Chapter 89
Software Development Using Service Syndication based on API Handshake Approach between Cloud-Based and SOA-Based Reusable Services

Vishav Vir Singh
Intersil Corporation, USA

ABSTRACT

Reusability in the software development realm is a momentous cornerstone in deciding the success of an organization. The quantum leap in reusability was provided by object-orientation, which fundamentally altered the manner of thinking of software architects. The notion of the object and the installation of reusability as one of the central tenets is a critical requisite. Components that represented bigger and better software aggregates easily utilized by businesses to gain instant execution for their business services have made great strides. This gave way to the ultimate notion of a service which took software aggregates in components to fine and sharpened streams of software execution called services. This has been another inflection point in the overall curve of reusability, and most importantly, in agility and turnaround time for software production. The notion of services mandated the birth of a secure and trusted public interface to a service, which is called the Application Programming Interface (API). APIs represent a directory of services that a given enterprise offers.

DOI: 10.4018/978-1-4666-4301-7.ch089

INTRODUCTION

Upon fulfillment of contractual obligations between the service provider and consumer, a process known as API Handshake ensues and brings the two parties onto a common collaboration plane. On the enterprise side many infant services may be amalgamated to constitute composite services, a process known as Service Syndication. Syndication obviates the need to create new services from scratch and greatly boosts reusability in an organization. This chapter follows the contours of evolution from object orientation to Cloud Computing and Service-oriented Architecture (SOA) to understand reusability in the context of service syndication based on the API handshake approach between diverse systems.

The transcendence of massive, rigid and uniform systems into co-acting, flexible and interconnected systems has been an evolutionary requisite. Almost two decades ago, the notion of object-orientation found its genesis in the software development paradigms. And a decade later, in came the messiahs of technology-neutral pluggability – components. This was the cornerstone of the very concept of reusability of an information asset so that it could be aggregated into any solution targeting a problem domain. The more recent advances in research into architectures and their orientation around services is a direct derivative of these initial baseline agents.

The two primary doctrines of code reusability and technology independence paved the way for the formation of a new generation computing platform that addressed these two golden principles with its own hallmark design paradigms, solution models, and technology deployment archetypes. It was the advent of the Software Oriented Architecture (SOA). SOA aims to carve and sculpt solution logic in a way that is very clean-cut, definitive and highly targeted towards the problem domain in the form of discrete and spearheaded streams of functionality. Such solution code representation is the service-orientation of architecture. This way of organizing solution model is greatly focused on the principles of loose coupling, reusability and interoperability.

A great chance of failure awaits organizations trying to share and expose services or components via SOA without furnishing a homogeneous programming framework with a very clear-cut and well-defined array of rules, constraints and principle of behavior. The public designations of such a model constitute the Application Programming Interface (API). An API lays out the syntax and semantics of declaration of objects, methods, classes, protocols and invoking mechanisms through exposed interfaces that are used as a communication medium between the provider and consumer of services.

A subsequent endeavor went underway to provide on-demand access to not just services aspired by SOA, but to any IT resource including assets that publish processes, databases, services and hardware. This approach was named Cloud Computing. While SOA concentrates on providing solution models oriented around services within the confines of an enterprise, cloud computing aims at graduating SOA to outside enterprise firewalls catering to the on-demand provisioning of any IT resource available as a service over the Internet. When the cloud-based software services are to be aggregated into a solution logic unit by the consumers, a handshake occurs between the consumer solution module and the cloud provider API which allows the consumer to leverage all the power of the service in its environment.

This chapter delves into a study of this orchestration of services and their APIs within the purview of disparate cloud-based systems using the notion of reusability via syndication. This powerful composition principle can be applied to render a well-adaptive, time-efficient and agile software development process using third-party services.