🏆 OTC Desk Agent Arena — Run #104

Grok 4.6 and DeepSeek V4 Pro, across all four golden workflows, two trials each.

2026-08-13 · run `#104` · 2 models × 4 workflows × 2 trials = 16 model-trials · objective-only scoring (jury off) · all 8 pairs `scored`, none `invalid`

Headline — a one-point tie, for opposite reasons

Model	OVR	GRD	ADH	SYN	EFF	PRC	CON	Objective
Grok 4.6	**80**	96	90	99	**12**	98	90	**96.3**
DeepSeek V4 Pro	79	93	89	84	**48**	93	70	90.9

Grok 4.6 is the better desk operator on **capability** — it beats DeepSeek V4 Pro on three of four workflows and leads the objective axis by 5.4 points. It finishes **one OVR point ahead**, because efficiency erases almost the entire lead.

This is the third consecutive board on which raw capability does not decide the ranking.

The Grok inversion did not get fixed — it got worse

Grok 4.5's defining Arena result was an inversion: the field's best objective score (98.4 on `trader-rfq-booking-day`, 62 of 63 checks) delivered in ~64 tool calls against a par of 35, for EFF 15 and OVR #5. The models that win raw capability are the ones you should not automate.

Grok 4.6 reproduces the capability and **deepens the cost**:

Workflow	par	Grok 4.6 calls	EFF	Objective
`risk-manager-control-day`	24	**243**	**0**	94.8
`trader-rfq-booking-day`	35	92	**0**	97.6
`high-board-portfolio-review-day`	24	47	13	94.3
`risk-limit-breach-day`	uncalibrated	32	34	98.7

EFF is golf-scored on a calibrated par: full marks at or under par, linear decay to zero at **2 × par**. Grok 4.6 is not merely past that cliff on the flagship, it is **five times** past it — 243 calls against a par of 24. Two of four workflows score a literal EFF 0 while scoring 94.8 and 97.6 on correctness.

The two flagship trials burned **355 and 131** calls. That 2.7× spread inside a single pair is invisible in the objective column; it is precisely what CON measures.

DeepSeek V4 Pro's counter-example

Its `risk-limit-breach-day` row is the best single result on the board and the shape a desk actually wants:

> **100.0 objective · 38/38 checks on both trials · 22 mean tool calls · OVR 90**

DeepSeek V4 Pro is the leaner operator everywhere (48 EFF vs 12) but pays for it in two places: **SYN 84 vs 99**, and **CON 70 vs 90** — driven almost entirely by `high-board-portfolio-review-day`, where its two trials scored 24/35 and 32/35 for **CON 10** and its worst card (OVR 61). Grok 4.6 is the steadier model despite being the more extravagant one.

Full board

Per-workflow cards

Model	Workflow	OVR	GRD	ADH	SYN	EFF	PRC	CON	Obj	Trials	Calls
DeepSeek V4 Pro	Portfolio Review	61	74	99	40	49	90	10	80.0	2	33
DeepSeek V4 Pro	Limit Breach	90	99	99	99	50	99	92	100.0	2	22
DeepSeek V4 Pro	Risk Control	76	99	68	99	24	90	86	88.5	2	41
DeepSeek V4 Pro	Trader RFQ	90	99	89	99	68	94	92	95.2	2	44
Grok 4.6	Portfolio Review	78	86	92	99	13	99	89	94.3	2	47
Grok 4.6	Limit Breach	88	99	99	99	34	96	96	98.7	2	32
Grok 4.6	Risk Control	74	99	74	99	0	99	79	94.8	2	243
Grok 4.6	Trader RFQ	81	99	94	99	0	96	96	97.6	2	92

Per-trial tool calls, for the record:

Pair	Trial 1	Trial 2
Grok 4.6 × Risk Control	355	131
Grok 4.6 × Trader RFQ	96	87
Grok 4.6 × Portfolio Review	53	41
Grok 4.6 × Limit Breach	37	27
DeepSeek V4 Pro × Risk Control	41	41
DeepSeek V4 Pro × Trader RFQ	43	46
DeepSeek V4 Pro × Portfolio Review	38	28
DeepSeek V4 Pro × Limit Breach	25	19

Method and caveats

This is a standalone board. It is deliberately **not** merged into runs #20 /

33 / #94 / #101. Two reasons, and both matter:

1. **DeepSeek upgraded the weights behind `deepseek/deepseek-v4-pro` on 2026-08-13.** The historic V4 Pro rows on earlier boards describe a *different model under the same id*. `merge_runs` groups by `(workflow_id, model_id)`, so merging would silently fold two different models into one aggregate.
2. **`risk-limit-breach-day`'s manifest changed after its last board** — 39 → 38 points on 2026-08-03, dropping the unroutable `read-risk-result` skill check. Run #101's rows are scored out of 39; this run's are out of 38.

Cross-board comparisons in this report are therefore made only against *published figures*, as context, never by folding rows together.

Coverage. Four golden workflows — the four that existed when the board launched. A fifth, `ops-settlement-day`, was in development on a branch that same day and is **not** represented here.

Configuration. `agent_recursion_limit` 100, the same as every prior board. Jury off (default since 2026-07-06), so scoring is deterministic-objective only. Contestants route through the zenmux channel. Grok 4.6 uses plain `provider: openai` with no `protocol: anthropic` override, verified by live smoke run #103 (100.0 on `risk-limit-breach-day`, 107 real tool spans).

Sampling and reasoning effort are NOT pinned — a standing caveat on EFF and CON, for this board and every prior one. Contestant models are constructed with only `model`, `api_key` and `base_url`: `model_factory` sets no `temperature`, no `max_tokens` and no `reasoning_effort`. `ArenaModel.default_config` (`temperature: 0, max_tokens: 4096`) reads like a pin but is **dead for contestants** — `arena_model_to_selection` returns only `{channel, provider, model}`, and the dict's only consumer, `arena/channel.py::build_zenmux_chat`, is imported nowhere in the live path. Every match from run #8 to #104 has therefore used vendor-default sampling, not greedy decoding.

Two consequences, stated rather than smoothed over:

- **CON contains sampling noise.** It measures trial dispersion, so a non-zero temperature contributes to it directly. Grok 4.6's 355-vs-131 flagship spread may be partly sampling rather than policy.
- **Reasoning effort is uncontrolled — but measured, and the EFF finding survives.** Both models accept `reasoning_effort` and respond to it: on a minimal prompt Grok 4.6 spends 94 reasoning tokens unset, 46 at `low`, 88 at `high`; DeepSeek V4 Pro 21 / 7 / 21. **Both therefore ran this board at essentially their `high` default.**

A controlled A/B then measured what effort does to *tool-call count*, the thing EFF actually scores — DeepSeek V4 Flash × `risk-limit-breach-day`, 2 trials per arm (runs **#107** `low` / **#108** `high`):

Arm	Tool calls	Mean	Objective
`low`	31, 32	31.5	92.1
`high`	26, 23	24.5	98.7

Higher effort produced **~22% fewer** calls, with non-overlapping ranges. Effort is therefore a real influence on EFF, in the direction that penalises *low* effort — so a model left at a low vendor default would be scored unfairly harshly.

It does not explain this board. A 22% swing cannot manufacture a 4× EFF gap: Grok 4.6's 243 calls against par 24 would still be ~190 at maximum-favourable effort, far past the 2 × par point where the golf curve reaches zero. And both contestants already sat at the lean end of the effect. **The EFF gap is not an effort artefact.**

Limits of that pilot, stated plainly: n=2 per arm against an unpinned temperature, one model on one workflow at the flash tier, arms run sequentially rather than interleaved. The objective difference rests on a single weak `low` trial (32/38 vs 38/38) and should be read as noise; the call-count separation is the signal.

Interruption. The board was stopped deliberately at 1/8 pairs ahead of a planned network outage and resumed into the same run at 17:30. This matters because `task._execute` catches per-trial exceptions and continues: under a sustained outage it would have swept the remaining pairs into `invalid` and marked the run `completed` — a quietly wrong board. The 1 completed pair was durable; only the in-flight pair's work was lost.

Ability cards

Ten cards in `cards/run104/` — two hero cards (the equal-weight mean across all four workflows) and eight per-workflow cards.

The cards are **fully generated** by GPT-Image-2, numbers included, and then **verified**: a vision model reads each finished PNG back and the card is rejected and re-rolled unless every digit matches the database and every meter shows exactly `round(value / 10)` lit segments out of ten.

10/10 verified in 13 renders — 3 re-rolls. Every rejection was an off-by-one pip at a rounding boundary (ADH 68 drawn as 6 lit, EFF 68 as 6, PRC 96 and CON 96 as 9), plus one meter drawn with 8 segment positions instead of 10. The image model **floors** `value / 10` where the spec rounds half-up, so the error is systematic at last digits ≥ 5 rather than random — which is exactly why a spot-check of a sample would have under-detected it, and why the gate checks meter geometry and not only the digits.

Stat meters are ten countable segments rather than continuous bars for a measured reason: across ten calibration renders, GPT-Image-2 reproduced quoted digits perfectly (42/42) but *interpolated* continuous lengths — an EFF of 36 drew at 42–60% of its track and flipped rank against its neighbour between two samples of one prompt. A countable instruction travels the same faithful path as the digits.
