A TOGAF-based Method for Migrating Applications to Clouds

[1]Jyhjong Lin and [2]Chaoyu Lin
Department of Information Management
Ming Chuan University, Taoyuan, Taiwan
[1]jlin@mail.mcu.edu.tw; [2]lendy.lin@gmail.com

Abstract: - For the advances of Cloud Computing technologies in recent years, cloud applications have been popularity for their rich set of features. The advantages of cloud applications include that users can utilize them in a low cost-, threshold-, and risk-way; these applications can be quickly deployed on the clouds without duplication of work such that developers can focus on enhancing their QoS to improve core competitiveness. Therefore, their practical use on business with promising values can be expected. As such, cloud applications are recognized as a trend for the next generation of business applications, and hence how to migrate these on-premise applications to the clouds becomes a desired field in the literature. For this need, we present a migration method that employs the well-known TOGAF framework to support an effective migration of on-premise applications into the clouds. For illustration, the method is applied to the migration of a CSS application to its cloud version.

Key-Words: - cloud computing, migration method, on-premise application, cloud application, TOGAF

1 Introduction

For the advances of Cloud Computing technologies in recent years, their utilization on applications has been most widely addressed due to the rich set of features in such cloud applications. These applications can be quickly deployed on the clouds that make users utilize them in a low cost-, threshold-, and risk-way. Therefore, their practical use on business can be expected as a trend for the next generation of business applications.

In terms of the architecture for on-premise applications (e.g., web information systems), client-server or distributed patterns were most commonly used in the past decades; almost all existing applications were constructed using these styles. However, as stated above, cloud applications have been recognized as a trend for the next generation of business applications; how to migrate these on-premise applications to the clouds becomes a desired field. For this need, some discussions about the migration work have been presented in [1-12]. In general, these discussions have clarified some important issues about the migration and then proposed various approaches for addressing these issues. However, some shortcomings can still be found: (1) few considerations except for [2] are addressed on the architecture of both on-premise applications and clouds; (2) few sentences are stated about the cloud requirements for the distributed styles of application architectures/profiles and (3) few words are stated about the employment of BPM [13] for enhancing the effectiveness of the migration.

Such shortcomings, in our opinion, should not be negligible since a well-considered process is critical for directing the migration of the many on-premise applications in a systematic and managed manner. Therefore, we present herein a method for directing the migration process. The method is based on the TOGAF framework [14] that (1) starts from the identification of the architecture and profile of the application, and then the discussion of the requirements for clouds via the BPM lifecycle, (2) through the identification and selection of the clouds whose service models satisfy the cloud requirements, and (3) finally ends at the deployment of the application into selected clouds. For illustration, the method is applied to the migration of a Customer Support System (CSS) [15,16] to its cloud version that emphasizes on both of collecting customer information (i.e., knowledge about/from customers) for enterprises and reversely delivering services information from enterprises to benefit customers.

This paper is organized as follows. Section 2 presents the migration method that encompasses the eight phases of the TOGAF framework. The method is illustrated in Section 3 by applying it to the migration of a CSS on to its cloud version. Finally, Section 4 has the conclusions and future work.

2 The Migration Method

2.1 Phase 1 (TOGAF phases A-D): Baseline Architecture Identification

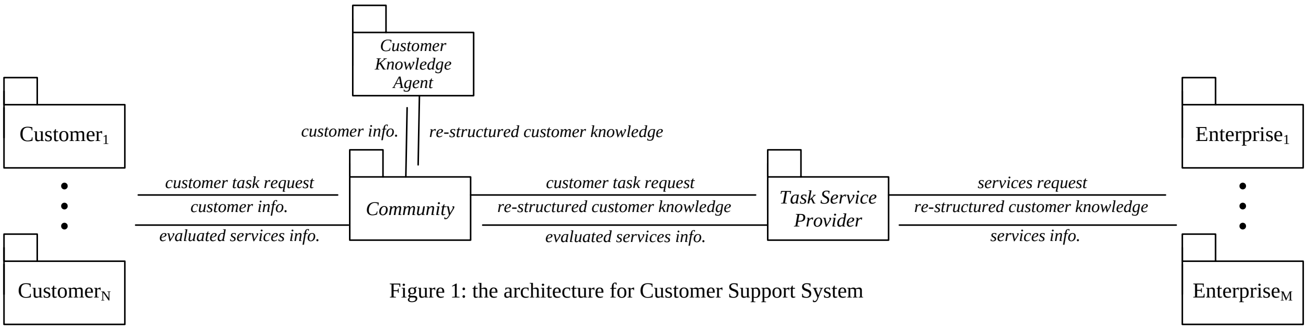

Figure 1: the architecture for Customer Support System

Based the TOGAF framework, the method starts from the identification of the architecture and profile of the on-premise application (i.e., the baseline architecture). As an example, Figure 1 shows the architecture of a CSS where

1. It is a 4-layer of collaborative components where ***Customers*** interact with ***Enterprises*** via three intermediaries: ***Community***, ***Customer Knowledge Agent***, and ***Task Service Provider***.
2. ***Community*** helps ***Customers*** share information about their desired tasks (e.g., buy/rent services from ***Enterprises***).
3. ***Customer Knowledge Agent*** collects Customer knowledge to help ***Enterprises*** catch customer needs.
4. ***Enterprises*** provide services information about the desired tasks to help ***Customers*** make recognition and comparisons.

With the application architecture, the next is to capture its profile to size the application. In general, the application profile should be collected for at least 10 to 14 days to allow figuring out any variances in daily or weekly usage patterns. There are two kinds of profile data about the application: (1) use data about its executions (e.g., CPU, memory, storage, I/O, and network uses); and (2) action data about its users (e.g., # of active users, request rates, transaction rates, and request/ transaction latencies).

2.2 Phase 2 (TOGAF phases A-D): Target (Clouds) Architecture Identification

With the baseline architecture and profile, the next is to identify the cloud requirements for satisfying its target cloud-deployed ones. This can in general be achieved by imposing the BPM lifecycle [13] for identifying the limits of the baselines and the enhancements of the target ones via its Strategy, Design, Execution, and Control lifecycle phases. The identified requirements may include (1) for architectural components, their deployments on the configuration elements in selected clouds are required to support enhanced functional purposes; and (2) for execution profiles, their QoSs in selected clouds are required to support enhanced non-functional purposes such as customized user interfaces and access modes, performance, reliability, security, and scalability.

For the CSS example, its five components may require respective deployments on various cloud environments to support its architecture and profile requirements. Further, for its purposes of collecting knowledge from and delivering information to customers, it may require such QoSs from these deployed clouds as customized user interfaces and information accesses, and reliability of the services information.

2.3 Phase 3 (TOGAF phase E): Candidate Clouds Identification

Based on the TOGAF phase E (i.e., opportunities & solutions), the method continues to identify the candidate clouds whose configurations and service models (i.e., SaaS or PaaS or IaaS) satisfy the cloud requirements. For this, therefore, it is good to consider all of the available environments that provide either of the following service models:

1. In SaaS model, the services can replace those in the application where specific QoS features are required for ensuring their replacement such as Service-Level-Agreements (SLAs), compatibility of services, and portability of data/access control.
2. In PaaS model, the cloud provides platform services on which the application can be deployed under such QoS features as SLAs, application deployment, compatibility of services, and portability of data/access control.
3. In IaaS model, the cloud provides infrastructure services like servers, storages, and networks where the application and its residual platforms can use under such QoS features as SLAs,

application deployment, compatibility of services, and portability of data/access control.

As a result, some clouds may be identified to satisfy the cloud requirements and then become the candidates for the migration. For example, Figure 2 shows the possible candidate clouds for the CSS where as an instance for ***Community***, some IaaS clouds are identified as candidates since its services are expected to be provided by some infrastructures that support well the storage and manipulation capabilities for the information sharing among ***Customers***.

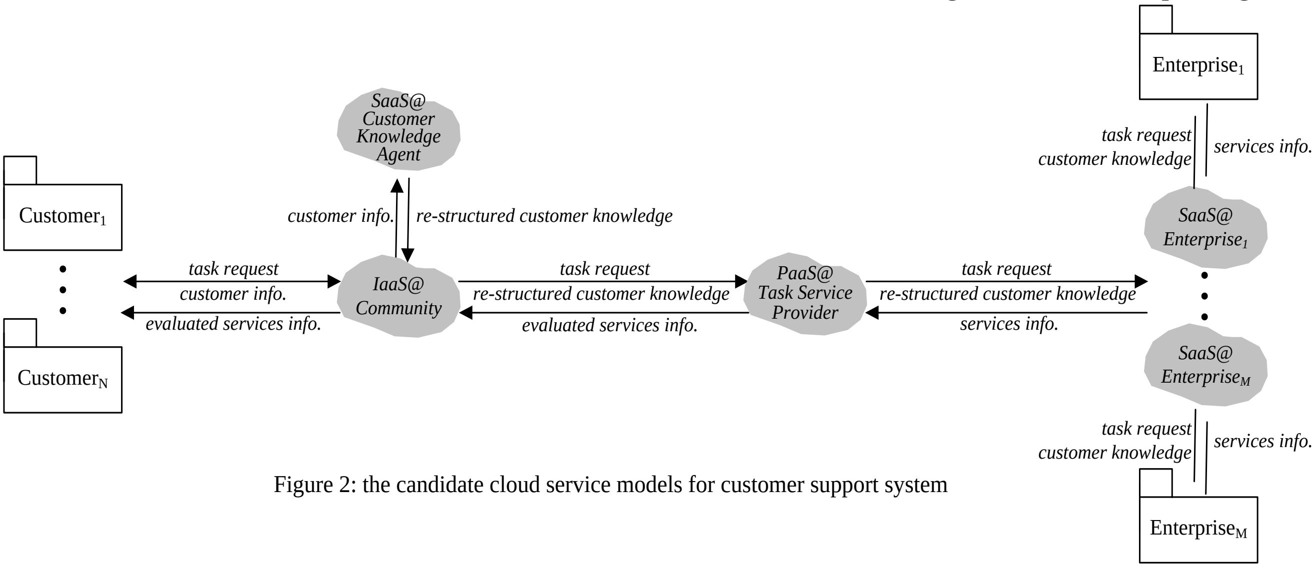

Figure 2: the candidate cloud service models for customer support system

2.4 Phase 4 (TOGAF phase E): Clouds Selection

With the candidate clouds identified, the next is to select from them the clouds to be migrated. In general, the selection can be achieved by some evaluation criteria (e.g., the QoS features identified above) for satisfying the cloud requirements. For example, based on the above QoS features, a candidate whose service models have the best assessments may be selected as the targeted cloud.

2.5 Phase 5 (TOGAF phase F): Clouds Migration Plan

Based on the TOGAF phase F (i.e., migration planning), the method continues to specify the plan about the activities involved in the application migration to the targeted clouds identified above. In general, the activities include

1. Deploy the application components on the configuration elements in respective clouds.
2. Deploy the interaction mechanisms among application components on the inter/intra-cloud interaction solutions over/in respective clouds.
3. Refactor any deployed components for satisfying the usage and user actions requirements such as customized user interfaces and access modes, performance, reliability, security, and scalability.

2.6 Phase 6 (TOGAF phases G-H): Clouds Migration and Testing

Based on the TOGAF phases G & H (i.e., implementation governance & change management), the method continues to realize the migration to the selected clouds in accordance with the migration plan. In addition, the migration is

3 Migration of the CSS to Clouds

3.1 Phase 1 (TOGAF phases A-D): Baseline Architecture of the CSS

As shown in Figure 1, the CSS is a four-layer of collaborative components where ***Customers*** are interacting with ***Enterprises*** via three intermediaries: ***Community***, ***Customer Knowledge Agent***, and ***Task Service Provider***. After then, it also needs to clarify the constituents in each component. As an example, ***Community*** is organized for ***Customers*** to share information about their desired tasks (e.g., buy or rent services from ***Enterprises***). In addition, it is also responsible for forwarding the shared info. to ***Customer Knowledge Agent*** for re-structuring into specific styles of knowledge (i.e., knowledge of customers) [19]. It then sends the knowledge to ***Task Service Provider*** for forwarding to ***Enterprises*** to catch their needs (e.g., provide services satisfying their desired tasks). Finally, it also cooperates with ***Task Service Provider*** to receive services info. relevant to these requests for ***Customers*** to make recognition and comparisons.

In summary, these requirements for ***Community*** can be described as follows:

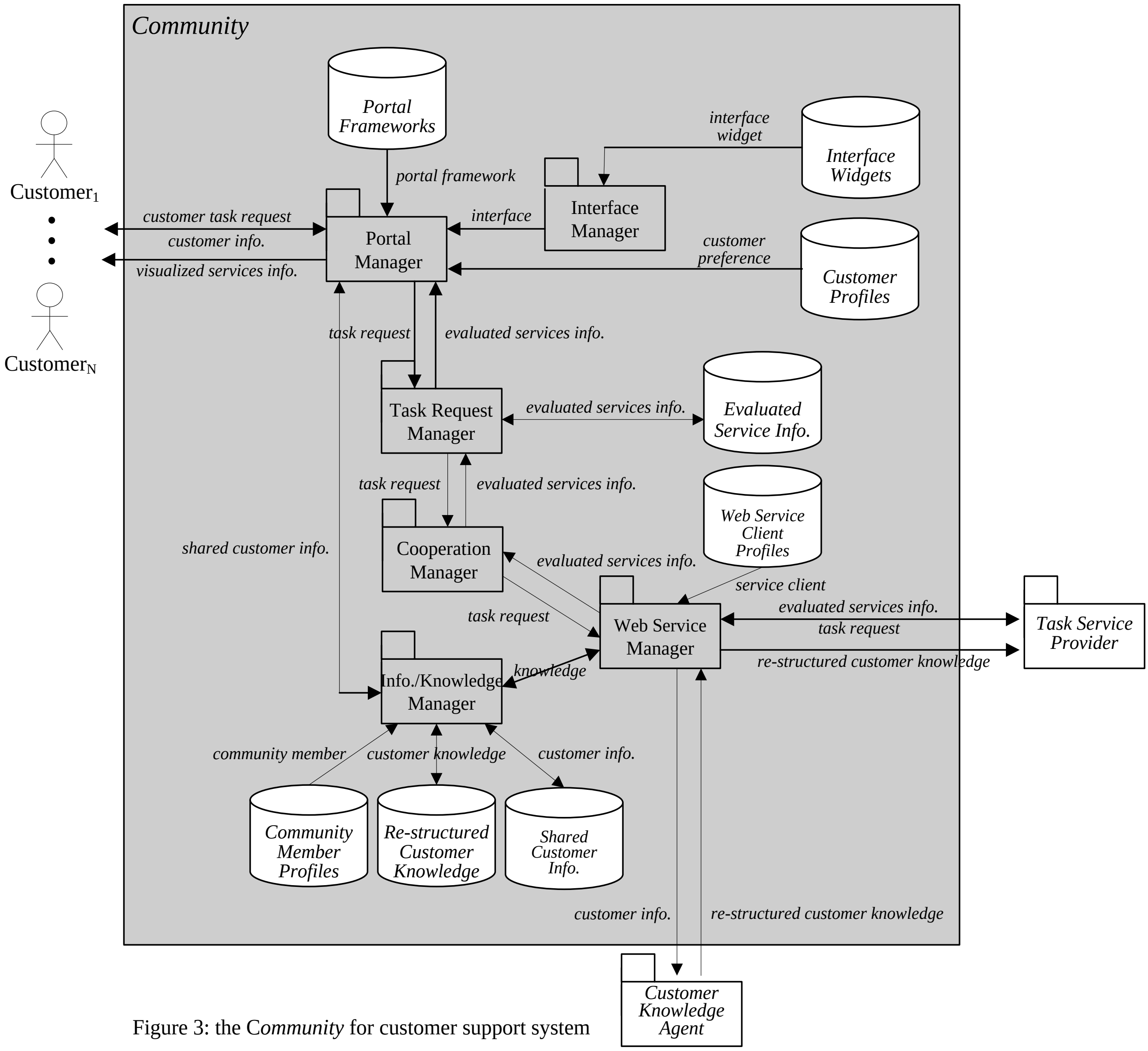

Figure 3: the C*ommunity* for customer support system

1. **Share customer information** that helps on the information sharing among Customer$_{1...N}$.
2. **Process shared information** that forwards the shared info. to ***Customer_Knowledge_Agent*** for re-structuring into knowledge, and then sends the knowledge to ***Task_Service_Provider***.
3. **Process task request** that receives task requests from and return evaluated information about the task-relevant services to Customer$_{1...N}$.
4. **Cooperate with *Task_Service_Provider*** that receives the evaluated information about task-relevant services.
5. **Present services information** that provides Customer$_{1...N}$ with rich user interface controls for visualizing the task-relevant services information from ***Task_Service_Provider***.

Based on the above requirements for ***Community***, Figure 3 shows its five constituents that realize these requirements. In particular, a 'Interface Manager' is imposed to realize the customization/personalization of user interfaces for Customer$_{1...N}$ where *Customer Profiles* are used to determine which interface components are preferred by them; in addition, with such customized/personalized user interfaces, their containing interface widgets may withhold by a 'Portal Manager' visualized information for sharing or about task-relevant services to form customized/ personalized portals (under available *Portal Frameworks*) that deliver to Customer$_{1...N}$ their desired information according to their interactive requirements. Further, the 'Info./Knowledge

Manager' accesses *Community Member Profiles* and *Shared Customer Info.* to help on information sharing among interested customers; *Shared ustomer Info.* is also retrieved for re-structuring into knowledge by ***Customer_Knowledge_Agent***. In addition, the 'Task Request Manager' forwards task requests from $\text{Customer}_{1\ldots N}$ to the 'Cooperation Manager' that cooperates with ***Task_Service_ Provider*** to receive evaluated information about these requests; the evaluated information is then visualized and returned to $\text{Customer}_{1\ldots N}$ through the 'Portal Manager'. Finally, the 'Web Service Manager' is responsible for inter-operating with the two external architectural components through *Web Service Client* APIs for accessing those remote services provided from the two components.

3.2 Phase 2 (TOGAF phases A-D): Target Architecture of the CSS

The second step is to identify the cloud requirements for the CSS based on its architecture and profile. Initially, considering its five distributed components, various cloud environments may be required for their respective deployments on the prospective configuration elements in these clouds to support its functional/non-functional purposes. Further, for its purposes of collecting customer knowledge for enterprises and delivering services information to benefit customers, it may require such QoS about its usage and user actions from these deployed clouds as customized user interfaces and access modes, performance, reliability, security, and scalability.

3.3 Phase 3 (TOGAF phase E): Candidate Clouds for the CSS

The third step is to identify the candidate clouds whose configurations and services (i.e., service models – SaaS or PaaS or IaaS – provided in clouds) satisfy the cloud requirements for the CSS. For this, therefore, it is common to consider all of the cloud environments available on-line whose service models may satisfy the cloud requirements. As one may conceive, there is usually more than one cloud that satisfies the requirements; such clouds hence become the candidates from which specific ones are then selected for the realization of the migration. As an example, Figure 2 shows some candidate clouds identified for the CSS where

1. For its components, ***Customer_Knowledge_ Agent*** and ***Enterprises***, some SaaS clouds are specifically identified since their requirements are expected to be satisfied by those collaborative agents/enterprises that host these SaaS services.
2. For ***Task_Service_Provider***, some PaaS clouds are identified since its requirements are expected to be satisfied by a PaaS cloud whose platform supports well the inter-cloud capability

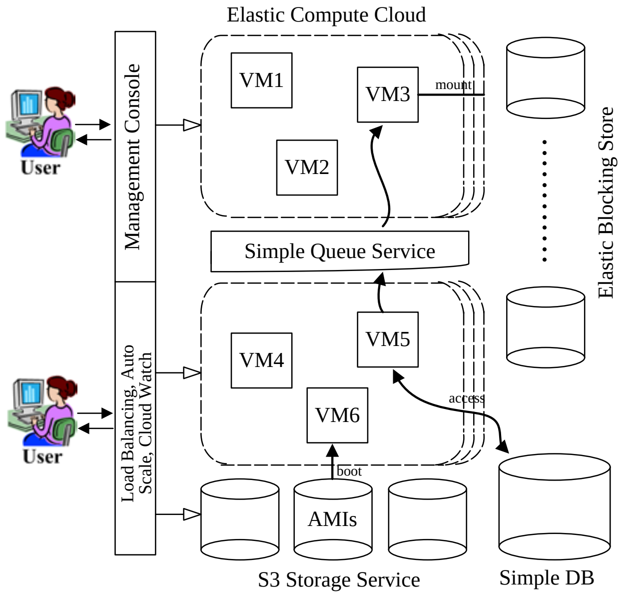

Figure 4 : the Amazon EC2 IaaS cloud

for the cooperation with multiple enterprises (***Enterprises***) to forward knowledge/ receive information and the analysis capability for the evaluation of the received services information into a comparative model for customers to make recognition and comparisons.
3. For ***Community***, some IaaS clouds are identified since its requirements are expected to be satisfied by a IaaS cloud whose infrastructure supports well the storage and manipulation capability for the large amount of information shared among ***Customers*** and the inter-cloud capability for the cooperation with ***Customer_ Knowledge_Agent*** to re-structure the shared info. into specific styles of customer knowledge.

3.4 Phase 4 (TOGAF phase E): Clouds Selection for the CSS

With the candidate clouds identified, the next is to select from them the clouds to be migrated. In general, the selection can be achieved by the QoS features identified above for satisfying the cloud requirements. As an illustration for the CSS:

1. Those SaaS clouds hosted by the collaborative agents/enterprises are undoubtedly selected for the migration of its ***Customer_Knowledge_ Agent*** and ***Enterprises*** components.

2. Among such available PaaS clouds as Google GAE and Microsoft Azure, the GAE may be selected for the migration of its ***Task_Service_Provider*** component due to its well-known inter-cloud and analysis capabilities for the cooperation and evaluation requirements of the component.

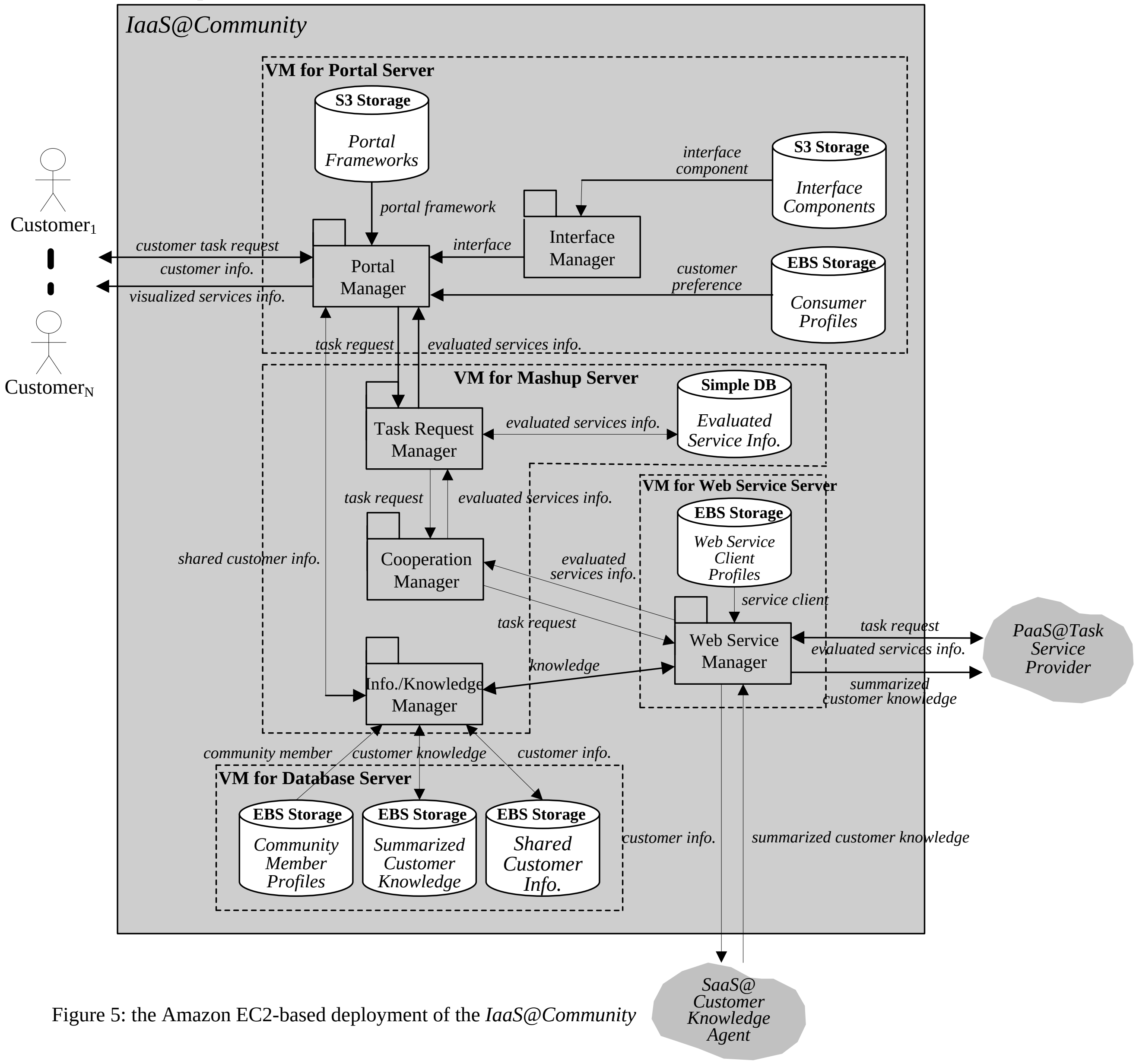

Figure 5: the Amazon EC2-based deployment of the *IaaS@Community*

3. Among such available IaaS clouds as Google GCE [17] and Amazon EC2 [18], the EC2 as shown in Figure 4 may be selected for the migration of its ***Community*** component due to its well-known storage and inter-cloud capabilities for the information sharing among customers and the cooperation with other components.

3.5 Phase 5 (TOGAF phase F): Clouds Migration Plan for the CSS

After determining the clouds selection, the plan about the activities involved in the application migration to these clouds can then be specified. In general, the activities include (1) deploying the CSS components on the configuration elements in respective clouds; (2) deploying the interaction mechanisms among CSS components on the inter/intra-cloud interaction solutions over/in respective clouds; and (3) refactoring/restructuring any deployed components for satisfying the usage and user actions requirements. As an illustration for ***Community*** to be migrated to the EC2 IaaS cloud, denoted as ***IaaS@Community***, Figure 5 shows the deployment of its five constituents on the four Virtual Machines (VMs) in EC2 and each one may use some storage services such as S3 storage, EBS storage, and Simple DB.

It should be noticed that with the aforementioned deployment, refactoring the CSS constituents to satisfy the usage and user actions requirements for the CSS then needs to be specifically concerned. For example, for satisfying the requirements for customized user interfaces and access modes, the 'Interface Manager' constituent of the ***Community*** component (migrated into the EC 2 cloud) may need to be refactored to fit into any interface/access constraints imposed on the EC2 cloud where its *Customer Profiles* data store may also need to be restructured for complying with any structure/style constraints on the cloud.

3.6 Phase 6 (TOGAF phases G-H): Clouds Migration and Testing for the CSS

The last step is to realize the migration of the CSS into selected or constructed clouds in accordance with the migration or construction plan identified above. As in usual, testing of the migration proceeds in accordance with the activities involved in the migration process.

4 Conclusion

In this paper, we present a method for directing the migration of on-premise applications to selected clouds. In particular, the method employs such well-known constructs as BPM lifecycle and TOGAF framework to support an effective migration of the on-premise applications into the clouds. For illustration, the method is applied to the migration of a CSS application to its cloud version that takes advantages of cloud configurations and services to help not only enterprises collect customer knowledge but also customers receive services information in a low cost-, threshold-, and risk-way .

Since cloud applications have been recognized in recent years as a trend for the next generation of business applications, how to migrate the many existing on-premise applications to the clouds for taking advantage of cloud applications has thus become a desired field in the literature. For this need, some discussions about the migration work have been presented that clarify some important issues about the migration and then propose various approaches for addressing these issues. However, some shortcomings can still be found that, in our opinion, should not be negligible since a well-considered process is critical for directing the migration of the many on-premise applications in a systematic and managed manner. Therefore, the method presented herein provides an effort on this need.

As our future work, we will continue to explore the real migrations of the CSS applications to the clouds where the most popular clouds such as Google GAE and Amazon EC2 will be used as the deployed platforms. In fact, with the TOGAF-based phases that gradually identify the application/ cloud architectures and features and then conduct the deployment on the most suitable clouds, the quality of the migrated applications can be expected. Finally, for the most critical issues in the migration such as identifying available clouds and then selecting desired ones from them, we will also study explicit formal approaches such as semantic ontologies that support the identification and selection from available clouds in a systematic and managed manner.

Acknowledgements

The work in this paper is supported by the Ministry of Science and Technology in Taiwan under Grant MOST 103-2410-H-130 -015.

References:

[1] M. A. Chauhan, M. A. Babar, "Towards Process Support for Migrating Applications to Cloud Computing," Proc. of 2012 International Conference on Cloud Computing and Service Computing, 2012, pp. 80-87.

[2] W. Abderrahim and Z. Choukair, "A frame-work architecture based model for cloud computing adaptive migration," Proc. of IEEE 2014 Global Information Infrastructure and Networking Symposium (GIIS), 2004, pp. 1-6.

[3] U. Banerjee, "Five Examples of Migration to Cloud," May 2010, http://setandbma.wordpress.com/2010/05/28/5-examples-of-migration-to-cloud/

[4] R. Blaisdell, "How to Plan Your Migration to the Cloud," Nov. 2011, http://www.rickscloud.com/how-to-plan-your-migration-to-the-cloud/.

[5] N. Carr, "Best Practices for Achieving Migration to a Cloud Model," March 2011, http://blogs.dlt.com/practices-achieving-migration-cloud-model/.

[6] Francisco Garcia-Sanchez, et al., "SITIO - Semantic Business Processes based on SaaS and Cloud Computing," Proc. of 9th WSEAS International Conference on E-Activities (E-Activities '10) and 9th WEAS International

Conference on Information Security and Privacy (ISP ’10), 2010, pp. 130-135.
[7] Francisco Garcia-Sanchez, et al., “Adding Semantics to SaaS and Cloud Computing,” *WSEAS Trans. on Computers*, vol. 9, no. 1, Jan. 2010, pp. 154-163.
[8] A. Gunka, S. Seycek and H. Kühn, “Moving an Application to the Cloud: An Evolutionary Approach,” Proc. of 2013 International Workshop on Multi-Cloud Applications and Federated Clouds, ACM, 2013, pp. 35-42.
[9] A. Koschel and S. Hofmann, “Standardization in Cloud Computing,” Proc. of 8th WSEAS International Conference on Computer Engineering and Applications (CEA ’14), 2014, pp. 199-204.
[10] MomentumSI, “Cloud Migration,” 2013, http://cloudmigration.com/.
[11] S. Zardari, R. Bahsoon, “Cloud Adoption: A Goal-Oriented Requirements Engineering Approach,” Proc. of 2nd International Workshop on Software Engineering for Cloud Computing, ACM, 2011, pp. 29-35.
[12] R. Zota and A. Fratila, “Toward the Selection of an Enterprise Architecture Model for a Cloud Environment,” Proc. of 11th RoEduNet International Conference, IEEE Computer Society, 2013, pp. 149-154.
[13] N. Palmer, “What is BPM,” 2014, http://what-is-bpm.com/
[14] The Open Group Architecture Framework (TOGAF), http://www.opengroup.org/togaf/
[15] J. Lin, “An Object-Oriented Development Method for Consumer Support Systems,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 19, no. 7, Sep. 2009, pp. 933-960.
[16] L. Orman, “Consumer Support Systems,” *Communications of the ACM*, vol. 50, no. 4, April 2007, pp. 49-54.
[17] Google Compute Engine (GCE), https://cloud.google.com/compute/
[18] Amazon EC2, http://aws.amazon.com/ec2/
[19] B. Mobasher, et al., “Five Styles of Customer Knowledge Management, and How Smart Companies Use Them To Create Value”, *European Management Journal*, vol. 20. no. 5, Oct. 2002, pp. 459-469.