

Microservices and Monolithic , difference between REST and SOAP

monolithic architecture

- **A monolithic architecture** is a traditional model of a software program, which is built as a unified unit that is self-contained and independent from other applications.
- The word “monolith” is often attributed to something large and glacial, which isn’t far from the truth of a monolith architecture for software design.
- A monolithic architecture is a singular, large computing network with one code base that couples all of the business concerns together.
- To make a change to this sort of application requires updating the entire stack by accessing the code base and building and deploying an updated version of the service-side interface.
- This makes updates restrictive and time-consuming.

microservices architecture

- A microservices architecture, also simply known as microservices, is an architectural method that relies on a series of independently deployable services.
- These services have their own business logic and database with a specific goal. Updating, testing, deployment, and scaling occur within each service.
- Microservices decouple major business, domain-specific concerns into separate, independent code bases.
- Microservices don’t reduce complexity, but they make any complexity visible and more manageable by separating tasks into smaller processes that function independently of each other and contribute to the overall whole.

SOAP

SOAP stands for Simple Object Access Protocol

SOAP is a protocol. SOAP was designed with a specification. It includes a WSDL file which has the required information on what the web service does in addition to the location of the web service.

SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern.

SOAP uses service interfaces to expose its functionality to client applications. In SOAP, the WSDL file provides the client with the necessary information which can be used to understand what services the web service can offer.

REST

REST stands for Representational State Transfer

REST is an Architectural style in which a web service can only be treated as a RESTful service if it follows the constraints of being

Client Server

Stateless

Cacheable

Layered System

Uniform Interface

REST can make use of SOAP as the underlying protocol for web services, because in the end it is just an architectural pattern.

REST use Uniform Service locators to access to the components on the hardware device. For example, if there is an object which represents the data of an employee hosted on a URL .