Decentralizing Video Game Matches on the Blockchain

By Blake Copeland

Twitter: @wbcopeland
LinkedIn: wbcopeland

December 2021

There are many types of blockchain technologies that exist today. When it comes to the realm of blockchain and gaming, there are many new games out there that are implementing blockchains. But when you look at the blockchain technologies from 10,000 ft, it’s obvious they work best when doing transactions between two parties.

Online multiplier games are incredibly commonplace. Consequently, for blockchain and gaming to really have a happy marriage, a new technical model is needed. Thus, this paper is about a new way of minting coin to players, which is achieved through spending time playing a game.

In our use case, players can play any variant of our multiplayer playlists in “two flavors.” They can play a multiplayer match just like any other game, or they can choose to play a multiplayer match “On the blockchain.”

Some context: I’m a Lead Software Engineer at a gaming firm called GAD Studios and we are currently finishing development on a game that uses this novel blockchain technology. I refer to our game in this paper as “our use case.”

Proof of Play

Blockchains can be mined in a few different ways. The two most common are Proof of Work (*PoW*) and Proof of Stake (*PoS*). In our blockchain architecture we are prescribing a new method for minting coins. This is done through chaining group transactions on the block and rewarding the parties involved with minted coin. We call this novel approach *Proof of Play (PoP)* because in our use case the group transactions are multiplayer matches in a video game. But any scenario that has a start point, an end point, and requires smaller group consensus can work on this blockchain architecture.

In more direct terms, the blockchain rewards parties for participating in a group task. In our use case, this means we reward players for completing a multiplayer match.

When a group of players confirms that they have played a game match together, the blockchain verifies and then mints coin to the players.

This is the spirit of *Proof of Play:* to reward players (or anyone else associated in the process) who spend time playing our game.

The Economics

In traditional blockchains there are economic incentives for miners to play fair. For example, *PoW* requires both time and energy to mine a block. If someone tries to submit a fraudulent block, the rest of the network will reject it, and actual money would be lost in the form of the energy bill for the failed mining effort. In *PoS*, users must stake existing coin in order to mine. If they try and approve fraudulent transactions, the network will remove their stake, and actual money is lost in the form of whatever the dollar value of the stake coin was.

PoP actually is a combination of some aspects of both *PoS* and *PoW*. In *PoP*, players spend real time playing a multiplayer match. But players can also stake coin on a match itself.

Players that finish games are rewarded with newly minted coin, with winners getting additional coin.

It's also worth noting that the energy impact of *PoP* is conceptually the same as *PoS*. A player is already spending energy playing a particular game, adding *PoP* on top of that does not add any power cost, and a *PoP* blockchain existing outside of the game only uses a much power as the process that is used to mine the actual blocks. Which leads to this disclaimer:

This blockchain architecture still requires one of the traditional methods of mining to put blocks on the chain. In our use case, we use *PoS*.

The Start Game Transaction

In most games, when a group of players want to start a match in a game they enter a connected lobby that allows data to be transmitted between them before the match starts. This is done in order to customize rules or settings of a match, to pick team names and colors, or other identifying features.

In our model, when all the players are ready to start the match and are happy with the settings, they "lock in." This simply means each player takes the lobby and team roster data and signs it. Then they share this signature with all the other players in the lobby. Likewise, all the other parties do the same and locally validate each signature.

The lobby generates a random address, called the *Output Address*. This is where the match data will be stored once the game is finished, as well as any other data that players want to write there (such as reporting another player or other comments on the match).

The lobby also contains a randomly selected *Officiator*. This is a *Verified Address* who, similarly to *PoS*, has staked coin to participate in the *Officiator* role. The *Officiator* connects to the game with the players and keeps a record of scores and metrics using an offline side chain. There are a few reasons this is done, which will be explained later.

If all the signatures are valid and exchanged between players, then they submit the data to the blockchain and start the match. This is called a *Start Game* transaction and is a way to have a group of parties sign and agree on a set of starting parameters, as well as who is involved in the match, what team they are on, and other starting criteria.

Start Game transactions can also require stakes in order for players to be able participate. These stakes are always given back to the player provided they "play fair." One foreseeable use of staking is in competitive matchmaking. Developers could also program the blockchain to use this stake in other ways, such as going to the winners of the match. This felt to poker-like for our use case, so we simply use it as insurance for good behavior.

The Game Side Chain

When the game match is started, an offline side-chain is shared between the players. In our use case, a block on the side chain is created every 5 minutes and keeps track of the scores. But instead of having the players sign off on each side block, we have the O*fficiator* do it. There are a few reasons we chose to sign the side-chain this way.

Let's look at the scenario where there is no *Officiator* and players sign each side-chain block. One issue is that if even one player disagrees on the scores of a given block, say halfway through a match, the game must come to a halt until the disagreement has been sorted or all the players forfeit the game. This also leads to another issue.

One of the inherent issues in blockchain technology is that it is susceptible to at 51% attack. When the network has millions of people, this becomes very difficult and impractical to achieve. When the blockchain contains just a few people, say in a game match, the issue becomes much more real. The majority of players could "bully" the minority into signing off on a fraudulent score, or a single player could hold all other players hostage by not signing on a given block. Players in these scenarios risk losing the potential coin that comes with mining in *PoP* and might succumb to signing the fraudulent scores of the majority.

This is where the *Officiator* comes in. They, as an unknown and un-biased 3rd party, can keep track of all the match metrics, similar to a referee or scorekeeper. If a majority of players in a match are trying to push fraudulent scores on a minority, the officiator keeps them from being able to do that, and can actually note this behavior.

The End Game Transaction

When the match is over, the *Officiator* posts the side-chain to the blockchain at the *Output Address* as an *End Game* transaction with his signature. Since the *Start Game* block contains the *Officiator's* address and is signed by all the players, the *Officiator* has the authority to do this.

The blockchain then looks at the game results and mints coin to the players that participated. The amount of coins minted is described below in the *Reward Ratio* section.

In the case of a bad or malicious *Officiator*, players can sign a dissent at the *Output Address*. These dissent signatures are taken very seriously and players have a very limited amount they can give out. It is a dox on the *Officiator* and his reporting, not on other players in that match or the outcome of the match.

The math is a little more complex than this, but basically if an *Officiator* receives three games where 80% of the players are challenging the results of the match, they lose their stake and can no longer officiate. This ruleset can also be tweaked by the developer.

Concession Signatures (The Disconnect Case)

In the case where a player no longer wishes to finish the game, they can sign a *Concession Signature* at the game's *Output Address*. This forfeits their right to win the game which in turn also forfeits their right to earn coin from the game match.

This allows the player to enter a new *Start Game* transaction before the previous game's *End Game* transaction has been posted to the blockchain.

Reward Ratio

In terms of how *PoP* distributes coin, the developer has the discretion to determine how rewards are given out to the players. Below is the point breakdown in our use case. These are the amount of coins minted to each player in the match per side-chain block.

	Non-Staked Match	Staked Match
Winners	2.5 Coins	5 Coins
Losers	1 Coins	2 Coins

If you reward players a static amount regardless of the amount of time played, they will be incentivized to finish the match as quickly as possible. With each side-block being worth 5 coins in a staked match, that's 15 coins for the winner. An hour long game would be 60 coins for the winner.

Losing players are still minted coin since they participated in the "mining process."

Players who disconnect or sign a concession signature forfeit their entire position and earn no coin.

In terms of our use case and perception to the player, we say all players mine at the rate of the loser, and receive at 2x multiplier on that coin if they win.

Verified Wallets

Inherent in every blockchain are addresses to which coin can be sent. In our use case, players create a wallet address to participate in *PoP*. However, there is nothing preventing players from creating as many wallet addresses as they want. This can lead to players being in multiple games at the same time with no one the wiser. Or players could even have multiple wallet addresses they possess enter into matches together, effectively playing against themselves.

These problems are solved by having two class of addresses:

<u>Normal Address</u>
- Can send and receive coin
- Can participate and stake in mining blocks via *PoS*

<u>Verified Address</u>
- Can do the same things as a normal address
- Can participate in *PoP*

A *Verified Address* is tied to a player's unique gamer tag. Only *Verified Addresses* are allowed to participate in *PoP*.

For an address to be verified, a player must first post their gamertag, an auth token proving their identity, and the wallet address they want to tie their identity to. All this gets posted to the blockchain. Other users in the ecosystem can check their auth token to verify that the user is actually who they claim to be. Users are incentivized to check as they earn 0.1 coin by "voting" yes or no on confirming the player's identity. After 15 blocks (30 minutes) of voting, the network looks at all the votes made on an identity. If there were over 1000 votes, and the ratio of yes/no is 95%, then the network sees the address as verified.

While some auth tokens are permanent, most that are provided (in our use case from game providers such as Steam or Epic Games) expire after an hour. This is why the network has a 30 minute time slot for verification. We also want to "spool up" time for players to be as minimal as possible.

If a player fails to get their identity proven, whether because they lacked the minimum number of signatures or they didn't get a high enough pass ratio, they can try to get their identity verified endless times.

Already verified players can go through the same process and give a different wallet address to associate their gamertag with a new address.

Rules

There are a few rules that our new blockchain incorporates in order to prevent abusive generating of coin in *PoP*. Some of these have been stated above. They are re-stated here for clarity:

- Only *Verified Wallets* can participate in *PoP*. This includes being a player in a match or an *Officiator*.

- There is a minimum time between *Start Game* and *End Game* transactions. This minimum can be set by the developer. In our use case, and because we are a strategy game, we set the minimum time to 15 minutes. That is a minimum of 8 of blocks on the blockchain, and a minimum of 3 blocks on a side-chain.

- An *Officiator* can only officiate one match at a time, and he cannot officiate his next match until he posts the previous match's *End Game* transaction.

- Players can only be in one game at a time. However, players can start a new match before the *Officiator* has posted the *End Game* transaction to the blockchain. This could be due to the player signing a *Concession Signature*, or an *Officiator* posting the *End Game* transaction late or not posting it at all. We do not want players to be delayed or inhibited from playing new matches, so we allow them to join another *Start Game* transaction. However, when the *End Game* transaction is posted by the Officiator, if the timestamp of when the game ended is after a new *Start Game* for the player (and the player did not post a *Concession Signature* on the previous match), then the network knows that the player tried to enter a new match while still in an old one, and it forfeits the coined earned in the previous match. This keeps players from gaming the system and being in multiple matches at the same time, while not being bottlenecked by the *Officiator's* posting time.

Caveats

Because of the nature of *PoP*, more than one address can be minted coin in each block. Thus the number of users playing the game concurrently directly influences the amount of coins that get minted into circulation. This is not necessarily a weakness, but it does have the potential to cause slight inflation when there are many matches concurrently finishing a the same time (aka high activity on the network). Further study is needed by an economist to really understand if this is really an issue when deployed at scale.

In the case where an *Officiator* never posts an *End Game* transaction, if players had stakes in the match they do not get them back. This should be an incredibly rare occurrence as *Officiators* who do not post cannot continue to officiate.