

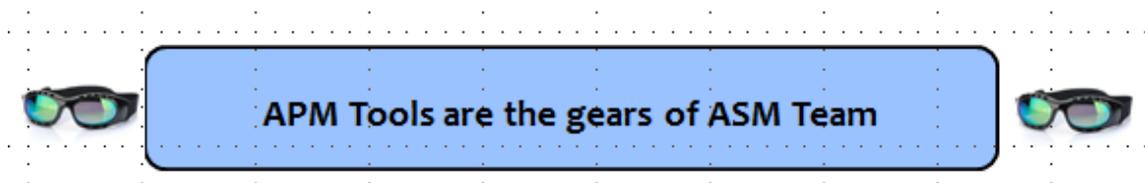
## **APM in Globally Distributed Enterprises**

### **04/01/2014**

**Sameer S. Paradkar**

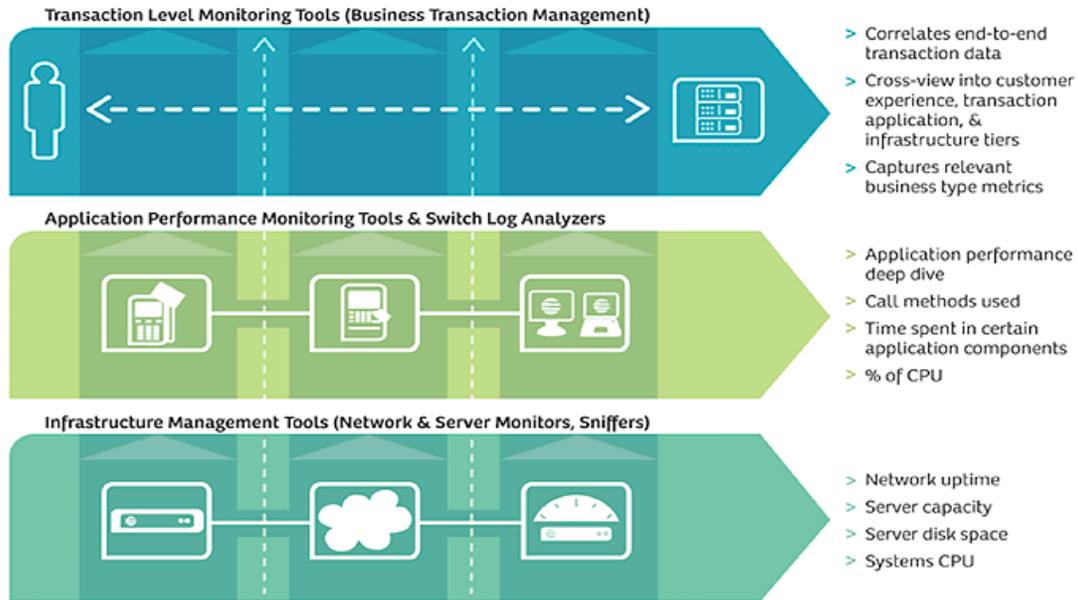
### **Introduction**

Application Performance Management (APM) is the name given to the use of technology to initiate, deploy, monitor, fix, update and optimize systems within an organization. Application management software employs measurements of response times and other components and resource interactions to help manage the overall stability and usability of the software within its purview.



**Figure 1: APM Tools**

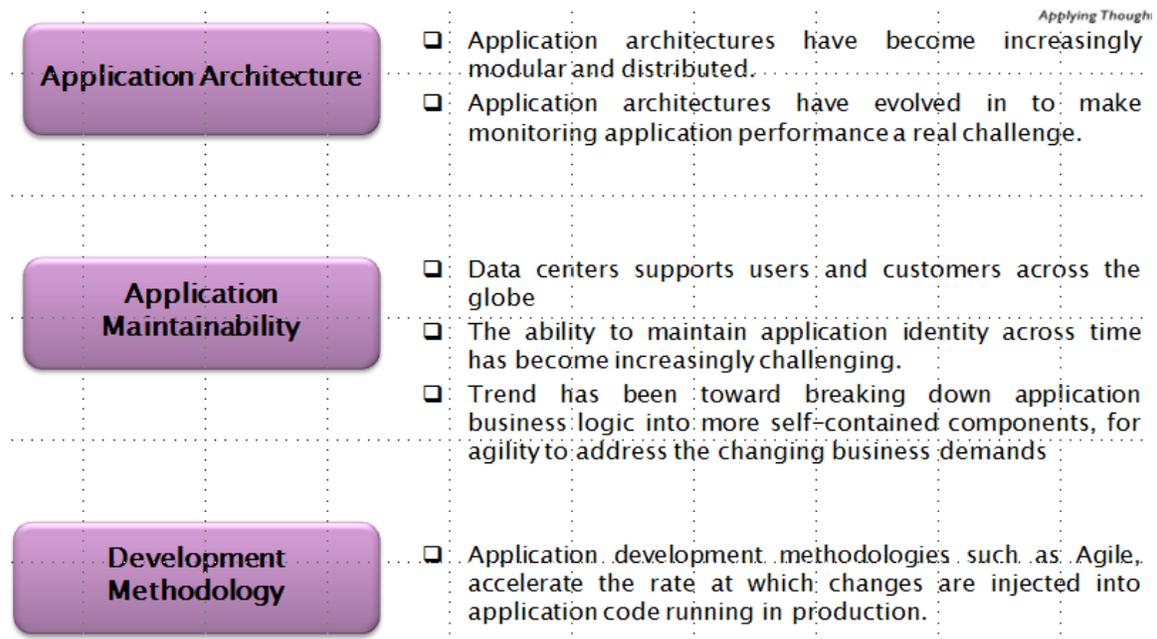
APM is a \$2 billion market for application performance monitoring as predicated by Gartner. Business and C-level executives recognize that IT is not just an infrastructure that supports background workflows, but is a direct generator of revenue and a key enabler of strategy.



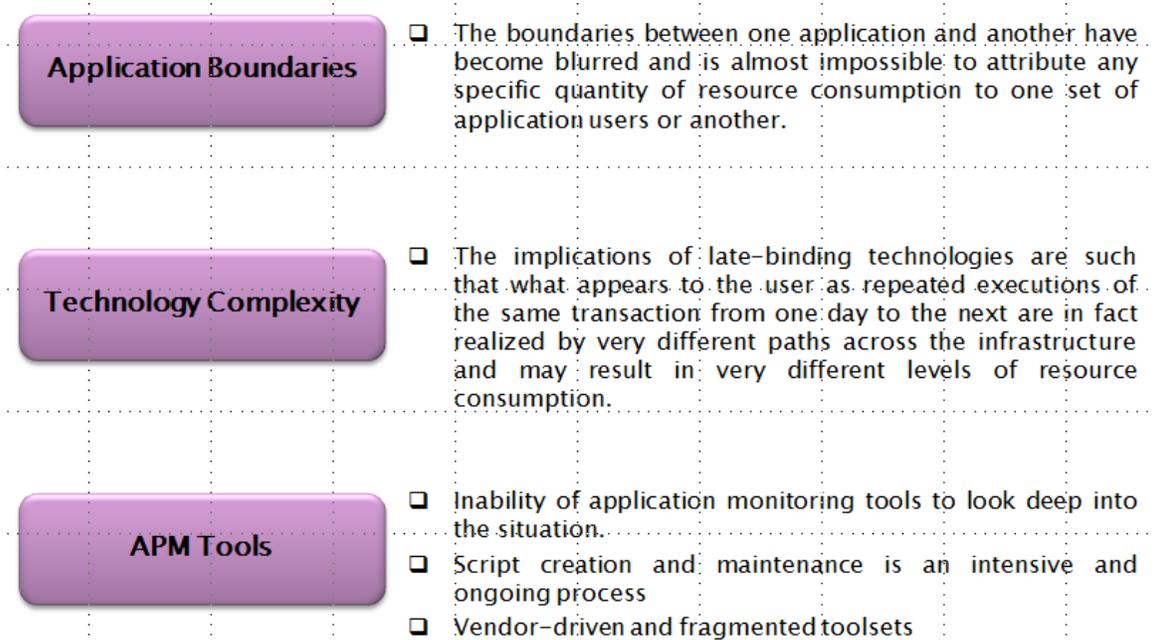
**Figure 2: APM Layers**

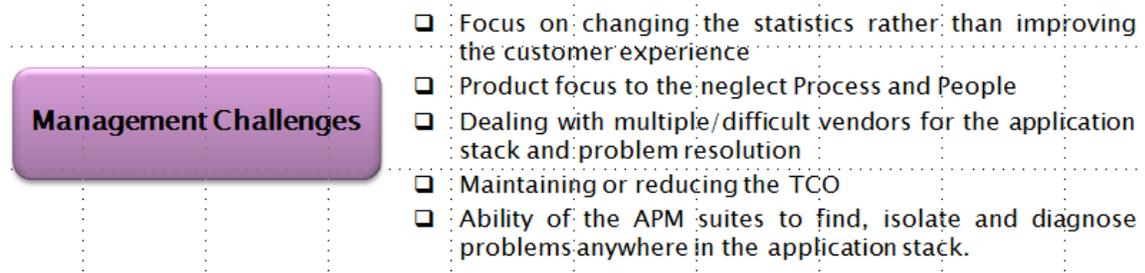
### Challenges faced by the ASM teams today

Applications have become difficult to monitor for a number of reasons: architectures have become more modular, redundant, distributed, dynamic and boundaries are blurred, often the code execution path taking multiple routes during execution. The combined impact of modularity, redundancy, distribution and dynamism reduces the effectiveness of the technologies and techniques that traditionally support the APM landscape. The following diagram lists the typical challenges faced by ASM teams.



**Figure 3: APM Challenges**

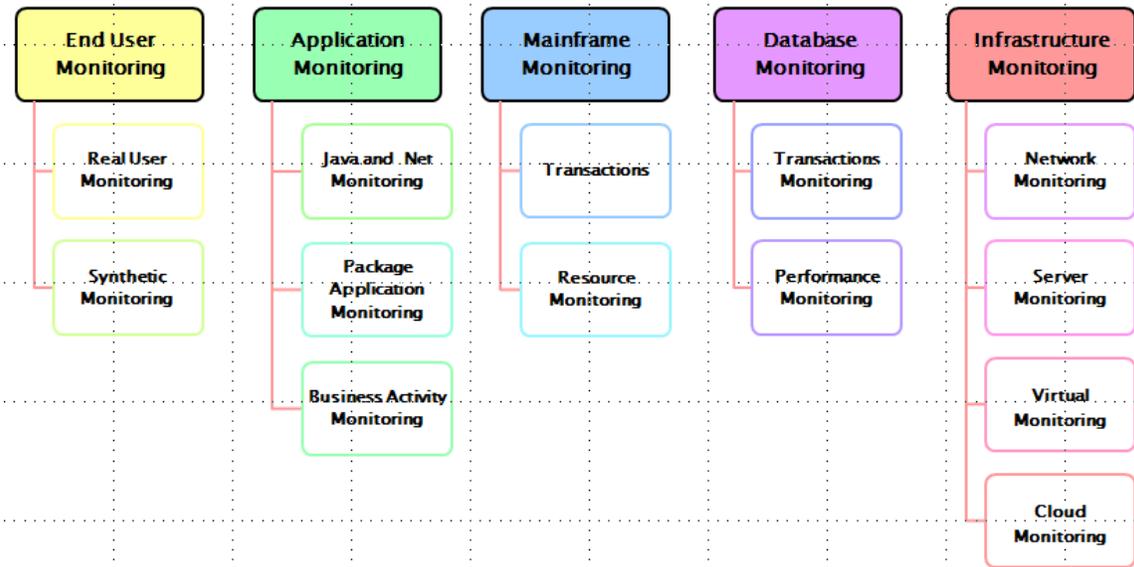




**Figure 4: APM Challenges Cont.**

### Monitoring Landscape

APM is not simply an end-user monitoring application. APM is part of the IT infrastructure and the use of the tools that is distributed among many stakeholders. The following diagram is an APM reference architecture that incorporates best practices, standards and the substance of what we've learned from past engagements. Figure 5 illustrates the different areas of the landscape that are monitored by a combination of various tools. It is important to correlate the information from various sources within the application landscape which is necessary for creating a dashboard with the entire enterprise wide view to provide health and early warning indicators in the application landscape.



**Figure 5: APM Monitoring Landscape**

## Dimensions of functionality

APM tools essentially work around these dimensions to generate meaningful data for the IT support and operations teams. APM technologies are sub-divided into five dimensions of functionality.

- End-User Experience Monitoring
- Discovery, Modeling and Display
- User-Defined Transaction Profiling
- Component Deep-Dive Monitoring
- Application Performance Analytics

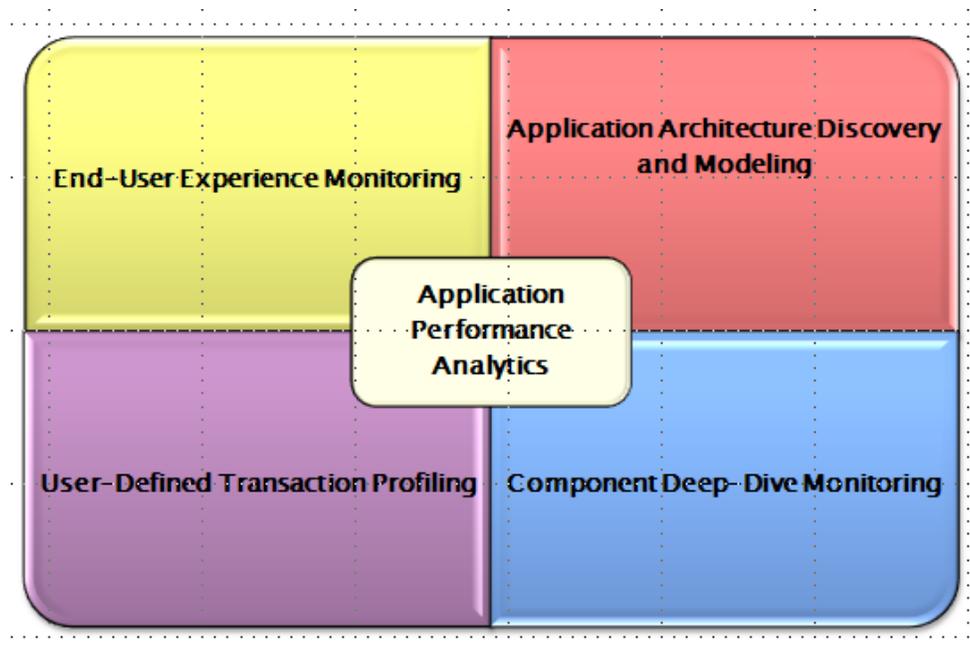


Figure 6: APM Dimensions

### End-User Experience Monitoring

There are 2 different types of end user experience monitoring which are explained in the following:

#### Synthetic:

The oldest approach to capturing end-user experience data where scripts are run against an application in production and record response time and availability results. This is also known as Last Mile monitoring

**Real User Monitoring:**

This technology requires the placement of software robots at various locations within the data center predominantly around the dynamic elements like switches, bridges, routes, gateways. This is also known as the First Mile monitoring.

**Discovery, Modeling and Display**

These technologies discover what software and hardware components are exercised as user-defined transactions are executed, and how those components are related to one another.

- Service Dependency Mapping: Technologies for discovering how different types of traffic flow among different types of physical and virtual infrastructure elements
- Transaction Profile: Models built from reports generated from the user defined transaction dimension of APM
- Network Topologies: Intended primarily to display physical devices and the physical links that allow these devices to communicate with one another.
- Bayesian networks: Intended to display random variables describing statistical properties of a system's behavior, while the links among the variables represent the conditional observational dependence of one variable on another.

**User Defined Transactions**

Historically, it meant a logical unit of work. In the context of APM, however, the term "transaction" means something different. In composite application, users or customers will typically perform a number of distinct operations that however un-connected they may be from the systems being accessed and exercised, form an integral action from the perspective of that user or customer. This "integral action" is what most APM vendors mean by the term "transaction." The user-defined transaction-profiling technologies will attempt to trace the effects of this transaction across an array of components and network paths.

**Component Deep Dive Monitoring**

The fourth dimension refers to a broad collection of technologies and products designed to monitor the performance of the various hardware and software components that support the execution of an application. What distinguishes component deep-dive monitoring in an application context from performance-monitoring technology in general is the ability to associate the latency or resource consumption being measured with the application causing the latency or resource consumption. In practice, such monitoring is largely confined to off-the-shelf application stacks, middleware (e.g., application servers, message queuing systems and service buses), database management systems (DBMSs) and aspects of network packet flow.

## Application Performance Analytics

Analytics are brought to bear to establish the root cause in the midst of the vast volumes of data generated in the first four steps, as well as to better anticipate and prepare for end-user experience problems that could emerge in the future.

## Monitoring Tools: Managing the Problem

APM Monitoring Model: The following section details the list of steps required for managing a problem in the application landscape.

- Collect response times by transaction and determine the first level alerting criterion. This is best achieved by using:
  - Passive agent that provides true end user performance,
  - Active agent that provides availability data.
- Understand and map all the components of the transaction. Several solutions are possible, but we believe that this model must be able to track each type of transaction, or each transaction, through the infrastructure, provide a template for debugging performance problems, and give full visibility into the transaction path. In addition, this dependency data should be available to improve the mapping of dependencies in a CMDB – Configuration Management Database.
- Monitor applications themselves. This includes Java EE and J2EE application servers, Microsoft .NET Framework, portal and Web server monitoring, a connector to collect performance data coming from mainframe-based transactions using IBM CICS/DB2 or IMS, messaging technologies such as WebSphere MQ or MQ-Series between distributed systems and mainframes, packaged applications provided by vendors such as SAP, Oracle, or other ISVs, and custom applications not written in Java.
- Monitor performance of the database(s). This includes the ability to analyze specific database performance issues.
- Monitor the physical and virtual components of the infrastructure.
- Combine all these parameters. This provides the ability to determine an alert, identify the root cause of this alert, and if possible predict an impending performance issue.
- Provide all of this information on a “single-pane-of-glass” dashboard

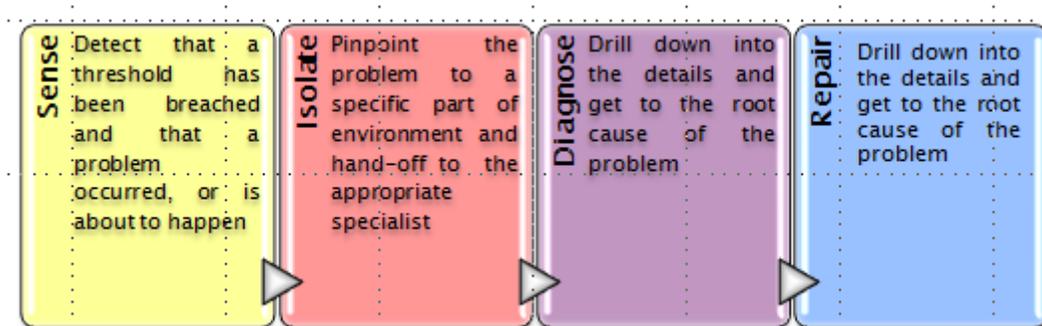


Figure 7: APM Problem Resolution

## APM Dashboards

The following sections deep dives into the various aspects of dashboards for different APM domains.

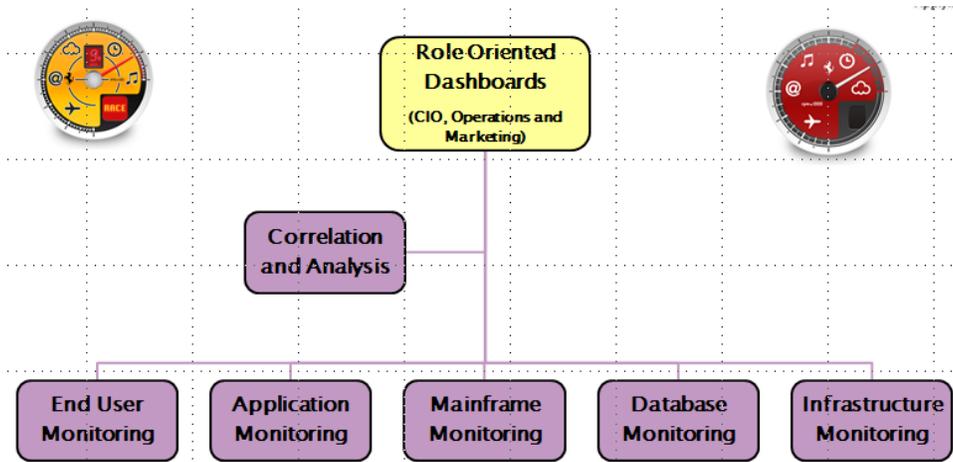


Figure 9: APM Dashboards

- **Real User Monitoring:** Performance from your actual users, visitor paths, impact of geography, ISP, browser size and type, Operating System, device and cache, performance degradation, availability data
- **Synthetic Monitoring:** Performance based on the scripts, browser size and type, Operating System, device and cache, performance degradation, availability data
- **Application Monitoring:** Response times, Transactions per second, User logged in per second
- **Business Activity Monitoring:** Customer, Journeys(Average Journey Fulfillment) and Holdings
- **Database Monitoring:** Transaction rates, Database query response times, Disk space Utilization, CPU Utilization, Number of active users
- **Mainframes:** Transaction Mapping Processor complex, LPAR and Operating System, Address Spaces / Jobs, STC's, IDMS Subsystems, Datacom subsystems, DB2 subsystems, CICS regions, IMS subsystems, MQ subsystems, tape and DASD
- **Infrastructure Servers Monitoring:** Hard Disk Utilization, Files Open/Owner, File Existence Monitor, File Size, Memory Utilization, CPU Utilization, Processes
- **Network:** Network Throughput, Current Logon, Failover Monitoring, Other Network Monitoring Points
- **Role Based Dashboards:** Performance and availability indicators for critical applications, locations and supporting infrastructure across the enterprise,

performance problems for the busiest URLs, supplemented with usage, performance and availability metrics, including the number of Users.

## Agent vs. Agentless Monitoring

The following graphic compares the pros and cons of agent vs. agentless monitoring.

	Agent-Based Deployment	Agentless Deployment
Pros	<ul style="list-style-type: none"> <li>❑ Light-weight - designed to limit impact and operate efficiently</li> <li>❑ Task-specific - designed to provide critical management-specific data</li> <li>❑ Secured communications - uses SSL or other encrypted means to provide data</li> <li>❑ Auto-updating - designed to update/upgrade itself to avoid maintenance overhead</li> <li>❑ Local intelligence - implements local "smarts" to adapt to changing conditions</li> </ul>	<ul style="list-style-type: none"> <li>❑ No risk - no additional software installed on critical systems</li> <li>❑ No/low impact - does not consume resources of critical systems</li> <li>❑ Rapid deployment - typically, low-to-no effort to deploy as standalone system</li> <li>❑ Scalable - can grow as fast as your system</li> <li>❑ Full Coverage - all participating servers can be monitored for the same low deployment effort, not just critical machines</li> </ul>
Cons	<ul style="list-style-type: none"> <li>❑ High risk - poses hygiene threat to stability of critical server/device environments</li> <li>❑ High impact - may consume significant percentage of server/device CPU, memory or storage</li> <li>❑ Time-consuming - deployment/maintenance overhead multiplied by number of servers/devices to be monitored</li> <li>❑ Limited to system where deployed - cannot see beyond the system it is installed on</li> <li>❑ Limited coverage - only able to see what agent was designed for; only able to see systems that have been instrumented</li> <li>❑ Yet-another-agent - there may be numerous agents already deployed for other systems</li> </ul>	<ul style="list-style-type: none"> <li>❑ Best-effort - WYSIWYG</li> <li>❑ Limited by richness of data source - data sources may not offer full visibility</li> <li>❑ Configuration - typically requires pre-existing services or agents be turned on or configured may also require changes to networking and firewalls</li> <li>❑ Imprecise - nature of data sources may limit amount of data available</li> </ul>

Source: Inetco

## Conclusion

**APM Future Roadmap:** One approach is to analyze the current APM maturity and then recommend a technology adoption roadmap based on the business goals and objectives, considering the complexity involved.

There is additional emphasis on end user monitoring hence the support team has to ensure that the end user experience always meets expectations. The rule of thumb is if there is a problem find out before the customer complains. The APM landscape consists of a combination of agent and agentless monitoring, starting with agentless for rapid initial deployment.

One of the major concerns in APM landscape is the virtualized infrastructure which impacts the metrics that are gathered from the guest OS and requiring a new approach to APM aligned with the needs of virtualized systems.

---

### About the Author:

#### Acknowledgements

I would like to thank **EY IT Advisory Leadership** for giving me the opportunity to work in this area and for providing with the guidance and valuable inputs.



**Sameer S Paradkar** is an Enterprise Architect with the EY IT Advisory. Sameer is responsible for the Presales, Consulting and Advisory and Enterprise Transformation. In the past Sameer has engaged with customers at Departmental Heads, Technical and Business leader levels and advised them on technology, adoption strategy and roadmaps. Sameer is creative and an out-of-the-box thinker. He has a deep understanding of different systems and is able to articulate advantages and disadvantages of each as they relate to a customer business model. Sameer has extensive experience in the ICT industry and he has worked in the U.S.A, UK, Europe, Asia Pacific and the Middle East regions extensively.

#### Disclaimer

The views expressed in this Article are solely those of the author.