DEFINING A STANDARD FOR CLOUD AND SAAS QUALITY

THE CHALLENGE OF SATISFACTORY SERVICE DELIVERY

Today, we spend more and more of our lives interfacing with internet-connected services, whether computers, mobile phones, tablets or wearable electronics. Services are increasingly delivered through SaaS applications and, more generally, cloud-based APIs. Users expect their services to function quickly and without error, and those creating and distributing services expect their providers to guarantee quality that meets customer demand. But how does one determine the quality of a cloud-based API?

Understanding the quality of any given API is a complex problem. Multiple factors come into play when trying to quantify whether one has functioning and reliable cloud-based APIs or SaaS products. Both location of the end-user and location of the hosting service can and do affect performance. In our research, we have also found that not all clouds are created equal; the cloud provider can significantly impact performance for the end-user. Performance-affecting issues can crop up due to unforeseen problems both between cloud providers and from a given cloud provider to a given end-user. Cloud and SaaS providers measure what they supply, not necessarily what their users consume. This can lead to issues where a service distributor has been assured there should be none.

The impact of poor performance cannot be understated in today's increasingly cloud- and SaaS reliant environment. In terms of end-user experience, poor performance will manifest in longer wait times for content (be it a response, a login or a result), higher user churn, and poor perception of the service they were attempting to use. In terms of cost to the group using a given cloud or SaaS service, poor performance results in lost time and money. Fixing issues and reporting on the causes takes time and effort, and depending on how poorly the service that one uses is performing, this can significantly affect a group's annual IT spending.

There are currently no established guidelines on what is and is not acceptable in terms of real-world API performance. Through extensive research and testing with our clients and many major cloud service providers, we have identified key areas that can be used to measure the performance of a given API. Latency, or how long a call or sequence of calls to an API takes from the request of certain information to its delivery, is key in determining performance. Along with this, variability in performance can greatly affect the experience of end-users. Is your service consistent? Are there variations in latency across regions or different calls? Are there times of day or extended periods when your service isn't effective or completely unusable? These factors and more need to be constantly monitored, analyzed and reported on in order to keep your end-user satisfied.

INTRODUCTION

After more than three year years of implementing our intelligent monitoring and reporting system, APImetrics has developed a new industry standard that can be applied to all SaaS and cloud-based services using APIs to deliver services to customers. The Cloud API Service Consistency (CASC) standard and the accompanying CASC Score will allow those who distribute, create, and consume data to be confident that the services they are using are working as they should. Standards in technology are an essential part of any widely used system, ensuring better reliability and a higher quality of services. Historical data shows that users and consumers perceive products complying to standards as more dependable. This raises user confidence, leads to greater sales, improves adoption of new technologies, and reduces churn of products. Whether or not the end user is aware of the standards that drive a given system, the benefits of standardization affect them positively. Standards have been widely adopted in the world of physical products, but with the rapid pace of innovation in field of technology and software, the standards for API performance have not yet been fully realized.

If you are purchasing, consuming, or providing API-reliant products, consider the following:

- Are you able to compare the value of the service, like for like with another?
- Are you able to hold your SaaS or cloud provider accountable based on metrics beyond a lackluster SLA?
- Is there an accepted methodology for measurement of performance?
- Can you easily identify trends and anomalies in the services you are utilizing?

WHAT AND WHEN TO MEASURE THE QUALITY OF CLOUD AND SAAS SERVICES

As a provider of Cloud and SaaS services, it is best to know of any potential issues in an API before external complaints, whether from clients or end-users, make you aware of them. The best time to measure the quality of Cloud and SaaS services is as frequently as possible, though this is not necessarily what most providers do. Constantly monitoring all possible calls and sequences that could be made through an API is extremely time-consuming and costly if not automated correctly, so many providers consider API monitoring as a reactive rather than proactive decision. The "how" of measuring quality is more complex.

When looking to measure the quality of a given API, a few key metrics can be used to determine overall API health: latency, consistency, and sample size depth.

THE CASC STANDARD

LATENCY

The latency of a given API call or a sequence of calls is a key indicator of overall API health and end-user satisfaction. If latency is kept low, end-users will never notice a problem. If the latency of a given API call or sequence of calls is high (for example, more than one second), end-users will often refresh the page or simply leave, the former not solving their problem and often leading to the latter.

CONSISTENCY OF SERVICE

A healthy API will consistently succeed at returning data in a timely manner regardless of the time of day, location of the data being requested, or location of the end-user requesting the data. In our research, we found that many publicly available APIs provided by both commercial and federal groups did not meet our baseline for consistency. Some consistently failed to provide information if requested from a certain location, while others had hours or even days where the information was inaccessible from any location. To confidently and truthfully claim that your data is accessible, whether to the public or to paying customers, you must provide a consistent, stable experience when accessing the given API.

> **ONE ORGANIZATION LOST 10% OF THEIR QUARTERLY REVENUE AS A RESULT OF LATENCY ISSUES — ISSUES THEY DIDN’T DETECT UNTIL AFTER THEY HAD LOST MANY CUSTOMERS.**

SAMPLE SIZE DEPTH

To maintain a healthy API, multiple call sequences must be tested with high frequency. Simulation of actual end-user activity and reporting on the results of said activity is critical. This is often overlooked or intentionally disregarded, as setting up realistic and strenuous testing that mirrors an end-user is an expensive and time-consuming process. Along with setting up the right call sequences, they must be frequently run in order to fully understand the health of your API. If you choose to monitor your API infrequently, you will risk missing outages or other important and actionable data. Further, these calls ought to be made from multiple locations in order to inspect whether an API is performing consistently in all regions.

THE CASC SCORE EXPLAINED

RATING	SCORE	IMPACT/RATING
Poor	0-399	End users will have significant problems using the APIs with a high percentage. You can expect dozens of outliers and problems in a given month. User complaints and frustration with the service will be unacceptably high.
Below Average	400-599	Multiple events and incidents per month will affect end users. The API will require significant engineer man-months of effort per month to fix problems as they occur. End users are likely to have a poor perception of the quality of the service.
Fair	600-799	You will have incidents on a regular basis that will appear to have no root cause and impact end users. Deep investigation of issues and a potential re-architecture may be required.
Good	800-999	Very few incidents - keep it up. You are still likely to get a few incidents a month where a client or user will notice a problem.

REAL WORLD IMPACT OF UNHEALTHY APIS

The CASC Score grid above reveals deeper insights into the impact that a problematic API can cause for all those who use it.

For example, consider that your users have cyclical use patterns - that is to say, they use the product during the day, but the service goes down during the night. Without proper monitoring, the malfunctioning systems would not be noticed until early users the next day had a chance to see that they cannot access the services. Another example is that users typically do a particular action weekly or monthly, such as paying bills or checking their bank accounts. If a problem arises during non-peak days, a large number of your users may find the service inaccessible or substandard when they attempt to access the service on a peak day. These sorts of problems can be avoided by early recognition with the proper monitoring tool.

Practically speaking, if services are poor, your clients will look elsewhere for a similar service. Consider this real world use case: one organization's API and Cloud services experienced a drop in performance relative to competing alternatives. Users experienced high latency, and many simply abandoned the service in favor of the competition, which had smoothly functioning API and Cloud services. It wasn't until the following quarter that the organization realized their revenue had dropped significantly -- far too late to do anything about the issue. A CASC score and proper monitoring tool would have alerted them immediately to the problem, giving them a chance to fix it before losing customers.

AN API WITH A GOOD SCORE will still experience between four and six incidents a month with no readily apparent root cause - for example, a server failure or outage. These incidents will need to be investigated. By using APImetrics in conjunction with our CASC Score, a provider can significantly narrow any search for the cause of a problem. By monitoring the API from multiple locations with a high frequency and using complex sequences of calls approximating end user activity, APImetrics helps cut down on the high cost of maintenance and repair.

AN API WITH A FAIR SCORE will generate, on average, twice as many issues as an API with a good score. As an example, APImetrics has been running calls on over two dozen of the most popular federally published APIs from Data.Gov. We found that over fifty percent of these fell, at least, into the range of fair. This indicates that multiple incidents will disrupt your end user's activity every month, and will cost a provider substantial time and money to fix.

AN API WITH A BELOW AVERAGE OR POOR SCORE will have dozens of incidents each month and will be almost completely unusable for any given end user. The costs associated with maintenance and issue resolution will quickly eat away at even a sizable IT budget as headcount is diverted from core tasks to impact investigation and resolution. In order to resolve problems that crop up, you must know the scale of the problem. Is the problem affecting only a certain area or all users? Where on the site is the issue being experienced, or is it site wide? Constant monitoring of an API allows engineering teams to respond to issues affecting a service quickly. By automating call sequences to mirror the activity of end users, engineers can rapidly narrow the potential problem areas in an API, saving time and money.

WHAT DOES AN UNHEALTHY API COST?

Through proprietary research, we have developed a system for understanding the potential costs associated with an unhealthy API. Though initially developed for the private sector and its pricing, we have modified it here with the standard pricing of a network engineer as per CALC, a contractor pricing site run by the GSA. The numbers that follow assume a fully loaded price per hour of $103 for a network engineer.

We have found that the number of incidents, which covers any and all problems requiring investigation, can be estimated for each level of the CASC scale. An API with a poor or below average rating will experience about 11 incidents per month, which each incident lasting approximately 28 hours from the time the incident starts to its resolution. An API with a fair rating will experience about 9 incidents per month, with each incident lasting approximately 24 hours. An API with a good rating will experience about 7 incidents per month, with each incident lasting approximately 14 hours. With these numbers combined with a standardized pricing for a network engineer, we are able to estimate the cost of repairs to an organization per annum.

COST OF REPAIR WITH INVESTIGATION & RESOLUTION:

$350K-$400K	Poor/below average score
$200K–$250K	Fair score
$100K-$150K	Good score

Based on our extensive research, an API with a poor or below average score could cost its organization $350,000 - $400,000 per year to repair and investigate after incidents. An API with a fair score will cost its organization $200,000-$250,000 per year to repair and investigate after incidents. An API with a good score will cost its organization $100,000-$150,000 per year to repair and investigate after incidents. As is plain from the amounts presented, API health is a large factor when it comes to determining the cost of an API over time. The price of a poor or below average score can be hefty - an organization using our calculation can expect to spend more than a quarter of a million dollars more if their API is scored as poor or below average than they would if their API was scored as good.

It is our hope that this mental exercise in incident cost drives home the need for rapid adoption of standards. The need for good monitoring solutions and automated reporting can be a lifesaver, as when adding in reporting costs you can expect to increase your overall API budget by many more thousands of dollars per year.

API MEASUREMENT

Testing is necessary to determine the overall health of an API. Organizations looking to monitor their existing systems have a number of options to choose from.

MANUAL TESTING

This is the least effective and, in some ways, most time consuming method. An engineer working for an organization can manually test the various possible calls to an API to retrieve relevant data concerning the health of an API. This will require a large allocation of time on the part of the engineer and, consequently, a large allocation of funds to support the manual testing of the system. This is the least effective method of API measurement.

THIRD PARTY SOLUTIONS

Existing third party solutions for API monitoring can effectively give you a partial picture of the overall health of your APIs. Many of the available solutions provide the ability to track the latency and uptime of your system over a given time period, and some provide the ability to easily report on results of their testing. However, potential problems can crop up with third party solutions, including availability of post-purchase support, training, and implementation. These issues can end up costing an organization a significant amount of resources. Not uncommonly, the engineer who contracted a third party solution without continuing support sets up the solution to monitor, but after a few months the engineer leaves.

The third party solution is now useless until another team member can be trained on the use of the system or another solution is acquired.

In addition to pulling together all the monitoring services, the data generated must be manually reported and analyzed, often requiring multiple time-consuming steps involving other products such as Excel. With the vast quantities of data involved, there is a strong possibility that something important will be missed.

ENGINEERED SOLUTIONS

When looking at the cost of putting together solutions, engineers will often overlook their own cost. This will lead to engineered solutions where many pieces are cobbled together over time that either work only with the engineer who initially installed them or become a costly endeavor for others to use. Stack monitoring, web monitoring, and network monitoring tools can be layered together to ultimately deliver actionable reports to an organization, but this can take extraordinary amounts of time and money to accomplish. In a worst case scenario, when the engineer who created the solution leaves an organization, they are left with no one able to use it. It is far better to start with a product that is easy to learn and reports automatically on the health of your Cloud and SaaS API services.

APIMETRICS

APImetrics is our solution to the problem of monitoring API health. With a comprehensive suite of measurement tools and automated reporting to keep organizations up to date on their service quality, we provide everything needed to monitor all essential cloud based API systems. The CASC Score is automatically delivered on a weekly and monthly basis to all APImetrics users, whether they implement our constant monitoring services or a more infrequent health check option. All organizations that work with APImetrics have access to our reporting tools, giving them insight into their API health beyond what is normally available with any other commercial or open source alternatives. Our system is easy to learn, thanks to extensive documentation, and our engineers are available to support organizations in setting up their monitoring, as well as providing continued support throughout the lifecycle of any given API.

SUMMARY

With API reliability becoming an increasingly vital issue, it has become essential to establish standards that outline what is considered healthy and acceptable for an API and what is not. Our development of the CASC score is the first step towards laying a foundation for industry-wide standards. With its easy-to-read numerical measurement, this 'credit score for APIs' provides a simple, convenient way to understand at a glance the health of any given API as well as that API's health compared to others. This information can be then used to identify and correct any issues before they cause damage to an organization's revenue or reputation, ultimately resulting in happier end-users and happier organizations.

ABOUT APIMETRICS

APImetrics provides the first end-to-end performance monitoring solution aimed at developers, API providers and enterprises. With APImetrics you can set up and schedule complex and authenticated API calls that test the latency and performance of the APIs you provide or use, then provide real time reporting via public and private dashboards and our own reporting APIs.

For more information:

http://apimetrics.io
info@apimetrics.io