Counterparty

An Extension to Bitcoin by State-Machine Replication

Adam Krellenstein

2024-03-29

Contents

Background

Development of Counterparty began in late November of 2013. On January 2nd of the following year, Counterparty was announced with a post on the Bitcoin Forum and immediately released into production.[1] Since that date, the Counterparty network has been live and in active use by the public. This whitepaper is thus being written ten years late—the original intention was to allow the Counterparty codebase itself to document its feasibility; however there is nevertheless a benefit to formalizing a description of the protocol design and history, especially in light of persistent misunderstandings regarding certain features of the design, the peculiar role that Counterparty has played in the history of the blockchain ecosystem, and the recent resurgance of interest in Bitcoin "L2" protocols.

The idea for Counterparty originally came from the Mastercoin project.[2] Mastercoin had, a few months before the launch of Counterparty, been described by J.R. Willett as part of the first-ever ICO.[3] The founders of Counterparty—Adam Krellenstein, Evan Wagner and Robby Dermody—saw the immense potential in the design for Mastercoin, but were unsatisfied with the general manner in which that Mastercoin was launched and developed. Counterparty was an effort to realize the architectural vision behind Mastercoin but, in contrast, in accordance with the principles and values of Satoshi and his creation.

Even before Mastercoin had any functionality beyond simple transfers of its own native token, Counterparty was able to demonstrate the extensive capabilities of a feasibility of what would later be called a **metaprotocol** or **metachain**, becoming the model implementation of the architecture originally devised by Willett. Counterparty has since inspired a number of imitators and second-order metaprotocols, such as STAMPS, SRC-20, BRC-20 and Mastercoin itself. But the architectural features of Counterparty are not widely understood, and this is a consequence partly of the idiosyncratic design of Bitcoin, the incidental features of which are incorrectly held to be essential characteristics of all blockchains.

State-Machine Replication

Blockchains are commonly referred to as "immutable databases", because, unlike a traditional *mutable* database, a blockchain has no cen-

[1] https://bitcointalk.org/index.php?topic=395761.0

[2] https://bitcointalk.org/index.php?topic=56901.0

[3] https://www.forbes.com/sites/laurashin/2017/09/21/heres-the-man-who-created-icos-and-this-is-the-new-token-hes-backing/

tral authority that is capable of changing its state at will. With a blockchain, the data are log-structured, replicated globally by thousands of anonymous entities, and secured by proof-of-work. The term “blockchain”, however, does not refer only to the historical log of blocks and transactions, but also to the decentralized system itself, i.e. to the protocol and the instantiation of that protocol in a particular network. The Bitcoin blockchain in particular is more than just a database—more than just the data structure which comprises the collection of canonical blocks and transactions—it also describes the protocol that is manifest in a reference implementation and all clones thereof.

Without the Bitcoin Core codebase, Bitcoin would not be completely specified. The blockchain *qua* historical log does not determine future block reward halvings, for instance. More relevantly, it does not determine *per se* how those data are to be interpreted; for instance, how balances are calculated as the sum of the values of UTXOs spendable by that address. These balances must be calculated deterministically, so that each node in the network reports the same balance for a given copy of the blockchain log, and the balances are not themselves recorded in blocks nor verified by miners.

If some alternative implementation of the Bitcoin protocol were ever to report a balance different from the one reported by Bitcoin Core, then it would, definitionally, be an incorrect implementation of Bitcoin. Indeed, it is an interesting feature of the design of Bitcoin that the vast majority of the Bitcoin protocol is validated by miners, so invalid transactions themselves generally never make it into the transaction log, say. However, one can easily imagine an alternative design in which double-spend transactions are included as useless data in blocks and ignored by each node when calculating the list of (valid) UTXOs for a given address. Strictly speaking, a blockchain is not just data—it is also *logic* and *state*.

The defining feature of a blockchain is the architectural pattern of **state-machine replication**. There are two components to the architecture: a distributed log (i.e. the list of blocks) and a state machine that parses that log (deterministically) and stores a local state that is identical to that of every other node. The log is replicated across all Bitcoin nodes, and each node reports the same network state based on the interpretation of the data in that log.

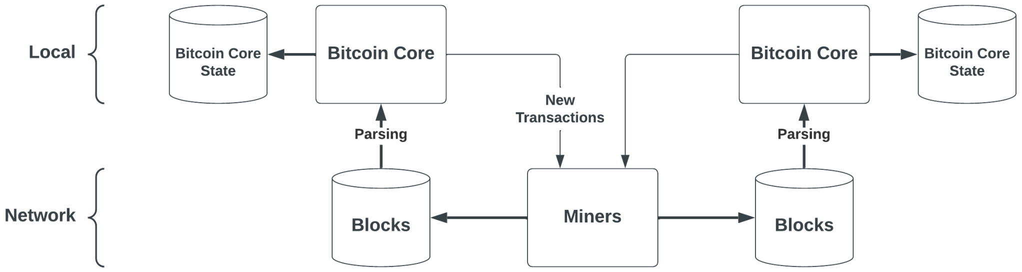

Figure 1: Hölder's reconstruction of Archimedes' proof

Counterparty, a Blockchain Metaprotocol

As a true metaprotocol, Counterparty takes the architectural pattern of a blockchain to its logical conclusion: Counterparty extends the Bitcoin protocol by adding parsing logic and derivative state with the aid of a novel state machine that treats the Bitcoin blockchain log as a store for new protocol messages. Data in the blockchain data structure that are explicitly ignored by Bitcoin nodes are parsed by the Counterparty software and additional state is derived deterministically.

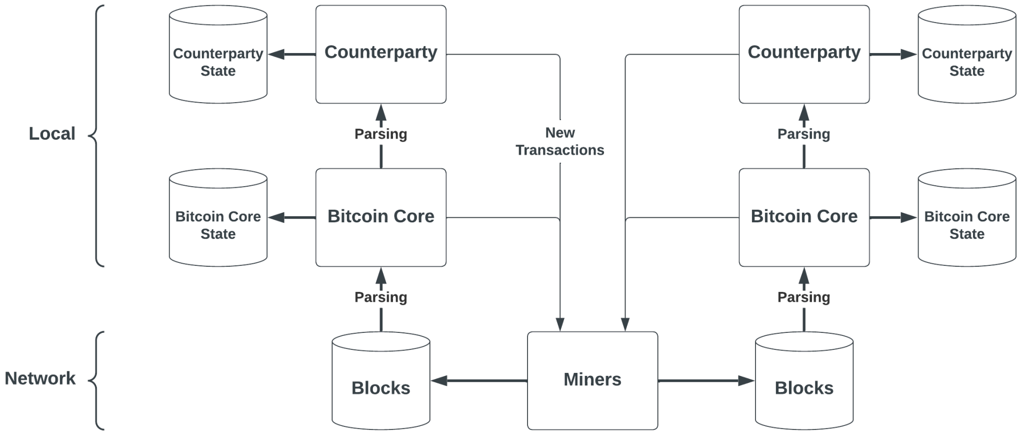

Figure 2: Hölder's reconstruction of Archimedes' proof

Counterparty has the same security model as Bitcoin: every Counterparty transaction is a Bitcoin transaction, and the complete history of Counterparty transactions is thus secured with the full hash power of the Bitcoin mining network. The only difference is that users of Counterparty must rely on *two* codebases—Bitcoin Core and Counterparty Core—rather than Bitcoin Core alone. Bitcoin miners, being 'unaware' of Counterparty, validate a smaller part of the Counterparty protocol than the Bitcoin protocol; but this distinction is quantitative rather than qualitative. This was also the case with the various colored coins protocols, including Ordinals.

If someone runs a version of the Counterparty software that is not compliant with the Counterparty protocol, as it is implicitly specified in the reference implementation, the consequences are exactly the same as if someone were to run a Bitcoin full node that disagreed with Bitcoin Core. The miners of the Counterparty network are simply Bitcoin miners, which presents no more of a problem than the fact that a Bitcoin full node need not be a miner, and a Bitcoin miner need not run a full node. Nevertheless, the Counterparty protocol cannot experience hard or soft forks that are not also hard or soft forks of Bitcoin.

Counterparty Transactions

The Counterparty protocol is thus truly an extension to the Bitcoin protocol. It implements a number of features that Bitcoin does not provide for. These include, but are not limited to, token issuance, a fully decentralized and trustless asset exchange, contracts for difference, native oracles and trustless gameplay. Counterparty 'writes in the margins' of Bitcoin transactions: to create a Counterparty transaction, the Counterparty software constructs a Bitcoin transaction that includes within it metadata that constitute the messages of its protocol.

For instance, one may issue a token on Counterparty by building a Bitcoin transaction that sends dust to oneself and includes an additional output with the `OP_RETURN` opcode from Bitcoin Script.[4] This opcode, when parsed by a Bitcoin node, causes the execution of the script to ignore the subsequent bytes of data. So, in the case of the simplest `issuance` transaction, Counterparty encodes the name of the token and the quantity to be issued in this output. Once the Bitcoin transaction has been broadcast to the network and mined, every instance of the Counterparty software sees the transaction in the Bitcoin blockchain and parses the issuance, storing the event in its local database. Each Counterparty user may then agree that the token has been issued with the specified identifier in the desired quantity.

Counterparty implements its own rules for parsing that metadata. In the case that a Counterparty transaction is invalid—for instance, it comprises a `send` for a token that the sender does not hold enough of—that transaction will be stored in the Bitcoin blockchain but be recognized as invalid by all Counterparty nodes. Counterparty data is thus stored in valid Bitcoin transactions, in the Bitcoin blockchain, and are simply ignored by Bitcoin full nodes. An invalid transaction is then simply a Bitcoin transaction like any other, except as defined by the Counterparty protocol.

[4] https://en.bitcoin.it/wiki/OP_RETURN

Transaction Encoding

Counterparty transaction data are embedded in Bitcoin transactions using a number of methods. The simplest, using the `OP_RETURN` opcode, is also the default. However, arbitrary restrictions on the relaying of transactions with larger `OP_RETURN` data have made it necessary for Counterparty to encode data with other methods, even those which do not allow for the data to be pruned from the Bitcoin blockchain later. Counterparty prefers the use of prunable outputs wherever possible, however. With prunable data, Bitcoin nodes are under no compulsion to store Counterparty transactions indefinitely. Nevertheless, as long as *some* Bitcoin nodes do not prune this data (for instance the Bitcoin nodes that Counterparty users themselves run), those data can never be lost.

A longstanding controversy relates to question of the ethics of using the Bitcoin blockchain for storing transactions not related to Bitcoin itself. Data from metaprotocols such as Counterparty are even sometimes labeled “SPAM”, however, Counterparty data is semantically meaningful and not generated in bulk. The operation of Counterparty, and of other Bitcoin Layer-2 (L2) networks, does not involve the exploitation of any vulnerabilities in the Bitcoin protocol, such as would allow them, for instance, to bypass the Bitcoin fee system. Rather, creating a Counterparty transaction involves paying Bitcoin fees as with a normal Bitcoin transaction: the Bitcoin fees paid are directly proportional to the burden placed on the network for the relaying and mining of Counterparty data. Precisely if the economic value associated with a Counterparty transaction is greater than the required fees does the creation of the transaction mean the addition of value to the network.

Most importantly, Bitcoin is first and foremost a permissionless platform with a protocol driven by economic incentives for honest participation. Lamenting the creation of fee-paying Counterparty transactions is tantamount to lamenting the use of Bitcoin itself for particular ‘undesirable activities’. Counterparty transactions, due to their popularity, do have the potential to increase the cost of non-Counterparty Bitcoin transactions. However, they also work to increase the value of the Bitcoin network as a whole—with miners receiving significant fees for them—especially given that Counterparty assets are able to be exchanged for bitcoins in a trustless manner on-chain. The Bitcoin protocol, being a global network with a capacity of only a few transactions per second, has never been and never will be a protocol for the transfer of small amounts of value, at least without Layer-2 protocols such as the Lightning Network. The technical challenge of making

Bitcoin scalable is thus entirely independent of the quantity of Counterparty or other L2 protocol messages.

The historical peak of this controversy occurred in 2014, during the so-called "`OP_RETURN` Wars".[5] At that time, the Bitcoin Core devs artificially limited the rules that the Bitcoin reference implementation used for relaying (rather than for mining) transactions, in an attempt to prevent Counterparty from storing anything more than 40 bytes in an `OP_RETURN` payload, since 40 bytes was sufficient for storing hashes.[6] Of course, the very fact that hashes were considered acceptable payloads, while semantic data were not, demonstrates both the hypocrisy and the ineffectiveness of such a ruleset—hashes are just data, and the effect that a data payload has on the Bitcoin network is obviously independent of its content. It is telling that this 40-byte limit was later raised to the arbitrary size of 80 bytes (its planned size before Counterparty was launched), and for no particular reason.[7] Today, it is even possible to store kilobytes of prunable data in individual Bitcoin transactions by employing the segregated witness scheme.[8]

Native Currency

Counterparty, being a general extension to the Bitcoin protocol, also provides its own native cryptocurrency—**XCP**. The XCP token is used in all cases where the Bitcoin token is not capable of acting as the working currency either as a denominational unit or for paying network fees: the Bitcoin protocol, not being aware of the Counterparty metaprotocol, cannot be used except as a minimal anti-SPAM mechanism based on the gross size of the Bitcoin transactions that contain Counterparty data. Nevertheless, there is a number of Counterparty transactions that do not require the use of the XCP token due to their simplicity, most notably straightforward asset transfers and issuances of numeric assets (Counterparty assets without a human-readable identifier), which create a low computational burden for Counterparty nodes.

The creation and initial distribution of XCP was designed to be as decentralized and trustless as possible, without there being any mining network to provide Sybil resistance. In early 2014, a few cryptocurrencies had already been launched through what would later be called "initial coin offerings" (ICOs), however such a launch necessarily requires trust in the creators of the protocol, who must distribute (read

[5] https://x.com/VitalikButerin/status/929804867568373760?s=20

[6] https://github.com/bitcoin/bitcoin/pull/3737

[7] https://github.com/bitcoin/bitcoin/pull/5286

[8] https://github.com/bitcoin/bitcoin/blob/fc06881f13495154c888a64a38c7d538baf00435/src/policy/policy.h#L46

"sell") the tokens in a centralized manner. **Proof-of-burn** provides an elegant alternative that mirrors the security model of Bitcoin mining: just as, with Bitcoin, energy is destroyed in the creation of bitcoins; with proof-of-burn, bitcoins are destroyed in the creation of XCP. The very energy used to create those bitcoins was repurposed to create XCP, and without the expenditure of any additional energy. Proof-of-burn had been theorized in 2013, but had never before been implemented;[9] it provides strong Sybil resistance without introducing any centralization.

Proof-of-burn involves the publication of a provably unspendable address (`1CounterpartyXXXXXXXXXXXXXXXUWLpVr`), the low entropy of which shows that there is no corresponding private key with which coins sent to that address could be spent. The Counterparty Core code detected transactions sending BTC to this address between January 2nd and February 3rd, 2014, automatically and trustlessly creating XCP as BTC were destroyed. Between 1,500 and 1,000 XCP were created for each 1 BTC (decreasing linearly) so as to provide an incentive for users to destroy BTC earlier in the burn period 2,130 BTC were destroyed in total (approximately US$2,000,000 at the time), leading to the generation of 2,648,755 XCP.

The creators of Counterparty had no special rights or privileges whatsoever in the creation of XCP, and they do not receive any fees for the use of the Counterparty protocol. This means that the Counterparty development has had to seek alternative sources of funding, like the ecosystem for Bitcoin. XCP is a deflationary currency: since the proof-of-burn period ended, no new XCP have been created. XCP are regularly destroyed by Counterparty users for the payment of Counterparty network fees; but because XCP is Counterparty-aware, they are able simply to be deleted and do not have to be burned by being sent to an unspendable address.

Future Development

As of the writing of this document, there are two primary limitations of the Counterparty protocol, both of which may be overcome with moderate additional development effort. These are:

1. Deeper Integration with BTC

Counterparty, as currently implemented, maintains a strong abstraction layer between the Bitcoin UTXO system and the Counterparty state machine. That is, Counterparty treats Bitcoin as a simple dis-

[9] https://en.bitcoin.it/w/index.php?title=Proof_of_burn&oldid=33833

tributed immutable log and largely ignores the semantic value of individual Bitcoin transaction outputs. As a consequence, trades between Counterparty-native tokens and BTC over the decentralized exchange are slow and expensive, requiring multiple block confirmations to settle a match.

By breaking this abstraction boundary and treating UTXOs as first-class objects in the protocol, Counterparty will be able to provide seamless integration with the Bitcoin token: Counterparty assets will be able to be attached directly to UTXOs, allowing for them to be held and transfered using standard Bitcoin wallet software. Perhaps more significantly, this upgrade will allow for trustless, atomic swaps between native Counterparty assets and BTC, such as Ordinals has. Indeed, it will be possible to trade Ordinals assets for Counterparty assets as well.

2. Implementation of a General-Purpose Virtual Machine

It is a core feature of the state-machine replication model that it supports *arbitrary deterministic computation*. Indeed, Counterparty demonstrated this by porting the entirety of the Ethereum Virtual Machine to the Bitcoin blockchain in 2014.[10] This functionality was never merged into mainline simply due to a lack of development resources necessary to maintain it.

Smart contract systems have been used in public blockchains for many years now, but they are still generally used for building only very small applications. The primary reasons for this are a combination of (a) poor language safety, (b) no modularity, (c) difficult syntax. The Counterparty smart contracts language will address these limitations of existing systems, so as to bring general-purpose computation to the Bitcoin blockchain, and without the use of a sidechain. Naturally, XCP will serve as the gas token for computation and storage using this virtual machine, and fees will be dynamic based on network load.

Comparison with other Layer-2 Protocols

Bitcoin Layer-2 protocols can be broadly categorized based on their fundamental level of integration with the Bitcoin blockchain:

- **Overlay Networks** are extensions to Bitcoin that add no value or functionality to that of Bitcoin, but which may provide significant improvements to performance for instance. The Lightning Network falls under this category.

[10] https://www.ccn.com/counterparty-brings-ethereum-smart-contracts-to-the-bitcoin-blockchain/

- **Colored Coins** are extensions to Bitcoin that are still reliant on the UTXO system and Script: these protocols add semantic value and functionality to the Bitcoin blockchain, but they are heavily limited by the simplicity and inflexibility of Script. The Ordinals protocol is an example of a colored-coins protocol.
- **Metaprotocols** are extensions to Bitcoin that implement their own state machines. Metaprotocols are able to add arbitrary functionality to Bitcoin. Counterparty was the first working instance of a metaprotocol.
- **Sidechains** are independent protocols and networks that simply allow for a two-way peg to the Bitcoin token, and they thus are not true "L2 protocols". Liquid, Rootstock and Stacks are all in essence sidechains.

As compared with overlay networks and colored coins, the Counterparty metaprotocol allows for dramatically more functionality, while simultaneously preserving the security model of the underlying blockchain and also allowing for a deep integration with Bitcoin. Sidechain protocols, because they have independent networks, must implement their own consensus system and then rely on a set of central entities that are also running Bitcoin nodes to validate all logic that is outside of the Bitcoin protocol itself.

A priori, the only significant architectural advantage of a protocol like Ordinals over Counterparty is in its inherent simplicity: Ordinals, for instance, is necessarily both very simple and very limited—it supports token creation and transfers, but nothing more. Of course, the continued operation of Counterparty over the past ten years has demonstrated the practicality of the metaprotocol model.

With the above features implemented, Counterparty will offer the best features of colored coins and sidechain protocols providing both the deep, native integration with Bitcoin that characterizes Ordinals and also the power and flexibility of altcoins and sidechains such as Rootstock and Stacks. Counterparty's technology allows for the creation of a truly trustless alternative to Ethereum on Bitcoin, and this effort will end with Counterparty reclaiming its status as a focal point of innovation and value-creation on Bitcoin in the near future.