# ADR-5

## Use of a Dedicated API Gateway Microservice for Routing

Date: 15-04-2025

## Status

Accepted

## Author

System Architect

## Context

With multiple microservices powering different domains of the system (user, facility, booking, and payment), secure and efficient communication between clients and services is crucial. Direct exposure of internal services can introduce security vulnerabilities and tight coupling. A routing solution is needed to centralize request entry and policy enforcement.

## Decision

We will introduce a dedicated API Gateway microservice responsible for routing all incoming requests to appropriate downstream services. This gateway will be the only exposed entry point for clients and will handle routing, rate limiting, request authentication, and basic logging.

## Alternatives

| Option | Pros | Cons |
| --- | --- | --- |
| No Gateway (Expose each service) | Simple to start, no added layer | Insecure, harder to manage auth/routing, multiple open ports |
| Shared monolithic API router | Unified control logic | Violates separation of concerns, low scalability |

| | | |
|---|---|---|
| Use existing open-source gateway | Mature, feature-rich, community-supported | Might be overkill or hard to customize for a course-scale project |
| Dedicated API Gateway Microservice (Chosen) | Centralizes access, improves security, supports cross-cutting concerns like auth and logging | Adds one more deployment unit, potential bottleneck if not scaled |

## Rationale

Using a gateway abstracts and centralizes routing logic, reduces security risks by not exposing internal service ports, and improves maintainability. It enables consistent enforcement of access policies and eases future enhancements like load balancing, service versioning, and API rate-limiting.

## Consequences

- **Positive**: Improved security, modularity, and scalability; simpler client interface; better observability.
- **Negative**: Slight added complexity and potential single point of failure if gateway is not properly managed.

## References

- https://microservices.io/patterns/apigateway.html