Argument:

“... The user interface of the Kleros Cooperative erroneously produced the case closure without giving the opportunity to appeal the first round result. Beyond the economic damage produced to the appeal founder, it is important to stress that the resulting lack of justice can potentially undermine confidence in the protocol, should the issue remain ignored. ...”

Refutation:

There was no economic damage to appeal funder because the bug actually prevented the economic damage: The maximum he could win was 0.868 ETH only if the counterparty funds the appeal as well and only if jurors rule in favor of Mente's side. And in case of appeal there is also the possibility that he could lose up to 1.736 ETH, reducing his total expected return.

Argument:

“... On March 15, the developers of the Kleros Coopérative, created a button to withdraw the ETH blocked due to the UI mistake (Issue closed on github). The ETH of the appellant could be released, four days after the appeal was funded. …”

Refutation:

Appeal funds were never blocked. There was no button on UI to withdraw it, but you could always directly interact with the contract and withdraw it. Also, the button was implemented within 2 days after the dispute got resolved.

Argument:

“... In our understanding, the Kleros Cooperative must be held responsible for the UI error that prevented the effective appealing of case 164, failing in providing justice and putting at risk the Bribe Resistance property of the protocol. Furthermore, we remark that the appealing System Dapp does not have a disclaimer statement informing, for example, if it is a development in beta testing. As a consequence, the user is implicitly encouraged to use it as a bug free application. Actually, for some dapps (Linguo or GTCR at this moment) exist an advice of “product under development”, so any user is advised that these products are not 100% bugs free, or close to be launched (see Figure 3). Why not use some similar disclaimer that the UI could hold some bugs in the interaction with the smart contract?. Why not encourage you to show how to interact with the smart contract? …”

Refutation:

There is no such thing as %100 bug-free software, you can only prove bugs existence but not their inexistence. Kleros is not liable for any UI bugs. Furthermore, UI bugs cannot prevent appealing a case as you can always directly interact with the smart contract. User interfaces provide an easier way to interact with smart contracts.

Argument:

*“... If you are promoting, and pushing to Decentralize the Justice, you provide a tool and dapps
to create disputes, the environment for the jurors to vote, and the appealing system, you must provide a reliable system, or at least, advice to the users that this UI could do exactly
the opposite that you are looking for, justice.
For the reasons described above, the Claimants request an indemnification of 10 ETH. The
amount is based on the compensation that the Kleros Coopérative pays as rewards to people who reveal errors in some of the Dapps they develop (TCR Bounty or Kleros Bounty),
but mainly on the damages due to the lack of justice caused to the appellant in the case 164.
Also, a bonus that recognizes the loss of profit of the money stacked in the appeal. …”*

Refutation:

UI bugs don’t undermine Kleros' systems resistance or reliability, you can always directly interact with the contract. UI and smart contracts are separate things. Kleros is a system on the Ethereum network. Kleros pays for smart contract bugs, not UI bugs. And only in bounty time periods, not open-endedly.