Gateway Pattern

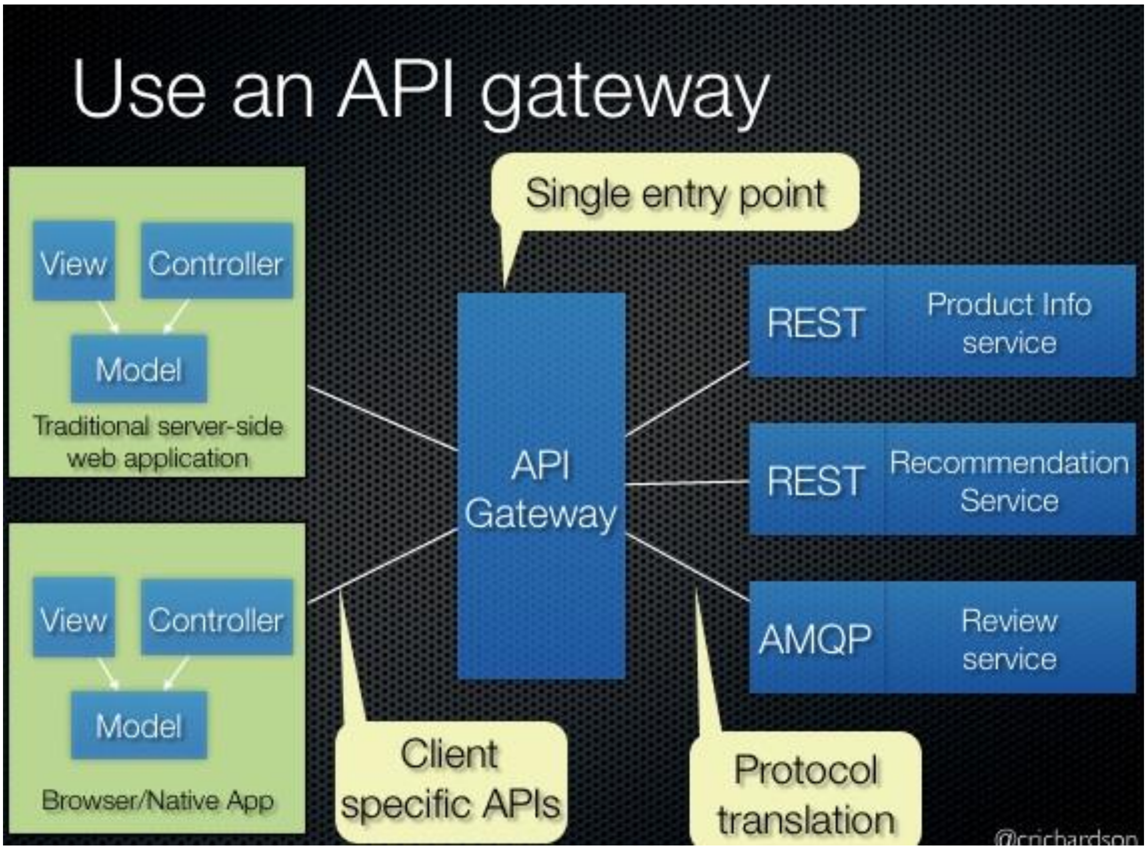
Problem

In complex microservice landscape clients must access individual dynamic service instances. There are the following challenges in this scenario:

1. The number of individual microservice instances and their endpoints can be changed dynamically that affects all clients
2. Partitioning and granularity of services can change over time and should be hidden from clients
3. Common functionalities like audit, security, monitoring must be applied for all service requests
4. Content type or header conversions can be required in some cases

Solution

Implement an API gateway that is the single-entry point for all clients. The API gateway handles requests in one of two ways. Some requests are simply proxied/routed to the appropriate service. It handles other requests by fanning out to multiple services



Rather than provide a one-size-fits-all style API, the API gateway can expose a different API for each client. There some out the box implementation of Gateway: Apigee, Amazon Gateway, Netflix API Gateway. SpringBoot Cloud provide easy way to implement own Gateway solution.

Tutorial

To run tutorial the following steps are necessary:

- Checkout maven projects from GitHub Gateway
- Start service-foo
- Start service-bar
- Test direct requests:
  `[curl http://localhost:8081/foo/hello]`
  `[curl http://localhost:8082/bar/hello]`
- Start gateway project
- Access microservices using central gateway:
  `[curl http://localhost:8080/foo/hello]`
  `[curl http://localhost:8080/bar/hello]`