

# HashUp - Decentralized distribution of games and software through ERC20-compliant license media

Szymon Jankowski, Softoshi Gamermoto

## Summary

In the early days of computer games, it was all about cartridges, the physical carriers of the software and licenses needed to run it. Games in that time could also be easily swapped or sold on the secondary market. Over time, they got supplanted by digital distribution on platforms that completely dispensed with license carriers and thus eliminated the possibility of exchanging and owning games. Once these platforms took over much of the market, they enforced their distribution terms and basically took away our freedom to use software. There is no direct mechanism for pricing in the free market for software, and the secondary market for games exists only for the physical versions, from which developers make no profit.

HashUp is the first open software distribution based on ERC20-compliant licensed media. Each game released in this form comes with separate ERC20 licenses, called **ERC20 Cartridges**, reminiscent of the physical cartridges of the 1980s. This solution restores the proprietary nature of the trade and is compatible with both the Web2 and Web3 worlds. It allows you to buy any game or software using Smart Contracts. HashUp's cartridge compatibility with the ERC20 standard makes it possible to store games in any cryptocurrency wallet and provide liquidity on decentralized exchanges to determine the true price of software using an Automated Market Maker (AMM) - the mechanism we called **GameSwapping**.

HashUp introduces game distribution platforms inspired by the most popular cryptocurrency platforms and compatible with Web2 and Web3 games. **GameCap.io** will act as a crypto capitalization and ranking aggregator for gamers and games, and **GameExplorer.io** will act as a blockchain explorer for gamers, combining EtherScan, EpicGames and OpenSea into one portal. **GameWallet** will serve as a wallet and messenger for gamers. All platforms work with the HashUp PC Launcher, which is connected to Metamask and contains games in the form of ERC20 Cartridges. The HashUp platforms bring software distribution to the Web3 world and decentralize it through the openness of ERC20-compliant media.

## Introduction

In the 1970s, physical licensed media such as floppy disks and cassettes were developed. They were carriers of both the license and the software itself. The license was designed in such a way that the software carrier was also the carrier of the license that entitled us to use the software. With the carrier, we had the rights to use the programme. Thanks to game stores there was a small market for games, and their prices on the secondary market reflected their quality. This allowed the free exchange of software, but the loss was the developer who could not control or profit from the physical secondary market. In addition, physical licensed media could easily be lost or broken. If you wanted to buy a game, you had to physically go to the shop, which was an additional time commitment.

With advances in technology, digital software distribution platforms such as Steam, Epic Games and GOG have emerged. Digital distribution is primarily characterized by convenience. However, by eliminating the physical media of the software, it has stripped us of the proprietary nature of the license we purchase. The software license is permanently assigned only to the account belonging to the platform. This prevents any trading outside the platform. Furthermore, the absence of the proprietary nature of trade leads to a centralization of software distribution. Not everyone can set up online game shops, as was the case with physical retailers. The platform driving traffic can rob developers of their revenue in an instant if they do something against their will.

It is also worth mentioning the extreme variant of digital distribution that is becoming increasingly popular - the subscription model. In this solution, players pay a subscription fee to access titles on the platform. The subscription model leads to a greater centralization of software distribution than ever before, and the platform freely manages the available titles.

Blockchain Technology, on the other hand, makes it possible to transfer the laws of the physical world to the digital world. The best proof of this is Bitcoin, which was created by Satoshi Nakamoto in 2008<sup>[1]</sup>. Blockchain makes it possible to combine the advantages of both worlds by transferring the properties of physical money to its digital form, without any intermediaries. Public blockchain explorers have been created to view the contents of the blockchain. In 2013, CoinMarketCap was established to index the prices of all cryptocurrencies in one place.

However, Bitcoin's blockchain is not programmable. This means that we cannot write our own contracts or licenses on it. Vitalik Buterin combined blockchain technology with a virtual machine in 2015 to create Ethereum<sup>[2]</sup>. It was the first blockchain to enable the creation of smart contracts. Shortly afterward, Vitalik Buterin and Fabian Vogelsteller created the ERC20 standard<sup>[3]</sup>. This is a standard interface that allows all tokens on Ethereum to be reused by other applications: from wallets to decentralised exchanges. However, the current transaction costs on the Ethereum network prevent the spread of software. The problem of scaling Ethereum is being solved by other networks, such as Polygon<sup>[4]</sup> and Avalanche<sup>[5]</sup>, which were founded in 2019-2020. Transactions on these networks are much

cheaper and faster, making it possible to build a convenient game shop with network-owned licensees in the form of just ERC20.

Furthermore, the HashUp architecture is designed for easy scaling to other networks, such as Hadera or Scale.network. The final part of HashUp is DeFi (Decentralised Finance), or essentially liquidity pools, created in 2019 to provide liquidity to ERC20-compliant tokens using the Automated Market Maker (AMM) mechanism. It allows tokens to be bought and sold without requiring an order from the second party (beyond just providing liquidity). In addition, with liquidity pools you can swap both ERC20 tokens and ERC20 cartridges with one another - we call this **GameSwapping**. HashUp uses the Uniswap [\[6\]](#) V2/V3 protocol and future versions for this.

HashUp has integrated the above mechanics into a single platform, resulting in decentralized software distribution:

- blockchain - acting as a decentralized network of license carriers,
- smart contract ERC20 - acting as a software license carrier,
- smart contracts - acting as a store of license carriers that anyone can use/add,
- Blockchain Explorer - in the form of GameXplorer.io, serving as a blockchain browser for gamers,
- DEX with AMM, Uniswap - acting as an engine for free market pricing of games on the secondary market,
- CoinMarketCap - in the form of GameCap.io, as a secondary market software price indexing platform.

It is worth noting that HashUp carriers, despite their compatibility with the Web3 world, are not intended for exclusive distribution of Web3 software. They are designed to **distribute any kind of software**, including games.

### **Problems of Web2 Software Distribution Issues of the Web2 world and solutions in the Web3 world:**

By bringing the above mechanisms together in a single platform, HashUp solves a number of problems that software distribution faces today. At the same time, it brings software distribution into the new world of Web3.

### **Problems of the Web2 world and solutions in the Web3 world:**

**Lack of ownership of purchased games** - using blockchain technology, i.e. decentralizing the license holder base, HashUp has put ownership of games back into the hands of players. With ERC20 Cartridges, it is possible to own games in exactly the same way as physical media.

**Lack of a secondary market** - by using the ERC20 standard medium for software licenses, a secondary market for software can be brought to life. At the same time, creators are satisfied thanks to arbitrarily high fees for exchanging the medium between players. Such a mechanism means that a player can resell his game at any time.

**Lack of a free market mechanism for setting the price of games** - the HashUp Cartridge standard is compatible with ERC20, thanks to which liquidity can be provided for licenses on all decentralized exchanges. Moreover, the price of the software is set by the AMM - Automated Market Making mechanism and does not require any external party, only people providing liquidity. Besides, the creator will earn profits for each trade of the license/copy/article.

**Lack of immediate remuneration of creators** - with smart contracts it is possible to transfer payment for the software immediately to the creator, the platform itself or the person providing the reflink. Furthermore, thanks to the openness of the blockchain, settlements with creators become more transparent than ever before.

**Absence of a game sales platform designed for players** - HashUp solves this problem with a liquidity/platform token # that automatically transfers profits for selling games on the platform to token holders. It is possible thanks to the Buyback&Burn mechanism, which enables buying the token # from the liquidity pool and then burning it. Thus, the price of the token increases every time a game is purchased on the platform. In this way, the sales effectiveness of a distribution platform directly affects the value of its payment token.

**The problem that we are not able to create our own online game stores** - to explain the problem in more detail: let us assume we cannot create our own Steam or Epic Games because then we would have to discuss distribution rights, and, as a result, our store would be half empty, devoid of real value. Fortunately, we can solve this problem with the openness of blockchain technology. Just as we can now create cryptocurrency exchanges and stock markets without Satoshi Nakamoto's permission, with the openness of blockchain technology and the ERC20 cartridge, it will be possible to create stores that sell all games published with ERC20 media. This opens up huge opportunities for influencers and anyone who wants to have their own game store. HashUp has created an open store that anyone can join. You are given a chance to sell and monetize games while connecting to a token # and increasing its value with each sale.

**Inability to store all the games in one place** - thanks to the transparency of ERC20 cartridges, you can store games in any cryptocurrency wallet compatible with this standard, regardless of whether the game was published by Ubisoft or EA or was exclusive to one of the platforms.

**Excluding players from investing in games** - due to the features of the ERC20 standard, it is possible to sell copies of games at, for example,  $\frac{1}{3}$  of the price at launch and share profits from production directly between investors and developers. Moreover, the openness of the blockchain allows anyone to participate in such investments. Currently, when games are funded through platforms like Kickstarter, we do not share in the profits associated with production. In the case of ERC20 backers, the profits can be shared with investors.

Web3 world problems associated with current digital distribution:

**The problem of not being able to buy games directly through smart contracts** - currently, we cannot buy any software licenses through Metamask or smart contracts, and thus buy P2P games. HashUp solves this problem with ERC20 Cartridges, the purchase of which on the primary market is similar to buying via smart contract for ICO, while its purchase on the secondary market is realized via DEX using contracts with liquidity pools.

**The problem of not being able to provide liquidity under software swaps** - thanks to the compatibility of ERC20 Cartridges with the ERC20 standard, it is possible to provide liquidity for game licenses and monetize the ability to swap games for tokens and other games.

**The problem with license storage location** - in today's popular solutions, licenses are usually stored on centralized platforms. They cannot be moved outside of them, including holding them at home. Customers of such platforms are entirely dependent on them. There is currently no way to easily hold licenses in your digital wallet, such as it is possible with Metamask for NFT tokens.

## **ERC20 cartridge as a software license carrier**

The ERC20 Cartridge is a software license carrier on the blockchain. Each game has its own ERC20 Cartridge, just as in the past each game possessed its own separate and independent series of cartridges, floppy disks or CDs. The cartridge, like the ERC20 standard, is a homogeneous entity. This means that one cartridge with game X is exchangeable with another cartridge with the very same game. Compatibility with the ERC20 standard enables an entire ecosystem of contracts created under the utility for homogeneous tokens. **The primary benefit of the ERC20 Cartridge is the ability to use the software if you have  $\geq 1$  cartridge, as was the case with a physical distribution.**

**The price of an ERC20 Cartridge Unit represents the value of a single carrier that enables the use of the software.**

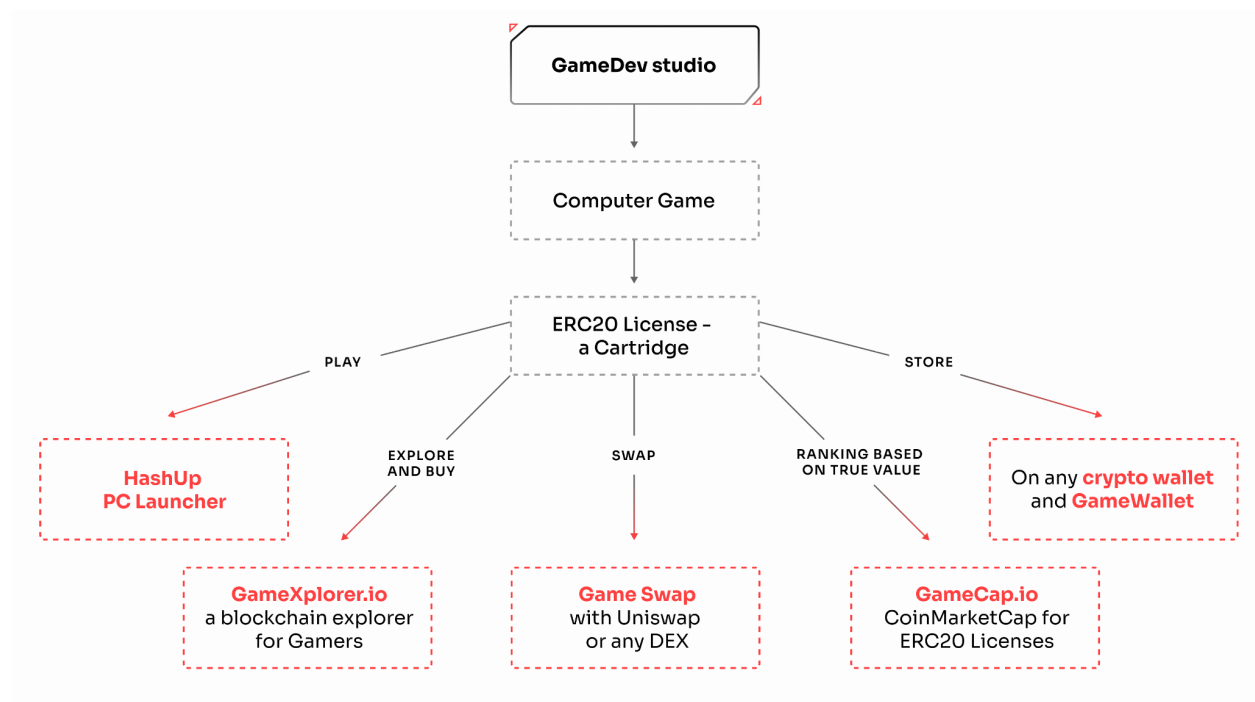
**The capitalization of an ERC20 Cartridge represents the value of all license mediums of a particular game that is currently in circulation.**

An explanation of the ERC20 standard in the context of a license carrier:

- **function** name() **public** view returns (string) - The name of the game/software title
- **function** symbol() **public** view returns (string) - Abbreviation of the title
- **function** decimals() **public** view returns (uint8) - Number of decimal places, HashUp creates cartridges with 2 decimal places.
- **function** totalSupply() **public** view returns (uint256) - Total supply of license carriers for the given software.
- **function** balanceOf(address \_owner) **public** view returns (uint256 balance) - The number of cartridges located at address \_owner. As in the case of physical media, one cartridge is required to have the right to use the software.
- **function** transfer(address \_to, uint256 \_value) **public** returns (bool success) - A functionality that enables software exchange between users.
- **function** transferFrom(address \_from, address \_to, uint256 \_value) **public** returns (bool success) - funkcja umożliwiająca chwytnie przez inne smart kontrakty w celu programowania dokładnego obrotu licencjom - A function that allows other smart contracts to capture in order to program the exact number of licenses.
- **function** approve(address \_spender, uint256 \_value) **public** returns (bool success) - a function that makes a given number of licenses available for other contracts to use.
- **function** allowance(address \_owner, address \_spender) **public** view returns (uint256 remaining) - the number of license carriers available for being captured by another contract.

In addition, the HashUp Cartridge is an **extended** version of the ERC20 standard that includes features designed to fully decentralize the software licensing medium, such as cartridge metadata in the form of **function tokenURI() public** view returns (string) and Ownership transfer over the contract in accordance with the Open Zeppelin standard.

### *Benefits of the ERC20 Cartridge*



## Why don't we use NFT/ERC721 as a license carrier for software?

The ERC20 standard is a much better license carrier than ERC721 because:

1. **It is homogeneous**
2. **It stores value rather than information**
3. **It is less expensive to manage in larger quantities**
4. **It is interchangeable**
5. **It is divisible**
6. **It represents a single asset**

We are aware of the **ERC1155** standard, which combines the functionalities of the ERC20 and ERC721/NFT standards - we are considering its implementation in the future.

## Profits from trading the ERC20 cartridge for the owner/creator of the cartridge

HashUp cartridges allow you to make profits on the secondary market. The rules under which you can earn commissions require that an existing pool of liquid licensed media be transferred to another liquid asset. In the **transfer function**, the license itself sells a portion of itself (as determined by the creator) to its own pool of liquid assets and passes the resulting value on to the creator. The remaining portion goes



to the recipient. The mechanics of cartridge trading commissions means that the transfer function automatically passes the commission to the creator, while at the same time passing the cost of the transfer to the receiver.

## Types of ERC20 Cartridges

There are 3 main types of ERC20 HashUp Cartridges:

- **Gold** - collectable, characterized by a maximum supply of 133,700 units and a maximum commission of 10% per trade. Such cartridge can be used for speculation due to its limited supply, which never fluctuates.
- **Grey** - standard cartridge, which has no supply limit at the time of creation and no limit on the amount of commission for trading on the secondary market. Standard License Carrier.
- **Custom** - modifiable, distinguished by complete freedom in shaping the aftermarket model and the 3D model.

## ERC20 Cartridge creation

The creation of ERC20 Cartridges is accomplished using the **GameContract.io** platform. The name **GameContract** refers to the "Smart Contract", which is the license carrier for a computer game. Creating ERC20 Cartridges can be outlined with a few simple steps.

1. Selecting the type of cartridge (gold, gray, custom).
2. Filling in the basic cartridge data, the secondary market trading rules, and protection time before aftermarket. Protective period means no exchange for a certain time after release to prevent the emergence of a secondary market.
3. Populating the metadata and visual aspects of the profile on **GameExplorer.io**.
4. Creating the ERC20 Cartridge with the help of Metamask.
5. Connecting the ERC20 Cartridge with the game files.
6. Setting the price and number of cartridges for sale.
7. Verification of the game's copyright is performed by the HashUp team.
8. Official sale on GameExplorer.io and GameCap.io platforms.
9. After the protection period before the secondary market, it is possible to transfer games between players and provide liquidity on DEX. From then on, the price of the software is set by the free market using AMM (Automated Market Maker).

Games are being sold on the platform with **any token that is approved by the creator**. This indicates that the creator can earn in USDC, WBTC, ETH, MATIC. The solution is scalable and facilitates easy switching from FIAT => CRYPTO .

## New capabilities for ERC20 Cartridges

The ERC20 standard opens up a number of possibilities such as:

- **Compatibility of ERC20 cartridges with crypto wallets** - thanks to the ERC20 standard, it will be possible to store games along with Bitcoin, Ethereum and tokens in cryptocurrency wallets. This is very convenient, especially for Web3 users.
- **Compatibility of ERC20 cartridges with DEX** - a free market for games made available through DeFi, DEX and AMM.
- **Swapping games and tokens with one another.**
- **Exchange p2p games in the same way as you do with ERC20 tokens** - with cryptocurrency wallets, it is quite possible to exchange ERC20 cartridges. This means that all available token infrastructures work natively with HashUp Cartridges.
- **Game license carrier as in-game currency** - by using the ERC20 standard, you can use such a cartridge as an internal currency. The free market will force additional usability of ERC20 cartridges over time, as it will demand mechanisms to generate demand for licenses. This implies that in the future, additional items in the game will be paid for in "units of the game itself". The capitalization of such a cartridge will represent the value of the entire network associated with the game, condensed into a single medium. GameCap.io was born out of the need for a single place to index all capitalizations of game-related networks.
- **Ownership of the game** - you can own ERC20 cartridge just like the cryptocurrencies. No one can take them away from us. Moreover, they are stored only in crypto wallets and not on central exchanges, so the probability of losing them is minimal.

## The concept of decentralized games and the ERC20 Modification License

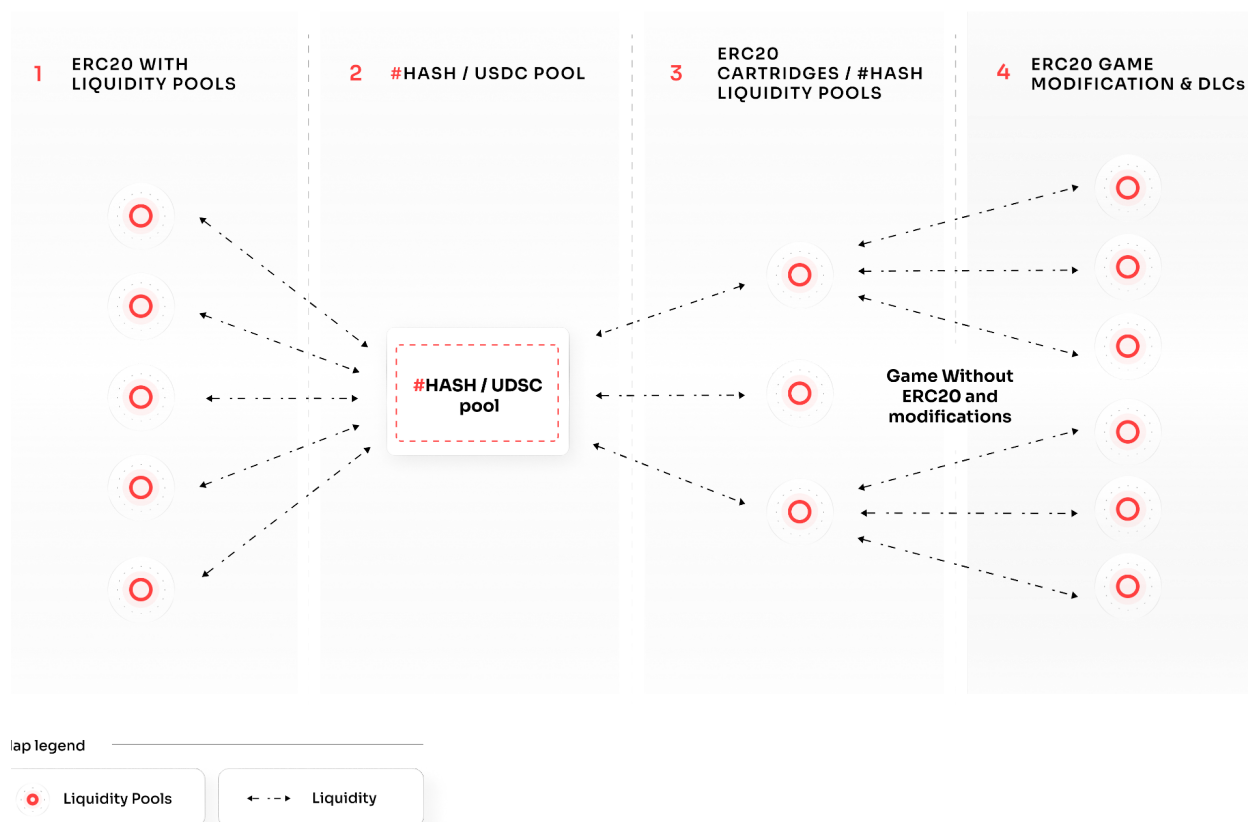
The idea for the concept of decentralized software came to us when the release of the game Cyberpunk 2077 failed. Cyberpunk, like most games at the time, was developed in a completely centralized way. However, when thinking about the concept of decentralized games, we ran into another problem. In our opinion, it will be solved before the end of this decade (perhaps by HashUp itself). The problem is that we cannot develop and sell our own games and expansions with their own storyline. The world created by CD Projekt Red - Cyberpunk - can only be changed and improved by a centralized group of people close to the company. However, we believe that the future lies in decentralization, which should allow us to create our own stories within games in the future. The ERC20 Cartridge technology is scalable to **ERC20 Modification Cartridges**. These are modification licenses that are technically identical to ERC20 Cartridges, but are a license carrier for modifications to the game. By modifications we mean not only mods, but also quests and entire storylines that anyone can create and combine.

### Scalability of ERC20 Cartridges

- **ERC20 Cartridge as a Decentralized Software License Carrier** - the games of the future will be developed in a decentralized manner. ERC20 cartridge distribution can be handled through

smart contracts programmed to give developers a fixed fee for actual game sales. Creators of decentralized worlds will be able to be rewarded in the same way that Bitcoin is automatically distributed at proof-of-work.

- **ERC20 Cartridge as liquidity for ERC20 modifications** - this means that modifications will in some way be based on the core license, and the purchase of such customizations will generate demand for the ERC20 Cartridge. In this sense, the core game license holder will be the capitalization of the entire game-based modification network.



*Example of liquidity flow for game modifications and extensions in the form of ERC20 license media*

The image displays an example of value flows between liquidity pools.

1. It represents liquidity pools up to \$USDC, so all ERC20 tokens that have value.
2. It represents the \$HASH/USDC liquidity pool, which mediates value flows between tokens and ERC20 Cartridges.
3. The ERC20 Cartridge and \$HASH liquidity pools, which make it possible to exchange the game for other tokens.
4. Modifications and extensions to the game in the form of ERC20 liquidity licenses built on the basic version of the game.

**ERC20 distribution of ERC20 Cartridges that can be designed by you** - the process by which new homogeneous tokens are put into circulation, is called distribution. In the case of the Bitcoin network, the people who maintain the network are called Miners and they acquire new Bitcoins through a process called **mining**. With ERC20 Cartridges, it is possible to freely set the distribution rules for new game units in circulation.

The above examples were only intended to describe, in general terms, the possible design of ERC20 cartridge distribution conditions. As with cryptocurrencies, these are unlimited.

### **GameSwapping as the heir to the subscription model**

GameSwapping is swapping one game for another with DEXes. All ERC20 Cartridges, even if they were released completely independently, can be swapped with each other as long as they share a common liquidity. The ERC20 cartridge-based software distribution model, through its versatility with a single licensing infrastructure, satisfies both: collectors who want to build a large game collection and people who just want to play and pay regularly. At the same time, this model combines both ownership and access to a library that grows with each new game released this way. So this is the future of game distribution, which will thwart the biggest of the gaming world to introduce a subscription model, benefiting both developers and players.

Thanks to GameSwapping you are able to:

1. Exchange one game for another p2p game and at the same time generate profit for the creators of the titles in question.
2. Exchange any liquidity token for a game with liquidity.
3. Exchange any game that has liquidity for any other liquidity token.

### **Why will GameSwapping replace the subscription model?**

GameSwapping will substitute the subscription model because of the **network effect** created by a single standard for ERC20-compliant licensed media. **Each subsequent game released with ERC20-compliant licenses will extend the functionality of the previous game. At the same time, it will be possible to interchange these games. Thus, a game released with an ERC20-compatible standard enters the pool of games issued this way, regardless of the platform on which it was published. ERC20 licenses will mostly replace the subscription model for this reason.**

The bottom line is this: by conforming to a single standard, games published on ERC20-compliant carriers will become part of a large pool of games that can be traded and collected with each other, regardless of the platform on which they were created. This will satisfy title collectors and fans of the subscription model at the same time.

### Where do we keep the game files?

The game files are hosted on AWS servers. The Launcher technology is the responsibility of the PatchKit team, who have been working on their product for over 5 years. However, we are ultimately interested in decentralizing the file storage location.

We are currently reviewing the following solutions:

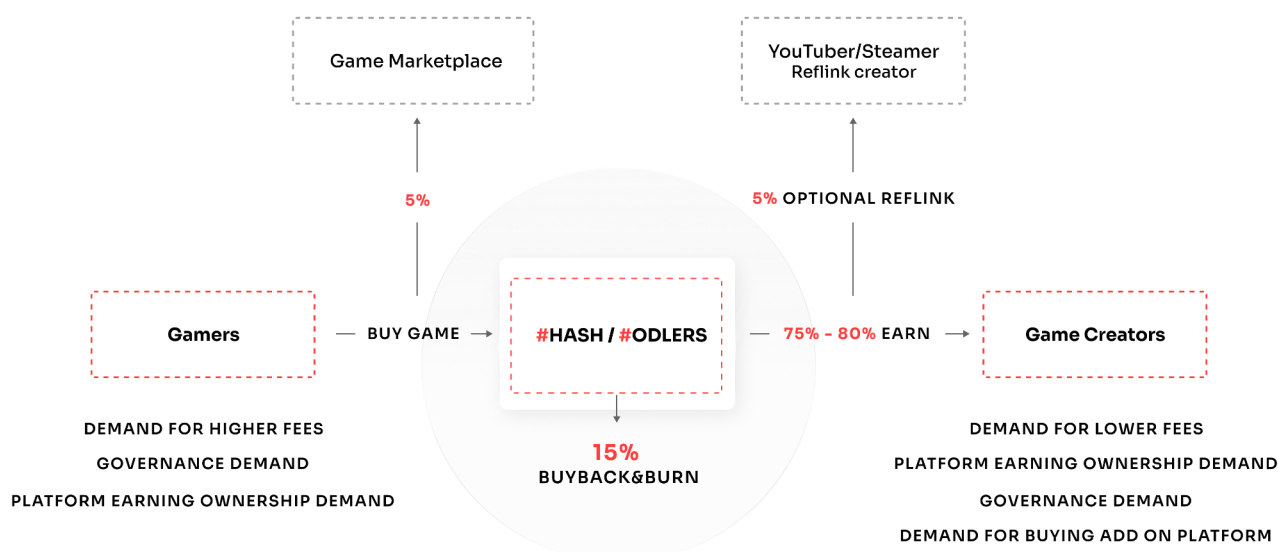
- **Torrent stored files** - we will be testing torrent integration in our launcher. However, we are concerned about possible overloads with AAA games downloaded by hundreds of thousands of players after release.
- **Sia, Filecoin** - decentralized CDNs - content delivery network. We will test storing game files on these networks. However, we believe that they are not yet ready for AAA games and network overload for game launches.

However, we believe that decentralized CDNs will enable full decentralization of license holders as technology advances, and we will support this type of technology.

### Token \$HASH / #

The # token is a liquidity/platform token of the HashUp platform, which is based on the distribution of ERC20 Cartridges. The name HashUp is a direct allusion to the underlying mechanism behind the token - once a game is purchased on the primary market, the price of \$HASH (UP) increases thanks to a 15% buyback&burn mechanism every time a game is purchased.

*Architecture of the \$HASH token economy*



### Glossary:

**Liquidity token** - a token that is being distributed in the course of providing liquidity to other tokens. Examples of liquidity tokens include Cake ( Pancake Swap ) or Uniswap. Typically, the primary use of liquidity tokens is to manage the protocol on which their distribution process takes place. The problem with liquidity tokens is that there is no constant source of demand for the token.

**Platform token** - a token associated with a specific platform. Many launchpads offer their own platform token and part of the revenue generated on the platforms goes into the payment token in the form of buyback&burn or staking.

**Buyback** - repurchase of tokens from a liquidity pool.

**Burn** - burning tokens to prevent their further trading and sale.

**Staking** - locking tokens of different types over time to obtain new tokens in circulation or tokens from a specific source.

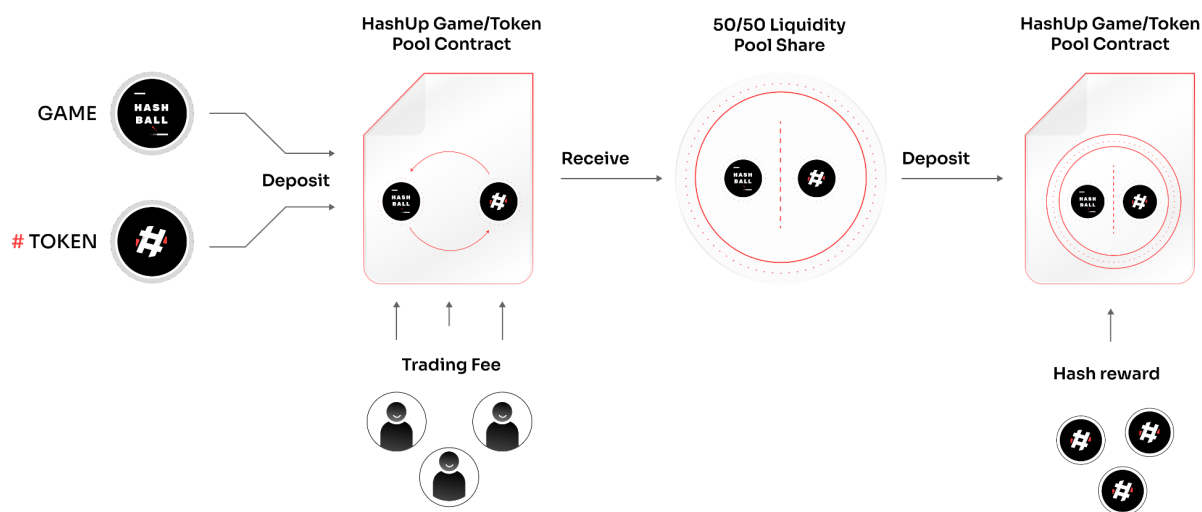
**Liquidity pool** - a pair of tokens whose mutual ratio determines the price. If you buy A tokens for B tokens in an A / B pair, the number of B tokens increases in proportion to the A tokens, resulting in a higher price for A tokens. Liquidity pools intermingle to create routes. The exchange between intermediary tokens creates demand for all intermediary pools along the route.

In this example, we have omitted the issue of commissions for the liquidity provider and greatly simplified the calculation.

**Liquidity** - the depth and ease of exchange between value carriers. The greater the liquidity, the smaller the price fluctuations during the exchange. Liquidity Providers make money when trading the liquidity they provide and during liquidity mining.

**Liquidity Mining** - an additional incentive to provide liquidity is so-called process of liquidity mining. During regular liquidity provision, providers earn a total of around 0.3% of the value of each trade (Uniswap V2), where profits are split proportionally to the share in the pool. After liquidity provision, LP tokens are received as proof of the process. By staking LP tokens we earn double, both from providing liquidity and in new tokens in circulation.

### *Liquidity Mining Process*



Basic data of the \$HASH/# token:

- Name: HASH / #
- Initial/maximum supply: 1 000 000 000
- Type: ERC20, Burnable
- Tokenomics:
  - 30% fundraising & listing, 6 months vesting
  - 35% positive inflation in the form of liquidity mining
  - 35% HashUp company and contributors (1/3 of tokens owned by HashUp company will be dedicated to game acquisition and liquidity on CEX), 1 year cliff, 5 years vesting

Details:

<https://docs.google.com/spreadsheets/d/1Gt7eZp1ZapTNP5GqbDUtdIism1IPSYLRkXxGqtWfVQI/edit#gid=1407805475>

In designing the \$HASH token, we focused on:

- Generating a constant demand for \$HASH
- Generating a constant trading volume for \$HASH

### **How do we generate demand for the \$HASH token?**

#### **The sale of games on the platform generates demand for the # and burns it to the tune of 15% of turnover.**

HashUp platforms are primarily geared towards selling games in the form of ERC20 cartridges. When selling games on platforms like Steam, Epic Games or GOG, a commission of 20-30% is due, and the developers have no problem with that. So we combined the demand for games sold on the platform with the demand for the \$HASH token. Instead of ending up in the company, the game sales commission buys \$HASH out of the liquidity pool and burns it.

The amount of commission at the start of the platform is as follows:

- 15% Buyback&Burn,
- 5% is given to the platform where the sale of the game took place,
- 75%-80% goes to the creator,
- 0%-5% *optionally contributes to the reflink.*

This basically means that if you purchase a game for \$100 USD, \$80 USD goes to the creator, \$5 USD to the reflink, \$5 USD to the platform where the sale took place, and \$15 USD to the pool of \$HASH for burning. The exact mechanics of how the commission works are described in the section that describes the Game Shop Agreement.

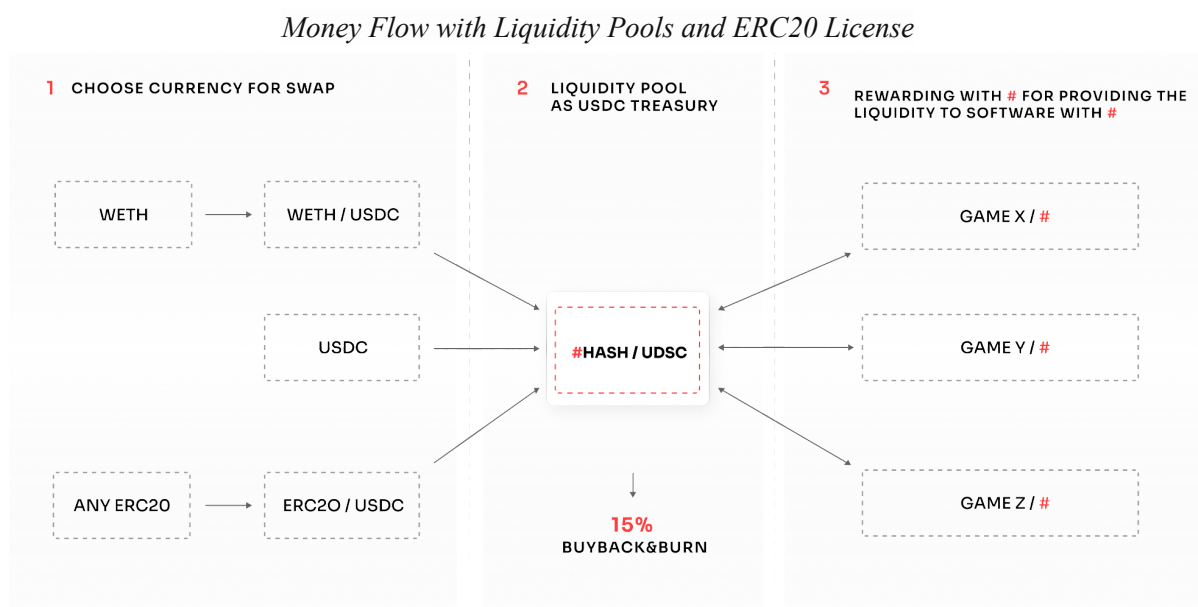
The 15% BUYBACK & BURN mechanism holds great potential, as the demand for games available on the platform is visible in real time in the valuation of the \$HASH token. In practice, each game sold on the platform boosts the token # price. Moreover, the mechanism burns \$HASH with each purchase, reducing its supply.

### **How do we generate \$HASH volume?**

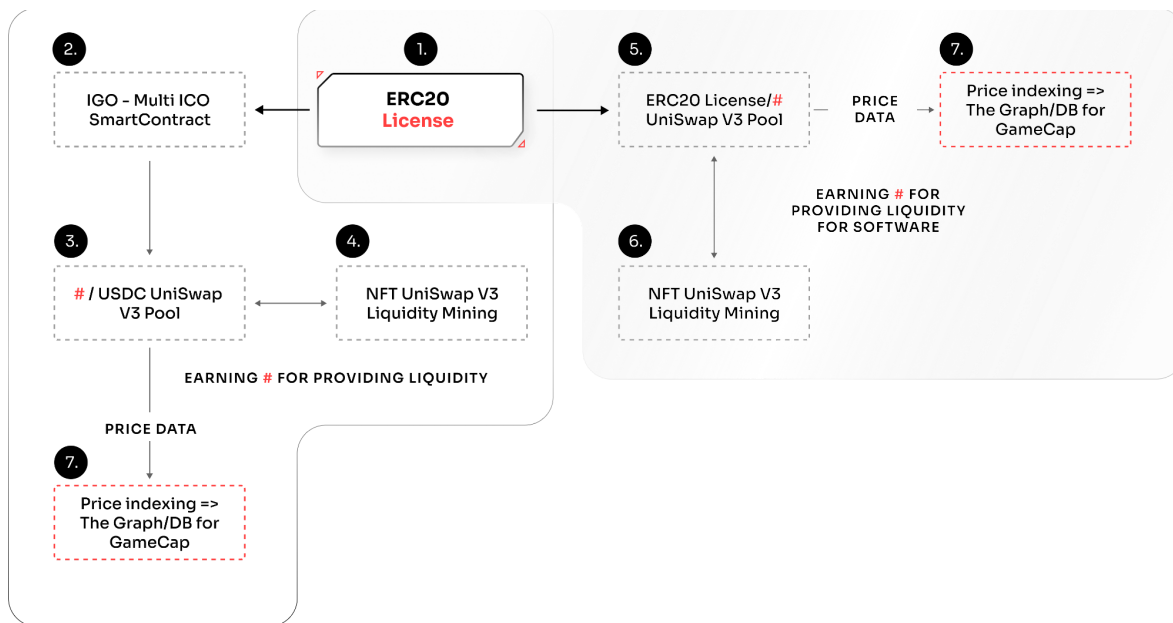
**The \$HASH token is liquid for ERC20 Cartridges and any exchange between them on DEX generates \$HASH revenue.**



The HashUp platform, based on ERC20 Cartridges, allows licensees to be liquid in any token. By leveraging the mechanisms of Liquidity Mining and creating appropriate facilitation on the platform, we are able to direct the creation of liquidity in \$HASH to ERC20 Cartridges, which are also issued independently of HashUp. HashUp provides up to 35% of all \$HASH tokens for this purpose. **Rewards for providing liquidity in \$HASH are given to games** chosen by token holders. This is done to create a positive inflation effect while minimizing the provision of liquidity to weak games that do not provide much network value.



## Smart Contract Architecture of the HashUp platform



The graphic above represents the HashUp smart contract architecture. As such, the architecture shows compatibility with Uniswap V3, but also with V1 and V2. It is also scalable to future ERC20 compliant versions of DEX. The architecture is compatible with all ERC20 compliant licenses issued without HashUp.

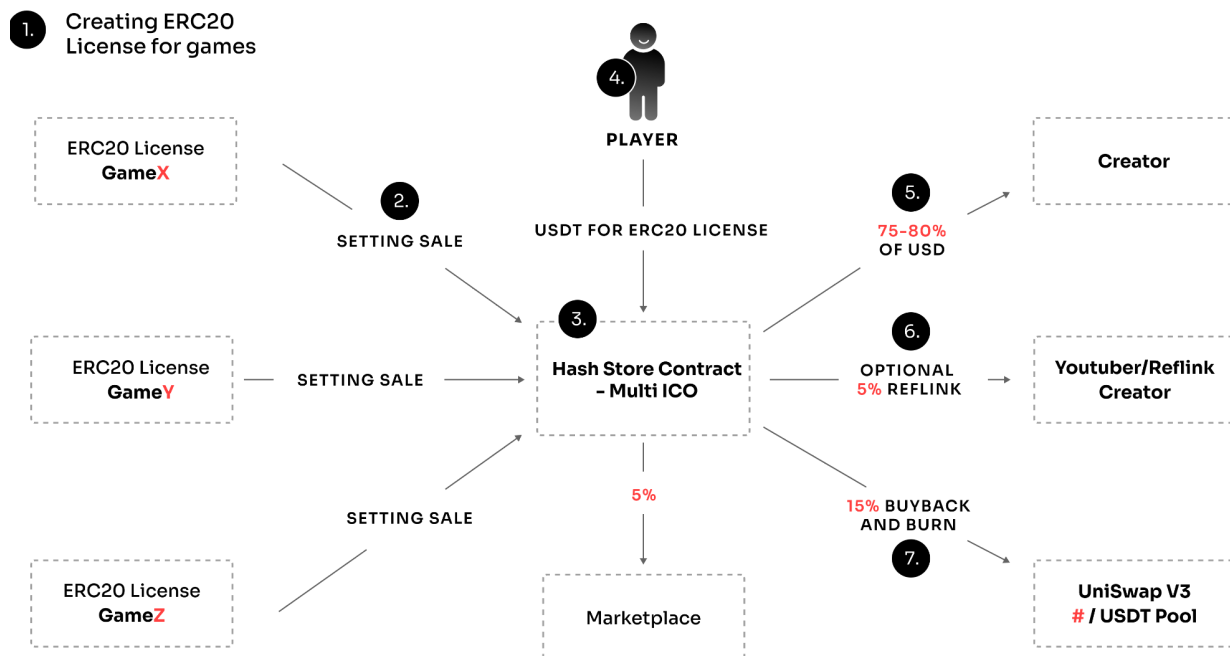
- 1. ERC20 Cartridge** - An ERC20-compliant license, called the ERC20 Cartridge, created by developers on the GameContract.io platform. The ERC20 Cartridge is associated with one (2) MultiICO contract called a Shop. There it is sold at a fixed price (like in game stores) and has its own liquidity pool to collect commissions for trading.
- 2. Multi ICO** - A shop where ERC20 Cartridges are received and sold at a price set by the creators. Technically, the contract is similar to an ICO where ERC20 tokens were sold at a fixed price, but it is rather a Multi ICO contract, meaning that with one contract it is possible to sell any ERC20 for any other ERC20.
- 3. The \$HASH/USDC liquidity pool** is linked to the shop (2) and each time a game is purchased, the \$HASH token is redeemed at 15% of the purchase value from the pool and burned immediately. This mechanism makes it possible to monitor the flow of value between players (consumers) and developers in real time.
- 4. Staking \$HASH liquidity** - the \$HASH liquidity pool is tied to contract (4). By providing liquidity, we obtain LP tokens that can be placed on contract 4 to earn new \$HASH in trade. This process is called **Liquidity Mining** and is used to improve liquidity.
- 5. Liquidity pool \$ERC20 Cartridge/\$HASH** - the liquidity pool where the license carrier goes to, which generates LP tokens for providing license liquidity.

6. **Staking LP \$ERC20 Cartridge** - Liquidity Mining \$HASH tokens to provide liquidity for licenses in the form of ERC20. A license in the form of ERC20 does not need to have a relationship with the HashUp platform to receive the award.
7. Data regarding price changes is aggregated by The Graph/Moralis, which allows graphs with # and software prices to be displayed.

### Summary:

The ERC20 license (1) is transferred to a shop (2), where it is sold at a fixed price. The shop (2) is connected to a liquidity pool (3), by a 15% Buyback&Burn mechanism, which makes each sale of the game boost the value of the token. By providing \$HASH liquidity (4), players earn new \$HASH units as they drain liquidity. Each game has a separate liquidity pool (5) into which ERC20 Cartridges can flow after creators unlock the secondary market. Each liquidity pool generates LP tokens that can be staked in a contract (6). The liquidity pool data is indexed (7) and can be reordered at will.

### IGO - contract, games shop



The ERC20 Cartridges (1), developed by the creators, after setting the sales rules (2) end up in the HashUp Store Contract (3). The shop offers the possibility to sell all ERC20 compliant contracts to other ERC20 contracts at a fixed price. This implies that all payment methods accepted by game studios are available. However, developers using our system will most likely want to earn in stablecoin. The commission for selling games on the HashUp platform ranges from 20-25% and splits the tokens among 4 sources:

1. 75% - 80% of the profits go immediately to the creator (5)
2. 5% **optionally** contributes the reflink (6)
3. 15% goes to the \$HASH token which is bought from the pool and burned - Buyback&Burn mechanism (7)
4. 5% is given to the platform where the sale of the game took place (7)

### **The openness of the HashUp store**

The transparency of the blockchain makes it possible to openly connect to the store and sell games outside of the HashUp platforms. This solves the problem of not being able to sell digital games outside of the platform they are distributed on, while also providing a collection of titles (ERC20 cartridges located in the store). To encourage the sale of games outside the HashUp platforms, we offer a 5% commission to the provider of the store where the transaction took place.

### **NFT platform advertisements that represent ownership of days on banner ads**

Certain ads on HashUp platforms are tied to associated NFT collections. Suppose that a 365 NFT collection represents 365 consecutive days. By owning an NFT with an index of 10, we "own" the day of January 10. This allows us, as the owner, to freely design the content of the ad. The advertising days issued in this way are auctioned on the GameContract.io platform. The highest bid wins. Ads can only be bought for \$HASH and all the profit is then burned, which is an important demand factor in the long run. The number of \$HASH used to pay for the advertising day is public, so ads that burn a lot of \$HASH are usually clicked more often by users of the platform. Of course, the HashUp platform reserves the right not to display content that does not comply with the terms of the platform and the applicable standards.

## **HashUp Platforms**

When HashUp introduces a carrier in the ERC20 standard, it must create an environment that is compatible with it. For this reason, the platforms it presents refer to the cryptocurrency market and the natural solutions that accompanied its development.

### **GameCap.io - indexing and ranking of the ERC20 Cartridges**

The counterpart of a cryptocurrency capitalization and ranking platform personalized for gamers. GameCap.io is a Dapp that allows you to buy games directly and is compatible with all game licenses published on the ERC20 standard.

### **GameXplorer.io - ERC20 cartridge store and explorer**

Combines a digital game store with a blockchain explorer and social media for gamers. It also creates an environment for browsing games and user profiles while introducing an element of social media.

### **GameWallet - a wallet for gamers**

GameWallet is a wallet for gamers integrated into HashUp platforms that also acts as a messenger between public addresses. Its core functionality is based on the ownership of computer games.

### **GameContract.io - a development platform for Game Creators**

The GameContract platform is used to create, manage and connect ERC20 cartridges to game files.

## **Bibliography**

[1] Satoshi Nakamoto: *Bitcoin: A Peer-to-Peer Electronic Cash System*. <https://bitcoin.org/bitcoin.pdf>, 200

[2] Ethereum White Paper, A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM, By Vitalik Buterin:  
[https://blockchainlab.com/pdf/Ethereum\\_white\\_paper-a\\_next\\_generation\\_smart\\_contract\\_and\\_decentralized\\_application\\_platform-vitalik-buterin.pdf](https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf)

[3] EIP20, ERC20 Standard, Vitalic Buterin, Fabian Vogelsteller 2015  
<https://eips.ethereum.org/EIPS/eip-20>

[4] Poligon Lightpaper, <https://polygon.technology/lightpaper-polygon.pdf>

[5] Avalanche whitepaper,  
[https://assets.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16\\_Avalanche%20Platform%20Whitepaper.pdf](https://assets.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16_Avalanche%20Platform%20Whitepaper.pdf)

[6] Uniswap V2 Whitepaper, decentralized exchange <https://uniswap.org/whitepaper.pdf>