Bitcoin Core & Bitcoin Cash Network Histories (2009 — 2024)

Legend
- HARD FORK
- SOFT FORK
- TECH RELEASE DATE

BitcoinCore Wallets

Bitcoin.com · muun · Wallet of Satoshi

Bitcoin Cash Wallets

Paytaca (Your money. Your control.) · Bitcoin.com · zapit · ELECTRON CASH · Cashonize (A Bitcoin Cash Wallet)

Bitcoin Core Technologies & Ecosystems

Bitcoin Cash Technologies & Ecosystems

2009-01-03 **Bitcoin**

2009-10-28 **Addition of nLockTime Enforcement**

2010-07-31 **Addition of OP_NOP Functions**

2010-08-01 **Separation of Evaluation of scriptSig and scriptPubKey**

2010-08-15 **Value Overflow Incident**

2010-10-12 **1MB Block Size Limit**

Notable commentary:

"The limit was originally Hal Finney's idea. Both Satoshi and I objected that it wouldn't scale at 1MB. Hal was concerned aboutz potential DoS attack though, and after discussion, Satoshi agreed... But all 3 of us agreed that 1MB had to be temporary because it would never scale." - Ray Dillinger

"The plan from the begining was to support huge blocks. The 1MB hard limit was always a temporary denial-of-service prevention measure." - Gavin Andresen

"It can be phased in, like:

if (blocknumber > 115000)
maxblocksize = largerlimit

- Satoshi Nakomoto

2012-03-15 **Disallow Transactions with Same TXID**

2012-04-01 **Pay-to-Script-Hash**

2013-03-24 **Block Height in Coinbase**

2013-05-15 **Migration from Berkeley DB to LevelDB**

2015-07-04 **Strict DER Encoding for Signatures**

2015-12-14 **OP_CHECKLOCKTIMEVERIFY**

2016-02-23 **Addition of Opt-In Replace-By-Fee ("RBF")**

2016-07-04 **OP_CHECKSEQUENCEVERIFY**

("BTC") **Bitcoin Core**
2017-08-24
User Activated Soft Fork
Segrigated Witness ("SegWit")

Segregated Witness ("SegWit") is an implemented protocol upgrade intended to provide protection from transaction malleability and increase block capacity. SegWit separates the witness from the list of inputs. The witness contains data required to check transaction validity but is not required to determine transaction effects. Additionally, a new weight parameter is defined, and blocks are allowed to have at most 4 million weight units (kU). Non-witness and pre-segwit witness bytes weigh 4 WU, but each byte of SegWit witness data only weighs 1 WU, allowing blocks that are larger than 1 MB without a hardforking change.

Because the new witness field contains Script versioning, it is also possible to make changes to or introduce new opcodes to SegWit scripts that would have originally required additional complexity to function without SegWit.

The SegWit upgrade accomplished the following:
- *Enabled Implementation of Lightning Network*
- *Stealth block size increase (some "block size" but more data allowed)*
- *Solved malleability*
- *Allowed for data posting with discount*

BitcoinCash ("BCH")
2017-08-01
User Activated Hard Fork
Block Size Limit to 8MB |
Removal of RBF

2017-11-13
Difficulty Adjustment Algorithm Change
cw-144 algorithm
With rapid price, and thereby hash, fluctuations, particularly in relation to BTC, an algorithm change was necessary to ensure more reliable blocktimes.

2018-05-15
Monolith
Block Size Limit to 32MB | Extra Opcodes (including OP_CAT)

OP_CAT, OP_SPLIT, OP_NUM2BIN, OP_CHECKDATASIG, OP_CHECKDATASIGVERIFY, OP_NOT, OP_AND, OP_OR, OP_XOR, OP_DIV, OP_MOD, OP_BIN2NUM

In 2010 and 2011 the discovery of serious bugs prompted the deactivation of many opcodes in the Bitcoin language. Rather than simply re-enable the opcodes, the functionality that they provide has been re-examined and in some cases the opcodes have been re-designed or new opcodes have been added to address specific issues.

2018-09-17
Emergency Fix for Inflation Bug
CVE-2018-17144 Inflation Bug

CVE-2018-17144 was a bug that could have allowed an attacker to spend the same BTC more than once.

An optimization included in Bitcoin Core 0.14.0 skipped a check on newly-received blocks that was thought to be redundant, making the node willing to partly process blocks that tried to spend the same input more than once. This would trigger some code meant to catch unexpected node operation and cause the node to deliberately crash. This allowed a denial-of-service attack that could be exploited by miners.

The code meant to catch unexpected operation (an assert) was rewritten for Bitcoin Core 0.15.0, unintentionally allowing the node to continue operation in the case of duplication inputs. This allowed a miner to spend the same BTC more than once.

These changes were ported to Bitcoin Cash as well.

*The vulnerability was discovered and disclosed by **Bitcoin Cash** developer Awemany on September 17th, 2018. Patches were available within hours, and a binary release for the current release branch within 36 hours. After several developers reviewing the patch were able to reverse engineer the bug and discover the inflation risk, a full disclosure was made by Bitcoin Core developers on September 20th, 2018.*

The bug was never exploited on the Bitcoin Core nor BitcoinCash mainnet and an overwhelming majority of full nodes subsequently updated to unaffected versions of Bitcoin Core / Bitcoin Cash.

After the full disclosure, the bug was exploited on Bitcoin Core testnet, causing any nodes still running Bitcoin Core 0.14.x to crash and nodes running 0.15.0 to 0.16.2 to accept a transaction that spent the same funds more than once. When testnet miners managed to produce a valid chain with more proof-of-work than the invalid chain, those non-upgraded nodes attempted to switch to the new chain but were unable to fully un-spend a double-spent input. This left the old testnet nodes stuck on a lower-proof-of-work block chain, requiring manual user intervention to fix them.

2018-11-15
Magnetic Anomaly
OP_CHECKDATASIG & OP_CHECKDATASIGVERIFY | Canonical Transaction Ordering ("CTOR")

OP_CHECKDATASIG and OP_CHECKDATASIGVERIFY check whether a signature is valid with respect to a message and a public key.

OP_CHECKDATASIG permits data to be imported into a script, and have its validity checked against some signing authority such as an "Oracle".

With the exception of the coinbase transaction, transactions within a block MUST be sorted in numerically ascending order of the transaction id, interpreted as 256-bit little-endian integers. The coinbase transaction MUST be the first transaction in a block.

Context:
All nodes already have all the transactions in the mempool. When a block is found, it is sending all transactions to all nodes a second time. This delays block propagation and block validation. In order to improve this, Compact/Xthinlet utilize shorter transaction IDs to minimize the data needing to be send a second time when a block is is found. Graphene takes this a step further by utilizing bloom filters to check which transactions are in which block. However, the bloom filters only allow us to know what transactions go into the block, not in what order. The ordering data makes up 81% of the data in a graphene block. By utilizing canonical ordering, 81% of the block data no longer needs to be sent, allowing for 7x faster block propagation and validation.

CTOR also enables better parallelization which enables far better scaling with technology. An additional benefit of faster block propagation and validation is the mitigation of selfish mining (where miners that find the first block have an advantage over others for the time it takes their nodes to receive and validate the recently found block).

2019-05-15
Great Wall
Schnorr Signatures | Segwit Recovery

Schnorr signatures have some slightly improved properties over the ECDSA signatures currently used in bitcoin:
- *Provable security assuming discrete log*
- *Proven resistance to all known third party malleability mechanisms*
- *Linearity allows some simple multi-party signature aggregation protocols. (compactness / privacy / malleability benefits)*
- *Possibility to do batch validation, resulting a slight speedup during validation of large transactions or initial block download*

2019-11-15
Graviton
Schnorr for Multisig | MINIMALDATAINSCRIPT

OP_CHECKMULTISIG/VERIFY upgraded to accept Schnorr signatures in a way that increases verification efficiency and is compatible with batch verification.

MINIMALDATA flag, which restricts malleability vectors, has been active at the mempool layer of most nodes but not at the consensus layer. The upgrade converts the existing MINIMALDATA rules to consensus.

2020-05-15
Phonon
OP_REVERSEBYTES | SigChecks

OP_REVERSEBYTES reverses the bytes of the top stackitem.

Since early days, Bitcoin has had a SigOps counting rule for limiting the amount of CPU usage possible in a given transaction or block, based on the principle that signature verifications are by far the most CPU-intense operations.

Although partly effective, there are well known issues with sigops, which mainly stem from the fact that SigOps are judged by parsing scripts, rather than executing them. Bitcoin splits scripts into two transactions (the scriptPubKey of the transaction that creates a coin, and the scriptSig of the transaction that spends it), yet the actual CPU work of verifying a transaction solely happens in the spending transaction, and this leads to some paradoxical situations: a transaction/block that contains high sigops might involve very little CPU work, and conversely a transaction with low sigops may require very high CPU work.

The essential idea of SigChecks is to perform counting solely in the spending transaction, and count actual executed signature check operations.

2020-11-15
Axion
Difficulty Adjustment Algorithm Change: ASERT (aserti3-2d DAA)

The DAA introduced in November 2017 exhibited susceptibility to a daily periodic difficulty oscillation stemming directly from the simple moving-average design of the algorithm. The periodic difficulty oscillations incentivized switch mining and disincentivized steady hashrate mining.

The oscillations in difficulty and hashrate have resulted in a daily pattern of long confirmation times, followed by bursts of rapid blocks. Average confirmation time of transactions are significantly increased as few transactions are included in the rapidly mined blocks.

ASERT does not exhibit the above-mentioned oscillations and has a range of other attractive qualities such as robustness against singularities without a need for additional rules, and absence of accumulation of rounding/approximation error.

2021-05-15
BigBlockIfTrue
Enabling Transactions with Multiple OP_RETURNs | Removal of Unconfirmed Transaction Chain Limit

Multiple OP_RETURNs enable developers to use OP_RETURN in a manner that justifies its core benefit - experimentation of new features/protocol updates without breaking existing systems or stressing the block chain with extremely UTXOs. This is still subject to the existing 223 byte limit.

When a transaction is first transmitted on the Bitcoin Cash network, it is considered "unconfirmed" until it is "mined" into a block. These transactions that are not yet mined are also referred to as "zero-conf" transactions. Transactions are dependent upon other transactions, such that they are chained together; the value allocated by one transaction is then spent by a subsequent transaction.

Until this upgrade, the Bitcoin Cash network only permitted transactions to be chained together 50 times before a block must include them. Transactions exceeding the 50th chain weresften ignored by the network, despite being valid. Once a transaction is submitted to the network, it cannot be revoked. This situation, when encountered, can be extremely difficult to remedy with today's available tools and simultaneously creates an unnecessary amount of complexity when being accounted for by network and applications developers.

This limit is now removed.

2021-11-14
Taproot
Schnorr Signatures | Taproot | Tapscript

Taproot was designed to improve privacy, scaling, and security in addition to improving smart contract efficiency.

Taproot allows payments to schnorr public keys that may optionally commit to a script that can be revealed at spend time.

Coins protected by taproot may be spent either by satisfying one of the committed scripts or by simply providing a signature that verifies against the public key following the script to be kept private). Taproot is intended for use with schnorr signatures that simplify multiparty construction (e.g. using MuSig) and with MAST to allow committing to more than one script, any one of which may be used at spend time.

Technologies stemming from Taproot include:
- *bitvm*
- *Inscriptions*
- *BRC-20 tokens*
- *Ordinals*

2022-05-15
U8
Addition of Introspection Op-codes | Bigger Integers & OP_MUL Re-enabling

Since the introduction of OP_CHECKDATASIG, it has been possible to implement practical Bitcoin Cash covenants – contracts which validate properties of the transaction in which the contract funds are spent.

Covenants enable a wide range of new innovation, but the strategy by which they were implemented was extremely inefficient; most of a transaction's content must be duplicated for each input which utilizes a covenant. In practice, this doubles or triples the size of even the simplest covenant transactions, and advanced covenant applications quickly exceed 1MB limits. This severely limited the extent of BCH covenant development and usage.

This upgrade adds a set of new virtual machine (VM) operations which enable BCH contracts to efficiently access details about the current transaction like output values, recipients, and more – without increasing transaction validation costs.

BCH virtual machine (VM) math operations were limited to signed 32-bit integers, preventing contracts from operating on values larger than 2147483647. Workarounds which allow contracts to emulate higher-precision math are often impractical, difficult to secure, and significantly increase transaction sizes.

This unusually-low limit on arithmetic operations has been present since the earliest Bitcoin release to avoid standardizing a strategy for overflow handling (though 64-bit math was always used internally). Since the development of covenants, this unusual overflow-handling strategy has become a source of contract vulnerabilities and a barrier to real-world applications. Few remaining computing environments operate using 32-bit integers, and this limit has never been relevant to work-case transaction-validation performance.

This upgrade expands the integer range allowed in BCH contracts (from 32-bit to 64-bit numbers) and re-enables the multiplication opcode.

2023-05-15
CashTokens
P2SH32 | Restrict Transaction Version | Minimum Transaction Size | Native Layer 1, UTXO, Miner-validated Tokens

Increases cryptographic security of existing P2SH feature by enabling a 32-byte variant of it.

Tightens enforcement of allowed transaction version field values by making it a consensus rule instead of the network relay rule that is currently in force.

Making allowed values part of consensus specification enables its use in tracking future consensus specification upgrades. This would make it easier to roll-out future hard fork upgrades in a non-breaking and backwards-compatible way for non-node software.

Reduces the current consensus minimum transaction size of 100 bytes down to a minimum size of 61 bytes. This more relaxed 63-byte minimum size, like the previous 100-byte minimum size, applies to both coinbase and non-coinbase transactions uniformly.

BitcoinCash contracts lacked primitives for issuing messages that can be verified by other contracts, preventing the development of decentralized application ecosystems on BitcoinCash. This upgrade introduced Byte-String Commitments and Numeric Commitments.

Contract-issued Commitments
In the context of the Bitcoin Cash virtual machine (VM), a commitment can be defined as an irrevocable message that was provably issued by a particular entity. Two forms of commitments were already available to Bitcoin Cash contracts:
- *Transaction signatures – a commitment made by a private key attesting to the signing serialization of a transaction.*
- *Data signatures – a commitment made by a private key attesting to the hash of an arbitrary message (introduced in 2018 by OP_CHECKDATASIG).*

Each of these commitment types require the presence of a trusted private key. Because contracts cannot themselves hold a private key, any use case that requires a contract to issue a verifiable message must necessarily rely on trusted entities to provide signatures. This limitation prevented Bitcoin Cash contracts from offering or using decentralized oracles – multiparty consensus systems that produce verifiable messages upon which other contracts can act.

By providing a commitment primitive that can be used directly by contracts, the BitcoinCash contract system can support advanced, decentralized applications without increasing transaction or block validation costs.

Byte-String Commitments ("Non-Fungible Tokens" or "NFTs")
The most general type of contract-issued commitment is a simple string of bytes. This type can be used to commit to any type of contract state: certifications of ownership, authorizations, credit, debt, contract-internal time or epoch, vote counts, receipts (e.g. to support refunds or future redemption), etc. Identities may commit to state within a hash structure (e.g. a merkle tree), or – to reduce transaction sizes – as a raw byte string (e.g. public keys, static numbers, boolean values).

Numeric Commitments ("Fungible Tokens" or "FTs")
The BitcoinCash virtual machine (VM) supports two primary data types in VM bytecode evaluation: byte strings and numbers. Given the existence of a primitive allowing contracts to commit to byte strings, another commitment primitive can be inferred: numeric commitments.

Numeric commitments are a specialization of byte-string commitments – they are commitments with numeric values which can be divided and merged in the same way as the Bitcoin Cash currency. With numeric commitments, contracts can efficiently represent fractional parts of abstract concepts – shares, pegged assets, bonds, loans, options, tickets, loyalty points, voting outcomes, etc.

While many use cases for numeric commitments can be emulated with only byte-string commitments, a numeric primitive enables many contracts to reduce or offload state management altogether (e.g. shareholder voting), simplifying contract audits and reducing transaction sizes.

2024-05-15
ABLA
Adaptive Blocksize Limit Algorithm

Needing to coordinate manual increases to BitcoinCash's blocksize limit incurs a meta cost on all network participants. The need to regularly come to agreement makes the network vulnerable to social attacks, as had occured on Bitcoin Core from its early history.

To reduce Bitcoin Cash's social attack surface and save on meta costs for all network participants, ABLA automatically adjusts the blocksize limit after each block, based on the exponentially weighted moving average size of previous blocks.

The algorithm preserves the existing 32 MB limit as the floor "stand-by" value, and any increase by the algorithm can be thought of as a bonus on top of that, sustained by actual transaction load.

January 2018
Rootstock (RSK)

Rootstock is a Bitcoin Core side-chain protocol designed to bring Turing complete smart contracts to the Bitcoin Core network. RSK is a separate blockchain secured through Bitcoin Core miners who participate in merge mining and use a two-way peer peg ("Powpeg") to bridge BTC from the main chain to its network. RSK is compatible with Ethereum Virtual Machine (EVM) smart contracts, enabling developers to deploy existing ETH smart contracts on RSK with minimal changes. The native currency of the RSK network is Smart Bitcoin (RBTC), which is pegged 1:1 with BTC.

March 2018
Lightning Network

Lightning Network launched as a means to scaling the transaction throughput of Bitcoin Core off-chain, thereby limiting the need for larger Block sizes on the main network.

The basic structure has users opening Lightning Nodes that can maintain payment channels. Users can broadcast a transaction on the main Bitcoin Core chain to open a channel with a BTC liquidity between the two parties. Parties can expand or close this liquidity at any time by broadcasting another on-chain transaction. As long as some route exists between a sending user and the final receiving user, a payment can go through.

The benefits are that users do not need to transact consistently on chain, only enough to open the necessary channels with the necessary liquidity with the parties they will frequently transact with, thereby eliminating the need to wait for confirmations or pay high fees. Users can further optimize their usage of Lightning by waiting to open/close channels for times with lower fees on the main chain.

The drawbacks are that usage of the main chain is still required, and on occasion users may face routing issues if they do not have the necessary connections. Users must also create "invoices," which are specific to receive a specific amount. This makes tipping challenging if tips are not all of the same amount. Custodial Lightning accounts (such as Wo$ and Strike) are that they are able to transfer any amount without direct use of an invoice. Another drawback is that a user's Lightning Node must remain online at all times or otherwise face a forced channel closure(s) and payment of on-chain fees. The primary instigator of this has been to use services such as Phaxnix Wallet and Wallet of Satoshi/Strike (as described below).

Today many applications exist. Applications like Phaxnix Wallet enable largely non-custodial nodes to be run by users, while others such as Wallet of Satoshi & Strike offer custodial lightning accounts aimed at simpler setup and more successful payment routing through larger hubs/internal ledger changes. Additionally, a custodial nature allows systems to be more compliant with local laws and regulations.

October 2018
Stacks (STX)

Stacks is a Bitcoin Core Layer-2 solution enabling smart contracts and decentralized applications ("dapps") to utilize BTC as a base layer. Stacks uses the Proof of Transfer (PoX) consensus mechanism.

March 2020
Nostr

Nostr: "Notes and Other Stuff Transmitted by Relays"

Nostr is a protocol, designed for simplicity, that aims to create a censorship-resistant global social network.

While not directly built on top of Bitcoin Core, nostr uses BTC as its native currency. Users can tip others (called "Zaps") BTC through Lightning Network invoices.

January 2023
Ordinals

Ordinals are NFTs on Bitcoin Core. Unlike earlier attempts at BTC NFTs, which existed on layer-2 blockchains, Ordinals are BTC native. They're created by attaching data, such as images, videos, and GIFs, to an individual satoshi on the base BTC blockchain (cost effective as a result of the SegWit upgrade). Due to this novel method, there has been an explosion of creative development on BTC, leading to a robust ecosystem of NFTs, apps, and more.

2017
ElectronCash

Robust, fully-featured Bitcoin Cash SPV wallet. Tooling includes support for tools such as CashFusion, Flipstarter, CashTokens, and more.

2018
Simple Ledger Protocol ("SLP")

SLP is a simple token protocol built on top of Bitcoin Cash. It enables the creation of both fixed-supply and multiple issuance tokens with a variety of options. Issuer-controlled whitelists can also be enabled particularly useful for regulated securities. Bitcoin Script can also be used with SLP tokens.

SLP runs through the use of OP_RETURNs - where an OP_RETURN is appended to a dust transaction containing the token transaction details.

April 2018
Memo.Cash

On-chain social network built on Bitcoin Cash.

Memo is both a protocol and a front-end application. The protocol works by writing transactions to the Bitcoin Cash blockchain. Client applications can read these transactions to see the activity on the network.

With current social networks, users do not control their data and can have their accounts removed without reason. The blockchain is more than an uncensorable way to send money, it's an uncensorable way to store data. Memo is an experiment leveraging the blockchain's data storing capabilities.

Each action is stored on the blockchain in an OP_RETURN transaction - whether it's posting a memo, following someone, or liking a post. You can also include a tip (it is on-chain after all).

June 2018
Libauth

Libauth is an ultra-lightweight, zero-dependency JavaScript library for Bitcoin Cash, Bitcoin Core, and bitauth applications.

December 2018
AnyHedge

A Bitcoin Cash futures contract that aims to mitigate volatility through trading of risk in a peer-to-peer, non-custodial, blockchain enforced, fully collateralized way. With trust reduced to a blind oracle, the futures contract offers a unique set of advantages and disadvantages over existing stability solutions. AnyHedge is transactional in nature and does not require a central point of failure. The user experience is different from traditional fiat or stablecoin based solutions. As liquidity increases, it enables a variety of automated, behind-the-scenes use cases aimed at taming volatility. Compared to alternative solutions, AnyHedge has a unique set of difficulties and respective mitigations.

February 2019
Bitauth IDE

An online, integrated development environment for Bitcoin Cash contracts.

July 2019
CashScript

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

November 2019
CashFusion

CashFusion is a fully decentralized privacy protocol that allows anyone to create multi-party transactions with other network participants. This process obscures your real spending and makes it difficult for chain-analysis companies to track your coins.

CashFusion is different than coin mixing. "Coin mixing" commonly refers to the use of services that allow a user to replace his or her coins with a different set of coins of the same exact amount, which significantly inhibits privacy. CashFusion is different. It allows users to combine their transactions with others, of different amounts, creating obfuscation.
Coin mixing has other disadvantages too. A trusted service is required, meaning the user must trust the service and usually has to pay a fee. However, with CashFusion, there is no risk of lost funds and no service fees.

Tor is integrated into the CashFusion protocol. It hides the user IP address from the server, thus assuring that the server cannot spy on users by linking their transactions components based on IP.

November 2019
Chaingraph

Chaingraph is a multi-node blockchain indexer and GraphQL API with support for efficient, nested queries and powerful blockchain analysis.

January 2020
Fulcrum

A fast & nimble SPV server for Bitcoin Cash, Bitcoin Core, and Litecoin.

- *Fast: Written in 100% modern C++20 using multi-threaded and asynchronous programming techniques.*
- *A drop-in replacement for Electrum/Electrumx: Fulcrum is 100% protocol-level compatible with the Electrum Cash 1.5.3 protocol. Existing server admins should feel right at home with this software since installation and management of it is nearly identical to Electrumx/Electrumx server.*
- *Cross-platform: While this codebase was mainly developed and tested on MacOS, Windows and Linux, it should theoretically work on any modern OS (such as *BSD) that has Qt5 Core and Qt5 Networking available.*

March 2020
Flipstarter

A crowdfunding platform built on Bitcoin Cash. The tool is fully decentralized , non-custodial, and uncensorable enabling anyone to create a campaign (be it a technical organization, social media influencer, or charity without the need of a centralized service such as Kickstarter or GoFundme which have the ability to censor projects.

August 2020
Electrum-Cash

Electrum-Cash is a set of lightweight JavaScript libraries that lets you work with one or more Electrum servers.

July 2021
smartBCH

Smart Bitcoin Cash ("smartBCH") is a sidechain for Bitcoin Cash that is compatible with Ethereum's EVM and Web3 API (enabling EVM developers to bridge projects with minimal changes) and provides high throughput for DApps in a fast, secure, and decentralized manner.

Today smartBCH is run independent from the main chain. However, CashTokens (introduced May-2023), has brought the tools and capabilities of smartBCH on the main chain as UTXOs which offer further scaling and processing benefits, and many smartBCH projects have ported over.

April 2022
Rostrum

Rostrum is an efficient implementation of Electrum Server with token support and can be used as a drop-in replacement for Electrumx. In addition to the TCP RPC interface, it also provides WebSocket support.

October 2022
BCHBull

Permissionless, on-chain, non-custodial leverage trading and hedging with Bitcoin Cash.

BCH BULL is a unique web application that allows users to speculate with leverage, or hedge with Bitcoin Cash against a number of other metals, fiat, and crypto assets, all done entirely using smart contracts on the Bitcoin Cash blockchain. The application is exceptional as it is one of the few decentralized and non-custodial applications in operation on a layer 1 UTXO blockchain, thus placing it at the forefront of a new frontier in cryptocurrency technology.

BCH BULL utilises AnyHedge, a permissionless, non-custodial, and on-chain protocol created by General Protocols.

May 2023
TapSwap

The first decentralized marketplace to trade NFTs on Bitcoin Cash utilizing the CashTokens protocol.

July 2023
Cauldron DEX

The Cauldron Protocol is an open-source smart contract for providing liquidity and trading CashTokens on the Bitcoin Cash network. Cauldron operates as a non-custodial automated liquidity protocol, emphasizing openness, transparency, and user accessibility.

Cauldron is a novel implementation of Constant Product Market Maker (CPMM) contracts on Bitcoin Cash using a micro-pools approach. By harnessing the inherent properties of Bitcoin Cash, particularly the UTXO model's ability to hold tokens and BCH simultaneously, Cauldron enables decentralized liquidity pools tailored for the Bitcoin Cash ecosystem. The micro-pools approach circumvents the need for a global state by allowing individual users to operate their own CPMM contracts, offering unparalleled scalability. Through micro-pools, Cauldron efficiently facilitates decentralized trading, which is particularly critical for a blockchain network that is actively accommodating an expanding suite of tokens.

December 2023
CashTokens Studio

A tool that enables users to create both fungible and non-fungible tokens on the Bitcoin Cash blockchain using the CashTokens protocol.

January 2024
BCH Gurus

BCH Gurus launched the first wave of its NFT collection (built on CashTokens) in November 2023. January 2024 BCH Gurus launched their decentralized price prediction app (on-chain (BCH) utilizing oracles) for betting on price of different in addition to the FURU fungible token which acts as the utility token for the app.

Additions are in the works to enable full prediction markets on-chain where users can bet on the outcome on anything (for which an oracle exists).

July 2024
FundMe.Cash

A website for interacting with decentralized crowdfunding contracts on the BitcoinCash blockchain.

CashStarter is the smart contract running on the Bitcoin Cash blockchain. This contract is what handles the BCH and manages the campaign.

July 2024
Nano-GPT

A website with a pay-as-you-go system for different language and image generation models. Users can deposit BCH which funds the use of these models. ChatGPT 4o, DALL-E 3, Stable Diffusion, CrystalClear, DreamShaper, and more are available.

August 2024
BCHC

BCHC is a fast, reliable, and extensible indexer built for BCH app developers.

BitcoinCore · btcd · Bitcoin Knots

BITCOINCASH node · bitcoinunlimited · BITCOIN VERDE · kth · BCHD

Bitcoin Core & BitcoinCash Network Histories: v4.7

X: @_minisatoshi
Telegram: @mini_satoshi