

# Chapter 5

## Middleware



### Abstract

In the context of IT applications and especially in large organizations, integration of existing information systems into new IT environments poses many challenges. One of the biggest issue in this regard is dealing with the systems' heterogeneity in terms of used programming languages, operating systems, or even data formats. In order to ensure communication between different information systems, developers must establish common interfaces. This chapter introduces middleware as a type of software which manages and facilitates interactions between applications across computing platforms. Besides a brief definition and overview of middleware, several of its characteristics are described. Furthermore, the differences between the three middleware categories (message-oriented, transaction-oriented and object-oriented middleware) are defined. In addition to these theoretical foundations, some practical implementations are presented.

### Learning Objectives of this Chapter

The primary learning objective of this chapter is to flesh out the concept, history, and current use of middleware. The chapter's main goal is to point out the difference between transaction-oriented middleware, message-oriented middleware, and object-oriented middleware. The subchapters present the key characteristics of each of these categories and their commercial implementation. Students will also learn about transaction processing monitors and remote procedure calls. The latter will be explained step by step and contrasted with local procedure calls. Finally, the chapter provides information about the well-known CORBA standard.

### Structure of this Chapter

The chapter's first section defines the overall concept of middleware and gives some brief historical background. This section is followed by a description of the best-known type of middleware: the remote procedure call. The third section forms the main part of this work and gives an overview of the different categories of middleware: message-oriented middleware, transaction-oriented middleware, and object-oriented middleware. Definitions, functionalities, commercial implementations, and well-known standards like CORBA are all included in the middleware categories' descriptions. The final section summarizes the whole chapter.

## 5.1 Introduction to Middleware

In today's digitally interconnected world, infrastructures are becoming more complex and diverse than ever before. At the same time, developers are facing increasing demands from an equally heterogeneous assortment of consumers around the globe for more powerful, reliable, and efficient applications. One way to facilitate the development process and create more flexible and reliable applications is to use sophisticated middleware. Middleware refers to application-neutral programs that mediate communication between applications in such a way that the complexity of these applications and their infrastructure is hidden from the user. Middleware can also be understood as a distribution platform (Ruh et al. 2002): It provides a solution to developers seeking to integrate a collection of servers and applications into a common service interface. From an application perspective, middleware is a service layer, which is used instead of an operating system interface. This is an appropriate course of action if the services offered are more powerful or the interface guarantees platform independence. For instance, in the context of mobile computing, middleware facilitates the development of platform-independent applications that are developed once but can be used across different mobile operating systems (e.g., Android and iOS) or even accessed via the Web.

At present, the term tends to be associated with a specific service area and three competing technologies. Two of these technologies are Common Object Request Broker Architecture (CORBA) and Distributed Component Object Model (DCOM). The most commonly used programming language, Java, contains a third, competing technology called Remote Method Invocation (RMI). These technologies will be discussed later in the chapter.

Middleware systems enable the distribution of applications to multiple computers in the network. The concept of 'Web services,' which will be introduced in Chapter 6, can be seen as a special type of middleware systems. Web services are an extension of the middleware concept, especially in terms of cross-organizational data exchange, which is of particular interest in business-to-business integration. The present chapter focuses on conventional types of middleware systems, which are explained in detail in the subchapters below.

### Middleware

Middleware is a type of software used to manage and facilitate interactions between applications across computing platforms.

In other areas, the term middleware has a different meaning. For example, in the field of computer game development, middleware may refer to subsystems for areas like game physics. Such middleware is often produced and offered by third-party developers.

The term itself has been in use since the late 1960s (Naur and Randell 1986). The understanding of middleware as a solution to the problem of linking newer