

— CASE STUDIES

Five teams. One question: what would a coworker take off your plate?

Short case studies from SaaS, financial services, healthcare, e-commerce and managed service providers — each showing what changed when a specialist coworker, not another dashboard, started doing the work.

5 case studies 5 sectors 4 coworkers

— CONTENTS

What these five have in common

None of these teams lacked skill or tooling. Each had dashboards, alerting, and engineers who knew their stack. What they shared was that the work between an alert and a fix — reconstructing context, correlating across layers, deciding whether an action was safe — still needed a person to do it, every time. The change was not another view of the same signals. It was a coworker that could act on them.

01 B2B SaaS & DevOps	p03
The pager stopped ringing for noise	
02 Financial Services & ITSM	p04
Tickets that arrive already half-solved	
03 Healthcare & Database	p05
Slow queries fixed before the portal felt it	
04 E-commerce & Retail	p06
Peak weekend, four coworkers, no war room	
05 Managed Service Providers	p07
One console, every client's estate	

THE RECURRING TRADE-OFFS

- **Specialists beat generalists.** A crew that knows compute, storage, database and network individually resolves more without escalating than one generic bot ever does.
- **Trust is built per pattern, not all at once.** Every team started narrow — advisory or human-in-the-loop — and widened autonomy only as the record of clean resolutions grew.
- **Context beats another alert.** Reading tickets, Slack and email alongside telemetry is what turns a page into a pre-solved ticket instead of a blank form.
- **The cost lands outside the SLA.** It shows up as attrition, audit overhead, over-staffed peak events and coverage that only scales with headcount.

01 B2B SAAS & DEVOPS

The pager stopped ringing for noise

SECTOR

Multi-cloud SaaS platform

COWORKER

Compute Coworker

RESOLUTION MODE

Autonomous for known patterns

DRIVER

Overnight page fatigue

THE CHALLENGE

- Noisy-neighbour pods **paged engineers overnight** for pressure that would self-correct
- Scaling was reactive, arriving after user-facing latency had already degraded
- Three cloud consoles made confirming a transient issue slower than fixing it
- Senior engineers were the informal escalation path for anything unfamiliar

THE SOLUTION

- Compute Coworker **reads pod, node and cluster telemetry** continuously across all three clouds
- Distinguishes transient pressure from genuine shortfall using prior-incident history
- Right-sizes and scales autonomously** for known patterns; escalates anything unfamiliar
- Every autonomous decision logged and reviewable, sharpening the next judgement call

THE OUTCOME

- Transient pressure **resolved before anyone is paged**
- One place to see resource state across all three clouds
- Escalation reserved for genuinely novel patterns
- A record of every decision, reviewable after the fact

We were not short of automation. We were short of judgement about which page was real. Once the coworker held that line, the on-call rotation stopped dreading the noisy nights.

— HEAD OF PLATFORM ENGINEERING

02 FINANCIAL SERVICES & ITSM

Tickets that arrive **already half-solved**

SECTOR

Enterprise financial services

COWORKERS

Network · Database · Compute

RESOLUTION MODE

Human-in-the-loop for production changes

DRIVER

Audit & reconstruction cost

THE CHALLENGE

- Every ticket began as a **blank form**, with root cause and impact reconstructed by whoever picked it up
- Regulated change meant evidencing the reasoning behind an action, which most tickets never had
- Cross-team handoffs for network or database work reset the investigation each time
- Repeat incidents were reinvestigated because nothing linked them to a prior pattern

THE SOLUTION

- Reads telemetry, tickets and email to **reconstruct root cause and impact** before assignment
- Drafts a **root-caused remediation** with evidence attached, not a bare recommendation
- Production-touching changes **wait for a named approver** who reviews evidence, not a blank page
- Every recommendation, approval and outcome logged and attributed for audit

THE OUTCOME

- Tickets arriving with **root cause and impact already attached**
- Evidence generated as a property of the investigation itself
- Named approval on production changes, evidence-backed
- Repeat patterns recognised and linked automatically

Our auditors do not want speed at the expense of evidence. What changed is that the evidence now arrives with the ticket instead of being assembled after the fact.

— DIRECTOR OF IT SERVICE MANAGEMENT

03 HEALTHCARE & DATABASE

Slow queries fixed before the portal felt it

SECTOR

Digital health provider

COWORKER

Database Coworker

RESOLUTION MODE

Auto-fix + pre-filled
DBA ticket

DRIVER

Small DBA team,
patient-facing risk

THE CHALLENGE

- Reporting queries competed with patient-facing traffic, **degrading the portal** before anyone noticed
- Index tuning was reactive, applied only after a slow-query report surfaced
- Scaling was always an option, but never actually weighed against a targeted fix
- Every issue, however minor, routed to a small DBA team regardless of size

THE SOLUTION

- **Continuous query profiling** correlating reporting load against patient-facing traffic
- Tunes indexes and configuration **autonomously** for patterns resolved cleanly before
- Anything touching schema arrives as a **pre-filled, root-caused ticket** for a DBA
- Recommendations weigh **FinOps and GreenOps context** — a fix against a scale-up

THE OUTCOME

- Query pressure **caught before it reaches patients**
- Index tuning applied autonomously for known patterns
- DBA tickets arriving pre-profiled and root-caused
- Portal latency stable through peak hours

Our DBAs were spending their time on tickets that turned out to be one bad index. Now the ones that reach them already have the profiling done.

— VP OF ENGINEERING

04 E-COMMERCE & RETAIL

Peak weekend, four coworkers, no war room

SECTOR

Global online retailer

COWORKERS

Compute · Storage ·
Network

RESOLUTION MODE

Auto: compute &
storage; advisory:
network

DRIVER

Peak bridge cost

THE CHALLENGE

- Checkout incidents **crossed layers by default** — a storage bottleneck read as compute or network
- Each layer had its own team and on-call, so a cross-layer incident meant three people joining a call
- The bridge ran the whole peak window as insurance, regardless of active incidents
- Post-incident learning was thin — everyone was managing, not documenting

THE SOLUTION

- Compute pressure, storage saturation and network drift **recognised as one incident**, not three pages
- The orchestrator decides which layer moves first, rather than three teams reasoning it live
- **Compute and Storage Coworkers act autonomously** on known patterns; Network holds for approval
- Each incident updates the pattern library for next year

THE OUTCOME

- Compute and storage patterns **resolved autonomously**
- The bridge convened only when genuinely needed
- Cross-layer sequencing decided before humans are paged
- Peak readiness measured in what closes itself

We used to measure peak readiness in how fast we could get three teams on a call. This year the question was whether the call needed to happen at all — and most of the time, it didn't.

— VP OF SITE RELIABILITY

05 MANAGED SERVICE PROVIDERS

One console, **every client's estate**

SECTOR

Regional datacenter operator

COWORKERS

Compute · Storage · Database · Network

RESOLUTION MODE

Autonomous for known patterns, per client

DRIVER

Coverage vs. headcount

THE CHALLENGE

- Each client's monitoring stack was **separate**, so context-switching consumed as much time as the work
- Facilities telemetry sat apart from IT telemetry, so a cooling issue became two investigations
- Overnight coverage meant a rotating on-call across every client at once
- Growth meant hiring — one more client meant one more set of eyes on one more dashboard

THE SOLUTION

- Continuous telemetry across **every client environment**, facilities data included via DCIM and BMS
- Each client's signals, patterns and history **kept separate**, so coverage scales without collapsing tenants
- **Known patterns resolve autonomously** per client, per coworker
- Facilities and IT incidents **correlated automatically**, closing a costly investigation gap

THE OUTCOME

- One coworker watching every client, data isolated
- Facilities and compute incidents correlated automatically
- More clients served without proportional hiring
- Consistent coverage regardless of shift or headcount

We were selling coverage and delivering headcount. Now we sell coverage and deliver a coworker that watches forty environments as attentively as it would watch one.

— CHIEF OPERATING OFFICER

— THE PATTERN

A dashboard is not a coworker

The distinction is structural. A dashboard shows you the estate and waits for a person to act on it. A coworker reads the same signals and does the work — autonomously where the pattern is known, alongside you where it is not.

WHAT A DASHBOARD ASSUMES

- That someone is watching it at the moment it matters
- That the person watching can tell noise from a real incident
- That a fix, once identified, is theirs to apply
- That context from six months ago is still in someone's head

WHAT A COWORKER PROVIDES INSTEAD

- A **specialist per layer** — compute, storage, database, network
- Resolution mode set **per pattern**: autonomous, human-in-the-loop, or advisory
- Action taken directly, inside the guardrails you set
- Every decision logged, attributed, and folded into the next one

BOTTOM LINE

See it take a real incident from alert to resolved — on your stack.

Meet the coworker who works the night shift.

Get a live demo, or talk to our team about ITOps Coworker for your estate.



GET IN TOUCH
lumiops.ai/contact-us



POC
~2 weeks to value



LINKEDIN
linkedin.com/company/lumiops