

# An introduction to Application Performance Management (APM)



With the rise of digital and the proliferation of on-demand services and software, applications are no longer just part of a business—in most cases, **they are the business.**

No matter how easy it is for customers to use your application on the front-end, modern digital experiences depend on the successful orchestration of a broad collection of technologies on the backend—from mainframes, multicloud, and microservices, to machine learning, APIs, and more. Left untamed, that kind of complexity can translate to monumental challenges related to monitoring, measuring, and optimizing the performance of the digital experience, and that puts the application experience and your business at significant risk.

To combat these challenges, many businesses make use of application performance management, or “APM” for short. But not all APM solutions are created equal: Some fall short of the observability they promise, while others make it difficult to put performance problems in the context of the business. That’s why, in this guide, we’re going to provide a short introduction to the basics of APM as well as provide advice about evaluating an APM solution to meet the needs of your digital business.

**Before we go any further, however, there’s one important call-out we want to make around the term application performance management, and a related concept, application performance monitoring.**

While both are referred to as “APM,” the differences between the two terms can be a bit murky. Instead of focusing on the difference between these two terms, let’s focus on what really matters when it

comes to APM, which is proactively monitoring the many facets of the modern application environment in order to identify and mitigate issues before they become major problems.

With the right approach to application performance management, businesses are able to determine whether their IT environment meets internal performance standards, identify bugs and potential issues, and enable better user experiences via close monitoring of IT resources.

The best APM solutions should help you:

- Reduce mean time to resolution (MTTR)
- Identify issues before they impact the end-user
- Provide your IT team with the information they need to connect application performance to business outcomes and end-user experiences

## Key capabilities of an APM solution

As applications have evolved from standalone to client-server to distributed, and ultimately, to cloud-based elastic services, APM has evolved as well. Today, when we refer to APM solutions, we refer to tools that allow you to observe and manage all your application environments, identify the root cause of issues, and quickly resolve them via automated or manual responses.

So, what specific monitoring capabilities can you expect from APM, and what kind of visibility do they provide? Let’s dig into the details.

Most solutions allow you to observe the front-end, back-end, and infrastructure performance of your application.

**Front-end monitoring** shows the performance of your applications through the lens of an end-user. This can take the form of **end-user monitoring**, which helps you analyze performance across both browser and mobile user sessions, or **synthetic monitoring**, which leverages simulated sessions to help diagnose

pre-production issues or to retest performance issues to avoid false-positive alerting.

**Back-end monitoring** shows performance across the various services built into your applications to support the functionality required by your end-users. Back-end monitoring gives visibility ranging from the code-level, across the various languages used to develop your application, to the performance of any middleware or third-party APIs integrated into your application.

Lastly, **infrastructure monitoring** helps you visualize the performance of the infrastructure supporting your applications. This includes monitoring servers, databases, the network, and the various cloud services you may be using.

Traditionally, with APM, you might observe things like:

- Application errors experienced by end-users across different geographical locations
- The physical hardware upon which the application is running
- The virtual machines in which the application is running
- The JVM that is hosting the application environment
- The container (application server or web container) in which the application is running
- The behavior of the application itself
- Supporting infrastructure, such as network communications, databases, caches, external web services, and legacy systems
- The cloud platforms and services being utilized
- Microservices and APIs that have been instrumented

But modern application ecosystems have become extremely distributed and complex, and are usually available to customers via multiple channels, mainly mobile and browser versions. When considering an APM solution, it's important to evaluate whether it provides the ability to observe the performance of your back-end technology and infrastructure along with the front-end experiences of your application.



Once we have captured performance metrics from all of these sources, we then need to interpret and correlate them with respect to the impact on your **business transactions**. A business transaction is a mapping of the digital touchpoints required by an application to perform a specific function. This is where the magic of APM really kicks in. APM vendors employ experts in different technologies so that they can understand, at a deep level, what performance metrics mean in each individual system and then aggregate those metrics into a holistic view of your application and its business transactions.

The next step is to analyze this holistic view of your application performance against what constitutes normalcy. For example, if key business transactions typically respond in less than 4 seconds on Friday mornings at 9 a.m., but they're responding in 8 seconds on this particular Friday morning at 9 a.m., the question is: Why? An APM solution needs to identify the paths through your application for those business transactions, including external dependencies and environmental infrastructure, to determine where they're deviating from the norm. It then needs to bundle all that information into a digestible format and alert you to the problem, so that you can easily identify the root cause of the performance anomaly and respond accordingly. This is where things start to get more advanced.

Although it's helpful to know that an issue is occurring, a single application could have dozens, if not hundreds, of issues occurring simultaneously. How does your team know which problem to solve first? Modern, more advanced APM solutions can help address this issue by correlating application performance with business metrics. This allows organizations to visualize how application performance issues are impacting their business outcomes. This insight can be invaluable, as it allows you to prioritize the issues that will have the biggest impact on the business.

Thus, in a modern sense, we can refine our definition of APM to include the following activities:

- The collection of performance metrics across an entire application environment
- The interpretation of those metrics in the light of your business applications
- The analysis of those metrics against what constitutes normalcy
- The capture of relevant contextual information when abnormalities are detected
- Alerts informing you of abnormal behavior
- Bonus: Visibility into the business impact associated with performance issues

The above capabilities, however, represent only the tip of the iceberg when it comes to APM's true potential. Enter: AIOps.

## Moving from reactive to proactive monitoring

We've established that traditional APM solutions provide leaders with real-time insights needed to take action. But with the ever-rising volume of data in IT ecosystems, many professionals are struggling to find a proactive approach to managing this information overload. While the automation of tasks has helped teams free up some bandwidth for operations and planning, automation alone is no match for today's increasingly complex environments.

What's needed is a strategy focused on reducing the burden of mounting IT Ops responsibilities and surfacing insights that matter most to business leaders, enabling them to take the right actions. So what are forward-thinking IT professionals doing to stay ahead of the curve?

As IT systems continue to grow more complex and distributed, many companies are turning to artificial intelligence for IT operations—AIOps—to help alleviate the increasing resources needed for manual management.

Since AIOps is an emerging technology, its definition is highly fluid, but here's a simple way to think about it. Its core elements consist of machine learning (ML), performance baselining, anomaly detection, automated root cause analysis, and predictive insights. Essentially, an AIOps approach leverages these advances in ML and artificial intelligence (AI) to proactively solve problems that arise in the application environment.

Though relatively new, AIOps is gaining momentum, and for good reason. Using AI to identify potential challenges within the application environment doesn't just help IT professionals get ahead of problems—it helps companies avoid revenue-impacting outages that jeopardize the customer experience and the business. Using AI/ML to leverage data about the application environment can make a massive

difference. Instead of merely ingesting data from every dimension of the application environment, AIOps tools can help IT professionals build a more proactive approach to APM.

We'll talk more about AIOps later, but let's take a quick detour. To understand why AIOps is important, we must first address a commonly asked question: Are there other options besides APM?

## APM alternatives

It may seem obvious to you at this point that APM is an essential component of optimal performance. However, some companies may opt to go a different route. In this section, we'll evaluate some alternatives to a complete APM solution, and discuss their limitations.

First, let's consider how to detect problems. An APM solution alerts you to abnormal application behavior. But if you don't have an APM solution, you have a few options:

- 1. Build synthetic transactions**
- 2. Instrument your environment manually**
- 3. Wait for your users to call customer support (definitely not recommended)**

A synthetic transaction is one that you execute against your application to measure performance. Depending on the complexity of your app, it's not difficult to build a small program that calls a service and validates the response. But what do you do with that program? If it runs on your machine, what happens when you're out of the office? Furthermore, if you do detect a functional or performance issue, what do you do with that information? Do you connect to an email server and send alerts? How do you know if this is a real problem or a normal application slowdown at this particular hour/day of the week? Finally, detecting the problem is one thing, but how do you find the root cause of the problem?



The next option is to manually instrument your application, which involves adding performance monitoring code directly to your app and recording it somewhere, like a database or file system. This approach raises, but fails to address, questions like:

- What parts of my code do I instrument?
- How do I analyze it?
- How do I determine normalcy?
- How do I propagate those problems up to someone to analyze?
- What contextual information is important?

And that's just the start. You'll also need to maintain the performance monitoring code you've introduced into your application. This leads to even more questions: Can you dynamically turn the performance monitoring code on and off to avoid negatively impacting application performance? If you identify additional application metrics you want to capture, do you need to rebuild the app and redeploy it to production? And what if your performance monitoring code has bugs?

There are other technical options, but what we find most often is that companies are alerted to performance problems when their customer service organization receives complaints from users. Obviously, this is a bad idea.

Next, let's consider how we identify the root cause of a performance problem without an APM solution. Most often, we see companies do one of two things:

- 1. Review runtime logs**
- 2. Attempt to reproduce the problem in a development or test environment**

Log files are great sources of information, and many times they can identify functional defects in your application by capturing exception stack traces. But when experiencing performance issues that do not raise exceptions, they typically only introduce additional confusion. You may have heard of, or been directly involved in, a production

war room. These war rooms are characterized by finger-pointing and attempts to indemnify one's own components so that the pressure to resolve the issue falls on someone else. The bottom line: These meetings cause unnecessary stress and anxiety. Neither fun nor productive.

Alternatively, and usually in parallel, the development team is tasked with reproducing the problem in a test environment. The challenge here is that you usually don't have enough context for these attempts to be fruitful. Furthermore, if you're able to reproduce the problem in a test environment, that's only the first step—you now need to identify the root cause of the problem and resolve it!

To summarize, yes, there are alternatives to APM, but they may not yield the results you are looking for. APM is important because it enables you to understand the behavior of your application, detect problems before your users are impacted, and rapidly resolve those issues. In business terms, APM matters because it reduces the mean time to resolution (MTTR), allowing you to resolve performance issues faster and more efficiently while limiting the impact on your bottom line.

But performance issues can only be solved as quickly as they're found—which brings us back to AIOps.

## The Future of APM: AIOps Ecosystems

The APM space is undergoing rapid change. Old-school APM software that focuses solely on manual resolution of issues is on the way out. On the other hand, Modern APM solutions focus on your application behavior. They place far greater importance on the performance and availability of business transactions than on the underlying systems that support them.

A modern APM solution uses machine learning to automate anomaly detection and dramatically reduce mean time to resolution (MTTR) for instant root-cause analysis. It correlates software and business performance in seconds, and takes action on performance problems in real-time. In short, modern APM provides an AIOps ecosystem—a comprehensive framework that companies can use to integrate complementary technologies with their APM platform.

A prime use case of the AIOps ecosystem is the ability to remediate problems automatically. For example, if your application is running in a cloud-based environment and your application has been architected in an elastic manner, you can automate

the addition of servers to your infrastructure under certain conditions that may be causing a performance anomaly. By intelligently automating this remediation, the performance is restored to normal without any human intervention.

This ecosystem benefits the APM customer by providing extended cross-domain visibility, business insights, and automation across a broad ecosystem of technologies. Benefits include:

- Visibility across the entire IT landscape throughout the customer journey—from infrastructure through application to the end-user experience. This highlights any potential impact to users so remediation action can be taken.
- AIOps-driven cross-domain capabilities that continually evolve insights to detect anomalies, automate root cause analysis, and predict future performance.
- IT automation for the best user experience, enabling them to rapidly fix performance issues or even proactively optimize cloud infrastructure resources.



## Why invest in APM?

By this point in the guide, you should have a good grasp of the overall importance of APM. However, as you move forward in your APM journey, you'll likely need to answer the "Why do we need APM?" question in a way that convinces your boss, CIO, CTO, or CFO.

The return on investment (ROI) of an APM tool can be demonstrated in many different ways, but to win over your C-level, you should focus on two main areas: operational expenses and preventing revenue losses. By combining cost savings across these two areas, you can not only justify the investment but build a strong sense of urgency across stakeholders.

### Operational Costs

Whenever your application experiences a problem or an outage, there is a cost associated with resolving it. This refers to the support activities required to resolve the issue—or the cost per ticket—and accounts for the time and energy spent finding and resolving the root cause of an issue. Keep in mind that when the root cause can't be easily determined, it often results in a war room scenario as discussed earlier, which drastically increases the number of people working on the issue—thus increasing the costs.

As IT teams are often asked to do more with less, a solid APM solution can help quickly drive efficiencies across the way your teams work, leading to monetary benefits the C-level can easily get on board with.

### Revenue and Reputation Damages

End-user expectations have never been higher when it comes to the applications they rely on. The evidence is clear: If your application is not performing properly or is down completely, and a customer can't complete their transaction, they tend to take their business elsewhere and are likely not to come back. If, on the more extreme side, your application has frequent performance issues,

this could start defining your company's reputation and even its failure.

It may be hard to put a dollar amount on the detrimental impacts to your business, but most C-level executives fully understand that damaging a customer's impressions of your company tends to lead to a loss in confidence and sales.

A quality APM solution can help you mitigate the costs and risks associated with suboptimal application performance, making it easier for you to build the case on why to invest. Additionally, [APM value assessment calculators](#) are widely available, and APM vendors are more than happy to help you calculate the return on investment (ROI) for your specific application and use case.

## Buy versus build

Now that you can justify the investment, you're faced with a choice: Do you evaluate modern, AIOps-focused APM solutions and choose the one that best fits your needs, or do you try to build your own? To make this buy-versus-build decision, ask yourself two questions:

- 1. What is your core business?**
- 2. Does it make financial sense to build your own solution?**

If your core business is selling widgets, it probably doesn't make a lot of sense to build your own application performance management system. Conversely, if your core business is building technology infrastructure and middleware for your clients, it might make sense...but consider where your expertise lies. If you're a rock star at building an eCommerce site but lack the years of technological experience that APM vendors have in interpreting performance metrics, you run the risk of leaving your domain of expertise and building a subpar APM solution—one that doesn't get the job done and only creates more headaches for you.

The next question: Does it make financial sense to build? This depends on the complexity of your applications and how downtime or performance problems affect your business. If your applications leverage a lot of different technologies (e.g. Java, .NET, PHP, web services, databases, NoSQL data stores, and so on), developing performance management code for each of these environments will be a herculean task. But if you have a simple servlet that calls a database, the effort will be less complex. Generally, the more complex your application, the more complex it will be to build your own solution.

Also, while considering if a build makes financial sense, don't forget to properly quantify the long-term investment. A build might seem like a "cheaper" path to APM, but it tends to come with numerous hidden costs associated with long-term maintenance. Additionally, as your technologies and personnel change, it becomes increasingly difficult to keep in-house APM solutions in working order.

Finally, ask yourself about the impact of downtime or performance issues on your business, as discussed in the previous section. If you have a complex environment and performance issues or downtime are impacting your business, you're far better off buying an AIOps-driven APM solution—one that allows you to focus on your core business and not on building an APM system to support your business.

## Conclusion

Application performance management involves measuring the performance of your applications, capturing performance metrics from the individual systems that support your applications and then correlating them into a holistic view. An APM solution works by observing your application to determine normalcy; then, when it detects abnormal behavior, capturing contextual information about the anomaly and notifying you of the problem.

And ideally, an APM solution uses machine learning to automate anomaly detection and reduce MTTR for instant root-cause analysis, correlating software and business performance in seconds to take action on performance problems in real-time. It should also provide an AIOps ecosystem—a comprehensive framework that companies can use to integrate complementary technologies with their APM platform. If application performance or downtime issues can negatively affect your business, it's in your best interest to evaluate APM solutions for these capabilities and choose the best fit for your environment.

Try it free at  
[appdynamics.com](https://appdynamics.com)

Create better digital experiences  
with the #1 fastest-growing APM  
solution. Try a free trial today.

→ [appdynamics.com](https://appdynamics.com)