**GlassCasino Presentation Appendix**

# Figures

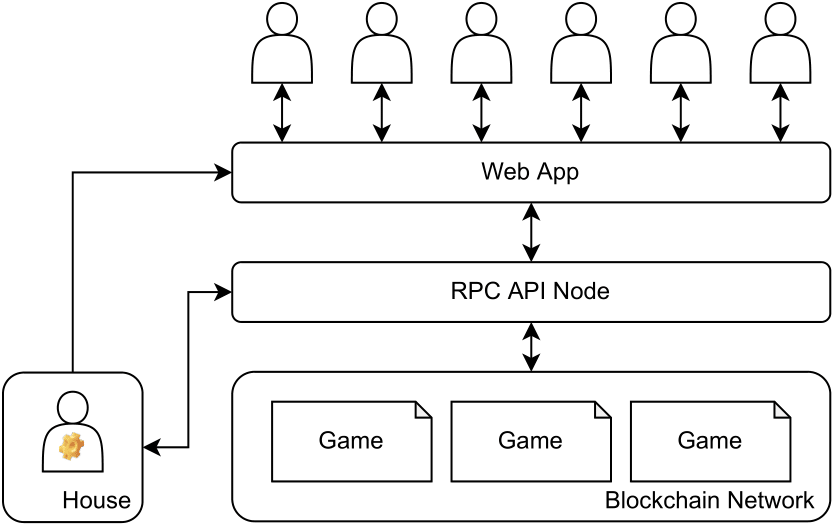


Figure 1 - The project architecture showing smart contracts deployed to a blockchain and how a user or automated user like the house can interface with said contracts.

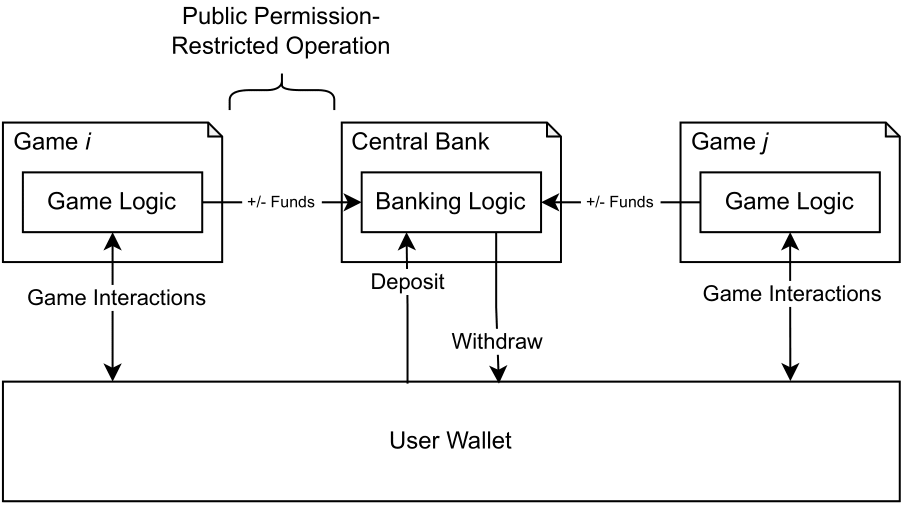


Figure 2 - Single-contract, low-cost on-chain central banking and its interaction with game smart contracts and the user.

Diagram

Description automatically generated

Figure - Deployment chain for game, banking, and migrations smart contracts.

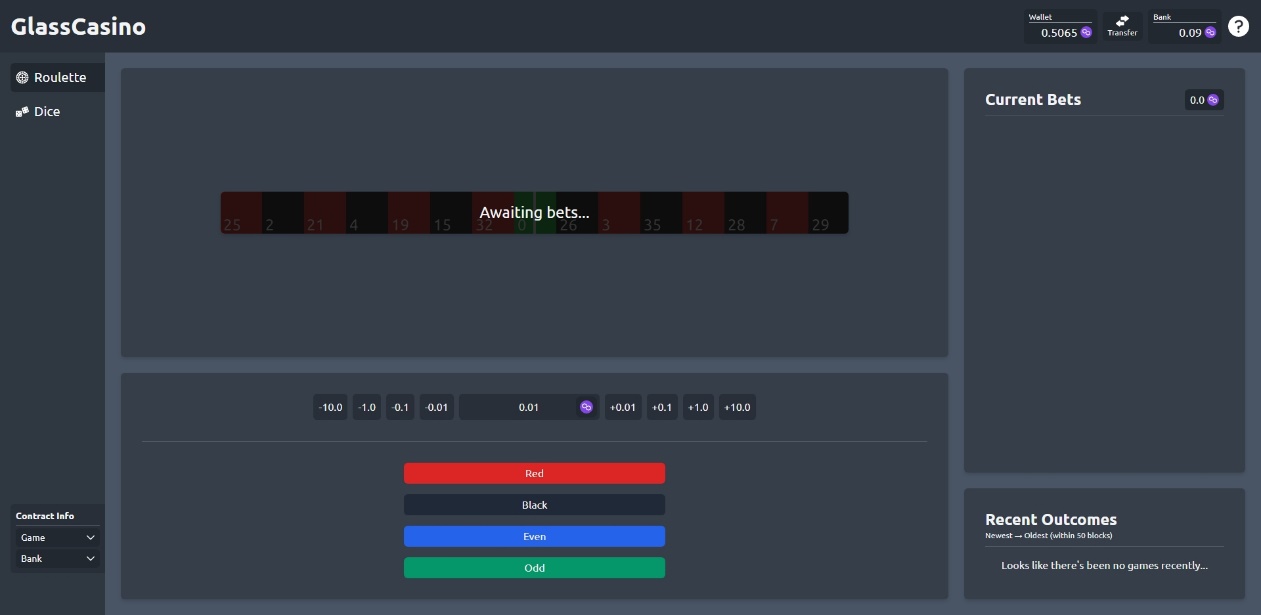


Figure 4 – Example of the completed roulette user interface on a 1920x1080 desktop viewport.

Diagram

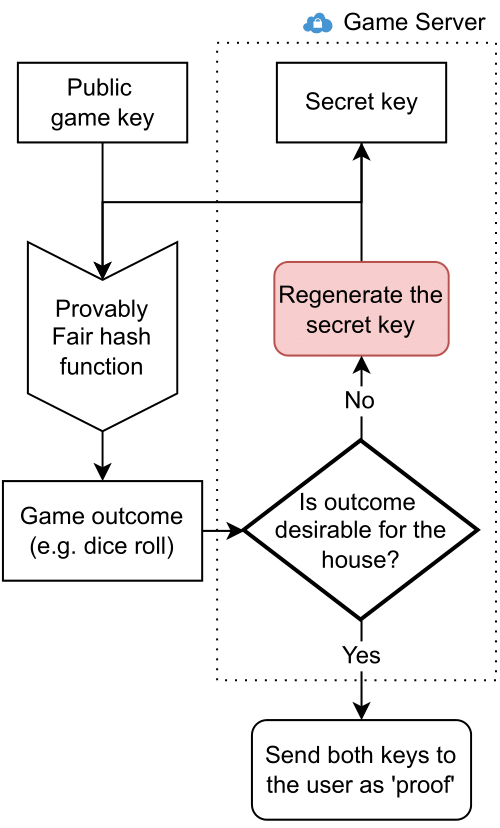
Description automatically generated

Figure 5 – Feedback loop that centralized iGaming platforms can use to produce malicious yet ‘provably fair’ game outcomes (left) compared against ChainLink VRF’s on-chain random value and its proof (note that ChainLink VRF takes ~10 blocks to generate as it uses future block headers as part of its seed).

Chart, line chart

Description automatically generated

Figure – Visual progress performance of GlassCasino compared against bc.game and CroissantGames. Visual progress measures speed to render a user interface.

Chart, histogram

Description automatically generated

Figure - Distribution of timings of entry-point transactions for each of our smart contracts against the density at which they occurred when executed 100 times.

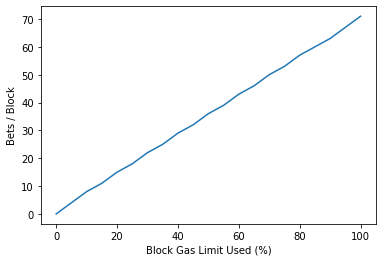
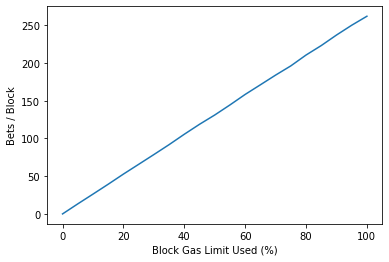


Figure – Scalability graphs for the roulette (left) and Chuck-a-Luck (right) smart contracts. These graphs indicate the maximum number of bets each game can fit in a block at a capped usage of the network’s global gas.