Horizontal Scalability of Blockchain Games Using the GSP Model

Daniel Kraft

Abstract In recent years, blockchain has emerged as an interesting technology not just for digital currency but also more complex decentralised applications including computer games. Smart-contract platforms like Ethereum are a popular choice for developers of blockchain games. However, due to their architecture they are unsuitable for complex on-chain computations, as those would lead to unfeasibly high gas costs. Consequently, most blockchain games today are either trivial in terms of game mechanics or run on a centralised server, defeating the original purpose of using a blockchain. In this paper, we present an alternate architecture based on game-state processors (GSPs). This model decouples computations from the core blockchain layer (miners and full nodes), effectively sharding computation and storage of state between the different applications and games. With this, a blockchain platform can be scaled horizontally to support many and highly-complex games with fully decentralised, on-chain computations.

Keywords Blockchain gaming · Decentralised gaming · Play-to-earn · Game-state processor · Scalability

1 Introduction

Only a few years after Nakamoto's invention of blockchain technology [21] in 2008 his idea was being applied to things beyond just a currency. The first such application was Namecoin [14], but already soon after that a fully decentralised computer game based on blockchain was described [8] and launched with Huntercoin [10]. Apart from trivial gambling (e.g. SatoshiDice [4]), this was the first blockchain game ever

D. Kraft (✉)
Xaya Team, Autonomous Worlds Ltd, Zurich, Switzerland
e-mail: daniel@xaya.io

A. Dingli et al. (eds.), *Disruptive Technologies in Media, Arts and Design*, Lecture Notes in Networks and Systems 382, https://doi.org/10.1007/978-3-030-93780-5_7

created, and it also pioneered the concept of *human mining*. Soon after, a second decentralised game (Motocoin [30]) introduced another interesting concept, *proof-of-play*. Based on the original ethos and cypherpunk spirit of the Bitcoin community at the time, those early projects were indeed fully decentralised like Bitcoin itself.

Later on, smart-contract platforms like Ethereum [1] made it much easier to create blockchain-based games. During the Bitcoin bullrun of 2017, CryptoKitties was such a smart-contract game that gained widespread attention [27], and made larger audiences aware of blockchain as a tool for computer games. Unfortunately, the monolithic design of smart-contract platforms makes them unsuitable for decentralised and complex game logic—this means that most blockchain games to date are either almost trivial or using the blockchain mainly to store game assets, while the game itself still runs on centralised infrastructure similar to traditional, non-blockchain online games. CryptoKitties itself arguably falls in both categories at the same time; the main interaction with the game is breeding digital cats, which can hardly be seen as a particularly deep or complex game concept. But even with this simple design, a crucial part of the game (namely the pictures associated to cats, i.e. the phenotype derived from the on-chain genes) is not decentralised and instead hosted on the game studio's servers. Enjin is even an entire platform built around the concept of centralised games that use blockchain-based assets [26]. Even for centralised games, the use of blockchain-based assets in the game certainly has many benefits. For instance, it makes it much easier for users to hold, transfer and trade their assets, even if the game developer was not able or willing to build a dedicated in-game marketplace. For Ethereum-based assets, OpenSea [11] has developed into a de-facto standard, universal marketplace where such assets can be traded.

However, game assets always have to be considered in the context of the games themselves. Assets of a centralised game may be owned by the users in a trustless manner on a public, decentralised blockchain (like Ethereum); but if the developers of the game decide to shut down the game servers, ban a user or inflate the supply of a supposedly rare item, then items will become worthless even if the game developers may not be able to take them directly off a user. Especially with smart contracts, it is nowadays quite common for the developers to give themselves special power to upgrade the contract or otherwise control the system at will; such backdoors have in the past been used to shut down supposedly decentralised applications such as the IOTA network [7] or the tBTC application [12].

Therefore, it is important to distinguish between *true ownership* of assets in fully decentralised games and just *non-custodial ownership* of assets in centralised games. The latter is only an evolution, which makes asset-ownership and trading easier; the former is a true revolution, which shows the real disruptive potential of blockchain technology. This can be compared to how Bitcoin enabled the emergence of value without any trusted, central instance for the first time in history, while "permissioned" blockchains are just a handy tool to optimise some processes [13]. For a more in-depth discussion of the benefits of fully decentralised gaming, see [28], specifically the sections about CryptoKitties and hybrid architectures.

Xaya is a blockchain, platform and open-source project that strives to enable fully decentralised, on-chain games that are still able to feature highly complex game logic and computations. In contrast to games built solely on smart-contract platforms, this is enabled by decoupling the core blockchain from the potentially heavy computations and states of individual games. This is possible through an architecture based on *game-state processors* (GSPs), which we will describe in this paper.

Note that while we work on this topic ourselves to build decentralised computer games, the term *game* in the following can just as well be understood to mean any type of decentralised application (dApp), not necessarily restricted to entertainment purposes.

2 Scalability for Blockchain Games

At least since the blocksize debate in Bitcoin [5], it is clear that scalability in general is a hot topic and potential issue for any blockchain-based system. In fact, blockchain as a data structure is very inefficient in terms of scalability; only because it enables otherwise-impossible full decentralisation should it be used at all [31].

For "simple" applications of blockchain technology like Bitcoin itself, the main scalability issue is the data that needs to be put into blocks, transmitted through the network, and stored forever in the blockchain. But for more complex applications (e.g. smart contracts on Ethereum) and especially non-trivial games, another important bottleneck is the computing power needed to execute the contract or game logic. In many cases of complex games, this may be the limiting factor rather than the volume of raw data. Therefore, it makes sense to distinguish between two slightly different types of scalability in this context.

2.1 Vertical Scalability

Even a monolithic design (where a single game runs on its own blockchain) can easily run into scalability limits. For instance, Fig. 1 shows the size of the Huntercoin blockchain over the first 18 months after its launch. Especially right at the beginning, the growth (i.e. transaction volume) was tremendous, considering also that this was back in 2014 where even Bitcoin itself was not yet hitting the block-size limit. It became clear that this growth was not sustainable, and a couple of network-wide changes were implemented to address the issue (some of which are marked with vertical lines in the chart). As can be seen, they took effect and slowed down blockchain growth. However, this solution is far from perfect, as it came with restrictions and severe changes to how Huntercoin could be used and played. Also, since these changes were tailored specifically to Huntercoin, they cannot be applied directly to other games.

Huntercoin was created as an experiment to see how blockchains can be used for MMO-type games. As it became clear that some scalability solution is needed, we developed *game channels* [19] as an off-chain scaling technology. Inspired by payment channels [23] and developed independently from state channels [20] at around the same time, they allow complex game interactions (and not just payments) to be done off-chain while keeping the whole system fully decentralised and trustless. This enables massive scalability *within a single game* as well as *near-realtime blockchain-gaming experiences*. In the following, however, we want to focus on a different aspect of scalability, and refer to [19] for more details about game channels and vertical scalability.

2.2 Horizontal Scalability

If a blockchain platform is not specific to a single game and instead built to run many games and applications in parallel, then the limits imposed on each individual application have to be even tighter. For instance, Ethereum as the leading smart-contract platform uses *gas fees* to charge each user according to the computations their transactions require in smart contracts. As a consequence, it becomes prohibitively expensive to execute anything beyond the most trivial game logic. For instance, it is even impossible to directly check whether or not a position in chess is check mate

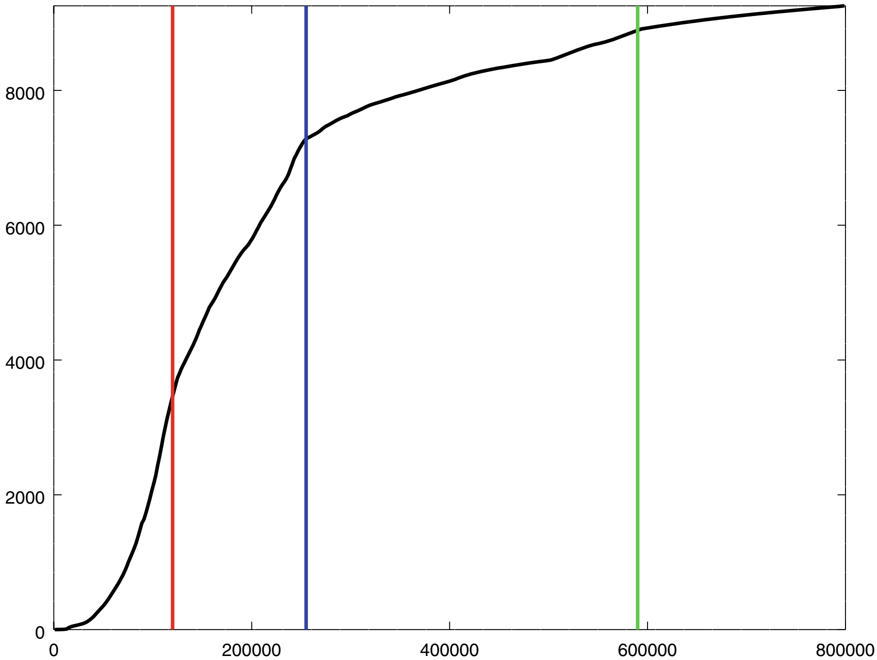

Fig. 1 Size of the Huntercoin blockchain in MiB during the first 800,000 blocks (about 18 months) after launch. The vertical lines indicate changes to the network rules

[15]; and while it is certainly challenging for a human to play well in chess, the rules themselves are much simpler than e.g. the computations required for a real MMO strategy game. Similarly, CryptoKitties famously clogged the Ethereum network due to its massive traffic after launch [16].

Strictly speaking, both of these are of course examples of single games running into scalability limits, and thus could be considered vertical scalability issues. However, a decentralised network is certainly able to verify check-mate positions. But if the network as a whole must be able to execute the computations of many other smart contracts as well, then the limits have to be a lot more restrictive, which is exactly what happens with Ethereum.

This is especially the case if the cost in computation (and perhaps state storage) outweighs the raw transaction size stored in the blockchain, as is the case for the chess example and many other potential gaming applications. For these situations, the monolithic approach of smart-contract platforms is not very cost effective or even outright prohibitive. To enable *horizontal scalability* of a blockchain to *many and complex applications*, a different model is required, which we will discuss below.

3 The Role of Miners

It is a common misconception, especially in mainstream media, that "the job of a Bitcoin miner is to validate transactions, and in exchange get a portion of the reward in BTC for doing it" [24]. Of course, Bitcoin miners typically also validate transactions before they put them into blocks; if they did not do this, they would risk losing money for mining invalid blocks. (In fact, in 2015 an incident related to "SPV mining" actually caused miners who did not properly validate blocks to lose block rewards in this way [25].) However, from the network's point of view, *all full nodes* (which are typically a superset of the miners) validate transactions.

The real function of mining is to achieve consensus with proof-of-work (PoW). More specifically, through mining the network establishes a *time order* in which transactions occurred, thus enabling to determine which of a possible set of conflicting transactions is the "correct" one, solving the *double-spending problem*. This is described as a "timestamp server" already in the original Bitcoin whitepaper [21].

4 Game-State Processors

Now all necessary preliminaries are in place, and we can discuss how the GSP architecture allows to decouple the potentially expensive computations for some game from the core blockchain network, and thus greatly improve the network's horizontal scalability.

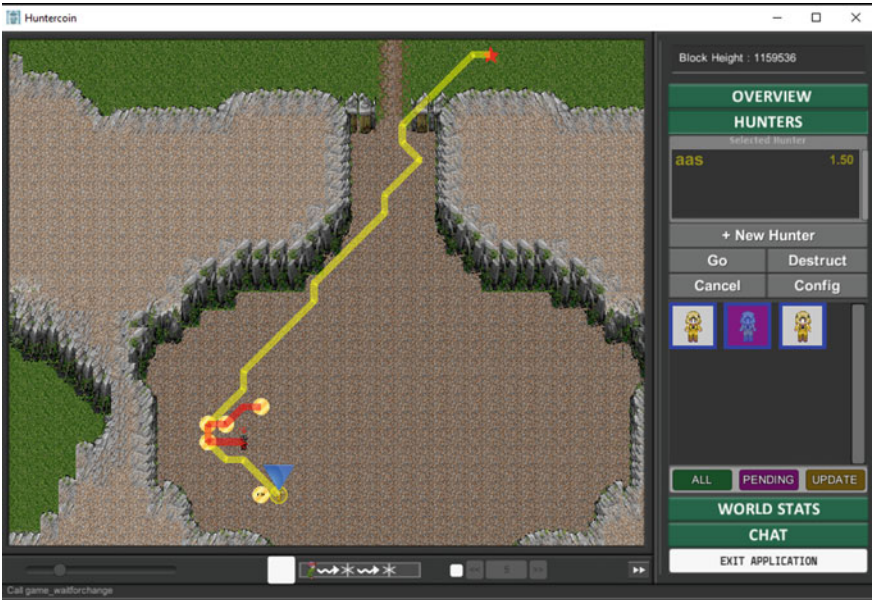

Fig. 2 Screenshot of Huntercoin, showing some of the game's basic concepts

4.1 Game States

Blockchains in general can be interpreted as a *state-transition system* [1]. Associated to each block is a current state. It can be computed deterministically from the state associated to the previous block, the transactions inside the block as "update data", and the network's hardcoded rules (the state-transition function). In the case of Bitcoin, the state is the current ledger, represented by the UTXO set. The updates applied are transactions that spend some inputs (remove them from the UTXO set) and create new outputs (that get added to the UTXO set). Other blockchains typically have more complex states and state-transition functions, e.g. Namecoin or Ethereum.

Figure 2 shows a screenshot of Huntercoin. In the game, players can move around with their avatars ("hunters") on a map. They are able to pick up coins placed on the map, and then need to bring them to a banking location while others may try to attack them and steal the coins. Once banked, the coins appear in the user's wallet, and act like an ordinary cryptocurrency. Consequently, the *game state* of Huntercoin contains like Bitcoin's the ledger of banked coins in form of an UTXO set; however, it also contains the state of the game world, i.e. the positions of all hunters on the map, their waypoints and all coins that are not yet banked. This state can be affected by each player by sending their moves as transactions in the blockchain. These moves are the creation of new hunters, changes to their waypoints (thus controlling movement) and an "attack" action.

It is important to understand that the state itself is never stored in the blockchain, only moves (player actions) are. The state is computed independently from those moves by each network node for themselves. Since mining ensures consensus about the set and order of moves that have occurred and each node is using the same game rules (state-transition function), everyone is guaranteed to obtain the same state as well.

4.2 Decoupling States from the Core Blockchain

Let us now briefly summarise what we have learned so far: It is not necessary to store the actual game state on chain; instead, it is enough to store the *moves* on the blockchain, and then have every node compute the game state independently. The only thing that is required to make this work and ensure everyone arrives at the same state is to ensure that they are using the same state-transition function (i.e. are using the same consensus rules and follow the same hard forks), and that PoW mining yields a network consensus on the sequence of moves that have taken place. But on the other hand, it is *not* necessary for miners to actually interpret what those moves mean in the context of a particular game, and thus it is also not necessary for miners to know any current game state (or even the rules of individual games).

This insight allows us to *decouple* computation of game states from the underlying core network; thus, we can split the architecture of such a platform into two layers:

1. The core layer consists of the P2P network of nodes, the actual blockchain, and PoW mining of transactions. It also needs a cryptocurrency to pay for transaction fees and in this way protect against blockchain spam (by having a fee market). This layer allows arbitrary data for moves to be put on chain, without validating or interpreting them in any game-specific way. In the case of Xaya, the core layer also manages a concept of "user accounts" (based on Namecoin names), and ensures through digital signatures that only the owner of an account can send moves for it.
2. Each game running on the platform provides a *game-state processor* (GSP): This is a software process that runs alongside the core blockchain node, reads the sequence of moves from the blockchain, and uses the game-specific state-transition function to compute the current state. This can then e.g. be used to show the state to the user in a game frontend, or for other purposes (perhaps as part of an exchange that allows trading of assets from one of the games).

The crucial observation here is that miners and full nodes are only required to handle the first part of this. All heavy computation for a particular game (as well as storing that game's state) is done by the GSP instead. And the GSP for some game only needs to be run by users who need to know the state of that game, i.e. who are actively interested in the game (e.g. playing it). This allows a single blockchain platform to support hundreds or thousands of complex games in parallel, because each user is only required to do computations for the games they are using themselves.

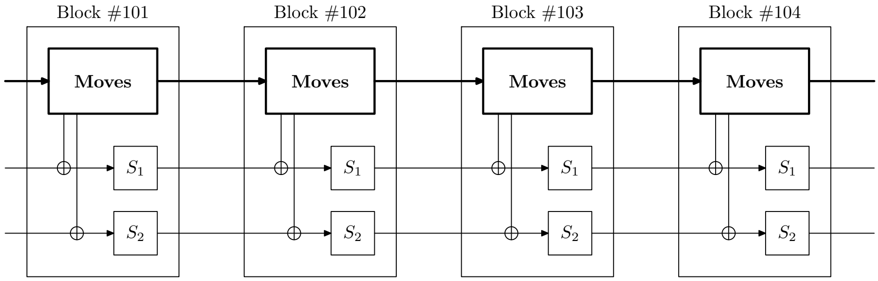

Fig. 3 Schematic diagram of two different games (GSPs) running on top of a single core blockchain (bold)

In some sense, this can be seen as (partial) *sharding*, namely sharding of computation power and storage (but not of the raw blockchain data).

This model is in complete contrast to a monolithic platform like Ethereum, where every full node (and thus miner) always has to perform all computations of the entire network. Hence, a platform based on GSPs is highly scalable horizontally, while a monolithic platform is not.

This split into a core layer and multiple GSPs on top of it is sketched in Fig. 3: The core blockchain corresponds to the bold parts in the diagram, i.e. blocks that contain just some moves. Those moves are then used by two GSPs running on the platform to derive the states S_1 and S_2 associated to each of the blocks. The $\oplus$ indicates the state-transition function (game logic) for each game, which derives the new state from the previous state and the current block's moves.

Note that each GSP is purely local, and does not concern the core network at all. This has an interesting consequence: Even if at some point in time no one is playing a particular game at all, the game still carries on (it is unstoppable), although there does not exist any instance of the current state at that time. If at a later time someone starts playing again and runs the GSP, they will compute the game state for the entire history (including those blocks when no GSPs were running) and "re-materialise" the game world. This also means that unlike platforms that have a "one blockchain per game" model (e.g. Planetarium [6]), there is no need to incentivise and worry about nodes running GSPs all the time.

Let us also remark that the decoupling of states from a core blockchain (which is used to achieve consensus) is not a completely new idea. A couple of overlay protocols have been built on top of Bitcoin in the past, e.g. Mastercoin [29] (now called Omni) or Counterparty [22]. To the best of our knowledge, however, all of these platforms were built to place a specific use case on top of Bitcoin and take advantage of the Bitcoin network's security. In our proposal, we use the overlay architecture for a different purpose, namely for horizontal scaling of a custom blockchain. In a sense, this can be compared rather to *rollups*, which have become the scaling solution of choice in Ethereum [3]: Each GSP can be thought of as its own rollup, except radically simplified and not relying on any complex security assumptions and mechanisms to bridge state back to the main chain.

5 Drawbacks of Game-State Decoupling

We have seen that decoupling the game state from the core network leads to drastic improvements in scalability. On the other hand, it also means that the core network does not and can not (by design) fully verify game-specific operations, which imposes two main restrictions on games built with this architecture.

5.1 SPV Security

SPV (simplified payment verification; see [21]) is a method to run cryptocurrency wallets without the need to download and verify the entire blockchain. It is used by popular tools such as Electrum or mobile Bitcoin wallets. SPV wallets assume that all transactions inside the longest (most work) chain of blocks are valid without verifying them for themselves. This works as long as the majority of mining power is honest, since miners have a strong incentive to only include valid transactions into blocks (as discussed previously).

Strictly speaking, this of course remains true for GSP-based blockchains. However, for them SPV security obviously only applies to things validated by the core network (e.g. CHI transactions and name operations in the case of Xaya). Since the game-specific interpretation of move data is left to GSPs, the mere presence of a particular move in the longest chain does not automatically imply that the move has a particular effect on the game state or is even valid for a specific game.

For example, Alice could send a move that indicates "send a Vorpal sword to Bob", and it could be confirmed and included into the longest chain of blocks just fine. However, it may still be invalid with respect to the game, e.g. because Alice does not own such a sword. Hence, Bob must not rely on the sword being transferred to him just because Alice's transaction has been confirmed. Instead, he has to check the game state himself by running a GSP.

However, it is still possible to build mobile and light clients for GSP-based games (just not with the SPV technique alone). Depending on the desired balance between decentralisation and convenience, users may simply trust a remote GSP that is run by the game developer, especially if they do not yet have a lot of money at stake inside the game. If their interest and stake in the game increases, they can at any later time decide to run their own GSP for more security and reliability instead. The game developer (or a third party) can even provide ready-made VM or container images for GSPs, so that also non-technical users are easily able to run their own GSP on a cheap VPS and still play on mobile devices.

It is even possible to design a system that allows for a balance between decentralisation and convenience that is strictly between the two extremes of "trusting a single remote GSP" and "running your own GSP". For instance, users could query multiple remote GSPs and only accept a current game state if all of them agree. Or the operators of remote GSPs could sign the states they deliver, so that a wrong answer can be proven to the network; with this, it is possible to build a "GSP-as-a-service" system based on deposits and penalties similar to some of the more recent proof-of-stake consensus algorithms [2] or optimistic rollups like Arbitrum [17].

5.2 No Control Over the Coin

The second restriction implied by state decoupling is that GSPs and specific games cannot control the native on-chain currency directly; this is in direct contrast to smart contracts on monolithic platforms (e.g. contracts on Ethereum have their own ETH balance). As coin transactions are part of the core network layer and thus need to be validated by all nodes, it is not possible that any coin transfers are conditional on the state of some game (which is not known by those nodes). It is, however, possible for a GSP to *observe* coin transactions that happen on the core layer and then update the game state accordingly (for instance, mint in-game assets whenever coins are paid to the game developer's address). This is a direct consequence of the unidirectional flow of information between the two layers (from the core network to GSPs, but not the other way round).

This restricts how value can flow into and out of games through the blockchain's native currency (especially for human-mining/play-to-earn games). It is, however, possible to add a game-specific currency (e.g. gold pieces, credits, gems, …) to the game state. Together with all other game assets, balances of this currency can then be fully controlled by arbitrary rules coded into the GSP. As long as this game currency can be exchanged to some fiat or other cryptocurrency, it will enable free flow of value again.

The latter, finally, is easily possible: Centralised exchanges can integrate an in-game currency just as easily as a "normal" cryptocurrency; all they have to do is run the game's GSP. Additionally, the core blockchain can be designed to facilitate atomic trades between its coin and game assets, so that trustless and decentralised exchanges can be built as well. In the case of Xaya, *atomic name updates* [18] can be used for this purpose; they are based on *atomic name trades* originally proposed for Namecoin [9].

6 Conclusion

We have seen that scalability is a major concern for blockchain projects in general, but even more so for blockchain-based games. Monolithic approaches based on smart contracts are popular for blockchain games today, but they lead to games that are either centralised (with only assets on the blockchain) or have trivial rules and shallow game play. In other words, Ethereum may not be the best choice for building a fully decentralised game.

Hence, we proposed to decouple game-logic processing from the core blockchain network. This GSP architecture improves horizontal scalability drastically and, in particular, allows to build complex and still fully decentralised games. Game-state decoupling comes not without its own trade-offs, though, as it means that the core network is not aware of game-specific states and rules. This leads to some restrictions, but we believe that they can be dealt with very well in practice using, among other techniques, atomic transactions for trading between the native coin of the core blockchain and in-game assets.

With the methods described, it is possible to bring blockchain games to the next level of their evolution, and build highly complex but still fully trustless, scalable and decentralised games and applications.

Acknowledgements The author is employed by Autonomous Worlds Ltd, the startup backing the Xaya project. This work was created through that employment.

References

1. Buterin, V.: A Next-Generation Smart Contract and Decentralized Application Platform (2013). https://ethereum.org/whitepaper/
2. Buterin, V.: A Proof of Stake Design Philosophy (2016). https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51
3. Buterin, V.: A rollup-centric Ethereum roadmap (2020). https://ethereum-magicians.org/t/a-rollup-centric-ethereum-roadmap/4698
4. Buterin, V.: The bitcoin gambling diaspora. Bitcoin Magazine (2013). https://bitcoinmagazine.com/articles/the-bitcoin-gambling-diaspora-1375548799
5. Caffyn, G.: What is the bitcoin block size debate and why does it matter? CoinDesk (2015). https://www.coindesk.com/what-is-the-bitcoin-block-size-debate-and-why-does-it-matter
6. Chung, I.: Every game this South Korean startup makes has its own blockchain. CoinDesk (2019). https://www.coindesk.com/every-game-this-south-korean-startup-makes-has-its-own-blockchain
7. Cimpanu, C.: IOTA cryptocurrency shuts down entire network after wallet hack. ZDNet (2020). https://www.zdnet.com/article/iota-cryptocurrency-shuts-down-entire-network-after-wallet-hack/
8. Colosimo, A.: Chrono Kings—a game/world within the block chain (2013). https://bitcointalk.org/index.php?topic=262599.0
9. Colosimo, A.: Fair exchange of a Namecoin name to NMC (2013). https://forum.namecoin.org/viewtopic.php?t=1316
10. Colosimo, A.: Huntercoin—Worlds First Decentralized Game/World on the Blockchain (2014). https://bitcointalk.org/index.php?topic=435170.0
11. Dale, B.: 'Ebay for cryptoKitties' raises $2 million from all-star VCs. CoinDesk (2018). https://www.coindesk.com/ebay-cryptokitties-raises-2-million-star-vcs
12. Dale, B.: Bug forces shutdown of bitcoin-backed ethereum token tBTC. CoinDesk (2020). https://www.coindesk.com/bug-forces-shutdown-of-bitcoin-backed-ethereum-token-tbtc

13. Dale, B.: Decentralized or nothing: Song duels IBM over blockchain hype at SXSW. CoinDesk (2019). https://www.coindesk.com/decentralized-or-nothing-song-slams-ibm-over-blockchain-hype-at-sxsw
14. Durham, V.: Namecoin—a distributed naming system based on bitcoin (2011). https://bitcointalk.org/index.php?topic=6017.0
15. Grau, P., et al.: Lessons learned from making a chess game for ethereum (2016). https://medium.com/@graycoding/lessons-learned-from-making-a-chess-game-for-ethereum-6917c01178b6
16. Hertig, A.: Loveable digital kittens are clogging ethereum's blockchain. CoinDesk (2017). https://www.coindesk.com/loveable-digital-kittens-clogging-ethereums-blockchain
17. Kalodner, H., Goldfeder, S., Chen, X., Weinberg, S.M., Felten, E.W.: Arbitrum: scalable, private smart contracts. In: 27th USENIX Security Symposium (USENIX Security 18), pp. 1353–1370. USENIX Association (2018). https://www.usenix.org/conference/usenixsecurity18/presentation/kalodner
18. Kraft, D.: Atomic trading (2019). https://github.com/xaya/xaya/blob/master/doc/xaya/trading.md
19. Kraft, D.: Game channels for trustless off-chain interactions in decentralized virtual worlds. Ledger **1**, 84–98. https://doi.org/10.5195/ledger.2016.15
20. McCone, R.: Ethereum lightning network and beyond (2015). http://www.arcturnus.com/ethereum-lightning-network-and-beyond/. Accessible in the WayBack Machine
21. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008). https://bitcoin.org/bitcoin.pdf
22. PhantomPhreak: Counterparty—Pioneering Peer-to-Peer Finance—Official Thread (2014). https://bitcointalk.org/index.php?topic=395761.0
23. Rapidly-adjusted (micro)payments to a pre-determined party. https://en.bitcoin.it/wiki/Contract#Example_7:_Rapidly-adjusted_.28micro.29payments_to_a_pre-determined_party
24. Raza, A.: A new bitcoin bug could cause chaos in the crypto world. Invezz (2020). https://invezz.com/news/2020/06/01/new-bitcoin-bug-found/
25. Rizzo, P.: Double spending risk remains after July 4th bitcoin fork. CoinDesk (2015). https://www.coindesk.com/double-spending-risk-bitcoin-network-fork
26. Sinclair, S.: Enjin launches game development platform on ethereum. CoinDesk (2020). https://www.coindesk.com/new-enjin-platform-aims-to-make-it-easier-for-game-developers-to-integrate-blockchain-assets
27. Takahashi, D.: CryptoKitties explained: why players have bred over a million blockchain felines. VentureBeat (2018). https://venturebeat.com/2018/10/06/cryptokitties-explained-why-players-have-bred-over-a-million-blockchain-felines/
28. Thomson, E.: The thick and thin of blockchain gaming architectures (2019). https://medium.com/@edward.thomson/the-thick-and-thin-of-blockhain-gaming-architectures-a51795156420
29. Willet, J.R.: MasterCoin: New Protocol Layer Starting from "The Exodus Address" (2013). https://bitcointalk.org/index.php?topic=265488.0
30. WilliamLie2: Motocoin: the first proof-of-play cryptocurrency (2014). https://bitcointalk.org/index.php?topic=591724.0
31. Woo, W.: 'Bitcoin is the future, blockchain is bullshit': Andreas Antonopoulos preempts consensus. Bitcoinist (2018). https://bitcoinist.com/blockchain-bullshit-andreas-antonopoulos/