Kleros

Short Paper v1.0.7

Clément Lesaege, Federico Ast, and William George

September 2019

Abstract

Kleros is a decentralized application built on top of Ethereum that works as a decentralized third party to arbitrate disputes in every kind of contract, from very simple to highly complex ones. It relies on game theoretic incentives to have jurors rule cases correctly. The result is a dispute resolution system that renders ultimate judgments in a fast, inexpensive, reliable and decentralized way.

1 Introduction

"Whoever controls the courts, controls the state". Aristotle.

The world is experiencing an accelerated pace of globalization and digitalization. An exponentially growing number of transactions are being conducted online between people across jurisdictional boundaries. If the blockchain promise comes to a reality, in a not so distant future, most goods, labor and capital will be allocated through decentralized global platforms. Disputes will certainly arise. Users of decentralized eBay will claim that sellers failed to send the goods as specified in the agreement, guests in decentralized Airbnb will claim that the rented house was not "as shown in the pictures" and backers in a crowdfunding platform will claim a refund as teams fail to deliver the promised results.

Smart contracts are smart enough to automatically execute as programmed, but not to render subjective judgments or to include elements from outside the blockchain. Existing dispute resolution technologies are too slow, too expensive and too unreliable for a decentralized global economy operating in real time. A fast, inexpensive, transparent, reliable and decentralized dispute resolution mechanism that renders ultimate judgments about the enforceability of smart contracts is a key institution for the blockchain era.

Kleros is a decision protocol for a multipurpose court system able to solve every kind of dispute. It is an Ethereum autonomous organization that works as a decentralized third party to arbitrate disputes in every kind of contract, from very simple to highly complex ones. Every step of the arbitration process (securing evidence, selecting jurors, etc.) is fully automated. Kleros does not rely on the honesty of a few individuals but on game-theoretical economic incentives.

It is based on a fundamental insight from legal epistemology: a court is an epistemic engine, a tool for ferreting out the truth about events from a confusing array of clues. An agent (jury) follows a procedure where an input (evidence) is used to produce an output (decision) (20). Kleros leverages the technologies of crowdsourcing, blockchain and game theory to develop a justice system that produces true decisions in a secure and inexpensive way.

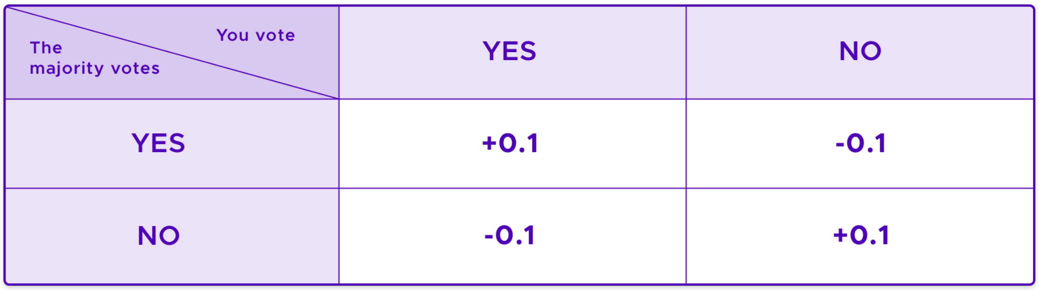
2 Previous Work: SchellingCoin Mechanism

Game theorist Thomas Schelling developed the concept of Schelling Point (also known as Focal Points) (25) as a solution that people tend to use to coordinate their behavior in the absence of communication, because it seems natural or relevant to them. Schelling illustrated the concept with the following example: “Tomorrow you have to meet a stranger in NYC. Where and when do you meet him?”. While any place and time in the city could be a solution, the most common answer is “noon at the information booth at Grand Central Terminal”. There is nothing that makes noon at Grand Central Terminal a location with a higher payoff (any other place and time would be good, provided that both agents coordinate there), but its tradition as a meeting place makes it a natural focal point.

Schelling Points typically arise when communication is impossible, but also when, while communication is possible, no party can provide the other with a reason to believe that what he says is true (16). Based on the concept of Schelling Points, Ethereum founder Vitalik Buterin has proposed the creation of the SchellingCoin (9), a token that aligns telling the truth with economic incentives. If we wanted to know if it rained in Paris this morning, we could ask every owner of a SchellingCoin: “Has it rained in Paris this morning? Yes or No”. Each coin holder votes by secret ballot and after they have all voted, results are revealed. Parties who voted as the majority are rewarded with 10% of their coins. Parties who voted differently from the majority lose 10% of their coins.

Thomas Schelling (25) described “focal point(s) for each person’s expectation of what the other expects him to expect to be expected to do”. SchellingCoin uses this principle to provide incentives to a number of agents who do not know or trust each other to tell the truth. We expect agents to vote the true answer because they expect others to vote the true answer, because they expect others to vote for the true answer... In this simple case, the Schelling Point is honesty.

SchellingCoin mechanisms have been used for decentralized oracles and prediction markets (26) (23) (3). The fundamental insight is that voting coherently with others is a desirable behavior that has to be incentivized. The incentives design underlying Kleros is based on a mechanism similar to SchellingCoin, slightly modified in order to answer to a number of specific challenges regarding scaling, subjectivity and privacy to make agents engage in adequate behavior.



You vote The majority votes	YES	NO
YES	+0.1	-0.1
NO	-0.1	+0.1

Figure 1: Payoff table for a basic Schelling game

3 A Use Case

Alice is an entrepreneur based in France. She hires Bob, a programmer from Guatemala, on a P2P freelancing platform to build a new website for her company. After they agree on a price, terms and conditions, Bob gets to work. A couple of weeks later, he delivers the product. But Alice is not satisfied. She argues that the quality of Bob's work is considerably lower than expected. Bob replies that he did exactly what was in the agreement. Alice is frustrated. She cannot hire a lawyer for a claim of just a couple hundred dollars with someone who is halfway around the world.

What if the contract had a clause stating that, should a dispute arise, it would be solved by a Kleros court? Kleros is a decentralized application built on Ethereum. After Bob stops answering her email, Alice taps a button that says "Send to Kleros" and fills a simple form explaining her claim.

Thousands of miles away, in Nairobi, Chief is a software developer. In his "dead time" on the bus commuting to his job, he is checking Kleros Court website (court.kleros.io) to find some arbitration work. He makes a couple thousand dollars a year on the side of his primary job by serving as a juror in software development disputes between freelancers and their clients. He usually rules cases in the Website Quality subcourt. This court requires skills in html, javascript and web design to solve disputes between freelancers and their customers. Chief stakes 2000 PNK, the token used by Kleros to select jurors for disputes. The more tokens he stakes, the more likely is that he will be selected as juror.

About an hour later, an email hits Chief's inbox: "You have been selected as a juror on a website quality dispute. Download the evidence here. You have three days to submit your decision". Similar email are received by Benito, a programmer from Cusco and Alexandru, from Romania, who had also staked their PNK in the Website Quality subcourt. They were selected randomly from a pool of almost 3,000 candidates. They will never know each other, but they will collaborate to settle the dispute between Alice and Bob. On the bus back home, Chief analyzes the evidence and votes who is right.

Two days later, after the three juries have voted, Alice and Bob receive an email: "The jury has ruled for Alice. The website was not delivered in accordance to the terms and conditions agreed by the parties. A smart contract has transferred the money to Alice". Jurors are rewarded for their work and the case is closed.

4 Project Description

4.1 Arbitrated Contracts

Kleros is an opt-in court system. Smart contracts have to designate Kleros as their arbitrator. When they opt-in, contracts creators choose how many jurors and which court will rule their contract in case a dispute occurs[1]. The idea is that they will choose a type of court specialized in the topic of the contract. A software development contract will choose a software development court, an insurance contract will select an insurance court, etc. Figure 2 shows an example of the court arborescence from which users can choose. The Kleros team has developed a number of standard contracts using Kleros as a dispute resolution mechanism. Moreover, we have proposed standards (21) (27) that would allow others to develop other contracts in a way that does not require anticipating which dispute resolution mechanism will ultimately be used.

4.1.1 Options for jurors

Contracts will specify the options available for jurors to vote. In the introductory example, options may be: "Reimburse Alice", "Give Bob one extra week to finish the website" and "Pay Bob".

The smart contract will also specify the behavior of the contract after the ruling is done. In the example:

[1] For more information about courts, see the Court Arborescence section.

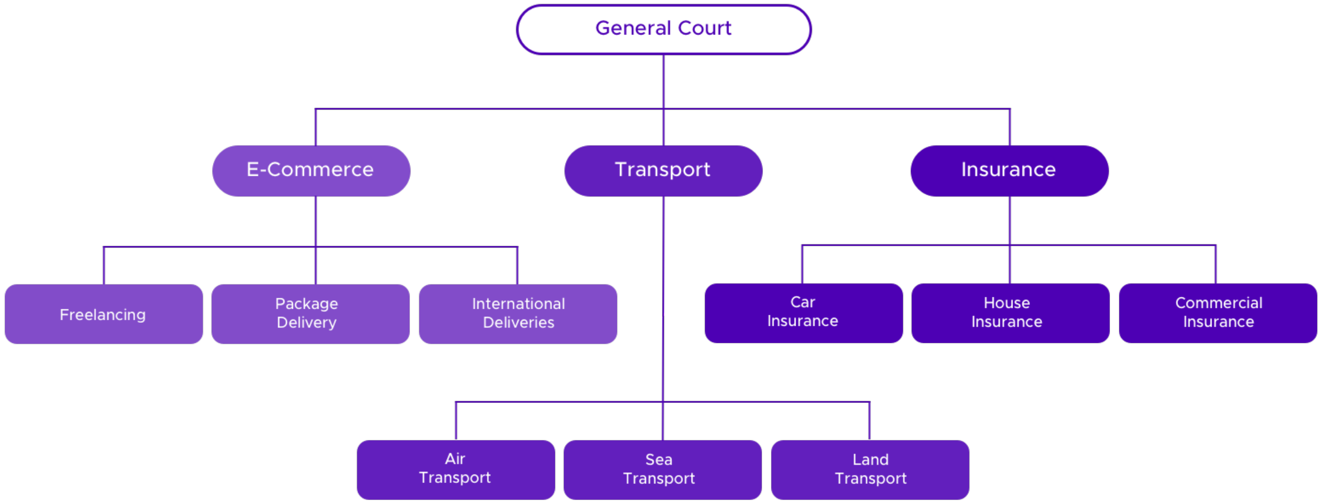


Figure 2: Example of court arborescence from which smart contract creators must select a court.

- “Reimburse Alice” transfers funds to Alice’s address.
- “Give Bob one extra week to finish the website” advances the timers for how long Bob has to finish the website one week, i.e. blocks Alice from creating new disputes during this time. Furthermore, the smart contract might be written so that if this option has been chosen once, it cannot be selected in further disputes.
- “Pay Bob” transfers funds to Bob’s address.

4.2 Drawing Jurors

4.2.1 System token: the pinakion (PNK)

Users have an economic interest in serving as jurors in Kleros: collecting arbitration fees in exchange for their work. Candidates self-select to serve as jurors using a token called pinakion (PNK)[2].

The probability of being drawn as a juror for a specific dispute is proportional to the amount of tokens a juror stakes. The higher the amount of tokens he stakes, the higher the probability that he will be drawn as juror. Jurors that do not stake PNK do not have the chance of being drawn. This prevents inactive jurors from being selected.

PNK plays two key functions in Kleros design.

First, it protects the system against the sybil attack (14). If jurors were simply drawn randomly, a malicious party could create a high number of addresses to be drawn a high number of times in each dispute. By being drawn more times than all honest jurors, the malicious party would control the system.

Second, PNK provides jurors the incentive to vote honestly[3] by making incoherent jurors, i.e. jurors whose votes do not agree with the ultimate ruling, pay part of their stake to coherent ones.

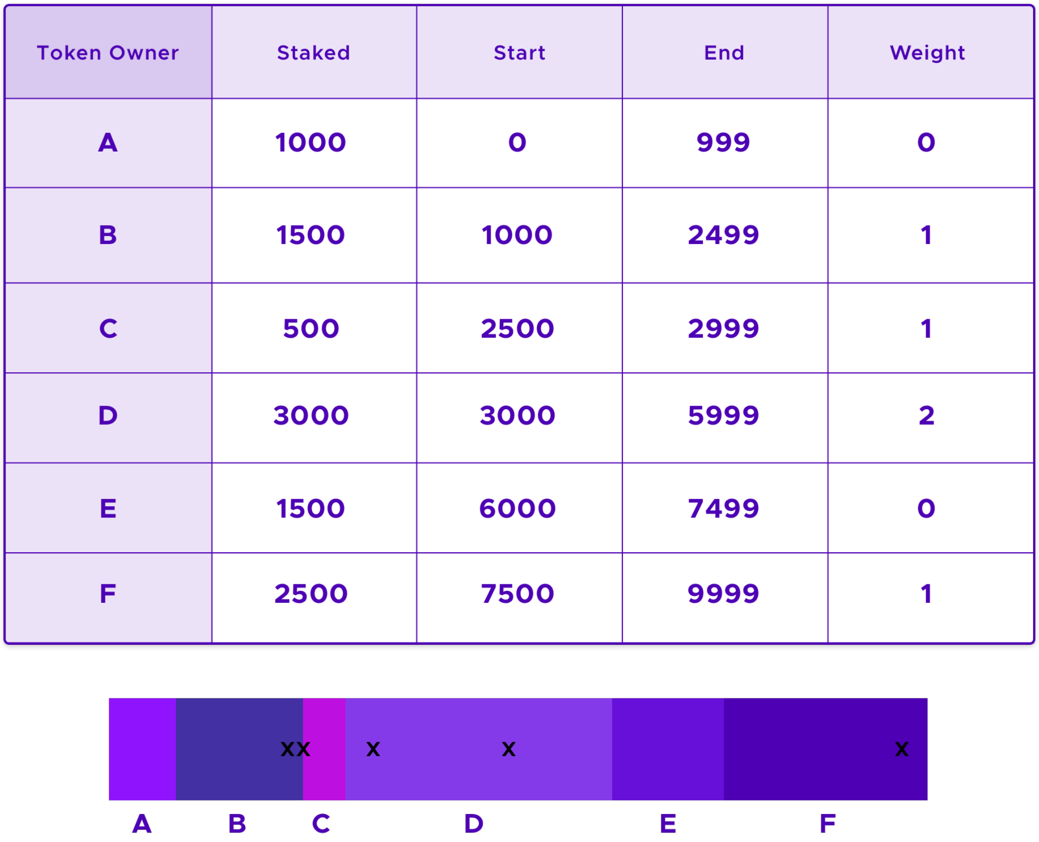
[2] The name is a reference to the pinakion, the bronze plaque that each Athenian citizen used as an ID. The pinakion was used as a token for jury selection in Athens popular trials.

[3] See the section Incentive System.

4.2.2 Jury selection

After candidates have self-selected into specific courts and staked their tokens, the final selection of jurors is done randomly. The probability of being drawn as a juror is proportional to the amount of staked tokens. Theoretically, a candidate may be drawn more than once for a specific dispute (but in practice it is unlikely). The amount of times a user is drawn for a dispute (called its weight) determines the number of votes he will get in the dispute and the amount of tokens he will win or lose during the token redistribution.

Imagine that 6 token owners signed up for the dispute and staked 10,000 in total with the following distribution:



Token Owner	Staked	Start	End	Weight
A	1000	0	999	0
B	1500	1000	2499	1
C	500	2500	2999	1
D	3000	3000	5999	2
E	1500	6000	7499	0
F	2500	7500	9999	1

Figure 3: Example of tokens staked and drawn jury members.

For a dispute that requires 5 votes, 5 tokens are drawn out of the 10,000 that were staked. The drawn tokens (as represented in Figure 3) are number 2519, 4953, 2264, 3342 and 9531. Token owners B, C and F are drawn with a weight of 1. Token owner D is drawn with a weight of 2. Staked PNK (except those paid by incoherent jurors) can be taken back after the court reaches a final decision.

4.2.3 Random number generation

In order to draw jurors, we need a process to draw random numbers resistant to manipulation. Using a protocol for creating a random number between two parties (5) does not work. An attacker could create disputes between himself, select himself as a juror multiple times and select another victim juror. He would then coordinate his own votes in a way that they would be considered coherent but not those of the victim in order to steal tokens from the victim when PNK are redistributed (see the section Incentive System).

Currently, the random numbers used to select jurors are drawn from blockhashes of Ethereum blocks. While these values remain impossible to predict in advance, miners can opt not to release a block that would result in random numbers unfavorable to them at the expense of forfeiting a block reward. In the future, to produce random numbers that cannot even be manipulated by attackers with the capabilities of large miners, these values will be generated with sequential proof of work (12) using a scheme similar to Bünz et al. (13) (See the section on Future Work.)

4.3 Votes

After assessing the evidence, jurors commit (7) their vote to one of the options. They submit $\text{hash}(\text{vote}, \text{salt}, \text{address})$[4]. The salt is a random value generated locally in order to add entropy to prevent the use of rainbow tables. The address is the Ethereum address of the juror. It is required in order to make the commitment of each juror different, thus preventing a juror from copying the commitment of another. When the vote is over, they reveal $\{\text{vote}, \text{salt}\}$, and a Kleros smart contract verifies that it matches the commitment. Jurors failing to reveal their vote are penalized (see Incentive System section).

After a juror has made a commitment, his vote cannot be changed. But it is still not visible to other jurors or to the parties. This prevents the vote of a juror from influencing the votes of others.

Jurors can still declare that they voted in a certain way, but it is challenging for them to provide other jurors a reason to think that what they say is true. This is an important feature for the Schelling Point to arise. If jurors knew the votes of other jurors, they could vote like them instead of voting for the Schelling Point. See the Future Work section for a further discussion of these ideas.

After all jurors have voted (or after the time to vote is over), votes are revealed by jurors. Jurors that fail to reveal their vote are penalized. Finally, votes are aggregated and the smart contract is executed. The option with the highest amount of votes is considered as the winning one[5].

As this two step processes of committing and then revealing one's vote requires additional user interactions, in some low stakes subcourts, one might want votes to be issued publicly to simplify the user experience[6]. Which system is used is determined via a subcourt parameter (see the Governance Mechanism section).

4.4 Arbitration Fees

In order to compensate jurors for their work and avoid an attacker from spamming the system, creating disputes and appealing requires arbitration fees. Each juror who is coherent with the final ruling will be paid a fee determined by the subcourt where the dispute is solved. The arbitrable smart contract will determine which party will pay the arbitration fee.

The rules can be simple. For example, they may require the party creating the dispute or the party appealing to pay the fee. But we may think of more complex rules to create better incentives. For example:

[4] Throughout this paper we use hash referring to a cryptographic hash function, in Ethereum the one used is keccak256.

[5] We are considering methods more complex than first-pass-the-post. But the challenge is to deal with the asymmetry in the incentive matrix they lead to. This asymmetry could affect the Schelling Point. For example taking the median value in case of values which can be ordered could lead to a bias toward center values.

[6] Note that as Kleros uses an appeal system, even if a majority of votes in a voting round have already been cast for a given choice, voting for that choice does not guarantee coherence with the ultimate result (see the Incentive System section). This limits the effectiveness of a vote copying strategy. Hence public votes might be acceptable in some cases.

- In first instance, each party will deposit an amount equal to the arbitration fee in the smart contract. If one party fails to do so, the smart contract will consider that the court ruled in favor of the party who deposited the arbitration fee (without even creating a dispute in the court). If both parties deposit the funds, the winning party will be reimbursed when the dispute is over.
- In appeals, both parties have to deposit the arbitration fees. The appellant also has to deposit an extra stake proportional to the appeal fees which will be given to the party winning the dispute. This way if a party makes frivolous appeals to harm the opposing party, the opposing party will get a compensation for lost time, while if the appeals are finally ruled to be legitimate, the stake will be returned to the appellant[7].

A discussion about the fee structure defined by arbitrable smart contracts will be part of future work.

4.5 Appeals

If, after the jury has reached a decision, a party is not satisfied (because it thinks the result was unfair), it can appeal and have the dispute ruled again. Each new appeal instance will have twice the previous number of jurors plus one. Due to the increased number of jurors, appeal fees must be paid ($\text{appeal\_fees} = \text{new\_amount\_jurors} \cdot \text{average\_fee\_per\_juror}$).

Due to arbitration fees being paid to a number of jurors which increases exponentially as one appeals, arbitration fees also rise exponentially with the number of appeals. This means that, in most cases, parties won't appeal, or will only appeal a moderate amount of times. However, the possibility of appealing a high number of times is important to prevent an attacker from bribing the jurors (See the section on Bribe resistance).

4.6 Incentive System

Again, we say a jury member voted coherently if it voted for the option chosen by the majority.

Jurors rule disputes in order to collect arbitration fees. They are incentivized to rule honestly because, after a dispute is over, jurors whose vote is not coherent with the group will not receive their arbitration fees and furthermore they lose will some tokens. Then both the arbitration fees they would have received if coherent and their lost stakes of tokens are given to coherent jurors.

After Kleros has reached a decision on the dispute, tokens are unfrozen and redistributed among jurors. The redistribution mechanism is inspired by the SchellingCoin[8], where jurors gain or lose tokens depending on whether their vote was consistent with the others jurors.

The amount of tokens lost per incoherent juror is : $\alpha \cdot \text{min\_stake} \cdot \text{weight}$. The α parameter determines the number of tokens to be redistributed after a ruling. It is an endogenous variable that will be defined by the governance mechanism as a consequence of the internal dynamics of the voting environment. The min_stake parameter is the minimum amount of tokens which can be staked in the subcourt.

Arbitration fees and the lost tokens are divided between the coherent parties proportionally to their weight[9]. An example of token redistribution is shown in Figure 4. Jurors could fail to reveal

[7] This requires post-dispute insurers for parties not having a sufficient capital to deposit appeal and stake deposits. The insurer would pay the deposit of a party in exchange of part of the stake if the dispute is won. All of this can be smart contract enforced.

[8] See section Previous Work: SchellingCoin mechanism

[9] Token redistribution mechanisms are still under active research and we may end up with a more sophisticated protocol in future work.

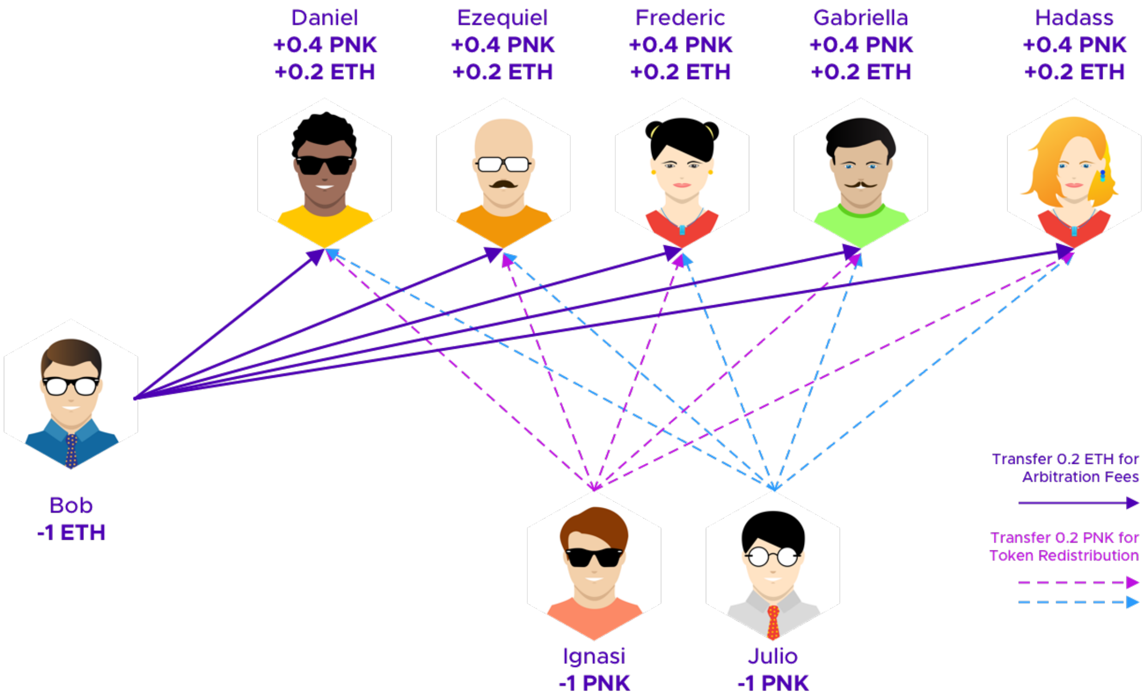


Figure 4: Token redistribution after the vote with seven jurors. Tokens are redistributed from jurors who voted incoherently to jurors who voted coherently. Bob lost the dispute and pays the arbitration fees. The other deposits are refunded.

their vote. To disincentivize this behaviour, the penalty for not revealing one's vote is at least as large as the penalty for voting incoherently. This incentivizes jurors to always reveal their vote. In case of appeal, arbitration fees and tokens are redistributed at each level according to the result of the final appeal[10].

When there is no attack, parties are incentivized to vote what they think, other parties think, other parties think… is honest and fair. In Kleros, the Schelling Point is honesty and fairness. One could argue that those decisions being subjective (for example, compared to a SchellingCoin mechanism for a prediction market), no Schelling Point would arise. In (25), the informal experiments run by Thomas Schelling showed that in most situations a Schelling Point plebiscited by all parties does not exist. But Schelling found that some options were more likely to be chosen than others. Therefore even if a particularly obvious option does not exist, some options will be perceived as more likely to be chosen by others parties and will effectively be chosen. We cannot expect jurors to be right 100% of the time. No arbitration procedure could ever achieve that. Some times, honest jurors will lose coins. But as long as overall they lose less value than what they win as arbitration fees and as coins for other incoherent parties, the system will work[11].

[10] If at one level no one voted coherently, what to do with the amounts from that level can be determined by the governance procedure; for example the tokens could be given to the winning party.

[11] Indeed, note that, so far, this idea has largely worked as expected in experiments such as those considered in (17) as well as in practical applications such as those of (19) and (23).

4.7 Attack Resistance

4.7.1 Buying half of the tokens

If a party (or a group of parties colluding) were to buy half of the tokens, it would control the results in the General Court and therefore could ultimately decide all results. However, having a party buying more than half the tokens is highly unlikely if these are fairly distributed. First, half of the tokens should be available for sale, which is not guaranteed. Moreover, the fact that one party could afford all the tokens at current market price does not mean it would be able to buy half of them. Tokens, contrary to most physical assets, have increasing marginal costs. They will be dynamically priced on exchanges. Should one party buy a significant part, the price would go up due to market depth making it increasingly more costly to acquire tokens.

4.7.2 Bribe resistance

Appeals are an important mechanism against bribes. Bribing a small jury is relatively easy. But since the victim always has the right to appeal, the attacker would have to keep bribing larger and larger juries at a steeply rising cost. The attacker would have to be prepared to spend an enormous amount of money to bribe jurors all the way to the General Court and would very likely lose in the end. To control the verdict of the whole court, the attacker would need to bribe token holders holding more than 50% of the PNK in total.

This attack doesn't work in the honest majority model (where more than half of the tokens are controlled by honest parties who won't accept the bribe). But even with a dishonest majority (majority of token holder only searching to optimize their profit), the system can withstand bribing attacks under certain conditions.

A successful bribe of the General Court would dramatically decrease the value of the tokens (who wants his contracts to be arbitrated by a dishonest court?). Therefore, an attacker should be able to provide more value than 50% of the expected loss from the price drop in order for his bribing offer to be successful (which in almost all cases would exceed the value at stake in the dispute). In practice, a party appealing every decision all the way to the General Court would be extremely unlikely. However, the possibility needs to exist for incentives to be correctly balanced.

A more elaborate attack (the p + epsilon attack) could be done, promising to pay the bribe only if the attack is unsuccessful. This attack requires a high budget but has zero cost if successful. Already in (11) there is a proposed game theoretic response against this attack where jurors use a mixed strategy (jurors only accept the bribe with a defined probability which increases their expected reward compared to accepting the bribe). Furthermore, we conducted experiments on the Kleros "Doges on Trial" pilot testing user behaviour in the event of a p + epsilon attack. See (17) for the results of these experiments and comments on how the Kleros appeal system makes such attacks less viable.

4.8 Court Arborescence

When registering as jurors, users start in the General Court and follow a path to a specific subcourt according to their skills. Each subcourt has some specific features regarding policies, session time, cost, number of drawn jury members and tokens staked. Each token holder can register a given token in at most one subcourt of each court in which they have token staked. Figure 5 show an example of allowable registrations.

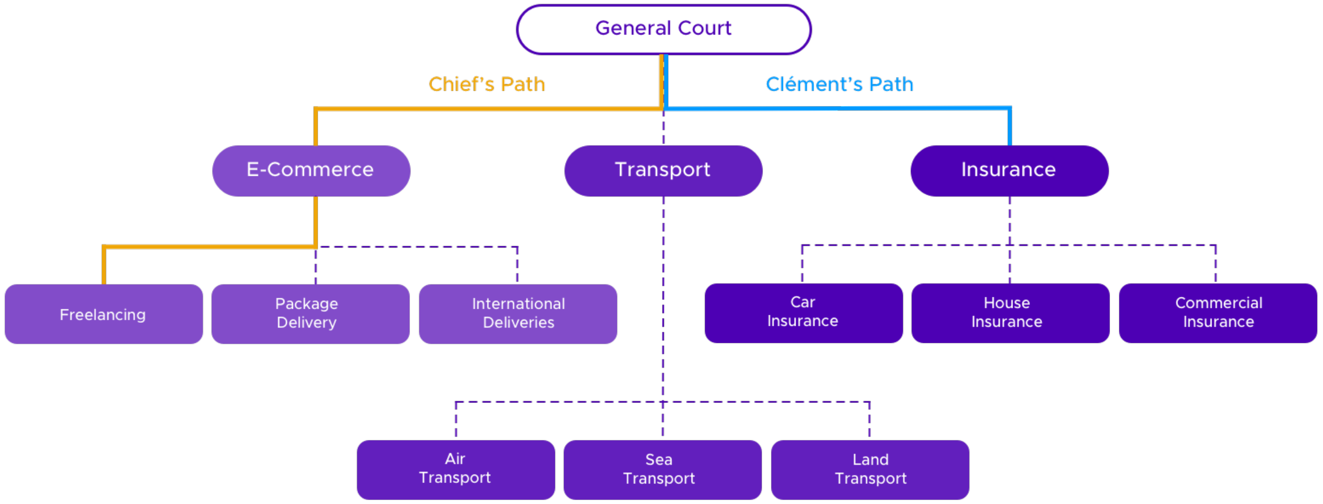


Figure 5: Example of paths chosen by jurors in the subcourt system. Clément can be drawn as juror in the General Court and in the Insurance Subcourt. Chief can be drawn as juror in the General Court, in the E-Commerce Subcourt and in the Freelancing Subcourt.

Asking jurors to choose between subcourts incentivizes them to choose the subcourts they are the most skilled at. If they were able to choose every subcourt, some would choose all of them to get the maximum amount of arbitration fees from their tokens.

4.9 Governance Mechanism

As the Kleros protocol gains users and use cases, it will be necessary to create new subcourts, to make changes in subcourt policies and parameters and to update the platform to new versions with additional features. Such decisions will be made by token holders using a liquid voting mechanism (15). Token holders will have an amount of votes equal to the amount of PNK they hold. The governance mechanism can be used to:

1. Set policies: Policies are guidelines about how to arbitrate disputes. They are the equivalent of the laws in traditional justice systems. They determine which party should win a dispute when particular conditions are met. They can be specific to a particular subcourt.
2. Add new subcourts.
3. Modify parameters in subcourts:
   (a) Arbitration fees.
   (b) Time of each court session.
   (c) Minimum amount of tokens to be staked.
4. Change one of the smart contracts Kleros rely on. This allows arbitrary changes. This can be used for improvements or in an emergency if it appears that some elements of Kleros are not working properly[12].

[12] Audits and reviews will be made before the code is deployed. But we can never guarantee at 100% that there is

5 Future Work

In this section we discuss a number of planned improvements to the protocol.

5.1 Privacy of Contracts

Solving disputes may require parties to disclose privileged information with jurors. In order to prevent outside observers from accessing this information, in the future, the natural language contracts (English or other) and the labels of the jurors voting options will not be publicly released, and in particular they will not be put on the blockchain. When the contract is created, the creator will submit hash(contract_text, option_list, salt) (where contract_text is the plain English text of the contract, option_list the labels of the options which can be voted by jurors and salt is a random number to avoid the use of rainbow tables).

The contract creator will send {contract_text, option_list, salt} to each party using asymmetric encryption. This way, parties can verify that the submitted hash corresponds to what was sent to them. In case of a dispute, each party can reveal {contract_text, option_list, salt} to jurors which can verify that they correspond to the hash submitted. They can do so using asymmetric encryption such that only the jurors receives the text of the contract and of the options. All these steps will be handled by the application users will run while using Kleros.

5.2 Improved Random Number Generation

As we saw above, jurors are currently drawn using randomness based on the blockhashes of Ethereum blocks. However this has the drawback that large miners can bias juror selection at the expense of forfeiting block rewards. In this section we will detail a planned protocol that provides a more robust source of randomness based on sequential proof of work (12) using a scheme similar to Bünz et al. (13) adapted to also work for Proof-Of-Stake blockchains as follows[13].

1. **Initialization**: We start with seed=blockhash and let all parties input a value localRandom to change the seed such that seed=hash(seed,localRandom). This allows any party to change the seed. We want the seed to not be chosen by any one party. This way every party can change the seed, but not choose it, because choosing a particular seedAttack would require the attacker to determine localRandom such that hash(seed,localRandom)=seedAttack which is difficult due to the preimage resistance of cryptographic hash functions.

2. **Computing the master random value**: Every party who has a stake in the random number runs sequential proof of work on the seed. Starting with h_0 = seed, they compute h_{n+1} = hash(h_n) up h_d where d is the difficulty parameter. Computing h_d takes time and assures that a certain amount of time passed between someone gets the knowledge of the seed and that he gets the result. The difficulty d is fixed such that no hardware can compute h_d during the time of the initialization phase. Because we need the result of the previous step before starting the next one, this process can't be parallelized. This means that no party will be able to obtain the results significantly faster than the others.

3. **Getting the results on the blockchain**: Every party can post the h_d with a deposit they found. Then other parties can disprove results which are wrong using interactive verification

not a bug (either on the code or incentives) somewhere. Having this failsafe provides extra security.

[13] In Proof-Of-Work blockchains, as the blockhash remains impossible to exactly predict, we can remove this step and only use the blockhash as a seed. However, Ethereum has planed to switch to Proof-Of-Stake.

(24). It consists of a dichotomic search on the results of the attacker. If an attacker submits a false h_d, an honest party can ask him his $h_{d/2}$ value. If he gives the wrong value, there is an error in the attacker values between h_0 and $h_{d/2}$. If he gives the right value, there is an error between $h_{d/2}$ and h_d. Either way, the search space is divided by two. The honest party continues this process on a reduced space (where the error is) until two values are left. Then the honest party can exhibit x such that $h_{x+1} \neq \text{hash}(h_x)$ in the attacker answer which invalidates his answer. Parties whose answer is invalidated lose their deposit. Part of it is burnt and the other part is given to the party that invalidated them. Note that the number of interactions required to invalidate a false result is only $O(\log(d))$.

4. **Getting all random values**: After the honest parties have invalidated the results, there is only the correct result h_d left. From this master random value we derive all the random values such that $r_n = \text{hash}(h_d, n)$.

The output of this process is a random number as long as there is at least one honest party. Computing the sequential proof of work and the interactive verification takes time. But for most disputes waiting a few hours from the moment the dispute starts and the moment jurors are drawn will not be a problem. However, for some subcourts with a particularly low session time (for example a subcourt solving disputes in a web to blockchain Oracle) this random number generation method could be too slow. Another possible random number generator, which is less secure, that could possibly be used by such subcourts is one based on threshold signatures (6).

5.3 Penalizing Jurors Who Reveal Their Vote Too Early

Above we described a commit and reveal scheme that allows jurors to keep their vote hidden until the votes of all jurors are committed to. However, this scheme does not in itself prevent jurors from nonetheless publishing their votes in an attempt to influence other jurors. In this section we discuss a potential future improvement that would incentivize jurors to not reveal their votes prior to the reveal stage.

We propose to let any party able to show the commitment of a juror to Kleros before the vote is closed steal the PNK of this juror and invalidate the vote of this juror. Then, if a juror wants to reveal its vote to another party, it has two options:

1. Reveal only its vote. The party won't have any proof that it effectively voted that way. The juror could lie about it and the other party has no way to verify.

2. Reveal its vote and its commitment. The party would have the proof of its vote, but the party would also be able to steal the PNK of this juror.

This scheme will impede jurors from revealing their votes trustlessly[14]. A more complete potential solution that we are examining for eventual inclusion in Kleros is detailed in (10).

5.4 Liquid Voting in Governance

In the section on the Governance mechanism above we described how token holders can make a number of decisions for the platform. In this section, we describe a future plan to allow token holders

[14] It is still possible for jurors to give insight about their votes. For example by making a smart contract with themselves committing to vote in a certain manner and burning a deposit if they vote differently. A discussion about these kinds of behaviour will be included in future work.

to delegate their vote if they choose not to vote directly. When a user fails to vote, his voting power will be automatically transferred to his delegate. You can see an illustration of the liquid voting mechanism in Figure 6. Vote delegation can also be subcourt specific. Users could choose to delegate their vote in some subcourts but not in others. Note that delegates do not need to be humans. They can be smart contract implementing arbitrarily complex voting rules (for example voting on updating fees based on market data).

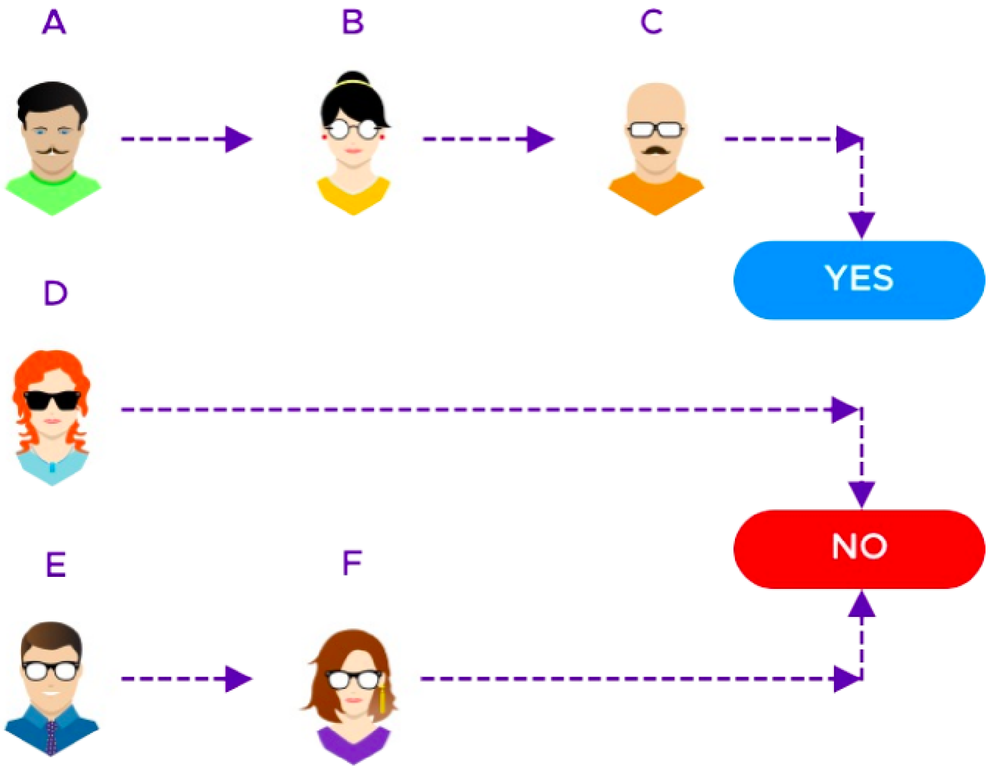


Figure 6: Illustration of a liquid vote

6 Applications

Kleros is a general, multipurpose system which can be used in a large number of situations. We present some examples of possible use cases:

- **Escrow :** To pay for an off-chain good or service, the funds can be put in a smart contract. After receiving the good or service, the buyer can unlock the funds to the seller. In case of dispute, Kleros can be used to have the smart contract either reimburse the buyer or pay the seller. Such a Kleros based escrow system is already available, see (18).

  Escrows can also be more complex. For example for a rental agreement, the renter can be required to pay a deposit. In case the property is damaged and the renter does not agree on a compensation, a dispute can be created by the owner to claim part of the security deposit.

- **Micro tasking:** Decentralized platforms could pay for microtasks (in the manner of the Amazon Mechanical Turk (1)). Taskers would put a security deposit and submit answers to microtasks. The tasks would be replicated. If a task gets different answers, taskers could admit their mistake, this would transfer a part of security deposit to the taskers who performed the task correctly. In case multiple taskers stay on their position, a dispute resolution process would ensue and the losing taskers would have part of their security deposit transferred to the winning ones.

- **Insurance :** The insuree will pay a fee to the insurer to get a compensation in case a particular event would happen. The insurer will have to put some security deposit which could be common to multiple insurees (respecting risk management rules). When an insured event happens, the insurer can validate it and compensate the insuree. If the insurer does not validate the event, a dispute resolution process would ensue. If the insuree wins the dispute resolution process, funds from the security deposit of the insurer would be transferred to the insuree. In case the security deposit is linked to multiple insurees claiming more than the deposit, a dispute resolution process would also be needed to determine how those funds should be split between insurees.

- **Oracle :** A decentralized data feed to be used by smart contracts was one of the early envisioned use cases of Ethereum (8). A party (which can be a smart contract) asks a question. Everyone can give a deposit and submit an answer. If everyone gives the same answer, it is returned by the Oracle. If there are multiple answers, a dispute resolution procedure ensues. The Oracle returns the answer given by the dispute resolution process and parties who put wrong answers lose their deposits which are given to honest submitters. Realitio provides an oracle service that is based on such principles, giving the option to use Kleros for the ensuing disputes (4). Moreover, other applications that use the Realitio oracle, such as CryptoUnlocked, (22), indirectly depend on this dispute resolution.

- **Curated lists:** Curated lists can be whitelists or blacklists. For example, a whitelist can list smart contracts having undertaken proper audit procedures. A blacklist can list the ENS (Ethereum Name Service (2)) names registered by parties having nothing to do with that name (for example, a malicious party could register "kleros-token-sale.eth", to scam people into sending funds to that address). Parties could submit items to the list by putting a security deposit. If no one contests that the item belongs to the list for a sufficient amount of time, the name is added and the deposit refunded. If some parties contest by putting a security deposit, a dispute resolution process ensues. If the item is considered belonging to the list, it is added and the submitter gets the deposits of the contesting parties. Otherwise, the deposit of the submitter is given to the contesting parties. Kleros is already being used for a token curated list of tokens that satisfy various properties (for example, such as being ERC20) (19).

- **Social networks:** Preventing spam, scams and other abuses is a challenge for decentralized social networks. Parties can report violations of the network policies and put a security deposit. If the violation is contested, a dispute resolution process ensues. If it is ruled that no violation happened, the reporter loses his security deposit to the accused party. If the violation is not contested or confirmed by Kleros various effects can be implemented: the content can be removed, the content poster can lose a sign-up deposit and the reach of his other posts can be lowered.

7 Conclusion

We have introduced Kleros, a decentralized court system allowing arbitration of smart contracts by crowdsourced jurors relying on economics incentives. You can see a summary of how Kleros works in Figure 7.

The rise of the digital economy created labor, capital and product markets that operate in real time across national boundaries. The P2P economy requires a fast, inexpensive, decentralized and reliable dispute resolution mechanism. Kleros uses game theory and blockchain in a multipurpose arbitration protocol capable of supporting a large number of applications in ecommerce, finance, insurance, travel, international trade, consumer protection, intellectual property and academia among many others. Cryptocurrencies are giving many the possibility of having their first bank account to send and receive money in a secure way. Cryptocurrencies are helping millions achieve financial inclusion. Kleros will do the same in access to justice by enabling arbitration in a large number of contracts that are too costly to pursue in court. Just as Bitcoin brought "banking for the unbanked", Kleros has the potential to bring "justice for the unjusticed".

References

[1] Amazon mechanical turk. https://www.mturk.com/.

[2] Ethereum name service. https://ens.domains/.

[3] Gnosis. https://gnosis.pm/.

[4] Ast, F. Kleros-realitio oracle service - getting real information on-chain. https://blog.kleros.io/the-kleros-realit-io-oracle/, 2019.

[5] Blum, M. Coin flipping by telephone a protocol for solving impossible problems. *SIGACT News 15*, 1 (Jan. 1983), 23–27.

[6] Boneh, D., Lynn, B., and Shacham, H. Short signatures from the weil pairing. *Journal of Cryptology 17*, 4 (Sep 2004), 297–319.

[7] Brassard, G., Chaum, D., and Crépeau, C. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci. 37*, 2 (Oct. 1988), 156–189.

[8] Buterin, V. Ethereum, a next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper, 2014.

[9] Buterin, V. Schellingcoin: A minimal-trust universal data feed. https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed/, 2014.

[10] Buterin, V. On anti-pre-revelation games. https://blog.ethereum.org/2015/08/28/on-anti-pre-revelation-games/, 2015.

[11] Buterin, V. The p + epsilon attack. https://blog.ethereum.org/2015/01/28/p-epsilon-attack/, 2015.

[12] Buterin, V. Introduction to cryptoeconomics. https://edcon.io/ppt/one/Vitalik%20Buterin_Introduction%20to%20Cryptoeconomics_EDCON.pdf, 2017.

[13] Bünzy, B., Goldfeder, S., and Bonneauy, J. Proofs-of-delay and randomness beacons in ethereum.

[14] Douceur, J. R. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems* (London, UK, UK, 2002), IPTPS '01, Springer-Verlag, pp. 251–260.

[15] Ford, B. Delegative democracy. http://www.brynosaurus.com/deleg/deleg.pdf, 2002.

[16] Friedman, D. A positive account of property rights. *Social Philosophy Policy 11* (1994).

[17] George, W. Doges on trial curated list observations part 2 - deep dive edition. https://blog.kleros.io/cryptoeconomic-deep-dive-doges-on-trial/, 2018.

[18] James, S. Kleros escrow explainer - secure your blockchain transactions today. https://blog.kleros.io/kleros-escrow-secure-your-blockchain-transactions-today/, 2019.

[19] James, S. Kleros TCR - a deep dive explainer. https://blog.kleros.io/kleros-ethfinex-tcr-an-explainer/, 2019.

[20] Laudan, L. *Truth, Error, and Criminal Law: An Essay in Legal Epistemology.* Cambridge Studies in Philosophy and Law. Cambridge University Press, 2006.

[21] Lesaege, C. ERC 792: Arbitration standard. https://github.com/ethereum/EIPs/issues/792, 2017.

[22] Long, P. Cryptounlocked oracle upgrade. https://blog.wetrust.io/cryptounlocked-oracle-upgrade-5c8b22e3375b, 2019.

[23] Peterson, J., and Krug, J. Augur: a decentralized, open-source platform for prediction markets. http://bravenewcoin.com/assets/Whitepapers/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf/, 2015.

[24] Reitwiessner, C. From smart contracts to courts with not so smart judges. https://blog.ethereum.org/2016/02/17/smart-contracts-courts-not-smart-judges/, 2016.

[25] Schelling, T. C. *The strategy of conflict.* Oxford University Press, 1960.

[26] Sztorc, P. Truthcoin, peer-to-peer oracle system and prediction marketplace. http://www.truthcoin.info/papers/truthcoin-whitepaper.pdf/, 2015.

[27] Vitello, S., Lesaege, C., and Piqueras, E. ERC 1497: Evidence standard. https://github.com/ethereum/EIPs/issues/1497, 2018.

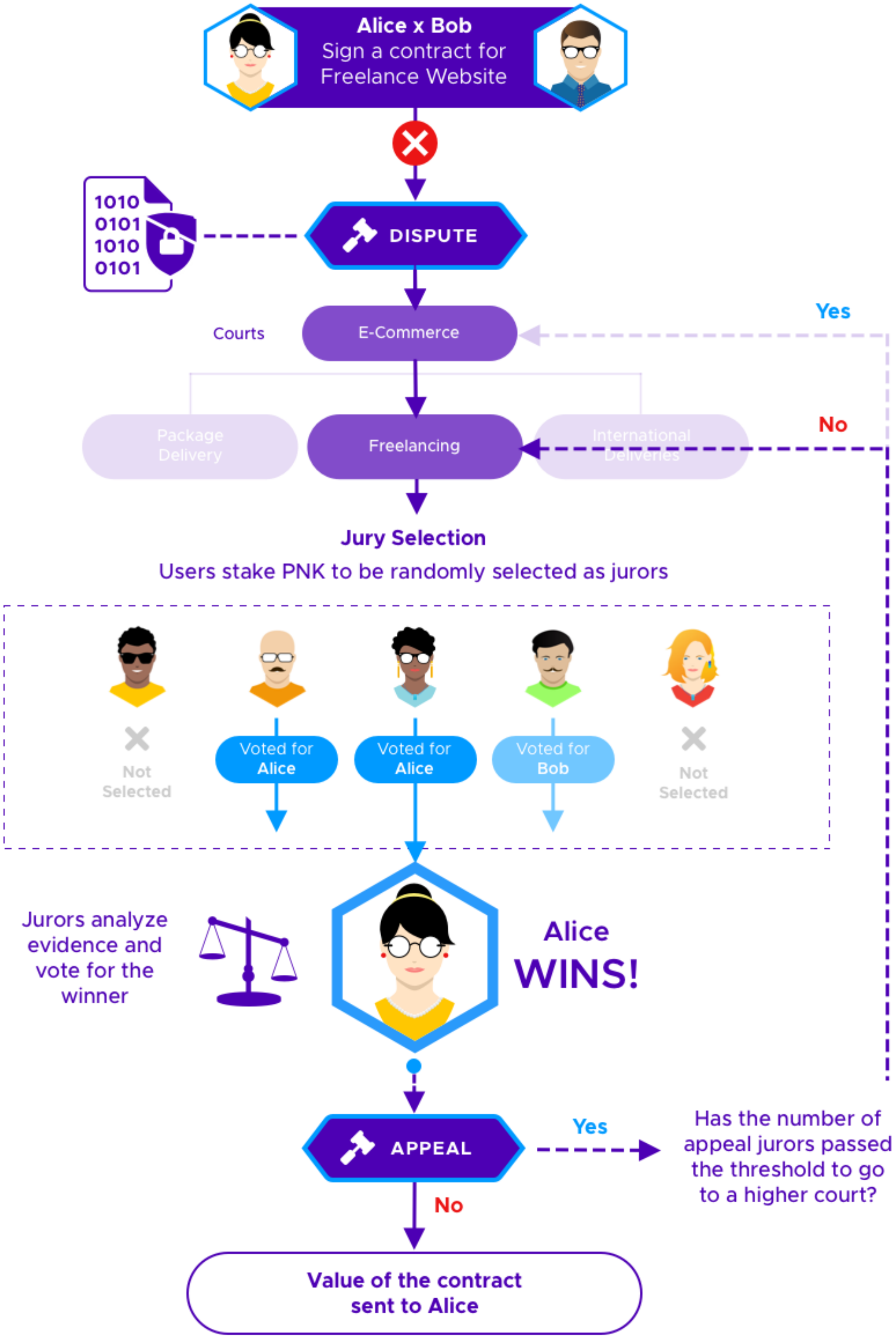


Figure 7: Example of dispute summing up how Kleros works.