A TAXONOMY MODEL FOR CLOUD COMPUTING SERVICES

Nelson M. Gonzalez, Charles C. Miers, Fernando F. Redígolo, Marcos Simplício,
Tereza C. M. B. Carvalho
Laboratory of Computer Architecture and Networks, University of São Paulo
Av. Prof. Luciano Gualberto, Trav. 3 - Nr. 158 - C1 50a, São Paulo, Brazil

Mats Näslund[1], Makan Pourzandi[2]
[1]*Ericsson Research, Stockholm, Sweden*
[2]*Ericsson Research, Ville Mont-Royal, Canada*

Keywords: Cloud Computing, SPI, Taxonomy.

Abstract: The continuous development of cloud computing is in evidence in several academic and non-academic researches. However, the relative youth of this field has produced several distinct definitions and taxonomies regarding the concept of cloud computing, as well as the classification and organization of such services. The appearance of commercial cloud solutions in this context with no firmly established standards only complicates the matter, making it difficult to determine how solutions should be technically identified and qualified. Therefore, with the growing complexity of the area, identifying, clarifying and classifying cloud services are essential steps to understand their organization, purpose and interaction with other services. With this goal in mind, this article presents a study on existing concepts and taxonomies, and then harmonizes these approaches in an extensible taxonomy model for cloud computing services. More specifically, this proposal builds on the SPI (Software, Platform, and Infrastructure) taxonomy created by NIST (National Institute of Standards and Technology), creating a hierarchical organization that groups different services according to their characteristics; the result is a taxonomy model that allows finer-grained analyses to be performed, while essentially keeping the simplicity of SPI itself. Finally, we present a specific instance of this model focused on existing and representative cloud services.

1 INTRODUCTION

The concept of cloud computing is the center of attention in several academic and non-academic research efforts. Cloud computing is sometimes considered the natural evolution of Internet (Velte et al., 2009), offering services able to replace hardware and software purchased and maintained by companies and customers at their own risk and expense. The first main advantage of this approach is the clearer distinction between the business itself (e.g., selling multimedia content) and the tools required by the business (Coombe, 2009) (e.g., datacenters and computer programs). Hence, cloud services allow customers to focus on their projects, leaving all unrelated aspects to the service provider.

Since cloud computing is a recent and evolving concept, it still lacks a precise classification. Indeed, even core aspects such as the possible structures, architectures and application scenarios still remain under continuous discussion, which is reasonable to expect for a technology whose impact and grandiosity is compared to the development of Internet itself (Armbrust et al., 2009). In this context, it is important to understand how cloud services interact with other services, as well as the mechanisms that can (or must) be used in order to assure a proper communication between them. This task, on the other hand, requires a robust model capable of classifying the services according to their types, which has lead to the creation of several taxonomies with this purpose (Jaekel and Luhn, 2009). One of the most accepted and adopted models for building such taxonomies is the SPI model (Mell and Grance, 2009), which divides services into three main categories: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

In this paper, we present an enhanced taxonomy

M. Gonzalez N., C. Miers C., F. Redígolo F., Simplício M., C. M. B. Carvalho T., Näslund M. and Pourzandi M..
A TAXONOMY MODEL FOR CLOUD COMPUTING SERVICES.
DOI: 10.5220/0003384800560065
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 56-65
ISBN: 978-989-8425-52-2

model for cloud computing solutions, which employs an hierarchical structure to harmonize the concise approach adopted by the SPI model with the finer-grained structure from other taxonomies, namely those presented in (Youseff et al., 2008; Johnston, 2010; Rimal et al., 2009a; Laird, 2009; Linthicum, 2009). Since these taxonomies were designed specifically to identify how services can be compared and combined (Youseff et al., 2008), the proposed model allows a more involving organization of cloud solutions. This model can thus be seen as an enhanced version of (rather than a substitution for) the SPI model, offering relevant criteria for classifying existing services and also allowing reusability and composability of new ones. Moreover, in order to illustrate these capabilities, we also present a specific instance of the proposed taxonomy model, focusing on existing cloud services. Both the model and this specific instance aim to guide further studies on cloud computing and the lifecycle of related services, focusing on how such services may be created and configured to perform different and complementary tasks.

The rest of this document is organized as follows: Section 2 provides an overview of cloud computing based on official definitions conceived by NIST (National Institute of Standards and Technology) (Mell and Grance, 2009) and ENISA (European Network and Information Security Agency), emphasizing the key characteristics of cloud services. Section 3 presents and briefly describes some relevant taxonomies that compose the basis of our proposal. These taxonomies are then compared in section 4, which explores points to be reproduced and main characteristics taken into account in the proposed taxonomy model, presented in section 5. The related work is covered in section 6. Section 7 provides our final considerations, comparing the proposed model with existing approaches and also presenting suggestion of future work in the area.

2 CLOUD COMPUTING OVERVIEW

Cloud computing solutions are being developed and becoming available for final users despite the lack of a clear classification, and while new features are still under research (IDC, 2010). A recurring issue in this scenario is the free and unrestrained creation of definitions and classification categories by service providers, which are commonly attempting to promote their solutions' features rather than following a technical perspective (Willis, 2009). Probably motivated by this concern, both NIST and the European Community started the process of standardizing the key aspects of cloud computing, aiming to unify the existent concepts and to simplify the task of identifying the main purposes of each cloud solution (Hoover, 2009).

NIST efforts to standardize the basic cloud computing concepts are associated to the increasing use of cloud computing solutions by USA governmental agencies. Therefore, most of the institute's work focus on USA government requirements for cloud computing, both from the technical and legal points of view. According to NIST, cloud computing can be defined as follows (Mell and Grance, 2009):

> *"Cloud computing is a model for enabling 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. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models."*

The main characteristics of cloud computing as defined by NIST are (Mell and Grance, 2009):

- On-demand self-service: The resources provided are easily scaled, so the customer is able to allocate or free them automatically whenever needed, without human interaction with providers;
- Broad network access: Services are available through Internet using standardized interfaces (such as browsers or remote console control), being accessible from any device able to connect and use these interfaces;
- Resource pooling: Provider infrastructure is organized to offer services for multiple customers using the concept of multi-tenancy, according to which resources are managed according to the demand. The resource management is transparent to users.
- Rapid elasticity: Cloud capabilities are promptly and easily scaled to satisfy the customers' needs, in such a manner that the cloud resources appear to be unlimited from the users' point of view; and
- Measured service: Services are automatically controlled and optimized to offer metering capabilities according to the type of service.

In comparison, ENISA (ENISA, 2009) defines cloud computing as an on-demand service model for IT provision, based on virtualization and distributed computing technologies. According to this agency, the key characteristics of cloud computing are:

- Highly abstracted resources: Virtualized infrastructures are the core of cloud computing, either by directly offering a virtualized platform or delivering services based on these structures;
- Scalability and flexibility: Clouds are highly scalable, provisioning customers with the required resources; they are also flexible enough to conform itself to the customers' needs;
- Instantaneous provisioning: The resources appear to the customers as unlimited and promptly provisioned as necessary;
- Shared resources: The provider infrastructure (including hardware, database and memory) is shared by customers. This also has security implications, as the logical layers can be separated but the hardware cannot;
- Service on demand with "pay as you go" billing system: The elasticity of the cloud enables the precise definition of what resources the customer needs, and when they are needed; and
- Programmatic management: Web services programming interfaces allow the management of the cloud infrastructure and also the integration between local software and cloud resources.

We note that both definitions (and also others (CPNI, 2010; CSA, 2009)) highlight the "pay as you go" feature for highly abstracted resources, provided by a same shared infrastructure that, as such, must achieve a minimum level of scalability and flexibility in order to adapt itself to the demand from multiple tenants.

3 EXISTING TAXONOMIES

Cloud computing comprises several types of services, from virtualized storage to development platforms. This complexity lead to the development of many taxonomies aiming to classify the different services and group those with similar characteristics, which is crucial for a more systematic study of such multifaceted area. Some of these taxonomies are very concise, while others create different categories with supposedly heterogeneous characteristics, as discussed in the following.

3.1 NIST's SPI Model

As briefly discussed in section 1, NIST's SPI model (Mell and Grance, 2009) is currently one of the most well-accepted taxonomies for cloud computing (CSA, 2009; Marks and Lozano, 2010). Its main objective is to encompass all various cloud approaches (models, vendors and market niches) in order to standardize concepts. In this manner, the SPI model classifies the cloud solutions in three categories:

- Software as a Service (SaaS): This category is related to cloud services that are developed, deployed, run and maintained by the provider and which are accessible through a Web browser. The customer does not manage or control the underlying infrastructure which supplies the required resources while securing the assets involved and guaranteeing service level agreements;
- Platform as a Service (PaaS): Services that provide a platform to develop and deploy web applications. The customer again does not have control over the cloud infrastructure but is able to directly access and use its resources, commonly made available through the use of specialized APIs (Application Programming Interfaces); and
- Infrastructure as a Service (IaaS): Cloud solutions able to offer infrastructural utilities and tools, such as processing, storage, virtual machines or other computing basic resources. The customer has considerable control over the infrastructure, even though there is no direct access to the physical hardware itself;

This taxonomy presents a succinct but comprehensive classification of cloud services abstracting three main possibilities of exploring them: to offer online browser-ready applications running over a virtualized infrastructure (SaaS); to provide platforms for the deployment of applications which use the underlying resources from the provider's infrastructure (PaaS); or to deliver the virtualized infrastructure itself (IaaS).

3.2 Youseff's 5-Layer Ontology

Youseff *et al.* (Youseff et al., 2008) conducted a research on cloud computing aiming at a final and consolidated taxonomy to be used for general purposes. The proposed taxonomy features five layers containing altogether six different categories depicted in Figure 1 and described as follows:

- Software as a Service (SaaS): Aggregates all software-oriented solutions, e.g., Salesforce CRM (Salesforce, 2010b), Google Apps (Google, 2010d);
- Platform as a service (PaaS): Application development platforms, e.g. Google App Engine

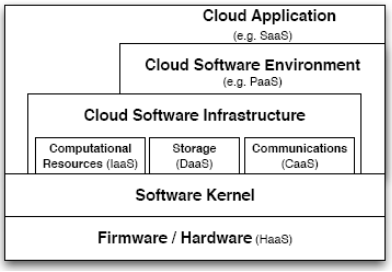

Figure 1: Youseff's 5-layer ontology (Youseff et al., 2008).

(Google, 2010a), Salesforce Apex (Salesforce, 2010a), Hadoop (Borthakur, 2007), Yahoo Pig! (Olston et al., 2008);

- Infrastructure as a Service (IaaS): Computational resources that can be moved to the cloud, mainly represented by virtualization, e.g. Amazon EC2 (Amazon, 2010b), Enomalism (Enomaly, 2010), Eucalyptus (Nurmi et al., 2009), OpenNebula (Fontán et al., 2008);
- Data as a Service (DaaS): Refers to remotely managed solutions for data storage, e.g., Google FileSystem (Google, 2010b), Bayou (Demers et al., 1994), Dynamo (DeCandia et al., 2007), Amazon S3 (Amazon, 2010e), EMC (EMC, 2010);
- Communication as a Service (CaaS): Solutions providing service-oriented communication for cloud users, including VoIP systems, instant messaging, and audio/video conferencing, e.g. Microsoft Connected Services Framework (Microsoft, 2010a; Hofstader, 2007); and
- Hardware as a Service (HaaS): Refers to services where the provider operates, manages and upgrades the hardware used by the customer, e.g., Morgan Stanley IBM utility computing (Hines, 2004) PXE (Johnston, 1999), UBL (UBOOT, 2010), IBM Kittyhawk (Appavoo et al., 2008).

The five layers abstract computing environments, starting from the application (top layer), going through the environment (which provides the resources to the application to run), software infrastructure (the resources that are enabled by the operating system), kernel and finally hardware. These layers are used to identify possible categories for cloud services, resulting in the taxonomy presented.

3.3 Sam Johnston's 6-Layer Stack

Judging that NIST's classification based on three layers/categories was oversimplified, Johnston (Johnston, 2010) created a cloud computing taxonomy focusing on the software for the final users. The resulting taxonomy organizes the cloud ecosystem in six layers, illustrated in Figure 2 and detailed as follows:

- Clients: Computer hardware and/or software that rely on the cloud to deliver services and applications, e.g., browsers in general, cloud operating systems, Cirtas (Cirtas, 2010), g-Eclipse (Gjermundrod et al., 2008), StorSimple (StorSimple, 2010);
- Services: Web services that allow the direct interaction between customers and providers' solutions;
- Application: Solutions that eliminate the need of installing or running software on local machines, e.g., Google Apps (Google, 2010d), Zoho (Zoho, 2010);
- Platform: Computing platform to deploy applications without having to purchase the required infrastructure, e.g., Force.com (SalesForce, 2010), Google app Engine (Google, 2010a), Heroku (Heroku, 2010), OrangeScape (OrangeScape, 2010), Rackspace (Rackspace, 2010);
- Storage: Delivering storage as a service, either in the form of raw storage or database utilities, e.g., Amazon S3 (Amazon, 2010e), Amazon SimpleDB (Amazon, 2010f), Dropbox (Dropbox, 2010), Evernote (Evernote, 2010), Mozy (Decho, 2010), Windows Live Skydrive (Microsoft, 2010b); and
- Infrastructure: Services that offer virtualization solutions for customers, allowing direct access to the cloud infrastructure, e.g., Amazon EBS (Amazon, 2010a), Amazon MI (Amazon, 2010c), AppNexus (AppNexus, 2010), Cloudera (Cloudera, 2010), ElasticHosts (ElasticHosts, 2010), GoGrid (GoGrid, 2010), Hadoop (Borthakur, 2007), RightScale (RightScale, 2010).

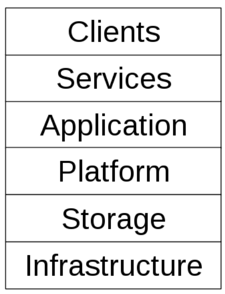

Figure 2: Sam Johnston's 6-layer stack.

The resulting stack shares many similarities with the one used by Youseff *et al.*, but includes the

"Clients" layer, which represents elements that effectively use the services from the adjacent layer.

The same taxonomy can also be presented as in Figure 3, which shows the categories and the examples used by the authors to contextualize them.

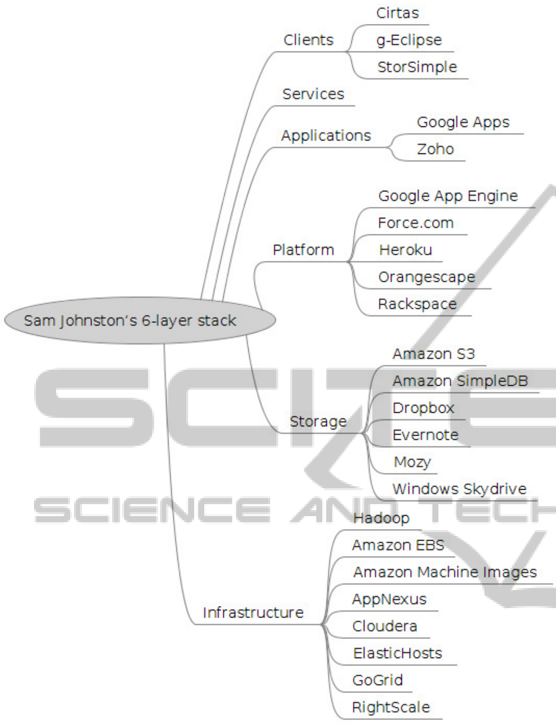

Figure 3: Sam Johnston's 6-layer stack.

3.4 Dave Linthicum's 10-Layer Taxonomy

Linthicum's taxonomy (Linthicum, 2009) aggregates all possible services into ten categories on the same level (see Figure 4).

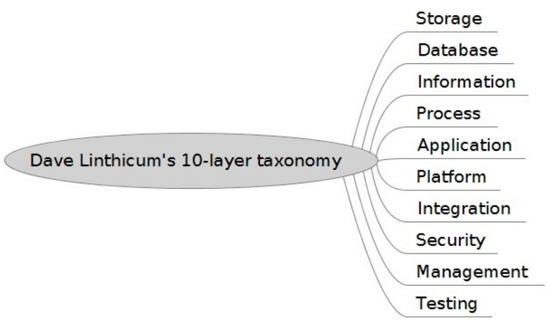

Figure 4: Dave Linthicum's 10-layer taxonomy.

These categories are described as follows:

- Storage as a Service: Ability to leverage storage handled by the customer, so it becomes physically remote but logically local, e.g., Amazon S3 (Amazon, 2010e), Box.net (Box.net, 2010), Google Base (Google, 2010c);
- Database as a Service: Follows the same principle of Storage, but focuses exclusively on databases, offering the space need for storing the data and also interfaces to use the database tools, e.g., Amazon SimpleDB (Amazon, 2010f), Trackvia (Trackvia, 2010);
- Information as a Service: Offers APIs and interfaces to access remotely hosted information, such as stock prices, address validation, credit reporting or meteorology;
- Process as a Service: Provides structures able to bind other resources altogether, creating business processes easily changeable, e.g., Appian Anywhere (Appian, 2010), Itensil (Itensil, 2010);
- Application as a Service: Software accessible through browser, e.g., Salesforce CRM (Salesforce, 2010b), Google Apps (Google, 2010d);
- Platform as a Service: Includes the application development platform, interfaces to access the cloud infrastructure (including database utilities), raw storage and testing, e.g., Google App Engine (Google, 2010a), Heroku (Heroku, 2010);
- Integration as a Service: An integration stack that includes application interfaces, flow control and integration designs, allowing the intercommunication between different programs or processes, e.g., Amazon SQS (Amazon, 2010d);
- Security as a Service: Mainly related to identity management, e.g., Ping Identity (Identity, 2010);
- Management as a Service: On-demand services that provide resources and structure to manage other cloud services in terms of topology, resource utilization, virtualization and task management, e.g., RightScale (RightScale, 2010); and
- Testing as a Service: Services able to test other cloud services, eliminating performance issues and bugs during development, deployment and production stage, e.g., SOASTA (SOASTA, 2010).

This taxonomy introduces many categories to divide the cloud ecosystem. However, it lacks some important aspects, as the IaaS set of services is not entirely covered (there is not a clear category for virtualization services) and there are no clear interrelationships between different categories.

4 COMPARISON AND LIMITATIONS OF EXISTING PROPOSALS

The first point worth noticing when comparing the four taxonomies previously presented refers to the number of categories adopted: the most succinct is the SPI model with only three categories, followed by Youseff's five-layers ontology, then by Johnston's six-layers stack and finally by Linthicum's taxonomy with ten different categories. Despite this difference, they all follow a similar organization and assume a linear or flat structure. Hence, it is plausible to use the SPI model as reference to compare the other taxonomies, as depicted in Table 1. Even though the SPI model was not necessarily used by the other taxonomies presented, this table helps to identify the main points where each layer can be subdivided.

Table 1: Taxonomies comparison using SPI model as reference.

Taxonomy	*SaaS*	*PaaS*	*IaaS*
Youseff's 5-layer	Software	Platform	Comp. resources Storage Communications
Johnston's 6-layer	Services Application	Platform	Storage Infrastructure
Linthicum's 10-layer	Application Testing Security Information Process	Platform Integration Management	Storage Database

One of the most important requirements when specifying cloud taxonomies is to permit an easier identification of interrelationships between different services; this allows the services to be studied separately, as well as the creation of new services by composing simpler ones.

The straightforward approach proposed in the SPI model is helpful when a simple and direct classification is needed, but encumbers the identification of interrelationships between services. In comparison, the other models described have more categories in a flat distribution; however, they rely basically on existing services to create the categories, in such a manner that the categories themselves usually do not have clear connections. One of the main disadvantages of this more empiric approach is that the resulting categories are likely unable to fully or properly cover future services, or even existing services that were not considered when the taxonomy was first designed. One example is the Amazon EC2 (Amazon, 2010b), which does not fit in any of Linthicum's taxonomy (Linthicum, 2009) categories: even though it is a storage service, it also offers virtualization and database features.

We can conclude that, despite their importance, the above-mentioned taxonomies are either too broad – not providing the granularity needed for deeper and more complex comparisons between distinct services – or too narrow – lacking the flexibility for adding new, more complex services, – and do not offer an explicit connection between the different categories. In this context, the development of a more flexible and comprehensive approach gains interest.

5 PROPOSED TAXONOMY

The proposed taxonomy model builds on the widely accepted SPI, but adopts an hierarchical organization rather than a flat one. This approach leads to a more structured taxonomy, not only conserving the simplicity of the SPI model in its higher level, but also enabling a deeper analysis of cloud services in its lower levels, as aimed by the other taxonomies described in section 3. Indeed, one of the most interesting features of the SPI model is its ability to offer an overview of services and to initially categorize them; the proposed taxonomy model extends this property with a finer-grained structure, enabling the development of the more sophisticated analyses that are not possible with the SPI model alone.

The result is a tree-like structure that aggregates cloud solutions, favoring the identification of similarities, dependencies and complementary points between these services. Figure 5 illustrates the proposed model, showing a taxonomy based on existing services. The specific categories depicted are:

- Software:
  - e-Commerce: Services that enable the creation of online stores and catalogs for local physical products, and application stores for mobile platforms;
  - Software management: Services that communicate with other software (hosted or not in a cloud) with management purposes; such as a hosted web application or an antivirus server. This category can be further divided into monitoring services (e.g., for checking service uptime, performance, security or SLA compliance), controlling (for creating, starting and stopping other services) and integrating (for data synchronization and conversion);
  - Financial management: Financial transactions, such as payment and billing services;

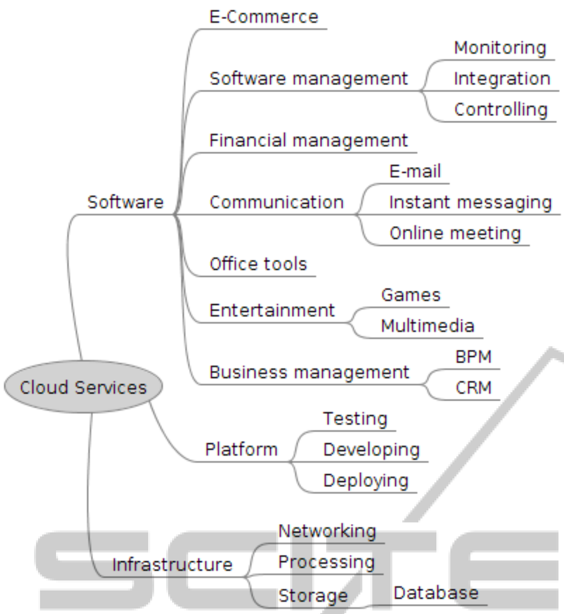

Figure 5: A service-based instance of the proposed taxonomy model.

- Communication: General-purpose types of communication tied to an online event (e.g., VoIP, webcasts, webinars and web conferencing solutions);
- Office tools: Word processors, spreadsheets, presentation software, file conversion, calendars, etc.;
- Entertainment: Online games, video/audio streaming, which demand specific QoS requirements; and
- Business management: Corporate applications such as CRM (Customer Relationship Management) and BPM (Business Process Management).

• Platform:
  - Developing: Hosted environment for software development, like IDEs (Integrated Development Environment), versioning and bug-tracking systems;
  - Deploying: Services that offer an environment to deploy applications, from static to dynamic web pages and web services; and
  - Testing: Solutions for automated tests of new applications.

• Infrastructure:
  - Processing: Virtualized processing resources, e.g., virtual machines execution;
  - Storage: Virtualized storage, including database, file storage, and disk storage; and
  - Networking: Virtualized networks and services on top of an existing network infrastructure, such as wireless clouds.

The proposed format enables not only to classify a service using specific categories (represented by the leaves), but also using more generic levels, which are especially recommended when a service is very complex or when it aggregates different solutions. Some Amazon web services are interesting examples to contextualize both situations. The Amazon S3 (Amazon, 2010e) is a storage service, which is initially classified as an IaaS; using the proposed model, this service is classified as a Infrastructure-Storage. On the other hand, Amazon EC2 (Amazon, 2010b), which is related to virtualization, offers a complete platform to run software over a virtualized infrastructure; it is classified as an Infrastructure service, because it offers resources related to its leaves. In this case it is interesting to note that both services would be simply classified as IaaS by the SPI model – which is correct and adequate if a simple analysis is required, but insufficient if one requires a comparison between various IaaS providers, each of which offering different solutions.

We emphasize, however, that the categories enumerated above are not necessarily exhaustive. Instead, the taxonomy instance depicted in Figure 5 is better regarded as an application of the proposed model for common cloud services; as such, this taxonomy can be easily extended without losing its main characteristics, as long as the SPI root is kept untouched. This extensibility feature also enables a clear evolution path, similar to the way networking standards are created and maintained by standardization entities such as IEEE and IETF (for instance, using tags to identify the taxonomy version).

It is important to emphasize that the combinations between services may be very complex. For example, while using Heroku to develop a dynamic website for an enterprise, the service consumer may decide to include Netsuite Ecommerce to integrate local products to an online store, to use Amazon S3 or SimpleDB to store any information related to business processes, customers and commercial transactions, and finally to adopt SOASTA to improve the website performance. This example illustrates that classifying services into more complex categories whilst keeping a solid foundation enables a swift and useful identification of potential services that, when combined, can create a complete toolkit for developing the desired business.

Table 2 shows the classification of some existing services according to this taxonomy.

Table 2: Classification of existing services using proposed taxonomy.

Service	Main category	Subdivisions
Amazon EC2 (Amazon, 2010b)	Infrastructure	-
Amazon S3 (Amazon, 2010e)	Infrastructure	Storage
Amazon SimpleDB (Amazon, 2010f)	Infrastructure	Storage – Database
Amazon SQS (Amazon, 2010d)	Sofware	Software management – Integration
Appian Anywhere (Appian, 2010)	Software	Business management – BPM
Boomi (Boomi, 2010)	Software	Software management – Integration
Box.net (Box.net, 2010)	Infrastructure	Storage
Elastra (Das et al., 2010)	Software	Software management
EMC SMS (EMC, 2010)	Infrastructure	Storage
Enomaly ECP (Enomaly, 2010)	Infrastructure	-
Google App Engine (Google, 2010d)	Platform	Deploying
Heroku (Heroku, 2010)	Platform	Developing, Deploying
Netsuite Apps (Netsuite, 2010)	Software	Financial management
OpSource Connect (OpSource, 2010)	Software	Software management – Integration
Oracle OnDemand (Oracle, 2010)	Software	Business management – CRM
RightScale (RightScale, 2010)	Software	Cloud management
rPath (rPath, 2010)	Software	Cloud management
Salesforce CRM (Salesforce, 2010b)	Software	CRM
SOASTA (SOASTA, 2010)	Platform	Testing
Trackvia (Trackvia, 2010)	Infrastructure	Database

6 RELATED WORK

In addition to the taxonomies for cloud computing services discussed in Section 3, there are also other taxonomies that aim to identify further criteria to classify those services.

In (Prodan and Ostermann, 2009), Prodan and Ostermann focus on the growth of cloud solutions and the constant need of fast and scalable resources for solving complex and demanding problems from both scientific and business applications The authors provide an organization of cloud computing environments based on eight main aspects: the service type, how resources are deployed, the hardware, runtime tuning, security concerns, the business model adopted, any middleware involved and performance requirements. The organization adopted is very similar to the SPI model, having as main difference the addition of a fourth category denominated *specialized services*, which includes web and file hosting. The cloud solutions are then categorized according to several different criteria, such as environments, service type, deployment, business model, Application Programming Interface (API), and performance. As a result, the proposed classification ends up resembling a comparison/benchmark of the listed cloud solutions rather than a taxonomy, requiring a considerable amount of information on how the solution was built and a deep understanding on its specification and requirements. In comparison, the taxonomy model and its instance proposed in this paper provides a more straightforward classification, letting the benchmark comparison to be verified on demand and according to the consumer's desire (in terms of requirements and criteria).

Rimal *et al.* (Rimal et al., 2009b), on the other hand, highlight the acceptance of pay-for-use models and of the *everything-as-a-Service* approach. They state that the taxonomies currently available were created from the perspective of vendors, not from the actual consumers of the cloud services. With this issue in mind, the authors define many criteria to classify cloud solutions, such as cloud architecture, virtualization management, services, fault tolerance, security, and other issues. The proposed taxonomy is also based on the SPI model, but includes an additional layer denominated Hardware as a Service (HaaS), located between the Paas and IaaS categories. This new category is, however, very similar to the IaaS layer from the original SPI Model, while the then redefined IaaS includes functionalities such as payment, which is usually classified in the SaaS category. The taxonomy model, on the other hand, extends the SPI model in a more elegant manner; moreover, the presented instance focus only on the services category, since it is related only to the taxonomy of cloud services.

7 DISCUSSION AND CONSIDERATIONS

Cloud computing is a powerful technology, enabling productive and constructive use of computational resources, easing software development and distributing applications through the Web. In this context, it is important to have a concise yet comprehensive taxonomy for organizing and classifying cloud solutions, which is many times provided by the solid and well established SPI model. One shortcoming of this model, however, is that it lacks of a clear connection between services inside each category.

In this paper, we address this issue by proposing an enhanced taxonomy model that extends the SPI model by means of an hierarchical structure. In a first level, this new model conserves the simplicity of SPI, allowing an easy identification of the overall characteristics of cloud services. Then, as it goes deeper in the tree-like classification, it enables more sophisticated analyses, which includes the capability of determining existing correlations between distinct services. These correlations allow the identification of complementary points that can be explored in order to offer a complete set of virtualized tools (from software to hardware), so the cloud customer is able to focus on projects rather than the required infrastructure. Therefore, the aim of the proposed model is an extension of (rather than a replacement for) the SPI model, empowering the latter with an organization that allows a finer-grained analysis to be performed whenever needed.

Another essential feature of the proposed model is the extensibility from the SPI root. Hence, its instances should not be regarded as fixed and absolute taxonomies, but as evolving structures that can follow the cloud computing evolution itself. This also means that future studies on criteria and other functional aspects of the cloud may very well improve the proposed model, organizing the services in order to facilitate the development of more sophisticated cloud computing solutions.

ACKNOWLEDGEMENTS

This work was supported by the Research and Development Centre, Ericsson Telecomunicações S.A., Brazil.

REFERENCES

Amazon (2010a). Amazon elastic block storage. http://aws. amazon.com/ebs/.

Amazon (2010b). Amazon elastic compute cloud. http://aws.amazon.com/ec2/.

Amazon (2010c). Amazon machine images. http://aws.amazon.com/amis/.

Amazon (2010d). Amazon simple queue service. http://aws.amazon.com/sqs/.

Amazon (2010e). Amazon simple storage service. http://aws.amazon.com/s3/.

Amazon (2010f). Amazon simpledb. http://aws.amazon.com/simpledb/.

Appavoo, J., Uhlig, V., and Waterland, A. (2008). Project Kittyhawk: building a global-scale computer: Blue Gene/P as a generic computing platform. *SIGOPS Oper. Syst. Rev.*, 42:77–84.

Appian (2010). Appian anywhere. http://www.appian.com/bpm-saas.jsp.

AppNexus (2010). Appnexus cloud computing. http://www.appnexus.com/.

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing. Technical report.

Boomi (2010). http://www.boomi.com/solutions.

Borthakur, D. (2007). The hadoop distributed file system: Architecture and design. http://hadoop.apache.org/.

Box.net (2010). Box. http://www.box.net/solutions.

Cirtas (2010). Cirtas bluejet. http://www.cirtas.com/.

Cloudera (2010). Cloudera. http://www.cloudera.com/.

Coombe, B. (2009). Cloud computing - overview, advantages and challenges for enterprise deployment. Technical report, Bechtel Corporation.

CPNI (2010). Information security briefing 01/2010 - cloud computing. Technical report, CPNI - Centre for Protection of National Infrastructure.

CSA (2009). Security guidance for critical areas of focus in cloud computing. Technical report, CSA - Cloud Security Alliance.

Das, S., Agrawal, D., and Abbadi, A. E. (2010). ElasTraS: An elastic, scalable, and self managing transactional database for the cloud. Technical report, UCSB. http://www.cs.ucsb.edu/research/tech_reports/.

DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels, W. (2007). Dynamo: Amazon's highly available key-value store. In *Proc. of ACM Symposium on Operating Systems Principles*, pages 205–220. ACM.

Decho (2010). Mozy. http://mozy.com/.

Demers, A., Petersen, K., Spreitzer, M., Terry, D., Theimer, M., and Welch, B. (1994). The Bayou architecture: Support for data sharing among mobile users. In *Proc. of the Workshop on Mobile Computing Systems and Applications*, pages 2–7.

Dropbox (2010). Dropbox. http://www.dropbox.com/.

ElasticHosts (2010). Elastichosts – flexible servers in the cloud. http://www.elastichosts.com/.

EMC (2010). Emc storage managed services. http://www. emc.com/.

ENISA (2009). Benefits, risks and recommendations for information security. Technical report, ENISA - European Network and Information Security Agency.

Enomaly (2010). Enomalism elastic computing infrastructure. http://www.enomalism.com/.

Evernote (2010). Evernote. http://www.evernote.com/.

Fontán, J., Vázquez, T., Gonzalez, L., Montero, R., and Llorente, I. (2008). Opennebula: The open source virtual machine manager for cluster computing. In *Open Source Grid and Cluster Software Conference - Book of Abstracts*. http://www.opennebula.org/.

Gjermundrod, H., Dikaiakos, M., Stumpert, M., Wolniewicz, P., and Kornmayer, H. (2008). g-Eclipse - an integrated framework to access and maintain grid resources. In *Proc. of the 2008 9th IEEE/ACM International Conference on Grid Computing*, GRID'08, pages 57–64, Washington, DC, USA. IEEE Computer Society.

GoGrid (2010). http://www.gogrid.com/.

Google (2010a). http://code.google.com/appengine/.

Google (2010b). http://labs.google.com/papers/gfs.html.

Google (2010c). http://code.google.com/apis/base/.

Google (2010d). Google. http://www.google.com/apps/.

Heroku (2010). Heroku - instant ruby platform. http://heroku.com/.

Hines, M. (2004). Morgan stanley, ibm ink utility computing deal. http://news.cnet.com/2100-7339-5200970.html.

Hofstader, J. (2007). Communications as a service. http://msdn.microsoft.com/en-us/library/bb896003. aspx.

Hoover, J. N. (2009). NIST team deeply studying cloud computing – cloud computing – InformationWeek. http://www.informationweek.com/news/government/enterprise-architecture/showArticle.jhtml?articleID=217701603.

IDC (2010). IDC: cloud computing implementations to double by 2012 - network computing. http://www.networkcomputing.com/ other/ idc-cloud-computing-implementations-to-double-by-2012.php?type=article

Identity, P. (2010). PingFederate. http://www.pingidentity.com/.

Itensil (2010). Itensil. http://itensil.com/.

Jaekel, M. and Luhn, A. (2009). Cloud computing - business models, value creation dynamics and advantages for customers. Technical report, Siemens AG.

Johnston, M. (1999). Preboot execution environment (PXE). www.pix.net/software/pxeboot/archive/pxespec.pdf.

Johnston, S. (2010). Taxonomy: The 6 layer cloud computing stack. http://samj.net/2008/09/taxonomy-6-layer-cloud-computing-stack.html.

Laird, P. (2009). Laird OnDemand: cloud computing taxonomy at interop las vegas, may 2009. http://peterlaird.blogspot.com/2009/05/cloud-computing-taxonomy-at-interop-las.html.

Linthicum, D. (2009). Defining the cloud computing framework. http://cloudcomputing.sys-con.com/node/811519.

Marks, E. A. and Lozano, B. (2010). *Executive's Guide to Cloud Computing*. John Wiley & Sons.

Mell, P. and Grance, T. (2009). The nist definition of cloud computing. Technical report, NIST.

Microsoft (2010a). Microsoft connected services framework. http://msdn.microsoft.com/en-us/library/aa306083.aspx.

Microsoft (2010b). Windows live skydrive. http://www.windowslive.com.br/public/product.aspx/view/5.

Netsuite (2010). Suitecloud. http://www.netsuite.com.

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. (2009). The Eucalyptus open-source cloud-computing system. In *Proc. of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, CCGRID'09, pages 124–131, Washington, DC, USA. IEEE Computer Society.

Olston, C., Reed, B., Srivastava, U., Kumar, R., and Tomkins, A. (2008). Pig latin: a not-so-foreign language for data processing. In *Proc. of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD'08, pages 1099–1110, New York, NY, USA. ACM.

OpSource (2010). http://www.opsource.net/Solutions/Cloud-Computing.

Oracle (2010). Oracle on demand. http://www.oracle.com/.

OrangeScape (2010). http://www.orangescape.com/.

Prodan, R. and Ostermann, S. (2009). A survey and taxonomy of infrastructure as a service and web hosting cloud providers. *10th IEEE/ACM Internation Conference on Grid Computing*.

Rackspace (2010). Rackspace managed hosting. http://www.rackspace.com/.

RightScale (2010). Rightscale cloud computing management platform. http://www.rightscale.com/.

Rimal, B. P., Choi, E., and Lumb, I. (2009a). A taxonomy and survey of cloud computing systems. In *Networked Computing and Advanced Information Management, International Conference on*, pages 44–51, Los Alamitos, CA, USA. IEEE Computer Society.

Rimal, B. P., Choi, E., and Lumb, I. (2009b). A taxonomy and survey of cloud computing systems. *Fifth International Joint Conference on INC, IMS and IDC*.

rPath (2010). rpath solutions. www.rpath.com/.

SalesForce (2010). Force.com. http://www.salesforce.com/platform/.

Salesforce (2010a). Salesforce apex. http://wiki.developer force.com/.

Salesforce (2010b). Salesforce customer relationship management (crm). http://www.salesforce.com/crm/.

SOASTA (2010). Soasta cloudtest. http://www.soasta.com/.

StorSimple (2010). http://www.storsimple.com.

Trackvia (2010). Trackvia cloud database. http://www.trackvia.com/products/cloud-database.

UBOOT (2010). Das U–Boot: The universal boot loader. http://www.denx.de/wiki/U-Boot/.

Velte, A. T., Velte, T. J., and Elsenpeter, R. (2009). *Cloud Computing - A Practical Approach*. McGraw-Hill.

Willis, J. M. (2009). Unified ontology of cloud computing | IT management and cloud blog. http://www.johnmwillis.com/cloud-computing/unified-ontology-of-cloud-computing/.

Youseff, L., Butrico, M., and da Silva, D. (2008). Toward a unified ontology of cloud computing. *Grid Computing Environments Workshop, GCE'08*, pages 1–10.

Zoho (2010). Zoho.com. http://www.zoho.com/.