API Workshop

Gateways

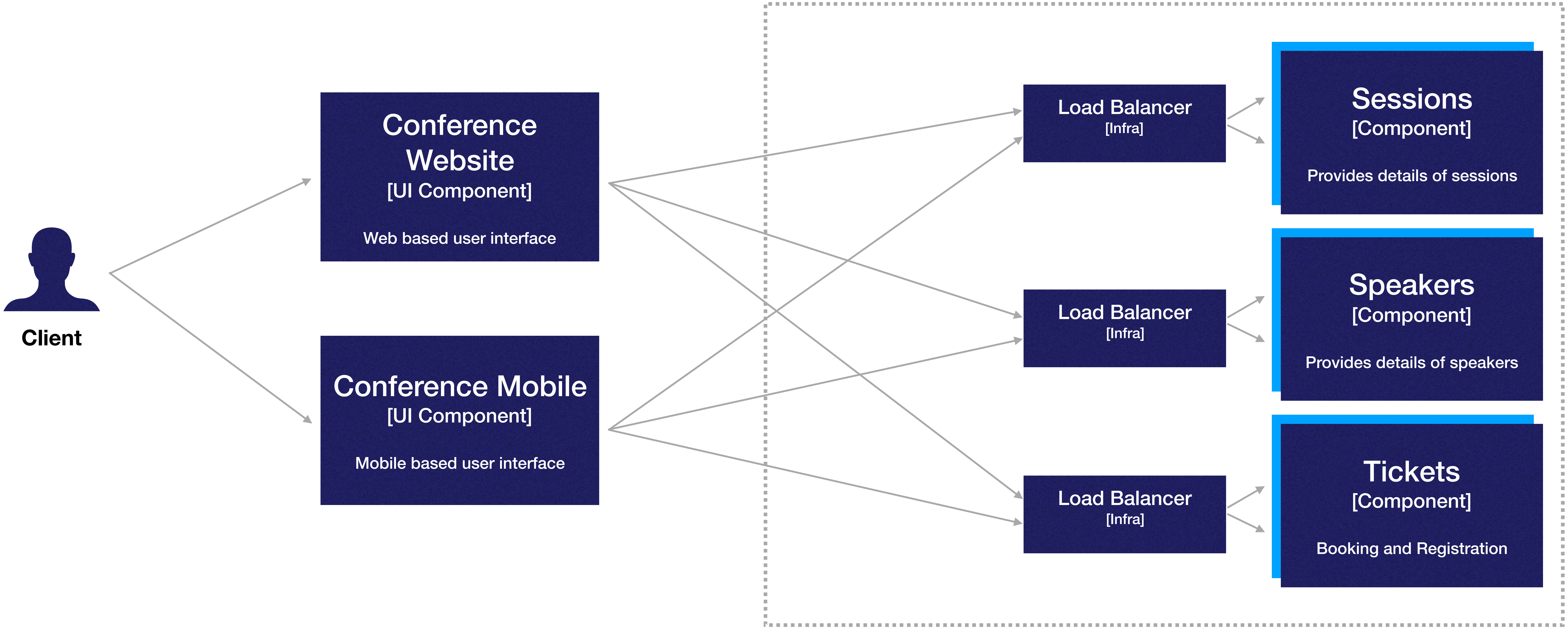
O'REILLY®

Software Architecture

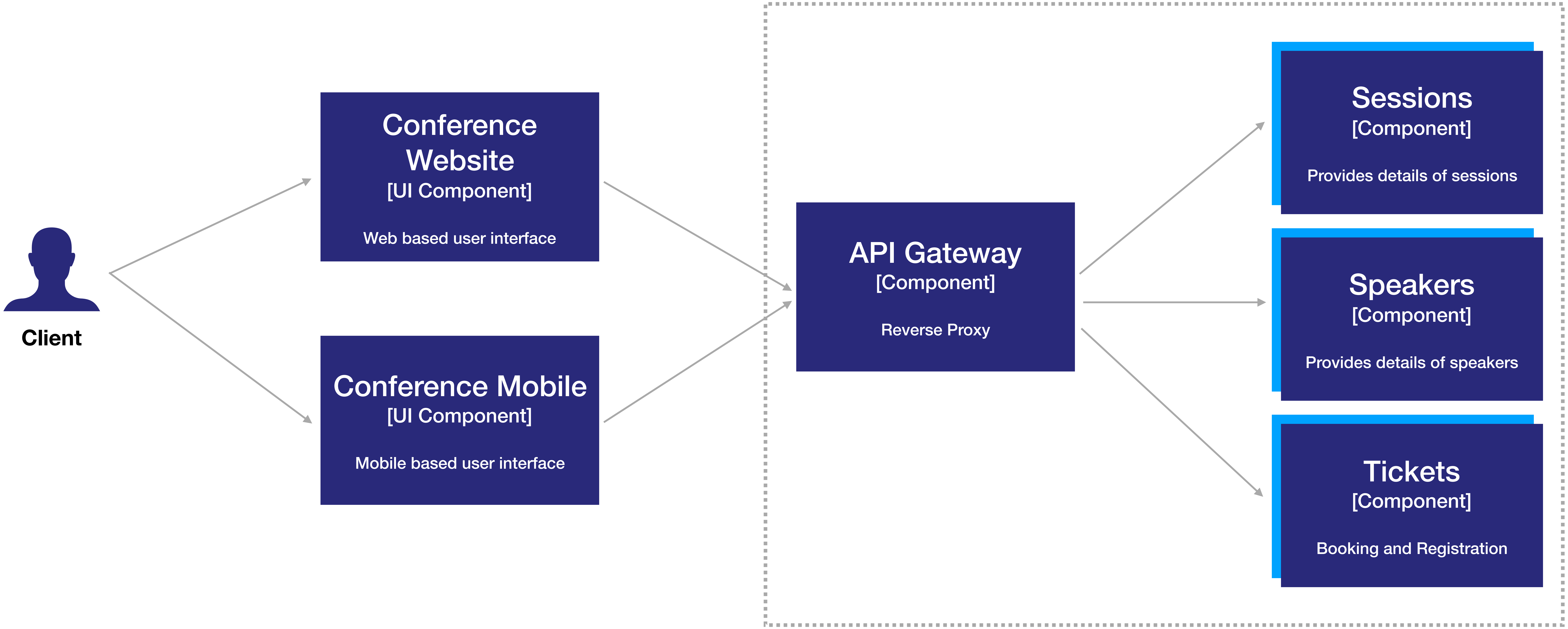
Agenda

- What is a gateway
- Different types of gateway
- Words of caution
- Demo - Applying a gateway to our solution

What is a Gateway?



What is a Gateway?



Client

Conference Website
[UI Component]

Web based user interface

Conference Mobile
[UI Component]

Mobile based user interface

API Gateway
[Component]

Reverse Proxy

Sessions
[Component]

Provides details of sessions

Speakers
[Component]

Provides details of speakers

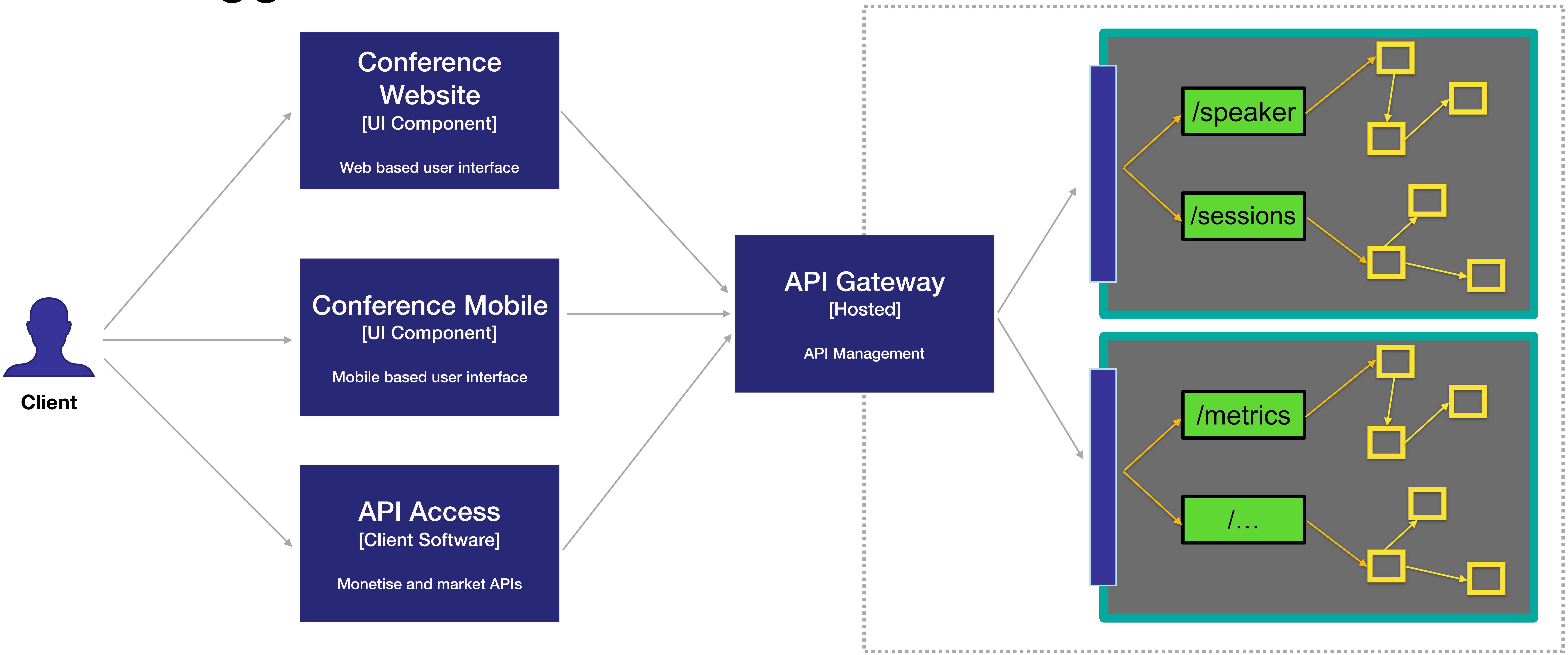
Tickets
[Component]

Booking and Registration

Differences Between Gateway

Credit - Ambassador Docs (https://www.getambassador.io/about/microservices-api-gateways/)	Enterprise API Gateway	Microservice Gateway
Goal	Provide an API Marketplace	Internal Services
Deployment	Admin API or Team Managed	DevOps Deployed
Metrics	Invocation Rate/HTTP Status	Latency, Traffic
Errors	Custom Errors for Clients	Full Detail of Error
Testing	Staging and Production Promotion	Canary Releases
Development	Docker if Needed	Local Docker/Kubernetes Deployment

The Bigger Picture



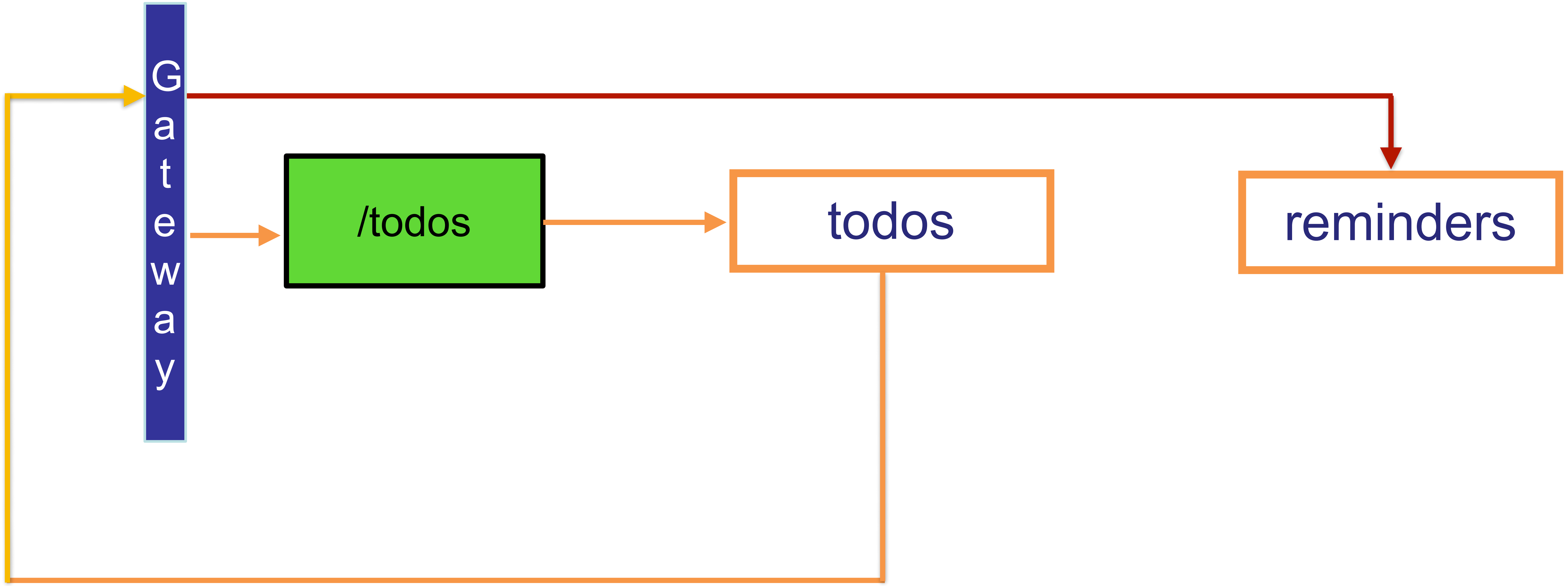
Words of Caution

We remain concerned about business logic and process orchestration implemented in middleware, especially where it requires expert skills and tooling while creating single points of scaling and control. Vendors in the highly competitive API gateway market are continuing this trend by adding features through which they attempt to differentiate their products.

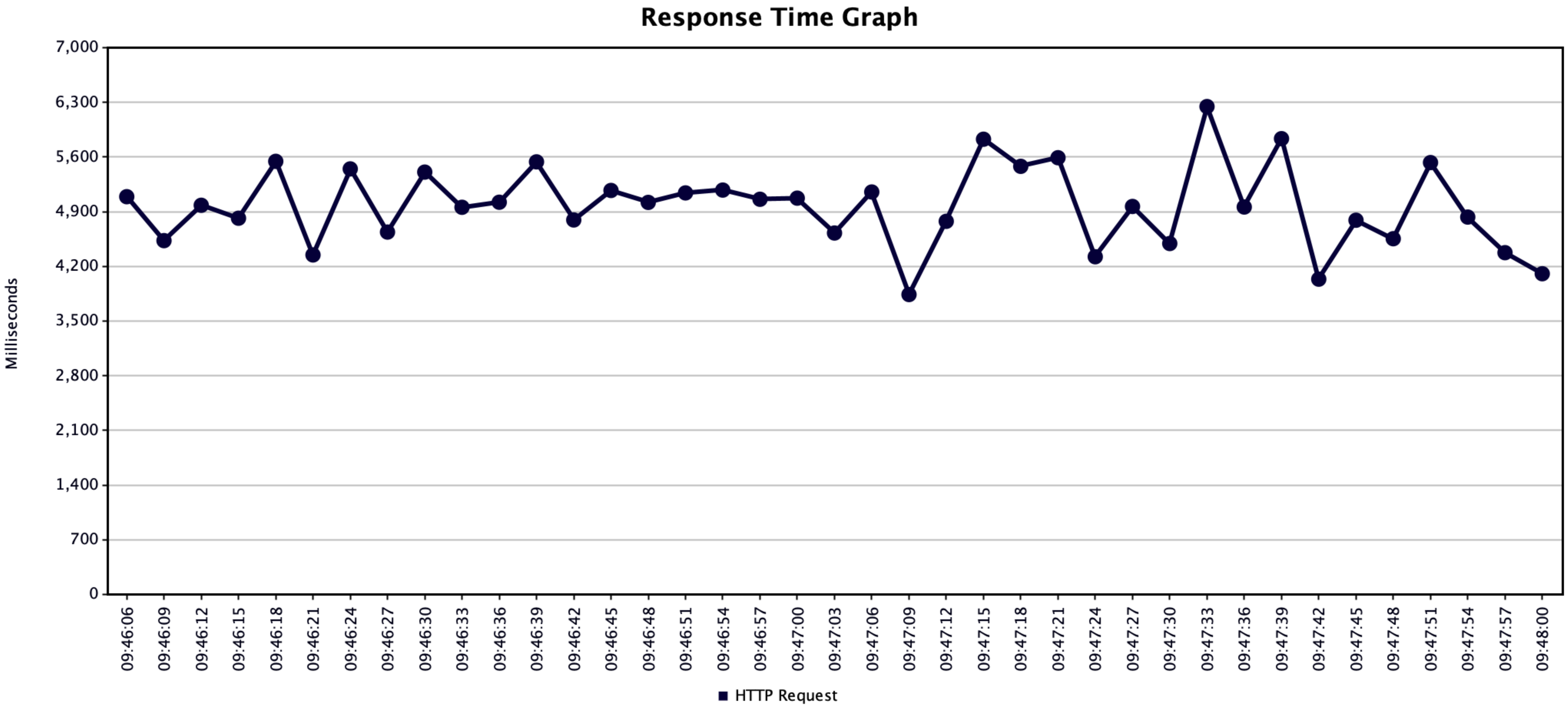
This results in **overambitious API gateway** products whose functionality — on top of what is essentially a reverse proxy — encourages designs that continue to be difficult to test and deploy. API gateways do provide utility in dealing with some specific concerns — such as authentication and rate limiting — but any domain smarts should live in applications or services.

Thoughtworks Technology Radar

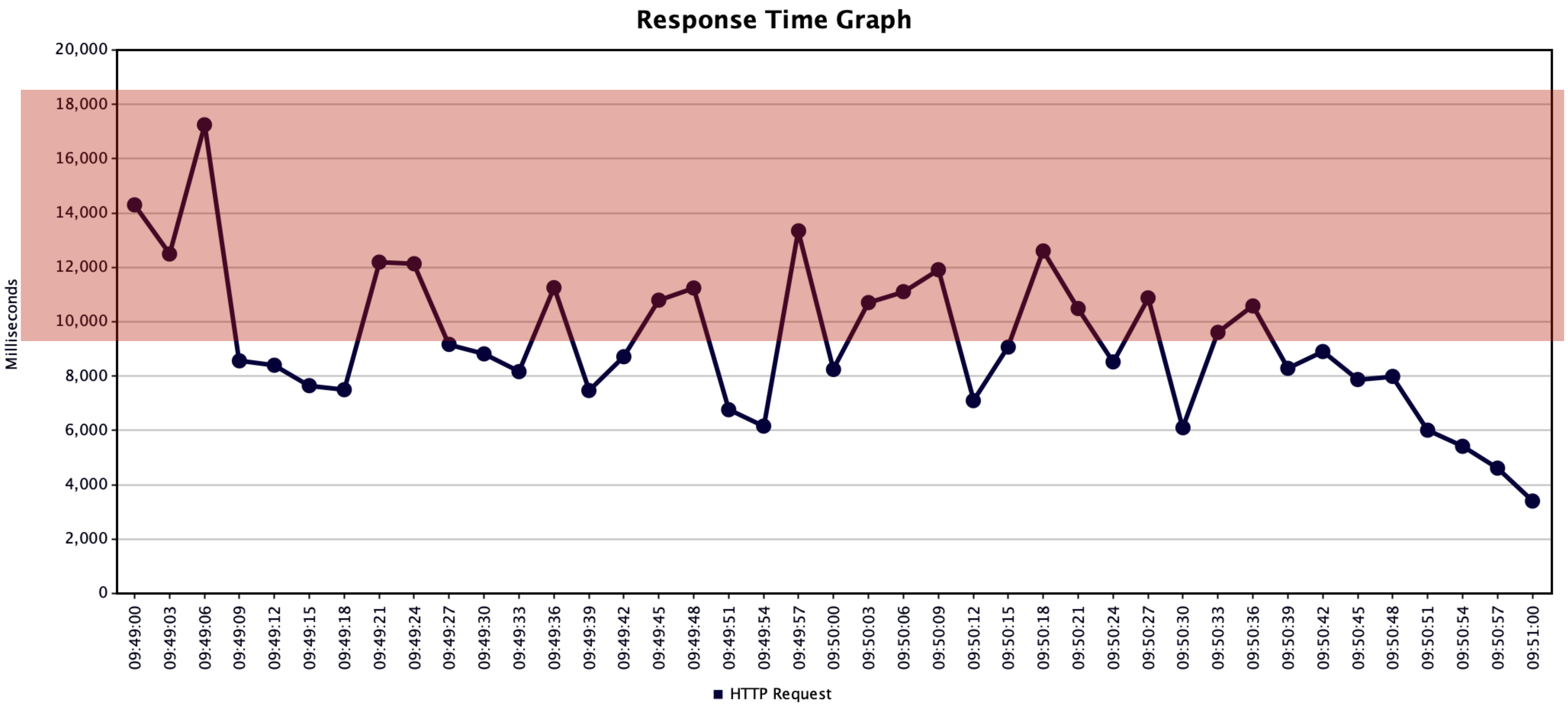
Saturation of Gateways Happens Quickly



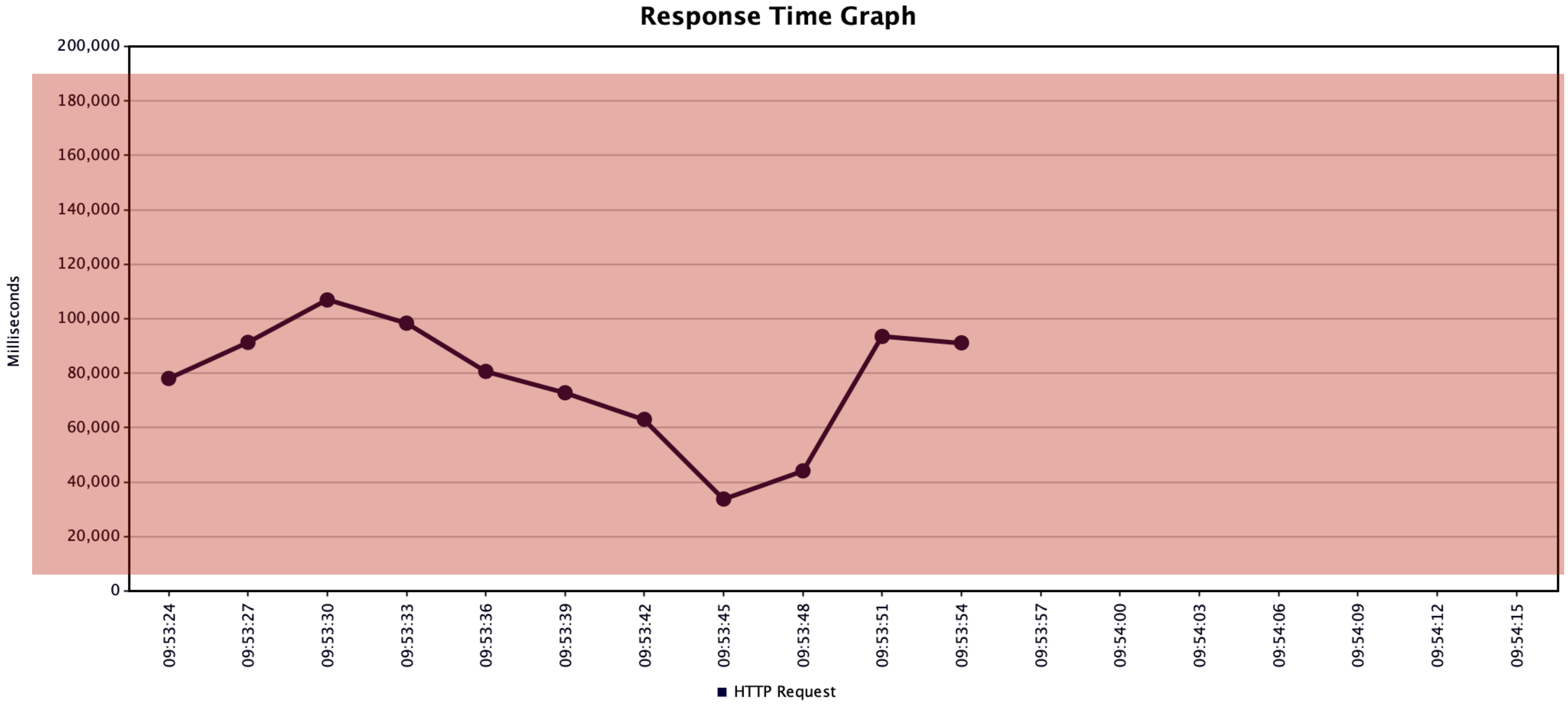
Gateway Loopback - 50 users, 2 Internal Requests, Delay 1s



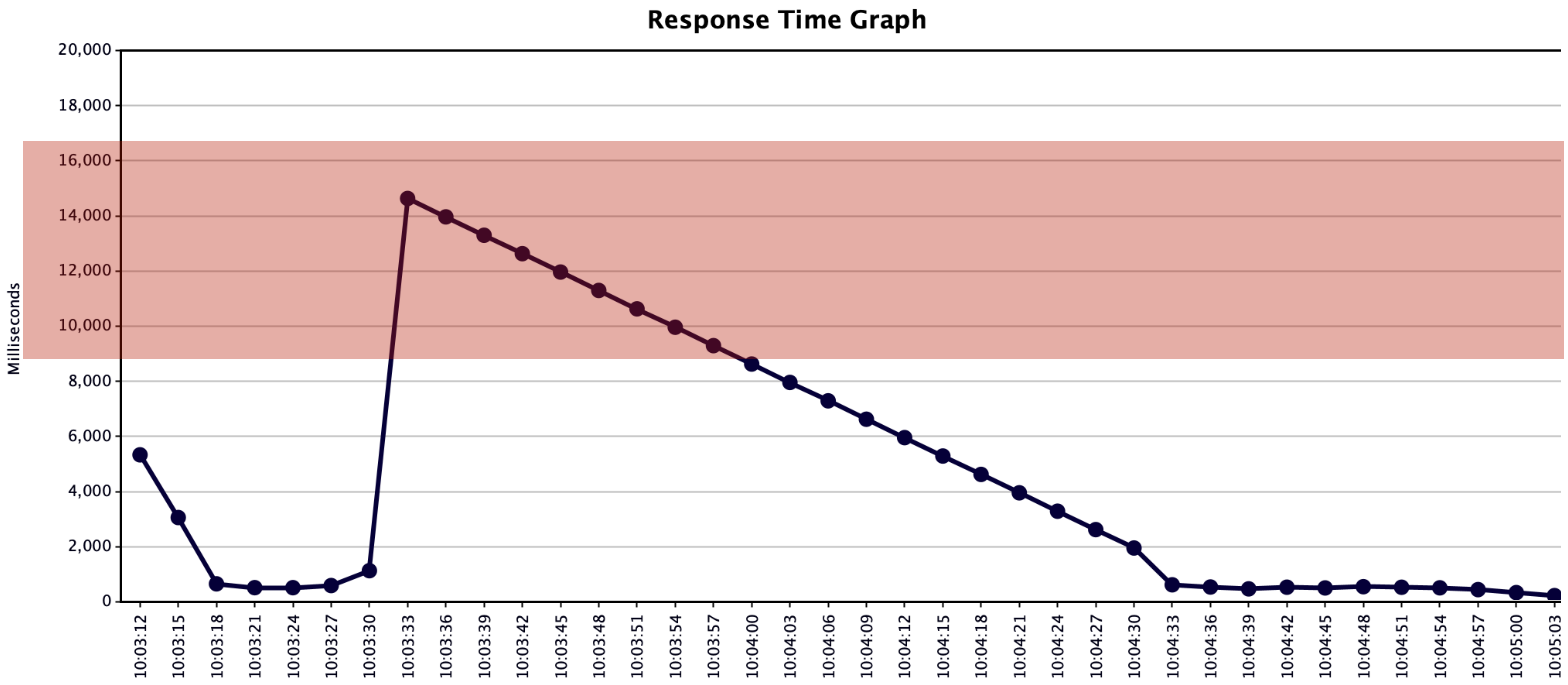
Gateway Loopback - 100 users, 2 Internal Requests, Delay 1s



Gateway Loopback - 150 users, 2 Internal Requests, Delay 1s



Gateway Loopback - 200 users, 2 Internal Requests, Delay 0s



Unexpected Edge Conditions

- Can a RESTful **Delete** request have a body?
- A payload within a **DELETE** request message has no defined semantics; sending a payload body on a **DELETE** request might cause some existing implementations to reject the request.
- The latest un-approved version of the spec removes this requirement. The latest approved version is still the **RFC2616** quoted above
- **RFC 7231** section 4.3.5 finalizes the language from version 26 with A payload within a DELETE request message has no defined semantics. So the body is allowed.

Sample Discussion from Stackoverfow

Demo - Applying a Gateway

Creates a Network

Builds out an image

Pulls in a gateway

```
version: '3'
services:
  task-service:
    build:
      context: ./
      dockerfile: Dockerfile
    image: "task-service:latest"
    ports:
      - "8080:8080"
  api-gateway:
    image: jpgough/api-workshop-gateway
    ports:
      - "8081:8081"
```

Demo - Applying a Gateway

```java
@SpringBootApplication
public class GatewayApplication {

    @Bean
    public RouteLocator customRouteLocator(RouteLocatorBuilder builder) {
        return builder.routes()
               .route("tasks", r -> r.path("/tasks/**")
               .filters(f -> f.rewritePath("/tasks/(?<segment>.*)", "/${segment}"))
               .uri("http://task-service:8080"))
               .build();
    }

    public static void main(String[] args) {
        SpringApplication.run(GatewayApplication.class, args);
    }

}
```