THE AI EXPEDITER

An agent team that tracks every material line-item from purchase order to job site, chases vendors automatically, predicts delays weeks in advance — and gets smarter on every project, because it remembers how this company's vendors actually behave.

Track: Track 2 — Supply Chain
Team: Khwahish Vaid — University of Toronto · Samarth Sharma — University of British Columbia
Date: July 2026

1. The Problem: Delays Nobody Saw Coming

The Track 2 brief opens with the questions every project team asks on every job: what has been approved, what is in fabrication, what is delayed, where each item is right now, and whether it will actually arrive when the site needs it. It then states the consequence plainly: nobody has good answers, and the delays cascade into missed schedules and blown budgets. [1]

The economics behind those five questions are brutal and well documented. McKinsey's research on capital projects finds the average large project finishing roughly **20 months behind schedule and up to 80 percent over budget** [2], with **98 percent of megaprojects** suffering cost overruns or delays [3]. Materials are a leading trigger: **65 percent of construction firms report projects delayed by supply-chain problems** [4]. And the failure is concentrated in long-lead items. Medium-voltage switchgear — the equipment that powers a finished building — now carries **40–80 week lead times**, more than double pre-pandemic norms [5]; panelboards that once arrived in 8–12 weeks now quote 28–48, and power-circuit-breaker switchboards exceed 84 weeks [6]. Worse than the length of a lead time is its unreliability: in one documented case a vendor quoted 40 weeks at bid time and delivered after more than 100 [7].

The killer is late discovery. A delay found three weeks early is a re-sequencing exercise; found three days early, it idles crews, extends site overheads, and cascades through every downstream trade. Yet between the purchase order and the site gate, every line-item crosses a paper trail — PO issued, submittal sent for approval, approved, fabrication, shipping, delivery — and every arrow in that chain is a place where the item can silently stall. The truth lives scattered across PDF purchase orders, vendor emails, spreadsheets, and phone calls, and a mid-size commercial project carries hundreds to thousands of line-items across dozens of vendors. Nobody has a single view.

The industry's existing fix is a person: the **expediter**, whose entire job is calling vendors so materials arrive on time. Expediters work — but they are expensive, they do not scale across hundreds of items, and their most valuable asset, an internalized model of how each vendor actually behaves, lives in their head and walks out the door when they do. That last clause is our wedge. **The knowledge should live in an organizational memory, not in a person's head.**

2. Proposed Solution: The AI Expediter

We propose an agent team that does the expediter's job end to end — and, unlike the human, never forgets. Five components:

Component 1 · Ingest (document intelligence) — Purchase orders, submittal logs, vendor quotes, status emails, delivery schedules and packing slips are parsed by a vision-language model with structured-output extraction into a canonical line-item schema — item, spec section, vendor, quantity, PO references, quoted lead time, promised date, current state, source document. This is document intelligence beyond OCR: the model reads mixed-format construction paperwork the way a project engineer does.

Component 2 · State Tracker — Every line-item lives on a pipeline — PO ISSUED → SUBMITTAL PENDING → APPROVED → FABRICATING → SHIPPED → DELIVERED — and state changes are triggered only by ingested documents and emails, never typed by hand. A single board shows all of a project's items colour-coded green, amber or red by delay risk. This one screen answers four of the track's five questions.

Component 3 · Agentic Chase — For any item stale beyond a threshold — no update in N days, or a promised date approaching while the state is not yet SHIPPED — the agent drafts a status-request email to the vendor, sends it after one-click human approval, parses the reply, updates the state, and escalates when the reply is evasive or absent. Every outbound message passes a human-in-the-loop gate: a trust-first design for a conservative industry, and precisely the agentic pattern the brief invites — systems that retrieve documents, send messages and complete workflows [1].

Component 4 · Delay Forecaster — For every line-item the system outputs a probability of lateness and a probabilistic ETA, not a binary flag. Features include the vendor's quoted lead time against that same vendor's historical actuals, the item category, submittal review cycles observed so far, and days since last vendor contact. The model starts deliberately simple — quoted lead time multiplied by the vendor's historical slip ratio — and graduates to gradient boosting as data accumulates. Probabilistic and honest beats deep and fake.

Component 5 · Vendor Memory (the differentiator) — A per-company, capped, human-readable playbook that the system writes after every completed order and every surprise, and that the Forecaster and the Chaser read on every decision. Section 3 describes it in full.

Components 1–4 are deliberately boring: proven techniques assembled sensibly. The innovation is concentrated in Component 5, where it can be measured.

3. The Differentiator: A Supply Chain That Remembers

Every dashboard competing in this space is stateless: it reports the status the vendor claims, and its value is the same on day 1,000 as on day one. Our system's value compounds, because it maintains an explicit organizational memory of how this company's supply chain actually behaves:

```
vendor_memory.md   (excerpt — written by the Scribe, read by Forecast & Chase)

## VENDOR PATTERNS         (how suppliers actually behave)
- SteelCo Ltd: quotes 4 wks, delivers in 6.5 wks (n=9). Reliable on rebar, slow on
plate.
- ElectroSwitch: never answers email; responds to phone within 1 day.

## PROCESS PATTERNS        (how this org's paperwork actually flows)
- Submittals routed through Consultant Y average 2 extra review cycles.
- Orders placed after the 20th slip into next month's fabrication slot.

## MISTAKES                (never repeat)
- Trusted a "shipped" status without a tracking number; item sat in a yard
  for 11 days. Always demand the tracking number.
```

Write policy. When an order closes or a prediction misses, a dedicated Scribe agent reviews the event history and edits the memory — adding, revising or deleting entries under a hard token cap — so the file stays small, current, and readable by any project manager. **Read policy.** The Forecaster conditions its predictions on the memory, and the Chaser conditions its tactics on it (it phones — or asks a human to phone — the vendor that never answers email). The memory is auditable by construction: a PM can open the file and see, in plain language, exactly why the system distrusts a vendor's "on schedule."

This design transplants a live frontier-research thesis into construction. Letta — the team behind MemGPT at UC Berkeley — argues that agents that truly learn will be powered by specialized memory systems that create and curate durable, token-space notes, because today's LLMs are excellent at using context but do not reliably create durable context for future tasks [8]. Construction supply chains are the ideal host for that thesis: the same vendors, materials and failure patterns recur project after project, which is precisely the condition under which a learned notebook compounds in value.

The claim is measurable, and we commit to measuring it: forecast error on late deliveries must fall as memory accumulates. Our headline demo artifact is a single chart — month-1 versus month-6 forecast error on the same synthetic company — showing a stateless baseline staying flat while the memory-equipped system improves. If the line does not fall, the differentiator has failed, and we will say so.

4. Technical Approach

Architecture and stack. A multi-agent system (Google Agent Development Kit or LangChain) with five roles — Ingestor, Tracker, Chaser, Forecaster, Scribe — over Firestore (items, events, memory) and Cloud Run, with a Gemini-class VLM for extraction and drafting, and the Gmail API in sandbox for the chase loop. The frontend is deliberately one page: the line-item board, an item drill-down, and the memory file rendered as plain text.

Data. We do not have a real contractor's data, and we will not pretend to. We will build a synthetic-company generator: a fictional general contractor with three projects and roughly forty vendors, where each vendor has hidden behavioural parameters — true lead-time distributions, reply latency, slip tendencies — from which twelve months of POs, submittal logs, vendor emails and delivery events are generated. Because vendor behaviour is generated from hidden parameters, Vendor Memory has genuine signal to discover, and the Forecaster can be scored against ground truth the models never see: the memory must learn general patterns, not memorize instances. Recent results support the external validity of this methodology — an open-source model RL-trained purely on a synthetic email corpus outperformed frontier models on real, unseen data [9]. For visual authenticity, generated documents will be formatted on real, publicly available PO and invoice templates.

Evaluation plan. (i) Extraction: field-level exact-match accuracy on held-out generated documents. (ii) Forecasting: mean absolute ETA error and calibration of the lateness probability, tracked as a function of memory size against the stateless baseline. (iii) Chase loop: end-to-end state updates from parsed replies, with the human approval gate exercised on video.

Roadmap — the moat. Once one company's document formats are known, extraction should not need a frontier model. We will apply the now-standard specialization recipe — GRPO fine-tuning of a small open model against a verifiable exact-match reward — to build a per-company extractor. OpenPipe's ART·E demonstrated the economics: a 14B open model, trained for under $80 on a single GPU, answered 60 percent of the questions the frontier model missed while running 5× faster and 64× cheaper [10]; enterprises such as Ramp now commission exactly this style of RL specialization from platforms like

Prime Intellect [11]. Construction documents are high-volume and margins are thin, so a two-orders-of-magnitude inference-cost reduction is a business feature, not a benchmark stunt. This is the "technique nobody else thought to apply" that the brief's own tip section invites [1].

5. Feasibility, Team & Stage-2 Plan

The team is Khwahish Vaid (University of Toronto) and Samarth Sharma (University of British Columbia). The build splits across four workstreams — agents & backend, synthetic data & evaluation, forecasting, and frontend & demo — with each member owning two. [Add one sentence per member on prior projects, internships, or relevant coursework.]

Stage-2 build sequence (strictly gated — no stage begins until the previous one's check is green). **Days 1–2:** synthetic generator, schema, and ingest — done means 200 line-items tracked from raw documents. **Days 3–4:** board UI and Forecaster v0 — done means a red/amber/green board with probabilistic ETAs. **Day 5:** the chase loop on sandbox Gmail — done means one full chase cycle captured on video. **Days 6–7:** the Vendor Memory Scribe and the before/after accuracy chart — done means the headline chart is measured, not mocked. **Stretch, only if all gates are green:** the RL-fine-tuned extractor. Estimated compute cost: under $150 in API usage for the prototype; $100–300 of rented GPU time for the stretch fine-tune.

6. Impact & Scalability

India is the sharpest version of this problem and the largest market for its solution. Construction ranks among the least digitized major sectors on the McKinsey Global Institute's digitization index — second-to-last in the United States and last in Europe [12] — while India's FY 2026-27 budget allocates ₹12.2 lakh crore (about $136 billion) to capital expenditure [13], on top of the ₹111-lakh-crore National Infrastructure Pipeline spanning more than 7,000 projects [14]. Every one of those projects orders materials; almost none of them can answer the five questions.

On a single project the payoff mechanism is simple: converting late discoveries into early ones converts budget events into re-sequencing exercises, and one avoided idle week on a large site pays for the software many times over. Scaling is structural: more projects and more sites are simply more rows in the tracker, while every closed order makes the memory — and therefore the product — better. That compounding is also the business moat. A stateless dashboard can be swapped for a competitor's overnight; an organizational memory that has learned your forty vendors cannot.

In one sentence: we take the track's five unanswered questions, answer four of them with solid, boring, proven engineering — ingest, track, chase, forecast — and win on the fifth dimension no other team will have: **a system that remembers.**

7. References

[1] Kaya AI IIT India Hackathon 2026 — Track descriptions, technical guidance and judging criteria. Devpost. https://kaya-ai-iit-hackathon-2026.devpost.com/

[2] McKinsey & Company, "Navigating the digital future: The disruption of capital projects" (2017) — average capital project ~20 months behind schedule and 80% over budget. https://www.mckinsey.com/capabilities/operations/our-insights/navigating-the-digital-future-the-disruption-of-capital-projects

[3] McKinsey & Company, "The construction productivity imperative" (2015) — 98% of megaprojects face cost overruns or delays; average delay 20 months. https://www.mckinsey.com/~/media/McKinsey/Industries/Capital%20Projects%20and%20Infrastructure/Our%20Insights/The%20construction%20productivity%20imperative/The%20construction%20productivity%20imperative.pdf

[4] Associated General Contractors of America survey data, cited in OpenSpace, “Construction project delay statistics” (2026) — 65% of firms report projects delayed by supply-chain challenges. https://www.openspace.ai/blog/construction-project-delay-statistics/

[5] DEI Power, “Electrical Switchgear Shortage: Supply Chain Impact & Solutions” (June 2026) — medium-voltage switchgear lead times of 40–80 weeks, more than double pre-pandemic averages. https://feeds.deipower.com/blog/electrical-switchgear-shortage

[6] Electronate, “Switchgear Lead Times in 2026: What Every Estimator Must Build Into Their Bid” (March 2026) — panelboards 28–48 weeks (vs. 8–12 historically); switchboards ~52 weeks; power-circuit-breaker switchboards 84+ weeks. https://electronate.app/blog/switchgear-lead-times-2026-data-center-boom

[7] Electrical Trends, “Switchgear Delays” (April 2024) — contractor account of equipment quoted at 40 weeks and delivered after more than 100 weeks. https://electricaltrends.com/2024/04/01/switchgear-delays/

[8] Letta, “Memory Models: Towards Agents That Learn” (2026) — specialized memory models trained with memory-native RL; LLMs use context well but do not reliably create durable context. https://www.letta.com/blog/towards-agents-that-learn/

[9] Tonic.ai, “Synthetic Data Is All You Need For Reinforcement Learning” (March 2026) — a model RL-trained on a fully synthetic email corpus outperformed o3 on real, unseen Enron data. https://www.tonic.ai/blog/synthetic-data-is-all-you-need-for-reinforcement-learning

[10] OpenPipe, “ART·E: How We Built an Email Research Agent That Beats o3” (2025) — Qwen-14B agent answering 60% of the questions o3 missed, 5× faster and 64× cheaper; trained for under $80 on a single H100. https://openpipe.ai/blog/art-e-mail-agent

[11] TechCrunch, “Prime Intellect raises $130M Series A to help enterprises build their own AI agents” (July 8, 2026) — RL-environment platform; customers include Ramp and Zapier. https://techcrunch.com/2026/07/08/prime-intellect-raises-130m-series-a-to-help-enterprises-build-their-own-ai-agents/

[12] McKinsey Global Institute, “Reinventing construction through a productivity revolution” (2017) — construction among the least digitized sectors; second-to-last in the US, last in Europe. https://www.mckinsey.com/capabilities/operations/our-insights/reinventing-construction-through-a-productivity-revolution

[13] ResearchAndMarkets via GlobeNewswire, “India Construction Market Analysis and Outlook to 2030” (May 2026) — FY 2026-27 capital expenditure of ₹12.2 trillion (~$136B); construction growing 6.4% in real terms. https://www.globenewswire.com/news-release/2026/05/05/3288002/28124/en/India-Construction-Market-Analysis-and-Outlook-to-2030-FY2026-27-Budget-Allocates-INR12-2-Trillion-for-Capital-Expenditure-Boosting-India-s-GDP.html

[14] Invest India, “Building New India: The National Infrastructure Pipeline” — ₹111 lakh crore identified investment across 7,000+ projects. https://www.investindia.gov.in/team-india-blogs/building-new-india-national-infrastructure-pipeline