12 Types of Integration Performance Metrics | MuleSoft Blog

12 Types of Integration Performance Metrics for Different Teams

Reading Time: 12 minutes

We have previously discussed how to choose the right API metrics to track integration performance, covering what metrics are, how they measure integration behaviors, and the best practices for choosing the right metrics.

Using that knowledge, and through engagements with our MuleSoft customers, we've collected KPIs that can be useful in tracking the performance of API programs and integrations. We will explore some most commonly tracked metrics, how they impact different personas, and what tools are used to help capture them.

Different personas track different metrics

Each team in an organization will have different needs and reasons to track different sets of metrics when dealing with APIs and integrations. For example, the set of metrics that a developer would like to track may be entirely different from that of a tester. Similarly, the business leaders will want to track a different set of metrics than an integration architect.

The type of metrics will depend on the stage of the integration or API in its product life cycle. For a newly developed API, we may be interested in its design and functionality, or the cost or development effort, whereas for a matured integration, we may be interested in tracking its stability, reliability or performance.

Let's look at how different personas will want to track different metrics in an API factory.

Development team

- **Application or platform monitoring:** Here the focus will be on tracking the performance of the APIs such as throughput and response times, success and error rates, defects and incidents, etc.
- **Application productivity:** Here the developers may be interested in development effort, API consumption rate, reuse rate, etc.
- **Application quality:** This can include code coverage and policy compliance metrics.

Operations team

- **Platform availability and scalability:** Operations teams may be interested in tracking the uptime of the systems and applications, resource usage of the systems such as memory, CPU utilization, etc.
- **Application or platform monitoring:** Teams will be interested in tracking the performance of the APIs such as throughput and response times, success and error rates, defects and incidents, etc.

IT management team

- **Platform performance, usage, and cost:** IT managers will be interested in tracing the performance stats such as throughput and response times. They will also want to track the application density (how many applications are deployed per CPU?), and the cost of their platform
- **System quality:** They will want to measure the code coverage, track the error and success rates, testing and refactoring effort, defects, incidents and system downtime
- **Consumption rate:** They may want to track the reuse of assets and consumption rate by channel, line of business teams etc.
- **Productivity:** Focus will be on tracking the development and QA efforts

Business leaders

- **Platform usage:** The business leaders or the sponsors will be interested in tracking metrics such as the total cost of ownership (TCO), return on investment (ROI), productivity gains and customer satisfaction
- **Consumption:** They will be interested in tracking the reuse rate of integrations and assets, and consumption rate by channel, LOB etc.
- **Business process monitoring (BPM):** Commonly tracked metrics are revenue/sales by channel, customer conversion rates

Monitoring and reporting of actionable metrics

It's important to identify the right set of metrics to track, but also to report the metrics in such a way that the organizations can convert the metrics to actions and results. The reports must be designed in a way that they are easily read and digested by leaders and decision-makers.

The reports must be relevant to the audience at that moment. They don't want to be bothered by irrelevant metrics or timeframes. Therefore, the frequency and timing of the reports is critical. The reports must aid leaders to make informed decisions about any improvements or modifications that are needed.

Different teams in an organization will end up developing siloed reporting capabilities. The reason could be that they have different interpretations of the metrics, and needs. They may also have different technology stacks or varying levels of skill sets. However, it is important to rationalize the reporting tools and come up with an enterprise standard.

At MuleSoft, we have developed a tool metrics toolkit to collect, aggregate, and load platform metrics into different visualization systems, providing out of the box integrations and visualization options, including useful dashboards and charts.

To learn more about how to transform the metrics from just data to visualizations, dive into our explainer on how to move from definition to KPI dashboard.

How to choose the right tool to collect metrics

Equally important to choosing the right set of metrics is instrumenting the means to collect them. API monitoring helps with evaluating the performance or behavior of the APIs by collecting metrics from various points of views (e.g. testers, developers, product owners, and deployment admins).

A performance architect’s focus might be on the scalability of the performance load of a system, while a tester might examine the input and output of an API to validate the structure and expected results. In this way, API monitoring can be effective at informing many different perspectives across the organization, making for an efficient optimization tool.

Integrate the monitoring tool into the CI/CD pipeline

API monitoring can be incorporated into the deployment automation process on your CI/CD pipeline so that it can establish continuous observability to collect metrics and derive actionable insights. The intent is to instrument Anypoint monitoring and other enterprise solution monitoring into the CI/CD framework and collect metrics even during the deployments so we can get more operational insights.

Consider API privacy when choosing a monitoring solution

Some monitoring tools may use third-party solutions that require you to open certain firewall ports to monitor internal APIs that should not be publicly reachable. These, in turn, may expose a security risk for the organization. So, it is very important to pick the right monitoring solution depending on whether what you want to monitor is an external asset or an internal asset. An example solution from a trusted product vendor, such as Anypoint Monitoring from MuleSoft, is best suitable for this use case.

Combine API testing with monitoring

Organizations may want to know immediately when an API is slow or down. This creates a perfect need to pick a tool that does both monitoring and testing. It creates a comprehensive end-to-end assessment of the performance of applications. With such a combined solution, organizations will get a 360-degree view of API quality and performance in a single view. Anypoint Functional Monitoring is a great example for monitoring the status of applications

Final thoughts

Metrics are used in every organization to describe the behavior of the systems. In almost all the enterprises, metrics are often applied to technology as well as business functions to track the performance and progress towards certain objectives. Clearly, metrics are a good means to evaluate whether a system is working, or not, depending on the organization’s specific measures.

To measure the success of your integration strategy, it is very important to identify the right set of metrics, design the right instrumentation to capture them and develop meaningful dashboards to visualize them. If you are a MuleSoft customer, you can use our metrics toolkit to begin your journey into KPIs and visualization of metrics.