THE IDEA OF PIXELCASH

BY KYLE D. USHER

INTRODUCTION

Pixelcash is a conceptual PoW (proof of work), truly decentralized, peer to peer, and completely anonymous cryptocurrency with smart contract (built-in escrow) capabilities and, unlike most, if not all, other cryptocurrencies, it is not artificially supplied and even has a community-driven circulating supply. It can also be called "Pixelnote" in regards to the units of value themselves. See the Pixelcash logo here: https://drive.google.com/file/d/1RWYjKPWEPqHOltP_kppFOIra7_WXtSXn/view?usp=drivesdk. The keyboard symbol for Pixelcash could look something like this: 【P̃】. Hopefully, though, Pixelcash doesn't turn out to be yet another "shitcoin", as there are already too many of those on the market being perpetrated by the likes of these two fine fellas: https://youtu.be/Z4gZGHP1y8U.

Let me just be honest and say that Pixelcash (PXC) was conceptualized as another one of those "novelty" cryptocurrencies, like Dogecoin (DOGE). If you think about it, cryptocurrencies are a lot like fireworks, in the fact that there are thousands of varieties of both, and many of the varieties are themed, and many varieties are simply for fun. Pixelcash would be one of those varieties. Pixelcash basically takes the old "needs more jpeg" meme from the early 2010s (see here: http://needsmorejpeg.com) and turns it into something more modern and useful. They say the a picture is worth a thousand words. Well, because of Pixelcash, now it's worth money. The official tagline for Pixelcash is "Pupa Pictura is Minarum Mille", which is a Latin phrase that literally translates into "A Picture is Worth a Thousand Dollars".

MINING

The premise behind Pixelcash is fairly simple; a whole Pixelcash unit (one Pixelnote) itself starts life as a grouping of exactly 1,000,000 (one million) freshly mined, original quality .jpeg format images, hosted in a cloud. They can be images of anything and everything, although it actually doesn't matter exactly what the images are, because the human eye never even sees the images (only the amount of Pixelcash in their wallet), they are randomly selected by the cloud's web crawling algorithm, which means they could come from an obscure blog written by a delusional teenager, the website of a large company such as Walmart, or even from your own personal Facebook page. It also doesn't matter if the images in question are not already in .jpeg format on the original web source, because the Pixelcash algorithm automatically saves them in .jpeg format when they are mined. Note that because the images mined in question are not viewable by anybody, and thus there is no way to determine exactly which images were mined and when, how many times they have been transacted, or their condition, their distribution within the Pixelcash network would not violate any copyright laws that would normally apply to any of the images, ex. logos, exclusive celebrity photos, images mined from Google Books, etc.

The task of running this web crawling software that locates, copies and encrypts these random images into the cloud is a component of the "mining" for this whole scheme, and just like (early) Bitcoin mining, anybody with a laptop or gaming PC can join in. However, unlike Bitcoin, the Pixelcash mining algorithm will be programmed to be ASIC resistant, i.e. remaining resistant to ASICs through regularly changing its adaptive mining algorithm, as ASICs cannot adapt to change. Choosing the right consensus tool is key to keeping Pixelcash decentralized and in the hands of the entire community, not just people with the most powerful machines, ASICs cannot achieve this. This is one of Bitcoin's main disadvantages. Pixelcash mining would also probably be a more suitable alternative (as opposed to Bitcoin or Ethereum mining) for the many homebuilt mining rigs that utilize GPU graphics cards, such as Nvidia, not only because Pixelcash mining is ASIC resistant, but also for the fact that it involves actual

graphics, albeit in a different form. It is not clear what the unit of mining power would be called in this case. For example, Bitcoin typically uses “terahashes per second”, or TH/s, as a measurement of power, because the SHA-256 algorithm is, well, a hash. Other cryptocurrencies, like Bitcoin Gold, use “solutions per second”, or “sols/s”. Perhaps because .jpeg uses a lossy form of compression based on a mathematical operation called the “discrete cosine transform”, or DCT (see this article: https://en.wikipedia.org/wiki/Discrete_cosine_transform), the unit of mining power for Pixelcash could be called “transformations per second”, expressed as “trans/s”.

It would probably be a wise idea for the Pixelcash network to have its own independently operated and self-contained indexing mechanism (search engine) working in conjunction with the cloud, allowing access to the web’s content for the purpose of mining, simply because it would eliminate the need for reliance on third party search engines such as Google, Yahoo!, Bing, etc., as these third parties could choose to block the Pixelcash network from using their services to mine for any reason, or even for no reason at all. “Self-contained” means that the indexing mechanism itself is an extension of the cloud and not accessible by the public (cannot be used by people to surf the web), in a similar way to how the images that make up Pixelcash cannot be viewed, it is only accessible by the Pixelcash mining software and the cloud. The Pixelcash indexing mechanism would also likely need to be programmed to index images only, and exclude everything else (text, videos, GIFs, etc.) that is of no use.

Each .jpeg image is assigned a unique serial number by the cloud, and the exact arrangement of pixels within the image, the date and time it was mined, and it's date and time of mining are also recorded. Pixelcash has six decimal places, displayed as 0.000000. The smallest unit of Pixelcash is called a “jpenny”, named after, you guessed it, the .jpeg file format. It is one one-millionth of a Pixelnote.

SPENDING

When Pixelcash is spent from a wallet, the individually encrypted images (jpennies) are decrypted, and each is matched against the data saved earlier in the cloud's ledger (i.e. the serial number and exact arrangement of pixels within the image), and once everything checks out, the jpennies are processed by the cloud's algorithm (more detail on how below) and delivered to the recipient's wallet, before being encrypted again. Just like Bitcoin, odd amounts of Pixelcash can be spent, for example; if 25.327496 (twenty five point three-two-seven-four-nine-six) Pixelnotes were spent, that means that 25,327,496 (twenty five million, three hundred and twenty seven thousand, four hundred and ninety six) individual jpennies, each with their own unique serial numbers, would be processed by the cloud's ledger and "given more jpeg", or compressed into a reduced file size, incurring loss of information and introducing compression artefacts as a result of the "lossy compression" characteristic of the .jpeg format. There would also be a small transaction fee awarded to the miners.

After each transaction, the exact pixel arrangement of each and every jpenny is again recorded, this data is saved again to the cloud ledger, and the cycle repeats when the Pixelcash is spent again. Compressing these large amounts of .jpeg images and processing their credentials to verify and complete transactions are the other component of Pixelcash mining, in addition to collecting the original images to be used as jpennies.

Unlike Bitcoin, Pixelcash transactions are completely anonymous, as the Pixelcash ledger is private. The only details displayed on the ledger are the transaction ID (copies of which are included in both of the users' transaction histories for their own reference), the amount of Pixelcash sent or received, the date and time of the transaction, and the size of the transaction in either megabytes, gigabytes, terabytes, or petabytes. For example; according to online-convert.com (see this article: https://blog.online-convert.com/average-file-sizes/), the average file size of a .jpeg image on the web is 12 kilobytes, or 12 kB. If a whole Pixelnote

is comprised of 1,000,000 such images, that means a transaction of one whole Pixelcash unit would "weigh" around 12 gigabytes, or 12 GB, in size. Of course, because the file size of each individual jpenny would vary, and because fractions of Pixelcash can be spent, the ledger's algorithm would need to choose the appropriate prefix (mega, giga, tera, or peta) in regards to the number of bytes in each transaction. Megabytes are the smallest unit that can be used for this, for the simple fact that not less than 100 jpennies, with a combined file size of 1.2 MB, or 0.000100 Pixelcash, could be spent at a time due to network fees.

Naturally, because the file size of each jpenny is reduced with each transaction, the transaction size for those jpennies will be displayed on the ledger as being smaller and smaller until the end of their life. For example; since one newly mined Pixelnote has a combined file size of roughly 12 GB, a transaction of the same amount of Pixelcash that only has a transaction size of 3.2 MB could indicate that the jpennies that make up the unit are nearing the end of their life. This is an important detail to include in the ledger, especially for recipients, to help them gauge the viability of their funds and request new payment in fresh Pixelcash from the sender if desired. Wallet addresses or their final balances are not displayed, similar to the Monero ledger.

Due to the nature of this particular system, it would not be computationally practical for a full client wallet to exist. Full client wallets verify transactions directly on a local copy of the blockchain saved on the user's device. The file size of 657,000 Pixelnotes (see the source of that number below) alone would weigh over 7.8 petabytes, or 7.8 PB, which is far more than any household PC and even the entire Sia cloud (https://sia.tech) can store. Therefore, only lightweight client wallets (such as smartphone apps like Coinomi), online wallets, paper/physical wallets, and hardware wallets would be practical for holding and spending Pixelcash. The official wallet client (particularly the smartphone version) could be equipped with APIs that allow users not concerned with anonymity to connect their Facebook, Instagram, or Twitter accounts to the network and "donate" their own images (i.e. consciously allow their social media profiles to be mined) as well as share wallet activity and request Pixelcash from friends. An example of a Pixelcash paper wallet can be seen here:

https://drive.google.com/file/d/101zltxfIHLXRX9S8FlMRFdznQYIIZTRn/view?usp=drivesdk. Pixelcash smart contracts would use a form of escrow dubbed "Mutually Assured Destruction" (MAD) escrow (see here: https://bitcoinmagazine.com/articles/particl-takes-mad-approach-escrows-maximizing-privacy/).

If you're not familiar, a smart contract is an agreement that can be enforced through a blockchain. Rather than relying on trust or a legal framework to ensure that each party that enters into a contract will adhere to its terms, you can use the blockchain to create a contract that is automatically enforced, between two people, in a decentralized fashion. Ethereum has become the most popular blockchain for creating smart contracts. One of the major design goals of the Ethereum platform was to support smart contracts. From the start, this set Ethereum apart from Bitcoin, which was created first and foremost as a digital currency platform. Both lightweight client and hardware Pixelcash wallets would have the smart contract capability built in. These wallets would enable smart contracts basically by locking (temporarily rendering unspendable) deposited funds until all of the parties sign off on the transaction. MAD escrow is a technique that effectively prevents fraud in a transaction without requiring the oversight of a third party. In a MAD escrow contract, a buyer and seller both place funds into escrow. The seller starts by depositing an amount they want the buyer to match to symbolize a virtual handshake. This could be between 1 and 100 percent of the item's purchase price. The seller then creates an "invoice" for the transaction, which is essentially the contract itself, containing the seller's payment address and the exact amount required to complete the transaction. A copy of the invoice is then sent to the buyer either in the form of an auto-generated alphanumeric code (ex. A1BB23C4D567E890) which is the transaction ID itself, and is to be copy and pasted into the "pay to" field of the buyer's wallet, or as a QR code that is to be scanned. The buyer then deposits an amount equal to the handshake amount plus the price of the item they are buying. The escrowed funds are not released to anyone until both parties confirm that the transaction has been completed satisfactorily. When both parties do confirm this, the smart contract will be fulfilled and will be displayed as "confirmed" on the ledger. If a dispute arises, and the transaction is ultimately canceled, then it will be displayed as "unconfirmed". In either situation, the status of the contract is final and cannot be altered or

supplemented in any way, ex. if a seller wanted to ask for more money later on, the original invoice and corresponding transaction ID would at that point be rendered invalid and could not be used to receive any further payments; the seller would have to create a new contract. This technique prevents either party from profiting through cheating in a transaction. With this approach, buyers and sellers using the Pixelcash network can operate without worrying about fraud or paying unnecessary fees. They also don't have to sacrifice privacy because no third party is involved in the transaction. Furthermore, and perhaps most significantly, because there is only basic scripting involved, security concerns are minimal. While Ethereum provides more extensible support for smart contracts, that flexibility comes with a higher risk of security and privacy threats. The more code that goes into a smart contract, the greater the risk of introducing a vulnerability that could enable an intrusion. Pixelcash would serve as the foundation for a completely decentralized platform that supports a multitude of decentralized applications (dApps) and programmable functionality while offering high anonymity at the same time.

THE NETWORK

It is currently unclear as to exactly how the Pixelcash cloud itself would be maintained, i.e. who or what would provide the massive amount of storage space needed. With proper funding, large data centers like those owned by Google could be built and completely dedicated to the Pixelcash project, or storage space could be purchased from many different providers. An unconventional, yet perhaps more immediately feasible approach would be to put members of the Pixelcash community itself in charge of maintaining the cloud, by enabling them to operate through a concept that I like to call the "Decentralized Data Center", or DDC.

The DDC would be comprised of a large network of homebuilt miniature data centers, or "nodes" that are all synced with the Pixelcash network and store a certain number of jpennies based on their storage capacity. These nodes would consist of multiple external hard drives (such as this model: https://www.wdc.com/products/external-storage/my-book-duo.html) all connected

together via a USB hub (such as this model: https://www.sabrent.com/product/HB-U14P/13-port-usb-2-0-hub-power-adapter/) and plugged into the node operator's main USB port on their PC. In this particular example, the 13 port USB hub could host up to 13 of the 20 TB hard drives, making for a total storage capacity of 260 TB, enough to store data for roughly 21,666 (twenty one thousand, six hundred and sixty six) whole Pixelnotes.

Ideally, there would be hundreds of thousands of these homebuilt nodes all around the world, operated by supporters of the Pixelcash project (which could be anyone), and there could be very many variations of them; from even larger versions of the example shown here, to a single common flash drive plugged into a laptop. The PC would have a special program created by the Pixelcash developers installed on it that detects the node and syncs its available storage space with the Pixelcash network, allowing the network to interact with it as needed, i.e. by encrypting the hard drives for security (if they are not already encrypted), saving the .jpeg files to the hard drives and recursively compressing/deleting them as needed, etc. Other than keeping the PC and the node powered on 24/7 (each external hard drive would need its own power supply), no maintenance is required from the node operator.

Most likely the easiest way to achieve this in the early stages of the project would be to create a web application that allows the Pixelcash network to sync with any online cloud storage account, i.e. Google Drive, Microsoft OneDrive, DropBox, etc. and interact with it in the same manner as a physical external hard drive would in order to support the network. The specific cloud storage account(s) in question would be required to be completely clean before it could be accepted by the network, i.e. would need to be a new account(s) created specifically for this purpose, and not contain any personal items. Anyone could download/use this application, sync it with any (preferably free) cloud storage service, and that's it. It would be completely "set it and forget it". This application may also be simply run by a script that automatically registers new cloud storage accounts based on network need with cryptographically generated usernames and passwords for the purpose of preventing human access to the accounts. A script to do this task would be much more efficient than relying on human creation of accounts, as the network

would be able to "help itself" to whatever amount of storage space it needs as mining takes place. For example, if one Pixelnote is roughly 12 GB in size, and a free Google Drive account offers 15 GB of storage space, the script would open a new account for each 1.25 to 1.3 Pixelnotes mined, using two different randomly generated strings of characters as usernames and passwords, such as "BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2" (username) and "c1qar0srrr7xfkvy5l643lydnw9re59gtzzwf5mdq" (password). This method would have its own drawbacks as opposed to physical hard drives, but seeing as it would be very economical, it would work fine until the project gained enough attention for the community to want to dedicate physical storage hardware.

Although it may be more economical for the developers, this approach has the obvious drawback of relying on multiple parties becoming part of the cloud to keep the network up and running, versus simply storing all data within professionally owned and operated commercial data centers. If one of the homebuilt nodes were to be destroyed for any reason, then all Pixelcash units with data stored in that node would be lost forever, being rendered unspendable as a result of permanently losing contact with the network. In this respect, this is the only way that Pixelcash could be “burned” like Bitcoins. The ledger would be required to report these missing units.

Pixelcash wallet addresses and private keys would be long strings of randomly generated alphanumeric characters, similar to the ones automatically generated as file names for folders downloaded from external sources as a result of the MD5/SHA-1 functions of other systems (see here; http://www.tomshardware.com/answers/id-1752853/folders-long-random-words-letters.html). An example wallet address could be 1cc257829bebe0b3188a62beb7. Unlike many other cryptocurrencies, Pixelcash wallet addresses and private keys would be devoid of any universally known prefix, such as the “1” at the beginning of every Bitcoin address, which further strengthens Pixelcash’s anonymity by providing ambiguous addresses. Each Pixelcash address can begin with a different number or letter. This is simply to fit in with Pixelcash’s overall theme. Since Pixelcash is cloud based, wallet addresses and private keys act just like normal email addresses and their respective passwords would, i.e. they are credentials

generated as a pair by the platform and used as allocation and authentication tools. Because of this, it would be impossible to "burn" Pixelcash in the traditional sense, i.e. destroying it by sending it to a fictitious wallet address, because the units are hosted within a cloud, and just like email messages, they cannot reach the recipient if the wallet address is invalid, and the client software will also not allow sending without a valid wallet address.

SUPPLY

It is impossible to cap the total supply of Pixelnotes, because according to at least one source (see here: https://www.quora.com/How-many-images-are-on-the-internet/answer/Djordje-Velickovic-1?share=34a440d5&srid=pkMbP), over 1.8 billion digital images are uploaded by people to the web every single day, making for a total of roughly 657 billion digital images every year. Since the Pixelcash algorithm is programmed to reject any single image from appearing twice in the cloud, meaning each and every image in the cloud is unique, and one Pixelnote is made up of 1,000,000 (one million) unique images (jpennies), that means that a rough total of only 657,000 (six hundred and fifty seven thousand) whole Pixelnotes can be mined each year at the current natural human upload rate, however, due to the large number of images already present on the web, Pixelcash mining in the very early days would have a much larger yield than that, just like many other cryptocurrencies when they were first released.

For example; let's just guess that 657 billion images have been uploaded to the web every year since 2008, the year smartphones became popular, making for a rough total of 6.57e+12 images already on the web, which would equate to roughly 6,570,000 (six million, five hundred and seventy thousand) whole, undivided Pixelnotes that can be initially mined. Compared to other cryptocurrencies like Bitcoin, which has a limited total volume of 21 million units, that is still very scarce. It would not be the most scarce cryptocurrency on the market, however (see this article: https://themerkle.com/top-6-rare-cryptocurrencies/). These 6,570,000

initial Pixelnotes would have a combined file size of roughly 78 petabytes (PB). Only after these already present images have been mined would the mining difficulty increase and the 657,000 yearly rate apply. By this time, each miner would be competing for the same amount of jpennies, and thus each miner would receive less as a reward, just like Bitcoin.

Because it is a naturally sourced cryptocurrency, i.e. the supply is not solely dependent on the mining algorithm, there is no such thing as a "block reward" in Pixelcash mining. Miners simply earn an amount directly proportionate to their computing power. But because of this apparent scarcity, this would help Pixelcash attain a real monetary value just like many other cryptocurrencies already in existence, and it's probably safe to say that the majority of users would be using fractions of Pixelcash.

In addition, due to the fact that after being spent a certain number of times (be it hundreds or even thousands, depending on the original quality of each jpenny when it was mined, see this video: https://youtu.be/NzsbjwuWYYI), the amount of information within a single jpenny would be reduced so far that the image cannot be compressed any further; it would be a single, monotone block, and thus it would be rendered unspendable by the cloud's algorithm and automatically deleted before it could be spent again, further enforcing this notion of scarcity. A Pixelcash smart contract would detail exactly how viable a user's funds are,i.e how many times the units can be spent, the exact percentages of each jpenny in the wallet at a different stage of its lifespan (rated on condition; Mint, Good, Fair, or Poor), and would automatically allow the user the option to spend the best percentage of their wallet balance. This is synonymous with paper money becoming so worn out and unrecognizable over time that it is deemed unacceptable by merchants, banks, vending machines, etc. and must be discarded. Perhaps this limited lifespan could serve as a motive to HODL Pixelcash. Conversely, the prospect of a user's Pixelnotes being burned as a result of a hard drive being damaged or cloud account being closed could serve as a motive to spend it as quickly as possible, just like real cash typically is. In preparation for burns, however, it would perhaps be intuitive to program the network to execute automatic refunds to wallets that are affected by these circumstances, synonymous with the U.S. Bureau of Engraving

and Printing offering citizens new banknotes in exchange for mutilated ones. To prevent fraudulent refunds (i.e. users intentionally disconnecting hard drives or cloud accounts that they know contain their Pixelcash, getting a refund, then reconnecting them again to end with twice the original amount in their wallet), the network must recognize that a refund has already been given, and hold the previously lost Pixelcash in a master escrow until it is needed for distribution to other users of the network that are due refunds. Due to the fact that the demand for refunds could possibly outweigh the supply available from this method alone, the master escrow will be funded with 10% of each miner's profit, synonymous with taxes being withheld from regular income by federal or state governments. This 10% would be automatically deducted by the network before the proceeds are deposited into the miner's wallet, meaning for every 10 jpennies that are mined, 1 is taken by the network and the remaining 9 are to be kept as profit by the miner, and jpennies are deposited into wallets in intervals of 9. Any wallets due a refund would be immediately credited with these deductions. In the case of many refunds being due concurrently, the total amount available in the master escrow will be divided up evenly among all of the wallets owed until each is gradually paid off. Each jpenny held in the network escrow would be encoded into a Base64 string and saved as a .txt (text) file within a separate portion of the DDC (Distributed Data Center) cloud or a sidechain. An example of Base64 encoding can be found here: https://www.base64encode.net. Due to the extremely small file size of .txt files (converting an image of kilobytes into mere bytes), a copy of the entire network escrow could potentially be saved on a single hard drive. With this in mind, copies of the network escrow sidechain may be saved in multiple places within the DDC, making a loss of the network escrow virtually impossible. When the escrow is needed, the Base64 strings would be decoded back into .jpeg images and released into the mainchain. Likewise, jpennies recovered from a temporary network absence would be Base64 encoded and moved to the sidechain.

According to Forbes (see this infographic: https://www.forbes.com/sites/niallmccarthy/2014/09/12/how-many-years-do-us-banknotes-stay-in-circulation-infographic/), the United States banknote with the

longest average lifespan is the $100 bill, at 15 years, due to the simple fact that they are often retained for their value and only used for larger purchases. Other banknotes, like the $5 bill, last only 4.9 years. Based on this logic, it could be assumed that jpennies being consecutively used for smaller transactions (such as buying coffee, lottery tickets, small purchases online, etc.). would “wear out” sooner than whole or large portions of Pixelcash units used for more expensive purchases. It is hard to tell, however, just how long in years that individual jpennies will last being transferred from peer to peer. As the video demonstrating recursive .jpeg compression clearly showed, a .jpeg image on the highest quality settings can be compressed over 2,000 times and even still be visually recognizable. Even the unrecognizable images could be further compressed, until all pixels within the image eventually assumed the same color and grouped into a large singular block. A Federal Reserve survey found that physical currency turns over about 110 times a year, i.e. 2 times a week. However, it is possible that cryptocurrencies are traded much more often than that.

Pixelcash would be the only cryptocurrency to emulate physical fiat currency while being decentralized at the same time, save for the fact that Pixelcash, just like any other cryptocurrency, cannot be counterfeited. Also, the total supply of Pixelnotes is, simply put, completely dependent on mankind's interest in photography and art, and namely, sharing such photography and art publicly on the web. Once an image is saved in the cloud and recorded on the ledger, it can never be reused, even after being removed from circulation as a result of “wearing out” from recursive compression or by being “burned” as a result of data storage failure. The total supply is also capable of keeping itself proportionate, for example; if the number of images uploaded to the web each year increases as a natural consequence of a growing world population and more widespread internet access, but the number of people spending Pixelcash also increases for the same reasons, then there will be an equal number of jpennies eventually being destroyed as there are being mined.

Because anybody can contribute towards Pixelcash mining, not even by running the mining software or hosting a node, but by simply uploading images to the web as they normally would (and in fact every single image everyone ever

uploaded and will upload in the future will ultimately end up in the Pixelcash cloud eventually), everybody is a part of the Pixelcash project by design, and thus it is far easier for the entire world to relate to Pixelcash and take interest in it, versus other cryptocurrencies such as NewYorkCoin (NYC), which despite having one of the fastest block times of all cryptocurrencies and being designed for everyday use such as buying coffee, only seems to be majorly accepted in New York City, as could be expected.

It is possible that a third party may develop a bot programmed to automatically upload large numbers of mass produced, generic images to the web in an attempt to rig the total number of Pixelnotes that can me mined; for this reason it is important that the Pixelcash algorithm is able to identify bots and reject such automated uploads.

MORALE

The cryptocurrency community seems to believe that in order for any newly created cryptocurrency to gain traction and become a success, it must not only be unique, but solve an existing problem. One notable problem in the current cryptocurrency community is circulating supply (aka scalability) and market cap. Some cryptocurrencies like Bitcoin have a limited circulating supply, while others like Ethereum have an unlimited circulating supply. Regardless of their circulating supply, however, all cryptocurrencies currently on the market have created monetary value from nothing, quite literally “just numbers on a screen”, meaning that most, if not all other cryptocurrencies on the market are artificially generated by their own algorithms, no external input is needed, and their units of value are capable of remaining in circulation forever. Once again, Pixelcash is the only cryptocurrency that emulates physical currencies in the fact that its network limits the total circulating supply by not only generating units from a constantly changing source, but also enforces a limited lifespan that directly correlates to how many times they are transacted, giving the currency a much more “real” feel. In addition, due to their open source nature, most other cryptocurrencies can be either cloned or

hard forked. Hard forks tend to devalue the original currencies they were forked from, because Pixelcash, while also being open source, is not entirely clone or hard fork proof, but it is highly resistant, for reasons clearly explained in a second paper on this topic authored by me (see here: https://drive.google.com/open?id=1VuMEp_YYNez0uFJfFi-bnWWQ2Gf1KLXSA-1x3IJLqR0), namely the fact that all content within the Pixelcash cloud is encrypted and an insane amount of data storage would be required to support a clone or hard fork. The total amount of data in the Pixelcash network gets significantly larger everyday and by the second, so even if a team of developers were to bypass the encryption and create a hard fork of the Pixelcash network, they would need to have an amount of storage space equal to the amount of data in the already existing network immediately available at the planned time of the fork; but if the team's calculations were off, and the amount of data in the network were to exceed the amount of storage space immediately on hand, the launch of the fork would be a failure. Such a large amount of data would also likely take a very long time to transfer into the new network, considering that the original network obtained the data bit by bit via mining over a long period of time, and thus the wait time until the new fork currency could be used would likely act as a deterrent.

CLOSURE

Pixelcash; Turning the entire web into money since 2018. If you think your cryptocurrency portfolio could use more .jpeg, then you should get your hands on some Pixelcash.

Kyle D. Usher