Diagrama de Componentes da API de Reservas

API Gateway
[Roteia requisições externas.]

Sistema de Check-in
Valida reservas/passageiros.

Encaminha requisições *[HTTPS/JSON]*

Encaminha requisições *[HTTPS/JSON]*

API de Reservas (Java/Spring Boot)
[container]

Passenger Controller
[Spring MVC RestController]
Recebe requisições HTTP relacionadas a passageiros.

Reservation Controller
[Spring MVC RestController]
Recebe requisições HTTP relacionadas a reservas.

Encaminha requisições *[HTTPS/JSON]* ' Vindo do Sistema de Check-in

Chama
[Java API] ' Ou um PassengerService dedicado, se complexo

Chama
[Java API]

Reservation Service
[Spring Service]
Orquestra a lógica de negócio principal de criação, consulta e atualização de reservas.

Checkin Data Controller
[Spring MVC RestController]
Expõe dados para consulta do Sistema de Check-in.

Usa *[Java API]*

Usa *[Java API]*

Usa *[Java API]* ' Para validar trechos/voos

Usa *[Java API]*

Usa *[Java API]*

Chama *[Java API]*

Usa *[Java API]*

Payment Integration Service
[Spring Service]
Coordena a interação com o Sistema de Pagamentos.

Seat Management Service
[Spring Service]
Gerencia seleção, bloqueio temporário e reserva de assentos.

GDS Integration Service
[Spring Service]
Lida com a lógica de sincronização e reservas via GDS.

Checkin Data Service
[Spring Service]
Fornece dados de reserva/assento validados para check-in.

Passenger Repository
[Spring Data JPA Repository]
Abstrai acesso aos dados de Passageiro no BD.

Reservation Event Publisher
[Spring AMQP Template]
Publica eventos de domínio (ex: ReservaConfirmada) na mensageria.

Usa *[Java API]*

Usa *[Java API]* ' Para locks

Usa *[Java API]* ' Para validar assentos na aeronave do trecho

Usa *[Java API]*

Usa *[Java API]*

Usa *[Java API]* ' Para criar/atualizar reservas do GDS

Usa *[Java API]*

Usa *[Java API]*

Publica mensagens *[AMQP]*

Payment Gateway Client
[SDK / RestTemplate]
Cliente para interagir com a API do Sistema de Pagamentos.

Seat Availability Cache
[Spring Data Redis]
Interage com o Cache Redis para locks e disponibilidade de assentos.

Flight API Client
[Feign Client / RestTemplate]
Cliente HTTP para consumir a API de Voos.

GDS API Client
[SDK / SOAP/REST Client]
Cliente para interagir com as APIs dos Sistemas GDS.

Reserved Seat Repository
[Spring Data JPA Repository]
Abstrai acesso aos dados de Reserva_Assento no BD.

Reservation Repository
[Spring Data JPA Repository]
Abstrai acesso aos dados de Reserva no BD.

Sistema de Mensageria
[RabbitMQ]
Fila para eventos assíncronos.

Lê/Escreve dados de Passageiro *[JDBC]*

Faz chamadas HTTP API *[HTTPS/JSON]*

Lê/Escreve chaves *[Redis Protocol]*

Faz chamadas HTTP API *[HTTPS/JSON]*

Faz chamadas API *[HTTPS/SOAP/JSON]*

Lê/Escreve dados de Reserva_Assento *[JDBC]*

Lê/Escreve dados de Reserva *[JDBC]*

Sistema de Pagamentos
Processa pagamentos.

Cache
[Redis]
Cache de dados e locks.

API de Voos
[Python/Django]
Gerencia inventário de voos.

Sistemas GDS
Distribuem inventário e originam reservas.

Banco de Dados Central
[PostgreSQL]
Armazena todos os dados.

Legend
- person
- system
- container
- component
- external person
- external system
- external container
- external component