Kleros

Short Paper v0.4

Clément Lesaege and Federico Ast

June 2017

Abstract

Kleros 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. It relies on game theoretic incentives to have jurors rule cases correctly. The result is a dispute resolution system that renders ultimate judgements in a fast, inexpensive, transparent 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 disputes. 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 fast and inexpensive way.

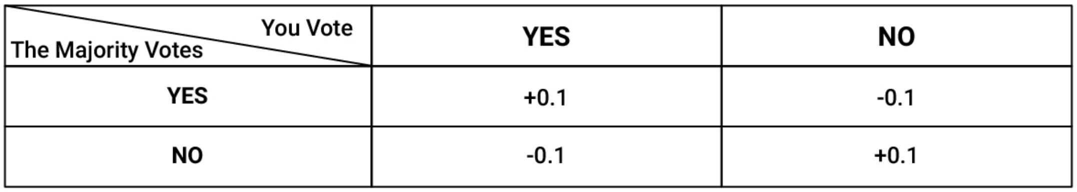
2 Previous Work: SchellingCoin mechanism

Game theorist Thomas Schelling developed the concept of Schelling Point of 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 them?”. 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 (19). Based on the concept of Schelling Points, Ethereum founder Vitalik Buterin has proposed the creation of the Schelling Coin (12), 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 Schelling Coin: “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”. The Schelling Coin 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.

Schelling Coin mechanisms have been used for decentralized oracles and prediction markets (27) (23) (2). 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 Schelling Coin, 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

Giselle is an entrepreneur based on France. She contracts Miguel, a programmer from Guatemala, at a p2p freelancing platform to build a new website for her company. After they agree on a price, terms and conditions, Miguel gets to work. A couple of weeks later, he delivers the product. But Giselle is not satisfied. She argues that the quality of Miguel’s work is considerably lower than expected. Miguel replies that he did exactly what was in the agreement. Giselle 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 Kleros court? Kleros is a decentralized application built on Ethereum. After Miguel stops answering her email, Giselle 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. On the “dead time” on the bus commuting to his job, he is checking Kleros website 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 activates 2 pinakion, the token used by Kleros to select jurors for disputes. The more tokens he activates, 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 activated their pinakion for the dispute. 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 Giselle and Miguel. On the bus back home, Chief analyzes the evidence and votes who is right.

Two days later, after the three juries have voted, Giselle and Miguel receive an email: “The jury has ruled for Giselle. The website was not delivered in accordance to the terms and conditions agreed by the parties. A smart contract has transferred the money to Giselle”. 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 be designed to be arbitrated by Kleros. When they opt-in, contracts creators choose 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. Kleros team is developing a number of standard contracts using Kleros as a dispute resolution mechanism.

4.1.1 Options for jurors

Contracts will specify the options available for jurors to vote. In the introductory example, options may be: “Reimburse Giselle”, “Give Miguel one extra week to finish the website” and “Pay Miguel”.

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

- “Reimburse Giselle” transfers funds to Giselle address.
- “Give Miguel one extra week to finish the website” blocks new disputes for one week and removes this option from further dispute.
- “Pay Miguel” transfers funds to Miguel address.

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

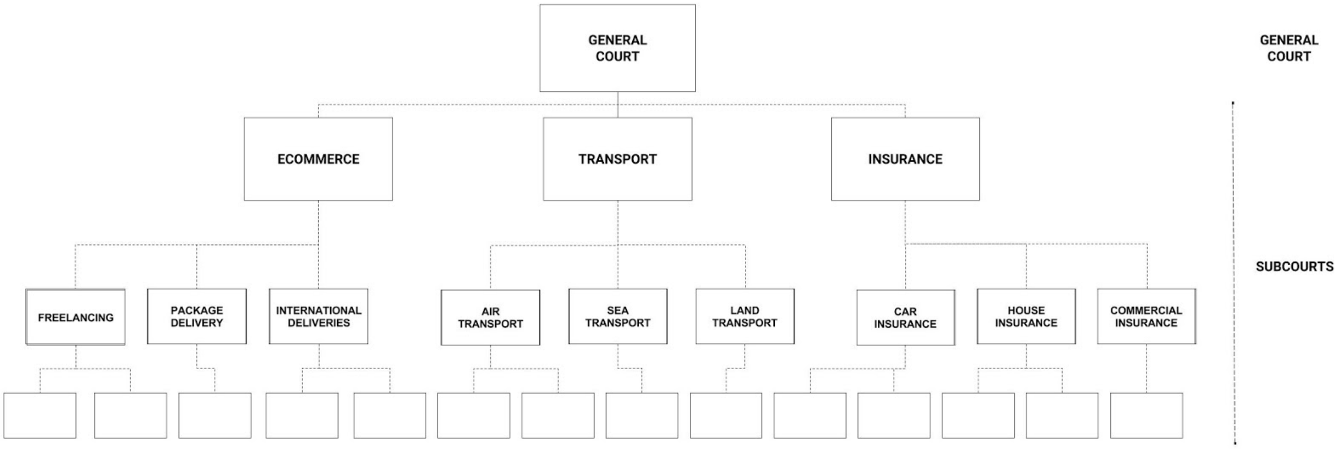


Figure 2: Example of court arborescence from which smart contract creators must select a court.

Options must be in one dimension and in order. In the previous example a juror whose favorite option is “Pay Miguel”, also prefers “Give Miguel one extra week to finish the website” over “Reimburse Giselle”.

4.1.2 Privacy

Solving disputes may require parties to disclose privileged information with jurors. In order to prevent outside observers from accessing this information, the natural language contracts (English or other) and the labels of the jurors voting options are not put on the blockchain. When the contract is created, the creator submits hash(contract_text, option_list, salt) [2] (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 sends {contract_text, option_list, salt} to each party using asymmetric encryption. This way, parties can verify that the submitted hash corresponds to what was send 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 are handled by the application users will run while using Kleros.

4.2 Drawing Jurors

4.2.1 System token: the pinakion

Users have an economic interest in serving as jurors in Kleros: collecting the arbitration fee that every juror gets for his work. Candidates will self-select to serve as jurors using a reputation token called pinakion (PNK)[3].

The pinakion represents a right to be drawn as a juror for a dispute. The probability of being drawn as a juror for a specific dispute is proportional to the amount of tokens a juror activates. The higher the amount of tokens it activates, the higher the probability that it will be drawn as juror.

[2] Note that thorough this paper we use hash referring to a cryptographic hash function, in Ethereum the one used is keccak256.

[3] 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. Most pinakion will be distributed in a token distribution event; a lower part will be given to project contributors and to early stage supporters.

Jurors that do not activate their pinakion do not have the chance of being drawn. This prevents inactive jurors from being selected.

Pinakion play two key functions in Kleros design.

First, they protect the system against the sybil attack (17). 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, pinakion provides jurors the incentive to vote honestly[4].

4.2.2 Jury selection

After candidates have self-selected specific courts and activated their tokens, the final selection of jurors is done randomly. The probability to be drawn as a juror is proportional to the amount of activated 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 reputation shift.

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

Token Owner	Activated	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

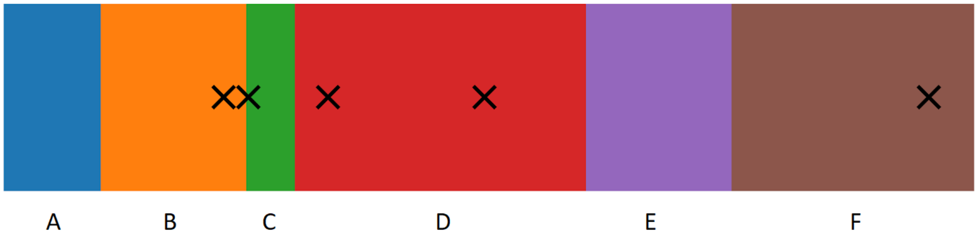


Figure 3: Example of tokens activated and drawn jury members.

For a dispute that requires 5 votes, 5 tokens are drawn out of the 10,000 that were activated. 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. Activated pinakions are frozen during the court session and will be unfrozen after the court has reached a decision.

[4] See the section Incentive System.

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 (8) 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 pinakion are redistributed (see the section Incentive System).

Random numbers are generated with sequential proof of work (14) using a scheme similar to Bünz et al. (15).

1. **Initialization**: We start with seed=blockhash and let any party 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 not to be chosen by 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.

   Note that if Proof-of-Stake the blockhash remains impossible to exactly predict, we can remove this step and only use the blockhash as a seed.

2. **Computing the master random value**: Every party who has a stake in the random number run 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 got the knowledge of the seed and that he got 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 which means that no party will be able to get the results order of magnitude 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 (24). It consists of a dichotomic search on the results of the attacker. If an attacker submits a false h_d, a 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 $h_{x+1} \neq$ hash(h_x) in the attacker answer which invalidate his answer. Parties having their answer 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 value such that r_n = 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 our application, it is not a problem waiting a few hours between the moment the dispute is created and the moment jurors are drawn. More details on this process will be available in future work.

4.3 Votes

After assessing the evidence, jurors commit (9) their vote to one of the options. They submit hash(option, salt). When the vote is over, they reveal {vote, salt}, and Kleros 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 jury members or to the parties. This prevents the vote of a juror to influence the vote of other jurors.

Jurors can still declare that they voted in a certain way, but they cannot provide other jurors a reason to think that what they say is true. This is an important feature of the vote for the Schelling Point to arise. If jurors knew the votes of others jurors, they could vote like them instead of voting for the Schelling Point.

We let any party able to show the commitment of a juror to Kleros before the vote is closed steal the pinakions of this juror and invalidate the vote of this juror.

If a juror wants to reveal its vote to another party, it has two options:

(a) 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.

(b) 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 pinakions of this juror.

This scheme prevent jurors from revealing their votes trustlessly.

Jurors are also required to provide a justification for their vote.

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 winning option is the median one which gives a consensual result and is robust to strategic voting. Assume that a contract with a client and a freelancer specify that if the job is not done entirely, the freelancer still get a portion of the payment (jurors have the possibility to award 0%, 20%, 50% or 100% of the payment to the freelancer).

If, as in Figure 4, 0% gets 2 vote, 20% 1 votes, 50% 2 votes and 100% 4 votes, the winning option would be 50%.

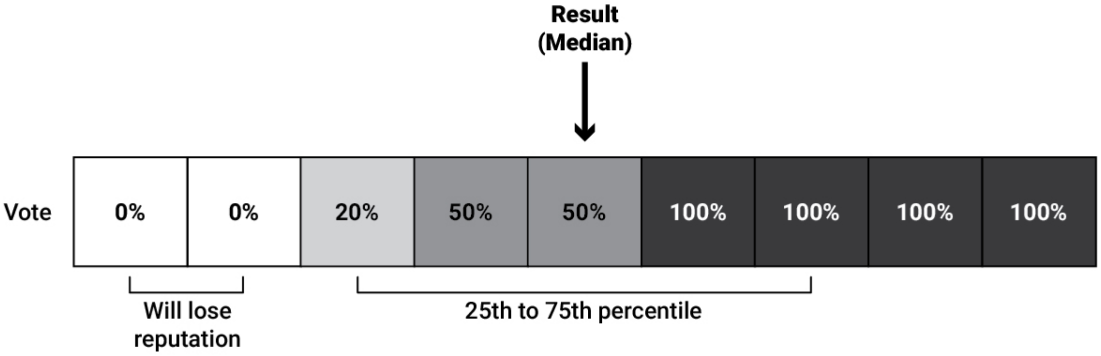


Figure 4: Example of votes, results and reputation shift.

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 will be paid a fee determined by the subcourt where the dispute is solved. The arbitrated smart contract will determine which party will pay the arbitration fee.

For most smart contracts we expect the following:

- 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, the appealing party will pay the arbitration fee.

4.5 Appeals

If, after the jury has reached a verdict, 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 arbitration fees being paid to each juror and appealing increasing the number of juror exponentially, arbitration fees rises 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 Bribe Resistance section).

4.6 Incentive system

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 not coherent with the group will lose some tokens which will be 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[5], where jurors gain or lose tokens depending on whether their vote was consistent with the others jurors.

We will assume a jury member voted coherently if his result is inside the 25th and 75th percentile. These boundaries make sure that if at least half of the jurors voted for an option, they will be considered to have voted coherently. Jurors who vote for an option outside will be penalized. If there is only two options, this means that if one option wins with more than 75% of the vote, those who voted for the other option are considered incoherent and lose tokens to those who voted for the winning option. Otherwise we consider that difference of vote is not significant enough to justify a token redistribution.

The amount of tokens lost per incoherent juror is : $\alpha \cdot \text{min\_activate} \cdot \text{weight}$. 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 that will be defined by the governance mechanism as a consequence of the internal dynamics of the voting environment.

The min_activate parameter is the minimum amount of token which can be activated in the subcourt.

The tokens are divided between the coherent parties proportionally to their weight. You can see an example of token redistribution in Figure 5. Jurors could failed to reveal their vote, to disincentivize

[5] See section Previous Work: SchellingCoin mechanism

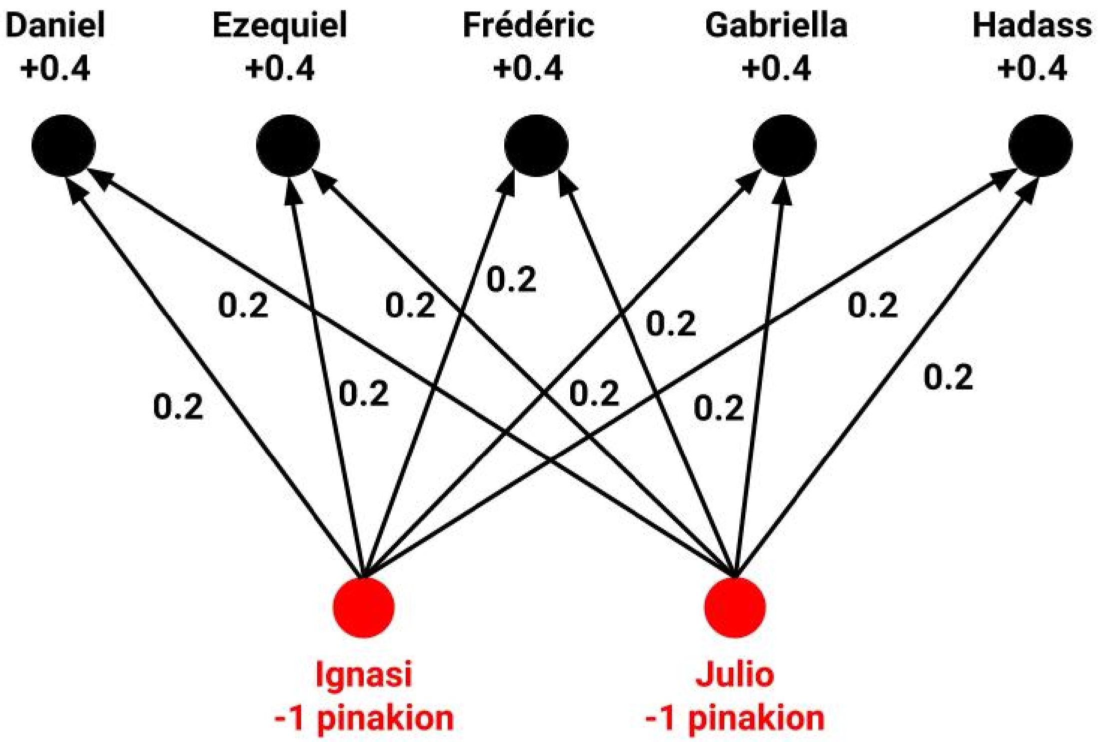


Figure 5: Token redistribution after the vote with seven jurors. Tokens are redistributed from jurors who voted incoherently to jurors who voted coherently.

this behaviour, the penalty for not revealing one's vote is the double than those for incoherent votes ($2 \cdot \alpha \cdot \text{min\_activate} \cdot \text{weight}$). This makes sure that it is always better to reveal one's vote.

In case of appeal, the tokens are redistributed at each level according to the result of the appeal. If at one level no one voted coherently, the tokens are given to the winning party. Note that when there is only 1 to 3 jurors, those jurors can't be penalized directly, but they can be in case of appeal, so they still have incentives to vote correctly.

Note the token redistribution mechanism is still under active research and we may end up with a more sophisticated protocol.

When there is no attack, parties are incentivized in voting 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 Schelling Coin mechanism for a prediction market), no Schelling Point would arise. However, in (25), the informal experiments run by Thomas Schelling show 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

4.7 Attack resistance

4.7.1 Buying half of the tokens

If a party 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 really 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 more. Tokens, contrary to most physical assets, have increasing marginal costs. They will be dynamically priced on exchanges, would 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

The appeal mechanism is 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 pinakions 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 reputation 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, making it an attack requiring a high budget, but zero cost if successful. However there exist a game theoretic response against this attack where jurors uses a mixed strategy (jurors only accept the bribe with a defined probability which increases their expected reward compared to accepting the bribe). More details about this attack and the response can be found in (13).

4.8 Court Arborescence

When activating their pinakions, 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 activated. Each token holder can activate tokens in at most one subcourt of each court they have token activated. Figure 6 show an example of legal activations in.

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.

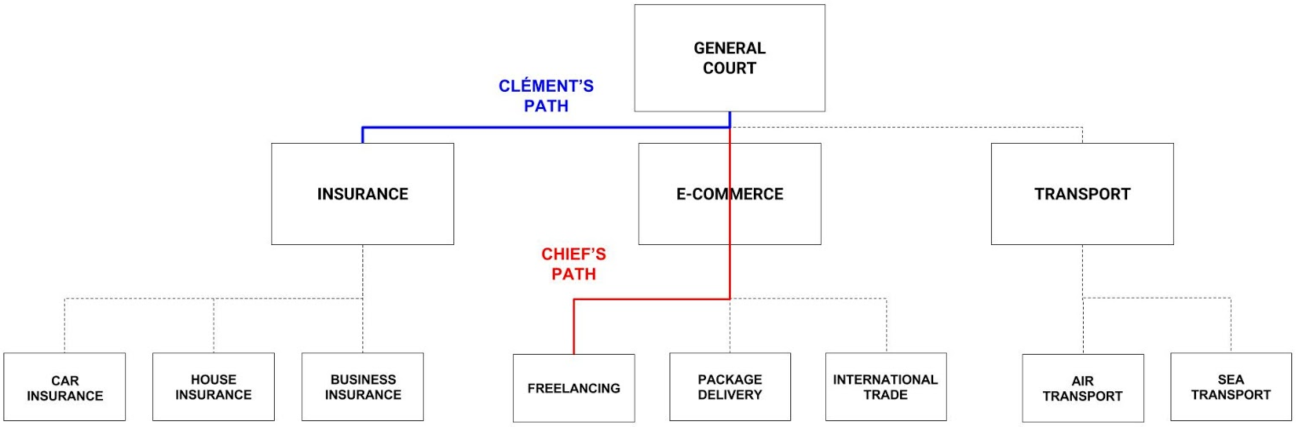


Figure 6: Example of paths chosen by jurors in the subcourt system. Clément can be drawn as juror in the General Court and in the Insurance Subcourt. Chief can be drawn as juror in the General Court, in the E-Commerce Subcourt and in the Freelancing Subcourt.

4.9 Governance Mechanism

As 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 democracy mechanism (18). Token holders will have an amount of votes equal to the amount of pinakions they hold. They will have the option of voting directly or delegating their vote. When a user fails to vote, his voting power is automatically transferred to his delegate. You can see an illustration of the liquid voting mechanism in Figure 7. Vote delegation can also be subcourt specific. Users could choose to delegate their vote in some subcourts but not in others.

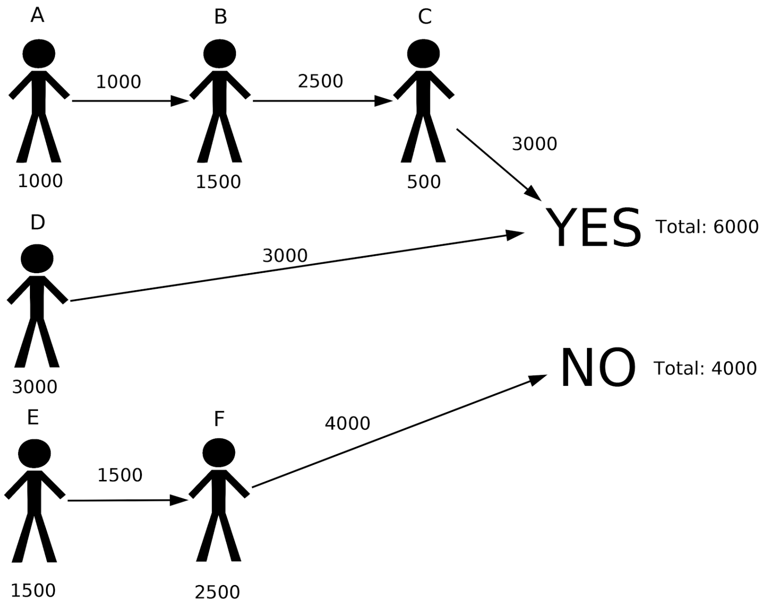


Figure 7: Illustration of a liquid vote

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. Modify the subcourts:
   (a) Add subcourts.
   (b) Remove subcourts.
   (c) Modify subcourt hierarchy.
3. Modify parameters in subcourts:
   (a) Arbitration fees.
   (b) Time of each court session.
   (c) Minimum amount of tokens to be activated.
4. Change one of the smart contracts Kleros rely on. This allows arbitrary changes. This can be used for improvements or in emergency if it appears that some elements of Kleros are not working properly[6].

5 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 8.

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. leroterion 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. The Bitcoin protocol is helping millions in financial inclusion. Kleros will do the same in access to justice by enabling arbitration in a high 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] Bitrared. https://www.bitrated.com/faq/.

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

[3] Oraclize. http://www.oraclize.it/.

[4] Swarm, serverless hosting incentivised peer-to-peer storage and content distribution. http://swarm-gateways.net/.

[6] Note that audits and reviews will be made before the code is deployed. But we can never guarantee at 100% that there isn’t a bug (either on the code or incentives) somewhere. Having this failsafe provides extra security.

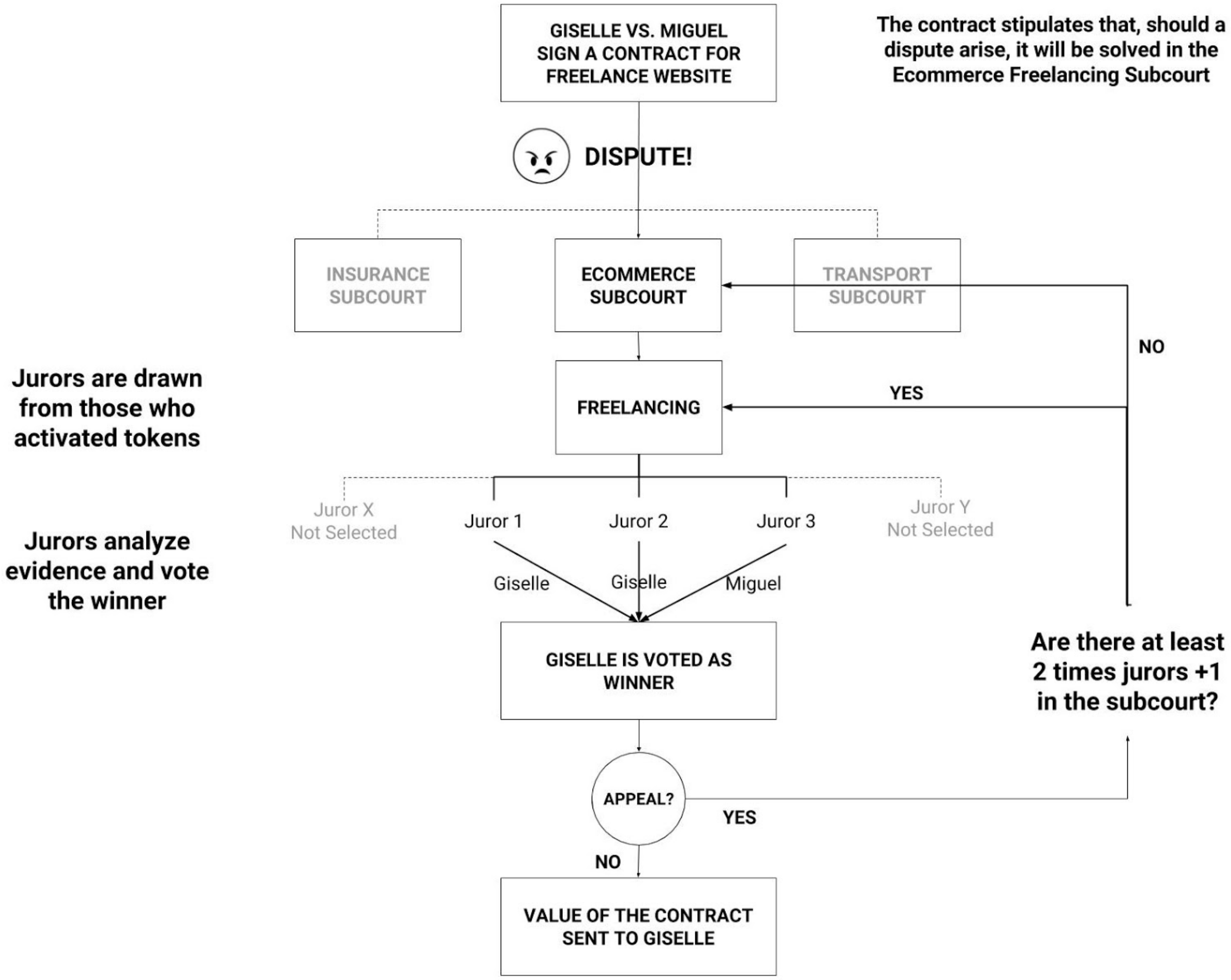

GISELLE VS. MIGUEL SIGN A CONTRACT FOR FREELANCE WEBSITE
The contract stipulates that, should a dispute arise, it will be solved in the Ecommerce Freelancing Subcourt
DISPUTE!
INSURANCE SUBCOURT
ECOMMERCE SUBCOURT
TRANSPORT SUBCOURT
NO
Jurors are drawn from those who activated tokens
FREELANCING
YES
Juror X Not Selected
Juror 1
Juror 2
Juror 3
Juror Y Not Selected
Jurors analyze evidence and vote the winner
Giselle
Giselle
Miguel
GISELLE IS VOTED AS WINNER
Are there at least 2 times jurors +1 in the subcourt?
APPEAL?
YES
NO
VALUE OF THE CONTRACT SENT TO GISELLE

Figure 8: Example of dispute summing up how Kleros work.

[5] Standards for efficientcryptography 2 (sec 2). `http://www.secg.org/sec2-v2.pdf`, 2010.

[6] Bell, J. Assassination politics. `http://jrbooksonline.com/PDF_Books/AP.pdf`, 1997.

[7] Black, D. On the rationale of group decision-making. *Journal of Political Economy 56* (1948).

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

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

[10] Buterin, V. Decentralized court, post on ethereum reddit. `https://www.reddit.com/r/ethereum/comments/4gigyd/decentralized_court/`.

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

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

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

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

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

[16] Clarkson, W. Time-lock cryptography, sending messages to the future. http://www.cs.tufts.edu/comp/116/archive/fall2013/wclarkson.pdf, 2013.

[17] 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.

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

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

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

[21] Martic, D. Blind arbitration. *Sintelnet WG5 Workshop on Crowd Intelligence: Foundations, Methods and Practices.* (2014).

[22] Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf/, 2008.

[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] Szabo, N. Normalizing and securing relationships on public networks. *First Monday* (1997).

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

[28] Teutsch, J., and Reitwiessner, C. A scalable verification solution for blockchains. http://people.cs.uchicago.edu/~teutsch/papers/truebit.pdf, 2017.

[29] van den Herik, J., and Dimov, D. Can the ebay's community review forum fairly resolve disputes? *Proceedings of the 23rd Benelux Conference on Artificial Intelligence 4* (2011).

[30] van den Herik, J., and Dimov, D. Towards crowdsourced online dispute resolution. *Journal of International Commercial Law and Technology 7* (2012).