CALYX CONTAINERS

Crypto Currency Acceptance Policy

Table of Contents

Introduction of crypto

What cryptocurrency does Calyx Containers accept?
For now, Calyx Containers only accepts Bitcoin — not Bitcoin fork products (like Bitcoin Cash and Bitcoin SV) or any other types of digital assets. (Our wallet will not receive or even detect any other digital assets.) Please make sure you only send us Bitcoin because any other digital asset sent might end up lost or destroyed (and we're not responsible if that happens).

Process of accepting Crypto

From time to time, we may, in our discretion, impose limits on the amount of Crypto Assets that can be accepted as sales consideration, which may include limits on the cash value or number of transactions in which you can

engage over particular periods of time. Standard transaction limits are as follows:

- Weekly purchase limit: $25,000
- Rolling annual purchase limit: $500,000

We may change the above limits for safety, security or other lawful reasons. This limit will be updated in our ecommerce and invoicing software. Once these limits are reached, Crypto assets will not be accepted. Any exception to this need to be approved by CEO and CFO jointly

Restricted Activities

Customer must not engage in the following "Restricted Activities":

- Breach these Cryptocurrency Terms;
- Use what we reasonably believe to be fraudulent funds in order to buy Crypto Assets;
- Initiate any transaction that is not intended to be completed, or is intended to abuse, manipulate, mislead or default other participants in the Crypto Asset market;
- Engage in any activity that operates to defraud Calyx, other Calyx customers, or any other person; or
- Control an account that is linked to another account that has engaged in any of these restricted activities

Calyx does not accept BitPay for the following types of purchases*:

- Gift Cards
- Subscription orders

- Pre-orders
- Return shipping labels

Architecture and apps for accepting Crypto

How do I buy something from Calyx Containers with Bitcoin?
First, you need what is called a “Bitcoin wallet” — a device, platform, app, or software that supports Bitcoin transfers. Calyx Containers will give you our Bitcoin wallet “address” in both an alphanumeric code and QR code for you to enter into your Bitcoin wallet so it knows where to transfer the Bitcoin. It’s your responsibility to make sure that you send the Bitcoin to Calyx Containers’s Bitcoin wallet accurately.

Invoicing

Items listed on an invoice are priced in fiat currency and the total is converted to cryptocurrency. Users have to select a preferred crypto for each payment.

The invoices are paid only by Bitpay. When an invoice is sent, both sides receive email notifications. Once payment is made via BitPay, the confirmation of payment need to be sent to both the parties.

Calyx will not accept any Crypto currency outside official Bitpay account. Calyx will also not accept the crypto currency not listed in Bitpay. Any exception to this need to be approved by CFO/CEO

Payment of Crypto

At the time of purchase, the US Dollars amounts from Calyx Containers are converted to Bitcoin using the exchange rate provided by BitPay.

Customers are required to review the amount of Bitcoin needed and use your Bitpay wallet to complete the transaction within 15 minutes. If an order

is not settled within that time, the order will expire and the checkout process will have to restart.

Here are the three ways to pay with Bitcoin:

- On your PC: select “Open in Wallet” and pay from your Bitpay wallet on the same device.
- On your phone: scan the QR code on the page to pay from your mobile wallet app.
- If your wallet is on another device, you can copy the receiving address and Bitcoin amount to the wallet on your device or the web and then make your purchase.

Once a Bitcoin transaction is submitted to the Bitcoin network, it will be unconfirmed for a period of time (usually about 30 minutes, but sometimes longer) pending full verification of the transaction by the Bitcoin network. A transaction is not complete until it is fully verified.

Bitcoin payments exceeding $10,000 US Dollars will require the customer to complete IRS Form 8300 -Report of Cash Payments Over $10,000.

Payment Discrepancies

If a customer overpays the specified number of bitcoins in their order, the overpayment amount will be processed pursuant to our cancellation policy. If the customer underpays the specified number of bitcoins, the order will be cancelled and Calyx Containers customer service will attempt to contact the customer to return any funds received pursuant to our cancellation policy

Return and refund of crypto

BitPay as your payment method, once clicked the “Pay with BitPay” button, customer will have only 15 minutes to complete your payment. If customer is unable to complete the payment, they will have two options: they can try

again later to place a new order or they can change the payment method fiat currency.

All orders fully paid by BitPay are final and cannot be returned for BitPay or hard currency.

Prevention of loss/limit of loss

Sometimes we'll suspend use of our crypto service so that we can make technical changes, add new features (such as new cryptocurrencies), make sure it runs smoothly or improve its security. We'll also try to limit any suspension so it lasts as short a period as possible.

We won't be responsible to customer for losses that arise:

- if our crypto service isn't available;
- if we don't meet our obligations under these terms and conditions because of a legal or regulatory requirement; or
- because there were unusual or unexpected events outside our control.

We will only be responsible for foreseeable losses

If we don't meet our obligations under these terms and conditions, we will not be responsible for any loss that we couldn't have thought customer would suffer at the time of transaction.

Accounting of Crypto

Calyx will treat Crypto as an intangible asset in it's book.

Intangible assets IAS 38 Intangible Assets defines an intangible asset as "an identifiable non-monetary asset without physical substance." A cryptocurrency has no physical attributes and will be in the scope of IAS 38, unless it is being held for sale in the ordinary course of business (see

Cryptocurrency held for sale). Initial measurement IAS 38 requires that an intangible asset be measured initially at cost. When an entity pays cash, or an equivalent, to acquire cryptocurrency the measurement of cost is straightforward. However, often the currency is received in exchange for goods or services or another cryptocurrency. When an entity accepts cryptocurrency in exchange for goods or services, the entity will need to assess the requirements in the relevant Standard. For example, when a retailer accepts cryptocurrency as payment, it is likely that the retailer will have made a sale in accordance with IFRS 15 Revenue from Contracts with Customers. IFRS 15 states that when a customer promises consideration in a form other than cash, an entity measures the non-cash consideration (i.e. the cryptocurrency) at fair value. If an entity cannot reasonably estimate the fair value of the non-cash consideration, the consideration is measured indirectly by reference to the stand-alone selling price of the goods or services delivered to the customer. Subsequent measurement IAS 38 has two models for the subsequent measurement of intangible assets—the Cost model and the Revaluation model. Neither model allows a cryptocurrency to be measured at fair value through profit or loss. Cost model When the cost model is applied, the cryptocurrency is carried at cost, less any accumulated impairment losses. Because a cryptocurrency is an indefinite-life intangible asset it would not be amortized. Impairment is assessed by comparing the carrying amount with its recoverable amount. Recoverable amount is the higher of the asset's fair value less costs of disposal and its value in use. A cryptocurrency such as Bitcoin has no 'use' other than as a medium of exchange. Hence, the impairment assessment will involve comparing the carrying amount with fair value less costs of disposal. That assessment must be made whenever there is an indication that it may be impaired, and at least annually. The practical implication is that if the fair value of a cryptocurrency is below it's carrying amount at a reporting date, an impairment loss would be recognized in profit or loss. IAS 38 requires the disclosure of the gross carrying amount of the cryptocurrency held at the end of the reporting period, along with any accumulated impairment losses

Hence, Calyx in line with IAS 38 along with IFRS 15 guidance

1. Book the intangible at the sales price (USD equivalent of sales consideration – Fees)
2. Inventory is reduced by the quantity of sales
3. Monthly Crypto currency balance is valued based on month end value as per Bitpay rates reduced by the fees to sell such crypto. Any gain/loss is booked to P&L at the other income line

GLOSSARY

51 percent attack
Majority attack
The ability of someone controlling a majority of network hash rate to revise transaction history and prevent new transactions from confirming.

Address
A 20-byte hash formatted using base58check to produce either a P2PKH or P2SH Bitcoin address. Currently the most common way users exchange payment information.
Not to be confused with: IP address

Base58check
The method used in Bitcoin for converting 160-bit hashes into P2PKH and P2SH addresses. Also used in other parts of Bitcoin, such as encoding private keys for backup in WIP format. Not the same as other base58 implementations.
Not to be confused with: P2PKH address, P2SH address, IP address

Block

One or more transactions prefaced by a block header and protected by proof of work. Blocks are the data stored on the block chain.

Block chain

Best block chain

A chain of blocks with each block referencing the block that preceded it. The most-difficult-to-recreate chain is the best block chain.

Not to be confused with: Header chain

Block header

Header

An 80-byte header belonging to a single block which is hashed repeatedly to create proof of work.

Height

Block height

The number of blocks preceding a particular block on a block chain. For example, the genesis block has a height of zero because zero block preceded it.

Block reward

The amount that miners may claim as a reward for creating a block. Equal to the sum of the block subsidy (newly available satoshis) plus the transactions fees paid by transactions included in the block.

Not to be confused with: Block subsidy, Transaction fees

Maximum Block Size

The maximum size of a block according to the consensus rules. The current block size limit is 4 million weight units (1 million vbytes).

Not to be confused with: Block, Blockchain, Blockchain size

Blocks-first sync

Synchronizing the block chain by downloading each block from a peer and then validating it.

Not to be confused with: Headers-first sync

Bloom filter

A filter used primarily by SPV clients to request only matching transactions and merkle blocks from full nodes.

Not to be confused with: Bloom filter (general computer science term, of which Bitcoin's bloom filters are a specific implementation)

Chain code

In HD wallets, 256 bits of entropy added to the public and private keys to help them generate secure child keys; the master chain code is usually derived from a seed along with the master private key

Change address

Change output

An output in a transaction which returns satoshis to the spender, thus preventing too much of the input value from going to transaction fees.
Not to be confused with: Address reuse

Child key

Child public key

Child private key

In HD wallets, a key derived from a parent key. The key can be either a private key or a public key, and the key derivation may also require a chain code.
Not to be confused with: Public key (derived from a private key, not a parent key)

Coinbase

A special field used as the sole input for coinbase transactions. The coinbase allows claiming the block reward and provides up to 100 bytes for arbitrary data.
Not to be confused with: Coinbase transaction, Coinbase.com

Coinbase transaction

Generation transaction

The first transaction in a block. Always created by a miner, it includes a single coinbase.
Not to be confused with: Coinbase (the unique part of a coinbase transaction)

CompactSize

A type of variable-length integer commonly used in the Bitcoin P2P protocol and Bitcoin serialized data structures.
Not to be confused with: VarInt (a data type Bitcoin Core uses for local data storage), Compact (the data type used for nBits in the block header)

Compressed public key

An ECDSA public key that is 33 bytes long rather than the 65 bytes of an uncompressed public key.

Confirmation score

Confirmations

Confirmed transaction

Unconfirmed transaction

A score indicating the number of blocks on the best block chain that would need to be modified to remove or modify a particular transaction. A confirmed transaction has a confirmation score of one or higher.

Consensus

When several nodes (usually most nodes on the network) all have the same blocks in their locally-validated best block chain.

Not to be confused with: Social consensus (often used in discussion among developers to indicate that most people agree with a particular plan), Consensus rules (the rules that allow nodes to maintain consensus)

Consensus rules

The block validation rules that full nodes follow to stay in consensus with other nodes.

Not to be confused with: Consensus (what happens when nodes follow the same consensus rules)

Child pays for parent

CPFP

Ancestor mining

Selecting transactions for mining not just based on their fees but also based on the fees of their ancestors (parents) and descendants (children).

Not to be confused with: Replace by Fee, RBF

Denomination

Bitcoins

Satoshis

Denominations of Bitcoin value, usually measured in fractions of a bitcoin but sometimes measured in multiples of a satoshi. One bitcoin equals 100,000,000 satoshis.

Not to be confused with: Binary bits, a unit of data with two possible values

Difficulty

Network difficulty

How difficult it is to find a block relative to the difficulty of finding the easiest possible block. The easiest possible block has a proof-of-work difficulty of 1.

Not to be confused with: Target threshold (the value from which difficulty is calculated)

DNS seed

A DNS server which returns IP addresses of full nodes on the Bitcoin network to assist in peer discovery.

Not to be confused with: HD wallet seeds

Double spend

A transaction that uses the same input as an already broadcast transaction. The attempt of duplication, deceit, or conversion, will be adjudicated when only one of the transactions is recorded in the blockchain.

Escrow contract

A transaction in which a spender and receiver place funds in a 2-of-2 (or other m-of-n) multisig output so that neither can spend the funds until they're both satisfied with some external outcome.

Extended key

Public extended key

Private extended key

In the context of HD wallets, a public key or private key extended with the chain code to allow them to derive child keys.

Fork

When two or more blocks have the same block height, forking the block chain. Typically occurs when two or more miners find blocks at nearly the same time. Can also happen as part of an attack.

Not to be confused with: Hard fork (a change in consensus rules that breaks security for nodes that don't upgrade), Soft fork (a change in consensus rules that weakens security for nodes that don't upgrade), Software fork (when one or more developers permanently develops a codebase separately from other developers), Git fork (when one or more developers temporarily develops a codebase separately from other developers)

Genesis block

Block 0

The first block in the Bitcoin block chain.

Not to be confused with: Generation transaction (the first transaction in a block)

Hard fork

A permanent divergence in the block chain, commonly occurs when non-upgraded nodes can't validate blocks created by upgraded nodes that follow newer consensus rules.

Not to be confused with: Fork (a regular fork where all nodes follow the same consensus rules, so the fork is resolved once one chain has more proof of work than another), Soft fork (a temporary divergence in the block chain caused by non-upgraded nodes not following new consensus rules), Software fork (when one or more developers permanently develops a codebase separately from other developers), Git fork (when one or more developers temporarily develops a codebase separately from other developers

Hardened extended key

A variation on HD wallet extended keys where only the hardened extended private key can derive child keys. This prevents compromise of the chain code plus any private key from putting the whole wallet at risk.

HD protocol

HD wallet

The Hierarchical Deterministic (HD) key creation and transfer protocol (BIP32), which allows creating child keys from parent keys in a hierarchy. Wallets using the HD protocol are called HD wallets.

HD wallet seed

Root seed

A potentially-short value used as a seed to generate the master private key and master chain code for an HD wallet.
Not to be confused with: Mnemonic code / mnemonic seed (a binary root seed formatted as words to make it easier for humans to transcribe and possibly remember)

Header chain

Best header chain

A chain of block headers with each header linking to the header that preceded it; the most-difficult-to-recreate chain is the best header chain
Not to be confused with: Block chain

Headers-first sync

Synchronizing the block chain by downloading block headers before downloading the full blocks.
Not to be confused with: Blocks-first sync (Downloading entire blocks immediately without first getting their headers)

High-priority transaction

Free transaction

Transactions that don't have to pay a transaction fee because their inputs have been idle long enough to accumulated large amounts of priority. Note: miners choose whether to accept free transactions.

Initial block download

IBD

The process used by a new node (or long-offline node) to download a large number of blocks to catch up to the tip of the best block chain.
Not to be confused with: Blocks-first sync (syncing includes getting any amount of blocks; IBD is only used for large numbers of blocks)

Input

TxIn

An input in a transaction which contains three fields: an outpoint, a signature script, and a sequence number. The outpoint references a previous output and the signature script allows spending it.

Internal byte order

The standard order in which hash digests are displayed as strings—the same format used in serialized blocks and transactions.
Not to be confused with: RPC byte order (where the byte order is reversed)

Inventory

A data type identifier and a hash; used to identify transactions and blocks available for download through the Bitcoin P2P network.
Not to be confused with: Inv message (one of the P2P messages that transmits inventories)

Locktime

nLockTime

Part of a transaction which indicates the earliest time or earliest block when that transaction may be added to the block chain.

Mainnet

The original and main network for Bitcoin transactions, where satoshis have real economic value.

Not to be confused with: Testnet (an open network very similar to mainnet where satoshis have no value), Regtest (a private testing node similar to testnet)

Transaction malleability

Transaction mutability

The ability of someone to change (mutate) unconfirmed transactions without making them invalid, which changes the transaction's txid, making child transactions invalid.

Not to be confused with: BIP62 (a proposal for an optional new transaction version that reduces the set of known mutations for common transactions)

Miner-activated soft fork

MASF

A Soft Fork activated by through miner signalling.

Not to be confused with: User Activated Soft Fork (a soft fork activated by flag day or node enforcement instead of miner signalling.), Fork (a regular fork where all nodes follow the same consensus rules, so the fork is resolved once one chain has more proof of work than another), Hard fork (a permanent divergence in the block chain caused by non-upgraded nodes not following new consensus rules), Soft fork (a temporary divergence in the block chain caused by non-upgraded nodes not following new consensus rules), Software fork (when one or more developers permanently develops a codebase separately from other developers), Git fork (when one or more developers temporarily develops a codebase separately from other developers

Master chain code

Master private key

In HD wallets, the master chain code and master private key are the two pieces of data derived from the root seed.

Merkle block

A partial merkle tree connecting transactions matching a bloom filter to the merkle root of a block.

Not to be confused with: MerkleBlock message (a P2P protocol message that transmits a merkle block)

Merkle root

The root node of a merkle tree, a descendant of all the hashed pairs in the tree. Block headers must include a valid merkle root descended from all transactions in that block.
Not to be confused with: Merkle tree (the tree of which the merkle root is the root node), Merkle block (a partial merkle branch connecting the root to one or more leaves [transactions])

Merkle tree

A tree constructed by hashing paired data (the leaves), then pairing and hashing the results until a single hash remains, the merkle root. In Bitcoin, the leaves are almost always transactions from a single block.
Not to be confused with: Partial merkle branch (a branch connecting one or more leaves to the root), Merkle block (a partial merkle branch connecting one or more transactions from a single block to the block merkle root)

Message header

The four header fields prefixed to all messages on the Bitcoin P2P network.

Minimum relay fee
Relay fee

The minimum transaction fee a transaction must pay (if it isn't a high-priority transaction) for a full node to relay that transaction to other nodes. There is no one minimum relay fee—each node chooses its own policy.
Not to be confused with: Transaction fee (the minimum relay fee is a policy setting that filters out transactions with too-low transaction fees)

Mining
Miner

Mining is the act of creating valid Bitcoin blocks, which requires demonstrating proof of work, and miners are devices that mine or people who own those devices.

Multisig
Bare multisig

A pubkey script that provides N number of pubkeys and requires the corresponding signature script provide M minimum number signatures corresponding to the provided pubkeys.
Not to be confused with: P2SH multisig (a multisig script contained inside P2SH), Advanced scripts that require multiple signatures without using OP_CHECKMULTISIG or OP_CHECKMULTISIGVERIFY

nBits
Target

The target is the threshold below which a block header hash must be in order for the block to be valid, and nBits is the encoded form of the target threshold as it appears in the block header.

Not to be confused with: Difficulty (a number measuring the difficulty of finding a header hash relative to the difficulty of finding a header hash with the easiest target)

Node

Full node

Archival node

Pruned node

Peer

A computer that connects to the Bitcoin network.
Not to be confused with: Lightweight node, SPV node

Null data transaction

OP_RETURN transaction

Data carrier transaction

A transaction type relayed and mined by default in Bitcoin Core 0.9.0 and later that adds arbitrary data to a provably unspendable pubkey script that full nodes don't have to store in their UTXO database.
Not to be confused with: OP_RETURN (an opcode used in one of the outputs in an OP_RETURN transaction)

Opcode

Data-pushing opcode

Non-data-pushing opcode

Operation codes from the Bitcoin Script language which push data or perform functions within a pubkey script or signature script.

Orphan block

Blocks whose parent block has not been processed by the local node, so they can't be fully validated yet.
Not to be confused with: Stale block

Outpoint

The data structure used to refer to a particular transaction output, consisting of a 32-byte TXID and a 4-byte output index number (vout).
Not to be confused with: Output (an entire output from a transaction), TxOut (same as output)

Output

TxOut

An output in a transaction which contains two fields: a value field for transferring zero or more satoshis and a pubkey script for indicating what conditions must be fulfilled for those satoshis to be further spent.
Not to be confused with: Outpoint (a reference to a particular output)

P2PKH address

P2PKH output

A Bitcoin payment address comprising a hashed public key, allowing the spender to create a standard pubkey script that Pays To PubKey Hash (P2PKH).
Not to be confused with: P2PK output (an output paying a public key directly), P2SH address, P2SH output (an address comprising a hashed script, and its corresponding output)

P2SH address

P2SH output

A Bitcoin payment address comprising a hashed script, allowing the spender to create a standard pubkey script that Pays To Script Hash (P2SH). The script can be almost any valid pubkey script.
Not to be confused with: P2PK output (an output paying a public key directly), P2PKH address, P2PKH output (an address comprising a hashed pubkey, and its corresponding output), P2SH multisig (a particular instance of P2SH where the script uses a multisig opcode)

P2SH multisig

A P2SH output where the redeem script uses one of the multisig opcodes. Up until Bitcoin Core 0.10.0, P2SH multisig scripts were standard transactions, but most other P2SH scripts were not.
Not to be confused with: Multisig pubkey scripts (also called “bare multisig”, these multisig scripts don’t use P2SH encapsulation), P2SH (general P2SH, of which P2SH multisig is a specific instance that was special cased up until Bitcoin Core 0.10.0)

Parent key

Parent public key

Parent private key

In HD wallets, a key used to derive child keys. The key can be either a private key or a public key, and the key derivation may also require a chain code.
Not to be confused with: Public key (derived from a private key, not a parent key)

Payment protocol

Payment request

The deprecated protocol defined in BIP70 (and other BIPs) which lets spenders get signed payment details from receivers.
Not to be confused with: IP-to-IP payment protocol (an insecure, discontinued protocol included in early versions of Bitcoin)

Private key

The private portion of a keypair which can create signatures that other people can verify using the public key.
Not to be confused with: Public key (data derived from the private key), Parent key (a key used to create child keys, not necessarily a private key)

Proof of work
POW

A hash below a target value which can only be obtained, on average, by performing a certain amount of brute force work—therefore demonstrating proof of work.

Pubkey script
ScriptPubKey

A script included in outputs which sets the conditions that must be fulfilled for those satoshis to be spent. Data for fulfilling the conditions can be provided in a signature script. Pubkey Scripts are called a scriptPubKey in code.
Not to be confused with: Pubkey (a public key, which can be used as part of a pubkey script but don't provide a programmable authentication mechanism), Signature script (a script that provides data to the pubkey script)

Public key

The public portion of a keypair which can be used to verify signatures made with the private portion of the keypair.
Not to be confused with: Private key (data from which the public key is derived), Parent key (a key used to create child keys, not necessarily a public key)

Replace by fee
RBF
Opt-in replace by fee

Replacing one version of an unconfirmed transaction with a different version of the transaction that pays a higher transaction fee. May use BIP125 signaling.
Not to be confused with: Child pays for parent, CPFP

Redeem script
RedeemScript

A script similar in function to a pubkey script. One copy of it is hashed to create a P2SH address (used in an actual pubkey script) and another copy is placed in the spending signature script to enforce its conditions.
Not to be confused with: Signature script (a script that provides data to the pubkey script, which includes the redeem script in a P2SH input)

Regtest
Regression test mode

A local testing environment in which developers can almost instantly generate blocks on demand for testing events, and can create private satoshis with no real-world value.
Not to be confused with: Testnet (a global testing environment which mostly mimics mainnet)

RPC byte order

A hash digest displayed with the byte order reversed; used in Bitcoin Core RPCs, many block explorers, and other software.
Not to be confused with: Internal byte order (hash digests displayed in their typical order; used in serialized blocks and serialized transactions)

Sequence number

Part of all transactions. A number intended to allow unconfirmed time-locked transactions to be updated before being finalized; not currently used except to disable locktime in a transaction
Not to be confused with: Output index number / vout (this is the 0-indexed number of an output within a transaction used by a later transaction to refer to that specific output)

Serialized block

A complete block in its binary format—the same format used to calculate total block byte size; often represented using hexadecimal.

Serialized transaction

Raw transaction

Complete transactions in their binary format; often represented using hexadecimal. Sometimes called raw format because of the various Bitcoin Core commands with “raw” in their names.

SIGHASH_ALL

Default signature hash type which signs the entire transaction except any signature scripts, preventing modification of the signed parts.

SIGHASH_ANYONECANPAY

A signature hash type which signs only the current input.
Not to be confused with: SIGHASH_SINGLE (which signs this input, its corresponding output, and other inputs partially)

SIGHASH_NONE

Signature hash type which only signs the inputs, allowing anyone to change the outputs however they’d like.

SIGHASH_SINGLE

Signature hash type that signs the output corresponding to this input (the one with the same index value), this input, and any other inputs partially. Allows modification of other outputs and the sequence number of other inputs.
Not to be confused with: SIGHASH_ANYONECANPAY (a flag to signature hash types that only signs this single input)

Signature

A value related to a public key which could only have reasonably been created by someone who has the private key that created that public key. Used in Bitcoin to authorize spending satoshis previously sent to a public key.

Signature hash

Sighash

A flag to Bitcoin signatures that indicates what parts of the transaction the signature signs. (The default is SIGHASH_ALL.) The unsigned parts of the transaction may be modified.
Not to be confused with: Signed hash (a hash of the data to be signed), Transaction malleability / transaction mutability (although non-default sighash flags do allow optional malleability, malleability comprises any way a transaction may be mutated)

Signature script
ScriptSig

Data generated by a spender which is almost always used as variables to satisfy a pubkey script. Signature Scripts are called scriptSig in code.
Not to be confused with: ECDSA signature (a signature, which can be used as part of a pubkey script in addition to other data)

SPV
Simplified Payment Verification
Lightweight client
Thin client

A method for verifying if particular transactions are included in a block without downloading the entire block. The method is used by some lightweight Bitcoin clients.

Soft fork

A softfork is a change to the bitcoin protocol wherein only previously valid blocks/transactions are made invalid. Since old nodes will recognise the new blocks as valid, a softfork is backward-compatible.
Not to be confused with: Fork (a regular fork where all nodes follow the same consensus rules, so the fork is resolved once one chain has more proof of work than another), Hard fork (a permanent divergence in the block chain caused by non-upgraded nodes not following new consensus rules), Software fork (when one or more developers permanently develops a codebase separately from other developers), Git fork (when one or more developers temporarily develops a codebase separately from other developers

Stale block

Blocks which were successfully mined but which aren't included on the current best block chain, likely because some other block at the same height had its chain extended first.
Not to be confused with: Orphan block (a block whose previous (parent) hash field points to an unknown block, meaning the orphan can't be validated)

Standard Transaction

A transaction that passes Bitcoin Core's IsStandard() and IsStandardTx() tests. Only standard transactions are mined or broadcast by peers running the default Bitcoin Core software.

Start string

Network magic

Four defined bytes which start every message in the Bitcoin P2P protocol to allow seeking to the next message.

Testnet

A global testing environment in which developers can obtain and spend satoshis that have no real-world value on a network that is very similar to the Bitcoin mainnet.

Not to be confused with: Regtest (a local testing environment where developers can control block generation)

Token

A token is a programmable digital asset with its own codebase that resides on an already existing block chain. Tokens are used to help facilitate the creation of decentralized applications.

Not to be confused with: Bitcoins, Satoshis, Security token, Denominations

Transaction fee

Miners fee

The amount remaining when the value of all outputs in a transaction are subtracted from all inputs in a transaction; the fee is paid to the miner who includes that transaction in a block.

Not to be confused with: Minimum relay fee (the lowest fee a transaction must pay to be accepted into the memory pool and relayed by Bitcoin Core nodes)

Txid

An identifier used to uniquely identify a particular transaction; specifically, the sha256d hash of the transaction.

Not to be confused with: Outpoint (the combination of a txid with a vout used to identify a specific output)

User-activated soft fork

UASF

A Soft Fork activated by flag day or node enforcement instead of miner signalling.

Not to be confused with: Miner Activated Soft Fork (a soft fork activated through miner signalling), Fork (a regular fork where all nodes follow the same consensus rules, so the fork is resolved once one chain has more proof of work than another), Hard fork (a permanent divergence in the block chain caused by non-upgraded nodes not following new consensus rules), Soft fork (a temporary divergence in the block chain caused by non-upgraded nodes not following new consensus rules), Software fork (when one or more developers permanently develops a codebase separately from other

developers), Git fork (when one or more developers temporarily develops a codebase separately from other developers

UTXO

An Unspent Transaction Output (UTXO) that can be spent as an input in a new transaction.

Not to be confused with: Output (any output, whether spent or not. Outputs are a superset of UTXOs)

Wallet

Software that stores private keys and monitors the block chain (sometimes as a client of a server that does the processing) to allow users to spend and receive satoshis.

Not to be confused with: HD wallet (a protocol that allows all of a wallet's keys to be created from a single seed)

WIF

Wallet Import Format

A data interchange format designed to allow exporting and importing a single private key with a flag indicating whether or not it uses a compressed public key.

Not to be confused with: Extended private keys (which allow importing a hierarchy of private keys)

Watch-only address

An address or pubkey script stored in the wallet without the corresponding private key, allowing the wallet to watch for outputs but not spend them.