Mao is a card game played with a regular deck a bit similar to Uno (you win by getting rid of all of your cards) but with a custom rule mechanic: each round, the winner of the round creates a new rule, which applies to future rounds, but which they only reveal by enforcing it (and other players must guess what the rule is). Mao is (subjectively) a really fun game and also a good demonstration of function privacy in practice - hence perhaps a neat application of the Anoma architecture. Mao is also an episodic game and needs only local ordering, which makes it suitable for a local temporary fractal instance (where bets, if placed, could be settled elsewhere after gameplay completes).

Required components:

- Random deck shuffling and dealing to individual players
- Iterated gameplay (players playing cards in order)
- Validity conditions for what cards can be played
- Specific cards have specific effects, for example:
- Reversal of direction
- Next player plays two
- Changes in validity conditions
- Reversal of direction
- Next player plays two
- Changes in validity conditions
- User-defined rules, for example:
- Can change validity conditions
- Can create additional actions which must be taken
- Can change order of play
- Can change validity conditions
- Can create additional actions which must be taken
- Can change order of play

Mao only requires local ordering, so the basic flow should be as follows:

- Counterparty discovery to seek counterparts for a game of Mao.
- In case of bets taken (with funds transferred from elsewhere), a transaction on the parent fractal instance in order to register the game
- Players’ public keys
- Initial ruleset state
- Number of rounds
- Players’ public keys
- Initial ruleset state
- Number of rounds
- Creation of a temporary fractal instance between the players to play the game
- PoA network between players
- Each round, the winner can create a new rule
- Private to them, but committed to
- Private to them, but committed to
- PoA network between players

- Each round, the winner can create a new rule
- Private to them, but committed to
- Private to them, but committed to
- Gameplay with privacy
- Game completion & submission of results to parent instance (for bets and/or leaderboard, prizes)
- Results should consist of
- Signature by involved players
- Composed sequence of state transitions (all moves played) which can be validated
- Signature by involved players
- Composed sequence of state transitions (all moves played) which can be validated
- Mao leaderboard can be updated
- Results should consist of
- Signature by involved players
- Composed sequence of state transitions (all moves played) which can be validated
- Signature by involved players
- Composed sequence of state transitions (all moves played) which can be validated
- Mao leaderboard can be updated

This is a pretty sophisticated application and a good demonstration of what kinds of experiences generalized information flow can enable. It's also a fun game!

/cc @apriori @awa penny for your thoughts