JOURNAL OF OBJECT TECHNOLOGY

Online at www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2010**

Vol. 9, No. 2, March-April 2010

“Architected” Cloud Solutions Revealed

Mahesh H. Dodani, IBM, U.S.A.

1 PRACTICAL CLOUD SOLUTIONS BASED ON ARCHITECTURE

“It's getting harder to focus on the vision of cloud computing these days. While there are still plenty of critical and complex problems to solve, and many, many implications of this disruptive operations model that have yet to be understood, the truth is that we've entered a new phase in the evolution of cloud adoption. Real work now exceeds theory when it comes to both new online content and work produced…

Development and test services, such as SkyTap and Soasta, are thriving. The cloud model really works well for the dynamic resource usage model of software engineering. In fact, it works so well that IBM is putting some real muscle into the game...

If you are wondering if cloud computing is a fad, the evidence to the contrary is all around you. I heartily recommend that you really listen to what is being said, understand how the cloud is being used, and seriously evaluate how this disruptive model will change your projects, your organization, and even your career. Clearly, there are many technologists who already have.” – Practice Overtaking Theory in Cloud Computing ... James Urquhart Manes Blog

This paper continues with the practice of designing cloud solutions based on an architecture that I described in my first article for 2010. This paper focuses on showing real solutions that have been implemented using the cloud architecture. The cloud architecture, shown in Figure 1, has been employed in numerous IBM client engagements and IBM’s public cloud offerings. These examples demonstrate its use in delivering value to the client from adopting cloud. This paper will focus on describing three of these examples to showcase the use of the architecture in delivering cloud capabilities. We focus on a common type of cloud service – the development and test platform services, and show three solutions that span different delivery models –public, private, and collaborative. Our experiences have shown that development and test platform services are a common entry point into cloud computing for many enterprises. For each cloud scenario, we summarize the business requirements, the architected solution, and show how the architectural components can be utilized to drive the implementation. As we have discussed over many articles, cloud adoption is a journey, and many of the scenarios that are described in this section are snapshots of the cloud journeys in progress. Since these journeys are ongoing, it is not possible to provide details of the ROI and business

Mahesh Dodani: “Architected” Cloud Solutions Revealed, in *Journal of Object Technology*, vol. 9, no. 2, March - April 2010, pages 27-36, http://www.jot.fm/issues/issue_2010_03/column4/

value delivered. However, these are being tracked by the individual projects, and as they become available, I will collate the results and report these in an upcoming paper.

Let us start by summarizing the components and capabilities of the cloud architecture shown in Figure 1. The three main roles in this architectural model are the service consumer (left hand side), the service provider (middle) and the service creator (right hand side). The service provider hosts services which are created by the service creator, based on a management platform consisting of an Operational Support System (OSS) and a Business Support System (BSS).

With a strict separation of concerns the Cloud Architecture enables specific perspectives:

- From the service consumer's perspective, a simplified interface/API is needed with well-understood service offerings, pricing and contracts. The value proposition for the service consumer is to get much faster whatever is needed (e.g. getting a financial model simulation IT environment in a few hours vs. in 6 weeks) while they only have to pay for the period of time the service is used.
- From the service Provider's perspective, a highly efficient service delivery and service support infrastructure and organization is needed in order to provide differentiated, well-understood, standardized and high-quality services to end users. Service management and a dynamic infrastructure make it possible for significant economies of scale to be achieved. A self-service portal allows exposing a well-defined set of services in a highly automated fashion to a very attractive cost point.
- From the service creator’s perspective, a tooling environment for modeling, assembling service elements (virtual images for example) and an effective means of managing the service lifecycle.

Business requirements drive the cloud service offerings and the business support systems. The architecture must be able to support a range of service offerings, including infrastructure, platform and software/applications that are needed to support the business needs. These services offerings should be able to address both enterprises using cloud computing to supplement traditional IT as well as service providers that support multiple customers. The discussion of how service providers can handle multiple customers (aka multi-tenancy) is beyond the scope of this paper. Furthermore, with different cloud-suitable services emerging, the cloud architecture will need to provide support for workload focused offerings, including analytics, application development/test, and collaboration/e-mail services along with industry specific services. The business support systems focus on managing the business side of delivering cloud services, including managing customers, accounts, orders, subscribers, etc. Underlying these management services is the need for reporting (on usage, meeting SLAs, licenses, etc.) as well as all the capabilities for charging (including billing, invoices, settlement, etc.)

Technical requirements drive the underlying IT management patterns, including a focus on handling the top adoption factors influencing cloud services – i.e. trust, security, availability, and SLA management. The main capabilities are shown within the architecture in the operational support systems. The architecture must focus on handling the major concerns of enterprises by facilitating internal/external cloud interoperability. This requires the architecture, for example, to handle licensing and security issues to span traditional IT, private and public clouds. Additionally, the architecture must support a self service paradigm to manage clouds using a portal which requires a robust and easy to use service management solution. A portal facilitates access to the catalog of services and to manage security services. Of course, all of these services must be provided on top of a virtualized infrastructure of the underlying IT resources.

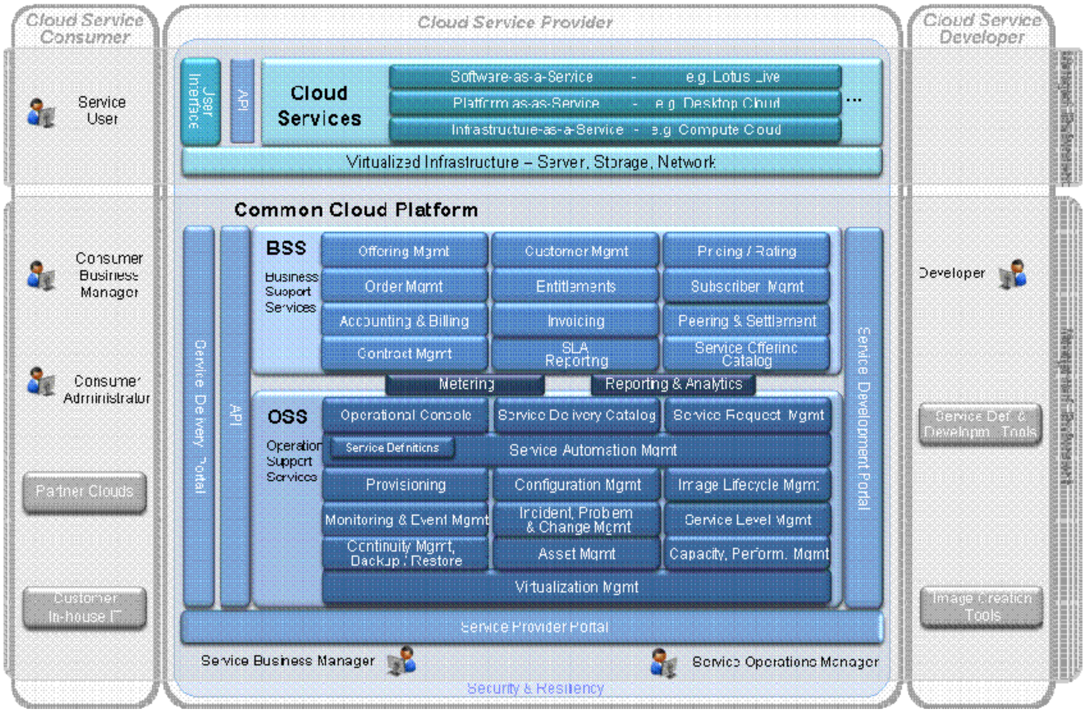

Figure 1: Cloud Reference Architecture

2 DEVELOPMENT AND TEST SERVICES ON THE IBM PUBLIC CLOUD

The Smart Business Development and Test on the IBM Cloud is designed to augment and enhance software development and delivery capabilities, particularly in large enterprises where IT departments handle hundreds of development projects every year. Unlike traditional development environments, developers can log on to IBM Smart Business

Development and Test on the IBM Cloud and get access to customizable virtual machines in minutes.

This new environment provides a range of services to help application developers and testers speed the development and delivery of software applications including compute and storage infrastructure services, software delivery platform services, and middleware software (e.g. application servers, database servers) platform services. To help customers leverage existing investments, these new services support development across heterogeneous environments, including Java, Open Source, and .NET. In addition, pre-configured integrations of some of these services are available based on the IBM Rational Jazz framework, which dynamically integrates and synchronizes people, processes and assets associated with software development projects.

The IBM Smart Business Development and Test on the IBM Cloud provides the following features to software developers and testers:

- Instant self-service provisioning of development and test environments.
- A dynamic and elastic environment to support an organizations' test lab and build infrastructure.
- Flexible deployment pricing options covering several delivery models. The private hosted option features fixed price, time and materials, or pay as you go pricing. The shared private or public cloud options facilitate multi-tenancy and features a utility or metered pricing.

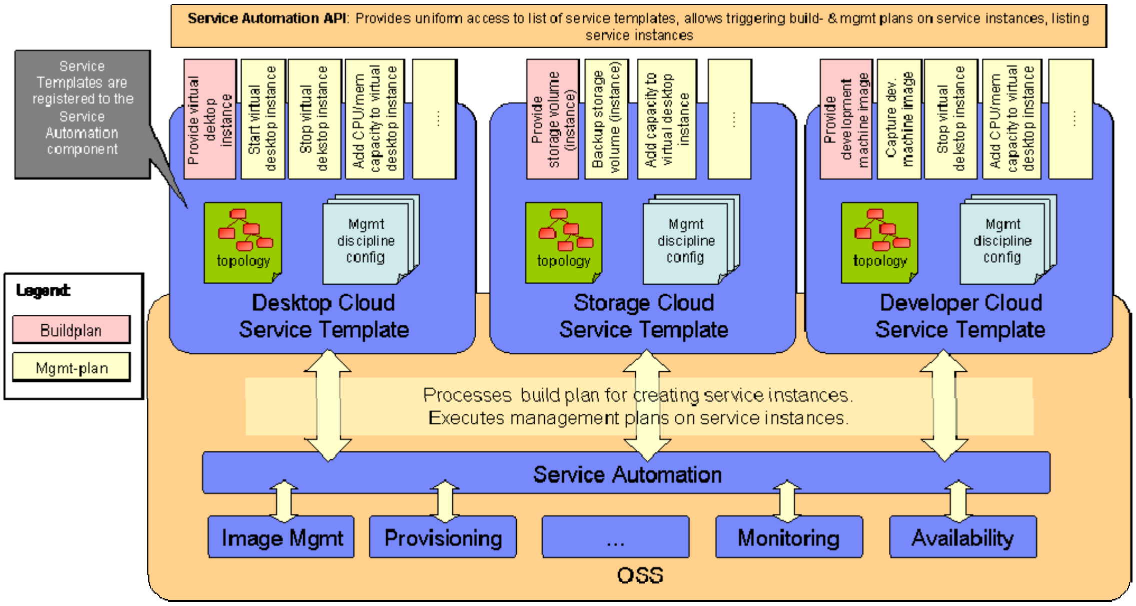
Service Automation API: Provides uniform access to list of service templates, allows triggering build- & mgmt plans on service instances, listing service instances
Service Templates are registered to the Service Automation component
Provide virtual dektop instance
Start virtual desktop instance
Stop virtual dekstop instance
Add CPU/mem capacity to virtual desktop instance
...
Provide storage volume (instance)
Backup storage volume (instance)
Add capacity to virtual desktop instance
...
Provide development machine image
Capture dev. machine image
Stop virtual dekstop instance
Add CPU/mem capacity to virtual desktop instance
...
topology
Mgmt discipline config
Desktop Cloud Service Template
Storage Cloud Service Template
Developer Cloud Service Template
Legend
Buildplan
Mgmt-plan
Processes build plan for creating service instances.
Executes management plans on service instances.
Service Automation
Image Mgmt
Provisioning
...
Monitoring
Availability
OSS

Figure 2: The OSS Capabilities of a Public Cloud

As is evident from the requirements above, the implementation of the IBM Smart Business Development and Test on the IBM Cloud requires capabilities from both the

BSS and OSS layers of the architectural model. Figure 2 shows how the underlying OSS capabilities are leveraged to deliver on the development and test cloud services.

The metering and reporting & analytics architectural components support the required BSS capabilities inherent in being able to manage the service offerings, accounts, and billing in the public cloud. In particular, the following BSS capabilities are used in delivering the cloud services – the service offering catalog component to make the services available to the consumers, the subscriber management component to handle users, the pricing/rating component to handle the various pricing options, and the accounting & billing component to manage payments.

3 A PRIVATE DEVELOPMENT & TEST CLOUD

The IBM Tivoli Development Services (TDS) organization provides IT services for Tivoli and other organizations in IBM Software Group in the form of lab services, host services, build/390 services, and virtual resource services to enable the software development cycle. The TDS organization faced several key business challenges. First, the organization had 24 development labs spread across the globe with minimal virtualization, resulting in much higher capital expense, management and administration costs, and less than optimal efficiency because of limited ability to reuse and share IT resources and best practices. In addition, request workflows, capacity management and administration processes were mostly manual, leading to average delivery times for new resource requests of weeks to months and driving up management costs. Lastly, Tivoli’s physical resources were largely underutilized, with an average utilization of about 45%.

The specific functional requirements are summarized from the perspective of the primary users (developers and testers) and the administrators who monitor and manage the cloud environment. In particular, the developer and tester use cases facilitate interactions with the dev/test environments through the entire lifecycle, including the ability to manage dev/test environments, request these environments, use them, snapshot/restore the environments, and release them. In addition, a dev/test manager has the ability to reserve resources to cover all the development and test for the projects under their control. The administrator requirements define how the cloud environment is monitored and managed from the perspective of IT resources, virtualized environments, and cloud services. In particular, the resource administrator is interested in monitoring different types of IT resources involved in delivering cloud services, including compute, memory and storage; and the ability to manage different aspects of the IT resources, including utilization and capacity. The virtualization administrator focuses on monitoring VMs, and manages workloads associated with the cloud services as well as the performance (by increasing the IT resources allocated to the VM.) Finally, the cloud administrator monitors the entire cloud environment, and manages any incidents and events to ensure efficient and effective delivery of the cloud services to the established SLAs.

Based on the above requirements, the overall objectives of the Tivoli Development Cloud include the following:

- Developers/testers reserve dev/test environments from a service catalog, use and release the environments that are handled and managed from "nearby" virtualized infrastructures.
- A central site monitors the geographically dispersed cloud environments and manages the cloud environment in areas such as performance, availability, utilization, and capacity.
- Capacity is increased by "plugging in" a virtualized infrastructure anywhere in the world.

Note that the design of both the geographical cloud component and the central monitoring & management components utilize primarily the capabilities defined in the OSS layer of the architecture.

The geographical cloud component includes both the managed and management environments that provide services to its users through the service catalog. The managed environment itself can be an independent resource pool available in the geography, and can be plugged in to support the needs of the Tivoli Development Cloud. The service catalog provides standard services for developers and testers. Each geographic cloud can track the usage of resources based on the services requested, and provide input to support keeping track of usage against reservations of resources. The implementation of the geographical cloud environment utilizes the key OSS layer architectural components, primarily the service automation manager to orchestrate the cloud service delivery along with the supporting components necessary to ensure effective service delivery; the respective portals for requesting, activating and accessing the services for both the service consumer and provider; the service delivery catalog and service request manager components to handle the requests from the service consumer; the provisioning component to build up the virtual development/test environments associated with the request; and the virtualization management component to manage the underlying IT (compute, storage, and networking) resources.

The central monitoring and management component ensures effective and efficient service delivery across the entire Tivoli Development Cloud. It monitors all of the resources, service requests, operating systems, and energy across all the geographic cloud environments. It allows this monitoring information to be collected and presented through dashboards suitable for administrators to manage different aspects of the Tivoli Development Cloud. The capabilities include the ability to monitor and manage performance and availability of the resources, the ability to monitor and manage utilization of resources and optimize it for the test workloads, and the ability to analyze the usage of resources to forecast and plan for capacity needs in the future. The implementation of the central monitoring and management site utilizes many of the associated OSS layer architectural components as depicted in Figure 3.

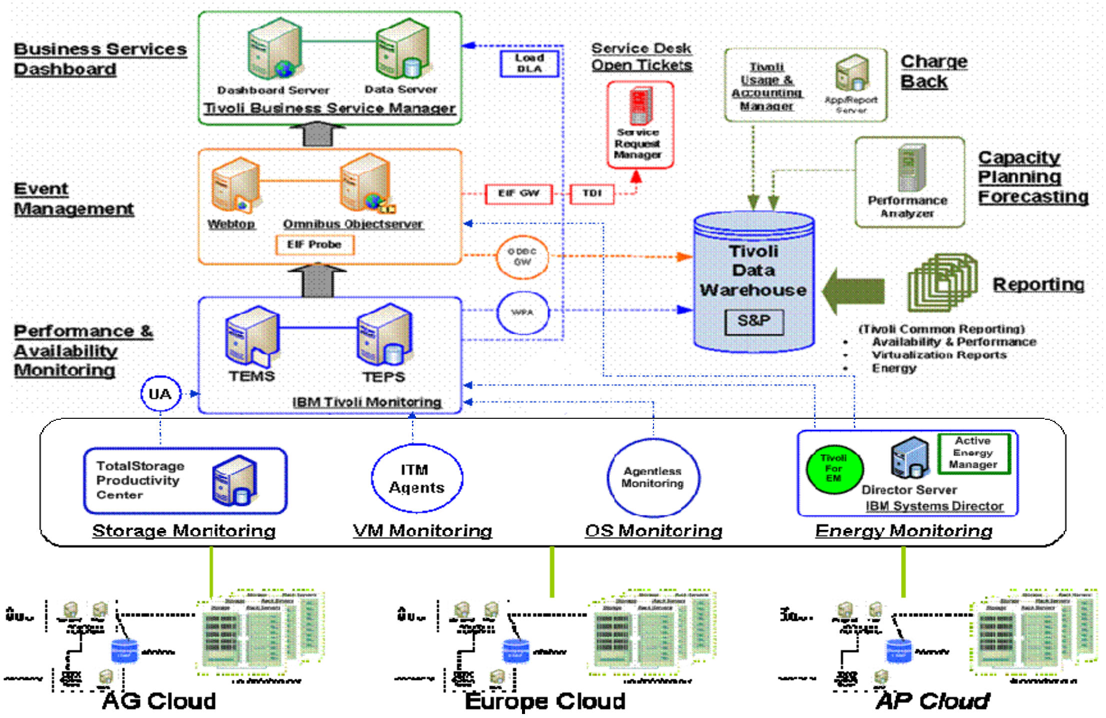

Figure 3: The Central Monitoring and Management Implementation

4 A COLLABORATIVE CLOUD FOR BUSINESS PARTNERS TO DEVELOP AND TEST SERVICES

SK Telecom is the leading provider of mobile telecommunications services in South Korea, with 50.5% of the market share in 2008. It was the 6th company in the world to surpass the ten million subscriber mark, and currently has a presence also in China, Japan, Vietnam, and the whole Asian continent. As a telecommunications operator, SK Telecom has a number of business partners that provide content and services that are offered to the end user (mobile phone user). SK Telecom provided systems on which the business partners developed these services, and then moved them into production. These systems were known as incubation systems, and in early 2009 it was decided to industrialize this system and provide a cloud to host these systems.

There were to be two server farms in the cloud: a development server farm in which the business partners would develop and test their services, and a production server farm to host the services when they were ready to be offered to the end users. This was to be a relatively contained system in terms of the size of the managed environment, but it was to be a full production cloud with functional requirements that exceeded the out-of-the-box capabilities of any existing commercial offerings. It was also to be a showcase for SK Telecom as an innovator, and key proof-point for larger and more ambitious Cloud projects within SK Telecom and for other companies in Asia Pacific.

The implementation of the SK Telecom Cloud addresses the requirements above as shown in Figure 4. As is evident from the figure, the implementation utilizes many of the key OSS architectural components: service request portal, service automation manager, service request manager, provisioning, monitoring & event management, security & resiliency, and the virtualization management (to interface with the managed environment.)

The Service Portal provides a number of services and applications to the user. In the context of this project, the Service Portal acts as a Korean language front-end into the SK Telecom Cloud, offering services by means of a Service Catalog and managing the workflows required to fully specify and submit a Service Request. It also supports the workflows required for the analysis and approval of the Service Requests by the authorized approvers.

The management environment contains the management servers that are needed to make the Cloud resources functional. The management environment includes the core components of a Service Request Manager, an Automation Manager, a workflow-based provisioning manager and a Configuration Management Data Base (CMDB) which is used to store the data in an Information Technology Infrastructure Library (ITIL) compliant service management best practice format. The Service Request Manager advertises Cloud Services by means of a Service Catalog, and provides the interfaces that allow the Service Portal to fully specify and submit a Service Request. It then supports the workflows for approval of the request (if required) before invoking the Automation Manager to fulfill it. The Automation Manager interprets the Service Requests, and uses predefined workflows and automation programs (Management Plans) to fulfill the requested service. It does this through the invocation of management tasks, and in particular through tasks that invoke workflows in the Provisioning Manager. The Provisioning Manager is triggered by the Automation Manager to provision or de-provision virtual servers, to install or uninstall software packages, or to invoke configuration actions on virtual servers. Provisioning actions are driven by workflows. The Provisioning Manager is also responsible for discovering resources in the Managed Environment, in order that they can be populated into the CCMDB and therefore be referenced by the Automation Manager. The Directory Server provides the authentication service for all of the elements in the Management Environment that require it, and for the Service Portal. It stores the user IDs, passwords, and security groups defined in the environment. The extended management environment contains the management systems to integrate with existing systems to monitor the managed environment, along with the ability to authenticate and authorize users of the cloud.

5 SUMMARY

Most CIOs are eager to see if the potential of cloud computing can be transformed into tangible business value that will help them reduce the cost of running their enterprises while continuing to deliver new, innovative business services. Deriving business value from cloud computing requires a careful "architected" design of the solution that meets the business' requirements. This paper showed several examples of how to architect a cloud solution by looking at the typical requirements inherent in any cloud implementation and showing how to derive a solution that addresses these requirements using capabilities provided by the cloud architecture. As we have discussed over the last few articles, it is important to establish the cloud architecture that will be used to architect the cloud solution, along with an implementation approach that provides a guide on how a cloud solution can be "architected", and then strengthen the architecture and implementation approach through several cloud solutions that derive business value. We use such an architected cloud solution guide to help our clients through their cloud journeys and ensure that they are deriving tangible business value from cloud computing.

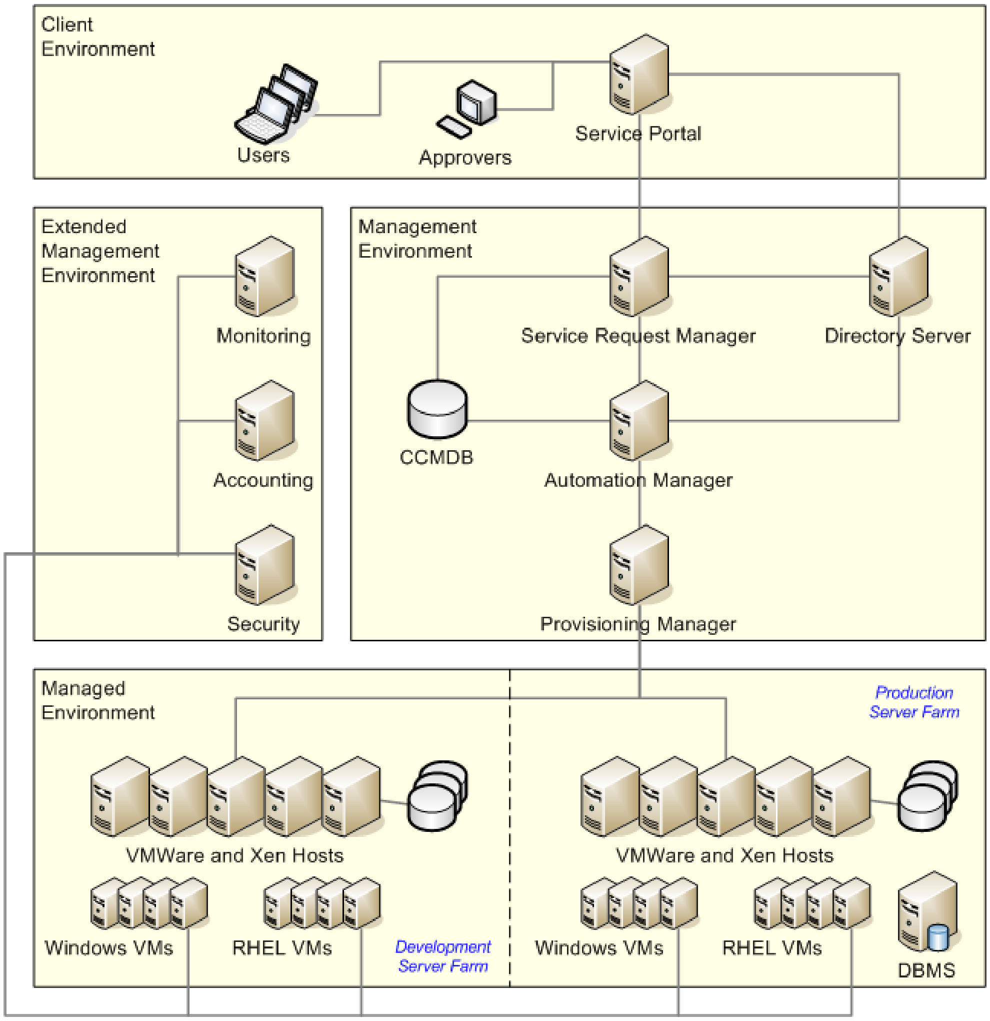

Figure 4: The SK Telecom Cloud Implementation

About the author

Mahesh Dodani is a software architect at IBM focusing on Cloud Computing. His primary interests are in enabling communities of practitioners to design and build solutions that address complex business needs and deliver value. He can be reached at dodani@us.ibm.com.