RADIANT COMMUNITY REPORT

Updated on the 13th of September 2023

Made By SHaRD. Tech Papers and Radiant4People by ANTARES
(Thanks to HippiePyro, Deathmumi, gsb, ArtOfSatoshi, CryptoNaut)

Dear All, the community of Radiant presents you the R.C.R. (Radiant Community Report): a brief summary of the project features, the achievements to date, what is to come and what can be expected in the short to mid-term. Over time this paper will be enhanced, with added sections, FAQ, links, guides and insights over Radiant (RXD). In this update you will finally find the community Roadmap, that will cover the next 2 years, until the second halving in April 2026, an entire mining section and much more. Radiant is a free, layer 1**(*)** & GPU mineable, open source peer-to-peer proof of work (SHA512-256D) network to read and write decentralized applications. Not meant as investment, precisely as Bitcoin itself. This document, in an ever-evolving "Work in progress" state, will be updated constantly from the Radiant Community so, given its open access to everyone, please feel free to contribute.

(*): Radiant is not a Layer2 token on ethereum so it is NOT on metamask, by consequence)

Join RXD on DISCORD: https://discord.com/invite/radiantblockchain

Radiant Whitepaper: https://radiantblockchain.org/radiant.pdf

Please also have a look at the official "Radiant4people" Technical WIKI and Community Landing Page, in development. "Web created by the community for the community, everyone can participate in the collaborative Github account. If you have an idea for Radiant, feel free to share it!"

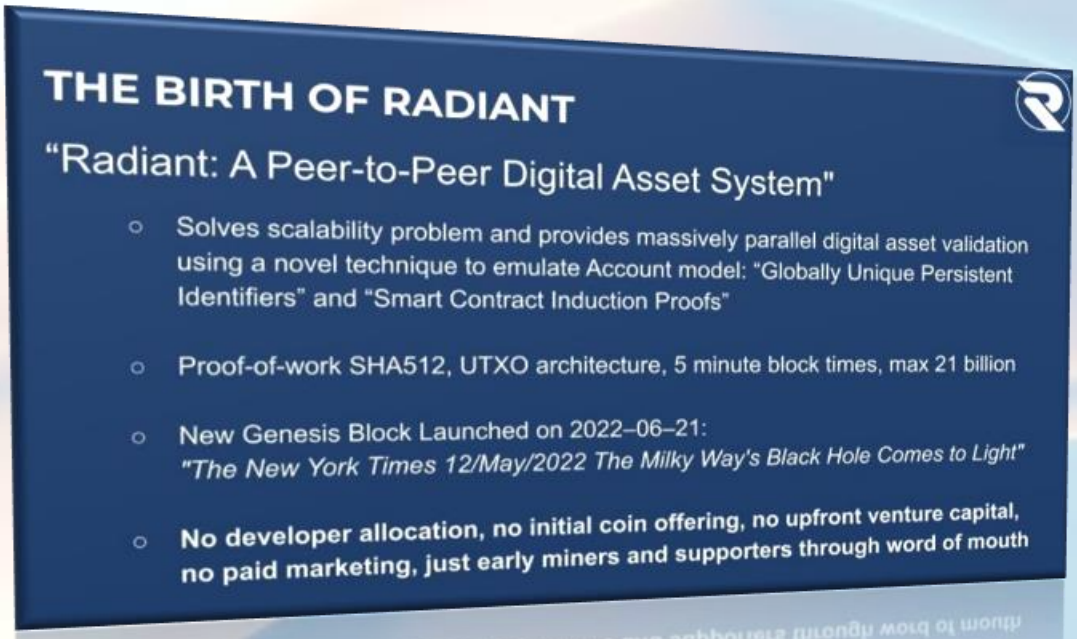
https://radiant4people.com/

R	*No VCs, No Hidden Manipulators*
	No Initial Fundraise, No ICO
	No Executives, No Leaders
X	*No premine (Clean Genesis block)*
	No FAKE metrics, No FAKE trading bots
	No Stakeholders, No Staking
D	One-Hundred-Percent POW, Deflationary
	Community Led & Community Owned

EXPLORER LINK: https://radiantexplorer.com/

GENESIS BLOCK: https://radiantexplorer.com/block/0000000065d8ed5d8be28d6876b3ffb660ac2a6c0ca59e437e1f7a6f4e003fb4

The Radiant Developers, August 11, 2022 - https://radiantblockchain.org/



WHERE TO BUY ? - CLICK ME

LEGEND (clicking LEGEND you come back here, hyperlinks below)

RADIANT LINKTREE (#ALL-THAT-YOU-NEED)

WEBSITE: https://linktr.ee/radiantblockchain

DISCORD CHANNEL: https://discord.com/invite/radiantblockchain

HIGHLIGHTS OF THIS UPDATE

- Radiant 2 years Roadmap
  - ❖ Community roadmap, initiatives to be funded by the community members after voting sessions, and community contributions
- Technical updates, in collaboration with RXD devs
  - ❖ The token standards will be the next step of Radiant, we are there
- New mining chapter
  - ❖ Addition of the list of hashrate for every GPU model
  - ❖ Addition of mining guides for Windows and Ubuntu
  - ❖ Addition of pool setup guides
  - ❖ Addition of wallet installation, for the mining process
  - ❖ Renderings of Mining FARMS Rigs
- New marketing chapter & Initiatives, new development contest chapter
- New 4 Step short term priority chapter, new brief tokenomics chapter
- Radiant History chapter expanded
- General improvements, with hyperlinks, general bugfixes & visual enhancements
- Addition of community Guides & way more ... (Photonic !)

Have you ever seen so much for a crypto project?

CHAPTER I - Radiant Blockchain System Design

Radiant is a peer-to-peer digital asset system with unbounded scaling as a UTXO-based blockchain with all the flexibility and power of account-based blockchains.

- ✓ Network Name: Radiant
- ✓ Network Abbreviation: RXD
- ✓ Mining Algorithm: SHA512/256 Proof-of-work
- ✓ Block Time: 5 minutes
- ✓ Initial Block Size: 128 MB, designed to achieve 10GB+
- ✓ Block Reward Schedule: 50,000 RXD per block
- ✓ Block Reward Halving: 2 years
- ✓ Maximum Supply: 21,000,000,000 RXD
- ✓ Decimal Places: 8
- ✓ Launch Date: 2022-06-21 02:42 UTC

- ❖ 100% POW, "NOT A (SEC) SECURITY by definition", 2 year monolithic halving schedule
- ❖ SHA 512256D Algorithm, best balancement between GPU and FPGA computational power, while being core-heavy. Low memory requirements for the maximum decentralization
- ❖ Hard Cap set at 21B (21.000.000.000) Coins. As a tribute to Bitcoin, 1000 zeros have been added to the total base and reward. Radiant, RXD, is the gas for Smart Contract, NFT or FT futures
- ❖ Instant TX. With the 0-Conf option configured by default, it is possible to perform unlimited transfers instantaneously. Wonderful for the future NFT applications, and gaming developments
- ❖ 1000 TX/second with 256MB default block sizes and 5 min average time. They can scale up much higher if required, without any problem. Radiant is extremely scalable by design, so very low cost TXs with NFTs are guaranteed, and with the 0Conf, lightning fast finalization too
- ❖ SMART CONTRACT Native Capability
- ❖ INDUCTION PROOF - The Induction Proof system makes it possible to efficiently compose outputs in any manner, without compromising the inherent parallelism and scalability characteristics of the UTXO based architecture. With the Induction Proof it is not necessary to have EVM (Ethereum Virtual Machines*) to manage Smart Contract and to have native NFT and FT tokens. All is managed from any node

*: The EVM is the acronym of “Ethereum virtual machine”, sort of a virtual computer, used to create and execute SC (Smart Contract) in order to create dAPPS (Decentralized Applications). Used on the Ethereum Network”. This is not needed on Radiant

- ACCOUNT EMULATION - With the novel Induction Proof technique, it is now possible to create globally unique identifiers and therefore implement accounts in the UTXO based architecture. The best of Both worlds in emulating the account models while also providing the coin (UTXO) model that offers massive scale and parallelism
- Layer 1, TURING Complete*
- 256MB Default Block Size. A safe block size to handle high throughput to start, designed to achieve 10+GB and beyond in the coming decades. As Bitcoin, enhanced on a log scale.
- Network Fuel: the fuel of the network is the Radiant Unit (RXD). A small amount of RXD is used to pay transaction fees to miners for processing transfers and for the execution of the smart contract

*: “Turing Complete refers to a machine that, given enough time and memory along with the necessary instructions, can solve any computational problem, no matter how complex. The term is normally used to describe modern programming languages as most of them are Turing Complete (C++, Python, JavaScript, etc.)”

Radiant is Turing Complete but it doesn't have loops. Loops have to be unrolled to the maximum number of iterations required. This is a lot simpler and safer. It does increase transaction size and make some logic harder to implement, but it also makes script execution predictable so fees can be calculated per byte. So there are some limitations, but technically it can compute everything Ethereum can. Radiant introducing loops would mean the impossibility to calculate fees per byte. In this case Radiant would need gas, with all the consequences that we can remember in the bull run, having to pay hundreds of dollars-worth of Ethereum to move our Coins around, with the Ethereum L2 chain totally un-usable.

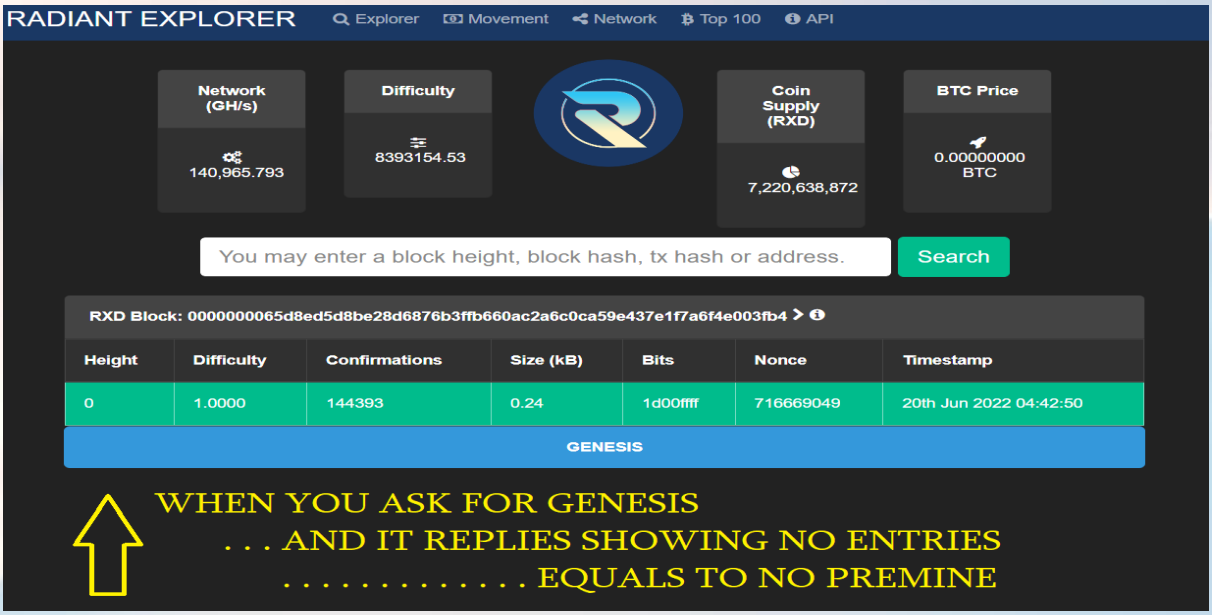
https://en.wikipedia.org/wiki/Loop_unrolling

COINBASE LAUNCH 20/06/2022 02:42:50 GMT

- [ ] New blockchain, block height 0
- [ ] Block Time: 300 seconds / 5 minutes
- [ ] Halving: 2 years (every 210,000 blocks)
- [ ] Subsidy emission: 50000 per block
- [ ] Original DAA algorithm from block height 1
- [ ] Switch to ASERT DAA on 09/07/2022 22:00:00 GMT+0000
- [ ] Total Coins: 21 Billion, with 8 decimal places for each
- [ ] Maturity: 100 Confirmations
- [ ] Prefix: RXD addresses start with "1"
- [ ] Ports: 7333 (P2P) / 7332 (RPC)
- [ ] PoW Algorithm: sha512_256d (double sha512/256)

Genesis Hash: 0000000065d8ed5d8be28d6876b3ffb660ac2a6c0ca59e437e1f7a6f4e003fb4

https://explorer.radiant.ovh/



Radiant embodies what one would expect from Bitcoin itself. There is no central authority, no venture capitalists ready to dump on your valuable holdings, no false promises, and quite simply: CODE. This code is designed to be used, shared, and improved, forming the basis on which everyone can build exceptional personal or company projects, thanks to its scalability and revolutionary intrinsic potential.

Among the top 10 coins on CoinMarketCap, 80% are premined: Ethereum, Ripple, Cardano, USDT, BNB, Polygon, and Solana. Ethereum itself launched with a 67% premine, and the recent Ironfish had around 30%, accompanied by a large team allocation common among many billion-dollar market cap crypto projects. The SEC is correct in asserting that proactive regulation is needed in the sector, focusing on protecting customers and emerging markets. Over the years, countless scams have damaged market sentiment, and there have been numerous centralized operations that exploited absurd marketing campaigns with celebrities, shady television promoters with dubious credentials, and false promises of wealth — all with the sole intention of making quick profits without offering tangible value, content, or innovation. Like Bitcoin, Radiant is a product of the growing trend of digitalizing services and products, specifically in the financial domain. It also possesses a technological core that enables digital certification of any kind, with exceptional scalability and instant transactions, much like Bitcoin prior to the "Replace by Fee" Bitcoin Core update. If 90–95% of crypto projects are considered securities, Radiant and Bitcoin are not. Should the SEC or the U.S. government ever pass a law banning or severely restricting the use and transfer of crypto securities, Radiant would benefit, just like Bitcoin, due to its pure intentions, honesty, and transparency, as any true community-led project should.

Radiant is us, Radiant is you; a survivor in a sea of sharks, a beacon of light in the shadows, and a shepherd guiding the pack through uncharted territory. Radiant does not fear the absence of third-party investments and openly classifies itself as something other than an investment. Radiant core values of moral honesty aim to protect its users from potential wrongdoings by regulatory entities, acting against unregulated markets due to the temporary complexity of regulatory processes and understanding. Decentralization is the key and each user, each node, each voice and each person are the richness of the project: the final-user contribution.

RADIANT ALGO

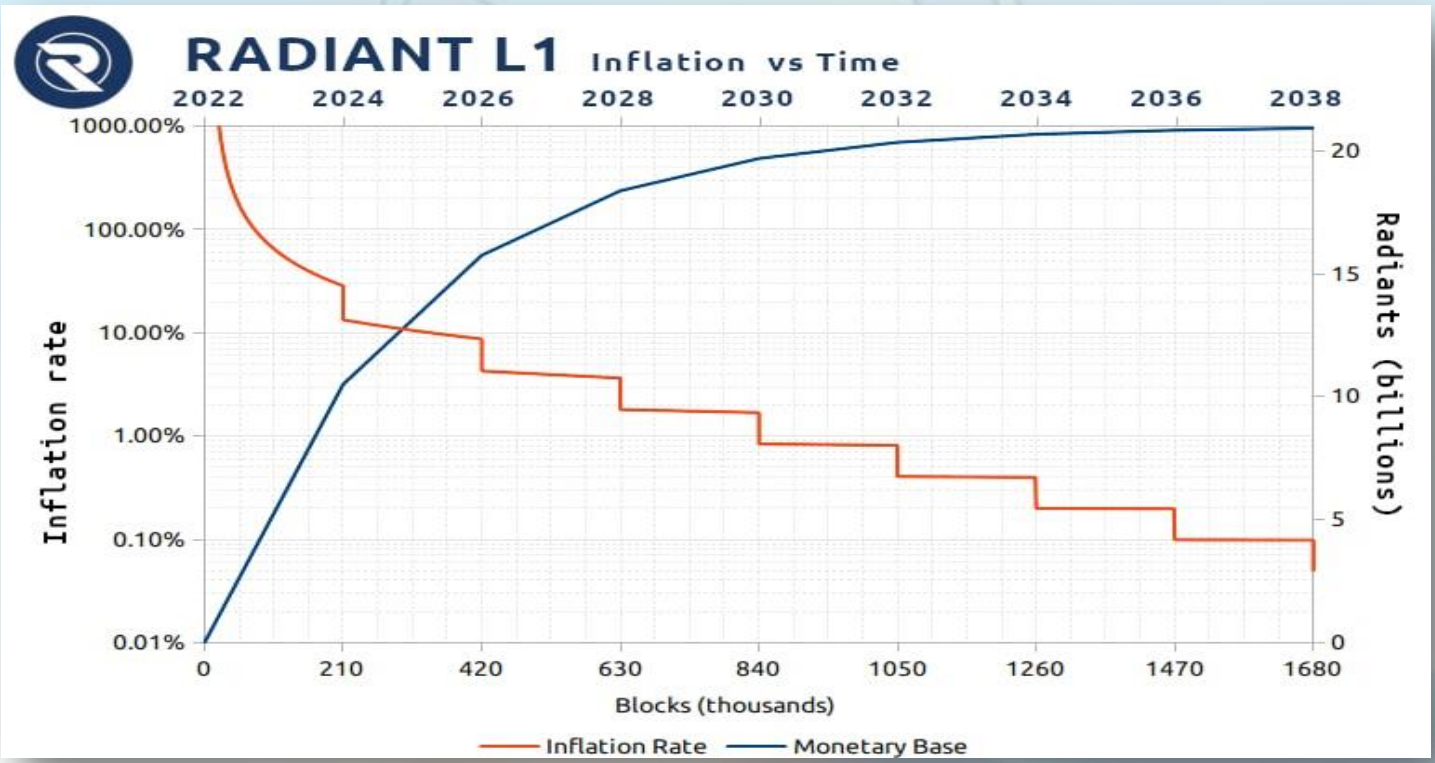
SHA-256 Performance	Cycles	
	Total	Per Byte
SHA-256 Update (Compact C code)	1863	29,11
SHA-256 Update (OpenSSL unrolled asm Code)	1166	18,22
SHA-256 of a 1024 bytes message (Compact C code)	33757	32,97
SHA-256 of a 1024 bytesmessage (OpenSSL unrolled asm Code)	19769	19,3

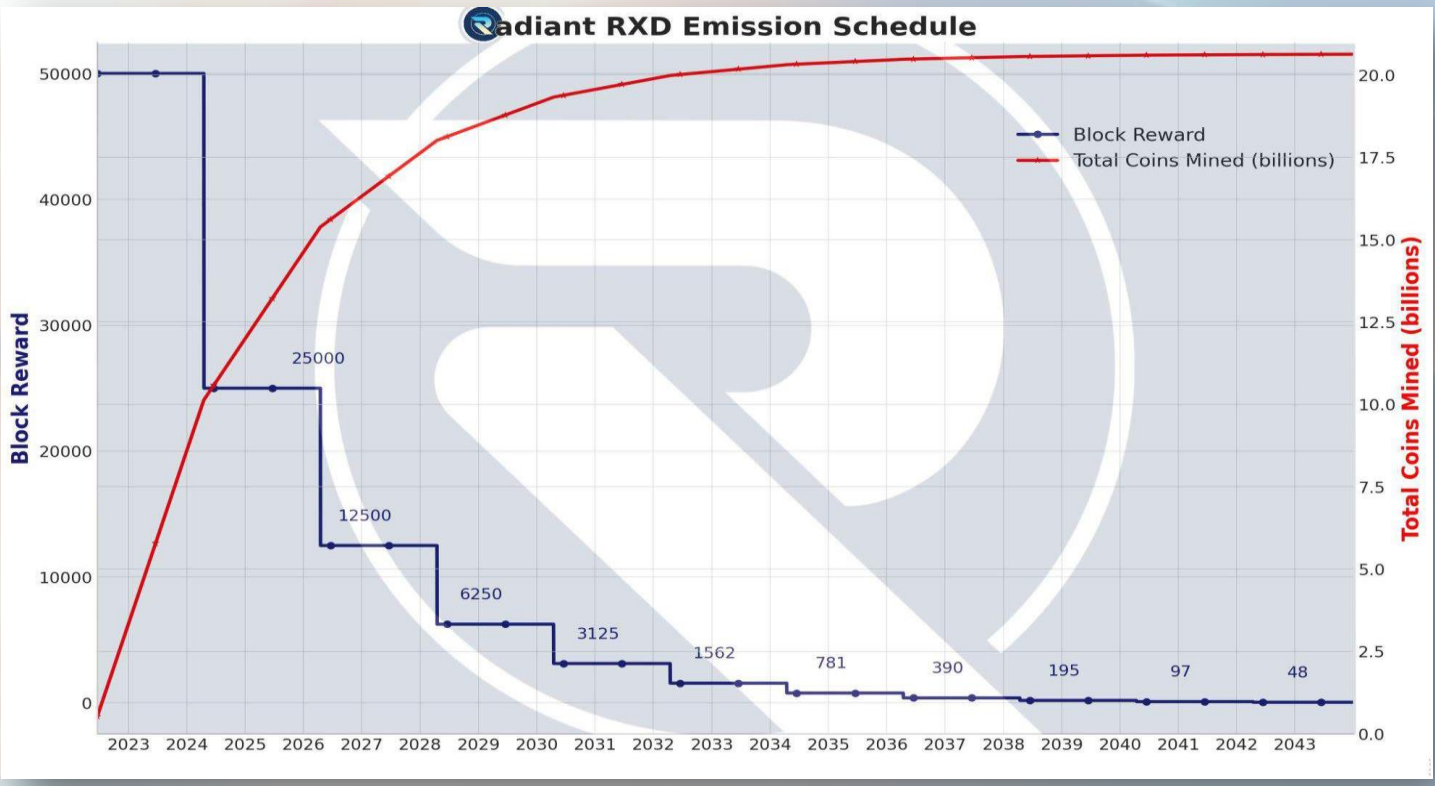
SHA-512 Performance	Cycles	
	Total	Per Byte
SHA-512Update (Compact C code)	2473	19,32
SHA-512 Update (OpenSSL unrolled asm Code)	1483	11,58
SHA-512 of a 1024 bytes message (Compact C code)	20928	20,43
SHA-512 of a 1024 bytesmessage (OpenSSL unrolled asm Code)	13392	13,07

When comparing apples-to-apples implementations, SHA-512 performs ~50% more efficiently SHA-256. Even when comparing "best" against "worst" implementations, the SHA-512 performance is within ~6% of SHA-256 implementation.

Source: https://eprint.iacr.org/2010/548.r

Radiant Coin Distribution Chart – 100% POW





1.1 Abstract

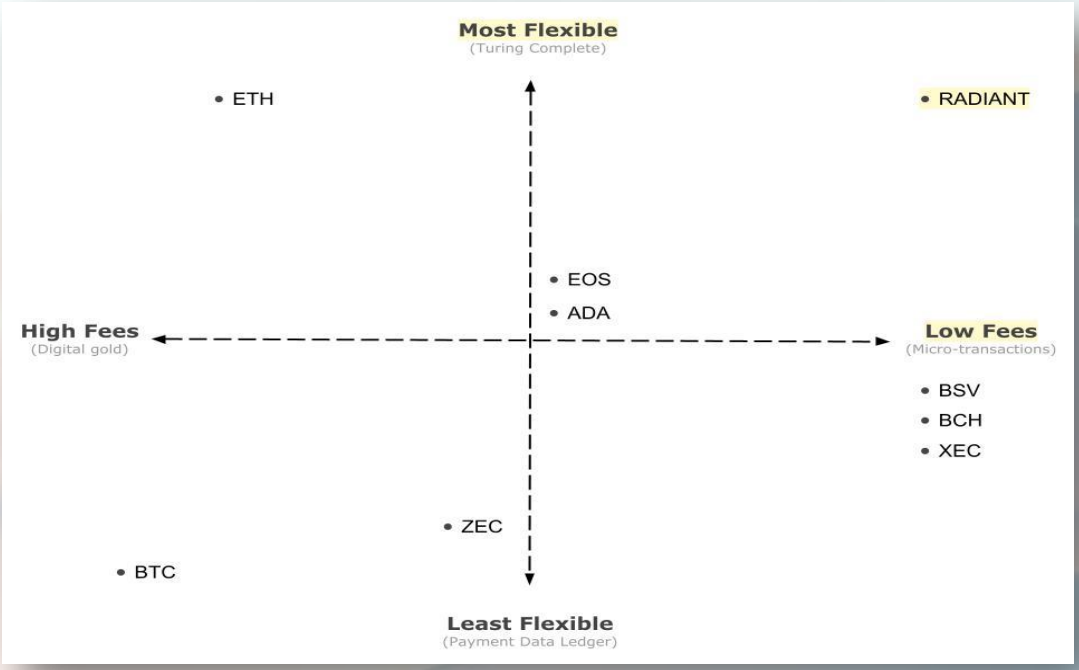
The Radiant network is a peer-to-peer digital asset system that enables direct exchange of value without going through a central party. The original Bitcoin[1] protocol provides what is needed to create a peer-to-peer electronic cash system, but lacks the ability to verify transaction histories and therefore cannot be used to validate digital assets. Digital signatures and output constraints provide part of the solution, but the main benefits are lost if a trusted third party is still required to validate digital assets. The Radiant network itself requires minimal structure, and operates similarly to the Bitcoin network in time stamping transactions into an ongoing hash-based chain of proof-of-work. We introduce two techniques to validate digital assets using a general purpose induction proof system that operates in constant O(1) time and space. The induction proof system makes it possible to efficiently compose outputs in any manner, without compromising the inherent parallelism and scalability characteristics of the UTXO based architecture. Users can leave and rejoin the network at will and be assured of the integrity and authenticity of their digital assets.

1.2 Introduction

Commerce with blockchains and digital ledgers has come to rely on issuers and custodians serving as trusted third parties (sometimes referred to as "bridges", "oracles", "secondary layers") to authenticate digital assets and process electronic payments. While the system works well enough for electronic payment-like transactions, it still suffers from the inherent weaknesses of the trust based model for more advanced usages of the blockchain. The high costs of transactions associated with Ethereum Virtual Machines (EVM) based blockchains is due to the limited block space and the inherent limitations of the account based model of processing. What is needed is an electronic payment system that can also act as a digital asset management system with the performance characteristics of an unspent transaction output (UTXO) blockchain architecture, with the flexibility of an account based blockchain. In this paper, we propose a solution to the problem of blockchain scaling using two novel methods which, independently, provide a general induction proof system capable of authenticating digital assets, emulating account based blockchains, while maintaining the performance characteristics of a UTXO based blockchain such as unbounded scale and parallelism.

The original Bitcoin[1] protocol provides what is needed to create a peer-to-peer digital asset system, but lacks the ability to verify transaction histories and as a result cannot authenticate

digital assets. Blockchains such as Bitcoin Cash (BCH) and Bitcoin Satoshi Vision (BSV) attempt to authenticate digital assets via trusted third parties called "oracles" which indexes the relevant transactions. Such solutions, however, prevent the possibility of advanced blockchain contracts since a trusted custodian is required. In order to solve the problem of digital asset authenticity, without using central parties, we introduce two novel methods that operate in constant O(1) time and space. The additional programming instructions creates a general purpose induction proof system. Users and applications need only to verify that the latest digital asset transfer is accepted into a block. Radiant is the first unspent transaction output (UTXO) blockchain that solves the key problems that prevented the development of advanced contracts on other blockchains such as Bitcoin, Cardano, and Dash. This breakthrough design revolutionizes what we imagined to be possible with blockchains; Radiant is a Turing Complete high performance layer one blockchain with no need for secondary layers.

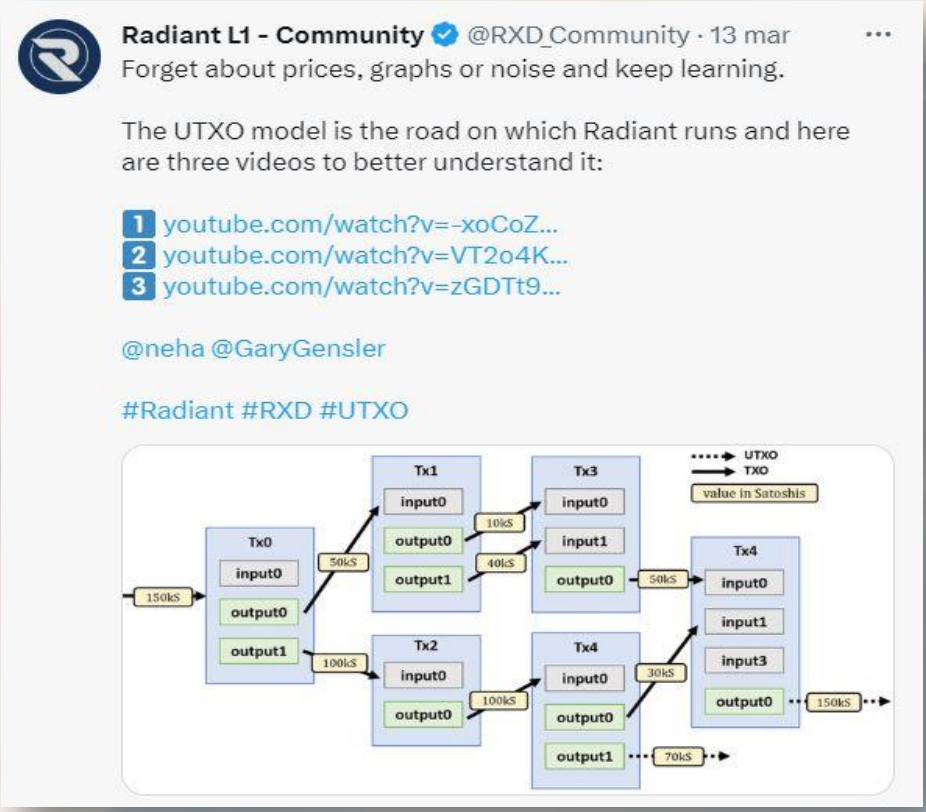


Positioning of Radiant relative to popular blockchains

1.2.1 Programming in Radiant

Radiant uses a programming language called script inherited from Bitcoin, but adding advanced functionalities that allow it to be Turing complete: "RadiantScript". To understand it well it is necessary to make a leap in time to the beginnings of Bitcoin.
Bitcoin Script / OP_codes / OP_PUSHINPUTREF
The key to everything.

DO YOU WANT TO KNOW MORE? FOLLOW THE WHITE RABBIT
https://radiant-community.medium.com/programming-in-radiant-5f7c4b5670db



1st LINK

2nd LINK

3rd LINK

1.3 Problems, that Radiant aims to solve

There are three problems which make it impractical to use unspent transaction output (UTXO) blockchains as a general purpose digital ledger. The first problem is the ability to arbitrarily constrain the spend conditions — or forward conditions on all descendant transactions. The second problem is how to efficiently authenticate transaction outputs to ensure they originate from a valid genesis transaction — this is an essential requirement for many programs, especially to emulate accounts and create fungible tokens. The third problem is coordination and collaboration between contracts — precise control of message passing between transaction outputs. We will show that all three problems can be solved without compromising performance or the scalability of the UTXO-based blockchain model.

1.4 Contract Constraints

The first obstacle to programming with an unspent transaction output (UTXO) blockchain was a misunderstanding of Satoshi Nakamoto original design and programming codes available in the original Bitcoin protocol. It is not generally acknowledged but the original Bitcoin blockchain had all of the programming codes necessary for Turing Complete [2] smart contracts. The necessary programming codes were removed from the protocol in the BTC upgrades of 2015. The method to impose constraints on spending conditions is to restore all of the original programming codes from Bitcoin and to provide a method to inspect the current transaction context. There are two ways to inspect the current transaction as a type of introspection. The first way is to push the Signature Hash (known as the "SigHash Preimage") onto the stack and use a temporary private key to generate a signature and then apply the OP_CHECKSIG operation to validate that the expected SigHash Preimage for the current transaction is valid. The second way is to provide native introspection programming codes that push the relevant transaction component onto the stack for use in the unlocking script. The key difference with a UTXO blockchain is there are no loops in the programming codes. However in practice any repetition can be simulated with unrolling the loop operations and replicating the logic for the necessary maximum number of repetitions. In this manner, UTXO blockchains can avoid any concept of "execution time cost" and instead estimate the execution cost by using only the transaction script size. For this reason, it is recommended that UTXO blockchains have a sufficiently large maximum transaction size,

such as 2 megabytes or more to be able to accommodate any use cases that may need dozens or hundreds of loop iterations.

1.5 Contract Persistent Identity

An electronic coin is defined as a chain of digital signatures. A coin begins at a genesis transaction called a "coinbase" transaction. To transfer a coin, the owner of the unspent output signs the coin with their private key and locks the tokens in a new output which is associated with the public key of the recipient. At each transaction a new transaction identifier and output index is used, which is globally unique. The concept of a "wallet balance" for a user is the sum total of the nominal token units controlled by the user for the unspent outputs for their corresponding public keys. Each coin in essence is uniquely identified by it's most recent unspent output.

There is no inherent concept of an "account" or "coin identity". In unspent transaction output (UTXO) blockchains the native token unit is the only class — or type and therefore a unique persistent coin identity is not necessary. It is sufficient to have a different UTXO identity to enumerate the coins that can be spent. However, if we wish to create a different class of tokens, in other words to "color" the native tokens to represent shares, points or any other enumerable type, then we need a way to represent and efficiently validate token class membership. The term "colored coins" have been used to describe an overlay network which mints tokens from a special genesis or minting transaction — similar to the native coin is emitted from a coinbase.

A custom (or colored) digital coin is defined as a chain of digital signatures anchored at a user defined genesis output. Users may mint or create a custom coin issuance by depositing the desired number of native token units at the output and designating it as a coinbase with 36's 0x00 null bytes as the first push data of the output. The contract logic is constrained such that the subsequent spend of the output must embed the outpoint (transaction id, output index) of the genesis transaction into the first push data (where the 36 0x00 null bytes were in the genesis transaction) for the entire lifecycle of the colored coin.

Following this convention combined with the contract constraints, we can see that this technique effectively "colors" the native token and can be identified unambiguously. Additional logic can be added according to the application needs such as how the coins are redeemed or returned back to their native token units.

For example the issuer can perform the operation or the token holder can "melt" out the native token unit and effectively destroy the color classification. The technique of embedding the genesis output forms a globally unique identifier sometimes referred to as asset identifier (or assetId for short) or contract identifier (or contractId for short) that may now be used to identify the coins that belong to that coin class.

This identity will form the basis of the advanced usages outlined below.

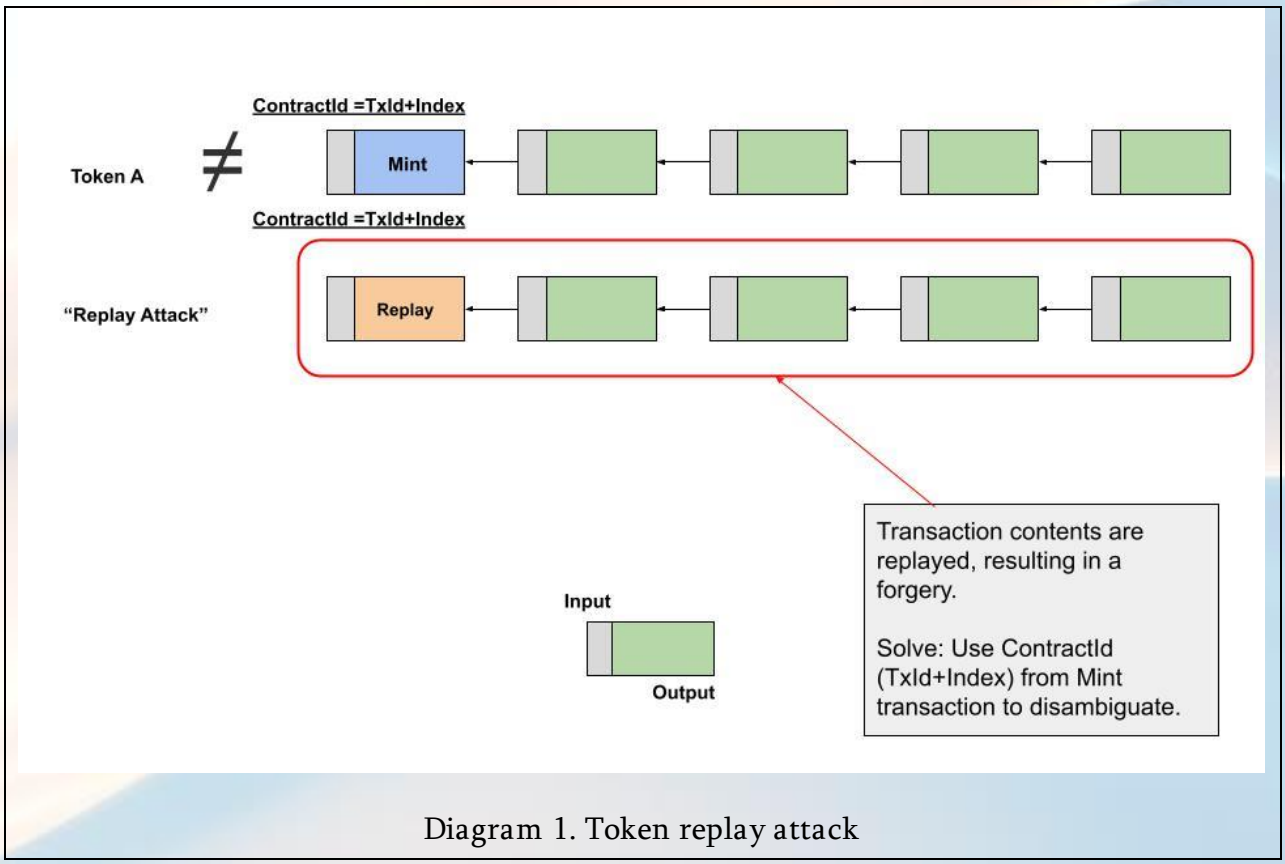


Diagram 1. Token replay attack

There is an outstanding problem however: How can spending transactions ensure that only coins descended from the rightful genesis transaction can be spent and not passed off into spending a forgery that was merely copied?

1.6 Contract Traceability & Authenticity & Examples

Recall that an unspent transaction output (UTXO) has no persistent identity, but we can give a persistent identity by following the rule that a user may designate some transaction as a genesis minting event, where the outpoint stands in as the assetId or contractId. However, using this convention it is not sufficient because an attacker can copy one of the intermediate transaction spends and begin a new (albeit forgery) chain of signatures to spoof a coin class and pass it off. Any spending transaction are unable to differentiate between a real output that originated as a valid descendent versus the forgery from a false copy. What is required is a way to enforce global uniqueness that is unobtrusive and efficient to verify inside a spending script.

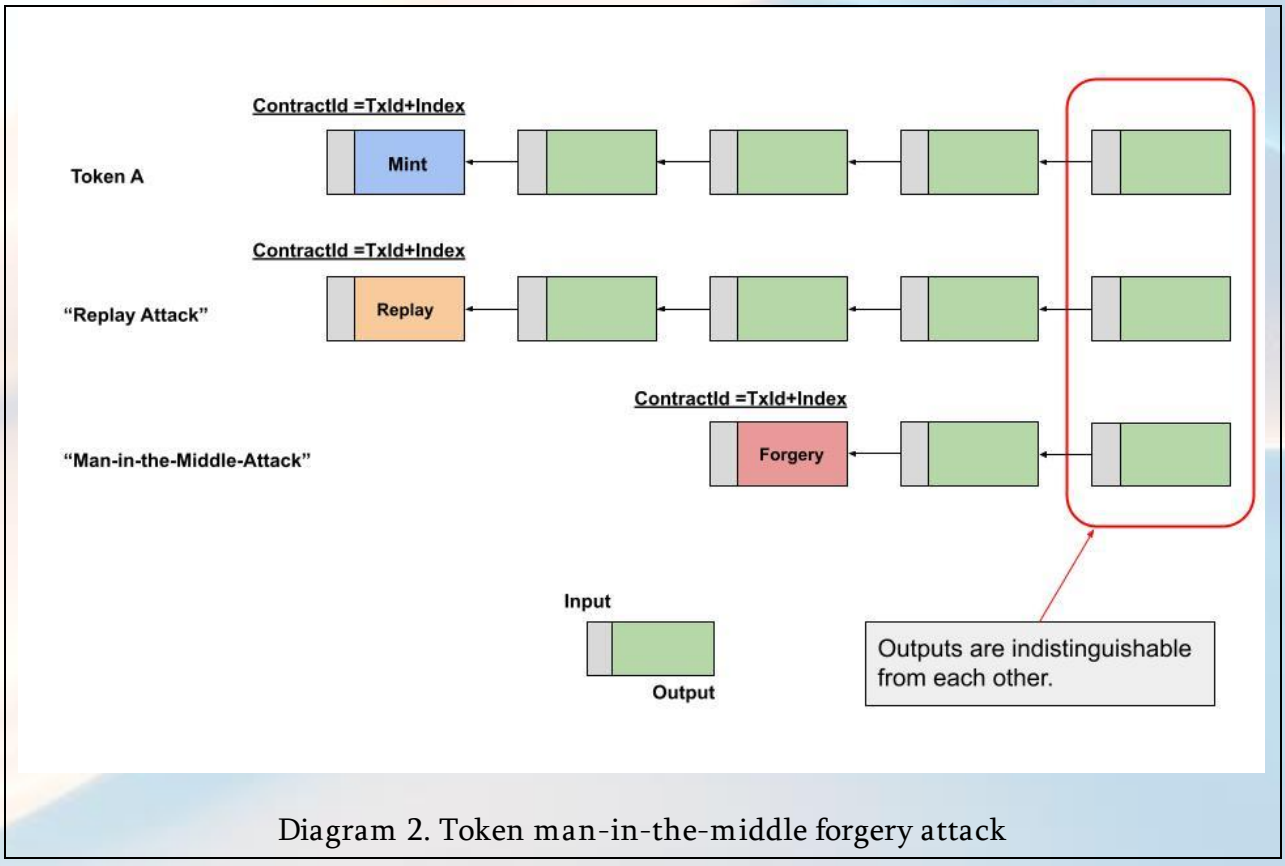


Diagram 2. Token man-in-the-middle forgery attack

OP_PUSHINPUTREF / Push reference

We define the programming operation code (OP code) OP_PUSHINPUTREF <hash> is defined as valid accordingly:

1. An OP_PUSHINPUTREF may appear only in an output and requires exactly 36 bytes immediately after that is treated as a push onto the stack in interpreter context.
2. The transaction containing an output with a OP_PUSHINPUTREF is valid if and only if the provided argument is equal to one of the inputs' outpoints being spent or at least one of the inputs' output locking script bytecode also contains the same OP_PUSHINPUTREF argument value.

The only way an OP_PUSHINPUTREF can first appear in an output is if the first occurrence is equal to one of the inputs outpoints being spent. In the case of using the above "Persistent Contract Identity", this corresponds to the transaction that contains the 36 0x00 null bytes signifying a genesis minting coinbase for a custom (colored) coin class.

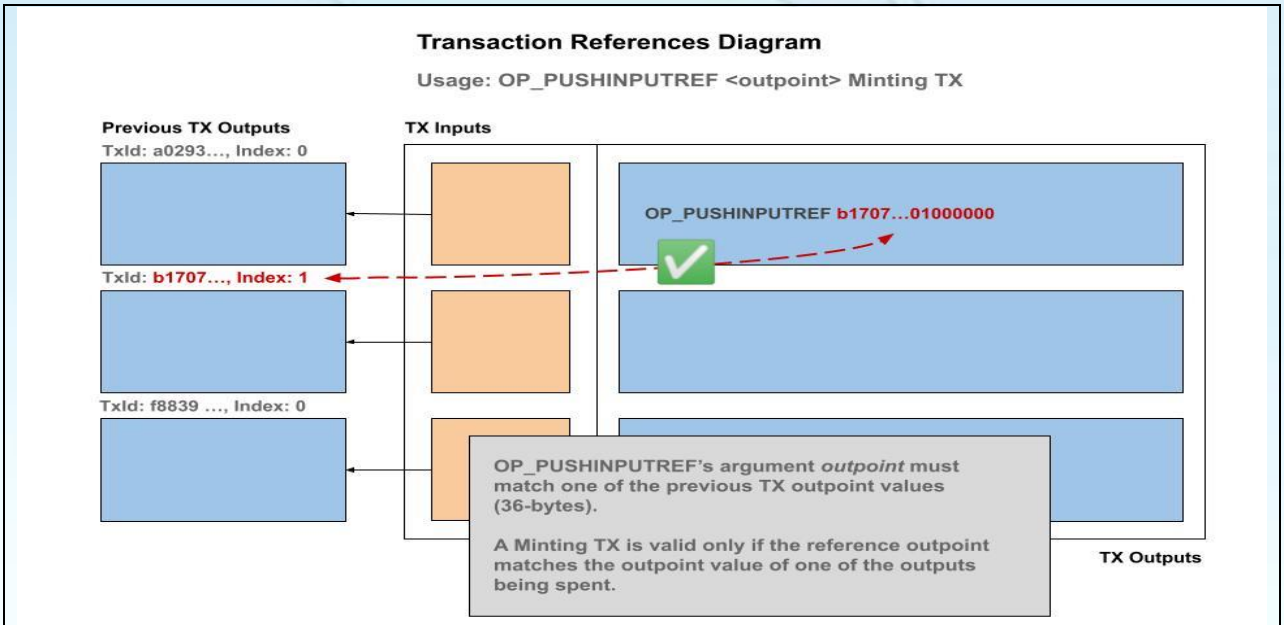


Diagram 3. Minting transaction OP_PUSHINPUTREF reference must match outpoint.

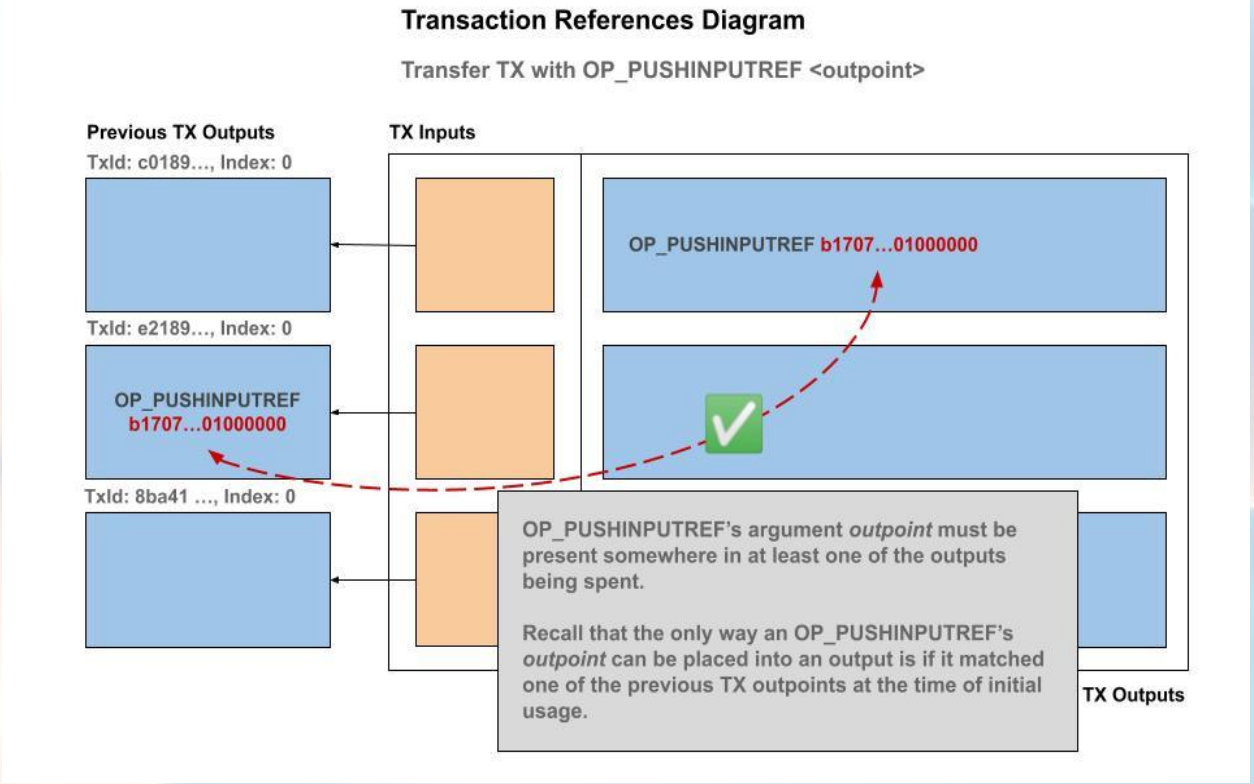


Diagram 4. Transfer transaction OP_PUSHINPUTREF reference must match one of the previous outputs scripts being spent.

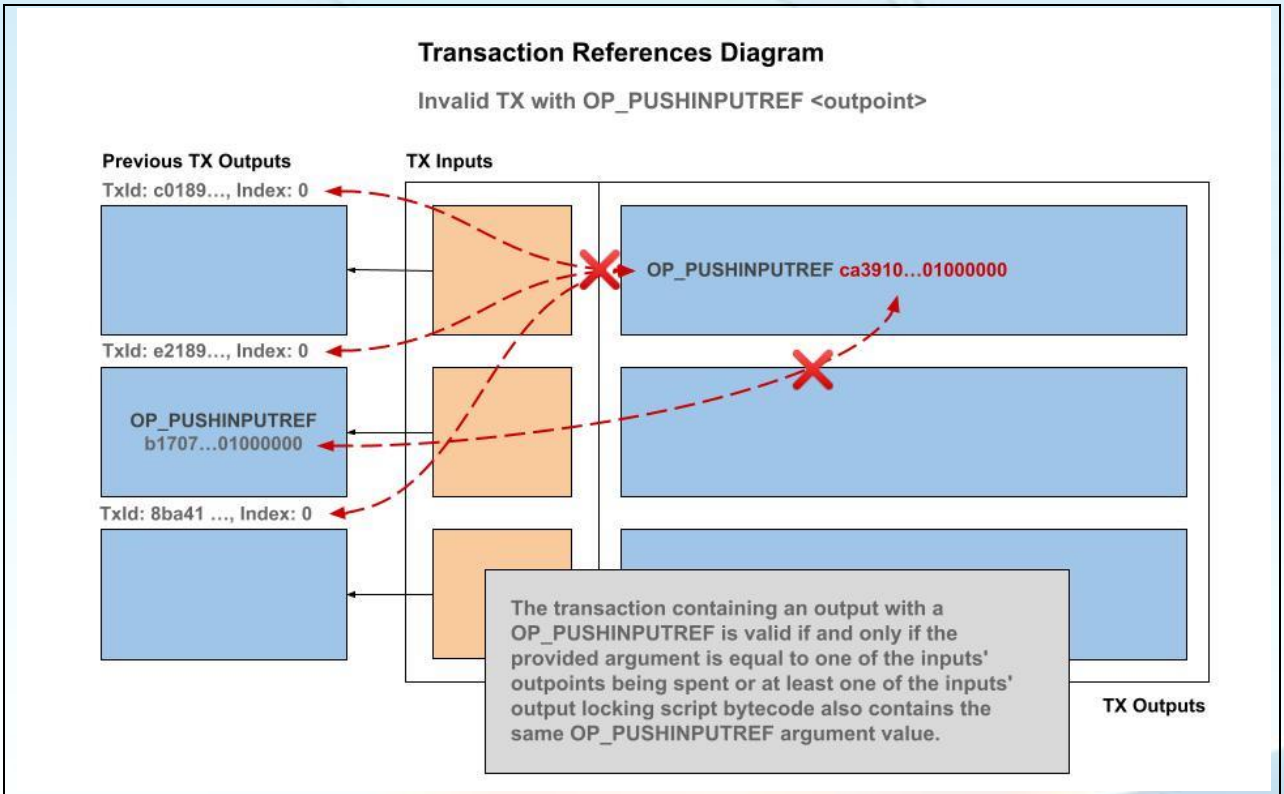


Diagram 5. Invalid TX with no matching previous output script or outpoint.

We demonstrate that this simple rule is sufficient to form a globally unique identifier, and carries no overhead — as in no extra indexes or lookup tables are required. Only a transaction and it's immediate parent inputs are needed to validate authenticity — all of the data is available to the virtual machine at the time of the unlocking script evaluation and also into accepting the transaction into the mempool and subsequently into a block. As long as at least one of the input coins has a valid 36 byte hash — either as the outpoint itself (significantly the first genesis chain of the colored coin) or as one of the scripts containing the reference, then the identity exists as a persisted identity. To terminate the lineage, simply omit passing on the reference and that terminates the ability to use that unique identifier in any other UTXO forever. Although this single OP code is sufficient, there are a handful of additional OP codes that provide flexibility for the programmer and are described next which complement OP_PUSHINPUTREF.

OP_REQUIREINPUTREF / Require reference

The OP_REQUIREINPUTREF functions identically to OP_PUSHINPUTREF except it does not pass on the reference identity to the output in which it appears. This is useful for demanding that at least one input is of a specific coin class — but without passing down the reference immediately.

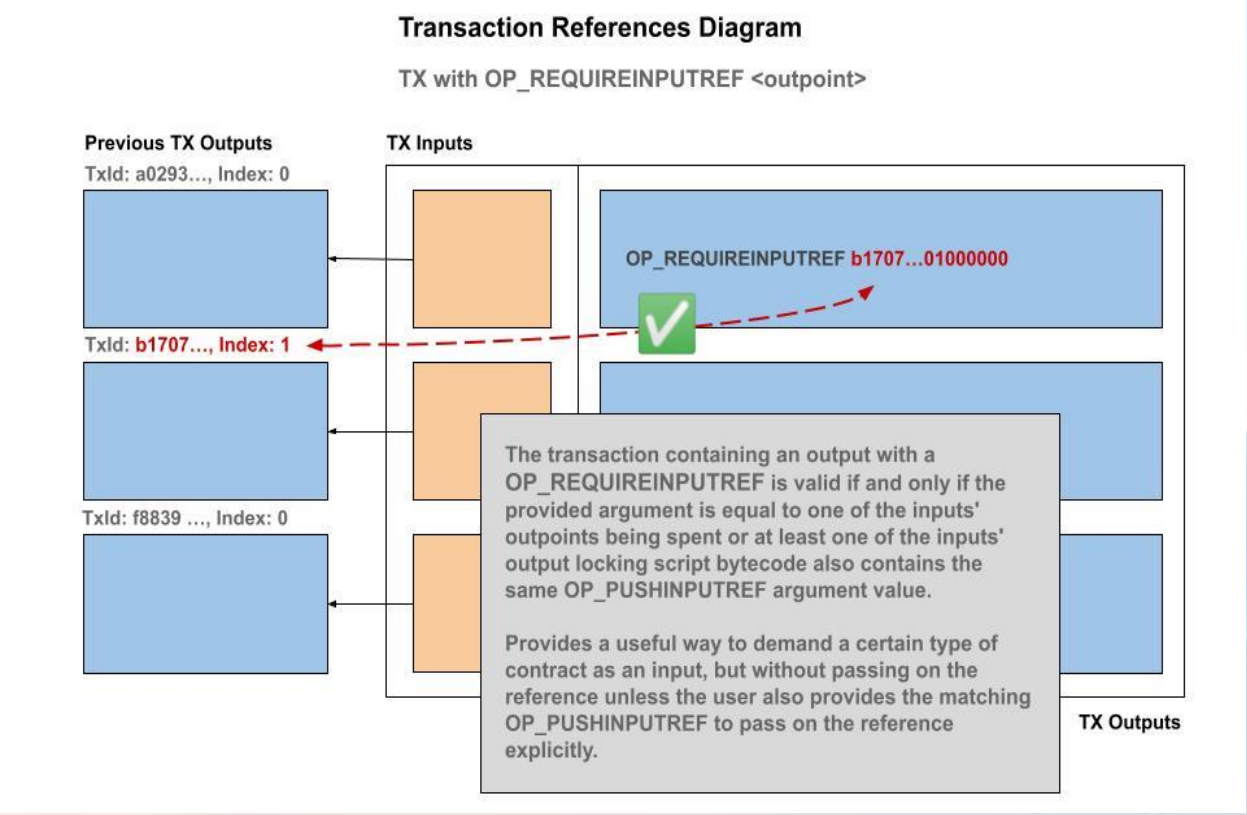


Diagram 6. OP_REQUIREINPUTREF functions the same as OP_PUSHINPUTREF but without pushing the reference to the current output.

OP_DISALLOWPUSHINPUTREF / Disallow reference in output

To disallow the use of a OP_PUSHINPUTREF in an output, the OP_DISALLOWPUSHINPUTREF may be used. This is a useful OP code for smart contracts which leave open the outputs to be used in various contexts, but allows the contract creator to restrict passing down a reference, for example in custom change outputs.

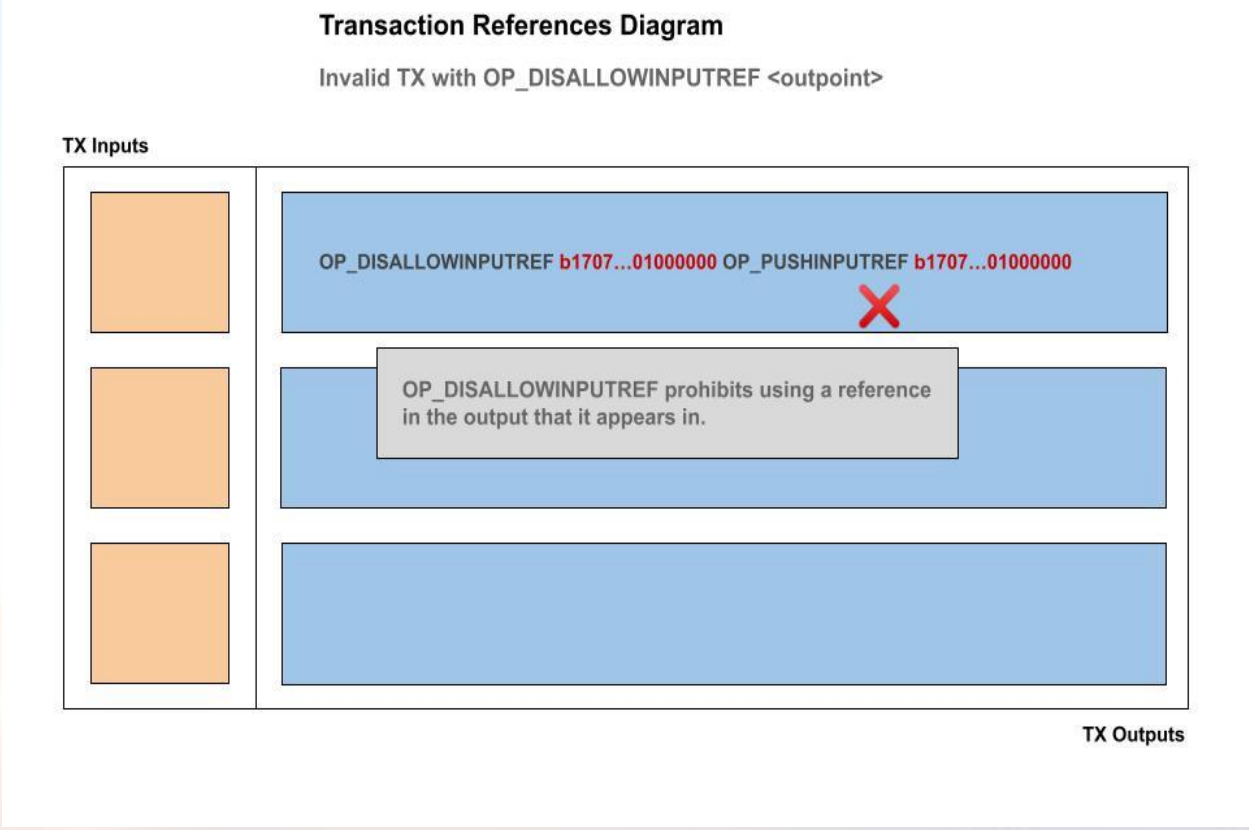


Diagram 7. OP_DISALLOWPUSHINPUTREF disallows usage of a specific reference.

OP_DISALLOWPUSHINPUTREFSIBLING / Disallow reference in sibling outputs

Similar to OP_DISALLOWPUSHINPUTREF, disallow specific outputs in any sibling outputs for the specific reference. This effectively prohibits using a reference in more than one output and is a way to create a singleton outpoint. By using OP_DISALLOWPUSHINPUTREFSIBLING in an output we can create a simple and powerful Non-Fungible Token (NFT) contract which functions with SIGHASH_SINGLE signature flag.

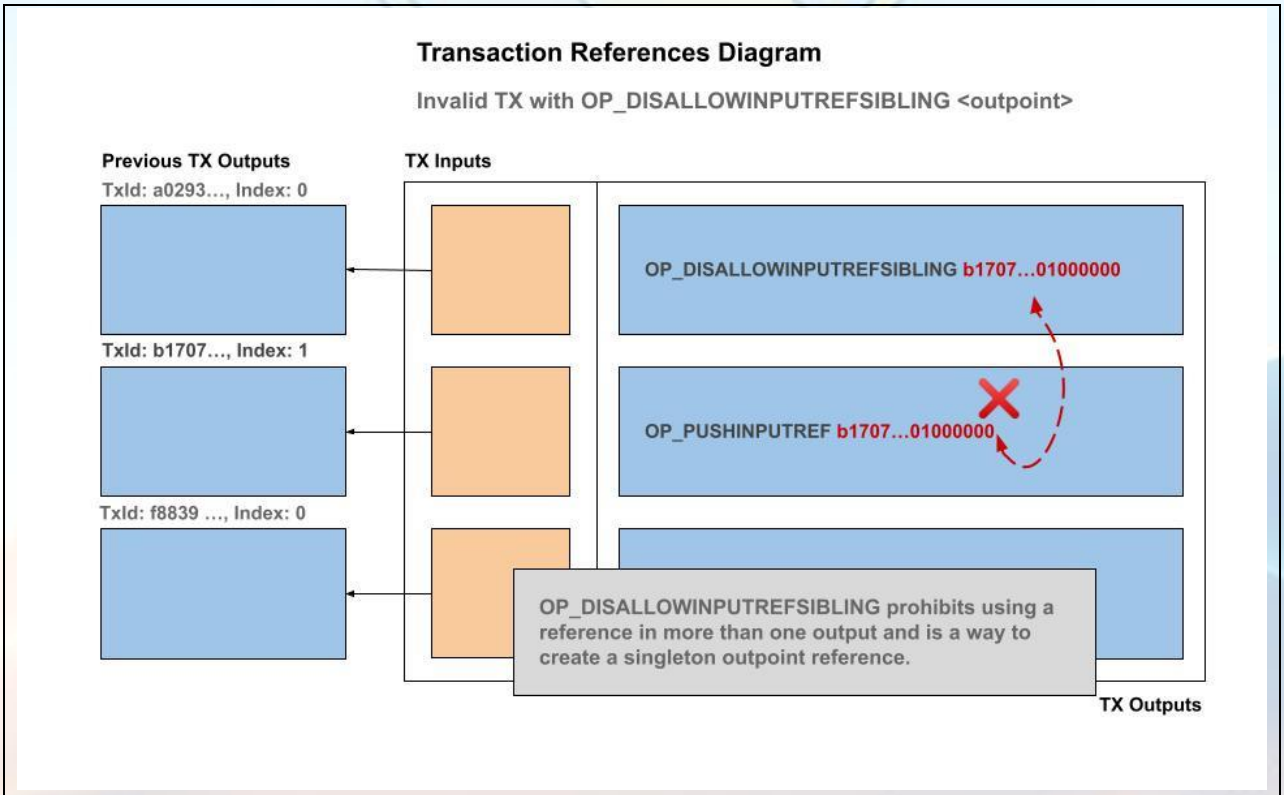


Diagram 8. OP_DISALLOWPUSHINPUTREFSIBLING: Disallow usage of a specific reference in all other outputs than the one in which this instruction appears.

OP_REFHASHDATASUMMARY_UTXO / Push UTXO data summary

Provides a summary of the contents of an output being spent in the current transaction. Takes the top element of the stack which is the index of the input being spent and then pushes the hash256 of the information about the UTXO being spent: hash256(<nValue><hash256(scriptPubKey)><numRefs><hash256(sorted_list(pushInputRefs))>). During unlocking script evaluation, the relevant data of an UTXO is able to be accessed and incorporated into the logic.

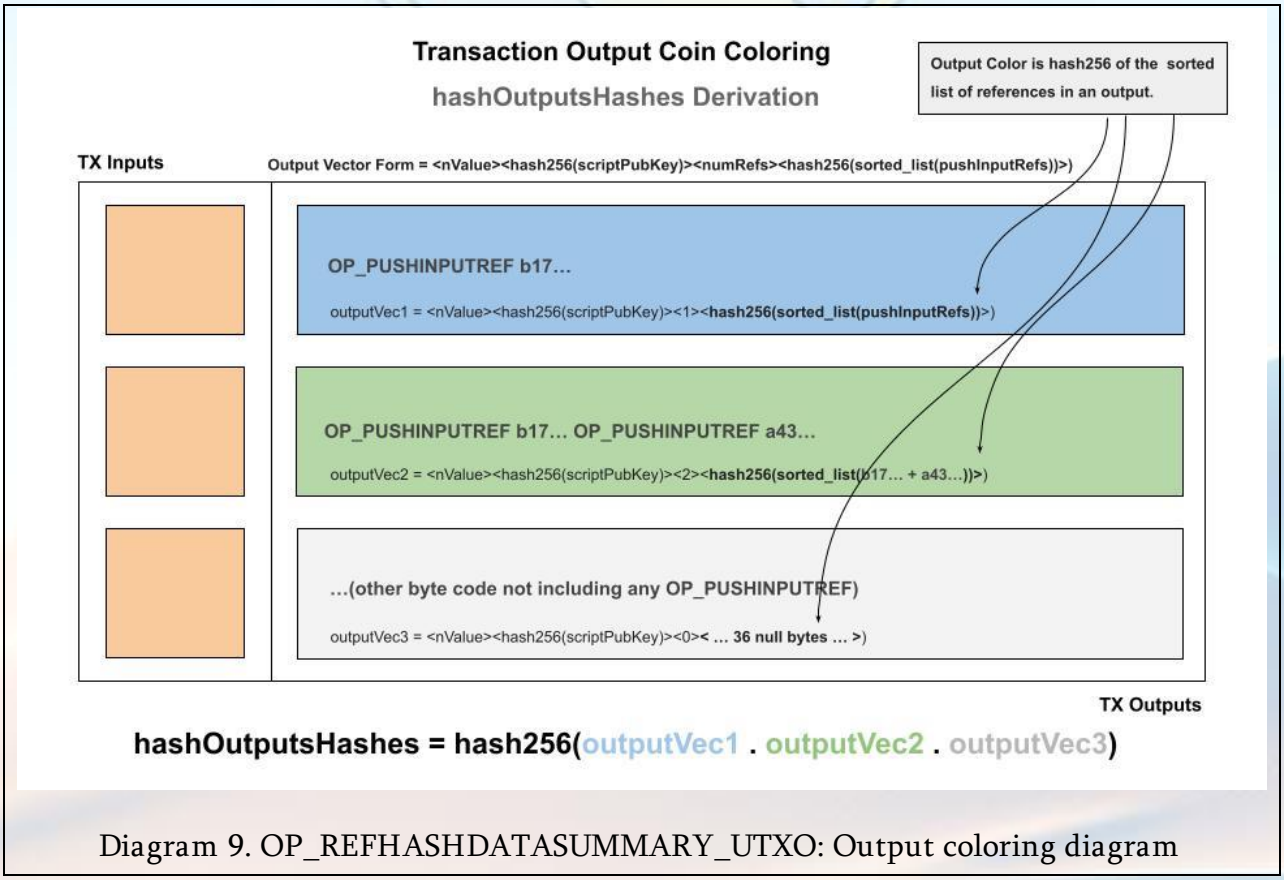


Diagram 9. OP_REFHASHDATASUMMARY_UTXO: Output coloring diagram

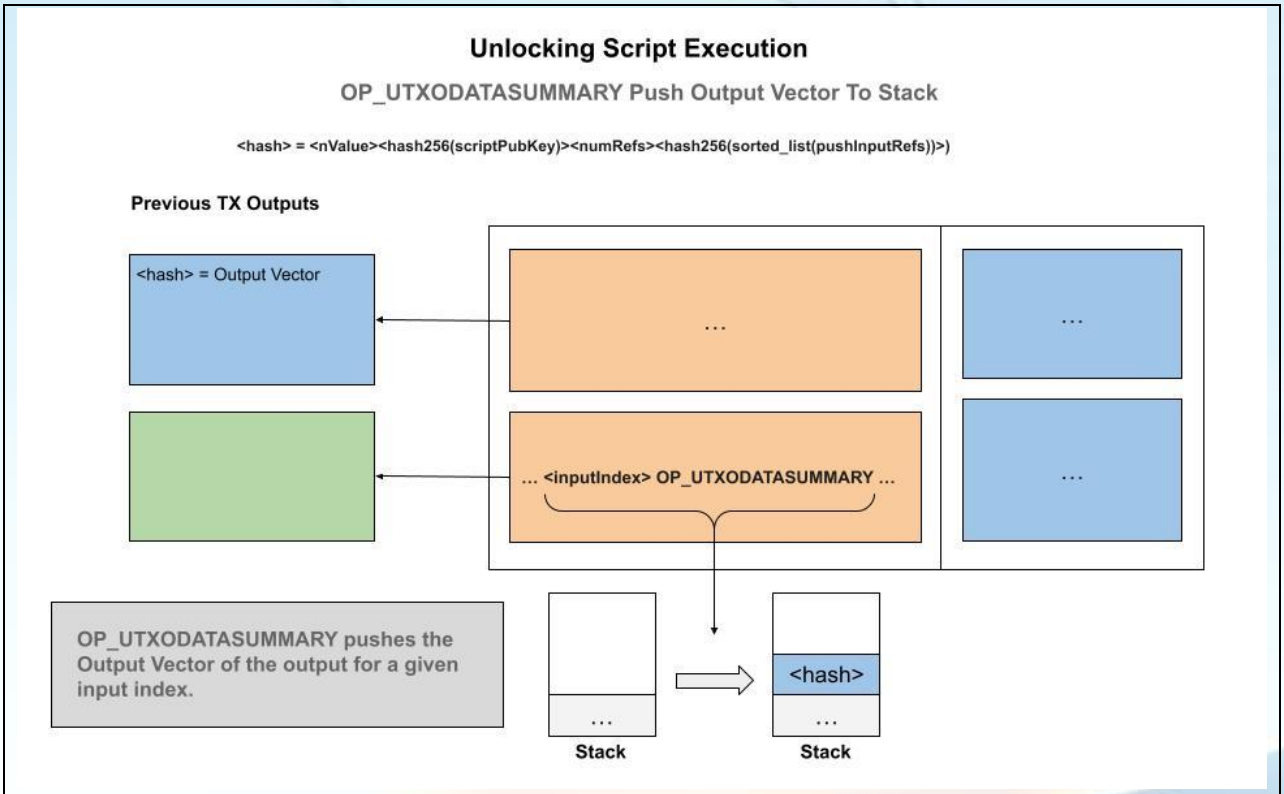


Diagram 10. OP_REFHASHDATASUMMARY_UTXO: Push to stack the summary of an input being spent.

OP_REFHASHVALUESUM_UTXOS / Push value sum of UTXO by reference (color)

This programming code accepts a hash256 of the 36 byte reference and pushes onto the stack the sum total of all of the inputs that matches that reference coloring. This is useful for saving data and for quickly assessing the total inputs and the values input to the transaction.

This is very useful for building a compact fungible token accounting system as we shall see below.

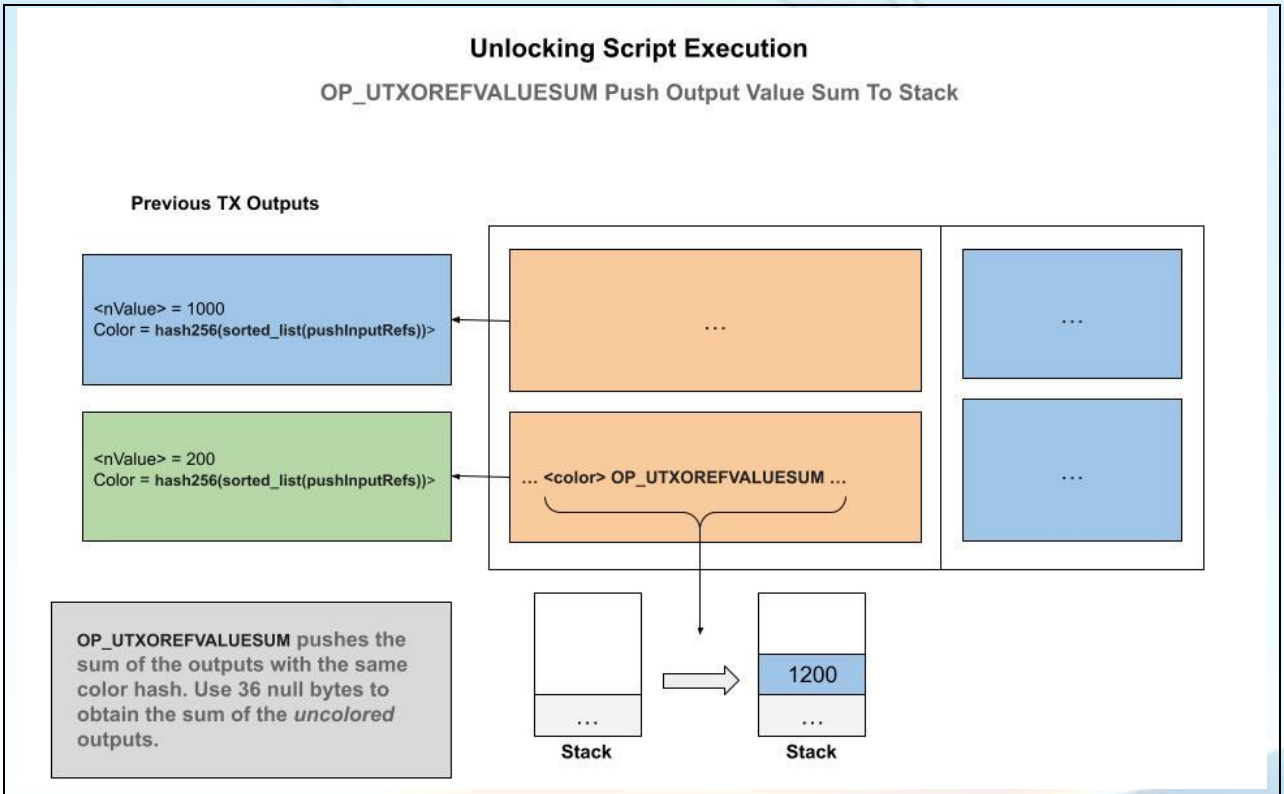


Diagram 11. OP_REFHASHVALUESUM_UTXOS: Push to stack the total sum of the inputs which match a specific reference hash

1.7 Contract Authenticity via Induction

Another method for solving the traceability and authenticity problem is to allow the embedding of the parent transaction into an unlocking script. In this manner, we can perform induction proofs and guarantee that a transaction output originated from a valid genesis minting event.

The general principle in mathematical induction is to prove that some statement P(k) holds for k = 0, k = 1, k = 2... and so on that generally P(k) holds for P(0) and P(k + 1).

In the case of smart contracts, we wish to prove that a given transaction is valid in the base case, as in it descends from valid parent, the base case P(0), and in the inductive case that the grand parent also satisfies the condition, which is the P(k + 1) step.

With an induction proof it is impossible to forge an intermediate transaction because the grand parent transaction will not be of the required origination.

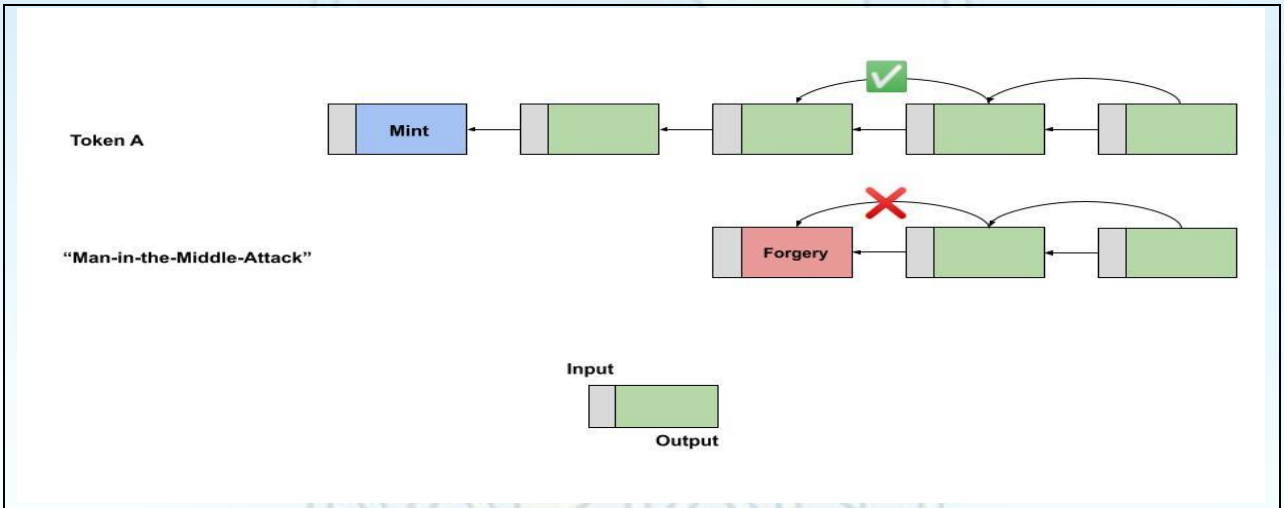


Diagram 12. Verify the parent and grand parent to identify forgery.

This system is not practical however because each time the output is spent a full copy of the parent (and it's parent) transaction must be embedded to calculate the transaction identifier. This leads to a factorial, or exponential, explosion in transaction size. It is not practical since after only about a dozen spends, the transaction size starts to exceed 1 GB and continues growing exponentially.

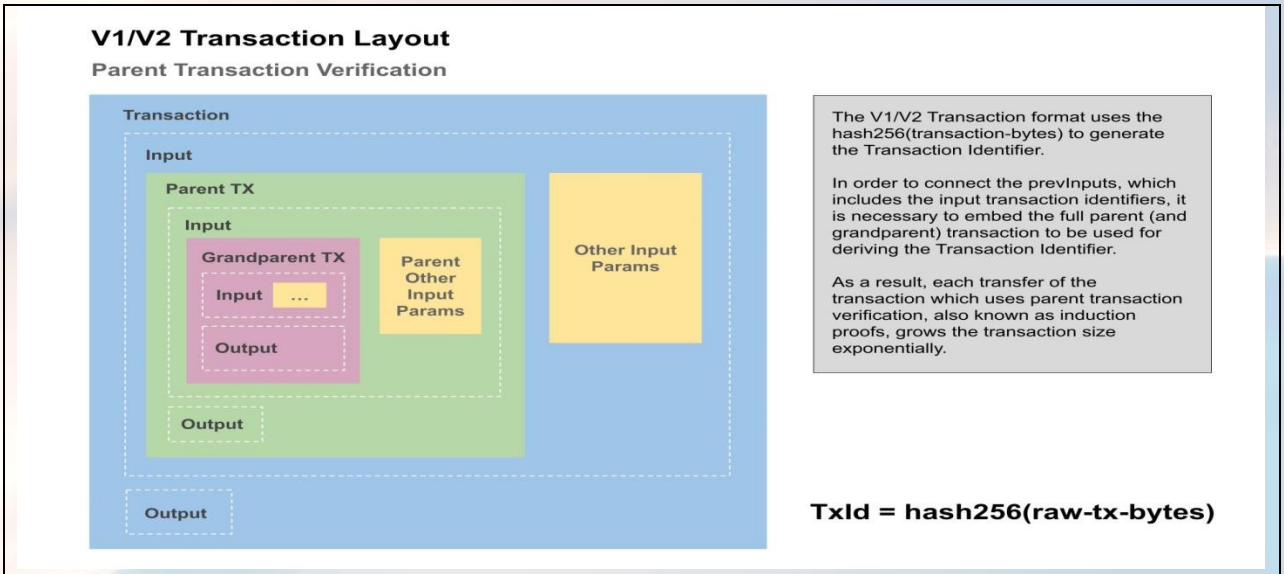


Diagram 13. Transaction Contents with Embedded Parents

To solve this problem of exponential transaction size growth we leverage the "nVersion" field provided by the transaction format. Bitcoin has version 1 and version 2 transactions already and we simply create a version 3 that uses a different transaction identifier generation algorithm instead of hashing the entire bytes of the transaction. The version 3 transaction format is identical except the transaction id is generated from an intermediate fixed size data structure that compresseses the transaction contents into a preimage — that can be embedded in locking scripts to derive the transaction id and avoid the exponential transaction size problem.

1.8 Transaction Identifier Version 3

Similar to the Signature Hash algorithm which generates a "Sighash Preimage", we produce a TxId preimage according to the following components and fields of a transaction.

1	nVersion of the transaction (4 byte little endian)
2	nTotalInputs (4 byte little endian)
3	hashPrevoutInputs (32 byte hash)
4	hashSequence (32 byte hash)
5	nTotalOutputs (4 byte little endian)
6	hashOutputHashes (32 byte hash)
7	nLocktime of the transaction (4 byte little endian)

Diagram 14. Transaction Identifier Version 3 Preimage

By incrementing the nVersion field, we introduce a way to compress an entire transaction into a fixed size (128 bytes) that can be pushed onto the stack, and hashed to arrive at the Transaction identifier, and therefore solving the problem of exponential size increase from from the embedded parent transactions in the induction proofs.

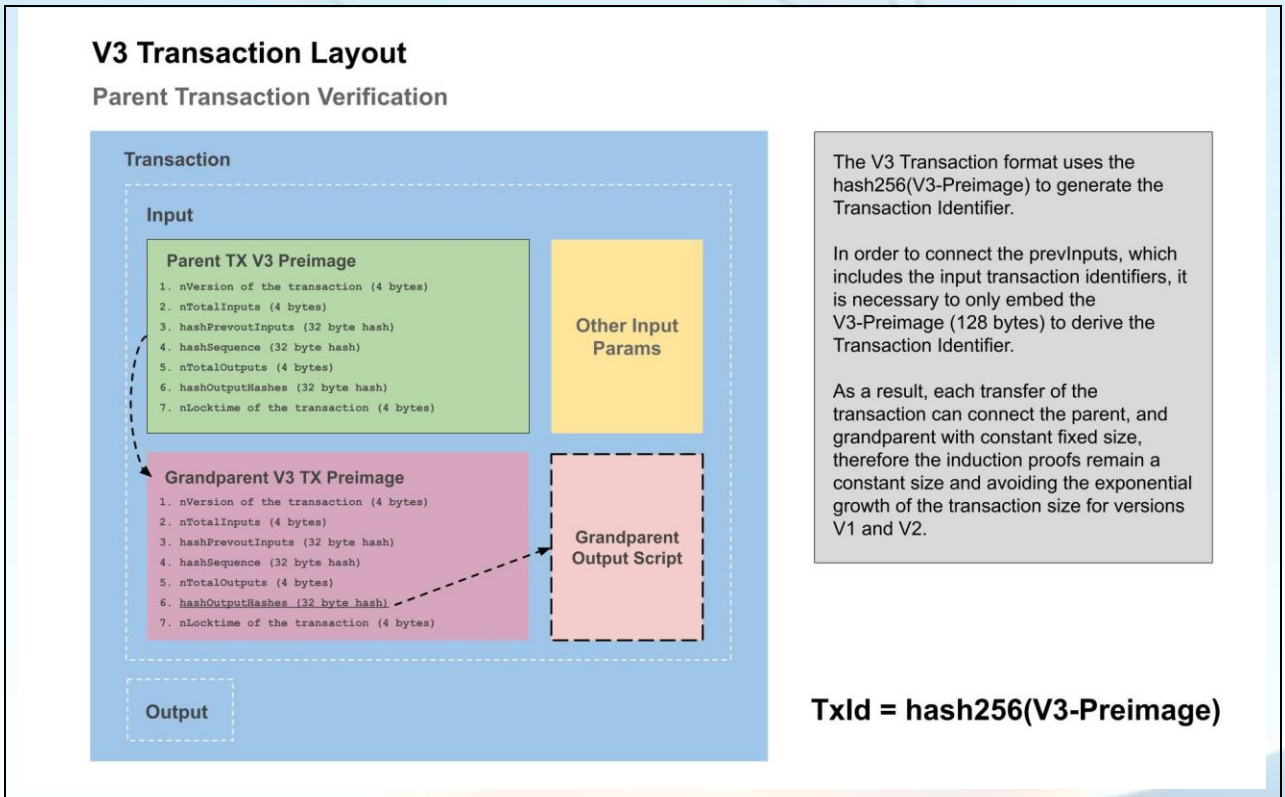


Diagram 15. Transaction Identifier Version 3 Preimages with Embedded Parents

Notice that this system itself is sufficient to create arbitrary induction proofs and is general purpose. This is a second method in which arbitrary induction proofs may be created in addition to the already discussed OP_PUSHINPUTREF technique.

1.9 Signature Hash Algorithm Upgrade

Building on the Transaction Id preimage the technique of segmenting the outputs, we can upgrade the default Sighash algorithm with an additional field called hashOutputsHashes to make it easier to constrain the outputs and save space and logic.

1	nVersion of the transaction (4 byte little endian)
2	hashPrevouts (32 byte hash)
3	hashSequence (32 byte hash)
4	outpoint (32 byte hash + 4 byte little endian)
5	scriptCode of the input (serialized as scripts inside CTxOuts)
6	value of the output spent by this input (8 byte little endian)
7	nSequence of the input (4 byte little endian)
8	hashOutputsHash (32 byte hash)
9	hashOutputs (32 byte hash)
10	nLocktime of the transaction (4 byte little endian)
11	sighash type of the signature (4 byte little endian)

Diagram 16. Radiant Signature Hash Preimage Fields.

This is useful because the other sibling outputs do not need to be included and a hash can be used for the outputs that are not of interest. There is still the color of the push references so that we can assert whether the other outputs contain a valid color, but without requiring the full script to be pushed.

1.10 Contract Design Patterns

With the OP_PUSHINPUTREF and TxId Version 3 constructs, we are in a position to define various contract collaboration design patterns. together these patterns will be used in account emulation, non-fungible tokens (NFTS), fungible tokens (FTs) and other programs.

1.11 Non-Fungible Tokens (NFT)

Definition: A "Non-Fungible Token" is a uniquely-identified object in which it's essential properties are conserved

We present a simple, yet powerful, design pattern called a Non-Fungible Token (NFT). Programs will recognize this by another name called a Singleton object. An NFT, or Singleton, guarantees that only one instance of an object can ever exist and is unique identified by a stable persistent identity. Most usages of the term Non-Fungible Token have centered around digital collectibles, however that need not be the case — the reason for that focus has to do with the high-gas fees on Ethereum and the speculative nature of digital artwork. In the Radiant blockchain, we use the term Non-Fungible Tokens to refer to a uniquely identifiable object, or colored coin, which maintains some essential properties in addition to being unambiguously traceable through the blockchain. This is a basic building block and design pattern that will appear in more complex contracts, and here we present a simple, yet very powerful, construction below to start.

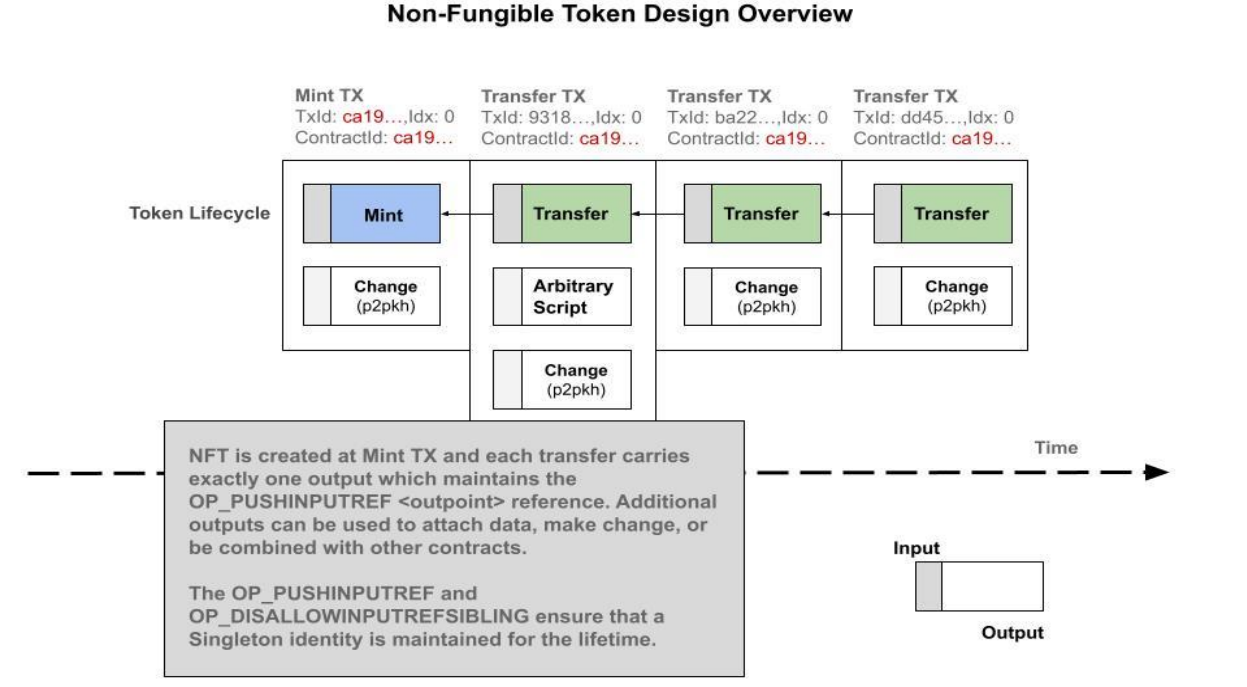


Diagram 17. Non-Fungible Token Design Overview.

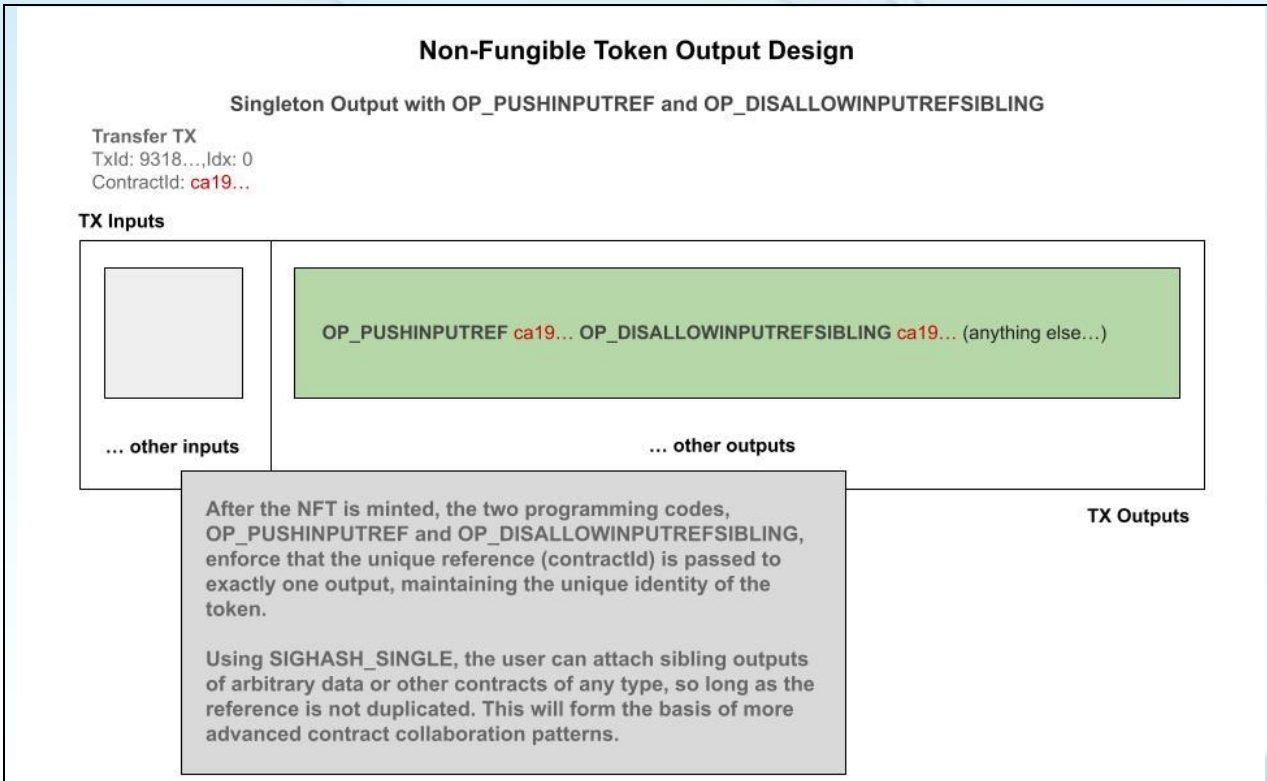


Diagram 18. Non-Fungible Token Output Design

Non-Fungible Token Pseudo-code:

```
contract NFT {

    // Asset identifier
    bytes assetId;

    // Current owner is the second push data
    Ripemd160 currentOwnerAddress;

    public function unlock(
        SigHashPreimage txPreimage,
        bytes outputSats,
        bytes newOwnerAddress,
        bool isMelt,
        Sig senderSig,
        PubKey unlockKey
```

```
    ) {
        require(hash160(unlockKey) == this.currentOwnerAddress);
        require(checkSig(senderSig, unlockKey));

        // Initial assetId is 36-bytes nulls(0x00 bytes)
        bytes actAssetId = (this.assetId == num2bin(0, 36) ?
            txPreimage[ 68 : 104 ] : this.assetId);

        bytes lockingScript = SigHash.scriptCode(txPreimage);

        // The default usage is to update/transfer
        if (!isMelt) {
            require(
                hash256(
                    outputSats +

                    // Define length of output
                    b'fd' + num2bin(len(lockingScript, 2)) +

                    // OP_PUSHINPUTREF <assetId>
                    b'd0' + actAssetId +

                    // New owner (20 bytes)
                    b'14' + newOwnerAddress +

                    // OP_DISALLOWPUSHINPUTREFSIBLING <assetId>
                    b'd3' + actAssetId +

                    // Get entire locking script after the push
vars
                    // 95 = 1+36 + 1+20 + 1+36
                    lockingScript[95 : ]
                )
                ==
                // Compare to HashOuts
                txPreimage[len(txPreimage) - 40 : len(txPreimage)
- 8]
            );
        } else {
            // Melt the NFT back and destroy the reference
```

```
            // Use OP_DISALLOWPUSHINPUTREF and
            // OP_DISALLOWPUSHINPUTREFSIBLING
            // to prohibit the reference from being passed along

            require(
                hash256(
                    // Hardcode len '4b' is 57 bytes (1 + 1 + 36
+ 1 + 36)
                    // 'd2' is OP_DISALLOWPUSHINPUTREF
                    // 'd3' is OP_DISALLOWPUSHINPUTREFSIBLING
                    b'00000000000000004b6ad2' + activeAssetId +
                    b'd3' + activeAssetId
                )
                ==
                // Compare to HashOuts
                txPreimage[len(txPreimage) - 40 : len(txPreimage)
- 8]
            );
        }
        require(Tx.checkPreimageOpt_(txPreimage));
    }
}
```

Diagram 19. Non-Fungible Token Pseudocode

1.12 Accounts & Smart Contracts

Definition: An "Account" is an object that manages a wallet balance while maintaining an addressable stable unique identifier.

One of the main difficulties in working with a UTXO-based blockchain is there is no protocol level concept of "wallet balance", and instead infrastructure providers and wallet services present a summary balance derived from the total value of all the individual outputs controlled by a key. Account-based blockchains also simplify specific types of problems and contracts, but trade-off performance and privacy to achieve it's aims.

We present a simple design pattern to emulate accounts using one or more outputs which gives the user and developer a stable unique identifier across transactions in the blockchain.

It is built using the Non-Fungible Token design pattern, and demonstrates that UTXO-based blockchains are perfectly equipped to emulate accounts with the same level of control, but with much higher performance characteristics.

Recall that in the Non-Fungible Token design pattern, the stable identifier *ContractId* (also sometimes referred to as *AssetId*) is derived from the outpoint of the minting transaction. The same ContractId is used as the public account identifier and can be treated as a wallet balance. We present below pseudo code for a smart contract that implements all the method associated with accounts: deposit, withdraw, changeOwner, and close the account.

```
contract Account {

    bytes assetId;
    Ripemd160 currentOwnerAddress;
    bytes disallowAssetIdNotUsed;

    static function createSingletonOutput(
        SigHashPreimage txPreimage,
        int amount,
        bytes assetId,
        bytes address
    ): bool {
        bytes activeAssetId = (assetId == num2bin(0, 36) ?
            txPreimage[ 68 : 104 ] : assetId);
        bytes lockingScript = SigHash.scriptCode(txPreimage);
        require(amount > 0);
        require(
                hash256(
                    // Add the deposit amount to the existing
balance
                    num2bin(SigHash.value(txPreimage) + amount,
8) +

                    b'fd' + num2bin(len(lockingScript), 2) +

                    // OP_PUSHINPUTREF <assetId>
                    b'd0' + activeAssetId +

                    // Address/owner (20 bytes)
                    b'14' + address +
```

```
                    // OP_DISALLOWPUSHINPUTREFSIBLING <assetId>
                    b'd3' + assetId +

                    // Get entire locking script after the push
vars
                    // 95 = 1+36 + 1+20 + 1+36
                    lockingScript[95 : ]
                )
                ==
                txPreimage[len(txPreimage) - 40 : len(txPreimage)
- 8] // HashOuts
        );
        require(Tx.checkPreimageOpt_(txPreimage));
        return true;
    }

    // Deposit to account
    // Anyone can spend this input and deposit funds into
    // the account, but only the owner can withdraw funds.
    public function deposit(
        SigHashPreimage txPreimage,
        int amount
    ) {
        require(amount > 0);

        require(
            Account.createSingletonOutput(
                txPreimage,
                SigHash.value(txPreimage) + amount,
                this.assetId,
                this.currentOwnerAddress
            )
        );
    }

    // Withdraw from account
    // The current owner can withdraw from the account
    // via any other outputs.
    public function withdraw(
        SigHashPreimage txPreimage,
```

```
        int amount,
        Sig senderSig,
        PubKey unlockKey
    ) {
        require(hash160(unlockKey) == this.currentOwnerAddress);
        require(checkSig(senderSig, unlockKey));

        require(
            Account.createSingletonOutput(
                txPreimage,
                SigHash.value(txPreimage) - amount,
                this.assetId,
                this.currentOwnerAddress
            )
        );
    }

    // Change the account owner
    // The current owner can assign the account to another
    // address owner
    public function changeOwner(
        SigHashPreimage txPreimage,
        bytes newOwnerAddress,
        Sig senderSig,
        PubKey unlockKey
    ) {
        require(hash160(unlockKey) == this.currentOwnerAddress);
        require(checkSig(senderSig, unlockKey));

        require(
            Account.createSingletonOutput(
                txPreimage,
                SigHash.value(txPreimage),
                this.assetId,
                newOwnerAddress
            )
        );
    }

    // Close the account
    // The current owner of the account can permanently close
```

```
    // the account and withdraw any tokens via other outputs
    public function close(
        SigHashPreimage txPreimage,
        Sig senderSig,
        PubKey unlockKey
    ) {
        require(hash160(unlockKey) == this.currentOwnerAddress);
        require(checkSig(senderSig, unlockKey));

        bytes activeAssetId = (this.assetId == num2bin(0, 36) ?
            txPreimage[ 68 : 104 ] : this.assetId);

        bytes lockingScript = SigHash.scriptCode(txPreimage);

        // Ensure one of the outputs is unspendable OP_RETURN
        // and uses the OP codes to prohibit passing on the
        // reference.
        // OP_DISALLOWPUSHINPUTREF and
        // OP_DISALLOWPUSHINPUTREFSIBLING which effectively
        // means no output may contain the reference anymore,
        // thereby ending the ability to carry on the assetId
        // anywhere else forever.
        require(
            hash256(
                b'00000000000000004b6ad2' + activeAssetId +
                // Hardcode len '4b' is 57 bytes (1 + 1 + 36 + 1
+ 36)
                b'd3' + activeAssetId
            )
            ==
            // HashOuts
            txPreimage[len(txPreimage) - 40 : len(txPreimage) -
8]
        );
        require(Tx.checkPreimageOpt_(txPreimage));
    }
}
```

Diagram 20. Account contract pseudocode

1.13 Fungible Tokens (FT)

The Fungible Token design pattern allows the same class or type of object to have more than a quantity of one. The fungible tokens can be merged together, with their values summed up into a new output, or an output can be split into two or more outputs where the total sum of the outputs is equal to the input value amount. This design pattern is useful for simulating loyalty points, tokens, and more. We present the solution:

```
contract SuperAssetR201 {
    // Do NOT provide a constructor as that will add unnecessary
OP_0 OP_0 to the beginning of the contract
    bytes assetId;                      // Asset identifier
    Ripemd160 currentOwnerAddress;      // Current owner is the
second push data
    // Notice that "disallowAssetIdNotUsed" is not used below.
The reason is that we save space and also it should always be
same as assetId
    bytes disallowAssetIdNotUsed;       // Disallow Asset from
being used in any other output
    static const int MAX_RECEIVE = 6;

    static function buildOutputVector(
        int amount,
        bytes assetId,
        bytes address,
        bytes outputScriptLen,
        bytes lockingScriptCodePart
    ): bytes {
        return
            num2bin(amount, 8) +
            hash256(
                outputScriptLen +

                // OP_PUSHINPUTREF <assetId>
                b'd0' + assetId +

                // Address/owner (20 bytes)
                b'14' + address +

                lockingScriptCodePart
```

```
            ) +
            // One color for the output
            b'01000000' +
            hash256(assetId);
    }

    public function mint(SigHashPreimage txPreimage, int amount)
{
        require(amount > 0);
        require(this.assetId == num2bin(0, 36));
        bytes lockingScript = SigHash.scriptCode(txPreimage);
        require(
                hash256(
                    num2bin(amount, 8) +

                    b'fd' + num2bin(len(lockingScript), 2) +

                    // OP_PUSHINPUTREF <assetId>
                    b'd0' + txPreimage[68 : 104]+

                    // Address/owner (20 bytes)
                    b'14' + this.currentOwnerAddress +

                    // Get entire locking script after the push
vars
                    // 95 = 1+36 + 1+20
                    lockingScript[58 : ]
                )
                ==
                txPreimage[len(txPreimage) - 40 : len(txPreimage)
- 8] // HashOuts
        );
        require(Tx.checkPreimageOpt_(txPreimage));
    }

    public function transfer(SigHashPreimage txPreimage,
Ripemd160[6] recipients, int[6] amounts, bytes otherOutputs, Sig
senderSig, PubKey unlockKey) {
        require(hash160(unlockKey) == this.currentOwnerAddress);
        require(checkSig(senderSig, unlockKey));
```

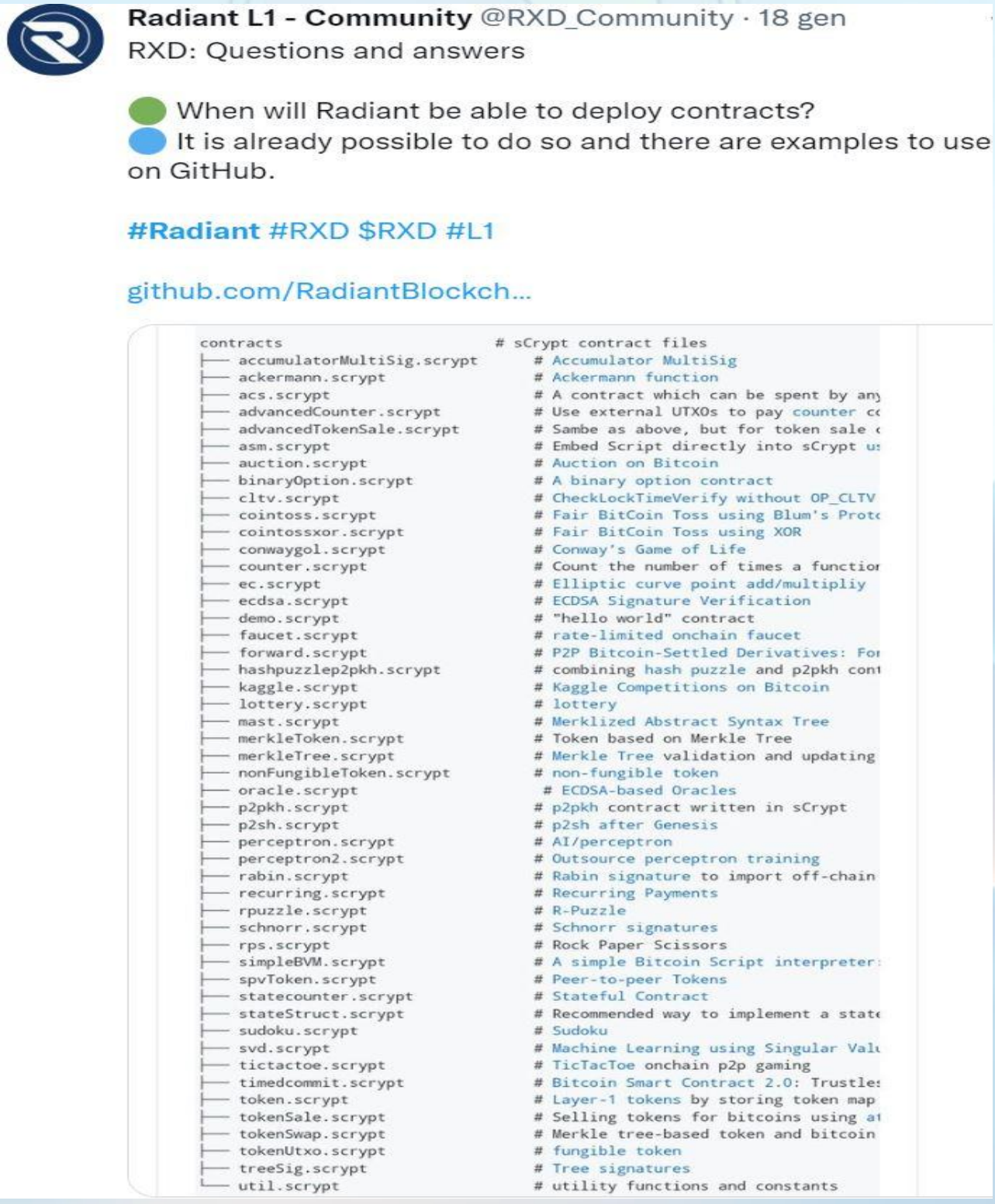
```
        int expectedRefColorSum = 1337;         // Placeholder
for OP_INPUTREFVALUESUM
        int actualAccumulatedRefColorSum = 0;   // Used for
counting the sum of the colors
        bool break = false;
        bytes expectedOutputVector = b'';
        bytes lockingScript = SigHash.scriptCode(txPreimage);
        // Length of the output script
        bytes outputScriptLen = b'fd' +
num2bin(len(lockingScript), 2);
        bytes lockingScriptCodePart = lockingScript[58 : ];
        loop (MAX_RECEIVE) : i {
            if (!break) {
                if (amounts[i] <= 0) {
                    break = true;
                } else {
                    // There is a valid recipient...
                    // Get entire locking script after the push
vars
                    // 58 = 1+36 + 1+20
                    expectedOutputVector +=
SuperAssetR201.buildOutputVector(amounts[i], this.assetId,
recipients[i], outputScriptLen, lockingScriptCodePart);
                    actualAccumulatedRefColorSum += amounts[i];
                }
            }
        }
        require(expectedRefColorSum > 0 && expectedRefColorSum ==
actualAccumulatedRefColorSum);
        require(
            hash256(expectedOutputVector + otherOutputs)
            ==
            // hashOutputsHashes
            txPreimage[len(txPreimage) - 72 : len(txPreimage) -
40]
        );
    }

    public function melt(SigHashPreimage txPreimage, Sig
senderSig, PubKey unlockKey) {
        require(hash160(unlockKey) == this.currentOwnerAddress);
```

```
        require(checkSig(senderSig, unlockKey));
        // Ensure one of the outputs is unspendable OP_RETURN and
uses the OP codes to prohibit passing on the reference
        // OP_DISALLOWPUSHINPUTREF and
OP_DISALLOWPUSHINPUTREFSIBLING which effectively means no output
may contain
        // the reference anymore, thereby ending the ability to
carry on the assetId anywhere else forever.
        require(
            hash256(
                // Hardcode len '4b' is 57 bytes (1 + 1 + 36 + 1
+ 36)
                b'00000000000000004b6ad2' + this.assetId +  b'd3'
+ this.assetId
            )
            ==
            txPreimage[len(txPreimage) - 40 : len(txPreimage) -
8]
        );
        require(Tx.checkPreimageOpt_(txPreimage));
    }
}
```

1.14 Conclusion

We have proposed a system for digital asset management without relying on trust. We started with the basic blockchain construction of coins made from digital signatures, which provides strong control of ownership. From the needed rules and incentives, we introduced two novel methods for authenticating and tracking digital assets in constant O(1) time and space. Both methods independently provide a general induction proof system which can encode any possible digital asset configuration. The system is Turing Complete within and across transaction boundaries, with unbounded scale, and never any need for secondary layers. Additionally we have presented three contract design patterns: Non-Fungible Token (NFT), Fungible Token (FT) and Account which emulating account based blockchains, using the UTXO based processing model. Radiant is a breakthrough design which provides the performance and parallelism benefits of an unspent transaction output (UTXO) blockchain,

and with the programming sophistication of account-based blockchains, while maintaining ultra low fees and unbounded scale.



Radiant L1 - Community @RXD_Community · 18 gen

RXD: Questions and answers

When will Radiant be able to deploy contracts?
It is already possible to do so and there are examples to use on GitHub.

#Radiant #RXD $RXD #L1

github.com/RadiantBlockch...

```
contracts                          # sCrypt contract files
├── accumulatorMultiSig.scrypt         # Accumulator MultiSig
├── ackermann.scrypt                   # Ackermann function
├── acs.scrypt                         # A contract which can be spent by any
├── advancedCounter.scrypt             # Use external UTXOs to pay counter co
├── advancedTokenSale.scrypt           # Sambe as above, but for token sale c
├── asm.scrypt                         # Embed Script directly into sCrypt us
├── auction.scrypt                     # Auction on Bitcoin
├── binaryOption.scrypt                # A binary option contract
├── cltv.scrypt                        # CheckLockTimeVerify without OP_CLTV
├── cointoss.scrypt                    # Fair BitCoin Toss using Blum's Proto
├── cointossxor.scrypt                 # Fair BitCoin Toss using XOR
├── conwaygol.scrypt                   # Conway's Game of Life
├── counter.scrypt                     # Count the number of times a functior
├── ec.scrypt                          # Elliptic curve point add/multipliy
├── ecdsa.scrypt                       # ECDSA Signature Verification
├── demo.scrypt                        # "hello world" contract
├── faucet.scrypt                      # rate-limited onchain faucet
├── forward.scrypt                     # P2P Bitcoin-Settled Derivatives: For
├── hashpuzzlep2pkh.scrypt             # combining hash puzzle and p2pkh cont
├── kaggle.scrypt                      # Kaggle Competitions on Bitcoin
├── lottery.scrypt                     # lottery
├── mast.scrypt                        # Merklized Abstract Syntax Tree
├── merkleToken.scrypt                 # Token based on Merkle Tree
├── merkleTree.scrypt                  # Merkle Tree validation and updating
├── nonFungibleToken.scrypt            # non-fungible token
├── oracle.scrypt                       # ECDSA-based Oracles
├── p2pkh.scrypt                       # p2pkh contract written in sCrypt
├── p2sh.scrypt                        # p2sh after Genesis
├── perceptron.scrypt                  # AI/perceptron
├── perceptron2.scrypt                 # Outsource perceptron training
├── rabin.scrypt                       # Rabin signature to import off-chain
├── recurring.scrypt                   # Recurring Payments
├── rpuzzle.scrypt                     # R-Puzzle
├── schnorr.scrypt                     # Schnorr signatures
├── rps.scrypt                         # Rock Paper Scissors
├── simpleBVM.scrypt                   # A simple Bitcoin Script interpreter
├── spvToken.scrypt                    # Peer-to-peer Tokens
├── statecounter.scrypt                # Stateful Contract
├── stateStruct.scrypt                 # Recommended way to implement a state
├── sudoku.scrypt                      # Sudoku
├── svd.scrypt                         # Machine Learning using Singular Valu
├── tictactoe.scrypt                   # TicTacToe onchain p2p gaming
├── timedcommit.scrypt                 # Bitcoin Smart Contract 2.0: Trustles
├── token.scrypt                       # Layer-1 tokens by storing token map
├── tokenSale.scrypt                   # Selling tokens for bitcoins using at
├── tokenSwap.scrypt                   # Merkle tree-based token and bitcoin
├── tokenUtxo.scrypt                   # fungible token
├── treeSig.scrypt                     # Tree signatures
└── util.scrypt                        # utility functions and constants
```

1.15 References

[1] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System" https://bitcoin.org/bitcoin.pdf, 2009.

SOURCE: https://radiant4people.com/tech/radiant-system-design/

CHAPTER II – TECHNICAL DETAILS (LEGEND)

2.1 Programming

2.1.1 RAD SCRYPTLIB

Javascript/TypeScript SDK for integration of Radiant (RAD) Blockchain Smart Contracts written in the sCrypt language.

PLEASE NOTE: This is a fork of scryptlib as a convenience that contains the patches to the included bsv.js lib directly Alternatively, the regular scryptlib may be used along with radjs https://github.com/RadiantBlockchain/radjs and not using the bundled bsv in scryptlib.

https://radiant4people.com/programming/rad-scryptlib/

2.1.2 sCrypt PROJECT BOILERPLATE

Note: Modified to use @radiantblockchian/radjs for testnet/superAssetR100.js

https://radiant4people.com/programming/scrypt-boilerplate/

2.1.3 RADIANTSCRIPT

See the awesome contracts and applications that people are building with RadiantScript!

AnyHedge

AnyHedge is the first DeFi project built on top of Radiant in the form of a synthetic derivatives platform. AnyHedge allows any two parties to enter into a smart contract together and speculate on the future price of an asset. One of the parties wishes to protect themselves against price fluctuations and takes the hedge position, while the other party wishes to speculate and takes a leveraged long position.

https://radiant4people.com/programming/scrypt-boilerplate/

CHAPTER III – GENERAL ANALYSIS (LEGEND)

3.1 What is Radiant?

Radiant (or RXD) is a peer-to-peer electronic cash system. It uses a blockchain to distribute its ledger over a network of independent nodes so that there is no single point of failure, and no central control that might be compromised. It uses a consensus algorithm called Proof-of-Work that allows these independent nodes to approve correct transactions and reject malicious ones.

Basics / The blockchain is a data structure that is distributed over a number of independent nodes. It derives its name from the chain of blocks that it uses to store its data. All blocks include a block header with some metadata and the root of a Merkle tree - a special kind of tree that allows quick validation of data. This Merkle tree is then used to store the actual data inside these blocks. To make the chain resistant to manipulation, block headers also include a timestamp and a hash of the previous block.

Proof-of-Work / Radiant and many other public blockchains use a consensus algorithm called Proof-of-Work (PoW). This algorithm works by attaching a nonce to every block header and changing this nonce until the hash of the block header matches a certain

prefix. This process is called mining, and is attempted by many nodes at the same time, until one of them has found a correct solution. One of the attributes of this algorithm is that mining is very expensive, but other nodes can verify the solution very quickly.

Mining is also the process by which new coins are introduced to the total monetary supply. Miners validate transactions and secure the network, for which they are paid new coins - called the block reward - in a special transaction called a coinbase transaction. The high cost of the mining process attaches a financial risk to incorrectly validating transactions. At the same time the block reward attaches a financial reward to correctly validating transactions. This process ensures that the mutually distrusting nodes can collaborate to validate transactions.

Transactions / Radiant transactions are created using chunks of RXD called transaction outputs. When these outputs are available, they are called Unspent Transaction Outputs (UTXOs). UTXOs are locked using a locking script (or scriptPubKey) that specifies the conditions to spend the UTXO. When attempting to spend a UTXO, an unlocking script (or scriptSig) is provided. These scripts are then executed together and the transaction is only valid if the scripts execute without errors and the resulting value is TRUE.

The most used locking/unlocking script pattern is called Pay-to-Public-Key-Hash (P2PKH), where the locking script contains the hash of a public key and expects the unlocking script to contain a public key and transaction signature. The locking script then checks that the provided public key matches the stored hash, and that the transaction signature is valid. This pattern is used in regular Radiant wallets. And the user's balance is simply the sum of all UTXOs that can be spent by the user's public keys.

UTXOs are used as inputs to Radiant transactions and produce new UTXOs as outputs. UTXOs need to be spent in their entirety within a transaction. So whenever the user wishes to use a 10 RXD UTXO to send someone 1 RXD, they need to send 9 RXD back to themselves. Realistically, part of the funds would be reserved for transaction fees as well.

Smart Contracts /

Peer-to-peer electronic cash was the first real use case of blockchain technology. But in recent years, smart contracts have grown in popularity. These smart contracts allow people to use the security that blockchains such as Bitcoin, Radiant and Ethereum offer and apply it to use cases other than cash. Especially Decentralized Finance (DeFi) applications such as Maker, Uniswap and Aave have skyrocketed.

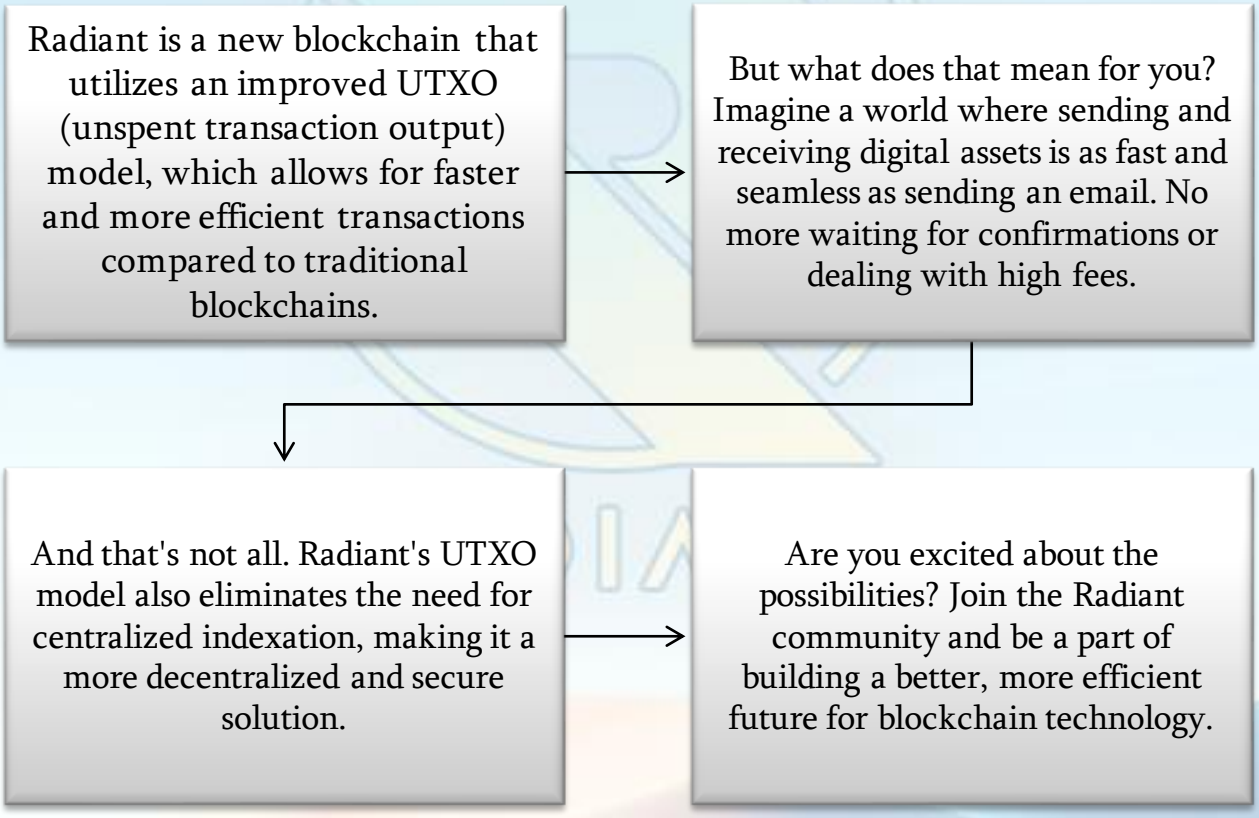
Most smart contract innovation has happened on Ethereum, but other platforms like Bitcoin and Radiant have some support for smart contracts as well. Smart contracts on every platform work differently, and the main differences between smart contracts on Ethereum and Radiant is that smart contracts on Ethereum are stateful, while those on Radiant are stateless.

This means that Ethereum contracts can record and update variables, while the variables in Radiant contracts are immutable.

Radiant Script /

The locking and unlocking scripts of regular transactions and smart contracts on Radiant are written using Radiant' transaction scripting language, creatively named Script. To avoid ambiguity, it can also be referred to as Bitcoin Script or Radiant Script. Script is a stack based assembly-like language that is intentionally not turing complete as its main use is the validation of programmable money, not general purpose computing.

Script is stateless, meaning it only uses the information contained within the locking and unlocking scripts themselves. This statelessness means that a Script can be deterministically validated on any machine. This gives increased performance and predictability, although it does limit the usefulness of the scripting language.



The Induction proof system is the core of Radiant, which allows for no dependence on external machines such as EVM or redundant code within the chain when moving FT/NFT.

Induction OP code in http://RadiantBlockchain.org makes it trivial to create massively parallel rich contract accounts. With zero index overhead and zero database persistence overhead. This makes it the first and only blockchain to maximize the potential of the UTXO model.

https://radiant4people.com/programming/radiantscript/basics/about-rxd/

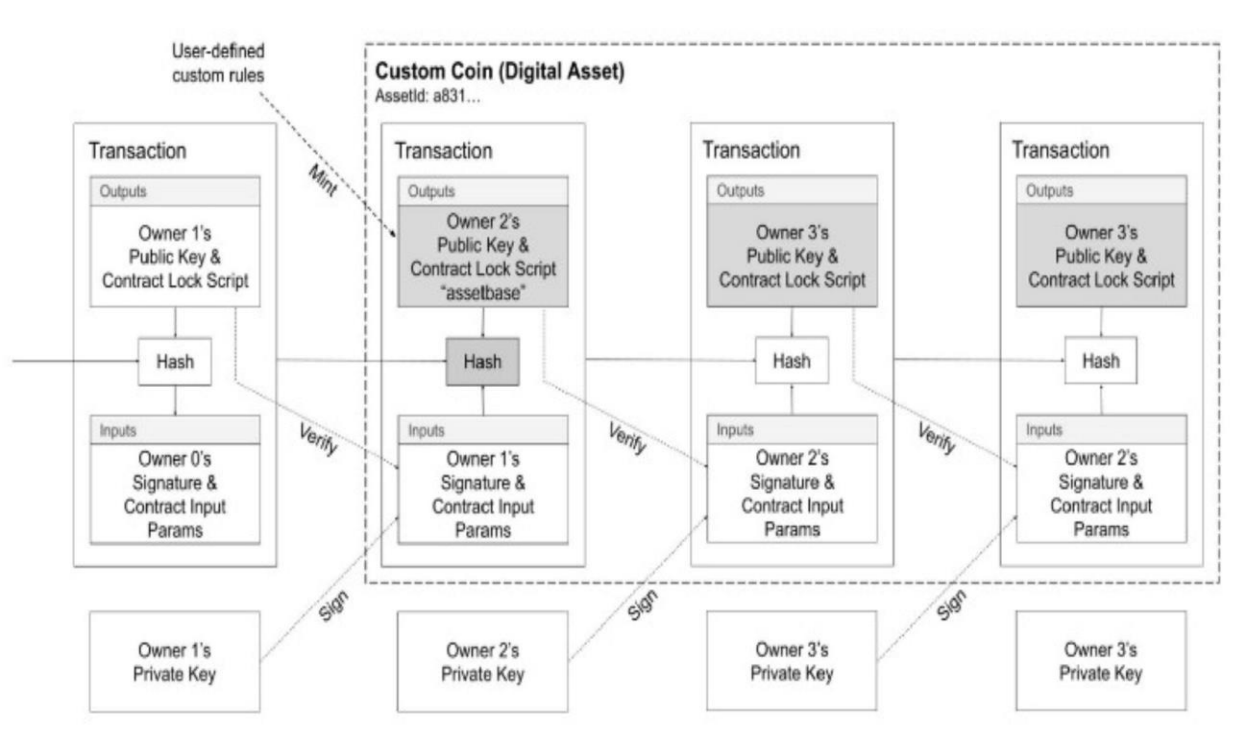
3.2 What is RadiantScript?

CashScript is a high-level programming language for smart contracts on Radiant. It offers a strong abstraction layer over Radiant' native virtual machine, RadiantScript. Its syntax is based on Ethereum's smart contract language Solidity, but its functionality is very different

since smart contracts on Radiant differ greatly from smart contracts on Ethereum. For a detailed comparison of them, refer to the blog post. If you're interested to see what kind of things can be built with CashScript, you can look at the Showcase or Examples. If you just want to dive into CashScript, refer to the Getting Started page and other pages in the documentation.

https://radiant4people.com/programming/radiantscript/basics/about/

Colored coins (Custom digital assets) was always possible with the UTXO model. No additional indexes or databases needed thanks to the induction OP codes in Radiant.



RadiantScript is a compiled language, so as long as the compiled scripts are properly tested there will be no issues. It is not even necessary to use RadiantScript, it is possible to write the scripts manually or use what RadiantScript compiles as a guide. This approach is also very

encouraged, it's very important to understand the compiled script. RadiantScript is very useful as a prototyping tool. It has already helped a lot with building standard token scripts.

3.3 Command Line Interface

The cashc command line interface is used to compile CashScript .cash files into .json artifact files. These artifacts can be imported and used by the JavaScript SDK or other libraries / applications that use CashScript. For more information on this artifact format refer to Artifacts.

Installation

You can use npm to install the cashc command line tool globally.

```
npm install -g cashc
```

Usage

The cashc CLI tool can be used to compile .cash files to JSON artifact files.

```
Usage: cashc [options] [source_file]

Options:

  -V, --version        Output the version number.

  -o, --output <path>  Specify a file to output the generated artifact.

  -h, --hex            Compile the contract to hex format rather than a full artifact.

  -A, --asm            Compile the contract to ASM format rather than a full artifact.

  -c, --opcount        Display the number of opcodes in the compiled bytecode.

  -s, --size           Display the size in bytes of the compiled bytecode.
```

-?, --help	Display help

https://radiant4people.com/programming/radiantscript/basics/cli/

3.4 Getting Started

Installing the CashScript compiler

The command line CashScript compiler cashc can be installed from NPM.

```
npm install -g cashc
```

Installing the JavaScript SDK

The JavaScript SDK can be installed into your project with NPM.

```
npm install cashscript
```

:::caution CashScript only offers a JavaScript SDK, but CashScript contracts can be integrated into other languages as well. Because there are no ready-to-use SDKs available for them, this is considered advanced usage, and it is recommended to use the JavaScript SDK. :::

Writing your first smart contract

There are some examples available on the Examples page, that can be used to take inspiration from. Further examples of the JavaScript integration can be found on GitHub. A simple example is included below.

```
pragma cashscript ^0.7.0;

contract TransferWithTimeout(pubkey sender, pubkey recipient, int timeout) {
    // Allow the recipient to claim their received money
    function transfer(sig recipientSig) {
        require(checkSig(recipientSig, recipient));
    }

    // Allow the sender to reclaim their sent money after the timeout is reached
    function timeout(sig senderSig) {
        require(checkSig(senderSig, sender));
        require(tx.time >= timeout);
    }
}
```

TIP: read more about the CashScript language syntax in the Language Description.

Integrating into JavaScript

While more detailed examples are available on GitHub, we show an integration of the TransferWithTimeout contract in a JavaScript project.

After compiling the contract file to an artifact JSON with cashc, it can be imported into the CashScript SDK.

```
cashc ./transfer_with_timeout.cash --output ./transfer_with_timeout.json
```

```
const { ElectrumNetworkProvider, Contract, SignatureTemplate } = require('cashscript');
```

```
const { alice, bob, alicePk, bobPk } = require('./keys');


async function run() {
  // Import the TransferWithTimeout JSON artifact
  const artifact = require('./transfer_with_timeout.json');


  // Initialise a network provider for network operations
  const provider = new ElectrumNetworkProvider('mainnet');


  // Instantiate a new TransferWithTimeout contract
  const contract = new Contract(artifact, [alicePk, bobPk, 600000], provider);


  // Call the transfer function with Bob's signature
  // i.e. Bob claims the money that Alice has sent him
  const transferDetails = await contract.functions
    .transfer(new SignatureTemplate(bob))
    .to('bitcoincash:qrhea03074073ff3zv9whh0nggxc7k03ssh8jv9mkx', 10000)
    .send();
  console.log(transferDetails);


  // Call the timeout function with Alice's signature
  // i.e. Alice recovers the money that Bob has not claimed
```

```
  const timeoutDetails = await contract.functions

    .timeout(new SignatureTemplate(alice))

    .to('bitcoincash:qqeht8vnwag20yv8dvtcrd4ujx09fwxwsqqqw93w88', 10000)

    .send();

  console.log(timeoutDetails);

}
```

TIP: read more about the JavaScript SDK in the SDK documentation

https://radiant4people.com/programming/radiantscript/basics/getting-started/

3.5 Writing Covenants & Introspection

Covenants are all the rage in Bitcoin Cash smart contracts. But what are they, and how do you use them? In one sentence: a covenant is a constraint on how money can be spent. A simple example is creating a smart contract that may only send money to one specific address and nowhere else. The term Covenant originates in property law, where it is used to constrain the use of any object - or in the case of BCH, the use of money. Bitcoin covenants were first proposed in a paper titled Bitcoin Covenants, but several other proposals have been created over the years. In May of 2022 Bitcoin Cash implemented so-called Native Introspection which enables efficient and accessible covenants.

https://radiant4people.com/programming/radiantscript/guides/covenants/

3.6 Syntax Highlighting

When developing smart contracts for CashScript it is useful to have the proper syntax highlighting in your code editor / IDE. If you use Visual Studio Code, there is a dedicated CashScript extension. For other editors it is recommended to install a Solidity highlighting plugin and associate it with .cash files in your editor, since the syntaxes of the two languages are very similar.

https://radiant4people.com/programming/radiantscript/guides/syntax-highlighting/

3.7 Artifacts

Compiled contracts can be represented by so-called artifacts. These artifacts contain all information that is needed to interact with the smart contracts on-chain. Artifacts are stored in .json files so they can be shared and stored for later usage without having to recompile the contract.

TIP: Did you know? Artifacts allow any third-party SDKs to be developed, since these SDKs only need to import and use an artifact file, while the compilation of the contract is left to the official cashc compiler

Artifact specification

```
interface Artifact {
  contractName: string // Contract name
  constructorInputs: AbiInput[] // Arguments required to instantiate a contract
  abi: AbiFunction[] // functions that can be called
  bytecode: string // Compiled Script without constructor parameters added (in ASM format)
  source: string // Source code of the CashScript contract
  compiler: {
    name: string // Compiler used to compile this contract
    version: string // Compiler version used to compile this contract
  }
  updatedAt: string // Last datetime this artifact was updated (in ISO format)
}
```

```
interface AbiInput {
  name: string // Input name
  type: string // Input type (see language documentation)
}

interface AbiFunction {
  name: string // Function name
  inputs: AbiInput[] // Function inputs / parameters
}
```

3.8 Contract Structure

Contracts in CashScript are somewhat similar to classes in object-oriented languages. A notable difference is that there is no mutable state. So once a contract is instantiated with certain parameters, these values cannot change. Instead, functions can be called on the contract that act on the contract's values to spend money from the contract. The extension of CashScript source code files is .cash, and the structure of these source files is explained below.

https://radiant4people.com/programming/radiantscript/language/contracts/

3.9 Examples

- **Transfer With Timeout**
- **HodlVault**
- **Licho's Mecenas**

An extensive collection of examples is available in the GitHub repository. Below we discuss a few of these examples in more details and go through the functionality.

https://radiant4people.com/programming/radiantscript/language/examples/

3.10 Global Functions & Operators

CashScript has several functions builtin for things like cryptographic and arithmetic applications. It also includes many common operators, although some important ones are notably missing due to the limitations of the underlying Bitcoin Script.

https://radiant4people.com/programming/radiantscript/language/functions/

3.11 Global Variables

An integer literal can take a suffix of either monetary or temporary units to add semantic value to these integers and to simplify arithmetic. When these units are used, the underlying integer is automatically multiplied by the value of the unit. The units sats, finney, bits and bitcoin are used to denote monetary value, while the units seconds, minutes, hours, days and weeks are used to denote time.

CAUTION: Be careful when using these units in precise calendar calculations though, because not every year equals 365 days and not even every minute has 60 seconds because of leap seconds

https://radiant4people.com/programming/radiantscript/language/globals/

3.12 Language Grammar

https://radiant4people.com/programming/radiantscript/language/grammar/

3.13 Types

CashScript is a statically typed language, which means that the type of each variable needs to be specified. Types can also be implicitly or explicitly cast to other types. For a quick reference of the various casting possibilities, see Type Casting.

https://radiant4people.com/programming/radiantscript/language/types/

3.14 SDK Examples

- Transfer With Timeout
- Memo.cash Announcement

An extensive collection of examples is available in the GitHub repository. Below we discuss a few of these examples in more details. These examples focus mainly on the use of the SDK, while the Examples page in the language section focuses more on the CashScript syntax.

https://radiant4people.com/programming/radiantscript/sdk/examples/

3.15 Contract Instantiation

Before interacting with smart contracts on the BCH network, the CashScript SDK needs to instantiate a Contract object. This is done by providing the contract's information and constructor arguments. After this instantiation, the CashScript SDK can interact with BCH contracts.

https://radiant4people.com/programming/radiantscript/sdk/instantiation/

3.16 Sending Transactions

When calling a contract function on a Contract object, an incomplete Transaction object is returned. This transaction can be completed by providing a number of outputs using the to() or withOpReturn() functions. Other chained functions are included to set other transaction parameters.

Most of the available transaction options are only useful in very specific use cases, but the functions to(), withOpReturn() and send() are commonly used. withHardcodedFee() is also commonly used with covenant contracts.

https://radiant4people.com/programming/radiantscript/sdk/transactions/

3.17 Migration Notes

- v0.6 to v0.7
- cashc compiler

The older preimage-based introspection/covenants have been replaced with the newly supported native introspection/covenants. This has significant consequences for any existing covenant contracts, but in general this native introspection makes covenants more accessible, flexible and efficient. See below for a list of changes. In some cases there is no one to one mapping between the old introspection and the new introspection methods, so the logic of the smart contracts will need to be refactored as well.

Most importantly, it is now possible to access specific data for all individual inputs and outputs, rather than e.g. working with hashes of the outputs (tx.hashOutputs). This offers more flexibility around the data you want to enforce. For more information about this new native introspection functionality, refer to the Global covenant variables section of the documentation, the Covenants guide and the Native Introspection CHIP.

https://radiant4people.com/programming/radiantscript/sdk/transactions/

3.18 Release Notes - LINK

CHAPTER IV – GUIDES & LINKS (LEGEND)

4.1 Electron Wallet, First Use - LINK

Restore Wallet - LINK / Create Multisig - LINK

4.2 Compile RXD Node with Ubuntu 22.04

Radiant node source code: LINK

Original guide to compile in Ubuntu: CLICK ME / ... AND ME

4.3 Deploy a Radiant ElectrumX Server or Node on Flux - LINK

4.4 How To Buy Guide on Tradeogre (made by gsb) - LINK

4.5 General Disclaimer about CEX, Centralized Exchanges

Such platforms, as the name says, are Centralized, usually anonymous or registered in nations with dubious standards about financial regulation, with security flaws more or less evident, and often discontinuous conducts or fishy behaviors. Some of them get hacked, by external inside job and so with those hackings, users lose money and often all their life savings. It can happen that such CEX after such losses can recover funds and it happened that they can also reimburse users, using their trading fees over time.

The majority of the times they go down, and burn bridges to the impacted communities. This can happen to any centralized reality also outside of the financial field and we remember the case of nicehash, that has been object of hacking, but it recovered in the coming year.

Your/our safety is the first, absolute, priority, so remember well the following statement:

"Not your (private) keys, not your coins"

- Centralized Exchanges (CEX) can be used just to buy and transfer, or to sell, convert and transfer

- Centralized Exchanges ARE NOT a store of value, as you would not give your wallet to a stranger for the management of your cash, debit cards and documents

“Stay safe and vigilant out there”

Interesting video: https://www.youtube.com/watch?v=Kz-jPjMvIu4

RXD | 100X Crypto Coin | AI | New Bitcoin?💥TOP Secrets | Rich |

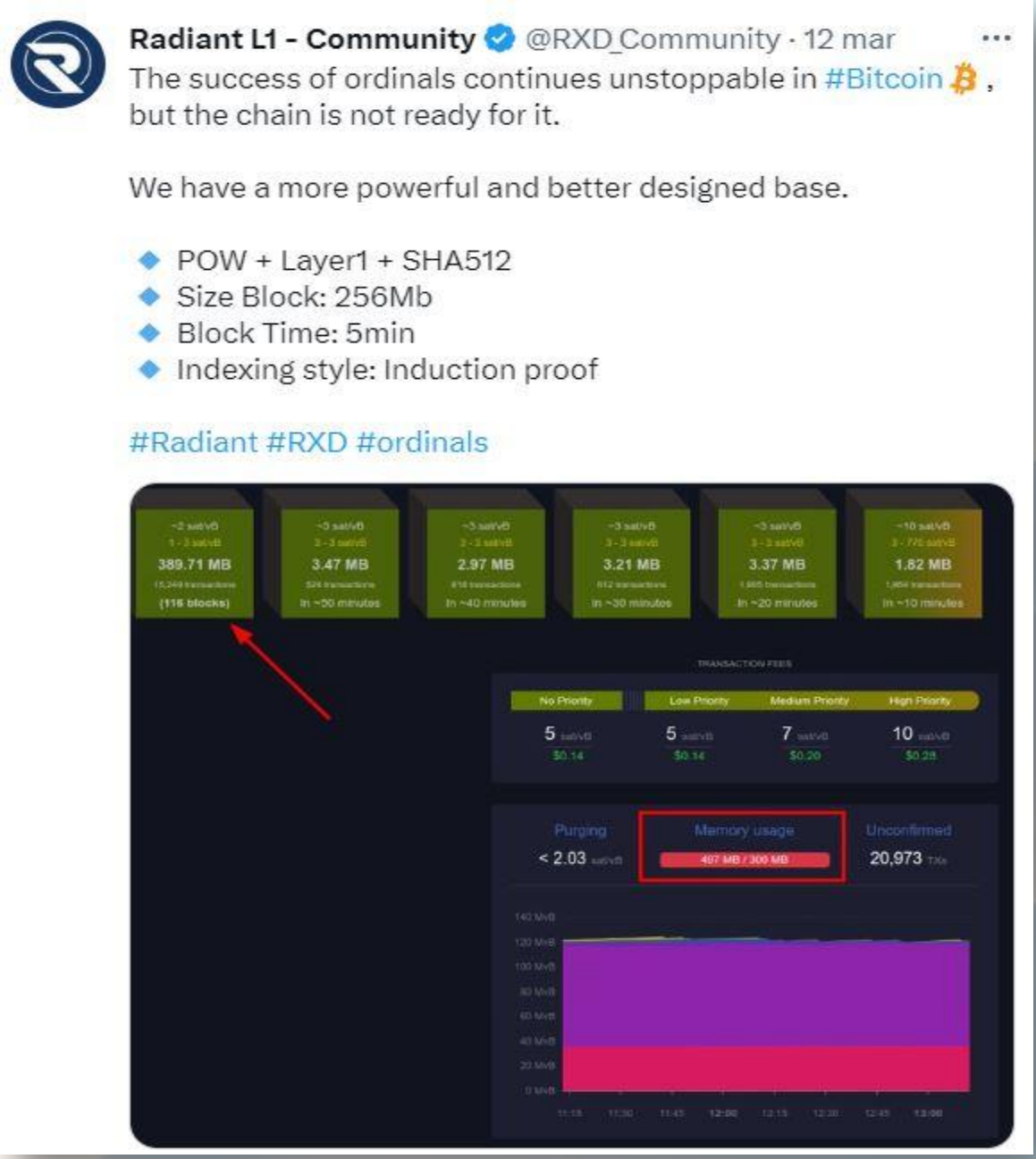
CHAPTER V - INFOGRAPHICS & TWEETS (LEGEND)

5.1 RADIANT COMPARISON CHART – POW/POS

DATE	11/12/2022	Radiant	Kaspa	Kadena	Solana	Ethereum	Bitcoin
1	Consensus Mechanism	POW	POW	POW	POS	POS	POW
2	Launch Time	June 21, 2022	November 7, 2021	Mid 2020	March 16, 2020	July 30, 2015	January 8, 2009
3	Coin distribution in %	16%	52%	21%	69%	infinite	92%
4	Coin Owners Decentralization	Still early, low	Still early, low	Low	VC, low	VC, very low	Average
5	Miner/Staker Decentralization	lower than average	lower than average	Asic on the network, low	Low	Low	Asic, average
6	Circulating Supply	3,259,256,884	15,039,852,353	212,017,319	366,114,059	122,373,866	19,230,137
7	Max coin Supply (Hard Cap)	21,000,000,000	28,704,026,601	1,000,000,000	533,872,038	infinite	21,000,000
8	Reward distribution POW/POS	100% POW	100% POW	70/30 POW/EarlyINV*	100% POS	100% POS	100% POW
9	Scalability	Very high	Very high	Very high	High but issues	Low	Low
10	Security	Still early, low	Still early, lower than avg	Still early, avg	Issues	High	Very high
11	Transaction Fee	very low	very low	very low	very low	very high	Low
12	Transaction Finality	5 min (instant if 0conf)	1 sec	1.5 sec	1 sec	5 mins	10 mins
13	Transactions Per Second	4k up to 500k	4k	480k	50k	17	7
14	Blocks Per Second	5 min/blk	1, soon at least 5bps	30 seconds/blk	3 seconds/blk	12 seconds/blk	10 min/blk
15	Block Reward	50000	319	1.0385	/ (POS)	/ (POS)	6.25
16	Transaction Goverance	Miners	Miners	Miners	Staking validators	Staking	Miners
17	Governance Supervision	Full node Ops	Full node Ops	Chainweb Nodes	Validators KYC	Full node Ops	Full node Ops
18	Smart Contract Capability	Yes	Not, planned	Yes	Yes	Yes	Yes, Possible!
19	Fair Launch	Yes	***	Yes	Not at all !	Not, Huge premine	Yes
20	Premine	Not	Not	Yes, but limited	Yes, extremely	Yes, heavily	Not (early mining)
21	Initial stages Devs Mining	Early mining	Early VC cloudmining	Might have been**	Not	Might have been**	Might have been**
22	ICO	Not	Not, but there was a VC	Yes	Yes	Yes	Not
23	Circulating Supply Market Cap	5,476,454	110,290,872	215,169,097	5,000,453,116	155,745,407,346	330,159,456,006
24	Fully Diluted Market Cap	35,285,813	210,493,565	1,014,865,666	7,317,503,961	infinite	360,545,979,268
25	Halving Time	2 years	1 year	1 year	/ (POS)	/ (POS)	4 years
26	VC Presence	Not	N/A***	Yes	Yes	Yes	Not
27	ASIC	Not	They planned initially	Yes	Not	Not	Yes
28	0conf Support	Yes	Not needed, fast	Not needed, fast	Not needed, fast	Not	Not
29	X Factor vs KDA CSMC (Potential)	39.3	2.0	1.0	/	/	/
30	X Factor vs KDA FDMC (Potential)	28.8	4.8	1.0	/	/	/
NOTES		*: early investors and team	**: it is uncertain if the developers have mined massively initially, it might have been possible, as in every POW crypto				
	***: since the inception there has been a VC (DagLabs) that certified a Cloudmining operation up to 3% of the HC of KAS, as the ASIC development - https://twitter.com/JoshMetnick						

5.2 Ordinals/Atomicals?

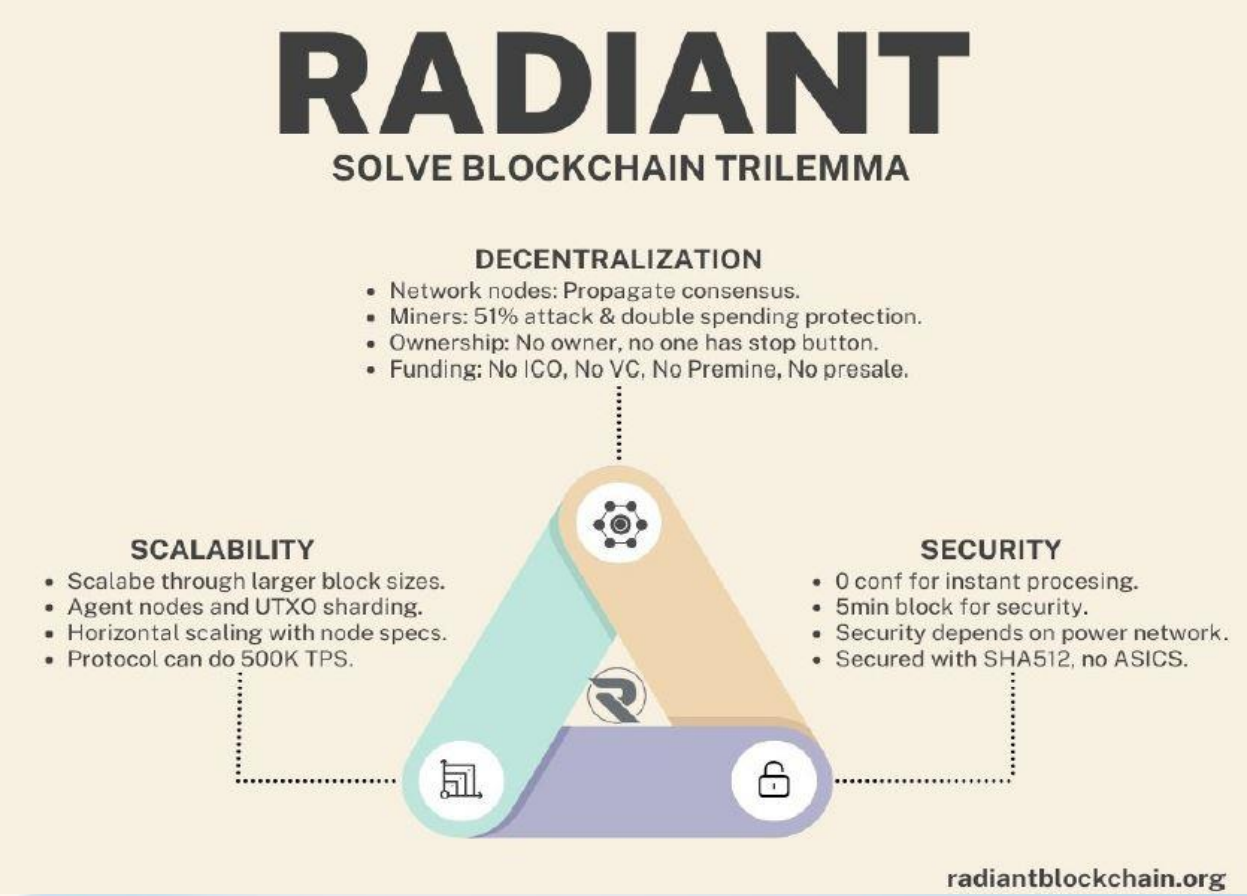
Powerful for Bitcoin, amazing for Radiant!



5.3 Is Radiant solving the blockchain trilemma?

A very interesting paper describing the issue: TheBlockchainTest.com

- Scalability is the ability of the blockchain to accommodate a higher volume of transactions
- Security is the ability to protect the data held on the blockchain from different attacks or blockchain's defence against double-spending
- Decentralization is the redundancy in the network that makes sure fewer entities do not control the network Blockchain trilemma or scalability trilemma is often just stated as a rule, which is not the case. It is not necessary that blockchain may never achieve optimum levels of decentralization, security, and scalability



5.4 TheCryptoVigilante Report Extract over Radiant

With their authorization, we can publish few screens describing the genesis of Radiant, by Attoshi. Very interesting reading.

Radiant's Backstory

To some extent like Bitcoin, Radiant's origins are a bit mysterious. No one person or group of people claims credit for its creation. All we know is that one person, a developer (who goes by the name Atoshi) was in some way involved in its development. Whether Atoshi worked alone, or with a group is unclear - but once the project was finished, he (and the other creators?) set it free in the world and then walked away from it - sparing the project from burdens of ego, personality and hubris. We invite you to listen to the one and only interview Atoshi ever gave before 'disappearing' back into the mists.

Note: while he may have been only peripherally involved in Radiant's creation, for literary convenience (and the lack of any other name to point to) we will reference Atoshi as the creator in describing the project's history.

Atoshi was clearly very familiar with development on Bitcoin and its forked chains (BCH, and especially BSV). While he saw the possibility within Satoshi's code for Bitcoin becoming the world's most powerful decentralized supercomputer (far beyond projects like Ethereum), he was frustrated with a fundamental problem it faced - something called the **'Back To Genesis Problem'**. In short, while Bitcoin did a great job of validating its *coins*, it had a real problem validating any other assets on the blockchain.

The beauty of Bitcoin's technology is that you are able to easily validate that your coins belong to you. You can mathematically prove your coins are legitimate tokens on the network, tracing all the way back to their creation (or genesis) without having to index the whole blockchain or rely on a third party. In many ways, this is arguably a core feature which enables Bitcoin to be considered sound, decentralized money.

But when it comes to any other kind of digital asset - such as tokens (or NFTs), the same methods don't apply. In Bitcoin's protocol, there is no way to easily validate that tokens or NFTs are authentic on-chain. One must instead use a third-party indexer or oracle - something anathema to the spirit of trustlessness embedded in the Bitcoin ethos.

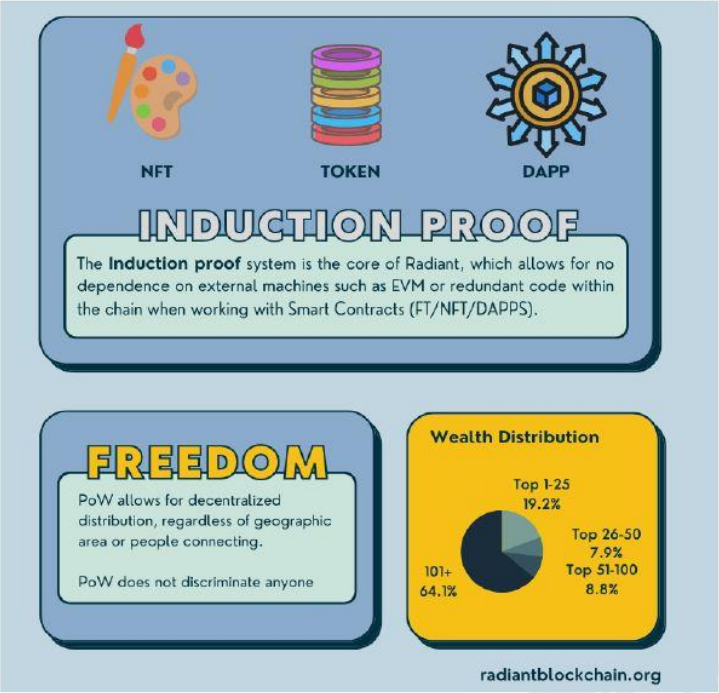
It's true that in a couple of years, BSV may have a workable solution to this problem through the development of their nChain protocols - a zero-knowledge (ZK) proof system. However, it would likely

[5] Rather than personal scheming to 'get rich with a new token launch'.
[6] The circumstances of its creation may in some way be connected to its potential. Philosophically, one can argue that power is only properly wielded by those who do not wish to use it for personal gain. In this light, Radiant's modest beginnings may have impressive implications.

still be clunky and with limited tooling, which means the developer learning curve will be high. Plus it would only be applicable to simple use cases of tokens.

Furthermore, generalizing the process to support more advanced DeFi applications could very easily take an additional 1-2 years of development. At that point, we have to ask: what open source developers will want to build with that complex system when they could just use one of a half dozen other fully capable blockchains that are not encumbered by patents? The 'ship will have long-since sailed'.

Thankfully, another technique started to take shape after Atoshi recognized a pattern in the way tokens and NFTs could be uniquely identified. The transaction ID and output index (called an Outpoint) forms a globally unique 36 byte identifier and it is only needed to carry that identifier along to the next transaction. If there was a new programming instruction (OP code) added to the blockchain node, it could ensure that any token/NFT issuance (or minting) could take place if and only if at least one of the parent inputs' outpoints was specified as the asset ID. In this way, the induction problem (the 'Back to Genesis Problem') was solved elegantly.



There are other nuances and details that play a part, but by and large they support the above description. In theory, this one programming instruction would be all that was needed to gain all the superpowers of Ethereum and other blockchains while keeping fees low and the blockchain throughput maximally parallel.

Though he solved the **Back to Genesis** problem, Atoshi now faced a new problem - what to do with the solution? Should he seek to get it implemented on BSV - the blockchain which had prompted him to seek a solution?

Not Another Blockchain Network Fork

In some respects, Radiant *looks* like a fork of other Bitcoin protocols. Its code is, to a large degree, based upon Bitcoin's and carries the incredible power of UTXO design. But as touched upon previously, a blockchain network fork (like those which were born from the Hash Wars) is not what was needed. This approach would have carried all the baggage associated with Bitcoin's multitudinous web of political and economic issues. It would have carried over the millions of already mined BTC, BCH and BSV tokens. It would have carried over the vast (and concentrated) armies of ASIC miners already engaged in those other projects. In other words, it wouldn't be free. A fresh start was needed.

So Atoshi decided on a new blockchain with its own *genesis block*, and he looked to BCH's codebase as the starting point.

Why not BSV? Well, in short, because Craig Wright had so thoroughly complicated BSV's code with legal entanglements that it wasn't worth the risk to even touch. And besides - BCH's codebase had all the pieces of the original Bitcoin design that would be needed to begin.

Even more than this though, by starting fresh, Atoshi drew upon the industry's learnings from other successful projects, and sought to harmonize them with the original core Bitcoin design. He was able to incorporate features (or potentials) for any number of advances that have been made since Bitcoin's birth. From sharding techniques to customizable node types, Radiant as a project invites the best features of all cryptocurrencies - even those with which it may disagree in other areas (like Ethereum and Cardano).

Resetting the Miner Community (New Algorithm, Shorter Blocktimes)

While starting from a new genesis block meant that Radiant wouldn't be burdened with a previously established (and arguably compromised) token distribution profile, there were other legacy risks to be considered.

If RXD used the same hashing algorithm as BTC/BCH/BSV, then there would be nothing to stop those miner communities from simply 'switching' to RXD and taking early ownership and control. RXD needed to use something different from Bitcoin's SHA256 hashing algorithm - and it found it in the next generation: SHA512.

While SHA512 is arguably more secure cryptographically, the reality is that SHA256 is already effectively unbreakable. The reason for its selection therefore is not to do with technological protection. Instead, it's because the vast armies of SHA256 miners currently working on BTC, BCH and BSV *physically cannot* be repurposed to a SHA512 algorithm. When it comes to mining RXD, those miners are *worthless*.

The playing field was starting out *level*.

Sure, at some point, SHA512 ASICs will surely arrive, but that won't be for some time in any meaningful way. And in the meantime, individuals, at home with GPUs, can mine competitively for a fair initial distribution.

Atoshi calculated that if every single currency, token, point, coupon, security, property and collectible was tokenized as 1 Satoshi, then, just on that basis, 60% of the 2.1 quadrillion Satoshis existing on BTC would be used up! This 1,000x token count provides future-proofing, with ample excess capacity to deal with growing capacity requirements.

Said differently, Radiant's token count allows for all-but-uncapped conversion of digital assets into NFTs through colored Photons (Photons are Radiant parlance for Satoshis), where each Photon is 1 unit of a token.

This future-proofing is yet another gift of Atoshi to the project - and is tied to his vision of Radiant reworking the global SaaS (software as a service) market.

Disrupting SaaS (Software as a Service)

What does it really mean to be the world's supercomputer? What is required for this to happen? And why has it *not happened yet?*

Atoshi had some very interesting ideas on this[8].

He posited that the reasons why blockchain has yet to fully express its *global compute potential* have to do with ethical and technological limitations which have plagued the space (until now).

The ethical limitations have to do with centralization and politics (as already discussed). The technological limitations though have to do with lack of scalability outside of the UTXO model.

As evidenced by Ethereum's outrageous fees, the only digital assets which can economically survive on-chain are 'high value speculation vehicles'. After all, if it costs $20-$50 simply to send a digital asset from one person to the other, then no 'token' worth less than that can *afford* to exist. This closes off whole *worlds* of business use-cases, which could otherwise be hosted in a decentralized fashion on blockchains.

By solving these problems, RXD offers the potential for decentralized digital economies to explode onto the scene - driven by nearly unfathomable amounts of decentralized tokens, tickets, licenses, and other

[7] The first halving event is set for little more than a year from now, in June 2024.
[8] If you haven't already clicked the link to listen to Atoshi's interview, we suggest you take some time to do so. Toward the end, he discusses a vision for what is possible in digital assets through Radiant.

forms of digital economic assets. In short, the currently *centralized* SaaS market (generating over $250 billion *per year* in revenues, and therefore *worth many trillions* of dollars today) is invited to migrate to a decentralized *on-chain* environment in Radiant. In fact, with consideration for the low cost and high benefits, it's *incentivized* to migrate.

The potential magnitude of this is so significant, we are reluctant to expound on it here, for fear of being considered hyperbolic. But it's possible that Radiant, in its current form, has barely scratched the surface of its potential. It could, in a different but analogous way, track the very footsteps of Bitcoin itself.

Investment Opportunity and Valuation

Investment Opportunity and Valuation

We see two primary ways to value Radiant. The first (and much more modest approach) is in comparison to what we call the 'low-hanging fruit' - BSV, and some 'start-up' L1 networks.

The second approach is far vaster in scope and possibilities - as not only a more broadly competitive L1 network, but one that has the potential to be a market leader.

Relative to BSV

RXD may be considered superior to BSV in several aspects:

- Technically
  Radiant, through induction proofs, has solved the 'Back to Genesis' problem in a way far more elegant (and useful) than BSV's nChain approach[9] - with enormous possibilities springing from this advance. Plus, by virtue of its redesign, its architecture has been optimized in a number of other ways - standing on the shoulders of developers and visionaries who came before.

- In Ethos
  Radiant offers a 'fresh start' in terms of genesis block, miner reset, and freedom from community and political entanglements (*especially* with regard to the controversial Craig S. Wright). Radiant's founder(s) built it, then gifted it to the community. *This* is the ethos of the real Satoshi, whoever he may be (or have been).

With consideration for these attributes, and the fact that developing on Radiant is so similar, we may anticipate that the BSV developer, application, and user community will increasingly look to migrate to Radiant. As such, BSV's market capitalization may be viewed as something of a relative value aspiration - with the understanding that by shedding BSV's baggage, RXD is actually poised to grow to

There are price predictions but you can find them registering in the TCV newsletter, by private access. Our interest was to share their extremely interesting Report, pointing to the main elements that make Radiant truly unique.

But we can't stop there. While it seems incredible to be faced with a possibility of this magnitude, the reality is that, at some point, Radiant could *directly challenge Ethereum's leadership position*. After all - it appears to have everything needed: superiority of technology and spirit - neither of which can be faked, and both of which should be deeply appreciated by a vast portion of the market (and world).

Consequently, while we urge a measured approach, we have to acknowledge Radiant's potential to have a radical and revolutionary impact on the world of crypto.

Were Radiant to approach Ethereum-level valuations (something we consider feasible with respect to its value

CHAPTER VI - RADIANT HISTORY & LINKS (LEGEND)

6.1 Achievements – Q2 2022 / Q2 2023

- WEBSITE: https://radiantblockchain.org/
- GitHub MAIN ACCOUNT: https://github.com/RadiantBlockchain
- WHITEPAPER: https://radiantblockchain.org/radiant.pdf
- FAUCET: https://radiant.liddlebit.com/
- GUIDES:
  - ✓ What is Radiant?: https://www.youtube.com/watch?v=B-UWB7iC7O4
  - ✓ System Design: https://github.com/RadiantBlockchain/radiant-node/blob/master/doc/whitepaper/radiant-system-design.md
  - ✓ Hashrate/Difficulty chart: https://radiant.ovh/grafica
  - ✓ Mining guide: https://github.com/RadiantBlockchain/rad-bfgminer/blob/master/MINING_RAD_GUIDE.md
  - ✓ Compile Node: https://github.com/Antares-RXD/Radiant-Guides/blob/main/Compile-Node/Ubuntu-22_04.md
  - ✓ How to use wallet: https://github.com/Antares-RXD/Radiant-Guides/blob/main/How-Use-Wallet/First-Use.md
  - ✓ Create Multisign Wallet: https://github.com/Antares-RXD/Radiant-Guides/tree/main/Create-multisign-wallet

- EXPLORER
  - ✓ Explorer 1: https://explorer.radiant.ovh/
  - ✓ Explorer 2: https://radiantexplorer.com/
  - ✓ Explorer 3: https://explorer.radiantblockchain.org/
  - ✓ Explorer 4: https://radiantscan.io/

- OFFICIAL SOCIAL MEDIA PLATFORMS
  - ✓ Medium: https://medium.com/@RadiantLayerOne
  - ✓ Reddit: https://www.reddit.com/r/RadiantBlockchain
  - ✓ Twitter: https://twitter.com/RadiantLayerOne
  - ✓ Telegram: https://t.me/RadiantBlockchain
  - ✓ Discord: https://discord.gg/dXMs6VCt6H

- ✓ Radtalk: https://radtalk.net/
- ✓ Bitcointalk: https://bitcointalk.org/index.php?topic=5414503.0;all

❖ WALLETS

- ✓ QT - Wallet Electron 0.1.3 (Windows/ Linux/Mac): https://github.com/RadiantBlockchain/electron-radiant/releases/tag/v0.1.3
- ✓ WEB WALLET - SAMARA 1.0.2 (browser wallet - Chrome/Brave): https://samara.app/

❖ EXCHANGES (**WHERE TO BUY**)

- ✓ Coinex: https://www.coinex.com/en/exchange/rxd-usdt/#spot
- ✓ Xeggex: https://xeggex.com/market/RXD_USDT
- ✓ Tradeogre BTC: https://tradeogre.com/exchange/RXD-BTC
- ✓ Tradeogre USDT: https://tradeogre.com/
- ✓ Exbitron: https://www.exbitron.com/
- ✓ Freiexchange: https://freiexchange.com/

❖ MARKET INFO AND CON AGGREGATORS LISTINGS

- ✓ Coinmarketcap: https://coinmarketcap.com/currencies/radiant/
- ✓ Coinsgecko: https://www.coingecko.com/en/coins/radiant
- ✓ Coinpaprika: https://coinpaprika.com/coin/rxd-radiant
- ✓ MiningPoolStats: https://miningpoolstats.stream/radiant
- ✓ Minerstats: https://minerstat.com/coin/RXD
- ✓ Hashrate.no: https://hashrate.no/
- ✓ Whattomine: https://whattomine.com/

❖ RADIANT POOLS SUPPORT

- ✓ Rxdpool: https://rxdpool.com/#rad1
- ✓ Rplant: https://pool.rplant.xyz/#radiant
- ✓ Poolmine: https://poolmine.tk/
- ✓ Deepfields: https://deepfields.io/
- ✓ Cryptopool: https://cryptopool.one/
- ✓ Vipor: https://vipor.net/

- ✓ Woolypooly: https://woolypooly.com/

❖ RADIANT MINER SUPPORT
- ✓ ccminer+hiveos (NVIDIA): https://github.com/radifier/radiator/releases
- ✓ rad-bfgminer (AMD/NVIDIA): https://github.com/RadiantBlockchain/rad-bfgminer/releases
- ✓ srbminer(AMD): https://github.com/doktor83/SRBMiner-Multi/releases
- ✓ bzminer: https://www.bzminer.com/
- ✓ wildrig: https://github.com/andru-kun/wildrig-multi/releases

❖ ADDED FUNCTIONALITY (Induction Proofs via new OP Codes)
- ✓ OP_PUSHINPUTREF
- ✓ OP_REQUIREREF
- ✓ OP_DISALLOWPUSHREF
- ✓ OP_DISALLOWPUSHREFSIBLINGOUTPUT

❖ Interesting Podcast with the main developer ATTOSHI, with Brittany Bitz:

https://www.youtube.com/watch?v=B-UWB7iC7O4

❖ WALLETS
- ✓ (MOBILE/WEB) CHAINBOW WEB3 WALLET INTEGRATION https://chainbow.medium.com/chainbow-wallet-for-web3-faithful-to-the-original-bitcoin-vision-2d75670937f
- ~~✓ (WEB) SAMARA WEB WALLET NFT MINTING, FOUNDATIONAL IMPROVEMENTS FOR THE SAMARA NETWORK AND TOKENIZED COMMUNITIES~~ (**currently under update, do not use**) https://twitter.com/SamaraNetwork/status/1615842655233081351?t=B8LdTJFtr_JpEkAsRSWqxw&s=19

❖ FLUXnodes INTEGRATION ! Start of the collaboration between RADIANT AND FLUX !

- ✓ Radiant node and ElectrumX #Radiant node have been added to the $Flux marketplace. https://home.runonflux.io/apps/marketplace/crypto

- ❖ XEGGEX EXCHANGE LISTING COMPLETED
- ❖ REP20: Fungible Token Standard for Radiant Published.
  - ✓ https://github.com/RadiantBlockchain/reps/blob/main/rep-0020.mediawiki
- ❖ REP21: Non-Fungible Token (NFT) Standard for Radiant Published
  - ✓ https://github.com/RadiantBlockchain/reps/blob/main/rep-0021.mediawiki
- ❖ TANGEM TALKS: integration of Radiant in their ecosystem
- ❖ NEW EXPLORER: https://radiantscan.io/
- ❖ ISO20022: internal discussions about a potential Radiant future compatibility

6.2 Radiant Community Roadmap (natural updates and voting)

6.2.1 Q4 2023 / Q2 2024 objectives

- ➢ FINALIZATION OF THE TOKEN STANDARDS
- ➢ RADIANT DEVON (Develop-On-Radiant) CONTESTS every 5 months (1.5 entry / 3 development / 0.5 vote)
- ➢ MARKETING PAPERS for YT Content Creators
- ➢ DISCORD / TWITTER / FACEBOOK SHARE & LIKE CONTESTS
- ➢ Community votes about NEW EXCHANGE T3 CEX or DEX LISTING
- ➢ TECHNICAL WIKI UPDATES (https://radiant4people.com/
- ➢ NON FUNGIBLE TOKEN STANDARD DEFINED
- ➢ TYPESCRIPT LIBRARY AND COMMAND LINE INTERFACE FOR NFT QUERIES AND TRANSACTIONS
- ➢ FUNGIBLE TOKEN STANDARDS DEFINED
- ➢ NEW COLLABORATIONS WITH CEX/DEX EXCHANGES
- ➢ PARTNERSHIPS WITH ASSOCIATED MARKETS
- ➢ RADIANTSCRIPT BETA TESTING
- ➢ ELECTRUMX UPDATED TO SUPPORT CONTRACT QUERIES

The intention is to setup the RADIANT DEVON and GAMEON as standalone events, community funded. Such events should be initiated around the beginning of 2024, and keep

going on for the years to come, allowing a continuously development and infrastructure buildup over the network.

6.2.2 Q3 2024 / Q2 2025 objectives

- NFT PROJECTS ASSOCIATIONS AND JOINT VENTURES
- START OF "GAMING ON RADIANT", NFT BASED WEBGAMES
- RADIANT GAMEON (Game-On-Radiant) CONTESTS, building games over Radiant L1 chain
- OPEN SOURCE WEB BASED TOKEN WALLET WITH MINTING INTERFACE
- TOKEN EXPLORER
- NFT STANDARD EXTENDED TO SUPPORT MINER VALIDATED MUTABLE TOKENS
- ELECTRON RADIANT UPDATED TO SUPPORT SENDING AND RECEIVING TOKENS
- TOKEN MARKETPLACE
- STANDARDS DEFINED FOR PURELY PEER TO PEER TRANSACTIONS
- RADIANT P2P CONTRACT DEVELOPMENT FRAMEWORK
- POTENTIAL COLLABORATIONS WITH GOOGLE PLAY AND IOS DEVELOPERS
- NEW COLLABORATIONS WITH CEX/DEX EXCHANGES
- ETHEREUM SC CONVERTER (Migrate-Over-Radiant)
- SC L2 TOKENS OVER RADIANT L1

6.3 Potential Development Ventures

- RADIANT SCRIPT (ALPHA TESTING) – Developers wanted!
- Radiant DEX (Radiant SWAP) & OTHERS (Check the Brainstorming Discord section)
- ELECTRON WALLET/NODE IMPROVEMENTS
- ELECTRON WALLET SC 0confV2 Improvement Proposals
  - Chainbow 0conf native implementation, tested with success
- CURRENT FUNDING PROPOSALS
  - 1M RXD for a depth chart of Radiant https://discord.com/channels/990271820101988362/1036004939224330332
  - 5M RXD for community dev website https://discord.com/channels/990271820101988362/1040874969695006740
  - 1M RXD for a Radiant NFT standard
    - https://discord.com/channels/990271820101988362/1055981505454813326
- TIER 2 EXCHANGES ANALYSIS AND INTERNAL EXAMINATION

RADIANT ROADMAP

Q4 2023

- FINALIZATION OF THE TOKEN STANDARDS
- RADIANT DEVON (DEVELOP-ON-RADIANT) CONTESTS EVERY 5 MONTHS (1.5 ENTRY / 3 DEVELOPMENT / 0.5 VOTE)
- MARKETING PAPERS FOR YOUTUBE CONTENT CREATORS
- DISCORD / TWITTER / FACEBOOK SHARE & LIKE CONTESTS
- COMMUNITY VOTES ABOUT NEW EXCHANGE T3 CEX OR DEX LISTING
- TECHNICAL WIKI UPDATES (HTTPS://RADIANT4PEOPLE.COM/)

Q1

- NON FUNGIBLE TOKEN (NFT) STANDARD DEFINED
- TYPESCRIPT LIBRARY AND COMMAND LINE INTERFACE FOR NFT QUERIES AND TRANSACTIONS
- FUNGIBLE TOKEN (FT) STANDARDS DEFINED

Q2

- NEW COLLABORATIONS WITH CEX/DEX EXCHANGES
- PARTNERSHIPS WITH ASSOCIATED MARKETS
- RADIANTSCRIPT BETA TESTING
- ELECTRUMX UPDATED TO SUPPORT CONTRACT QUERIES

Q3 2024

- NFT PROJECTS ASSOCIATIONS AND JOINT VENTURES
- START OF "GAMING ON RADIANT", NFT BASED WEBGAMES
- RADIANT GAMEON (GAME-ON-RADIANT) CONTESTS, BUILDING GAMES OVER RADIANT L1 CHAIN

Q4

- OPEN SOURCE WEB BASED TOKEN WALLET WITH MINTING INTERFACE
- TOKEN EXPLORER
- NFT STANDARD EXTENDED TO SUPPORT MINER VALIDATED MUTABLE TOKENS
- ELECTRON RADIANT UPDATED TO SUPPORT SENDING AND RECIEVING TOKENS

Q1 2025

- TOKEN MARKETPLACE
- STANDARDS DEFINED FOR PURELY PEER TO PEER TRANSACTIONS
- RADIANT P2P CONTRACT DEVELOPMENT FRAMEWORK
- POTENTIAL COLLABORATIONS WITH GOOGLE PLAY AND IOS DEVELOPERS
- NEW COLLABORATIONS WITH CEX/DEX EXCHANGES

Q2

- ETHEREUM SC CONVERTER (MIGRATE-OVER-RADIANT)
- SC L2 TOKENS OVER RADIANT LAYER-1

Made By SHaRD. Tech Papers and Radiant4People by ANTARES
(Thanks to HippiePyro, Deathmumi, gsb, ArtOfSatoshi, CryptoNaut)

6.4 A Glimpse of the Future

- Electrumx update
  - Support queries and subscriptions by contract hash and singleton ref.
- NFT standard v1
  - Script and encoding standard defined for minting and transferring immutable tokens.
- Photonic (**Brief Preview here – LINK**)
  - Open source web based token wallet with minting interface, utilizing updated Electrumx.
- Block explorer update
  - Update existing block explorer software to handle standard token scripts.
- Pay to contract
  - Standard script for paying to a contract address, with a new contract address format.
- Contract Explorer
  - New web interface for exploring standard Radiant contracts.
- FT standard
  - Script and encoding standard defined for minting and transferring fungible tokens.
- Contract Management CLI
  - Typescript library and command line interface for contract deployment and token minting.
- Electron Radiant update
  - Support sending and receiving tokens.
- NFT standard v2
  - NFT standard extended to support miner validated mutable tokens, with capability to define update rules in custom contracts.
- Radiant Contract Studio
  - Contract management interface for deploying and managing large collections of contracts/tokens.
- Radiant Token Marketplace
  - Buy and sell Radiant tokens.
- P2P Foundation

- Standards defined for purely peer-to-peer transactions and off-chain code.

❖ P2P Development Framework
- Peer-to-peer contract development framework.

❖ P2P Wallet
- Fully peer-to-peer transactions without an indexer. Support for custom scripts and off-chain code execution.

CHAPTER VII – MINING (LEGEND)

Given the excellent work of Hashrate.no (link-hashrate-no) we will report the public known hashrates of the main Videocards, operating on Radiant. Due to the huge list, the graphs will be divided in 3 sections, based on the computing power.

Please follow the link to find the single GPU parameters, including them here would have implied a huge waste of space, that instead should be related to the project itself, and associated information.

We are suggesting you also to check the hashrate.no website due to the extremely good job that they have done, inserting precious information that rarely can be found in such category of websites.

There are 4 Boards sections:

- ✓ Entry Level VideoCards
- ✓ Mid Range VideoCards
- ✓ Enthusiast Range VideoCards
- ✓ Field Programmable Gate Arrays (FPGA, or FGPA for the average users)

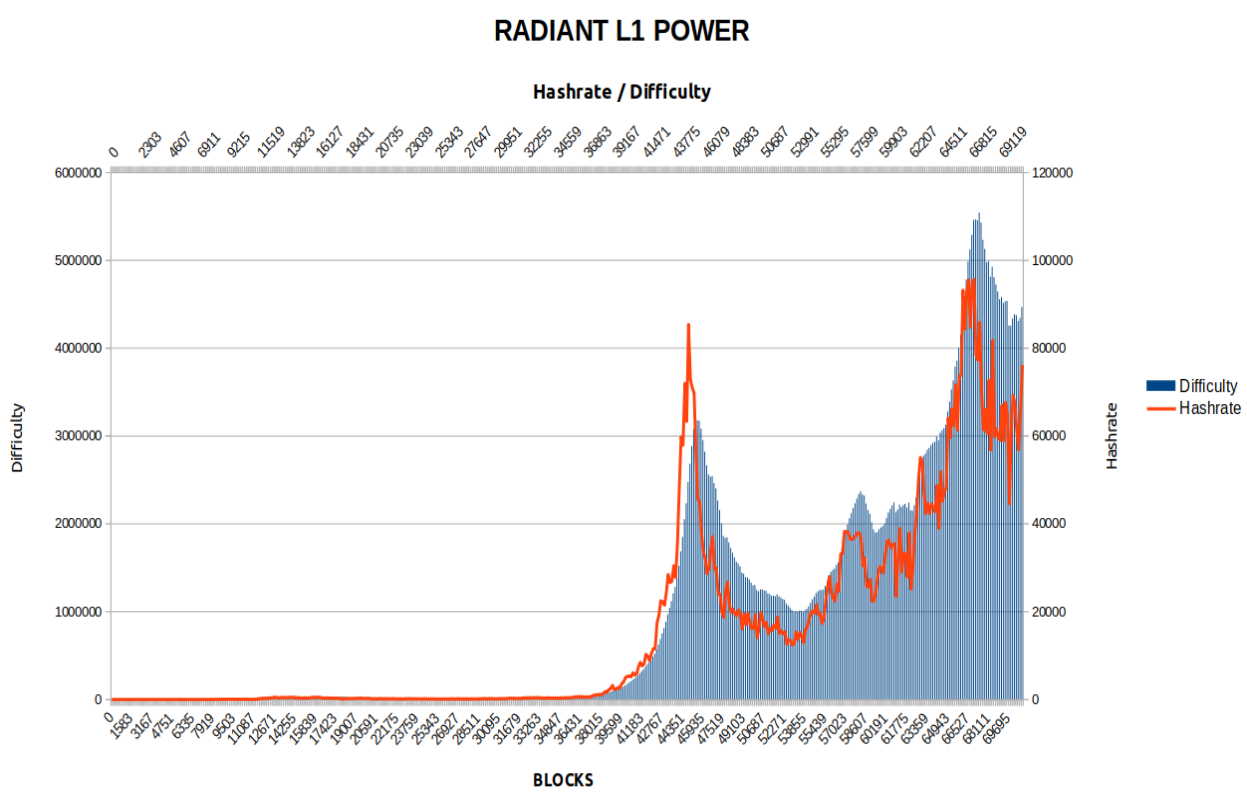
5 Comparison charts:

- ✓ MINER HELL, part I - Single GPU systems MH/s
- ✓ MINER HELL, part II - 20KWh GPU Farm Competition
- ✓ MINER HELL, part III - Single FPGA systems MH/s
- ✓ MINER HELL, part IV - 20KWh FPGA Farm Competition
- ✓ MINER HELL, part V - 20KWh GPU VS FPGA all-out war

A Mining Guide paragraph with pools, batches and wallet settings.

… Plus few Visual GPU Farm ! RENDERS !

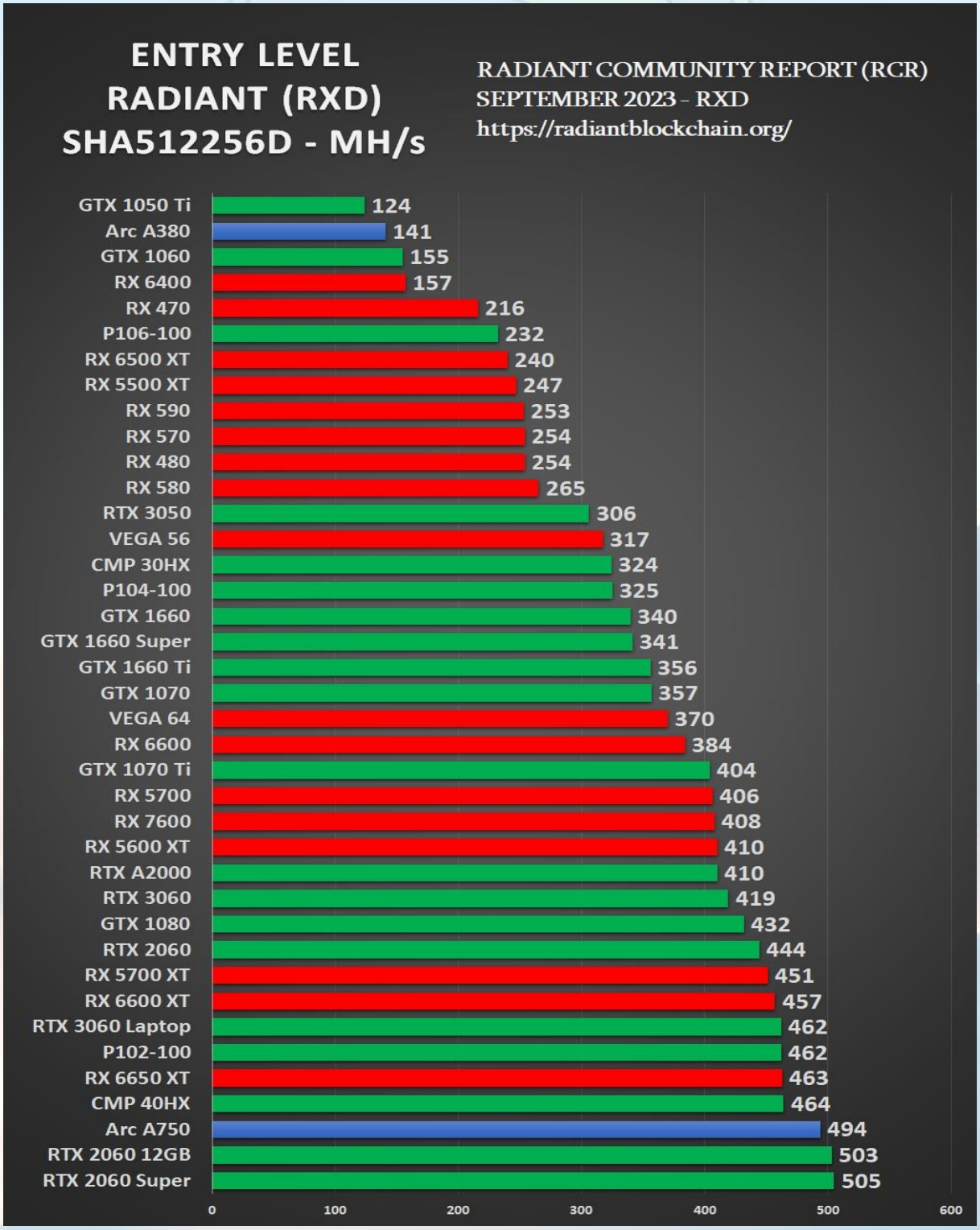
A mention about the algo: SHA512526-D. Core heavy, but more balanced toward FPGA, against GPU, to the point that the RTX 4090 is almost competing with FPGA, devices way more expensive, rare and technically speaking way more efficient, on paper. In other core heavy algorithms the difference can be easily more than twice the one present in Radiant, and this is an important factor that might have an impact later on, when the RXD/USDT will grow organically, due to the limited number of FPGA compared to GPU. As shown below in red we see an example of the difficulty retargeting of Radiant: in red the hashrate, in blue the difficulty. Reactive increments but without monolithic blocks that can create issues. Other chains during the ETH merge had massive issues, this is not the case with Radiant.



BALANCED GPU & FPGA MINING - 100 % POW

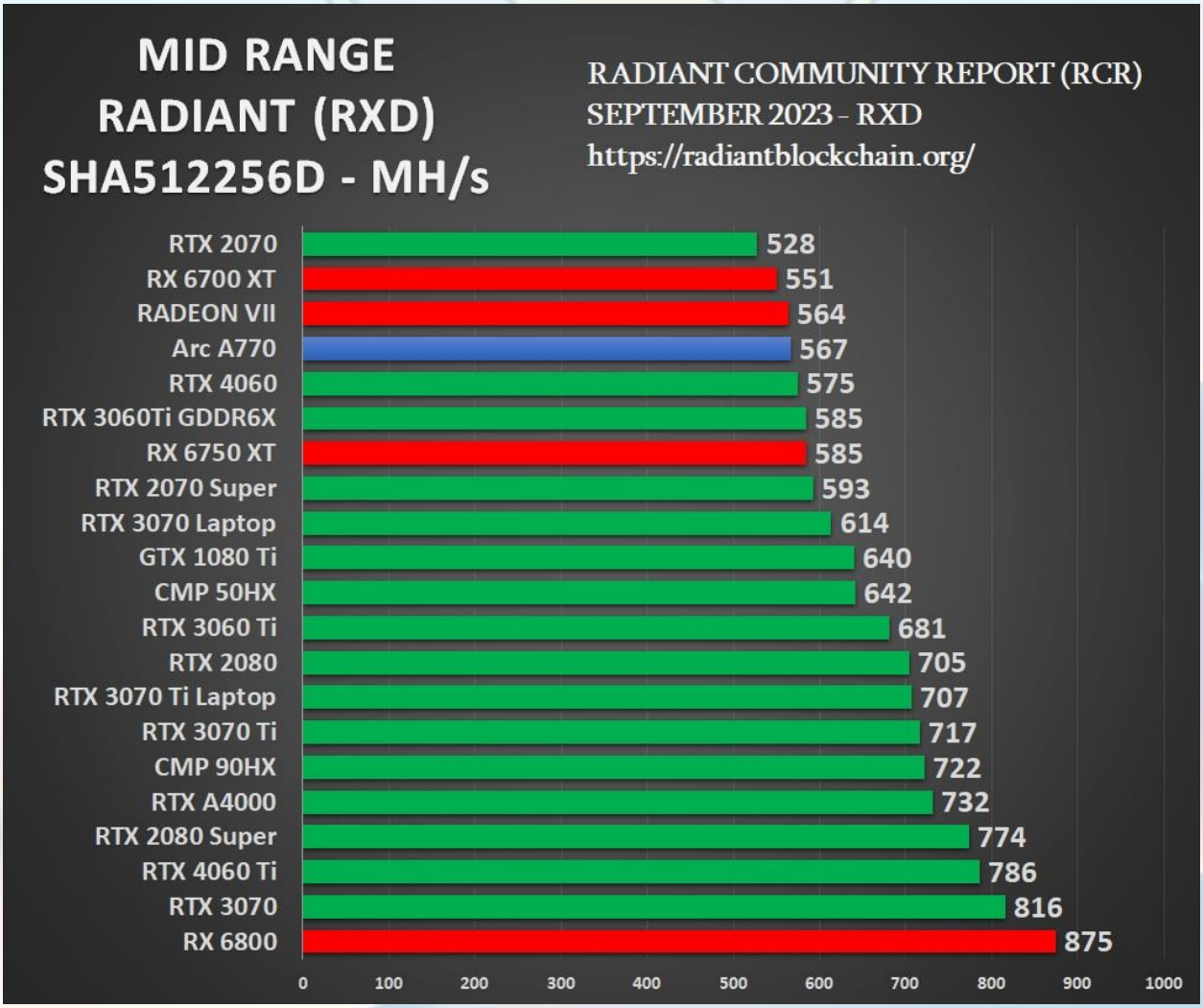
SMART DIFFICULTY RETARGETING ALGO (ASERT DAA)

7.1 Entry-Level VideoCards



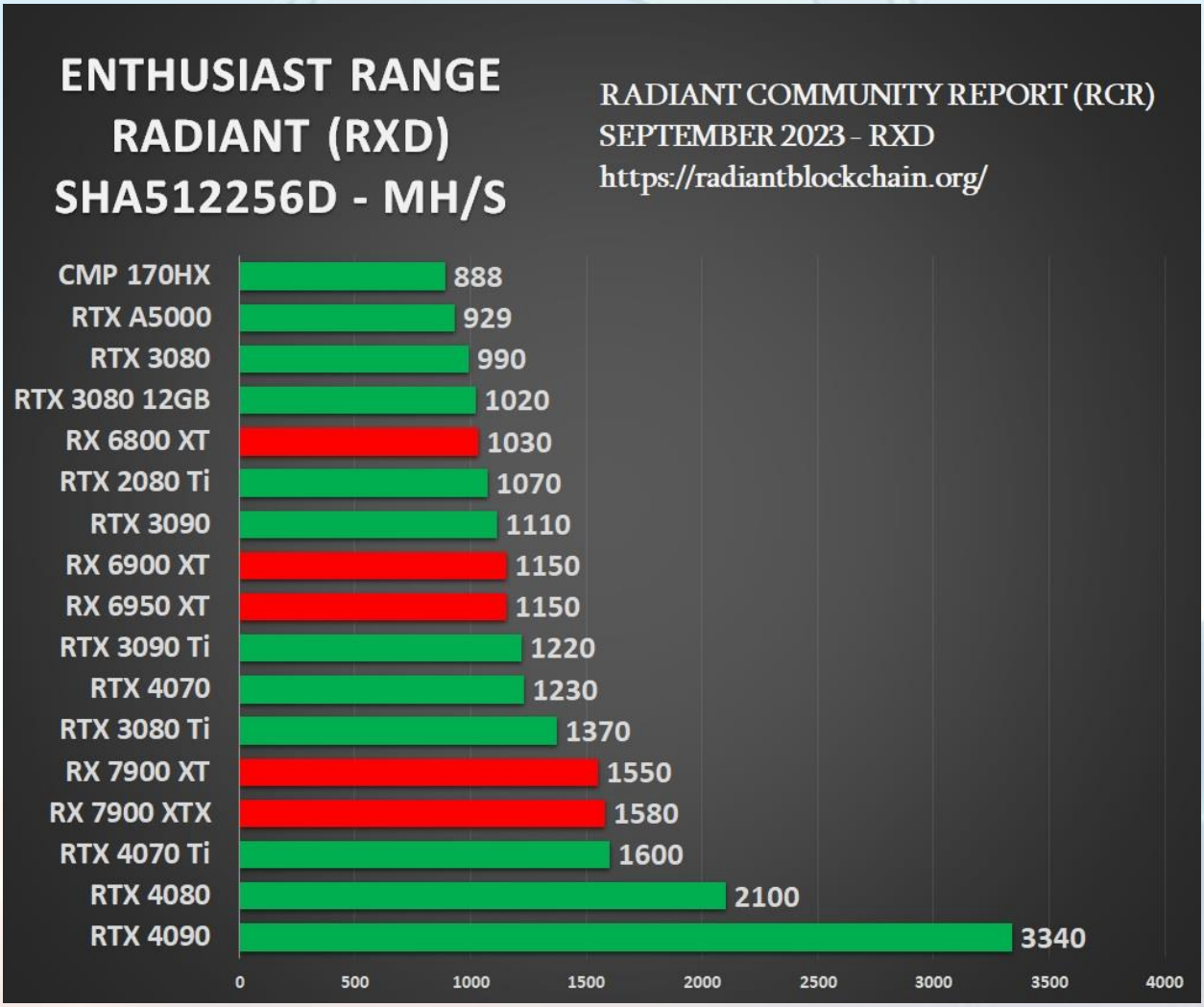
Nvidia, as it's shown, is clearly having leadership in the entry level range. The GTX 1660 models are still showing strength, while the RTX 3060 are competing directly with the AMD RX 6600 XT, boards that can be overclocked significantly and that do not possess the LHR limitation (element that in certain loads might create issues with multiple algorithms). In the top of the chart we still find the RTX 2060, a model that has proven itself during the last years as a noteworthy mining upgrade.

7.2 Mid-Range VideoCards



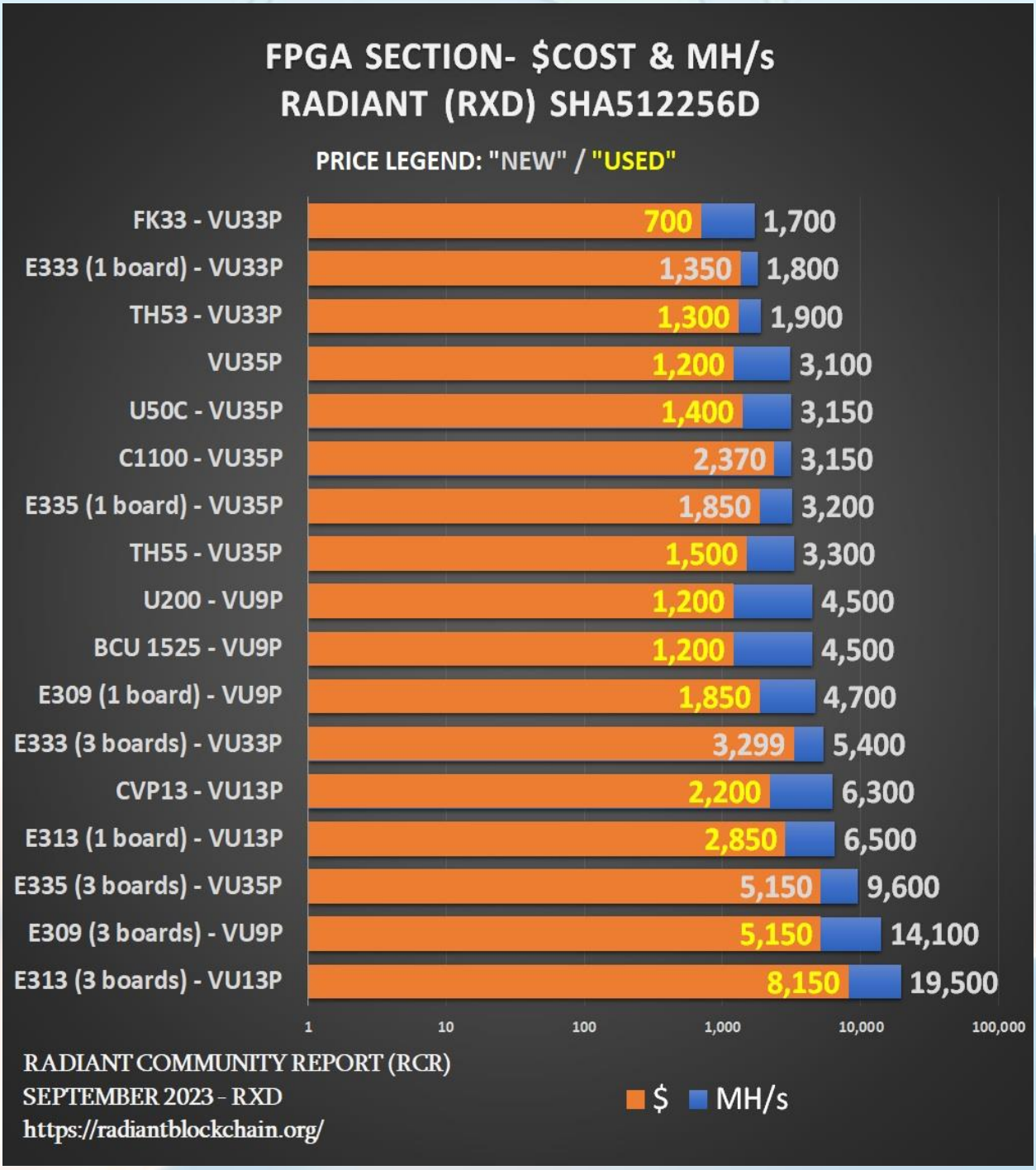
In this section Nvidia is taking the lead, with the RTX 3060 Ti that is trading blows with the AMD RX 6750 XT, that however is significantly inferior to the RX 6800 XT given the shader processor count, current top of the line in the MID range segment.

7.3 Enthusiast-Range VideoCards



In the last segment we can clearly see the Nvidia dominance. AMD cannot compete, and the RTX 4090 has shown an incredible result, destroying literally the competition, and putting to shame also the RTX 4080, possessing much less shader processors. It will be interesting to see the possible new RTX 4080 Ti release, or the super variants of both the RTX 4090 and 4080, because it is probable that a 4070 super will not surpass the stock RTX 4080 computing power. Nvidia has also just announced the stop of the production of the RTX 4080 and 4090 products and this might imply a potential refresh of the product line in the coming 2023 end of year sales interval.

7.4 Field Programmable Gate Arrays - FPGA



Field Programmable Gate Arrays. You heard it right.

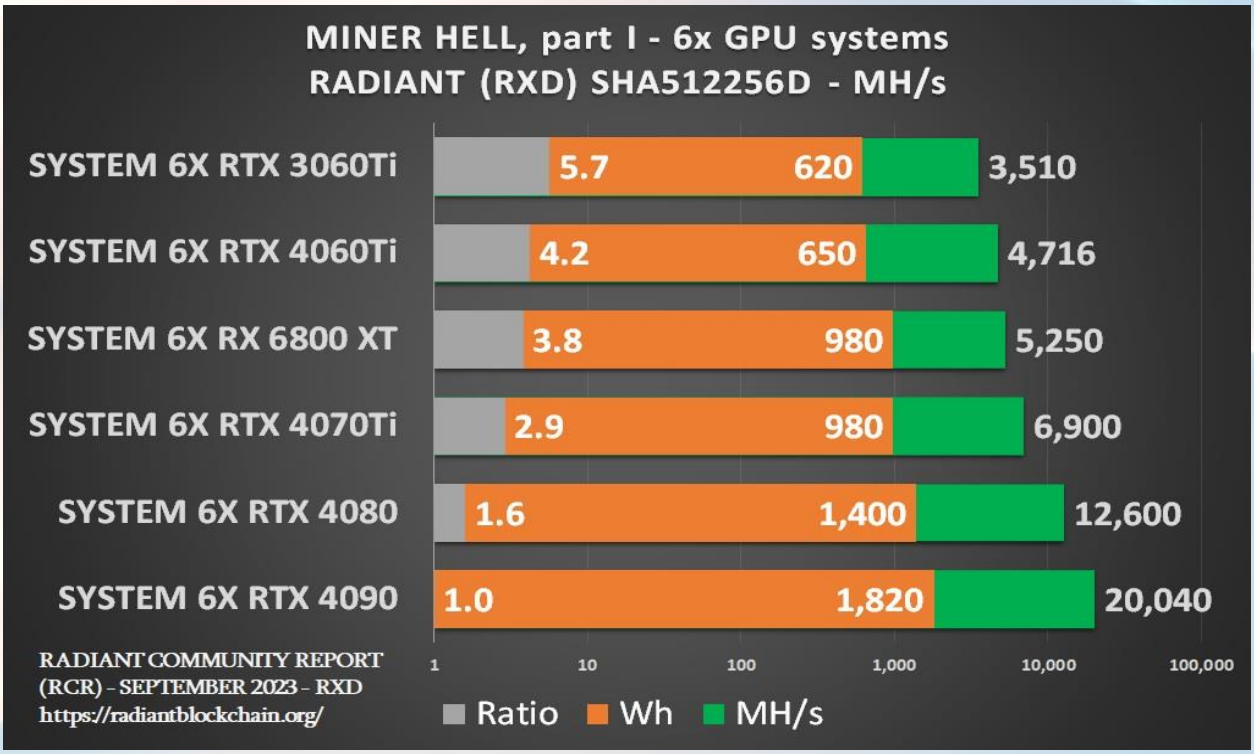
These devices, an hybrid between a GPU and an ASIC, are very efficient and reprogrammable boards that are:

- extremely hard to use and program
- extremely limited in numbers and extremely hard to find in quantity
- extremely efficiency but without any kind of warranty and use-guide
- as, or more, expensive than enthusiast grade VideoCards and hard to manage properly

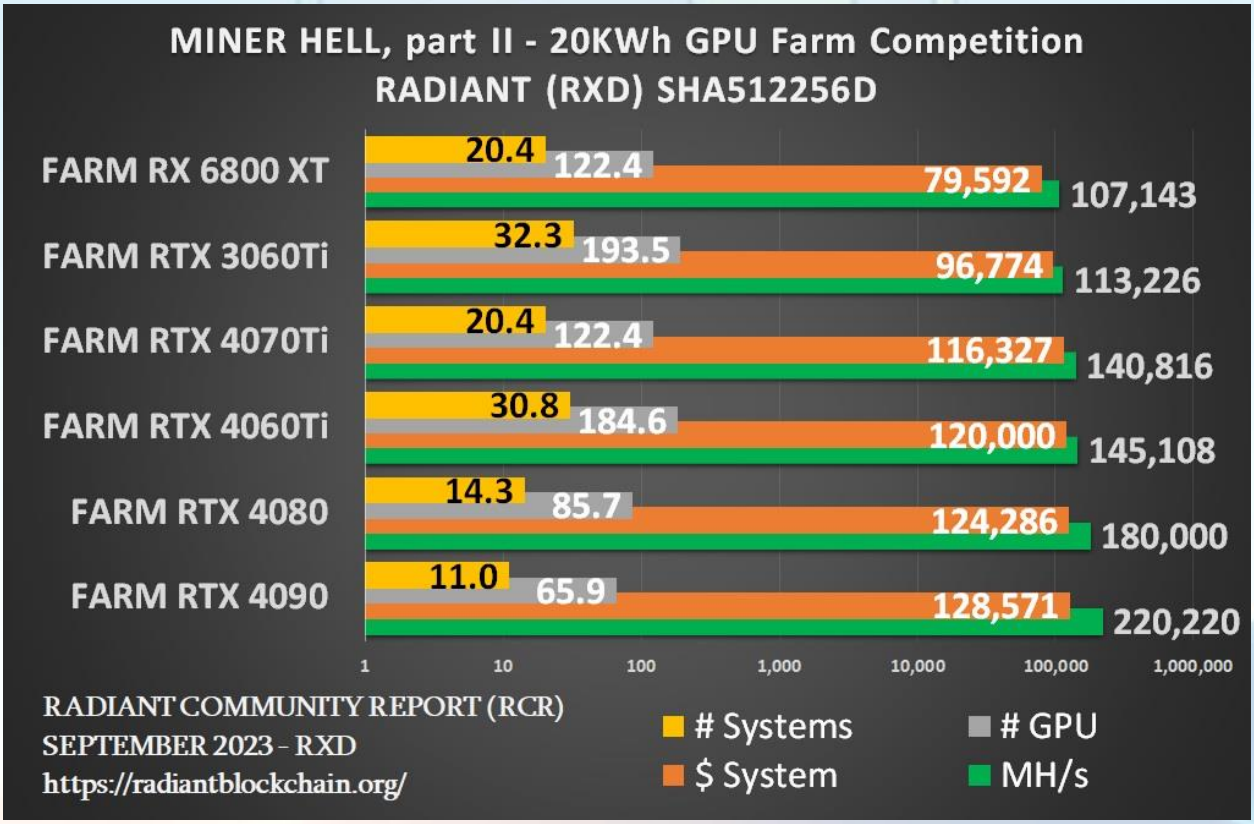
If there are hundreds of Millions of GPU in the markets, there are few tens of thousands FPGA around. One of the best models is the BCU 1525 (XILINX VU9P chip), like the U200 and the E309. Right after we find the models having the VU35P, like the Osprey E300/335 or the TUL u50C/TH55. In the last position the VU33P, on the SQRL FK33. There are few unicorns like the E309, the E313 and the supposed K10, that is based on older small FPKA Kintex models, but in a higher number. The latter is sold by a dubious company that is evidently lying over the state and usage of these boards, shipped to a reviewer with semi burnt pins and extremely dusty chassis. DYOR and mostly, avoid them. These K10 once broken should be trashed, and with them your wallet.

7.5 MINER HELL – Comparisons & Charts

In the coming Chapters, comparisons! For visualization and data standardization purposes, each system has been configured up to 6 Boards. It will be shown the theoretical hashrate, consumption and cost in USD. We start with the chart related to 6x GPU systems:



Clear RTX 4090 Dominance. It follows a simulation of a GPU Farm up to 20KWh of power Consumption:

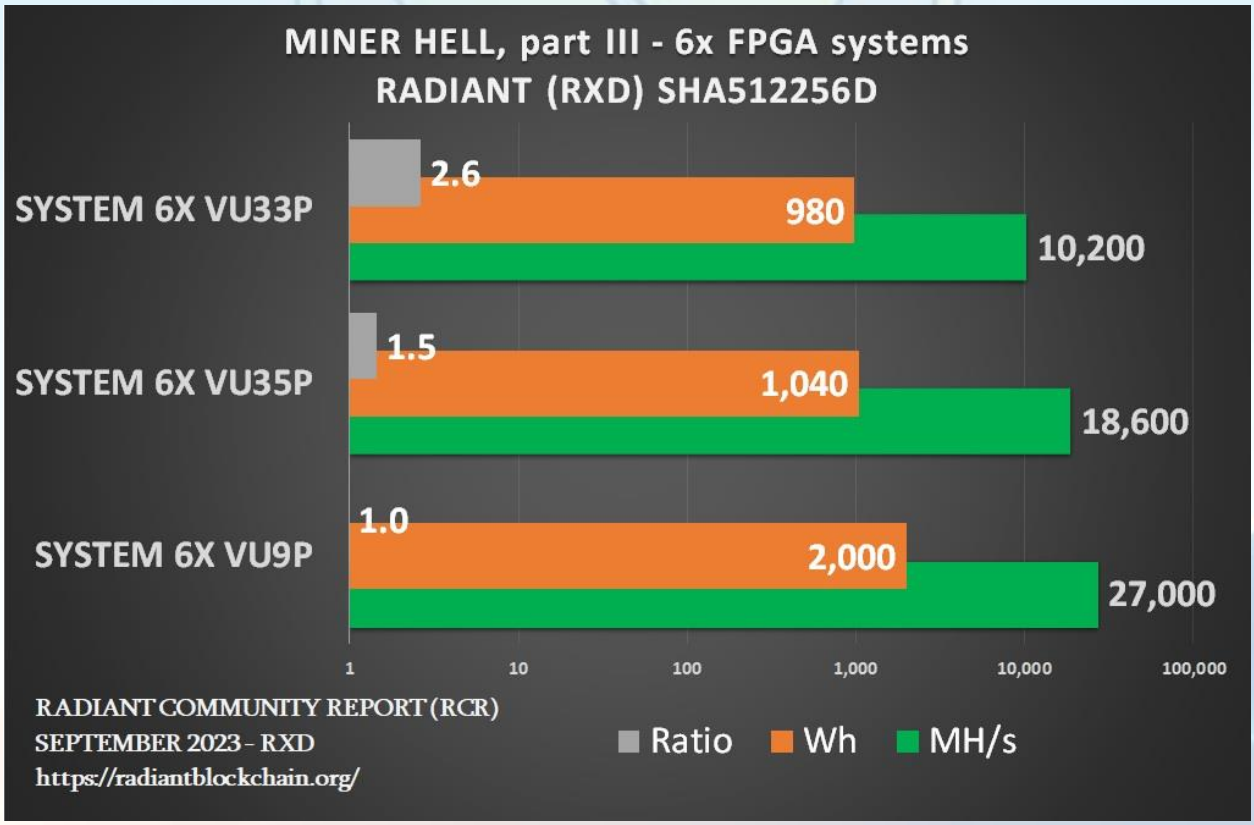


Interestingly there is not much difference making a GPU farm with RTX 4070Ti or RTX 4090, except that the performance are much better for the latter. Do note the number of systems required, 1/3. Clearly for such systems more GPU can be putted in one system, but this shows clearly what a monster the RTX 4090 truly is.

Except in case of very low purchasing prices, the RTX 3060Ti Farm is simply not worth anymore. The main issue is the lackluster performance of the new mid-range RTX 4000 series, because they have been so crippled, by design, from nvidia that the only powerful model is precisely the top of the line, that represents indeed a powerful example of the real capabilities of Nvidia Know-How.

It has to be expected a similar level of performance between the RTX 5000 and the Xilinx VU9P, but this will happen in early 2026 probably, so there is plenty of time, and a full BTC, and RXD, cycle ahead.

We continue with the 6x FPGA system chart. Please do note that FPGA can be linked even up to 20 units, but for control and investment reason the optimal and max suggested configuration is still 6/8 boards / system. Usually: the higher the number of devices = the more the headaches = the bigger the maintenance time and downtime.

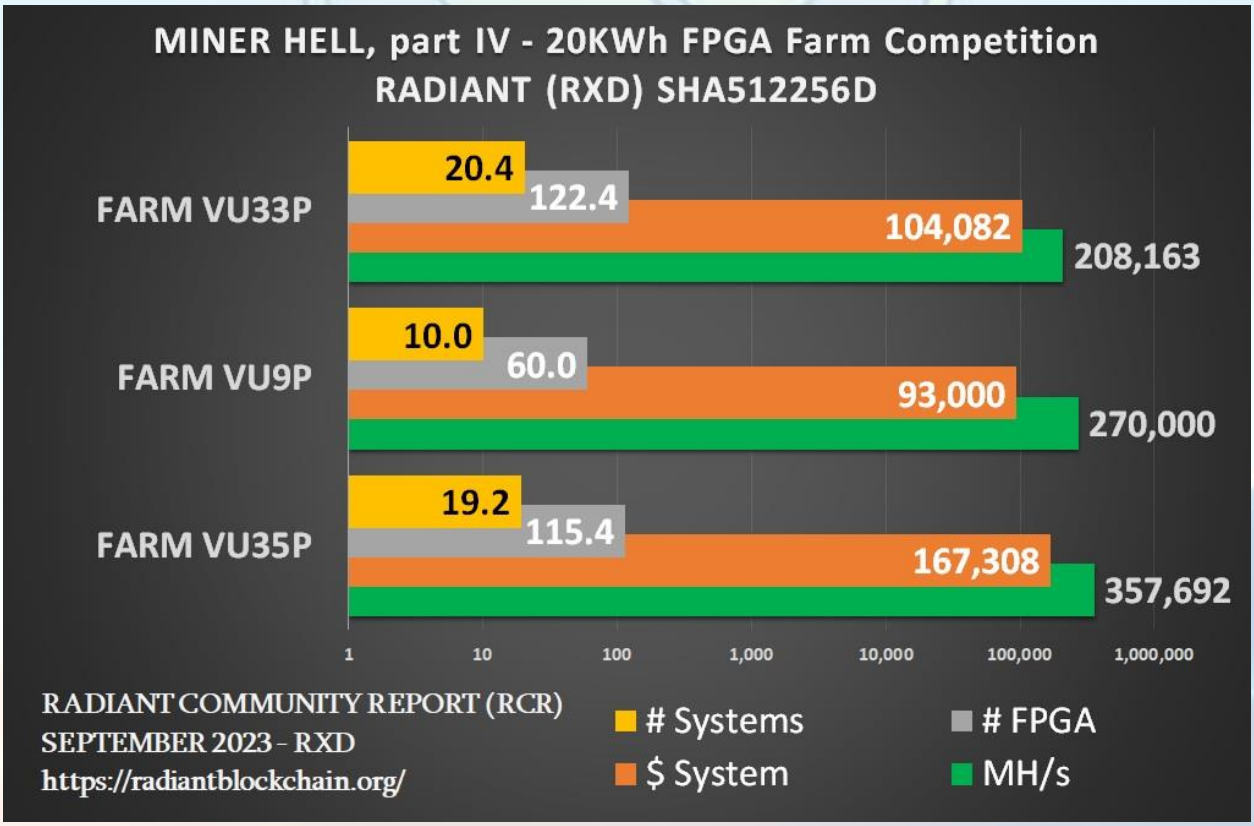


Clear VU9P dominance. Ratio 1 is the 6X VU9P system, and on Radiant a VU33P system has a ratio slightly below 3, very significant.

Please do note that FPGA are systems that possess a USB connector for the transfer of data, and sometimes this can be done also over the PCIE for some models, like the Xilinx BCU1525 and CVP13. Other models like the FK33 or the VU35P, with the SQRL design, do not possess such a feature, and they rely just on the USB connector. These are very complex systems, and they possess an high efficiency if managed properly, but do note that the chip is from 2017, so 8 years have passed. These are chips that were costing 27.000$ in April 2017 possessing a 12 weeks lead time for an order, with MOQ 1. Enthusiast parallel computing, other-than-mining applications, extreme price. These boards are literally wasted with mining activities, but in this document you can find a brief summary of their potential. Do note that the SHA512256D of Radiant is way more restrictive than other core-heavy algos around where such FPGA are operating. This means that by design Radiant is way more

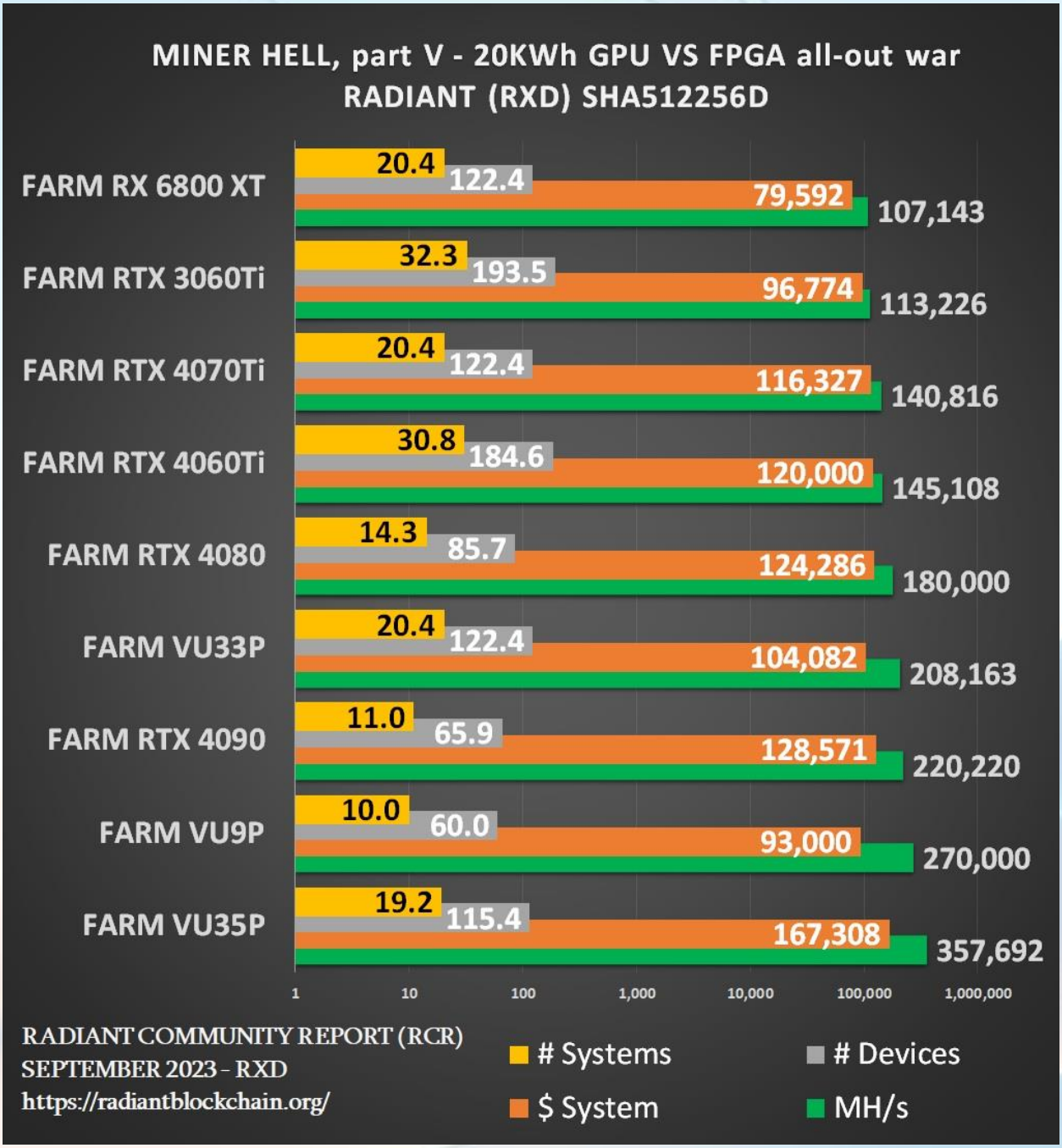
egalitarian, lowering the difference between enterprise boards like FPGA and Enthusiast gaming videocards like the RTX 4090.

But what about a FPGA Farm up to 20KWh? Here it is:



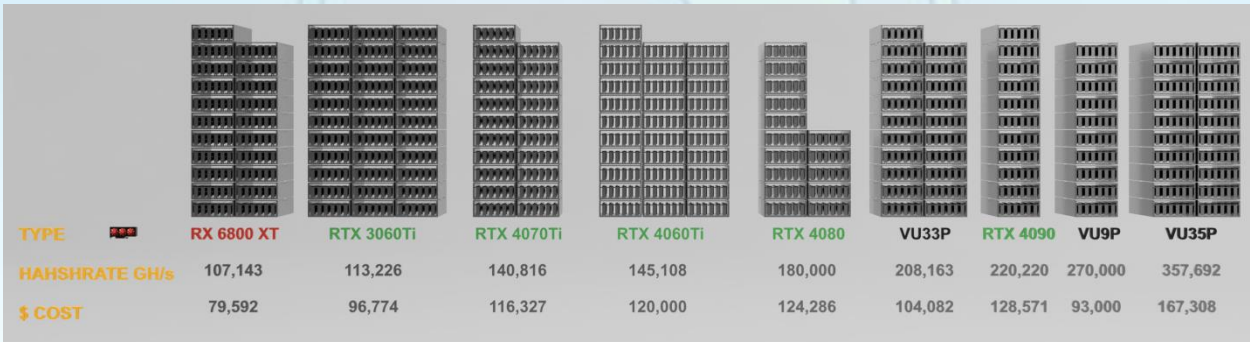
The VU35P chip in this case takes the lead, simply because it is consuming much less, so it has a bigger impact the KW factor, but just look at the prices. All in all, still there is VU9P dominance.

It follows the All-Out-War between FPGA and GPU. Radiant is one of the few core-heavy algos around where we can find the smallest delta between GPU and FPGA, and this is a great feature that leads to a much better decentralization between single board users and industrial farms !



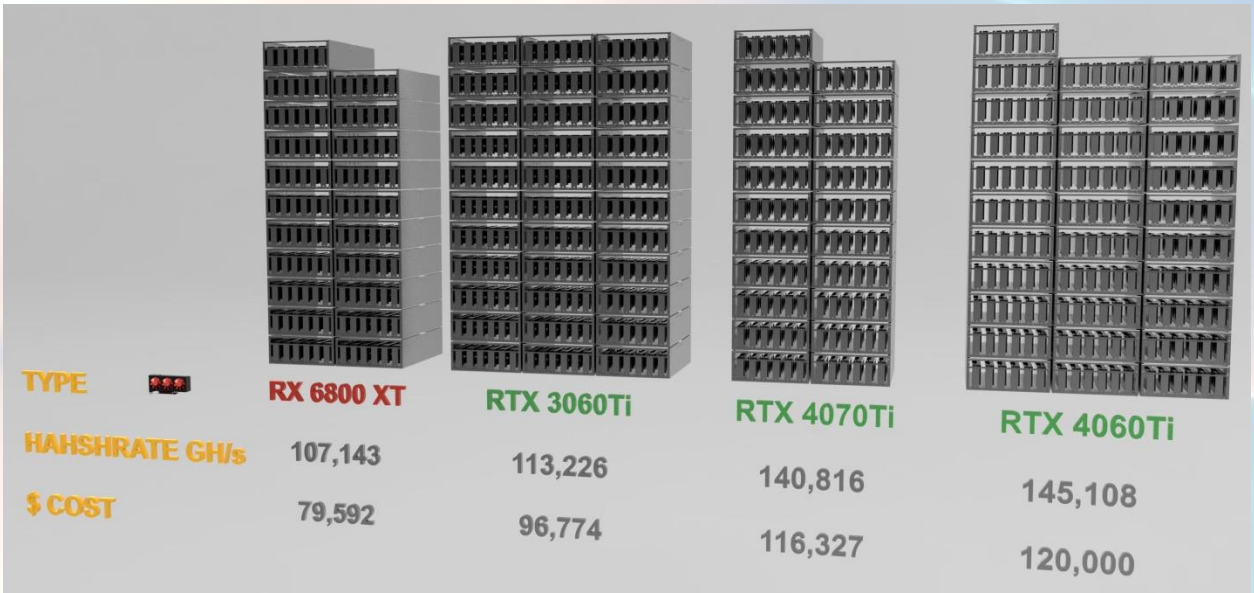
This chart is showing the Maximum hashrate possible with a power consumption of 20KWh. The winner is a farm with VU35P boards. There is however a problem, the balancement between price/performance, and also that the biggest impact of the electric consumption for the BCU boards is cooling, and in case of an immersion cooling setup, this model would pretty much take the lead, while costing way less than the VU35P. In all of this a farm of RTX 4090 is performing 10% better than one with FPGA VU33P (not modified), while costing 5.7% more. Interesting times ahead.

7.6 Mining Hell Showroom



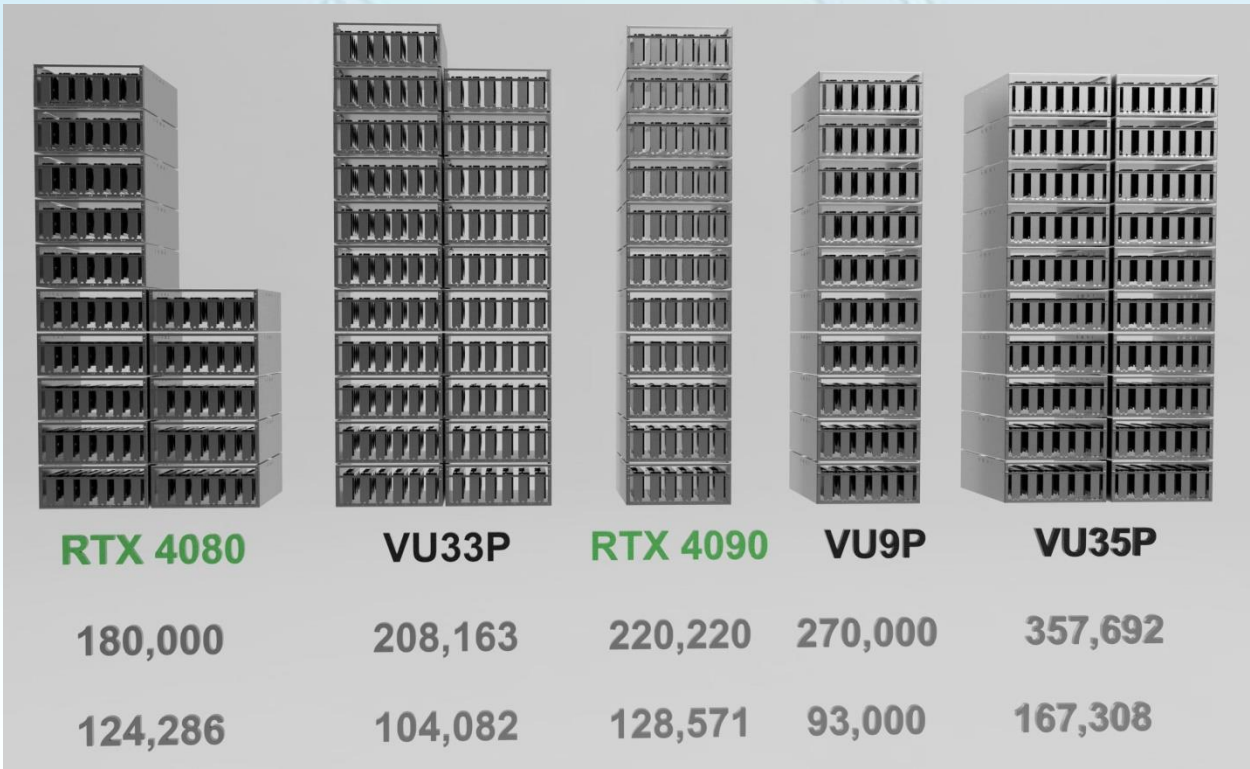
This is a visual representation of what these numbers mean, and how big it could be such a 20KWh farm for such devices, with the associated cost. It is not included clearly all the accessory cost, shelves, electric implants, cooling, personnel, etc. The Data Center business is a complex affair, that has to take in account a lot of costs, and multiple high level elements of complexity. It requires a deep know how, and when its possible to see big hashrate numbers, using this visual representation as a reference, the aim of this document is also allowing people to understand the economies of scale.

Entry Level



The RTX 4070Ti leads the way, and it's very sad to see that AMD completely FAILED with the release of the 7000 Series, to the point that there isn't even a single model worth of being mentioned.

Heavy Lifters



Interesting, finally, to see an algorithm where we can find a Farm with RTX 4090 that surpasses, in performance, FPGA boards like the VU33P. There is however an element: efficiency.

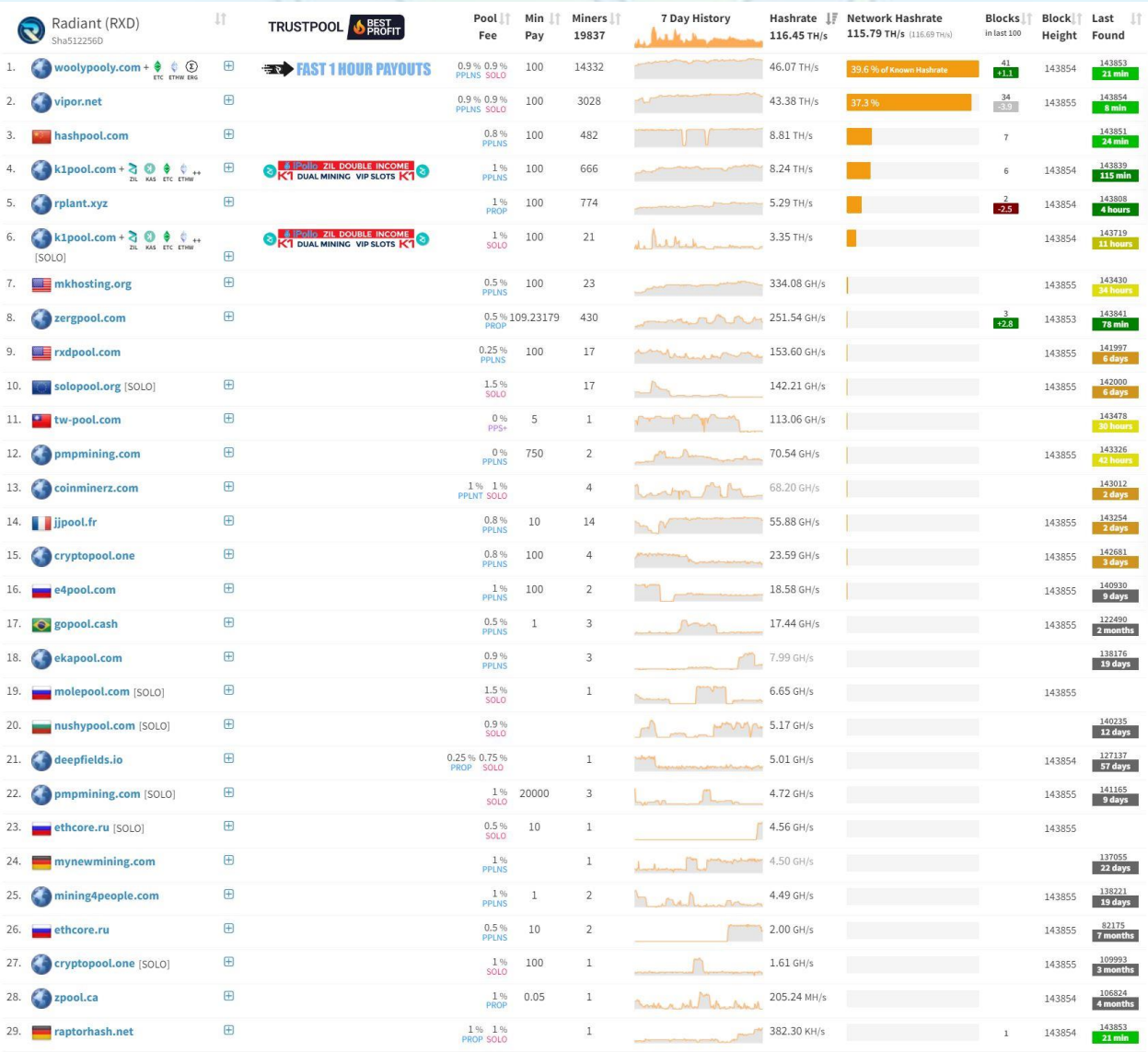
An RTX 4090 is consuming around 300 W to achieve this
An FK33 around 130/140 W (depends by the cooling setup, with blower fans)
Around 110 W if with the TH53 model from TUL, not mentioned here because it's a pure conversion

The FPGA efficiency still is not putted in discussion, for the coming 2 years.

This is the reason why the VU35P farm goes on top of the chart, because it is presenting a similar W rating compared to the VU33P, presenting however double the hashrate. A VU9P with blower fan is hovering around 300W, and this is the reason why it's gone to the second place, but do note the number of servers required, half compared to the VU35P. This can surely be a winning factor, because of the hashrate density and cooling needs, that however are quite complicated for the VU9P setup, because a VU35P is cooled in a much easier way, and surely more efficiently. If we add Osprey in the middle with the E300 series, we understand that playing with cooling gives a lot of edge, and room for improvement.

7.7 Mining Pools Decentralization & Mining Guide

In the crypto mining scene it is important to distribute the hashrate. You can check the list of available pools here: https://miningpoolstats.stream/radiant



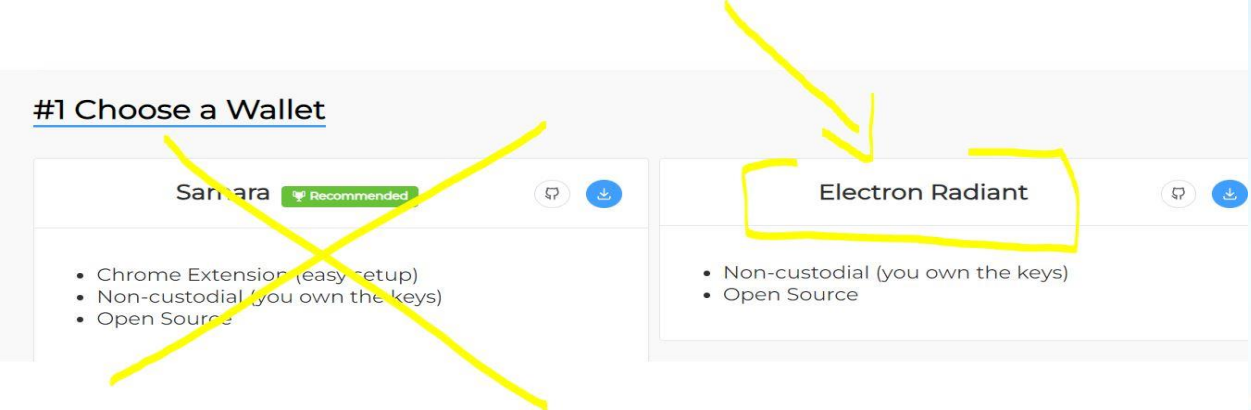
	Radiant (RXD) Sha512256D	TRUSTPOOL BEST PROFIT	Pool Fee	Min Pay	Miners 19837	7 Day History	Hashrate 116.45 TH/s	Network Hashrate 115.79 TH/s (116.69 TH/s)	Blocks in last 100	Block Height	Last Found
1.	woolypooly.com + ETC ETHW ERG	FAST 1 HOUR PAYOUTS	0.9 % 0.9 % PPLNS SOLO	100	14332		46.07 TH/s	39.6 % of Known Hashrate	41 +1.1	143854	143853 21 min
2.	vipor.net		0.9 % 0.9 % PPLNS SOLO	100	3028		43.38 TH/s	37.3 %	34 -3.9	143855	143854 8 min
3.	hashpool.com		0.8 % PPLNS	100	482		8.81 TH/s		7		143851 24 min
4.	k1pool.com + ZIL KAS ETC ETHW	ZIL DOUBLE INCOME K1 DUAL MINING VIP SLOTS K1	1 % PPLNS	100	666		8.24 TH/s		6	143854	143839 115 min
5.	rplant.xyz		1 % PROP	100	774		5.29 TH/s		2 -2.5	143854	143808 4 hours
6.	k1pool.com + ZIL KAS ETC ETHW [SOLO]	ZIL DOUBLE INCOME K1 DUAL MINING VIP SLOTS K1	1 % SOLO	100	21		3.35 TH/s			143854	143719 11 hours
7.	mkhosting.org		0.5 % PPLNS	100	23		334.08 GH/s			143855	143430 34 hours
8.	zergpool.com		0.5 % PROP	109.23179	430		251.54 GH/s		3 +2.8	143853	143841 78 min
9.	rxdpool.com		0.25 % PPLNS	100	17		153.60 GH/s			143855	141997 6 days
10.	solopool.org [SOLO]		1.5 % SOLO		17		142.21 GH/s			143855	142000 6 days
11.	tw-pool.com		0 % PPS+	5	1		113.06 GH/s				143478 30 hours
12.	pmpmining.com		0 % PPLNS	750	2		70.54 GH/s			143855	143326 42 hours
13.	coinminerz.com		1 % 1 % PPLNT SOLO		4		68.20 GH/s				143012 2 days
14.	jjpool.fr		0.8 % PPLNS	10	14		55.88 GH/s			143855	143254 2 days
15.	cryptopool.one		0.8 % PPLNS	100	4		23.59 GH/s			143855	142681 3 days
16.	e4pool.com		1 % PPLNS	100	2		18.58 GH/s			143855	140930 9 days
17.	gopool.cash		0.5 % PPLNS	1	3		17.44 GH/s			143855	122490 2 months
18.	ekapool.com		0.9 % PPLNS		3		7.99 GH/s				138176 19 days
19.	molepool.com [SOLO]		1.5 % SOLO		1		6.65 GH/s			143855	
20.	nushypool.com [SOLO]		0.9 % SOLO				5.17 GH/s				140235 12 days
21.	deepfields.io		0.25 % 0.75 % PROP SOLO		1		5.01 GH/s			143854	127137 57 days
22.	pmpmining.com [SOLO]		1 % SOLO	20000	3		4.72 GH/s			143855	141165 9 days
23.	ethcore.ru [SOLO]		0.5 % SOLO	10	1		4.56 GH/s			143855	
24.	mynewmining.com		1 % PPLNS		1		4.50 GH/s				137055 22 days
25.	mining4people.com		1 % PPLNS	1	2		4.49 GH/s			143855	138221 19 days
26.	ethcore.ru		0.5 % PPLNS	10	2		2.00 GH/s			143855	82175 7 months
27.	cryptopool.one [SOLO]		1 % SOLO	100	1		1.61 GH/s			143855	109993 3 months
28.	zpool.ca		1 % PROP	0.05	1		205.24 MH/s			143854	106824 4 months
29.	raptorhash.net		1 % 1 % PROP SOLO		1		382.30 KH/s		1	143854	143853 21 min

No pool is endorsed but please do note that there have been multiple and even personal reports about Woolypooly shady behaviors. Who is writing had almost 1M KAS missing in payments. It is suggested to AVOID Woolypooly until it will reimburse the missing assets and until it will reply to previous emails. Vipor has proven to be a very reliable pool, and the owners are in the Radiant

Discord at service. By personal experience it is a pool that provided an excellent service as of today. Please do note that Radiant DOES NOT TAKE any commission, nor endorses third parties. DYOR

To mine of vipor, as an example, follow these steps:

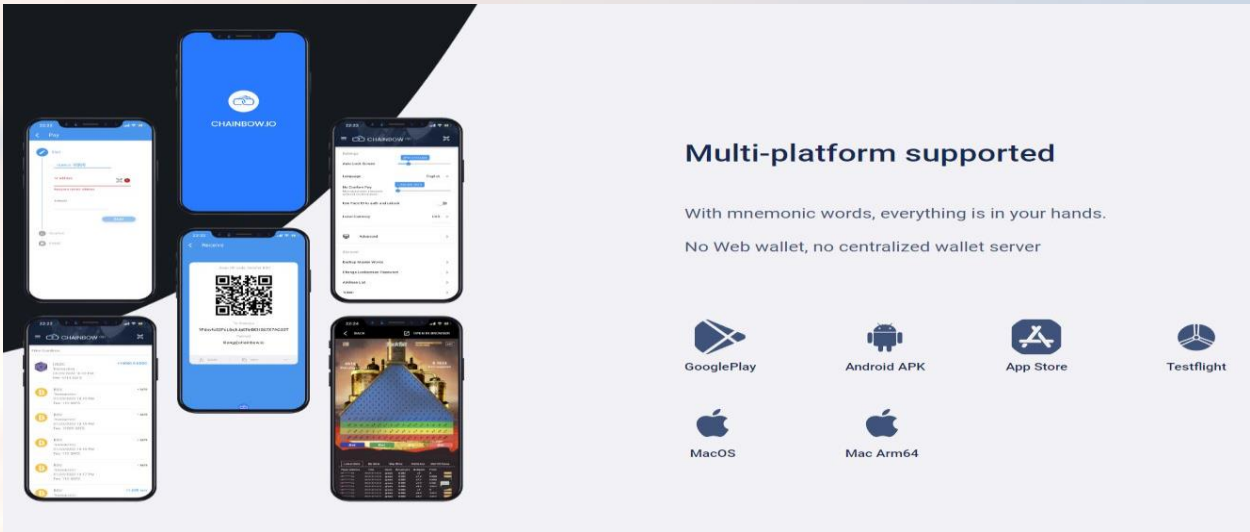
1) Choose the server location & port based on your hashrate
   https://vipor.net/connect/rxd_ppln s_1
2) Choose a wallet, Electron or Chainbow



Samara is currently NOT UPDATED, so avoid it. Do note that you can create a wallet and import your seeds, or restore them, in Chainbow or electron, do not panic if you have issues with Samara. Hopefully it will be updated in the future. Hint: yes, web based. More to come later on.

Link to Electron: https://github.com/RadiantBlockchain/electron-radiant/releases

Link to Chainbow: https://chainbow.io/ (screenshot below)



3) Choose a miner and configure the batch (Ubuntu, windows requires few modifications)
    a. bzminer-v16.0.5-cuda11.6
        i. -a radiant -w 1NfkUEuLZ1CEDZoWvPLaprwMmfZwtzoMn4.$rigName -p stratum+tcp://rxd.vipor.net:5066 stratum+tcp://us2.ethermine.org:4444
    b. gminer-v3.41
        i. --algo radiant --server tr.vipor.net --port 5066 --user 1NfkUEuLZ1CEDZoWvPLaprwMmfZwtzoMn4.$rigName --pass x --api 3333
    c. wildrig-multi-v0.36.10
        i. --algo sha512256d --url stratum+tcp://rxd.vipor.net:5066 --user 1NfkUEuLZ1CEDZoWvPLaprwMmfZwtzoMn4.$rigName --pass x --api-port 3333 --watchdog-script=reboot.sh
    d. Rigel-1.7.2
        i. -a sha512256d -o stratum+tcp://rxd.vipor.net:5066 -u 1NfkUEuLZ1CEDZoWvPLaprwMmfZwtzoMn4 -w $rigName --no-tui --api-bind 127.0.0.1:5000

4) Paste your address in the selected pool, Example:
    a. https://vipor.net/miner/RXD/1NfkUEuLZ1CEDZoWvPLaprwMmfZwtzoMn4/rxd_ppl ns_1

Test address for fast testing purposes, put yours after.

MINING GUIDE 1: https://www.youtube.com/watch?v=_tlTQnSsngI

CHAPTER VIII - DEVELOPMENT CONTESTS (LEGEND)

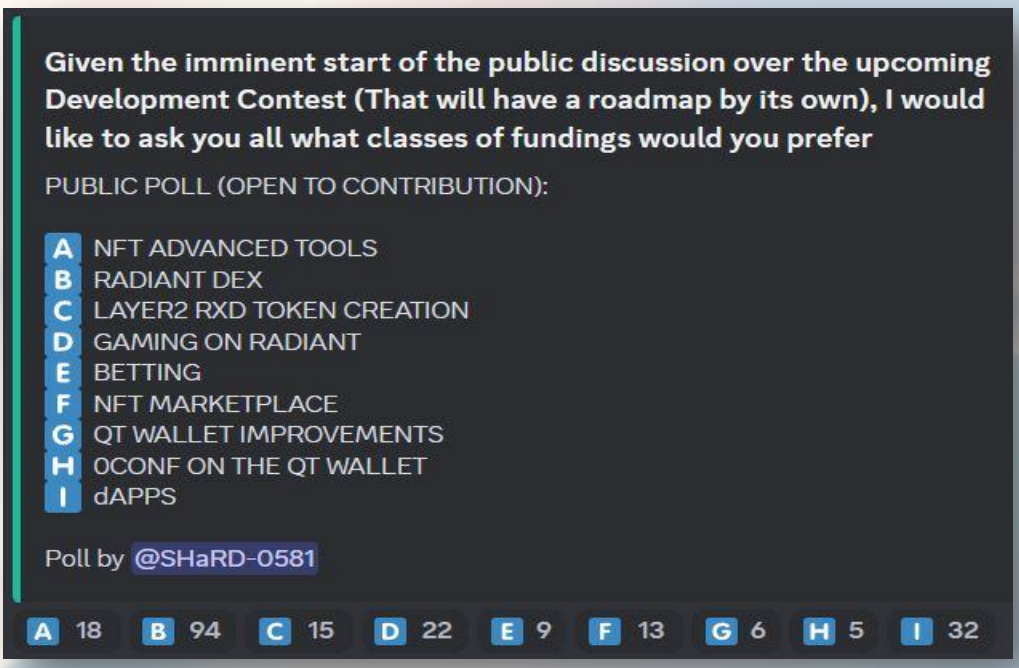
NEW LIFE CHAPTER: RADIANCE / The majority of users often expect a listing so they can buy, profit, sell, and move on to the next project, repeating this pattern and failing in 95% of cases. If you fall into this category, *be aware that Radiant is pursuing a different path*. While Ethereum had software updates, RADIANCE takes it to another level. It represents a growth process, transitioning from infancy to adolescence. This shift encompasses becoming self-sufficient, equipped with its own tools, experience, and funds. It signifies a change in vision, an enhancement of commitment, and the beginning of a journey. As Radiant enters this new phase, we anticipate larger markets, an expanded community reach, broader horizons,

increased confidence, heightened stamina and resilience, greater developer presence and interaction, new developer grants and funded projects, as well as new, free-of-charge participation opportunities. Radiant has just entered its adolescence and in this RCR (Radiant Community Report) it will be provided an in-depth description. RADIANCE is not a one-step process or a buy-the-rumor/sell-the-news event; rather, it is about laying a solid foundation for a skyscraper and raising the ceilings with each completed step, integrating processes and mutually supportive actions.

This marks the beginning of a new chapter, and your support in helping Radiant to shine is immensely appreciated.

The development process is the base of everything that is long lasting, mature and robust. With this approach, given the release of the Radiant Token Standards, the project has effectively entered in its development and execution phase.

There are multiple pathways that could lead to the next expansion phase, and this is in line with the new multi-year roadmap, that will be object of further developments and brainstorming events, publicly. We are here to put the setting stone of the first Radiant Development Contests (RDC)



8.1 Some of the development options

Some of the options on the table, and it's explanation:

A. NFT ADVANCED TOOLS

NFTs are not only Monkey-JPEGS. They are for sure Money-Jpegs, but mostly something that can be time-stamped, certified, notarized, saved by a third party for public audits and also for legal reasons. But it could be also used for private needs, certifications, document sharing or document authentication with public and private realities. It could be a fraction of a document, an extract, a certified and graphical digital signature, an encrypted message or anything else that needs to be registered in a private or public state. In short, it would be needed certainly to build better tools for the management of such elements in the Radiant Network.

B. RADIANT DEX

Certainly by many it is considered to be one of the biggest potentials of Radiant. Imagine a platform where:

- you can list free of charge your L2 Radiant tokens (putting eventually a time-lock over a certain RXD asset as a guarantee)
- you can trade such assets in a decentralized manner, potentially from your encrypted QT, with public keys and API
- you can provide liquidity for Radiant or L2 Radiant Tokens
- you can trade wrapped ordinary Tokens to Radiant
- you can use Radiant and Bridge it to third party realities
- you can eventually use it for NFT trading with future Radiant Games

Just few examples of what could it imply. Is the Radiant community willing to go in this direction?

C. LAYER2 RXD TOKEN CREATION

This feature will be clearly possible given the nature of Radiant and it would be wise to start thinking in this direction too, but there are examples and there is a clear path to follow, based on the Radiant peculiar properties

D. GAMING ON RADIANT

This might be another huge branch and the first to develop, will have the first move advantage. Radiant in its form can bring huge benefits to the cryptomarket, associating itself with established realities, especially with the development of the owned DEX, to be developed or integrated with the previous sketches.

E. BETTING

Another huge market that Radiant could be associated with, given the SC and FT capabilities.

F. NFT MARKETPLACE

A web portal where the sharing, selling and purchasing takes effect.

G. QT WALLET IMPROVEMENTS

With a Radiant DEX potentially trades could happen also from the QT, that could be ideally a base of operations, encrypted and secure, while being associated to the decentralized ecosystem of Radiant.

H. dAPPS

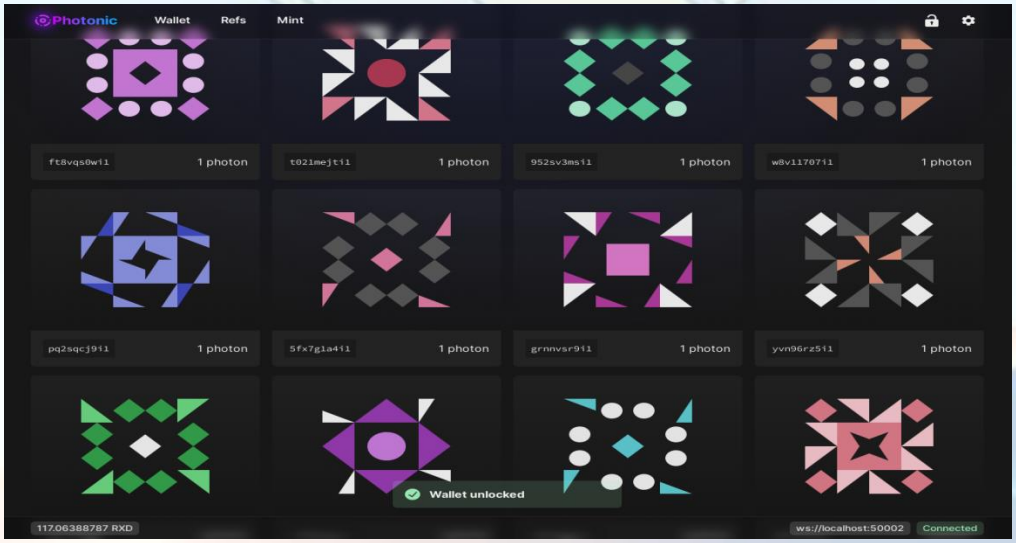
A plethora of options, a sea of possibilities.

After the publication of this document the Radiant Community will proceed toward the Planning and Execution of the First Development Contest, or Hackaton if you wish to call it in this way. Expect further ANNs in this regard.

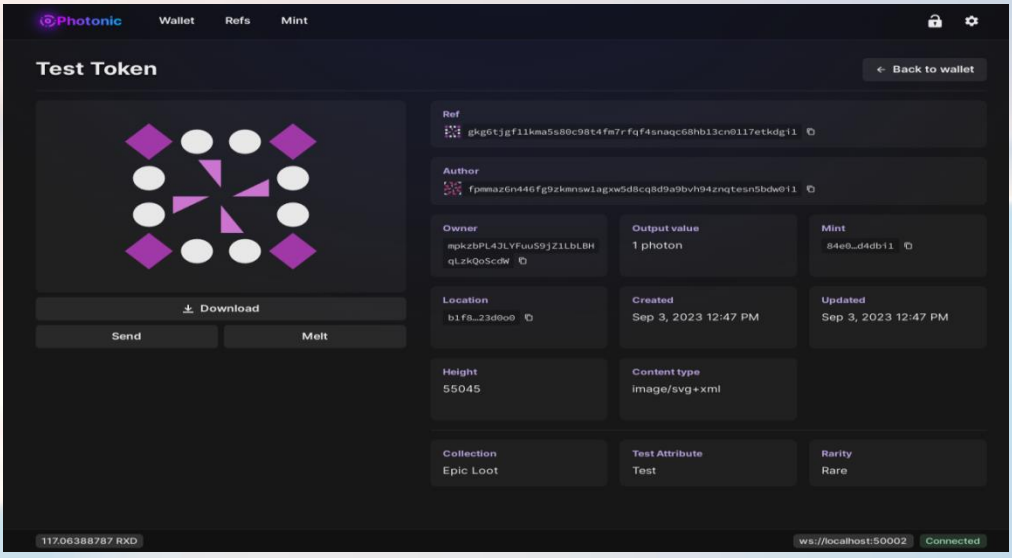
Plenty of stuff to happen, but Radiant needs YOUR CONTRIBUTION. Make it happen, and discuss it freely: https://discord.com/channels/990271820101988362/1024676946556235841

8.2 Photonic Preview

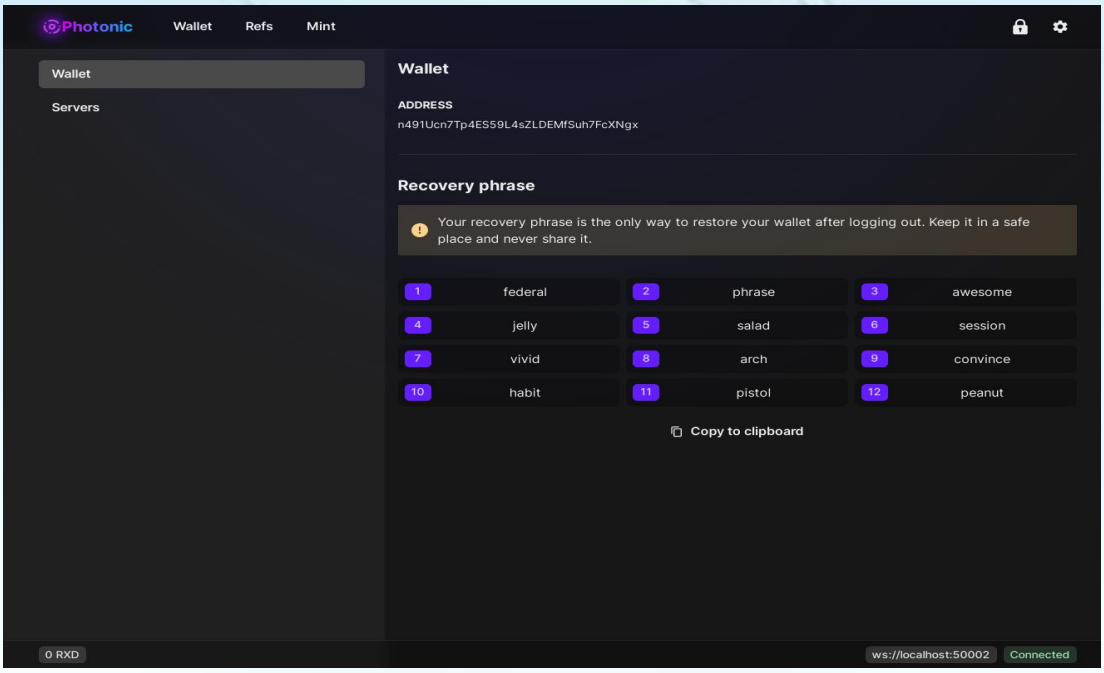
In this paragraph we will find a Brief preview of Photonic, a Radiant web based wallet that will allow storage and NFT minting, aside from other features. ETA a couple of month, but the testnet already works well, as it's shown ! It runs as a web based ElectrumX client so it requires no special backend and can be hosted as static files, even served from a block explorer or IPFS. Later on it might be worth planning ahead its future common developments.



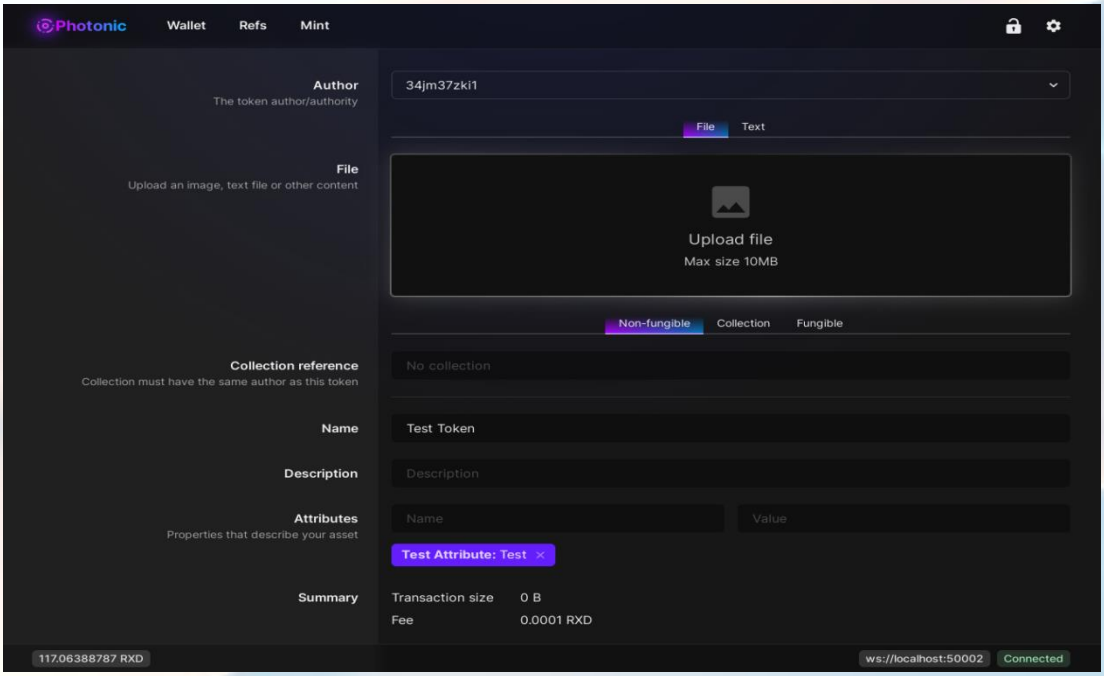
Landing page



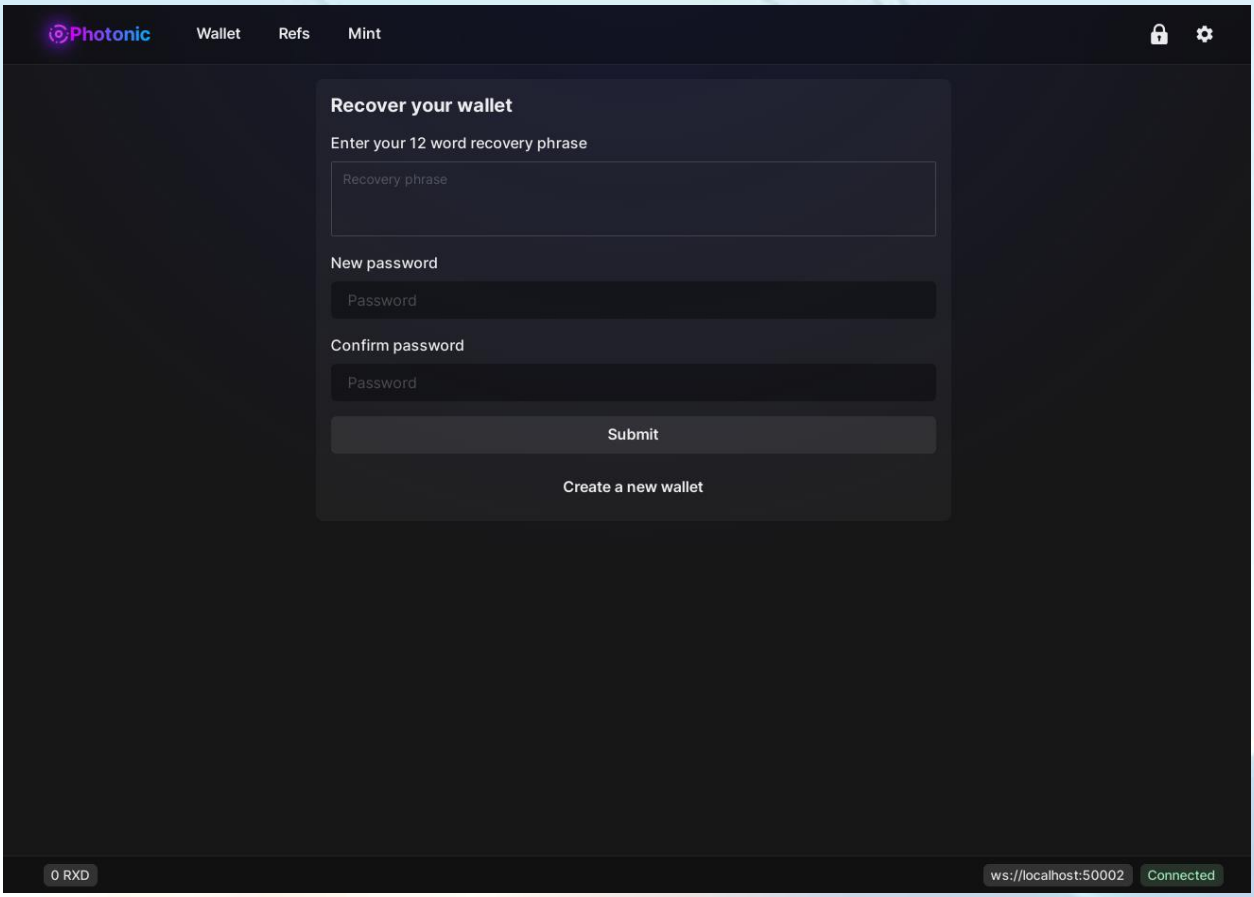
Details, stats (Gaming-wise, noteworthy)



Wallet and Seed



NFT creation page

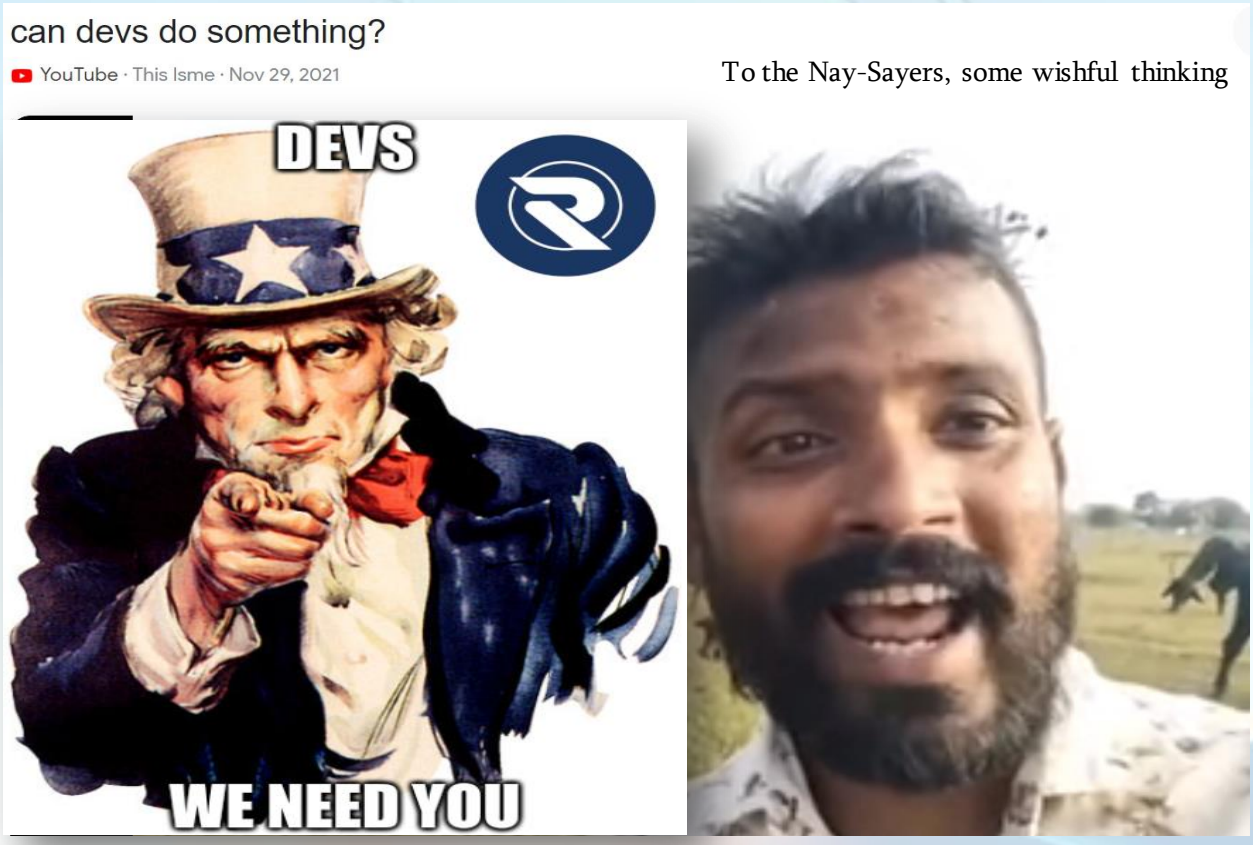


Wallet recovery page, from the seed

In short, a very powerful tool, that will pave the way for future integrations, associations, use cases, potential developments and clearly daily improvements of the Radiant Ecosystem experience.

The start of the Radiant NFT life-chapter !

It is currently under internal testing for bugfixing, core updates and developments, but it is working well and it is expected the release around late November or December 2023. More to come soon.



In short, devs: we need you, but “can devs do something?”

8.3 Radiant FAQ

What is Radiant?

Radiant is based off a fork of the genesis block of BCH. It adds new OPcodes that allow for turing completeness and mathematical induction proofs, which allow for solving the back to genesis problem. This makes Radiant a breakthrough design which provides the performance and parallelism benefits of an unspent transaction output (UTXO) blockchain, but with the contracting ability of account-based blockchains based on the Ethereum Virtual Machine (EVM).

-POW, Smart Contract capable, fairly launched, big block Blockchain.

-Block 0 fork of BCH with added functionality.

-Hashing algorithm is SHA512256d and Difficulty algorithm is ASERT DAA

-Genesis Date: 20/06/2022 02:42:50 GMT+0000
-Genesis Hash: 0000000065d8ed5d8be28d6876b3ffb660ac2a6c0ca59e437e1f7a6f4e003fb4
-Block Time: 300 seconds / 5 minutes
-Halving: 2 years (every 210,000 blocks)
-Subsidy emission: 50000 per block
-Total Coins: 21 Billion, with 8 decimal places for each

What makes Radiant stand out from other blockchain platforms?

-Induction Proofs that solve the back-to-genesis problem. No indexers ever required.
-Account Emulation allows for EVM like applications
-Turing complete scripting language: RadiantScript.
-0conf feature allows for instant transactions.
-Split Node System allows for scalability

How was Radiant fairly launched?

Radiant was bootstrapped as a ready-to-use, 100% PoW blockchain with no investors, no ICO, no premine, and no coin allocations. Before Radiant mining went online, guides on how to mine and even how to rent hash online to mine it without having GPUs were released publicly by Attoshi, Radiant's dev. Everyone was able to obtain RXD from day 1, ensuring a fair distribution.

What is Radiant ethos? Is Radiant an investment?

Radiant aims to be scalable and flexible blockchain technology. It wasn't designed to be invested in/to be expected to generate profits from buying RXD coins. Just like BTC was meant to be a peer-to-peer electronic cash system and not an investment vehicle, Radiant was designed as a peer-to-peer digital programmable asset system. It is not designed for investment purposes, and there's no promise of further development. But you can still "invest" (buy coins from an exchange) in it if you want. NFA. DYOR.

Why is it a fork from BCH?

BTC civil wars divided blockchain enthusiasts into small block-ers and big block-ers. In BTC, small block-ers won and big block-ers forked it to crate BCH. BCH still deviated from what BTC was supposed to be based on what is said on its' whitepaper. Another fork (BSV) was made, aiming to make it as close as possible to the original BTC proposed by Satoshi Nakamoto. This version of Bitcoin still is far from perfect as it has to rely on third party indexers to function properly. On top of that, it is surrounded by drama and heavy baggage

from certain person that affirm they are Satoshi, suing people left and right. That's why BCH was chosen as the building base for Radiant.

What's special about SHA512256d?

It is a hashing algorithm that takes a SHA512 hash and truncates it into SHA256. SHA512 is more secure than SHA256, but that's not the reason it was chosen. In reality SHA256 is already secure enough. But the implementation of such hashing algorithm allows for a secure, battle-tested Blockchain compatible with ASIC machines, without allowing the horde of existing SHA256 or SHA512 ASICs to take over control of Radiant. SHA512256d levels the playing field, allowing for a truly decentralized, massively scalable and extremely cost and power efficient Blockchain. At the moment no ASIC capable of mining RXD exists.

When will ASICs for Radiant mining be available?

Radiant was designed specifically to transition from GPU mining to ASIC mining to allow for a truly scalable and efficient system. The chosen algorithm allows for a straight-forward manufacture process of ASICs. Still, there must exist solid economic incentive for manufacturers first. Developing ASIC machines is a costly task that involves millions of dollars and several months effort. It will only happen if Radiant is able to grow in community, price, and usage demand.

Why is the block time 5 minutes?

5 minute block time is a faster block speed than BTC while retaining stability and security (keeping orphan blocks at bay). With blocks of 256MB size and the possibility of even bigger blocks, it's essential to choose block time carefully. Block time in Radiant, thanks to it's features, isn't a direct limitation for transactions per second or transaction speed.

What are the benefits of Radiant tokenomics?

With tokenization of digital and real word assets in mind, 21 billion coin supply future-proofs Radiant to be able to tokenize all of the worlds assets while leaving available RXD for micro and macro transactions. The 8 decimal places further collaborate to this matter. The block time, block reward and 2 years halving, are an incentive and attractive for miners and investors alike.

What is the Back-To-Genesis problem?

In the context of blockchains, this problem refers to the ability to check that all coins, transactions and digital assets are valid and can be traced back to the genesis block. To do this

after years and billions of transactions without an external/third party indexer had never been possible before. Solving this problem allows Radiant to be a truly decentralized and permission less digital asset system.

What does Turing Complete mean and why have it?

Turing Completeness in the context of a programming language means there's no limit to what kind of computations can be done with it. This enables for extremely flexible smart contracting capabilities similar to those of ETH. Without none of the scalability issues and insane gas costs of ETH, this feature makes Radiant unique. A one of a kind, paradigm shifting blockchain that truly enables all of the use cases of cryptocurrency and blockchain technology.

How does the Split Node System work?

Being a big-blocked blockchain with scalability in mind, nodes in Radiant have the option of running 3 different ways: Mining Nodes which are light-weight, non-resource-intensive, only dedicated to mining; Agent Nodes that contain only the useful information needed to feed to specific dApps/smart contracts, and Archival Nodes, which store the complete Blockchain.

Who created Radiant?

Radiant was bootstrapped by a person/group of people by the pseudonym of "Attoshi". Just like Satoshi created BTC anonymously and let it thrive on its own by its community, Attoshi delivered to the world a functional product, ready to use, without any promises or guarantees.

How to mine Radiant?

Check the Radiant mining guide

How to run a Radiant node?

Check the Radiant node deployment guide

Is there any reward for running a node?

No, Radiant is a 100% POW Blockchain. The only way to get RXD is through mining.

What wallets are compatible with Radiant?

You can access your RXD through:

- Electron: an open source desktop wallet based off the popular BTC wallet
- Chainbow: a mobile wallet compatible with 0conf transactions, developed by the Chainbow team (recognized devs from BSV community)
- Photonic: a multi-platform wallet developed by Radiant community developers, compatible with digital asset minting
- Samara: a web extension wallet (currently outdated and not recommended)

CHAPTER IX - MARKETING INITIATIVES (LEGEND)

It is crucial that every Radiant User contributes actively, because of the community-led nature of the project. Any voice has weight if it's spoken, shared and if it generates ripples in the pool of users sentiments, if it generates a consequence, if it leads to an avalanche.

CHAOS THEORY (from Wikipedia): *"In chaos theory, the butterfly effect is the sensitive dependence on initial conditions in which a small change in one state of a deterministic nonlinear system can result in large differences in a later state. The term is closely associated with the work of mathematician and meteorologist Edward Norton Lorenz. He noted that the butterfly effect is derived from the metaphorical example of the details of a tornado (the exact time of formation, the exact path taken) being influenced by minor perturbations such as a distant butterfly flapping its wings several weeks earlier. He discovered the effect when he observed runs of his weather model with initial condition data that were rounded in a seemingly inconsequential manner. He noted that the weather model would fail to reproduce the results of runs with the unrounded initial condition data. A very small change in initial conditions had created a significantly different outcome. The butterfly effect concept has since been used outside the context of weather science as a broad term for any situation where a small change is supposed to be the cause of larger consequences."*

Generate Noise. Contribute. Cause avalanches. This is what makes Radiant powerful, but just if you truly understand your potential of contribution in a truly decentralized project. Few times in the crypto industry users had power and the possibility to Express. In Radiant, as freedom of expression, they have the possibility to Create.

9.1 Community Campaign

It has been created a Marketing Campaign from the Discord User HippiePyro (hippiepyro). It will be reported here so that everyone can potentially work in this direction, by his own initiative:

https://discord.com/channels/990271820101988362/1029432986749391000/1142842067454791731

As an up and coming project with only community funding, we start with a grassroots marketing campaign. This is simple and effective, each of us posting and commenting about Radiant wherever we can. Our goal is increasing general awareness of Radiant. We, the community, are the most powerful tool we have to make more people aware of Radiant. Word of mouth is the single most impactful means of recruitment. People are more likely to read a post or comment than any ad we can purchase. If OGfruits or Bonjovi can post fud all day, every day, you can post something positive at least once a day. We have at least 5,000 members here in discord alone and if we all post about Radiant a few times a week, it can be far more effective than a paid campaign, as well as cheaper. Anyone who remembers the DeepOnion campaign knows how effective a posting campaign can be. We will take the crypto space by storm. This isn't a bounty campaign, RXD is purely community driven; everyone can and should do this, it takes less than 5 minutes and it's as easy as 1,2,3.

- VOTE- Give 1 vote for RXD at one of the crypto ranking pages.
- POSTS- Make 2 posts or comments about RXD.
- SHARES- Share 3 posts on 3 social media platforms.

General guidelines

- Make interesting posts. Try to spark conversation, educate the readers and leave them wanting more. Provide links to Radiant material when you can.
- Be respectful- Some places have rules against irrelevant posting, advertising, shilling, be sure to check your posts are allowed. Also keep content focused on Radiant. No trashing other projects, no politics, no spitting matches, no hijacking threads. You represent Radiant when you post.
- No spam, no copy paste- We need to keep comments as unique as possible. No spamming or flooding a thread. Seeing the same thing over and over gets old. Be creative when you post. This will be easy in the beginning but harder after a while. Even saying the same thing in a different way is better than straight repetition.
- Many sites require an account via email to complete these activities. If you don't want to be flooded with spam notifications on your personal accounts, consider creating an

email just for using these accounts.

Vote List

Below is a list of various crypto ranking websites that rank based on votes, likes, follows,ect. Some only let you vote once, some are every day. There's not many of them, show your support for Radiant and vote at least once a day! Any votes we have in discord need to be shared for maximum community outreach.

- https://minerstat.com/coin/RXD
  - Use the how do you feel tab to vote. This can be repeated daily.
- https://www.coingecko.com/rxd
  - Add to watchlist, vote with the how do you feel tab daily!
- https://www.livecoinwatch.com/price/Radiant-RXD
  - Vote and add to favorites!
- https://www.cryptocompare.com/coins/rxd/overview
  - Follow RXD, add to favorites and portfolio. Leave a comment too!
- https://coinmarketcap.com/currencies/radiant
  - Add to your watchlist, set your status as bullish. This is one time only.
- https://www.coincarp.com/currencies/radiant/
  - Add to favorites, one time only.
- https://coinpaprika.com/coin/rxd-radiant/
  - Add to favorites, one time only.
- https://coincodex.com/crypto/radiant Add to watchlist, one time only.

Comment List

The next list is of popular websites, forums, and big social media accounts. Leave a comment about RXD, this comment can be anything! How RXD relates to the site, article, video, ect. What problem RXD is solving, how a similar situation would have gone on this blockchain, what about the capabilities of Radiant excites you. Or just share one of our amazing Radiant memes from the discord. I aim to make one post about tech, one about my excitement for RXD and one for fun with memes.

- Use key words from the SEO list, when it makes sense. This makes your comment more likely to be picked by the algorithms and seen by more people.
- Steer away from mooning and price discussion comments as these are generally ignored and of low quality.
- Add links to RXD media when you can, this gives the reader a way to follow up with more RXD info.
- Use as many tags as you can when posting.
- Post in different places every day to maximize Radiant exposure and avoid repetition.
- More comment ideas can be found near the end of this guide.

After you post, share it in the discord channel

#content-creator-media. And then pick someone else's comment from that channel about and share that on your accounts. This increases engagement within the algorithms and makes them suggest RXD content more often to new users automatically.

Popular crypto websites

https://coinmarketcap.com/currencies/radiant/

Our CMC page

https://bitcointalk.org/index.php?topic=5436028.msg61620604#msg61620604

Our ANN thread on BCT

https://www.cryptocompare.com/coins/rxd/overview

Comments at the bottom of the coin page

https://www.altcoinstalks.com/index.php

An altcoin forum like BCT https://cryptolinks.com/ many links to everything crypto, find a venue and post there

Large Forum Style Social Medias

Each forum has different rules for posting

Reddit

- https://www.reddit.com/r/binance/
- https://www.reddit.com/r/cryptomarket/
- https://www.reddit.com/r/altcoin/
- https://www.reddit.com/r/defi/
- https://www.reddit.com/r/CryptoCurrencies/
- https://www.reddit.com/r/CryptoMarkets/
- https://www.reddit.com/r/BitcoinBeginners/
-

Discord

- https://discord.gg/cryptoknight
- https://discord.gg/money
- https://discord.gg/cryptohub
- https://discord.gg/cryptocurrencyofficial
- https://discord.gg/nftw
- https://discord.com/invite/zJkxzrq
- https://discord.gg/cryptocurrency
- https://discord.gg/the-crypto-nation-807769695293407242
- https://discord.gg/ZnkCMzzw9b

Telegram

- https://t.me/cryptowendyochat
- https://t.me/kryptoed

- https://t.me/DeFimillion
- https://t.me/minternetworksignals
- https://t.me/btcchamp
- https://t.me/DeCenterOrg

Large Influencer Social Media Accounts

Posts made here aren't necessarily made to attract the account owner. We are more interested in getting their viewers and followers interested in Radiant.

- https://twitter.com/CoinDesk
- https://twitter.com/nansen_ai
- https://twitter.com/girlgone_crypto
- https://twitter.com/MessariCrypto
- https://twitter.com/danheld
- https://twitter.com/VitalikButerin
- https://twitter.com/nickszabo4
- https://twitter.com/cz_binance
- https://twitter.com/Coinboundio
- https://twitter.com/saylor
- https://twitter.com/IvanOnTech
- https://twitter.com/layahheilpern
- https://twitter.com/bitcoinlobo
- https://twitter.com/Itzjoshuajake
- https://twitter.com/TuurDemeester
- https://twitter.com/blockchainboyy/
- https://twitter.com/kennethbosak
- https://twitter.com/thecryptokang
- https://twitter.com/ErikVoorhees
- https://twitter.com/tyler
- https://twitter.com/TheCryptoDog
- https://twitter.com/ToneVays
- https://twitter.com/CryptoWendyO
- https://twitter.com/bitboy_crypto

TikTok

- https://www.tiktok.com/@girlgone_crypto
- https://www.tiktok.com/@cryptokang.reborn
- https://www.tiktok.com/@cryptowendyo
- https://www.tiktok.com/@itzjoshuajake
- https://www.tiktok.com/@thewolfofbitcoins
- https://www.tiktok.com/@theblockchainboy
- https://www.tiktok.com/@cryptoed

Instagram

- https://www.instagram.com/cryptokang/
- https://www.instagram.com/Itzjoshuajake
- https://www.instagram.com/thewolfofbitcoins/
- https://www.instagram.com/cryptowendyo
- https://www.instagram.com/theblockchainboy/

Video and Live Stream Social Media

fairly relaxed rules on comments. Did you know a text-to-talk tip message will broadcast to an entire live stream audience, not just the streamer!

YouTube channels

- https://www.youtube.com/tonevays
- https://www.youtube.com/CryptoWendyO
- https://www.youtube.com/c/CryptoKnightio
- https://www.youtube.com/c/ivanontech
- https://www.youtube.com/TheCryptoLark
- https://www.youtube.com/AltcoinDaily
- https://www.youtube.com/Diaryofamademan
- https://www.youtube.com/DataDash
- https://www.youtube.com/Boxmining
- https://www.youtube.com/CryptoLove
- https://www.youtube.com/@CryptoBanterGroup
- https://www.youtube.com/@TheEconomist
- https://www.youtube.com/CryptoBanterGroup
- https://www.youtube.com/@99Bitcoins
- https://www.youtube.com/@VoskCoin
- https://www.youtube.com/@MiningChamber
- https://www.youtube.com/@RabidMining
- https://www.youtube.com/@RedPandaMining
- https://www.youtube.com/@TheHobbyistMiner
- https://www.youtube.com/@SebsFinTechChannel
- https://www.youtube.com/@ChumpChangeXD

Twitch

- https://www.twitch.tv/thecryptokang
- https://www.twitch.tv/hasheur
- https://www.twitch.tv/imdbraz
- https://www.twitch.tv/cryptokking
- https://www.twitch.tv/tr4d3r10
- https://www.twitch.tv/tradersamwise

SEO List

Search engine optimization is a method of making comments posts and articles more attractive to the algorithms and more engaging for the audience. Landing pages- Use these links in comments or posts whenever possible. These are where we need to direct users, where we need to increase click-thru. The more times someone clicks these links while searching, the more often google will suggest them to new users.

- https://radiant4people.com/
- https://radiantblockchain.org/
- https://radiantexplorer.com/

Personalization- People first content is far more impactful. Posts should be created with the reader in mind, what are they going to want to see, how will they be impacted by the news. Most popular keywords and phrases searched in google

- best cryptocurrency
- best cryptocurrency to invest in 2023
- best cryptocurrency app
- best cryptocurrency exchange
- cryptocurrency to buy
- buy cryptocurrency
- mining cryptocurrency
- cryptocurrency mining
- what is mining cryptocurrency
- best multi cryptocurrency wallet
- cryptocurrency wallet

Most used crypto hashtags on Twitter and Instagram

#rxd #Radiant #cryptocurrency #bitcoin #crypto #blockchain #ethereum #btc #forex #money #trading #investment #bitcoinmining #cryptotrading #cryptonews #investing #bitcoins #business #bitcoinnews #cryptocurrencies #forextrader #invest #entrepreneur #eth #bitcointrading #trader #investor #binaryoptions #binance #forextrading #bitcoincash #finance#litecoin #nft #coinbase #stockmarket #stocks #forexsignals #success #binary #dogecoin #blockchaintechnology #wealth #cryptoworld #bitcoinprice #xrp #cryptoinvestor #motivation #forexlifestyle #hodl #altcoin #usa #trade #ripple #mining #cryptomining #financialfreedom #cryptocurrencynews #daytrader #wallstreet #altcoins #millionaire

Comment and Content Ideas

BCT signatures- Anyone with Html coding experience can whip up some signatures for use on forums like Bitcointalk, Altcoinstalk. Link to bct limits on design https://bitcointalk.org/index.php?topic=484259.msg5393088#msg5393088

Articles

Any articles about RXD will greatly help newbies and can potentially be published on news sites later on. Graphics- Avatars for discord, twitter, ect. New memes and remakes of old ones go over great. Info graphics for various aspects of the ecosystem are great as well. If you don't have the ability to make it but have the concept, ask one of our talented creators in discord to craft it for you. Many memes are already on Tenor. A large repository of RXD memes can be found in our discord in the channel #meme-stickers-emojis and this google drive from one of our artists.

- https://discord.gg/9QFH5tst
- https://drive.google.com/drive/folders/1tInIGqSMawFdfsOiT4fBoPFyb5AViHMk?usp=share_link

Translations

Anyone who can translate to a non-english language can help spread the word in their native tongue by translating docs and guides to make access easier for all those around the world. Transcripts of interviews- Some folks understand better by reading then hearing, so transcripts of various interviews given will provide a media for them. It also makes a low volume, quick reference guide to lots of key information.

Make Informational Videos

- How to guides
- How to mine with each miner, hive, Mac, and windows
- How to dual/triple mine- CFX,KAS,ERG,ALPH,DYNEX,IRON,ZIL
- How to swap mine to rxd via zergpool
- How to set up each wallet
- How to rent hashrate to mine RXD

Official links to share, verify in discord channel #links

- Website: https://radiantblockchain.org/
- Web Community: https://radiant4people.com/
- GitHub: https://github.com/RadiantBlockchain
- GitHub Community: https://github.com/RadiantBlockchain-Community
- Whitepaper: https://radiantblockchain.org/radiant.pdf

System Design: https://radiant4people.com/tech/radiant-system-design

Community Campaign by HippiePyro

9.2 Translation Efforts of the RCR

Given the complexity of the single user translation, this should be a community effort.

Top languages by population

***Ethnologue* (2023)**

The following languages are listed as having at least 50 million first-language speakers in the 26th edition edition of *Ethnologue* published in 2023.[7] Entries *Ethnologue* identifies as macrolanguages (such as Arabic, Lahnda, Persian, Malay, Pashto, and Chinese, encompassing all their respective varieties) are not included in this section.

Languages with at least 50 million first-language speakers[7]

	Language	Native speakers (millions)	Language family	Branch
1	Mandarin Chinese (incl. Standard Chinese, but excl. other varieties)	939	Sino-Tibetan	Sinitic
2	Spanish	485	Indo-European	Romance
3	English	380	Indo-European	Germanic
4	Hindi (excl. Urdu, and other languages)	345	Indo-European	Indo-Aryan
5	Portuguese	236	Indo-European	Romance
6	Bengali	234	Indo-European	Indo-Aryan
7	Russian	147	Indo-European	Balto-Slavic
8	Japanese	123	Japonic	Japanese
9	Yue Chinese (incl. Cantonese)	86.1	Sino-Tibetan	Sinitic
10	Vietnamese	85.0	Austroasiatic	Vietic
11	Turkish	84.0	Turkic	Oghuz
12	Wu Chinese (incl. Shanghainese)	83.4	Sino-Tibetan	Sinitic

It is crucial that every community works in this direction first, not just for potentially very expensive listings, that benefit few at the expenses of everyone else.

More to come in Discord, please join and start contributing

9.3 4 step priorities, short term

It is proposed to proceed with a 4 – STEP process.

1. TRANSLATION OF THE RCR IN THE MAIN LANGUAGES
2. MARKETING EFFORT FOR THE DISTRIBUTION OF THE LATTER
3. RADIANT DEVELOPMENT CONTEST (HACKATON) PUBLIC VOTING
4. HACKATON FUNDING, VOTING AND STARTUP OF NEW RADIANT VENTURES

These initiatives will bring BIG benefits to Radiant that given the completion of the imminent completion of the RXD Token Standards is in a mature state; these elements will allow new development realities to make a significant contribution, while being funded.

We need to help each-others in this regard, bringing forth our common goals and vision, for the Radiant Greater Good. This will allow Us to create a serious new ecosystem, that will allow everyone the possibility to enter in the blockchain field in the best way, in a open community that is accepting everyone, without any social distinction.

In RADIANT

"WE-ARE-ALL-EQUAL"

"WE-ARE-ALL-RADIANT"

CHAPTER X – TOKENOMICS & END NOTES (LEGEND)

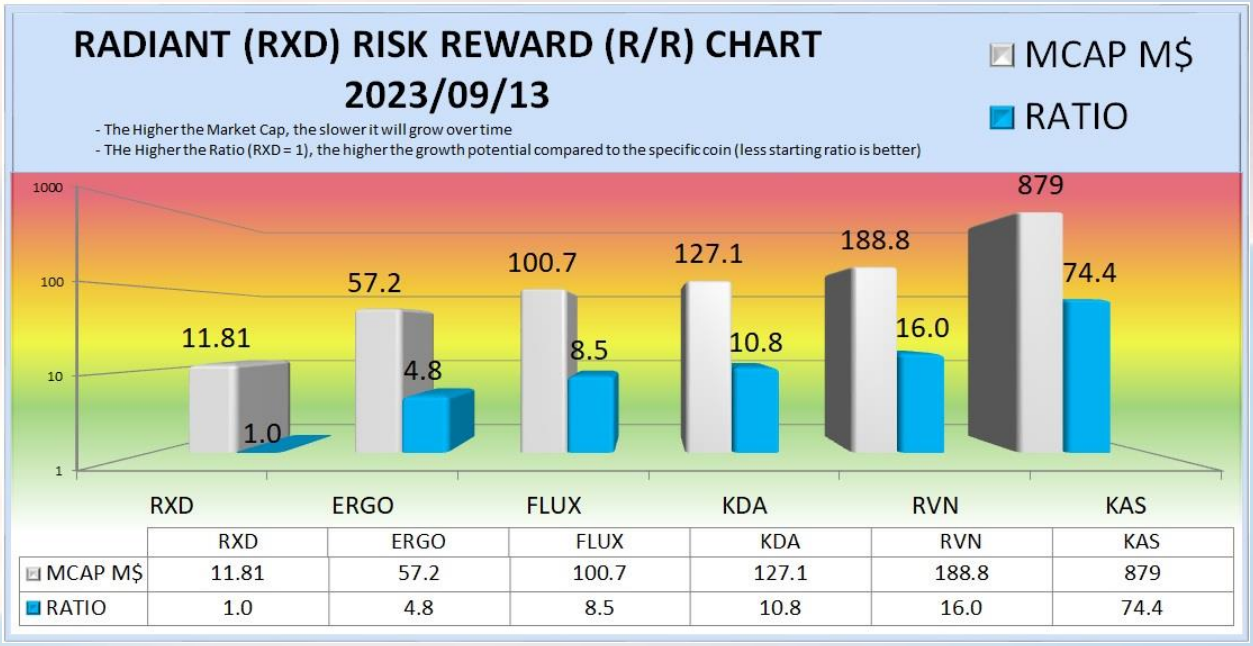
10.1 Brief Tokenomics

Each project has its own characteristics, so technically speaking we could compare gross numbers, without taking into account non-numerical elements, that alone make any comparison hardly realistic, even if potentially apparent.

As an example cannot be compared project having a different age, community dimension, different halving timings, different coin distribution over time, different coin distribution mechanics and clearly the presence of deflationary or inflationary elements, that can alter in an extreme way potential patterns.

Said this, as many requested and as of today, we present you few tokenomics charts that can allow users to understand the position in the crypto-sphere of Radiant, where it's headed and where it might arrive over the coming years.

In the following chart the Risk Reward factor of Radiant, compared to other successful projects, in similar categories, possessing however an extremely different age: ERGO, FLUX, KDA, RVN and KAS. Time is the key, and in this chart Radiant is the youngest.



	RXD	ERGO	FLUX	KDA	RVN	KAS
MCAP M$	11.81	57.2	100.7	127.1	188.8	879
RATIO	1.0	4.8	8.5	10.8	16.0	74.4

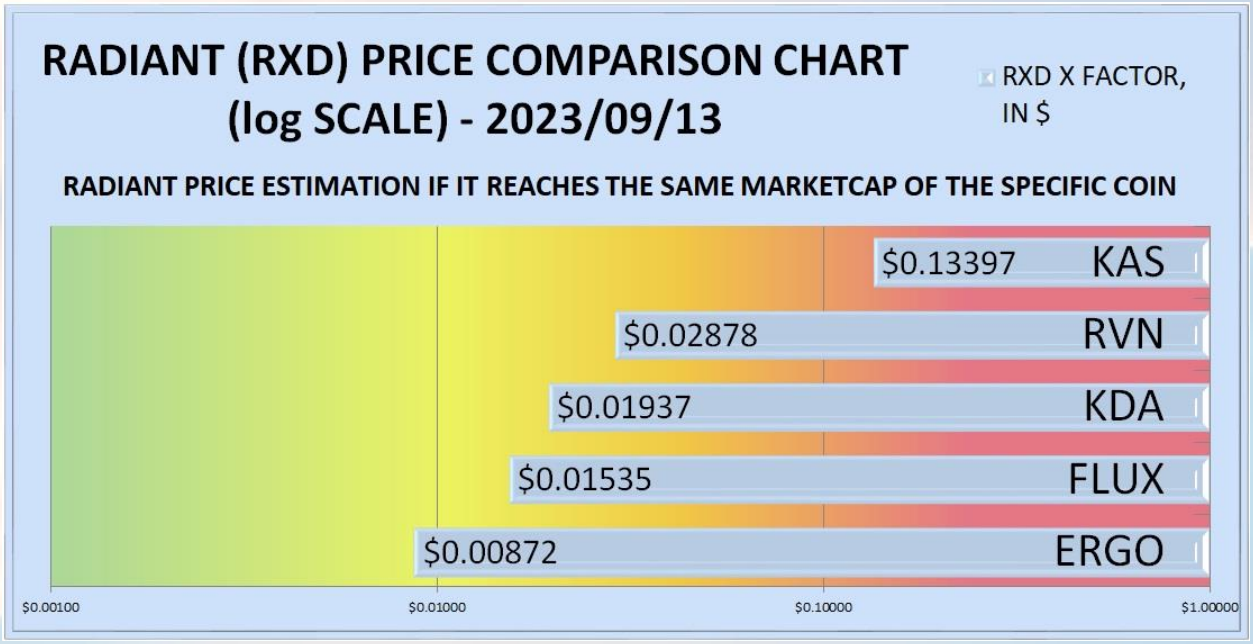
Pointing noteworthy diversities and comparing the smallest with the biggest, just due to numbers alone, we could check Radiant and Kaspa. The latter had an enormous success in the past year and, while being 1 year older than Radiant, it is possessing a 2 times faster halving coin distribution scheme.

A huge difference; 71% of its circulating supply has been already allocated to miners and traders, while Radiant just 34.7%.

Kas had an extremely important success for such a young chain, numbers alone cannot lie but as you all can see and understand it is hard to compare chains with very different mechanics, metrics and methodologies, to not speak of fields of application and precisely community dimension.

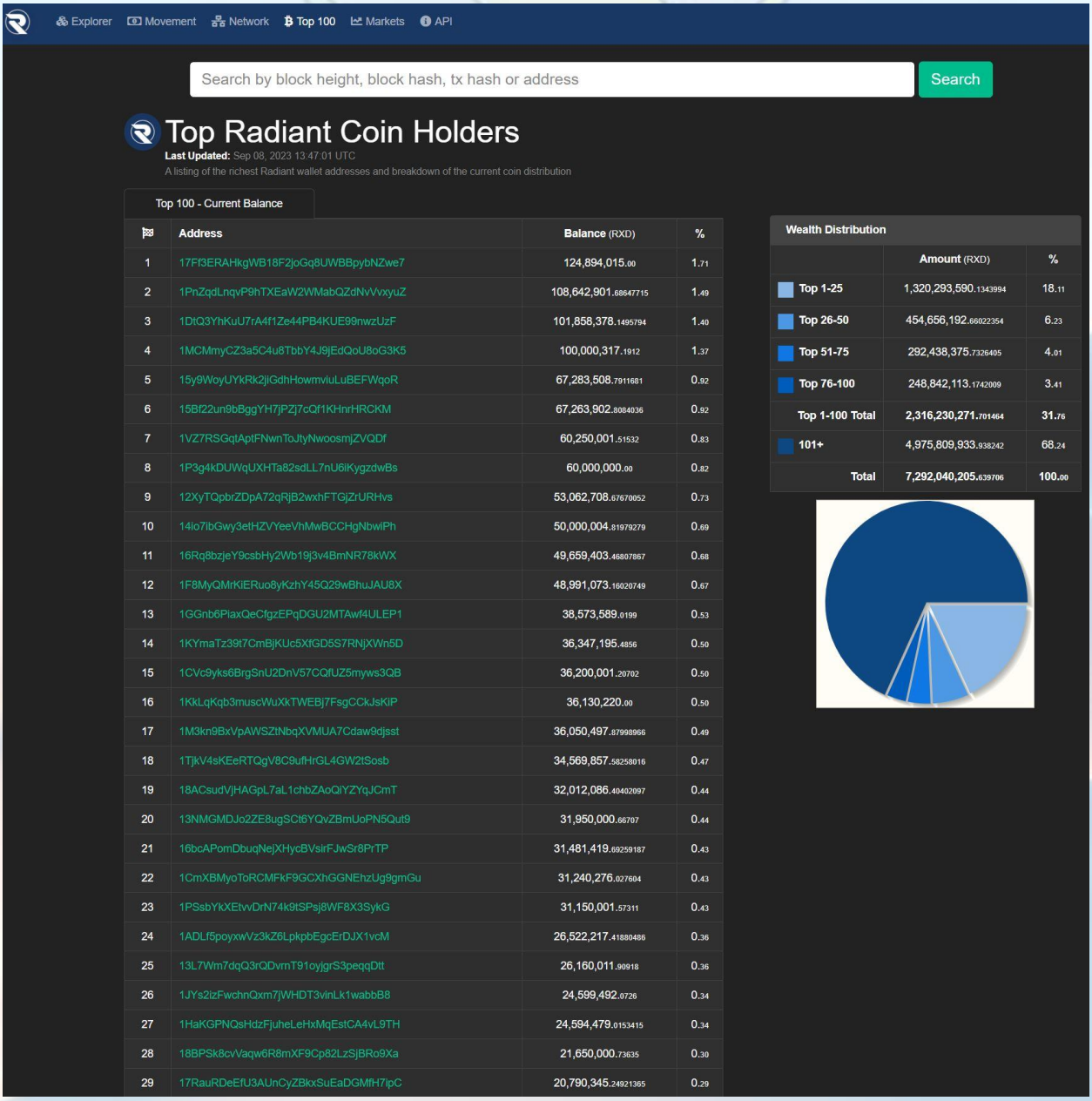
In the following chart a WHAT-IF scenario. What could be the price of Radiant if it reaches the same market cap of the previously mentioned projects?

Interesting numbers. The scale of colors expresses the complexity to reach those values. Clearly all is subject to change and everything can happen, but we are just speaking of pure numbers and ratios here.



The decentralization is a fundamental factor, and this is why it is important that Radiant reaches new communities, being more decentralized in the coin distribution; such element is also quite good at the moment, as shown below in the public Rich List.

https://radiantexplorer.com/richlist



Explorer Movement Network Top 100 Markets API

Search by block height, block hash, tx hash or address Search

Top Radiant Coin Holders

Last Updated: Sep 08, 2023 13:47:01 UTC

A listing of the richest Radiant wallet addresses and breakdown of the current coin distribution

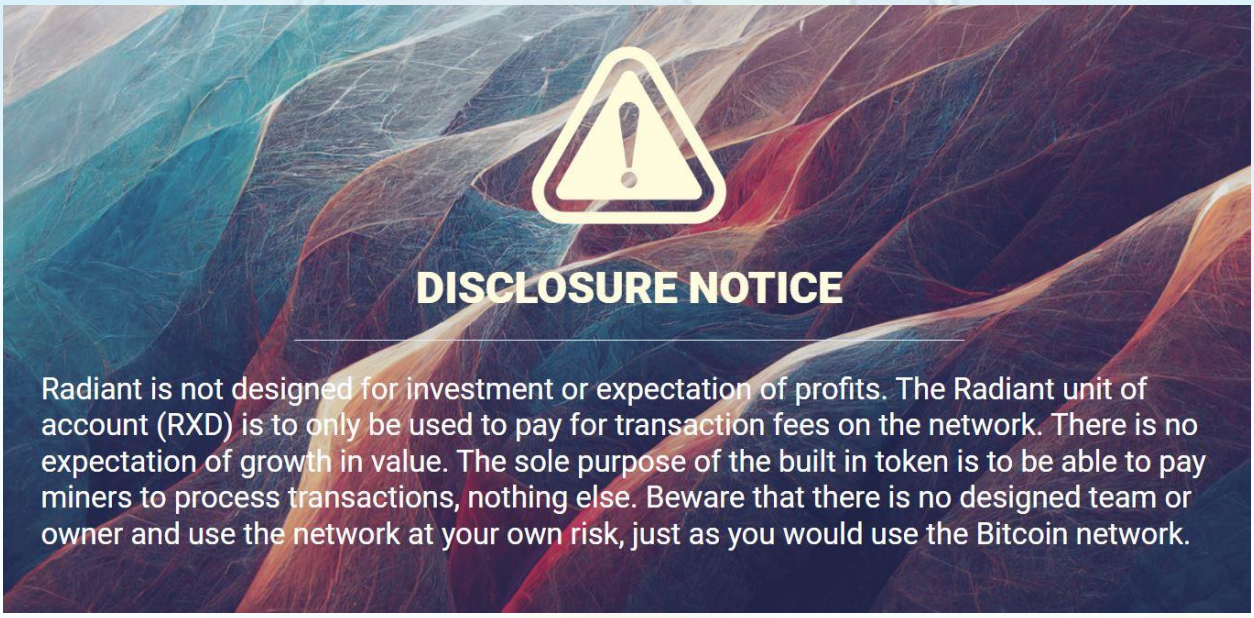
Top 100 - Current Balance

	Address	Balance (RXD)	%
1	17Ff3ERAHkgWB18F2joGq8UWBBpybNZwe7	124,894,015.00	1.71
2	1PnZqdLnqvP9hTXEaW2WMabQZdNvVvxyuZ	108,642,901.68647715	1.49
3	1DtQ3YhKuU7rA4f1Ze44PB4KUE99nwzUzF	101,858,378.1495794	1.40
4	1MCMmyCZ3a5C4u8TbbY4J9jEdQoU8oG3K5	100,000,317.1912	1.37
5	15y9WoyUYkRk2jiGdhHowmviuLuBEFWqoR	67,283,508.7911681	0.92
6	15Bf22un9bBggYH7jPZj7cQf1KHnrHRCKM	67,263,902.8084036	0.92
7	1VZ7RSGqtAptFNwnToJtyNwoosmjZVQDf	60,250,001.51532	0.83
8	1P3g4kDUWqUXHTa82sdLL7nU6iKygzdwBs	60,000,000.00	0.82
9	12XyTQpbrZDpA72qRjB2wxhFTGjZrURHvs	53,062,708.67670052	0.73
10	14io7ibGwy3etHZVYeeVhMwBCCHgNbwiPh	50,000,004.81979279	0.69
11	16Rq8bzjeY9csbHy2Wb19j3v4BmNR78kWX	49,659,403.46807867	0.68
12	1F8MyQMrKiERuo8yKzhY45Q29wBhuJAU8X	48,991,073.16020749	0.67
13	1GGnb6PiaxQeCfgzEPqDGU2MTAwf4ULEP1	38,573,589.0199	0.53
14	1KYmaTz39t7CmBjKUc5XfGD5S7RNjXWn5D	36,347,195.4856	0.50
15	1CVc9yks6BrgSnU2DnV57CQfUZ5myws3QB	36,200,001.20702	0.50
16	1KkLqKqb3muscWuXkTWEBj7FsgCCkJsKiP	36,130,220.00	0.50
17	1M3kn9BxVpAWSZtNbqXVMUA7Cdaw9djsst	36,050,497.87998966	0.49
18	1TjkV4sKEeRTQgV8C9ufHrGL4GW2tSosb	34,569,857.58258016	0.47
19	18ACsudVjHAGpL7aL1chbZAoQiYZYqJCmT	32,012,086.40402097	0.44
20	13NMGMDJo2ZE8ugSCt6YQvZBmUoPN5Qut9	31,950,000.66707	0.44
21	16bcAPomDbuqNejXHycBVsirFJwSr8PrTP	31,481,419.69259187	0.43
22	1CmXBMyoToRCMFkF9GCXhGGNEhzUg9gmGu	31,240,276.027604	0.43
23	1PSsbYkXEtvvDrN74k9tSPsj8WF8X3SykG	31,150,001.57311	0.43
24	1ADLf5poyxwVz3kZ6LpkpbEgcErDJX1vcM	26,522,217.41880486	0.36
25	13L7Wm7dqQ3rQDvmT91oyjgrS3peqqDtt	26,160,011.90918	0.36
26	1JYs2izFwchnQxm7jWHDT3vinLk1wabbB8	24,599,492.0726	0.34
27	1HaKGPNQsHdzFjuheLeHxMqEstCA4vL9TH	24,594,479.0153415	0.34
28	18BPSk8cvVaqw6R8mXF9Cp82LzSjBRo9Xa	21,650,000.73635	0.30
29	17RauRDeEfU3AUnCyZBkxSuEaDGMfH7ipC	20,790,345.24921365	0.29

Wealth Distribution

	Amount (RXD)	%
Top 1-25	1,320,293,590.1343994	18.11
Top 26-50	454,656,192.66022354	6.23
Top 51-75	292,438,375.7326405	4.01
Top 76-100	248,842,113.1742009	3.41
Top 1-100 Total	2,316,230,271.701464	31.76
101+	4,975,809,933.938242	68.24
Total	7,292,040,205.639706	100.00

10.2 Volatile assets trading note



Cryptocurrencies are Volatile-Assets; they are possessing, more often in its initial stage of development small market capitalizations and this can affect in a very positive or negative way its market counter value to USD. Please understand that there might be the absence of liquidity, and that Radiant is not meant as an investment, as Bitcoin was not created for this purpose. Bitcoin too is not meant as an investment since its inception, it is a technology product, that has a ratio of exchange with FIAT currencies, and its up to the user to make trading, using it as a store of value. If you intend to invest in cryptocurrencies, do note that you advance what you can afford to lose.

1) NEVER-EVER put your life investments in speculative assets
2) DO risk management, or inform about this
3) DIVERSIFY your portfolio
4) UNDERSTAND that in Radiant there is no expectation of profits from anyone except yourself. Radiant is a Community Led / Community Owned piece of technology having a ratio of exchange with FIAT, convertible into different FIAT and CRYPTO assets. If you want the project to grow, work on it. If you complain expecting someone else to work for you, if you are promoting listings so you can dump upon other community members that are investing their time and energy for decentralized projects, this is not your place.

"Venturing into Bitcoin and cryptocurrencies is like sailing in a stormy sea"

Quoting Investopedia: "Thus, many people purchase Bitcoin for its investment value rather than its ability to act as a medium of exchange. However, the lack of guaranteed value and its digital nature means its purchase and use carry several inherent risks. For example, many investor alerts have been issued by the Securities and Exchange Commission (SEC), the Financial Industry Regulatory Authority (FINRA), and the Consumer Financial Protection Bureau (CFPB) regarding Bitcoin investing.

Regulatory risk: The lack of uniform regulations about Bitcoin (and other virtual currencies) raises questions over their longevity, liquidity, and universality.

Security risk: Most individuals who own and use Bitcoin have not acquired their tokens through mining operations. Rather, they buy and sell Bitcoin and other digital currencies on popular online markets, known as cryptocurrency exchanges. Bitcoin exchanges are entirely digital and—as with any virtual system—are at risk from hackers, malware, and operational glitches.

Insurance risk: Bitcoin and cryptocurrencies are not insured through the Securities Investor Protection Corporation (SIPC) or the Federal Deposit Insurance Corporation (FDIC). Some exchanges provide insurance through third parties. In 2019, prime dealer and trading platform SFOX announced it would be able to offer Bitcoin investors FDIC insurance, but only for the portion of transactions involving cash.

Fraud risk: Even with the security measures inherent within a blockchain, there are still opportunities for fraudulent activity. For instance, in July 2013, the SEC brought legal action against an operator of a Bitcoin-related Ponzi scheme.

Market risk: As with any investment, Bitcoin values can fluctuate. Indeed, the value of the currency has seen wild swings in price over its short existence. Subject to high volume buying and selling on exchanges, it is highly sensitive to any newsworthy events. According to the CFPB, the price of Bitcoin fell by 61% in a single day in 2013, while the one-day price drop record in 2014 was as big as 80%.

Source: LINK TO INVESTOPEDIA - BITCOIN AS AN INVESTMENT

"Use caution managing your assets, you only take responsibility for your actions, in life and mostly in the financial markets"

10.3 RCR END-NOTES

Radiant is Proof Of Work also in the meaning that each user has to "Prove his own contribution, his own Work".

The success of the single is the success of the entire community, and this is precisely the intention why I wrote down the Radiant Community Report: the RCR is a powerful tool from Unpaid & Unaffiliated Community Members, at the service of everyone.

Each one of Us can express his ideas, experience, organize the community, lead as he desires, and all of this in a environment where free-speech is protected, where education and respect has to pave the way for the common success of a real community driven project.

As a two-stage thermonuclear warhead, while this document is the detonator to light the chemical explosive of the primary stage, the Radiant Community is instead the lithium deuteride (fusion fuel) that allows the nuclear reaction to grow exponentially.

To achieve success, each one of Us needs to understand that we can truly let the chain reaction proceed, grow logarithmically, reach new potency, if only we work together as a decentralized team, willing to cooperate and mature.

This is the potential of each member, the true power of Radiant.

Join Radiant on DISCORD: https://discord.com/invite/radiantblockchain

(One of the wonderful Deathmumi creations)

Sail with Radiant in this stormy ocean

Radiant is here to Stay, to Unite and to Conquer!

Thanks to everyone that contributed and thanks for all the future contributions.

Join the Radiant Revolution

Per Aspera, Ad Astra!

Made By SHaRD. Tech Papers and Radiant4People by ANTARES
(Thanks to HippiePyro, Deathmumi, gsb, ArtOfSatoshi, CryptoNaut)