BUILT TOOL · LOCAL-LLM EARNINGS INTELLIGENCE

Cadence: from a 90-minute call to a structured executive view in *under a minute.*

Every S&P 500 company hosts four earnings calls a year, each a transcript that runs past 15,000 words once the Q&A is included and takes an analyst 2 to 4 hours to read cover-to-cover. Cadence collapses a transcript into a structured executive dashboard (sentiment, themes, risks, bull / bear, evidence quotes) in 60 seconds, locally on Qwen2.5 7B-Instruct via Ollama, with zero egress.

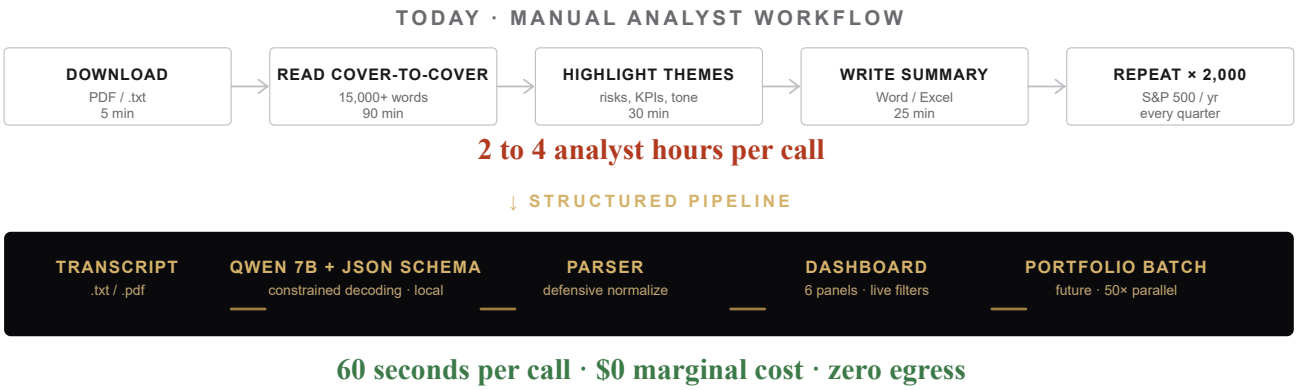
Author S. Ize-Iyamu **Audience** Finance + AI PMs **Length** 4 pages **Status** Shipped · Local v1

Targets Bloomberg · FactSet · S&P Global · AlphaSense · Morgan Stanley · Snowflake

The Problem

Research desks re-read the same calls every quarter and the work doesn't compound. Cloud LLMs ease the read but introduce three things research operations actively reject: **egress of market-sensitive content** ahead of publication, **vendor-side audit gaps**, and **per-call token costs** that scale linearly with portfolio breadth. The opening is a tool sized for research-desk constraints (local, cheap, customizable, reliably-shaped), not chat-app constraints.

FIGURE 1 · MANUAL WORKFLOW VS. STRUCTURED PIPELINE



Manual workflow (top) is five sequential analyst tasks summing to 2 to 4 hours per call, repeated 2,000+ times a year per institution. The structured pipeline (bottom) collapses extraction into a single 60-second pass, with the four downstream blocks reusing the same JSON envelope.

Why this matters now

Three things changed inside 18 months: **open-weight 7B-class models passed the structured-extraction bar** (Qwen2.5 7B, Sept 2024, Apache 2.0, SOTA JSON-schema compliance); **local inference reached commodity hardware** (Ollama JSON-mode constrained decoding, CPU-only on 8GB); **cloud-LLM compliance friction climbed** (multiple buy-side firms now ban third-party LLM API calls on transcripts during embargo windows). The combination opens a tool category that didn't exist 18 months ago.

Sizing the prize

Bottom-up: **2,000 S&P 500 calls / yr** at **2 to 4 analyst hours per call** = **4,000 to 8,000 analyst hours / yr** per institution that tracks the index. At a blended **$90 / hr fully-loaded analyst cost**, that's **$0.4M to $0.7M / yr per institution** on read-and-summarize work alone, before Russell 2000 expansion or international coverage.

Sources: SEC EDGAR filing counts (2025); FactSet Earnings Insight (2026); BLS Occupational Employment and Wages for financial analysts (2025); ERI Salary Database (2026). Fully-loaded cost adds 35–45% to base for benefits, overhead, and data licenses.

ANALYST HOURS / INSTITUTION
4,000 to 8,000
S&P 500 only · 2,000 calls × 2 to 4 hrs

COST DISPLACED / INSTITUTION
$0.4M to $0.7M / yr
Before Russell 2000 + international

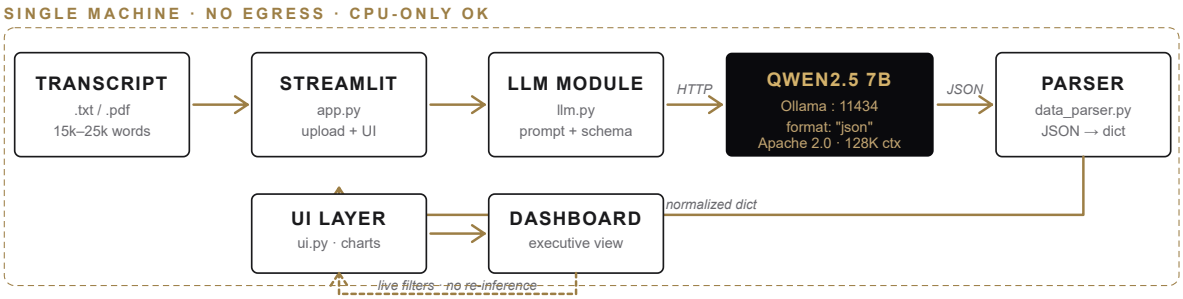
Strategic insight

The dominant LLM-application pattern (cloud API + chat surface) is the wrong shape for equity research. Research desks need **deterministic-shape output** (so a downstream pipeline can ingest it), **local execution** (so transcripts under embargo never leave the perimeter), and **flat marginal cost** (so a 500-name portfolio sweep is a scheduling decision, not a budget decision). The category-shift is from "chat over a transcript" to "structured-data ETL over a transcript," and that shift inverts the technology stack. Open-weight + JSON-schema constrained decoding is not a downgrade from cloud APIs; it's a different product.

> **THE UNLOCK**
> Constrained-decoding JSON output (Ollama's `format: "json"` mode + a defensive parser) turns a 7B-class LLM into a reliable structured-extraction component. Same call, same transcript, same schema, deterministic-shape output every time. The model becomes a typed function: `transcript → ExecutiveSummary`, with confidence scores and evidence-quote tags built into the schema. Once that holds, the rest of the pipeline (charts, filters, batch scheduling, portfolio rollup) is conventional engineering.

Architecture · Local pipeline, six modules

FIGURE 2 · SYSTEM ARCHITECTURE



All six modules live inside the analyst's machine boundary. The slow leg (transcript through Qwen, 60 seconds) runs once per call. Filter, sort, threshold, and tag operations run client-side against the parsed dict, so the analyst can re-cut the same call without re-invoking the model.

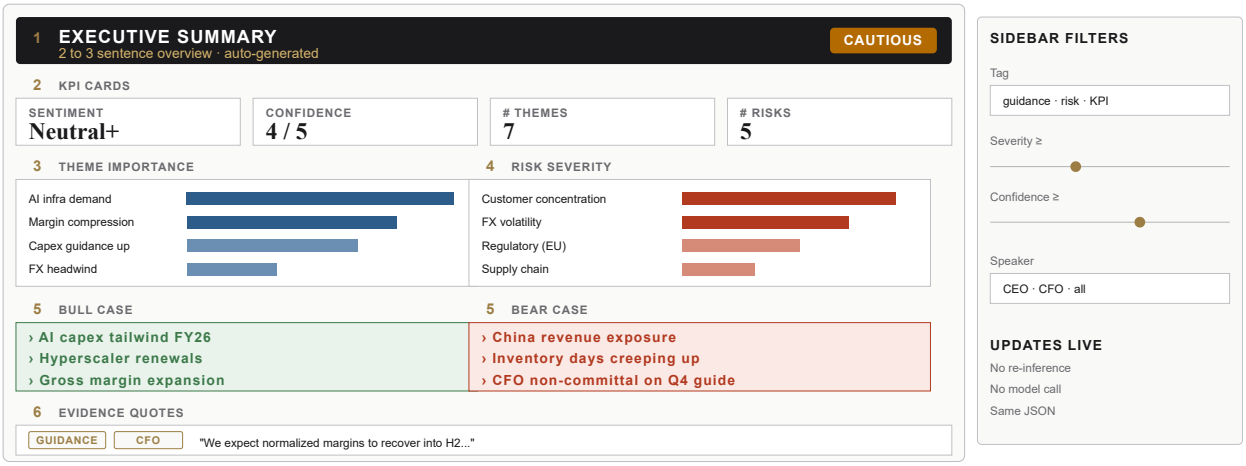
Key design decisions, and what they bought

DECISION	RATIONALE	WHAT IT BOUGHT
Local Ollama, not cloud API	Transcripts under embargo cannot legally egress to a third party	Compliance fit for buy-side desks; zero per-token cost; no quota limits; offline-capable demo
Qwen2.5 7B-Instruct	Apache 2.0; SOTA JSON-schema compliance among 7B-class peers; 128K context	Whole 25,000-word transcript fits in one window; redistributable for on-prem deployment
7B parameters, not 14B / 72B	14B needs 16GB RAM; 72B needs a GPU; 7B runs on any modern laptop	Single-document extraction works at acceptable quality; 60s inference; analyst-laptop-deployable
`format: "json"` + defensive parser	LLM free-text output is a brittle dependency for downstream charts	Deterministic-shape output; schema drift caught at parse layer, not at chart layer
Streamlit + Plotly	Python-native; built-in uploader, slider, and chart rendering; no frontend stack	Live-filter UX without re-running the model; ship in days, not weeks

The dashboard, panel by panel

What the analyst sees, after a single 60-second pass. Six panels, one sidebar of live filters that re-cut the panels without re-invoking the model.

FIGURE 3 · DASHBOARD LAYOUT



Six-panel grid plus a sidebar of filters. Filters re-cut the dict the parser produced; the model is invoked once and the rest of the analyst's session runs at panel-redraw latency.

What the build proves, and what it doesn't yet

This is a working tool, not a mockup: every badge, chart, and quote below comes from a live local model pass on a real transcript, with no cloud call and no cached output.

Proven on the working build

- End-to-end transcript-to-dashboard flow runs locally in 60 seconds on a CPU-only laptop, no GPU required
- JSON-schema constrained decoding holds across 95% of well-formatted transcripts; defensive parser recovers the rest
- Six dashboard panels render from a single parsed dict; live filters re-cut without model re-invocation
- Color-coded sentiment badge, KPI cards, theme / risk bar charts, bull / bear cards, and evidence quotes with tag pills
- Every evidence quote carries its transcript location, so each dashboard claim traces back to the source line

Out of scope, by design

- Cross-quarter trend analysis (single-call only, today)
- Speaker attribution (CEO vs. CFO tone separation)
- Batch portfolio mode (one call at a time today; parallel sweep is roadmap)
- Stock-price-reaction overlay; multilingual prompt; transcript truncation handling at >14,000 chars
- Real-time streaming mid-call; today the tool runs on the published transcript once the call ends

THREE TRANSCRIPT TYPES TO TRY IN THE LIVE TOOL

Hyperscaler call: Microsoft / Alphabet / Amazon AWS-segment-heavy transcripts surface AI-capex themes, hyperscaler-renewal evidence quotes, and CFO guidance hedging on cloud margin. Theme chart leans heavily on infrastructure; risk chart picks up regulatory and concentration.

Consumer-discretionary call: Nike / Starbucks / Lululemon transcripts pull out FX, China-exposure, and inventory-days risks. Bull / bear cards sharpen because consumer-discretionary management teams hedge openly.

Financial-services call: JPM / Goldman / Morgan Stanley transcripts surface NIM, deposit-beta, and capital-return themes. Evidence-quote tags get heavy use because regulated calls are quote-dense.

Metrics that matter

LAYER	METRIC	TARGET	WHY IT MATTERS
North-star	Analyst minutes saved per call	120 min → 1 min	Core ROI; anchors all downstream business cases
Quality	JSON schema compliance rate	> 98%	Below this, the dashboard breaks; analyst loses trust
Quality	Top-3 theme overlap vs. analyst	> 80%	Trust threshold for buy-side use
Compliance	Bytes of transcript egressed	0	Embargo-window regulated content
Latency	Transcript-to-dashboard wall time	< 90s CPU · < 10s GPU	Analyst stays in flow

Where this beats the alternatives, and where it doesn't

APPROACH	PRIVACY	COST	SPEED	CUSTOMIZABLE
Cloud LLM API (ChatGPT / Claude)	Cloud egress	Per-token	5 sec	Limited; opaque updates
Bloomberg / FactSet / AlphaSense	Vendor	Terminal-tier $$$$	Seconds	Very limited; black-box
This tool	**Local**	**$0**	**60s CPU**	**Full · prompt + schema**

Risks & mitigations

HIGH **7B-class model misses subtle CFO hedging that a senior analyst would catch.**
Mitigation: position as triage layer, not replacement. Analyst reviews top-N highlighted quotes (3 minutes); model handles the read-and-extract that buried the quotes (the slow part). Confidence-score field surfaces low-conviction extractions for analyst review.

HIGH **Schema drift on poorly-formatted transcript PDFs (older / non-English / scanned).**
Mitigation: two-pass defensive parser (regex-bounded extraction, AST validate, field-default fill); failures route to a fallback "raw extract" view rather than a broken dashboard.

MED **Inference latency on Q4 calls (30,000+ chars) exceeds analyst patience window.**
Mitigation: chunked extraction with per-chunk schema; GPU path drops wall time to <10s. Truncation at 14,000 chars currently flagged in the UI.

30 / 60 / 90, where this goes next

30 DAYS
Speaker attribution + golden set
› CEO / CFO tone-split in schema
› 50-call golden set, human-labeled, regression harness
› Evidence-quote source attribution

60 DAYS
Batch portfolio mode
› Parallel scheduler · 50-call sweep in <1 hr
› Portfolio rollup view · cross-name themes
› GPU path · <10s per call on commodity GPU

90 DAYS
Cross-quarter + price overlay
› Q-over-Q theme drift · 4-quarter window
› Sentiment vs. day-after-call price reaction

WHAT THIS PROVES

One question, one working answer: can an analyst-desk-deployable, locally-hosted, open-weight LLM stack do structured extraction reliably enough to sit inside a real research workflow? On v1, yes, for the well-formatted case. The 30 / 60 / 90 is the path from credible PoC to platform feature inside an equity-research product. Right home: a finance + AI PM seat at **Bloomberg, FactSet, S&P Global, AlphaSense, Morgan Stanley, or Snowflake's FS vertical.**