowledge for paid dispute resolution.

3- Structure and Operation

The Jury.Online platform functions on the basis of smart contracts that contain the constituent elements of a deal and its participants in an immutable format, thus ensuring the execution or resolution of the parties' contractual obligations.

Below are the requirements for a smart contract, the information it stores, the functions to be called and the participants that will call them when the deal becomes active.

The Jury.Online smart contract stores the following information:

1. The counterparty identifiers.
2. The subject of the deal, links to related documents and attachments.
3. The starting time of the deal, time of execution, time of acceptance, the moment of dispute fee payment.
4. The counterparties' deposits and collateral for dispute resolution.
5. The type of dispute resolution, e.g. identifiers of the pool of judges for random judges.
6. The identifiers of other smart contracts used in the protocol:
    (a) The rating of the smart contracts (Rater).
    (b) The responsible for choosing a judge based on a random number generator (RNG).

This data is vital for smart contract operation. Typically, it would hold more details, such as information about public visibility of the deal on the website or the number of possible appeals.

After the counterparty confirms its consent to the terms and conditions of the deal and specifies the necessary information, the deal is considered to be concluded.

The parties then have a certain period of time to fulfill their obligations. In the event each party is satisfied with the result, the money from the smart contract of the deal is transferred to the counterparty (or elsewhere, if specified by the deal), and the deal is considered to be successfully completed.

Given the fact that working with blockchain technologies is still challenging for people who are unaware of the technical details involved. For normal operation, one needs to have the entire Ethereum blockchain, which currently occupies a considerable amount of hard drive memory, up to 290GB. Such a volume of memory is too much for an average user, not to mention mobile devices. Therefore, Jury.Online operates as an intermediary that enables users to send transactions to the blockchain. The transactions cannot be changed by Jury.Online, because the platform does not store or have access to the users' private keys.

Jury.Online maintains its own nodes and servers for flawless service operation, and the user does not have to manually call contract functions through Parity or Mist. The user visits the website and performs an action, which is then translated into blockchain transaction and signs it by their private key stored on their side.

4-Getting Started

Any person willing to initiate a deal should open the website or mobile application of Jury.Online and create a deal using one of the ready templates and specify all the important and relevant details. The deal is then placed into the blockchain as a smart contract. After the deal is accepted by the other party, it cannot be deleted or changed by anyone. Even the platform administration has no control over this smart contract. In addition, the deal is assigned a link, which leads to the website page displaying information about the deal. The initiator sends this link to the counterparty or makes it accessible to the public if they do not yet know who is going to be the counterparty. For example, a deal is intended for product delivery or service rendering, but the initiator does not know who is going to perform it.

Every deal splits the investments in it into a series of Milestones. Each of the Milestones is a specific and logical step of the deal procedure which is assigned a certain amount of the funds of the total deal

value. The Milestones serve as benchmarks for the parties to evaluate the fulfillment of contractual obligations and act as an important factor in dispute resolution. Each Milestone is assigned a three day grace period within which any of the parties can open a dispute regarding deal execution. In case of failure by either of the parties to achieve a certain Milestone, the funds remaining on the deal will be up for dispute and subsequent refund.

When creating a deal, the initiator indicates a sum they are ready to pay for the job (or want to receive as a rendered service, if the initiator is a contractor). This sum, converted into cryptocurrency form, is then sent to the balance of the smart contract and is deposited on it. This money cannot be withdrawn until the deal is completed, regardless of whether it was successful, whether the parties were content with its execution or need to start legal proceedings. Part of the balance is allocated to resolve possible disputes, or to pay court fees.

All the participants, the counterparties, the judges and the pool operators must have a pair of keys used in an asymmetric cryptosystem for their identification and to sign their actions. Such a key pair is given to every owner of an “account” of a typical blockchain.

The counterparties of the deal need a secure way for sharing files and documents, and these attachments later must be exposed to the judges in case of a dispute. Moreover, the jurors need complete information about all the actions of the parties, and they also need to be sure that the files presented were really sent by a certain counterparty. Fortunately, modern cryptography successfully solves this problem via proven asymmetric crypto-systems. During a dispute resolution, the counterparties decide which documents are vital for the correct outcome and disclose them to the jurors.

Pools are another party to the deal that should maintain up-to-date lists of active judges who are online and ready to consider disputes. In order for a smart contract to recognize them, part or all of the information about the list should be kept in the blockchain.

Basic services provided under the arbitration of a deal include random selection of judges for dispute resolution. Pools provide up-to-date lists of active judges. In case of a dispute, a pseudorandom number generator selects the required number k of specific judges from this list who receive information about the deal. Random number generator operates via smart contract so that it is not controlled by any party.

However, for proper operation, it needs some initial state that is defined by a numeric parameter called a seed. Since all information in the blockchain is open and accessible to any user, a pool operator or a party that knows the seed can adjust the mediators' order so that a certain deal is considered by certain judges who may be in collusion with a party. Therefore, the seed cannot be calculated on the basis of any public information, but should use parameters provided by the counterparties. Since the counterparties to a dispute pursue opposite goals, they are interested in safe and high-quality seed that would lead to an unpredictable judge choice.

5- Refund

Jury.Online is an escrow platform that provides a set of features tied to a deal initiated by parties with an underlying monetary reward in token form. The deal has a certain set of Milestones that need to be reached as a process to fulfilling the contractual obligations set therein. After the end of each Milestone, there is a period of at least three days before the next Milestone can begin. During this period, any investor can start a dispute. In this case, the pool of arbitrators comes into effect.

If the decision made on a dispute by the arbitrators is in favor of the investor, the latter will get the remaining Ethers or cryptocurrencies on the deal as a refund. The Ether or cryptocurrencies set for the previous Milestones will no longer be returned. In the opposite case, the investor continues to receive the tokens set for the deal.

6- Initiating a Dispute

In the event one of the parties is not satisfied with the deal's execution and believes that the counterparty has not fulfilled its obligations, the deal is sent to the judges for consideration. Any party may initiate a dispute and send the deal to the judges, but as a rule, it is the party that deposited the money.

The main rules for initiating a dispute are the following:
1) Only an investor (the holder of the tokens of Jury.Online) can open a dispute regarding a deal;
2) The dispute can be opened only in a period specially allocated for such a procedure within the deal's parameters (3 days after the end of a Milestone).

7- Mediator Decision-Making Procedure

When initiating a dispute, the parties have a certain period of time to set forth their arguments and comments on the issue. After that, the deal is sent to the judges for consideration. The form of dispute resolution may vary, but usually, the system chooses an N number of random judges who receive anonymized information about the deal and take the decision by an absolute majority of votes.

Judges are provided by a source called a judge pool. In its simplest form, this is done by Jury.Online, but third parties specializing in a certain sector may also offer dispute resolution services. The identities of the judges are unknown to the parties, but their competence is. Judges have a fixed period of time to take a decision. The information about the decision of a particular judge is encrypted and unavailable to other judges.

There are other forms of judging accessible. Instead of several random, unknown judges, the parties may agree on a specific judge they consider fair.

Most deals imply performance of contractual obligations for payment. In this case, one party has automatically fulfilled its obligations, as it has transferred the payment to a smart contract, and it is the only party interested in the dispute. However, we do not want to limit the range of possible deals, for example, services for services or other options, so it is possible to choose who and when is to pay for possible litigation.

8- Dispute Decision Consequences

Since judges are rated based on the judgments they pass and are rewarded for their actions, the economic and rating component motivates and forces judges to investigate and resolve disputes fairly and correctly, rather than to randomly pass their verdicts. Judges should not know the verdicts of other judges to prevent them from taking the decision voted for by a majority. Therefore, a mechanism for hiding the verdicts must be used, and verdicts cannot be stored in the blockchain as the number of for/against votes. To avoid collusion, Jury.Online uses probabilistic encryption.

The algorithm used in this case is encrypted with some additional, randomly generated data, for instance “salt”. This data is generated by the parties of the deal. The parties to encrypt the verdict are chosen in rotation. The party uses a symmetric-key algorithm to encrypt the “salt” and to store it in the smart contract of the deal. Later, the party will disclose the encryption key and reveal the verdict. Refusal to disclose the key would mean that this party lost the dispute.

Then a party encrypts the “salt” using the judge’s public key and sends it to the judge via side channel, so it is not published in the blockchain. Judge can reveal the salt and make their verdict open, however, it can be done only by publishing their private key, so the judge will lose control of their account and all the funds on their balance.

Another approach is to use probabilistic encryption, though it is rather complicated and is therefore described in the technical protocol specification. A significant advantage of this method is that there is no need in side channel communication.

The consequences for the parties are just as important, as the losing party will have to refund the counterparty with the Ether or cryptocurrencies allocated to the project and pay any ensuing court fees.

9- Receiving Project Funding

Cryptocurrencies transfers arise in the workflow in following forms:
1. Transaction fee of the blockchain. This part is paid by the initiator of the deal. By agreement with the other counterparty, this fee can count as a part of the deal sum.
2. Deal amount. Both counterparties may deposit funds to the smart contract of the deal. Hence, the counterparty that is meant to receive the deal amount may provide a pledge to show their serious intentions to the deal.
3. The first dispute is free for the investor. The platform charges a fee for ensuing disputes to avoid “dispute spam”. Also, after any counterparty pushes the button “withdraw token/ETH”, this party will have to pay for the blockchain transaction (gas).

The first two items are nominated in the internal blockchain cryptocurrency. Jury.Online has no control over these amounts and charges no commission from these parts. Jury.Online charges a commissions only from dispute resolution payment. The dispute resolution fee is paid in Jury.Online Tokens (JOT), which are issued at the ICO. The fee is only charged in case of a resolution, so a deal without a dispute takes no other fee than that for the blockchain transaction.

Jury.Online receives the same fee both for the $10,000 deal and the $100 deal. However, we expect that a $10,000 deal requires more qualified jurors for dispute resolution, hence it will involve larger dispute resolution payment. The moment of fee payment is specified beforehand, so a deal can be created with tokens deposited for a potential dispute, or tokens can be purchased when the dispute is

started. If the deal is successfully executed without any dispute, the tokens are returned to the party that provided them.

10- Jury.Online Smart Contracts

Jury.Online applies a number of smart contracts within the system, each of which is responsible for executing certain functions related to deals. The following section contains a detailed description of the Cluster, Arbiters Pool, and Responsible Crowdsale smart contracts and relevant documentation.

The successful development and operation of the Jury.Online platform requires the implementation of 3 main smart contracts and the necessary procedure:
- Cluster Smart Contract
- Arbiters Pool Smart Contract
- Responsible Crowdsale Smart Contract
- How to start working with Jury.Online smart contracts.

General Repository:
https://git.kryptonhub.com/jury-online/yet-another-contract-repository/tree/ararat_tonoyan

Cluster Smart Contract

The Cluster smart contract needs to be deployed only once by the owner of the platform. It saves information about Operators and Crowdsale contracts addresses, and also connects the Responsible Crowdsale with the Mediators Pool contract, when the investors open a dispute.

The contract has 2 key roles inside the platform:
1) The Owner of the Cluster:
- This address can withdraw fees from the Cluster contract and transfer them to its address;
- Add new Backend addresses;
- Call the emergency transfer function in any Crowdsale contract.
2) The Backend addresses:
- This addresses will be based on servers and will be triggered by platform admins;
- It can create a new Crowdsale;
- Add or remove an Arbiter from the Arbiter Pool.

The Cluster is also responsible for:
- Adding/removing Mediators (only Backend addresses can call this method);
- Adding new Crowdsales (only Backend addresses can call this method);
- Collecting all fees from Crowdsales (only the Owner can call this method);
- Emergency transferring of all tokens and ETH from any Crowdsale contract to another, safer smart contract (only the Owner can call this method, the Operator of that Crowdsale needs to call this method also);
- Receiving disputes from Investors and add new dispute in the ArbitersPool smart contract (the investor need to call addDispute method from Cluster contract);

Cluster Smart Contract –
https://ropsten.etherscan.io/address/0x68ca57d1d7b5dee55162c48455a522a38437d6e5

Arbiters Pool Smart Contract

The Arbiters pool smart contract is the contract is used by Arbiters so they can vote on disputes and solve them. The Arbiters can be added or removed from the Cluster smart contract (Backend role). Each dispute has its own dispute ID (number from 0) and personal information like an investor address, who opened the dispute, Crowdsale address, the reason of the dispute etc. Each dispute has 2 states - "Waiting" and "Solved". The Arbiters have 2 choices - Operator wins or Investor Wins. All disputes need 3 votes to be solved, but if there are 2 Arbiters' voices with the same result, the dispute will be solved automatically. For voting, the investors need to call the Vote method from the Arbiters Pool contract, writing there the ID of the dispute and their Choice (0 - Operator Wins / 1 - Investor Wins). The quorum of judges involved may change for every project and deal depending on the requirements of their owners.

Methods:

getCycleDetails - you need to pass on the ID of the cycle (0,1,2 ...), returns:
- Percentage of cycle Ether;
- Percentage of cycle Token;
- Milestone hashes of this cycle.

getCyclesAmount - returns the total number of active loops;
getMilestonesHashes - returns all of the Milestone hashes, sorted by time;

getMilestoneDetails - users need to pass the Milestone hash, returns:
- Titles;
- Start time;
- The start time of the dispute (-3 days from the start time);
- ID of the cycle which the Milestone is part of;
- Percentage of tokens in the cycle;
- Percentage of Ethers in the cycle;
- Number of active disputes;
- Milestone status (PENDING, DISPUTS_PERIOD, APPROVED).

getMilestoneStatus – returns the status of the Milestone separately
getTotalPercents – returns the total amount of tokens and Ethers of all cycles
isMilestoneHasActiveDisputes – separately returns the bool, whether the Milestone has active disputes
haveMilestonesSetted – returns a bool operator, whether 1 cycle is added, only after that investors can send Ether.
didInvestorOpenedDisputeBefore – users need to pass the hash of the cycle and the address of the investor, returns the bool if the investor opened a disputed on the given Milestone before.
didInvestorWithdraw – users need to pass the hash of the cycle and the address of the investor, returns the bool if the investor removed tokens from the Milestone before.

getCrowdsaleDetails - returns:
- The price of the token;
- The address of the token contract;
- The number of minimum investments;
- The commission amount;
- The bonus time;
- The interest bonuses;

getInvestorsArray – returns all the addresses of investors in an array.
getRaisedWei – returns the amount of investment received.
getSoldTokens – returns the number of tokens purchased.

getInvestorBalances – users need to pass the address of the investor and it returns:
- How much has the investor invested;

- How many tokens should the investor receive;
- How much Ether did the investor withdraw (if he won the dispute);
- How many tokens the investor withdrew;
- Did the investor call the refundETH method or not.

The contracts are deployed in Ropsten.

Mediators Pool Smart Contract –
https://ropsten.etherscan.io/address/0x966E524800E268B0Ee271C98B572416e90d6C28D

Responsible Crowdsale Smart Contract

The Responsible Crowdsale contract is the main place, where the Operator and Investors need to interact. This smart contact contains all the information about circles, Milestones, investors' balances, checks, validations etc. Each Milestone has its unique Hash and all sides need to use that hash if they want to interact with that Milestone. Each Milestone has its hard coded start time date and other information, which is immutable after its creation. The investors can open disputes starting from 3 days until the Milestone start date. The funds for the first Milestone can be collected by the Operator immediately after the Milestone start time and the Investors cannot open disputes, but for second and other Milestones, the Investors can open disputes. The Investor and Operator can withdraw their assets only after 3 days have passed from the Milestone date and after all the disputes have been resolved.

The Operator can:
- Add new circles / Milestones;
- Collect Milestone investments;
- Transfer all tokens and ETH from the Crowdsale contract to another contract in emergency form (after the Cluster owner's transaction).

The Investors can:
- Refund their funds if the Crowdsale is still open;
- Collect the Milestone tokens (or ETH, if the investor won the dispute);
- If they want to open a dispute, they need to call the method from the Cluster contract.

Responsible Crowdsale Smart Contract –

https://git.kryptonhub.com/jury-online/yet-another-contract-repository/tree/ararat_tonoyan/src

How to start working with Jury.Online smart contracts.

For launching the platform, users need to:
- Deploy the Cluster smart contract;
- Add new Backend address, which can deploy new Crowdsales;
- The backend needs to deploy a Crowdsale using the method with this method: function addCrowdsale(uint256 rate, address token, uint256 openingTime, uint256 closingTime, address payable operator, uint256[] calldata bonusFinishTimestamp, uint256[] calldata bonuses, uint256 minInvestmentAmount, uint256 fee), where:
1. The rate of ETH/TOKEN;
2. Address of the user's ERC20 token contract;
3. Crowdsale opening time (UNIX format);
4. Crowdsale closing time (UNIX format);
5. Address of the Operator;
6. Time until them the investors can get bonuses (UNIX format);
7. Amount of bonus for each time;
8. Minimum investments amount;
9. The fee which will be transferred to cluster smart contract automatically when somebody will invest. This method will return the address of the new Crowdsale contract.

- After Crowdsale creation, the Operator needs to set up Circles with Milestones, only after that Investors can send investments to Crowdsale contract address (when it will be opened).

For adding a new Circle, the Operator needs to call the function: addCircle(uint256 tokenPercent, uint256 ethPercent, bytes32[] memory MilestonesNames, uint256[] memory MilestonesTokenPercent, uint256[] memory MilestonesEthPercent,uint256[] memory MilestonesStartTimestamps), where:
1. The token percent of the circle;
2. The ETH percent of the circle;
3. The array of the Milestones names (bytes32 format);
4. The array of the Milestones token percentage;
5. The array of the Milestones ETH percentage;
6. The array of start dates of Milestones (UNIX format).

(NOTE, each timestamp should be bigger than the previous, and the time of the start of the first Milestone should be bigger than the time of his previous Circles last Milestone time)
- That's all, for receiving ETH, the Investors need to transfer ETH to the Crowdsale address, within the closing timestamp.

The Jury.Online smart contracts have taken into account all of the situations that could arise in a dispute and their resolution is provided for in a fair and transparent manner. Jury.Online is neither a new blockchain nor an old service attempting to integrate blockchain technologies. Jury.Online is a project that aims to use the full power of contemporary technologies to solve problems that were considered unsolvable for decades.

11- Company Information

Website: https://about.jury.online/
Jury.Online Foundation OU
Registration code: 14314305
Tartu mnt 83-701 Tallinn Harjumaa 10115
Email: support@jury.online