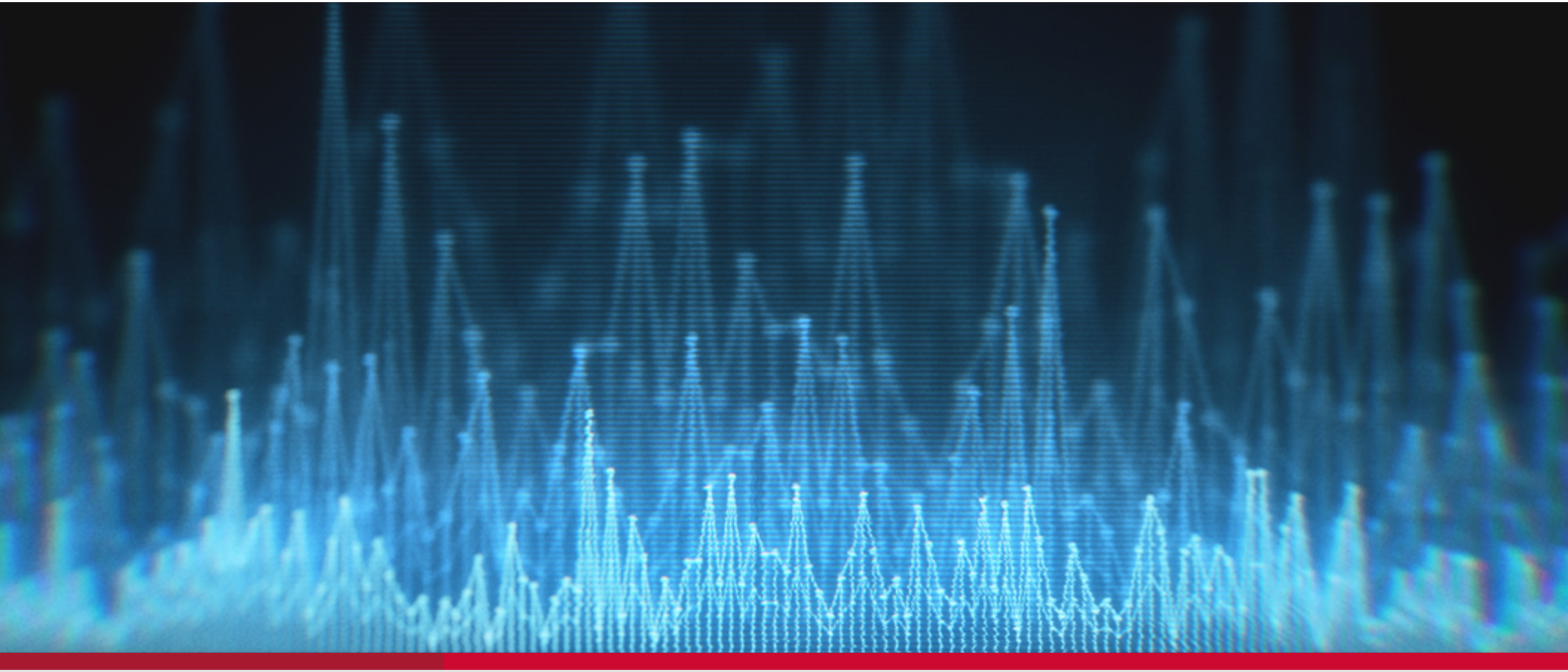
A Broadcom Company

technologies



Building an API Monitoring Practice

for Modern Apps, Containers and Microservices and Microservices

The Importance of API Monitoring

One of the most significant differences between today's software applications and those of the past is the role played by APIs.

Although APIs have existed for decades, the widespread adoption of microservices architectures, containers, serverless computing, IoT smart devices and other next-generation technologies has made them much more important. Modern applications rely extensively on APIs to facilitate internal and external communications.

The large-scale embrace of APIs has not only changed the way applications are written and deployed; it has also created new imperatives for maintaining software quality.

In a world where APIs serve as the glue that melds disparate applications or parts of applications, API performance problems, whether they involve complete API unavailability, API data validation errors, or slow-to-respond APIs, can quickly make an application unusable and undercut user experience.

What's more, it's often not possible to detect API performance issues using the monitoring tools DevOps engineers have traditionally had at their disposal. Typical application performance monitoring (APM), infrastructure monitoring, security monitoring and other types of monitoring tools do little to identify API-related problems. As a result, an application that appears to be functioning normally according to conventional monitoring software may actually be suffering significant API performance issues. And when an application is experiencing a problem, it can be difficult to trace its cause to an API using conventional monitoring tools.

This is why DevOps teams are now adding another category of monitoring—API monitoring—to their toolset. As this eBook explains, API monitoring tools are essential for helping to ensure software performance and quality and, by extension, for driving business value. While conventional monitoring tools will not disappear, monitoring toolsets that lack the ability to discover API performance and reliability issues leave a critical blind spot in the vision of DevOps teams.

What is API Monitoring?

API monitoring is the practice of identifying and resolving availability or performance problems that affect APIs. API monitoring delivers several critical types of visibility into application performance and quality by supporting several types of use cases:

- Uptime monitoring: API monitoring tools detect APIs that stop responding or respond intermittently.
- Performance monitoring: APIs that are available but are responding slowly to requests can cause the applications that they support to perform slowly too.
- Data validation: When an API sends or receives the wrong data, significant application performance problems can result. API monitoring tools that support data validation help prevent this problem.
- SLA satisfaction: Because APIs play a vital role in connecting applications to third-party resources, SLAs may require certain levels of API performance to be maintained. API monitoring tools can help to meet SLA requirements.

API monitoring applies to any type of API architecture. Whether an application uses REST-style APIs, SOAP, RPC or something different, modern API monitoring tools can help ensure the availability and quality of the APIs that the application needs to meet user expectations.

In addition, API monitoring tools are valuable for managing the availability and performance of internal APIs (which different microservices inside the same application use to communicate) and third-party APIs (which link one application to an outside application or service).

API Monitoring vs. Other Types of Monitoring

To a limited extent, the functionality of API monitoring tools and other monitoring tools overlaps. APM and load testing tools, in particular, can also identify application performance issues whose underlying cause may be an API.

However, while APM and load testing tools might be able to detect when an application is slow to respond, they are ill-suited for identifying why if the root cause of the issue lies with an API. That is because these tools focus on applications themselves, not the APIs that serve as communication vectors between applications.

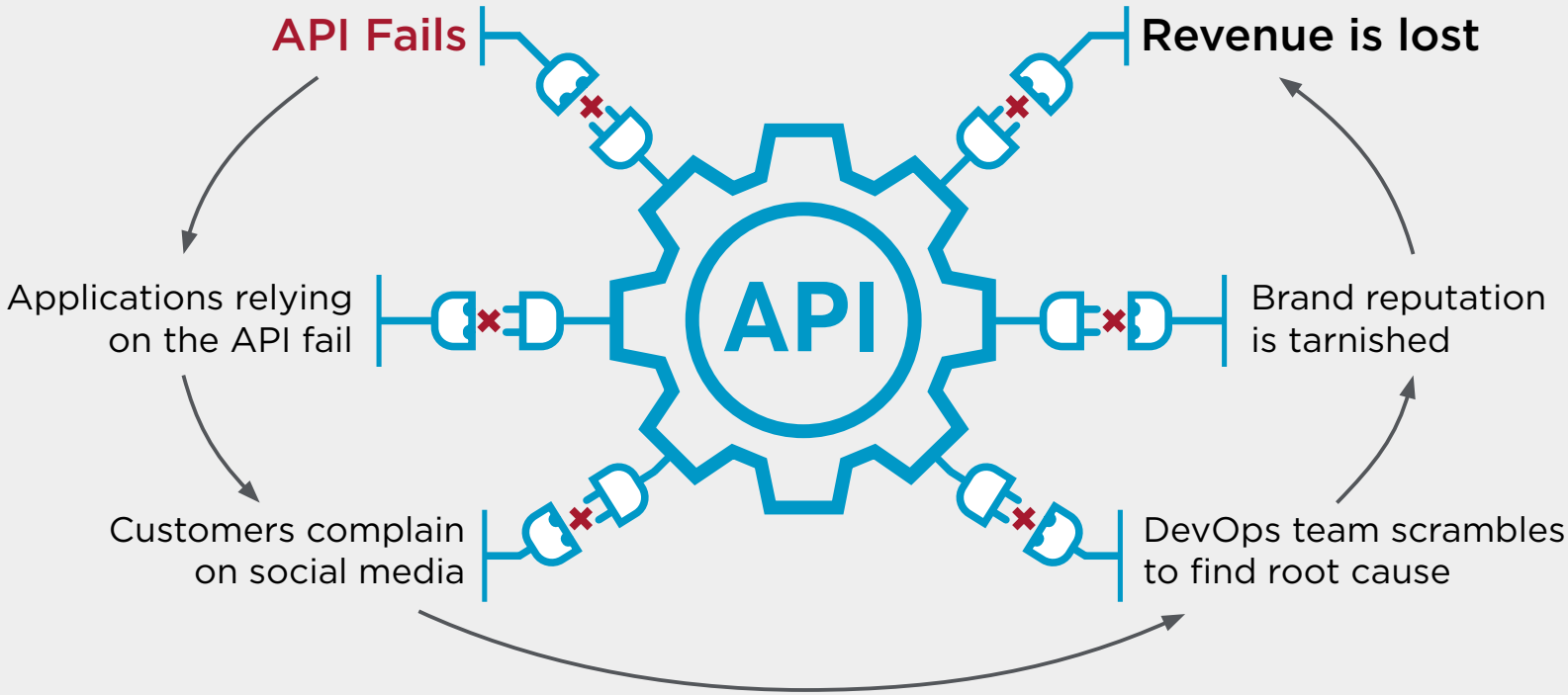
As for other types of monitoring software, such as infrastructure monitoring and security monitoring tools, they focus on very different monitoring workloads and provide virtually none of the insight that API monitoring offers.

In contrast, API monitoring tools are designed to identify the root cause of API performance problems quickly. If an API is slow to respond due to limited network connectivity, API monitoring tools will lead engineers quickly to the source of the problem. If data formatting problems are causing a miscommunication between two services that exchange data using an API, the monitoring tools will identify that issue, too.

The Risk of API Failures

APM and infrastructure monitoring tools tend to dominate the IT monitoring conversation. It can therefore be easy to assume that the types of issues these tools are designed to monitor are the main risks to application quality and user experience today.

An API failure is an application failure. What happens when no one is watching...



Most applications now have hundreds of APIs, making remediation time consuming. When even one API is down, so is your business.

Yet API performance is just as critical, especially because, as previously noted, APIs have assumed a more important role than ever in connecting applications. To understand how seriously API performance problems can harm software quality, consider the API failure scenarios on the following pages.

Internal API Failures in a Docker Environment

In an environment where applications are written as sets of microservices and deployed using Docker containers, APIs are essential for managing the containers, as well as for allowing different microservices and containers to exchange data. If the API that a Dockerized application uses to communicate internally stops responding or handles requests slowly, the application itself will stop operating properly.

For example, if a microservice that handles user authentication can't connect to a credentials database microservice because the API that connects these two services becomes unavailable, the application will cease to be able to authenticate users, rendering it unusable. This type of internal API failure will harm the user experience and possibly cause an application to fall short of meeting SLA commitments.

Third-Party API Failures and a Banking App

Delivering a positive customer experience often requires integrating an application with other applications or services provided by partners. Third-party APIs enable this interaction.

For example, consider a financial application that uses an open banking API to allow customers to transfer funds and pay bills between accounts held at different banks. A data validation error or slow response from the third-party APIs on which the application relies might cause transactions to fail, leaving customers dissatisfied and possibly concerned about the security and stability of an application that controls their bank accounts.

What's particularly challenging about this type of problem, from the perspective of DevOps teams, is that a failure related to a third-party API might result from an issue beyond the control of a company's developers and admins. An external application might send invalid data because it has not been updated to the latest version of the API, for instance, or an API could become unresponsive when an external resource that it connects to becomes overloaded.

From the user's standpoint, the reason for an API failure does not matter. Even if an external resource connected by a thirdparty API is at fault, users will still become frustrated. That is why DevOps teams must be able to detect third-party API issues quickly, so that they can take steps to mitigate the impact of the issues on their applications even if they cannot fix them entirely.

External APIs and Data Backup Services

As more and more data moves to the cloud, APIs are playing an increasingly important role in getting it there and keeping it accessible. That is because cloud providers make their storage services available for integration with third-party applications via APIs.

If those APIs experience performance problems, the applications that depend on them for mission-critical functionality may also fail. For example, a data backup application that relies on a cloud host's data API to perform nightly backups of a company's data might stop performing backups when the API goes down. If the API error is not addressed quickly, critical business data could be left at risk. Compliance requirements could be violated, too.

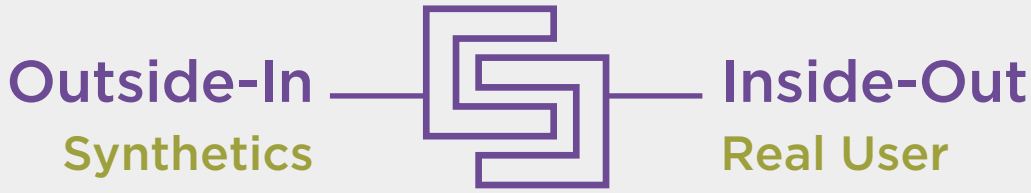
APIs are Everything

In each of these scenarios, API performance can make or break the ability of an application to perform its job and meet user expectations. And because the types of failures described above are not likely to be detected by monitoring tools that are not designed specifically to monitor APIs, they may go unnoticed by DevOps teams that rely on conventional monitoring tools alone.

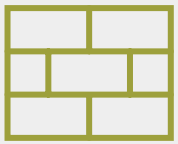
Synthetic vs. Real-User API Monitoring: Why You Need Both

Once an organization has made the decision to adopt API monitoring tools, it must decide which specific types of API monitoring (synthetic or real-user) to deploy in different scenarios.

This is an important issue, because synthetic monitoring and real-user monitoring cater to distinct types of use cases.



- API Performance
- SLA Monitoring
- Data Integrity
- Performance at Scale



- PI Performance
- Full Trace Visibility
- Component Break-Down

Synthetic Monitoring

Synthetic monitoring solutions provide insight into API uptime, performance and data integrity so you can find and fix API issues before they impact the customer experience. Synthetic monitoring uses simulation or emulation to mimic an application environment. As a result, it enables a range of different environment variables to be tested easily because the test environments can be virtualized. In this way, synthetic monitoring provides a global view of API performance and availability across a range of software environments.

One of the limitations of synthetic monitoring is that, because tests are scripted rather than carried out using real users, test results may not always reflect real-world conditions with complete accuracy. In addition, testers lack the ability to trace a problem into back-end systems and determine which users it affected.

In the context of API monitoring, API mocks and service virtualization enable synthetic monitoring insights by simulating APIs without requiring resources to be fully present.

Real-User Monitoring

Real-user monitoring solutions provide insights into the user experience and transaction performance across APIs and backend services with the ability to quickly drill down into the root cause of API issues. Real-user monitoring (sometimes abbreviated as RUM) involves testing software in real-world environments, with “live” users.

While real-user monitoring provides more detailed and accurate insight into software behavior and user interactions, performing such monitoring on every application is impractical. To perform real-user testing and monitoring of APIs, DevOps teams must set up real-world environments and trigger API actions in response to actual user behavior.

While the investment of time and resources required to perform real-user monitoring is not practical in all situations, it is often worth the effort for testing and monitoring mission-critical APIs. Real-user monitoring can also be paired with APM tools to help quickly identify the root cause of an API performance issue by homing in on the particular environment variables that have triggered the failure.

A Comprehensive API Monitoring Strategy

Because synthetic monitoring and real-user monitoring cater to different use cases, it's best to factor both into your monitoring strategy. An API monitoring tool set that supports only one or the other will lack either the monitoring breadth provided by API mocks and service virtualization or the depth delivered by real-user monitoring.

Outside-In

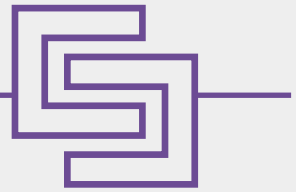
BlazeMeter® Continuous Testing

Be the first to know when an API is down or slow by continuously monitoring APIs

Gain a global view of API performance by synthetically monitoring from locations around the world

Verify data integrity to ensure that APIs are returning the expected response

Ensure SLAs are being met for your APIs and the third-party APIs your app relies on



Inside-Out

DX Application Performance Management

Gain real-time, in-depth and customizable visibility into critical API performance data

Optimize and monitor API gateway traffic health via analytics and instrumentation

Speed root cause analysis with targeted triage of API problems and anomalies

Leverage detailed visualizations, automated detection and alerting of API services

Over 1,000 customers depend on Broadcom for API monitoring

Integrating API Monitoring into Your Monitoring Stack

As noted above, API monitoring tools provide functionality that other types of monitoring solutions can't match. To make the most of API monitoring tools, DevOps teams must understand when and when not to deploy them, or, in other words, how to integrate API monitoring solutions effectively into the stack of monitoring tools at their disposal.

API monitoring tools can enhance an existing monitoring tool set in the following ways:

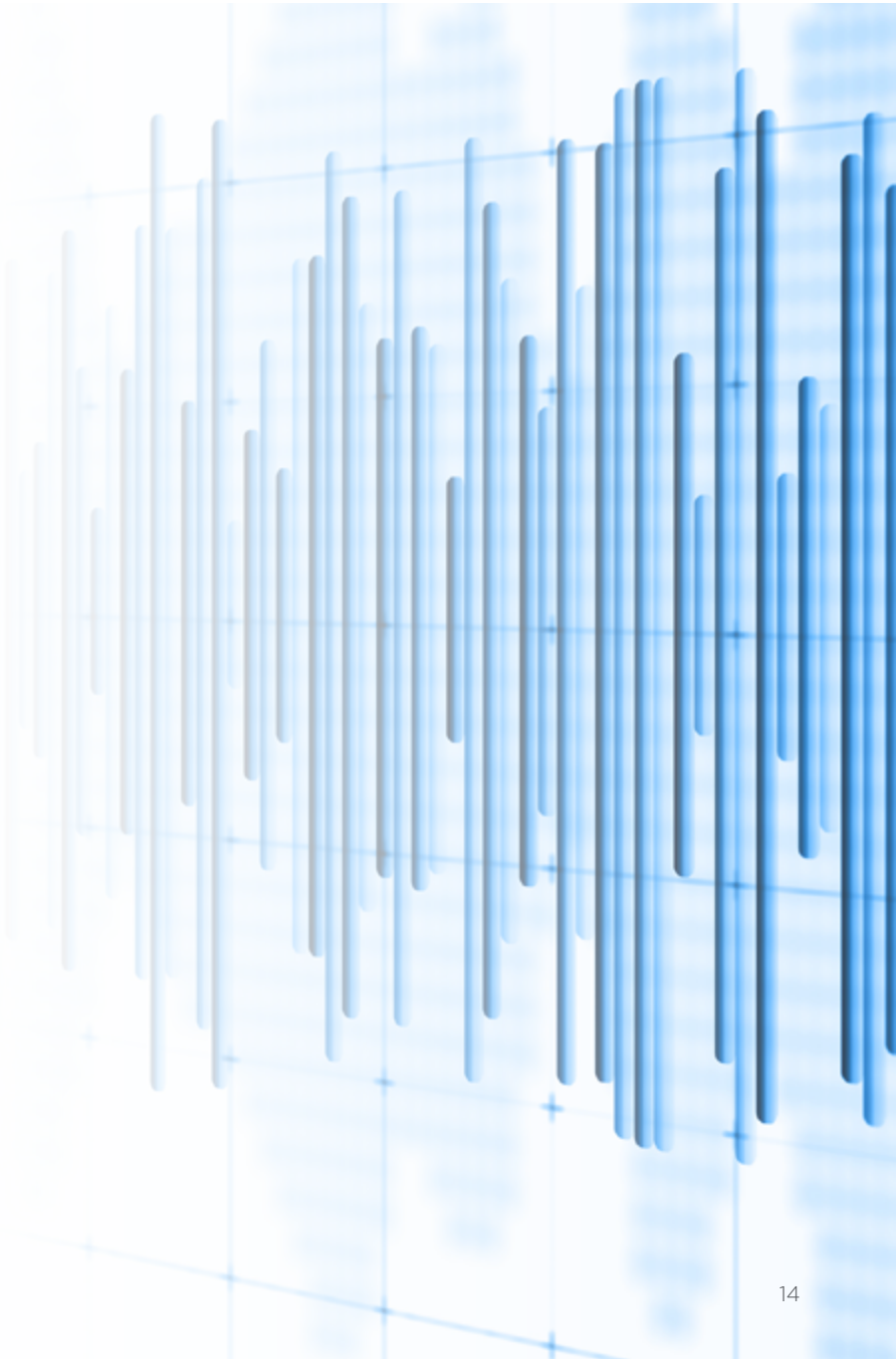
- When used alone, they can help to test and validate API performance prior to an application's release into production.
- When used alone, they can identify API performance issues within a production environment in real time.
- When used in conjunction with APM tools, API monitoring tools can help to diagnose and fix an application performance problem whose root cause lies with an API. In this case, the APM tool might identify the overarching issue while the API monitoring tool traces it to an API.
- When used in conjunction with infrastructure monitoring tools, API monitoring software can help to determine when an infrastructure problem (such as a network switch failure or slow disk I/O) is contributing to poor API performance.

API tools thus play valuable roles at all stages of the continuous delivery pipeline. During software development and testing, they help to identify and resolve application performance issues before software is released into production. And in production environments, API monitoring enables real-time visibility into API performance to help DevOps teams find and address issues before they impact the business.

Conclusion

In short, no monitoring strategy is complete in today's highly dynamic, microservices-oriented world if it does not include API monitoring tools. These tools should be able to support the range of API monitoring use cases described above (API uptime, API performance, API data validation and more) while also enabling both synthetic and real-user monitoring. And they should be paired effectively with other types of monitoring tools in order to provide holistic visibility into software environments and enable quick response to performance issues of all varieties.

The suite of monitoring solutions available from CA Technologies, a Broadcom company meets all of these criteria. BlazeMeter Continuous Testing platform enables a range of API monitoring and testing, while DX Application Performance Management solutions provide the complementary insights necessary to leverage API monitoring to its fullest potential.



Broadcom Inc. is a global infrastructure technology leader built on 50 years of innovation, collaboration and engineering excellence.

Broadcom Inc. (NASDAQ: AVGO) is a global technology leader that designs, develops and supplies a broad range of semiconductor and infrastructure software solutions. Broadcom's category-leading product portfolio serves critical markets including data center, networking, enterprise software, broadband, wireless, storage and industrial. Our solutions include data center networking and storage, enterprise, mainframe and cyber security software focused on automation, monitoring and security, smartphone components, telecoms and factory automation. For more information, go to www.broadcom.com.

For product information please visit our website at: broadcom.com

BC-0571EN January 20, 2020