# Measuring What Matters: Experience Report on Continuous Improvement of Large-Scale Government APIs

Estêvão Bissoli Saleme
Federal Data Processing Service (Serpro)
Belo Horizonte-MG, Brazil
estevao.saleme@serpro.gov.br

## Abstract

APIs are a core component of modern, large-scale digital public services. At Serpro, the Brazilian Federal Data Processing Service, APIs serve as the primary mechanism for integrating systems and delivering data to both private-sector customers and federal government agencies. These APIs support more than 15 million daily interactions across domains such as civil registries, taxation, foreign trade, digital identity, etc., through a portfolio of over 115 APIs exposed to external public users. Operating APIs at this scale comes with engineering and operational challenges, including performance degradation, service unavailability, lack of standardization, infrastructure constraints, and fragmented observability. This extended abstract reports on an industry experience with GUAPI, a gateway-agnostic API governance platform designed to support unified observability and governance in production environments. GUAPI uses an event-driven architecture with near-real-time processing to consolidate metrics related to performance, availability, and quality into a data-driven view of API behavior. The reported experience suggests that unified indicators can improve operational transparency, support continuous improvement, and enable more proactive decision-making. It thus offers practical lessons learned and insights that may be relevant to organizations facing similar challenges when engineering, governing, and evolving large-scale API platforms.

## Keywords

API Governance, Industry Experience, Software Operations, Event Streaming, Digital Government

## 1 Motivation, Context, and Challenges

Application Programming Interfaces (APIs) play a central role for integrating heterogeneous software systems and enabling digital services [6, 8]. In digital government, their role is even broader, as they support data exchange among public agencies, private organizations, and citizens [11]. At Serpro, the Brazilian Federal Data Processing Service, APIs provide access to critical domains such as civil registries, taxation, vehicles, biometrics, foreign trade, digital identity, among others. This ecosystem supports more than 15 million daily interactions through more than 115 APIs for external public users and contributes to the modernization and expansion of digital public services.

Before GUAPI (*Gestão Unificada de APIs*, or Unified API Management), an API tool developed at Serpro to support governance, the organization faced recurring operational challenges in managing APIs in production. Incidents often led to large "war rooms" involving development, operations, infrastructure, architecture, support, and management teams. These situations were frequently caused or intensified by the lack of a unified view of API governance. Key indicators related to performance, reliability, availability, and errors were either unavailable, inconsistently measured, or distributed across multiple tools and teams, a common issue in large organizations [1, 2, 6]. As a result, there was no objective or standardized way to assess whether an API met operational quality expectations.

This fragmentation made incident diagnosis costly and time-consuming. When an API presented latency, instability, or unavailability, teams had to manually correlate data from API gateways, backend services, infrastructure monitoring tools, logs, and support reports. Root-cause analysis thus depended heavily on human coordination and prior knowledge of the ecosystem. This reactive model also limited continuous improvement, since recurring problems were not consistently translated into standardized indicators, measurable goals, or feedback loops for development and operations teams.

According to Pillai et al. [7] from Gartner, "APIs provide the foundation for digital transformation, modernization, and digital business ecosystems, but they are difficult to manage and govern." Indeed, the diversity of Serpro's API ecosystem created additional scalability and governance challenges. APIs were operated by different teams, supported by different systems, and exposed through heterogeneous infrastructures. These challenges motivated the development of GUAPI as a scalable, data-driven solution for continuous API quality management and operational governance. Rather than treating API monitoring as a set of isolated dashboards, GUAPI provides a unified view of API quality. Its event-driven architecture supports near-real-time processing of API access data [9] and consolidates indicators related to performance, throughput, availability, error rates, and reliability. This approach reduces fragmentation, improves operational transparency, and accelerates incident diagnosis.

## 2 Relevance, Impact, and Benefits for the Software Industry

The experience of GUAPI addresses a common challenge in large-scale API ecosystems: how to govern, measure, and continuously improve API quality in production [4]. Modern software platforms increasingly rely on APIs to integrate internal systems, external partners, digital products, and distributed services. As these ecosystems grow, API governance becomes not only a technical activity but also an organizational capability that affects reliability, maintainability, scalability, transparency, and user experience.

GUAPI is particularly relevant in environments where API ownership and operation are distributed across multiple teams, services, and infrastructure layers. In such contexts, fragmented observability can hinder communication, delay incident diagnosis, and increase coordination effort during production incidents. GUAPI's shared view improves transparency among teams, reduces ambiguity during incident analysis, and supports the prioritization of improvements based on measurable evidence rather than assumptions.

GUAPI establishes a continuous feedback loop in which production data is transformed into indicators that support decision-making at different levels:

- For developers and operations teams, comparable metrics such as latency, error rate, throughput, availability, mean time to recovery, and mean time between failures enable the identification of performance bottlenecks, recurring failures, abnormal behavior, and opportunities for refactoring or infrastructure improvements.
- For managers, aggregated indicators provide a consolidated view of service quality, allowing them to monitor API evolution, compare operational conditions across services, and prioritize improvement initiatives based on evidence.

Acting as a shared decision-support layer, GUAPI makes API governance more proactive and evidence-based while enabling the systematic identification of standardization issues and architectural improvement opportunities. Combined with its gateway-agnostic design, which supports heterogeneous environments shaped by legacy systems, multiple technology stacks, or gradual modernization, this approach is readily transferable to other organizations managing large and distributed API portfolios.

To position GUAPI within the API management landscape, it is instructive to contrast its approach with widely adopted platforms such as Google Apigee, MuleSoft Anypoint Platform, Kong Gateway, Amazon API Gateway, Microsoft Azure API Management, and WSO2 API Manager. These solutions offer comprehensive support for the API lifecycle, encompassing design, deployment, security, traffic governance, analytics, and monetisation. However, their architectures are often tightly coupled to proprietary gateway infrastructures [12]. While this integration can simplify platform operations, it may also limit flexibility in heterogeneous environments and contribute to vendor lock-in, a particularly relevant concern for public organisations operating across multiple technologies and infrastructure providers.

More recent platforms, such as Sensedia and Gravitee, have introduced multi-gateway governance layers that consolidate API catalogs and metrics across providers. GUAPI differs in three main respects: (i) it is gateway-agnostic by design, consuming logs from heterogeneous gateways without requiring replacement or migration; (ii) it focuses specifically on operational quality governance, providing a unified analytical view of production indicators rather than managing the entire API lifecycle; and (iii) it adopts an event-driven architecture based on Apache Kafka, Kafka Connect, and Kafka Streams, enabling near-real-time processing instead of batch-oriented analytics.

GUAPI's architecture continuously collects, normalizes, enriches, and processes API access events from heterogeneous gateways, transforming production data into governance indicators without intrusive changes to existing API implementations. As a result, GUAPI complements rather than replaces API management platforms by addressing a common gap in large-scale, multi-gateway environments: the lack of unified, real-time, and comparable quality indicators across APIs in heterogeneous environments. While this report focuses on the organizational experience and the continuous-improvement outcomes enabled by GUAPI, the detailed architectural design and implementation are described in the companion Tools Track paper by Saleme et al. [9].

Therefore, even though GUAPI was developed in a public-sector environment, the challenges it addresses are equally relevant to organizations that manage large and heterogeneous API ecosystems, regardless of domain. From both research and industrial perspectives, the reported experience provides empirical evidence on how gateway-agnostic governance, event-driven architectures, and unified operational metrics can support continuous improvement at scale. These lessons are particularly timely as APIs become the primary interface for AI-enabled systems, with Large Language Models (LLMs) increasingly exposed and consumed through API ecosystems.

## 3 Evidence and Success Indicators

Since August 2025, GUAPI has operated in production, processing more than 15 million events per day. This evaluation analyzes daily API event data collected from 4 August 2025 to 30 April 2026 to assess changes in operational behavior. The monitored indicators include APDEX, uptime score, downtime, unavailability events, response time, error rate, Mean Time to Recovery (MTTR), and Mean Time Between Failures (MTBF). Atypical days were kept for separate incident analysis. Monthly request-event metrics use medians and IQR (Interquartile Range) bands, while downtime and unavailability events use monthly totals. Uptime score, daily MTTR, and daily MTBF also use medians and IQR bands. Linear regression indicates only descriptive trends.

These aforementioned indicators are commonly used in API management, microservice monitoring, production readiness, observability, and site reliability engineering to assess service quality and incident response [3, 5, 6, 8, 10, 13].

Figure 1 shows monthly trends for request-event quality metrics. APDEX, scaled from 0 to 100, increased from 93.74 in August 2025 to 95.80 in April 2026. Median request duration decreased from 261.92 ms to 174.25 ms, indicating faster request completion. The median total error rate also decreased from 12.98% to 8.65%. Although November 2025 showed temporary degradation across these indicators, the broader trend suggests improved responsiveness and fewer errors under typical operating conditions.

Figure 2 presents the availability indicators. Downtime and unavailability events were analyzed as monthly totals. From August 2025 to April 2026, downtime decreased from 1,486 to 1,057 minutes, and unavailability events decreased from 813 to 679. The lowest values were observed in March 2026, with 580 downtime minutes and 349 unavailability events, followed by an increase in April 2026. Median uptime score increased from 99.883% to 99.961%, median daily MTTR decreased from 0.57 to 0.33 minutes, and median daily MTBF increased from 1,266.06 to 1,366.28 minutes. These results indicate improved availability indicators during the analyzed period, despite the deterioration observed in April 2026.

The analysis presented in this study is descriptive and should not be interpreted as causal evidence that GUAPI alone produced the

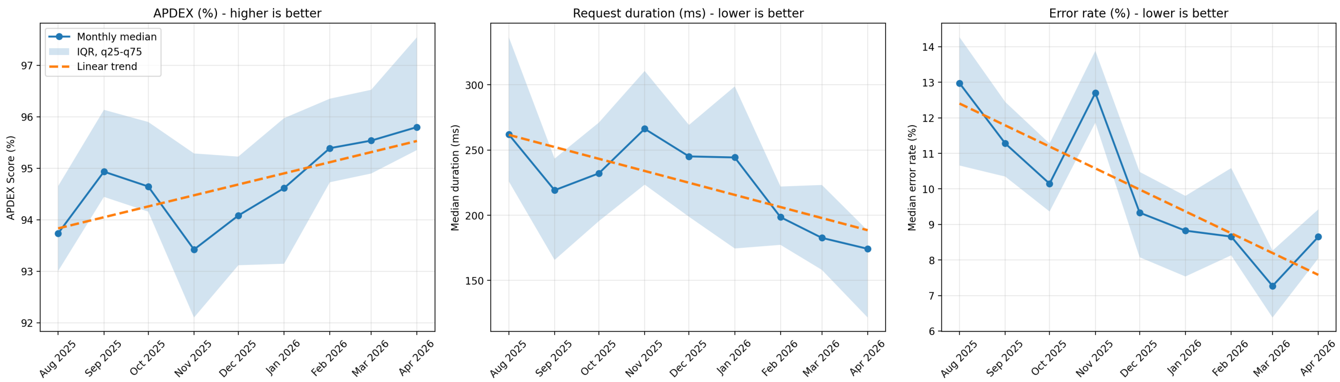


**Figure 1: Monthly trends show improving request-event performance, with APDEX increasing while average duration and error rate decline over time.**

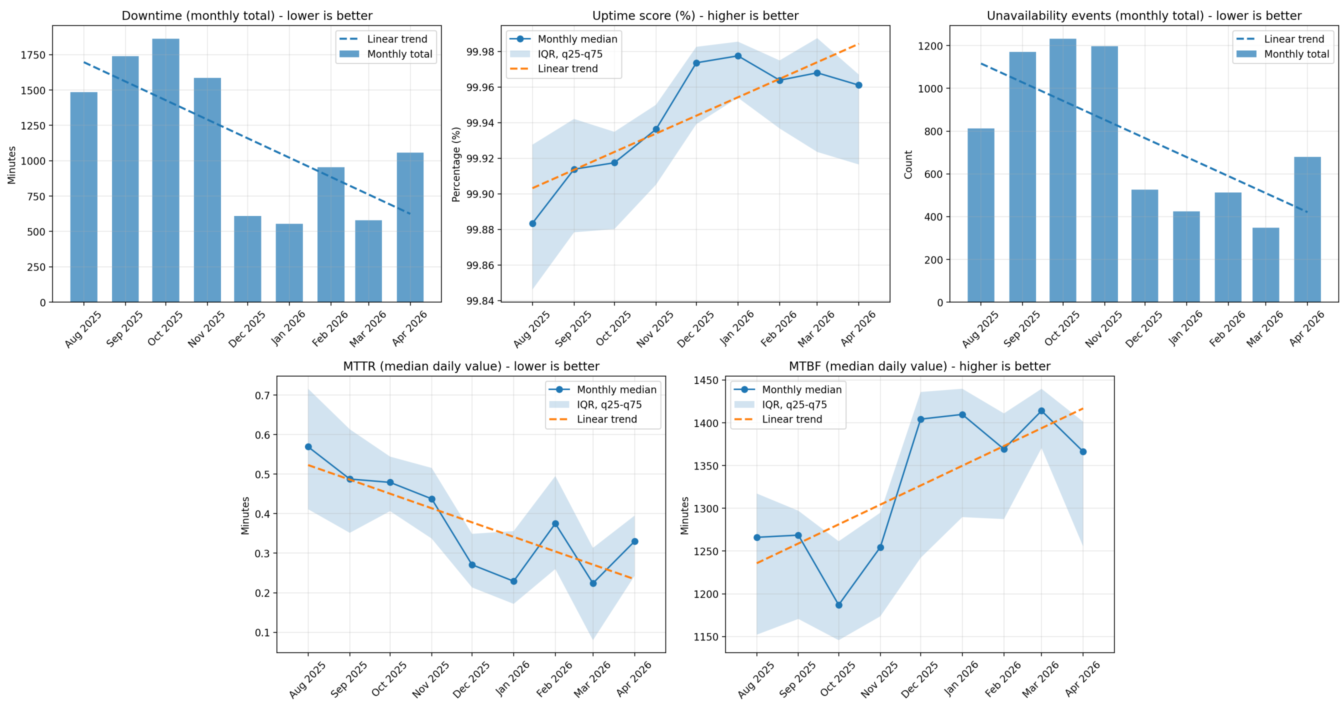


**Figure 2: Availability metrics trend positively over time, with decreasing downtime, unavailability events, and MTTR, alongside slight increases in uptime and MTBF.**

observed improvements, since operational, organizational, infrastructure, and workload factors may also have influenced the results. Rather than acting as an isolated intervention, GUAPI should be understood as an enabling mechanism for operational learning and supporting strategic decisions. After consolidating indicators that were previously fragmented across servers, tools, and logs, it improved visibility into production behavior, reduced ambiguity during incident analysis, and established a common factual basis for discussions among teams. This shared view enabled more systematic monitoring of service quality, facilitated the identification of deviations and improvement opportunities, and supported a shift from reactive troubleshooting toward evidence-based API governance.

At the same time, the experience exposed important trade-offs. Supporting heterogeneous gateways required effort to normalize event formats and align operational data with API metadata, while the effectiveness of log-based governance remained dependent on the completeness, consistency, and semantic richness of the collected events.

## 4 Lessons Learned

GUAPI's experience at Serpro highlights six lessons for organizations operating large-scale, heterogeneous API ecosystems.

**From monitoring to governance**: monitoring alone is insufficient for sustained API quality. Effective governance requires integrated, historical, and comparable indicators that support operational learning, evidence-based prioritization, and continuous improvement, not just alerting.

**Event-streaming as an architectural enabler**: event-streaming architectures are well suited to high-volume production environments. GUAPI provides actionable indicators without batch delays, enabling faster feedback loops and more timely decision-making processing API access events in near-real-time. Events are durably retained in Kafka, allowing recovery from failures and replay historical data when processing resumes. Thus, temporary interruptions in downstream components, such as database ingestion, do not lead to data loss, since accumulated events are processed retrospectively once the affected components become available.

**Gateway-agnostic design for heterogeneous environments**: it reduces vendor lock-in and preserves architectural flexibility. Because GUAPI consumes logs from existing gateways rather than replacing them, it integrates into diverse technology stacks without requiring infrastructure migration, a huge advantage in public-sector environments with long procurement cycles.

**Frictionless adoption through transparent integration**: GUAPI required no adoption effort from development or operations teams. API specifications already published to gateways in OpenAPI format are imported directly, and Kafka event data is automatically associated with the corresponding APIs and paths. No new interfaces, instrumentation, or workflow changes were needed, which reduced organizational resistance and accelerated rollout.

**Objective metrics as a shared language**: standardized, objective metrics reduce ambiguity during incidents, improve cross-team communication, and shift prioritization from assumptions to data. A shared indicator set enables developers, operators, and architects to coordinate faster and diagnose problems more effectively, while providing managers with objective evidence to support strategic planning, resource allocation, and long-term decision-making.

**Toward multi-perspective quality evaluation**: API quality should be evaluated from multiple perspectives, not solely from infrastructure or backend reliability. GUAPI is being extended with indicators covering consumer experience and specification quality - First-Success Bounce Rate (FSBR), Time to First Call (TTFC), Anomaly Detection Rate (ADR), and Static Analysis Score (SAS). These metrics will incorporate user-experience, adoption, behavioral, and design-quality dimensions into API governance.

These lessons are relevant to both industry and research. Rather than prescribing guidelines or best practices, this experience report provides practical insights that practitioners and researchers alike can interpret and adapt to their own organizational and technological contexts. In line with the central theme of measuring what matters, GUAPI illustrates for industry how scalable, non-intrusive, and vendor-independent API governance can improve operational visibility, support evidence-based prioritization, and facilitate coordination through shared quality indicators. More fundamentally, systematic measurement provides an objective basis for distinguishing desirable from undesirable conditions, enabling organizations to identify strengths, detect quality deficiencies, and make informed decisions grounded in empirical evidence rather than intuition. For research, GUAPI offers a real-world case of continuous API quality evaluation in a large-scale production environment, providing empirical evidence to inform future studies on data-driven, multi-perspective API governance and the evolution of quality assessment practices in API ecosystems.

## Artifact Availability

The study did not produce software, datasets, models, or other digital assets intended for public distribution.

## References


[1] Mak Ahmad, J. J. Geewax, Andrew Macvean, David R. Karger, and Kwan-Liu Ma. 2024. API Governance at Scale. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '24)*. ACM, Lisbon, Portugal, 430–440. doi:10.1145/3639477.3639713

[2] David Bermbach and Erik Wittern. 2020. Benchmarking Web API Quality — Revisited. *Journal of Web Engineering* 19, 5–6 (2020), 603–646.

[3] Betsy Beyer et al. 2016. *Site Reliability Engineering: How Google Runs Production Systems.* O'Reilly Media.

[4] Eder dos Santos and Sandra Casas. 2024. Unveiling Quality in API Management: A Systematic Mapping Study. In *2024 L Latin American Computer Conference (CLEI)*. IEEE, 1–10. doi:10.1109/clei64178.2024.10700447

[5] Susan J. Fowler. 2017. *Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization.* O'Reilly Media.

[6] Mehdi Medjaoui, Erik Wilde, Ronnie Mitra, and Mike Amundsen. 2021. *Continuous API Management: Making the Right Decisions in an Evolving Landscape.* O'Reilly Media, Sebastopol, CA, USA.

[7] Sanjiv Pillai et al. 2024. *API Management Market Guide.* Technical Report. Gartner. Industry Report.

[8] Leonard Richardson, Mike Amundsen, and Sam Ruby. 2019. *RESTful Web APIs.* O'Reilly Media, Sebastopol, CA, USA.

[9] Estêvão Bissoli Saleme, Hellyan Alves de Oliveira, João Vitor da Silva Gomes, and Flávio Gomes da Silva Lisboa. 2026. GUAPI: A Gateway-Agnostic, Unified, Data-Driven Platform for API Governance Based on Event Streaming. In *Proceedings of the Brazilian Symposium on Software Engineering (SBES 2026) - Tools Track.* Brazilian Computer Society (SBC), São Paulo, SP, Brazil.

[10] Cindy Sridharan. 2018. *Distributed Systems Observability: A Guide to Building Robust Systems.* O'Reilly Media.

[11] United Nations Department of Economic and Social Affairs. 2024. *United Nations E-Government Survey 2024: Accelerating Digital Transformation for Sustainable Development.* Technical Report. United Nations, New York. https://publicadministration.un.org/egovkb/en-us/Reports/UN-E-Government-Survey-2024 Accessed: 2026-05-06.

[12] Padmanabhan Venkiteela. 2025. Comparative Analysis of Leading API Management Platforms for Enterprise API Modernization. *International Journal of Computer Applications* 187, 54 (2025).

[13] Muhammad Waseem et al. 2021. Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective. *Journal of Systems and Software* 182 (2021).