An APM solution tailored for the modern software-defined business

The Application Intelligence Platform and five architectural innovations that redefine Application Performance Monitoring

Organizations of all sizes have struggled to deploy monitoring and analytics solutions that can keep up with the rapid pace of change and fundamental architectural shifts in their application stack. Mobile, service-orientation, massive data sets, virtualization/containerization/cloud, and the increasing adoption of continuous deployment methodologies have rendered traditional approaches obsolete.

Traditional Application Performance Management (APM) solutions have been very complex and unable to keep up with the complexity of agile development and highly complex architectures and environments. To address these fundamental changes in application architectures and shifts in operational practices, AppDynamics has created the next-generation APM solution.

Prospects and customers evaluating APM solutions have questions about the data collection, processing and analytics capabilities of our platform. Specifically:

– What makes our approach so different, and how does it transform the way modern, software-defined enterprises implement and leverage APM?

– What makes our solution so easy to deploy out of the box, with minimal configuration?

– How do we enable customer to so quickly get results — often in hours?

– How do we capture so much detailed information for production applications without impacting application performance and response times?

– How does our solution scale to handle large deployments?

In this note, we will highlight five core architectural principles of the AppDynamics Application Intelligence Platform that make it possible to rapidly and flexibly deploy in the most demanding production environments, quickly gain visibility into all user interactions and application flows, and provide real-time business impact and operational insights.

## 1. Monitor what matters to your customers and their end-users

Many legacy APM solutions focus on bottom-up health metrics monitoring for discrete application components such as the web server, application server, databases, servers, etc. This bottom-up approach to monitoring ignores the single most important measure to ensure great application performance — the customer (or end-user) experience of the application. Application users and customers don't complain that the server or database connection is slow. Instead, users say they cannot log into the application or notice that it takes too long to complete a checkout transaction. This is why AppDynamics focuses on and uses business transactions to identify and troubleshoot real-world problems in production.

This business transaction-centric approach focuses on how your users are experiencing the site. At a very high level, business transactions are unique user interactions with the application that have a business impact. For example, business transactions include transferring funds in a financial application, booking a flight in a travel application, or checking the status of an order in an eCommerce application. Monitoring business transactions allows you to know if the functionality is currently available (users can log in, check out, view their data, etc), whether user requests are being responded to in a timely manner, and where the cause of the problem is when identified. Organizing application traffic into business transactions aligns the requests with the primary functions of the software applications — engaging and satisfying the user.
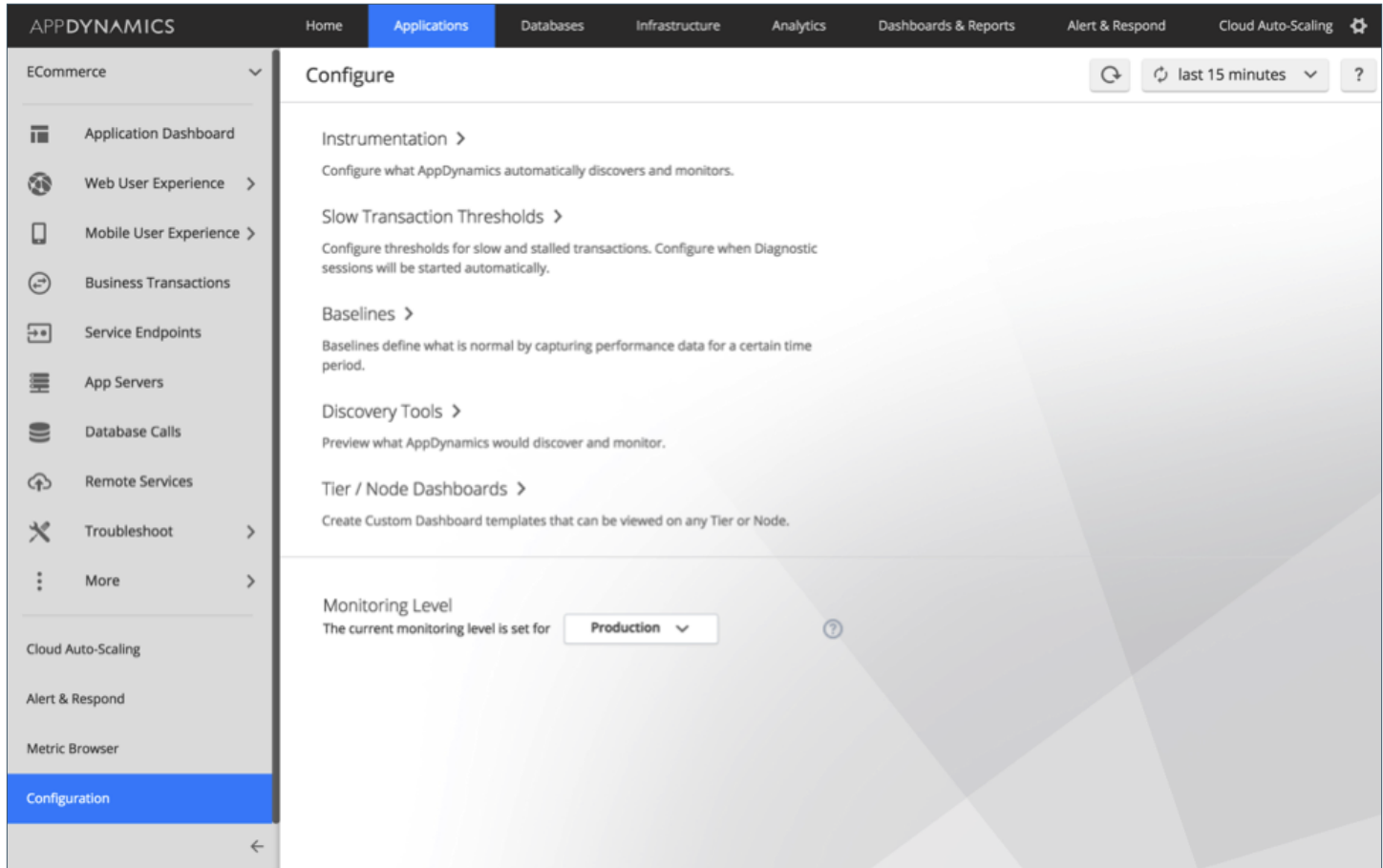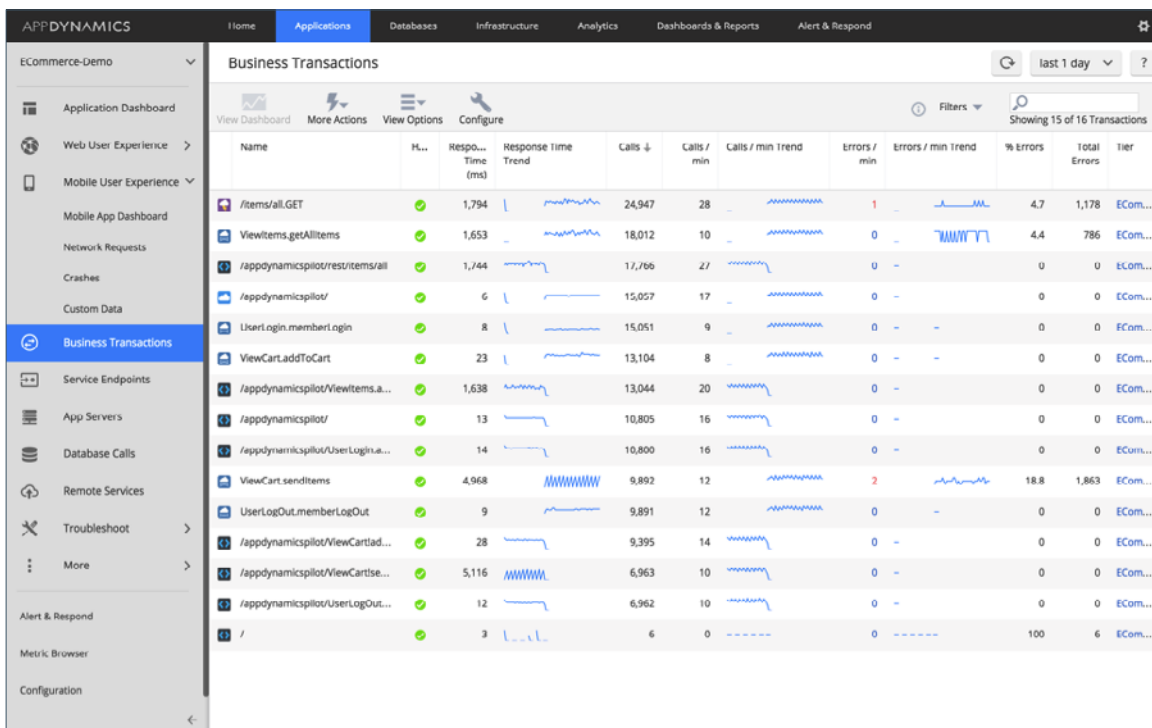
By default, AppDynamics automatically discovers and detects the business transactions in your distributed application using its proprietary tag and follow technology, which requires no configuration. Out of the box, AppDynamics can also detect transaction flows between applications. Once discovered, AppDynamics constantly monitors every aspect of every business transaction being executed in the application including response times, errors, stalls, etc. Also, as the application changes, this definition of a business transaction automatically changes to adapt to your application structure and code.

> *"Business transactions gave the ING team an entirely new perspective on how to view requests across their distributed system."*
>
> **ING** 🦁
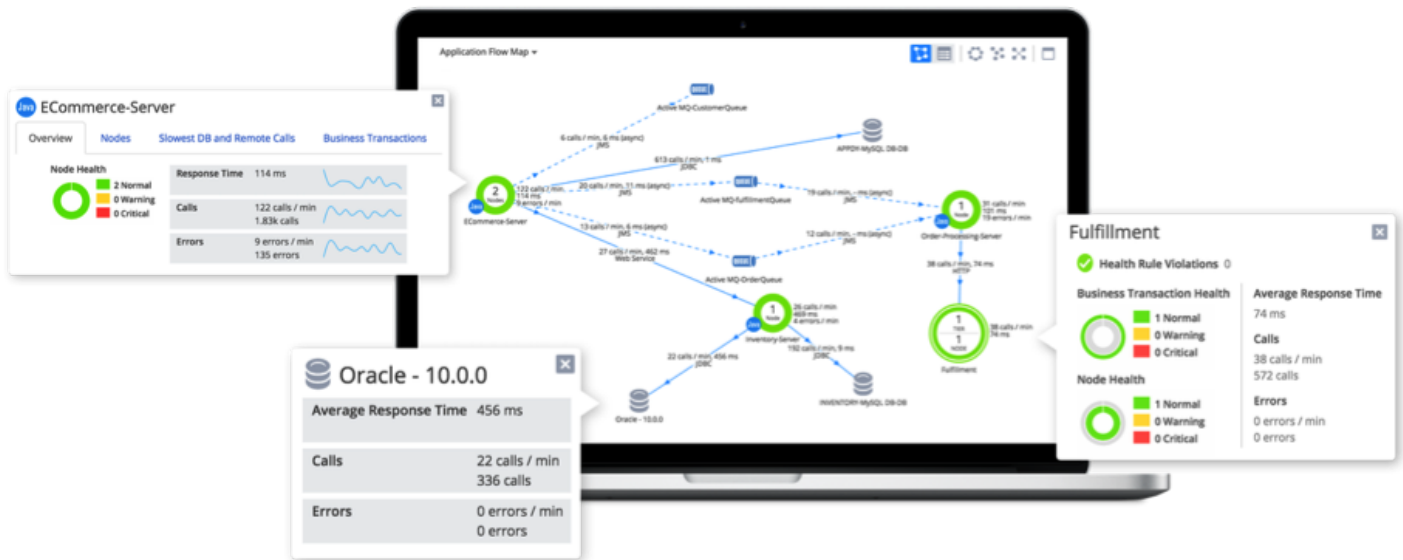
2. Configure the AppDynamics solution with minimal effort

At the core of the AppDynamics platform are collection agents that enable real-time events, metadata, and metric collection from applications and their associated infrastructure components. AppDynamics provides agents to monitor a wide range of user, application, infrastructure platforms, and technologies, such as Java, .NET, SQL and NoSQL databases, Android and iOS applications, web browsers, PHP, Node.js, C/C++, Python, AWS and Azure public cloud infrastructures and servers.

Unlike other APM solutions, AppDynamics' agents are self-configuring; they require minimal effort to monitor the application, infrastructure, and its code (method, classes etc.). Once deployed, the agents automatically identify the application framework (for example, in the case of Java — Struts or Spring); the agents then leverage the business transactions flow to identify the right methods to monitor and automatically collect the metrics from the method. As the application and the business transactions change, the agents automatically adapt by monitoring the transaction and application code without any manual intervention.

"It was simple to deploy. It took minutes, literally. In our proof of concept, we saw results within an hour."

Nasdaq

3. Intelligently identify normal and anomalies with dynamic baselines

Most APM solutions require IT operations to configure baselines and thresholds to generate alerts when the application performance is not up to par. However, in a complex, distributed application with many business transactions and supporting components, understanding the normal and abnormal performance is quite hard. In such environments, it is difficult to:

– Determine the values or ranges that are normal for a particular metric. The normal value of a metric can also vary depending on the time of the day, week, or month. As an example, for a financial trading application, the performance at the first hour of trading is different from the hour after the trading has closed; these metrics will also be very different from the metrics during the weekends.

– Set meaningful thresholds on which to base and receive relevant alerts. In the example above, it is clear that the transaction thresholds have to be dynamic, depending on the hour of the day.

– Determine what is a normal metric when the application or business transaction undergoes change.

For these reasons, the AppDynamics platform performs anomaly detection based on dynamic baselines or thresholds.
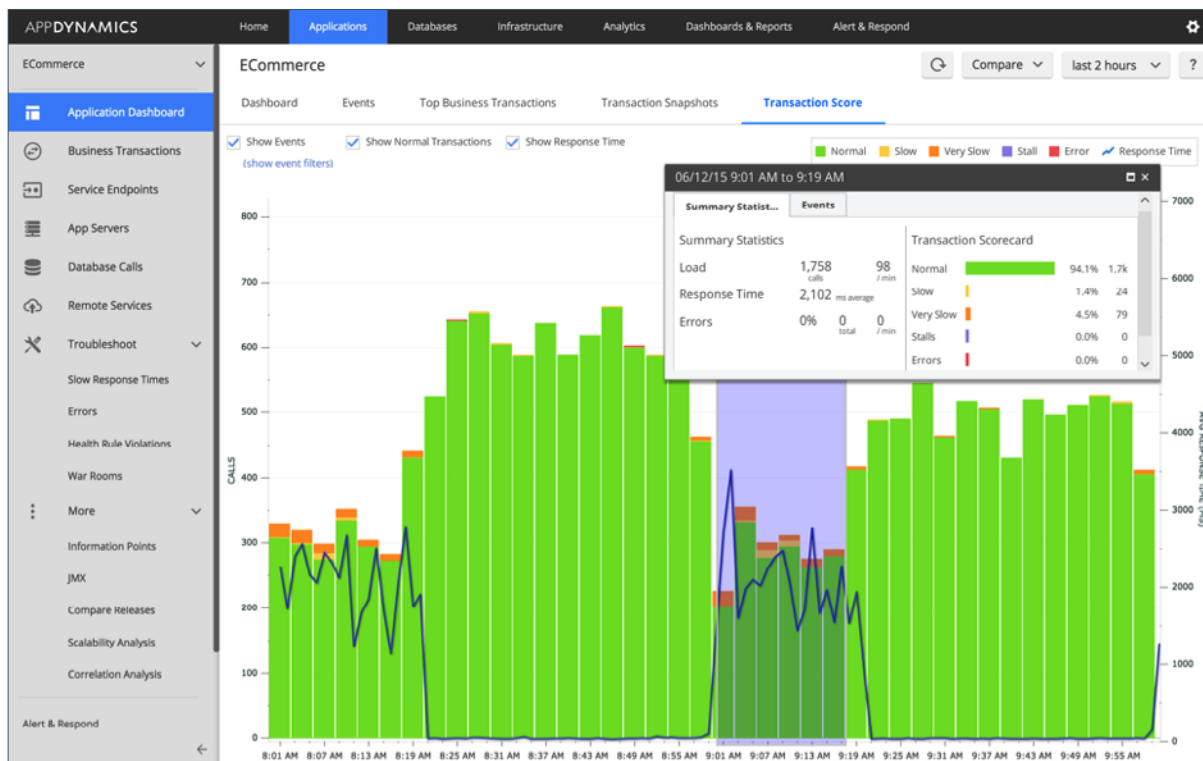
The AppDynamics platform automatically calculates dynamic baselines for business transactions and all application and infrastructure components it monitors. Once implemented, the platform collects and defines what is "normal" for each metric based on actual usage. Then, the platform uses these baselines to identify subsequent metrics whose values fall out of this normal range.

In this new approach, AppDynamics monitors every single business transaction that flows through your applications. We automatically learn and develop a dynamic baseline for end-to-end response time as well as the response time and load of every entry and exit call along a distributed transaction flow, and also for all critical business metrics within your application.

4. Present all the detailed troubleshooting data when needed

AppDynamics scores each transaction by comparing the actual response time to the self-learned baseline. When we determine that the performance of a business transaction has deviated too far from normal behavior (using a tunable algorithm), our agent knows to automatically collect detailed data (i.e. call stack details) to identify the root cause of the problem. This analytics-based methodology allows AppDynamics to detect and ensure that it consumes minimal overhead as it monitors the application.

Unlike AppDynamics, legacy APM solutions use a "capture and store everything" approach to collect all call graph and component metrics all the time. This approach consumes lots of overhead to collect, report, and store irrelevant information, consuming valuable CPU, memory, and network resources. Additionally, it is more complex and expensive to deploy, as the scalability and storage requirements of the centralized management system are affected significantly.

In contrast, there are multiple benefits to the intelligent approach followed by AppDynamics:

– Any CPU or memory resources consumed by the agent reduces the resources available to the business application to support its mission-critical functions. Monitoring solutions should not become the bottleneck. AppDynamics is architected to adhere to this cardinal rule of management software.

– Most APM tools have built-in overhead limiters to keep themselves from introducing too much overhead within a running application. So when legacy APM solutions are collecting deep dive data all the time, they are inducing the maximum allowed overhead all the time (assuming reasonable load). The problem with this approach is that as the application load gets higher and when the performance problems are most likely to surface, the legacy APM solution overhead skyrockets (due to massive amounts of code execution and deep collection being always on), triggering the overhead limiters and reducing the amount of data being collected. In plain English, this means that legacy always-on APM tools that collect all data all the time will provide you with the least amount of data at the time when you need the most data.

– Another critical benefit of the get-details-only-when-needed approach is that AppDynamics dramatically reduces the information overload on IT teams. Operations teams are already busy maintaining and managing applications and systems. They are looking for a solution that collect the appropriate data and alerts them as needed. By separating the real issues from noise (the unnecessary call-stack details for good transactions), AppDynamics makes it easier for ops to spot outliers and focus on the issues that truly impact application performance.

Of course, there are times when deep data capture of every transaction is advantageous—such as during development—and the AppDynamics APM solution has another intelligent feature to address this need. AppDynamics offers a simple, one-click button to enable full data recording system-wide. Developer Mode is ideal for pre-production environments when engineers are profiling and load-testing the application. Developer Mode will capture a transaction snapshot for every single request.

5. Distribute intelligence for massive scalability

All of the intelligent data collection mentioned above requires a very small amount of extra processing to determine when to go deep and what to save. This is a place where the implementation details make a difference.

At AppDynamics, we put the intelligence where it's suited best – at the agent level. It's a simple paradigm shift that distributes the workload across your install base (where it's not even noticed) rather than concentrating on a single point. The AppDynamics agents have several capabilities to intelligently manage their workload requirements:

– Reduce server resource consumption by the agent through intelligently stepping back if the application system is under heavy or critical load

– Reduce storage consumption by collecting only diagnostic data when necessary via intelligent evaluations of transaction performance

– Reduce network communications by performing per minute burst transmissions of transaction metrics and data

The legacy APM solutions use a different approach to data collection and processing. In these solutions, all the data processing is done at central monitoring server(s), and this requires a resource-intensive, complex architecture to support the additional collection & processing needs. To scale such deployments, for example, the architecture shown below is commonly used by legacy APM vendors to support large deployments. This architecture requires the use of additional collectors and data stores to collect and process data. TCO (total cost of ownership).

The challenges with this complex architecture approach are several fold:

– With additional collector servers, the TCO of the solution increases significantly

– The management cost — setup, upgrade, maintenance, security etc. — of the architecture is also high

– The enterprise also needs to hire and train resources to manage this complex deployment

**AppDynamics
Scalable Architecture**

**Traditional APM Vendors:
Non-Scalable & High Maintenance Architecture**
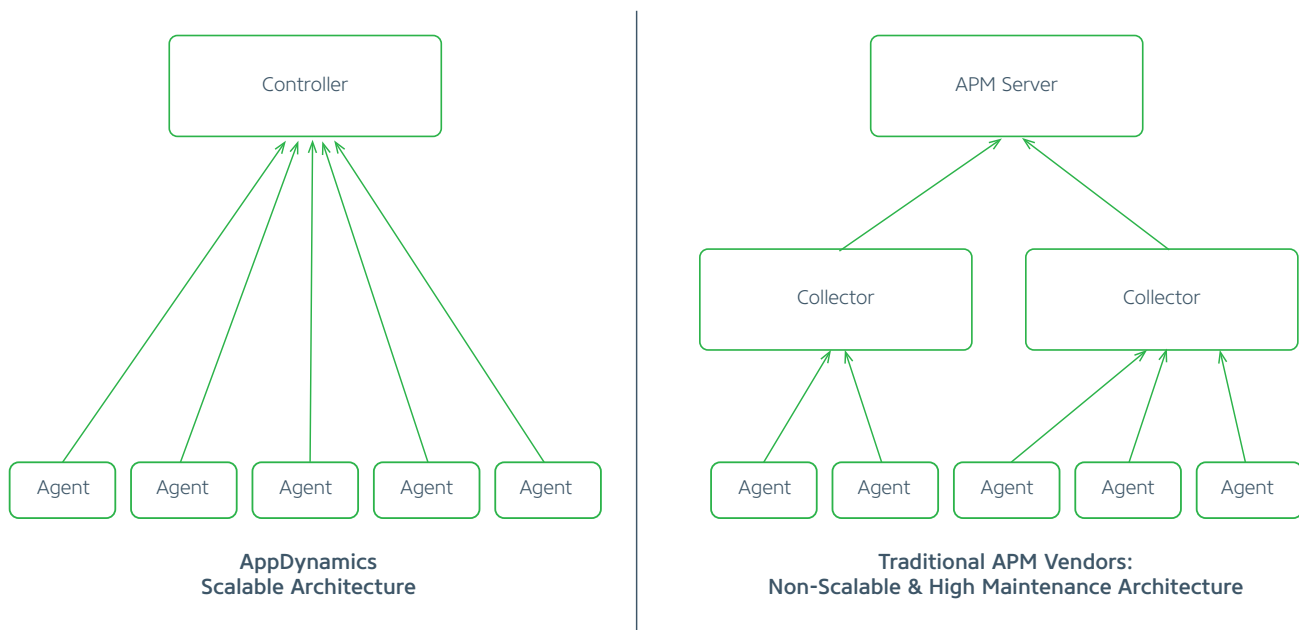
"AppDynamics was rolled out globally across 23 data centers — deploying a total of 15,000 agents in just one week — all to a single AppDynamics report server."

CISCO™

Summary

AppDynamics was architected to manage the performance of modern distributed applications. The solution was built with the notion that monitoring solutions should be easy to install and configure, and quickly deliver value. The solution intelligently manages what metrics to collect, how to collect, and when to collect. Lastly and most importantly, AppDynamics monitoring solution is focused on making the job of the operator easy and better — allowing them to prioritize and focus on what is important to the business. And the solution has provided exactly that value to thousands of customers. To learn more about the AppDynamics solution, please visit www.appdynamics.com

Try it FREE at appdynamics.com