Blockchain-based Online Casino

Adam Hodson
L3 BSc Computer Science Project
Durham Univeristy, UK
qvhf99@durham.ac.uk

***Abstract* —**

Background: -

Aims: -

Method: -

Results: -

Conclusions: -

***Keywords* —** Blockchain, Ethereum, Smart Contracts, Polygon, iGaming, gambling, Web3, Vue, Solidity

I. Introduction

In recent years, blockchain technologies have exploded in popularity. Particularly during the 2020-2021 COVID-19 pandemic where we saw a sharp rise in interest in these technologies; largely credited to NFTs (Dowling, 2021). The industry is now far beyond scope of an "electronic cash system" proposed in the seminal paper on blockchain (Nakamoto, 2008) as interest now lies in how classical, centralized systems can be moved to incorporate blockchain.

This project considers one of the practical applications of blockchain as a replacement for the status quo in the industry of iGaming (internet-based chance games). We aim to use to use blockchain technologies to create a transparent, cryptographically secure, and fair gaming platform.

A. *Background*

A blockchain is fundamentally a widely distributed ledger of transactions. It is backed by a consensus algorithm to stop bad actors from manipulating the transaction history for their benefit. A group of transactions is grouped into a 'block' with a unique identifier in the form of a hash. This hash, along with the transactions in the block, is used to create the next hash. For a new block to be added to the chain, consensus must be reached that it is valid. From this it is obvious that the longest, valid chain on the network is the source of truth.

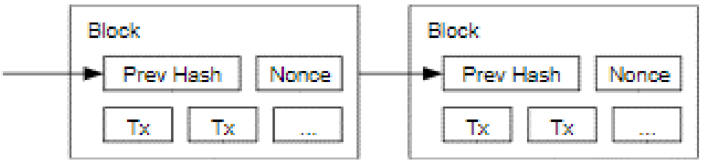

Figure 1 – The block structure used in Bitcoin (Nakamoto 2008).

The consensus algorithm proposed by Nakamoto in his 2008 paper is Proof-of-Work (PoW). Bitcoin's PoW algorithm works by incrementing a nonce value in the block until a value is found that satisfies the hash. Since editing the block would change the hash (and thus create the need to re-validate it) we end up with an immutable chain of validated transactions. To attack the network, the attacker would have to control more than half of the nodes of the network as otherwise the non-malicious nodes chain will outpace the attacker's chain. The act of validating a block is called mining and is often a pooled effort due to the enormous amount of computing power required. Miners are rewarded with 'coins' which are then traded around the network in the blocks' transactions, thus forming a currency.

However, PoW is slowly falling out of favor in the industry due to severe bottlenecking as the blocks' hash difficulty is raised and the high energy consumption (Li et al., 2019) raising environmental concerns. More modern blockchain projects like ADA or ADA are backed by Proof of Stake (PoS) consensus protocols. It is more sustainable in terms of energy consumption and is less vulnerable to malicious miners attacking the network (Nguyen et al., 2019). This is because blocks can only be validated by miners who have staked that much in a coin. This discourages powering up the most powerful computers en-masse and instead encourages investment into the cryptocurrency to be a validator.

Second generation blockchains like Ethereum (ETH) introduced the concept of a 'smart contract' (Buterin, 2014). This effectively allows Turing-complete programs to be written and deployed to the network. In this regard, Ethereum is more of a distribute state machine than a distributed ledger as each node on the Ethereum network runs an instance of the EVM to allow them to process the operations of a smart contract. This allows for anyone to deploy a smart contract with their own code, run it and receive completely deterministic results in a trustless decentralized network. This is an incredibly powerful concept and has countless possible applications that are only just beginning to be explored.

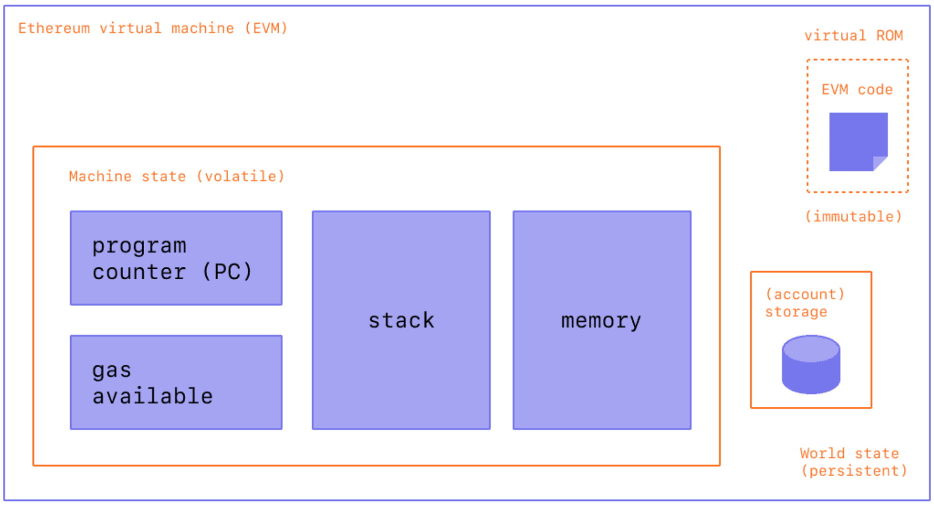

Figure 2 – Diagram of the Ethereum virtual machine (EVM)

II. Related Work

A. *Ethereum*

Ethereum forms the backbone of much of the work this project is built on in the form of smart contracts. Ethereum mart contracts are usually written in a high-level language called Solidity which is a classic curly-braced language. This was built for developer QoL as the EVM can, as a Turing machine does, only complete simple bitwise operations like AND, XOR, etc. with some extra blockchain specific.

Ethereum relies on two things: its primary token ETH and the concept of gas. ETH is sometimes called the Ethereum networks gas token. All chains implementing smart contracts follow suit with this concept. Gas is a way of measuring how much compute power a particular operation on a particular smart contract will require. The invoker of the transaction will pay a certain amount of ETH to perform their transaction on the network.

Ethereum was not built with scalability in mind and thus as a PoW, bloated network with uncapped gas fees it often leads to incredibly high fees to perform even simple operations as users bid each other up to get the miners to process their transaction in the next available block. Ethereum’s second major flaw is its impractically low TPS (transactions per second) of ~15 across the whole network. Bitcoin suffers a similar issue with ~6 TPS. For reference a centralized payment provider like Visa can handle ~24,000 TPS. This severely limits the practical applications of Ethereum in real-time applications like gaming. The upcoming 2.0 upgrade should help to fix these issues but for now, many dApp (decentralized apps) developers are looking towards alternatives.

B. *Polygon (formerly Matic)*

Polygon is a network that aims to solve the numerous problems faced by Ethereum (Kanani et al., 2021). It tackles this issue using an adapted version of the Plasma framework (Poon & Buterin, 2017). In essence, a Plasma chain runs on top of an existing blockchain (the rootchain) so that one does not need to create transactions on the rootchain for every single transaction on the Plasma chain.

Plasma blockchains are level 2 chains or ‘chains on a chain’. It does not disclose the contents of the Plasma chain to the rootchain, instead only block hashes are submitted by Plasma validators utilizing PoS consensus. If there is proof of fraud on the root chain, i.e., the block, cannot be validated by the Ethereum network, the block on the Plasma chain will be rolled back and its creator penalized via deducting some of their stake.

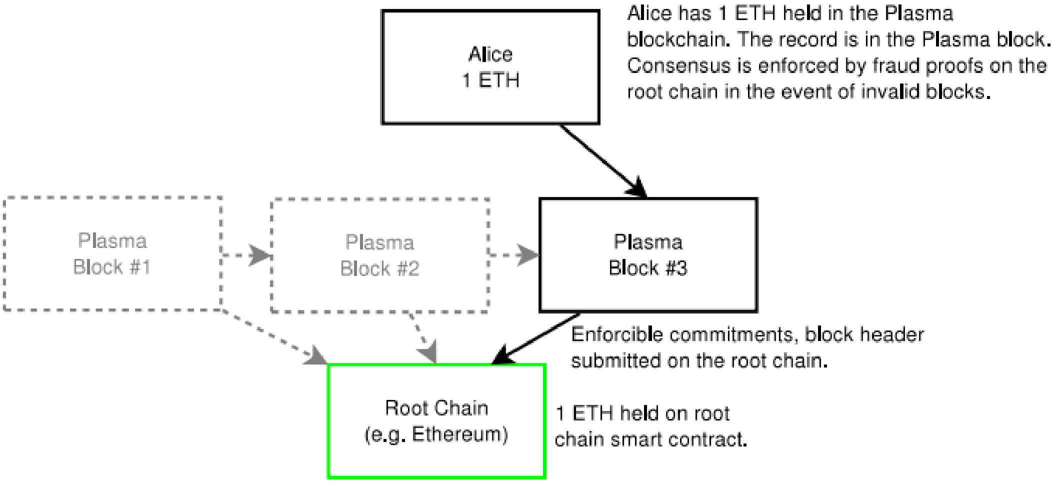
Alice
1 ETH
Alice has 1 ETH held in the Plasma blockchain. The record is in the Plasma block. Consensus is enforced by fraud proofs on the root chain in the event of invalid blocks.
Plasma Block #1
Plasma Block #2
Plasma Block #3
Enforcible commitments, block header submitted on the root chain.
Root Chain (e.g. Ethereum)
1 ETH held on root chain smart contract.

Figure 3 - Demonstration of Plasma implementation (Poon & Buterin, 2017)

All this is to say, a Plasma blockchain like Polygon will have much lower gas fees as state updates are validated in bulk by the root network (in this case, Ethereum). Secondly, transaction bandwidth will be considerably higher as minimal data is being passed down to the overloaded Ethereum network. Internal testing of Polygon's main network achieved ~7000 TPS.

Polygon can be thought of as a second-layer scaling platform to enable secure off-chain smart contracts and transactions that will still eventually be validated by a reliable rootchain. Thus, providing the speed of a mid-tier centralized network with the security of Ethereum's vast network of validators. All while providing complete interop available with the Ethereum network i.e., 1 ETH on Ethereum is always 1 ETH on Polygon – the same token can be traded between the two blockchains.

This also has the disadvantage of any vulnerability on Ethereum will spill over into Polygon and Polygon's gas token (MATIC) will always be dependent on the price of ETH.

C. Avalanche

Avalanche is a high-performance, scalable, customizable, and secure blockchain platform (Sekniqi et al., 2020) built from the ground up, unlike Matic. It introduces a new consensus protocol called Snow, which is part of a family of leaderless Byzantine fault tolerance protocols (Rocket et al., 2019). Touting ~3400 TPS under heavy load with ~1.35 second transaction confirmation times - it is the current fastest standalone smart contract platform available for use by developers today. This new consensus protocol is also very secure and resilient to a 51% attack.

The avalanche network consists of three chains: X, P and C. For *exchanging*, building *platforms* and deploying/running *contracts* respectively. In simple terms this means exchanges aren't bogged down by slow smart contracts and vice-versa as each chain is validated by a subnet of validators which all have a stake in the primary network. The C-Chain (Contract chain) is built on the Ethereum virtual machine meaning most Ethereum smart contracts will work effortlessly.

These numerous benefits make Avalanche a strong contender for developers when considering what platform to build dApps on as unlike other smart-contract platforms it does not sacrifice decentralization for speed.

However, Avalanche lacks support for Chainlink VRF (Verifiable random function) which, as the name implies, is quite essential to building chance-based games in a trustless network.

D. Chainlink VRF

ChainLink VRF is a provably fair and verifiable source of randomness designed for use in smart contracts. This was previously impossible to do on-chain while being tamperproof as older methods like using block hash or timestamp to seed a pseudo-random number generator are vulnerable to attacks from miners via withholding mined blocks to change the outcome of a game. Every random number provided by the ChainLink oracle comes with a cryptographic proof to verify its randomness and this proof is verified on-chain before the number is consumed by the smart contract. This technology is integral to building any on-chain game that relies on non-deterministic behavior i.e., the roll of a die, or the spin of a roulette wheel. To call the VRF the contract must pay some LINK token depending on which chain it was called from. On the Ethereum network it costs 2 LINK (an Ethereum-based token to cover gas fees/ChainLink's running cost) per VRF call (~60 USD as of 30/10/2021) whereas on Polygon it costs 0.0001 LINK (0.00301 USD as of 30/10/2021).

III. Solution

A. *Project Outline*

Traditionally, the only thing stopping a casino (online or physical) from cheating the player is its own reputation, as casinos that cheat will lose business. Online casinos can conceal this cheating as their games run on centralized servers where the user will only ever see the outcome of the game and not how it was calculated. This is not the case with a blockchain-based platform thus giving even a new platform instant-legitimacy and creating a viable alternative to much disliked and distrusted giants of the iGaming industry.

This is where our project begins, the industry currently suffers from distrust due to centralization of the games themselves, with 45.5% of users complaining about 'unfair software', 59.1% complaining about not being paid properly and, 25.8% 'not being paid at all' (Gainsbury et al., 2013, p. 241). A blockchain-based iGaming platform solves these issues completely if the games' smart contracts are built to be robust, secure, and understandable and the blockchain they are based on is reliable and secure.

In this work, we build a platform called GlassCasino to play online multiplayer casino games that run entirely on-chain to reap the benefits stated earlier. The platform consists of three parts: on-chain smart contracts containing game state and transactions to modify game state, an owner of these contracts that deploys games and manages game flow e.g., spinning a roulette wheel, and a front-end web app to allow users to interact with the games.

TODO – insert app design diagram here from notes and reference

	Ethereum (ETH)	Avalanche (AVAX)	Polygon (MATIC)
Market cap (USD)	**$511.7b**	$13.8b	$12.95b
Average TPS	14	4500+	**~7000**
Gas price	High	**Lowest**	Low
ChainLink VRF Support?	**Yes – very expensive**	None	**Yes**
ETH interop	**Native**	None	**Excellent with Polygon Bridge**

Figure 4 – Comparison of prospective blockchains for our platform (as of 30/10/2021)

The blockchain of choice for our smart contracts is the Polygon network as discussed in Section II. Figure 4 shows that with high TPS, low gas fees, 1:1 inter-op with Ethereum and Chainlink VRF support it makes no compromises feature-wise for a high-throughput dApp.

To build the final frontend app we use Vue.js due to prior experience building responsive, modular web apps with the framework previously. However, the frontend will need some wallet integration and a method of sending transactions to our smart contracts. For this I selected the ethers.js library due to its low bundle size as, ideally, the dApp should be accessible even with poor internet connection.

Lastly, to build the controller server or *The House* we use Node.js, again, with ethers.js for simplicity and to share code across both domains. This Node app will be deployed to a DigitalOcean cloud node and contain the house's wallet keys to top up LINK.

B. *Project Goals*

Basic	Intermediate	Advanced
- Learn Solidity - Write roulette smart contract using naïve randomness - Iteratively test roulette smart contract (winnings allocation, exploits etc.) - Deploy contract to Polygon testnet - Learn ethers.js - Build simple vanilla web app for user interaction - Write node app to control game flow and run locally	- Integrate ChainLink VRF to roulette smart contract - Advanced, styled web app built in Vue.js for roulette - Full user wallet integration i.e., MetaMask - Live deployment of whole system with web app on a public-facing domain - Providing links to easily view cryptographic proofs when a game ends	- Write more game smart contracts - Extend web app to allow users to access multiple games - Integrate polygon bridge to allow users to exchange assets from Ethereum to Polygon - User login system to allow for profiles (with managed "serverless" infrastructure) - Customizable user profiles (username, proven NFT user icons)

References

Buterin, V. (2014). *A next-generation smart contract and decentralized application platform*.

Dowling, M. (2021). Fertile LAND: Pricing non-fungible tokens. *Finance Research Letters*. https://doi.org/10.1016/j.frl.2021.102096

Gainsbury, S., Parke, J., & Suhonen, N. (2013). Consumer attitudes towards Internet gambling: Perceptions of responsible gambling policies, consumer protection, and regulation of online gambling sites. *Computers in Human Behavior*, *29*(1). https://doi.org/10.1016/j.chb.2012.08.010

Kanani, J., Arjun, A., Nailwal, S., & Bjelic, M. (2021). *Polygon Lightpaper*.

Li, J., Li, N., Peng, J., Cui, H., & Wu, Z. (2019). Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies. *Energy*, *168*. https://doi.org/10.1016/j.energy.2018.11.046

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*.

Nguyen, C. T., Hoang, D. T., Nguyen, D. N., Niyato, D., Nguyen, H. T., & Dutkiewicz, E. (2019). Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2925010

Poon, J., & Buterin, V. (2017). *Plasma: Scalable Autonomous Smart Contracts*.

Rocket, T., Yin, M., Sekniqi, K., van Renesse, R., & Sirer, E. G. (2019). *Scalable and Probabilistic Leaderless BFT Consensus through Metastability*. http://arxiv.org/abs/1906.08936

Sekniqi, K., Laine, D., Buttolph, S., & Sirer, E. G. (2020). *Avalanche Platform Whitepaper*. https://assets.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16_Avalanche%20Platform%20Whitepaper.pdf