

Terminology

The following terms are used throughout the Solana documentation and development ecosystem.

account#

A record in the Solana ledger that either holds data or is an executable program.

Like an account at a traditional bank, a Solana account may hold funds called [lamports](#) . Like a file in Linux, it is addressable by a key, often referred to as a [public key](#) or pubkey.

The key may be one of:

- an ed25519 public key
- a program-derived account address (32byte value forced off the ed25519 curve)
- a hash of an ed25519 public key with a 32 character string

account owner#

The address of the program that owns the account. Only the owning program is capable of modifying the account.

app#

A front-end application that interacts with a Solana cluster.

bank state#

The result of interpreting all programs on the ledger at a given [tick height](#) . It includes at least the set of all [accounts](#) holding nonzero [native tokens](#) .

block#

A contiguous set of [entries](#) on the ledger covered by a [vote](#) . A [leader](#) produces at most one block per [slot](#) .

blockhash#

A unique value ([hash](#)) that identifies a record (block). Solana computes a blockhash from the last [entry id](#) of the block.

block height#

The number of [blocks](#) beneath the current block. The first block after the [genesis block](#) has height one.

bootstrap validator#

The [validator](#) that produces the genesis (first) [block](#) of a block chain.

BPF loader#

The Solana program that owns and loads [BPF onchain programs](#) , allowing the program to interface with the runtime.

client#

A computer program that accesses the Solana server network [cluster](#) .

commitment#

A measure of the network confirmation for the [block](#) .

cluster#

A set of [validators](#) maintaining a single [ledger](#) .

compute budget#

The maximum number of [compute units](#) consumed per transaction.

compute units#

The smallest unit of measure for consumption of computational resources of the blockchain.

confirmation time#

The wallclock duration between a [leader](#) creating a [tick entry](#) and creating a [confirmed block](#) .

confirmed block#

A [block](#) that has received a [super majority](#) of [ledger votes](#) .

control plane#

A gossip network connecting all [nodes](#) of a [cluster](#) .

cooldown period#

Some number of [epochs](#) after [stake](#) has been deactivated while it progressively becomes available for withdrawal. During this period, the stake is considered to be "deactivating". More info about: [warmup and cooldown](#)

credit#

See [vote credit](#) .

cross-program invocation (CPI)#

A call from one [onchain program](#) to another. For more information, see [calling between programs](#) .

data plane#

A multicast network used to efficiently validate [entries](#) and gain consensus.

drone#

An off-chain service that acts as a custodian for a user's private key. It typically serves to validate and sign transactions.

entry#

An entry on the [ledger](#) either a [tick](#) or a [transaction's entry](#) .

entry id#

A preimage resistant [hash](#) over the final contents of an entry, which acts as the [entry's](#) globally unique identifier. The hash serves as evidence of:

- The entry being generated after a duration of time
- The specified [transactions](#)
- are those included in the entry
- The entry's position with respect to other entries in [ledger](#)

See [proof of history](#) .

epoch#

The time, i.e. number of [slots](#) , for which a [leader schedule](#) is valid.

fee account#

The fee account in the transaction is the account that pays for the cost of including the transaction in the ledger. This is the first account in the transaction. This account must be declared as Read-Write (writable) in the transaction since paying for the transaction reduces the account balance.

finality#

When nodes representing 2/3rd of the [stake](#) have a common [root](#) .

fork#

A [ledger](#) derived from common entries but then diverged.

genesis block#

The first [block](#) in the chain.

genesis config#

The configuration file that prepares the [ledger](#) for the [genesis block](#) .

hash#

A digital fingerprint of a sequence of bytes.

inflation#

An increase in token supply over time used to fund rewards for validation and to fund continued development of Solana.

inner instruction#

See [cross-program invocation](#) .

instruction#

A call to invoke a specific [instruction handler](#) in a [program](#) . An instruction also specifies which accounts it wants to read or modify, and additional data that serves as auxiliary input to the [instruction handler](#) . A [client](#) must include at least one instruction in a [transaction](#) , and all instructions must complete for the transaction to be considered successful.

instruction handler#

Instruction handlers are [program](#) functions that process [instructions](#) from [transactions](#) . An instruction handler may contain one or more [cross-program invocations](#) .

keypair#

A [public key](#) and corresponding [private key](#) for accessing an account.

lamport#

A fractional [native token](#) with the value of 0.000000001 [sol](#) .

Within the compute budget, a quantity of [micro-lamports](#) is used in the calculation of [prioritization fees](#) .

leader#

The role of a [validator](#) when it is appending [entries](#) to the [ledger](#) .

leader schedule#

A sequence of [validator public keys](#) mapped to [slots](#) . The cluster uses the leader schedule to determine which validator is the [leader](#) at any moment in time.

ledger#

A list of [entries](#) containing [transactions](#) signed by [clients](#) . Conceptually, this can be traced back to the [genesis block](#) , but an actual [validator](#) 's ledger may have only newer [blocks](#) to reduce storage, as older ones are not needed for validation of future blocks by design.

ledger vote#

A [hash](#) of the [validator's state](#) at a given [tick height](#) . It comprises a [validator's](#) affirmation that a [block](#) it has received has been verified, as well as a promise not to vote for a conflicting [block](#) (i.e. [fork](#)) for a specific amount of time, the [lockout](#) period.

light client#

A type of [client](#) that can verify it's pointing to a valid [cluster](#) . It performs more ledger verification than a [thin client](#) and less than a [validator](#) .

loader#

A [program](#) with the ability to interpret the binary encoding of other on-chain programs.

lockout#

The duration of time for which a [validator](#) is unable to [vote](#) on another [fork](#) .

message#

The structured contents of a [transaction](#) . Generally containing a header, array of account addresses, recent [blockhash](#) , and an array of [instructions](#) .

Learn more about the [message formatting inside of transactions](#) here.

Nakamoto coefficient#

A measure of decentralization, the Nakamoto Coefficient is the smallest number of independent entities that can act collectively to shut down a blockchain. The term was coined by Balaji S. Srinivasan and Leland Lee in [Quantifying Decentralization](#) .

native token#

The [token](#) used to track work done by [nodes](#) in a cluster.

node#

A computer participating in a [cluster](#) .

node count#

The number of [validators](#) participating in a [cluster](#) .

onchain program#

The executable code on Solana blockchain that interprets the [instructions](#) sent inside of each [transaction](#) to read and modify accounts over which it has control. These programs are often referred to as "[smart contracts](#) " on other blockchains.

PoH#

See [Proof of History](#) .

point#

A weighted [credit](#) in a rewards regime. In the [validator rewards regime](#) , the number of points owed to [a stake](#) during redemption is the product of the [vote credits](#) earned and the number of lamports staked.

private key#

The private key of a [keypair](#) .

program#

See [onchain program](#) .

program derived account (PDA)#

An account whose signing authority is a program and thus is not controlled by a private key like other accounts.

program id#

The public key of the [account](#) containing a [program](#) .

proof of history (PoH)#

A stack of proofs, each of which proves that some data existed before the proof was created and that a precise duration of time passed before the previous proof. Like a [VDF](#) , a Proof of History can be verified in less time than it took to produce.

prioritization fee#

An additional fee user can specify in the compute budget [instruction](#) to prioritize their [transactions](#) .

The prioritization fee is calculated by multiplying the requested maximum compute units by the compute-unit price (specified in increments of 0.000001 lamports per compute unit) rounded up to the nearest lamport.

Transactions should request the minimum amount of compute units required for execution to minimize fees.

public key (pubkey)#

The public key of a [keypair](#) .

rent#

Fee paid by [Accounts](#) and [Programs](#) to store data on the blockchain. When accounts do not have enough balance to pay rent, they may be Garbage Collected.

See also [rent exempt](#) below. Learn more about rent here [What is rent?](#) .

rent exempt#

Accounts that maintain more than 2 years with of rent payments in their account are considered "rent exempt " and will not incur the [collection of rent](#) .

root#

A [block](#) or [slot](#) that has reached maximum [lockout](#) on a [validator](#) . The root is the highest block that is an ancestor of all active forks on a validator. All ancestor blocks of a root are also transitively a root. Blocks that are not an ancestor and not a descendant of the root are excluded from consideration for consensus and can be discarded.

runtime#

The component of a [validator](#) responsible for [program](#) execution.

Sealevel#

Solana's parallel run-time for [onchain programs](#) .

shred#

A fraction of a [block](#) ; the smallest unit sent between [validators](#) .

signature#

A 64-byte ed25519 signature of R (32-bytes) and S (32-bytes). With the requirement that R is a packed Edwards point not of small order and S is a scalar in the range of $0 \leq S < L$. This requirement ensures no signature malleability. Each transaction must have at least one signature for [fee account](#) . Thus, the first signature in transaction can be treated as [transaction id](#)

skip rate#

The percentage of [skipped slots](#) out of the total leader slots in the current epoch. This metric can be misleading as it has high variance after the epoch boundary when the sample size is small, as well as for validators with a low number of leader slots, however can also be useful in identifying node misconfigurations at times.

skipped slot#

A past [slot](#) that did not produce a [block](#) , because the leader was offline or the [fork](#) containing the slot was abandoned for a better alternative by cluster consensus. A skipped slot will not appear as an ancestor for blocks at subsequent slots, nor increment the [block height](#) , nor expire the `oldestrecent_blockhash` .

Whether a slot has been skipped can only be determined when it becomes older than the latest [rooted](#) (thus not-skipped) slot.

slot#

The period of time for which each [leader](#) ingests transactions and produces a [block](#) .

Collectively, slots create a logical clock. Slots are ordered sequentially and non-overlapping, comprising roughly equal real-world time as per [PoH](#) .

smart contract#

See [onchain program](#) .

sol#

The [native token](#) of a Solana [cluster](#) .

Solana Program Library (SPL)#

A [library of programs](#) on Solana such as spl-token that facilitates tasks such as creating and using tokens.

stake#

Tokens forfeit to the [cluster](#) if malicious [validator](#) behavior can be proven.

supermajority#

2/3 of a [cluster](#) .

sysvar#

A system [account](#) . [Sysvars](#) provide cluster state information such as current tick height, reward [points](#) values, etc. Programs can access Sysvars via a Sysvar account (pubkey) or by querying via a syscall.

thin client#

A type of [client](#) that trusts it is communicating with a valid [cluster](#) .

tick#

A ledger[entry](#) that estimates wallclock duration.

tick height#

The Nth[tick](#) in the[ledger](#) .

token#

A digitally transferable asset.

Token Extensions Program#

The[Token Extensions Program](#) has the program IDTokenzQdBNbLqP5VEhdkAS6EPFLC1PHnBqCXEpPxEb and includes all the same features as the[Token Program](#) , but comes with extensions such as confidential transfers, custom transfer logic, extended metadata, and much more.

Token Program#

The[Token Program](#) has the program IDTokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA , and provides the basic capabilities of transferring, freezing, and minting tokens.

tps#

[Transactions](#) per second.

tpu#

[Transaction processing unit](#) .

transaction#

One or more[instructions](#) signed by a [client](#) using one or more[keypairs](#) and executed atomically with only two possible outcomes: success or failure.

transaction id#

The first[signature](#) in a [transaction](#) , which can be used to uniquely identify the transaction across the complete[ledger](#) .

transaction confirmations#

The number of[confirmed blocks](#) since the transaction was accepted onto the[ledger](#) . A transaction is finalized when its block becomes a [root](#) .

transactions entry#

A set of[transactions](#) that may be executed in parallel.

tvu#

[Transaction validation unit](#) .

validator#

A full participant in a Solana network [cluster](#) that produces new [blocks](#) . A validator validates the transactions added to the[ledger](#)

VDF#

See[verifiable delay function](#) .

verifiable delay function (VDF)#

A function that takes a fixed amount of time to execute that produces a proof that it ran, which can then be verified in less time than it took to produce.

vote#

See [ledger vote](#) .

vote credit#

A reward tally for [validators](#) . A vote credit is awarded to a validator in its vote account when the validator reaches [quorum](#) .

wallet#

A collection of [keypairs](#) that allows users to manage their funds.

warmup period#

Some number of [epochs](#) after [stake](#) has been delegated while it progressively becomes effective. During this period, the stake is considered to be "activating". More info about: [warmup and cooldown](#)