ADR-1

Adoption of Microservices Architecture

Date: 13-04-2025

Status

Accepted

Author

System Architect Team

Context

PitchPlease involves distinct domains: user management, facility listings, bookings, and payments. Each has different scaling, availability, and maintenance needs. A monolithic approach would lead to tight coupling and reduced agility.

Decision

Adopt a **microservices architecture**, with four main services:

- User
- Facility
- Booking
- Payment

Each service will have its own database, API, and deployment lifecycle.

Alternatives

Approach	Pros	Cons
Monolith	Simple, fast to build	Hard to scale, tightly coupled, single point of failure
Modular Monolith	Logical separation	Still one deployable unit

Microservices (Chosen)	Independent scaling, better isolation, team autonomy	DevOps overhead, inter-service complexity

Rationale

Microservices offer better fault isolation, team parallelism, independent scaling, and maintainability — all aligning with project needs.

Consequences

- Faster development and deployment
- Fault isolation, independent scaling
- Added complexity in ops, monitoring, and service communication

References

- https://martinfowler.com/articles/microservices.html