Notes: Analysis of models for communication between dapps and between courts in scaled Kleros

January 8, 2021

A fundamental theme of many scaling approaches is to allow for systems where nodes do not have to process all transactions. Of course, participants to a sharded consensus protocol will only generally be required to process the transactions of their shard (and potentially those of a beacon chain). Similarly though, one does not expect *all* of Ethereum to migrate to the *same* rollup (and particularly not same optimistic rollup) as this would just push the issue of nodes verifying all transactions to the rollup participants[1].

Kleros should ideally be able to interact with a wide variety of applications on different rollups. Message transfers from rollups to L1, and in some cases from one rollup to another, are of course forseen by rollup developers. This communication generally requires delays, which are not necessarily critical to Kleros as Kleros already has execution times on the order of days. To the degree that such communication goes through L1, it also has costs.

This raises the question of whether different courts should all be on the same rollup, or whether they should be on different rollups (or even if some of them should be on L1). Note, however, that if the courts are not all on the same rollup, it becomes problematic to keep stakes across courts synchronized. Namely, Alice may stake 1000 PNK in court A, a child of the general court and then proceed to lose some of her stake due to incoherent votes in court A. If court A is not on the same rollup as the general court, then either the general court does not realize that Alice has lost PNK and continues to draw her with higher odds than she deserves, or potentially expensive L1 to L2 or cross-rollup communication is required between the the two courts to synchronize Alice's general court stake.

In these notes we begin to explore the choices we might make in efficiently structuring the court tree over (potentially different) rollups. The following three choices are natural:

- SR: All courts on single rollup

[1] See https://newsletter.banklesshq.com/p/when-defi-meets-rollup for discussion on this point from a member of Offchain Labs.

- DRL1GC: Courts on different rollups, generally along with dapp(s) that use them. General court on-chain
- DRL2GC: Courts on different rollups, General court also on some rollup

Before we analyze the pros and cons of each of these models, recall the threat model for optimistic rollup.

- The L2 generally itself has some consensus mechanism, which we consider less trustworthy than the L1 consensus[2]
- If the L2 consensus acts in a hostile manner (approving transactions incorrectly, selectively censuring transactions, or just completely freezing), L1 is capable of eventually determining what should be the state of L2, using the L2 transaction data which is publicly available in calldata.

Namely, a malicious L2 operator can result in whatever dapps are on that L2 being frozen for some period of time.

SR:

- Pros:
  - Court stakes remain synchronized
  - Minimal cross-rollup transactions for *jurors* (all stakind, destaking, voting on same rollup)
  - Probably the solution that is the easiest to implement, at least from the perspective of the court contract, which would look very similar to current contract
- Cons:
  - Dapps using Kleros generally on other rollups.
  - Delays and costs for dapps triggering dispute, paying appeal fees, etc.

DRL1GC:

- Pros:
  - Can put various courts on rollups with the dapp(s) that they are mostly used for - only have to worry about costs and delays from cross-rollup or L1 transactions when jurors want to destake/restake in different courts
  - By having general court on L1, if a rollup freezes/takes time to resolve due to malicious operator actions, the effects are limited to that rollup and don't mess with the general court or by extension Kleros courts and Kleros enabled arbitrable dapps on other rollups

[2] For example, Optimism intends to initially launch a "Single Sequencer Mainnet", namely a consensus mechanism where Optimism themselves acts as a PoA. They plan to launch a "Decentralized Sequencer Mainnet" in "202a".

  - Depending on how cross-rollup communication implemented in a given optimistic rollup scheme, this might reduce delays and costs versus having general court on L2. (For example, naive L2 to L2 communication might go through L1.)

- Cons:
  - Issues keeping stakes synchronized over courts on different rollups[3].
  - Still need to pay for L1 transaction when voting in general court
  - Still need to pay for L1 transactions when staking in general court and typically when adjusting stake from one court to another

DRL2GC:

- Pros:
  - Can put various courts on rollups with the dapp(s) that they are mostly used for - only have to worry about costs and delays from cross-rollup or L1 transactions when jurors want to destake/restake in different courts
  - Don't need to pay L1 gas costs to vote in general court
- Cons:
  - Issues keeping stakes synchronized over courts on different rollups.
  - If rollup on which the general court is located freezes due to operator misbehaviour and one has to wait for it to resolve on-chain, can potentially have weird effects on courts/dapps on other rollups
  - Still need to pay for L1 or cross-rollup transactions when adjusting stake from one court to another

Finally, note that many of these consideration are also applicable to courts/dapps on different shards or even on different chains connected by bridges.

[3] Can have mechanisms to mitigate this. E.g. if Alice loses more than x% of her stake in a rollup might have a transaction that can be issued by anyone to force general court to update its record of Alice's stake. However, can't do this at current level of granularity without eroding efficiency gains from L2. Typically will be the case that stakes over different courts somewhat out of date until user destakes and cross-rollup communication is done to sort out what they should be owed.