

13 integrations, zero visibility — fixed as a program

A pan-European parking operator — multi-tenant platform for operations, claims and billing

THE STAKES

The platform ran on 13 external partner integrations (Azure Service Bus, Event Hubs) with no distributed tracing and no latency visibility. The operational reality: recurring incidents, hours-long investigations across a chain of services with no way to see where a request died, and heavy manual intervention. Every one of those hours is engineer time the business pays for twice — once in salary, once in delayed fixes.

WHAT WE DID

Ran the fix as a program — one accountable engineer, three months:

- **Discovery audit** of the production codebase — findings with file-and-line evidence, not opinions
- **Implementation playbook** — the exact library and pattern chosen per gap — and a tiered, execution-ready backlog; presented and defended to stakeholders
- **Built the tracing and metrics foundation:** OpenTelemetry across Django, DRF, Celery, PostgreSQL, Redis and outbound HTTP, through a hardened collector (secret scrubbing, sampling, ingest-cost guardrails), with a staged production rollout runbook — so an incident shows its own cause
- **Designed the failure containment** for all 13 partner integrations: circuit breakers, bulkhead queues, retry budgets with jitter, transactional outbox, explicit fail-open/fail-closed policies — one partner's outage stays one partner's outage
- Parallelized the test suite: **4,300+ tests in about a minute** — fast enough that engineers actually run it

OUTCOMES

- Tracing and metrics foundation delivered end to end; program adopted via stakeholder sign-off.
- Partner integrations moved from implicit trust to explicit, designed failure policy.
- Incident diagnosis moved from hours of manual archaeology to reading a trace.

WHY IT MATTERS

This engagement is the delivery-audit motion itself: evidence-based findings, a playbook, a backlog, then the implementation — run end to end by one accountable engineer on a live 13-integration production platform.

STACK

Python · Django/DRF · Celery · OpenTelemetry · New Relic · Azure Service Bus · Event Hubs · PostgreSQL · Redis · pytest-xdist

This is the engineering standard behind Alpine's fixed-price delivery audit: findings with file-and-line evidence, an implementation playbook, and an execution-ready backlog your team can run with. References on request.

Alpine LTD · contact@alpine-engineering.eu · alpine-engineering.eu · linkedin.com/company/138193958