# Slide 1 – The Problem

We begin in establishing the problem to address.

That is, there is a lack of transparency and reliance on blind trust in the online gambling industry.

Online casinos, unlike physical ones, can conceal cheating their players nearly undetectably as their games run on their own centralized servers where the user will only ever see the outcome of the game like a black box.

This centralized model is susceptible to various types of scams and fraud but most importantly it causes distrust amongst players when they lose as shown by the stats here.

The UKGC finds 71% of the population don’t trust casinos!

# Slide 2 – The Project

Our work looks at a potential solution to these problem – blockchain. Thus, we pose the question:

**[CLICK]**

“How technically viable is it to implement an online casino on the blockchain?”

**[CLICK]**

We ask this because we know blockchain networks to have innate security and trustability, as per their design as listed here.

More recently, with the advancement of smart contracts popularized by Buterin in the 2014 Ethereum white paper, building out fully-fledged apps is possible and thus our work begins.

# Slide 3 – Objectives

Now that we’ve covered the posed question and why it needs to be asked - we can outline the objectives required to get us to a point where we can answer that question.

To do so we introduce a handful deliverables here:

- The smart contracts themselves. This the code governing either game logic or banking logic – i.e., deposits, withdrawals, and prize allocation etc. These are annotated ‘Game’ on Figure 1.

- The migration chain to get the contracts onto the blockchain and this ties in heavily with the banking logic

- An interface allowing easy user interaction with the platform just using their crypto wallet of choice

You might also have noticed the ‘house’ on Figure 1. This is some necessary off-chain logic required to manage certain aspects of certain games. In our completed work this is to manage the timing of spinning the roulette wheel.

With all these deliverables built and connected as shown in Figure 1 - we can measure the three tenets of mainstream technical viability: scalability, speed, and cost.

# Slide 4 – Achievements

Now let’s cover what this work consisted of a little bit more in-depth.

**Firstly, we’ve got two games – roulette and a dice game called Chuck-a-Luck. These are interesting as they’re both of slightly different paradigms in that they are both played against the house but either in a group or alone. This has some interesting ramifications in terms of throughput and cost particularly.**

Next up we’ve got the banking system as shown by figure 2 on your appendix. This was a solution to a problem that presented itself in development.

Originally, the plan was for each game to be self-contained. In that, you never deposit any money anywhere except your own wallet.

This led to poor usability and high fees.

Thus, we produced the system shown in figure 2.

It uses a permissions system to retain trust-lessness despite now having a centralized bank.

Effectively the bank has two roles it can assign to people or contracts: operators (who can add or subtract value to/from users’ bank balances) and admins who can designate new operators.

**Thirdly, we have the migration chain. This is the code to put the contracts on the blockchain. The flowchart in figure 4 walks through the process of designating and revoking admin and operator roles I spoke about earlier regarding banking.**

**The result of this chain is that there will be no bank admins and thus the list of operators (which will be a list of all deployed game contracts) is fixed, public and enumerable.**

Next, we have the web interface. While there’s not too much novel work here - it’s worth noting that the interface is very quick, very light and has been fully integrated with the most common EVM-blockchain wallet MetaMask to prevent the need for any sign-ups.

**Finally, coming on to the largest pain point for blockchain-backed online gambling: random number generation.**

**Figure 5 on your appendix show a traditional method for generating fair random numbers on the left and a blockchain method on the right.**

**The left flowchart has an obvious flaw as it partially seeds itself with hidden data and thus the vendor can manipulate it.**

**The right flowchart is an implementation from an the ChainLink organization which implements a network of oracle nodes running a Goldberg verifiable random function (VRF).**

**The basic flow of how it works is shown in the figure but it’s important to note that ChainLink has no control over the number generated, nor do miners or the casino owner as it’s seeded by all those parties individually with block headers, ChainLink private keys and random nonces and transaction data.**

# Slide 5 – Results

So, the scores we achieve in speed, cost and scalability are given to you Figures 6 through 8.

Figure 6 shows excellent performance for the web app, but it must be kept in mind that GlassCasino is not nearly as feature rich as bc.game.

However, for the smart contract speed shown in Figure 7, we see poor speeds. Even on the better performing games and the bank it still takes around 10 seconds for transaction confirmation.

Dice, as the VRF-backed game, sees speeds of an order of magnitude slower! This is something innate to ChainLink’s implementation. In fact, while developing this project they came out with a complete re-implementation of their oracle network.

As for scalability shown in Figure 8, we see some poor numbers. In the paper we make some very optimistic estimates of the scalability required to rival mainstream platforms and find we would need at minimum 20% of the Polygon network’s global gas limit to run these games 24/7. However, this could easily spike to 100% or more during peak times, thus flooding the network.

Because of all this we conclude **[CLICK].**

This.

Effectively, GlassCasino a good proof-of-concept for blockchain based online gambling. So, we find that while the technical possibility of decentralized iGaming exists, our platform cannot be deemed technically viable for mainstream adoption. This is due to speed and throughput limitations.

However **[CLICK]** a key stipulation is that this work is limited by the current smart contract ecosystem which, while growing rapidly, is still relatively immature. Given a better platform that hopefully doesn’t sacrifice decentralization for speed/throughput we believe this threshold could easily be passed.