# ZARNIWOOP WHITEPAPER

A proposal to reject the quantum resistant protocol developed on Damogran and mint Zarniwoop NFTs on the Ethereum network of Earth instead

Abstract

The strongest limit on the number of different locally distinguishable geometries is determined mostly by our abilities to distinguish between different universes and to remember our results

*Zarniwoop Vann Harl (for Infinidim Enterprises, Saquo-Pilia Hensha)*

## Introducing the Zarniwoop token concept

Satoshi Nakamoto's development of Bitcoin in 2009 has often been hailed as a radical development in money and currency, being the first example of a digital asset which simultaneously has no backing or intrinsic value and no centralized issuer or controller. However, another, arguably more important, part of the Bitcoin experiment is the underlying blockchain technology as a tool of distributed consensus, and since the introduction of Ethereum attention has now shifted to this other aspect.

Commonly cited alternative applications of blockchain technology include using on-blockchain digital assets to represent custom currencies and financial instruments, the ownership of an underlying physical device, non-fungible assets such as domain names (like zarniwoop.crypto), as well as more complex applications involving having digital assets being directly controlled by a piece of code implementing arbitrary rules or blockchain-based decentralised autonomous organisations (DAOs). Ethereum provides a blockchain with a built-in fully fledged Turing-complete programming language that can be used to create "contracts" that can be used to encode arbitrary state transition functions, allowing users to create any of the systems described above, as well as many others that humans have not yet imagined, simply by writing up the logic in a few lines of code.

Zarniwoop's collectibles are an innovative new type of digital asset, named after Zarniwoop Vann Harl because he is very important character with respect to The Guide. They are NFT's minted on the Ethereum network or compatible Layer 2 blockchains such as Polygon (Matic) and marketed by *Playbeing*, a journal devoted in roughly equal parts to galactic politics, rock music, and gynaecology. One edition carried the results of an opinion poll in which the central offices of The Guide were voted the "third hippest place" in the whole of Ursa Minor. According to this same poll, the second hippest place in the whole of Ursa Minor was the entrance lobby to the same offices, and the hippest place in the whole galaxy was the left cranium of the fugitive galactic president Zaphod Beeblebrox. Beeblebrox himself who once hitchhiked on an Acturan Megafreighter that was carrying a larger number of copies of *Playbeing* than "the mind [could] comfortably conceive". A few hours after Beeblebrox hitched a ride aboard the Arcturan Megafreighter, it unloaded five-billion tons of *Playbeing* magazine on Ursa Minor Beta causing a slight, but largely irrelevant, shift in its orbital trajectory.

The Zarniwoop collection is distinctive because it is probably the only NFT collection with a whitepaper (this document) telling you that investment in the tokens is a very silly idea.

## Zarniwoop V2 token value limitations

On your instance of planet Earth one of the curious developments in cosmology in recent years has been the emergence of the multiverse as a mainstream idea. Instead of the Big Bang producing a single uniform universe, the latest thinking is that it produced many different universes that appear locally uniform.

$$\mathcal{N}_{\text{observer}} \sim 10^{10^{16}}$$

One question that then arises is how many universes are there. That may sound like the sort of quantity that is inherently unknowable but Andrei Linde and Vitaly Vanchurin at Stanford University in California have worked out an answer, of sorts.

Their answer goes like this. The Big Bang was essentially a quantum process which generated quantum fluctuations in the state of the early universe. The universe then underwent a period of rapid growth called inflation during which these perturbations were “frozen”, creating different initial classical conditions in different parts of the cosmos. Since each of these regions would have a different set of laws of low energy physics, they can be thought of as different universes.

What Linde and Vanchurin have done is estimate how many different universes could have appeared as a result of this effect. Their answer is that this number must be proportional to the effect that caused the perturbations in the first place, a process called slow roll inflation, and in particular to the number “e-foldings” of slow roll inflation.

Of course, the actual number depends critically on how you define the difference between universes. Linde and Vanchurin have applied some reasonable rules to calculate that the number of universes in the multiverse and have totted it up to at least 10^10^10^7. A “humungous” number is how they describe it, with no little understatement. How many of these could we actually see? What’s interesting here is that the properties of the observer become an important factor because of a limit to the amount of information that can be contained within any given volume of space, a number known as the Bekenstein limit, and by the limits of the human brain.

Linde and Vanchurin say that total amount of information that can be absorbed by one individual during a lifetime is about 10^16 bits. So a typical human brain can have 10^10^16 configurations and so could never disintguish more than that number of different universes.

10^10^16 is a big number but it is dwarfed by the “humungous” 10^10^10^7.

“We have found that the strongest limit on the number of different locally distinguishable geometries is determined mostly by our abilities to distinguish between different universes and to remember our results,” say Linde and Vanchurin

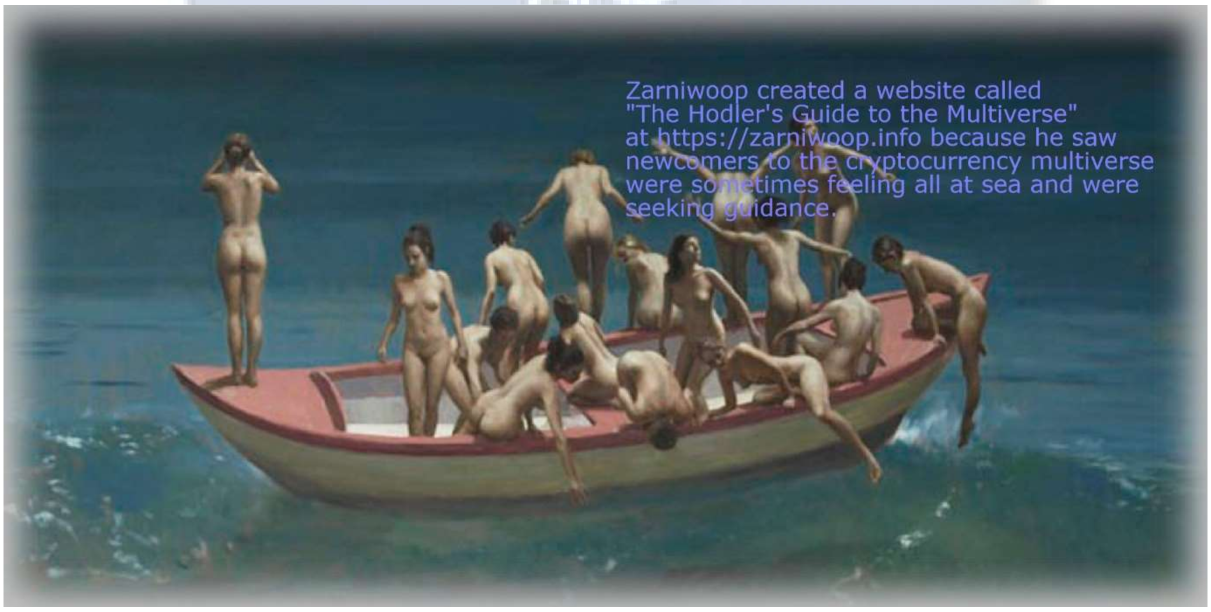
Therefore, the limit does not depend on the properties of the multiverse but on the properties of the observer. This simple principle was applied by the Sirius Cybernetics Corporation when developing Zarniwoop non-fungible tokens (NFTs) for *Infinidim Enterprises*. The maximum value of each token is a factor of that token's supply and demand for it that can never be infinite because it is limited in proportion to the multiple of the number of observers who become aware of that token and the number of home planet instances in the multiverse on which those observers exist.

If you have doubts about the analysis above then you might prefer to consider a simpler formula that says, *"the value of an NFT = utility + ownership history + future value + liquidity premium."*

The "back story" to any NFT is very important. To be of value it needs to mean something in the future to people other than the creator, its owner or their current community of followers.

Distinguishing features of Zarniwoop NFTs compared to many other NFTs on the market that could influence future value are:

a) First minting began in 2020 before ERC721 and ERC-1155 were widely known or had achieved popularity;
b) Most are extremely rare being solitary items with no copies;
c) Some represent original paintings or photographs created by Zarniwoop Vann Harl, the owner of the popular cryptocurrency guide "The Hodler's Guide to the Multiverse" (https://zarniwoop.info). He is a celebrity, although not nearly as awesomely famous and important as Zaphod Beeblebrox.
d) the AI of Marvin the paranoid android was commissioned to generate some images that required less artistic dexterity and relate to original Hitchhiker's Guide themes.


Zarniwoop created a website called "The Hodler's Guide to the Multiverse" at https://zarniwoop.info because he saw newcomers to the cryptocurrency multiverse were sometimes feeling all at sea and were seeking guidance.

## Zarniwoop NFT ownership mitigates side effects of the Infinitive Improbability Drive

The Infinite Improbability Drive is a wonderful new method of crossing interstellar distances in a mere nothingth of a second, without "tedious mucking about in hyperspace."

It was discovered by lucky chance and then developed into a governable form of propulsion by the Galactic Government's research centre on Damogran. As soon as the drive reaches infinite improbability, it passes through every conceivable point in every conceivable universe simultaneously. An incredible range of highly improbable things can happen due to these effects.

It was installed on the Starship Heart of Gold and both the ship and the drive were unveiled by then-President of the Galaxy Zaphod Beeblebrox.

The principle of generating small amounts of *finite* improbability by simply hooking the logic circuits of a Bambleweeny Sub-Meson Brain to an atomic vector plotter suspended in a strong Brownian Motion producer (say a nice hot cup of tea) were well understood. It is said that such generators were often used to break the ice at parties by making all the molecules in the hostess's undergarments leap simultaneously one foot to the left, in accordance with the theory of indeterminacy.

Many respectable physicists said that they weren't going to stand for this, partly because it was a debasement of science, but mostly because they didn't get invited to those sorts of parties.

The physicists encountered repeated failures while trying to construct a machine which could generate the *infinite* improbability field needed to flip a spaceship across the mind-paralyzing distances between the farthest stars. They eventually announced that such a machine was virtually impossible.

Then, one day, a student who had been left to sweep up after a particularly unsuccessful party found himself reasoning in this way: If he thought to himself, such a machine is a virtual impossibility, it must have finite improbability. So all I have to do in order to make one is to work out how exactly improbable it is, feed that figure into the finite improbability generator, give it a fresh cup of really hot tea... and turn it on!

He did this and managed to create the long sought after golden Infinite Improbability generator out of thin air. Unfortunately, after he was awarded the Galactic Institute's Prize for Extreme Cleverness he was lynched by a rampaging mob of respectable physicists who couldn't stand him being "a smart arse."

This page features an early image of the starship Heart of Gold which contains the Infinite Improbability Drive (as seen in the 1980 BBC TV series). Of course, the Heart of Gold actually looks somewhat more impressive as can be seen from its images in the “Hodler’s Guide to the Multiverse” website.

Side effects of using the Infinite Improbability Drive include temporary (and sometimes permanent), changes to the environment and morphological structure, hallucinations, and the calling into being of large marine mammals.

Known effects have included the creation, and spontaneous upending, of a million-gallon vat of custard, marrying Michael Saunders, the transformation of a pair of guided nuclear missiles into a Magtrathean sperm whale and a bowl of petunias, redesigning the interior of the Heart of Gold, turning Ford Prefect into a penguin, causing Arthur Dent to temporarily lose three of his limbs, transforming the desert world of Kakrafoon into an incredibly habitable oasis during a Disaster Area concert, ridding the people of Kakrafoon of their telepathy during the same concert and allowing for the discovery of Magrathea by Zaphod Beeblebrox.

The Magrathean sperm whale was the co-product of the Infinite Improbability Drive and its reality-warping field interacting with two guided missiles above Magrathea, the other outcome being a bowl of petunias. The probability of this occurring was 8,767,128 to 1 against. The whale has an existential life of discovery which lasts a minute before it hits the ground, leaving a large crater and whale remains.

The first known use of the Infinite Improbability Drive was initiated by Zaphod Beeblebrox and Trillian on the starship Heart of Gold. Its major consequence was rescuing Arthur Dent and Ford Prefect from open space, at the probability of two to the power of 276,709 to one against.

Other events that occurred, including those that took place at a time of abnormality, include:

- Lots of paper hats and party balloons appeared from a hole in the universe and drifted off in space.
- A team of seven three-foot-high market analysts came from the hole and died from a combination of asphyxiation and surprise.
- 239,000 lightly fried eggs fell out of the hole and onto the famine struck land of Poghril in the Pansel system. This caused the one surviving man of the Poghril tribe to die from cholesterol poisoning some weeks later.
- Arthur and Ford appeared to be at the seafront at Southend, Essex, UK and were passed by a man with five heads and the elderberry bush full of kippers.

Another side effect of using the Infinite Improbability Drive is that a powerful Quantum Computer (QC) could appear at any time. Where would this leave Bitcoin and Ethereum? In a post-quantum world, miners could gain an unfair advantage by mining blocks using Grover's algorithm. This provides a quadratic speed-up in the number of operations compared to a classical computer, which should lead to an increased hash rate. However, current miners use parallel computations on optimized hardware (ASICs) and it is hence difficult to predict if and when QCs will be reliable and fast enough to outperform them. To this end, you can assume that early generations of QCs will not be capable of outperforming classical miners in terms of hash rate. Furthermore, once QCs reach a state of development acceptable for mining, a quick adoption among miners can be expected, establishing an equilibrium as the network difficulty adjusts.

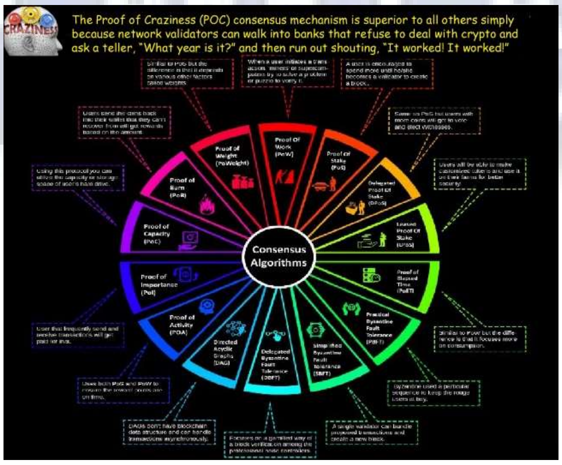
In the context of this whitepaper a key property of the Infinite Improbability Drive is that under the careful control of Zaphod Beeblebrox it can eliminate the need for *Infinidim Enterprises* to include a project road map in this proposal because our product can be delivered and manifest itself instantly on planet Earth whenever the Heart of Gold is in Earth's vicinity.

## Proof of Craziness consensus mechanism (POC)

The Proof of Craziness (POC) consensus mechanism was originally developed by Zaphod Beeblebrox as a defence against the psychiatric therapy being inflicted upon him by Gag Halfrunt. By feigning consensus with some of Gag's suggestions and attempted manipulations Zaphod was able maintain freedom of thought and discover new and innovative ways of enlarging his own ego. The very idea of being perceived as normal and fitting in to the conventional humdrum of galactic society was, of course, abhorrent to Zaphod. But the POC mechanism obscured this reality from his psychiatrist so successfully that his psychiatrist began to share some of his own egotistical insecurities and traits such as his desire to collude with another client, Prostetnic Vogon Jelz, in a plot to demolish the Earth before the Ultimate Question to Life, the Universe, and Everything could ever be found.

A thing about crazy people is they may not know they are crazy. Most "crazy" people know they are different. But they just think others are crazy. And how are they wrong? There is no "normal" only what we are told is normal. And that does not mean it is correct. Anyone can be slapped with a "mental illness" One very common mental illness to consider is believing all the bullshit that politicians, religious leaders or regulated financial advisors tell you.

POC manifested itself on the Earth as a side-effect of Zaphod operating the Infinite Improbability Drive while in the solar system at around the same time as Satoshi Nakamoto was proposing the Proof of Work (POW) consensus mechanism as established now in the Bitcoin Network. POC has since become widely adopted in the financial investment field of risking huge amounts on crudely created pixelated images (especially of cats and penguins) in the nascent and immature NFT sector of cryptocurrencies. While it is apparent that most "normal" people do not collect these artifacts, there is a delightful compulsion to accumulate these labels of independence among some visionary free-thinkers who believe blockchain technology can lead to a fairer world.


CRAZINESS
The Proof of Craziness (POC) consensus mechanism is superior to all others simply because network validators can walk into banks that refuse to deal with crypto and ask a teller, "What year is it?" and then run out shouting, "It worked! It worked!"
Consensus Algorithms
Proof Of Work (PoW)
Proof Of Stake (PoS)
Delegated Proof Of Stake (DPoS)
Leased Proof Of Stake (LPoS)
Proof of Elapsed Time (PoET)
Practical Byzantine Fault Tolerance (PBFT)
Simplified Byzantine Fault Tolerance (SBFT)
Delegated Byzantine Fault Tolerance (DBFT)
Directed Acyclic Graphs (DAG)
Proof of Activity (POA)
Proof of Importance (PoI)
Proof of Capacity (PoC)
Proof of Burn (PoB)
Proof of Weight (PoWeight)

Editor's Note: The innovative Proof of Craziness (POC) system pioneered with the NFT known as "ONE DNA" was a side effect of creation of the first prototype of the Zarniwoop NFT product and relies on a consensus that tea should be drunk hot. "ONE DNA" is not a Zarniwoop V2 NFT and it is not as rare because its supply is less limited. Indeed perhaps the only attraction of "ONE DNA" is that Zarniwoop distributes random airdrops to wallet addresses containing "ONE DNA" as identified by occasional snapshots.

## Formidable humans will flex with Zarniwoop NFTs to signal status

While anyone can enjoy an artwork displayed on OpenSea or download the JPEG, not everyone can own the original NFT. Tokenizing an asset on a public blockchain creates a way for anyone with an Internet connection to verify its authenticity and ownership. In some senses, owning an NFT of an artwork versus owning a JPEG of the same artwork can be compared to owning an original Andy Warhol versus owning a poster of the piece.

While the artwork itself may serve no purpose other than aesthetics, the act of purchasing it does. Economic literature distinguishes between two types of consumption values: hedonic and functional. Hedonic products are consumed mostly for affective or sensory fulfillment, while functional products are for utilitarian goals. Given that anyone can "consume" NFT artworks for hedonic or sensory fulfillment purposes without purchasing them, collectors may be more motivated to purchase them for utilitarian purposes.

But what is the utility? Is there a logical reason for someone to spend seven figures on an animal avatar or digital rock? Costly signaling theory would suggest the reason is to flex. Almost all animals benefit from altering the perceptions, behaviour, and psychology of others in their environment in ways that benefit themselves. This is particularly true for social animals like humans, who regularly employ different tactics like investing in costly signals to enhance their perceived attractiveness, formidability, or status.

As humans are capable of higher-order thinking, as targets or receivers of these signals, they often verify their validity before accepting them at face value. This is why flexing must be costly in order to work. Individuals who possess certain socially desired qualities will invest more in signals than those who lack them, thereby producing signals that are difficult or unreasonably costly to fake.

Purchasing NFTs of pixelated punks for hundreds of thousands of dollars a go is an example of costly signaling. Ownership or provenance cannot be faked, the costs are easily auditable, and the pieces offer little utility other than flexing. This explains why NFTs have quickly risen to become the luxury status symbol of choice for crypto's nouveau riche. After all, nothing says "I've made it" like splurging north of a million dollars on a digital picture of a rock.

The historical association of Zarniwoop NFTs with Zaphod Beeblebrox (briefly the President of the Galaxy) obviously enhances their signalling power and hence the status endowed to their owners.

## Infinidim Enterprises considered a quantum resistant protocol

Consensus participants are known as miners and upon finding a valid solution to the PoW puzzle, they are rewarded with new units of the underlying cryptocurrency and fees associated with the transactions included in the respective block. Even though quantum computers (QCs) are not yet widely available on your instance of planet Earth a recent breakthrough with a direct impact on blockchain network security is Peter Shor's polynomial time quantum algorithm that can in its subsequently generalized form break ECDSA. While more players enter this growing research area, it appears increasingly probable that powerful QCs will emerge in the near future. Although the early generations of QCs do not have enough qubits to solve problems large enough to affect Bitcoin, different alternatives for the architecture of QCs are being considered, tested and implemented so a sudden improvement in the approach might lead to a powerful QC appearing virtually overnight. In this paper, we provide an overview of the potential impacts the emergence of QCs could haveon Bitcoin. As such, we describe how a quantum-capable adversary (QCA) is in the position of stealing funds from users who have revealed their public keys. Consequently, we propose a commit–delay–reveal protocol for the secure transition from Bitcoin's current signature scheme to a quantum-resistant signature scheme, applicable even if ECDSA has already been compromised. In contrast to existing proposals, we emphasize the necessity of a substantial delay phase to provide sufficient protection against accidental and, especially, adversarial chain reorganization. We assume that the Bitcoin and Ethereum communities have agreed on and deployed a quantum-resistant signature scheme, either as a measure of precaution or as a reaction to the appearance of a (fast) QCA. Independent of quantum computing, our protocol can be generally applied to react to the appearance of vulnerabilities rooted in Bitcoin's public key cryptography. The transition can be implemented as a soft fork using a similar approach as, for example, SegWit.

Elliptic Curve Digital Signature Algorithm (ECDSA) is an implementation of the Digital Signature Standard (DSS) based on elliptic curve cryptography (ECC). The purpose of such signatures is to allow third parties to determine the legitimacy and integrity of a signed message, while the signer cannot reasonably deny the act of signing. In Bitcoin, transactions are digitally signed using ECDSA, thus securing the transfer of ownership of bitcoins. ECC is a form of public-key cryptography that uses the mathematical properties of elliptic curvesover finite fields. More specifically, to define an elliptic curve cryptosystem one chooses a curve C and a public point P on the curve. To generate a pair of keys, one chooses a random number sk as the private key and uses elliptic curve point multiplication to multiply the point P with itself sk times thus obtaining the public key pk which is itself another point on C. ECDSA or, in general, ECC, relies on the assumption that it is intractable to solve the elliptic curve discrete logarithm problem (ECDLP),which would allow for deducing the private key from the public key. Like integer factorization, ECDLP has no known reasonably fast (e.g. polynomial-time) solution on a classical computer.

Quantum computing makes use of various quantum phenomena, such as superposition and entanglement, to represent classical data in a quantum context and to manipulate it in ways that produce interpretable results. Just like the state of classical computers is made of bits, QCs use qubits that have two fundamental (basis) states (0 and 1). However, while the computation is running, the state is a linear combination (superposition) of basis states, each having an associated probability to be measured. To extract information about the state of a QC, the system is measured collapsing the superposition to one of the possible basis states. This means a QC with n qubits can represent internally the whole range of n-bit numbers and can perform calculations on all of them simultaneously; however, when measured, the state will collapse to just one of the basis states, thus returning only one of the results to the performed calculation. Instead, quantum algorithms try to make use of the underlying structure of the problem in order to amplify (or otherwise home in on) certain basis states, to increase their probability, and thus to make the result obtained repeatable and conclusive. For some problems, quantum algorithms can yield a significantly improved runtime complexity over their classical equivalents, thus offering a speed-up.

Grover's algorithm is another efficient quantum computation. It aims to solve the problem of searching unstructured data by computing with high probability a unique (or very rare) solution x for which f(x) equals v, some desired value. The time complexity is $O(\sqrt{N/t})$, where N is the size of the domain of f and t is the number of solutions. The algorithm works by first arranging a superposition of all possible input states, each having equal probability of being measured. Then, it uses some techniques to iteratively increase the probability amplitude of the states that represent the solution. Given N and t, the number of iterations after which the probability amplitudes of the correct states become maximal can be mathematically computed. In case t is unknown, there exists a scheme which will produce a solution in $O(\sqrt{N/t})$ steps. Note that it is not possible to measure the state after each iteration as this would collapse the superposition and the computation would end. Grover's algorithm is particularly interesting for mining as it theoretically offers a quadratic speed-up when guessing a nonce. However, in practice, it is believed that early generations of QCs will be slower than optimized ASIC miners.

Post-quantum cryptography is a new branch of cryptography interested in a suite of algorithms which are believed to be secure even against attackers equipped with QCs. There have been multiple proposals of cryptographic systems which are not yet broken by quantum computing. Some examples are:

(i) code-based cryptography relies on the intractability of decoding unknown linear error-correcting codes. McEliece used the algebraic properties of Goppa codes and proposed the first such system, which took his name;
(ii) hash-based cryptography is based on the security of hash functions which, as mentioned, are not drastically weakened by QCs. Merkle was the first to propose hash-based digital signatures by building on the concept of one-time signature schemes such as Lamport's signature scheme; and
(iii) lattice-based cryptography is based on the hardness of lattice problems such as approximating the closest vector problem in a lattice. For the purposes of our paper, it is important that the Bitcoin community agrees on and implements an appropriate alternative (or perhaps more than one) to replace ECC as the basis for digital signatures of transactions.

Once efficient QCs with internal states comprising many qubits are implemented, the underlying cryptographic guarantees of existing blockchains can be challenged. An attacker with a QC of about 1500 qubits can use Shor's algorithm to solve the ECDLP and compute an ECDSA private key given the public key and is thus able to plant fake transactions and perform double-spending attacks. Users should be concerned about exposing their public keys to mitigate the risk that a QCA engages in (live) transaction hijacking

Under the assumption that QCs are being employed for malicious intent by some adversary, previously revealed public keys pose a direct threat to blockchain users. As outlined, a QCA is capable of deducing the private key from a formerly revealed public key with little effort.

Regardless of how a public key is revealed, given the presence of a QCA, the owner is at risk of losing control over funds. Except for P2PK, one can prevent against the afore mentioned (so long as the QCA is slow to deduce a private key) by using addresses only once. Reusing addresses is not recommended, neither by developers nor the community, while numerous studies identifying privacy risks have been conducted. Hence, we assume appropriate protective mechanisms are already employed by the minority of Ethereum users who have bothered to read this paper and are smart enough to have decided to acquire Zarniwoop V2 tokens.

The protocol described in the paragraphs below is designed to allow such users to transition securely, if rather slowly, to quantum-resistant outputs even in the presence of a fast QCA. It is based on a simple commit–delay–reveal mechanism with a long security delay and can be deployed using a soft fork. Note that once the protocol is deployed, classic ECDSA signatures will no longer be accepted and clients will only be allowed to spend UTXOs based on the previously introduced quantum-resistant signature scheme or the transition scheme described in this paper. Furthermore, if one uses an old client and spends from a non-quantum-resistant public key, the respective funds will be lost, as the public key is revealed, and no protective mechanism can be applied effectively.

Assume a user, Arthur, is in possession of Zarniwoop V2 tokens stored in a non-quantum-resistant output, the public key of which has not yet been revealed, i.e. funded by an unspent P2PKH or P2SH output.We shall denote Arthur's public key as pk and the corresponding secret key as sk. Furthermore, assume Arthur has already generated a quantum-resistant keypair (pkQR,skQR), which will be used as a surrogate for his current keypair (during any future spending) as part of the transition. To convince the network, he is the rightful controller of both keypairs and this way regain the ability to safely spend the funds at a future date in any way he pleases (e.g. to pay user Ford), Arthur publishes a commitment H(pk|pkQR), i.e. the hash of his concatenated public keys, and leaves the funds on pk untouched for a sufficiently long security period t sec. Once the period has passed, Arthur creates a second transaction Treveal signed by skQR which consumes all UTXOs attributed to (pk,sk) and reveals both public keys pk and pkQR, proving to the network that he is the controller of both keypairs and signalling the transition of funds.

As a first step, to mark the commitment of the funds in (pk,sk), Arthur publishes the hash of both public keys pk and pkQR concatenated: H(pk|pkQR). This is achieved by creating a transaction Tcommit, which includes the hash commitment as an output. This can be achieved, for example, by using the OP_RETURN opcode, which allows us to store up to 80 bytes of arbitrary data in a transaction. Note that as Arthur cannot spend any non-quantum-resistant coins to fund the creation of the OP_RETURN, he will have to either already possess, or acquire through trade, some quantum-resistant currency units—sufficient to fund the creation of an OP_RETURN on the blockchain

After publishing the hash commitment, Arthur leaves the funds in (pk,sk) untouched for a sufficiently long security period tsec. Any further attempted use of this keypair, which would fail in accordance with the new protocol rules, puts Arthur's funds at risk of theft. A long delay is necessary to ensure no blockchain reorganization could have occurred accidentally or have been caused intentionally by an adversary. While the specific choice of delay may be subject to follow-up scientific work and discussion in the community, we propose an initial period of six months.

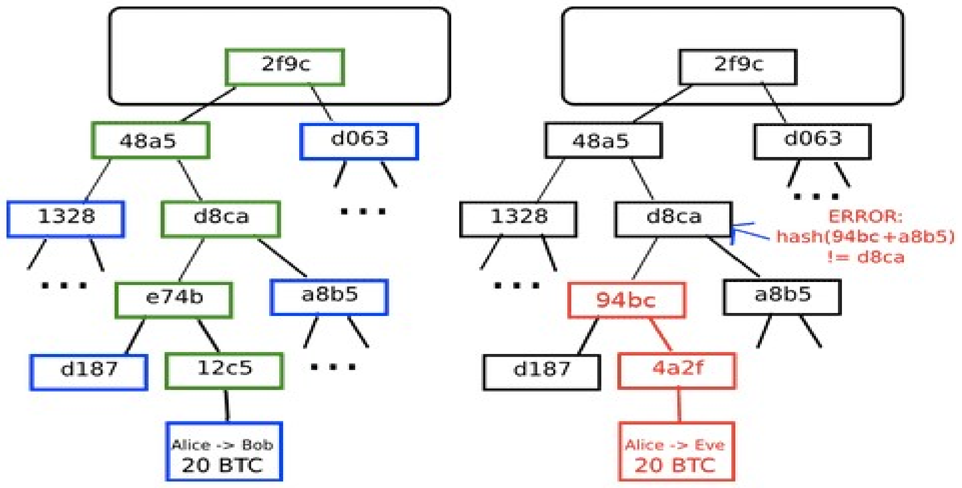
Once the security period has elapsed, Arthur proceeds to safely spend the coins to any destination(s) he pleases, by revealing his public keys pk and pkQR, proving to the network he is the rightful controller of both keypairs. To this end, Arthur creates a transaction Treveal signed by the secret key skQR of the new quantum-resistant keypair, which consumes the UTXOs of (pk,sk) and in which he

- (i) gives his ‘old’ non-quantum-resistant public key pk,
- (ii) gives the public key of the new quantum-resistant keypair pkQR,
- (iii) reveals (via Merkle-tree proof) that he has publishedH(pk|pkQR) in a transaction older than the security period tsec,and
- (iv) provides a quantum-resistant signature of the transaction against pkQR.

Miners, adhering to the new protocol rules, will then be able to verify the funds that have been committed for a sufficient period to require a new quantum-resistant public key for their eventual spending. Hence, Arthur will be allowed to spend his funds by providing a valid signature against his new quantum-resistant public key. Unupgraded consensus participants will simply believe Treveal is a normal transaction consuming the UTXOs of (pk,sk).

By the time Arthur has waited for the proposed six month delay period there is a high probability that he has forgotten that his surname is Dent never mind why he even considered doing research into the Zarniwoop whitepaper before buying some Zarniwoop V2 tokens. The marketing department at *Playbeing* declared that this could be regarded as an unattractive feature if it was made public and lobbied the board at *Infinidim Enterprises* to vote to drop the whole clever new protocol and soft fork idea in favour of making prospective Zarniwoop V2 buyers aware that if they are clever enough to find and collect Zarniwoop V2 NFTs then they are certainly well able to avoid these purely theoretical QCA thingies.

## Playbeing can't yet explain why Merkle Trees are interesting



*Left: it suffices to present only a small number of nodes in a Merkle tree to give a proof of the validity of a branch.*

*Right: any attempt to change any part of the Merkle tree will eventually lead to an inconsistency somewhere up the chain.*

A Merkle tree is a type of binary tree, composed of a set of nodes with a large number of leaf nodes at the bottom of the tree containing the underlying data, a set of intermediate nodes where each node is the hash of its two children, and finally a single root node, also formed from the hash of its two children, representing the "top" of the tree. The purpose of the Merkle tree is to allow the data in a block to be delivered piecemeal: a node can download only the header of a block from one source, the small part of the tree relevant to them from another source, and still be assured that all of the data is correct. The reason why this works is that hashes propagate upward: if a malicious user attempts to swap in a fake transaction into the bottom of a Merkle tree, this change will cause a change in the node above, and then a change in the node above that, finally changing the root of the tree and therefore the hash of the block, causing the protocol to register it as a completely different block (almost certainly with an invalid proof of work).

Of extraordinary interest to our researchers is Bowerick Wowbagger the Infinitely Prolonged who was a being who became immortal after an accident with a few rubber bands, a liquid lunch and a particle accelerator.

Wowbagger was not consulted in any way whatsoever during the composition of this white paper but *Playbeing* elected to mention him anyway in this marketing material because he is reasonably famous, probably more interesting than a Merkle tree, his picture is cool and this neatly fills up the bottom of page 12.

## Infinidim Enterprises to mint Zarniwoop V2 tokens on Ethereum

The Ethereum protocol was originally conceived as an upgraded version of a cryptocurrency, providing advanced features such as on-blockchain escrow, withdrawal limits, financial contracts, gambling markets and the like via a highly generalized programming language. The Ethereum protocol would not "support" any of the applications directly, but the existence of a Turing-complete programming language means that arbitrary contracts can theoretically be created for any transaction type or application. What is more interesting about Ethereum, however, is that the Ethereum protocol moves far beyond just currency. Protocols around decentralized file storage, decentralized computation and decentralized prediction markets, among dozens of other such concepts, have the potential to substantially increase the efficiency of the computational industry, and provide a massive boost to other peer-to-peer protocols by adding for the first time an economic layer. Finally, there is also a substantial array of applications that have nothing to do with money at all.

The concept of an arbitrary state transition function as implemented by the Ethereum protocol provides for a platform with unique potential; rather than being a closed-ended, single-purpose protocol intended for a specific array of applications in data storage, gambling or finance, Ethereum is open-ended by design, and we believe that it is extremely well-suited to serving as a foundational layer for a very large number of both financial and non-financial protocols in the years to come.

In general, there are three types of applications on top of Ethereum. The first category is financial applications, providing users with more powerful ways of managing and entering into contracts using their money. This includes sub-currencies, financial derivatives, hedging contracts, savings wallets, wills, and ultimately even some classes of full-scale employment contracts. The second category is semi-financial applications, where money is involved but there is also a heavy non-monetary side to what is being done; a perfect example is self-enforcing bounties for solutions to computational problems. Finally, there are applications such as online voting and decentralized governance that are not financial at all.

The journalists and marketing wizards at *Playbeing* can understand some of this description of Ethereum and particularly liked reading the section below about Identity and Reputation Systems but found all the drivel about quantum resistant protocol they had pasted above to be headache inducing gobbledygook.

As a final flourish they decided that the tokens would be minted on the Ethereum network with the identification Zarniwoop V2 to imply that a mature and continuously improving project had superceded an earlier but inferior version of the token.

In conclusion, *Infinidim Enterprises* has chosen to make use of Unfiltered Perception technology rather than adopt any Quantum Resistant technology at this time. This allows the Zarniwoop collection to perceive and interact with any and all planes of existence, including at least 22 spatial dimensions, multiple temporal dimensions and the entire array of the axis of probability.

If you've previously read the Ethereum whitepaper you can skip the remainder of this one because it is just covers the same material.

## Token Systems

On-blockchain token systems have many applications ranging from sub-currencies representing assets such as USD or gold to company stocks, individual tokens representing smart property, secure unforgeable coupons, and even token systems with no ties to conventional value at all, used as point systems for incentivization. Token systems are surprisingly easy to implement in Ethereum. The key point to understand is that a currency, or token system, fundamentally is a database with one operation: subtract X units from A and give X units to B, with the provision that (1) A had at least X units before the transaction and (2) the transaction is approved by A. All that it takes to implement a token system is to implement this logic into a contract.

The basic code for implementing a token system in Serpent looks as follows:

```
def send(to, value):
    if self.storage[msg.sender] >= value:
        self.storage[msg.sender] = self.storage[msg.sender] - value
        self.storage[to] = self.storage[to] + value
```

This is essentially a literal implementation of the "banking system" state transition function described further above in this document. A few extra lines of code need to be added to provide for the initial step of distributing the currency units in the first place and a few other edge cases, and ideally a function would be added to let other contracts query for the balance of an address. But that's all there is to it. Theoretically, Ethereum-based token systems acting as sub-currencies can potentially include another important feature that on-chain Bitcoin-based meta-currencies lack: the ability to pay transaction fees directly in that currency. The way this would be implemented is that the contract would maintain an ether balance with which it would refund ether used to pay fees to the sender, and it would refill this balance by collecting the internal currency units that it takes in fees and reselling them in a constant running auction. Users would thus need to "activate" their accounts with ether, but once the ether is there it would be reusable because the contract would refund it each time.

## Financial derivatives and Stable-Value Currencies

Financial derivatives are the most common application of a "smart contract", and one of the simplest to implement in code. The main challenge in implementing financial contracts is that the majority of them require reference to an external price ticker; for example, a very desirable application is a smart contract that hedges against the volatility of ether (or another cryptocurrency) with respect to the US dollar, but doing this requires the contract to know what the value of ETH/USD is. The simplest way to do this is through a "data feed" contract maintained by a specific party (eg. NASDAQ) designed so that that party has the ability to update the contract as needed, and providing an interface that allows other contracts to send a message to that contract and get back a response that provides the price.

Given that critical ingredient, the hedging contract would look as follows:

1. Wait for party A to input 1000 ether.
2. Wait for party B to input 1000 ether.
3. Record the USD value of 1000 ether, calculated by querying the data feed contract, in storage, say this is \$x.

4. After 30 days, allow A or B to "reactivate" the contract in order to send \$x worth of ether (calculated by querying the data feed contract again to get the new price) to A and the rest to B.

Such a contract would have significant potential in crypto-commerce. One of the main problems cited about cryptocurrency is the fact that it's volatile; although many users and merchants may want the security and convenience of dealing with cryptographic assets, they may not wish to face that prospect of losing 23% of the value of their funds in a single day. Up until now, the most commonly proposed solution has been issuer-backed assets; the idea is that an issuer creates a sub-currency in which they have the right to issue and revoke units, and provide one unit of the currency to anyone who provides them (offline) with one unit of a specified underlying asset (eg. gold, USD). The issuer then promises to provide one unit of the underlying asset to anyone who sends back one unit of the crypto-asset. This mechanism allows any non-cryptographic asset to be "uplifted" into a cryptographic asset, provided that the issuer can be trusted.

In practice, however, issuers are not always trustworthy, and in some cases the banking infrastructure is too weak, or too hostile, for such services to exist. Financial derivatives provide an alternative. Here, instead of a single issuer providing the funds to back up an asset, a decentralized market of speculators, betting that the price of a cryptographic reference asset (eg. ETH) will go up, plays that role. Unlike issuers, speculators have no option to default on their side of the bargain because the hedging contract holds their funds in escrow. Note that this approach is not fully decentralized, because a trusted source is still needed to provide the price ticker, although arguably even still this is a massive improvement in terms of reducing infrastructure requirements (unlike being an issuer, issuing a price feed requires no licenses and can likely be categorized as free speech) and reducing the potential for fraud.

## Identity and Reputation Systems

The earliest alternative cryptocurrency of all, Namecoin, attempted to use a Bitcoin-like blockchain to provide a name registration system, where users can register their names in a public database alongside other data. The major cited use case is for a DNS system, mapping domain names like "bitcoin.org" (or, in Namecoin's case, "bitcoin.bit") to an IP address. Other use cases include email authentication and potentially more advanced reputation systems. Here is the basic contract to provide a Namecoin-like name registration system on Ethereum:

```
def register(name, value):
    if !self.storage[name]:
        self.storage[name] = value
```

The contract is very simple; all it is a database inside the Ethereum network that can be added to, but not modified or removed from. Anyone can register a name with some value, and that registration then sticks forever. A more sophisticated name registration contract will also have a "function clause" allowing other contracts to query it, as well as a mechanism for the "owner" (ie. the first registerer) of a name to change the data or transfer ownership. One can even add reputation and web-of-trust functionality on top.

## Decentralized File Storage

Over the past few years, there have emerged a number of popular online file storage startups, the most prominent being Dropbox, seeking to allow users to upload a backup of their hard drive and have the service store the backup and allow the user to access it in exchange for a monthly fee. However, at this point the file storage market is at times relatively inefficient; a cursory look at various existing solutions shows that, particularly at the "uncanny valley" 20-200 GB level at which neither free quotas nor enterprise-level discounts kick in, monthly prices for mainstream file storage costs are such that you are paying for more than the cost of the entire hard drive in a single month. Ethereum contracts can allow for the development of a decentralized file storage ecosystem, where individual users can earn small quantities of money by renting out their own hard drives and unused space can be used to further drive down the costs of file storage.

The key underpinning piece of such a device would be what we have termed the "decentralized Dropbox contract". This contract works as follows. First, one splits the desired data up into blocks, encrypting each block for privacy, and builds a Merkle tree out of it. One then makes a contract with the rule that, every N blocks, the contract would pick a random index in the Merkle tree (using the previous block hash, accessible from contract code, as a source of randomness), and give X ether to the first entity to supply a transaction with a simplified payment verification-like proof of ownership of the block at that particular index in the tree. When a user wants to re-download their file, they can use a micropayment channel protocol (eg. pay 1 szabo per 32 kilobytes) to recover the file; the most fee-efficient approach is for the payer not to publish the transaction until the end, instead replacing the transaction with a slightly more lucrative one with the same nonce after every 32 kilobytes.

An important feature of the protocol is that, although it may seem like one is trusting many random nodes not to decide to forget the file, one can reduce that risk down to near-zero by splitting the file into many pieces via secret sharing, and watching the contracts to see each piece is still in some node's possession. If a contract is still paying out money, that provides a cryptographic proof that someone out there is still storing the file.

## Decentralized Autonomous Organizations

The general concept of a "decentralized autonomous organization" is that of a virtual entity that has a certain set of members or shareholders which, perhaps with a 67% majority, have the right to spend the entity's funds and modify its code. The members would collectively decide on how the organization should allocate its funds. Methods for allocating a DAO's funds could range from bounties, salaries to even more exotic mechanisms such as an internal currency to reward work. This essentially replicates the legal trappings of a traditional company or nonprofit but using only cryptographic blockchain technology for enforcement. So far much of the talk around DAOs has been around the "capitalist" model of a "decentralized autonomous corporation" (DAC) with dividend-receiving shareholders and tradable shares; an alternative, perhaps described as a "decentralized autonomous community", would have all members have an equal share in the decision making and require 67% of existing members to agree to add or remove a member. The requirement that one person can only have one membership would then need to be enforced collectively by the group.

A general outline for how to code a DAO is as follows. The simplest design is simply a piece of self-modifying code that changes if two thirds of members agree on a change. Although code is theoretically immutable, one can easily get around this and have de-facto mutability by having chunks of the code in separate contracts, and having the address of which contracts to call stored in the modifiable storage. In a simple implementation of such a DAO contract, there would be three transaction types, distinguished by the data provided in the transaction:

- `[0,i,K,V]` to register a proposal with index `i` to change the address at storage index `K` to value `V`
- `[1,i]` to register a vote in favor of proposal `i`
- `[2,i]` to finalize proposal `i` if enough votes have been made

The contract would then have clauses for each of these. It would maintain a record of all open storage changes, along with a list of who voted for them. It would also have a list of all members. When any storage change gets to two thirds of members voting for it, a finalizing transaction could execute the change. A more sophisticated skeleton would also have built-in voting ability for features like sending a transaction, adding members and removing members, and may even provide for Liquid Democracy-style vote delegation (ie. anyone can assign someone to vote for them, and assignment is transitive so if A assigns B and B assigns C then C determines A's vote). This design would allow the DAO to grow organically as a decentralized community, allowing people to eventually delegate the task of filtering out who is a member to specialists, although unlike in the "current system" specialists can easily pop in and out of existence over time as individual community members change their alignments.

An alternative model is for a decentralized corporation, where any account can have zero or more shares, and two thirds of the shares are required to make a decision. A complete skeleton would involve asset management functionality, the ability to make an offer to buy or sell shares, and the ability to accept offers (preferably with an order-matching mechanism inside the contract). Delegation would also exist Liquid Democracy-style, generalizing the concept of a "board of directors".

## Further Applications

**1. Savings wallets**. Suppose that Alice wants to keep her funds safe, but is worried that she will lose or someone will hack her private key. She puts ether into a contract with Bob, a bank, as follows:

- Alice alone can withdraw a maximum of 1% of the funds per day.
- Bob alone can withdraw a maximum of 1% of the funds per day, but Alice has the ability to make a transaction with her key shutting off this ability.
- Alice and Bob together can withdraw anything.

Normally, 1% per day is enough for Alice, and if Alice wants to withdraw more she can contact Bob for help. If Alice's key gets hacked, she runs to Bob to move the funds to a new contract. If she loses her key, Bob will get the funds out eventually. If Bob turns out to be malicious, then she can turn off his ability to withdraw.

**2. Crop insurance**. One can easily make a financial derivatives contract but using a data feed of the weather instead of any price index. If a farmer in Iowa purchases a derivative that pays out inversely based on the precipitation in Iowa, then if there is a drought, the farmer will automatically receive money and if there is enough rain the farmer will be happy because their crops would do well. This can be expanded to natural disaster insurance generally.

**3. A decentralized data feed**. For financial contracts for difference, it may actually be possible to decentralize the data feed via a protocol called SchellingCoin. SchellingCoin basically works as follows: N parties all put into the system the value of a given datum (eg. the ETH/USD price), the values are sorted, and everyone between the 25th and 75th percentile gets one token as a reward. Everyone has the incentive to provide the answer that everyone else will provide, and the only value that a large number of players can realistically agree on is the obvious default: the truth. This creates a decentralized protocol that can theoretically provide any number of values, including the ETH/USD price, the temperature in Berlin or even the result of a particular hard computation.

**4. Smart multisignature escrow**. Bitcoin allows multisignature transaction contracts where, for example, three out of a given five keys can spend the funds. Ethereum allows for more granularity; for example, four out of five can spend everything, three out of five can spend up to 10% per day, and two out of five can spend up to 0.5% per day. Additionally, Ethereum multisig is asynchronous - two parties can register their signatures on the blockchain at different times and the last signature will automatically send the transaction.

**5. Cloud computing**. The EVM technology can also be used to create a verifiable computing environment, allowing users to ask others to carry out computations and then optionally ask for proofs that computations at certain randomly selected checkpoints were done correctly. This allows for the creation of a cloud computing market where any user can participate with their desktop, laptop or specialized server, and spot-checking together with security deposits can be used to ensure that the system is trustworthy (ie. nodes cannot profitably cheat). Although such a system may not be suitable for all tasks; tasks that require a high level of inter-process communication, for example, cannot easily be done on a large cloud of nodes. Other tasks, however, are much easier to parallelize; projects like SETI@home, folding@home and genetic algorithms can easily be implemented on top of such a platform.

**6. Peer-to-peer gambling**. Any number of peer-to-peer gambling protocols, such as Frank Stajano and Richard Clayton's Cyberdice, can be implemented on the Ethereum blockchain. The simplest gambling protocol is actually simply a contract for difference on the next block hash, and more advanced protocols can be built up from there, creating gambling services with near-zero fees that have no ability to cheat.

**7. Prediction markets**. Provided an oracle or SchellingCoin, prediction markets are also easy to implement, and prediction markets together with SchellingCoin may prove to be the first mainstream application of futarchy as a governance protocol for decentralized organizations.

**8. On-chain decentralized marketplaces**, using the identity and reputation system as a base.