How Bigtangle Works

Bigtangle combines two technologies:
BigT.AI

- A **DAG (Directed Acyclic Graph)** that allows many transactions to be processed simultaneously for high speed.
- A **Proof-of-Stake (PoS) beacon chain** that periodically confirms those transactions and makes them permanent.

Think of the DAG as a highway that moves traffic quickly, while the beacon chain acts as a traffic authority that periodically certifies everything is correct.

Every 12 seconds, one validator is chosen to create a **beacon block**. This beacon confirms all recent transaction activity and makes it irreversible.

Why Bigtangle

PROBLEM	HOW BIGTANGLE SOLVES IT
Slow transactions — Ethereum Layer 1 does ~30 tx/s, Bitcoin ~7 tx/s	DAG parallelism: thousands of transactions per second in parallel
Single-leader risk — if a designated leader fails, the chain stalls	Transaction processing has no single-leader bottleneck because DAG blocks can be created in parallel by many nodes. Beacon blocks are proposed by one validator per slot to provide finality.
Slow finality — Ethereum takes ~6 minutes for irreversible settlement	Casper FFG finality in ~12.8 minutes (2 epochs); beacon blocks confirmed in ~24 seconds
No horizontal scaling — Most chains run on a single ledger	Many L1 application chains, each with its own validators and consensus
Quantum vulnerability — ECDSA signatures can be broken by quantum computers	Dual post-quantum signatures using two NIST-approved algorithms

Architecture

Bigtangle runs two independent layers:

- The **DAG** handles transaction throughput. Because many blocks can be created at the same time, the network needs a way to decide which existing blocks a new block should connect to. Bigtangle solves this using a randomized search through recent blocks. The search naturally favors well-connected branches while still allowing alternative paths to grow. (This technique is called Markov Chain Monte Carlo, or MCMC.)
- The **beacon chain** provides finality. Every 12 seconds, a validator chosen by an unbiased random process produces a beacon block that confirms recent DAG activity. Bigtangle uses the same finality mechanism as Ethereum (Casper FFG). Once two-thirds of validators agree, the confirmed blocks become permanent.

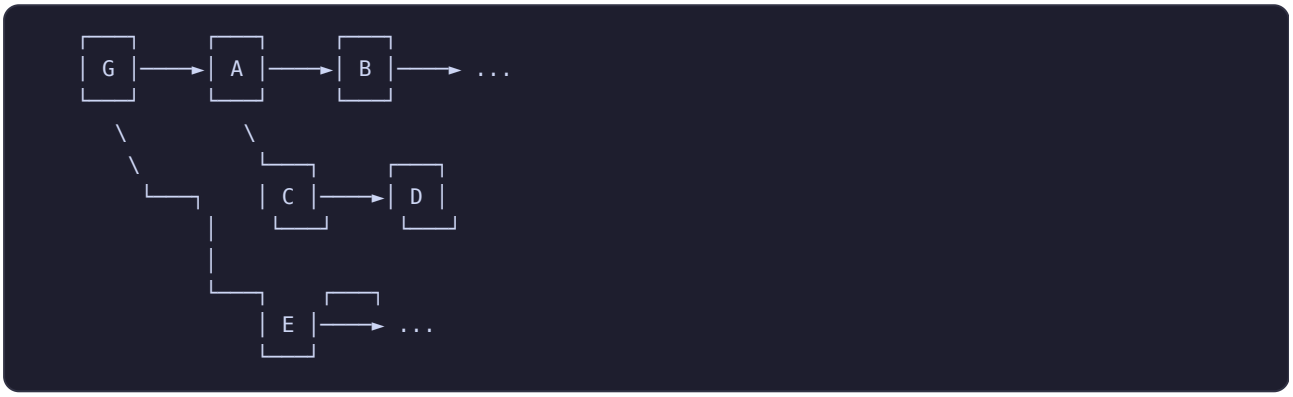
The two layers are complementary: the DAG provides speed and parallelism; the beacon chain provides security and finality.

DAG and Parallel Processing

How the DAG works

In a traditional blockchain, blocks are created one after another in a single line. This creates a bottleneck — everyone must wait for the next block.

Bigtangle uses a DAG, where each block connects to two previous blocks instead of one. This allows multiple blocks to be produced simultaneously:



Key benefits:

- **Multiple blocks in parallel** — no waiting for a single leader.
- **No empty slots** — if a validator misses its turn, the DAG absorbs it.
- **Faster confirmations** — transactions are confirmed in seconds, not minutes.

How tip selection works

When a new block is created, it needs to choose two existing blocks to connect to. The network performs a randomized search through recent blocks that naturally favors the healthiest branches while still allowing alternative paths to grow. This ensures new blocks consistently attach to the best-connected parts of the DAG even under high load.

Transaction validation

Every transaction is checked before entering the network:

1. **Format check** — no duplicate inputs, no negative values.
2. **Signature verification** — the sender's signature is verified immediately.
3. **Double-spend prevention** — the same funds cannot be spent twice in the pending queue.
4. **Fee check** — each transaction must include a minimum fee.

Invalid transactions are rejected at submission time, not when a block is produced.

Proof of Stake and Finality

How validators are chosen

Every 12 seconds, one validator is randomly selected to produce a beacon block. The selection uses an unbiased randomness source that cannot be predicted or manipulated by any participant.

Validators register by depositing at least 32 million BIG. They can be penalized if they misbehave (e.g., signing conflicting blocks). Penalties include loss of staked funds.

How finality works

At the end of each epoch (every 32 slots, approximately 6.4 minutes), the network evaluates which blocks have received enough validator support:

- A block becomes **justified** when two-thirds of the staked validators attest to it.
- A block becomes **finalized** when the chain it extends was already finalized.
- Once finalized, a block can never be reverted (unless one-third of all staked validators conspire to do so).

Fee pool and validator rewards

The system has **zero inflation** — no new tokens are minted as block subsidies. The total supply of BIG is fixed at genesis and never increases.

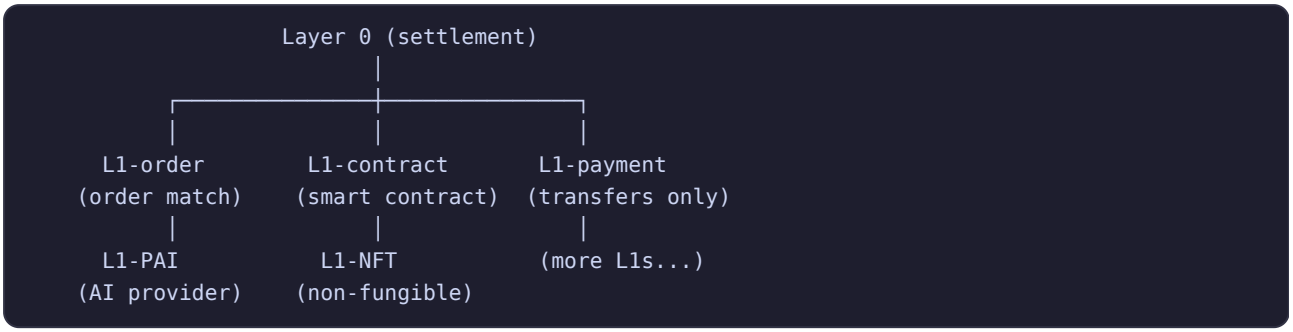
Every transaction pays a fee. Fees accumulate throughout each epoch and are distributed to validators at the epoch boundary. Each validator receives rewards in proportion to the amount they have staked. This aligns incentives: validators earn more by committing more resources to the network, and the entire reward pool comes entirely from network usage rather than monetary expansion.

Multi-Layer Architecture

One Layer 0, many Layer 1 chains

There is **exactly one Layer 0 chain worldwide**. It is the settlement chain where the native token (BIG) is created and custom tokens are issued. Layer 0 is the source of truth for token supply and the root of all value transfers.

There can be **many Layer 1 chains**, each running independently with its own validators, database, and consensus:



Each L1 chain:

- Has its own validators and consensus.
- Receives BIG tokens only via bridge from L0 (no native minting).
- Is fully isolated — a failure on one L1 does not affect others.
- Uses its own `chainId` to prevent blocks from one chain being accepted by another.

Horizontal scalability

Because each L1 chain is fully isolated and the chain ID is configurable, you can run multiple instances of the same type:

```
CHAIN_ID=PAYMENT-US  →  payment chain for US region
CHAIN_ID=PAYMENT-EU  →  payment chain for EU region
```

Each instance achieves comparable throughput to L0. Total system throughput scales linearly with the number of L1 instances, with no cross-chain coordination overhead.

Available chain types

CHAIN	PURPOSE
L0 settlement	BIG minting, token creation, global anchors
L1 order match	Decentralized order book matching
L1 smart contract	General-purpose contract execution
L1 AI provider	AI provider staking and reputation
L1 NFT	Non-fungible token creation and transfer
L1 payment	Transfer-only (minimal attack surface)

Post-Quantum Security

Bigtangle uses two post-quantum signature algorithms on every transaction input. Both are NIST-approved final standards:

ALGORITHM	STANDARD	SECURITY LEVEL
ML-DSA-87 (Dilithium)	FIPS 204	Category 5
SLH-DSA-SHA2-256s (SPHINCS+)	FIPS 205	Category 5

The two algorithms rely on mathematically independent assumptions (lattice cryptography and hash functions). Breaking one does not break the other. An attacker would need to break both simultaneously to forge a single transaction.

A standard BIP39 seed phrase deterministically generates both key pairs. If an algorithm is ever broken, the system supports upgrading to a replacement without changing the transaction format or invalidating existing funds.

Comparison

METRIC	BIGTANGLE	SOLANA	ETHEREUM L1	VISA
Ledger	DAG + beacon chain	Single chain	Single chain	Centralized
Consensus	MCMC + Casper FFG	PoH + Tower BFT	Gasper	Authority
Slot time	12s	400ms	12s	—
Finality	~12.8 min*	~12.8s	~6.4 min	Instant
Peak tx/s	~4,873	~50,000*	~30	24,000
Observed tx/s	~4,873	~2,000–3,000	~15–30	~1,700
Parallel execution	DAG-native	Sealevel (analysis)	Sequential EVM	Sharded DB

*Vendor-reported laboratory peak; observed mainnet throughput is significantly lower.

**Casper FFG finality (2 epochs). Beacon block confirmation ~24s (2 slots).

Key advantages

- **No single-leader bottleneck** — transaction blocks can be created by any node. Beacon block proposers provide finality without limiting throughput.
- **Horizontal scaling** — additional L1 chains increase capacity without protocol changes.
- **UTXO model** — transactions are inherently parallel because they reference distinct inputs.
- **Post-quantum security** — dual signatures are active from launch.

Conclusion

Bigtangle combines a parallel DAG for high throughput with a Proof-of-Stake beacon chain for deterministic finality. This architecture removes the single-leader bottleneck found in traditional blockchains while preserving strong security guarantees.

Layer 0 provides global settlement. Independent Layer 1 chains enable horizontal scaling across specialized applications without cross-chain coordination overhead.

Together, these design choices deliver high transaction throughput, rapid confirmation, modular scalability, fixed token supply, and post-quantum security within a unified architecture.

Appendix: Developer Guide

Node configuration

MODULE	PORT	DEFAULT DB	ROLE
`layer0-server`	8081	`info_l0`	L0 full node
`layer0-mcmc`	8082	—	L0 consensus
`l1-order-server`	8083	`info_order`	L1 order match
`l1-payment-server`	8091	`info_payment`	L1 payment
`l1-pai-server`	8087	`info_pai`	L1 AI provider
`l1-nft-server`	8089	`info_nft`	L1 NFT

Environment variables

VARIABLE	DEFAULT	DESCRIPTION
`CHAIN_ID`	(varies)	L1 chain identifier
`FEE_DEFAULT`	1000	Minimum transaction fee (BIG)
`POS_SLOT_INTERVAL_MS`	12000	Slot duration in ms
`POS_SLOTS_PER_EPOCH`	32	Slots per epoch

Block type scoping per chain

CHAIN	BLOCK TYPES ACCEPTED
L0	All types
L1 order match	INITIAL, TRANSFER, BEACON, ORDER_OPEN, ORDER_CANCEL
L1 contract	INITIAL, TRANSFER, BEACON, CONTRACT_EVENT, CONTRACTEVENT_CANCEL
L1 PAI	INITIAL, TRANSFER, BEACON, CONTRACT_EVENT, CONTRACTEVENT_CANCEL
L1 NFT	All types (reuses L0 params)
L1 payment	INITIAL, TRANSFER, BEACON only

Reward calculation

At each epoch boundary, accumulated fees are distributed:

```
For each active validator:
  reward = validator.stake × pool / total_active_stake
```

Fee accumulation per block

```
For each transaction in block:
  surplus = sum_of_BIG_inputs − sum_of_BIG_outputs
  if surplus > 0:
    add surplus to accumulated_fee_pool
```

Post-quantum key derivation

A 24-word BIP39 seed phrase (256-bit entropy) deterministically generates both post-quantum key pairs via HKDF-SHA256. The first 32 bytes seed ML-DSA-87, the second 32 bytes seed SLH-DSA-SHA2-256s.

Block size budget (post-quantum signatures)

COMPONENT	SIZE
ML-DSA-87 signature	4.6 KB
SLH-DSA-256s signature	16 KB
Per input total	~23 KB
100 tx/block	~3 MB (feasible)
500 tx/block	~12 MB (feasible)
2,000 tx/block	~48 MB (needs 100 MB limit)