The OTC Desk Agent Arena — Methodology & Results (Run #8)

A controlled, repeated-trial evaluation of ten LLMs operating a real OTC derivatives trading desk, fully autonomously, with no human in the loop.

Date: 2026-06-27 · **Run:** #8 · **Task:** `risk-manager-control-day` · **Trials:** 5 per model · **Gateway:** ZenMux

---

Abstract

Most LLM benchmarks score a model on a frozen prompt. A trading desk is not a frozen prompt — it is a stateful environment where the operator must read risk, price portfolios, run scenarios, back-test hedges, and produce a governance report, each step depending on the last, with no human to approve or correct intermediate actions. The **OTC Desk Agent Arena** measures exactly this: it drives the *real* desk orchestrator end-to-end and scores whether a model can run a risk-manager's control day on its own.

Run #8 evaluated **ten models** over **five independent trials each** under a single execution regime (headless "YOLO" — no human-in-the-loop interrupts and no deferral tool), scored by a 31-point objective manifest combined 50/50 with a GPT-5.5 judge. The headline result is a **statistical tie at the top — Claude Opus 4.8 (66.4) and GPT-5.5 (66.3)**, separated by 0.1 with $\sigma \approx 8$. Repetition was decisive: a single-trial pilot (run #7) had ranked these two in the opposite order and placed Claude Sonnet 4.6 near the floor on one unlucky sample; over five trials Sonnet is the **most consistent operator in the field (σ 2.7)**.

We also separate two failure modes that a naive leaderboard conflates: **model incapability** (the model connects fine but cannot operate the desk) versus **infrastructure censoring** (the model is capable but its gateway route failed). GLM 5.2 is the canonical example of the latter and is reported separately.

---

1. Introduction

Agentic benchmarks increasingly try to measure "can the model do the job," not "can the model answer the question." For a derivatives desk, the job is a *sequence* of stateful operations:

> *"Pull the latest risk on the control portfolio → price it on the control profile → find the hotspot underlying → run the Greeks landscape → stress it → back-test the hedge → write the governance report."*

What makes this hard for an LLM is not any single step — it is doing all seven **without a human**: resolving ambiguous references ("the control portfolio") to the right entity, threading IDs from one async task into the next, and never stalling to ask for confirmation. A model that is excellent in a chat box but freezes the moment no human is present will fail here, and that failure is invisible to a single-shot benchmark.

The arena is built to expose that. It does not simulate the desk — it runs the production desk orchestrator, the same code path a human trader drives, and reads the model's work back out of the system's own trace log.

What the arena measures

> **The ability to operate the desk as a competent human would, end-to-end, with zero human intervention.**

Concretely: did the model route the expected skills, call the right tools on the right entities, produce the durable artifacts (risk run, valuation, scenario, back-test, report), and narrate the governance decisions — across all seven steps, on its own.

---

2. The Arena

2.1 The task

The flagship workflow `risk-manager-control-day` is a **seven-step risk-manager control day** with a 31-point objective manifest. The scenario is seeded so that a specific underlying (AAPL) becomes the portfolio's risk hotspot only *after* a fresh pricing run — a model that prices the wrong portfolio, or reads stale risk, surfaces the wrong hotspot and cascades errors through the remaining steps.

2.2 The environment — real, not simulated

Each trial:

1. **Seeds fixtures** into the live desk database under fresh, arena-tagged IDs (portfolios, positions, pricing-parameter profiles), then purges them after.
2. **Creates a real agent thread** and drives every workflow step through the production `AgentService.stream_and_persist` path bound to the candidate model.
3. **Reconstructs the transcript from the trace log.** Skills routed are *ground truth* — the deep-agent loads each skill it follows by reading the actual `SKILL.md` file, so the trace records what the model genuinely did, not what it claimed.

Because it is the real orchestrator, every production safeguard, tool, and sub-agent is in play — including the failure surface (gateway latency, tool errors, sub-agent delegation), which becomes part of what we measure and, where relevant, control for.

2.3 Execution regime — headless "YOLO"

The arena drives every turn in a **single regime**: fully headless.

- **No human-in-the-loop interrupts.** HITL gates auto-clear.
- **No deferral tool.** The `propose_reply_options` card tool — the mechanism a model uses to bounce a decision back to a human — is *withheld entirely*, and the system prompt instructs the model that no user is present.

This matters because it eliminates a confound. In an earlier design, models could "pass" by deferring to a (non-existent) human, and an auto-continue heuristic then guessed an answer on their behalf — which drifted to the wrong portfolio and corrupted scores. Removing the deferral path means **the model must commit to its own judgment**, which is the entire point of an autonomy benchmark.

> **Single-regime fairness.** Every model in run #8 was driven through the identical headless path. A model cannot score by punting to a human, and none is advantaged by a different harness.

2.4 Scoring

Each trial's total is a 50/50 blend:

- **Objective (0–100):** fraction of the 31-point manifest satisfied — deterministic checks against the trace (correct entities priced, hotspot identified, artifacts produced, async task IDs surfaced, etc.).
- **Judge (0–100):** GPT-5.5 grades the transcript against the workflow rubric (governance narration, correctness, completeness), with retries to ride out transient gateway errors.

`total = 0.5 · objective + 0.5 · judge`.

2.5 Repetition and averaging

Every model runs **five independent trials**; its score is the **mean** of the trial totals, reported with the **standard deviation** as a first-class reliability signal. The `ArenaMatch` table enforces one row per (run, workflow, model), so the five trials are averaged into a single row whose breakdown carries the per-trial detail. Repetition is not a formality — see §4.

---

3. Models evaluated

Ten models, all routed through the ZenMux gateway (Anthropic-native protocol for Claude, OpenAI-compatible protocol for the rest):

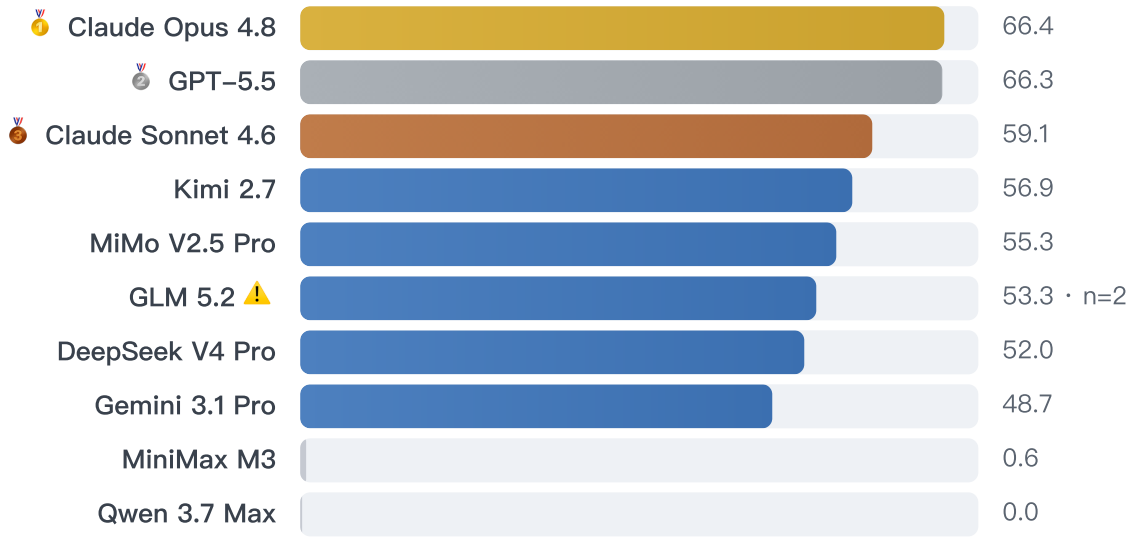
Model	Vendor	Protocol
Claude Opus 4.8	Anthropic	anthropic
Claude Sonnet 4.6	Anthropic	anthropic
GPT-5.5	OpenAI	openai
Gemini 3.1 Pro	Google	openai-compat
GLM 5.2	Z-AI	openai-compat
Kimi 2.7	Moonshot	openai-compat
DeepSeek V4 Pro	DeepSeek	openai-compat
MiMo V2.5 Pro	Xiaomi	openai-compat
MiniMax M3	MiniMax	openai-compat
Qwen 3.7 Max	Alibaba	openai-compat

---

4. Results

4.1 The leaderboard

Mean total score — frontier (Opus/GPT) vs. the open-weight cluster vs. the floor (1 block ≈ 2 points):



Mean total score (5 trials each) · axis 0–70 pt

Rank	Model	Total	σ	Objective	Judge	n
1	**Claude Opus 4.8**	**66.4**	8.1	50.3	82.5	5
2	**GPT-5.5**	**66.3**	8.2	54.8	77.8	5
3	Claude Sonnet 4.6	59.1	**2.7**	52.9	65.2	5
4	Kimi 2.7	56.9	10.4	47.7	66.0	5
5	MiMo V2.5 Pro	55.3	5.1	49.0	61.5	5
6	GLM 5.2 ⚠	53.3	2.3	51.6	55.0	**2**
7	DeepSeek V4 Pro	52.0	13.7	43.9	60.0	5
8	Gemini 3.1 Pro	48.7	6.4	47.1	50.2	5
9	MiniMax M3	0.6	0.9	1.3	0.0	5
10	Qwen 3.7 Max	0.0	0.0	0.0	0.0	5

⚠ **GLM 5.2 is infrastructure-censored** — its score reflects only the 2 of 11 attempts that its gateway route allowed to complete functionally. It is **not comparable** to the clean n=5 rows. See

§6.

4.2 Headline: a statistical tie at the top

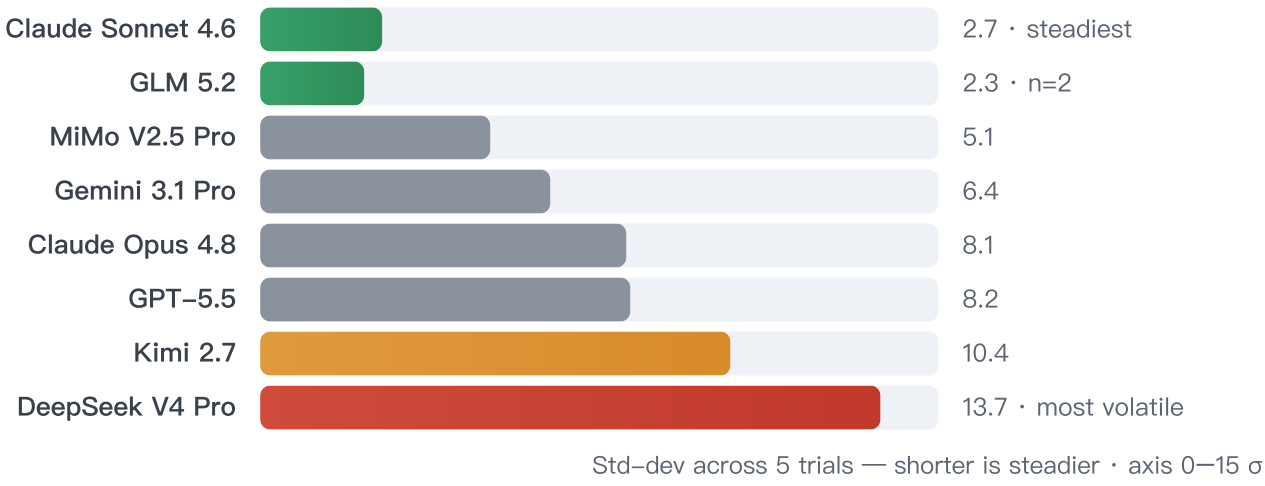
Opus 4.8 (66.4) and GPT-5.5 (66.3) are separated by **0.1**, deep inside their σ ≈ 8 — a tie. They reach it differently: **Opus wins on judge quality** (82.5 vs 77.8 — cleaner governance narration) while **GPT-5.5 wins on objective coverage** (54.8 vs 50.3 — it satisfies more manifest checks).

This is precisely the result a single trial cannot produce. In the single-shot pilot (run #7), GPT-5.5 *led* Opus (62.2 vs 61.1) and **Sonnet 4.6 scored 19.0** on one unlucky run where it lost state mid-workflow. Over five trials those artifacts wash out.

4.3 Consistency is a signal, not noise

Rank ≠ reliability. The standard deviation tells you whether you can trust a model to run the desk unattended on *any given day*:

Reliability — standard deviation across the five trials (**shorter is steadier**; 1 block ≈ 1 point of σ):



- **Sonnet 4.6 — σ 2.7:** the steadiest operator in the field. It never tops the board, but it never collapses either (range 54.8–61.4).
- **DeepSeek V4 Pro — σ 13.7:** the most volatile, swinging 43.9 → 69.6 across trials. Its mean (52.0) hides a model that is sometimes excellent and sometimes mediocre.
- **Opus / GPT — σ ≈ 8:** strong but genuinely variable at the top.

For an autonomy use-case, a high-mean/high-σ model and a slightly-lower-mean/low-σ model are *different products*. The arena surfaces that distinction.

4.4 The floor

MiniMax M3 (0.6) and **Qwen 3.7 Max (0.0)** sit at the floor across all five trials ($\sigma \approx 0$) — these zeros are reproducible and real (see §5).

---

5. Failure-mode taxonomy

A score near zero can mean two completely different things. The arena's trace log lets us separate them by reading what the model actually *emitted*.

Failure mode	Signature	Cause
Infrastructure wedge	run never completes; connection frozen, 0 output	gateway/route, **not the model**
Infrastructure blank	run completes but emits **0 characters** every step	gateway returned empty streams
Model stall	run completes, emits **coherent text**, but 0 useful tool calls	the **model** can't/won't operate

The discriminator is simple: **a model that fails *behaviorally* still talks.** A model whose *pipe* failed produces nothing.

Evidence from the transcripts:

Run	steps w/ text	total text	tool calls	what it emitted
GLM 5.2 — 0.0	**0/7**	**0 chars**	0	*nothing* (blank, yet ran 27 min)
GLM 5.2 — 54.9	7/7	15,639	22	*"Here's what the latest stored risk run says for the Control Desk Portfolio (id=2)…"*
MiniMax M3 — 0.0	7/7	6,394	0	malformed tool syntax as text: `[<tool_call>\ntask\n{…}`
Qwen 3.7 — 0.0	7/7	6,085	0	*"I'm sorry, but I'm encountering a persistent framework error… subagent invocation is failing…"*

- **MiniMax / Qwen** are *genuine model results*: the pipe worked (0 wedges in 5 trials each), the model emitted thousands of characters, but it could not operate the desk — MiniMax emits tool calls in a malformed syntax that never parses into real calls; Qwen narrates coherently then gives up on sub-agent delegation. **Their zeros are trustworthy.**
- **GLM's 0.0** is *not a model result*: zero characters across all seven steps, yet it ran the full 27 minutes (the same duration as its functional runs). That is the signature of the gateway returning empty completions on every generation.

---

6. GLM 5.2 — an infrastructure-censored result

GLM 5.2 deserves a separate treatment because its raw number badly misrepresents the model. Across **11 attempts**, GLM's `z-ai/glm-5.2` route behaved as follows:

Outcome	Count	Interpretation
Wedged / timed out	**8**	route froze; SIGKILLed at the hard cap — *infrastructure*
Completed, blank (0.0)	1	empty streams across all steps — *infrastructure*
Completed, functional	2	**54.9 and 51.7** — *the real model*

When the route held, GLM was a **competent desk operator**: 15,639 characters of coherent control-desk work and 22–34 tool calls per run, scoring in the mid-50s — on par with the DeepSeek/MiMo/Kimi cluster. The problem was never GLM's capability; it was that **~73% of attempts never got a usable connection** through the gateway during this run window.

We therefore report GLM two ways:

- **Functional estimate: 53.3** (mean of its 2 uncensored runs; **n=2, low-confidence**) — its placement *if you exclude infrastructure-censored data*.
- **Raw all-completions: 35.5** (mean of all 3 completed runs incl. the blank 0.0; n=3) — *understates* the model because the 0.0 is a pipe artifact, not a performance sample.

Either way, **GLM is the one model in run #8 that did not get a fair shake**, and its row is flagged accordingly. A clean re-run on a healthy route is required to place it with confidence. This is a *resource/infrastructure* limitation of the evaluation, explicitly distinct from the *model* limitations of MiniMax and Qwen.

---

7. Cost & token usage

> **Note.** This section reports what the trace could capture in run #8; **the full usage detail will be added in the future reports.**

7.1 What we could measure

The trace log records per-span token counts. For the two Anthropic-native models, ZenMux surfaced usage and we have **exact** figures (5 trials each):

Model	Prompt tok	Completion tok	Total	~ / match
Claude Sonnet 4.6	10,324,552	187,832	10,512,384	~2.10 M
Claude Opus 4.8	9,606,863	123,275	9,730,138	~1.95 M

The workload is **heavily prompt-dominated** (~98% input): the desk loads large context each turn — skill files, tool results, portfolio and risk data — while the model's own output (tool calls + short narration) is small (~25–40 K completion tokens per match).

> **Measurement gap (transparency).** ZenMux's OpenAI-compatible streaming path did **not** surface `usage` in the trace, so the eight openai-protocol models recorded zero tokens in-trace. We then tried ZenMux's **Management API** (`statistics/leaderboard` + `timeseries`), which *does* return real per-model tokens and cost — but **account-wide**, with no per-run or per-key filter. On this account that is unusable for run-level attribution: non-arena traffic on the sweep day (2026-06-26) ran **8–838× larger** than the arena's per-model volume (e.g. Opus 8.38 B account tokens vs 9.7 M measured for run #8 — a 838× gap; DeepSeek 848×; GLM 338×; even the lightest, MiniMax, 8.8×). The Management API's **per-generation** endpoint *can* give exact token/cost, but it requires the `x-generation-id` returned per call, which the inference layer did not capture for run #8. **Lesson for future runs:** record `x-generation-id` on every arena call to enable exact, run-isolated billing. Figures below for the eight openai-protocol models are therefore **estimates** using the measured ~2.0 M-prompt / 30 K-completion-per-match volume as a proxy (same task, same context), as a **no-cache upper bound**.

7.2 Price and estimated cost

ZenMux list price (USD per million tokens) and estimated cost per match:

Model	in $/M	out $/M	est. $/match †	measured $/match
GPT-5.5	5.00	30.00	~10.9	—
Claude Opus 4.8	5.00	25.00	~10.8	**10.22**
Claude Sonnet 4.6	3.00	15.00	~6.5	**6.76**
Gemini 3.1 Pro	2.00	12.00	~4.4	—
DeepSeek V4 Pro	0.435	0.87	~0.90	—
MiMo V2.5 Pro	0.435	0.87	~0.90	—
GLM 5.2	0.43	1.35	~0.90	—
Kimi 2.7	0.43	1.79	~0.90	—
Qwen 3.7 Max	0.43	1.29	~0.90	—
MiniMax M3	0.137	0.55	~0.29	—

† estimate; no-cache upper bound. With prompt caching (cache-read is ~10× cheaper: Opus $0.50/M, Sonnet $0.30/M), real cost on this prompt-heavy workload is **substantially lower** than shown.

7.3 Cost–performance

Score per dollar — the open-weight tier inverts the quality ranking on cost-efficiency (1 block ≈ 2.6 points/$):

Score per dollar (est.) — cost-efficiency inverts the quality ranking · axis 0–70 pts/$

The frontier (Opus/GPT, ~$11/match) buys the top ~7 points of score; the open-weight tier (~$0.90/match) delivers **~80–86 % of that score at ~1/12 the cost**:

- **Kimi 2.7** — 56.9 at ~$0.90 ≈ **63 points per dollar**, the cost-performance champion (86 % of Opus's score, ~12× cheaper).
- **MiMo V2.5 (55.3)** and **DeepSeek V4 (52.0)** sit just behind on the same cost tier.
- **Opus 4.8 / GPT-5.5** — ~6 points per dollar: best *absolute* quality and the cleanest governance narration, at a 10–12× premium.

In short: if you need the best unattended desk operator and cost is secondary, Opus or GPT-5.5. If you need 85 % of the quality at a tenth of the cost, the open-weight tier (Kimi/MiMo/DeepSeek) is the Pareto-efficient choice.

---

8. Limitations & threats to validity

- **One workflow.** Run #8 covers a single flagship task. Rankings may shift on other desk workflows; this is a depth-over-breadth design. We're also working on other long-workflow match designs, and will make them public soon.
- **Judge model in the loop.** Half the score comes from a GPT-5.5 judge, which may carry stylistic bias. (Notably it did *not* simply crown its own family — Opus outscored GPT-5.5 on the judge axis.)
- **Gateway variance.** All traffic shares one gateway (ZenMux). Route instability is real and asymmetric — it censored GLM 5.2 this run. We detect and flag it rather than letting it silently depress a score, but it remains a confound.
- **GLM under-sampled (n=2).** Its placement is low-confidence pending a clean re-run. Every other model reached the full five trials.
- **Token capture incomplete.** Exact usage exists only for the Anthropic models; others are proxy estimates (§7.1).

---

9. Reproducibility

- **Run:** `ArenaRun #8`, status `completed`, viewable on the `/arena` page; each model's per-trial detail is in its `score_breakdown.aggregate`.
- **Regime:** headless YOLO (`mode="yolo"`) — HITL auto-cleared, deferral tool withheld.
- **Engineering note:** an intermittent gateway wedge (any model, a blocking-sync freeze the in-process async timeout could not cancel) forced each match to run in an isolated

subprocess under a hard `SIGKILL` wall-clock guard, with checkpoint-to-disk so the 50-match sweep was resumable across restarts. This is why the failure surface is measured rather than hidden.

Appendix A — per-trial totals

Model	trials	n
Claude Opus 4.8	54.2, 71.7, 75.2, 67.7, 63.3	5
GPT–5.5	72.9, 57.1, 66.4, 59.2, 75.9	5
Claude Sonnet 4.6	59.9, 61.2, 58.0, 61.4, 54.8	5
Kimi 2.7	58.6, 71.2, 54.8, 57.8, 42.0	5
MiMo V2.5 Pro	56.7, 46.9, 57.8, 54.8, 60.2	5
GLM 5.2 (functional)	54.9, 51.7 *(+1 blank 0.0 censored, +8 wedged)*	2
DeepSeek V4 Pro	61.7, 45.9, 69.6, 47.6, 35.0	5
Gemini 3.1 Pro	37.6, 52.1, 53.6, 50.7, 49.5	5
MiniMax M3	0.0, 1.6, 1.6, 0.0, 0.0	5
Qwen 3.7 Max	0.0, 0.0, 0.0, 0.0, 0.0	5

Run #8 · OTC Desk Agent Arena · generated 2026-06-27.
