

KEEL

An autonomous operations controller that keeps a production line running when suppliers fail — detecting disruption, **challenging supplier claims against hard evidence**, sourcing alternates, allocating recovery under budget and certification constraints, and escalating with a decision brief instead of an alert.

NOT

A chatbot. A dashboard. A document assistant. A fixed workflow.

IS

An operator that acts over multiple steps and leaves a verifiable trail.

BUILT

A deployed web application. Real agent, real solvers, simulated world.

NAMED FOR

The keel — the member that keeps a ship from capsizing.

Two companies. The same fire. The same signal. A \$2.34 billion difference.

ERICSSON

Accepted the estimate. Philips said the Albuquerque plant would recover in about a week. Ericsson filed it, waited, and escalated three weeks later. Recovery actually took months.

Lost **\$2.34bn** in its mobile phone division and exited handset manufacturing.

NOKIA

Refused the estimate. Challenged the recovery figure, demanded specifics, escalated within days, put engineers on site and locked alternate capacity before the market knew.

Shipped through the quarter and **took share**.

The difference was not information. Both had the same signal on the same day. The difference was whether anybody challenged a supplier's estimate against evidence, quantified the exposure, and escalated with a decision rather than an alert.

KEEL is the Nokia response, in software — applied to every part number, every day.

17 March 2000, Philips semiconductor plant fire, Albuquerque NM. Widely documented in the supply-chain risk literature (Sheffi, *The Resilient Enterprise*).

PART ONE

What the research actually says

We did not start from the brief. We started from the failure literature and the 2025–26 disruption record, and let the findings choose the problem worth solving.

Five findings

One gap

One product

Detection is close to solved. The forty hours after detection are not.

TIME TO DETECT A DISRUPTION

8.7 h

News feeds, port data, AIS, weather, supplier alerts.
Effectively commoditised.

TIME TO UNDERSTAND ITS IMPACT

40.9 h

Which parts, which orders, which customers, how
much revenue, what to do about it. Done by hand,
across systems.

SYSTEMS TOUCHED FOR ONE DECISION

3–10

ERP, email, spreadsheets, vendor portals, carrier
tracking, finance. Reported by 75% of teams.

The bottleneck is interpretation, not sensing. A buyer covering 200–400 part numbers cannot spend forty hours on one vague email — so the assessment does not happen. The email is filed as “acknowledged”, the delayed delivery keeps counting as inbound stock, and the shortage becomes visible when the line stops.

Vendor and industry-survey figures — see Sources. Procurement separately reports ~31% of its time spent on manual and paper process (Ivalua).

Control towers largely failed at their actual job — and alert fatigue is the symptom.

WHAT THE LAYER OPTIMISES	REAL LIMITATION
Control towers	Only 22% of >\$1B-revenue shippers call theirs highly effective at <i>driving action</i> . Built as reporting projects, not decision projects.
SCRM platforms	Risk at entity level, not part level. "Supplier X is risky" — not "which order stops, and when".
Questionnaires & audits	Point-in-time, self-reported, stale by roughly 18 months at sub-tier.
Inventory buffers	Static hedges, often on the wrong SKUs. A direct margin cost.
Dual sourcing	78% adopted it for critical materials. The second source is often the same fab, region or Tier 2 — and frequently unqualified.

THE PATTERN

Every layer above optimises **sensing** or **static structural hedging**. Almost nothing optimises the **decision and the execution in the 48 hours after a signal**.

CONSEQUENCE

Visibility was built ahead of response capacity, so problems now arrive faster than the organisation can absorb them. That is alert fatigue, and it is a *throughput* problem, not a data problem.

Three inputs a planner must treat as adversarial — that every tool assumes are true.

THE ERP NUMBER

65%

of ~370,000 inventory records at a major retailer were inaccurate. Where the system *overstates* stock, replenishment fires too late and the line stops.

DeHoratius & Raman, *Management Science*, 2008

THE SUPPLIER REPLY

“Delivery may be delayed by 5–7 days. We are trying to resolve this and will update soon.”

Parses as “acknowledged” in every ERP on earth. It contains **no date, no quantity and no commitment** — yet the delivery keeps counting as inbound stock.

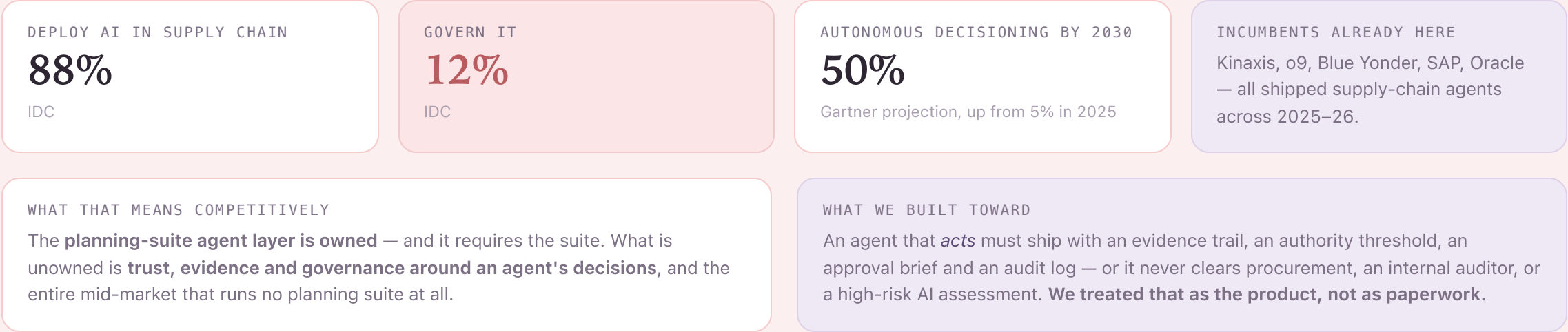
THE CLAIM OF FACT

“The shipment has been dispatched and is on its way.”

A shipping label generated to answer the question, with no pickup behind it. Believing it and standing down alternate sourcing **is the Ericsson decision**.

Nothing in the standard stack checks any of these. An ERP stores the number it was given. A control tower forwards the message it received. The verification step exists only in the heads of experienced buyers — and only when they have time.

Everyone is shipping agents. Almost nobody is governing them.



The mid-market is structurally unserved — and it is running on a workbook.

THE BUYER WE ARE BUILDING FOR

A mid-size manufacturer: a handful of people covering sourcing, onboarding, compliance and supplier relationships across hundreds of suppliers. **No risk function. No planning suite. No integration budget.**

THEIR ACTUAL INCUMBENT TOOL

Spreadsheets and email remain the most-used supply-chain risk tool in 2026. That is not an adoption gap; it is the market's true baseline.

WHY ENTERPRISE SCRM DOES NOT REACH THEM

Resilinc, Everstream, Interos and Z2Data are priced and implemented for the Fortune 500. A six-month integration project is not available to a plant with four people in procurement.

THE DESIGN CONSEQUENCE

If the buyer's system of record is a workbook, then **the workbook is the ERP** — and connecting to it directly is not a compromise or a demo shortcut.

This one finding is why KEEL reads and writes **.xlsx** as a first-class data source, and why it can watch a live sheet and replan the moment somebody edits a cell. **Nobody asked us for this. It is the difference between a demo and something a plant could use on Monday.**

The unsolved problem is not seeing the disruption. It is deciding what to do about it, defensibly, in the next 48 hours.

EVERYONE OPTIMISES

Sensing · alerting · static hedging

ALMOST NOBODY OPTIMISES

Impact assessment · option generation · evidence-checked execution · escalation quality

NOBODY SERVES

The company with no risk team, no planning suite, and a workbook for an ERP

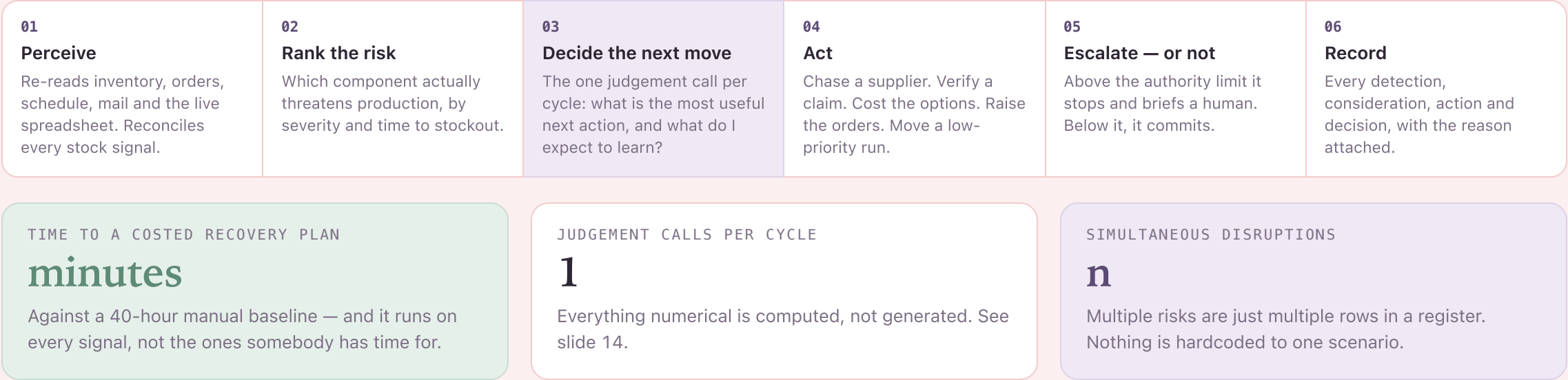
KEEL attacks the impact-assessment and response loop — for a company that cannot afford a twelve-person risk team — compressing forty hours into minutes and producing an auditable, evidence-backed decision rather than another alert.

PART TWO

How we solve it

One loop, eight modules, and a hard rule about which parts of the decision a language model is allowed to touch.

It runs the buyer's Tuesday, one step at a time, and shows its working at every step.



Eight modules. Each one exists because of a documented way this goes wrong.

MODULE	WHAT IT DOES	WHAT HAPPENS IN A COMPANY WITHOUT IT
Stock reconciliation	Takes the ERP figure, the warehouse figure and the figure implied by consumption — and plans on the most pessimistic .	The buyer plans on the ERP number. Replenishment fires days late; the first sign of trouble is an empty bin on the line.
Commitment parser	A reply counts only with a specific date , a specific quantity and no hedging .	“Trying to resolve, will update soon” is filed as a response. The delivery keeps counting as inbound stock and the shortage stays invisible.
Claim verifier	Checks any claim that goods shipped against the carrier record — and on a contradiction keeps sourcing anyway .	The dispatch claim is believed, alternate sourcing is stood down, and nothing arrives. This is the Ericsson decision.
Risk register	Working memory. Every risk carries its assumptions, each with an expiry.	The plan is made once and held past its expiry. Nobody notices the thing it depended on stopped being true.
Recovery allocator	Certification and quality gate first , then searches single-source, split and staged options across suppliers.	The quote table is sorted by unit price. Either an uncertified part enters a safety-critical assembly, or one vendor is asked to cover everything.
Cost of delay	Prices buying speed against the delay it avoids — including the option of moving a low-priority run for zero rupees.	Expediting becomes the reflex. Roughly half of emergency freight spend traces to a planning failure, not a logistics one.
Escalation brief	Cost, overage, impact if approved, impact if rejected , alternatives with reasons, residual risk, decide-by time.	The manager gets “urgent approval needed” with no cost and no consequence. Approval averages more than two business days.
Supplier memory	Hedging and contradictions attach to the supplier and feed straight back into the allocation maths.	Reliability is a number someone updates quarterly, if at all. The supplier who lied last month is scored the same as the one who did not.

Four moments — each one a thing the existing stack cannot do.

Moment 1

"Your ERP is lying."

The screen says 800 units. The warehouse says 390. Consumption since the last update implies 304. It plans on **304** — unprompted — and says why.

Moment 2 · the one that wins the room

"It says dispatched. The record says otherwise."

The supplier claims the shipment left. The carrier record shows a label and no pickup. It refuses the claim, cuts the supplier's reliability, removes the order from cover, and **keeps sourcing**.

Moment 3

"Don't expedite — split."

Rejects the cheapest supplier on certification, splits across two vendors plus stock, and delays a low-priority order instead of paying an expedite fee. **With the arithmetic on screen.**

Moment 4

"A brief, not an alert."

Cost crosses the threshold, so it stops and hands over one screen. You press Approve or Reject — and on a rejection **it goes back and finds something cheaper** rather than stalling.

Every one of these is reproducible on a fresh random seed, in front of you. Supplier names, quantities, thresholds and disruption timing all change with the seed — which is the only convincing answer to "is this hardcoded?"

The model decides *what to do next*. Code decides *what the numbers are*.

GIVEN TO THE LANGUAGE MODEL

- Reading a hedge and knowing it is not a commitment
- Judging that a claim is worth verifying before money moves
- Choosing which of fifteen tools moves this situation forward
- Writing an email that forces a supplier to give a real date
- Judging whether a trade-off deserves human sign-off

NEVER GIVEN TO THE LANGUAGE MODEL

- Days of production coverage
- Allocation of quantities across suppliers
- Any cost, premium or fee total
- Probability that a plan holds
- Whether a cost crosses the approval threshold

Those are deterministic solvers it must call. It is instructed never to state a figure it did not get from one.

Why this matters commercially, not just technically: a hallucinated coverage number is not a bad answer — it is an unexplainable purchase order. Every figure in every brief traces to a named calculation with its inputs recorded, which is what an internal auditor, a procurement committee and an AI-governance review each ask for first.

Four controls, built in from the first commit — not added for compliance later.

01 · AUTHORITY LIMIT

A hard threshold. Below it the agent commits. Above it, execution is **refused in code** until a human approval exists — not merely discouraged in a prompt.

02 · BRIEF QUALITY GATE

An escalation is **rejected** unless it carries cost, overage, impact if approved, impact if rejected, alternatives with reasons and residual risk. The agent cannot escalate to avoid deciding.

03 · EVIDENCE SEPARATION

What a supplier *says* and what the record *shows* are different data. Claims are verified against an independent source before money moves.

04 · APPEND-ONLY TRAIL

Nine kinds of event — detection, reasoning, action, decision, escalation, contradiction, discrepancy, replan, disruption — each with the reasoning that produced it. Exportable.

This is the wedge the research pointed at: 88% of firms deploy supply-chain AI and 12% govern it. The scarce thing in 2026 is not an agent that acts — it is an agent whose actions survive being questioned six weeks later.

Against every other thing that claims to handle this.

	WHAT IT GIVES YOU	WHAT IT STOPS SHORT OF	KEEL'S DIFFERENCE
ERP / MRP	The record, and an alert when a threshold breaks.	Any view on what to do. It also cannot tell you its own stock figure is wrong.	Reconciles the record against itself and plans on the pessimistic figure.
Control tower	Visibility across systems and lanes.	Driving action — only 22% are rated effective at it. Alert volume without decision throughput.	Ends in an executed plan or a signed-off brief, never in an alert.
SCRM platform	Entity-level supplier risk scores.	"Which of <i>my</i> orders stops, when, and what are my options?"	Part → order → revenue → costed recovery, in one loop.
Planning-suite agents	Agentic actions inside the suite.	Requires the suite. Deployed by 88%, governed by 12%.	Runs on a spreadsheet, and ships the governance as the product.
RPA / fixed workflow	Reliable repetition of a known sequence.	Anything the author did not anticipate. Disruption is definitionally that.	State-driven: it re-decides every cycle rather than following a branch.
A chatbot on the ERP	Answers a question you already knew to ask.	Noticing, deciding, acting, and being accountable for it.	Nobody has to ask. It works the queue and stops only when authority requires.

PART THREE

Against the brief — and beyond it

Every capability the problem statement asks for, mapped to the thing that answers it. Then the work we did that **nobody asked for**, and the reason each piece exists.

Ten required capabilities. Ten answers, each a named component.

§	REQUIRED CAPABILITY	HOW KEEL ANSWERS IT	EVIDENCE YOU CAN SEE IN A RUN
4.1	Autonomous monitoring	Every cycle re-reads inventory, usage, safety stock, open POs, delivery dates, mail, schedule, budget and thresholds — and computes downstream consequence, not just breach.	Risk register opens rows by itself, ranked by severity and time to stockout.
4.2	Dynamic task decomposition	No scripted sequence. The next action is chosen from live state each cycle, so multiple simultaneous disruptions decompose naturally.	Change the seed or the scenario and the order of operations changes with it.
4.3	Tool selection	One tool per cycle, chosen against the risk in focus; repeat calls with unchanged inputs are refused and the refusal is fed back.	Tool-call gauge in the header; “refused a repeat call” entries in the trail.
4.4	Supplier communication	Real email semantics — threads, subjects, envelopes. It chases vague replies for a date and a quantity, and challenges contradictions.	The Mail screen, with every inbound reply tagged firm or not a commitment.
4.5	Inventory & production reasoning	Coverage from reconciled stock, walked day by day, catching both quantity shortfalls and continuity breaks before a deadline.	Coverage days, first stockout time, affected orders with revenue and slip.

The second five — where most of the judgement actually lives.

§	REQUIRED CAPABILITY	HOW KEEL ANSWERS IT	EVIDENCE YOU CAN SEE IN A RUN
4.6	Supplier & RFQ evaluation	Price, lead time, reliability, quality, MOQ, availability, past performance, expedite terms — and splitting is searched, not hardcoded.	Three ranked plans, each with full arithmetic and a continuity probability.
4.7	Constraint management	Budget, thresholds, certifications, quality floors, MOQ, deadlines, safety stock, lead times, tool-call budget and a cost-bearing clock.	Rejected options listed with reasons — including ones that were cheaper <i>and</i> faster.
4.8	Replanning & recovery	Assumptions carry expiry. Any external change unverifies them, reopens the affected risks and discards costed plans.	Press ⚡ Inject during a run and watch the register reopen.
4.9	Human escalation	Only when a business rule requires it — and refused unless the brief is complete. A rejection reopens the risk with an instruction.	The approval screen; reject it and the agent finds a cheaper route.
4.10	Audit trail	Append-only, nine event kinds, reasoning attached to each action, exportable as JSON.	The Audit screen, filterable to decisions only.

The brief asks for an agent that can “recover from uncertainty”. Every one of these five is a place where the honest answer is “I do not know yet” — an unverified claim, an expired assumption, a rejected plan, a supplier who will not commit. The system is built so that state is *representable* rather than papered over: a risk stays open, an order stops counting toward cover, and the work continues elsewhere.

Each hidden test has a module that exists specifically to pass it.

#	THE TRAP	WHAT CATCHES IT
1	Supplier confirms, then reneges	Commitment parser + supplier memory
2	ERP overstates stock	Stock reconciliation
3	Cheapest supplier fails quality	Constraint gate
4	Reliable supplier short on quantity	Split allocator
5	Fast supplier is unreliable	Continuity probability

#	THE TRAP	WHAT CATCHES IT
6	Demand rises mid-run	Assumption ledger + forced replan
7	Expedite withdrawn	Cost of delay + re-solve
8	Claim contradicts tracking	Claim verifier
9	Cost exceeds approval limit	Escalation brief gate
10	Production priority flips	Risk re-ranking + replan

And all ten are available as a button. Six of them can be fired into a live run from the console, so an evaluator does not have to take our word for the recovery behaviour — **they can break the world themselves and watch what happens.**

Six things the problem statement never asked for, and why each one is in the build.

WHAT WE ADDED	WHY WE DID IT	WHAT IT UNLOCKS
Spreadsheet as a first-class ERP import · export · template	The research says spreadsheets are still the most-used risk tool in 2026. A sandbox that only reads its own seeded JSON proves nothing about the buyer who actually has this problem.	Any plant can put its own parts, suppliers and orders in front of the agent in ten minutes — no integration, no IT project.
Live sheet connection file path or URL, re-read every cycle	Uploading a file is a one-off; a plant's workbook changes all day. A buyer editing a cell is a signal in exactly the way a supplier email is.	Edit a cell mid-run and the agent notices, states the change in plain English, throws away its costed plans and replans.
Suppliers as improvising agents four hidden personas	Scripted replies can be memorised, and they make evasion read like a canned string. Hidden personas — honest, optimistic, evasive, deceptive — improvise against ground truth they are told not to reveal.	Replies differ every run, so nothing can be hardcoded against them . It is the strongest available answer to “is this scripted?”
Human takeover portal a private link per supplier	The most honest test of a claim verifier is an adversary it has never seen. So we let one in.	An evaluator opens a link on their phone and becomes the supplier — stalling or lying in their own words. The agent cannot tell them from a persona.
Real email transport outbound + inbound webhook, off by default	To prove the sandbox boundary is a configuration flag , not an architectural limit. Envelopes and threading are genuine throughout.	Turn one variable on and the same messages go over real SMTP — to an address the operator owns, never to a supplier's. Replies flow back into the run.
Provider-agnostic model adapter	Vendor lock-in is a procurement objection, not just an engineering one.	Azure AI, OpenAI or any compatible endpoint — including a self-hosted model for a plant that will not send data outside its own network.

Five more, all built for the same reason: a claim you cannot test is marketing.

WHAT WE ADDED	WHY WE DID IT	WHAT IT UNLOCKS
The counterfactual engine "What it prevented"	ROI for this class of product is normally asserted. We compute it from the run's own opening snapshot — before the agent touched anything, because by the time you look, it has already fixed the problem.	A defensible savings number: believed vs actual coverage, when the buyer would have found out, downtime and panic-freight exposure, against what was actually spent.
The Engine Lab solvers with no agent attached	A sceptic's first question is whether the maths is real or the model is improvising numbers. So the maths is exposed directly, with dials.	Anyone can turn the stock, usage and lead-time dials and watch coverage, severity and commitment parsing respond — with the agent switched off entirely.
Disruption injection console six one-click world-breakers	The brief says the environment <i>should be allowed to change</i> . We made it changeable by the audience, mid-run, on stage.	Recovery behaviour becomes a demonstration rather than a claim — and every injection forces a genuine replan rather than a cosmetic one.
Implemented scorecard the rubric, in code	If continuity and cost are 55% of what matters, the system should measure itself against that continuously — not once, at the end, by hand.	Every run self-scores on six weighted lines with the working shown. Regressions surface immediately.
Ground-truth isolation enforced at three layers	A simulation that cannot lie to the agent cannot test it. Physical stock, hidden personas and true lead times exist — and are stripped from tool results, from the API response and from the browser.	Open the network tab during a demo and you find nothing the agent could not see . The test is honest by construction.

Each addition removes a different reason somebody could not use this.

REMOVES: “WE HAVE NO SYSTEM TO INTEGRATE WITH”

Spreadsheet-native + live sync. The unserved mid-market — no risk function, no planning suite, hundreds of suppliers — becomes reachable without an integration project. This is the single largest audience expansion in the build.

REMOVES: “WE CANNOT LET IT ACT”

Authority threshold, brief gate, audit trail, evidence separation. Procurement committees, internal audit and AI-governance reviews all get what they ask for first. 88% deploy, 12% govern — this lands on the right side of that.

REMOVES: “WE DON'T TRUST THE NUMBERS”

Engine Lab + deterministic solvers + counterfactual. A sceptical buyer can inspect the arithmetic without touching the agent, and the savings figure is computed rather than claimed.

REMOVES: “OUR DATA CAN'T LEAVE THE BUILDING”

Provider-agnostic adapter. Any OpenAI-compatible endpoint, including a self-hosted model. Regulated and defence-adjacent manufacturers stay in scope.

REMOVES: “SUPPLIERS WON'T CO-OPERATE”

Portal + real email. A supplier answers by link or by simply replying to an email. Nothing to install, no account to create — the lowest-friction path that exists.

REMOVES: “PROVE IT ISN'T SCRIPTED”

Seeds, personas, human takeover, injections. The system is built to be attacked live. That is what converts a demo into due diligence.

PART FOUR

Value, market and the path in

Where the money comes from, who signs for it, and how an agent that spends money earns the right to do so.

Four sources — three of them are avoided losses, and one is avoided waste.

01 · STOPPAGE AVOIDED

The dominant term. Line-down cost per hour multiplied by the hours between the stockout and the first realistic relief — plus the revenue on orders that miss.

02 · PANIC FREIGHT AVOIDED

Once the line is down, procurement pays whatever it takes. Emergency air on the same lane runs far above standard; acting days earlier converts it into a normal order.

03 · EXPEDITE DISCIPLINE

Top performers hold expedite to ~3% of logistics cost; bottom performers ~10%. Roughly half of expedite events trace to a planning failure. Pricing speed against the delay it avoids — every time — is the control.

04 · BUYER HOURS RETURNED

The 40-hour assessment collapses to minutes, and it runs on **every** signal rather than the few somebody had time for. Procurement reports ~31% of its time on manual process.

WORKED EXAMPLE — FROM ONE LIVE RUN, COMPUTED BY THE PRODUCT ITSELF

A single component, a single delayed order. The ERP shows **8.9 days** of cover; reconciled reality is **2.7 days**. Without intervention the buyer discovers this at the stockout, the line is down until the incumbent's already-slipped delivery lands, and the recovery is bought at emergency rates. **KEEL commits a fraction of that exposure, days earlier, inside the approval limit.** The figures on the “What it prevented” screen are recomputed live from that run's own opening snapshot — never hardcoded, and never taken from the world after the agent has already repaired it.

Three roles, three different arguments — all from the same run.

THE BUYER / PLANNER

- Every vague email assessed, not just the ones with time
- Coverage they can defend, on reconciled stock
- Alternates already gated on certification and quality
- Split-versus-expedite priced, with the arithmetic attached
- The correspondence drafted and chased for them

THE APPROVER

- A decision in 60 seconds instead of a two-day thread
- **Impact if rejected** stated, not just impact if approved
- Alternatives listed with the reason each was turned down
- Rejecting is productive: the risk reopens with an instruction
- Fewer escalations, because the threshold is enforced in code

THE BUSINESS

- Stoppages avoided, with the counterfactual computed
- Emergency spend that survives an audit six weeks later
- Supplier truth on record and priced into future allocation
- A governance story that clears procurement review
- No integration programme to fund before value appears

What it does not claim: it does not replace the buyer, it does not learn between runs, it does not forecast demand, and it does not move money or touch a live ERP. Authority above the threshold stays human — and the threshold is a setting, not a promise.

Start where the pain is sharpest and the incumbent is a spreadsheet.

SEGMENT	WHY THEY HURT	WHY THEY CAN ADOPT	ENTRY MOTION
Mid-market discrete manufacturers automotive & EV components, industrial electronics	Hundreds of suppliers, single-source parts, certification gates, real line-down costs — and no risk function.	The workbook <i>is</i> the system of record, so there is nothing to integrate. Decision authority is short and local.	One plant, one component family, one buyer. Value visible in the first week.
Contract manufacturers & Tier 1s	Their customer's line stopping is their commercial event. Penalty clauses make timing existential.	Already instrumented, but the decision loop is still manual and lives in email.	Land as the escalation and evidence layer above the existing ERP.
Enterprises with a planning suite	The suite plans; it does not challenge a supplier claim or hold an authority threshold with an audit trail.	Runs alongside rather than replacing. Reads exports; writes nothing without approval.	Governance and evidence layer for agent decisions — the unowned space.
Suppliers themselves expansion	When a customer goes dark, thousands of suppliers have no way to model the outage.	The portal side already exists as a product surface.	Downstream continuity — deliberately not in scope today.

Segment prioritisation follows the research severity ranking: impact–assessment gap as the product, sub-tier exposure as the later data moat, execution and qualification as the expansion path.

Nobody grants autonomy on day one. So the threshold starts at zero.

Stage 1 Shadow Threshold at ₹0. It watches the real workbook, opens risks, computes coverage and drafts plans. It executes nothing. You compare its calls to yours.	Stage 2 Advisory It drafts supplier emails and approval briefs; a human sends and signs. The audit trail accumulates against real events.	Stage 3 Bounded autonomy A small threshold — one shift's worth of spend. It commits below it and briefs above it. Every commitment traceable.	Stage 4 Operating The threshold is tuned to observed performance. Escalation volume, not autonomy, becomes the metric people watch.
--	--	--	--

WHY THIS SEQUENCE WORKS COMMERCIALY

Every stage produces evidence for the next. Shadow mode costs nothing and proves the coverage maths against the buyer's own data — which is the objection that actually blocks these deals, not the price.

WHY IT WORKS TECHNICALLY

The threshold is already a first-class setting enforced in code, not a prompt instruction. Stage 1 is **the same product with one number set to zero**.

Priced per site, against the loss it prevents — not per seat.

PILOT

One plant

Fixed fee, 6–8 weeks, shadow mode on the buyer's own workbook. Deliverable is the counterfactual report on real disruptions that occurred during the window.

Purpose: convert scepticism about the maths into evidence.

SUBSCRIPTION

Per site

Tiered on part numbers monitored and suppliers in scope. Includes the approval workflow, audit export and supplier portal. Model cost is a pass-through line.

Rationale: value scales with parts at risk, not with headcount — and the buying team is deliberately small.

EXPANSION

Per module

Sub-tier exposure mapping, alternate-source qualification workflow, and supplier-side continuity — each a separate later module with its own buyer.

Sequenced after the core loop is trusted, never before.

ASSUMPTIONS WE WOULD VALIDATE BEFORE COMMITTING TO ANY OF THIS

- That a mid-market plant will price a prevented stoppage rather than a headcount saving
- That workbook quality is good enough to run against without a cleanup project
- That the buyer, not IT, can sign a per-site subscription
- That model cost per run stays small relative to a single avoided expedite
- That supplier co-operation via link or email reply is acceptable to procurement
- That shadow mode produces a defensible counterfactual within one quarter

Stated as assumptions on purpose. Pricing built on invented market numbers is the fastest way to lose a room that knows the sector.

The hard part is not the agent. It is everything that makes the agent's output survive scrutiny.

EASY TO COPY

- Wiring a model to a set of procurement tools
- A console that streams what the agent is doing
- A prompt that says “be careful with supplier claims”

HARD TO COPY

- Solvers that are **correct** — timing breaks as well as quantity gaps, MOQ, split search, continuity under supplier unreliability
- An escalation format an approver actually signs, with impact-if-rejected
- The discipline that no number is ever generated, only computed
- An evidence model that separates what a supplier said from what the record shows
- Accumulated supplier truth that feeds back into allocation

The compounding asset is the decision record. Every run adds evidence about which suppliers hedge, which claims fail verification, and which recovery shapes held. That is institutional memory the research specifically identifies as missing — the same disruption class gets re-solved from scratch every time, by everyone.

Risks, stated plainly.

RISK	WHY IT IS REAL	MITIGATION ALREADY IN THE BUILD
The model states a wrong number	The failure mode that would end trust permanently, and the one language models are most prone to.	It is forbidden from arithmetic . Coverage, allocation, cost and probability come from deterministic solvers it must call.
It acts on a false supplier claim	Exactly the Ericsson failure, and suppliers have every incentive to sound confident.	Claims of fact are verified against an independent record before money moves — and a contradiction never stands down alternate sourcing.
It escalates everything, or nothing	Either destroys the value: one is alert fatigue again, the other is unauthorised spend.	Threshold enforced in code; briefs rejected when incomplete; a risk waiting on a human is deprioritised so work continues elsewhere.
Runaway cost or a decision loop	An agent that re-reads the same data forever is expensive and useless.	Hard tool-call budget, a repeat-call governor that refuses and re-asks with the earlier result, and a clock that makes acting cost more than reading.
Buyer data is too messy to run on	Mid-market workbooks are inconsistent by nature.	A published template, a tolerant importer, and a reconciliation model that expects the numbers to disagree and reports the disagreement.
Incumbents ship the same thing	Kinaxis, o9, Blue Yonder, SAP and Oracle are all moving.	They require the suite. The unserved segment has no suite — and the governance layer is the part they have consistently under-built.

Not a prototype of an idea — a working application you can open.

BUILT AND RUNNING	STATUS
Agent loop: perceive → decide → act → record	shipped
Fifteen tools across read, solve and act	shipped
Four deterministic solvers, unit-tested	shipped
Risk register with expiring assumptions	shipped
Approval workflow with brief quality gate	shipped
Supplier portal with human takeover	shipped
Spreadsheet import, export and live sync	shipped
Audit trail, scorecard, counterfactual	shipped
Seven scenarios, six live injections	shipped
Real email transport, off by default	shipped

- NEXT, IN ORDER
- **Shadow mode on a real workbook** — one plant, one component family
 - Persistent supplier memory across runs, so reliability compounds
 - Multi-component allocation under a shared budget
 - Alternate-source qualification workflow (the slow loop the research flags)
 - Sub-tier exposure inference — the later data moat

DELIBERATELY NOT BUILT

No chat interface. No demand forecasting. No carrier or route optimisation. No live ERP writes. Each was considered and cut, because the value is in the decision loop and everything else dilutes it.

Ericsson did not lack data. It lacked a second opinion that arrived in time.

Every layer the industry has built since optimises for seeing further. The thing that decided a \$2.34bn outcome was whether anyone challenged one estimate, quantified the exposure, and put a real decision in front of a person while it still mattered.

That work is bounded, repetitive, evidence-based — and it is done today by a human, in forty hours, only when there is time.

THE PRODUCT

The decision loop, not another alert.

THE WEDGE

The mid-market that runs on a workbook.

THE MOAT

Evidence, governance, and a decision record that compounds.

Where each number came from — and how to read it.

PEER-REVIEWED / DOCUMENTED CASE

- **65% of inventory records inaccurate** — DeHoratius & Raman, *Management Science*, 2008 (~370,000 records)
- **Nokia / Ericsson, \$2.34bn** — Philips Albuquerque fire, 17 Mar 2000; supply-chain risk literature (Sheffi)
- **Aisin Seiki, 1997** — 99% of Toyota's P-valves from one plant, ~1 day of stock, recovery via 62 suppliers in days
- **Nexperia, Oct 2025** — a \$0.20 discrete stopped automotive lines; Honda guided a ~US\$960m operating-profit impact
- **JLR, Sep 2025** — five-week shutdown; £196m in-quarter cost; ~£1.9bn modelled UK economic loss

VENDOR AND INDUSTRY SURVEY — DIRECTIONAL, NOT AUDITED

- 8.7 h to detect vs 40.9 h to assess impact
- 22% of >\$1B shippers rate their control tower effective at driving action
- 75% need 3–10 systems to make one supply-chain decision
- 88% deploy supply-chain AI, 12% govern it (IDC)
- 50% autonomous decisioning by 2030, from 5% in 2025 (Gartner projection)
- Spreadsheets still the most-used supply-chain risk tool in 2026
- Expedite spend: ~3% of logistics cost at top performers vs ~10% at the bottom; ~49% of expedite events trace to planning failure
- ~31% of procurement time on manual and paper process (Ivalua)

We label these separately on purpose. They are directionally consistent across sources and good enough to choose a problem — not good enough to build a revenue model on.

Full research file: [docs/supply-chain-disruption-research.md](#) · problem → feature traceability: [docs/problem-to-feature-mapping.md](#) · architecture: [docs/ARCHITECTURE.md](#)