Gatling

2026

WHITEPAPER

Modern performance testing for financial services

How financial services leaders can leverage continuous performance testing and turn it into a board-level priority

Performance testing has always mattered in financial services. What has changed is the cost of getting it wrong.

In the past, a single large-scale test at the end of a release cycle was considered due diligence. It produced a report, confirmed a threshold, and gave teams confidence to go live.

By the time issues appeared (often in production, often during peak load) the damage was already done.

Today, that model is no longer adequate. Financial systems release daily, scale dynamically, and operate under regulatory frameworks that require continuous, demonstrable resilience.

The pace of software delivery doesn't allow for end-of-cycle testing. And regulators, increasingly, don't either.

The organizations that are ahead of this shift have redefined performance testing not as a milestone but as an ongoing discipline — embedded in the way they build, ship, and govern their systems.

Versioning load tests → Automating them → Sharing results → Building a feedback loop

Each step reinforces the next, turning performance testing from a single team's task into a shared governance practice — one that satisfies regulators, protects revenue, and gives leadership the visibility they need.

This whitepaper outlines what that discipline looks like in financial services: how leading institutions are moving from static reports to continuous confidence, and what it takes to get there.

Version your load tests like code

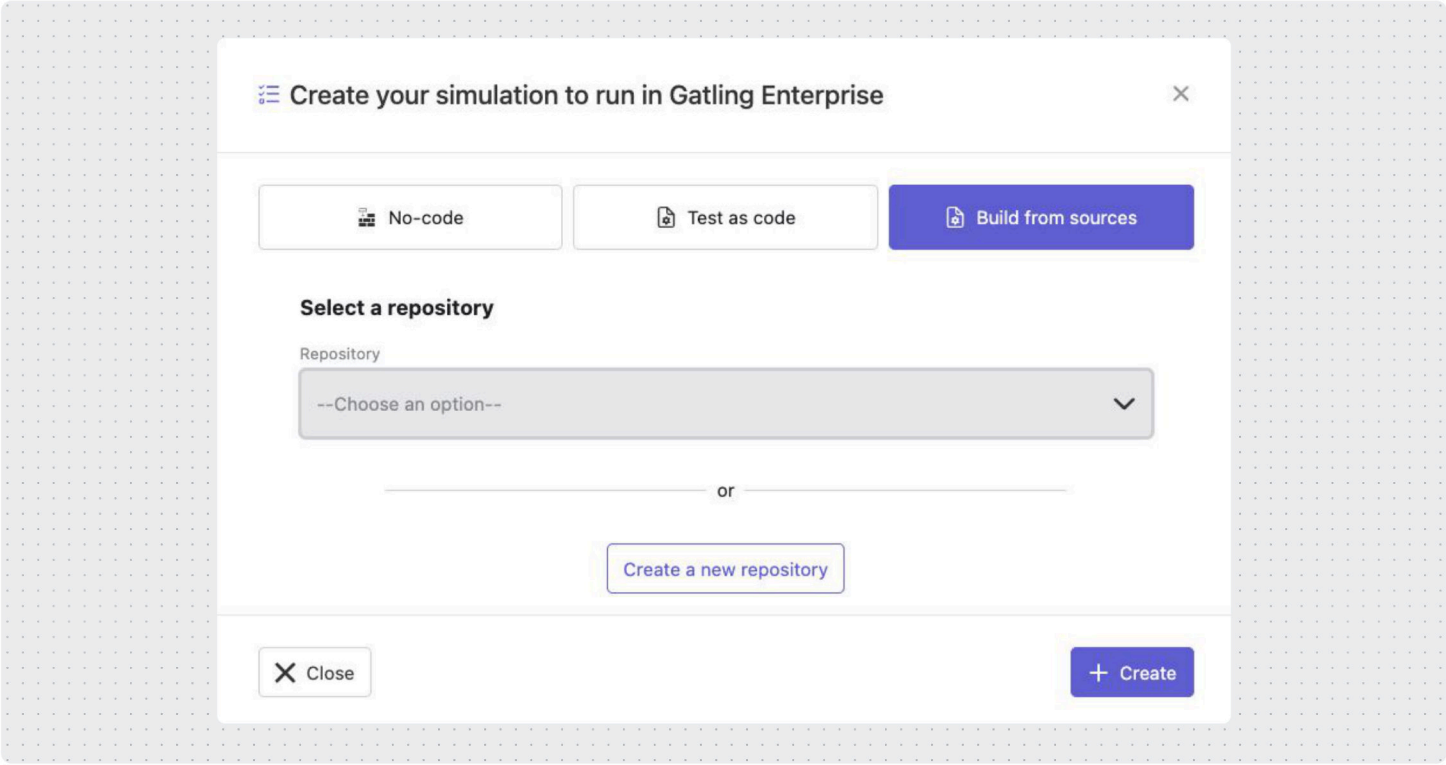
Performance testing starts with ownership and traceability.

If your load tests live in a shared folder that one engineer maintains, or in a tool that only a specialist knows how to run, they are already out of date. They reflect the system as it was, not as it is. And when a regulator or auditor asks how your systems were validated before the last release, the answer shouldn't be "we're not sure."

Treating load tests as code changes that dynamic entirely. When tests are versioned alongside application code, they evolve with the system. Every new feature, infrastructure change, or dependency update has a corresponding update in test coverage. There are no forgotten scenarios that only surface in production. And every test run is traceable to the exact release that triggered it — creating an auditable record of what was tested, under what conditions, and what the results showed.

For regulated financial institutions, that traceability is no longer just good practice. Under DORA and equivalent frameworks, organizations must be able to demonstrate that their systems have been tested under realistic conditions, repeatedly, over time. Versioned load tests are how that evidence is produced and preserved.

> **Pro tip for leadership:** Ask your engineering team where your load test scripts live and when they were last updated. If the answer involves a shared drive, a single specialist, or a tool disconnected from your codebase, your performance validation has a governance gap — and so does your regulatory evidence trail.



With build from sources you can create your own simulaton by connecting your own repository

Control access, enforce limits, centralize oversight

Large financial institutions don't run performance testing from a single team.

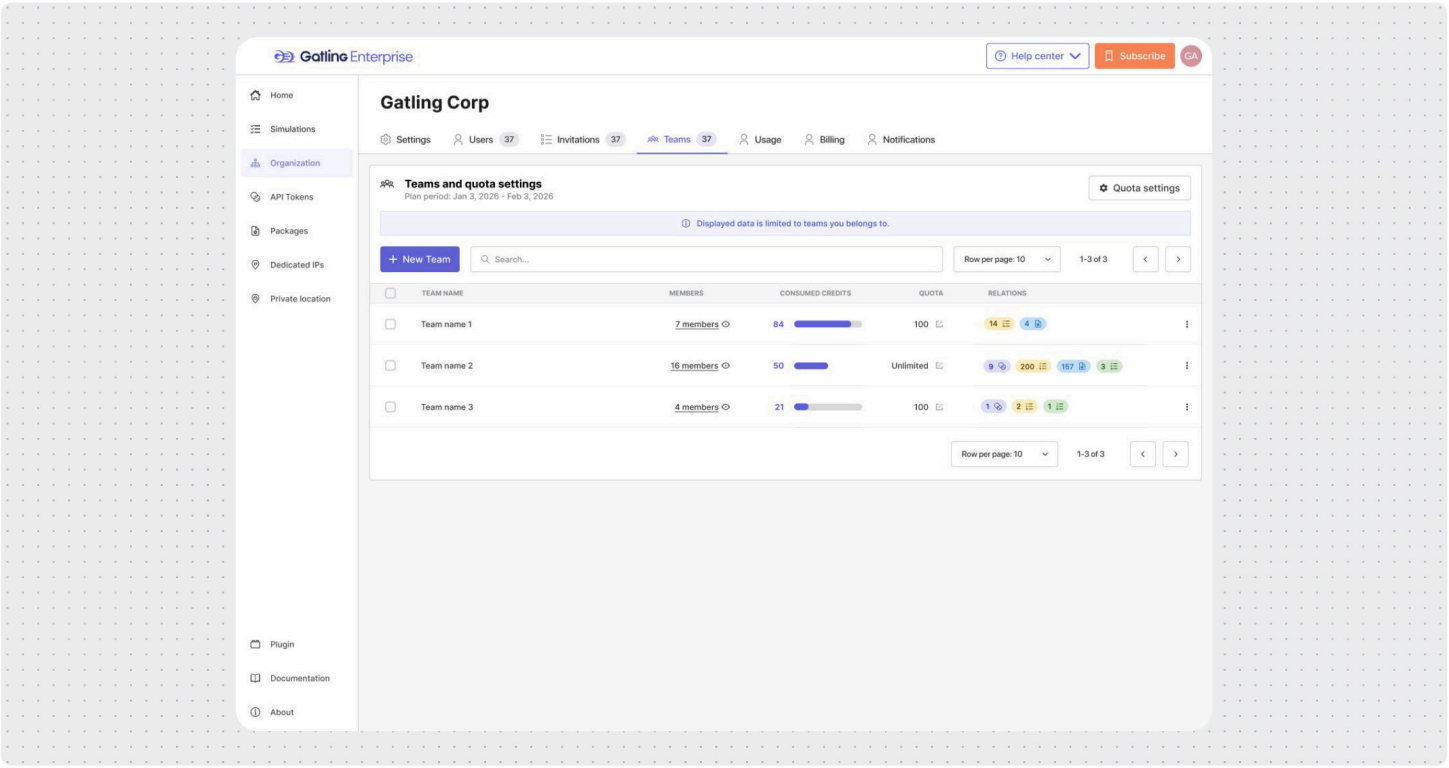
Payment engineering, core banking, digital channels, and platform teams all need to test, often simultaneously, often against shared infrastructure. Without governance, that creates conflicts, cost overruns, and blind spots.

Gatling Enterprise Edition gives organizations the controls to manage this at scale. Role-based access control ensures teams can only see and run what they're responsible for.

Usage quotas prevent any one squad from consuming shared testing resources at the expense of others. And centralized reporting gives platform and engineering leadership a unified view across all teams, without requiring them to aggregate results manually.

For regulated institutions, this governance layer also matters at audit time. When every test run is tied to a specific team, role, and set of permissions, the question of who tested what — and under what authority — has a clear answer.

> **Pro tip for leadership:** Ask whether performance testing in your organization has named owners, defined quotas, and a documented audit trail, or whether it runs on goodwill and informal coordination. The difference between those two things is the difference between governance and hope.



From a single dashboard, define teams, set usage quotas, and control access across every squad sharing your testing infrastructure.

Stay compliant with secrets management

Financial services load testing environments handle sensitive material by design.

Tests authenticate against real or near-real systems using API keys, client certificates, OAuth tokens, and encryption credentials.

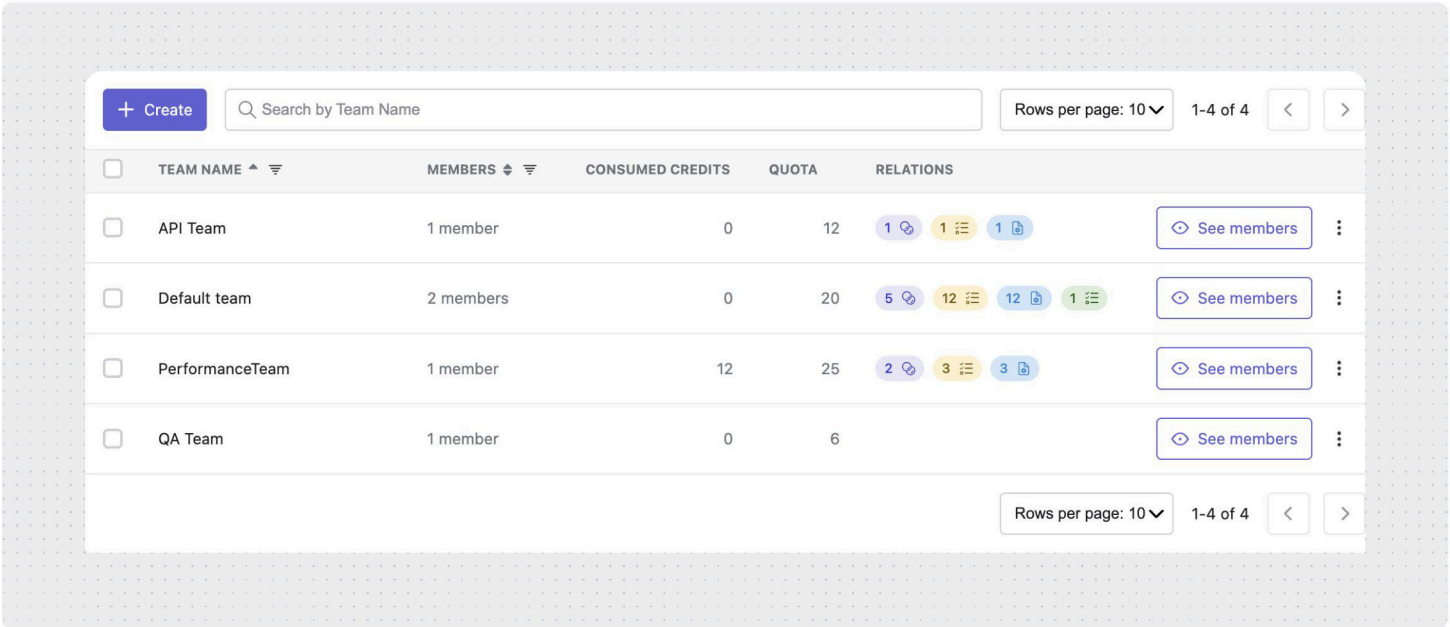
If that material isn't managed securely, the testing infrastructure itself becomes an attack surface.

Gatling Enterprise Edition integrates with enterprise secret managers to ensure sensitive credentials are never hardcoded in test scripts or stored in unsecured repositories.

API keys, certificates, and encryption material are managed centrally, rotated without requiring test code changes, and access-controlled at the same level as production credentials.

For institutions with strict information security requirements (which in financial services means most of them) this isn't a convenience feature. It's more of a prerequisite for running load tests against systems that handle customer data and financial transactions.

> **Pro tip for leadership:** Ask your security team whether they've reviewed how credentials are managed in your load testing environment. If test scripts contain hardcoded keys or if testing credentials aren't subject to the same rotation and access policies as production credentials, you have an exposure that sits outside your standard security perimeter.



Each team gets its own quota, members, and resource allocation, so API, QA, and performance teams can all test independently without stepping on each other.

Define what good looks like with SLOs

Automating tests is only half the equation. The other half is knowing, precisely, whether the results were acceptable.

Service Level Objectives solve this. An SLO is a performance contract defined directly in Gatling Enterprise Edition, no code changes required. You set a metric (a response time percentile or error ratio), a threshold, and the platform evaluates it continuously throughout every run.

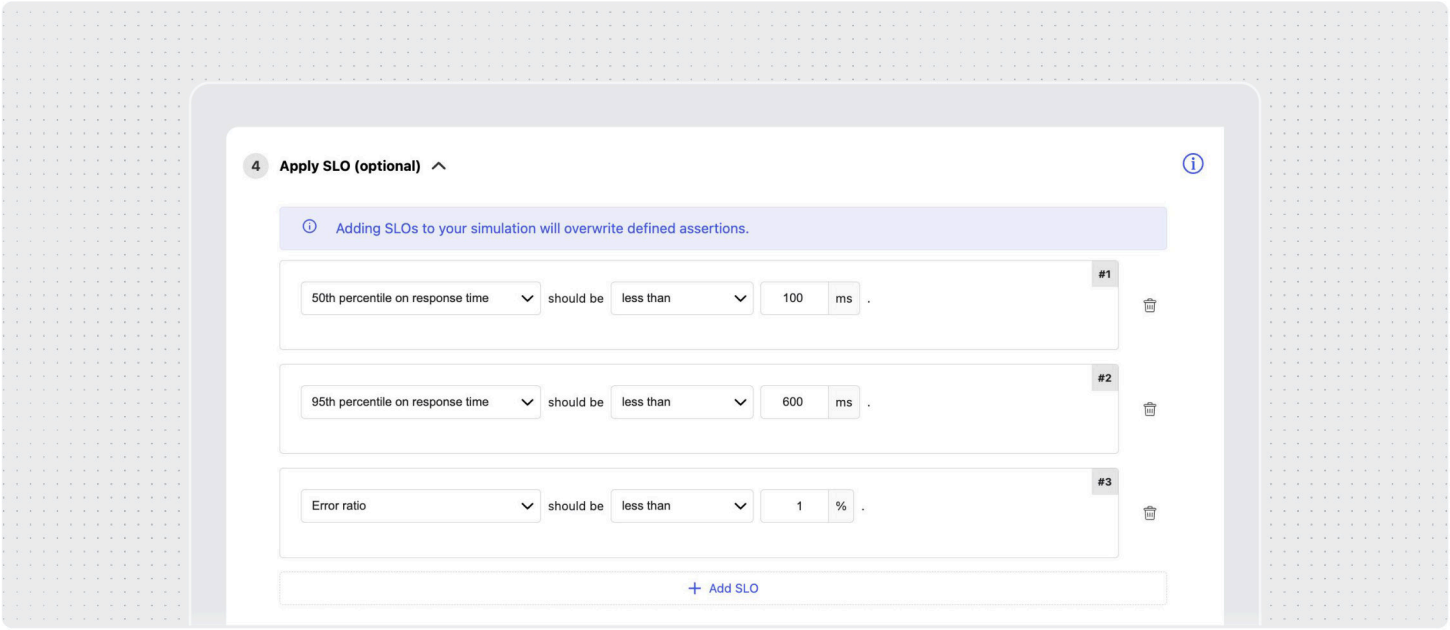
The output isn't only a pass/fail judgment; It's a compliance score: the exact percentage of time your system held to its defined standard. Green means it held 99% of the time or more. Orange means it was close but worth investigating. Red means the SLO was breached.

For financial services leaders, this distinction matters. A compliance score is auditable evidence because it’s an objective, tied to a specific run, and comparable across releases.

You can stack multiple SLOs on a single test: a response time target, an error ratio cap, and a stricter threshold for your most critical payment flows, each evaluated independently.

And because SLOs are configured in the platform rather than in test code, engineering managers and platform teams can define and adjust standards without depending on the engineer who wrote the simulation.

> **Pro tip for leadership:** Ask your team how they currently determine whether a load test "passed." If the answer involves someone reviewing a chart and making a judgment call, your performance standards aren't standards — they're opinions. SLOs replace that ambiguity with a compliance score that means the same thing to every stakeholder, every run.



Define response time and error ratio targets directly in Gatling Enterprise Edition, no code changes required.

Build a continuous feedback loop

Modern performance testing is a signal that accumulates value over time.

A single test run tells you how your system performed on a given day, against a given load profile, at a given point in its release history. That's useful. But the real intelligence comes from what happens when you track that signal across dozens or hundreds of runs or when you can see not just whether your system passed a threshold today, but how its performance has evolved over the last quarter.

This longitudinal view is what separates organizations that are genuinely building resilience from those that are performing compliance theater. It reveals gradual degradation that no single test would catch.

It quantifies the performance impact of architectural decisions. And it gives leadership a factual basis for conversations about capacity, risk, and investment rather than relying on engineering judgment after the fact.

For financial services organizations operating under DORA, the UK resilience framework, or equivalent regulations, this history is increasingly the substance of what regulators want to see.

Not a single test report from last quarter. A documented pattern of continuous validation, with trend data showing how performance has held (or where it has drifted) across releases over time.

> **Pro tip for leadership:** Ask your team what your payment flow's performance looked like six months ago compared to today. If that question requires someone to go digging through old reports (or can't be answered at all) your organization is producing test results but not building performance intelligence. The difference matters, both for operations and for regulatory evidence.

Your recent runs are displayed in 3 bar graphs, with up to 10 runs in total.
Only successfully executed runs (noted by green check marks in the left-side menu) are included in the graphs.

From reactive firefighting to predictive performance engineering

The goal isn't just better testing. It's better decisions, faster.

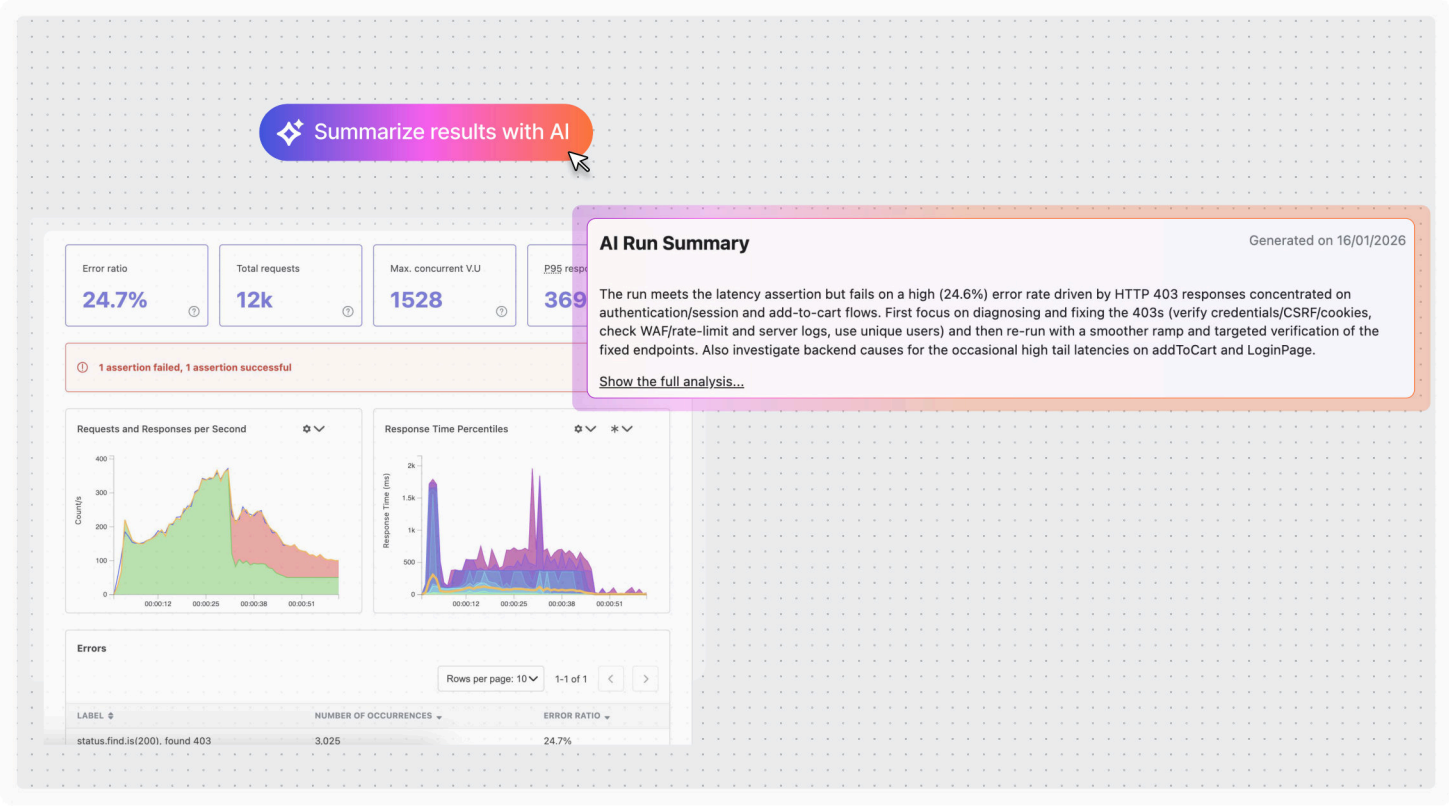
AI is changing how performance testing works and how quickly results turn into action. Gatling uses AI in three places that matter for financial services teams.

First, test creation: engineers can generate working simulations in minutes, update scripts when APIs change, and migrate legacy tests automatically, removing the specialist bottleneck that slows performance coverage down.

Second, result interpretation: instead of hours spent translating graphs into conclusions, AI Insights surfaces what changed, what looks abnormal, and where to focus so the right people can decide faster without needing a performance expert in the room.

Third, as financial institutions build AI-powered products of their own, Gatling enables load testing of LLM-based applications under real conditions like streaming responses, long-running requests, concurrency behavior before cost and reliability surprises hit production.

> **Pro tip for leadership:** One of the most common performance bottlenecks in financial institutions is organizational. Testing depends on a small group of specialists, and coverage is limited by their capacity. AI-assisted test creation and automated result interpretation change that equation, making performance testing something every engineering team can contribute to and every leader can understand.



From raw results to clear next steps: Gatling's AI summarizes every run in seconds, highlighting what failed, why, and where to focus.

What financial services companies actually test with Gatling?

EPI Company: building Europe's payment infrastructure with performance confidence

EPI Company, backed by 16 major European banks, is building Wero, a European payment scheme operating across five countries.

From early development, EPI made performance testing a core engineering practice. Using Gatling Enterprise Edition, their teams test security-heavy flows including API authentication (JWT, mTLS, HMAC), client certificate validation, mobile performance under real-world latency, and high-volume transactional traffic.

15+ engineers across multiple teams collaborate on performance tests. Private locations on AWS keep test traffic inside their infrastructure. SSO and Datadog integrations bring performance testing in line with their security and observability stack.

✓ **THE RESULT?** Performance testing is embedded in their release process, not a bottleneck in it.

Nickel: serving millions of accounts through traffic spikes

Nickel, a BNP Paribas subsidiary, serves 4.5 million accounts through 8,200 retail partners across France. During social benefit disbursements, traffic spikes 20 to 30 times baseline for one to two days. Every transaction must still complete in under three seconds.

Performance testing at Nickel isn't optional. It's mandatory before every release as part of their Change Advisory Board review. Each application must demonstrate no degradation compared to the previous version.

Using Gatling Enterprise Edition's private locations, load generators run inside Nickel's own on-premises environment, a non-negotiable for a banking application.

✓ **THE RESULT?** They're now expanding to automated testing across 20 critical applications and planning denial-of-service simulations.

Both followed the same journey: start with Gatling Community Edition, hit an operational ceiling around visibility, governance, and infrastructure control, then move to Enterprise to scale across teams. Both deploy load generators inside their own perimeter, integrate testing into release governance, and treat performance results as decision inputs, not just dashboards to check.

Choosing the right load testing platform for financial services

Embedding continuous performance testing into financial services delivery requires more than the right intent; on the contrary, it requires a platform built for the constraints and expectations of regulated, private-infrastructure environments.

Gatling Enterprise Edition enables financial companies to:

Analyze smarter and act faster

Real-time dashboards, trend comparisons across releases, and SLO compliance scores that give leadership the visibility they need, without requiring them to interpret raw engineering data.

Create tests your way

Build tests via code, low-code, or no-code. Script in JS/TS or Java, import existing collections, or design visually. Even people who've never written a load test can use Gatling

Deploy load tests from your coding agent

Gatling's MCP Server & Skills lets engineering teams trigger and manage load tests directly from their AI coding environment, embedding performance validation into the workflow.

Collaborate and share results easily

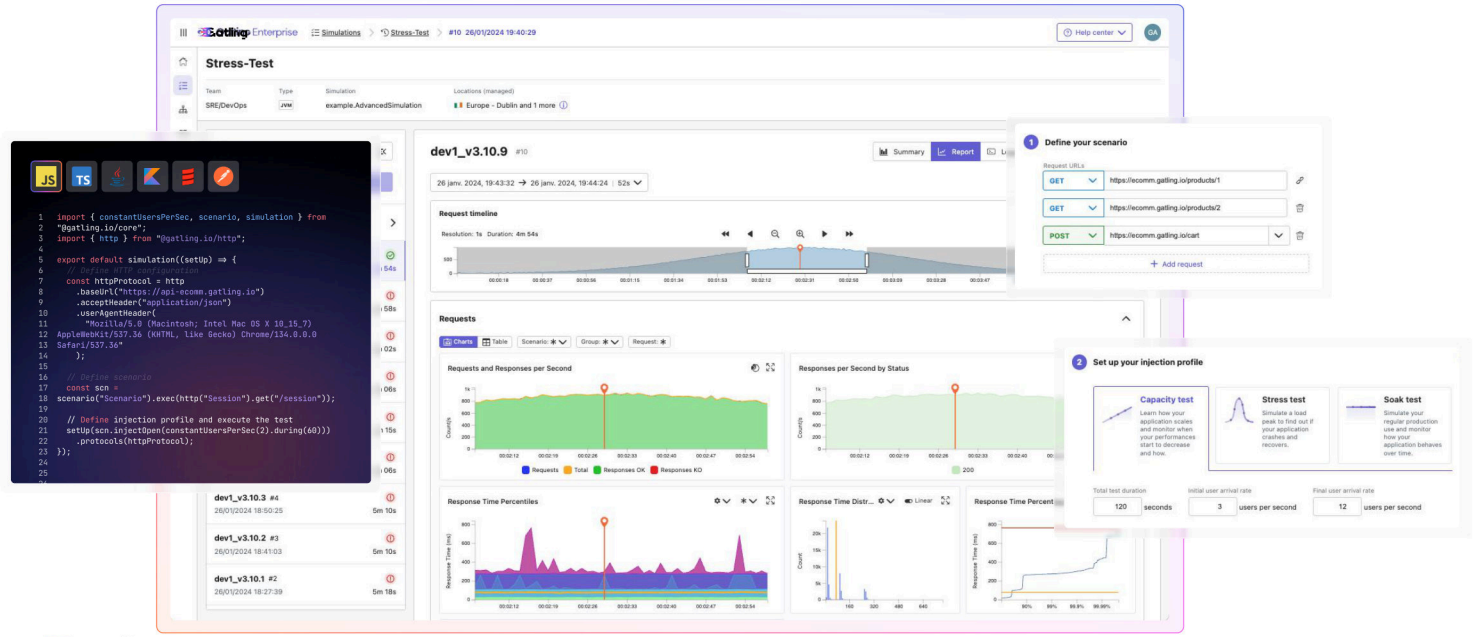
RBAC, SSO, quotas, integrations with Slack, Teams, and Jira mean performance signals surface where teams already work and shared reports ensure results reach the right people.

Deploy load generators anywhere

Financial services infrastructure requires testing that reflects real production conditions, not a public proxy. Run tests from inside your network or from Gatling managed regions.

Track performance compliance on every run

Define SLOs against response time and error rate thresholds. Gatling evaluates them continuously across each test run and produces a compliance score: objective, repeatable, and auditable.



Gatling

Gatling Enterprise Edition is the platform purpose-built to deliver Continuous Performance Intelligence: transforming load testing from a technical activity into an organizational capability that business leaders can govern, product teams can engage with, and engineering teams can scale.

With its powerful open-source and enterprise platforms, Gatling empowers teams to test APIs, microservices, and web apps in real-world conditions.

Trusted by thousands of companies worldwide, Gatling is the performance backbone for development, QA, and DevOps teams building the next generation of software.

Whether you're scaling APIs, migrating to the cloud, or handling flash traffic spikes, Gatling helps you deliver fast, reliable performance.

Ready to implement continous performance intelligence?

See how AI-assisted performance testing can take you to the next level

Talk to an expert ›