Shitcoin: A Peer-to-Peer Electronic Shitposting System

The Shitcoin Whitepaper - v1.0, September 2026

> *In the beginning there was Bitcoin. Then there were ten thousand copies of Bitcoin.

> This is the funniest one with the best tech.*

Abstract

Here's the pitch, and it's a short one: Shitcoin (SHIT) is a joke coin built like
it isn't one. Bitcoin miners earn it on the side for doing work they were already
doing. It has two layers, a UTXO payment layer and an EVM-compatible smart contract
layer, with a trustless bridge between them. And it ships with a DEX and an
on-chain lottery that burns 10% of every pot.

We're not asking you to take Shitcoin seriously. We're asking you to take the
engineering seriously while we all laugh at the name. Under the hood it's a fork
of battle-tested Bitcoin Core / Syscoin Core software: SHA-256 AuxPoW merged
mining, LLMQ chainlocks, DIP-3 deterministic masternodes, and a full EVM. The
meme is the marketing. The tech is the product.

Ticker: SHIT - Block time: 30 seconds - Consensus: PoW (merge-mined
with Bitcoin) plus masternode quorum services - Smart contracts: NEVM
(EVM-equivalent, chain ID 20026)

---

1. Introduction

Every cycle teaches the same lesson twice. First: people buy the coin with

personality. Second: the only coins still standing four years later are the ones with real infrastructure. Dogecoin proved the first half. Bitcoin proved the second. Shitcoin is our attempt at both at once - a coin you can laugh about, running on infrastructure you don't have to laugh about.

What we're going for, in order:

1. Be genuinely fun. The brand, the lottery, the community energy. If it isn't fun, none of the rest matters.
2. Be merge-mined with Bitcoin. Security should be inherited, not purchased.
3. Be useful. EVM smart contracts, a bridge, a DEX. DeFi toys that actually work.
4. Burn stuff. Every lottery round permanently destroys 10% of the pot.

Nothing in this paper is financial advice. SHIT is a meme coin with real tech - not real tech with a meme excuse. If you lose money on a coin called Shitcoin, that one's on you.

1.1 The Mascot

Every chain gets the mascot it deserves. Ours is Mr. Shitoshi - a golden poop with googly eyes, ethereum-logo sparkles, and the quiet confidence of someone who has never once checked the price.

Why a golden poop? Because every portfolio starts as a liability: brown, smelly, technically a biohazard. Mr. Shitoshi is what it looks like when it works out. He is the patron saint of everyone who bought the top and stayed anyway.

The little ETHs orbiting him are the project's one sincere belief, drawn right on the mascot: everything outside Bitcoin is a shitcoin - including this one. Shitcoin isn't competing with Bitcoin. It exists to serve it: more transactions flowing, and merge-mined rewards that keep Bitcoin mining profitable for longer. The joke coin with a day job.

---

2. Merged Mining with Bitcoin: Security for Free

2.1 The idea

Bitcoin miners do an absurd amount of SHA-256 hashing. Merged mining (AuxPoW - pioneered by Namecoin, refined by Syscoin and Dogecoin) lets those same hashes secure Shitcoin blocks at the same time, for essentially zero extra cost. A miner builds a Shitcoin block, tucks its hash into their Bitcoin block's coinbase, and if that Bitcoin block clears Shitcoin's (much lower) difficulty target, the Shitcoin block counts.

In the code, that looks like this:

CAuxPow::check() (src/auxpow.cpp) validates the parent Bitcoin block, the chain merkle branch, and the merged-mining header sitting in the coinbase - the standard, audited AuxPoW construction.

Shitcoin's AuxPoW chain ID is 16 (testnet: 8), registered so parent coinbases can commit to it without colliding with other merge-mined chains.

Strict chain-ID enforcement (fStrictChainId) rejects parent blocks that claim our own chain ID, which closes off a whole category of spoofing attacks.

2.2 Why it matters

A brand-new standalone PoW chain is born weak: low hashrate means a 51% attack costs pocket change. A merge-mined chain is born strong, renting Bitcoin's ~500 EH/s of security for the price of an extra merkle branch. Attacking Shitcoin costs as much as attacking Bitcoin, because the work is Bitcoin's work.

For miners the pitch is one sentence: point your existing SHA-256 rigs at a Shitcoin-aware pool and collect SHIT block rewards on top of your BTC. No new hardware, no new electricity, no new heat. Free money for work you were already doing.

---

3. Dual-Layer Architecture: UTXO + NEVM

Shitcoin runs two execution environments in one client, because payments and programmability want different things:

Layer 1 - UTXO (the Bitcoin layer). Fast, simple, auditable payments. 30-second blocks, native assets, aliases, and instant probabilistic confirmations (ZDAG-style). This is where SHIT lives as money.

Layer 2 - NEVM (the Ethereum layer). A full EVM-equivalent runtime (chain ID 20026 on mainnet) embedded in the node. Any Solidity contract, any Ethereum wallet, any EVM tooling - it all just works, except gas is paid in SHIT instead of ETH.

Both layers share one validator set (the masternode quorum network) and one security budget (Bitcoin-merged mining). Moving SHIT between layers goes through the bridge (Section 4), and you don't have to trust any custodian to use it. No IOUs, no trust-me-bro.

---

4. Shitbridge: The Trustless Two-Way Peg

Most whitepapers file the bridge under "roadmap," right next to the lambo. Ours is already in the codebase.

4.1 Shitbridge: UTXO <> NEVM (native)

Moving SHIT between the payment layer and the contract layer:

UTXO to NEVM (mint): you burn SHIT in a special UTXO transaction. Relayers submit an SPV proof of that burn to the Vault Manager contract (0x7904299b3D3dC1b03d1DdEb45E9fDF3576aCBd5f on mainnet parameters). Once the proof checks out against enough proof-of-work, the contract mints the same amount of SHIT on NEVM. No custodian ever touches the funds - the burn is the deposit.

NEVM to UTXO (withdraw): you destroy NEVM SHIT in the Vault Manager. The masternode quorum sees the event, co-signs a UTXO release, and the funds show up on the payment layer.

The trust model is SPV proofs plus a deterministic masternode quorum (DIP-3) - not a multisig committee pinky-swearing it'll behave. Shitbridge is as decentralized as the masternode set, and every step is verifiable on-chain.

4.2 Bridging outward

The same construction generalizes: any chain that can verify SPV proofs (or run a light client of Shitcoin) can peg SHIT in and out. The reference design follows the sysethereum pattern - a relay contract on the foreign chain, with

the masternode quorum acting as decentralized relayers. New bridges go through governance proposals (Section 8), not hard forks, because hard forks are how you tell everyone you didn't plan ahead.

---

5. EVM Compatibility and Codebase Updates

The NEVM is a fork of go-ethereum maintained alongside this repo. The plan for keeping it fresh:

1. Track upstream geth. Rebase onto every stable go-ethereum release. The only patches we carry are the Shitcoin precompiles (Shitbridge proof verification, quorum randomness hooks) and the SHIT-as-gas changes.
2. Keep the precompile surface minimal. Every custom precompile is consensus-critical, so any addition needs an audit note in doc/. Small surface, small blast radius.
3. Solidity first. All Shitcoin DeFi (ShitSwap, ShitLottery) ships as ordinary Solidity targeting the NEVM. No custom tooling - an Ethereum dev is productive here on day one, which is more than you can say for most "EVM-compatible" chains.

Because the EVM is equivalent (not just "compatible"), contracts deployed on Ethereum redeploy on Shitcoin's NEVM unchanged. The whole Ethereum DeFi toolbox, on a merge-mined chain, with joke-tier branding. Your auditors will hate the name and love the diff.

---

6. ShitSwap: The Native DEX

A fun coin needs a casino floor. ShitSwap is the reference automated market maker, deployed on NEVM.

6.1 Design

Constant-product AMM (x * y = k) - the battle-tested Uniswap V2 design, because it works and we're not here to reinvent the wheel badly. Permissionless pair creation, LP tokens, 0.3% swap fees going to liquidity providers.

Factory plus pair architecture (contracts/ShitSwap.sol): one factory spins up minimal pairs, and each pair is its own SHIT-20 LP token.

SHIT as the hub asset. Every serious pair routes through SHIT, so all liquidity ultimately deepens the SHIT market. WSHIT pairs let UTXO-native assets trade against EVM tokens once bridged.

No admin keys in the core. Fees and pair creation answer to the masternode proposal system, not a dev multisig. We've all seen how the other version ends.

6.2 Why an AMM fits Shitcoin

Order books need market makers; meme coins have gamblers. An AMM turns every holder into a passive market maker and every trade into exit liquidity with better vibes. And combined with the bridge, ShitSwap lets you swap a UTXO asset for an NEVM meme token in two transactions - the kind of UX that gets a fun chain actually used instead of just talked about.

---

7. ShitLottery: Provably-Fair Degeneracy

The centerpiece toy: an on-chain lottery. **You pay SHIT for tickets, one ticket wins the whole pot at random, and 10% of every pot gets burned forever.**

One honest note before the rules: the lottery running live on the site right now is a simpler version - 80% to the winner, 20% to the team, drawn by hand. It works and people win, but it's training wheels. What follows is the full on-chain design it's growing into.

7.1 Rules

1. Rounds. The lottery runs back-to-back rounds, each with a fixed ticket price (say, 100 SHIT) and a ticket window measured in blocks.
2. Tickets. Call buyTickets(n), send n times the price. Every ticket is one entry - more tickets, more chances. All funds sit in the contract. No custody, no operator, nobody to run off to the Bahamas.
3. The draw. Once the window closes, anyone can call drawWinner(). The winner is uint(keccak256(blockhash(closeBlock), roundId)) % ticketCount. The draw is permissionless - if nobody calls it, the pot just waits. It'll

still be there tomorrow. Patience, degen.

4. The split. The winner takes 90%. The other 10% goes to the burn address (0x000000000000000000000000000000000000dEaD) and leaves supply forever. Every round is a small deflationary event.
5. Next round. A fresh round opens automatically. Edge cases (a round with zero tickets) just roll over. Even the contract knows not to throw a party nobody came to.

Reference implementation: contracts/ShitLottery.sol.

7.2 On randomness, honestly

Block-hash randomness has a known weakness, and we're going to say it out loud instead of burying it on page 40: whoever makes the closing block can throw it away if it makes them lose (costing them the block reward). For a fun, low-stakes lottery that's an acceptable trade-off. The permissionless draw also means no operator can stall or rig the timing. When the pots get serious, the upgrade path is a masternode-quorum randomness beacon (the LLMQ network already does threshold signatures - it's the natural decentralized randomness source on this chain) or an external VRF oracle. Ship the fun version first; harden it as the money grows.

7.3 Why burn 10%?

Two reasons. First, every round becomes a tiny buy-and-burn engine pushing against emission - the more fun people have, the scarcer SHIT gets.
Second, watching the burn counter climb is half the entertainment. Our position: a meme coin should have at least one mechanism that's unironically good tokenomics, and this is it.

7.4 Native Vault: lock it and forget it

Not everyone wants to gamble. The native vault lets you lock SHIT for **30 days to 5 years** and earn yield while you wait - enforced by consensus, not by anyone's promise:

Lock duration	**APY**
30 days	2%

Lock duration	APY
6 months	3%
1 year	4%
5 years	15%

Here's the deal: vaultlock creates a P2WSH output with the witness script <locktime> CHECKLOCKTIMEVERIFY DROP <your-pubkey> CHECKSIG. In plain English: a timelock only your key can ever spend, and only after it matures. There is no early exit - lock it and forget it, literally. When you vaultclaim after maturity, the network itself mints your yield and pays it out next to your principal. Nobody approves it, nobody funds it, nobody can stop it. It works like a block reward because, mechanically, it is one.

Where the yield comes from. Vault yield is paid out of the 17% vault yield reserve (3,570,000,000 SHIT) - and here's the part that matters: **those coins don't exist yet, and nobody holds them.** The reserve was never sent to an address. It's a protocol-level number every node tracks, like the block subsidy schedule, and fresh SHIT for yield is minted straight out of it. No address means no private key, which means not even the team can touch it. Two consensus caps keep it honest: at most 47.6M SHIT of yield per year, 3.57B total - exactly the math for the full 75-year schedule. If it's ever exhausted, locks still pay back principal in full; they just stop earning. The floor is "you get your money back," which is a better floor than most of crypto offers.

One more thing: bitcoin holders only. To lock or claim, the official wallet asks you to hold any amount of bitcoin - literally a single satoshi counts - in a BTC address you prove you own by signing a challenge message. Full disclosure on how this works: the Shitcoin chain can't see the Bitcoin chain, so this is enforced by the wallet software as policy, not by consensus. No point pretending otherwise. It's our way of keeping the vault aligned with the Bitcoin community we're merge-mined alongside.

The EVM ShitVault contract (contracts/ShitVault.sol) still exists as a DeFi companion with its own 10%-burn early-exit mechanic - but the canonical, trustless reserve described above lives at consensus level, where it can't be

rugged by definition.

8. Tokenomics

Max supply: 21,000,000,000 SHIT (21 billion). No tail emission - the last new SHIT gets minted roughly 100 years after genesis. Where it all goes:

Bucket	Share	Amount	Notes
Presale	18%	3,780,000,000	Team-controlled block-1 payout; distributed to buyers via the ShitVesting contract (24 monthly unlocks, no cliff)
Team	14%	2,940,000,000	Core team and founders
Coin support	2%	420,000,000	Listings, liquidity, marketing, ops
Wrapping reserve	6%	1,260,000,000	Team-controlled block-1 payout; backs wrapped SHIT on Solana/Ethereum launchpads and exchanges
Vault yield reserve	17%	3,570,000,000	Protocol-level reserve (no custodian); minted as vault yield only, capped at 47.6M SHIT/year over 75 years
Mining	43%	9,030,000,000	Block rewards (see below)

The presale, team, coin-support and wrapping-reserve allocations are paid in block 1 - the coinbase has to contain exactly four outputs with exactly those amounts, enforced in consensus (CheckAllocationBlock() in src/validation.cpp). The schedule can't be changed after launch, by anyone. Including us. Especially us.

The 17% vault reserve is not paid in block 1: it only exists as a consensus-tracked reserve, and the only thing that can ever mint from it is vault yield (CheckVaultYield()), capped per year and in total. The genesis block's 50 SHIT is unspendable, as is tradition.

Emission. Mining starts at block 2 at about 198.07 SHIT per block, declining 2% per year over a 100-year tail. The base subsidy is tuned so lifetime mining lands on the 9.03B allocation - a century of rewards for merge-mining Bitcoin, covering Bitcoin's entire subsidy era (to around 2140) and then some.

Reward split (per block): 10% goes to the governance superblock. Of the remaining 90%: 25% to miners (merge-mined, Section 2), 75% to masternodes. Half of all transaction fees go to masternodes too.

Masternodes. 100,000 SHIT collateral, DIP-3 deterministic registration. They run the quorum services: ChainLocks (instant finality), Shitbridge relaying, governance voting.

Governance. Monthly superblocks fund whatever masternodes vote for - bridge deployments, EVM updates, lottery tweaks, marketing. Vault yield doesn't need governance: consensus mints it automatically inside the caps.

Burns. The lottery's 10%-per-round burn and the EVM vault's 10% early-exit penalty are the deflationary sinks. Both only ever push circulating supply further below the 21B cap.

---

9. Network & Consensus

The spec sheet:

30-second blocks, SHA-256 PoW, AuxPoW merged mining with Bitcoin.

LLMQ ChainLocks (llmq400_60): masternode quorums sign every block - near-instant finality and 51%-attack immunity, stacked on top of merged mining. Belt and suspenders.

DIP-3 deterministic masternodes, DIP-19 data handling, full Taproot/Segwit support from the Bitcoin Core base.

Network identity: mainnet P2P magic ce e2 ca ff, default port 8369, bech32 HRP shit (chain params in src/kernel/chainparams.cpp).

---

10. Roadmap (subject to vibes)

Phase 1 - Launch the joke properly. Rebrand the binaries, public testnet, mining pools pick up SHIT AuxPoW, first ShitLottery round on NEVM testnet.

Phase 2 - DeFi toys. ShitSwap on mainnet, lottery randomness beacon via LLMQ threshold signatures, a bridge UI your mom could use.

Phase 3 - Expansion. Outward bridges (Section 4.2), CEX listings through

community governance proposals, merch (you know you want the shirt).

Phase 4 - World domination. Unlikely. But funny to write down, and this is our whitepaper so it stays.

---

11. Risks & Disclaimers

SHIT is a meme coin. Buy it for fun, not as an investment strategy. Nothing in this paper is financial advice. If a coin named after poop is your retirement plan, please log off and touch grass.

The smart contracts (ShitSwap, ShitLottery, bridge contracts) can have bugs. The reference implementations are starting points, not audits. Code is law until the law has a bug.

The bridge and lottery trust assumptions are documented above - read them before bridging your life savings (don't bridge your life savings).

Meme coins, lotteries, and DeFi live under different rules in different places. Whether the lottery is legal where you live is your homework. We are not your lawyer.

References

1. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
2. Namecoin / Syscoin AuxPoW merged-mining construction (src/auxpow.cpp).
3. Dash DIP-0003 (deterministic masternodes), DIP-0019.
4. Uniswap V2 core (constant-product AMM design), 2020.
5. Syscoin SYSX / sysethereum bridge design docs.
6. This repository: src/kernel/chainparams.cpp (network and emission parameters), src/validation.cpp (GetBlockSubsidy), contracts/ (ShitSwap, ShitLottery, ShitVault - with compiler build artifacts in contracts/build/).
