Apica®

Service-Level Assurance Through Active and Passive Monitoring

Validate Critical Business Transactions with Apica

What is Service-level Assurance?

Service-level Assurance defines what is essential to your business and measuring the elements that support your Service-level Indicators (SLIs). Collectively, these SLI's support Service-level Objective (SLO) goals. And SLO's underpin the Service-level Agreements (SLA's) that you have with your upstream and downstream suppliers and vendors.

Monitoring independently validates your SLOs and SLAs. You measure success for all stakeholders with a comprehensive Service-level Assurance monitoring program that provides actionable feedback to achieve business outcomes.

Measure What Is Critical

What is a critical user journey for your company? Is it a user interacting with a mobile application? Is it an account login from users of a web portal to a SaaS-hosted application? Is it a retail check-out cart?

How do you measure these user journeys to understand their experience and the supporting services better? Do you know the global and regional availability for these potentially complex and interconnected services? In short: are you measuring what is truly critical?

You need to acquire reliable data on your website's uptime and performance to help proactively correct problems before they impact your customers, people, and your bottom line.

Performance Impacts

How much of a difference does a couple of seconds make? It's easy to gloss over the cost of a sliver of time, but research shows that a one-second delay in your website loading time can result in a 7% loss in eCommerce conversions. And a whopping 40% of web users abandon websites if they take longer than three seconds to load. Conversely, decreasing mobile site load times by just one-tenth of a second resulted in conversion rates increasing by 8.4%.

Website and application availability also have a tremendous impact on revenue. Many organizations have cited a revenue loss of more than $300,000 for every 60 minutes of downtime.

Measure the Server AND End-user

"What's measured improves," said Peter F. Drucker, acclaimed management theorist.

Relying on server-side monitoring techniques, measuring a server's system resources and capacity ignores the end-user digital experience. Once content leaves a server, the uncertainties of the Internet (analytics beacons, dynamic ads, DNS, and ISP outages) drastically change the end-user experience, especially when seen in a distant country.

So, measuring and monitoring both at the source and where the end users see it ensure that the business' view matches the envisioned experience.

[1] WebsiteHostingRating.com - "100+ Internet Statistics and Facts for 2021"
[2] Thinkwithgoogle.com - "How Speeding Up your Mobile Site can Improve your Bottom Line" June 2020

Apica

Service-level Assurance 101

IT's modern functions are no longer limited to just maintaining servers and operating systems. The complex web of business needs and outcomes is increasingly forcing IT to align itself at all levels, both internally and externally, so that agreements between business organizations are maintained, metrics are identified, and objectives are met. Businesses need to know what to measure and monitor to establish realistic service-level metrics and attain their commitments to all stakeholders.

Create Service-level Indicators (SLIs)

Google's Site Reliability Engineering (SRE) Handbook defines an SLI as "a carefully defined quantitative measure of some aspect of the level of service that is provided."

Examples of an indicator are either application (or network) latency or availability of that service:

- What are the milliseconds of the application response time?
- How many seconds did the end-users wait for a login?
- Is the application/login/database, webserver/website up or down?
- Was the account secured adequately behind a login?

In short, ask yourself if these indicators will add to the end-user experience. If you ignore these indicators, will users continue to stay or abandon your service without you knowing? There are real-world implications of not meeting these indicators. In turn, there is a negative cost of not meeting these indictors:

- Revenue Leakage: Losing revenue without noticing.
- Productivity Loss: If your service is down or you're your users cannot do their jobs.
- User Dissatisfaction: Users expect a responsive, available site on any device to remain loyal and happy.
- Non-Compliance: Not keeping within a contracted SLA can lead to monetary or other penalties.
- Security Breach: Unauthorized disclosure of Personally Identifiable Information (PII) or any confidential information can lose not only customers, but also brand trust and reputation.

A collection of SLIs create a target goal that an enterprise wants to achieve; a Service-level Objective (SLO).

Set Service-level Objectives (SLOs)

The SRE Handbook defines an SLO as a "target value or range of values for a service level that is measured by an SLI."

In short, SLO's are the objectives that you want to meet internally and defined over a fixed period.

Examples:

- Login availability is not less than 99.9% over a year.
- A daily average login response time of not over 2 seconds.
- A 100 percent failed daily login attempts with improper credentials.

Setting SLO's requires care and realistic indicators to support them so that you can meet your Service-level Agreements (SLA's).

Establishing realistic SLO's involves measuring and understanding normal behavior over time in all regions and devices where users consume your services.

By recreating the critical journeys in synthetic and measuring them repeatedly, you gain a sense of normal service levels over time; how well your services perform and how available they are.

As you continue to gather SLI metrics over time, you will see what a normal level of performance is supportable, where improvements will need to be made whether at the infrastructure, network, or geographical level.

Understanding the SLIs over time will allow you to set realistic thresholds for your SLOs and ultimately an SLA that you know is supportable, because you've empirically validated the promised SLA levels.

Set Service-level Agreements (SLAs)

SLA's answer the question, "Contractually, what happens when I do not meet the SLOs I've set with customers or organizations that rely on my services?"

SLAs define a business-level metric that carries defined penalties if not met. For example, if the SLA for an application/platform uptime is 99.99%, Company X will pay financial credits/penalties of Y% of the monthly subscription cost for every Z% of not meeting this SLA.

Active and Passive Monitoring. AKA Synthetic and Real-User Monitoring

So, with SLIs as the measurable building blocks of SLOs that underpin the SLA(s) an enterprise has with customers, setting up a comprehensive monitoring system that assures the service level you are striving for is paramount.

Service-level Assurance comes from an internal and external deployment of monitoring agents, working together to measurably demonstrate that the web of interconnected services is functioning as designed and promised.

Monitoring agents come in a few forms and functions, ideal for exercising an application or reporting metrics.

Active/Synthetic Monitoring

Synthetic, agents proactively test and measure the actions of an aspect of the service. In doing so, they measure SLI's.

These agents can be internal to the organization or external, depending on the viewpoint (e.g., application or end-user) needed.

These agents report these results over time for critical user journeys. The aggregated metrics provide insights into performance and availability over select periods, seasons, and regions—critical dimensions to understand the services. The SLOs can be evaluated against these results to assure the organization that their SLAs are being met.

Passive/Real-User Monitoring (RUM)

This is essentially a deployed sensor network of beacons that return a set of metrics if (and when) that analytics beacon/tag/some monitoring code is activated. This form of monitoring can be thought of as "agentless" and passive in the sense that they lay dormant indefinitely, waiting to be activated.

They are triggered when a user's browser parses a page that they are present on; their code then measures the target metrics and returns the analytics data to the parent system. Hence, they are known as Real User Monitoring (RUM, or more accurately, Real User Analytics)

Analytics information can be voluminous, extensive, and highly variable, such as where it happened, when, what was the device and browser, and how activated (SLIs are contained in these metrics). Additional SLO's can be calculated when the sampling size supports it, however, RUM normally serves to validate the data gathered actively.

Synthetic Monitoring and Service-level Assurance

Synthetic monitoring can simulate critical-path journeys and send traffic in scheduled intervals, providing granular data on the availability and performance of websites, applications, and APIs that support these journeys.

Often referred to as proactive or active monitoring, synthetic provides a suite of repeatable tests that deliver reliable "before and after" data, enabling IT Operations, Development and DevOps teams to measure the effectiveness of code changes and troubleshoot specific transactional issues.

So SLIs (and therefore SLOs) depend on synthetic monitoring platforms such as Apica's to measure selected metrics, active and drive supporting services, and trend these responses over time.

This proactive approach allows companies to understand response times and availabilities for selected locations and critical user transactions. And it also forces the underlying systems to activate and support the overarching company service regularly.

Real User Monitoring and Service-level Assurance

Real user monitoring (RUM) is a user-driven monitoring technology that tracks actual user activity on websites and applications. RUM observes websites and applications in real-time, tracking availability, responsiveness, and functionality.

It also provides IT Operations and Development teams with insight into how users experience an application. This type of monitoring observes users by device, browser, and network access to help the business put performance issues into context.

While some RUM tools analyze every user's transaction, others observe a smaller set of users representing the whole. Since RUM tracks audience demographics, behavior, and website or application performance metrics, it makes it an ideal complement to Synthetic Monitoring because it can help validate the assumptions that set your SLO's.

The SLA Monitoring Program

Synthetic Monitoring can regularly exercise critical user journeys that are important to the business. In contrast, Real user monitoring (RUM) is a verification and validation tool covering conditions and locations where your monitoring agents are not.

So, create and deploy synthetic monitoring wherever you need assurance that you are delivering what you are committing to your SLA. But use RUM to help validate that you're capturing the SLI metrics and that SLOs are properly calculated.

Monitoring Best Practices

Here are Apica's general recommendations for setting up an effective monitoring program:

Set up synthetic monitoring agents externally from as many locations and environments as you can reasonably afford

Cover your user base widely and deeply. Avoid having single monitoring agents as they have no redundancy nor the ability to be checked against another result from another agent.

Measure all aspects of a critical journey, so every component is measured and its effects on your services captured

DNS: Every HTTP request begins with DNS, so do you check your DNS responsiveness or availability (to enforce the SLA they might have committed to you)?

TLS: Most critical and modern transactions include TLS/SSL to secure transmission and content transferred between the end browser and the application. Are you monitoring the SSL certificate's validity for that hostname to ensure that you still own or control it?

Maintain the care and feeding of your APIs

If you have mobile apps or web applications that rely on APIs to function, do you check the API host provider's responsiveness? And do you know what the availability of that service is? Do you exchange data via API REST calls to vendors or providers? These vendors or providers need to either upload or download data with your API. Where do you measure these services?

Remember that API endpoints also need DNS and TLS monitoring to function securely and reliably.

Ensure every user experience

Create a synthetic, critical user journey through your service, something every user needs to traverse. Or you need to ensure that your service is serving them consistently and predictably—from DNS resolution to SSL handshake to Login and Logout. How can you do this without active monitoring? Active monitoring will ensure that you activate all the sub-services supporting your application (APIs, database calls, application servers, containers, etc.).

Monitor from desktop, mobile applications, tablets, mobile browsers and more

Users have expectations that differ depending on the device that they are using. Mobile networks differ drastically in their latency and speed. Mobile devices also have less processing power than a desktop. A responsive website may look good on a Desktop and a mobile device connected to the corporate Wi-Fi, but what about a page banner served full-size (but scaled down) to an older mobile device on 3G or 4G?

Active Compliments Passive Monitoring

You can now see the disadvantage of only relying on synthetic monitoring; for a given script deployed to an agent, you can only monitor one transaction from one location on one device.

Also, as your applications change, scripts can break and need maintenance because they must adjust to changing navigation paths or application features. And, if you are not capturing all metrics, you cannot correct what you cannot measure.

An example use case and transaction for banking would include navigating multi-factor authentication and checking a balance. A synthetic use case would include executing the chain APIs to complete a transaction for a mobile application.

As the number of service scenarios grows, it may become impractical to actively monitor all potential platforms, devices, and locations because you need to create scripts for each of these. For cases where active monitoring becomes impractical, RUM can take up that niche—it is excellent for measuring metrics from remote locations and devices. But you can set up synthetic monitoring to start validating your SLAs at a granular level: measure the SLIs, determine if the SLO's that you want are realistic and consistently achievable so you can make your legal commitments in binding SLAs.