Migration of Legacy Systems to Cloud Computing

Ana Sofía Zalazar[1,2], Silvio Gonnet[3], Horacio Leone[3]

[1] CONICET, Crisóstomo Álvarez 722, 4000, Tucumán, Argentina
[2] GITIA (UTN-FRT), Rivadavia 1050, 4000, Tucumán, Argentina.
ana.zalazar@gitia.org
[3] INGAR (UTN-CONICET), Avellaneda 3657, 3000, Santa Fe, Argentina.
{sgonnet,hleone}@santafe-conicet.gov.ar

Abstract. Cloud Computing is the dynamic provisioning of physical and virtual resources as services offering by providers, to optimize performance and utilization of their resources. Consumers contract Cloud services and negotiate service level agreements. Some consumers plan to migrate functionality of their legacy systems to Cloud Computing, to minimize investment on their own infrastructure and to obtain new software solutions that rapidly adapt to changes in the system environment. Therefore, the contribution of this work is to classify different types of migration of legacy systems to Cloud Computing, according to the characteristics of the applications and the deployment models of Cloud Computing. Thus, a workflow for functionality migration is proposed, based on the experience in system conversion projects and the analyzed characteristics of Cloud environments. Finally, some security risks are evaluated in the migration process and some recommendations are listed.

Keywords: Cloud Computing, Legacy Systems, Migration.

1 Introduction

Cloud Computing is a paradigm of external arrangement, where third party services are contracted according to *Service Level Agreements* (SLA) between Cloud providers and service consumers, by means of Internet protocols. In this way, Cloud providers optimize the usability of their own technology infrastructure offering storage solutions (*hosting*) and computer services (*outsourcing*), and Cloud consumers pay for Cloud services taking into account the type of service charge (i.e. pay per use, subscription, etc.) [12, 14].

The migration of *legacy systems* of one organization to Cloud Computing environments represents great advantages in the cost-benefits analysis (e.g. pay only for what it is used, less investment in hardware and technical maintenance, etc.) and growing opportunities for rapid adaptation to dynamic changes of the organization business market. Similarly, system migration provides the ability to integrate some applications in a single software solution, and to create collaborative processes among customers, partners and different vendors [13].

Most of the analyzed papers about Cloud migration in software engineering are focus on functional aspects. For instance, Andrikopoulos et al. [1] mention Cloud migration as an adaptation of legacy systems to be deployed in the infrastructure of Cloud providers. That contribution does not propose any mechanism or guideline to adapt legacy system to Cloud environments, however it is focused on defining related concepts to four types of migration: (a) Replace components with Cloud offerings; (b) Partially migrate some of the application functionality to the Cloud; (c) Migrate the whole software stack of the application to the Cloud; and (d) Cloudify the application.

Similarly, Kaisler and Money [8] do not consider migration processes, and they take into account aspects about acquisition, implementation, economic factors, security and privacy. Kaisler and Money also classify four type of migration: (a) Data migration; (b) Information migration; (c) Service migration; and (d) Autonomic migration.

Furthermore, Know and Tilevich [11] present the migration to Cloud Computing as the code refactoring of system functionality to be accessed by remote clients. That code refactoring is about the code transformation of legacy systems to create Cloud services and the configuration of client application to access to the created services. This latter work is limited to the reengineering and refactoring, and it does not cover other simpler types of migration, such as replacing a system component for an existing service from Internet.

Undoubtedly, the tendency in Cloud migration is to firstly transform the traditional applications, commonly structured and form-based, to applications based on *Software-Oriented Architecture* (SOA) [4, 9, 10]. Because this transformation promotes loose coupling, code reuse, service integration, and the abstraction of technological infrastructure; which are the key factors in Cloud environments.

For example, Chauhan and Babar [6] analyze the activities of migration based on SOA to the Cloud model "*Software as a Service*" (SaaS), and after evaluating the quality requirements, they propose a method for this migration. The method consists basically in four steps: (a) Evaluation of components for scalability; (b) Evaluation for orchestration; (c) Identification of the components for refactoring; and (d) Evaluation of the solution against the target Cloud environment.

The rest of this paper is organized as follows. Section 2 introduces Cloud Computing characteristics, services models, and deployments models. Section 3 presents different types of migrations for legacy systems to Cloud environment, considering the analysis of software system and the implementation models of Cloud Computing. Section 4 proposes the *workflow* for Cloud migration, based on system migration projects and the definition of adapted tasks to be applied to Cloud environments. Finally, Section 5 explains a case study of a beverage distributor company and Section 6 lists some security risk and recommendations to take into account during the migration of legacy systems to Cloud Computing. In this work, we consider that security aspect should be considered during the all process of system migration.

2 Characteristics of Cloud Computing

The concept "Cloud Computing" has been used as a marketing term in various contexts representing different ideas [15]. Cloud Computing is not a new technology, but it is a new business paradigm based on existing technologies as: "*Virtualization*": mechanism for running applications and storing data in the same physical resources, as the applications and the data were isolated in different servers; "*Grid computing*": processing on multiple servers; "*Broadband Internet*": fast networks that transports large amounts of data; "*Web 2.0*": applications and technologies that make the Web a collaborative channel; and "*SOA*" architecture that supports building applications using linked services.

There are many available definitions of Cloud Computing in the literature [3, 12, 14], and the most cited definition is offered by Mell and Grace [12] from the *National Institute of Standard and Technology* (NIST). There authors propose a definition that encompasses the broader aspects of the business model: "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*".

Moreover, the NIST defines five Cloud roles: (a) *Cloud Provider*: entity that owns the deployed service in its physical servers, and it is responsible for service maintenance and availability ; (b) *Cloud Consumer*: entity that use the service for completing its business process; (c) *Cloud Carrier* : intermediary that provides data transportation and service connectivity; (d) *Cloud Broker*: intermediary that is involved in the business contract and the relation between Cloud entities; and (e) *Cloud Auditor*: external agent that is responsible for keeping track all business process, reporting failures, and analyzing the quality of services considering the SLA.

Based on these definitions, five essential characteristics are attributed to Cloud Computing [12]: (a) *On-demand self-service*: Cloud consumers automatically access to Cloud resources according to their needs; (b*) Broad network access*: cloud resources are available in the network, and the access to resources is performed by different client platforms; (c) *Resource pooling*: Cloud provider resources are shared between multiple Cloud consumers, by using virtualization mechanism and tenancy; (d) *Rapid elasticity*: virtual and physical resources are added and released according to Cloud consumers demand; and (e) *Measured service*: Cloud systems automatically control and measure the services use in a transparent manner for all Cloud actors.

The dynamic of Cloud Computing consists in variable provisioning of physical and virtual resources by the Cloud provider, to optimize the performance and the utilization of information technology resources. Cloud providers rapidly vary the amount of resources adding new units (servers), replacing or disconnecting units according to the demand of Cloud services, without being noticeable by Cloud users and consumers [2].

2.1 Deployment Models

Before contracting a Cloud provider and migrating the functionality of legacy systems to Cloud environments, it is necessary to analyze the different deployment models of Cloud Computing and the specific requirements of the Cloud consumer organization. Moreover, the adoption of a particular deployment model depends on the criticality of the Cloud consumer business processes and the sensibility of the data associated to these processes.

The Cloud deployment models indicate if the services have been deployed in a private manner, in a place where users can shared the resources with a limited number of trusted partners, or in a third part location where the resources are acceded publicly. Depending on the physical location of the resources, Cloud services can be:

- "*On premises*": the deployment is inside of the security zone or the data center of the consumer organization. The service consumers can require that the physical resources are exclusively used by them, under a special protection policy (i.e. *firewall*) and jurisdiction.
- "*Off premises*": the deployment is outside of the physical control perimeter of the Cloud consumer, and the information about the physical location of the contracted resources may not be specified by the Cloud service provider. This deployment type can be very risky for a Cloud consumer.

The different deployment models defined by NIST [12] are:

- *Private Cloud:* The physical and virtual resources are exclusively acceded by one organization with multiple internal users. This type of cloud can be controlled and managed by the organization, the vendor, or both, sharing management responsibilities.
- *Community Cloud:* The physical and virtual resources are shared by a community or group of organizations that shared some special features or specific purposes. Generally, the communities require to the Cloud providers to implement specific policies among users of the same community cloud.
- *Public Cloud:* The Cloud infrastructure is shared by several independent consumers, using reinforced multi-tenancy mechanisms that insure the isolation between different Cloud environments. The Cloud services are allocated within the provider's servers or third parties.
- *Hybrid Cloud:* This type of model is a combination of a private Clouds, public Clouds or community Clouds. The hybrid Cloud is usually created to protect sensitive data, to safeguard information, and to exploit the rapid provisioning when there is workload in the private or community Cloud.

When an organization must ensure a high level of security and confidentiality, it may contract a private Cloud and deploy part of the services as "on-premises" to directly execute control routines inside of the physical server. Undoubtedly, the best solution is a combination of services inside of public Cloud and private Cloud, so cloud consumers can control their data in their own servers and simultaneously get the benefits of public Cloud (i.e. outsourcing, scalability, and rapid provisioning).

2.2 Service Models

Five layers can be considered for defining *Information Systems* [7]: (a) *Application Layer*: it includes software components, web services and the clients; (b) *Middleware Layer*: it allows developing applications and it performs the communication between different applications, databases and operating systems; (c) *Operating Systems Layer*: it is responsible for managing virtual or physical resources, where the applications are hosted; (d) *Hypervisor Layer*: it is a virtualization layer that is managed by the operating systems; and (e) *Infrastructure Layer*: it contains hardware, storage resources, and network devices.

In general, the management of all components in a legacy system is under the control of the service consumer. When the service consumer analyzes the possibility of software migration, the consumer pays special attention to the application layer and the data repositories associated to this layer.

The Cloud service models describe the types of services that can be used in Cloud Computing. Depending of the service model type, the Cloud provider will use different mechanisms of abstraction, resource management, and access control. There are three principal service models, and other models can be derived from them [12]:

- *Software as a Service (SaaS)*. The service consists of applications that end users can access through navigators, browsers, and web interfaces using Internet protocols of their devices. The provider is responsible for the maintenance of the platform and security mechanisms. The service accesses are generally conducted by token and password authentication. Depending of the service contract, the users may have some permission of application configurations.
- *Platform as a Service (PaaS)*. The offered service is a container within a programming environment, libraries, and tools that support the application developments. The container restricts the interactions between the development environments and other systems presented in the same physical infrastructure. The service providers control the access to the network and the platforms, and they are also responsible for the installations of applications, libraries, and tools that support the development in the virtual platforms.
- *Infrastructure as a Service (IaaS)*. The services are virtual machines, storage memory, database service, and network components. The service provider is responsible for all resource management details, technical staff, maintenance of physical resources, and Internet devices. A physical server can have many virtual servers, and the providers may offer each server individually to different consumers.

3 Migration of Legacy Systems

Legacy systems are software solutions found in a company for a long period of time [7]. The systems have probably survived in a company because of the software maintenance (corrective, preventive, and evolutionary), internal resistance to technology changes, or they may run critical processes.

Usually, these kinds of systems work in isolation and with an exclusive data repository (i.e. data files, databases, etc.). Therefore, the communication between the legacy systems to other newer application is a hard task and it requires the definition of complex communication interfaces and data conversion components. Additionally, the companies with old legacy systems need to invest money to integrate theirs tools, to adapt functionality to new technologies, and to make flexible their business processes.

Most of the companies firstly transform the traditional applications to software based on SOA before adopting the Cloud paradigm, because SOA is flexible to services composition and it encapsulates the business logic through *web services* (WSs). Those WSs can communicate each other by standard protocols and methods of message exchanging, which facilitate the services composition.

The SOA applications consist basically of three layers: (a) *Presentation Layer*: where is the user interface; (b) *Business Layer*: where are the business logic and its functionality implemented by algorithms, software components or WSs; and (c) *Data Layer*: where is the schema and application data. The applications can be migrated by layers.

A conceptual model of system migration to Cloud Computing is presented in Fig. 1). It is considered in the migration model the following techniques:

- *Application Replacement:* it entails replacing the whole application or part of it for one or more standard components available in the market. As a result, some configurations and adaptation task have to be conducted as part of the system migration. This replacement can generate the loss of information control, the need of creating new interfaces with other components or applications, the change of the normal workflow in the organization, and the lack of information about the internal structure of the acquired components.
- *Application Conversion:* it consists in transforming the whole application in a Cloud Computing solution. The transformation can be conducted automatically using a conversion engine that also maps the converted data, but it is necessary that the legacy system works normalized and without failures.
  - *Software Conversion*: it is the transformation of algorithms and software applications, keeping their functionality and their structures, but changing some programming aspects.
  - *Data Conversion*: it is the transformation of data according to a specific target schema or format.
  - *Schema Conversion*: it is the transformation of the data structure to a new equivalent structure of database.
- *Application Virtualization:* it basically involves the generation of virtual system container, and all system components and data schema are moved into the container without changing the source code. The legacy system is considered as a black box, within which all layers are encapsulated. During the users' interaction, all the inputs/outputs are analyzed to generate a "*wrapper*" [4] that is the nexus between the encapsulated legacy system and the new presentation layer deployed in Cloud Computing.

- *New Application Deployment:* it is about programming a new application compatible with Cloud Computing. This solution involves software reengineering to obtain the definition of the functions and the operations, and to generate a new application for Cloud Computing. As part of this reengineering processes, it should be analyzed those functionality unused, redundant, and obsolete. Some improvements may be realized and some new functionality may consolidate several functions.

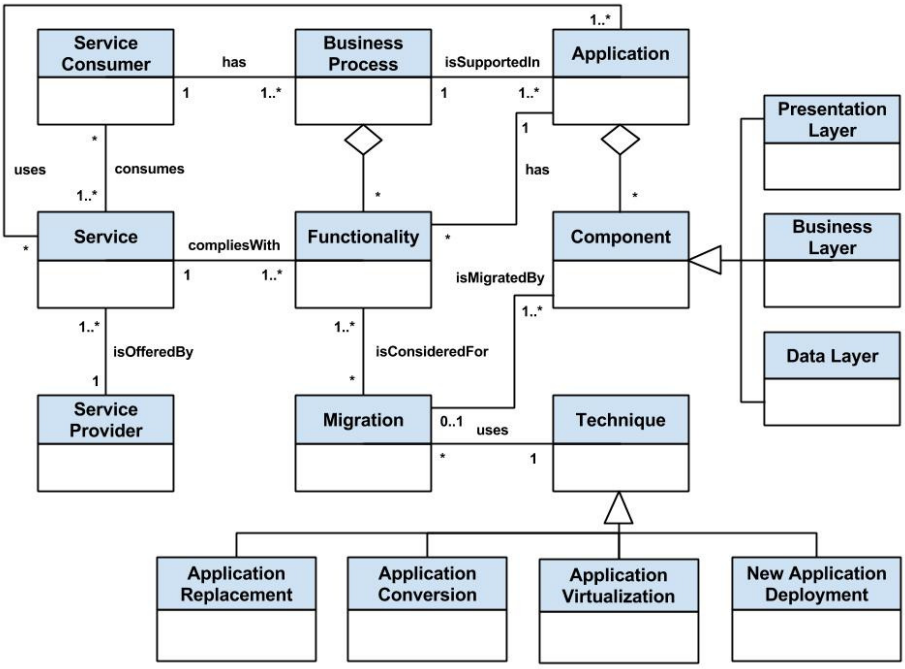

Fig. 1. Conceptual Model of the Migration to Cloud Computing

4 Proposed approach for migrating to Cloud Computing

After understanding the Cloud Computing characteristics, the legacy systems, and the key factors of system migration; the proposed approach is presented in this section for migrating systems to Cloud architecture. This approach considers the security analysis as an integrated activity to each task in the migration processes or workflow. Thus, the Cloud consumers should align the proposed workflow to their security politics and ensure that the Cloud service contract also complies with them.

The Fig. 2 presents the proposed workflow for Cloud migration. The diagram involves 13 processes conducted by the *Consumer*, the *Provider*, or the *Service Developer*. The role of Service Developer consists of a team of functional analysts, programmers, and developers that can belong to the Cloud consumer part, the Cloud provider

part, or a consulting company. Each process of the proposed workflow is explained below.

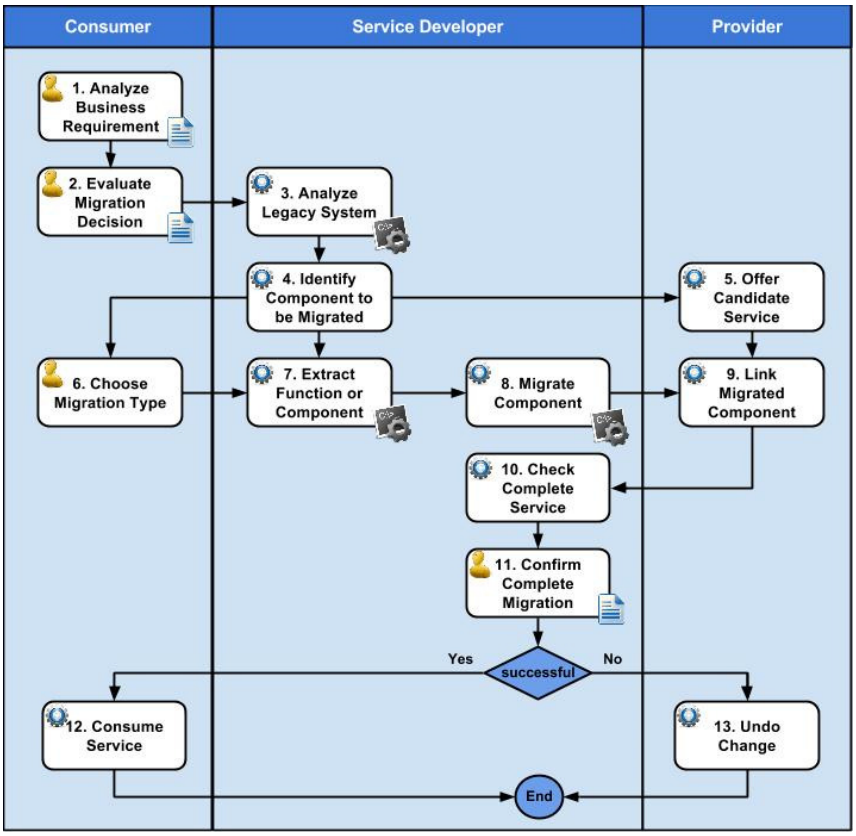

Fig. 2. Workflow for the Migration of applications to Cloud Computing

1. Analyze Business Requirement. The service consumer must make it clear why migrate a legacy systems, an application or a component to Cloud Computing. They should identify business objectives and goals, and in which way the Cloud migration project helps to archive them. Moreover, service consumers should analyze the data characteristics and requirements, before acquiring a new business model where sensitive data may be manipulated by other organization.

2. Evaluate Migration Decision. During this task, the migration project is defined along with its justification, its goals, its scope, and its limitations. Usually, the motivations for migrating systems to Cloud Computing are costs reductions, capabilities additions to adapt functionality to emerging markets, and the improvement of productivity

and safety. The consumer role should seriously discuss these requirements and the impact of migration on its organization, and then the plan for migration is defined.

3. Analyze Legacy System. This task is about the understanding of the current system, and in this regard all documentation and information about the implementation is collected by analyzing components. If the system is a "black box" (there is no access to the source code), the analysis is about inputs, outputs, and system responses. On the other hand, reverse engineering is useful for "white box" systems (available and accessible source) and it decomposes the system in functions and data. The final analysis of the system components from the legacy system will divide the system by types of functions, and then it will align these functions to the business goals, that is, the functionality mapping to business requirements. This task helps to discover the main functions that must be in the migrated system.

4. Identify Component to be Migrated. In this step, the components to be migrated are recognized, and its configuration is extracted. According to the migration objectives, the components to be migrated can be a database, an application layer, a function, an algorithm, or the whole legacy system. At this point, it should be considered a way to preserve the privacy and security of the components by data encryption or encoding mechanism.

5. Offer Candidate Service. This activity will identify services that have been deployed in Cloud Computing environments, and they may be contracted as a migration solution. Then, the consumer should select the candidate services that fulfill the identified requirements, considering the security level guaranteed in contracts and SLA. It is very common that the Cloud services are directly offered by the provider, or it may need a broker that performs the service search and contract negotiation with the service provider. Most of the time the Cloud services are like black box, so it should be compared the candidate Cloud service behavior (i.e. analyzing its input, its output, and its performance level) with the expected results. Furthermore, there are three possible situations in this step: (a) there is no found service that fulfills the requirements; (b) the found service is incomplete; (c) a new interface may be coded for message exchanging with the service; or (d) the service completely fits perfectly with the expected solution. Finally, some security considerations about Cloud contract are carried out in Section 6.

6. Choose Migration Type. To choose the type of migration, the consumer should validate the identified components, the service candidate (if there is any), and the needed resources for the migration process. The consumer should also indicate what type of migration (see Section 3 for more details) fulfills the goals and analyze the rules for services contract, if the candidate service meets all main objectives.

7. Function or Component. Once the component is identified and located, the service developer **7. Extract** team extract it for its migration. A mechanism of high modularity

and loose coupling is used in this process, so there is no way to access any routine or function from outside of the component.

8. Migrate Component. The first activity of this task is to make a backup of the current system, both the source code and data, because the consumer may need to undo changes ("*rollback*") made during the component migration. According with the migration type, it may be needed to develop some interfaces, to convert some functions, and to adapt data schema to the new data repository.

9. Link Migrated Component. In this step, the software solution is deployed into the Cloud Computing environment; also the software configuration of the Cloud service provider service is tested. In addition, new configurations are needed to linked the contracted services to the service consumer environment. A Cloud carrier may participate in this activity, so this service carrier can securely transport all migrated components to the provider systems.

10. Check Complete Service. The verification and the validation are crucial to ensure the correct behavior of the system after the migration. Thus, some verification tests are executed to validate that the functionality of the old system is also kept in the new system implementation. The quality analysts can perform unit test, integration test, acceptance test, and pilot test, before leaving all the control to the new system or integrating the solution to the organization processes.

11. Confirm Complete Migration. After verifying the correct functioning of the migrated service into the organization environment, the service should be evaluated considering the quality standards to be incorporated into the organization business processes. The confirmation of succeed migration is made in this step along with documentation of the service implementation, configuration settings, and all changes made during the migration.

12. Consume Service. The service consumer should be sufficiently prepared to use the service and to understand the new processes, since the migration of functionality to a new paradigm is generally associated to changes in the organization procedures.

13. Undo Change. If the migration has failed or it does not comply with the minimum level of quality standards required by the service consumer, the whole migration process must be reversed using a backup copy.

5 Case Study: Beverage Distributor Company

This case study briefly describes that the proposed workflow are applicable to a solution for migrating functionality to Cloud Computing environments.

It is considered a beverage distribution company that must adapt their purchase order system to allow beverage suppliers make daily, weekly, and monthly reports, for reserving and supplying orders in advance. To achieve competitiveness in the market of beverage dispensing, the distributor company must make sure that its provider import drinks on time according to the distributor demand, and therefore it is required a software module of purchasing that allows visualizing the stock and reserve units in real time.

After analyzing these requirements (*Step 1. Analyze Business Requirement* in *Fig. 2*), the distribution company executives recognized that their old server is too small to make reports required by their beverage suppliers, and to have a modulus of purchasing installed at another company may be a security risk. Therefore, the executives decided to migrate this functionality to cloud computing (*Step 2. Evaluate Decision Migration* in *Fig. 2*).

Finally, the solution of the strategic level was to migrate a database schema (*Step 4. Identify Component to be Migrated* in *Fig. 2*) to a Cloud Computing provider. To avoid security risks, only the data needed for supplier reports have been considered for the migration. The distribution company decided to keep a routine for updating regularly the data repository in the Cloud.

The beverage distribution company has subscribed to a database provider under the model pay-per-use, and the company only pays for the use of the provider network in the data transferring (*Step 5. Offer Candidate Service, Step 6. Choose Migration Type,* and *Step 7. Extract Function or Component* in *Fig. 2*).

In conclusion, the proposed solution was more cost effective than purchasing a new server. Thus, the suppliers and providers can access to the necessary information and create the sale reports and forecast (*Step 9. Linking Component Migrated* in *Fig. 2*) without representing a security risk to the company.

Furthermore, a wrapper was required to handle online orders and to interact with the external modules in the suppliers servers. Thus, an employee of the distribution company can enter an online form and the wrapper communicates the petition to the supplier system (*Step 12. Consuming the Service* in *Fig. 2*) after exchanging the security certificates.

6 Recommendations for Cloud Computing Contracts

The Cloud Computing acceptance depends on the way that the Cloud services comply with the functional and nonfunctional requirements of the consumers. The security aspects and service contracts are the most criticized factors of cloud solutions. That is why this section is dedicated to address some of the most important aspects that the Cloud consumers should consider about security and privacy to move some functions to Cloud architecture. One of the particular aspects of this business paradigm is that the

data centers are not located in the same geographic location of the service consumer, and for this reason the provider often has no obligation to comply with the same legal issues. Consequently, the service consumer must evaluate security policies presented in the SLA and ensure that the service meets the minimum security requirements of the consumer organization. The safety aspects considered by the *European Network and Information Security Agency* (ENISA) [5] and some recommendations about them are listed below:

- *Data Protection, Data Security, and Intellectual Property.* At this point, the most important aspects are authenticity, data and service integrity, availability, and information confidentiality. Before contracting a Cloud service, where the manipulation of the organization's sensitive data is delegated to other company, the consumer should analyze that the service contract explicitly provides some protecting mechanism, guarantees of lawful processing, and compensation in the case of any violation of terms and conditions. In addition, the Cloud provider should be obliged to notify when there are threats, hazards or incidents affecting the integrity, confidentiality and availability of customer information.
- *Information Transfer.* The service consumer should consider the guarantees of adequate protection of data, even though the origin or the destination of the data transfer is in different jurisdiction.
- *Confidentiality and Non-disclosure.* The service consumer should review the policies of confidentiality and non-disclosure, to exactly know what information will circulate in Cloud environments and the level of risk of unauthorized access, manipulation, or destruction of consumer information.
- *Limitation of liability.* After analyzing the risks and the limitations of liability, the associated compensation should be according to the risk level and criticality of the service.
- *Impact analysis of Change of control.* The consumer should grant responsibilities and contractual obligations to the service provider, when the provider makes control changes or subcontract without the consent of the consumer.
- *Data and Services Portability.* The contract should specify that transfer of data and documents can be performed without complications and that the consumer can freely migrate Cloud functionality whenever necessary.

The recommendations and security risks are not only in the service provider side, but also they should be considered in the internal consumer infrastructure and network used during the service contact.

7 Conclusion

This work presents the characteristics of services employment in "*Cloud Computing*", the reasons why this business paradigm continues to grow, and how are the mechanics for provisioning of physical and virtual resources. In addition, the deployment models

are presented by location (on premises and off premises) and by access (private cloud, public cloud, community cloud and hybrid cloud).

Moreover, physical and logical layers are identified in an information system, and according to the layer control that the provider gives to the consumer, the Cloud service models can be classified (IaaS, PaaS and SaaS).

The characteristics of legacy systems are then described, and the tendency of transforming these systems to SOA applications is also explained. Furthermore, the migration techniques of legacy systems to cloud computing are presented in this work, and a conceptual diagram of this migration is proposed.

In this paper, we have presented a preliminary workflow to perform the migration of legacy systems, which consists of 13 steps. The proposed approach has been developed using in system migration project and the deep analysis of Cloud Computing characteristics. Then, an illustrative example is presented using the proposed workflow in the scenario of a beverage distributor company.

The proposed workflow represents a useful and simple tool for decisions making, project organization, and tasks planning for Cloud Computing projects.

Finally, it is considered that the security plan should be integrated throughout the migration process, and also some points and recommendations are stated on safety and service contracts.

Acknowledgments. This work is funded jointly by CONICET and UTN. The support provided by these institutions is gratefully acknowledged.

References

1. V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch. How to adapt applications for the cloud environment. *Computing*, 95(6):493-535, 2013.
2. L. Badger, T. Grance, R. Patt-Corner, and J. Voas. Cloud computing synopsis and recommendations. *NIST Special Publication*, 800:146, 2012.
3. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599{616, 2009.
4. G. Canfora, A. R. Fasolino, G. Frattolillo, and P. Tramontana. Migrating interactive legacy systems to web services. In *Software Maintenance and Reengineering, 2006. CSMR 2006. Proceedings of the 10th European Conference on*, pages 10-pp.IEEE, 2006.
5. D. Catteddu and G. Hogben. Cloud Computing: benefits, risks and recommendations for information security. *European Network and Information Security Agency*, Crete, Greece, 2009.
6. M. A. Chauhan and M. A. Babar. Migrating service-oriented system to cloud computing: An experience report. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 404-411. IEEE, 2011.
7. J.L. Hainaut, A. Cleve, J. Henrard, and J.-M. Hick. Migration of legacy information systems. In *Software Evolution*, pages 105-138. Springer, 2008.
8. S. Kaisler and W. H. Money. Service migration in a cloud architecture. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1-10. IEEE, 2011.

9. R. Khadka, A. Saeidi, S. Jansen, and J. Hage. A structured legacy to SOA migration process and its evaluation in practice. In *Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 2013 IEEE 7th International Symposium on the*, pages 2-11. IEEE, 2013.
10. R. Khadka, A. Saeidi, S. Jansen, J. Hage, and G. P. Haas. Migrating a large scale legacy application to SOA: Challenges and lessons learned. In *Reverse Engineering (WCRE), 2013 20th Working Conference on*, pages 425-432. IEEE, 2013.
11. Y. W. Kwon and E. Tilevich. Cloud refactoring: automated transitioning to cloud-based services. *Automated Software Engineering*, 21(3):345-372, 2014.
12. P. Mell and T. Grance. The NIST definition of Cloud Computing. *National Institute of Standards and Technology*, 53(6):50, 2009.
13. I. Mezgár and U. Rauschecker. The challenge of networked enterprises for cloud computing interoperability. *Computers in Industry*, 65(4):657-674, 2014.
14. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50-55, 2008.
15. Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7-18, 2010.