

Chapter 14

Healthcare Application

Francesco D'andria and Roi Sucasas Font

14.1 Introduction

This chapter presents a real life case based on a case study from Atos which uses the MODAClouds framework to manage the design, deployment and governance of a telemedicine solution in a hybrid multi-Cloud environment.

The Atos eHealth telemedicine solution is a software application, based on the state-of-the-art in ICT that aims at developing an innovative and integrated solution for the general management of patients suffering from dementia. It provides an integrated online clinical, educational, and social network to support dementia sufferers and also their caregivers. Based on a set of monitoring parameters and measuring scales, this solution aims to early detect symptoms that predict decline, avoid emergencies and secondary effects and, ultimately, prolong the period that patients can remain safely cared at home, no matter where it is located. There are various stakeholders involved in this scenario that would benefit from the system capabilities offered by the eHealth application:

Patients and their caregivers:

- Access to services, like videos or games, recommended by clinicians or experts.
- Collect and register data and measurements (blood pressure, weight, activity levels, questionnaires, etc.).
- Management of warnings or requests sent to the clinicians.

R.S. Font (✉)

ATOS Spain SA, Subida al Mayorazgo, 24B Planta 1, 38110 Santa Cruz de Tenerife, Spain
e-mail: roi.sucasas@atos.net

F. D'andria

ATOS Spain SA, Av. Diagonal, 200, 08011 Barcelona, Spain
e-mail: francesco.dandria@atos.net

© The Author(s) 2017

E. Di Nitto et al. (eds.), *Model-Driven Development and Operation of Multi-Cloud Applications*, PoliMI SpringerBriefs,
DOI 10.1007/978-3-319-46031-4_14

133

- Improve awareness on the use of their sensitive data, like the patients monitoring parameters and the patients medication follow-up and drug adverse events.

Clinicians and Health System (organization of people, institutions and resources to deliver health care)

- Continuous monitoring and follow-up of the patients
- Management and assignment of tasks and questionnaires
- Services to meet the health needs of target populations
- Improve workload of assistance teams:
- Institutions/specialists dynamically added and removed on demand
- Allocation/De-allocation of Cloud resources depending on the workload.
- Rapid elasticity, i.e., the network can respond rapidly and automatically to changes in demand from particular doctor/specialist
- Improve access to and participation in the Knowledge and Information Society for citizens
- help monitoring of risks like: data breaches/inappropriate access, disruption of service and data)

Originally a monolithic web application, this new telemedicine solution has been re-designed for a multi-Cloud environment. This software solution that consists of two main software blocks: a multi-Cloud server-side block and a client-side block. The server-side block consists of a database and two server applications. All of them can be deployed in different multi-Cloud scenarios alternatives, like private Clouds, public Clouds or hybrid scenarios. The client-side block consists of a desktop application (used by the patients and their caregivers) that connects to one of the server applications.

14.2 EHealth Cloud Solution: Why to Deploy It in a Multi-Cloud Environment?

There are many real and potential benefits of a multi-Cloud based telemedicine solution. This section describes these benefits (also the cons and risks associated to this approach) from two different points of view: the application (including designers, developers, and administrators), and the final users (the patients and their caregivers, and the doctors).

Because of the number of users of this application, the amount of database transactions, and also the load and traffic conditions that can vary significantly, it is not trivial to accurately estimate the resources required by this solution. To be able to scale the resources on demand is one of the main advantages of a multi-Cloud approach. The Cloud providers allow us to administer these challenges in an easy way, also reducing costs and management times. This reduced technical maintenance (that also applies to the deployment and the backup and recovery tasks) is also a feature that will be very useful in this case study. Another important characteristic of the eHealth solution is the capability of integrating and connecting multiple third party tools or services.

This will allow us to scale the functionalities and features offered at a much lower cost compared with an own development and a later deployment in a private / public infrastructure. For example the eHealth GUI component could be deployed in a public PaaS or IaaS, making use of third party services offered by these providers, like email delivery services, monitoring services etc. This approach would save us money and time, and would also allow an easier growth of the application's functionalities and capabilities in the future. The developers won't need to create new applications from scratch with the same functionalities offered by these third party services. A multi-Cloud environment also offers us flexibility, for example it will allow us to easily move components from one provider to another when needed. Also the final users of this solution can benefit from it. The deployment of the eHealth solution in a multi-Cloud environment will make it possible to access it from everywhere. Patients and caregivers will connect from home with clinicians. This will facilitate the home health monitoring and follow-up of these users by the clinicians. Patients and their caregivers will send them different kind of measurements (activity, weight, blood pressure, etc.) and tests results, so that doctors can make a diagnostic with all these data. Also the eHealth solution will analyze these data in order to detect early deterioration symptoms. Patients with dementia won't need to go to hospital so frequently, and doctors can make a better and individualized follow up of them. It is expected that this set of tools and features will reduce the workload of these clinicians, and will also reduce the costs of the health system. And from the point of view of the patients and caregivers, it's also expected that with this direct and frequent connection (and monitoring) with clinicians, their quality of life will be better.

14.3 Risks and Problems

There are some risks associated with this approach. The most important of these risks may be the data and privacy protection and the different legislations of each country associated with the management of this data. The preservation of this medical information and the confidentiality is a major challenge in all telemedicine systems. Another very common risk in Cloud computing is the vendor lock-in. PaaS and IaaS providers are heterogeneous and the provided features are often incompatible. This can be a serious inconvenience in such a multi-Cloud approach. These providers present another problem: which ones are the best for our case study? Not only may the prices of the different providers vary, but also their availability and the offered services. In order to deploy all the eHealth components in a multi-Cloud environment, a great knowledge of the available Cloud providers will be needed. Other risks associated to this kind of telemedicine solutions are the availability and reliability of their services. In emergency cases, the access to these services can mean the difference between life and death. In particular, in those emergencies where a fast medical response time is needed, the availability of these services can be critical. The database and other components of the eHealth solution shall be accessible without interruption 24×7 . This could become a problem because it's very important to continue monitoring not only the health of these business-critical applications but also

their performance. In order to monitor these properties and also the performance of each of the components, it will be needed to use some kind of monitoring application that could react to issues related to them. Moreover, to define a set of rules, constraints or SLAs for the eHealth application components should be a mandatory requirement.

14.4 EHealth and MODAClouds: The Story

At this point, Juan, the responsible (administrator, designer and developer) of the eHealth solution, wants to use the MODAClouds platform to be able to design, deploy, monitor and manage this telemedicine application on a hybrid PaaS environment. Why MODAClouds? He expects to take advantage of all the Cloud environment benefits described before and at the same time he expects to solve most of the problems and risks of such approach. MODAClouds offers a set of tools for the design and run time. On one hand the design-time tools offer the capability of designing a Cloud agnostic model of the solution, the capability of defining the QoS/SLA rules that will be used during the run-time, and also the capability of avoiding the vendor lock-in by providing a list of all available Cloud providers, that match the eHealth technical and business requirements, where this solution could be deployed. On the other hand the run-time tools will monitor the different solution components and will react properly to these monitoring values by scaling the components up or down, or by migrating the components from one provider to another. This is the theory. Juan will prove it soon. Juan will use two different PaaS providers accounts: one in Pivotal (a public PaaS provider based on Cloud Foundry) and another in a private Cloud Foundry server with few resources available. He thinks that hosting the database and the web services application in this private PaaS will make it possible to handle some of the data and privacy protection risks mentioned before. It will allow him to restrict and control in an efficient way the access to the data that is stored there. Only the web services application (the application that handles the connections with the database) will be able to access the content of this database. But because he thinks that this is not enough, the critical data will also be encrypted by this web service application. This way the same application that handles the access to the database will be the responsible of encrypting the data. This application is also responsible of managing the roles and permissions of the users that want to access the content of the database by accepting or refusing the requests depending on the users roles. He also thinks that deploying the other eHealth component (Web GUI application) in the private server could be the best option for a first moment, where only a limited set of users will make use of the eHealth solution. Once the application starts to grow in number of users and starts to have difficulties in handling the incoming traffic, he could migrate some components from the private Cloud to a public Cloud. But instead of migrating the components manually, Juan will use MODAClouds to define a set of rules and SLAs so that the MODAClouds run-time components can do that in an automatic way. In order to achieve all this, Juan will follow the steps described below.

Cloud agnostic solution design and Cloud provider selection

First, Juan uses Creator4Clouds to do a Cloud agnostic model of the eHealth solution. This is the model of the components that will be deployed in the Cloud without specifying where and how they will be deployed. These components are the following: the main database, the web services application and the Web GUI application. For each of these components he defines the provided and required interfaces, which are needed to define the connections between each component and are also needed to define the methods or components that will be monitored. After refining all these models, the next step is to get a list of all available PaaS providers that match the functional and business requirements. Juan wants to know which the best options for this solution are. On one hand he needs to host in a private Cloud the database and the web services application. This way he can preserve the data and privacy protection. The other component, the web GUI application (called eHealth-gui in the models), can be deployed either in a private or in a public Cloud. Juan uses the Venues 4Clouds tool in order to get all the providers that match his business and functional requirements. As a result of this he gets Pivotal / Cloud Foundry as the best choice (in terms of costs, services offered, etc.).

Now Juan wants to define how the MODAClouds run-time tools will behave once the application components are deployed in the selected provider.

Modelling of the QoS and SLAs

In order to do that, Juan continues using the Creator 4Clouds tool to define the Quality of Service constraints and penalties that will be used by the MODAClouds run-time tools. He wants to monitor the performance of the application components. To do that he defines two constraints: Response Time (to check the response time of some operations) and Throughput (to check requests per second). These two constraints will be used to monitor the traffic and performance of the deployed web applications.

He uses the interfaces defined before to associate these constraints to the interface or to some of its methods. He decides that if there's too much traffic (by specifying a value in the Throughput constraint), the SLA tool should migrate the Web GUI application to a public PaaS. To define this behavior, he associates this constraint to a penalty (called Migrate Penalty). This way, once the number of users grows, both applications will have more resources to be scaled up. Also the SLA tool would send events (i.e. emails) to Juan if some other SLA violation occurs. This migration penalty or behavior could also be defined by using other MODAClouds tools. Creator 4Clouds will then transform these constraints and penalties models to generate the monitoring rules and penalties used by the Tower 4Clouds and the SLA tool during the run-time.

The data collector library

Juan will need to connect the deployed eHealth components with Tower 4Clouds, the MODAClouds monitoring tool. First Juan imports the data collector library (a remote component of the Tower 4Clouds tool) in the application, then he configures it, and finally he generates the packed files that will be deployed (.war files). But before deploying them, Juan needs to finish the models.

Refining the models

Before deploying the eHealth application, Juan needs to do a few more things. On one hand he associates the model components with the ‘physical’ applications. In the case of the eHealth cloud components, the ‘physical’ applications are packed in two .war files: one for each of the java web server applications. And on the other hand he specifies the cloud providers where these components will be deployed. At this point the eHealth application components are ready to be deployed.

EHealth deployment

Once Juan has modelled all the components, including all the QoS constraints and SLAs, its time to deploy them in the private Cloud Foundry server. The Creator 4Clouds tool offers the option of generating different models for different providers. But for now, Juan only creates one model for deploying the application components in the private Cloud Foundry. To do that, Creator 4Clouds connects with CloudML4Clouds, which is the responsible of deploying the application components in the selected providers. Juan selects the deploy option and waits until the application is deployed and ready. Then it starts the application execution and makes it available to his customers.

Run-time tools: Check the status of the application components

Juan accesses the private Cloud Foundry to see that the applications are deployed and running as expected. Then he connects with the Tower 4Clouds and SLA web tools to see if they are correctly connected with the application components. He does a few tests with the eHealth component and checks that all of them are working fine. After Juan checks that the eHealth solution cloud components are deployed and running in the private Cloud Foundry server, he tells his bosses that the application is ready to be used.

14.5 Conclusions

As we could see in this chapter, a multi-cloud approach offers several new possibilities and advantages for telemedicine applications, but it also presents some risks and disadvantages. As was shown in the “real life case”, by using the MODAClouds ecosystem, our main character was able of taken benefit of the advantages of such approach, and at the same time, he could avoid most of the risks and disadvantages associated to a multi-cloud deployment.

References

1. Deliverable 4.3.3 MODACloudML IDE final version
2. Deliverable 2.3.2 decision making toolkit requirements and architecture, and update on business methodology
3. Deliverable 7.1.2 case studies requirements—Final version
4. Deliverable 8.1 case study design analysis
5. Deliverable 8.4.2 Healthcare application design—Final prototype design
6. Deliverable 9.3.2 prototype of healthcare application final release

Open Access This chapter is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

