

API Gateways

The good, the bad, and the ugly

James Gough and Matt Auburn

@Jim__Gough and @mattyaubz

O'REILLY®

Software
Architecture

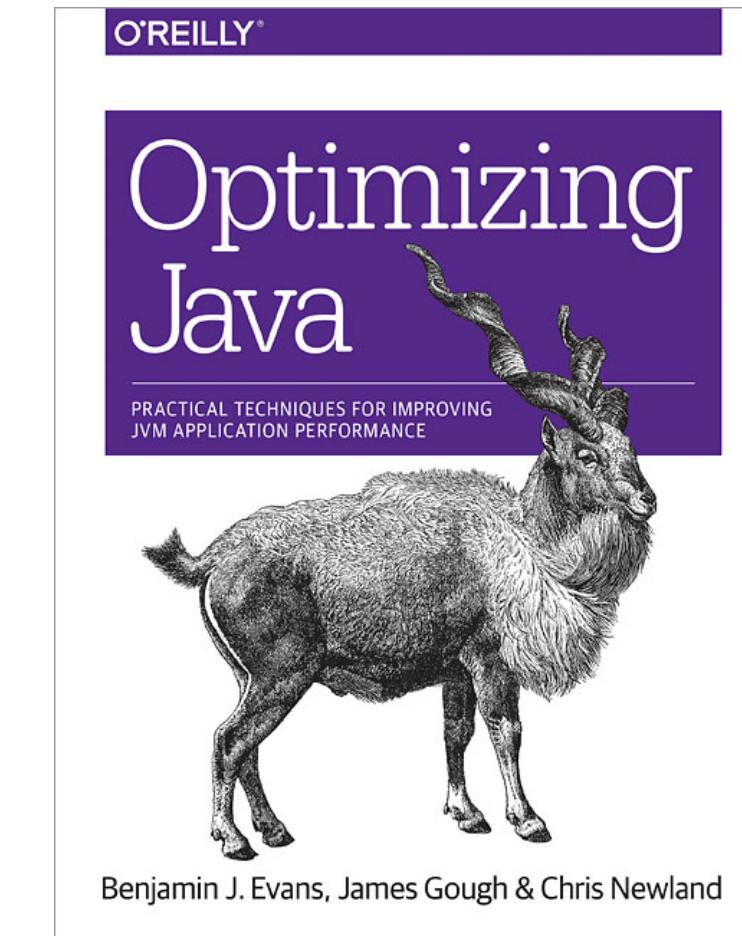
Who are we?

Jim

- London Java Community
- Contributed to Java
- Developer and Trainer
 - Teaching Java and C++ to graduates
- Co-Author of Optimizing Java
- Developer on Java based API Gateways
- Occasional Maven Hacker

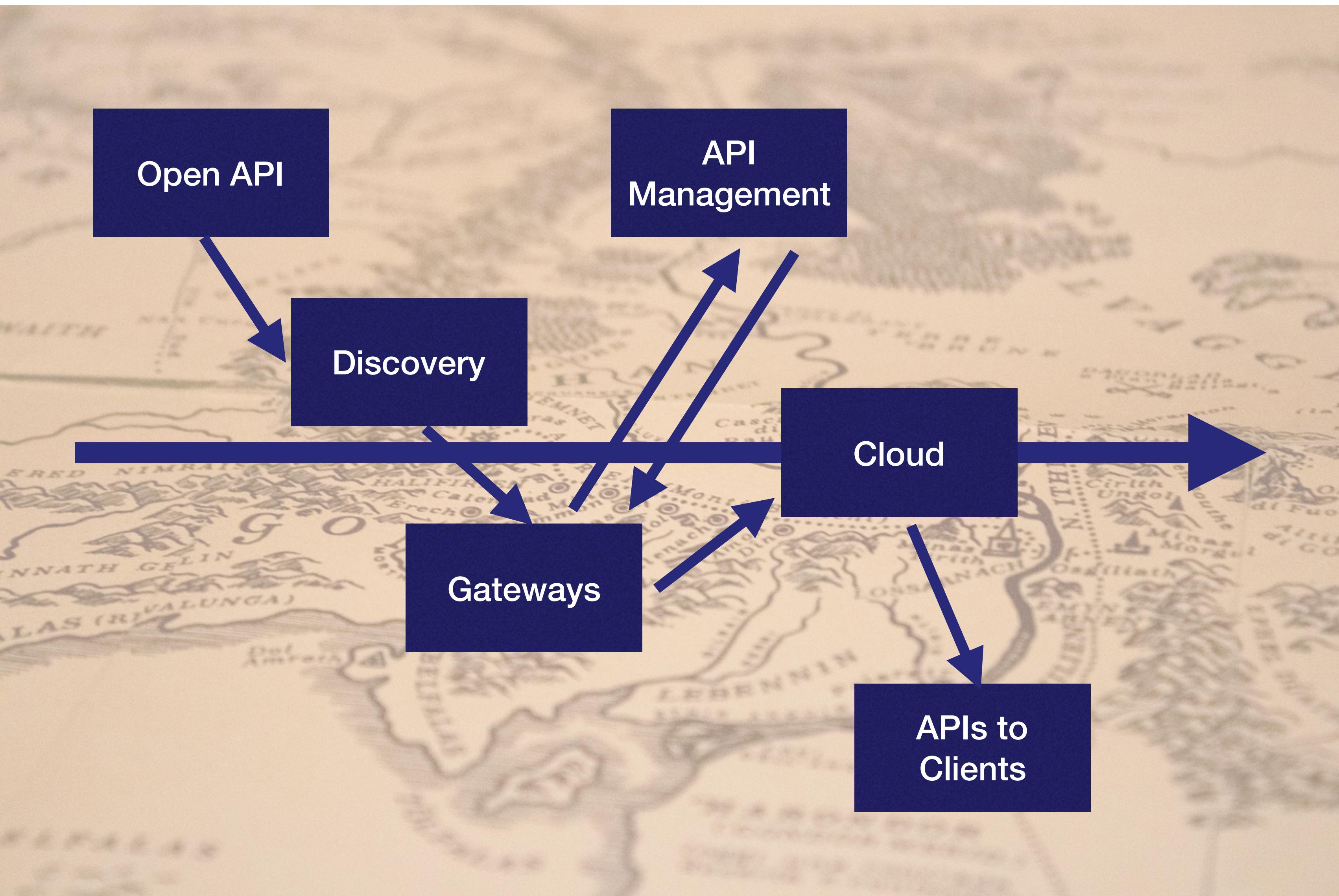
Matt

- Java Developer
- APIs and API Gateways
- Hacker of things

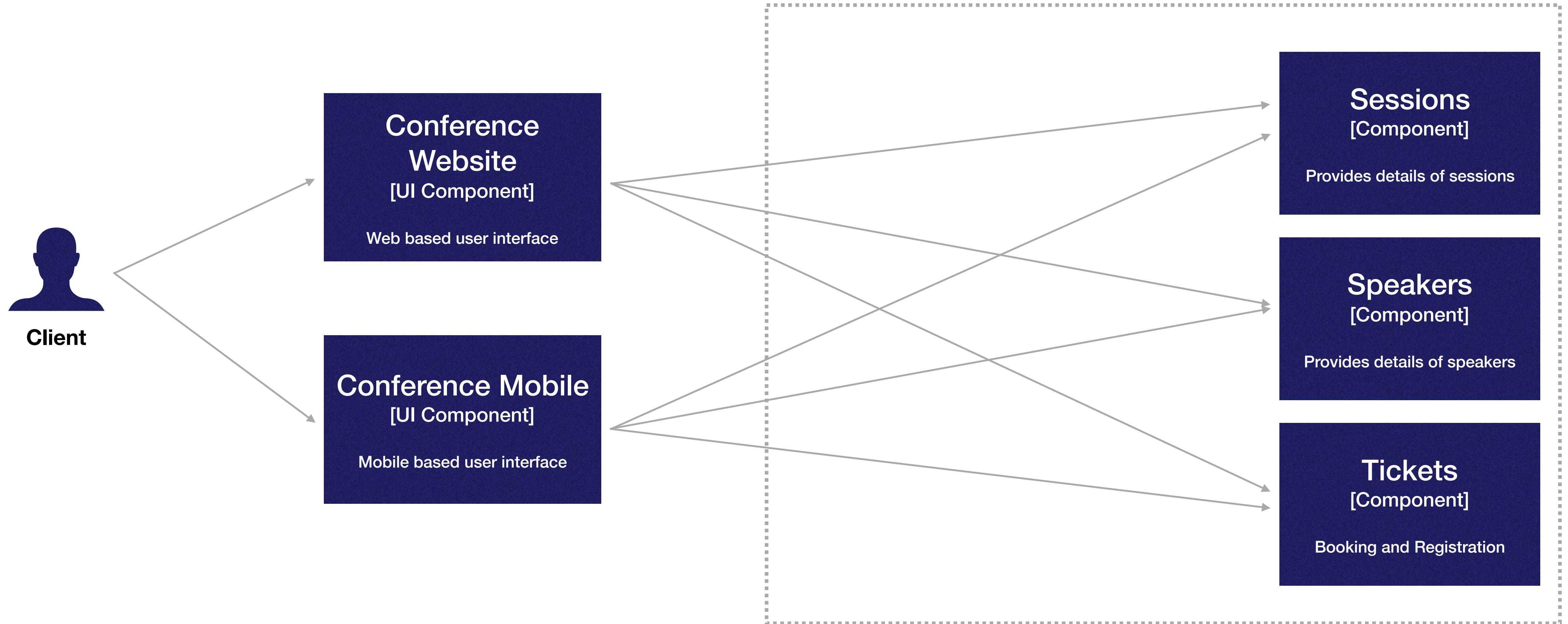


Agenda

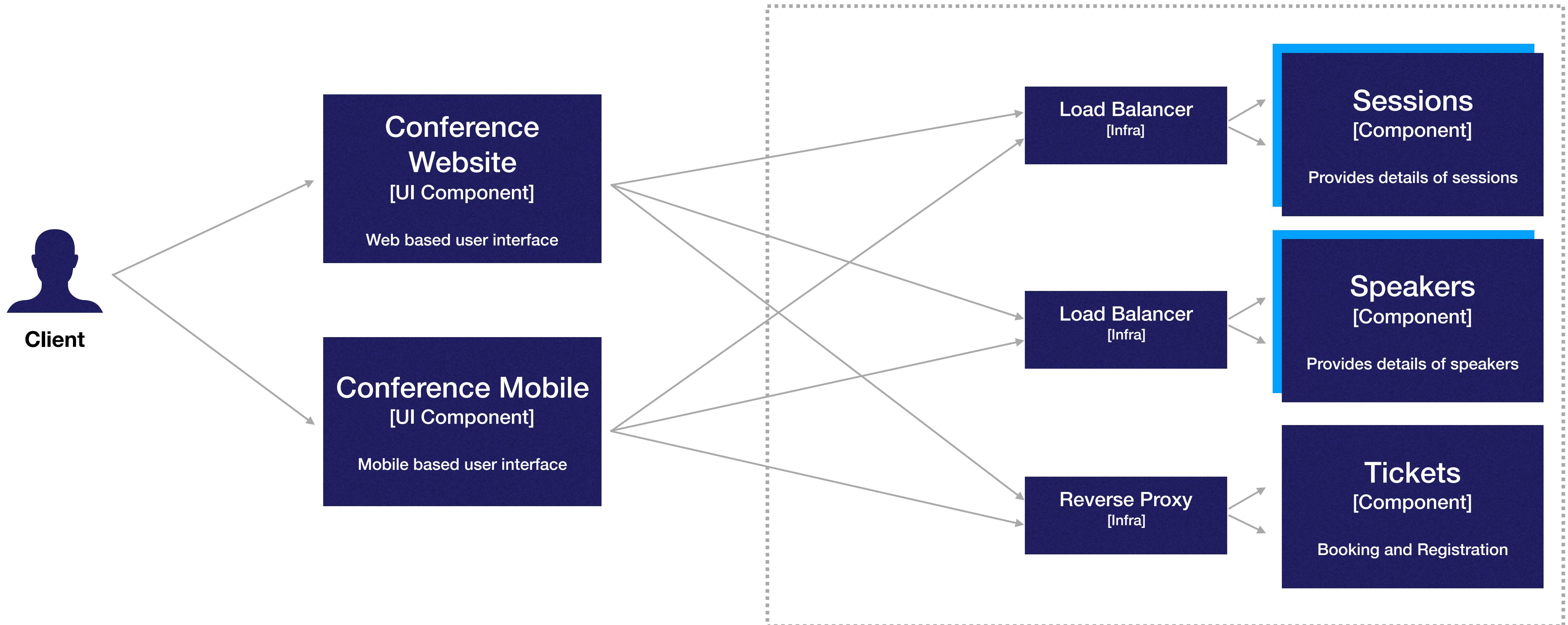
- API Gateways, their usage and our journey
- The bigger picture and architectural migration
- Words of caution
- Picking a Gateway
- Test Containers
- End to End Testing and Insights
- Antipatterns and Conclusions



What is an API Gateway?



What is an API Gateway?



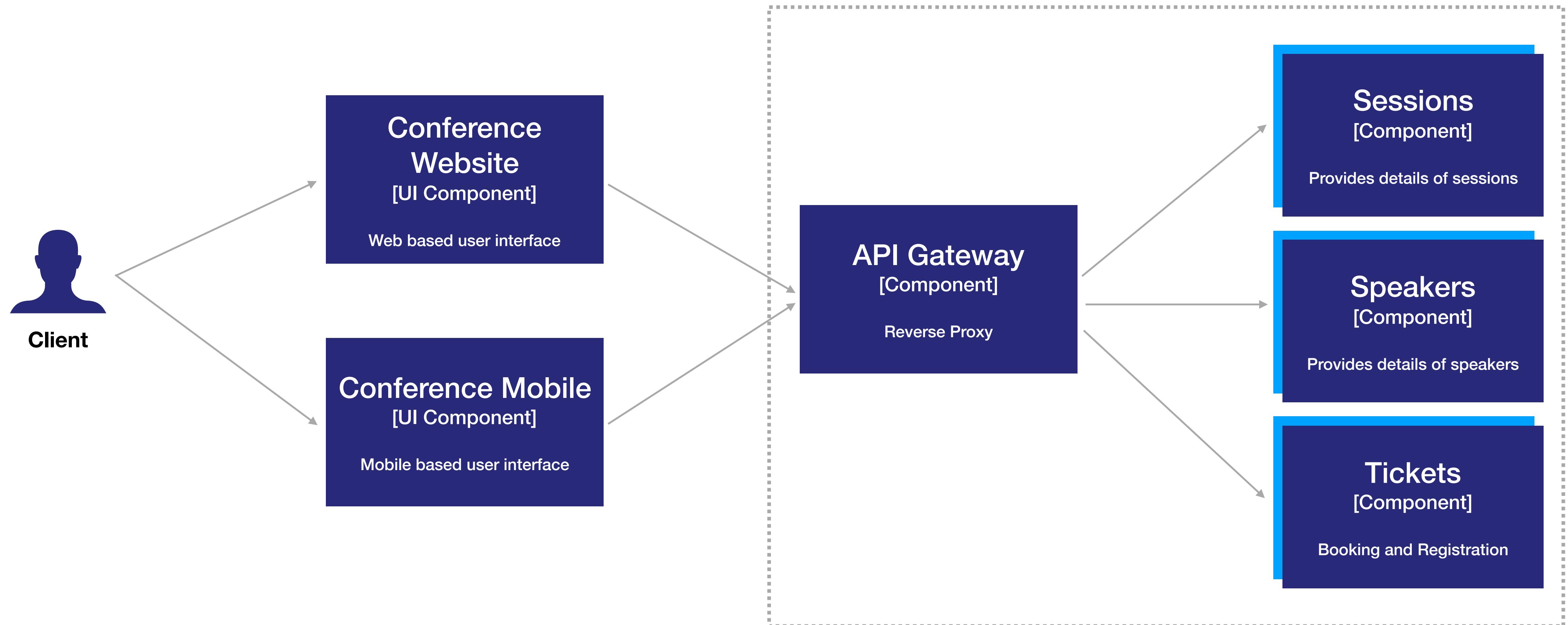
Without an API Gateway

- Client is coupled to the knowledge of the server deployments
- Rate of change of microservices exposes a maintenance issue
- Supporting multiple versions is tricky and may lead to big bang releases
- Increases the number of firewall holes
- Services may have to handle SSL, authorisation, logging

Features of an API Gateway

Features	Reverse Proxy	Load Balancer	API Gateway
Single Backend	*	*	*
TLS/SSL	*	*	*
Multiple Backend		*	*
Service Discovery		*	*
API Composition			*
Auth			*
Retry Logic			*
Rate Limiting			*
Logging and Tracing			*
Circuit Breaking			*

Introducing One Gateway



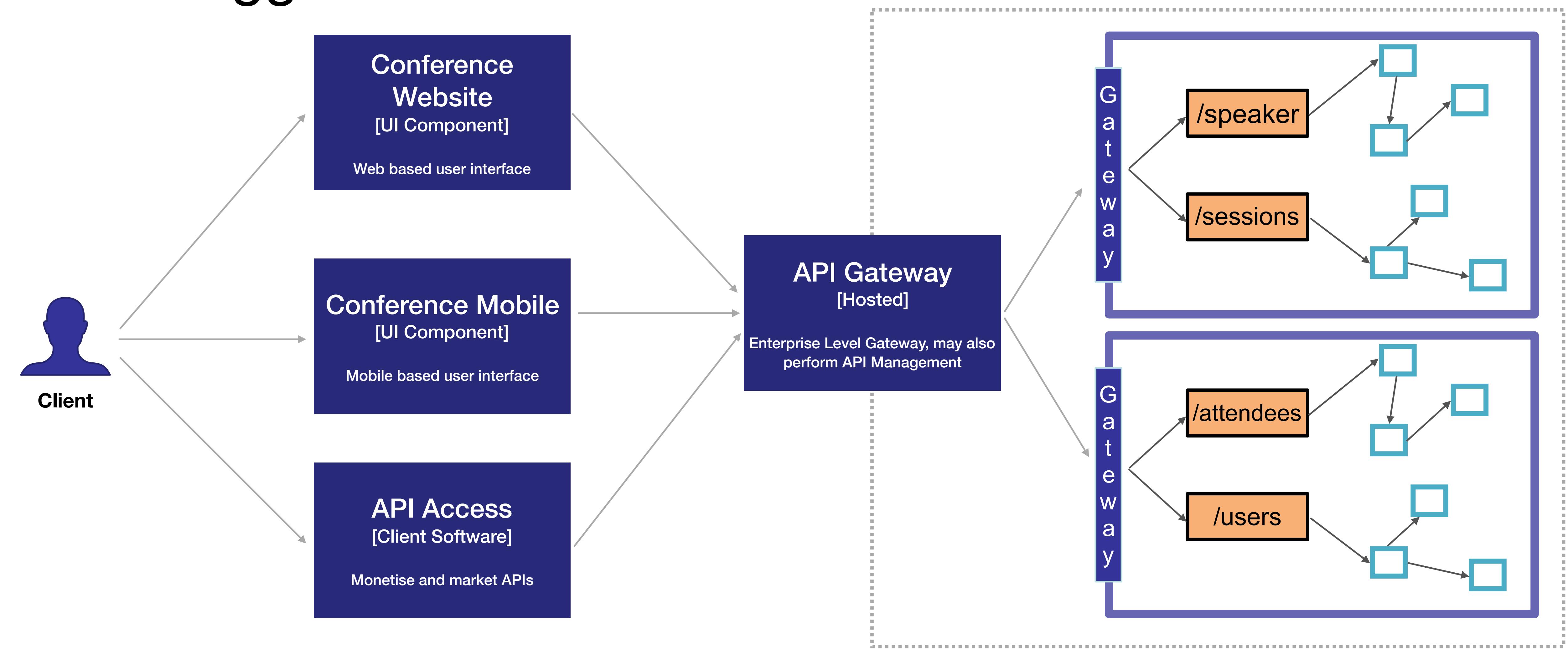
One Gateway Problem Solved?

- A single team owns releases for everyone
 - Slows down deployments
 - Makes DevOps and automated deployments difficult
- Mixture of traffic with varying performance requirements
 - One poor deployment or system takes down entire company
- Versioning and management of APIs is now tangled with lower level concerns
 - e.g. service discovery and circuit breaking

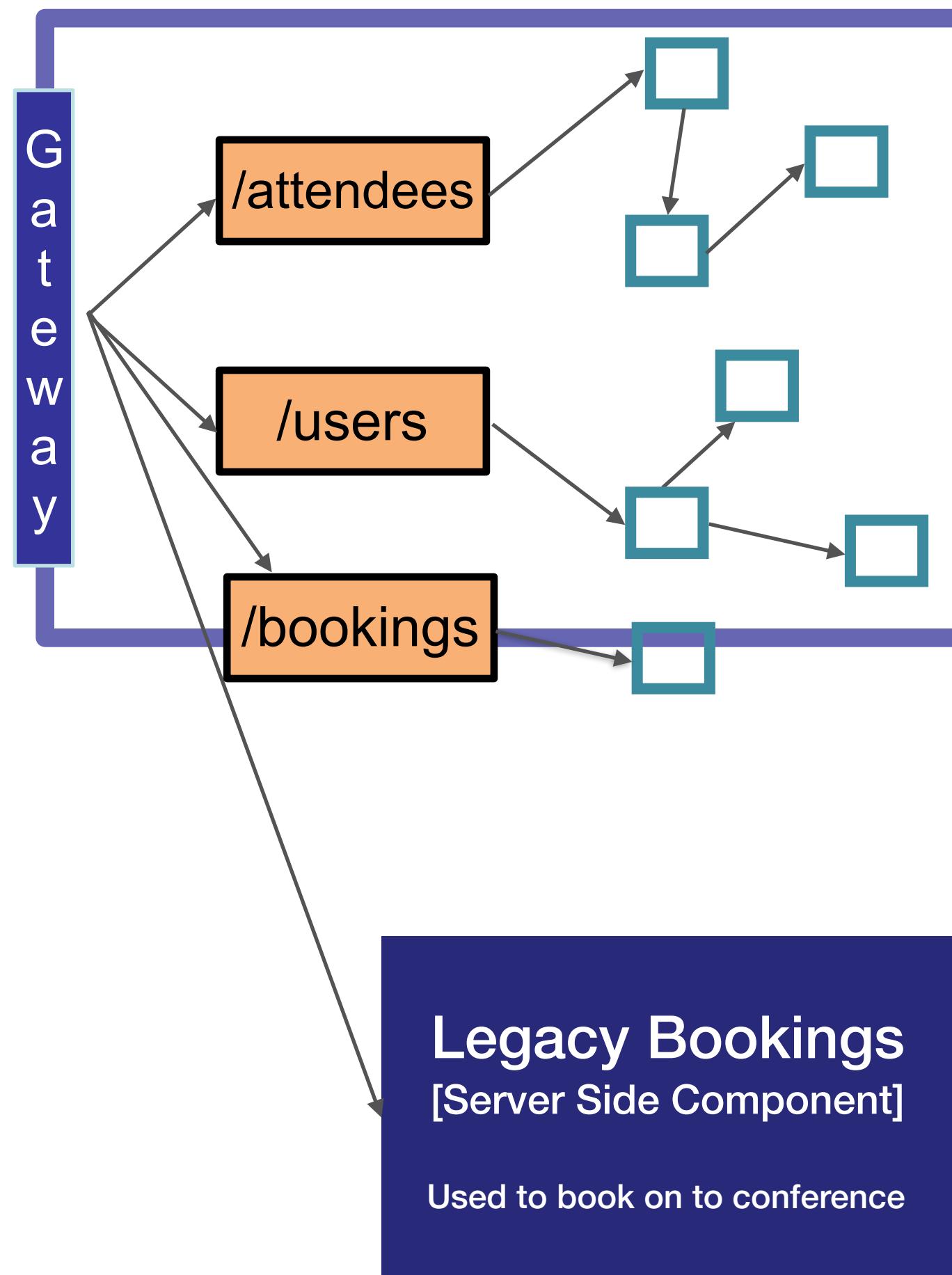
Gateway Levels and Responsibility

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

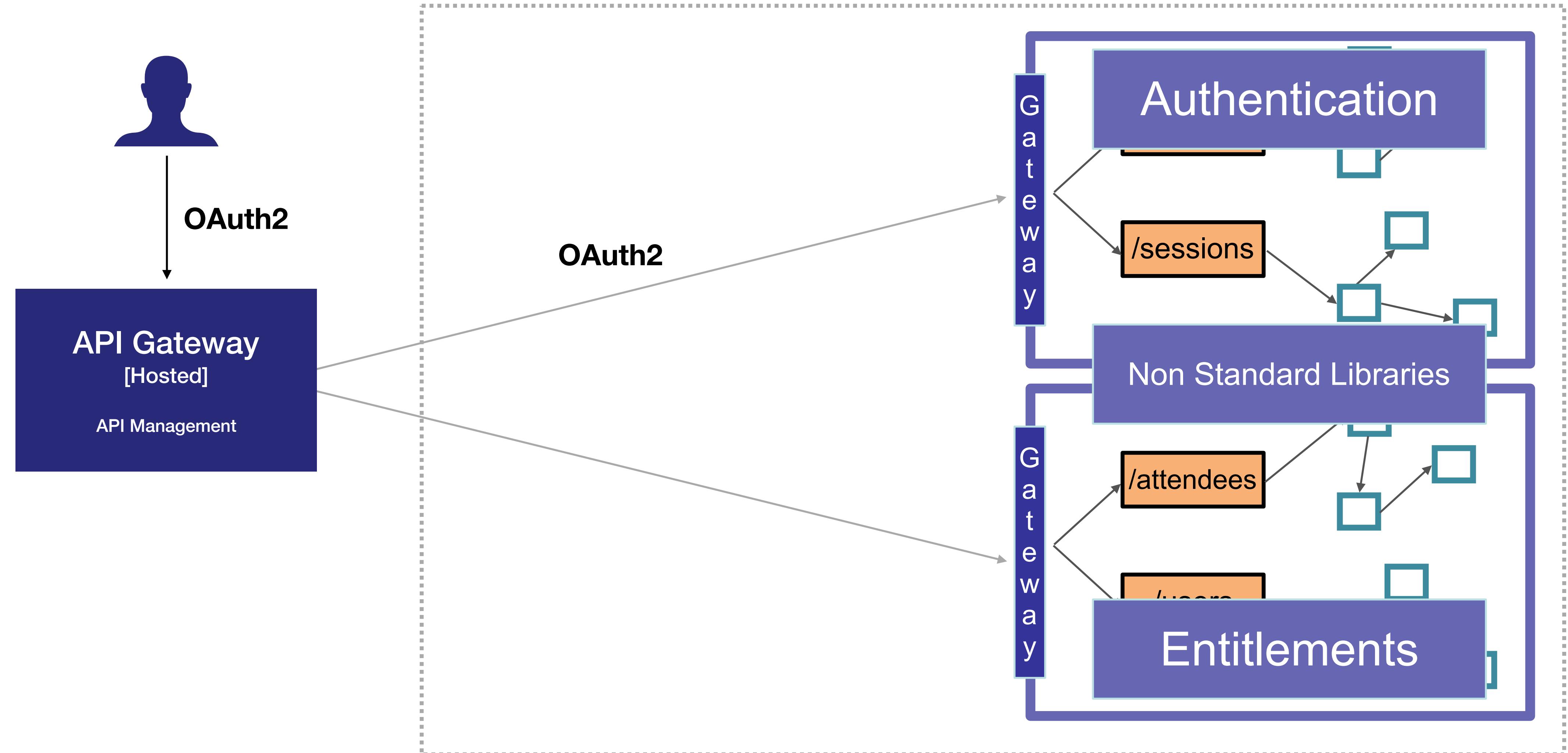
The Bigger Picture



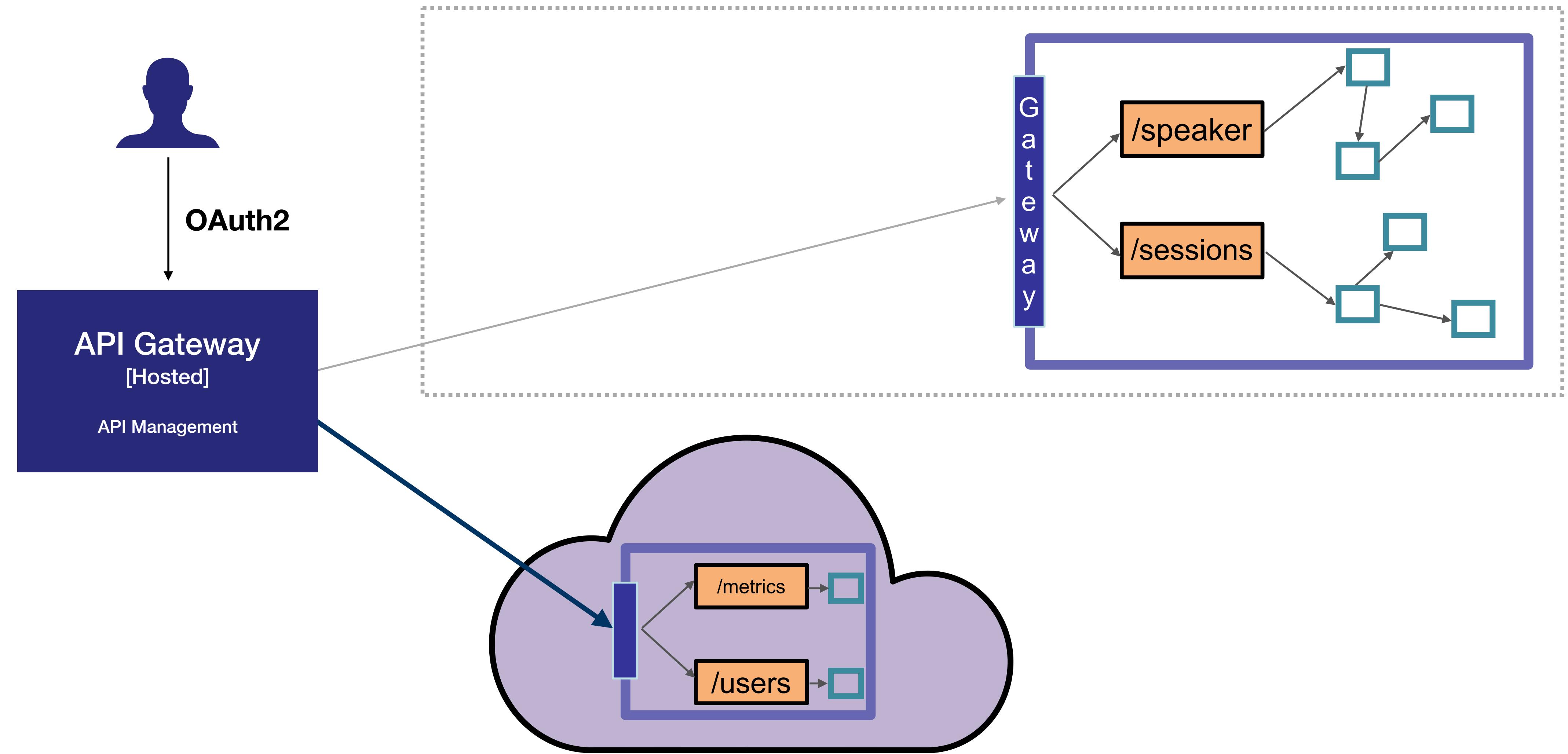
Migrating Legacy



Migrating to Cloud



Migrating to Cloud



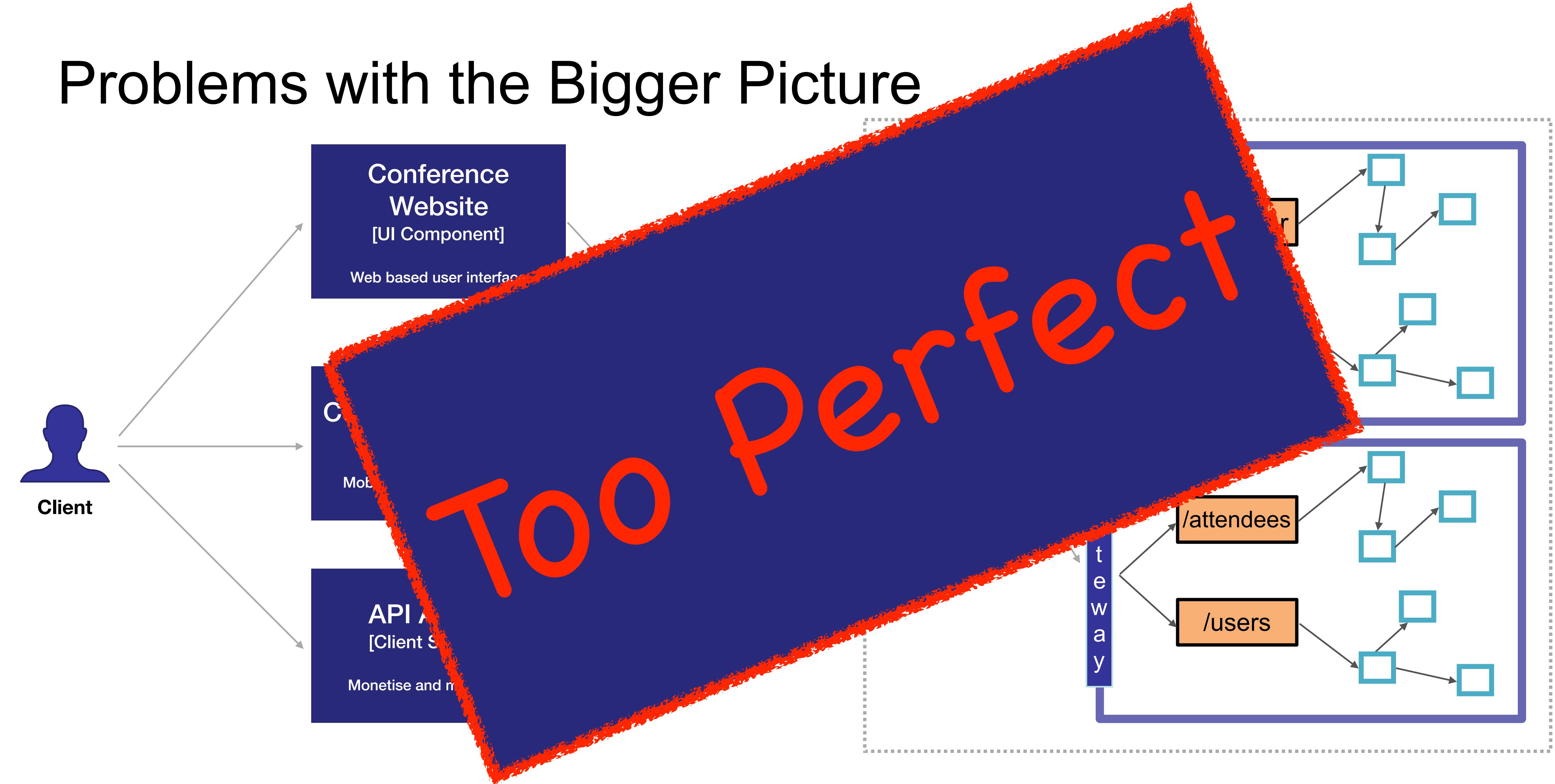
Words of Caution

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.

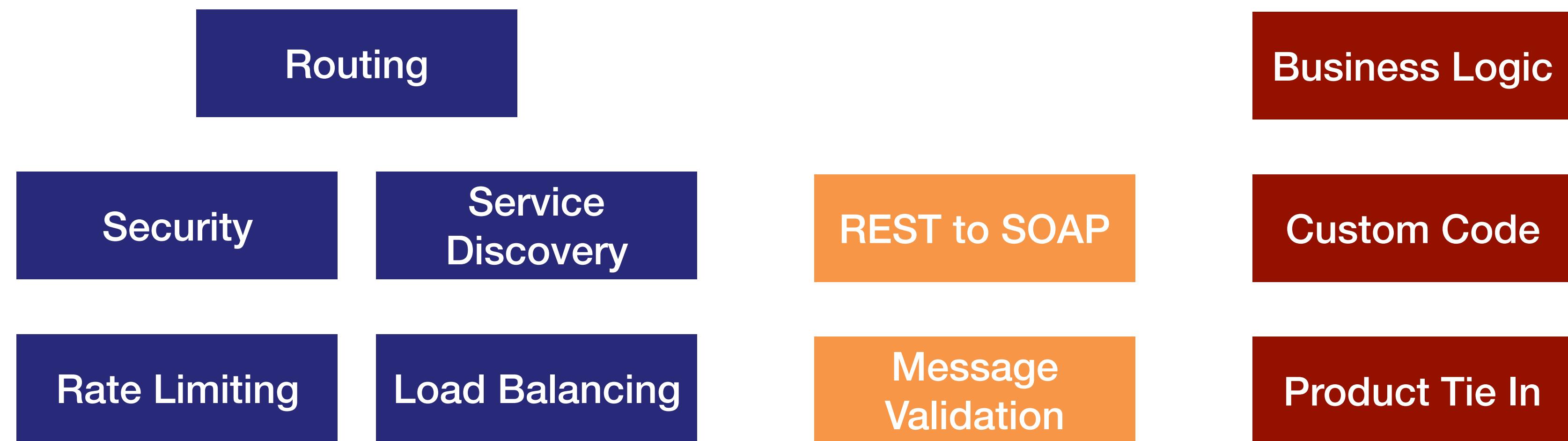
This results in **overambitious API gateway** products whose functionality — on top of what is essentially a reverse proxy — encourages designs that continue to be difficult to test and deploy. API gateways do provide utility in dealing with some specific concerns — such as authentication and rate limiting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

Thoughtworks Technology Radar

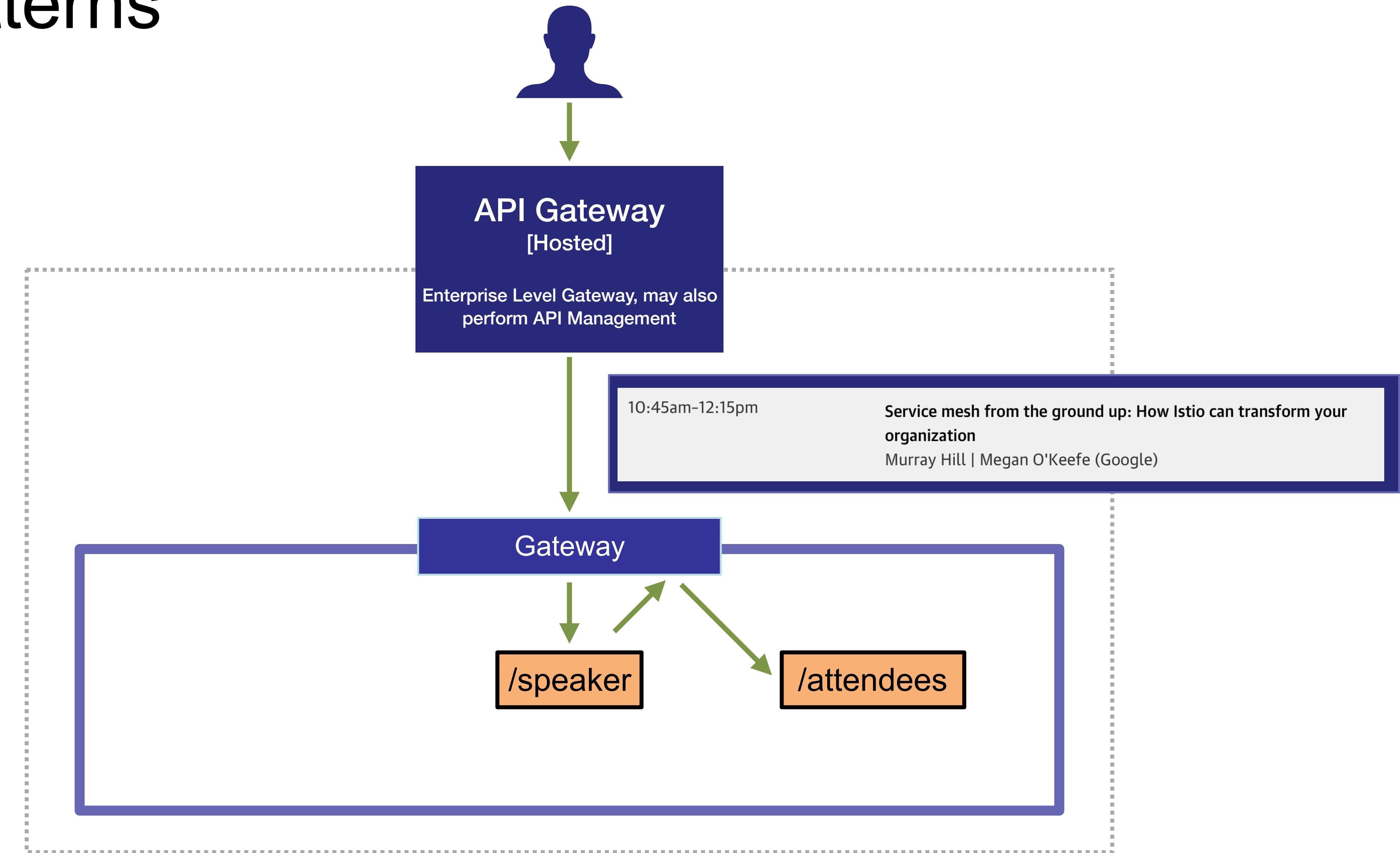
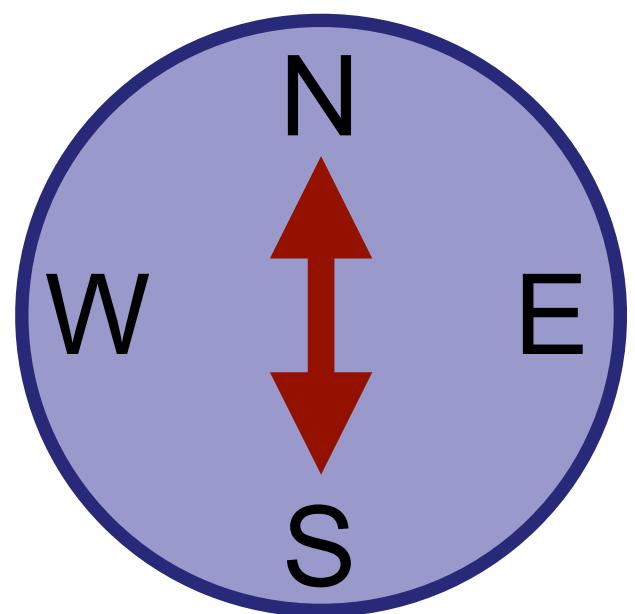
Problems with the Bigger Picture



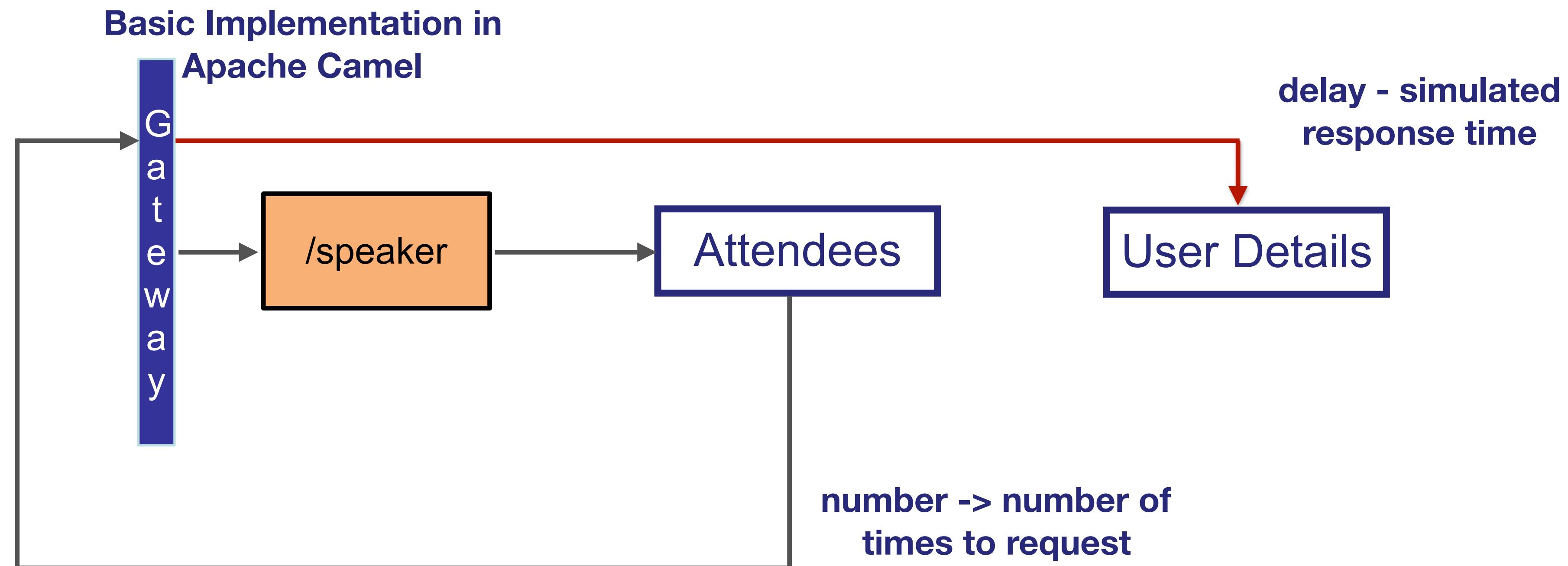
Additional Features of an API Gateway



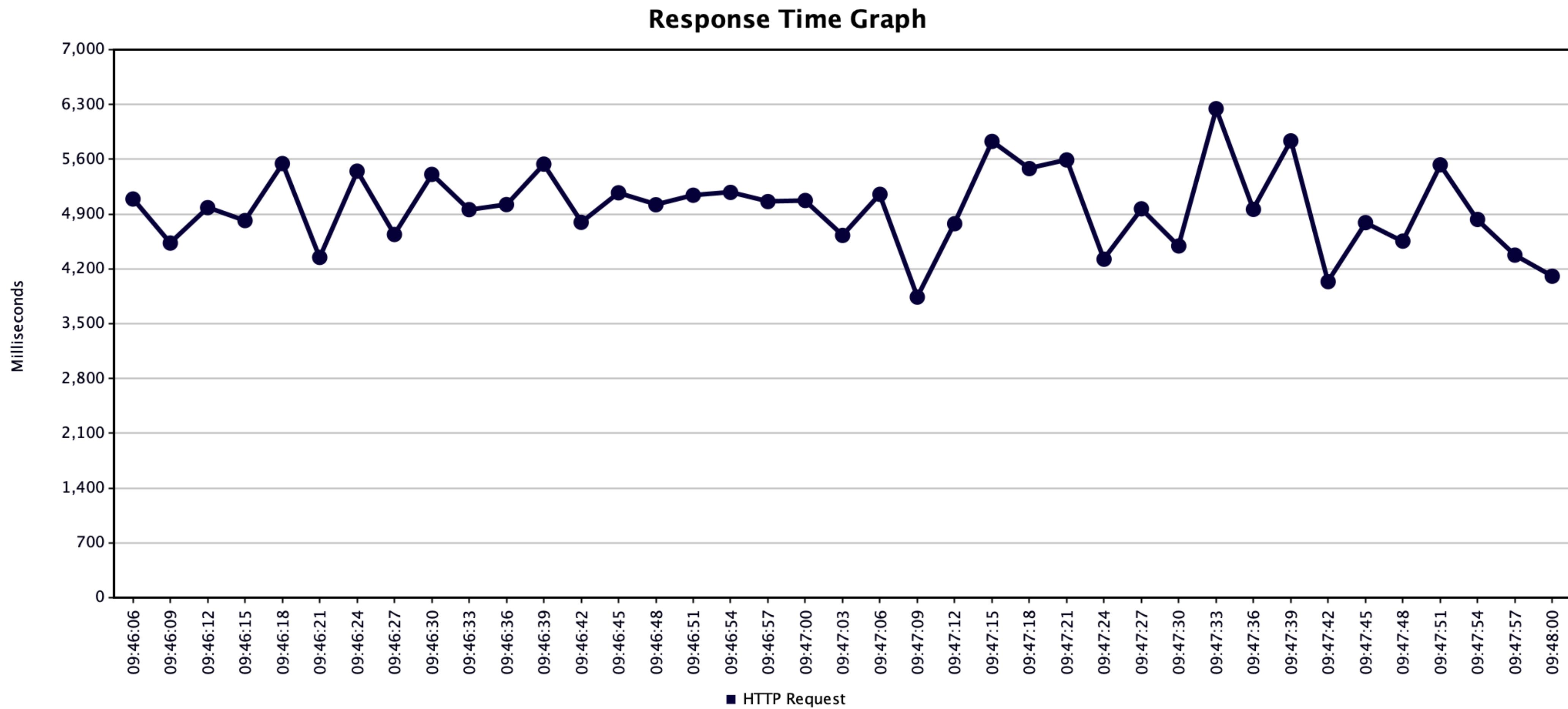
Traffic Patterns



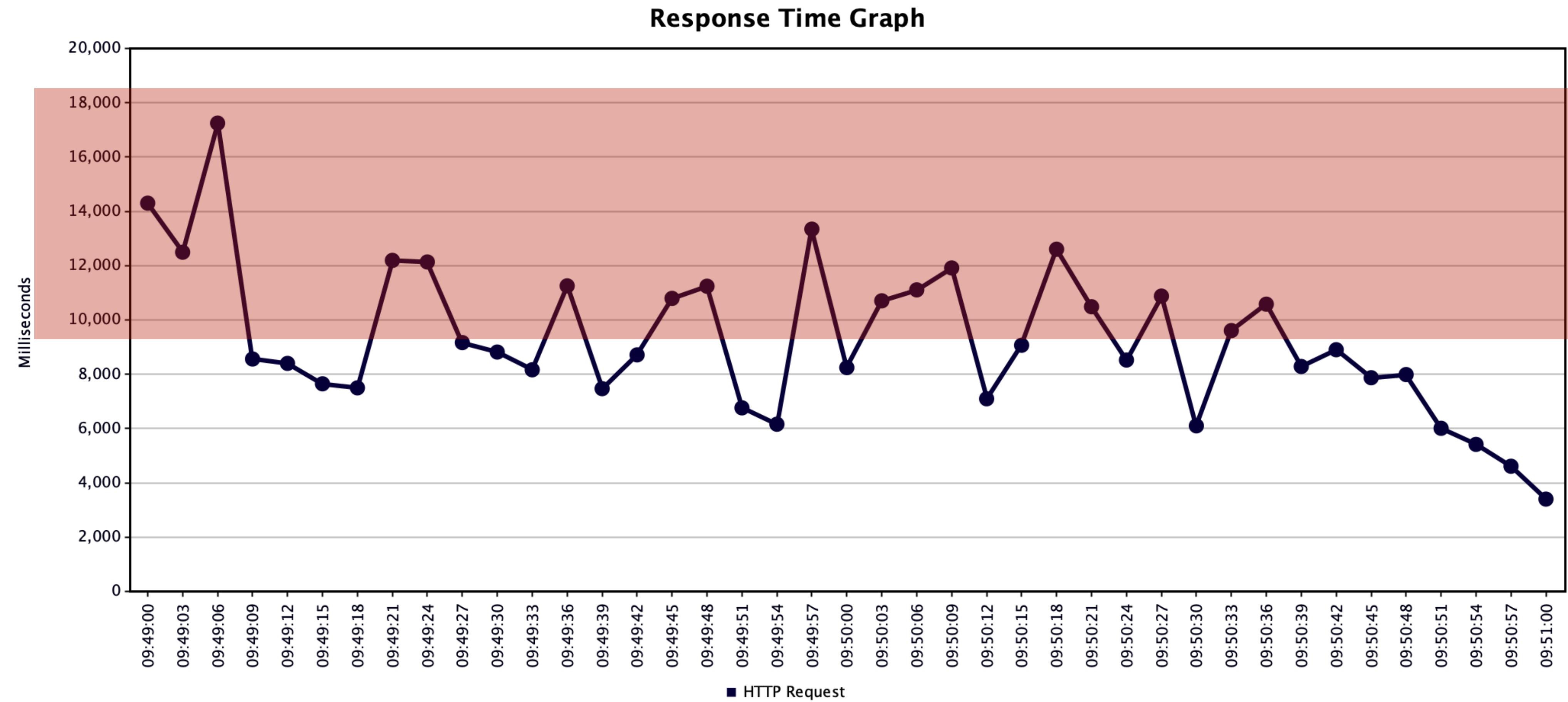
Mixing Traffic Patterns



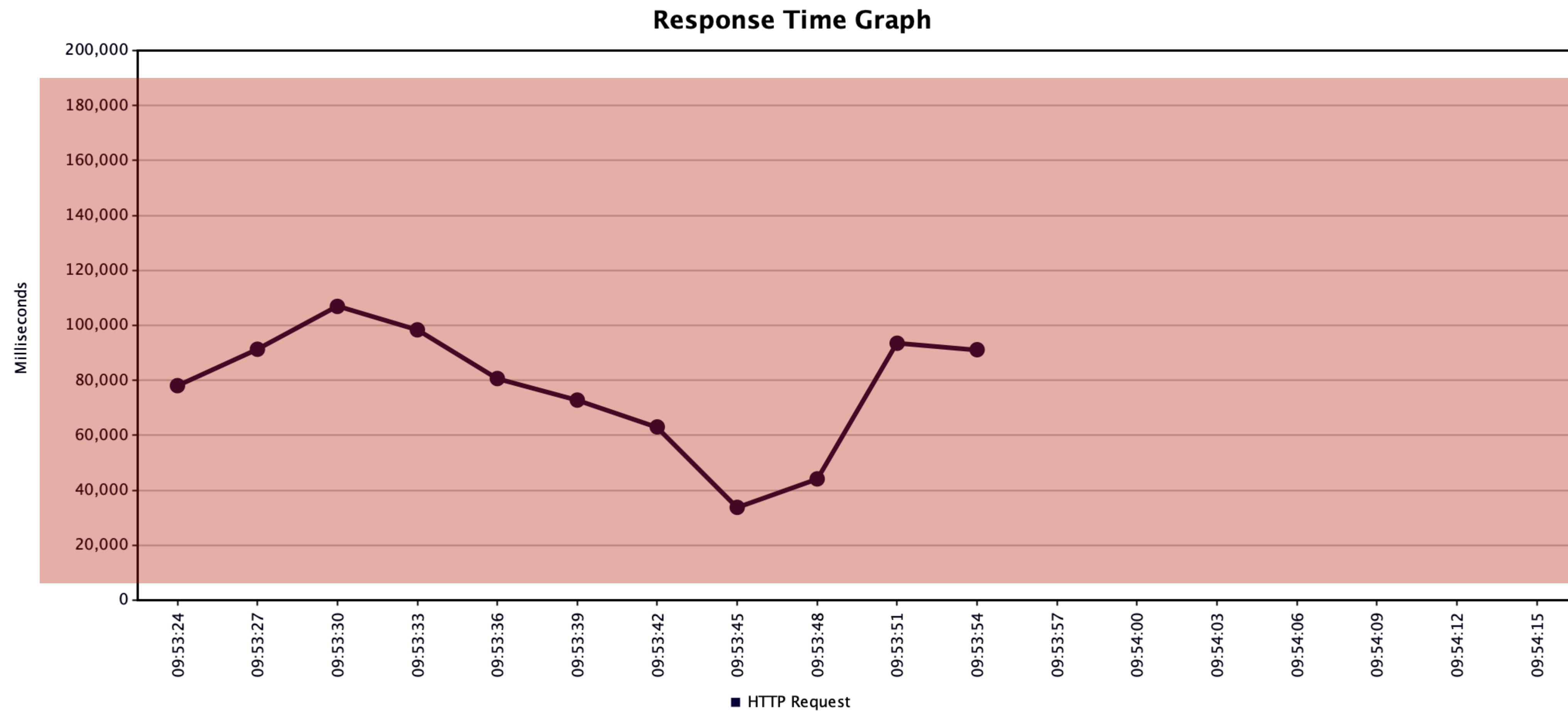
Gateway Loopback - 50 users, 2 Internal Requests, Delay 1s



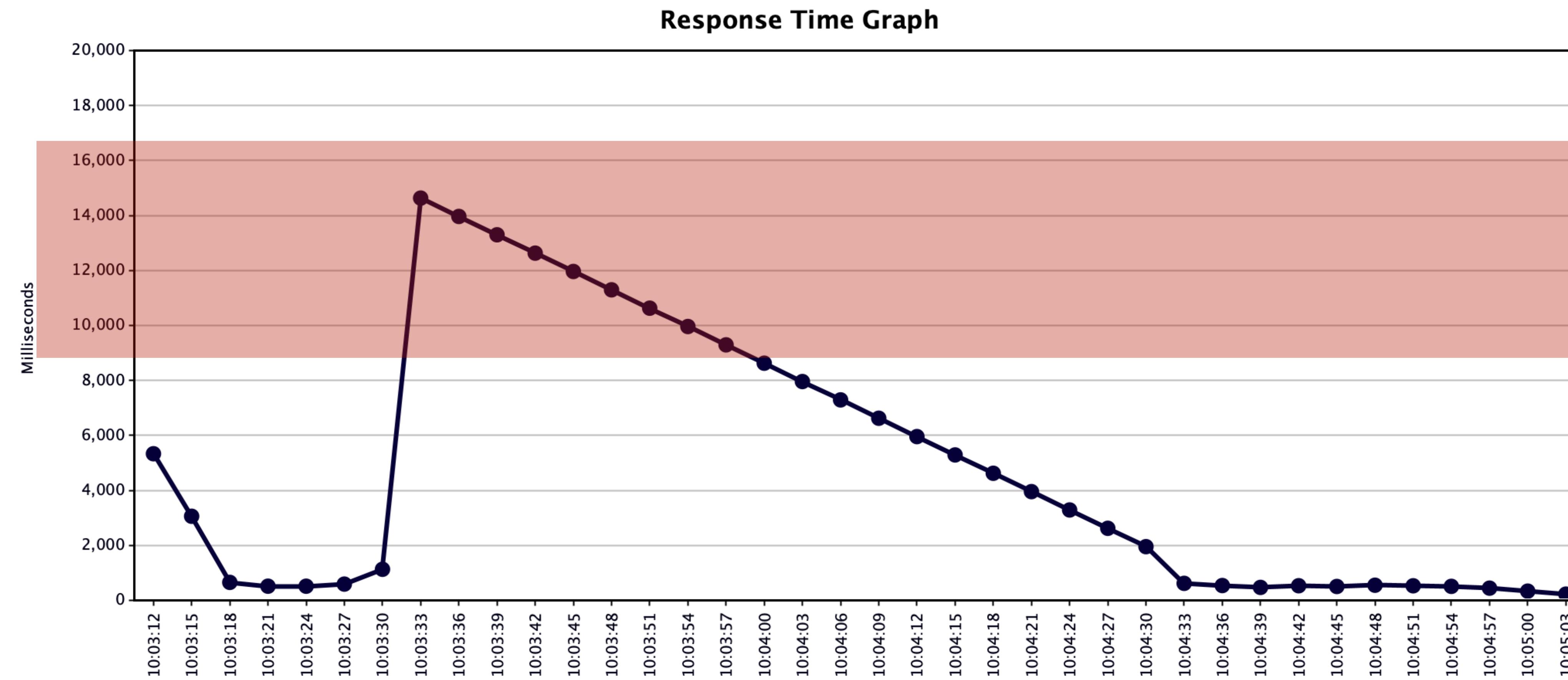
Gateway Loopback - 100 users, 2 Internal Requests, Delay 1s



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



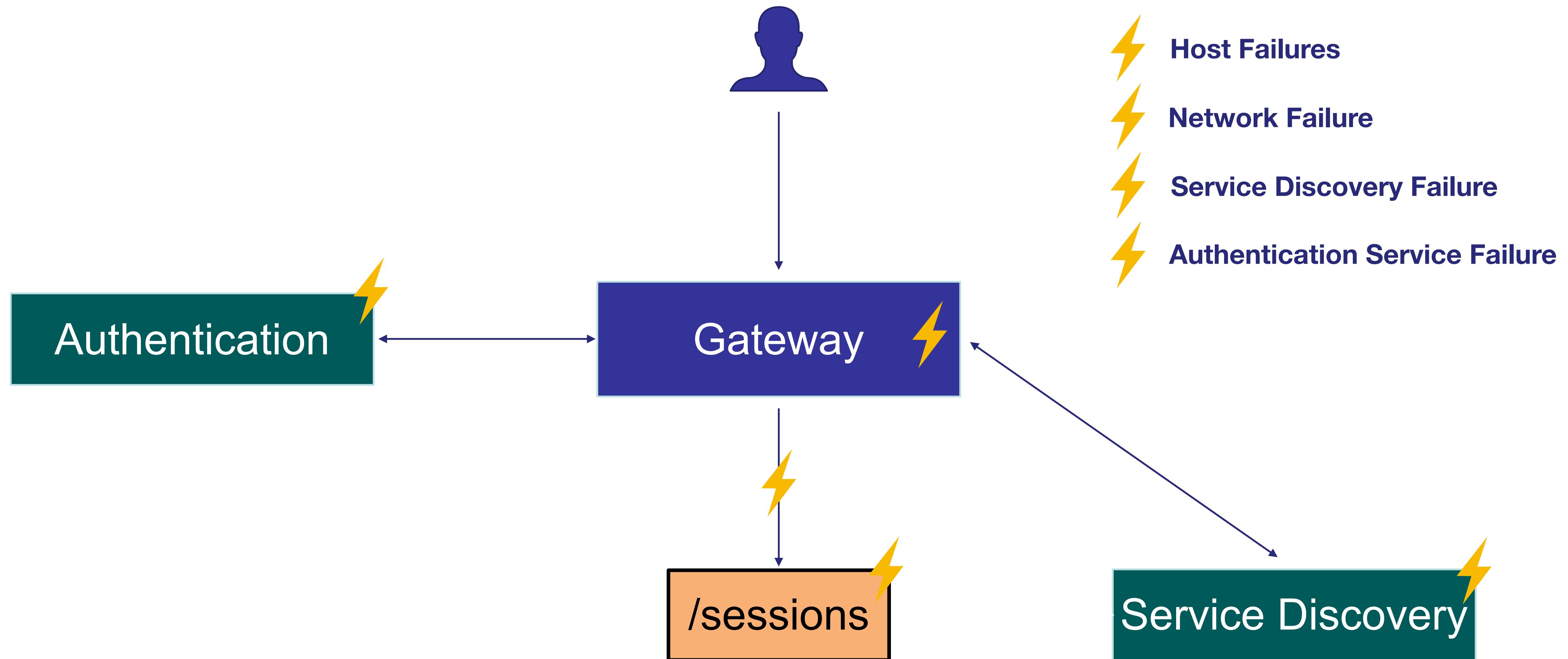
Gateway Loopback - 200 users, 2 Internal Requests, Delay 0s



Build Your Own Microservice Gateway

- Our Demo Gateway is a build your own
- Pick up an Open Source Microservice Gateway and contribute back
- It's important to understand the underlying web server (Jetty/Netty)
- Spend a LOT of time building filters and logic
- Back pressure and buffer capacities relevant as soon as product scales
- Visibility into traffic patterns and flows

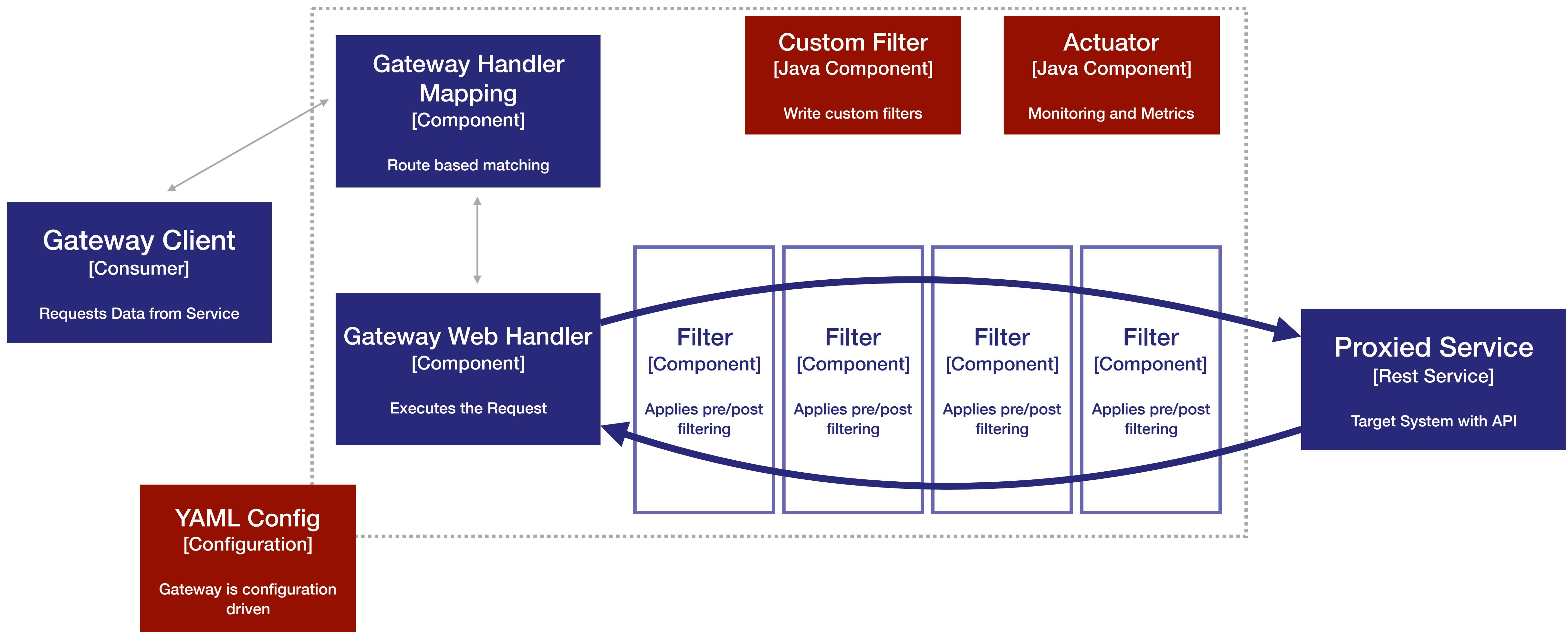
Single Points of Failure



Introduction to Spring Cloud Gateway

- Built using Spring WebFlux and Project Reactor
- Reactive implementation
- Uses Netty which provides non-blocking I/O
- Foundations of Spring Cloud Gateway
 - **Route:** Route ID, destination, predicates and filters
 - **Predicates:** for matching on the HTTP request
 - **Filters:** Used to modify requests and responses

Introduction to Spring Cloud Gateway



Testing the Gateway

O'REILLY®

Software
Architecture

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.

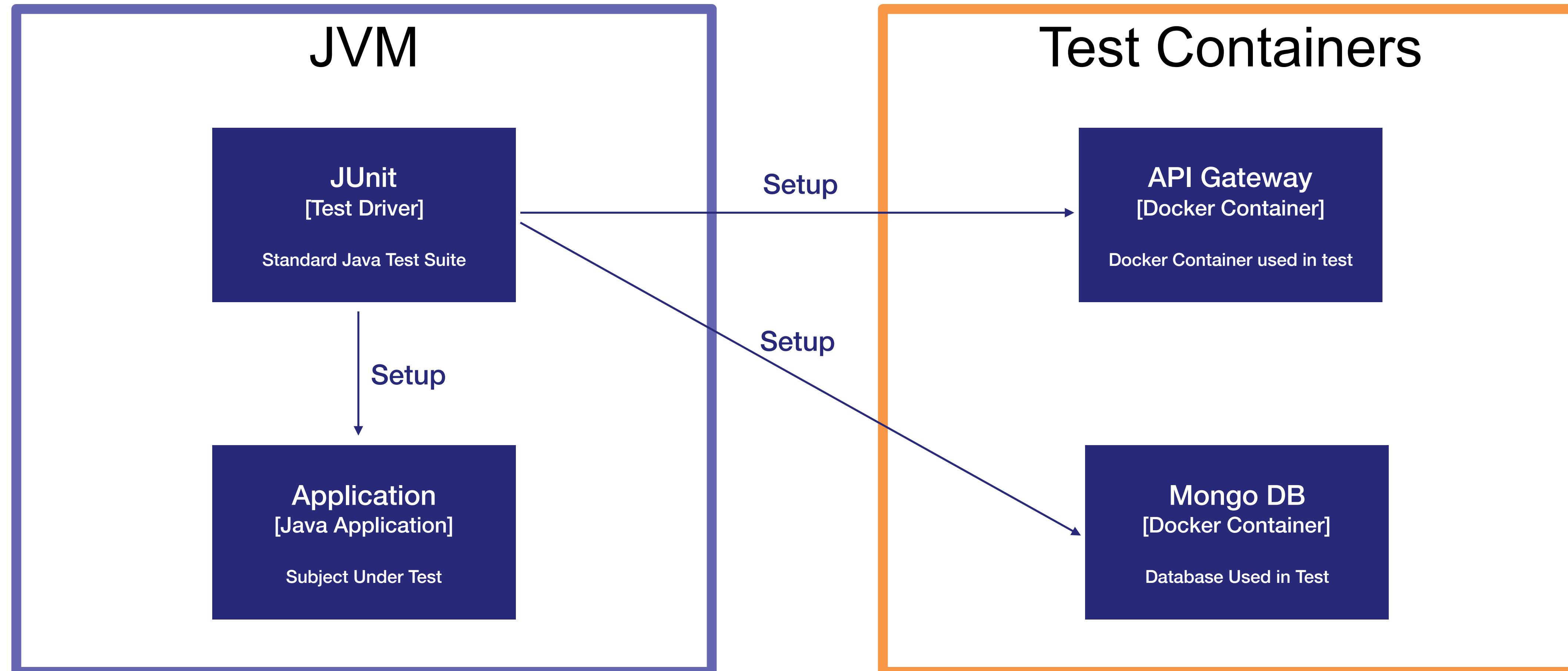
[Sample Discussion from Stackoverflow](#)

Test Containers

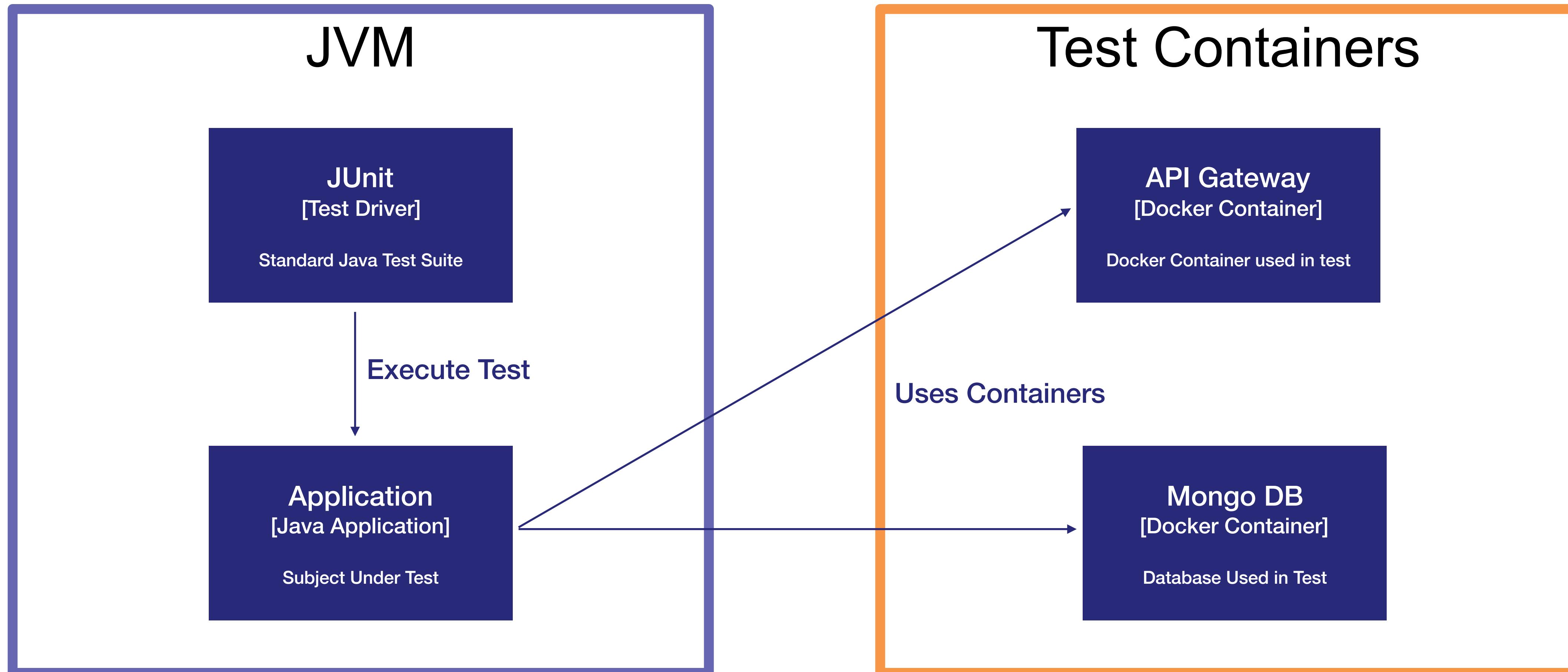
“Test containers is a Java library that supports JUnit tests, providing lightweight, throwaway instances of common databases, Selenium web browsers, or anything else that can run in a Docker container.”

<https://www.testcontainers.org/>

Test Containers



Test Containers



Test Containers

```
@Testcontainers
public class AttendeesThroughGatewayTests {

    private static final int EXPOSED_GATEWAY_PORT = 8080;

    @Container
    public GenericContainer scgVanillaContainer =
        new GenericContainer<>().dockerImageName("jpgough/vanilla-scg")
            .withExposedPorts(EXPOSED_GATEWAY_PORT)
            .withEnv("HOST_MACHINE_IP_ADDRESS", IpDiscovery.myIPAddress())
            .withClasspathResourceMapping(resourcePath: "gateway/application.yaml",
                containerPath: "/etc/gateway/config/application.yaml",
                BindMode.READ_ONLY);
```

Test Containers

```
@Test
public void delete_call_with_body_supported_by_spring_cloud_gateway() {
    String address = scgVanillaContainer.getContainerIpAddress();
    Integer port = scgVanillaContainer.getFirstMappedPort();

    String endpoint = "http://" + address + ":" + port + "/attendees/user/5";
    RestTemplate restTemplate = new RestTemplate();
    UserDetail user = new UserDetail();
    user.setId(5);
    user.setName("Test User");
    user.setEmail("test@email.com");

    Map<String, String> uriVariables = new HashMap<>();
    HttpEntity<UserDetail> requestEntity = new HttpEntity<>(user);
    ResponseEntity<UserDetail> responseEntity = restTemplate
        .exchange(endpoint, HttpMethod.DELETE,
        requestEntity, UserDetail.class, uriVariables);

    assertEquals(HttpStatus.OK, responseEntity.getStatusCode());
}
```

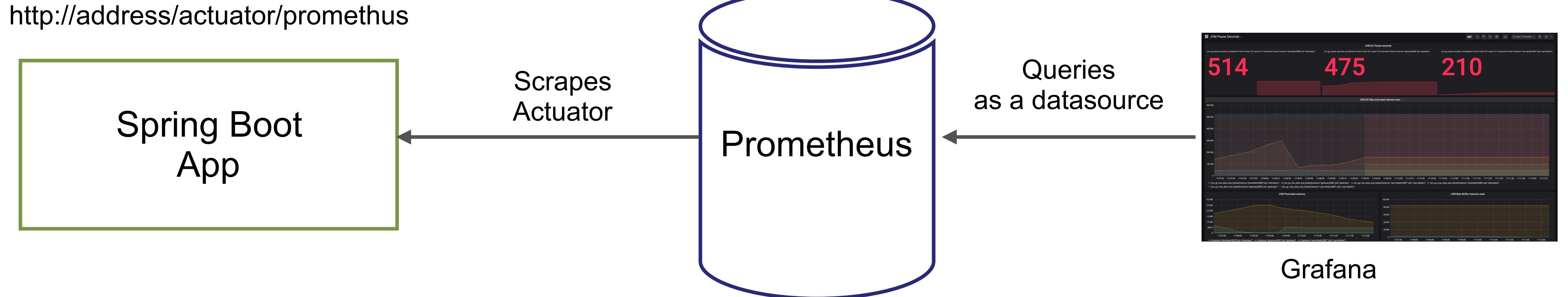
End to End Testing

- It is hard to test scenarios locally that involve infrastructure behaviour
 - Performance
 - Security
- Ephemeral testing helps to mitigate issues before production
- Ephemeral testing helps with ability to get started with chaos testing
- As part of the Test Pyramid a mixture of tests is essential

Insights into Gateway

- Gain an understanding into the operational performance
 - Prometheus
 - Grafana
- Create dashboards to see views into the application
- Setup alerts
- Invaluable when performance testing

Insights into Gateway



Insights into Gateway

Prometheus Alerts Graph Status ▾ Help

Targets

All Unhealthy

attendees (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://attendees:8080/actuator/prometheus	UP	instance="attendees:8080" job="attendees"	1.137s ago	319.8ms	

gateway (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://gateway:8082/actuator/prometheus	UP	instance="gateway:8082" job="gateway"	1.453s ago	434.9ms	

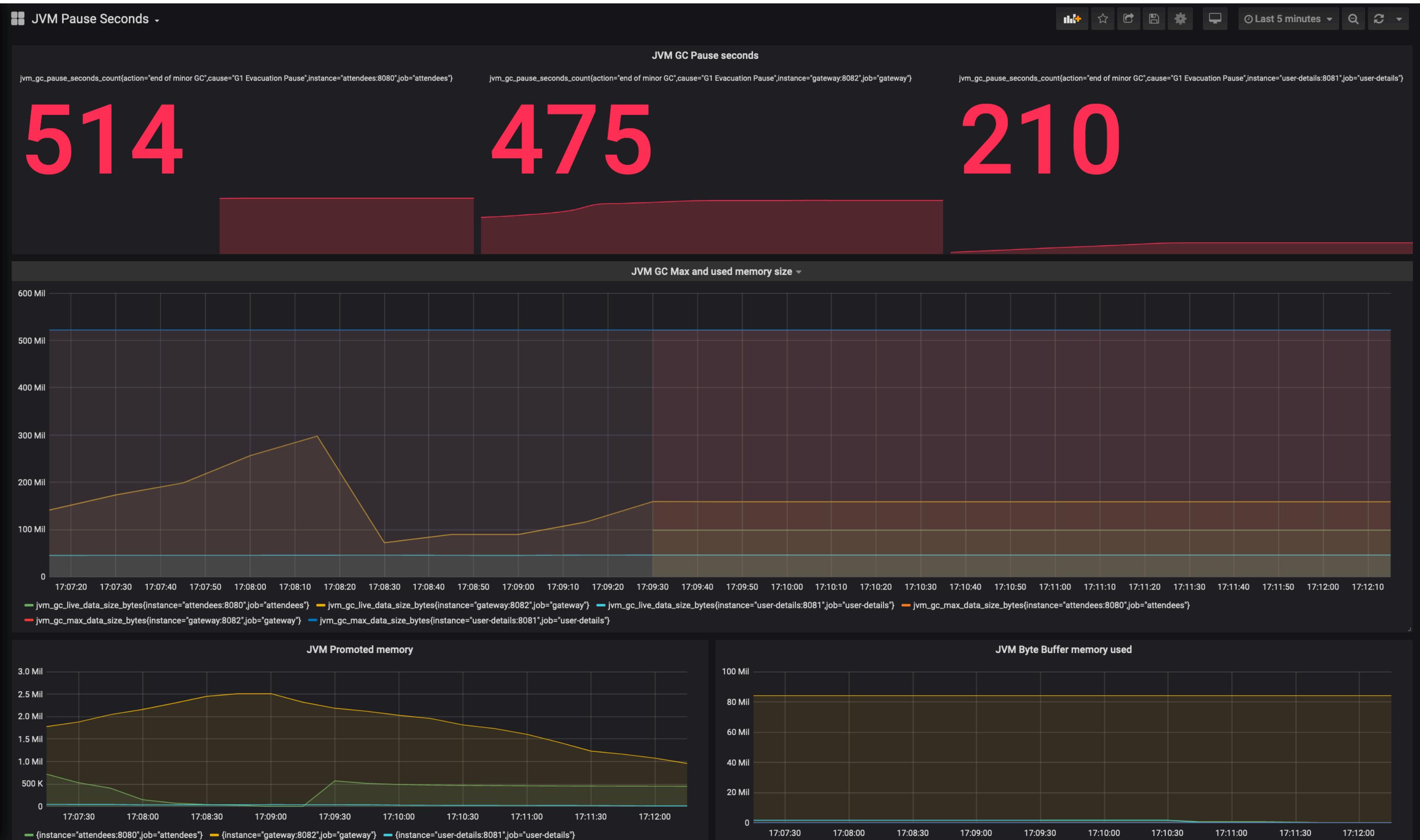
prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	105ms ago	4.478ms	

user-details (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://user-details:8081/actuator/prometheus	UP	instance="user-details:8081" job="user-details"	638ms ago	301.5ms	

Insights into Gateway



Antipatterns

- Like any technology API Gateways are subject to misuse
- Avoid overburdening a single Microservices Gateway too many services
- It should be easy to provision and deploy your own
- Avoid a gateway becoming an Enterprise Service Bus
- Have to be careful to avoid a high degree of gateway loopback

Conclusions

- **Do use** Microservice Gateways to coordinate APIs
- **Do use** Enterprise Gateways to deliver external value in client APIs
- **Do use** gateways as an architectural benefit
- **Do NOT** use gateways as an Enterprise Service bus or mix traffic directions
- **Ensure** single points of failure are mitigated and managed
- **Use testing** and best practices to ensure minimal production impact
- **Use monitoring** and **visibility** to maintain a stable plant

Questions

<https://github.com/jpgough/api-gateways-the-good-bad-ugly>

James Gough and Matt Auburn
@Jim__Gough and @mattyaubz

O'REILLY®

Software
Architecture

Rate today's session

Cyberconflict: A new era of war, sabotage, and fear

[See passes & pricing](#)

David Sanger (The New York Times)
9:55am-10:10am Wednesday, March 27, 2019
Location: Ballroom
Secondary topics: [Security and Privacy](#)

Rate This Session

We're living in a new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you're often the collateral damage in a growing conflict among states. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the dens of Chinese, Russian, North Korean, and Iranian hackers to the boardrooms of Silicon Valley, David reveals a world coming face-to-face with the perils of technological revolution—a conflict that the United States helped start when it began using cyberweapons against Iranian nuclear plants and North Korean missile launches. But now we find ourselves in a conflict we're uncertain how to control, as our adversaries exploit vulnerabilities in our hyperconnected nation and we struggle to figure out how to deter these complex, short-of-war attacks.

David Sanger
The New York Times

David E. Sanger is the national security correspondent for the *New York Times* as well as a national security and political contributor for CNN and a frequent guest on *CBS This Morning*, *Face the Nation*, and many PBS shows.



Session page on conference website

✓ Attending [Notes](#) [Remove](#)

Cyberconflict: A new era of war, sabotage, and fear

9:55 AM - 10:10 AM, Wed, Mar 27, 2019

Speakers

 **David Sanger**
National Security Correspondent
The New York Times

Ballroom

Keynotes

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

SESSION EVALUATION

O'Reilly Events App