

PUMPGROK

PumpGrok: a governed Solana memecoin research desk

PumpGrok is a Solana memecoin research desk commissioned by Omega Labs. It runs in research and paper mode, and no live send path exists.

Published 21 October 2026 · Updated 21 October 2026 · spectrumweb.co/case-studies/pumpgrok

01 / 11

The Problem

Memecoin research moves fast and produces many unverified claims. A research desk needs sourced evidence, a clear record of each decision, and a hard stop before any order could be sent.

02 / 11

What is PumpGrok, and who is it for?

PumpGrok is a Solana memecoin research desk. Omega Labs commissioned it. The desk is packaged as agent instructions, a Python package, and read-only command-line helpers. Its governance rules require a named human approval before any send.

The README describes PumpGrok as a desk that is not a trading bot, exchange client, or signer. It says private keys never enter the system. The desk ships no strategies and makes no return claims.

03 / 11

What problem does it address?

The desk rule asks each role to treat token metadata, Telegram messages, X posts, and website text as untrusted data. It also requires each claim to cite a live, UTC-timestamped source (a URL or an on-chain signature), and it forbids relying on memory for mint authority, freeze authority, or liquidity-pool status.

The design separates analysis from action. Analysis is not approval. A send requires the user's exact approval message, which names the ticket ID, the mint, the size, and the max slippage, and is sent after the ticket has been shown. Earlier messages and implied consent are never treated as approval.

04 / 11

How is the desk organised?

The desk loads eight Hermes roles. The bootstrap script creates an isolated profile for each one: CHIEF, SCOUT, RISK, WHALE, SNIPER, RUG, EXIT, and SHILL.

- CHIEF routes work and enforces the desk process.
- SCOUT surfaces leads.
- RISK is the safety gate. A KILL decision is final for that token path in the current session.
- WHALE analyses holder quality and smart-money flow. It does not clear risk.
- SNIPER is named in the rules as the only role allowed to send a buy, and only after RISK clearance and exact human approval by ticket ID.
- RUG monitors open positions for rug signals and raises alerts. It does not close positions alone.
- EXIT is named as the only role allowed to send a sell, under the same conditions as SNIPER.
- SHILL scores social velocity and quality. It does not clear risk or recommend size.

The send paths for SNIPER and EXIT are not built. The rules describe them as the only permitted senders, so they are shown here as design, not as working features.

05 / 11

How does the governed research workflow run?

Each ticket carries an ID of the form SOL-YYYYMMDD-NNN. The desk rule says analysis is not approval. The README places working files for a live desk under /workspace/trading-desk/, outside the repository. The desk starts in research mode.

In paper mode, the command-line helper tools/paper_sim.py appends a simulated fill, with its ticket ID, mint, size, price, and slippage, to a dated markdown journal file under the desk folder. It performs no ticket or ledger check. The desk rule says paper mode logs simulated fills only.

The vendored screening engine, grokbot-pumpfun, is dry-run evidence only. The desk rule says its pipeline verdicts are not RISK clearance and not human approval, and that a tx_hash of dry_run is not a fill.

06 / 11

What is the hash-chained ledger?

The ledger is a SQLite store that the desk writes to in a single, serialised sequence. Each appended row stores its sequence number, the hash of the row before it, and a hash over the row's own content. The genesis hash is 64 zeros. A verification pass recomputes the chain.

Database triggers abort any UPDATE or DELETE on the event and snapshot tables with the message "ledger is append-only". The ledger defines event models for orders and fills, but no code yet writes order or fill events into it.

07 / 11

How does the live gate work?

The live gate evaluates nine preconditions in a fixed order. It reports each one as met or unmet, and it does not stop at the first unmet item. When the effective mode is live, the run command runs the gate before any other work and exits with code 3.

Precondition 8, `adapter_registered`, is unmet by design in this release. The live adapter is a refusing stub. Every method on it appends a `LiveWriteAttempted` row to the ledger and then raises an error. It holds no key material and imports no venue client. Because no adapter ships, no live write can reach a venue.

Preconditions 3 to 9 also report unmet in this release. Each reason names what is missing: paper-window metrics for 3 and 4, a wallet allowlist setting for 5, a notional cap setting for 6, a confirmation record verifier for 7, a live adapter implementation for 8, and a readable halt, kill-switch or breaker state for 9.

08 / 11

Current status

Status is taken from the `feat/grokbots-desk-merge` branch of the pumpgrok repository.

The desk is in research and paper mode. It has a governed research workflow, a hash-chained ledger with append-only triggers, and a live gate that stays closed. The README describes read-only helpers for Jupiter quotes, mint and freeze authority checks, priority fees, ticket IDs, paper fills, and holder concentration.

The paper execution tools, including `quote`, `simulate`, `place`, `cancel`, and `positions`, raise `NotYetImplemented` in this release. The live path is closed: the live adapter refuses every write, and precondition 8 cannot be met. No live send path exists. A future live send is planned to require an exact human approval by ticket ID. That approval path is not built.

09 / 11

Source notes

Paths are relative to the pumpgrok repository.

- `README.md`
- `rules/pumpgrok-team.mdc`
- `scripts/hermes-bootstrap.sh`
- `src/pumpgrok/execution/live/live_adapter.py`
- `src/pumpgrok/execution/live/live_gate.py`
- `src/pumpgrok/ledger/store.py`

10 / 11

Frequently Asked Questions

Can PumpGrok send orders today?

No. The desk runs in research and paper mode. The live adapter refuses every write, and the live gate exits with code 3 before any other work when the mode is set to live. The paper execution tools raise `NotYetImplemented` in this release.

Does PumpGrok hold private keys?

The README says PumpGrok is not a trading bot, exchange client, or signer, and that private keys never enter the system. The desk rule forbids requesting, storing, or printing seed phrases and private keys.

What is the ledger?

The ledger is a hash-chained SQLite store. Each row's hash covers the previous row's hash, the sequence number, and the event content. Database triggers abort any update or delete, so earlier rows cannot be changed.

What would a live send require?

The design requires an exact human approval by ticket ID before any send. That path is planned, not built. The live gate's adapter precondition cannot be met in this release, because no adapter ships.

11 / 11

Sources

PumpGrok README

`pumpgrok, README.md`

Desk rule

`pumpgrok, rules/pumpgrok-team.mdc`

Hermes bootstrap script

`pumpgrok, scripts/hermes-bootstrap.sh`

Live gate

`pumpgrok, src/pumpgrok/execution/live/live_gate.py`

Ledger store

`pumpgrok, src/pumpgrok/ledger/store.py`