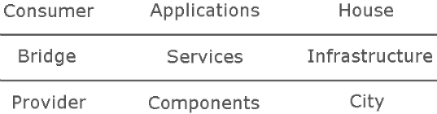
SOA could be described in three layers as seen in the picture below. On one side we have the **Provider** and on the other side we have the **Consumer**, separated by a **Bridge** where the two sides communicate.



The consumer uses a number of Applications necessary for it's business and the provider uses Components that provide these applications with information. They communicate through a set of Services using a common architecture.

If you're an architect SOA is the solution to design loosely coupled services over an integration platform, to just plug services into outlets.

If you're a developer, SOA is a programming paradigm where a service is in the centre of the design and the code.

Service-oriented architecture (SOA) is a multitier architecture in which application functionality is encapsulated in services.

SOA services are usually implemented as Web services.

Web services are accessible through HTTP and are based on XML-based standards such as Web Services Description Language (WSDL) and SOAP.

Service-oriented architecture (SOA) is an evolution of distributed computing based on the request/reply design paradigm for synchronous and asynchronous applications.

An application's business logic or individual functions are modularized and presented as services for consumer/client applications.

What's key to these services is their loosely coupled nature; i.e., the service interface is independent of the implementation.

Application developers can build applications by composing one or more services without knowing the services' underlying implementations.