RAPAX

Service Assurance for the Lights-Out NOC

WHITE PAPER • TIER 1 AUTOMATION

# Tired of Tickets?
# Ask Frank.

AI-Native Tier 1 Support Automation for Telecom Operators

> *FRANK — Frontline Response And Knowledge Navigator*

Deflect 40% of incoming support load. Auto-close 15% of tickets. Defer your next support-engineer hire — in under three weeks of payback.

**By Shawn Ennis**
Founder & CEO, Rapax
May 2026 • Rapax | Citus Technologies, LLC

SECTION 01

# Executive Summary

Your support team is drowning. Industry data consistently shows that **60–70% of all support tickets** don't actually require human judgment — they're questions answerable from your knowledge base, incomplete reports waiting for missing logs, duplicates, or false alarms. In a 50-person support organization, that's the equivalent of **10 engineers and $5M per year** spent on work a machine should be doing.

Frank is the Tier 1 AI agent built into Rapax incident management. Frank intercepts incoming tickets before they reach an engineer, deflects what your knowledge base can already answer, closes routine questions automatically, and routes everything else to a human with the right context attached. Frank does not replace engineers — Frank gives them back the hours they're losing to triage.

### What this paper covers

The economic case for AI-native Tier 1 automation, what Frank does today on the Rapax support desk, three deployment scenarios with payback math, and a 6-week adoption path that doesn't require you to rebuild your knowledge base first.

## The numbers, briefly

**$661K–$1.8M**
Annual value delivered, depending on team size and KB maturity

**7–19 days**
Payback period across all three deployment scenarios

**4–9 FTEs**
Support hires Frank lets you defer over 12 months

CONTENTS

# What's in this paper

SECTION 02

# The Last Engineer Who Knew

Twenty-five years ago, when I started in telecom operations, a NOC engineer carried a pager. They'd sleep at the office some nights. They knew every alarm, every rule, every quirk of the network by heart. When something broke at 2 AM, they were the person everyone called.

That engineer was invaluable because they'd seen it all. They knew that a 10% CPU spike on this ONT was fine, but the same spike on that router meant you had four minutes before the backhaul link went down. They could read tea leaves in raw log output.

But here's the brutal reality: that engineer is gone. They retired. Or they moved on. And if you're like most Tier 2 and Tier 3 operators, you're facing a problem that keeps you up at night.

> **Your support desk isn't broken — it's drowning.**

You've hired more people. You've bought better tools. You've implemented ITSM processes. And yet, your team still spends 70% of their time on work that doesn't actually require a six-figure engineer.

They're answering questions that are already in your knowledge base. They're asking customers for the same logs 50 times a month. They're being paged at midnight for false alarms any automated classifier could have caught. And they're frustrated because they went into telecom operations to solve hard problems — not to be a glorified search engine.

This is the paradox: **the busier your support team gets, the less strategic they become.**

SECTION 03

# Support Theater, Not Support Capacity

## What the data shows

Industry research consistently shows that 60–70% of support tickets don't actually require human judgment. They break down like this:

| Ticket Type | Share of Volume | Why it shouldn't need a human |
|---|---|---|
| **KB-answerable questions** | 40–50% | Password resets, SNMP field references, configuration look-ups |
| **Incomplete reports** | 15–20% | Missing logs, reproduction steps, or affected systems |
| **Duplicate tickets** | 5–10% | Same issue, multiple openers, no cross-link |
| **False alarms** | 3–5% | Monitoring false positives misclassified as incidents |
| **Real engineering work** | 15–25% | This is what your team should actually be solving |

## The business cost of triage

A fully loaded support engineer in the telecom sector costs $150,000–$200,000 per year when you include salary, benefits, and overhead. That's roughly $75–$100 per hour.

Each ticket — from creation to closure — costs money. Not just the engineer's time to solve it, but the time to read it, understand it, ask for missing information, classify it, route it, and manage it.

A typical support engineer spends 2–3 hours per working day on pure triage activities: reading incoming tickets, deciding which team should handle it, asking for more information, or looking up answers that already exist in documentation.

**$5,000,000**

Annual cost of triage work in a 50-engineer support team — equivalent to 10 full-time engineers

## The customer impact

The cost isn't just internal. The customer experience suffers. When your support team is drowning in triage, first-response times stretch. SLA breaches become common.

And here's the kicker: the customers calling in aren't naive. They know you have a knowledge base. They see documentation in your portal. They're frustrated they called a human instead of finding an answer in 30 seconds.

Frustrated customers escalate emotionally. They demand executive escalations. They threaten to switch vendors. And now your on-call engineer is dealing with a legal threat at midnight — when the real issue was “how do I enable DHCP snooping?”

## The strategic problem

The deepest cost is strategic. Your best engineers — the ones who could be redesigning your network, optimizing costs, or building the next-generation platform — are stuck answering email. They're not learning new skills. They're not moving into architect roles. They're not mentoring junior engineers. They're processing tickets.

And you can't hire your way out of this. Every engineer you hire spends 30% of their first year ramping. Every one costs $75+ per hour. And if you hand them a pile of triage work, you haven't solved the problem — you've just bought time.

SECTION 04

# Meet Frank — Your Tier 1 AI Agent

Frank is an AI-native support agent built into Rapax incident management. Frank runs in the background, 24/7, and intercepts support work before it reaches your engineers.

## Five decisions on every ticket

| Decision | What Frank does | Customer outcome |
|---|---|---|
| **Question deflection** | Customer emails a question answerable from your KB; Frank replies directly | Answer in 30 seconds — no ticket, no human |
| **Auto-classification** | Reads the ticket and categorizes as question, bug, or feature; corrects the opener if the signals point elsewhere | Right team gets the ticket the first time |
| **Intake validation** | Checks for required information; if missing, Frank asks for it automatically | Engineer starts solving — not waiting for logs |
| **False-alarm detection** | Cross-checks customer-reported outages against live telemetry; downgrades priority if monitoring shows green | Real outages stay urgent; noise stays normal |
| **Escalation & frustration** | Routes human-needed tickets with full context; pages on-call if customer tone shifts to angry or legal | SLA holds; emotional escalations get caught early |

## Why this works

Frank doesn't replace engineers. **Frank gives you back the time you're wasting on the wrong problems.** Every hour Frank saves your team is an hour they can spend on real incidents, strategic work, or mentoring junior staff.

And Frank is grounded. It doesn't make up answers. It queries Wade — your knowledge base — and only responds when it has a high-confidence match. If an answer isn't in your KB, the ticket goes to a human. No hallucinations. No invented fixes.

### Grounded, not generative.

Frank only acts when Wade returns a high-confidence match against your own documentation. Below the threshold, the ticket escalates to a human — every time. You set the threshold; Frank respects it.

## Market validation

Frank isn't theoretical. It's running live on Rapax's own support desk today, handling real tickets from telecom operators, managed service providers, and enterprises. We've seen it work on the exact scenarios your team faces:

- **Routine KB hit.** A customer asks “Which alert fields are used for Up/Down correlation?” Frank finds a KB article. The customer gets their answer. The ticket auto-closes. Zero human time spent.
- **Incomplete report.** A customer reports a crash with a stack trace but missing OS version and repro steps. Frank replies automatically asking for the missing pieces. By the time an engineer opens the ticket, it's a complete report.
- **Duplicate pattern.** Two identical “app is broken” reports come in an hour apart from the same company. Frank links them with an internal note so the engineer sees the pattern immediately.
- **Frustration spike.** A customer sends a follow-up threatening legal action over a 72-hour silence. Frank detects the escalation language and pages on-call before the SLA breach lands on the dashboard.

SECTION 05

# The Numbers: Three Deployment Scenarios

The value of Frank depends on your starting point: team size, ticket volume, and the maturity of your knowledge base. Here's what different operators can expect — all three scenarios assume a $50,000 annual investment ($25K for Frank, $25K for Wade).

| Scenario | Annual Value | Payback | FTEs Deferred |
|---|---|---|---|
| **Conservative**<br>Smaller team, newer KB | $661K | 19 days | 4 FTEs |
| **Moderate**<br>Industry-typical operator | $1.16M | 11 days | 6 FTEs |
| **Optimistic**<br>Mature KB, high adoption | $1.80M | 7 days | 9 FTEs |

## Where the value comes from

The $661K to $1.8M in annual value breaks down into three categories:

- **Pre-ticket deflection (35–40% of value).** Frank answers trivial questions before a ticket is ever created. No triage. No engineer time. The single highest-leverage lever.
- **Auto-close (40–45% of value).** Tickets confirmed as questions with a KB answer close automatically. Your engineers never read them. This stops the bleed.
- **Intake efficiency (15–20% of value).** For tickets that do need a human, Frank collects the required information upfront. The engineer starts solving immediately.

## The FTE-avoidance angle

This is the metric CFOs and CTOs gravitate to: Frank doesn't reduce cost in some abstract way. Frank lets you defer hiring your next support engineer — by six months to two years depending on your growth curve.

In the moderate scenario, Frank avoids six new hires over twelve months. That's six × $150K in salary, benefits, and ramp-up cost that stays in the bank. The investment in Frank is roughly 3% of the headcount it offsets.

SECTION 06

# The 6-Week Adoption Path

## Weeks 1–2: Assessment and KB readiness

First, we inventory what you have. If you already operate a knowledge base (Confluence, Zendesk Help Center, a custom wiki), we export it. If you don't, we build one from your most common support questions and internal runbooks.

**This is not a blocker.** Even a basic 50-article KB is enough to start deflecting 20–30% of incoming questions. You grow the KB over time. We want to start with the low-hanging fruit: “How do I reset my password?” “What's the difference between an ONT and an OLT?” “Where's the API documentation?” Those articles cover half your tickets today.

## Weeks 3–4: Data migration and integration

If you're moving from ServiceNow, Jira, or Zendesk to Rapax, we export your historical tickets via API or CSV and normalize the data (date formats, deduped users, standardized priority levels). Then we load everything into Rapax.

If you want to keep your existing ticketing system, we integrate via REST API so Frank reads tickets and pushes decisions back. Rapax professional services handles the integration — no custom coding on your side.

## Weeks 5–6: Configuration, warm-up, and go-live

We configure Frank's confidence thresholds — how confident Frank needs to be before it deflects, auto-closes, or escalates. Every operator has a different risk profile. A large MSP might run Frank at 0.85 confidence (deflect only on near-certain KB matches). A smaller team might run at 0.65 to catch more edge cases manually during the warm-up.

Then we run a warm-up period: all of Frank's decisions are logged and reviewed by your team, but no automatic actions fire yet. You see what Frank would do, tune the thresholds, and then you flip the switch.

## Risks and how we mitigate them

| Risk | Why it matters | How we handle it |
|---|---|---|
| **Knowledge base quality** | If your KB is incomplete or out-of-date, Frank's deflection rate drops | Frank ships with an audit dashboard. You see exactly what's deflecting, what's auto-closing, and what's escalating. You use that data to fill the gaps. Over 3–6 months you build a living KB. |
| **Threshold tuning** | Set too high, Frank never fires; set too low, customers get frustrated by wrong answers | We provide templates by industry and company size, then tune from the warm-up results before go-live. |
| **Change management** | Your team may resist — “what if Frank makes a mistake?” | Transparency. Show your team the warm-up results. Prove Frank reduces their workload, not their job. Engineers who were losing three hours a day to triage suddenly get those hours back. |

SECTION 07

# The Bottom Line

You're facing a choice. You can keep hiring support engineers to do work a machine should be doing. You can keep running SLA-breach reviews. You can keep watching your best people move into architect roles elsewhere because your team became a ticket factory.

### Or you can change the game.

Frank costs $50,000 per year. In every realistic scenario, it pays for itself in under three weeks. It deflects 40% of your incoming load. It closes 15% of tickets without human review. It catches false alarms and frustrated customers before they become crises.

Most importantly, Frank gives your engineers back their time. Time to solve hard problems. Time to learn. Time to build the next generation of your platform.

 

## That's worth $50K.

*And it's worth a lot more than $50K.*

### Next steps

If this resonates, we're ready to talk. No gatekeeping. No long contracts. No tie-ins. Your data stays yours. You can shut Frank off any time.

## Book 15 minutes with Rapax

We'll walk you through a sample warm-up, show you the deflection dashboard, and answer your questions — no demo theater.

**cal.com/shawn-ennis**

sales@rapax.app • rapax.app

ABOUT

# The Author & The Company

## Shawn Ennis — Founder & CEO, Rapax

Shawn Ennis is the founder and CEO of Rapax and Citus Technologies. With 25+ years of experience in telecom operations and service assurance, Shawn previously founded Assure1 (acquired by Oracle in 2021), where he led AI-first initiatives for service assurance. He's a published author on NOC automation, holds 12 patents in telecom OSS/BSS, and hosts the Transformation Leaders Podcast.

Shawn is based in the Dallas–Fort Worth area and is passionate about helping operators do less busy work and more strategic work.

> **Connect:** linkedin.com/in/shawnennis
> **Podcast:** Transformation Leaders Podcast (available on all platforms)
> **Email:** shawn@citus.cloud

## Rapax

Rapax is an AI-native service assurance platform built for telecom operators. The platform includes Frank (AI Tier 1 support agent), Wade (AI knowledge base and query engine), incident management, alert correlation, performance analytics, and Bruce (the AI-native integration builder). Rapax is deployed in production today and is trusted by operators managing network elements across North America.

**Learn more:** rapax.app
**Contact:** sales@rapax.app
**Book a call:** cal.com/shawn-ennis