SPECIMEN · ILLUSTRATIVE DATA

LUCIEN ALLEN

PROFIT LEAKAGE AUDIT

Performance & Profit-Leakage Audit

An end-to-end diagnostic of execution performance, profit attribution, latency, fill quality, and markout — culminating in a dollar-attributed profit-leakage audit, delivered from the artifacts you already produce.

$995 per system · 48-hour turnaround

Prepared for **Acme Capital** (illustrative sample)

Strategy **Cross-asset market-making · ES/NQ & ZN/ZB futures, cash USTs, SPX options · CME Globex (futures) + EDGA/EDGX (options)**

Period **2026-06-08 → 2026-06-21 (14 trading days)**

Method **Artifact-based · no production / repo / credential access · under mutual NDA**

Prepared by **Lucien Allen Corp** · lucienallen.com

Executive summary

Over 14 trading days the book generated **$368.0k** gross edge and captured **$312.8k** net (capture ratio **85.0%**, Sharpe 6.0 on ~$10M deployed, profit factor 2.6). We attribute **$55.2k of profit leakage — 15.0% of gross, ~$1.0M annualized** — to five sources, most of it **introduced by two releases** and therefore preventable at the release gate.

GROSS EDGE	NET CAPTURED	PROFIT LEAKAGE	% OF GROSS	RECOVERABLE (EST.)	ANNUALIZED
$368.0k	$312.8k	$55.2k	15.0%	$30–39k	~$1.0M

SEVEN-DIMENSION SCORECARD

B–
OVERALL 74/100

Read: the system is fundamentally sound (B). Weakness concentrates in **latency discipline (61)**, **delivery & deployment health (66)**, and **fill quality (68)** — the three dimensions that, together, account for ~70% of the leakage. Findings are marked **[measured]** (from your data) vs **[attested]** (from the intake questionnaire) throughout.

Strategy performance & risk

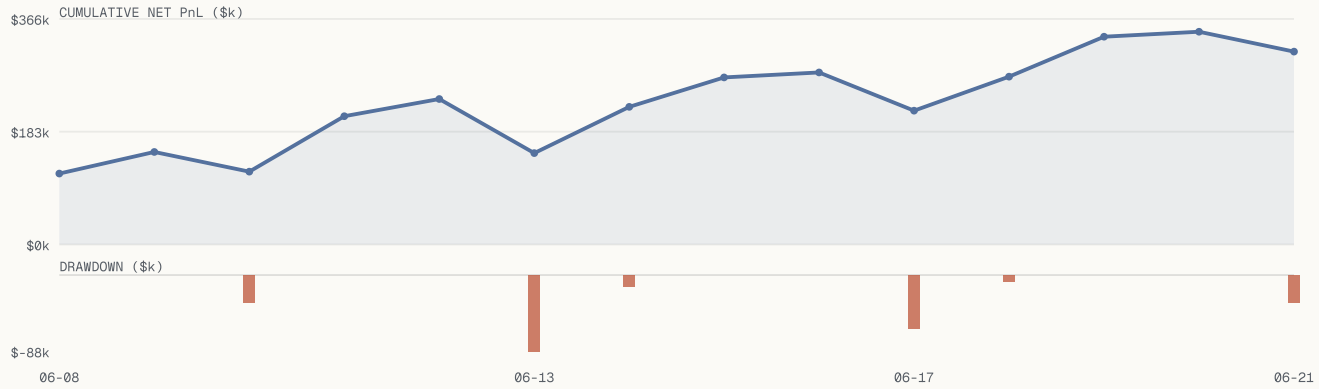
ACCOUNT METRICS

Net PnL	$312.8k	Fills	2.10M
Gross edge	$368.0k	Daily PnL — mean / median	$22.3k / $31.5k
Risk capital (avg deployed)	$10.0M	Daily PnL — std dev	$59.0k
Return on capital (ann.)	~56%	Sharpe (ann., on capital)	6.0
Profit factor	2.6	Sortino (ann.)	7.4
Capture ratio (net/gross)	85.0%	Max drawdown / capital	-0.9%
Net bps / notional	2.20 bps	Win rate — daily / per-fill	71% / 65%
PnL per $1M traded	$220	Realized edge — mean / median	+0.9 / +1.4 bps
Notional traded	$1.42B	Realized edge — std / skew	6.2 bps / -1.10

PER-SYMBOL BREAKDOWN

SYMBOL	NET PNL $K	BPS/NOTL	MARKOUT 5M	MAKER	HIT RATE	EDGE CAPTURE
ES	158.2	3.1	+0.6	88%	73%	92%
ZN	121.4	2.0	+0.1	86%	71%	88%
SPX ODTE	33.2	0.9	-3.1	83%	67%	57%
Total	**312.8**	**2.2**	**–**	**86%**	**71%**	**85%**

CUMULATIVE NET PNL & DRAWDOWN — 14-DAY WINDOW



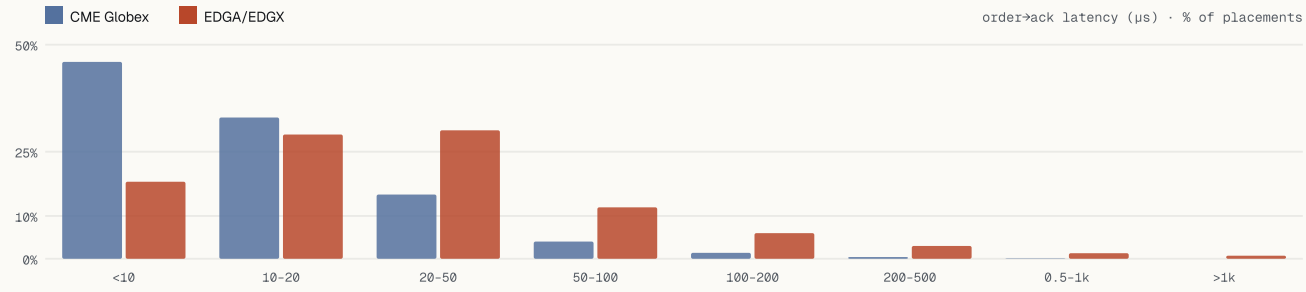
Recovery profile: returns and Sharpe are computed on **~$10.0M average deployed risk capital** (≥ the $6.9M peak gross inventory). Max drawdown **–$88.0k (–0.9% of capital)** on the 06-13 release; a second **–$62k** dip followed 06-17 — the two worst days are both release-driven. Recovery factor **3.6×**, typically recovers within a day. The drawdowns track **execution regressions, not market beta**.

Latency analysis

PER-STAGE LATENCY PERCENTILES (µs) — MD INGRESS & SIGNAL ON FPGA

STAGE	P50	P90	P95	P99	P99.9	MAX	MEAN	STD	P99/P50
Tick → order · FPGA	0.31	0.42	0.48	0.61	1.2	3.4	0.34	0.09	2.0×
Order → ack · CME Globex	8.2	16.0	24.1	61.5	210	880	11.4	9.1	7.5×
Order → ack · EDGA/EDGX	14.6	41.0	71.0	188.0	1840	2310	26.8	61.2	12.9×
Order → fill	42.0	120.0	180.0	520.0	2100	3400	71.0	96.0	12.4×
Cancel → ack	9.1	19.0	28.0	88.0	240	410	12.1	11.0	9.7×

ORDER→ACK LATENCY DISTRIBUTION (µs) — CME GLOBEX VS EDGA/EDGX



PIPELINE JITTER (NANOSECONDS)

STAGE	LAYER	JITTER	STATE
MD ingress → decode	FPGA	12 ns	clean
Book build + signal	FPGA	21 ns	clean
Order encode → NIC egress	FPGA	9 ns	clean
Risk gate + cancel path	host (C++)	640 ns	outlier
Switch fabric · L1 x-connect	Arista 7130	<0.1 ns	floor

jitter = cycle-to-cycle std at each stage. The FPGA tick-to-trade path (MD ingress → decode → signal → egress) holds **<25 ns**; the L1 fabric **<0.1 ns** (Arista 7130 / Cisco 3550-F class). The lone outlier is the **non-FPGA host risk/cancel path**.

READ

The data path is clean; the cancel path is not. MD ingress, book-build and signal run on **FPGA at <25 ns** jitter and the L1 fabric at **<0.1 ns** — deterministic. But the **host risk/cancel path is off-FPGA** and jitters at **640 ns**, 30–70× the data path. That variability lands on EDGA/EDGX: order→ack p99 **188 µs vs 61.5 µs** (CME), p99/p50 **12.9×** vs 7.5×, 4.1% over the budget — a **host-stall signature** (131 events), not a venue or wire problem (matching engines clear in tens of µs — CME ~52 µs, BrokerTec <200 µs, Dealerweb ~30 µs). At quoted size → **$9.9k** latency slippage.

Baselines for context: CME ~52 µs median inbound / sub-150 µs match (CME; Markets Media); Cboe options µs-level (RTLM, medians portal-gated); BrokerTec <200 µs, Dealerweb ~30 µs; FPGA tick-to-trade & L1 fabric jitter per Arista 7130 / Cisco 3550-F-class figures. Client execution figures illustrative.

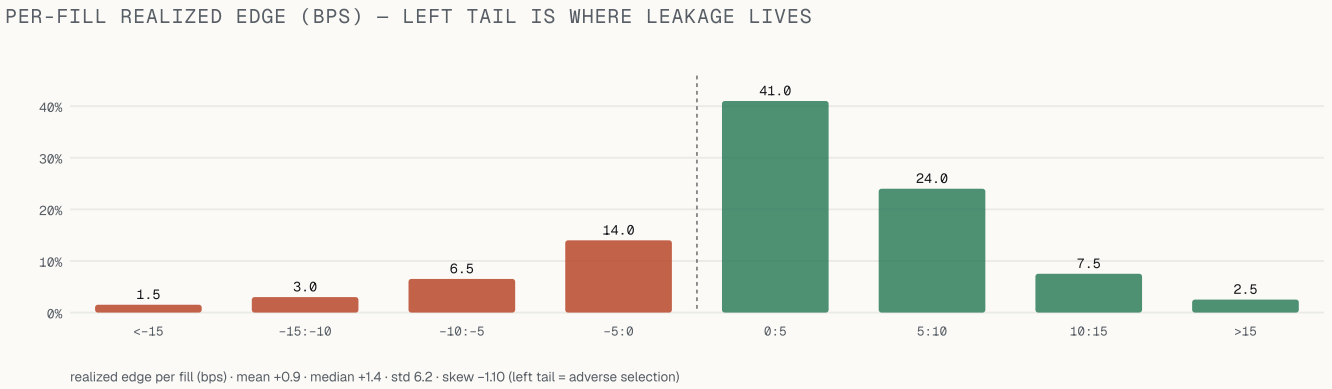
Fill attribution, quality & markout

FILL-QUALITY METRICS

Maker ratio	86%
Aggress share	14% (target 6%)
Hit rate	71%
Fill rate	94%
Order-to-trade	3.2 : 1
Avg slippage	1.4 bps
Quote uptime	98.6%
Avg lot size	2.4 contracts

MARKOUT SURFACE — BPS AFTER FILL

REALIZED-EDGE FREQUENCY DISTRIBUTION



Read: the edge distribution is positively centered (median +1.4 bps) but **left-skewed (−1.10) — 11% of fills clear below −5 bps**, concentrated in SPX 0DTE, whose markout is negative at every horizon (−3.1 bps @5m) and whose edge-capture is just **57%** vs 92% for ES. The book is quoting into informed flow on one leg; order-to-trade at 3.2:1 (23% over the CME messaging budget) discards queue position faster than it's earned.

Order lifecycle & reconciliation

Aggregates hide behaviour. We close the loop at the **order level**: a stratified sample of **2,140 transactions** (standard 500–3,000) reconstructed from **NEW** through every terminal state, hedge and markout — so **every order is accounted for** and the sample reconciles to the book.

ORDERS TRACED	ACCOUNTED FOR	CANCEL/REPLACE CHURN	ORDER-TO-TRADE	SAMPLE CAPTURE
2,140	100%	65.5%	3.2 : 1	85.0%

ORDER STATE FUNNEL

STATE	ORDERS	%
Orders placed (NEW)	2,140	100.0%
Acknowledged	2,131	99.6%
Cancelled / replaced	1,402	65.5%
Filled (full or partial)	668	31.2%
Stuck partial — never COMPLETE	23	1.1%
Expired / IOC-miss	38	1.8%
Rejected (risk / exchange)	9	0.4%

Acked = cancelled + filled + stuck + expired (2,131); + rejected = placed (2,140). Nothing unaccounted.

PNL RECONCILIATION — SAMPLE → BOOK

Gross edge (sample)	$6.35k
– Adverse selection	-$0.21k
– Latency slippage	-$0.17k
– Aggressing fees	-$0.16k
– Order-to-trade / messaging	-$0.13k
– Missed fills / unhedged	-$0.28k
Net captured (sample)	$5.40k
Capture — sample / book	85.0% / 85.0%

Sample capture 85.0% vs book 85.0% — the traced orders are representative; each leakage dollar maps to an order behaviour.

REPRESENTATIVE ORDER TRACES — FULL LIFECYCLE

ORDER	OUTCOME	LIFECYCLE (NEW → TERMINAL → HEDGE)	MARKOUT
#A714	**Cancel · queue-pos loss**	NEW → ACK +8.2 µs → working → CANCEL +210 µs (reprice churn)	–
#A902	**Fill · clean**	NEW → ACK +7.9 µs → FILL +41 µs → HEDGE +132 µs	+1.4 bps
#A958	**Fill · loaded (81% of limit)**	NEW → ACK +14.6 µs → FILL +96 µs → HEDGE +540 µs	-3.9 bps
#A977	**Stuck partial**	NEW → ACK +9 µs → PARTIAL 40% → never COMPLETE → leg unhedged 1.4 s	naked

The loop closes: 2,140 orders in, 2,140 accounted for; **65.5% cancel/replace** before filling (order-to-trade 3.2:1), and the orders that *do* fill skew toward the moments you're adversely selected while loaded. Per-order reconstruction is what lets every leakage dollar trace to a **specific order behaviour** — churned queue position, a stuck partial, a fill taken while full — not a vague aggregate.

Order & fill integrity

Closing the loop means the order/fill stream must itself be sound. We audit the message layer — cancels, reject storms, ack delays, stuck acks, ID uniqueness, fill-ID propagation and partial-fill completion — the failures that quietly corrupt attribution and leave risk naked.

CANCELS (SAMPLE)	CANCEL-ACK RATE	STUCK ACKS	REJECT STORMS	CLORDID COLLISIONS
1,402	99.4%	6	3	2

CANCEL & ACK HEALTH

Cancels (sample)	1,402
Cancel-ack rate	99.4%
Cancel-reject rate	0.6%
Cancel→ack p50 / p99	9.1 / 88 µs
Ack delays > 50 µs budget	4.1%
Stuck acks (orphaned)	6 (0.3%)

REJECT STORMS

Reject-storm episodes (≥5/100 ms)	3
Peak burst	14 rej / 80 ms
Cause · risk-gate throttle	2
Cause · CME MEP msg-rate	1
Total rejects (sample)	9 (0.4%)
Storm-driven re-quote cascade	1

ORDER-ID UNIQUENESS & FILL-ID PROPAGATION

clOrdID uniqueness	2 dup / 2.10M
Collision cause	post-reconnect reuse
execID / fillID propagation	99.97%
Propagation gaps	11 (feed-reconnect)
Ambiguous fill attribution	2 (patched by seq)
Out-of-order execIDs	4

PARTIAL-FILL ANALYSIS

Partial fills	56 (8.4% of filled)
Avg fill ratio on partials	47%
Partials → completed	33 (59%)
Partials stuck (never COMPLETE)	23 (41%)
Partials left unhedged	1.8% (naked)
Avg partial → complete	3.1 s

Three failure modes bite. (1) Message storms — 6 stuck acks and 3 reject storms (risk-gate throttle + CME MEP rate-limit) trigger re-quote cascades that burn the messaging budget. **(2) Identity gaps** — 2 clOrdID collisions after reconnect and 11 fill-ID propagation gaps create **ambiguous fill attribution**, a reconciliation and risk hazard, not cosmetics. **(3) Partial-fill rot** — 41% of partials never complete, leaving **1.8% of legs naked**. Fixes: reconnect-safe monotonic clOrdID, idempotent execID handling, and a partial-fill timeout → forced complete/hedge.

Inventory risk, hedging & fill timing

For a market maker, edge is conditional on **inventory**: quoting flat is profitable; quoting while loaded is where adverse selection bites. We reconstruct the intraday exposure path and condition markout, hedge time and fill time on it.

MAX NET EXPOSURE	MAX GROSS INVENTORY	PEAK LIMIT USE	TIME >75% LIMIT	PEAK-EXPOSURE DWELL	INTRADAY 99% VAR
$4.2M	$6.9M	84%	41 min	4.2 min	$128k

MARKOUT BY NET-INVENTORY LEVEL — BPS AFTER FILL

	1m	5m	30m	edge capt.
<25% of limit	+0.9	+0.8	+0.6	79%
25–50% of limit	+0.5	+0.2	-0.4	68%
50–75% of limit	-0.3	-1.6	-2.4	54%
>75% of limit	-1.9	-3.9	-5.6	38%

markout (bps) after fill by net-inventory bucket · the book is picked off precisely when most loaded

TIME TO HEDGE

Time-to-hedge p50	120 µs
p90	540 µs
p99	1.8 ms
Episodes > 500 µs	9%
Unhedged-leg windows	1.8%

FILL TIME (ORDER → FILL)

Order→fill p50	42 µs
p90	120 µs
p99	520 µs
ES / ZN median	38 / 51 µs
SPX ODTE median	96 µs

READ

You're adversely selected exactly when you're most loaded. Markout is positive only while flat-to-light (**+0.8 bps @5m** below 25% of limit) and **craters to −3.9 bps @5m above 75%**, where edge-capture falls to **38%**. The book spends **41 min/day** in that zone — and is slowest to hedge precisely then (time-to-hedge p90 **540 µs**, p99 1.8 ms; 1.8% of legs left open). This is **inventory-driven** toxicity, not symbol-driven alone: the same fill is fine when you're flat and ruinous when you're full.

> **Fix:** skew or withdraw quotes as inventory builds, cap exposure dwell, and pre-stage a co-located hedge. Most of the >75% bucket is recoverable — it overlaps the **$16.5k** missed-fills / unhedged-leg and **$12.5k** toxic-flow buckets on the waterfall.

Delivery & deployment health → execution quality

How a team *ships* is an execution-quality input: a release can introduce a latency regression or a bad quoting parameter that bleeds PnL silently. We correlate delivery metrics against the execution-quality timeline (from a git-history export and deploy/CI logs — no repository or production access).

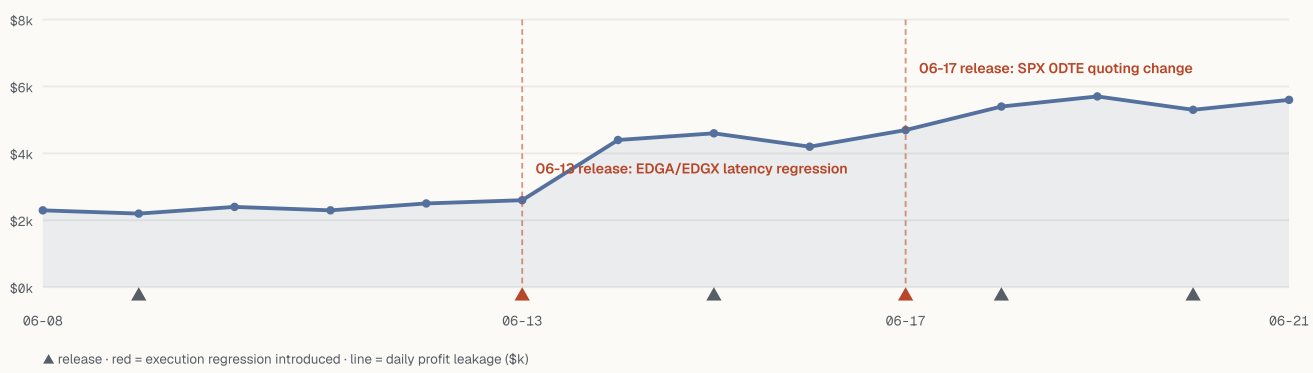
DELIVERY METRICS (DORA-STYLE)

METRIC	VALUE	TIER
Deploy frequency	0.6 / day	–
Change-failure rate	22%	Low (DORA)
Deploy success	78%	–
MTTR	4.7 h	Low-Med
Lead time (median)	3.2 d	Med
Commit velocity	142 commits	Bursty
Rollbacks	1	–

THE LINK

Most leakage was introduced by releases — it is not a baseline. Daily leakage sat near $2–3k until the **06-13 release** (EDGA/EDGX latency regression, ≈+$2k/day) and stepped again after the **06-17 release** (SPX 0DTE quoting change). Change-failure rate is **22%**; commits cluster pre-deploy — a large-batch failure signature.

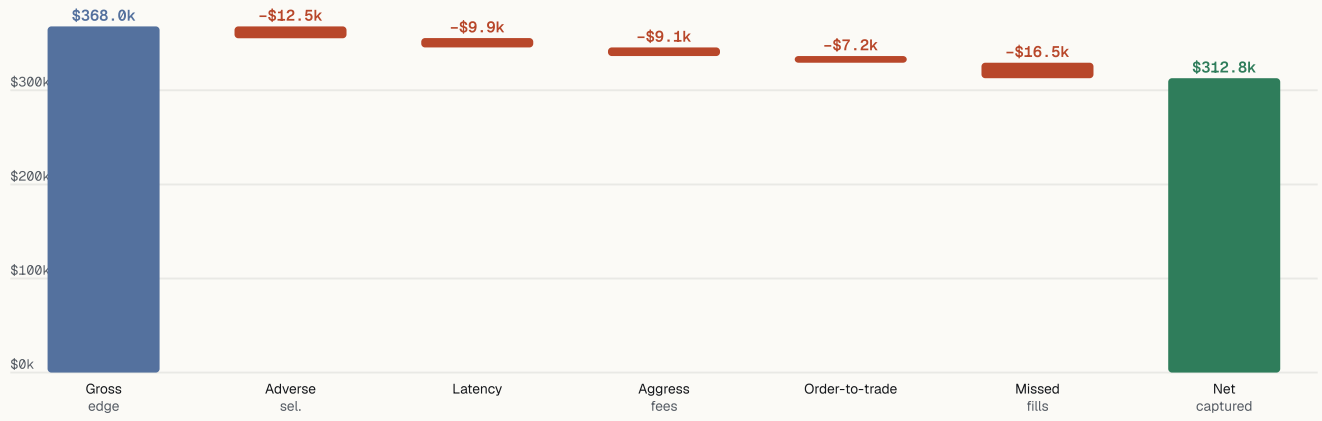
DAILY PROFIT LEAKAGE ($K) VS RELEASE EVENTS



Why it matters: the team has deployment *governance* but no **execution-quality gate** — nothing catches a markout or latency regression at release, so a clean-looking deploy still bleeds PnL. The bulk of the $55.2k is **preventable at the release gate**.

Profit-leakage audit

PROFIT-LEAKAGE WATERFALL ($K, 14-DAY WINDOW)



LEAKAGE BY SOURCE

SOURCE	$K	% GROSS
Adverse selection (toxic SPX 0DTE flow)	-12.5	3.4%
Latency slippage (EDGA/EDGX ack tail)	-9.9	2.7%
Excess aggressing fees (14% vs 6% target)	-9.1	2.5%
Order-to-trade / messaging penalty	-7.2	2.0%
Missed fills / unhedged leg	-16.5	4.5%
Total leakage	**-55.2**	**15.0%**

RECOVERY

About 70% of the leakage ($38.9k) is latency + toxic-flow + missed fills — quotes arriving late, into bad flow, or never completing. None requires a rewrite.

Identified (14d)	$55.2k
Annualized	~$1.0M
Est. addressable	55–70%
Recoverable / yr	**$0.55–0.70M**

Findings & remediation register

Each finding: evidence, severity, quantified impact (period / annualized), recommended action. Ordered by severity then dollar impact.

SEV	FINDING & EVIDENCE	DIMENSION	IMPACT	ACTION
HIGH	**SPX 0DTE quoted too tight into toxic flow** Markout −3.1 bps @5m, edge-capture 57% vs 92% ES; left-tail fills 11%	Fill quality	$12.5k / ~$225k yr	Widen 0DTE base spread + skew; short-horizon toxicity filter
HIGH	**Host risk/cancel path jitters 640 ns (30–70× the FPGA data path); EDGA/EDGX ack p99 3.1× CME** p99 188 vs 61.5 µs; p99/p50 12.9×; 4.1% over budget; host-stall signature (off-FPGA)	Latency	$9.9k / ~$178k yr	Move risk-gate/cancel onto FPGA or pin to isolated core; cap in-flight; bypass host scheduler
HIGH	**Execution regresses after releases — no execution-quality gate** 22% change-failure; EDGA/EDGX latency leakage began at 06-13 release; pre-deploy commit bursts	Delivery	≤~$22k preventable	Post-deploy markout/latency canary; auto-rollback; smaller batches
HIGH	**1.8% of futures fills left with an unhedged option leg** Episode reconstruction: reach PARTIALLY_HEDGED, never COMPLETE	Risk	Naked delta	Leg-timeout → forced hedge or flatten
MED	**Aggress share 14% vs 6% target — crossing spread on reprice** Excess aggressing fees on ~8% of volume	Execution	$9.1k / ~$164k yr	Slow reprice cadence; widen joins; queue-aware reprice
MED	**Order-to-trade 3.2:1; 23% exceed CME messaging budget** Queue-position loss + CME messaging surcharge	Execution	$7.2k / ~$130k yr	Replace-in-place; debounce micro-moves
MED	**Time-to-hedge p90 540 µs; option/future leg lags >500 µs on 9% of episodes** Markout craters at high inventory: −3.9 bps @5m above 75% of limit vs +0.8 bps below 25%	Risk	Delta drift / inventory	Co-locate & pre-stage hedge; inventory-aware quoting at high exposure
MED	**clOrdID reuse after reconnect (2 collisions / 2.10M) + 11 fill-ID propagation gaps** Ambiguous fill attribution on 2 orders; out-of-order execIDs (4)	Risk	Attribution / recon hazard	Reconnect-safe monotonic clOrdID; idempotent execID handling; seq-gap alarm
LOW	**Recurring CME / EDGA / EDGX feed-reconnect failures ~3×/day** Market-data staleness windows; source of the fill-ID propagation gaps	Operations	Staleness risk	Feed-health alert; redundant endpoint

> The **Sprint** tier turns this register into a costed remediation roadmap with drill design; the **Retainer** tier monitors these metrics continuously and re-runs the audit on a cadence.

How it works

WHAT YOU PROVIDE

- A 5-minute questionnaire (firm context, the trigger, a one-line architecture note)
- 3–7 days of standard trade execution logs (**CSV, JSON, or parquet**), latency telemetry, and order-to-fill records — redact what you choose
- Optional: a git-history export + deploy/CI logs (unlocks the delivery→execution analysis)
- A short architecture description — no source code

WHAT YOU RECEIVE

- This report: scorecard, performance & risk statistics, latency distribution analysis (benchmarked to published venue and switch-fabric floors), fill attribution & markout, order-lifecycle reconciliation (500–3,000 transactions traced end-to-end), inventory-risk & hedge timing, a quantified profit-leakage audit, and a prioritized findings register
- A recorded readout — **results within 48 hours of artifact receipt**

WHAT WE NEVER SEE

- ✕ Your production environment, repository, dashboards, or cloud
- ✕ Any API keys, credentials, or trading access
- ✕ Other clients' data — each engagement is isolated, under a mutual NDA

> **$995 per system, fixed scope** — one system = one independent trading strategy or target venue link. You choose what to share; every engagement runs under a mutual NDA, confidential and secure. The same statistical pipeline run on production books, applied to your artifacts — then **Sprint** (costed roadmap + drills) and **Retainer** (continuous monitoring) when you want to go deeper.

Start a diagnostic → lucienallen.com · contact@lucienallen.com

Beyond the audit — plugs for the leaks

The $995 diagnostic identifies **exactly where the edge is leaking** — to the dollar, ranked by impact. Here is how we help you **capture it**. Both tiers build directly on the findings register (page 10) — no new intake, no re-discovery.

SPRINT TIER · FIXED-SCOPE ENGAGEMENT

From findings to a costed roadmap

We turn this register into a step-by-step, costed remediation roadmap — the exact architectural drills and logic adjustments to plug each leak, sequenced by dollar impact.

- Re-spec the **ODTE base spread + skew** and ship a short-horizon toxicity filter — with a backtested before/after on markout and edge-capture.
- Move the **risk/cancel gate onto FPGA** (or pin it to an isolated, scheduler-free core) and cap in-flight on the EDGA/EDGX path — to a stated p99 latency target.
- Wire a **post-deploy markout/latency canary** with auto-rollback, so a release can't silently regress execution.
- Every drill costed, owner-assigned, and tied to the dollars it recovers.

RETAINER TIER · CONTINUOUS

Keep the leak plugged

We connect to your artifact-output pipeline on a rolling cadence and re-run the audit continuously — so a fresh release never silently introduces another ~$2k/day regression.

- Rolling re-runs on each release window; regression alarms on markout, latency tails, and order-to-trade.
- A standing, independent **execution-quality gate** — the control the book is missing today.
- A written readout each cadence for your risk committee or allocators.
- Same zero-access posture: artifacts only, under the mutual NDA.

HOW THE TIERS MAP TO THIS REPORT

FROM THE REGISTER (P.10)	IDENTIFIED	WHAT THE SPRINT PLUGS IT WITH
Toxic 0DTE flow	$12.5k / ~$225k yr	0DTE spread + skew re-spec; short-horizon toxicity filter
Host cancel-path jitter (off-FPGA)	$9.9k / ~$178k yr	Risk/cancel gate → FPGA or isolated core; in-flight cap to a p99 target
No execution-quality gate	≤~$22k preventable	Post-deploy markout/latency canary + auto-rollback
Unhedged option leg	Naked delta	Leg-timeout state machine → forced hedge or flatten

> **The diagnostic finds the leak. The Sprint plugs it. The Retainer keeps it plugged.** Scope and pricing for both tiers are quoted from this report's findings — ask for them in your readout.

Continue → lucienallen.com · contact@lucienallen.com