# Software Architecture:

## SOLID Principles

SOLID is an acronym for the first five object-oriented design (OOD) principles by Robert C. Martin (also known as [Uncle Bob](http://en.wikipedia.org/wiki/Robert_Cecil_Martin)).

* S: Single Responsibility Principle
* O: Open Close Principle
* L: Liskov’s Substitution principle
* I: Interface Segregation Principle
* D: Dependency Inversion Principle

**Single Responsibility Principle**:

Ability to define purpose of an entity in a single statement.

Entity must adhere to this purpose throughout its lifecycle

Benefits:

* Easy testing
* Controlled dependencies
* Advancement on monolith styled entities

**Open – Close Principle**

Open for extension, closed for modification

Benefits:

* Reduce risk of potential bugs in current stable application

**Liskov Substitution Principle:**

Child instance should be able to behave as Parent instance

Child instance should act as a substitute for parent instance

Child should seamlessly show behaviors of parent

**Interface Segregation Principle:**

Monolith interfaces should be segregated into smaller interfaces based on the purpose category

**Dependency Inversion:**

Develop loosely coupled software modules

High level modules should not be dependent on low level modules. Both should depend on abstractions

Abstract entities should be dependent on concrete/implemented entities

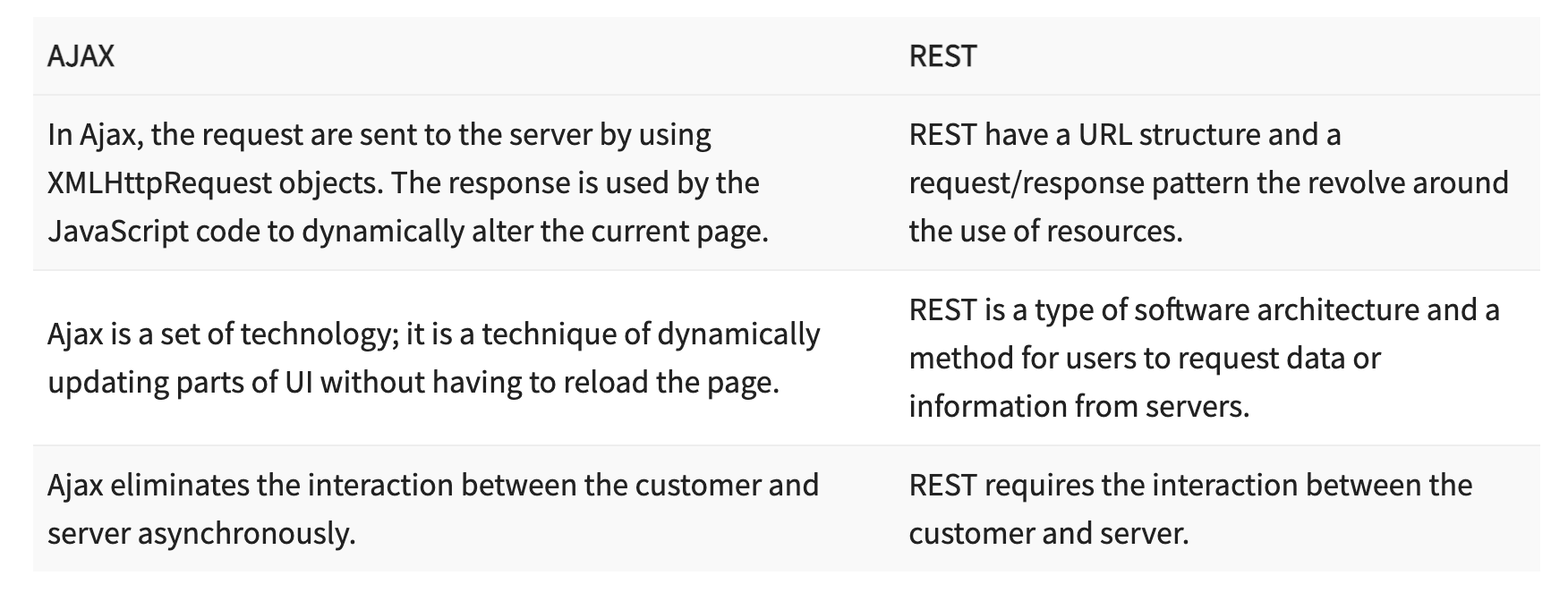
# ReST:

* Stands for: Representational State Transfer
* Stateless services, i.e, server has no session data
* Between two ReST service calls, we can restart the server, without impacting accuracy of operations

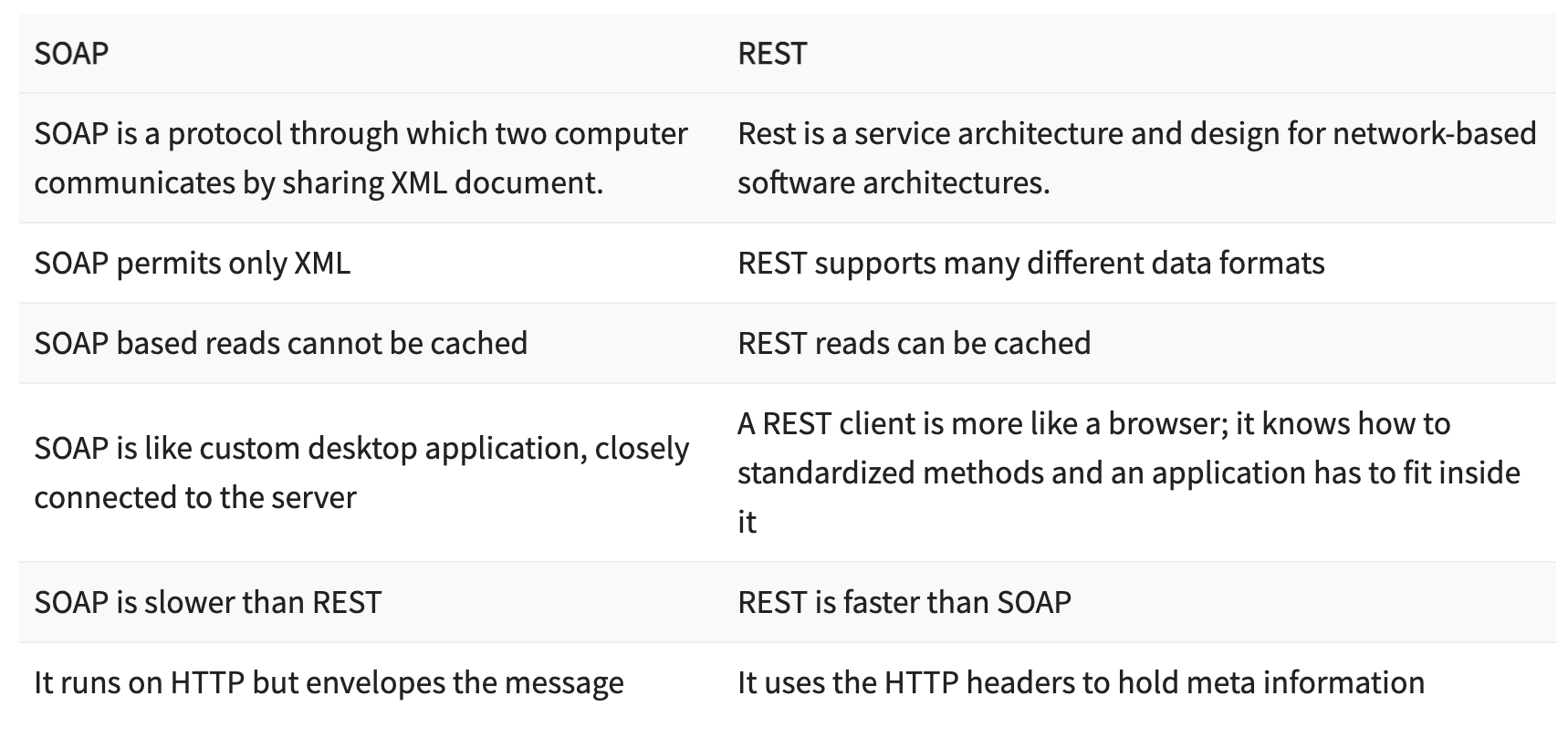
## HTTP methods supported in ReST:

* GET
* POST
* PUT
* DELETE
* OPTIONS
* HEAD

## Difference between AJAX and ReST



## Difference between SOAP and ReST



# Python

## Scopes in Python

* **Local**

Refers to objects inside a function/block

* **Global**

Refers to objects in main body of Python code

* **Module level**

Refers to global objects of current module

* **Outermost**

Included built-in functions and constants in Python

## Data types in Python

# Artificial Intelligence

## Reward based learning

* Learning based on rewards gained as result of an action

## Generalized learning

* Applying action to similar scenario based on learning of past