Chapter 7
Access Control in Service Compositions: Challenges and Solution Architecture

Aurélien Faravelon
Laboratoire d’Informatique de Grenoble, France

Stéphanie Chollet
Laboratoire de Conception et d’Intégration des Systèmes, France

ABSTRACT
Pervasive applications are entering the mainstream, but at the present time, exhibit significant security weaknesses. Service-driven architectural approaches facilitate the development of pervasive applications, however, security with respect to access control and data privacy of pervasive applications are currently not managed comprehensively from design time through run time. This chapter presents a use case emphasizing the security challenges for pervasive applications and proposes a novel, generative architectural approach, to include security in pervasive applications at design time. This is a model-driven approach based on models pertaining to access control management that respect the temporal constraints relating to pervasive applications. The approach is implemented with a design and runtime environment and the results of the validation applied to the pervasive use case are presented.

INTRODUCTION
Healthcare costs are skyrocketing as the world’s population ages, due to these aging members requiring on average more frequent hospitalization. A solution to mitigate medical costs is to enable these senior citizens to receive medical care while remaining in their homes, by adapting houses to their needs. This includes regular monitoring and interaction with medical staff and family. Pervasive computing, which is integrating computational artifacts into the fabric of our daily lives, is a promising paradigm to enable a safe, and well-connected medical experience in homes. Technologies to perpetuate this kind of home environment already exist, including residential high-speed Internet connections and a wide range of home sensors for health monitoring. Coordinat-
ing communication between these various available computing units and technologies therefore becomes crucial. Furthermore, for privacy and security reasons, the access to these individual units must be controlled from the perspective of the entire system, to prevent situations in which a unit causes data leakage or other harm.

The software engineering community strives to produce new answers to these kinds of relevant challenges. Service Oriented Computing (SOC) (Papazoglou, 2003) is one of the answers. SOC relies on the notion of service that can manifest as anything from software to infrastructure. A service provides a set of well-defined characteristics and possesses a description that expresses both its functional and non-functional properties. Web Services are the best known services, but several other implementations exist, including UPnP, DPWS, and OSGi, all of which are especially interesting because they allow for exposing devices as services.

Services are assembled using Service-driven approaches based on Service Oriented Architecture (SOA). SOA provides mechanisms to specify, publish, discover and compose services. Service compositions are defined according to the control, which can be internal or external to the service. When the control is internal to the service, services interact directly. When the control is external, services are composed according to a directed graph that represents a process. Processes are traditionally specified through languages such as WS-BPEL (Web Service Business Process Execution Language) and then interpreted by an execution engine.

Two features of Service-driven architectural approaches are of particular interest. First, service consumers do not have to deal with the heterogeneity of services. Composition’s elements are independent. Services can thus evolve independently. This is especially important when an application must rely on loosely coupled elements that were not meant to work together. Services may be substituted for one another as long as they implement the functional interface defined by the service provider. Next, since services are capable of late-binding, service-driven approaches address the dynamism of services.

However, Service-driven architectural approaches still suffer from some major hurdles. Dynamic service compositions are complex, especially when they have to enforce an access control policy defined at an application-wide level. Two main difficulties remain:

1. Services are highly heterogeneous as there are multiple technologies and varied implementations of them, as indicated earlier. As a consequence, developing dynamic service compositions require cross-technology skills that are rare.

2. Actual services are not necessarily known at design time of the application. Thus, the access control capabilities of the services cannot be trusted. Services may or may not offer access control capabilities and when they do so, they may not enforce the right access control features.

In this chapter, we will focus on the following pervasive application scenario:

- The house monitors the medical condition of its inhabitants.
- On a regular basis, the house generates reports:
  - Medical reports are available to physicians
  - Nurses can access reports necessary to perform their daily visits
  - Family members can read summarized reports to make sure that their relatives are OK.
- At any time, if the inhabitants are in danger, the house generates alerts and sends them to:
  - The physicians
  - The family members