service oriented architectures

- what is a service?

  independently deployed (has its own lifecycle), well-documented

- historical perspective

  monolithic, often compiled into a single executable

  ->

  standard web protocols led to decoupling, SOA became common

  ->

  microservices, containers, thus services became smaller & smaller

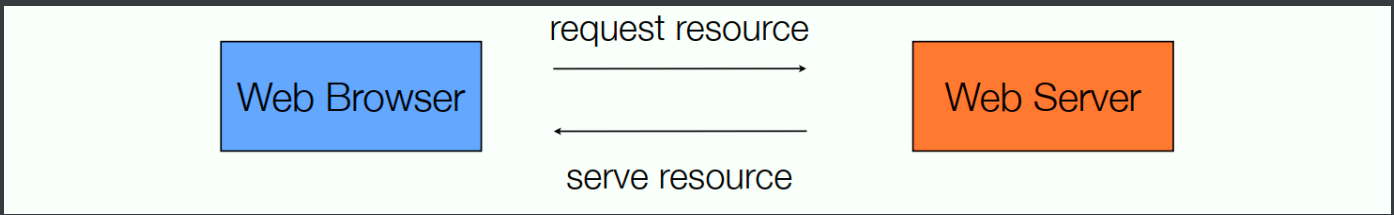
  SOAP-based -> **REST-based**

the (world wide) web

distributed information system for publishing & accessing resources & services across the internet,

key feature is hyperlinked documents

client-server architecture: user agent (browser) & web server



HTTP

application-layer request-response protocol, methods (GET/POST...)

GET should be idempotent: 多次发送request不会影响resource的状态

POST is not idempotent: 多次发送是不安全的，agent需要询问用户意见

stateless: each request is independent from others, easy to scale

session: 分组request，有session id和cookies等方法实现

web services

machine to machine interaction

HTTP was designed to fetch resources, not to invoke operations

-> RPC on top of HTTP, such as SOAP

SOAP-based web services

SOAP = simple object access protocol, an RPC protocol

SOAP + XML

but SOAP-based services layer an extra request-reply RPC protocol on top of HTTP, which si already a request-reply protocol; and XML's encoding is very not light-weighted,

-> REST-based + json