L&T Technology Services

The network optimises itself.

The intelligence behind it is engineered.

CASE STUDY



EXECUTIVE SUMMARY

The client is a **Tier-1 network equipment OEM and one of the world's top-10 brands.** It sells operators a promise: a radio network that optimises itself. Honouring that promise operator by operator, on live networks and across mixed-vendor estates, is an engineering workload no OEM absorbs alone.

LTTS has carried a large part of it for **eight years.** Our team customizes the OEM's network-automation platform for **25+ of the OEM's own operator customers,** sustains the LTE software running inside those operators' live networks, and built the machine learning and code generation that made the work scale. The engagement has moved from executing work packages to co-developing platform features with the OEM's product line.

THE CHALLENGE

Radio networks used to be tuned by hand. That stopped being possible years ago. A large operator runs millions of cells, each one affecting its neighbours, with fresh performance data arriving every fifteen minutes. Self-Organising Networks, or SON, exist to make those adjustments automatically.

Selling that automation is one thing; making it work inside each operator's network is another. Every operator has a different mix of radio equipment, a different software release and a different set of performance targets it is measured against. And underneath the automation sits the LTE software itself: still carrying live subscriber traffic, still under the OEM's service commitments, still needing new features and fixes on schedule.

The OEM needed a partner who could work at **both levels at once.**

THE ENGINEERING COMPLEXITY

Automating a live radio network is harder than it sounds.

01

The rulebook stops short of the answers. The industry standard says how this automation should be managed and measured. It does not say how the automation should decide anything: the standards body states outright that these functions are implementation specific and appear in no public standard. So every optimisation module is original work, and every mixed-vendor deployment is custom engineering.

The automation can work against itself and can be wrong even when it doesn't. Different functions want opposite things. One raises a cell's transmit power to reach users at the edge; another lowers it to stop interfering with the cell next door. Both move the cell boundary, which changes what a third function sees when it tries to balance traffic. The standard treats preventing that as a formal requirement, not an edge case. Coordination alone is still not enough: a perfectly non-conflicting set of changes can leave the network worse off, so every change is checked against real performance data after it lands and reversed if it was wrong.

02

03

Every software release is a compatibility problem. The platform sits above equipment it does not control, from several manufacturers, each on its own release cycle and each describing its settings differently. Basic plug-and-play between vendors only arrived four standard releases after the automation itself.

WHAT LTTS ENGINEERED

Three strands of one engagement, each answering a difficulty above.



Tuning the network.
LTTS delivers the OEM's central automation platform to its operator customers end to end: install, upgrade, module execution, handover into care. Where the standard modules cannot express what an operator needs, the team writes new ones on the platform's development kit, and brings in data sources the shipped product never carried. Energy savings, data hygiene and network sharing between operators were all built that way.



Holding the code together.
Beneath the automation sits software running inside live operator networks. LTTS inherited it without documentation, so the design was recovered first by working backwards through the code; that is what made safe change possible at all. Then came two million lines under automated test, purpose-built simulators, and a laboratory of fifteen base stations and a core network, so operator-reported faults could be reproduced locally instead of debugged against production. Ten or more Release 10 features, and every patch release since, have shipped from here to schedule, against the OEM's own commitments to its operators.



Automating the automation.
Writing an operator's performance modules meant repeating the same code by hand, over and over. LTTS built a generator that takes a plain specification of what to measure and produces the working module code from it; the OEM's own delivery leadership recognized the result. Machine-learning models now predict performance from platform data alongside it, shipped through the OEM's pipelines rather than run as analyst spreadsheets.

THE FEAT · EIGHT YEARS INSIDE THE PRODUCT

Sustaining the software that runs inside operators' live networks, while customising the automation platform above it for 25+ of the OEM's own customers, and then building the tool that writes the automation.

BUSINESS OUTCOMES

All figures as reported by the engagement. Percentages are client-assessed, not audited.

Outcome	What changed	Why it is credible
Module development effort	About 70% saving	The generator converts a plain specification of what to measure into working module code. What it removes is boilerplate, which is why the saving is both large and repeatable.
Reach through the OEM	25+ operator customers served	A team of 30 or more engineers delivering install, upgrade and customisation work packages across Europe, India, North America and Latin America.
Engagement depth	5,000+ person-months, continuing	Eight years sustained, and extended from work-package execution into feature co-development with the OEM's product line.
Legacy code confidence	2 million line under test	Automated test coverage plus purpose-built simulators and stubs, on a codebase whose design documentation the team had to reconstruct first.
Defect throughput	150+ change requests; 500+ escalated tickets	A dedicated laboratory of 15 base stations and a core network let faults be reproduced locally instead of debugged against a production network.
Operating model	Zero-touch provisioning; proactive performance monitoring	Continuous-learning prediction models deployed through the OEM's own delivery pipelines, acting on platform data rather than reporting on it.

ENGINEERING INTELLIGENCE IN ACTION

1. Depth in one domain, held for eight years. A software vendor could have written modules. A staffing firm could have supplied engineers. Neither, alone, sustains live network software while customising the automation platform above it for 25+ operators.

2. Close to the standard, the product line and the operator at once. The team sits inside the OEM's own organisation, works with its R&D on new platform features, and delivers to the OEM's operator customers. Few partners hold all three positions.

3. We automated our own work, then handed over the gain. The code generator removed the repetitive part of building performance modules. That changes the unit economics for the OEM: the marginal cost of the next operator's custom module falls.

This collateral includes AI-enhanced imagery

L&T Technology Services

ENGINEERING THE CHANGE

Copyright © L&T Technology Services