

We wanted to share the force-move games

framework - a small, proof-of-concept state channel framework capable of running 3rd-party 'applications'. We will be publishing a white paper containing a full description of the framework and its capabilities in the coming weeks but the code is already [available on github](#). Along with the framework, we provide the contracts and client-libraries for a few example applications that can be built on top of it (the game of [rock-paper-scissors](#) and a [payment-channel](#)).

The framework takes a pragmatic approach, sacrificing generality of the supported applications, for simplicity in the on-chain and off-chain logic. In particular, the framework can't (yet) support any applications that have state transitions whose validity depends on time, or other properties that are external to the channel's state. For example, as it currently stands, you couldn't build a state channel sports betting game, where bets have to be made by a deadline and the winner is determined through an oracle feed.

Writing an application to run on the framework involves creating a smart contract that defines a set of states and a `validTransition`

function, which specifies the allowed transitions. The framework provides a `forceMove`

action through which a player can challenge their opponent on-chain to take a move. If their opponent doesn't respond within the timeout period, the game ends and the players receive a 'fair' settlement, which depends on the current state and is determined as part of the application design process.

The framework doesn't currently support ledger channels or virtual channels, so at the moment an on-chain deposit is required for each game played. We've tried to decouple the funding of the channels from their behavior though, to allow us to add support for ledger and virtual channels at a later date. This also allows the channels to run in other state channel setups, such as that being created by L4.

We'd like to thank L4 for all of their support, collaboration, and helpful discussions throughout, and James Fickel for providing the funding for this work!