

Preface

Cloud computing is emerging as the most promising technology for software development, changing the way customers interact with their data and applications. There are many reasons that drive the choice of moving to cloud:

- companies no longer need to buy, store, and maintain expensive hardware infrastructures, reducing time and money involved in maintaining, updating, and repairing their own equipments;
- hardware dimensioning is not to be related to peak workload any more, but the infrastructure can be dynamically scaled according to the current needs. This results in a better use of the existing resources;
- customers pay only for the resources they actually use, following a “Pay as you Go” paradigm;
- using distributed resources, including data centers and computing nodes, can enhance systems’ resiliency and disaster recovery;
- the possibility to choose among a broad range of available resources and services can trigger strong competition between cloud providers, thus resulting in better quality and lower prices for customers.

These are only a few of the possible benefits that could derive from the adoption of the cloud computing paradigm but, despite the diffusion of cloud technologies, issues and limitations still exist. A major issue is the lack of portability and interoperability between cloud platforms at different service levels, affecting the cloud computing panorama in several ways and aspects. The brokering, negotiation, management, monitoring, and reconfiguration of cloud resources are nowadays challenging tasks for the developer or user of cloud applications due to different business models associated with resource consumption as well as due to the variety of services—and their features—offered by the variety of cloud providers. These points become very critical when the landscape is a multicloud environment and the main concern is represented by the *vendor lock-in* problem. In fact, cloud providers usually propose technological solutions that differentiate them from their competitors: these differences have the drawback of locking the customers as no

alternatives are offered. Thus, once customers have chosen a cloud provider, either they cannot change to another provider or they can do it but only at a huge cost. Vendor lock-in risk also includes reduced negotiation power in reaction to price increases and service discontinuation (if, e.g., the provider goes out of business).

In the following the structure of the book is illustrated.

In Chap. 1, the notions of cloud portability and interoperability are introduced, together with the issues and limitations arising when such features are lacking or ignored. The illustration starts with definitions of portability and interoperability as inherent to the generality of software systems, and then the concepts are tailored, specialized, and exemplified for the specificity of cloud computing. Some basic concepts of the cloud and reference architectures are reported to define the recurrent terms and roles.

A number of use cases, accompanied with a concrete case study, representing a variety of interoperability and portability scenarios are illustrated. Several definitions and use case scenarios are modeled by means of an n -dimensional feature space, where features represent the different characteristics and abstraction levels of the cloud domain.

The feature space, the use case scenarios, and the case study are utilized in the following chapters in order to position, classify, and demonstrate the different technologies and solutions presented.

Chapter 2 provides an overview of the state-of-the-art methodologies and technologies, which are currently used or are being investigated to enable cloud portability and interoperability. These include: Model-Driven Architecture (MDA) and languages, semantic technologies, cloud patterns, and agent systems. We illustrate in detail how the use of cloud patterns can enable robust cloud applications design and development with respect to portability and interoperability. We position the different methodologies and technologies illustrated with respect to the use case scenarios and features defined in Chap. 1, and we test them by analyzing their application to the case study illustrated in Chap. 1. We also mention and briefly illustrate the contributions coming from projects funded by the European Commission FP7 program.

Chapter 3 illustrates the main cross-platform cloud application programming interfaces and how they can solve interoperability and portability issues by bringing uniformity and standardization to the cloud. This chapter provides an overview of initiatives that provide cross-platform-based cloud APIs such as DeltaCloud, SimpleCloud, JCloud, Libcloud, and research projects whose aim is to provide multicloud APIs (such as mOSAIC). Such APIs are positioned with respect to the use case scenarios and features defined in Chap. 1, and tested by analyzing their application to the case study illustrated in the chapter.

Chapter 4 presents a set of ready-to-go solutions which, either for their wide diffusion in the cloud computing scenario or because they implement the established or emerging standards (see Chap. 5), have a fundamental role in providing interoperable and portable solutions. In particular, Amazon Web Services (AWS) and OpenStack have imposed themselves as “de facto standards” at the IaaS level, since their wide adoption has led other providers to develop APIs and interfaces

which are compatible with their offers. At the PaaS level, Microsoft Azure, Google App Engine, IBM Bluemix, and OpenShift with their multi-language support and ability to interface with other platform services, surely enhance the application portability and thus deserve to be cited here. Such solutions are positioned with respect to the use case scenarios and features defined in Chap. 1, and tested by analyzing their application to the case study illustrated in the chapter. Finally, we also present solutions that have been explicitly created for portability and interoperability purposes, such as Docker, ElasticBox, and Cloudify.

Chapter 5 presents an overview of the emerging standards for cloud interoperability and portability. In particular, here we consider efforts moving toward the definition of shared standards addressing different aspects of the cloud environment, spanning from services communication to data description. Among these, we consider standards such as TOSCA, CIMI, OCCI, and CDMI. Some of these standards are positioned with respect to the use case scenarios and features defined in Chap. 1, and tested by analyzing their application to the case study illustrated in the chapter.

We wish to thank Antonio Argenziano, Andrea Barbato, Graziella Carta, Salvatore D'Angelo, Salvatore Maisto, Stefania Nacchia, and Raffaele Sperandeo for their valuable contribution to the evaluation of the various platforms and solutions described in the book.

Last but not least, we would like to acknowledge the excellent and relentless support of the Springer staff members—in particular Viktoria Meyer, Ralf Gerstner, and Aliaksandr Birukou—during all phases of the development of this book.

Beniamino Di Martino
Giuseppina Cretella
Antonio Esposito

Cloud Portability and Interoperability
Issues and Current Trends

Di Martino, B.; Cretella, G.; Esposito, A.

2015, XIV, 121 p. 58 illus., Softcover

ISBN: 978-3-319-13700-1