2010 IEEE 6th World Congress on Services

Service Storm: A Self-Service Telecommunication Service Delivery Platform with Platform-as-a-Service Technology

Yu Chen Zhou, Liang Xue, Xin Peng Liu, Xi Ning Wang, Xiao Xing Liang and Chang Hua Sun
China SOA Design Center, IBM China Development Laboratory
Diamond Building, Zhongguancun Software Park, Haidian District, Beijing, P.R. China 100193
{zhouyuc, xueliang, xinpengl, wangxn, liangxx, schangh}@cn.ibm.com

Abstract—**To attract and serve an expanded customer base, telecommunication service providers need to provide more targeted, focused and personalized services, which brings challenges to both business model and technologies to development of value added services. In this paper, we introduce Service Storm, a novel self-service telecommunication Service Delivery Platform with Platform-as-a-Service technologies to support rapid and flexible construction and delivery of value added services in the cloud environment as an open developer community. Firstly, we analyze the characteristics of four categories of users including Individual User, Organization User, Telecommunication Operator, and Business Partner, and present the architecture of Service Storm to support new business model targeting at long tail applications as well as conventional business models. Then we highlight the key technologies including 1) Rapid assembly model and tools for codeless and off-premise integration of telecom services and application logic based on Web 2.0 technologies, 2) Cloud based resource isolation, management, capacity planning and scaling for dynamic topology deployment and elastic infrastructure support, and 3) Automatic deployment and monitoring in runtime. Finally we illustrate the sample of SMS and Web based mobile workflow for insurance order management to show the advantages of Service Storm. We demonstrate how these technologies and architectures enable the new business model for telecommunication operator and significantly enhance the diversity of value added services with lower cost and shorter time to market.**

Keywords-Telecommunication SDP; Cloud Computing; Platform-as-a-Service

I. INTRODUCTION

Telecommunication Service Delivery Platform (SDP) is the open, horizontal, and standard based platform which enables Service Providers to create and deliver new innovative services for end users. To attract and serve an expanded customer base, Telecommunication Service Providers are moving from voice centered legacy network service, through cross-industry and cross-community, to services that target at the "long tail". Services will become more targeted, focused, personalized, which brings challenges to both business model and technologies to develop and consume such value added services. Telecommunication networks and the Internet have evolved as disjoint worlds with regard to software applications and application development technologies [1].

Lots of efforts have been taken on telecommunication SDP in industry and academia. In [4], Service Template is defined to represent service pattern, and Service Configurable Point is defined to configure the basic properties of one service. It provides a flow diagram editor based on desktop tools via the XML-based scripting language. In [2] and [3], SDP via SOA is introduced, in which service elements are shared to reduce development time and cost. The real-time status of products, employees, and business process are visualized for better understanding of the Supply Chain Management. Whereas, more efforts should be taken to meet the changing requirements and enable the new business models by leveraging the emerging technologies such as cloud computing.

According to the characteristics of SDP, cloud becomes the best way to host the solutions with its low cost for value added services and Platform-as-a-Service (PaaS) implements the best way to build the open development community like business models for telecommunication services. A new range of services and applications are delivered to customers through a number of business models, most prominently in PaaS models.

The PaaS model is a novel approach for software suppliers that want to focus primarily on the software development cycle and the monetization of new applications, thus bypassing the investment in and maintenance of the underlying infrastructure/services for application design, development, testing, deployment and hosting [4]. PaaS creates a virtual platform for application development and deployment. In PaaS, the system's provider makes most of the choices that determine how the application infrastructure operates. Users build their applications with the provider's on-demand tools and collaborative development environment [5].

PaaS enables centralized cloud computing model by which different roles in the ecosystem are magnetized around value added services, including telecommunication operator, partner for development of value added services, enterprises using telecommunication related application, and individual customers, into a centralized hosting environment. It fits a double-sided market model where in a platform business is mediated between the demand and supply side, creating value for upstream developers and downstream customers and extracting revenues from both sides [1]. Telecommunication service provider will benefit from this new business model to achieve low cost, openness, faster time to market, rich application, and rich approaches for

978-0-7695-4129-7/10 $26.00 © 2010 IEEE
DOI 10.1109/SERVICES.2010.45

increasing revenue, etc. Since the telecommunication operator provides the environments for the entire software cycle, the developers may reduce the cost of provisioning and management of their own IT infrastructure and consider the runtime fees as their main costs based on the actual computing resources used by applications or the number of users who are using them.

By engaging several telecommunication and mobile communication operators with more than dozen millions of customers, and analyzing the requirements and challenges in both business and technical models, we summarized a number of factors that should be taken into account, and designed and implemented Service Storm, a cloud enabled self-service telecommunication SDP in PaaS model by leveraging Web 2.0 and other virtualization technologies to address the requirements of different roles in the ecosystem. Different from conventional enterprise or partner based development model, it enables the "open development community" model, to achieve the goal that every one, not only the professional developer, could rapidly develop the telecommunication based application according to the understanding of domain, community or even user specific requirements. Service Storm provides a rapid integration model for personalized value added services. This model facilitates the application creation for non-technical user by assembling the registered services, mashups [6], built-in rules, events, and database services, etc. To ensure the independency and security of the different user groups a three-layer isolation mechanism is provided to secure the resources from three layers. To facilitate the resource allocation, an effective management and capacity planning mechanism is provided to allocate the resources more appropriately. It provides an automatic deployment mechanism to support rapid and flexible deployment and scalability adjustment with the increased needs from the business users. It also provides an integrated view about the system health both from business and IT perspectives for the running business solutions and its hosting platform.

The rest of the paper is organized as follows: Section II systematically discusses the business model and technical challenges of self-service telecommunication SDP, and introduces the architecture and integration with the telecommunication services. Section III illustrates the rapid integration model and tools for off-premise and codeless development of application in Service Storm. Section IV describes the dynamic optimization for business agility, including system scaling, three-layer resource isolation mechanism, resource management and capacity planning. Section V addresses the problems of automatic deployment and monitoring for Service Storm where a configurable deployment model for flexible solution deployment is discussed. Section VI demonstrates the usage and advantages of Service Storm with a sample of development and deployment of a mobile workflow for Insurance Order Management. Section VII concludes the paper and summarizes the future works.

II. Service Storm Architecture Addresing Business and Technical Challengs

Telecommunication operators must innovate, differentiate the customer experience and be agile to win in the market. In their business model, the success factors for telecommunication operators are associated with the proper balance between the platform trade offs – open or closed, proprietary or nonproprietary, and free or paid elements – in order to avoid fragmentation [1]. From the perspective of business model, we classify the roles according to specific requirements in the ecosystem of telecommunication value added services into four categories, including individual user, telecommunication operator, business partner and organization user.

Individual User – previously only acts as end user and has limited options for personalized services, and lacks of IT knowledge for developing his/her own application though he/she has deep understanding of the requirements specific to individual or small community. It is required to have the environment as a black box with easy-to-use functionality to innovate for personal needs, or business opportunities. Such requirements from the individual users, who dream to start up from the scratch with similar model of "American Idol", bring brand new business model for telecommunication operator to increase revenue besides the conventional voice and SMS based business. End users may significantly benefit from the rich content, multimedia and highly personalized services in such model.

Telecommunication Operator – develops and hosts the value added services for public user communities, which falls into the conventional business model of value added services, with need for more powerful and flexible development model and infrastructure to shorten the time to market for more and more fine-grained value added services to smaller user community.

Business Partner – develops value added services for public user community, and splits the profit with telecommunication operator. It has the similar requirements with those from telecommunication operator, meanwhile, plus those for reducing cost for hosting the development, testing and provisioning product environment compliant with the standards specific to telecommunication industry.

Organization User – conventionally owns industry specific application as a part of enterprise system by leveraging telecommunication service, such as mobile workforce management and front-end customer relationship management system, etc, especially for small and middle enterprise lacking of IT capability. It is required to reduce the cost to maintain the IT infrastructure by partial out-sourcing, and develop and deploy new application rapidly with business agility.

Service Storm is designed to meet the requirement of all the four categories of users. The architecture of Service Storm is shown in Fig. 1.

1) Service Assembler: It is a visualized tool for the developer to define the application logic in a drag and drop manner, and integrate the telecommunicaiton services, such as SMS, MMS, Location Based Service (LBS), and other elements including Web 2.0 UI and external third party services, for implementing the value added services, and automatically generate the deployable artifacts without coding.

2) Management Components: By collaborating with BOSS, it provides management functionality for SDP. It consists of three components: Service Management for managing the telecommunication services exposed in SDP, external third party services registered in SDP, and the newly generated applications; Solution Management for managing the solution implementing the services and deployed in cloud environment; and Cloud Management for managing the computing resources as cloud infrastructure.

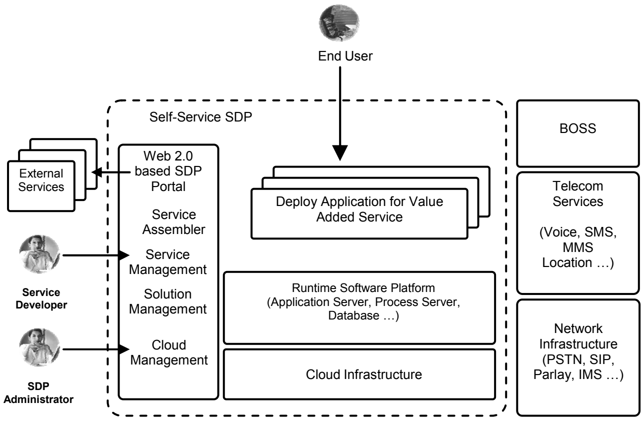


Figure 1. Architecture of Service Storm.

3) Runtime Software Platform: Built on cloud environment, it provides the runtime environment to host the applications for value added services. Normally it consists of application servers, process servers, and other runtime environment focusing on handling business rules, events, and operating status, etc.

4) Cloud Infrastructure: Cloud environment provides a centralized management on virtualized computing resources to provide the underlying capabilities needed by solution deployment for resource sharing based on cost saving.

Leveraging the capability provided by Service Storm, various types of applications as value added services could be constructed and deployed. These applications include the below two typical ones:

- *Web Services (Composite Services):* The organization users may build up application and expose services for integrating with their enterprise system. The telecommunication operator and business partner may also help to develop some key services that could be leveraged by the organization users.
- *Mashups:* Mashups are flourishing on current programmable and innovative Web. Developers can easily create mashups for richer user experience comparing with the coventional voice based services, for example, combination of map widget, location based service and MMS based navigation.

III. Rapid Assembling Model and Tool for Personalized Services

According to the above business and technical challenges, one of the key enablers to support rapid development of value added services in the open development community is the service assembly model and tool.

From the perspective of service assembly model, one of the key requirements to be fulfilled is to support a non-professional developer, even the business user, to create personalized application for value added service, with more focuses on the business scenario and less focuses on IT infrastructure and code. Accordingly, codeless service assembly model with visualized modeling of application logic, declarative language and automatic code generation should be taken into consideration. In Service Storm, Web Service Business Process Execution Language (BPEL) is the prominent standard to compose web services into higher-level web services recursively [7]. With BPEL, most of the telecommunication services exposed as web services, such as those Parlay X web services, could be easily integrated into an application. And the capabilities of BPEL are powerful enough to express the application logic in most cases of value added service. Besides the above reasons, BPEL is originally designed as an XML based declarative language so that it is easy to be generated, processed and deployed for the purpose of automation. Therefore, BPEL-based applications, business processes, and web services are supported as the targeted building blocks in Service Storm.

From the perspective of tooling, web based off-premise development tools become the best choice for the user to manage the whole application lifecycle by using Web browser in a drag-drop model. The application can be automatically generated and deployed to the cloud infrastructure. Testing and version control of the application are also available in Service Storm.

Fig. 2 depicts the integration model of Service Storm. It uses a web 2.0 based development tool, Service Assembler, to orchestrate the supported web services to build BPEL-based applications. These web services include telecommunication services like SMS, MMS, voice and LBS, external services registered in Service Storm, and user uploaded applications.

Besides orchestrating web services into process centered application, widgets and mashup capabilities are also supported and provided in Service Storm. The human task of the process is a natural glue point for integrating the web 2.0 mashups with the process. Generally, a mashup is treated as a web page or application that uses functionality from two or more widgets to create a new service. And a widget is treated as a web page or an application to provide certain service or functionality in Service Storm. The business process developed via Service Assembler can interact and exchange information with the widget pages at running time.

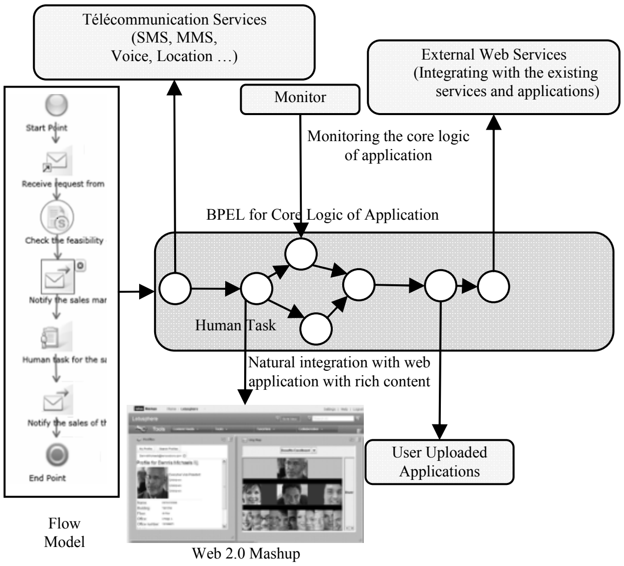
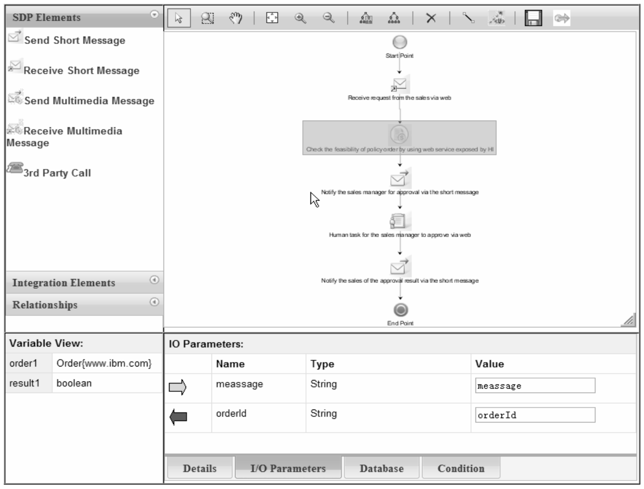


Figure 2. Service Assembly Model.

In sum, Service Assembler makes the application development much easier for non-technical user without expertise on programming language. And the human task is integrated with mashup to utilize the widgets, feeds, and other web resources to build meaningful Web UI. After the application logic is built by using Service Assembler, the BPEL modules and monitor modules can be generated automatically which can bridge the gap between business level and infrastructure level of process development.


SDP Elements
Send Short Message
Receive Short Message
Send Multimedia Message
Receive Multimedia Message
3rd Party Call
Integration Elements
Relationships
Start Point
Receive request from the sales via web
Check the feasibility of policy order by using web service exposed by HI
Notify the sales manager for approval via the short message
Human task for the sales manager to approve via web
Notify the sales of the approval result via the short message
End Point
Details
I/O Parameters
Database
Condition

Variable View:

order1	Order{www.ibm.com}
result1	boolean

IO Parameters:

	Name	Type	Value
⇨	meassage	String	meassage
⬅	orderId	String	orderId

Figure 3. Service Assembler Implemented as Web 2.0 Widget

Fig. 3 shows the web user interface of Service Assembler. It is a web based process development tool, aiming to provide an easy-to-use way to define business process for business users without the complex programming skills. The left top area is the Palette which contains the key elements of web services and mashups. Each element represents one or more built-in or external registered web services or other necessary web 2.0 elements for mashup. The left bottom is the variables used in one process. The right center is Canvas and Toolbar, and the right bottom is used to define properties and their configurations for the key elements in the process. By using Service Assembler the businesses user can drag elements from Palette and drop to Canvas. These elements are treated as nodes on Canvas. The business user can configure the detail of web service, like input/output parameters by changing the property of each node. Relationships between web services are defined by links between nodes on Canvas. After drag-dropping and properties configuration a final process flow model compliant with BPEL is generated that has business meaning for the non-technical users.

IV. Resource Management, Scaling and Capacity Planning

Cloud platforms host several independent applications on a shared resource pool with the ability to allocate computing power to applications on a per-demand basis. A key challenge for cloud providers is to manage the resources while taking into account both high-level quality of service requirements of hosted applications and resource management costs. Before application is deployed, the application runtime environment for hosting the deployed application should be allocated with necessary resources and be created automatically in SDP. For the sake of security, Service Storm ensures that the allocated resources have been isolated for different user groups via the three-layer resource isolation mechanism. Meanwhile, resources allocated to service providers can be scaled up and down depending on the user group size or the business growth requirement with the help of capacity planning mechanism which is provided to manage the resources more effectively by catering to the requirements from the different roles.

A. Resource Isolation

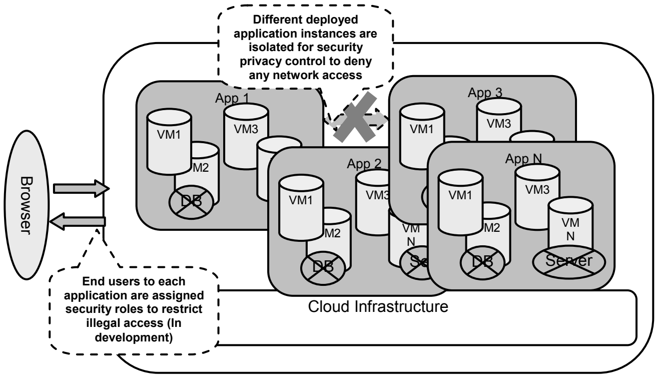
Cloud computing enables services running on leased computation platforms, and decouples the service provider from the platform owner and the cloud infrastructure provider. Such a model implies that the infrastructure provider should provide the ability to properly isolate services from each other. The resources in Service Storm consists of physical resources including hosts, CPU, memory, disk space, network, and software resources including OS, middleware, and application, etc. One group of service providers and their end users should be isolated from the other groups to avoid interferences or conflicts. Service Storm provides a three-layer isolation mechanism which helps to isolate the resources in the following ways:

- Application Layer

The first type of isolation we consider is application level isolation which ensures that the application data and business logic are kept adequately segregated and that the end users are not able to gain unauthorized access to each other’s data and business logic. Under such situation, the account information of end users is associated with the applications. Only the authorized end users can access the application which is exposed by the corresponding service provider. This ensures that end users will have accessibility for the applications created by a special group of telecommunication operator and business partner. Thus the applications are isolated from each other for different end users and telecommunication operators and business partners.

- Network Layer

Network isolation is often used to enhance the security and efficiency of the network by isolating certain network traffic to certain physical networks. Generally there are two types of isolation technologies: one is physical isolation which cuts off the physical conjunction between physical nodes, and the other is logical isolation which often uses the software to realize the isolation. Regarding for virtual machines, coupled with virtual LAN and virtual Switch technologies, the isolation of network can be effectively provided for physical server hosts and the virtual machines running on them.


Different deployed application instances are isolated for security privacy control to deny any network access

End users to each application are assigned security roles to restrict illegal access (In development)

Figure 4. Virtual Resource Isolation.

- Virtual Machine Layer

A third type of isolation ensures that different virtual machines running on the same physical machine are isolated from each other. The virtual machine monitor should offer the isolation among the virtualized guest environments and the host systems both for paravirtualization and full system virtualization. The software running in a virtual machine also can not access or modify the software running in the virtual machine monitor or in a separate virtual machine.

B. *Resource Management and Scaling*

Resource usage typically includes storage, I/O bandwidth, CPU usage, and memory usage. The agile resource management capability provides the flexible mechanism to build the business agility. Different roles of service provider require different functions and qualities or resource scaling in the telecommunication field. For example, the service provider of backbone business process requires more virtual resources to achieve shorter response time and higher availability. And for the service provider of assistant services, less or shared resources are allocated to cut down the expenses. All these can be implemented easily by adjusting the virtual resources on cloud infrastructure.

The horizontal and vertical scaling mechanisms are provided as two dimensional methods to resolve different performance tuning problems. The horizontal scaling denotes the scaling-up and scaling-down capabilities, while the vertical scaling specifies the scaling-out and shrinking capabilities. Furthermore, one certain business service may raise more requirements on resources during the business growth to increase the quality of services. For example, a service needs better capability when high-end user quantity increases which can be achieved by clustering of middleware. In Service Storm, built-in virtual templates provide topologies including standalone application server and cluster for both normal and larger system throughput. Business user only needs to specify service quality options in Service Storm to increase scalability including the amount of the end users or whether to enable high availability. The corresponding virtual system resources will be associated automatically to satisfy the service quality requirement.

Besides the basic resource scaling, telecommunication operator and business partner can define more detailed sub resource templates for different roles throughout the entire service lifecycle. Topology of virtual machines or the hardware and software resources for each virtual machine are customized for roles as business modeler, developer, tester and production environment administrator, etc.

In a word, initial resource allocation and scaling functionalities in Service Storm are customized according to the user role of his group (service provider groups or service end user groups) or even in the entire service lifecycle.

C. *Capacity Planning*

Capacity planning is the process of predicting how much capacity is needed when future load levels will saturate the system and of determining the most cost-effective way of delaying system saturation as much as possible. The capacity planning is provided by Service Storm to cater to different roles. Telecommunication operator and business partner can define constraints on resources as initial resource allocation and upper limit for potential growth in future. Also they can define the service level agreement to describe the conditions for resource extension and recovery.

For a service provider, when the usage of virtual resources for certain user reaches a predefined threshold, cloud infrastructure will extend the virtual machine resources by allocating more memory and disk space or adding more nodes into cluster to balance workload. On the contrary, when the usage of virtual system is lower than predefined threshold for a certain time, idle nodes will be removed or memory will be drawn back.

Based on automatic scripts described in Section V, the virtual resources will be scaled horizontally or vertically, and managed automatically based on capacity planning of service provider but with uninterrupted services that will make a remarkable operating costs decreasing for SDP.

V. Automatic Deployment and Monitoring

A flexible and efficient SDP requires that the runtime environment including OS, network, middleware and applications can be instantiated and configured automatically to provide services for end users in the cloud environment. In Service Storm a deployment model is defined to provide elastic combination of virtual machine template, network resources and automatic scripts. Using this as a support, the whole solution deployment becomes configurable and the components can be configured freely and flexibly for deployment rather than the traditional single solution deployment. Also the aggregated views of business and infrastructure monitoring provided by Service Storm are of great benefit for the runtime management after deployment.

A. Configurable Deployment Model

In Service Storm, a complete deployment model is defined with the following elements to implement the automatic deployment.

- Virtual Machine Template

Certain OS and middleware are integrated in one virtual machine template (shown in Fig. 5) such as application server with OS as an application runtime. Moreover, configuration points for OS and middleware are defined for certain virtual machine template to provide the customization capabilities for service provider. There are three types of configuration points including specific software configuration points, operation system level configuration points, and underlying hardware resource properties.

- Network Resource

Service Storm provides centralized networking management including allocation of internal network address for virtual machine, address and port configurations for external services and the routing to internal virtual systems. End user can not manually specify this type of value in order to make the deployed environment controllable.

- Automatic Scripts

Automatic scripts are used to configure or reconfigure one resource for certain configuration point as mentioned above. Based on a unified model, Service Storm associates special automatic script in certain configuration step during deployment. All the configuration steps are organized and the corresponding automatic scripts are invoked, which avoids the service provider to take burden in caring of the underlying infrastructure implementation.

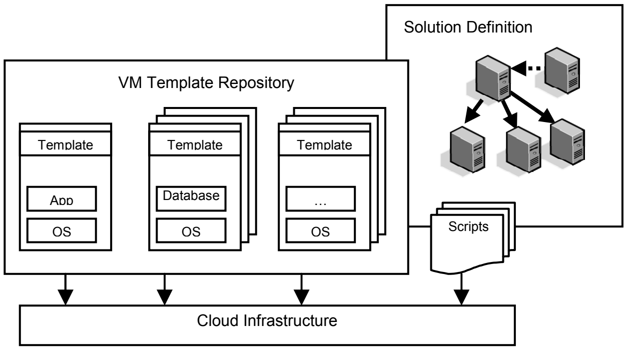


Figure 5. Application Runtime Automated Deployment

As shown in Fig. 5, when deployment is started, Service Storm selects virtual machine templates from template repository and associates necessary automatic configuration scripts.

B. Solution Deployment

Service Storm provides the ability of the whole solution deployment as well as one single component deployment and configuration. In telecommunication field, flexible business requires SDP to provide rapid and easy-to-use way to deploy various entire solutions automatically as well as independent resource component.

Service Storm builds in several typical middleware solution patterns such as application server cluster and high availability solution pattern, etc. These solution patterns include pre-built metadata of middleware and automatic scripts to regenerate solution. Also, user can customize these solution patterns by setting value on solution configuration points. For example, service provider can select the number of nodes in the cluster and whether to enable the high availability feature.

C. Aggregated Views of Business and Infrastructure Monitoring

Monitoring is another important function in SDP to get aware of the execution status including physical and software resources. In Service Storm, based on PaaS model, special resources are able to be selected for monitoring, such as response time of business process instance and virtual system resource status like CPU, memory and disk space, etc.

Both business and infrastructure resources monitoring are mashed up into a unified view for the user. It will provide a global and integrated status view for the whole SDP, so that it will help to reduce the risk of SDP operation and make it much easier to take quick response under emergent situation. The captured KPI of business process is presented into the Service Storm monitor portal as shown in Fig. 6.

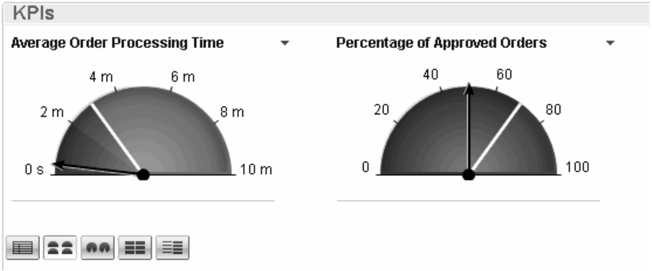


Figure 6. Monitored Business Process KPI.

VI. Sample Case: Mobile Workflow for Insurance Order Management

Service Storm has been implemented as a SDP with the above core technologies and mechanisms addressed. In this section, we take a mobile workflow for Insurance Order Management to evaluate the key features of Service Storm including: 1) Built-in Telecommunication Web Services; 2) Personalized Telecommunication Web Services; 3) Widgets based Mashup Capabilities; 4) Resource management for applications.

A. *Mobile Workflow*

JK Enterprise is a mobile communication company that provides self-service telecommunication Service Delivery Platform based on Service Storm for users to easily develop and deploy applications for their requirements by leveraging rich variability points of the solution. HI, an insurance company needs to extend its existing IT system with mobile workflow for its sales team to process order timely and efficiently. Fig. 7 illustrates the sample mobile workflow. When the workflow receives order request from sales (by Receive SMS web service), it needs to check the feasibility of the order (by Check Order Feasibility web service). After that, it notifies the sales manager to approve the order by short message (by Send SMS web service). After the approval of the sales manager (with a human task) finishes, the workflow notifies the approval result (by Send SMS web service) to the sales.

B. *Built-in and Personalized Web Services*

As described in Section III, in Service Storm, Receive SMS and Send SMS services are built-in web services. To better support the process flow HI IT manager needs to provide the Check Order Feasibility web service. In JKE SDP portal, HI IT manager registers the Check Order Feasibility web service to be referenced by the mobile workflow by uploading its WSDL documents to the Service Storm. In this way, the runtime instance of the web service could also be reached by Service Storm. Then the personalized web service, Check Order Feasibility, can be used in developing the mobile workflow.

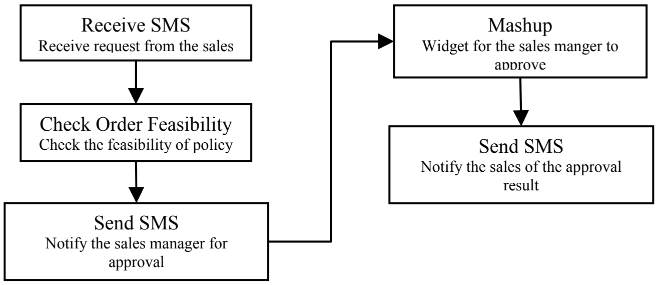

Receive SMS
Receive request from the sales
Check Order Feasibility
Check the feasibility of policy
Send SMS
Notify the sales manager for approval
Mashup
Widget for the sales manger to approve
Send SMS
Notify the sales of the approval result

Figure 7. Mobile Workflow for Insurance Order Management.

C. *Service Orchestration and Mashup*

HI sales manager rapidly defines the mobile workflow for her team using Service Assembler, by dragging and dropping the corresponding web services from the Palette of Service Assembler. To use a widget web page for the sales manager to approve the order, the Mashup capability is used in service orchestration. By means of this, the mobile workflow can pass and receive parameters from the predefined widget. After linking the member web services together, the mobile workflow application is automatically generated.

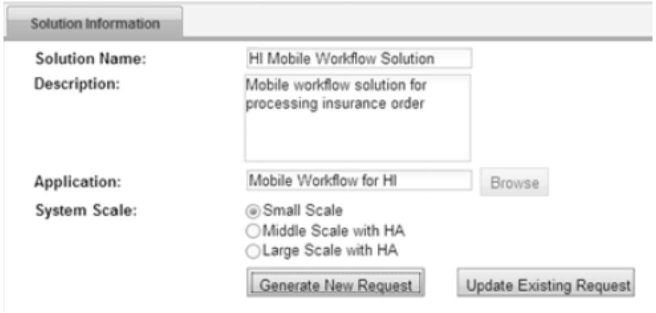
HI sales manager requests to deploy the application with JKE SDP portal. JKE SDP Administrator approves the request, and Service Storm automatically deploys the solution to the cloud infrastructure.

With the mobile workflow application, HI sales manager approves the order from her team members on a mobile phone. This confirms that the mobile workflow is successfully developed, deployed, and ready for use.

D. *Scaling the System*

With the number of sales and orders increased, HI sales manager applies for better computing capability. Service Storm provides one flexible mechanism to scale the runtime environment horizontally or vertically. HI sales manager may change the default “Small Scale” to “Middle Scale with HA” to apply for more computing resources. Service Storm extends the mobile workflow to a cluster topology automatically when the JKE SDP administrator approves the request. Fig. 8 provides the snapshot which is used to modify the scale of the solution topology to satisfy the business need.

Verified by this sample scenario, Service Storm can simplify the service orchestration by using the web-based Service Assembler and can generate and deploy the application automatically by using PaaS model. In addition, Service Storm can manage the computing resource with agility.


Solution Information

Solution Name: HI Mobile Workflow Solution

Description: Mobile workflow solution for processing insurance order

Application: Mobile Workflow for HI

System Scale:
● Small Scale
○ Middle Scale with HA
○ Large Scale with HA

Figure 8. Snapshot of Widget for Modifying the System Scale of applications

VII. FUTURE WORKS AND CONCLUSION

This article introduces Service Storm, a self-service telecommunication SDP, which mainly fulfils the requirements from the four roles in telecommunication SDP. It firstly provides a business user oriented application assembly model with the existing telecommunication services, external registered business services and the key Web 2.0 elements supported. Secondly, it provides an underlying elastic infrastructure and scaling capability for different business needs and implements the three-layer resource isolation, flexible resource management and effective capacity planning. Finally, it offers the automatic deployment to the runtime environment via a deployment model to demonstrate the capability of flexible and configurable solution deployment. Meanwhile, the aggregated views of monitoring both from business and infrastructure perspectives are illustrated. After verified in the JK enterprise and HI insurance company it proved that the Service Storm has provided more flexibility and conveniences to the individual and organization users and helped them save invest and cost. With the increasing amount of the end users the telecommunication operator and business partner of Service Storm can win more profit. In the next step, we will focus on the enrichment of the elements of Service Assembler and the security of the deployed solution environments.

References

[1] V. Goncalves and P. Ballon, "An Exploratory Analysis of Software as a Service and Platform as a Service Models for Mobile Operators", in *Proc. International Conference on Intelligence in Next Generation Networks (ICIN 2009)*, Oct 26-29, 2009.

L. Jin, P. Pan, C. Ying, J. H. Liu, and Q. M. Tian, "Rapid Service Creation Environment for Service Delivery Platform based on Service Templates", in *Proc. IFIP/IEEE International Symposium on Integrated Network Managemen*t, June 1-5, 2009.

[2] M. Hisatomi, K. Fukuda, M. Wilson, and T. Chujo, "Application of Service Delivery Platform for Supply Chain Management", in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Nov 30 -Dec 4, 2008.

[3] G. Lawton, "Developing Software Online with Platform-as-a-Service Technology", *Computer*, vol. 41, no. 6, pp. 13–15, June 2008.

[4] D. Mitchell, "Defining Platform-As-A-Service, or PaaS," 2008. [Online]. Available: http://blogs.bungeeconnect.com/2008/02/18/defining-platform-as-aservice-or-paas/.

[5] D. Merrill, "Mashups: The new breed of Web app," 2009. [Online]. Available: http://www.ibm.com/developerworks/xml/library/x-mashups.html.

[6] T. Anstett, F. Leymann, R. Mietzner, and S. Strauch, "Towards BPEL in the Cloud: exploiting different delivery models for the execution of business processes," in *Proc. IEEE World Congress on Services (SERVICES 2009*), July 06-10, 2009.