Chapter 1
Distributed Interoperability in Heterogeneous Cloud Systems

José C. Delgado
Universidade de Lisboa, Portugal

ABSTRACT

Cloud platforms constitute distributed and heterogeneous systems. Interacting applications, possibly in different clouds, face relevant interoperability challenges. This chapter details the interoperability problem and presents an interoperability framework, which provides a systematization of aspects such as coupling, compatibility, and the various levels at which interoperability must occur. After discussing the main limitations of current interoperability technologies, such as Web Services and RESTful applications, the chapter proposes an alternative technology. This entails a new distributed programming language, capable of describing both data and code in a platform-agnostic fashion. The underlying model is based on structured resources, each offering its own service. Service-oriented interfaces can be combined with the structured resources and hypermedia that characterize RESTful applications, instead of having to choose one style or the other. Coupling is reduced by checking interoperability structurally, based on the concepts of compliance and conformance. There is native support for binary data and full-duplex protocols.

1. INTRODUCTION

A distributed system has modules with independent lifecycles, each able to evolve to a new version without having to change, suspend or stop the behavior or interface of the others. These modules are built and executed in an independent way. Frequently, they are programmed in different programming languages and target different formats, platforms and processors. Distribution usually involves geographical dispersion, a network and static node addresses. Nevertheless, nothing prevents two different modules from sharing the same server, physical or virtual.

Modules are usually designed to interact, cooperating towards some goal. Since they are independent and make different assumptions, an interoperability problem arises. Interoperability, as old as networking, is a word formed by the juxtaposition of a prefix (inter) and the agglutination
of two other words (operate and ability), meaning literally “the ability of two or more system modules to operate together”.

With virtualization pervading all aspects of information technology, the cloud delivery model provides an increased decoupling between the availability of services and the details of their implementation, in a dynamic fashion. This hides many issues but raises others, namely:

- The variability introduced by cloud service models and deployment variants (private, managed, public hybrid cloud, integration with classical applications and combinations thereof);
- The plethora of cloud providers and brokers, with a vast range of computing resources, platforms, management services and user and developer APIs (Application Programming Interfaces).

This chapter adopts a broad perspective on cloud computing, in which a cloud is viewed as an elastic set of services, implemented by a set of heterogeneous resources that are interconnected by a set of heterogeneous networks. These resources can range from high-end servers with complex, enterprise-level applications, down to very simple sensors or actuators, without application deployment capabilities. Service unreliability and resource migration are expected characteristics.

Figure 1 illustrates various scenarios, in which services deployed on general-purpose clouds need to coexist with others, provided by a set of more specific resources. These can be managed in cloud fashion, dynamically using only what is needed and paying only for what is used. Although a cloud has been represented as the centerpiece in Figure 1, any system may connect directly to any other system.

In fact, the computer-based world is a giant, distributed, and heterogeneous cloud of clouds, some more dynamic than others, but all with the same basic problems:

- How to provide enough interoperability to enable systems to cooperate without requiring exact mutual knowledge;
- How to allow them to evolve independently;
- How to introduce context information, in particular in applications and networks involving mobility.

Typical general-purpose clouds provide a management API to support non-functional features, such as deploying resources, controlling and charging resource use, monitoring performance, and so on. This chapter emphasizes the functional slant. The cloud is viewed essentially as a dynamic and distributed platform, with on-demand provisioning and deployment (including migration). The main problem is to achieve interoperability between heterogeneous systems, deployed in heterogeneous platforms and expected to migrate and sometimes fail, while minimizing the required coupling between these systems.

The main goals of this chapter are:

- To contribute to the field of cloud computing, in particular in the interoperability area, by analyzing several of the problems involved and discussing possible solutions;
- To describe a framework that explores application interoperability at various levels, to better dissect and understand what is involved and how this can help in systematizing solutions;
- To assess the potential of structural compatibility (instead of nominal compatibility) as a means to reduce coupling and increase adaptability, changeability, and reliability, while maintaining interoperability requirements;
- To cater for cloud heterogeneity, by clearly separating the aspects involved in portability, transport-level communication, message protocol, syntax, semantics, and behavior, avoiding any dependence on any network protocol (such as HTTP – HyperText Transfer Protocol) or platform;