API Workshop

API Recommendations and API Gateways

O'REILLY®
Software Architecture

Agenda

- Recommendations for Building APIs
- Handling Multiple APIs
- The role of gateways
- API Centric Architecture
- Role of API Management
- Demo Spring Cloud Gateway

Recommendations for Building APIs

- Use API best practices guidelines
  - e.g. PayPal or Microsoft
- Use HTTP Verbs correctly
- Follow idempotency guidelines
- Version you API's from the outset
- Standardise Error structure

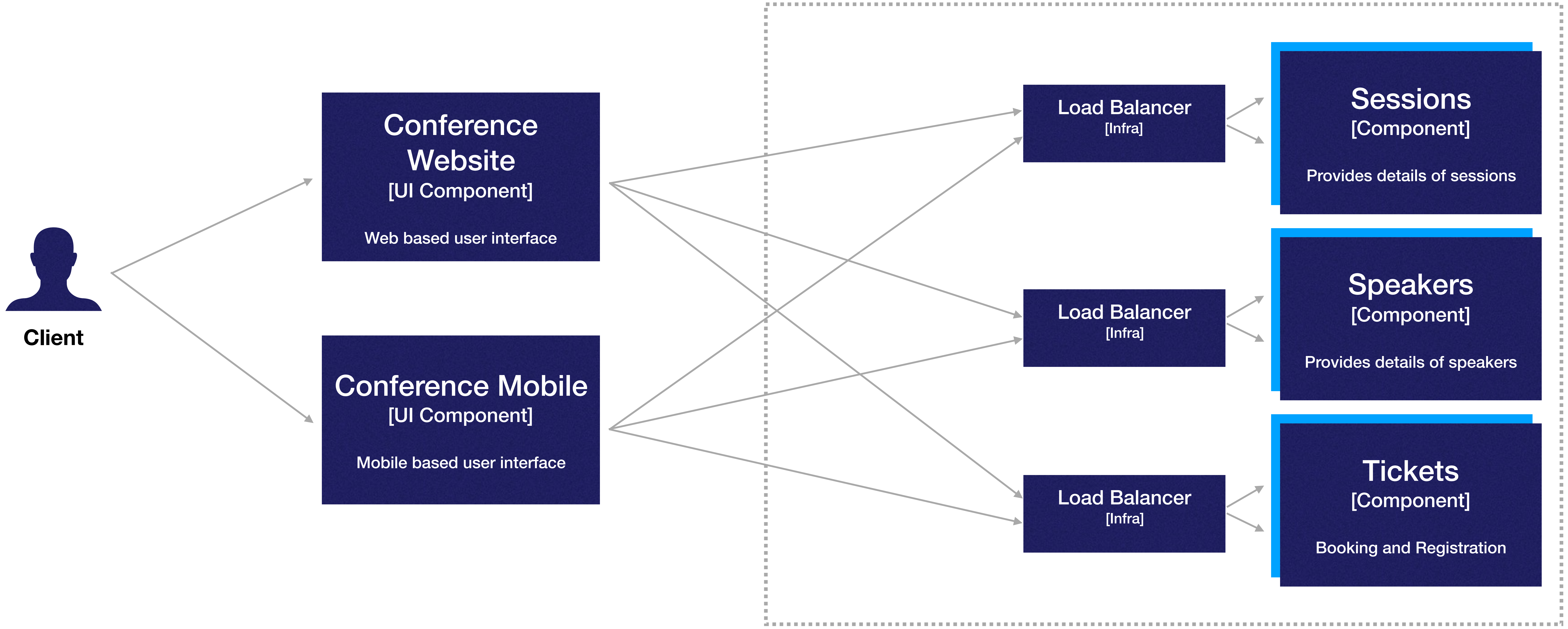
Recommendations for Building APIs

- Collections
  - Pagination
  - Filtering
- CORS (Cross Origin Resource Sharing)
- Consider security from the outset
  - e.g. OAuth2

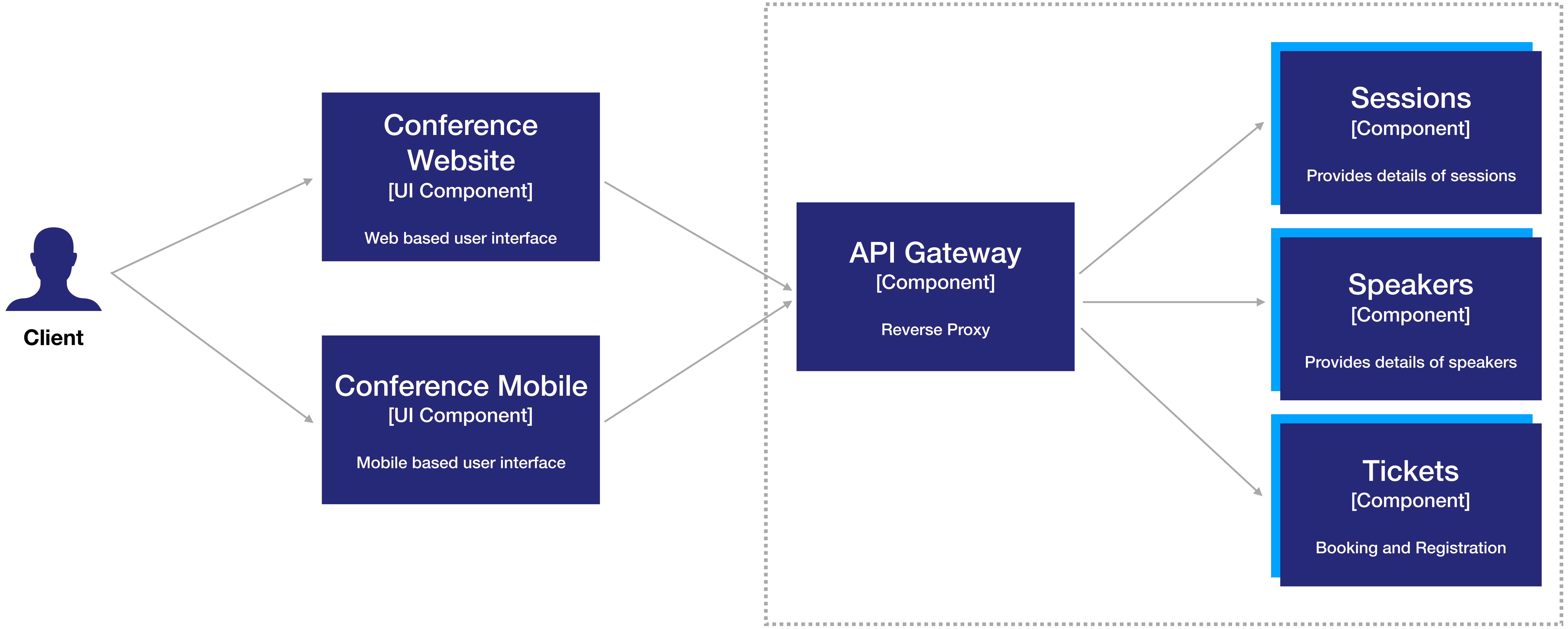
Handling Multiple APIs

- It is unlikely you will build a monolithic API system
- Routing traffic to services becomes an architectural point for consideration
- May wish to avoid each service implementing
  - Uniform Request Logging
  - Security Considerations (Entitlements/SSL Termination)
  - Circuit breaking/Load balancing

What is a Gateway?

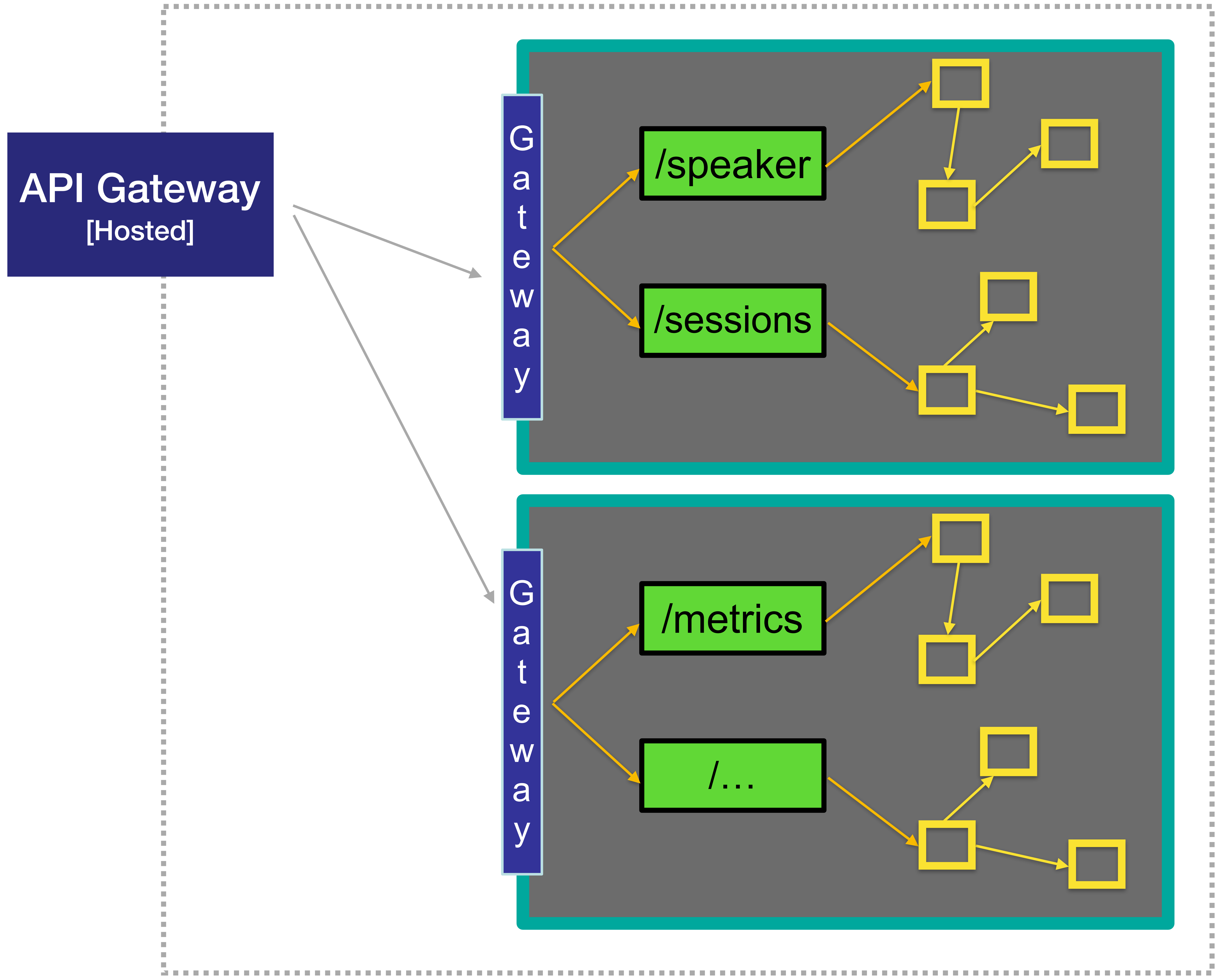


What is a Gateway?

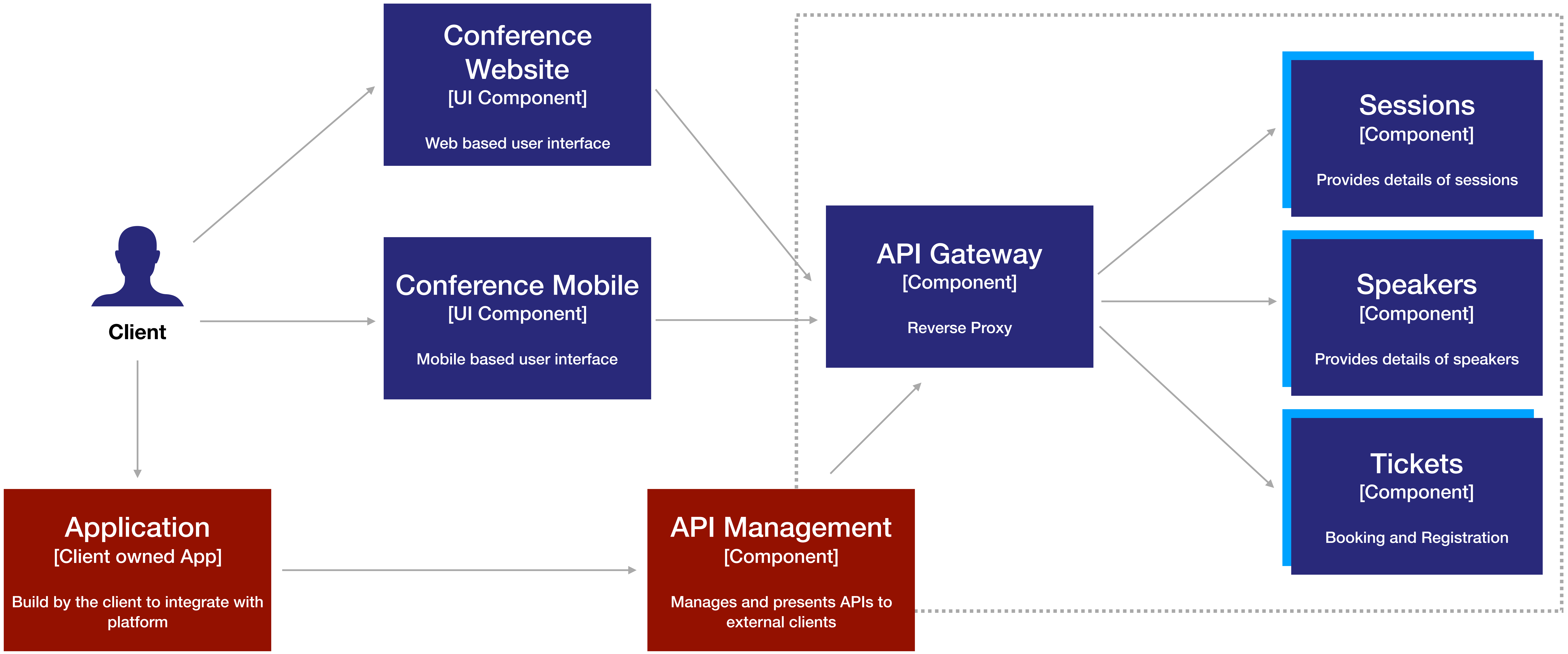


Building an API Centric Architecture

- Gateways allow for an API Centric strategy
- Build APIs as Microservices
- Delivery velocity at Microservices Gateway
- Repository of APIs at Enterprise Gateway
- Easy to extend and rapidly build
- API Governance and Curation Challenges



Role of API Management



Demo - Applying a Gateway

Creates a Network → todo-service

Builds out an image → image

Pulls in a gateway → image

```yaml
version: '3'
services:
  todo-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

```
@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://todo-service:8080"))
                .build();
    }

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

}
```