# Architectures

## Layered Architecture

## Client-Server Architecture

## Service Oriented Architecture

SOA architecture will package functionality as [*interoperable*](http://en.wikipedia.org/wiki/Interoperability) [*services*](http://en.wikipedia.org/wiki/Service_%28systems_architecture%29): software modules provided as a service can be integrated or used by several organizations, even if their respective [client](http://en.wikipedia.org/wiki/Client_%28computing%29) systems are substantially different.

SOA are based on a mesh of software services. Services comprise unassociated, [loosely coupled](http://en.wikipedia.org/wiki/Loosely_coupled) units of functionality that have no [calls](http://en.wikipedia.org/wiki/Subroutine) to each other embedded in them.Instead of services embedding calls to each other in their source code they use defined protocols that describe how services pass and parse messages, using description meta-data.

Often, [WSDL](http://en.wikipedia.org/wiki/Web_Services_Description_Language) typically describe the services themselves, while [SOAP](http://en.wikipedia.org/wiki/SOAP_%28protocol%29) describes the communications protocols.

* Reuse, [granularity](http://en.wikipedia.org/wiki/Granularity), [modularity](http://en.wikipedia.org/wiki/Modularity_%28programming%29), composability, componentization and [interoperability](http://en.wikipedia.org/wiki/Interoperability).
* Standards compliance (both common and industry-specific).
* Services identification and categorization, provisioning and delivery, and monitoring and tracking.

## Data Driven Architecture

## Inversion of Control (Architecture?)