Kleros

Long Paper v1.0.0

Clément Lesaege, William George, and Federico Ast

March 2020

Abstract

Kleros is a decentralized decision protocol for use on smart contract platforms, which has been implemented on Ethereum. It acts as a decentralized third party capable of providing decisions on the correct result when applying a set of rules to questions ranging from simple to highly complex. This is achieved by using game theoretic incentives to have crowdsourced jurors analyze and rule on cases correctly. Hence, Kleros provides judgments in an inexpensive, reliable, typically fast, and decentralized way. Of particular relevance is the use of this protocol to dispute resolution, creating for a form of decentralized justice.

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 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 resolve every kind of dispute. It is an autonomous organization, implemented on Ethereum, that works as a decentralized third party to arbitrate[1] 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.

[1] Here, a priori, the words "arbitrate" and "arbitration" are used in an informal sense. The status of Kleros decisions with respect to existing legal systems is a subject of research.

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) [30]. 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 Focal Points, which are also called Schelling Points, [40] as a solution that people tend to use to coordinate their behaviour in the absence of communication[2], 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.

Based on the concept of Schelling Points, Ethereum founder Vitalik Buterin has proposed the creation of the SchellingCoin [11], 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 [40] 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.

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

Figure 1: Payoff table for a basic Schelling game

[2]Note that for applications of Schelling points in blockchain systems it is often impossible to guarantee that agents will not communicate as they tend to be pseudo-anonymous. While this is the case of Kleros, we will see that Kleros has an appeal system that incentivizes participants to agree with how potentially not-yet-determined agents in some future appeal round would decide, recovering a partial impossibility of communication. Furthermore, we are undertaking research on how to incentivize participants to not trust any communication they might have between each other, building off of work in [23] that argues that Schelling points also arise in such situations, see Section 5.3.

SchellingCoin mechanisms have been used for decentralized oracles and prediction markets [43] [35] [3]. We note academic work, developed concurrently to Kleros, that also attempts to apply these principles to dispute resolution[3] [17]. The fundamental insight is that voting coherently with others is a desirable behaviour 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 behaviour.

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 Alice and Bob had used Kleros Escrow and put a clause in the contract stating that, should a dispute arise, it would be solved by Kleros Court? After Bob stops answering her email, Alice taps a button that says "Send to Kleros" and fills in 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 the Kleros Court website (`https://court.kleros.io`) to find some dispute resolution 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 court. 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. Assuming Chief rules well in the dispute, he gets his stake of 2000 PNK back afterwards in addition to a payment of arbitration fees, see Section 4.7.

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 court. 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.

[3] Compared to Kleros, the proposal of [17] uses a somewhat different structure for how arbitrators stake to be selected for, vote on, and then receive fees for disputes, with the Schelling game occurring in an additional "validation" phase. Other notable differences are that, while the proposal of [17] does not include a mechanism similar to the court tree we describe in Section 4.2 for global 51% attack resistance, it does include a novel proposal for a "forum" in which community members are incentivized to participate beyond the direct providing of votes in disputes, for example by submitting proposals for changes in governance or new template contracts.

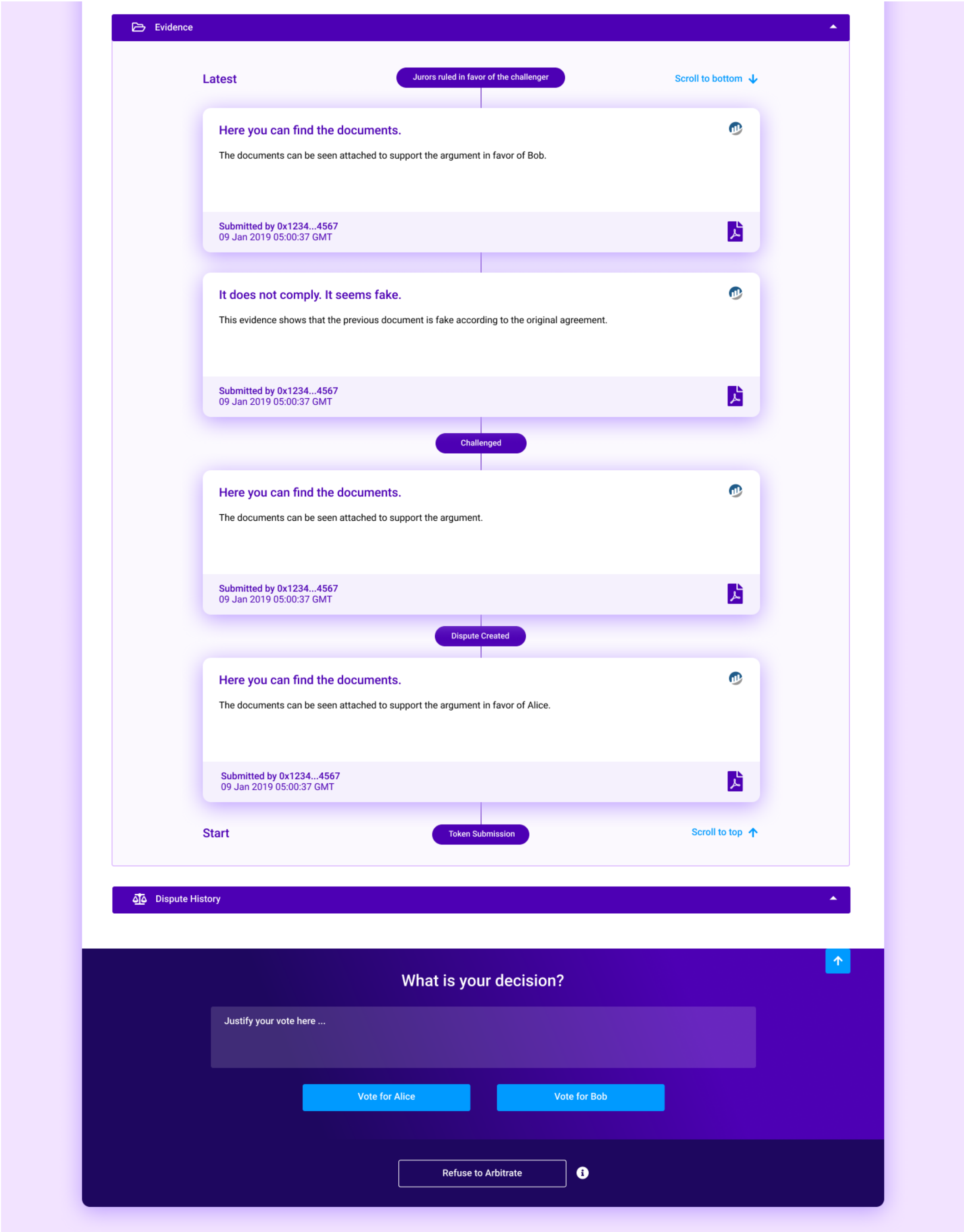

Figure 2: The evidence before a juror as she makes her vote.

4 Kleros Mechanism Design

In this section, we detail the architecture of Kleros[4].

4.1 Arbitrable and Arbitrator Contracts

Kleros is an opt-in court system. "Arbitrated" or "arbitrable" 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, see Section 4.2.

The Kleros team has developed a number of standard contracts using Kleros as a dispute resolution mechanism. Moreover, we have proposed standards [31] [45] that would allow others to develop other contracts in a way that does not require anticipating which dispute resolution mechanism will ultimately be used. This standardization allows parties that regularly require dispute resolution services to easily switching between dispute resolution providers.

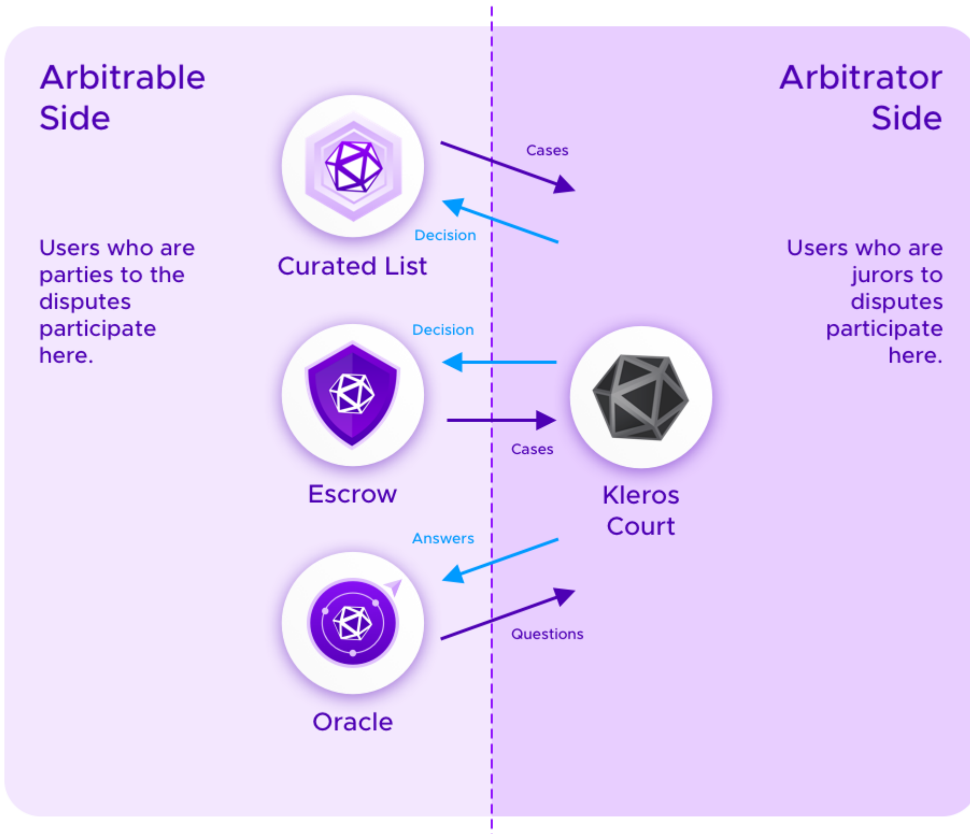

Figure 3: "Arbitrable" contracts, namely contracts that might require dispute resolution, are represented on the left. These include applications such as curated lists, escrows, and oracles, see Section 6. "Arbitrable" contracts designate an "arbitrator" contract such as that of the Kleros Court, represented on the right, to provide rulings on any eventual disputes.

Contracts can allow for a variety of possible behaviour in cases that do not require arbitration,

[4]Throughout this work we will indicate both the current state of Kleros as it is implemented at the time of writing as well as proposed mechanisms that we have researched for future versions. As we continue to research these issues, the proposed future mechanisms described below are subject to change.

particularly when there is unanimous consent of parties. For example, the current Kleros Escrow includes a limited settlement system.

Moreover, 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 behaviour of the contract after the ruling is done for each possible option. In the example:

- "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.

In general, any finite list of options can be proposed to jurors[5]. Sets of options having a different structure can be accommodated in some circumstances to the degree that they can be "discretized". For example, we have researched mechanisms by which jurors can choose a real-number value [25], as one might want for cases where one party receives a percentage refund. For examples, jurors might rule to refund 75% of an amount at stake to Alice while paying 25% to Bob.

4.2 Court Tree

When creating an arbitrable contract, parties should 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.

In parallel, when registering as jurors, users start in the General Court and follow a path to a specific court according to their skills[6]. Each token holder can register a given token in at most one child court of each court in which they have token staked[7]. Figure 4 shows the current court structure with examples of allowable registrations.

[5]One option that is always presented to jurors is to "refuse to arbitrate". For the sake of incentivization of jurors, see Section 4.7, this option is treated like any other option and hence jurors have an economic incentive to vote for it if and only if they believe that it will be the winning option. A court policy of the Kleros general court, and hence all Kleros courts, see Section 4.2, is that jurors should vote "refuse to arbitrate" if the decision is being used as part of illegal activities.

[6]In the language of graph theory, the structure of the set of courts forms an arborescence with the General Court as the root.

[7]Moreover, token holder *must* be staked in the parent court to stake in the child court, so when staking in a given court a token holder is staked in all the courts on the path from it to the General Court.

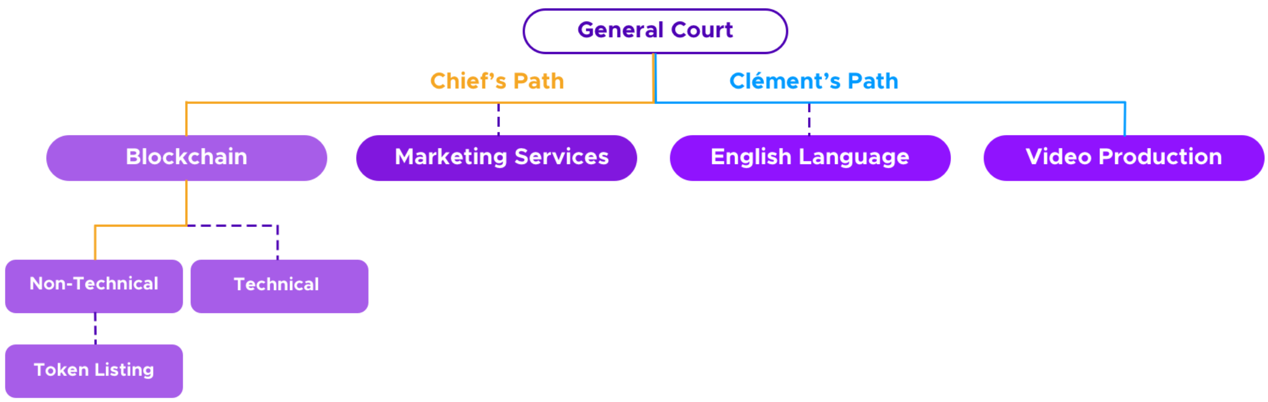

Figure 4: The current court tree from which smart contract creators must select a court. New courts will be added as Kleros is adopted to resolve additional types of disputes. The mechanism to add new courts is described in Section 4.11. Example of paths chosen by jurors in the court system are shown. Clément can be drawn as juror in the General Court and in the Video Production Court. Chief can be drawn as juror in the General Court, in the Blockchain Court, and in the Blockchain Non-Technical Court.

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

Each court has some specific features regarding policies. Also a number of parameters are chosen on a court by court basis including session time, cost, number of drawn jury members and tokens staked. We will consider these system parameters in detail below, see Section 4.7.

4.3 Raising a Dispute

The nature of an arbitrable contract determines its default behaviour and conditions under which a dispute can be raised[8] Typically, parties to an agreement can raise a dispute if there is a disagreement, or accept a default behaviour if there is agreement.

In the event of a dispute, parties can provide evidence and arguments on behalf of their case during an evidence period. This evidence conforms to the ERC 1497 standard [45], which sets forth requirements for how this evidence is organized and how it triggers smart contract events, providing for interoperability across arbitrator applications.

4.4 Drawing Jurors

4.4.1 The Pinakion Token (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)[9].

[8] Such as time limits, or fees required to Kleros jurors must be paid, see Section 4.5.

[9] 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.

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 she stakes, the higher the probability that she will be drawn as juror. Thus, staking PNK signals availability to be drawn as a juror; users that do not stake PNK do not have the chance of being drawn. This prevents inactive jurors from being selected.

PNK plays three key functions in the design of Kleros.

- First, it protects the system against the Sybil attack [20]. If jurors were simply drawn randomly, a malicious party could create many addresses to be drawn a high number of times in each case. 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 by making incoherent jurors, i.e. jurors whose votes do not agree with the ultimate ruling, pay part of their stake to coherent ones, see Section 4.7.
- Finally, PNK can be "forked" in a way that creates parallel versions of Kleros, serving as a fallback defense in the event of successful 51% attacks, see Section 4.9.

4.4.2 Random Selection

Candidate jurors self-selected into specific courts. By staking her tokens in a given court, a participant expresses an availability to be draw as a juror in that court. Then, the final selection of jurors is done randomly[10]. 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 she will get in the dispute and the amount of tokens she will win or lose during the token redistribution.

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

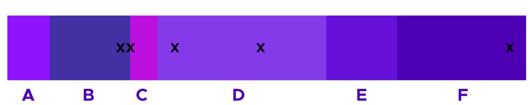

Figure 5: Imagine that 6 token owners staked 10,000 in total with the above distribution. Then 5 tokens are drawn: numbers 2519, 4953, 2264, 3342 and 9531. As a result, token owners B, C and F are drawn with a weight of 1. Token owner D is drawn with a weight of 2.

[10]Note that random draw has a long history of use in public decision making. In addition to its use in jury selection in many countries today, it was, for example, used in the selection of public office-holders in ancient Athens and Renaissance Venice [18]. For reflections on the legitimacy of such processes, see [21] [42]

See Figure 5 for an example of juror selection. Note that staked PNK (except those paid by incoherent jurors) can be taken back after the court reaches a final decision.

4.4.3 Random Number Generation

In order to draw jurors, a random number generation process that is resistant to manipulation is required. Using a protocol for creating a random number between two parties [6] does not work. An attacker could create disputes between herself, select herself as a juror multiple times and select another victim juror. She 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 discussion of the incentive system in Section 4.7.

4.4.3.1 Current Random Number Generator

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.

4.4.3.2 Improved Future Random Number Generation

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 using a scheme similar to Bünz et al. [16] [11].

1. **Initialization**: One starts 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. It is important that the seed is not chosen by any one party. Using the above process, 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 to 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 she gets the result. The difficulty d is fixed such that no hardware can compute h_d during the time of the initialization phase. Because one needs the result of the previous step before starting the next one, this process cannot 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 [36]. 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 her for her $h_{d/2}$ value. If she gives the wrong value, there is an error in the attacker values between h_0 and $h_{d/2}$. If she 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

[11] Note that the following protocol that we present is adapted to also work for Proof-Of-Stake blockchains. In Proof-Of-Work blockchains, as the blockhash remains impossible to exactly predict, one can remove the initialization step and only use the blockhash as a seed. However, Ethereum has planed to switch to Proof-Of-Stake.

honest party can exhibit x such that $h_{x+1} \neq \text{hash}(h_x)$ in the attacker answer which invalidates her 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 one derives 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 courts with a particularly low session time (for example, a court 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 courts is one based on threshold signatures [7].

4.5 Arbitration Fees

Kleros uses arbitration fees in order to compensate jurors for their work. These fees also make it more difficult for an attacker to spam the system by creating frivolous disputes and/or appealing as these actions require paying arbitration fees. Each juror who is coherent with the final ruling will be paid a fee determined by the court where the dispute is solved. The arbitrable smart contract will determine which party will pay the juror fee; this can vary from one application to another.

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

- In first instance, each party will deposit an amount equal to the juror 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 juror 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 arbitration 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[12].

4.6 Voting

4.6.1 Voting Process

Jurors assess evidence that has been submitted, typically by the parties to the dispute, and are provided with court policies, comparable to juror instructions[13], on how they should reason based on that evi-

[12]This requires a form of litigation funding to protect parties that do not have sufficient financial resources to deposit appeal and stake deposits. In such a framework, a funder would pay the deposit of a party in exchange for part of the stake if the dispute is won. All of this can be smart contract enforced.

[13]These policies vary by court, see Section 4.2 and can be changed by the governance procedure, see Section 4.11.

dence. Then jurors commit [9] their vote to one of the options. They submit $\text{hash}(\text{vote}, \text{salt}, \text{address})$[14]. The salt is a random value generated locally in order to add entropy to prevent the use of rainbow tables[15]. 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 Section 4.7.

After a juror has made a commitment, her vote cannot be changed. However, 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[16].

As this two step processes of committing and then revealing one's vote requires additional user interactions, in some low stakes courts, one might want votes to be issued publicly to simplify the user experience[17]. Which system is used is determined via a court parameter, see Section 4.11 on governance.

4.6.2 Vote Aggregation

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 according to a predetermined voting rule resulting in an option that is considered the winner.

4.6.2.1 Current Voting System

When jurors are presented with a binary choice, it is natural to use the Plurality, or "first-past-the-post", voting system[18]. Currently, Kleros employs the Plurality system even when there are more than two choices. (Specifically, we adopt the choice that wins a plurality of the vote in the last appeal, see Section 4.8).

When there are more than two options, under a Plurality voting system, the following may occur:

- If there are many very similar honest options (or "clones"), they will divide the votes of jurors that are attempting to vote honestly, decreasing the probability that any one of them wins. Anticipating this effect, jurors might instead vote for a distinguished but dishonest choice. For example, imagine that in our contractor use-case above there were several different options that gave Bob another week, another eight days, or another nine days respectively. Then the collective

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

[15] As currently implemented in the existing Kleros interface, the salt is generated by having the user use the Ethereum private key stored in Metamask to sign a block of text that includes an identifier for the specific dispute. Hence, the signatures on this text with different keys will be different and are computationally infeasible to obtain without the user's private key under the assumption of the security of the signature algorithm. Thus, a juror is prevented from copying the commitment of others. Nevertheless, if a user deletes her salt from her local memory, she can regenerate it as long as she retains access to her private key.

[16] See the discussion on future work in Section 5.3 for a further discussion of these ideas.

[17] 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 used for token redistribution, see Section 4.7. This limits the effectiveness of a vote copying strategy. Hence public votes might be acceptable in some cases.

[18] Plurality, or "first-past-the-post", is the voting system that in which voters express a vote for only one candidate, and then the candidate that receives the largest number of votes is selected, even if this candidate does not receive a majority of the total votes due to there being more than two candidates.

odds of any of these options being chosen would likely fall below the odds of a single "give Bob more time" option, see Figure 6.

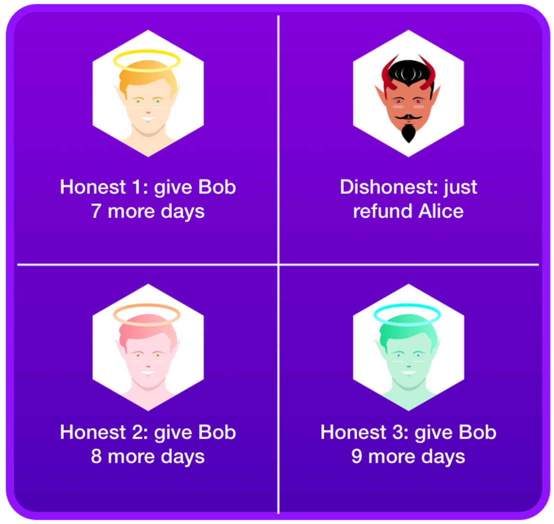

Figure 6: In the Plurality voting system, jurors can only vote for one option. Particularly, a voter cannot cast a vote such as "take one of the more time options, it doesn't matter which". Then if jurors are presented with a collection of similar, honest choices along with a single dishonest choice, the dishonest choice may seem distinguished and become the Schelling point.

- To the degree that no single option is likely to receive more than 50% of the votes, this lowers the bar for the number of votes that attackers need to corrupt to pass a dishonest result, see Figure 7.

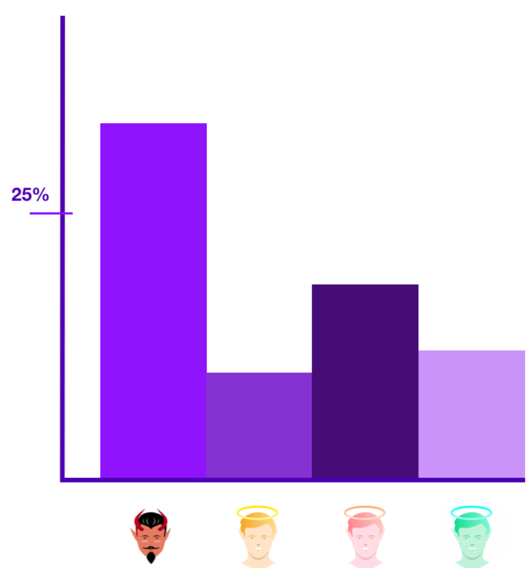

Figure 7: If the vote is split between several "honest" options, the attacker does not need to corrupt 50% of the vote in a Plurality system to have a malicious option adopted.

Considering these issues, one might expect Plurality voting to still produce generally "honest" results when one single choice has a very clear, winning case, which essentially binarizes the choice. However, Kleros should be able to cope with nuanced cases involving many choices.

4.6.2.2 Social Choice Theory and Future Voting System(s)

In this section, we consider a number of desirable properties for a vote aggregation rule in a system such as Kleros, remarking which rules satisfy or do not satisfy these properties. Note that for some of the properties we consider, an analysis of how a given vote aggregation rule performs will depend on the choice of incentive system that we discuss in Section 4.7. Indeed, the choices of voting aggregation rule and incentive system are deeply interwoven and should be considered together. Also note that, while some of the properties we consider have standard definitions, others will be somewhat subjective.

Ideally, the vote aggregation system should have the following properties:

- Clone independent - This is a standard property considered in social choice theory that informally means that having a "clone set" of multiple similar options does not increase or decrease the winning chances for other options outside of the clone set, see for example [44] for a formal definition. Using a clone independent aggregation rule allows parties to produce arbitrable contracts without having to worry that presenting two or more similar options to the jurors might lead to vote splitting in or against their interests. Examples of clone independent voting systems are Instant-Runoff, Ranked Pairs, and Schulze. The results in the clone independence column of the table below are all presented in [44] and [41].

- Easy for jurors to understand - The difficulty of understanding the voting system is particularly relevant for voters in a Kleros-type system, even compared with the situation of voting in democratic elections. Voters in a Kleros-type system need to be able to mentally simulate how a vote is likely to play out in order to maximize their chance of being coherent with its output.

- Satisfy the Condorcet criterion - A voting rule is said to be a Condorcet method if whenever there is an option w such that more voters rank w higher than a for all other options a, then w wins, see [8]. Note, however that such a "Condorcet winner" does not necessarily exist for all sets of votes expressed by jurors. The idea that, if there is an option that wins "head-to-head" against all others, then that option should be selected, is fairly intuitive. Hence, if Kleros cases often have Condorcet winners, then they are particularly straightforward to mentally simulate for jurors. However, Condorcet methods may have more complicated rules for how to handle situations in which there is no Condorcet winner; this can lead to a voting system whose effects are *usually* easy for jurors to understand, but may be complicated for jurors to analyze on cases where there is not likely to be a Condorcet winner. Moreover, the Condorcet criterion, namely to prefer options that have a consensus of the population against each other option, corresponds to certain notions of "fairness". On the other hand, if Kleros cases rarely produce Condorcet winners, this property is less important. Ranked Pairs and Schulze are examples of Condorcet methods.

- Likely to transform "honest" votes into an outcome that is largely seen as honest - In Section 4.7 we will consider an incentive structure designed to encourage Kleros jurors to submit votes that reflect their honest evaluation of the case. Indeed, if jurors are not incentivized to individually provide honest votes, it is unlikely that the resulting vote aggregation will be seen as honest. Then, the question of what vote aggregation rules are effective in taking the "noisy signal" of honesty given by the votes provided by the jurors and converting it into a collective outcome that is seen as honest is reminiscent of work of Conitzer and Sandholm [19]. Their work considers a setting where participants attempt to provide a ranked choice vote expressing their view of a given underlying truth that they see with some noise[19]. Then, one asks which voting systems are maximum likelihood estimators on producing the true outcome from this information. In the table below we indicate the results found in [19], where MLEWIV indicates that the given voting method is a maximum likelihood estimator when the "honest" answer is thought of as being a single outcome, and MLERIV indicates that the given voting method is a maximum likelihood estimator when the "honest" answer is thought of as being a full ranking of the possible outcomes[20]. While there may be some situations where it is natural for jurors to think of the options as being totally ordered, and hence as there being an "honest" order rather than just a single "honest" outcome, the situation of MLEWIV is more typical of this context. Plurality and Borda are maximum likelihood estimators in both models, whereas Instant-Runoff is seen to be a MLERIV but not a MLEWIV.

- Resistant to attacks - Note that attack resistance is essentially economic. For example, one wants the number of votes that would need to be changed to result in a "dishonest" outcome winning to be high, with the rationale that this increases the cost of attacks, such as bribes that would be necessary to change those votes. Hence, attack resistance touches on certain standard voting systems properties that have been studied in social choice theory, such as the "later no help" and the "participation" criteria [8], in combination with analysis of the penalties and rewards laid out in the incentive system, see Section 4.7. While one can exhibit attacks on individual systems, it is challenging to produce proofs that no attacks exist[21] The discussion in Section 4.6.2.1 already shows that the bar for a 51% attack is lowered on the Plurality system in cases where no single (honest) option receives more than 50% of the votes. In the Borda system, if the "honest" answer is a and an attacker wants the option b to win, she can submit many votes where a is ranked first and b is ranked second, and a small number of votes where b is ranked first and a is ranked last. Using the incentive system that will be described in Section 4.7, this attack has limited cost and risk for the attacker whereas for appropriate choices of parameters, Instant-Runoff and Ranked Pairs seem to be more resistant[22].

[19] An important difference between our setting and that of [19] is that [19] does not consider the voters as being economically motivated in whether they provide honest rankings. Hence, the other properties that are considered here, particularly attack resistance, are not relevant in that setting.

[20] Note that the results of [19] that show a voting system can be considered as a maximum likelihood estimator are based on specific, though plausible, models for the noise with which voters observe the honest answer. Their negative results hold for any noise model where the jurors' votes are independent and identically distributed. Further note that Schulze was not considered in [19], however a counterexample to it being a MLEWIV can easily be constructed using Lemma 1 of [19] where one takes sets of votes $V_1 = \{c > a > b * 9, a > c > b * 1, a > b > c * 8, b > c > a * 8\}$, $V_2 = \{b > a > c * 7, c > b > a * 8, a > c > b * 3\}$. This is unsurprising as one sees in Proposition 8.16 of [8] that the MLEWIV property is an alternative characterization of scoring rules. It is less clear whether Schulze can be viewed as a MLERIV.

[21] Hence the question marks on the claims of better attack resistance for Instant-Runoff, Ranked Pairs, and Schulze in the table below.

[22] All of our claims on attack resistance should be viewed as based on using the incentive system of Section 4.7. It is possible with another incentive system our conclusions would be different. Also note the role of β discussed in Section 4.7; for $\beta = 0$ the economic cost in lost deposits of the attack described on Borda is comparable to that of an attack on Instant-Runoff where the attacker bribes many voters to place b first and a second, though the attack on Instant-Runoff requires convincing a large number of jurors to accept a small in-protocol penalty, and the attack on Borda involves convincing a small number of jurors to accept a large penalty. However, with other choices of β, this attack on Instant-Runoff becomes more expensive while the attack on Borda does not. Indeed, heuristically, for Instant-Runoff as well as any Condorcet system, one would expect that in order to produce a dishonest answer, the attacker would need to corrupt many votes willing to place a dishonest answer over the "honest" answer; for appropriate choice of β one would expect the incentive system of Section 4.7 to heavily weight the pair between these two answers, and hence a narrowly failed attack should be expensive.

The following table summarizes how selected voting systems perform according to these criteria:

	CLONE INDEPENDENT	DIFFICULTY TO UNDERSTAND	CONDORCET	TRANSLATION OF VOTES TO OUTCOME	ATTACK RESISTANCE
PLURALITY	No	Easy	No	MILEWIV+MLERIV	Bad
BORDA	No	Easy	No	MLEWIV+MLERIV	Not great
INSTANT-RUNOFF	Yes	Medium	No	MLERIV	Better?
RANKED PAIRS	Yes	Hard	Yes	Neither	Better?
SCHULZE	Yes	Hard	Yes	Not MLEWIV	Better?

When jurors are presented with a binary choice, as is the case of most current applications of Kleros[23], these voting systems are equivalent. Indeed, such voting between two options satisfies all of the "good" properties in this table. Hence these comparisons are only relevant when there are at least three possible outcomes.

Weighing these considerations, it is planned to implement Instant-Runoff voting in future versions[24]. Ultimately, similar to the situation of Arrow's Impossibility Theorem [4] and the Gibbard-Satterthwaite Theorem [26] [39], it is not unlikely that it is the case that it is impossible to have a voting system-incentive system pair that satisfies all desirable properties, and that one is forced to make tradeoffs adapted to what is acceptable in a given use case[25].

Finally, note that while the design choices of Kleros are motivated by the particular challenges of being attack resistant in a setting without trusted authorities, the properties considered above are potentially relevant beyond blockchain applications to other crowdsourced platforms that, like Amazon's Mechanical Turk [1], require an aggregation of user feedback with users who may provide

[23] In fact, all Kleros disputes also have the possible outcome that jurors vote "refuse to arbitrate", see Section 4.1. However, it is expected that this option will rarely be voted for, so a case with two other possible outcomes are de facto binary with only those two outcomes having plausible chances of being voted for.

[24] An important technical point in the choice of a voting system is how to handle ties. For Instant-Runoff, a particularly simple choice of tiebreaker from the perspective of minimizing code complexity is to eliminate any options that are tied for last place in a voting round, and return "refuse to arbitrate" if all options are eliminated. However, as such an implementation deviates slightly from Instant-Runoff as it is typically defined, it may fail some properties in edge cases that normal Instant-Runoff satisfies (e.g. failures of clone independence could occur if all members of a set of clones were tied for last in a given voting round).

[25] Again, further research on this question may result in other choices eventually being made. Notably, if it is determined based on community feedback that the complexity of the Ranked Pairs or Schulze systems does not hinder the juror experience, one of these systems would also be a reasonable choice. Another possibility would be to use a Condorcet variant of Instant-Runoff that, before each step during the tabulation where the lowest remaining candidate is eliminated, determines whether there is a Condorcet winner for the choices expressed among the remaining candidates, and if so returns that choice.

incorrect or spam answers to minimize the effort required of them. Hence some of these ideas could potentially be used to improve the design of such systems in the spirit of [34].

4.7 Incentive System

Users are incentivized to become Kleros jurors as this gives them the opportunity to receive a portion of the arbitration fees paid for the dispute, as discussed in Section 4.5. As part of the incentive system that encourages jurors to provide honest rulings, in addition to the potential to gain arbitration fees, jurors can also lose some of their PNK stake for rulings that are out-of-line with those of the other jurors. These lost PNK stakes are then redistributed to other, more coherent, jurors as will be described below. Thus jurors are participating in a Schelling game similar to those described in Section 2.

In order to be drawn in a given court, users are required to stake a minimum amount of tokens, denoted by min_stake. Then, regardless of how a juror votes on a case, the number of tokens that she can lose from her stake per vote is limited to a fixed percentage of this minimum stake. This percentage will be denoted by α. However, was observed in Section 4.4.2 in the discussion of the "weight", a single juror can be drawn multiple times for a given case, giving her more votes on this case. Then the maximum amount that the juror can lose as a result of her vote increases corresponding to this weight. Namely, the maximum amount of tokens than can be lost on a given case per juror is:

$$D = \alpha \cdot \text{min\_stake} \cdot \text{weight}.$$

Both the α and the min_stake parameters are defined by the governance mechanism and can vary from one court to another.

4.7.1 Current Token Redistribution Model

Currently, in parallel with the first past the post voting system that we described in Section 4.6.2.1 (in which particularly jurors do not provide ranking of the options other than a single vote), any juror that does not select the outcome w that wins the last appeal round loses her deposit D. Then jurors that do vote for w receive a payment of:

$$\frac{\text{ETH fees and lost deposits}}{\#\text{ jurors that vote for } w}.$$

These calculations (i.e. how many deposits were lost and how many jurors voted for w) are done on a round-by-round basis.

4.7.2 Future Token Redistribution Model

Take w to be the option that wins via the vote aggregation methods described in Section 4.6.2. We speak about jurors voting "coherently" if they agree with the ultimate vote outcome; while being coherent is an all or nothing property for binary decisions, for non-binary decisions jurors' votes can be more or less "coherent". The goal is to incentivize users to place outcomes they believe to be "honest" high in their lists following the motivations of Section 4.6.2. Conversely, one also wants to strongly penalize a juror who has placed the winning choice w far down on her list. One option would be to have jurors lose:

$$\frac{\text{\# options ranked above } w}{\text{\# total options} - 1} D \tag{1}$$

to be redistributed between other jurors based on their coherence. However, in this framework, an attacker that ranks a malicious choice first and w second will risk relatively little of her deposit.

In equation 1, one rewards the jurors for the number of options a_i that they correctly place below the winner w with each a_i given the same weight. Alternatively, one can give extra weight for rewards and penalties for options a_i for which the margin of pairwise votes between w and a_i is particularly close. This is in the spirit that *narrowly* failed attacks should be particularly expensive, which is a common goal in the design of blockchain-based platforms [13]. If an attacker is attempting to commit a bribing attack so that a would-be Condorcet winner w^* no longer wins, this requires a sufficient number of bribes so that at least some a_i defeats w^*. Hence at least one pair must pass from the honest winner winning to not, so in narrowly failed attack this pair will be weighted heavily.

Then, taking $\beta \geq 0$, chosen via the governance mechanism[26], we define weights:

$$w(i) = \frac{\left(\frac{1}{|\text{margin of } a_i \text{ against } w| + 1}\right)^{\beta}}{\sum_{a_j \neq w} \left(\frac{1}{|\text{margin of } a_j \text{ against } w| + 1}\right)^{\beta}}.$$

Now we have voters lose:

$$D \sum_{a_j \neq w} \mathbf{1}_{\text{voter voted } a_j > w} \cdot w(j)$$

from their deposit D and receive redistributions of the form:

$$\frac{\text{ETH fees and lost deposits}}{\sum_{\text{voter}_k \in \mathcal{V}} \sum_{a_j \neq w} \mathbf{1}_{\text{voter}_k \text{ voted } a_j < w} \cdot w(j)} \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR} \text{ voted } a_j < w} \cdot w(j),$$

where $\mathcal{V}$ is the set of voters in the same voting round as $\mathcal{USR}$.

Under this payoff mechanism, the lower jurors place the winning outcome, the larger the portion of the deposit they lose and the less arbitration fees they receive. Indeed, if a juror places the ultimate winning outcome last, she receives no arbitration fees and loses her entire deposit, which is split between other jurors in accordance with how high they ranked w[27]. The above formulas do not involve a division by zero unless all voters in a given round place the winning outcome last, in which case all voters lose their deposit and do not receive a payout[28].

The β parameter allows one to tune how much narrowly decided pairs are weighted compared to less narrowly decided pairs. For example, if $\beta = 0$, all weights are the same and we recover the previous incentive system given by penalties of the form of Equation 1; if $\beta \to \infty$ then all the weight is concentrated on the narrowest pair (which is probably undesirable because, even though our comments above argue that a narrowly failed attack will lose some narrow pair, this pair may not be a priori

[26] Note that one might wish to choose different β in a way that depends on the dispute round as in early rounds, the number of jurors is small enough that which a_j are narrowly decided and hence which are weighted heavily is more variable, and hence to some degree arbitrary.

[27] The redistribution mechanism is inspired by the SchellingCoin, see Section 2, where jurors gain or lose tokens depending on whether their vote was consistent with the others jurors. Note that token redistribution mechanisms are still being actively researched and may further evolve in future versions.

[28] If at one level no one voted coherently, what to do with the amounts from that level can be determined by the governance procedure. See the descriptions on future work in Section 5 for further discussion on this point.

the narrowest, so placing all the weight on the narrowest pair may actually make the attack cheaper). The "plus 1"s here are to allow $w(i)$ to be defined even if some pairs are tied.

In the following proposition, one sees that this payoff system can have good properties with respect to incentivizing jurors to rank candidates who are likely to win higher, corresponding to the objectives laid out in Section 4.6.2.

Proposition 1. *Consider the above incentive system with $\beta = 0$. Suppose that a given voter has a probabilistic prior for the outcome of the dispute resolution process, i.e. she estimates probabilities for the votes of other jurors and for the probabilities of each outcome to win possibly after appeals, such that:*

- *she believes that the votes of other jurors in her voting round are independent of her vote,*
- *she believes that the outcome is independent of her vote and the votes of other jurors in her voting round,*
- *she assigns to the possible outcomes $a_1, \ldots, a_n$ probabilities $prob(a_1), \ldots prob(a_n)$ of ultimately winning.*

Then a weakly dominant strategy for this juror is to rank the outcomes a_j from highest to lowest by their chance of winning $prob(a_j)$.

See Appendix A for a proof of this result. Note that the perspective of a juror believing that her vote will not change the ultimate outcome can be justified in our setting if jurors believe that incorrect outcomes are likely to be appealed[29].

After Kleros has reached a decision, tokens are unfrozen and redistributed among jurors. An example of token redistribution is shown in Figure 8. Note that jurors could fail to reveal 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.

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 [40], 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 dispute resolution 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[30].

[29] Indeed, one possibility is to use $\beta = 0$ for redistribution of all rounds other than the last appeal round, and then some non-zero choice of β in the last round. In this case, narrowly failed attacks would still incur a high cost in the last round. On the other hand, if voters are certain that a case will be appealed, they have incentives equivalent to an incentive system without weighting. Then the good incentive compatibility properties we have see in Proposition 1 will also apply to such voters. More generally, when evaluating how to vote, voters will have to judge how likely they are to be in the last round; if rational voters believe they are likely in the last round they could potentially take large deviations from honest votes. However, one might expect such deviations to be tempered by the expectation that large collective deviations are likely to cause a result that is appealed.

[30] Indeed, note that, so far, this idea has largely worked as expected in experiments such as those considered in [24] as well as in practical applications such as those of [28] and [35].

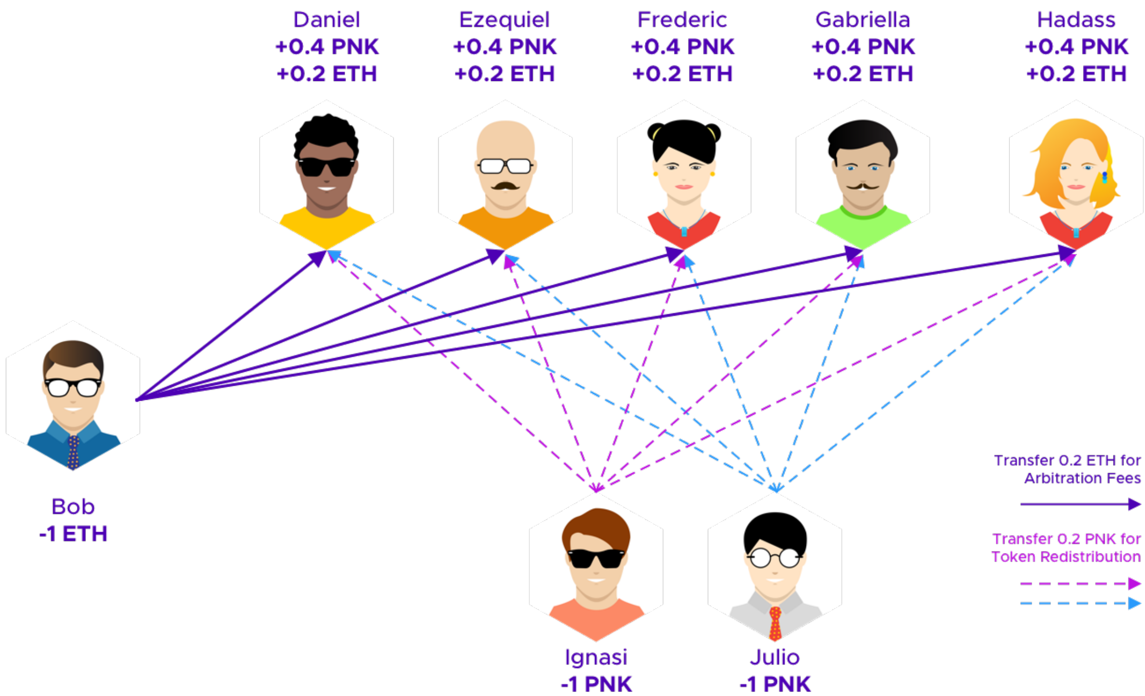

Figure 8: Seven jurors have a binary choice between ruling on behalf of Alice or on behalf of Bob. Tokens are redistributed from jurors who voted incoherently to jurors who voted coherently. Bob lost the dispute and pays the arbitration fees. The other deposits are refunded.

Remark 1. *Above we saw that the redistribution of arbitration fees and lost deposits is handled by round. Note that if a given voter then knows or suspects that other voters in her round have voted "incorrectly", this gives her more of an incentive to vote honestly. In the extreme, a single juror that agrees with the final outcome in a round where every other juror disagreed would receive all the arbitration fees and lost deposits for that round. We call this phenomenon the "lone voice of reason" effect. We will note further implications of this effect below.*

4.7.3 Parameterization of Arbitration Fees

Suppose we have a round of M jurors. One must choose f, the per juror average juror fee (i.e. so that the entire round requires $M \cdot f$ in arbitration fees), as well as the parameters α and min_stake described above that give the maximum deposit per juror vote that can be lost: $D = \alpha \cdot \text{min\_stake}$.

Again, denote the ultimate winning outcome by w. In the model of Section 4.7.2, imagine an honest juror that takes time and effort valued at e and as a result has:

$$E\left[\frac{\sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}: a_j < w} w(j)}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k: a_j < w} w(j)} \Big| \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array}\right] \geq \frac{1}{M},$$

namely that, on average, her return is at least as high as the average juror return.

Note that:

$$\frac{Mf + \sum_k D \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k: a_j > w} w(j)}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k: a_j < w} w(j)} \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}: a_j < w} w(j) - D \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}: a_j > w} w(j)$$

$$= \frac{Mf + MD}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k : a_j < w} w(j)} \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}: a_j < w} w(j) - D.$$

Denote

$$\pi_* = \text{prob}\left(\begin{array}{c} \mathcal{USR}_k \text{ puts} \\ w \text{ last } \forall k \end{array} \right)$$

Then, we can calculate the expected value of this honest strategy, based on the payoffs above:

$$E[\text{honest}] = (1 - \pi_*)\, E\left[\text{honest} | \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array} \right] - \pi_* (D + e)$$

$$= (1 - \pi_*)\, E\left[\frac{Mf + MD}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k : a_j < w} w(j)} \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}: a_j < w} w(j) - D - e | \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array} \right] - \pi_* (D + e)$$

$$\geq (1 - \pi_*)\,(f - e) - \pi_* (D + e).$$

On the other hand, noting that the special case where all voters in a round rank w last corresponds to the minimum payoff, a "lazy" strategy adopted by the voter $\mathcal{USR}_l$ that does not expend any effort in the case has expected value:

$$E[\text{lazy}] = (1 - \pi_*)\, E\left[\frac{Mf + MD}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k : a_j < w} w(j)} \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}: a_j < w} w(j) - D | \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array} \right] - \pi_* D$$

$$= (1 - \pi_*)\, M(f + D) E\left[\frac{\sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_l : a_j < w} w(j)}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k : a_j < w} w(j)} | \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array} \right] - D.$$

As long as the best available "lazy" strategy has a lower average payoff than that received by the average juror, i.e. as long as:

$$E\left[\frac{\sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_l : a_j < w} w(j)}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k : a_j < w} w(j)} | \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array} \right] \leq \frac{1}{M},$$

it is possible to choose D sufficiently large such that the lazy strategy has a negative expected return. Then, one should choose f and D such that:

$$(1 - \pi_*)\,(f - e) - \pi_* (D + e) > 0 > (1 - \pi_*)\, M(f + D) E\left[\frac{\sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_l : a_j < w} w(j)}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k : a_j < w} w(j)} | \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array} \right] - D.$$

Here one can view $\pi_* = \text{prob}\left(\begin{array}{c} \mathcal{USR}_k \text{ puts} \\ w \text{ last } \forall k \end{array} \right)$ and $E\left[\frac{\sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_l : a_j < w} w(j)}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k : a_j < w} w(j)} | \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array} \right]$ as quantities that can be observed empirically in a given court over time.

We can perform a similar analysis under the model of Section 4.7.1. We suppose that the honest strategy determines the winning answer with probability p. Suppose that all jurors other than $\mathcal{USR}$ adopt the honest strategy, so if we consider the jurors as choosing their answers independently, the number of other jurors who vote for the ultimate winning answer is binomial $X \sim Binom(M - 1, p)$. Suppose a "lazy" juror that does not evaluate the case can choose the right answer with probability $t \in [0, p]$ (for example because the court tends to side with a contractor over a business owner in a t proportion of cases). Then:

$$E\,[\text{honest}] = pE\left[\frac{Mf + (M - X - 1)D}{X + 1} \right] + (1 - p)(-D) - e = (f + D)(1 - (1 - p)^M) - D - e$$

and

$$E\left[\text{lazy}\right] = tE\left[\frac{Mf + (M - X - 1)D}{X + 1}\right] + (1 - t)(-D) = (f + D)\frac{t}{p}(1 - (1 - p)^M) - D.$$

Namely, in this case, the above constraints become:

$$(f + D)(1 - (1 - p)^M) - D - e > 0 > (f + D)\frac{t}{p}\left(1 - (1 - p)^M\right) - D.$$

If these are satisfied, the honest strategy is Bayesian-Nash incentive compatible. Note that in the special case of the model of Section 4.7.2 where jurors make a binary choice hence $w(j) = 1$, the two models are the same. We calculate the probability that no juror votes for the ultimate winner as $(1 - p)^M$, and

$$E\left[\frac{\sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_l : a_j < w} w(j)}{\sum_k \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR}_k : a_j < w} w(j)} \Big| \begin{array}{c} \exists k : \mathcal{USR}_k \text{ does} \\ \text{not put } w \text{ last} \end{array}\right] = \frac{t}{1 - (1 - p)^M} E\left[\frac{1}{X + 1}\right] = \frac{t}{Mp}.$$

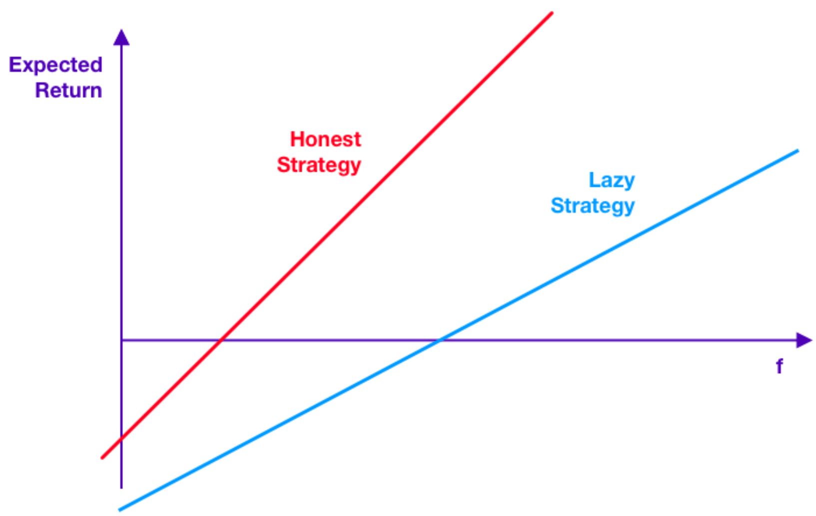

Figure 9: The parameters f and D should be chosen such that the strategy of performing effort to review the case and voting honestly has a positive expected value and such that the lazy strategies of voting randomly or always making the same vote have a negative expected return. Here the choice of D affects the positions of these curves and should be chosen such that there is an acceptable range from which to choose f.

Ultimately, the values f and D are chosen by the governance mechanism, see Section 4.11. Particularly, in order to follow this reasoning, the community must estimate values such as e[31]. Hence, if adequate data is available to justify using a more sophisticated model for how jurors' efforts vary over the population, the governance votes are capable of following more subtle versions of this argument.

[31] Note that one would want the different cases being considered by a given court to generally require similar levels of effort so that jurors do not apply the "honest" strategy on easier cases and the "lazy" strategy on difficult cases. Nevertheless, even if the cases in the court are of roughly equal difficulty, some jurors will need to exert less effort than others to come to honest rulings.

4.8 Appeals

If, after the jury has reached a decision, a party is not satisfied (because she thinks the result was unfair), she 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 ($\text{appeal\_fees} = \text{new\_amount\_jurors} \cdot \text{average\_fee\_per\_juror}$).

The number of jurors increases exponentially as one appeals; hence 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. Hence, via the appeal mechanism, Kleros manages to avoid the unnecessary duplication of effort and high costs that would be required by having a very large number of jurors consider every case while nonetheless the possibility of appealing a high number of times provides a defense against an attacker bribing the jurors. See Section 4.10 for a further discussion of this point.

Different models can be adopted by arbitrable contracts for gathering the required appeal fees, specifying different consequences when fees are not paid. These models present various tradeoffs that we will discuss below. Note that, based on our standards for arbitrable and arbitrator contracts [31], this logic is generally coded into arbitrable contracts. Hence, which model to use can be adapted to the specific requirements of different applications. Some of the models in this section have already been implemented in arbitrable contracts that use Kleros; others are proposed for future implementations.

A first, basic choice is the following:

- In the case of appeal, the appellant must pay any required appeal fees. This has the advantage that the party that won the previous round cannot be caused to lose without a further ruling by the jurors merely because she did not pay appeal fees. On the other hand, this has the disadvantage that parties who ultimately win their appeals do not have their appeal fees reimbursed, hence it may only be worth appealing relatively high value disputes.

- Alternatively, each side can be required to pay sufficient appeal fees to cover necessary costs in case they lose the appeal. Then, if only one option is (fully) funded, then that option would win by default without an additional appeal round. To mitigate the issue that a party that won a previous round can lose merely as a result of not paying adequate fees, we require additional stake to be paid by parties funding an appeal beyond the fees required to pay jurors for the subsequent round, where the option that won the previous round might require less stake. Then this stake is used to incentivize "fee funders". Namely one encourages third parties to pay the appeal fees for options, particularly those of the previous round winner, in exchange for the possibility of winning the stake of the other side. This structure is similar to litigation funding [37]. Note that fees can be funded collectively by a "crowd", as we will describe below[32].

4.8.1 Crowdfunding

In this section we will describe a mechanism by which third party "funders" are incentivized to cover appeal fees on behalf of parties to a dispute. We will provide several models for doing so, again each with its own tradeoffs. In designing these models, one has the following constraints:

[32] Other mechanisms designed to protect less well-financed parties that may not be able to pay large appeal fees are also possible and are an active subject of research. For example, parties to disputes can participate in a collective "appeal fee insurance". Here the parties would deposit amounts greater than the required first round juror fee when initially creating their dispute. Then the loser of the case does not receive back this difference, rather it goes into a pool of money that is used to pay appeal fees of parties that had won the previous round when required. These different models, fee litigation funding and fee insurance, each have their own tradeoffs, though they can be used together.

- at least the first funder in an appeal period has to stake on a given, single choice so that if the fees of no other choice are paid then there is no dispute and that choice becomes the default winner.
- at least one funder must be wrong per round so that they can cover the appeal fees for that round.

We use the following notation for this section:

- x is the total fees required by the jurors in the subsequent appeal
- s_{a_i} is the additional stake required beyond arbitration fees in order to fund the outcome a_i; namely, the total deposit that must be made to fund a_i is $x + s_{a_i}$[33].

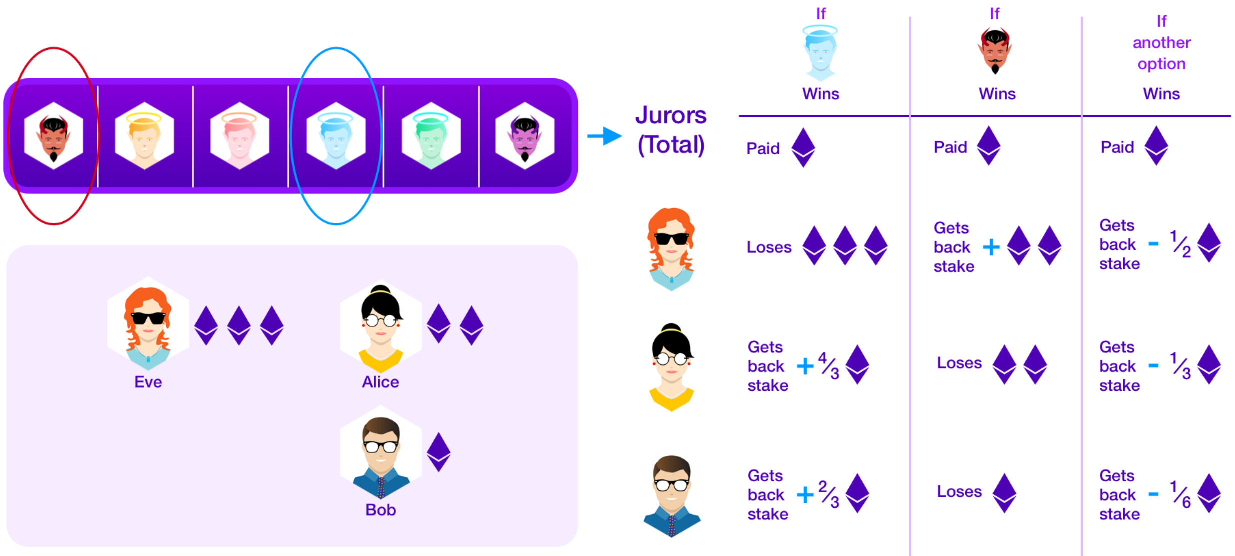

Figure 10: If Eve funds the appeal fees of a dishonest choice, Alice and Bob can collectively crowdfund the appeal fees of another (honest) choice. Crowdfunders are incentivized to participate because they can win a stake that is paid from the appeal fees of the opposing side beyond what is required to pay the jurors. Any situation where only two options are funded ressembles this schema, regardless of which of the crowdfunding models below is used.

Different designs can be adopted:

- In order to fund an option a, one must deposit $x + s_a$. Then once two options are funded, a dispute is raised. If some other option ultimately wins, the two funders share the arbitration fees for their round, but otherwise their deposits are returned. We will see that this model has reasonably good resistance to the "clone funding" grief described in Section 4.10. However if there are multiple options that could be considered to be honest, this model does not give the possibility to the funder to hedge by funding them together.

[33]It is possible that during the course of an appeal, the governance mechanism for a court will change its required arbitration fees. A variety of options for how to handle this, from requiring sides that had previously been fully funded to contribute more to requiring the difference from the remaining party, are possible.

- During the appeal period, users can pay $x + s_{a_i}$ for the choice a_i to be funded. Any choice that is not funded is eliminated from juror consideration in all future rounds. This has the advantage that cases are likely to binarize themselves after a small number of rounds, removing complications coming from having multiple choices. On the other hand, winnowing possible outcomes primarily through requirements to pay fees creates the risk that no good option is funded in a round and jurors are presented with non-sensical disputes. Moreover, an approach to winnowing the set of options that depends so heavily on fees paid could be seen as plutocratic. Also this has the disadvantage that it is incompatible with the current standard for arbitrator and arbitrable contracts [31].

- Suppose that option b won the previous appeal round. Then an appeal is called if for some option $a \neq b$ and some set of options S that does not contain a but may contain b, $x + s_a$ is staked on behalf of a by one or more funders and $\gamma(\#S)x + \sum_{a_i \in S} s_{a_i}$ is staked on behalf of S by one or more funders, where $\gamma(\#S)$ is an increasing function satisfying $1 \leq \gamma(\#S) \leq \#S$ for all values $\#S$[34]. If a wins, the funders of a receive back their deposits plus $(\gamma(\#S) - 1)x + \sum_{a_i \in S} s_{a_i}$ proportionally based on their contributions. If an option in S wins, then the funders who contributed to the fees of S receive s_a proportionally based on their contributions. If an option outside of $\{a\} \cup S$ wins, then the side insuring a receives back her deposit minus $\frac{x}{\#S+1}$ and the funders of S receives back their deposits minus $\frac{x\#S}{\#S+1}$. If there are multiple options that could be considered to be honest, this model gives the possibility to the funder to hedge by funding them together. However, for the simplest choice of γ, $\gamma(\#S) = 1$, this model has worse resistance to the "clone funding" grief described in Section 4.10.6 than the previous model. We will see in Proposition 6 that if one chooses $\gamma(k) = \frac{k+1}{2}$, this model has comparable resistance to clone funding as the first model, while maintaining its good properties with respect to allowing hedging for funders, at the expense of additional complexity.

In all of these models, if insufficient funds are raised for a given option or group of options to trigger an appeal, those funds should be returned to their contributors. This encourages funders to participate in this process without taking unnecessary risks on whether an option will manage to be fully funded.

Remark 2. *One notes that in our models above we allow the possibility for several choices to be funded together only on one "side" of the dispute. Ideally, one could place any elaborate "bet" on side of the dispute, funding some combination of outcomes to hedge as necessary. However, as mentioned in our constraints above, the first "side" that is funded must indicate a single option that becomes the default option if no other fees are paid. One could imagine attempts to adapt, such as taking the highest ranked choice from the previous round among a set of options that are funded together to be the default choice. However, this is vulnerable to attacks where a malicious party that has managed to corrupt a previous round can ensure that all high ranking choices are malicious and then fund one high ranking choice together with enough honest choices that it will not be viable for honest parties to fund another side to provoke a dispute.*

Remark 3. *This crowdfunding mechanism does not completely resolve the issue of making appeal fees accessible. While (different) funders can cover the fees of different sides of a given dispute, they are ultimately playing a negative sum game against each other as the funders collectively must pay the arbitration fees for the appeal round. In particular, funding two sides of a dispute would typically only be possible in cases where funders have very different priors for the chances of the victory of different outcomes. Indeed, there will generally be ranges of the funders' priors in which*

[34] Note that a compatible set S can be found if it exists by taking all options a_i for which at least s_{a_i} has been raised.

the case is not sufficiently clear-cut to fund any party. This issue becomes more acute as the number of possible outcomes grows; with many options, an attack could push a dispute to some clearly false option without there being a single honest option with sufficient chances of victory to be funded in appeal. This is particularly true in the presence of clones that may drive down the probability that any given honest answer wins. Hence, when designing an arbitrable contract, one should consider combining crowdfunding with pre-ruling appeal fee insurance as well as putting in careful reflection on the possible outcomes proposed, avoiding unnecessary clones.

4.8.2 Funding Sets of Choices Together

We briefly show a few results in the context of allowing multiple choices to be funded together, i.e. the third crowdfunding model discussed above. Particularly, we see that participants funding additional choices does not harm actors who had already (partially) funded. We first see that it can make sense for funders on the side of a set of choices to contribute to several different choices as this can maintain positive expected value of return while reducing variability.

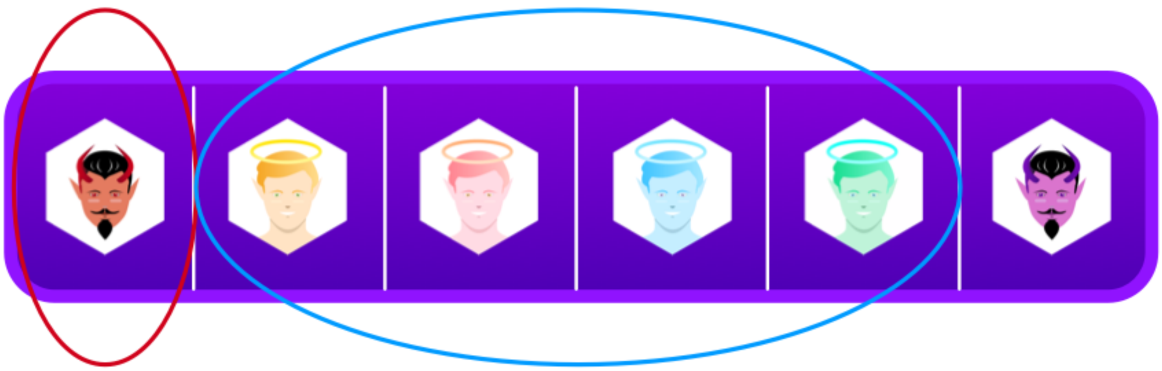

Figure 11: When one employs the third crowdfunding model above, funders can finance the fees of a set of options. Here, in contrast to the situation of Figure 10, crowdfunders stake on the honest options collectively and win if any of those options ultimately wins the dispute. This requires a larger contribution from the crowdfunders than if they had only funded a single option, however generally the additional cost of adding an option is sub-additive, with the degree of sub-additivity controlled by the choice of γ. This allows for a form of hedging that can preserve the effectiveness of crowdfunding even in the presence of clone options.

Proposition 2. *Suppose the function γ has the property that $\gamma(n) + \gamma(1) \geq \gamma(n+1)$, for all $n \in \mathbb{N}$. Then if funding the set of outcomes $\mathcal{S}$ has a non-negative expected value and funding the outcome a_j individually has a non-negative expected value, then funding $\mathcal{S} \cup \{a_j\}$ collectively has a non-negative expected value.*

Proof. Suppose that b is the option that has been funded by the opposing side. Take $\#S = n$. Then

$$E[S \cup \{a_j\}] \geq E[S] - E[a_j] \Leftrightarrow$$

$$p_b x\left(-\gamma(n+1)+\gamma(n)+\gamma(1)\right)+(1-p_b)\left(\frac{nx}{n+1}+\frac{x}{2}-\frac{(n+1)x}{n+2}\right)+\sum_i p_{a_i}\left(\frac{(n+1)x}{n+2}-\frac{nx}{n+1}\right)+p_{a_j}\left(\frac{(n+1)x}{n+2}-\frac{x}{2}\right)\geq 0.$$

The first term is non-negative by our assumption on γ. The other terms are clearly non-negative as $n \geq 0$.

□

Proposition 3. *Suppose that funding the option b has a non-negative expected value if the opposing side has funded a_j for any option $a_j \in \mathcal{S}$. Then funding b has a non-negative expected value if the opposing side has funded $\mathcal{S}$ collectively.*

Proof. Let $n = \#\mathcal{S}$. Then for $n \geq 1$,

$$E[b] = p_b\left((\gamma(n)-1)x+\sum_{a\in\mathcal{S}} s_a\right)+\left(\sum_{a\in\mathcal{S}} p_a\right)(-x-s_b)+\left(1-p_b-\sum_{a\in\mathcal{S}} p_a\right)\frac{-x}{n+1}.$$

Then for $n \geq 2$,

$$E[b]-E[b:\ \text{remove } a_1 \text{ from } \mathcal{S}]$$
$$= p_b(\gamma(n)-\gamma(n-1))+p_b s_{a_1}+p_{a_1}(-x-s_b)+\left(1-p_b-\sum_{a\in\mathcal{S}} p_a\right)\frac{-x}{n+1}-\left(1-p_b-\sum_{a\neq a_1\in\mathcal{S}} p_a\right)\frac{-x}{n}.$$

As $p_{a_1} \geq 0$,

$$\left(1-p_b-\sum_{a\in\mathcal{S}} p_a\right)\frac{-x}{n+1}-\left(1-p_b-\sum_{a\neq a_1\in\mathcal{S}} p_a\right)\frac{-x}{n}$$

is clearly non-negative. However,

$$p_b s_{a_1}+p_{a_1}(-x-s_b)\geq E[b \text{ against } a_1]\geq 0$$

by assumption. Furthermore, γ is increasing by assumption. One can complete the argument by induction with the $n = 1$ case holding as

$$E[b \text{ versus singleton}]-E[b \text{ if opposition unfunded}]=E[b \text{ versus singleton}]\geq 0$$

by assumption.

□

So, in the third crowdfunding model above, our expected behaviour is for funders on the side that must be funded as a single option to consider that option they evaluate as being the most likely to win. They will then expect that other funders will not fund very similar options or "clones"[35]. Hence, the funder will tend to either win by default (which has an expected value of zero) or to be pitted against choice(s) which are meaningfully different from the option he funded. Then Proposition 3 shows us that if a funder Frederick thinks an option is worth insuring against others individually, it will still be worth insuring even if he is pitted against several options collectively. On the other hand, following Proposition 2, funders who fund only a part of the fees for the side of the collective may reasonably fund one or multiple options as they hedge balancing expected returns with variability.

[35]Funding such a clone would have a negative expected return, see Proposition 6 in Section 4.10 for analysis of this subject. However, an attacker may be willing to accept a negative expected return in exchange for being able to "grief" the honest funders by also reducing their expected returns. Note, as we see in Proposition 6, if the number of possible clones is small this grief is not very effective.

4.8.3 Parameterization of Stakes

In Section 4.8.1, we saw that funders' returns for financing a successful appeal depend on the arbitration fees that must be paid to the jurors as well as the amounts of additional stake paid by the funders of each side. The choice of stakes in a given application depends on the arbitrable contract being used. In this section, we make some observations that are useful when choosing these values. We generally take the stake required for all options other than the option that won the previous round to be the same. Then funding the option that won the previous round might require less stake.

In the first crowdsourcing model, where only two options can be funded, funding the option a that won the previous round has a positive expected value if:

$$E[a] = p_a s_b + p_b(-x - s_a) + (1 - p_a - p_b)\frac{x}{2} > 0$$

for any option $b \neq a$. If we want to choose the stakes so that it is always worthwhile to fund a previous round winner that is also estimated by a funder to be the option with the single best chances of winning, then one can assume that $p_a \geq \frac{1}{n}$, where n is the total number of possible outcomes. Then the above inequality is satisfied if

$$s_b > s_a + \frac{nx}{2}.$$

Namely, the stake required of the losers of the previous round should be chosen in accordance with the stake required of the previous round winner[36]. Then, as one similarly has:

$$E[b] = p_b s_a + p_a(-x - s_b) + (1 - p_a - p_b)\frac{x}{2} > 0 \Leftrightarrow p_b > \frac{p_a\left(\frac{x}{2} + s_b\right) + \frac{x}{2}}{s_a + \frac{x}{2}},$$

one can choose s_b so that non-previous round winners are insurable at acceptable thresholds of p_b.

Reasoning for the other crowdfunding models is similar. In the third model, it is possible that only one option will be funded in the set S. Based on Propositions 2 and 3, this can be viewed as a worst case scenario for incentivizing fee funders. Hence, it is natural to require the same stakes in this case as in the first model. For the second model, which eliminates outcomes that are not funded, one might want funding the option a that won the previous round to have a positive expected return as long as $p_a \geq \frac{1}{\#S}$ for any possible set S of options that are funded. Then, as all options other than a require the same stake s, it is sufficient to have:

$$E[a] = p_a s\#S + (1 - p_a)(-x - s_a) > 0,$$

for which we see that it is sufficient to have $s > x + s_a$.

4.9 Forking

An attacker with a large percentage of active PNK who nevertheless fails a 51% on a given case loses a percentage of the stakes for each time she is drawn. However, as only a subset of PNK are drawn

[36] Note the linear dependence of this inequality on the number of possible outcomes. This may be inappropriate for a large number of outcomes, in which case one might choose the stakes so that a previous round winner only is guaranteed to have a positive expected return if it has a larger probability of winning. For example, if one assumes that $p_a \geq 1/2$, then $s_b > s_a + x$ is sufficient. Alternatively, one could slightly modify this model so that in the event that neither a nor b wins, the funders of the previous round winner receive back their full deposits with the additional burden of arbitration fees going to the funders of other options. This has the advantage that the relationship between the stakes no longer depends on n, at the expense of making funding previous round losers even less advantageous and increasing added code complexity.

in any given case, even in a late appeal round the attacker would only typically lose a relatively small proportion of her total holdings in lost deposits. In order to maximize the cost of 51% attacks, in this section we will propose an "ultimate vote round" in which all PNK holders vote. This is inspired by a similar mechanism in Augur [35], adapted to the situation of Kleros.

The idea is that one can have a "fork", creating two versions of the system depending on the result of a particularly contentious vote, where the fork on which an attacker held her tokens would be seen as malicious by the broader market, and her tokens would not be valuable compared to those on the "honest" fork. In current development there is no formal mechanism for facilitating such a fork; however, nothing prevents community members from creating a copy of Kleros in which the result of the contentious decision is reversed and attempt to build a social consensus around using this copy as the main version of Kleros going forward. The remainder of this section details a mechanism that would facilitate the community grouping around forks with desirable properties, as we will see below.

An important feature of Kleros is that cases are often subjective. Hence, it is possible that there will be cases where jurors legitimately disagree. Indeed, we can imagine that a very narrow result in a late appeal round could be the result of any of the following phenomena:

- An attempted 51% attack that is trying to pass through an obviously dishonest result.
- A deep ideological split in how certain types of cases are viewed[37].
- Honest disagreement on the specifics of a given case that would have little bearing on future cases.

In the first case, one would want there to be a fork to remove the attacker's influence; in the third case, one would not want there to be a fork as this would unnecessarily fracture the community. A fork over an ideological difference may be justified depending on how serious this difference was viewed by the community. In order to determine what situation we are in, in our "ultimate forking round" PNK holders will specify both their vote in the case as well as information indicating whether they think the case is worth forking over.

Of course, a mechanism that considers a majority vote on whether a case is worth forking over would be ineffective against a successful 51% attack as the attacker would control the outcome of both votes. Instead, PNK holders specify a percentage of total PNK that would go to a given fork at which point they would also be willing to fork. Then a given PNK holder can set this percentage so that her remaining tokens stay with the main fork regardless of the outcome if she thinks the case is a result of an honest disagreement, or so that she definitely forks away from a successful 51% attack. Alternatively, the PNK holder can set this percentage to some intermediate value so that she forks only if there is sufficient support on her side of an ideological split for the new fork to be viable.

We envisage writing arbitrable contracts in such a way that, with unanimous consent of the concerned parties, an arbitrator for an existing contract can be replaced with a fork of Kleros. Hence, this limits the ability of a successful 51% attacker to hold Kleros users hostage except in the relatively rare situation where the attacker has a direct interest. User interfaces can alert concerned parties to the fact that there has been a fork and over what case this fork was made.

[37]For example, the literalist/spirit of the rules type of split that one might have seen with the Augur "who will control the US house after the midterms" market if it had gone on long enough to force a fork [33]. Also compare to the Ethereum/Ethereum Classic fork over how to handle the DAO hack.

4.9.1 Forking Mechanism

We imagine that a given token holder's utility for a given case with a potential fork as a function:

$$\text{utility} = \text{fct}\begin{pmatrix} \text{case outcome} & & \text{percentage of tokens that are on} \\ \text{on the fork} & , & \text{the same fork as me after the case} \end{pmatrix}.$$

This allows for a possible trade-off between how egregiously incorrect/unacceptable the winning answer is and the breakdown of how the community splits. We can imagine cases where someone would think that outcome A is rather unjust, and it would be worth forking to a universe where outcome B won, but only if a large percentage of the community forked with her. Otherwise, if only some marginal amount of the community would have been willing to fork over this case, she prefers tolerating outcome A and remaining in the main branch[38].

We expect that the utility function should be monotonic in the percentage of tokens going to the same fork as you. Namely, all else being equal, we assume that participants would not prefer that the fork they are going to be smaller, as this would be a sign that it would be less likely to catch on. These dynamics are reminiscent of the "battle of the sexes" coordination problem in game theory, where two parties try to coordinate on two possible outcomes, and while they have different preferred outcomes, their preference for landing on the same outcome as the other party is stronger than their preference for their better outcome.

Hence we propose the following: after a case has been appealed the maximum number of times, a final voting round is triggered in which all PNK holder participate[39]. Each PNK holder USR_i submits an ordered list $(a_{ij}, r_{ij}^0) \in L(A) \times [0,1]^{\#A}$, where $r_{i1}^0 < r_{i2}^0 < r_{i3}^0 < \ldots$[40]. The user's choice of r_{ij}^0 will essentially allow her to specify a minimum threshold for community support for a fork where a_j is considered the winner at which USR_i would want to join this fork[41].

The "main fork"; is the one where the outcome a_{main} corresponds to the choice which is selected as the "winner" for settling any payments of ETH for this case and, by default, in other existing contracts. This winning option could be chosen as follows: we can consider the the ordering of options by r_{ij} as a ranking, which provides enough information about voter preferences that the same voting system used as in Section 4.6.2 can be used to determine a winner in this "ultimate forking vote round".

On each fork, the winner on that fork for the purposes of redistribution of PNK for coherence in previous votes is the option selected by that fork. Hence, if a juror in an earlier round believes that a given decision may require a forking round and she has confidence in her ability to choose the fork on which PNK retains market value, the reasoning around the incentivization of the earlier rounds that we have in Section 4.7 still holds. Note that it is thus possible that PNK which is "lost" to Bob on

[38] This possibly abstracts both the price they might expect the tokens to get on markets going forward after the case as well as their morality/altruism and willingness to participate in a system that they view as just or unjust.

[39] Note that, to a great degree, the forking procedure we describe here facilitates what could already be done by the community via social consensus outside of the formal protocol. In the case of social consensus forks it is also possible for a subset of the community to realize that a given outcome is going to win (or already has won), and fork without waiting for and paying appeal fees for a large number of appeals. In future work, we may consider how to expand the forking mechanism to allow a group to coordinate around a fork in an early appeal round, hence facilitating a healthy community in a fuller range of situations where a social consensus fork could have been employed.

[40] This is not optimally expressive; a voter might be willing to fork to a_1 at a lower level of support than she demands to fork to a_2 but nonetheless be more willing to be part of a "large" fork to a_2 than a "small" fork to a_1, a preference that is not expressible in this model. However, richer expression requires a more complicated user experience and may require a greater running time to resolve the result.

[41] Note that a user specifying a percentage r_{ij}^0 as a threshold for support for a fork at which she is willing to participate is equivalent to specifying a threshold for the number of tokens going to that fork $r_{ij} \in \mathbb{N}$; while it is particularly concrete for users in the interface to think in terms of percentages, we consider these perspectives interchangeably.

a fork where Alice is incoherent may still be held by Alice on the fork where she is coherent. On the other hand, we do not redistribute for the coherence of PNK holders' "votes" in the forking round except insofar as they determine which fork PNK holders wind up on[42].

Any PNK that is staked in some court, but for which a set of forking preferences is not provided during the forking period are slashed. Any PNK that is not staked is sent to the main fork[43]. After the result of each fork is used for PNK redistribution in previous rounds, the tokens that take part in the forking vote receive a bonus such that the total number of tokens on each fork is ultimately equal to the number of tokens on the original fork. Essentially from the point of each fork redistributing the PNK that went to the other forks to the "coherent" PNK that remained. However, unstaked PNK not used to vote do not receive this bonus; this incentivizes all PNK holders to participate in the forking vote. Indeed, it is essential to have very high turnout in any fork vote to maximize resistance to 51% attacks.

Now we will use a procedure that attempts to find the single largest fork(s) compatible with users' expressed preferences[44]. One can perform:

- For each choice $a_j \in A$ such that $a_j \neq a_{\text{main}}$
  - Take the list L_j of voters $\mathcal{USR}_i$ for whom a_j is listed as an acceptable choice for some r_{ij}.
  - Take the sum of the number of tokens over all voters on L_j. Denote this by s_j.
- For each option j:
  - Sort L_j by the user's r_{ij}.
  - Take the user USR_i with the largest r_{ij}. Compare r_{ij} to s_j. If $r_{ij} > s_j$, remove USR_i from L_j.
  - Recalculate s_j as the sum of the number of tokens over all voters on L_j.
  - Repeat the two previous steps until L_j stabilizes.
- Then, over the possible choices j, take the option that maximizes s_j and for which $L_j \neq \varnothing$.
- All remaining token holders on L_j go to an alternative fork where a_j wins.
- Remove the token holders on L_j and the option a_j from the originally provided preferences and repeat this process until there is no j produced such that $L_j \neq \varnothing$. This determines if there are still compatible forks possible.

[42] As the forking round will not have exactly the same incentive mechanisms as in normal appeal rounds, neither in terms of ETH nor PNK redistribution, one might worry that PNK holders will not give "full rankings". Indeed, if USR_i provides a single $r_{ij} = r_i$ for all "acceptable" choices, the system begins to resemble approval voting, which is very sensitive to where jurors draw the line for what is "acceptable" [38]. This may nonetheless be acceptable/appropriate for a forking round vote, as ultimately we want the forking system to be able to allow the community to move on in the case of extremely contentious decisions.

[43] Particularly there is no potential for token holders to wait until they see the market reaction to the different versions of PNK after the vote before making a choice.

[44] Note that in the context of non-binary decisions, there are various ways to translate this information on preferences into a set of consistent forks. Instead of using a rule based on the single largest fork, one could have, for example, attempted to maximize the total number of tokens that fork. This choice has its tradeoffs; for example in maximizing the single largest fork one may have situations where there was an alternative fork that was almost as large that also allowed secondary forks that satisfied a larger percentage of the token pool, an outcome that one might expect to produce higher utility. By choosing this forking rule, we make a choice that is more easily computable, avoids fragmenting the community more than what is necessary, and has good properties with respect to the presence of clones as we will see below.

	% of Total PNK	Vote	Threshold for willingness to fork to b	Destination Fork
	26%	a	NA	a
	20%	a	NA	a
	12%	b	12%	b
	11%	b	35%	b
	9%	b	40%	a
	9%	a	NA	a
	7%	b	20%	b
	6%	a	NA	a

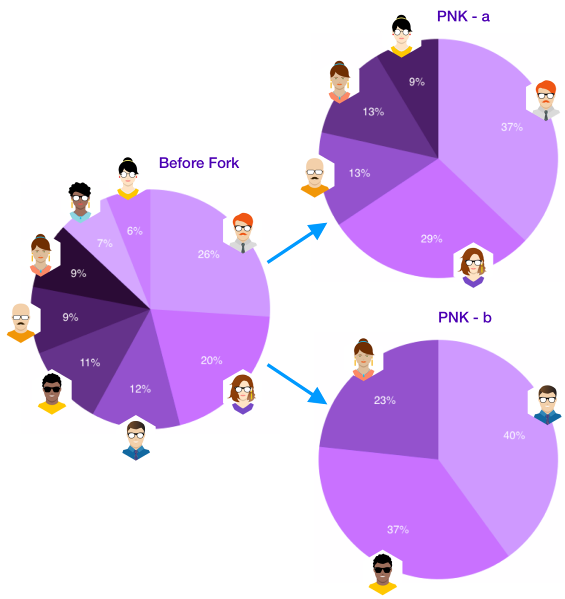

Figure 12: Here a pool of PNK holders vote in a forking round, also expressing their willingness to go to a minority fork. The PNK holders who vote for a would also generally express a threshold for their willingness to go to a minority fork where a wins, but as a already wins the vote here that information is not used and we suppress it for simplicity in the image. Note that one PNK holder voted for b in the forking round, but as a minority fork to b involving him would only have 39% support, lower than his threshold, he ultimately remains on the (main) a fork.

4.9.2 Properties of Forking Proposal

Proposition 4. *The above process determines the single largest possible fork.*

Proof. We see that the first two loops allow us to determine the largest possible percentage of tokens that can fork to each option j. Indeed, a simple inductive argument shows that none of the voters removed in the second for loop could have gone to this fork under any subdivision. On the other hand, all voters that remain on L_j after this process have $r_{ij} \leq s_j$ as the list is sorted; hence if they all go to the alternative fork together this is compatible with their choices. Then as we maximize over all outcomes, we find the largest possible fork.

□

The first for loop takes $O(\#AN)$ steps, where N is the number of voters. Sorting L_j takes $O(N \log N)$ time. The steps of the second for loop after the sorting require $O(N)$ time, so the entire for loop takes $O(N \log N)$ time, repeated $\#A$ times, so the second for loop takes $O(\#AN \log N)$ time. The remaining steps require $O(\#A + N)$ steps, and this whole process is repeated at most $\#A$ times for a total running time of $O(\#A^2 N \log N)$.

The above process could be coded directly in the smart contract controlling the forking process. However, to reduce gas costs, it is also possible to use an "optimistic" process where different fork possibilities are submitted during some interval of time, and the contract then determines which produces the single largest fork. Note that a smart contract, given a possible set of forks presented as a list of token holders sent to each outcome with the outcomes pre-sorted by the size of their forks, can determine if it is valid (with respect to users' expressed preferences) and if it has a greater largest fork than a current choice.

- For each choice $a_j \in A$ such that $a_j \neq a_{\text{main}}$
  - Take the list L_j of voters $\mathcal{USR}_i$ who fork to the choice a_j under the proposed set of forks.
  - Take the sum of the number of tokens over all voters on L_j. Denote this by s_j.
- Verify that the s_j are monotonic, i.e. that the forks where submitted order by size. (If the s_j are not monotonic, return "false".)
- Starting with the largest s_j, check that $s_j \geq f_j$, the size of the largest fork in the current fork choice. (If the algorithm arrives at a value for which $s_j < f_j$, return "false".) Halt at the first value of j for which $s_j > f_j$ and return "true".
- If the algorithm does not halt during the comparisons between s_j and f_j, i.e. if $s_j = f_j$ for all j, return "false".

The for loop requires $O(\#AN)$ steps, and then the remaining steps require $O(\#A)$ steps. Hence, the total running time for this on-chain verification is $O(\#AN)$ steps.

We introduce the idea of clone independence in the forking process. It is inspired by the idea of clone independence in ranked list voting systems, see [8].

Definition 1. *For a given set of preferences expressed by token holders, options a_1 and a_2 are said to be clones if the thresholds $r_{i1} = r_{i2}$ for all token holders $\mathcal{USR}_i$. A forking system is said to be clone independent, given a set of options $\{a_1, \ldots, a_k\}$ that are clones, deleting the option a_j from consideration does not change any of the forks produced for options outside of the set $\{a_1, \ldots, a_k\}$ (either by creating new such forks or deleting old ones).*

Proposition 5. *The single largest fork rule is clone independent.*

Proof. For computing how many tokens are willing to fork to an option from a set of clones, all the clones will have the same total. Furthermore, deleting one clone does not change this total.

□

4.10 Attack Resistance

In order to be a reliable dispute resolution system, Kleros needs to be able to withstand malicious behaviour from participants. In this section, we will discuss the resistance of Kleros to specific attacks that have been identified as being relevant.

4.10.1 Buying Half of the Tokens Attack

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 would need to 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. Indeed, tokens have increasing marginal costs; they will be dynamically priced on exchanges. Should one party buy a significant percentage of them, the price would go up due to market depth making it increasingly costly to acquire further tokens.

Finally, such an attack could lead to a fork as described in Section 4.9. While this fork would not prevent attackers that successfully obtained half of the tokens from being able to decide the outcome of a given case, it can isolate those attackers' tokens from the rest of the community limiting their resale value and forcing the attacker to absorb their cost. See Figure 13 for a summary of these effects.

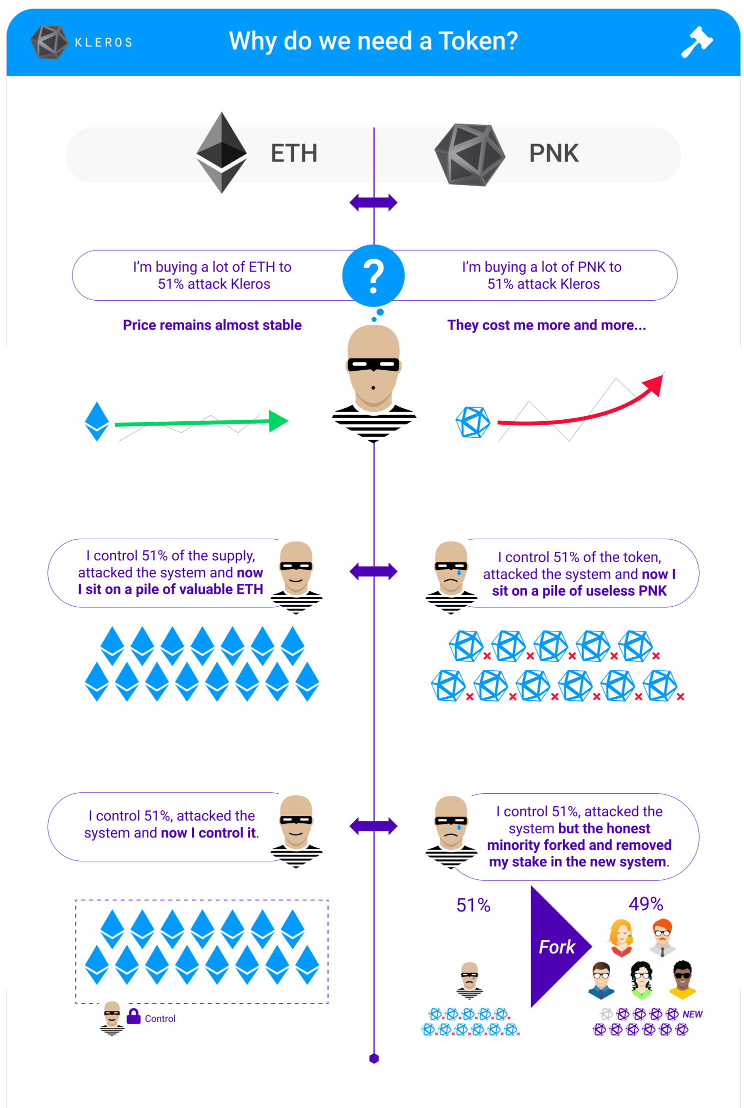

Figure 13: A summary of effects of having a system token, as described in Section 4.4.1, on resistance of Kleros against attacks that attempt to buy up a majority of tokens.

4.10.2 Bribe Attack

Appeals are an important mechanism against bribes. Bribing a small jury is relatively easy. However, 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[45].

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. 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.

4.10.3 $p + \epsilon$ Attack

A p+ϵ attack is a type of elaborate bribe that promises to pay the bribe only if the attack is unsuccessful, see [13]. Such attacks are particularly designed to target Schelling game systems, warping their incentive structure. These attacks require a high budget but have zero cost if successful. Already in [13] 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 [24] for the results of these experiments. One can also find in [24] comments on how round-based redistribution, which as we saw in Remark 1 leads to the "lone voice of reason" effect, reduces the viability of such attacks.

4.10.4 Frivolous Appeals Attack

While the appeal system serves as a defense against bribe attacks, it creates the possibility for attackers with substantial resources to attack the system by appealing cases beyond the point where the other party to case can afford to pay appeal fees[46]. We have detailed in Section 4.8.1 a number of possible mechanisms by which arbitrable contracts can incentivize third party funders to fund the appeals of parties they believe to be honest so that this effect is mitigated.

4.10.5 Delay Grief

A related attack would be for an attacker to appeal merely because she wants to delay the execution of a ruling. For example, perhaps she is a competitor of the honest party and hopes to delay when he will receive funds from the case. Unlike in the previous attack, we do not consider the attacker as having the goal of changing the result, so our defenses cannot prevent an attacker from spending money in

[45] Note that similarly to the "buying half of the tokens" attack, ultimately the viability of the attack depends on whether 50% of the token pool is corrupted. Here, as the attack does not hold the tokens, the market depth effects discussed above would interact somewhat differently with a potential attack. It is still the case, however, that a successful attack on the General Court would dramatically decrease the value of the tokens (who wants her contracts to be adjudicated 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 her bribing offer to be successful (which in almost all cases would exceed the value at stake in the dispute).

[46] Note that, as a pure attack, this is only relevant in the appeal fee model where both sides pay appeal fees and the winning party is refunded, see Section 4.8.

appeal fees to achieve her desired delay[47]. However, again, as appeal fees increase exponentially over the rounds, an attackers should not be able to maintain this delay for long.

4.10.6 Clone Funding Grief

We describe a grief on crowdfunding mechanisms, see Section 4.8.1, in the context of non-binary choices. We call this grief "clone funding". Here the idea is that Frederick funds some (honest) option a_j to which there is a very similar option a_k. Funding a_k is unlikely to be profitable as one would expect that jurors rule for the two options with roughly equal likelihood, however due to the fees that must be paid to the jurors, Frederick and the attacker would then be playing a negative sum game with even likelihood of victory. This grief is relevant in any of the crowdfunding models considered in Section 4.8.1. In the following proposition, for the models that do not allow hedging by funding choices together for the purposes of the following proposition, we take $\gamma(k)$ to be the number of options that are funded (excluding the one funded by Frederick).

Proposition 6. *Assume that all participants possess the same estimates for the probabilities of winning of each outcome. Suppose Frederick funds an option b that he estimates to have the highest probability of eventually winning. Then the strategy of funding k other option(s), excluding the previous round winner so all options considered require the same stake, with the aim of reducing Frederick's expected return has a griefing factor of at most $\frac{1-\gamma(k)+k}{\gamma(k)}$. In particular, if $\gamma(k) = \frac{k+1}{2}$, then the griefing factor is at most one.*

Proof. We use the notation of Section 4.8. Suppose Eve funds options $a_1, \ldots, a_k$ each of which has $p_{a_i} \le p_b$ and $s_{a_i} = s_b = s$. Then, using $\frac{p_b + \sum_i p_{a_i}}{k+1} \ge \frac{\sum_i p_{a_i}}{k}$ and $\sum_i p_{a_i} \le 1 - p_b \le 1 - \frac{1}{1+k}$, we have

$$\begin{aligned}
E[\text{Frederick}] &= p_b\left((\gamma(k)-1)x + \sum_i s_{a_i}\right) + \left(\sum_i p_{a_i}\right)(-x - s_b) - \left(1 - p_b - \sum_i p_{a_i}\right)\frac{x}{k+1} \\
&\ge s\left(kp_b - \sum_i p_{a_i}\right) + x\left(p_b(\gamma(k)-1) + \frac{\sum_i p_{a_i}}{k} - \frac{1}{k+1} - \sum_i p_{a_i}\right) \\
&\ge x\left[\frac{\gamma(k)-1}{k+1} + \left(1 - \frac{1}{k+1}\right)\left(\frac{1}{k} - 1\right) - \frac{1}{k+1}\right] = -x\frac{1-\gamma(k)+k}{k+1}.
\end{aligned}$$

Similarly, using $p_b \ge \frac{1}{k+1}$, we have

$$\begin{aligned}
E[\text{Eve}] &= \left(\sum_i p_{a_i}\right)s_b + p_b\left(-\gamma(k)x - \sum_i s_{a_i}\right) - \left(1 - p_b - \sum_i p_{a_i}\right)\frac{kx}{k+1} \\
&= s\left(\sum_i p_{a_i} - kp_b\right) - \gamma(k)xp_b \le \frac{-\gamma(k)x}{k+1}.
\end{aligned}$$

Note these bounds are sharp when $p_b = \frac{1}{k+1}$.

□

In particular, we note that in the model where only two choices can be financed, clone funding has a griefing factor of at most 1. This can also be obtained in the model that allows hedging by funding options together at the expense of a more complicated function for $\gamma(k)$.

[47] An attack where someone accepts to pay a cost to also harm a victim, namely where both the attacker and the victim are worse off, is referred to as a grief, see [14].

4.11 Governance Mechanism

As the Kleros protocol gains users and use cases, it will be necessary to create new courts, to make changes in court policies and parameters and to update the platform to new versions with additional features. Such decisions will be made by token holders who have a number 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 resolve 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 court.

2. Create new courts.

3. Modify parameters in courts such as:

   (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[48].

5 Additional Future Work

Above, we have already addressed several points on which we intend to improve existing aspects of the protocol in future work. In this section we consider a few other planned improvements not previously discussed.

5.1 Redistribution of Funds in Rounds Where No Juror is Coherent

As discussed above, arbitration fees and lost PNK deposits are redistributed on a per round basis between voters that are coherent with the final outcome. If no juror in a given round is coherent, these amounts are currently sent to the governor, and can be allocated by the governance process.

Even when there are only three jurors in a round, it will be relatively rare for a round to have no coherent juror. However, in order for Kleros to be a cost effective alternative to very small scale disputes, such as content moderation, see the examples in Section 6, it will sometimes be necessary to start with initial juries of a single voter. In this case, it is a problem that jurors in this first round will never win PNK deposits from other jurors (because there are no other jurors), however they can nevertheless lose their own PNK deposit to the governor, leading to a reduced incentivization. In future work, we are considering mechanisms by which the governor will automatically redistribute amounts that it accumulates due to rounds where no one is coherent back to the jurors drawn in the corresponding court. Hence, this will average out the lost deposits from situations where jurors attempt to rule honestly but nevertheless wind up being incoherent with the final ruling.

[48] Audits and reviews will be made before the code is deployed. However, it can never be guaranteed 100% that there is not a bug (either in the code or incentives) somewhere. Having this fail-safe provides extra security.

5.2 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 $\text{hash}(\text{contract\_text}, \text{option\_list}, \text{salt})$ (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 $\{\text{contract\_text}, \text{option\_list}, \text{salt}\}$ 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 $\{\text{contract\_text}, \text{option\_list}, \text{salt}\}$ 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.

5.3 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.

On some level, one might not expect seeing the vote of a given other juror as being particularly convincing, as even if a given outcome has a publicly visible majority of the votes in the current round, if the outcome is dishonest, jurors who would vote for the minority might expect there to be an appeal. By the "lone voice of reason" effect that we observed in Remark 1, in fact knowing that other voters in your round have voted "incorrectly" gives you *more* of an incentive to vote honestly.

Nevertheless, in many cases it would be useful to have mechanisms to punish jurors who reveal their vote prematurely. A simple approach in this sense would be 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 would impede jurors from revealing their votes trustlessly but nevertheless has limitations[49]. A more complete potential solution that we are examining for eventual inclusion in Kleros could be based off the work in [12]. Other possible approaches, drawing on the idea of collusion-resistance [15] could provide more complete security on this problem. There, due to a mechanism where jurors can change submitted votes prior to the deadline, jurors would not be able to credibly prove to others that they voted in a given way, ideally even after the case is resolved, which would also have the effect of inhibiting bribe attacks.

[49] It is still possible for jurors to give insight about their votes. For example, they can issue a zero-knowledge proof against their vote commitment, or by making a smart contract with themselves committing to vote in a certain manner which burns a deposit if they vote differently.

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 who choose not to vote directly to delegate their vote using a liquid voting mechanism [22]. When a user fails to vote, her voting power will be automatically transferred to her delegate. You can see an illustration of the liquid voting mechanism in Figure 14. Vote delegation can also be court specific. Users could choose to delegate their vote in some courts 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).

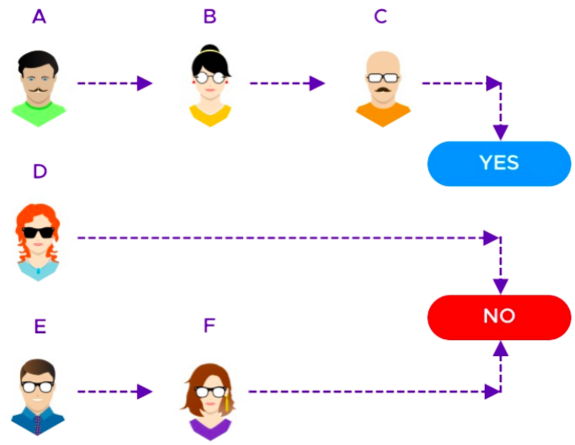

Figure 14: Illustration of a liquid vote

5.5 Settlement Logic

There may arise situations where, over the course of a series of appeals, the original parties to a dispute reach an agreement for how they want the original, disputed amount to be divided and wish to prematurely end the dispute resolution process. We intend to add logic to templates of standard arbitrable contracts to allow this[50].

However, during the course of the dispute other parties, such as jurors and fee funders, will have become financially interested in the result being correct. Indeed, Kleros depends on allowing third parties to finance and call appeals to protect jurors in the event of certain attacks. Hence, it is necessary that the dispute be allowed to continue on, even in the absence of the original parties.

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 [27].

[50] The current version of the Kleros Escrow has offers a limited structure for settlements, however only before the triggering of a dispute.

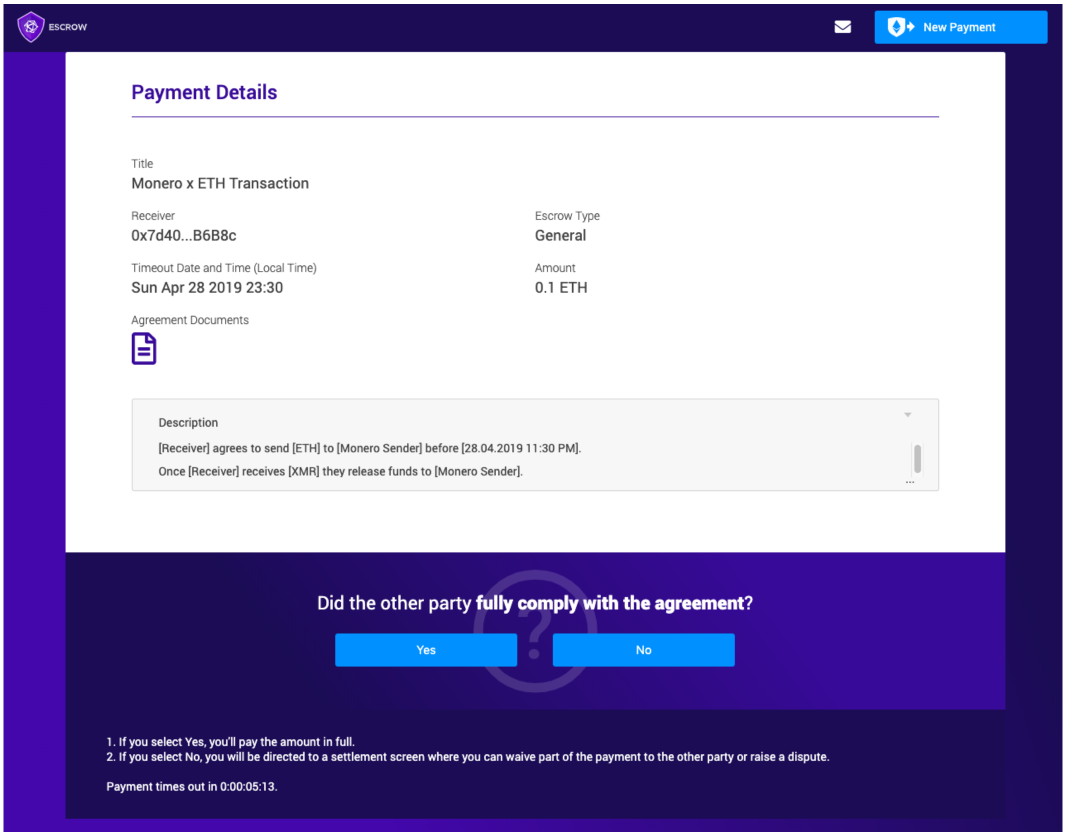

Figure 15: A user of a Kleros escrow deciding whether to raise a dispute.

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 a 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 [10]. 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 [5]. Moreover, other applications that use the Realitio oracle, such as CryptoUnlocked, [32], indirectly depend on this dispute resolution. Particularly, we have researched ways in which such processes can be efficiently adapted when the oracle is required to output a real-number value, such as in the case of a price oracle [25].

- **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) [28]. One particularly notable possible application of Kleros would be a curated list of distinct individual humans, where jurors would be required to consider limited biometric information (such as facial features) to mediate challenges that submissions are either of nonexistent people or are of people already on the list. Such a "proof-of-humanity" system would allow for a Sybil resistant list of people, which could have applications for quadratic voting schemes [29], more effective airdrops, etc.

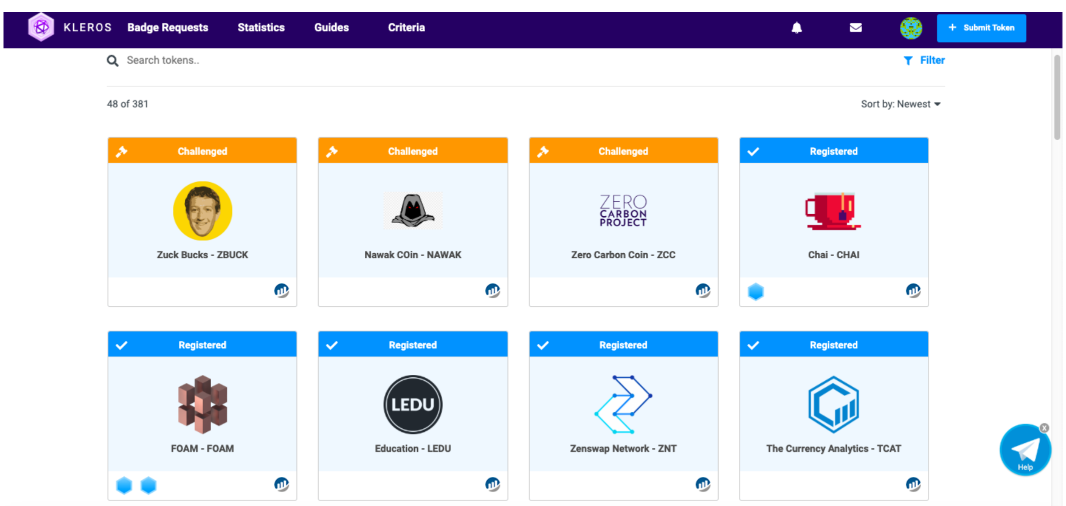

Figure 16: A Kleros-based token curated registry of tokens, where the address and logo of the token must be correct for the token to be allowed on the list.

- **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 her 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 her other posts can be lowered.

7 Conclusion

We have introduced Kleros, a decentralized court system allowing dispute resolution in smart contracts by crowdsourced jurors relying on economics incentives. You can see a summary of how Kleros works in Figure 17.

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 arbitration mechanism. Kleros uses game theory and blockchain in a multipurpose dispute resolution protocol capable of supporting a large number of applications in e-commerce, 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 dispute resolution 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] Kenneth Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58, 1950.

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

[6] Manuel Blum. Coin flipping by telephone a protocol for solving impossible problems. *SIGACT News*, 15(1):23–27, January 1983.

[7] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, Sep 2004.

[8] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, New York, NY, USA, 1st edition, 2016.

[9] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, October 1988.

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

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

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

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

[14] Vitalik Buterin. The triangle of harm. https://vitalik.ca/general/2017/07/16/triangle_of_harm.html, 2017.

[15] Vitalik Buterin. Minimal anti-collusion infrastructure. https://ethresear.ch/t/minimal-anti-collusion-infrastructure/5413, 2019.

[16] Benedikt Bünzy, Steven Goldfeder, and Joseph Bonneauy. Proofs-of-delay and randomness beacons in Ethereum. http://www.jbonneau.com/doc/BGB17-IEEESB-proof_of_delay_ethereum.pdf, 2017.

[17] Craig Calcaterra, Wulf A. Kaal, and Vlad Andrei. Blockchain infrastructure for measuring domain specific reputation in autonomous decentralized and anonymous systems. https://ssrn.com/abstract=3125822. U of St. Thomas (Minnesota) Legal Studies Research Paper No. 18-11.

[18] Lyn Carson and Brian Martin. *Random Selection in Politics*. Praeger, 1999.

[19] Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pages 145–152, Arlington, Virginia, United States, 2005. AUAI Press.

[20] John R. Douceur. The Sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, UK, 2002. Springer-Verlag.

[21] O. Dowlen. *The Political Potential of Sortition: A Study of the Random Selection of Citizens for Public Office*. Luck of the draw : sortition and public policy. Imprint Academic, 2008.

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

[23] David Friedman. A positive account of property rights. *Social Philosophy and Policy*, 11, 1994.

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

[25] William George and Clément Lesaege. A Smart Contract Oracle for Approximating Real-World, Real Number Values. In *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019)*, volume 71 of *OpenAccess Series in Informatics (OASIcs)*, pages 6:1–6:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[26] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.

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

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

[29] Steven P. Lalley and E. Glen Weyl. Quadratic Voting: How Mechanism Design Can Radicalize Democracy. *AEA Papers and Proceedings*, 108:33–37, May 2018.

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

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

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

[33] P. H. Madore. Augur House elections market: Alleged reporter says Republicans won the market. https://www.ccn.com/augur-house-elections-market-alleged-reporter-says-republicans-won-the-market/, 2018.

[34] Andrew Mao, Ariel D. Procaccia, and Yiling Chen. Better human computation through principled voting. In *AAAI*, June 2013.

[35] Jack Peterson and Joseph Krug. 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.

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

[37] Rachel Rothwell. The rise of global litigation funding. https://www.raconteur.net/risk-management/the-rise-of-global-litigation-funding, 2017.

[38] Donald G. Saari and Jill van Newenhizen. Is approval voting an 'unmitigated evil'?: A response to Brams, Fishburn, and Merrill. *Public Choice*, 59(2):133–147, 1988.

[39] Mark Allen Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

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

[41] Markus Schulze. A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method. *Social Choice and Welfare*, 36(2):267–303, February 2011.

[42] P. Stone. *The Luck of the Draw: The Role of Lotteries in Decision Making*. Oxford University Press, 2011.

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

[44] T.N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4, 1987.

[45] Sam Vitello, Clément Lesaege, and Enrique Piqueras. ERC 1497: Evidence standard. https://github.com/ethereum/EIPs/issues/1497, 2018.

A Proof of Proposition 1

Proof. Denote by L the number of options. Denote by F the arbitration fees paid to be split between the voters of this round. For a given winning option and collection of votes cast by the other voters in the round, denote by

$$K = \sum_{\mathcal{USR}_k \neq \mathcal{USR}} \sum_{a_j \text{ not winner}} \mathbf{1}_{\mathcal{USR}_k \text{ votes } a_j \text{ after winner}}$$

the total number of pairwise votes on which users other than $\mathcal{USR}$ are coherent with the ultimate winning choice. Similarly, denote by

$$K' = \sum_{\mathcal{USR}_k \neq \mathcal{USR}} \sum_{a_j \text{ not winner}} \mathbf{1}_{\mathcal{USR}_k \text{ votes } a_j \text{ before winner}}$$

the total number of pairwise votes on which users other than $\mathcal{USR}$ are incoherent with the ultimate winning choice.

Based on our assumptions, the ultimate winning choice, as well as the votes of the other jurors in the current round, can be considered to be fixed with respect to $\mathcal{USR}$'s vote. Hence, K and K' are also fixed. Suppose $\mathcal{USR}$ places the ultimate winning choice in the ith position. Then she is correct regarding $L-1-(i-1) = L-i$ pairs and incorrect regarding $i-1$ pairs. Hence, when $\beta = 0$, her net payoff accounting for lost deposits is:

$$\text{payoff}(i) = \left([K' + (i-1)] \frac{D}{L} + F \right) \frac{L-i}{K+L-i} - (i-1)\frac{D}{L}.$$

In particular, her payoff is a function only of i (which is notably not the case when $\beta \neq 0$). Note that increasing i causes $\mathcal{USR}$ to lose an additional deposit of $\frac{D}{L}$ which is split between her but also other jurors based on their coherence. Hence, $\mathcal{USR}$ can only recover a portion of this lost deposit through her reward so

$$\text{payoff}(i) \geq \text{payoff}(i+1).$$

Then, for any given set of votes by the other voters in the voting round, a standard argument shows that

$$E[\text{vote } a_1 > a_2 > \ldots > a_L] = \sum_{j=1}^{L} \text{payoff}(j)\text{prob}(a_j \text{ wins})$$

$$= \sum_{j=1}^{L-1} \text{prob}(a_j \text{ wins})\text{payoff}(j) - \text{payoff}(L)\left(1 - \text{prob}(a_L \text{ wins})\right) + \text{payoff}(L)$$

$$= \text{payoff}(L) + \sum_{j=1}^{L-1} \text{prob}(a_j \text{ wins})\left(\text{payoff}(j) - \text{payoff}(L)\right)$$

$$= \text{payoff}(L) + \sum_{j=1}^{L-1} \text{prob}(a_j \text{ wins}) \sum_{i=j}^{L-1} [\text{payoff}(i) - \text{payoff}(i+1)]$$

$$= \text{payoff}(L) + \sum_{i=1}^{L-1} \left([\text{payoff}(i) - \text{payoff}(i+1)] \sum_{j=1}^{i} \text{prob}(a_j \text{ wins}) \right).$$

is maximized by maximizing $\sum_{j=1}^{i} \text{prob}(a_j \text{ wins})$ and hence by voting the options a_i in order by their probability of winning. As this is true for any give set of votes by the other voters, and these votes are independent of $\mathcal{USR}$'s vote and the eventual outcome, we have the desired result. □

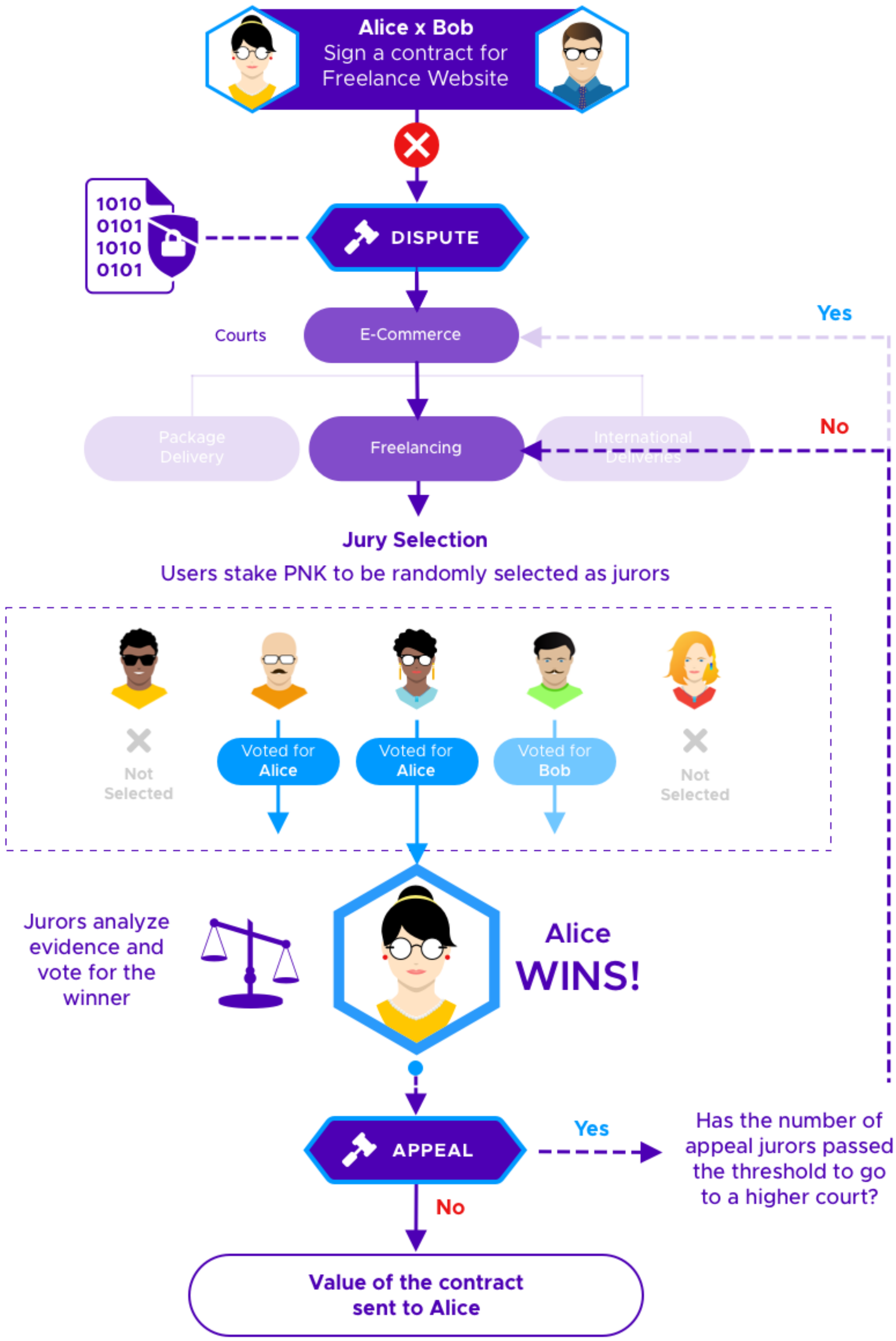

Figure 17: Example of dispute summing up how Kleros works.