Evrmore: A Peer to Peer System for Decentralized Finance Applications

Hans Schmidt
WhaleStreet
www.evrmore.foundation

14th September 2022

Thank you to Satoshi Nakamoto and all the Bitcoin developers for Bitcoin. Thank you to Bruce Fenton, Tron Black, BlondFrogs and all the Ravencoin developers for Ravencoin.

Abstract. Evrmore (EVR) is a blockchain with built-in asset and DeFi (decentralized finance) primitives. Evrmore is based on the Bitcoin (BTC) UTXO model, is mined publicly and transparently using Proof-of-Work, is free and open source and is open for use and development in any jurisdiction. Evrmore is built as a fork of Ravencoin (RVN) and grew out of DeFi discussions and research within the Ravencoin community. Evrmore forks Ravencoin's coins but not assets. It takes measures to protect Ravencoin because it seeks not to replace Ravencoin, but to co-exist with a different purpose. Ravencoin extended the Bitcoin UTXO model with assets thereby eliminating errors commonly caused by implementing assets as "colored coins." Evrmore similarly extends the Bitcoin UTXO model with DeFi primitives thereby eliminating errors commonly caused by implementing DeFi in general-purpose smart contracts – implementing DeFi functionality using powerful and general-purpose smart contracts is not the correct solution. In addition to DeFi primitives, Evrmore includes features that improve transaction speed, flexibility, zero-confirmation, minimal transaction fees, scaling and complex covenant financial derivatives. Evrmore also includes code development financing and uses the EvrProgPow mining algorithm to keep hardware requirements affordable and competitive.

1. Introduction

Bitcoin [1] has proven the value of the blockchain and the UTXO architecture as a better form of money by remaining the dominant cryptocurrency by a large margin. Extending Bitcoin to applications beyond cash replacement has been more complex, resulting in a large number of cryptocurrency projects with varying degrees of success. Support for assets as tradeable tokens has taken place using three primary strategies.

- A few projects, including Mastercoin [2] and Counterparty [3], have used the

Bitcoin blockchain as a protocol layer on which to build new coins.

- Other projects, including the SLP project [4], which uses BCH, have used the Bitcoin UTXO outputs directly to create new assets, often called "colored coins."
- Ethereum (ETH) [5] has emerged as the 2nd leading blockchain, with support for powerful smart contract programmability. Ethereum and similar projects have been wildly successful, especially as platforms for decentralized finance. But these platforms, due to the general-purpose nature of their programmability, have largely failed to deliver on the important requirement of security.

Ravencoin [6] successfully demonstrated that extending the UTXO protocol by implementing asset primitives results not only in greater simplicity, but also enables a low risk means to provide greater asset functionality with relative ease. It is in this context that Evrmore will extend the UTXO protocol by implementing DeFi primitives. Implementing DeFi primitives in the protocol layer necessarily leads to greater simplicity, which means easier development, lower cost, and more reliable security.

2. Equally Valid Purposes

Cryptocurrency projects have distinct philosophical, financial, and organizational characteristics that differentiate their project "styles." While it is not possible for anyone to speak on behalf of the entire decentralized Ravencoin community, that community has shown a preference for conservative choices. Proposals for any type of governance or funding have been declined, even with mechanisms to ensure that control and decision making remain decentralized. Ravencoin technical roadmap discussions have focused on following the Bitcoin codebase and philosophy, including Segwit and Lightning. Most of Ravencoin's funding was originally provided by Medici (a division of Overstock, Inc.), which made significant business investments in promoting the growth of centrally issued and controlled Security Token Offerings (STOs). The STO market has been slow to develop because of regulations and entrenched players, but it will, no doubt, grow over time. The slow initial growth of the STO market has been beneficial as it has given Ravencoin time to mature and Evrmore time to develop additional capabilities which can work in parallel.

Evrmore (EVR), an upgrade proposal originally called "Evermore," was proposed within the Ravencoin community as an aggressive roadmap to participate in the DeFi economy with new features and strategies. Evrmore's technology generated great interest within the Ravencoin community [14], but Evrmore clearly differed from Ravencoin's established cautious approach. The perceived issues of radically changing Ravencoin's feature set and style motivated some members to suggest that Evrmore should be a separate project within the Ravencoin community. Evrmore is being built by current and former Ravencoin developers, hodlers, and miners who understand these issues and concerns. Far from being a replacement of Ravencoin, Evrmore simply represents an alternative set of trade-offs with different features aimed at providing solutions to a different set of problems.

Evrmore is currently under development and plans to friendly-fork the Ravencoin blockchain in October 2022. This friendly fork will create an EVR for every RVN, but it will not create any RVN assets on the Evrmore chain. Evrmore will employ measures to protect Ravencoin because it seeks not to replace Ravencoin, but to co-exist with a different purpose and new capabilities.

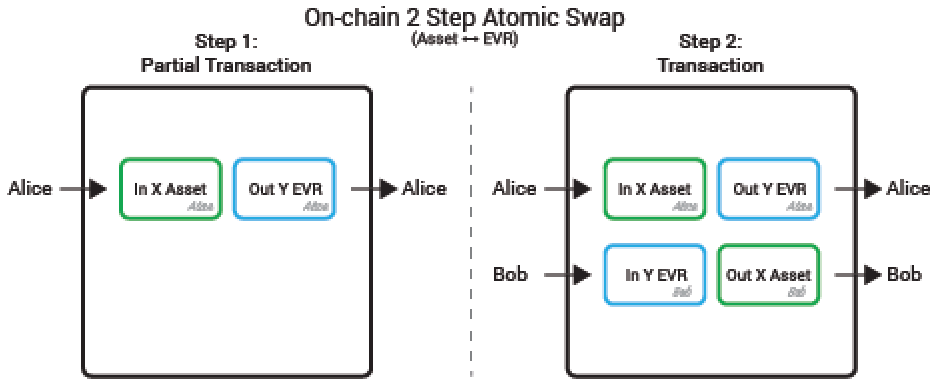
Like Ravencoin, Evrmore is based on the Bitcoin UTXO model, is mined publicly and transparently using Proof-of-Work, is free and open source, and is open for use and development in any jurisdiction. Evrmore's focus is DeFi. DeFi today is a quickly evolving field that requires active code developers and a mechanism to pay their salaries. In support of this need, Evrmore collects a fee on coinbase rewards – 10% that will be managed by the Evrmore Foundation for development contracts, grants etc. Evrmore values decentralization of mining. In support of the mining community, Evrmore uses a variation of the KawPow algorithm, EvrProgPow, to keep hardware requirements affordable and competitive.

3. Primitives

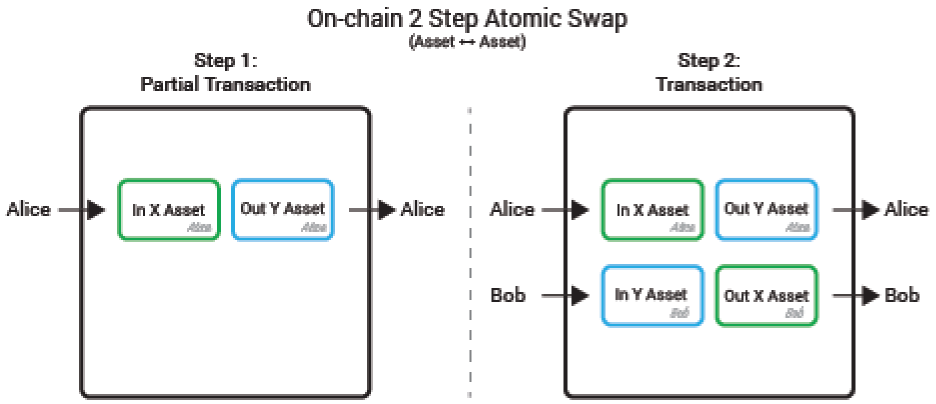
A primary goal of Bitcoin as decentralized money is to ensure only the owner controls the use of the cryptocurrency while making theft and confiscation impossible. Nevertheless, scams and hacks have been a growing problem in the cryptocurrency space since the beginning [7]. Many of the largest losses were caused by custodial centralized exchanges, the most famous of which, Mt Gox [8], has become a poster child for the motto "Not your keys, Not your Bitcoin." It was thought that DeFi would eliminate theft by providing non-custodial exchanges while also providing more attractive and more flexible blockchain-based alternatives to traditional mainstream savings and investment vehicles. The reality thus far has proven to be quite different [9]. Social-Engineering scams continue to be a problem for investors of all kinds and probably always will be. But failures of DeFi technology due to smart contract hacks have become an even larger ever-growing problem. Hacks in DeFi, due to poorly designed smart contract platforms, poorly written smart contracts, unexpected interactions between smart contracts and the improper use of smart contracts, have become a nearly daily occurrence. The size of monetary losses continues to grow, with multiple hacks in the hundreds of millions of dollars [10]. The first major hack of this type, the DAO hack, shook the Ethereum community and resulted in a fork of the Ethereum blockchain, which created Ethereum Classic [11]. The Poly Network [12] hack was much larger yet triggered far less emotion in the DeFi community, which has become numb to the constant exploitation of smart contract vulnerabilities. Implementing DeFi functionality using powerful and general-purpose smart contracts is not the correct solution.

The history of computer networking protocols has demonstrated a standard path to increase security and performance. As increasingly complex functions became common, standardized, and indispensable they were then built-in as native primitives. Using the internet as an example, TCP was added alongside UDP so that every network application or stack need not create its own connection-oriented stream solution. Likewise, HTTPS was added alongside HTTP so that each web page need not re-implement an encryption solution in order to benefit from secure connections. Supporting commonly used functions as protocol primitives is a powerful tool for improving security, usability and performance.

Atomicity is a requirement of most financial transactions, in order to ensure either all parts or no part of a transaction takes place. In Bitcoin *all* or *none* of a transaction’s inputs and outputs are completed. A withdrawal of funds should never be allowed, for instance, unless a corresponding deposit elsewhere also takes place. Yet ensuring atomicity in smart contracts is a complex task that often fails. Even for UTXO-based blockchains, complex methods such as Hash-Time-Locked-Contracts have been required to guarantee that *swaps* are atomic. Yet *transaction transfers* on the same blockchain exhibit atomicity that is used to great advantage. In traditional HTLC atomic swaps, the parties must participate in a multi-step procedure, including making an offer, the acceptance of the offer, each party making an interlocked payment, and each party accepting payment from the other party. At the BTC transaction level, a much simpler two-step atomic swap is possible using a protocol feature called "SIGHASH_SINGLE| SIGHASH_ANYONECANPAY" [13]. In a two-step swap with asset primitives, no upfront negotiation is needed. The seller posts an offer that is a partial transaction. This partial transaction communicates terms, "I am willing to sell X and I don't care about any other inputs or outputs as long as I get Y." Any buyer who likes the terms can accept the offer by completing the transaction and submitting it. With asset primitives as on Ravencoin and Evrmore, two-step swaps involving *multiple assets* and *asset types* become possible in a *single* transaction. This swap transaction has guaranteed atomicity and optimal performance. Such is the power of protocol level asset primitives.



Alice creates a partial transaction. In this partial transaction Alice is selling *X Asset* in exchange for *Y EVR*.
Alice broadcasts this partial transaction. Bob sees Alice's partial transaction.
Bob completes the partial transaction by buying Alice's *X Asset* in exchange for his *Y EVR*.
No further steps are required.



Alice creates a partial transaction. In this partial transaction Alice is selling *X Asset* in exchange for *Y Asset*.
Alice broadcasts this partial transaction. Bob sees Alice's partial transaction.
Bob completes the partial transaction by buying Alice's *X Asset* in exchange for his *Y Asset*.
No further steps are required.

Bitcoin's simple UTXO protocol design is its strength, which Ravencoin extended to natively support *asset primitives*. Evrmore takes the next evolutionary step by further extending Bitcoin’s UTXO model to natively support partial order fulfillment, complex orders, interest rates, oracles, and many other useful *DeFi primitives*. All are implemented at the lowest possible level. All are effortlessly reusable by DeFi applications with security and performance confidence. Compared to general purpose smart contracts, smart contracts on Evrmore are simpler, more secure and in many cases unnecessary as their functionality is built-in at the lowest protocol layer. Smart contract programmers benefit from built-in DeFi protocol primitives by being able to concentrate on the new functionality they are creating. They don’t have to re-create every wheel or worry they have broken something that worked previously. These advantages result in a much better user experience.

4. Evrmore Protocol Extensions

This section details some of the asset-related technical changes that Evrmore will implement. These may change as work progresses. The following details assume the reader has an understanding of the Ravencoin architecture, features and codebase. Only a short description is provided. These details do not fully describe Ravencoin's current asset capabilities including assets, subassets, unique assets, restricted assets, along with various characteristics and options for each capability. They also do not describe the transaction speed, flexibility, zero-confirmation, minimal transaction fees, scaling and complex covenant financial derivatives that Evrmore will merge.

Add a Second IPFS Field to Asset Metadata

- One of the IPFS fields is treated as immutable once the asset is issued, even during reissue. The second IPFS field is always mutable, even for unique assets and other non-reissuable assets. This change needs additional RPC and GUI support.

Add Owner-Required Asset Transfer Tolls Support

- Each asset has a metadata field indicating a required per-asset transfer toll. This toll is a quantity of EVR, or in the case of vault tokens a percent of face value. Another metadata field specifies the address to where the toll payment is made.
- The toll quantity and address fields will be selected at asset creation time. These field values can be changed during reissue, unless the toll amount is zero, in which case it will always be zero.
- The toll is paid as an EVR output to the toll address. This requirement is enforced as a consensus item by all nodes.

Add Expiring Unique Assets

- Unique assets are given a lifetime when they are created. There is a minimum lifetime (1 hour) and a minimum cost (0.05 EVR) to prevent spam and to limit computational load. Only a fixed set of lifetimes and associated prices will be supported.
- Proposed lifetimes and associated prices:
    - 1hour/0.05 EVR
    - 1day/0.1EVR
    - 1week/0.2EVR
    - 1month/0.5EVR
    - 3months/1EVR
    - 6months/2EVR
    - 1year/4EVR
    - Infinite/5EVR
- For example, these assets can represent concert tickets (serial numbered with assigned seating) that expire after the concert, 1-hr worth of VPN service or a 10-year bond.
- The expiring nature of this asset type makes it ideal for representing an expiring options or futures contract. Such securities exhibit unique pricing behavior because expiring options contracts trend to a value of zero at the time of expiration, while expiring futures contracts trend toward the spot market price.

Add Asset Consumption Support

- A new script opcode for a “Consume()” asset function for any asset. Note that while an asset can be "burned" by sending it to a non-spendable address, there are advantages in being able to destroy assets and remove them from the UTXO.

- If used for voting, destruction can be done after voting is completed. More generally, if the asset is a utility token, it can be consumed after it is used to obtain the service.
- If used as a "wrapped" asset, for example EVR-Wrapped-BTC, consuming provides a way for the supply of EVR-Wrapped-BTC to be made equal to the number of BTC locked on the Bitcoin blockchain.
- Adds a “Consumable(T/F)” metadata to every asset that determines whether to allow the “Consume()” asset function.
- During reissue, “Consumable(T/F)” can be changed from T to F but never from F to T.
- Owner tokens may not be consumed. Therefore there is no concern about whether a subasset name exists below any given asset name.

Implement Vault Assets

- Vault assets have a "face value" of EVR and can be "melted" to extract the EVR.
- Vault assets can pay a daily interest rate, or cost a daily demurrage rate, as a percentage of the “face value” or as a fixed amount of EVR.
- Vault assets are a special type of root asset or subasset. They follow the same naming rules as root assets and subassets except their name must always start with the ^ (carat) symbol (conveniently resembling an upside-down "v").
- This naming convention makes it clear that the asset is a vault asset that has a face value and pays/costs an interest/demurrage rate from/to an address.
- Vault assets look like ^MYNAME/^MYSUBNAME.
- Vault asset names may only be issued by the owner of the root asset of the same name. For example ^MYNAME may only be issued by the owner of MYNAME!.
- Vault assets are supported by new vault-related RPC commands "mintasset", "monetizeasset", "meltasset", "unmeltasset" and "remintasset."
- Vault assets can be further extended by two capabilities that allow extraction of face value from assets. These capabilities can be used as a mechanism for DAO cash-outs. The two capabilities, which must pass a vote by the token holders in order to be used, are:
    - Vote for Unmelt at face value
        - This capability doesn’t change the value of each outstanding token, rather it reduces equity share and thus voting power per token.
    - Vote for Remint at zero value
        - This capability is useful for funds withdrawal off-chain; it reduces value and equity share per token.

Add Level-1 DEX Support

- To communicate a desired trade, enhanced support for 2-step

SIGHASH_SINGLE|SIGHASH_ANYONECANPAY on-chain atomic swaps is added, by adding fields, to every UTXO:
    - Chain
    - Asset_Name
    - Ask_Qty
    - Expire_Time
    - Signature
- These new data fields do not alter the behavior of the transaction's output they are written into. They contain the information necessary for anyone to create a spending transaction that accepts the offer described by the new data fields.
- Traders can wait for the new UTXOs to be mined before submitting matching acceptance transactions. But nothing prevents buyers from snooping the P2P node traffic, in which case an offer and its acceptance could be mined sequentially into the same block.
- A separate database and separate communication channel can be used to track and broadcast offers in order to save on transaction fees.

Add Level-2 DEX Support

- On-chain 2-step atomic swaps are limited in efficiency because buyers must match the exact qty of an offer by creating a matching UTXO. Fixed pricing is another limitation. This enhancement addresses these weaknesses.
- On-chain 2-step atomic swaps can support partial order fulfillment with specified granularity. This is done by adding two new UTXO data fields "Granularity" and "Change Address," and related consensus rule changes.
- By adding new UTXO data fields: "Order_Type," "Stop_Qty" and "Limit_Qty," and related consensus rule changes, on-chain 2-step atomic swaps can support new order types: precise(legacy), market, stop, limit, stop limit and stop market.
- Note these new data fields do not alter the behavior of the transaction's output they are written into. However, during a future transaction that spends that output by using it as an input, the consensus rules read these new fields to determine whether the spending transaction satisfies the requirements set by these fields. The spending transaction may have a Vout value (final settlement quantity) that is different than "Ask_Qty." This is due to partial order fulfillment (based on Granularity) or price change (within the rules set by "Order_Type," "Stop," and "Limit").
- Independent vendors are responsible for matching offer and seller half-transactions for submission to the miners. These vendors can make a profit on the spread.

Add Oracle DEX Support

- On-chain 2-step atomic swap support for oracle-based price discovery provides another vast improvement in price discovery and flexibility. Markets use many different methods of price discovery. Some use traditional off-chain bid/ask order books. DeFi projects are experimenting with a variety of automatic-market-maker algorithms. Oracle DEX support allows an offer creator to select any oracle of their choice based on the oracle's public key and signed price reports. Only the price quote and signature format need to be standardized, and a few changes need to be made to the UTXO definition and consensus rules. The buyer can choose an oracle from any independent vendor who provides that service on a for-profit basis.

- Implementing oracles requires two new UTXO data fields: "Oracle_Pubkey" and “Oracle_Quote.” “Oracle_Pubkey” is written into the transaction output of the transaction making an offer. "Oracle_Quote" is included along with Vout in the spending transaction that accepts an offer.
- Note that the oracle is chosen by the owner of the UTXO making the offer. It is up to the other party to decide whether to accept that oracle by taking the deal.
- If a token being traded represents an off-chain asset, then even with oracle price setting, the token requires a stable coin mechanism to bind the token to the off-chain asset, or it must trade in a pair with a stable-coin asset. This is needed to provide an arbitrage mechanism to anchor the on-chain price to its off-chain value. Otherwise, oracle inaccuracies and computational round-off errors will accumulate to create growing token price inaccuracy.
- Note, however, that no off-chain anchor is needed if the value is fully on-chain. A token containing a predictions market contract, which accepts EVR from bet participants and pays out based on oracle information, needs no off-chain anchor.

Add Covenant Support

- Covenants allow Evrmore script to support more complex smart contracts within limitations. In particular, a token or stablecoin’s options or futures contract will be able to reside in a separate contract token. This contract token will be able to trade separately from the underlying value token.
- This capability is a great motivator to attract liquidity providers. People that lock up their Bitcoins on the BTC blockchain and accept EVR-WRAPPED-BTC assets on the EVR blockchain, can then write and sell covered options and futures contracts against their bitcoins.

5. Evrmore Foundation

Please see www.evrmore.foundation for more details about the Evrmore Foundation.

6. Naming Convention

- Platform: Evrmore (ev·r·more | /ˌevərˈmôr/)
- Ticker: EVR (ev·r | /ˈevər/)
- 10^8: evr (ev·r | /ˈevər/)

References

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

[2] "A Brief History of MasterCoin" https://blog.omni.foundation/2013/11/29/a-brief-history-of-mastercoin/

[3] Counterparty https://counterparty.io/

[4] Simple Ledger Protocol (SLP) https://simpleledger.cash/

[5] Ethereum https://ethereum.org

[6] "Ravencoin: A Peer to Peer Electronic System for the Creation and Transfer of Assets" https://ravencoin.org/assets/documents/Ravencoin.pdf

[7] "The History of Crypto Hacks — Top 10 Biggest Heists That Shocked The Crypto Industry" https://medium.com/ngrave/the-history-of-crypto-hacks-top-10-biggest-heists-that-shocked-the-crypto-industry-828a12495e76

[8] Mt Gox https://en.wikipedia.org/wiki/Mt._Gox

[9] "Documented Timeline of DeFi Exploits" https://cryptosec.info/defi-hacks/

[10] "The Speed of Crypto Hacks is Picking Up: This Month Alone Thieves Stole $71.5M" https://howmuch.net/articles/biggest-crypto-hacks-scams

[11] Ethereum Classic https://ethereumclassic.org/knowledge/roadmap

[12] "Cross-Chain DeFi Site Poly Network Hacked" https://finance.yahoo.com/news/poly-network-hacked-135650605.html

[13] Alex Mizrahi's usage description of SIGHASH_SINGLE|SIGHASH_ANYONECANPAY https://groups.google.com/forum/#!msg/bitcoinx/pON4XCIBeV4/IvzwkU8Vch0J

[14] "Ravencoin Community Discord" https://discord.gg/jn6uhur