

# APM Experts

---

*White Paper:*

## Application Response Time for Virtual Operations

**Bernd Harzog**

Analyst - Virtualization and Cloud Performance Management

August 2013

© 2013 The APM Experts, LLC. All Rights Reserved.  
All other marks are property of their respective owners.

---

### Abstract

Massive changes are occurring to how applications are built and how they are deployed and run. The benefits of these changes are dramatically increased responsiveness to the business (business agility), increased operational flexibility, and reduced operating costs.

The environments onto which these applications are deployed are also undergoing a fundamental change. Virtualized environments offer increased operational agility which translates into a more responsive IT Operations organization. Cloud Computing offers applications owners a complete out-sourced alternative to internal data center execution environments. IT organizations are in turn responding to public cloud with IT as a Service (IaaS) initiatives.

For applications running in virtualized, distributed and shared environments, it will no longer work to infer the “performance” of an application by looking at various resource utilization statistics. Rather it will become essential to define application performance as response time - and to directly measure the response time and throughput of every application in production. This paper makes the case for how application performance management for virtualized and cloud based environments needs to be modernized to suit these new environments.

---

# Table of Contents

<b>I.</b>	<b>Introduction – Application Operations</b> .....	<b>1</b>
<b>II.</b>	<b>The New Enterprise Application Operations Environment</b> .....	<b>1</b>
	Proliferation of Applications .....	2
	Agile Development and “DevOps” .....	3
	Application Operations (“AppOps”).....	3
	Mobile Devices .....	3
	Lower Cost and Open Source Platforms .....	3
	Scaled Out (not Up) Deployment Models .....	4
	Proliferation of Compound Applications .....	4
	Collapsed and Centralized Applications Infrastructure.....	4
	Business Demand for Service Level Management .....	4
	Density Based Interactions.....	5
	Dynamic Operations .....	5
	Private and Hybrid Clouds.....	5
	Deployment in Public Clouds.....	6
	Dynamic Application Topologies .....	6
<b>III.</b>	<b>Criteria for Virtualization and Cloud Aware APM</b> .....	<b>7</b>
	Broad Application Platform Support .....	7
	Dynamic Discovery of Applications and their Topology .....	7
	Real-Time and Comprehensive Data Collection .....	7
	Zero Initial and Ongoing Configuration.....	8
	Continuous, End-to-End Measurement of Application Response Time .....	8
	Cross Virtualization Platform Support .....	9
	Public Cloud Ready .....	9
	Automatic and Dynamic Baselineing.....	9
	Root Cause Analysis .....	10
<b>IV.</b>	<b>ROI from Application Performance Management</b> .....	<b>10</b>
<b>V.</b>	<b>Summary and Conclusions</b> .....	<b>12</b>
<b>VI.</b>	<b>About AppEnsure</b> .....	<b>13</b>
<b>VII.</b>	<b>About APM Experts</b> .....	<b>14</b>

## I. Introduction - Application Operations

Application Performance Management (APM) have existed in their modern form since 1996 when Wily founded the business of monitoring custom developed Java applications in production. Today many people still think of APM as just a solution for these custom developed applications. It is true that Agile Development and DevOps have fueled the need for more monitoring by APM solutions of custom developed applications, but these solutions only focus upon one part of the problem - issues with the code in production.

Traditional developer focused APM tools (let's call these tools DevOps focused tools), focus just upon the code and just upon helping the developer supporting custom code in production find issues with that code.

There is a requirement for a new and different set of APM solutions that is not limited to just code problems and developers as the audience. This requirement is for APM solutions that work for every application and that help the Operations team support every application in production.

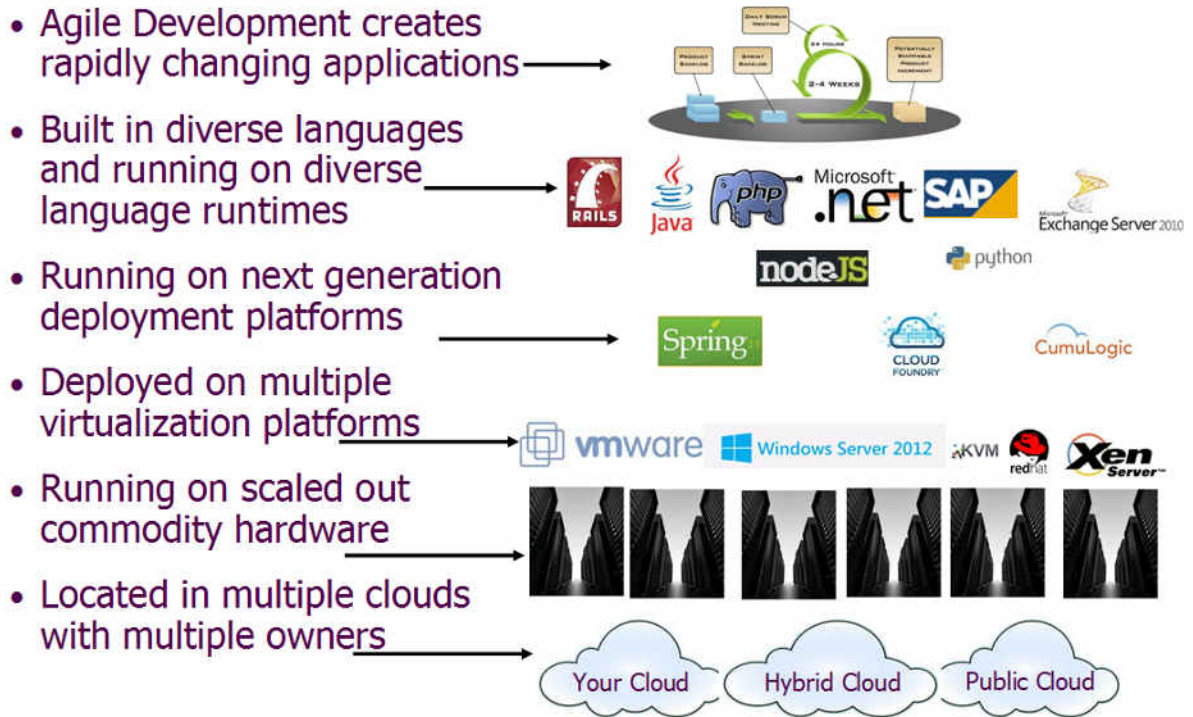
This new category of APM solutions is called the Application Operations or AppOps category of APM solutions and it focused upon the Operations staff who is responsible for supporting every application (physical, virtual or cloud), running across every operating system (Windows or Linux), and every source of application (purchased or custom developed). The key attributes of an AppOps focused APM solution is that it be able to measure the performance of every application in production, and that it be able to find problems in the infrastructure that the Operations team can resolve to help the applications perform up to the requirements.

## II. The New Enterprise Application Operations Environment

First generation APM solutions were built around a set of assumptions that are in many cases no longer true today. These solutions assumed that the application was going to get built or bought, then run inside the firewalls of the enterprise data center. They assumed that applications were going to get built in Java or .NET, which were for a while the dominant development environments used by developers. They assumed that the average application would only get enhanced once or at most twice a year. Finally, many first generation APM solutions completely ignore the fact that in most enterprises, 80% of the applications are purchased commercial applications and not custom developed by the enterprise themselves.

In fact all of the above assumptions are being invalidated in modern enterprise environments. The dynamics listed below are combining to create an entirely new and different enterprise

computing environment which must be addressed by entirely new application performance management tools. This new enterprise computing environment is depicted in the diagram below, and described in detail in the following sections.



### Proliferation of Applications

The unlimited demand for new application functionality and an ever-growing backlog of unaddressed application development and enhancement requests has resulted in two sets of pressures upon enterprises of all sizes. The first is to quickly procure and get into production new purchased applications - and to quickly update these applications in production as the vendor provides new releases and updates. The second is to deliver custom developed applications into the environment more quickly and update these more quickly as well. This demand for new application functionality has led to an astounding proliferation of applications within enterprises worldwide. Many enterprises report that they now have 1500 to 2000 applications that they consider “business critical.” Business pressures to compete online and with mobile applications will only continue to fuel the trend towards more applications. This proliferation of applications is creating unprecedented pressure upon the Operations teams who have to support these applications, and who then need enhanced visibility into the performance of all of these applications.

## Agile Development and “DevOps”

The unrelenting pressure to deliver more application functionality in less time has given rise to other important trends: Agile Development as a development methodology and “DevOps” as a methodology for managing applications in production.

Agile Development focuses upon having one developer responsible for each component of an application system, and then having those developers work as a self-coordinating team to deliver new functionality into production on regular and short time intervals (every week, two weeks or a month at most). DevOps is about eliminating the walls between application development and production application support - essentially creating one team that builds the application and supports it in production.

## Application Operations (“AppOps”)

When organizations deploy applications in virtualized environments, or in some combination of private, hybrid or public clouds, the IT Operations teams that support these dynamic and distributed environments often find themselves under pressure from application owners and business constituents to “prove” that their virtualized infrastructure or their private cloud is not at fault for real or perceived application performance issues.

The best possible way to deal with the “guilty until proven innocent” problem is to create a team of people responsible for the performance of every application in production. These teams are often called Application Operations (“AppOps”), and they are responsible for the performance (response time) of every application in production. These teams need a completely different class of APM tools (something that works for every application in production) than the Java/.NET tools that focus only on a narrow set of custom developed applications. They need a tool that tells them how their infrastructure is affecting the operation of every application in production.

## Mobile Devices

Since the inception of the Internet, it has been safe to assume that the client of an application was a browser, and that a page-oriented interaction between the browser and the web server took place over HTTP or HTTPS. It is now the case that for many applications, a client application resident on a mobile device is the most popular point of access into the back end application. These mobile “apps” represent the fastest growing access point for most applications.

## Lower Cost and Open Source Platforms

There have been continued improvements in the price and performance of commodity Intel-based servers. There has also been an emergence of lower cost, open source alternative application platforms such as Linux, JBoss Application Server, VMware vFabric, and Apache Tomcat. This means that it is now much less expensive to have large numbers of “smaller”

commodity servers than it is to have a small number of high end servers that maximize CPU count and memory size. As enterprises spend less on the application server tier, this has put the traditional pricing models of the first generation APM vendors under pressure.

### **Scaled Out (not Up) Deployment Models**

Agile Development has led to the modularization of applications and the continuous delivery of new functionality into production. The economics of commodity hardware and open source application platforms have made it inexpensive to scale server farms out - not up. The combination of modularized software and scaled out deployment models means that we now have rapidly changing application systems that run on hundreds, and in some cases thousands, of scaled out servers instead of just a few very large and expensive boxes. This creates another requirement that the first generation of APM tools are not designed to address. They are not able to deal with application systems comprised of hundreds or thousands of servers, nor are they priced and sold in a manner that makes their purchase feasible for this deployment scenario.

### **Proliferation of Compound Applications**

Whereas first generation APM tools did a great job for Java and .NET applications running on a small number of servers deployed inside of an enterprise's network, this is not the world we live in today.

Typically, 80% of the business critical applications that an enterprise relies upon are purchased. Yet, none of these purchased applications are instrumented for performance. It is increasingly the case that custom developed applications "wrap" these purchased applications producing compound applications that are part custom and part purchased. Tools that monitor code are completely unable to provide an end-to-end picture of the performance of these applications.

### **Collapsed and Centralized Applications Infrastructure**

Prior to virtualization, most business critical applications ran in their own resource silos. They ran on their own set of servers, often with dedicated edge networks, and also often with dedicated LUN's and arrays in the storage system. Virtualizing these systems often means that much more of this infrastructure becomes shared. Even if it is only shared with other elements of one application system, the underlying technology that enables the sharing is under the responsibility of IT Operations. Therefore, IT has to take more responsibility for the performance of these applications, and is often guilty until proven innocent.

### **Business Demand for Service Level Management**

When the applications owners and their constituents lose direct control over the resources that support their applications, these teams turn around and require service level assurances from IT Operations. These assurance demands come in two forms: The first is that the shared

infrastructure perform as required to support the required performance of the application. The second is that IT guarantees it will do its part to ensure that the application performs (from a response time perspective) to a level that meets the needs of the business constituents and their end-users. Virtualization therefore makes IT Operations at least partially responsible for the end-user experience associated with these applications.

### Density Based Interactions

Virtualization involves consolidating workloads that were previously on isolated hardware into shared pools of server, network and storage resources. Once this consolidation has been implemented, isolated peaks in resource consumption can cause resource conflicts with numerous other processes. This concentration of load can create new bottlenecks which did not exist before. The owners of the IT infrastructure must now prove that spikes in load and resource consumption on the part of one application, is not adversely affecting the performance of other applications.

### Dynamic Operations

VMware Vmotion, HA, and DRS create the ability for workloads to get moved from server to server automatically, based upon rules and resource thresholds. When an administrator logs onto a physical server, the environment and the demands placed upon the environment are typically well known and may have been established over a relatively long operating history of that one application on that one server. Moving workloads creates new issues with density and sharing of resources, especially since the products that are doing the automatic moving of guests do not really have any kind of end-to-end visibility into performance.

### Private and Hybrid Clouds

As public cloud vendors have made it easy for business constituents to “swipe a credit card and get IT services”, forward thinking IT organizations are also seeking to offer their IT services to their business constituents on a similarly flexible and easy to provision basis. This has given rise to the notion of “IT as a Service” in which business constituents order IT services from service catalogs in the same manner as they order AWS services from Amazon.com.

“IT as a Service” initiatives create the potential for not just dynamic operations for current workloads, but the dynamic instantiation of many new workloads ordered up by users and business constituents. This has profound implications upon both the management of performance for these dynamically created applications and the management of the capacity of the underlying environment. Imagine a situation where business constituents can create 10 or more new applications a day, serving thousands or ten’s of thousands of constituents.

Such a dynamic environment will require a degree of automation and simplicity in APM solutions that represents a radical change to the status quo. It will become necessary for APM

solutions to automatically discover applications as they are instantiated, identify them, and manage their performance automatically without human intervention or configuration.

### Deployment in Public Clouds

When Amazon launched its public cloud offering (EC2), the first users were developers and organizations who wanted to rapidly prototype and test new applications without having to provision internal resources (either themselves or through their IT departments). Since then, several other public cloud infrastructure and platform providers have come into being (IaaS offerings like Microsoft's Azure, VMware's Hybrid Cloud Service, Google's ComputeEngine, and PaaS offerings like Engine Yard and Heroku). This raises another set of issues that first generation APM solutions are completely unequipped to address. The main issue is that first generation APM solutions are designed to assume that the agents that monitor the JVM's in the application servers are residing on the same LAN subnet as the management system for the APM solution, and that the management system for the APM solution can poll the agents for their data.

These architectural assumptions on the part of first generation APM solutions are invalid for applications where all or a part of the application resides in a public cloud. In the case where an application resides in a public cloud, and the monitoring system resides inside the enterprises firewall, it is not possible for the monitoring system to poll the agents in the cloud. To address this use case, a first generation APM solution will need to be redesigned from scratch, around the assumption that the agents are autonomous and that they initiate communications, over public internet protocols like HTTP and HTTPS through commonly open ports like 80 and 443, to the back end management system.

### Dynamic Application Topologies

Given that modern applications will be running in a dynamic, elastic, scaled out, and potentially distributed production environment (across one or more data centers - for example one belonging to the IT organization and one belonging to a public cloud vendor), it will not be possible to manage the performance of these applications without understanding where the application components are running, and what communications flows are occurring between all of these components. The dynamic and elastic nature of this problem means that topology cannot be dealt with via configuration at install time of the APM tool, but that it must be continuously and automatically discovered. In fact, one of the hallmarks of second generation APM systems is the ability to dynamically and continuously construct a topology map of the application system, and include in that map both hop-by-hop and end-to-end throughput and response time information.



### III. Criteria for Virtualization and Cloud Aware APM

#### Broad Application Platform Support

Virtualization of business critical and performance critical applications gives rise to a new requirement for the IT staff that operates and supports the virtualized infrastructure. In the physical world, IT Operations typically supported the hardware and the operating system, but had no responsibility for the actual performance (response time) of the applications. The physical hardware was usually so massively over-provisioned that the risk of infrastructure causing performance issues was very low.

In virtualized environments, previously isolated applications run together on a shared set of infrastructure. The resources used by the applications are abstracted from the underlying hardware by the hypervisor. This puts the IT team in charge of the virtualization environment in the position of being “guilty until proven innocent” when performance issues arise. For this reason, it is critical that IT teams owning virtualized infrastructure that supports business critical and performance critical applications have a tool that measures applications response time across *every* applications system, no matter how that application was developed or procured. It is essential for modern enterprises to have an APM tool that can cover every application they support in production (including all of the purchased applications). If an enterprise has both custom developed and purchased applications, then two tools are likely necessary - one that understands all of the applications in production (used by the operations staff) and one that can find problems with code in production (used by the team supporting custom developed applications in production).

#### Dynamic Discovery of Applications and their Topology

The insatiable demand for business functionality implemented in software means that most enterprises have to cope with a deluge of new incoming applications and a deluge of changes to existing applications. For an APM tool to be viable, it must discover (and name) these applications as they arrive automatically, and then automatically map their topology. Included in this requirement is the ability to keep the topology map up to date in near real time as the application topology changes or the application is scaled out in response to demand.

#### Real-Time and Comprehensive Data Collection

There are hundreds of tools and products that monitor operating systems, servers and networks. However, the vast majority of these tools suffer from two fatal flaws when it comes to helping the operations team support the mix of modern applications in modern enterprise computing environments.

The first flaw is that the traditional infrastructure monitoring tools are blind to the actual applications that are running in the environment. They are blind as to which applications are running and they are blind as to the topology of these applications.

The second flaw is that traditional infrastructure monitoring tools collect just commodity resource utilization data from the standard API's in the operating systems, network devices and storage devices. The problem with this resource utilization data is that in a modern enterprise compute environment (virtualized and cloud based), you cannot infer the performance of application from resource utilization metrics.

Instead credible monitoring solutions must focus upon collecting data in the following manner:

- Real-time data collection. Collecting data every 5 minutes based upon 20 second samples (as is the case with the VMware vSphere API data that commodity vendors rely upon) leaves tremendous gaps in visibility. The modern environment is changing so quickly that real time collection of performance data is necessary in order for the monitoring tool to be able to keep up with the environment.
- Comprehensive data collection. In a rapidly changing environment it is critical not to miss momentary spike or peaks in data. Therefore it is necessary to collect all of the data, and not just samples of the data.
- Deterministic data collection. Any data that comes from a management API like WMI, SNMP, or SMIS is subject to sampling and averaging by the process that collects that data and makes it available via that API. Credible and modern operations focused APM solutions must collect their own data in a manner that ensures the accuracy of the data.

### Zero Initial and Ongoing Configuration

APM solutions in dynamic and cloud based environments need to work out-of-the-box with as close to zero initial configuration as possible, with no ongoing configuration required as applications are either changed or new applications arrive. The environment is simply too dynamic and potentially distributed for approaches that require heavy configuration and customization to work.

### Continuous, End-to-End Measurement of Application Response Time

The single, most important capability of an APM tool is to discover the applications automatically, map their topology automatically, and to measure end-to-end and hop-by-hop response time and throughput across the application topology. If the tool under consideration does not measure response time and throughput, it is not an APM tool and should not be considered for APM use cases.

The technical reason for focusing upon response time and throughput is these are the only really accurate measure of application performance that can be taken for a dynamic system (you cannot infer performance from resource utilization). The business reason is that the constituents and users of the application can relate to the response time (it is what they mean when they say it is slow). Response time is therefore the metric that can be the common language between users, application owners, and owners of the virtual infrastructure.

Response time needs to be measured both end-to-end and hop-by-hop through the entire topology of the application system. This needs to occur automatically, without requiring anyone to configure the APM solution for the topology of the application system.

### **Cross Virtualization Platform Support**

While the vast majority of virtual applications that warrant being managed by an APM solution are run on VMware vSphere today, it is important to consider the benefits of having the flexibility to move applications to another virtualization platform without having to consider another management tool. Almost all of the APM solutions contain no dependencies upon a particular platform like VMware vSphere. However, a few include virtual appliances that use the mirror port on the VMware vSwitch to collect applications response time data. Since these virtual appliances are currently only available on the VMware vSphere and Microsoft Hyper-V platforms, a switch to another platform would entail these products losing some data collection functionality. Solutions that are based upon an agent inside of the operating system have the flexibility to follow that operating system around as it is moved from physical to virtual to public cloud deployment models.

### **Public Cloud Ready**

APM solutions must also be cloud ready in the sense that if the agent monitoring an instance of an application is in a public cloud, it must be able to “phone home” and traverse firewalls to get back to its management system in the enterprise without requiring a VPN or firewall work around.

### **Automatic and Dynamic Baselineing**

In a world of many new and changing applications, there will be no time to manually set thresholds for anything but a few top level metrics like response time. Therefore a part of self- or zero-configuration is the ability of the APM solution to set baselines for underlying resource utilization and load metrics, and for these baselines to automatically change over time as the usage patterns of the application changes.

## Root Cause Analysis

Root cause analysis is one of the most difficult topics in the application performance monitoring industry, and one of the most difficult areas for vendors to get right in their products. There are the following approaches to root cause analysis:

- Inference from resource utilization data. This legacy approach involved creating baselines for lots of resource utilization metrics, and then trying to infer that there was an application problem when one of those metrics went out of bounds. This approach has been proven to be useless in the modern abstracted, shared, dynamic, and distributed data center.
- Statistical correlation of response time, throughput and resource utilization metrics. Most products cannot directly link the response time of an application with the chain of actions in the infrastructure that support that application. Therefore they rely on statistical correlation to say that if response time deviated at this point in time and these resource utilization metrics were out of bounds at the same time, then these resource utilization metrics probably point to a constraint that is the cause of the performance problem. The problem with this method of root cause analysis is that it is prone to either too many false alarms or too many instance of issues that should have been caught but were not (false negatives).
- Rule based approaches. In certain cases when there are known or discovered relationships between applications and their supporting hardware and software infrastructure, topology maps can be used to dramatically reduce the scope of the data that must be analyzed in order to come to a root cause.

## IV. ROI from Application Performance Management

While the table below lists some strong business justifications and returns on the investment from a virtualization-aware APM solution, there is an even simpler justification for these solutions than ROI. That justification is for any organization that is putting a performance critical or revenue generating application in a virtual or cloud base environment, it is essential that this organization ensure the performance of that application. The teams that own the virtual infrastructure will not be allowed to virtualize business critical and performance critical applications unless the performance (response time) of these applications can be assured. Attempting to infer the performance of applications that run in virtualized and cloud based environments will not work. This is why a solution that directly measures the response time of the application is needed. This is especially true for any application that runs in a public cloud, as the cloud vendor cannot or will not provide

infrastructure performance metrics that prove the quality and speed of the infrastructure. Therefore, both virtualized and cloud based applications will require modern, agile, self-configuring solutions that measure response time end-to-end across distributed environments that (in the case of clouds) span organizational boundaries.

Capability	Benefit
Accelerate virtualization ROI and Cloud savings	Drive more hard dollar ROI more quickly
Assure business performance dependent upon business critical applications	Protect Revenue
Deliver Business Services based upon app performance and end user experience	Business Agility (Revenue and Market Share)
Reduce time and effort required to address performance problems	Save Money
Redeploy expensive architects out of support and babysitting roles	More effective use of people in a challenging economy
Prevent business disruptions due to application brownouts and outages	Protect Revenue and IT Reputation
Enhance operational agility	Be more responsive to the business at a lower cost

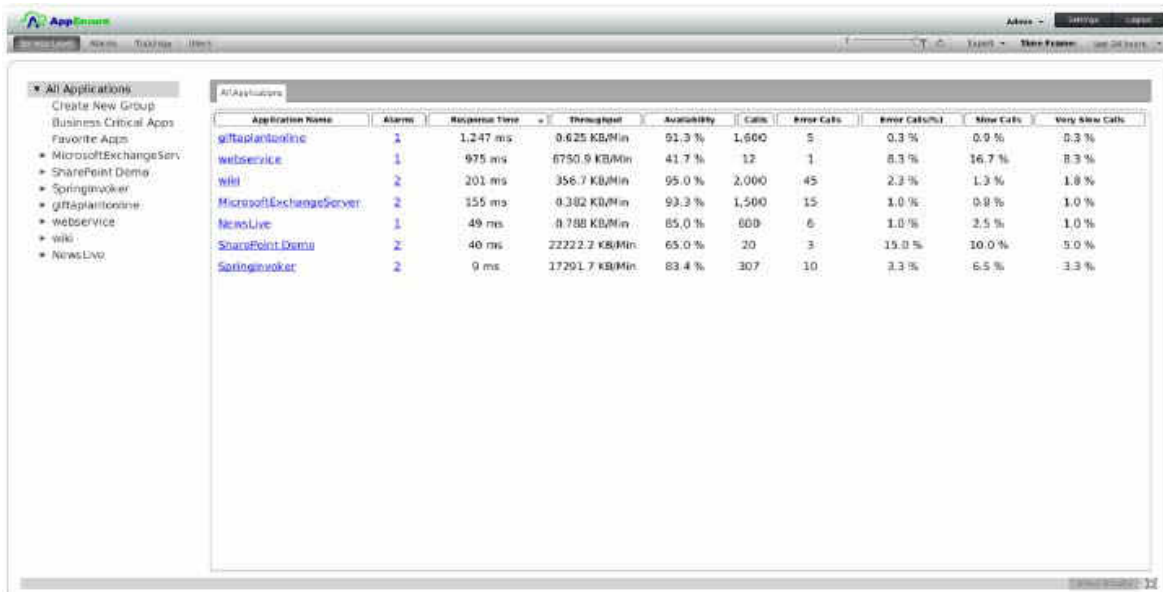
## V. Summary and Conclusions

In summary, monitoring the performance (response time) of a business critical application in virtualized and cloud based environments should be implemented around the following concepts:

- Application performance should be monitored end-to-end and hop-by-hop through the entire sequence of processes, servers, and networks that comprise an applications system.
- In a dynamic (virtual or cloud) environment, a fixed topology for an application cannot be assumed. The topology must be continuously discovered on a fully automatic basis by the APM solution.
- Virtualization breaks the historical methods of measuring application performance and introduces brand new issues that must be addressed by new APM solutions. Clouds run on virtualization, and therefore have all of the problems of managing performance created by virtualization, plus the additional ones associated with having an organizational boundary between the application and the infrastructure.
- Synchronize your applications development and applications monitoring strategies. Agile Development and brittle APM solutions that require constant reconfiguration by hand do not go together.
- Cloud Computing introduces additional challenges like distributed execution of applications components, and an organizational boundary between the application and its environment. APM solutions should be able to travel with an application as it moves between internal and external data centers.
- Moving an application into a virtualized or cloud based environment will increase the demand on the part of your business constituents for both infrastructure performance management and application performance management solutions.
- Moving an application into a virtualized or cloud based environment is the perfect opportunity to kill legacy, complex, expensive “heavy-weight” application management solutions.
- Virtualized and Cloud-aware APM solutions are required in order to address these needs.

## VI. About AppEnsure

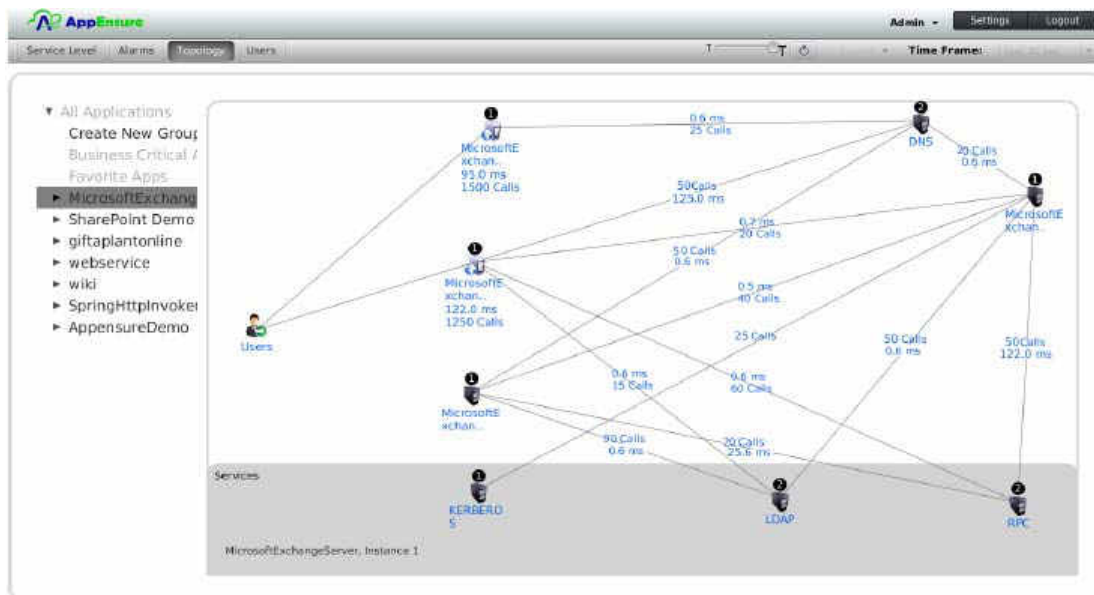
AppEnsure uniquely and automatically discovers every Windows and Linux application in your environment, identifies it by name, and measures the end-to-end and hop-by-hop response time and throughput of every application. This works for applications installed on physical servers, applications installed in virtualized guest operating systems, applications automatically provisioned in private or hybrid clouds, and applications running in public clouds. It also works irrespective of whether the application was custom developed or purchased.



The screenshot shows the AppEnsure application dashboard. On the left, there is a sidebar with 'All Applications' and a tree view of application groups including 'MicrosoftExchangeServ', 'SharePoint Demo', 'SpringInvoker', 'wiki', and 'NewsLive'. The main area displays a table titled 'All Applications' with the following columns: Application Name, Alarms, Response Time, Throughput, Availability, Calls, Error Calls, Error Calls(%), Slow Calls, and Very Slow Calls. The table contains data for several applications:

Application Name	Alarms	Response Time	Throughput	Availability	Calls	Error Calls	Error Calls(%)	Slow Calls	Very Slow Calls
giftaplantonline	1	1,247 ms	0.625 KB/Min	91.3 %	1,600	5	0.3 %	0.0 %	0.3 %
webservice	1	975 ms	6750.9 KB/Min	41.7 %	12	1	8.3 %	16.7 %	8.3 %
wiki	2	201 ms	356.7 KB/Min	95.0 %	2,000	45	2.3 %	1.3 %	1.8 %
MicrosoftExchangeServer	2	155 ms	0.382 KB/Min	93.3 %	1,500	15	1.0 %	0.9 %	1.0 %
NewsLive	1	49 ms	0.788 KB/Min	85.0 %	600	6	1.0 %	2.5 %	1.0 %
SharePoint Demo	2	40 ms	2222.2 KB/Min	65.0 %	20	3	15.0 %	10.0 %	5.0 %
SpringInvoker	2	9 ms	17291.7 KB/Min	83.4 %	307	10	3.3 %	6.5 %	3.3 %

AppEnsure then determines the Topology Map for each application system, automatically identifying how the tiers of the application system are communicating with each other, and how servers in each tier are communicating with their supporting services like LDAP and DNS.



## VII. About APM Experts

Bernd is the CEO of APM Experts, a consulting and analysis firm focusing advising enterprises upon next generation management software strategies, and upon advising vendors upon their product and marketing strategies. Bernd was formerly a Gartner Group Research Director focusing upon the Windows Server operating system, CEO of RTO Software, VP of Products of Netuitive and has been involved in vendor and IT strategy since 1980.