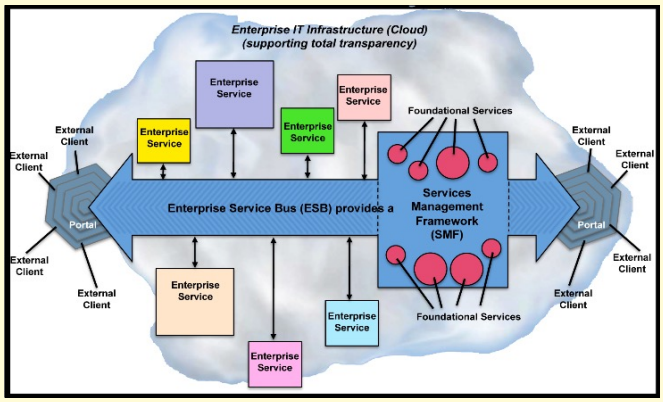
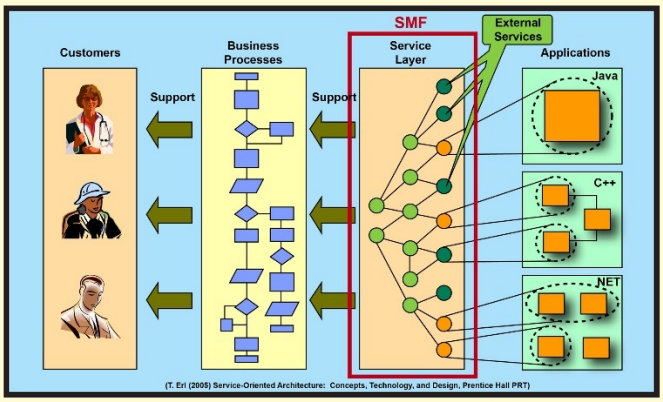
Service-Oriented Architecture (SOA)

The notion of service-oriented is ubiquitous. Everywhere we see countless examples of tasks being performed by a combination of services, which are able to interoperate in a manner that results in the achievement of a desired objective. Typically, each of these services is not only reusable but also sufficiently decoupled from the final objective to be useful for the performance of several somewhat similar tasks that may lead to quite different results. For example, a common knife can be used in the kitchen for preparing vegetables, or for peeling an orange, or for physical combat, or as a makeshift screwdriver. In each case the service provided by the knife is only one of the services that are required to complete the task. Clearly, the ability to design and implement a complex process through the application of many specialized services in a particular sequence has been responsible for most of mankind's achievements in the physical world. The key to the success of this approach is the interface, which allows each service to be utilized in a manner that ensures that the end-product of one service becomes the starting point of another service.





In the software domain these same concepts have gradually led to the adoption of Service-Oriented Architecture (SOA) principles. While SOA is by no means a new concept in the software industry it was not until Web services became available that these concepts could be readily implemented. In the broadest sense SOA is a software framework for computational resources to provide services to customers, such as other services or users. The Organization for the Advancement of Structured Information (OASIS) defines SOA as a "… paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains" and "…provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects with measurable preconditions and expectations". This definition underscores the fundamental intent that is embodied in the SOA paradigm, namely flexibility. To be as flexible as possible a SOA environment is highly modular, platform independent, compliant with standards, and incorporates mechanisms for identifying, categorizing, provisioning, delivering, and monitoring services.

A core component of a SOA-based software system is an Enterprise Service Bus (ESB) that provides the communication bridge through which the tasks performed by the services are coordinated. The management functions provided by the ESB are critical to the successful operation of the system, because each individual service is capable of performing only a narrow set of tasks, is unaware of the existence of any other services, and has no understanding of or interest in the end-result produced by the system. Therefore, the ESB needs to be able to route any request for services to a service that can perform the requested task, translate the request into the form that the task provider expects the request to be in, and again translate the results of the performed tasks into the form expected by the requester. For example, Service-A sends a request for the current location of a particular item that has been ordered by a customer. The ESB receives the request, translates it into one form that is expected by Service-B with responsibility for inventory and another form for Service-C with responsibility for in-transit tracking of shipments. Similarly, the ESB will translate the responses received by Service-B and Service-C into the required form for Service-A. In more technical terms ESB management services include, routing, protocol transformation, message transformation, service mapping, message processing, process orchestration, transaction management, and access control.