

PILOT, 10 AND 11 OCTOBER 2026

Lane Arena: local model vs. Claude and GPT

Two real tickets, six implementers, isolated clones, a blind review
and hidden tests.

THE QUESTION

Can a local open-weight model keep up as an implementer?

Background

Today the local lane takes on tasks that would otherwise go to the mid-tier Claude implementer. Security-critical and hard work stays with Claude.

Goal of the pilot

Build a repeatable harness and deliver a first comparison on one medium and one hard task.

PARTICIPANTS

Six implementers, same brief

● **Lane**

Qwen3.8-Flash-Next via opencode,
running locally on the Mac Studio

● **Haiku 5.5**

`claude --bare -p`

● **Sonnet 5.5**

`claude --bare -p`

● **Opus 5.5**

`claude --bare -p`

● **GPT-6.1-Sol**

`codex exec` with a ChatGPT account

● **GPT-6-Luna**

`codex exec` with a ChatGPT account

Claude: no MCP, no hooks, no memory, no subagents. Codex: no user config, no rules, no MCP, no memory. This matches the lane's context.

THE LOCAL LANE

Hardware and throughput

Mac Studio, Apple M5 Max

- 18 CPU and 40 GPU cores, 128 GB unified memory
- llama.cpp (Metal), 262k context, 2 slots
- MTP for speculative decoding

~826 tok/s

Prefill on a 16k prompt (measured live)

Qwen3.8-Flash-Next

- MoE with 125B parameters, 6B active
- 4-bit GGUF (UD-IQ4_XS), 93.7 GB
- Thinking on, effort xhigh by default

54 to 59 tok/s

Generation with thinking and MTP

THE TASKS

Two real open tickets

M: Role-dependent controls

Frontend (React/Next). Buttons the server rejects with 403 should be hidden for MEMBER and VIEWER, or disabled with a reason. Clear acceptance criteria, typical lane territory.

H: Reserved usernames

Backend (Express/Prisma), security. The service-account usernames should live in one constant and be blocked at registration. The traps: letter case after normalisation, invite codes, and a test that only runs with a database.

RULES

Isolation and fair scoring

Isolation

- Own repository clone per participant, without a remote
- Same base commit, own dependencies
- Strictly sequential, never in parallel

Scoring

- Fresh Opus reviewer per round, blinded by a random label
- Hidden tests, frozen beforehand; the required cases fail on the base state
- Passed = the reviewer accepts, at most 3 rounds

RESULTS

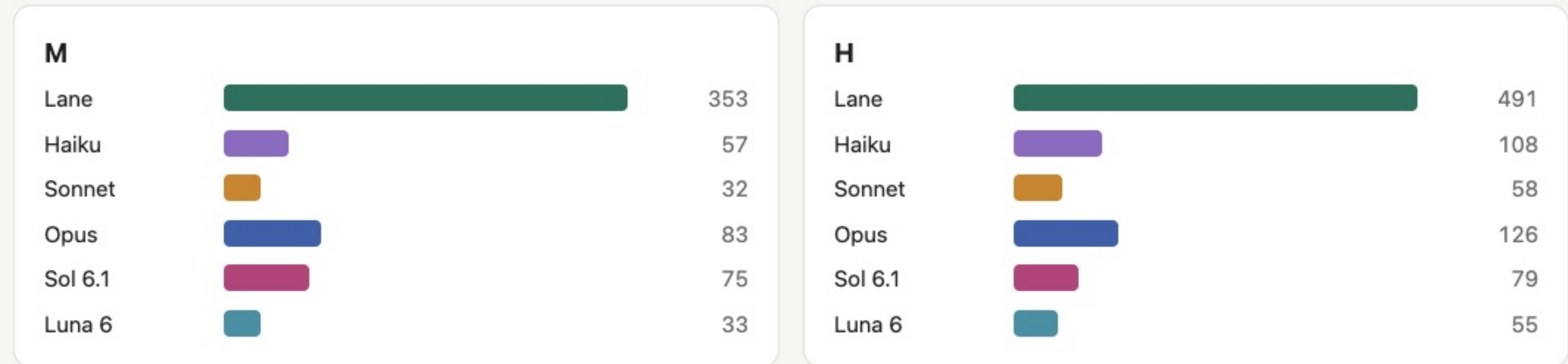
In which round did they pass?

TASK	LANE	HAIKU	SONNET	OPUS	SOL 6.1	LUNA 6
M: Controls	Round 1	Round 1	Round 1	Round 1	Round 1	Round 1
H: Usernames	Round 1	Round 2	Round 3	Round 1	Round 2	<u>failed</u>

The hidden tests were green for everyone in every round. Differences showed up in the review and in a database test measured afterwards.

TIME

Implementation time in round 1 (seconds)



In round 1 the lane is 3.9 to 11 times slower; until acceptance on H only 1.3 to 3.9 times, because others needed fix rounds. Every step of the agent loop costs prefill plus thinking.

COST

The review dominates cost, not the code

~\$0

Local implementation
(electricity not measured)

**\$0.01 to
\$0.39**

Implementation with Claude
per model and task; GPT at
API list price \$0.007 to \$0.28

**\$0.22 to
\$0.90**

per blind Opus review
(medium \$0.22 to \$0.35,
hard \$0.53 to \$0.90)

\$10.37

API cost of the scored
rounds, of which \$9.15
reviews; Codex via a
ChatGPT subscription

Every additional round costs another review. Fewer rounds are the biggest cost lever.

EFFORT

Effort was not evenly distributed

Claude: medium

--bare ignores the user settings (xhigh). The default `effort: medium` was sent, for the reviewer too. Proven by request capture.

Lane: xhigh

Without a variant, opencode sends no effort, so the chat-template default applies (xhigh). Thinking is on, but opencode does not count the reasoning tokens.

Codex: model default

Sol 6.1 at low, Luna 6 at medium (per the model cache, deliberately not pinned). The reasoning tokens match.

OBSERVATIONS

What the review and a follow-up measurement showed

Only the lane keeps an existing database test green

An existing database test expects 409. Measured afterwards, only the lane keeps it green. Opus, Haiku, Sol and Luna break it, and the review still accepted three of them. Sonnet changed the expectation to 400 without the approval the reviewer's suggestion required.

Fast is not the same as done

Luna 6 was among the fastest on M with 33 seconds, just behind Sonnet (32 s), but failed on H after three rounds. The lane was the slowest everywhere and passed both tasks in round 1.

Docs drove the review rounds

Everyone implemented the required block. The mediums on H were almost all about line references in the repository's knowledge base, the changelog and docs statements. Luna 6 failed on exactly this three times.

LIMITS

What the pilot does not show

- $n = 2$ tasks
- Uneven effort (lane xhigh, Claude and Luna medium, Sol low)
- Reviewer one tier below production

- Hidden tests too easy, everyone passed
- One review per solution, noise unmeasured
- Review only partially blind
- Local verification without database tests

CONCLUSION

Slower, but in this pilot not worse

Today

The rule "lane = mid-tier implementer" is supported. One good result is not enough to relax it for security-critical tasks.

Next run

- More tasks
- Pin the effort for everyone, reviewer at high
- Two reviews per diff, sharper hidden tests