

— CASE STUDIES

Five teams. One question: whose knowledge is doing the work?

Short case studies from retail, payments, SaaS, telecom and healthcare — each showing what changed once a team's own operational know-how became a governed agent instead of a person's standing habit.

5 case studies

5 sectors

4 stages

— INSIDE THIS BOOKLET

01

E-commerce & Retail

A backlog that stopped outgrowing the team

02

Fintech & Payments

Every rotation now has a name attached

03

B2B SaaS

Cleanup that finally pays for itself

04

Telecom & Network

The best engineer's instincts, every shift

05

Healthcare

Risk caught before it reached a patient

Governed AI agents for enterprise IT operations — built from what your team already knows.

— CONTENTS

What these five have in common

None of these teams lacked expertise. Each one already had a person who could explain the fix in a couple of sentences. What they lacked was a way to turn that sentence into something that ran the same way every time, without waiting on that person's calendar — and without giving up the approval step that made it safe to run at all.

01	E-commerce & Retail The ticket queue stopped growing faster than the team	p03
02	Fintech & Payments Every automated action now has a name attached to it	p04
03	B2B SaaS The cloud bill finally matched what the product actually used	p05
04	Telecom & Network Operations Routing incidents stopped waiting on one person's calendar	p06
05	Healthcare & Digital Health Database risk got caught before it reached a patient	p07

— THE RECURRING PATTERN

The objective was always the hard part to skip, not to write.

Once someone could state the fix in a sentence, drafting the workflow around it took minutes, not a project.

One gate did more work than ten restrictions.

Every team found that a single, well-placed approval step was what let them automate everything around it.

Promotion is what made it durable.

A working draft is fragile. A promoted agent, with its objective and thresholds attached, outlives the person who built it.

Nobody scaled by hiring for this.

In every case, the fix was headcount-proof — it ran the same way whether or not the original engineer was even at their desk.

— AT A GLANCE

SECTOR	STARTING OBJECTIVE	AUTONOMY	PRIMARY DRIVER
E-commerce & Retail	Investigate crash loops, clean up disks	Cleanup automatic, restarts gated	Backlog outgrowing the team
Fintech & Payments	Rotate aging IAM keys with approval	Detection automatic, rotation gated	Compliance review
B2B SaaS	Find and clean up unused disks	Non-prod automatic, prod tags gated	Untracked cloud waste
Telecom & Network	Network Specialist foundation	Investigation only, no actions	Single point of expert failure
Healthcare	Database Specialist foundation	Investigation continuous, fixes gated	A near-miss nobody wanted repeated

01 — E-COMMERCE & RETAIL

The ticket queue stopped growing faster than the team.

SECTOR

Global online retailer

START POINT

Two described objectives

AUTONOMY

Cleanup automatic;
restarts gated

DRIVER

Backlog outgrowing the
team

The situation

- Container restarts and crash loops in checkout services followed the same three or four patterns, diagnosed fresh each time.
- Tickets sat in a shared queue with no enrichment, so the first fifteen minutes of anyone's shift went to figuring out what a ticket even meant.
- The engineers who wrote the original runbooks had mostly moved to other teams, so the documents existed but nobody trusted them.
- Cleanup tasks — old snapshots, idle test environments — got batched into a quarterly sprint because nobody had time for them weekly.

How it was built

- **Describe.** Two objectives written in a sentence each — investigate the crash loop, reclaim unused disks — became draft workflows within the hour.
- **Compose.** The team added a flow-control branch for the two crash-loop root causes they actually see, plus a human checkpoint before any pod gets force-restarted in production.
- **Govern.** Restarts in production require a named approver; disk cleanup in non-production runs unattended within a fixed monthly ceiling.
- **Promote.** Both workflows now run as standing agents with defined success thresholds, so a new hire inherits the pattern instead of re-learning it.

What changed

- Crash-loop root causes tested automatically, restart gated
- Tickets arrive enriched with context attached
- Runbook logic encoded and versioned in the agent
- Disk cleanup running on a fixed monthly rhythm
- New hires inherit a working pattern on day one

"We didn't sit down and design an agent. We described what a good Tuesday afternoon fix looks like, and it built the first draft from that."

— STAFF ENGINEER, PLATFORM TEAM

— CAPABILITIES DOING THE WORK

Objective-led workflow drafting

Compute Specialist

Storage Specialist

Approval-gated restarts

ITSM ticket enrichment

Versioned promoted agents

Fixed cleanup ceilings

Full run audit trail

— GROWTH OUTRAN THE RUNBOOK

Order volume doubled two years running. Headcount for the platform team didn't. The gap got filled the usual way — more tickets sitting longer, more context rebuilt from scratch each time someone finally picked one up.

02 — FINTECH & PAYMENTS

Every automated action now has a name attached to it.

SECTOR

Payments platform

START POINT

Rotate aging IAM keys
with approval

AUTONOMY

Detection automatic;
rotation gated

DRIVER

Compliance review

The situation

- IAM key rotation ran on a cron job with a shared service account — effective, but attributable to nobody in particular.
- When a rotation broke something, reconstructing what changed meant reading commit history and hoping the right person remembered the context.
- Security wanted rotation to happen more often. Engineering wanted fewer things that could break payment flows unattended.
- Neither side trusted a plain script enough to widen its blast radius, so rotation stayed manual and overdue.

How it was built

- **Describe.** 'Rotate aging IAM keys with approval' became the seed objective — already close to how the team described the problem out loud.
- **Compose.** A skill checks key age against policy on a schedule; a human step inserts itself only where the credential touches production.
- **Govern.** Every rotation on a production key requires a named approver and records who signed off, when, and against what evidence.
- **Promote.** The workflow now runs as a standing agent with its own performance threshold — rotations completed on time, without a single unattributed change.

What changed

- Every rotation attributed to a named approver
- Detection continuous; the risky step still requires a person
- Security gets more frequent checks, engineering keeps the gate
- Evidence attached to every approval request automatically
- Compliance review treating automation as the safer option

"Neither team was wrong. Security wanted more rotation, engineering wanted less risk. The agent didn't resolve the argument — it just meant we didn't have to choose one before we started."

— HEAD OF PLATFORM SECURITY

— CAPABILITIES DOING THE WORK

Approval gates on production changes

Named-approver audit trail

Policy-driven key age checks

Deep identity-provider integration

Promoted, versioned agent

Evidence-attached approvals

Scoped access boundaries

Continuous detection

— SPEED WITHOUT A PAPER TRAIL IS A LIABILITY

A payments company can automate almost anything technically. What stops most of them is simpler: if an auditor asks who approved a production change six weeks ago, 'the script did it' is not an answer anyone wants to give.

03 — B2B SAAS

The cloud bill finally matched what the product actually used.

SECTOR

B2B SaaS platform

START POINT

Find and clean up unused disks

AUTONOMY

Non-prod automatic;
prod tags gated

DRIVER

Untracked cloud waste

The situation

- Engineers spun up test environments for a sprint and moved on; a meaningful share were never torn down.
- FinOps could see the anomaly in the dashboard but had no direct way to act on infrastructure they didn't own.
- A quarterly cost review would surface the same categories of waste, get triaged into tickets, and mostly not get done before the next review.
- The one engineer who reliably found and fixed this stuff was also the one everyone needed for actual incidents.

How it was built

- Describe.** 'Find and clean up unused cloud disks' — one of the quick-start objectives — became the first promoted agent inside a week.
- Compose.** The FinOps Agent correlates billing data with tags and usage, and a flow-control step separates test-tagged resources from anything touching production.
- Govern.** A per-run cost ceiling caps how much gets reclaimed in one pass, and anything tagged production routes to a human instead of executing.
- Promote.** The agent now runs on a schedule with its own cost-saved metric tracked alongside its own operating cost — the ROI is visible, not assumed.

What changed

- Non-production waste reclaimed on a standing schedule
- FinOps and the agent share the same view and the same action
- Cleanup running continuously instead of quarterly
- Production-tagged resources still routed to a human
- Cleanup work no longer bottlenecked on one person

"FinOps had always been able to see the problem. What changed is that seeing it and fixing it became the same step, instead of a dashboard and a ticket and a wait."

— VP OF ENGINEERING

— CAPABILITIES DOING THE WORK

FinOps Agent

Tag-based routing

Per-run cost ceilings

Continuous anomaly detection

Production approval gate

Cost-saved tracking

Light discovery start

Scheduled promoted agent

— WASTE THAT NOBODY OWNS

Idle compute and forgotten disks are nobody's job to hunt down. They're not urgent, they're not exciting, and by the time finance flags the bill, three more months of waste have already accumulated behind them.

04 — TELECOM & NETWORK OPERATIONS

Routing incidents stopped waiting for the one person who knew the topology.

SECTOR

Telecom & network operations

START POINT

Network Specialist foundation

AUTONOMY

Investigation only, no actions

DRIVER

Single point of expert failure

The situation

- Latency and connectivity issues across the hybrid and multicloud footprint needed someone who held the whole topology in their head.
- That person was one of three people on a rotating on-call schedule; the other two escalated more often than the numbers justified.
- Investigation meant pulling logs from four different tools by hand before anyone could even state the hypothesis.
- Postmortems captured what happened but rarely made it back into anything the next on-call person could actually use at 2am.

How it was built

- **Describe.** The team started from the Network Specialist foundation rather than a blank objective — the shape of the problem was already familiar.
- **Compose.** Skills for topology lookup, latency comparison and routing-table diffing were wired together with the team's own escalation logic.
- **Govern.** Access scope is read-only by design for this agent; it produces a ranked diagnosis, not an executed change.
- **Promote.** Now standing and available to every shift, with performance tracked against how often its top hypothesis matched what the senior engineer would have said.

What changed

- Every shift starts with the same depth of context
- Escalation reserved for genuinely novel cases
- Logs and topology correlated automatically at the first alert
- Each incident's diagnosis pattern folded back into the agent
- Expertise available on every shift, not just some

"We didn't need the agent to fix anything. We needed the two-in-the-morning version of every engineer to start where our best engineer would have started."

— DIRECTOR OF NETWORK OPERATIONS

— CAPABILITIES DOING THE WORK

Network Specialist

Read-only investigation scope

Topology-aware diagnosis

Routing-table comparison

On-call context handoff

Escalation reduction

Deep telemetry integration

Diagnosis pattern reuse

— TRIBAL KNOWLEDGE, SINGLE POINT OF FAILURE

Every network team has one person who can look at a routing table and just know where the problem is. That's a compliment to them and a risk to everyone else — the response is only as fast as their calendar.

05 — HEALTHCARE & DIGITAL HEALTH

Database risk got caught before it became a patient-facing outage.

SECTOR

Digital health platform

START POINT

Database Specialist
foundation

AUTONOMY

Investigation continuous;
fixes gated

DRIVER

A near-miss nobody
wanted repeated

The situation

- Capacity and error-pattern trends lived across several dashboards that nobody checked with the same discipline every day.
- The team that could act on database risk was small and mostly consumed by on-call and change requests.
- Sensitive patient data meant most tooling vendors were a hard no before a conversation about capability even started.
- A single connection-pool exhaustion incident had already touched a clinical workflow once — a near miss the team didn't want to repeat.

How it was built

- **Describe.** The objective was capacity and error-pattern risk on one clinical database tier — narrow on purpose, given what was at stake.
- **Compose.** Skills watch connection pool saturation, query latency and error rate together, rather than any one signal in isolation.
- **Govern.** Reasoning runs entirely in-boundary; any remediation step — killing a query, scaling a pool — requires a named approver.
- **Promote.** Now a standing agent with a defined error budget: past a set number of missed early-warnings, the pattern escalates for review rather than continuing quietly.

What changed

- Capacity and error patterns watched continuously, in-boundary
- Early warning reaching the team before a threshold breach
- Remediation still gated on a named approver
- No patient-adjacent telemetry leaving the environment
- A near miss that hasn't repeated since

"We could not send patient-adjacent telemetry to an external model, full stop. Reasoning that stays inside our boundary was the only version of this we were ever going to be allowed to run."

— DIRECTOR OF PLATFORM ENGINEERING

— CAPABILITIES DOING THE WORK

Database Specialist

In-boundary reasoning

Continuous capacity monitoring

Approval-gated remediation

Error budget & escalation

Light discovery start

No external data egress

Early-warning investigation

— SLOW DECAY, SUDDEN FAILURE

Database performance problems in a clinical system rarely announce themselves early. They creep — a slow query here, a connection pool a little tighter than last month — until the day they don't creep anymore.

— THE PATTERN

Knowledge scales when it stops living in one head.

Every team in this booklet already had the answer before they had the agent. What changed wasn't the expertise — it was whether that expertise could run without waiting on the person who held it, inside limits that made everyone else comfortable letting it run.

WHAT TRIBAL KNOWLEDGE ASSUMES

- That the right person is awake and available.
- That re-explaining context each time is an acceptable tax.
- That a fix survives its author leaving the team.
- That approval and speed are always in tension.

WHAT A PROMOTED AGENT PROVIDES INSTEAD

- The same starting point on every shift, every time.
- Context assembled automatically, not reconstructed by hand.
- A fix that outlives the person who first described it.
- One well-placed gate instead of a blanket no.

— ACROSS ALL FIVE TEAMS

5

SECTORS COVERED

4STAGES, START TO
STANDING AGENT**60%**LESS MANUAL TRIAGE, ON
AVERAGE**100%**RUNS LOGGED AND
ATTRIBUTED

Customers anonymised at their request; profiles and quotations shown are representative of deployments in each sector, not records of named engagements. Outcomes described are illustrative and vary by estate size, objective scope and guardrail configuration. Third-party product names are trademarks of their respective owners. © 2026–2027 LumiOps.AI.

— IN THEIR WORDS

"We didn't sit down and design an agent. We described what a good Tuesday afternoon fix looks like..."

**STAFF ENGINEER,
PLATFORM TEAM**

"Neither team was wrong. Security wanted more rotation, engineering wanted less risk. The agent..."

**HEAD OF PLATFORM
SECURITY**

"FinOps had always been able to see the problem. What changed is that seeing it and fixing it became..."

VP OF ENGINEERING

"We didn't need the agent to fix anything. We needed the two-in-the-morning version of every..."

**DIRECTOR OF
NETWORK
OPERATIONS**

"We could not send patient-adjacent telemetry to an external model, full stop. Reasoning that stays..."

**DIRECTOR OF
PLATFORM
ENGINEERING**

— THE SHAPE, EVERY TIME

1A person could already explain the
fix**2**The objective became a draft
workflow**3**

One gate made the risky step safe

4Promotion made the fix outlast its
author

— IN CLOSING

Your team already knows the fix. Give it somewhere to run.

Every story here started the same way — a sentence describing what a good fix looks like. The work after that was setting the boundary, not writing the automation from scratch.

60%

LESS MANUAL TRIAGE

45%

FASTER TO FIRST RUN

3×

MORE AGENTS PROMOTED

\$18

PER AUTOMATED RUN

— WHERE TO START

Pick one recurring fix.

The task your team already re-does by hand most often.

Describe it in a sentence.

That's the seed objective — the draft builds around it.

Gate the risky step, run the rest.

One approval point usually covers the whole conversation.

Put Agent Builder to work.

Describe an objective, watch it draft the workflow, and promote what works into a standing agent.

GET IN TOUCH

contact@lumiops.ai

PLAYGROUND

[Try it, no setup](#)

LINKEDIN

linkedin.com/lumiops