

# WHAT IS SOA? SERVICE-ORIENTED ARCHITECTURE EXPLAINED



## What Is Service-Oriented Architecture?

SOA is a structure that allows services to communicate with each other across different platforms and languages by implementing what is known as a "loose coupling" system. While the concept of SOA has been around for many years, it is only within the past decade that it has risen to the forefront of software-related technologies.

The term "loose coupling" refers to the client of a service, and its ability to remain independent of the service that it requires. The most important part of this concept is that the client, which in itself can be a service, can communicate with the service even if they are not closely related. This facilitated communication is achieved through the implementation of a specified interface that is able to perform the necessary actions to allow for the transmission of data.

A common example of this increased ability to communicate without service constraints involves coding languages used by these services. There is an array of different languages from which software platforms are created and not all of these languages can interact fluently, without encountering communication issues. By using an SOA, it is not necessary for the client to understand the language that is being used by the service, but instead, it relies on a structured interface that is able to process the transmission between the service and the client.

SOA is often differentiated from [Microservices Architecture \(MSA\)](#). Explore the [differences between](#)

[SOA and Microservices](#) in this blog post.

# The importance of Service-Oriented Architecture

There are a variety of ways that implementing an SOA structure can benefit a business, particularly, those that are based around web services. Here are some of the foremost:

- **Creates reusable code**

The primary motivator for companies to switch to an SOA is the ability to reuse code for different applications. By reusing code that already exists within a service, enterprises can significantly reduce the time that is spent during the development process. Not only does the ability to reuse services decrease time constraints, but it also lowers costs that are often incurred during the development of applications. Since SOA allows varying languages to communicate through a central interface, this means that application engineers do not need to be concerned with the type of environment in which these services will be run. Instead, they only need to focus on the public interface that is being used.

- **Promotes interaction**

A major advantage in using SOA is the level of interoperability that can be achieved when properly implemented. With SOA, no longer will communication between platforms be hindered in operation by the languages on which they are built. Once a standardized communication protocol has been put in place, the platform systems and the varying languages can remain independent of each other, while still being able to transmit data between clients and services. Adding to this level of interoperability is the fact that SOA can negotiate firewalls, thus ensuring that companies can share services that are vital to operations.

- **Allows for scalability**

When developing applications for web services, one issue that is of concern is the ability to increase the scale of the service to meet the needs of the client. All too often, the dependencies that are required for applications to communicate with different services inhibit the potential for scalability. However, with SOA this is not the case. By using an SOA where there is a standard communication protocol in place, enterprises can drastically reduce the level of interaction that is required between clients and services, and this reduction means that applications can be scaled without putting added pressure on the application, as would be the case in a tightly-coupled environment.

- **Reduced costs**

In business, the ability to reduce costs while still maintaining a desired level of output is vital to success, and this concept holds true with customized service solutions as well. By switching to an SOA-based system, businesses can limit the level of analysis that is often required when developing customized solutions for specific applications. This cost reduction is facilitated by the fact that loosely coupled systems are easier to maintain and do not necessitate the need for costly development and analysis. Furthermore, the increasing popularity in SOA means reusable business functions are becoming commonplace for web services which drive costs lower.

## Summary

There is no doubt that as web application technologies continue to evolve there will be an increase in the number of businesses that utilize the power of SOA. By switching to a standardized

communication protocol, engineers will be able to create software applications without having to worry about the languages on which platforms are built, and can instead rely upon the interoperability that the SOA structure creates. In addition, SOA can help ensure that applications can be easily scaled, while at the same time decreasing the costs that are often encountered when developing business service solutions.