OUTCRY

OUTCRY is a chain: a fork of Agave, the Solana validator, run by this project on four validators. Its order book is part of genesis, and the order in which transactions to that book execute inside a block is a public rule: cancels first. Day one the chain's money is test money from a faucet. It has no value.

What runs today

Two things are changed from stock Agave.

The book in genesis. The order book is Phoenix v1 (Ellipsis Labs, MIT licence), the executable read from Solana mainnet, installed in genesis at its own program id PhoeNiXZ8ByJGLkxNfZRnkUfjvmuYqLR89jjFHGqdXY under the upgradeable loader with no upgrade authority. solana program show prints Authority: none; nobody can change the program. One market runs: test SOL against test USD, taker fee 0.05% and maker fee 0, both in test money.

The line. When a validator builds a block, transactions that call the book are lined up by a public rule:

1. A book transaction is one with at least one top-level instruction to the book program.
2. Class 0: every book instruction is a cancel. Class 1: only cancels and post-only quotes, at least one quote. Class 2: anything else that calls the book, takes included. Top-level instructions to ComputeBudget, System, SPL Token and Associated Token Account are ignored; any other top-level program makes a book transaction class 2.
3. Book transactions are scheduled by class, then by the slot in which the block builder received them (earlier first), then by the SHA-256 hash of the first signature (lowest first). Fee priority plays no part among them, and every book transaction is scheduled before other transactions waiting at the same moment.

4. A class 2 book transaction becomes schedulable 50 ms after the block builder received it. A cancel or a quote that reaches the block builder no later than 50 ms after a take therefore executes before that take. A cancel that arrives more than 50 ms after a take can execute after it, in the same block or a later one.

What enforces the line is the block builder's scheduler, and only that: validators replaying a block do not check its order. Which block a transaction lands in is decided by the block builder, and every block builder on this chain is a validator this project runs. Transactions that reach the book only through another program are not book transactions for the rule.

The build: Agave v4.5.0-alpha.4 (commit ccddd3dffc7edb1d6ed8182ac8e9cbd73b24b159) with three patch files: line-rule.patch (the line, in the block builder's scheduler), genesis-no-authority.patch (a program installed in genesis with no upgrade authority records none, not the zero key) and ledger-floor.patch (a smaller minimum ledger size, so a small network stays inside its disk). The build record holds the hash of every patch and binary.

The network: four validators, all run by this project, reached by the public only through one RPC on the project's domain. Stake is equal once warm, so the chain keeps going with one validator down. The chain runs on test gas from a faucet. The token $OUTCRY is an SPL token on Solana mainnet; it is not this chain's gas at launch.

Measured

Every figure here comes from a rig.py record: the command, p50 and p99, the host and the network it ran on. A figure without a record is not stated.

RECORD	VALUE	CONDITIONS
line-test-cancel-first-pct	100 % (p50 and p99 of 3 runs): in 19 of 19 pairs that landed in the same block, the cancel executed first. 11 of 30 pairs landed in different blocks and are counted apart: in 10 of them the cancel's block came first, in 1 the take's block came first and the take filled the quote	3 runs of 10 pairs: a take of a resting post-only ask and that ask's cancel, sent in the same moment from one client; the ask one tick under the best ask of the house quotes, the take limited to the ask's price and size; 4 validators, all voting, stake still warming (one validator holds most of it); one host, AMD Ryzen 9 9950X, 32 vCPU, load average about 100 from other work on the host; 2026-11-19 09:33 UTC
quote-send-to-confirm-ms	p50 438.5 ms, p99 484 ms (min 422 ms)	10 runs: one post-only quote from a fresh funded key, from send to confirmed commitment, polled, client on the same host through the RPC; same network and host as above, load average about 100; 2026-11-19 09:34 UTC
quote-compute-units	p50 21,912 CU, p99 28,067 CU	3 runs: compute units the post-only quote transaction consumed, read from the transaction's metadata; same network as above; 2026-11-19 09:34 UTC

RECORD	VALUE	CONDITIONS
net-disk-gb-peak	29.76 GB at most (min 24.78 GB) for the whole test network: ledgers, accounts and snapshots of all 4 validators. Each validator keeps only a fixed window of the newest ledger data (2,876,094 data shreds on the RPC validator, 766,958 on each of the other three) and frees older data when its database compacts, so the total moved inside this band rather than growing over the series	1 run over 14 samples taken 30 min apart, 2026-11-19 13:30 to 19:23 UTC, every window already full; 4 validators, all voting; the band is what we saw in those 6 hours, not a bound proven for any load

How to reproduce

Only the URL changes: every command below uses the stock Solana CLI or web3.js against the public RPC, whose URL is on /docs.

The book cannot be changed:

```
solana program show PhoeNiXZ8ByJGLkxNfZRnkUfjvmuYqLR89jjFHGqdXY -u "$
```

Read the order inside a block: solana block <SLOT> -u "$RPC" --output json lists transactions in the order they executed. A book instruction's class comes from its first data byte, and the second when the first is 2 or 3: cancel 4-11; post-only quote 16, 17, or 2 or 3 followed by 0; everything else is class 2.

One line test by hand: with the two signatures the line test page shows, solana confirm -v <signature> -u "$RPC" prints each one's

slot; when the slots match, solana block <slot> lists the cancel before the take.

The figures: each row of *Measured* is rerun with rig.py run --name <record> --runs <k> -- <command>, and the line test with the client's pair command, which sends a take of a resting quote and that quote's cancel in the same moment.

The source: the patch files, the build command that applies them to the named Agave tag, the genesis configuration, the client and the house-quote bot are in the project's public repository.

Sources

- Agave v4.5.0-alpha.4, Anza, https://github.com/anza-xyz/agave (pre-release tag; our upstream).
- Phoenix v1, Ellipsis Labs, https://github.com/Ellipsis-Labs/phoenix-v1, MIT licence from commit 6ae54af5f07f4d0ae8430b08cbe81d10432cd34c.

What does not work yet

- The line is enforced by the block builder only. A validator replaying a block does not check its order.
- Every validator is run by this project. Outside validators are not possible on this network today.
- A cancel that reaches the block builder more than 50 ms after a take can execute after it.
- The chain trades test money only. $OUTCRY is not this chain's gas or stake yet; that path needs a security review and is not built.
- There is no comparison run of the same test on stock Agave, so nothing here says how stock ordering behaves.
- The RPC node keeps only recent transaction history; older blocks are in the project's own index behind /blocks.