



THE DEVELOPER'S  
CONFERENCE

# Chatbots com microserviços

Luís Hansen

---

Engenheiro de computação

Matheus Kumano

---

Engenheiro de computação



# Chatbots

Chatbots

Interface  
cliente

# Chatbots

Interface  
cliente

Serviço de  
NLU / NLC

# Chatbots

Interface  
cliente

Serviço de  
NLU / NLC

STT / TTS

# Chatbots

Interface  
cliente

Serviço de  
NLU / NLC

STT / TTS

Integrações  
com outros  
sistemas



# Seguro

TMA médio de 1 minuto

130 mil atendimentos

57% Resolução no bot

Nota média 6/10



# Varejo

9 mil atendimentos realizados

TMA de 2,4 minutos

Mais de 50 assuntos no chatbot

Assertividade 70%

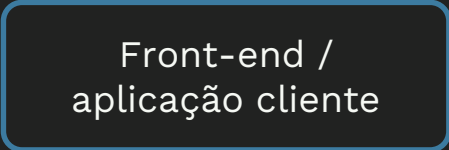


Service Desk

Arquitetura  
tradicional:  
**Monolítica**

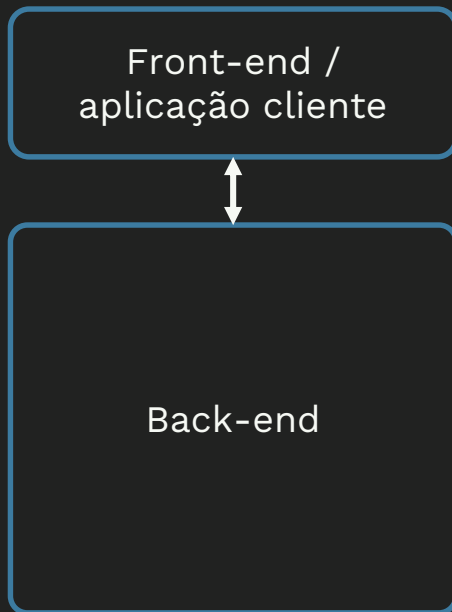
Arquitetura tradicional:

Monolítica

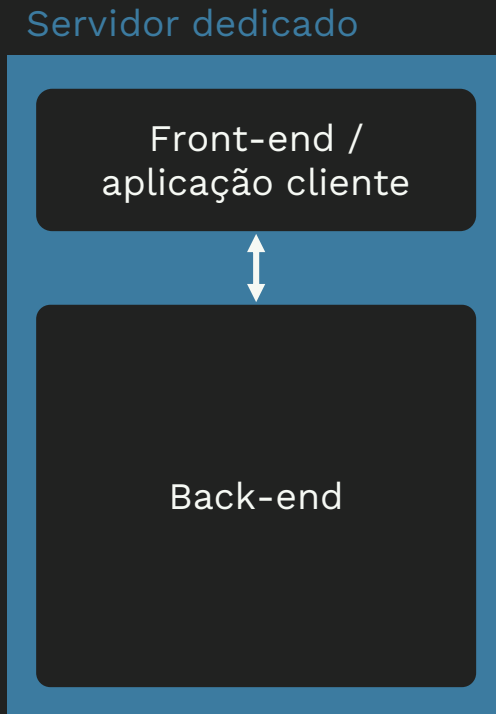


Front-end /  
aplicação cliente

Arquitetura tradicional:  
**Monolítica**

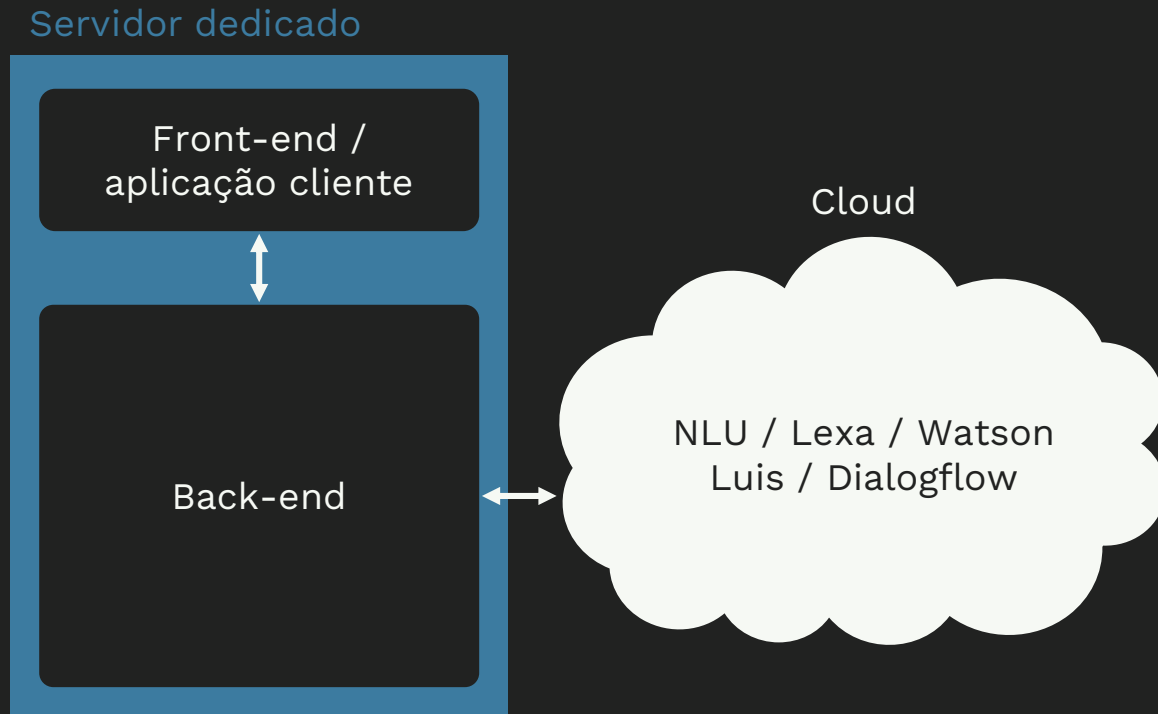


Arquitetura tradicional:  
**Monolítica**

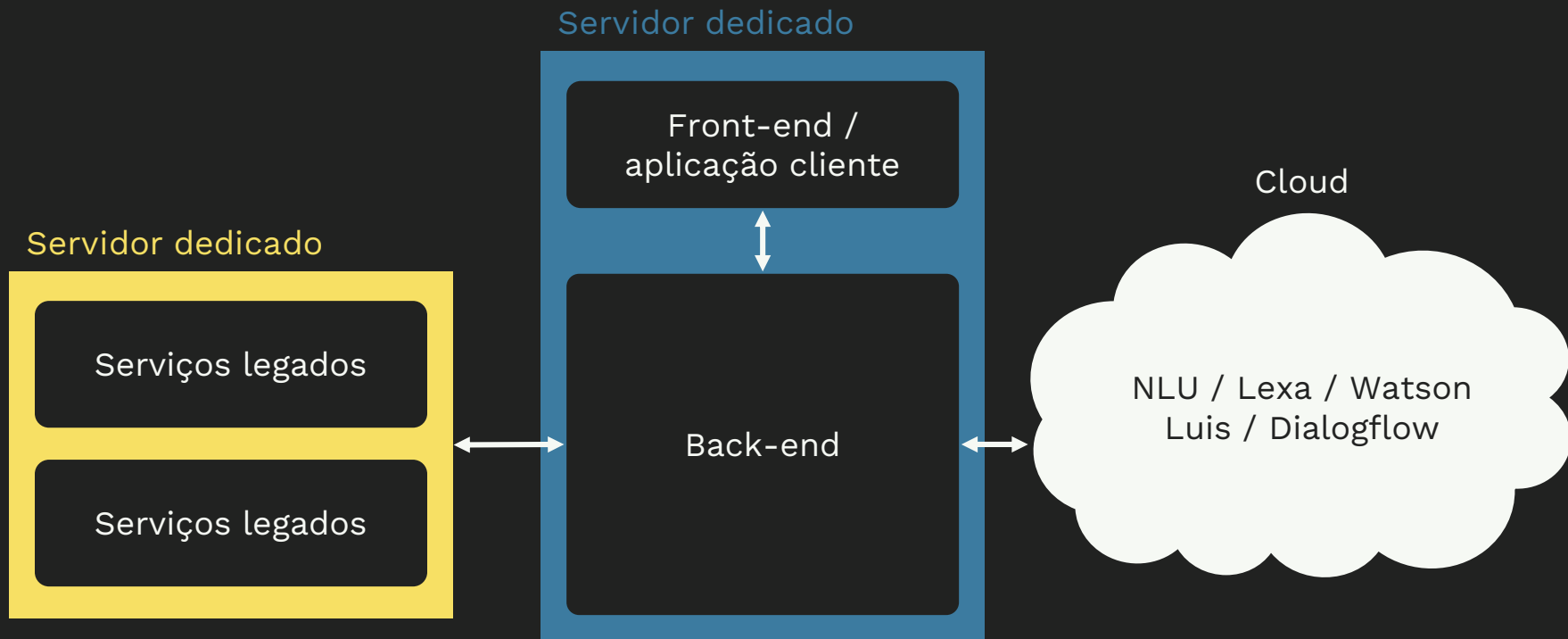


Arquitetura tradicional:

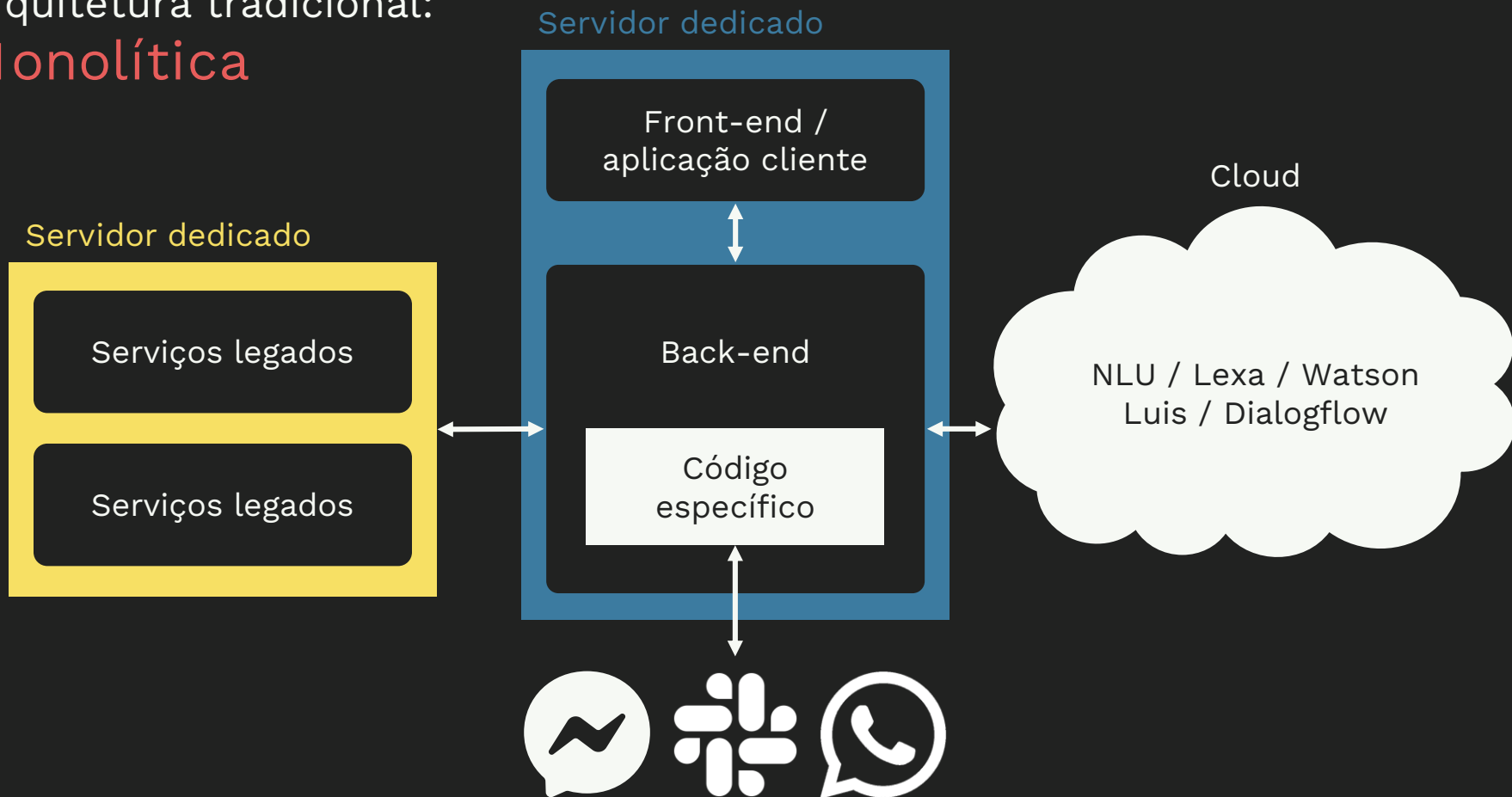
## Monolítica



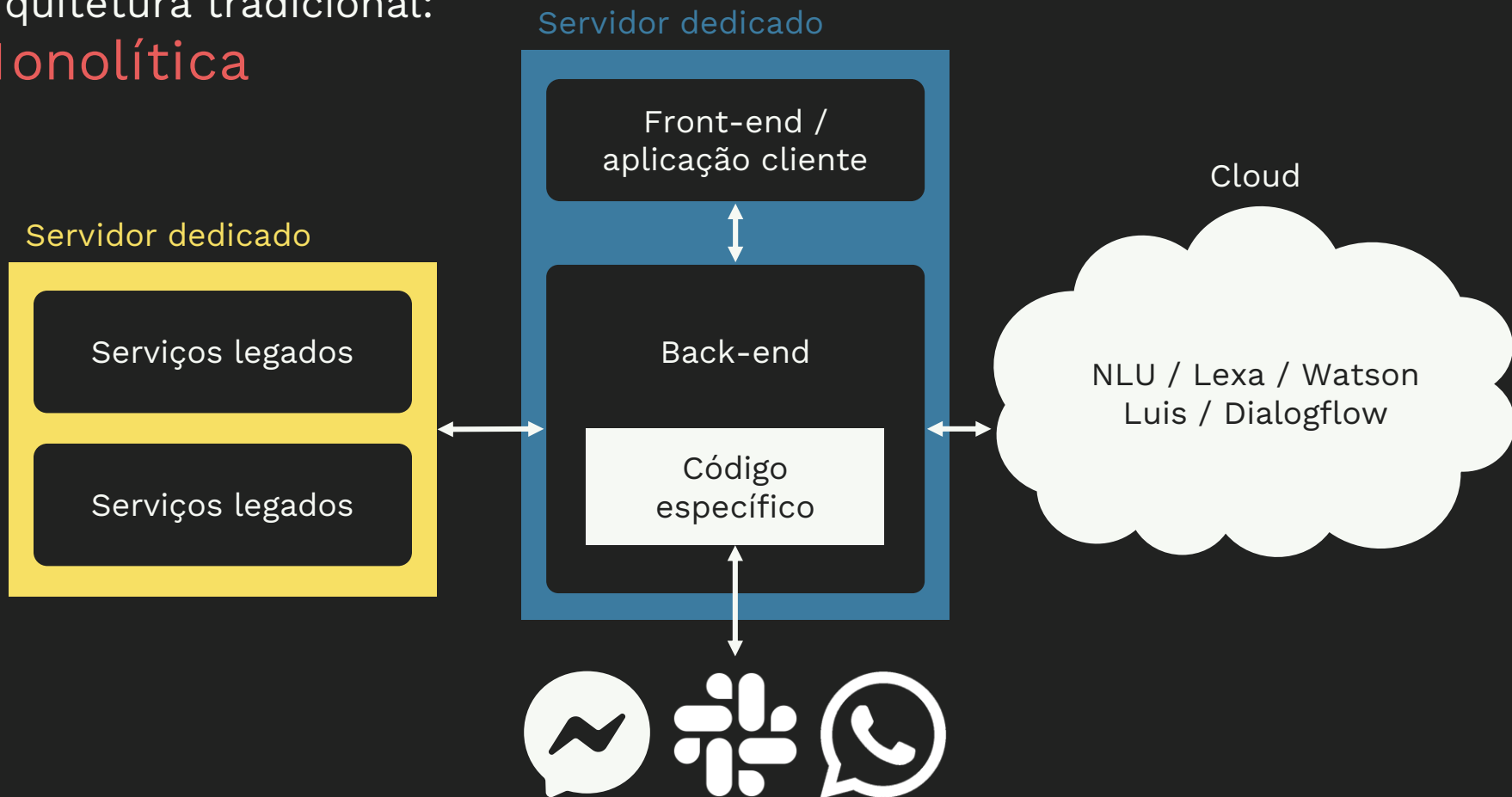
# Arquitetura tradicional: Monolítica



# Arquitetura tradicional: Monolítica

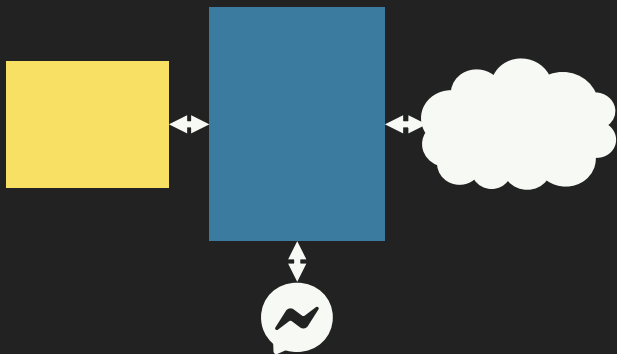


# Arquitetura tradicional: Monolítica





Arquitetura tradicional:  
**Monolítica**



- ✗ Novo deploy:
  - Atualização front/back
  - Mudanças nos legados
  - Mudanças em políticas da empresa
- ✗ Servidor dedicado
- ✗ Redundância / segurança / infraestrutura / load balancer
- ✗ Tende a se tornar um serviço legado
- ✓ Tudo dentro de casa

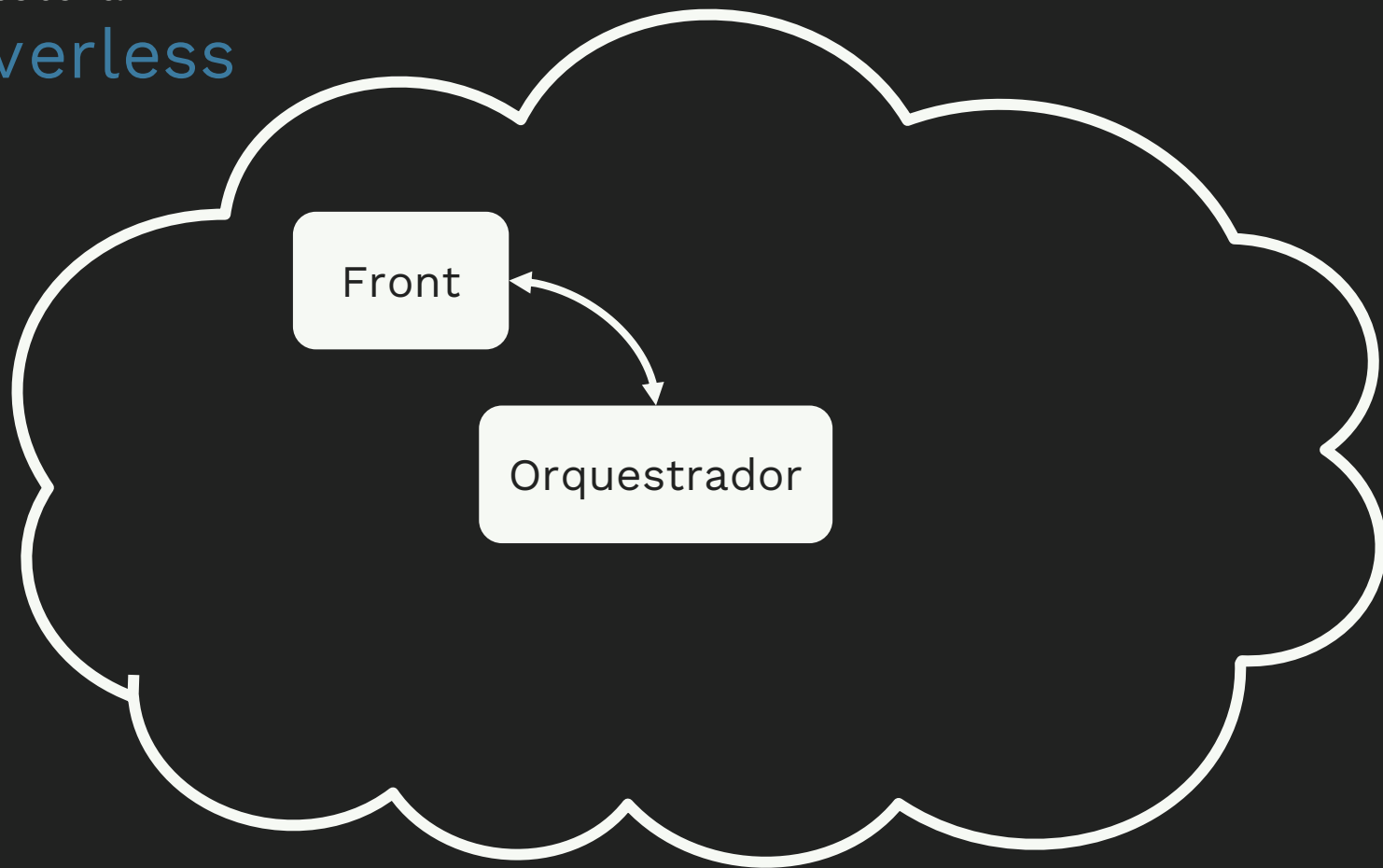
Arquitetura  
tradicional:  
**Monolítica**

Arquitetura  
tradicional:  
**Monolítica**

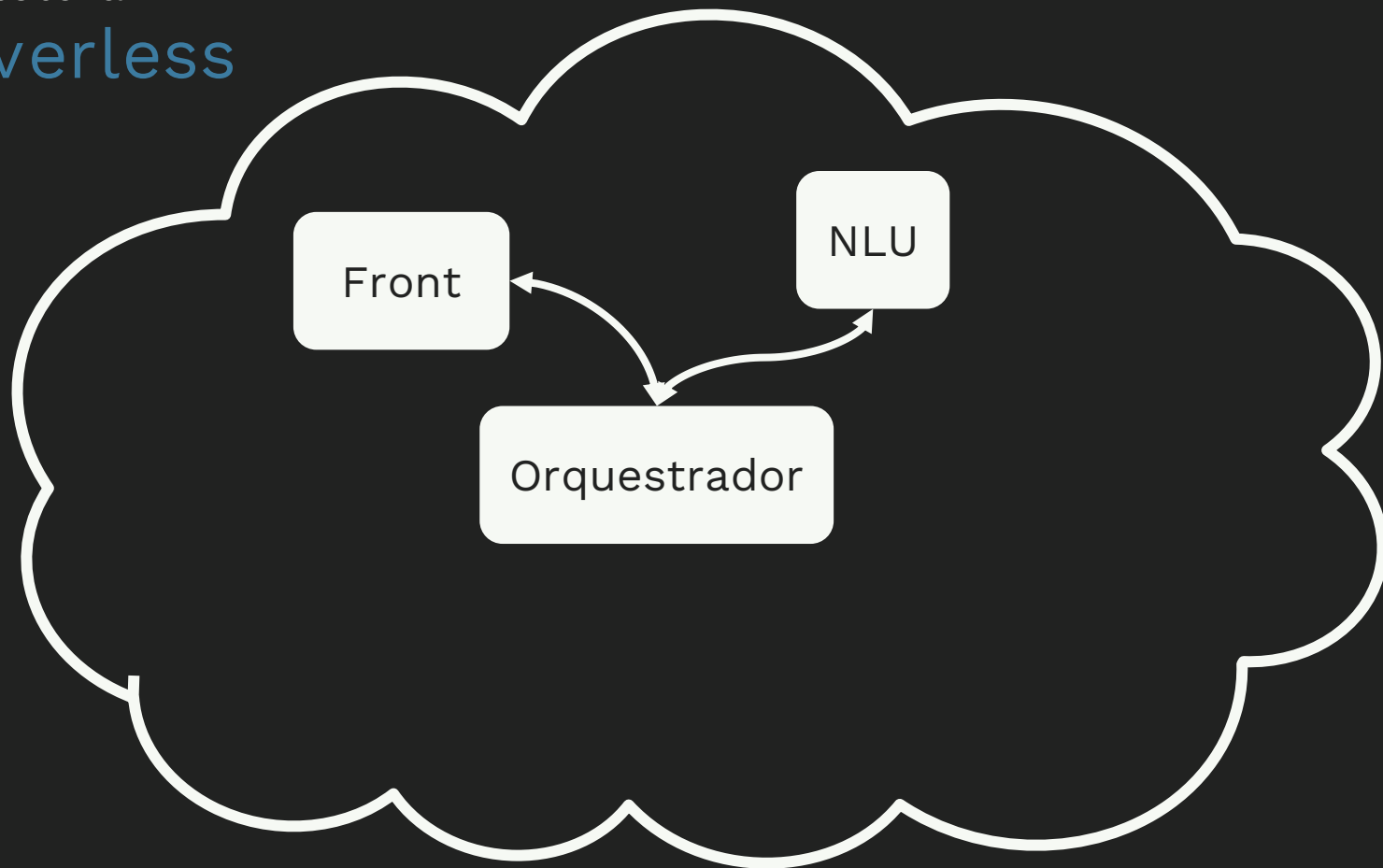
# Arquitetura Serverless

# Arquitetura Serverless

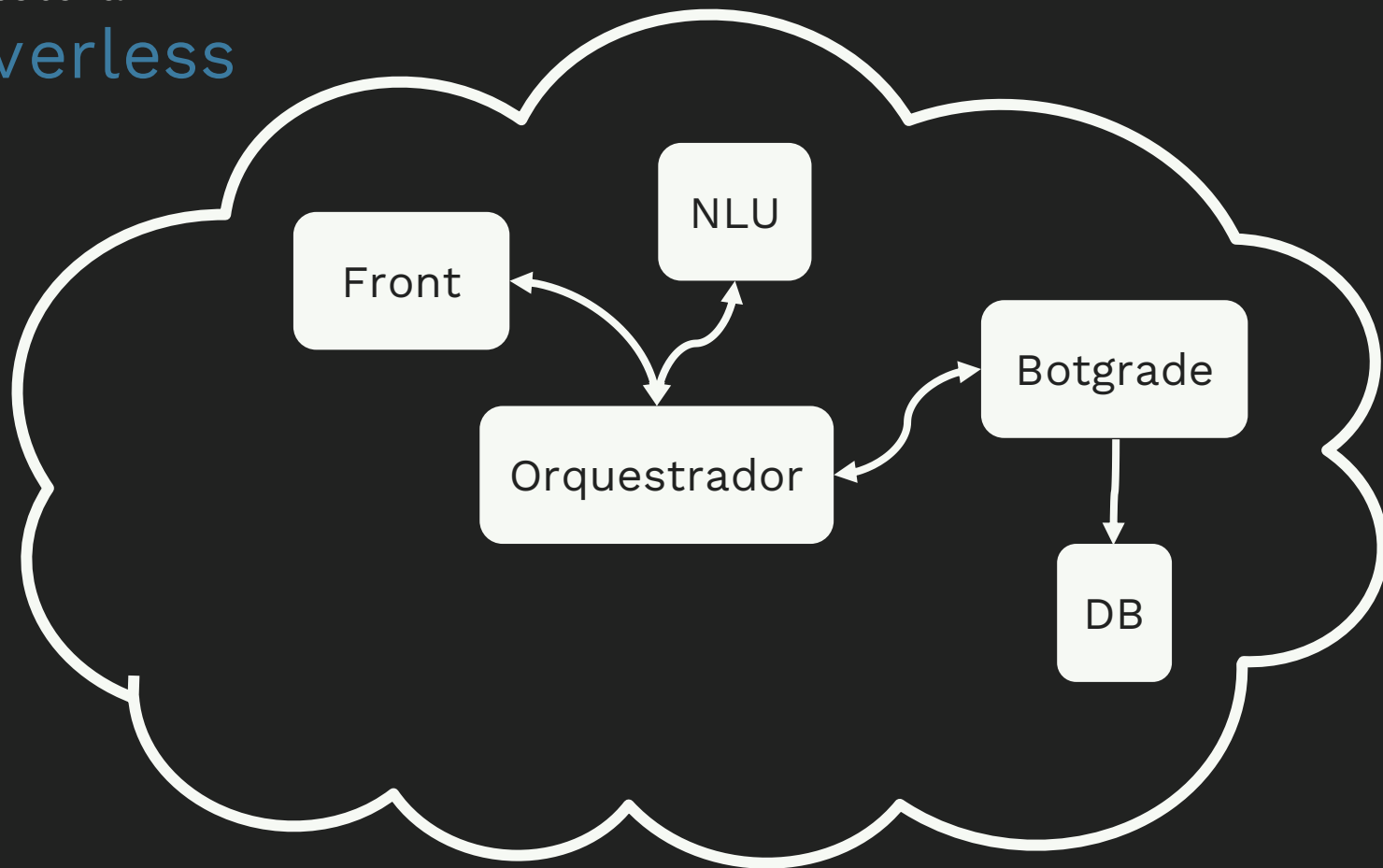
# Arquitectura Serverless



# Arquitectura Serverless

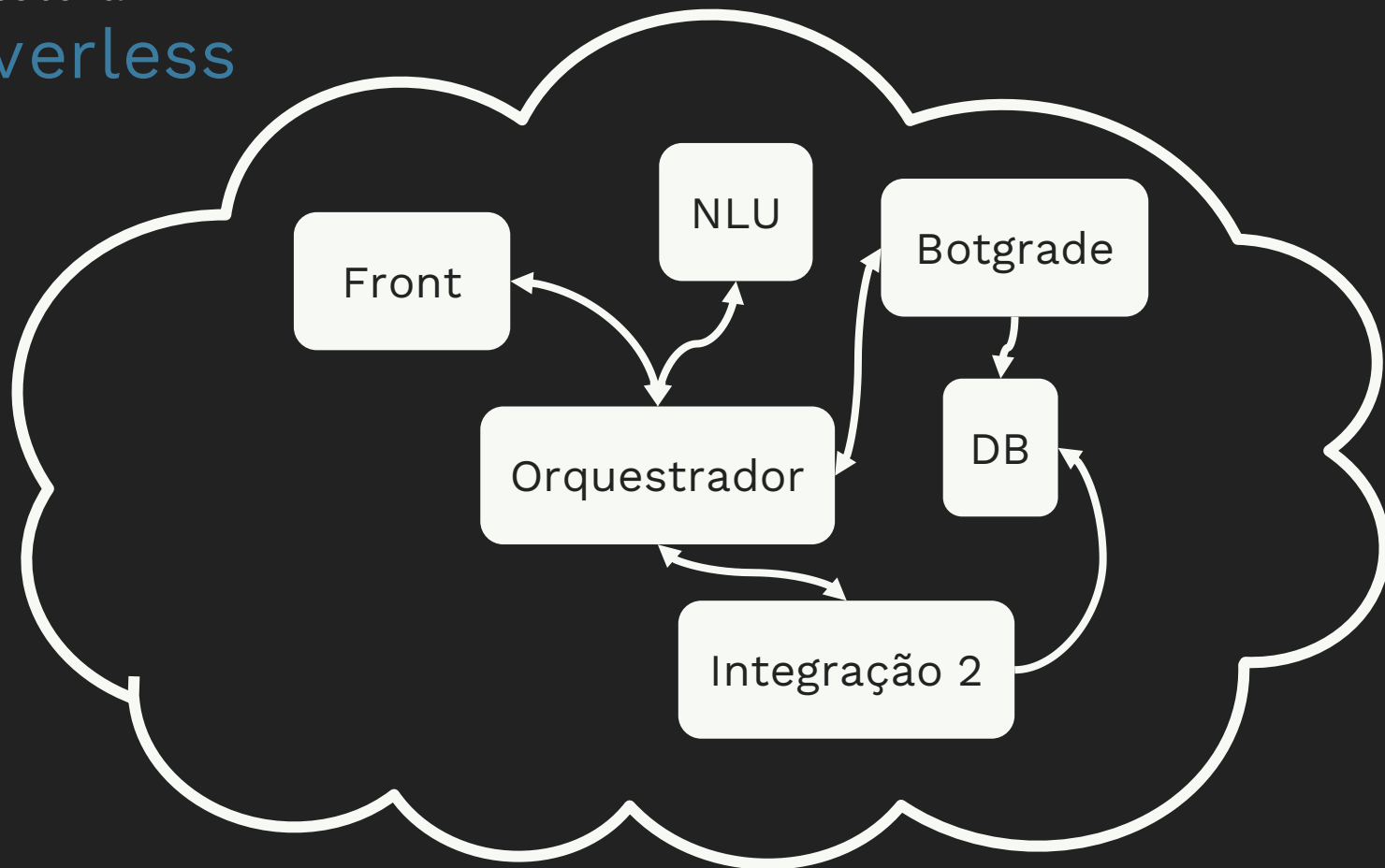


# Arquitectura Serverless

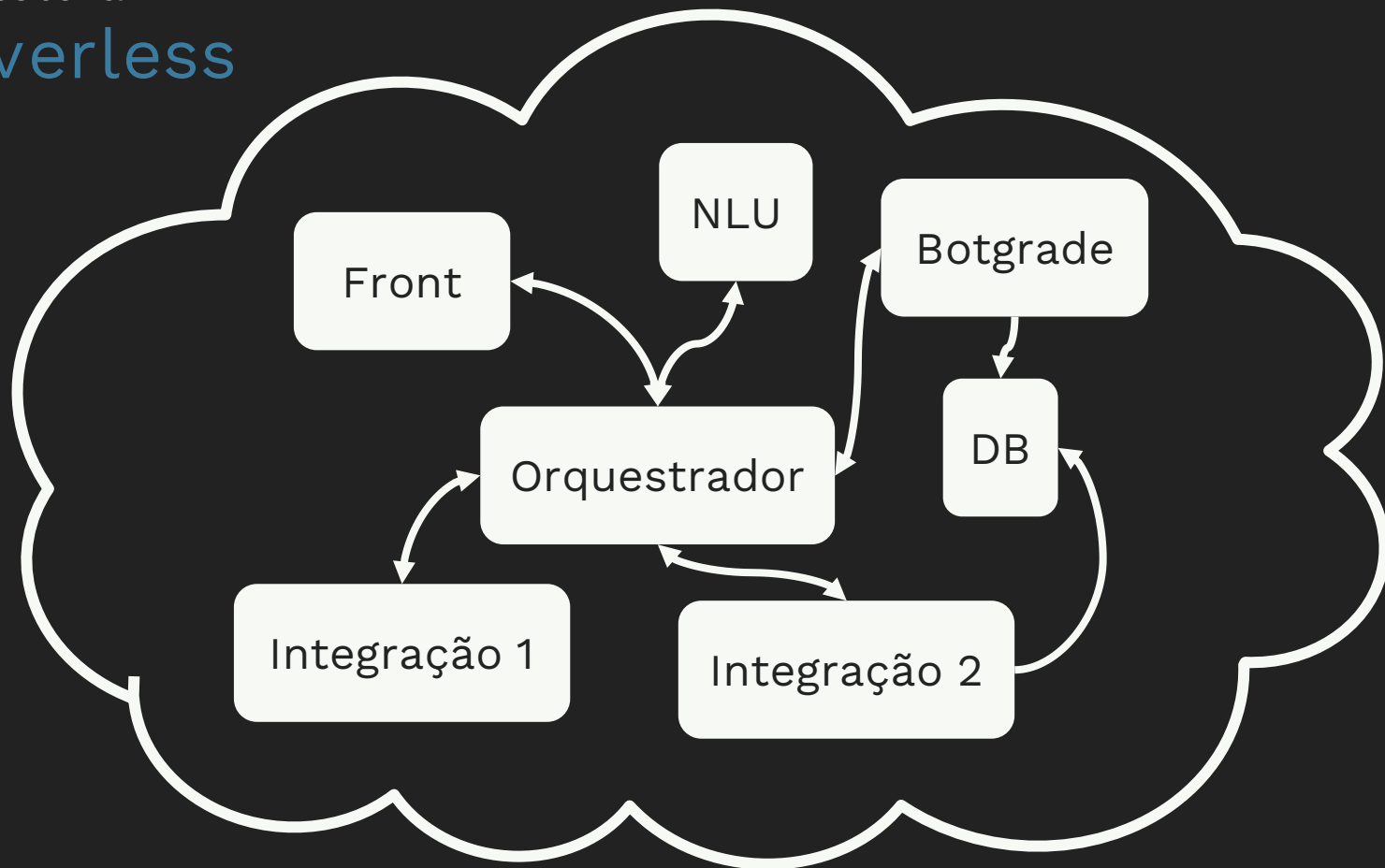




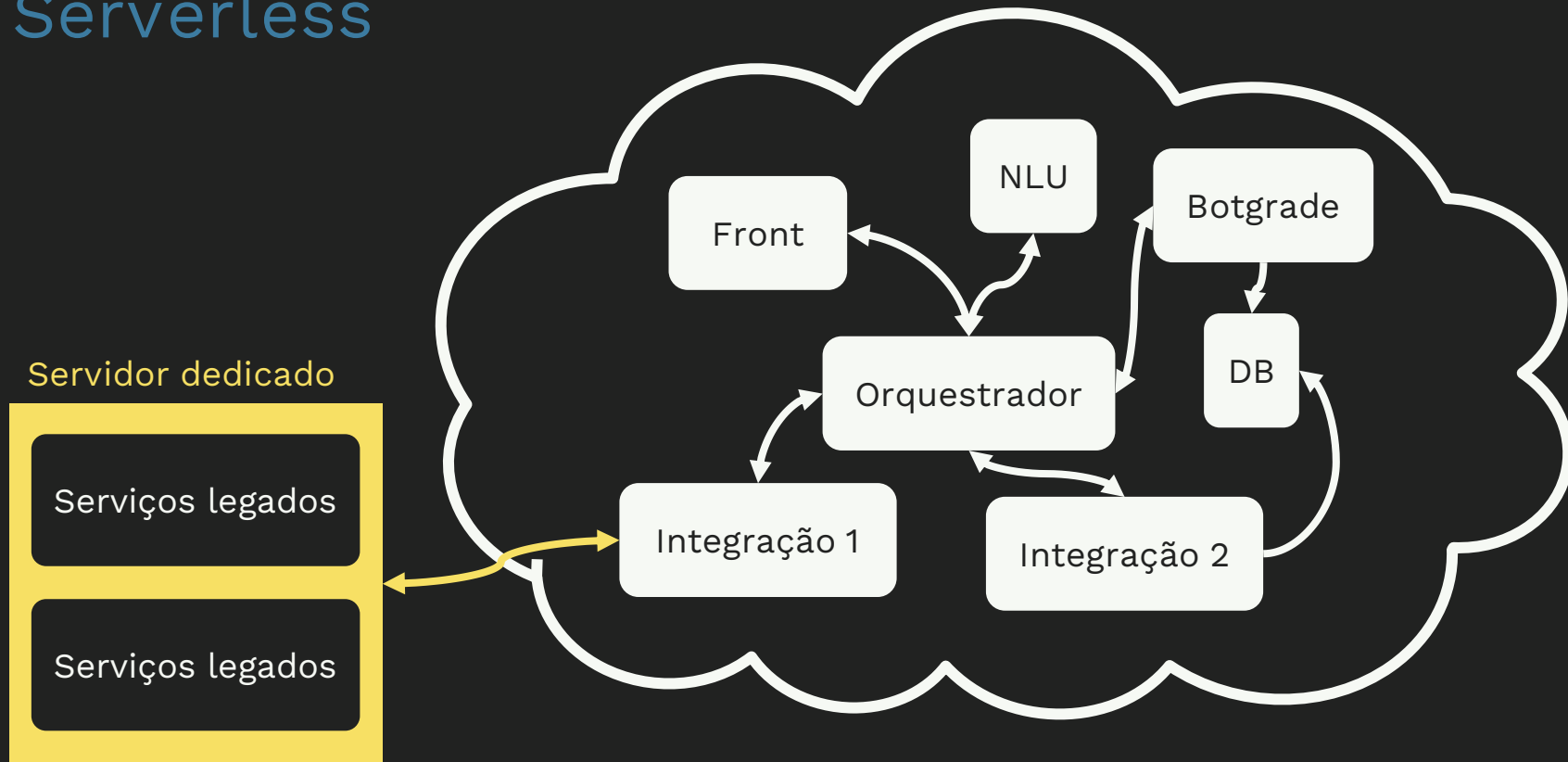
# Arquitetura Serverless



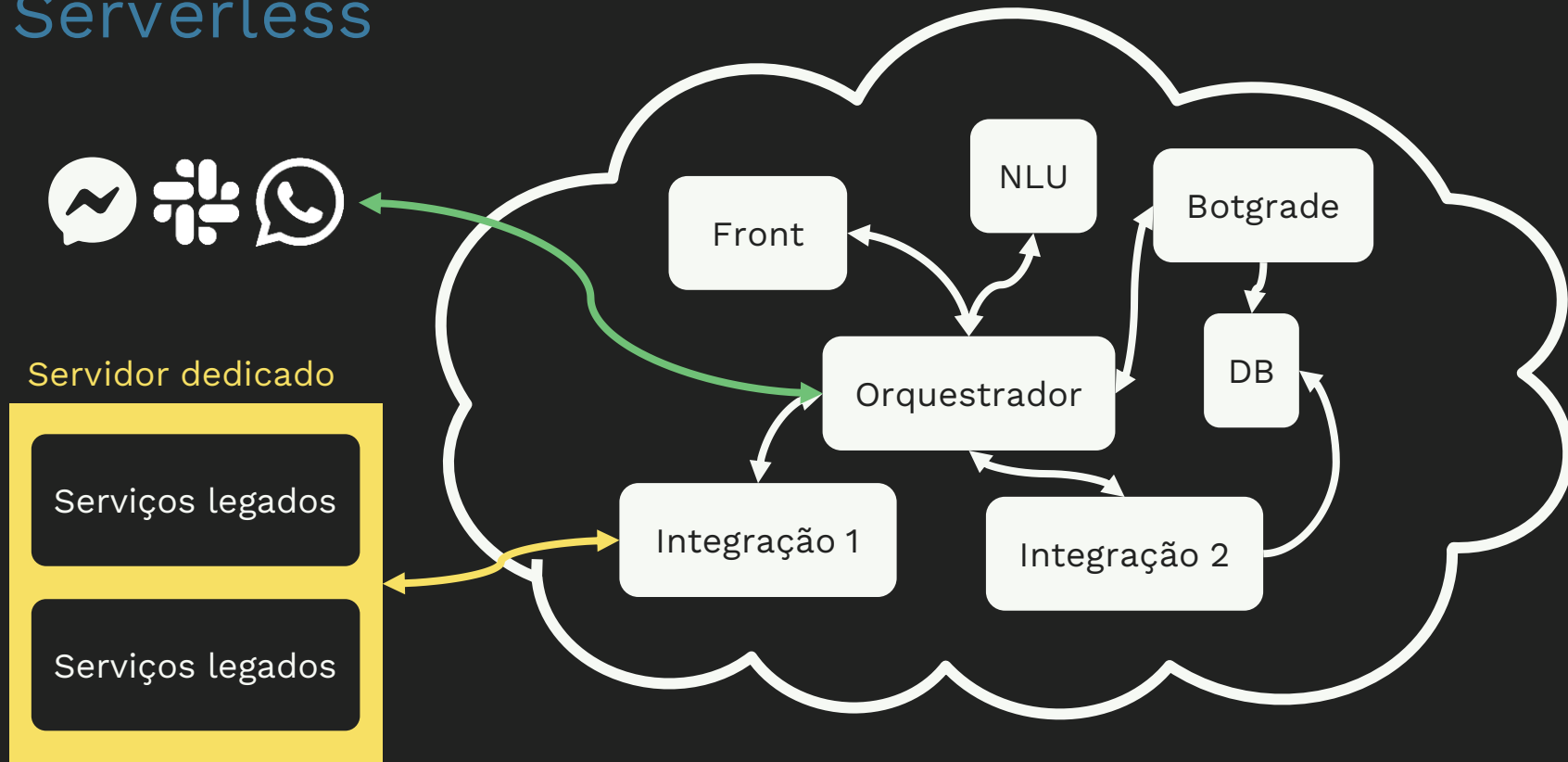
# Arquitetura Serverless



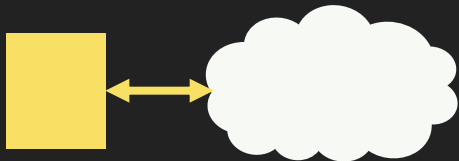
# Arquitetura Serverless



# Arquitetura Serverless



# Arquitetura Serverless



- ✓ Facilidade de implementação
- ✓ Escalabilidade
- ✓ Robustez
- ✓ Deploy individualizado com zero-downtime
- ✓ Integração com várias plataformas sem acoplamento
- ✗ Pode ficar caro
- ✗ Gerenciamento de aplicações mais complexo

# Arquitetura Serverless

Arquitetura  
Microserviços

Arquitetura  
Microserviços



Arquitetura

Microserviços



Orquestrador

# Arquitetura Microserviços

Orquestrador

Integração  
legada

# Arquitetura Microserviços

Orquestrador

Integração  
legada

Front-end

# Arquitetura Microserviços

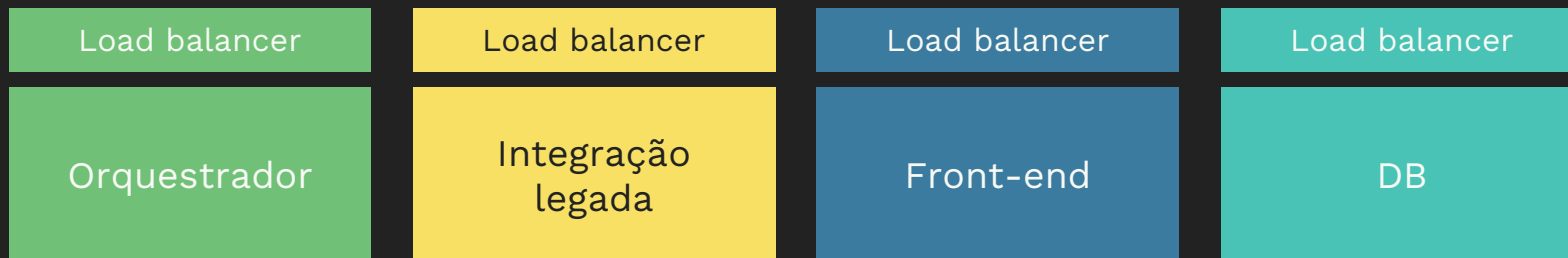
Orquestrador

Integração  
legada

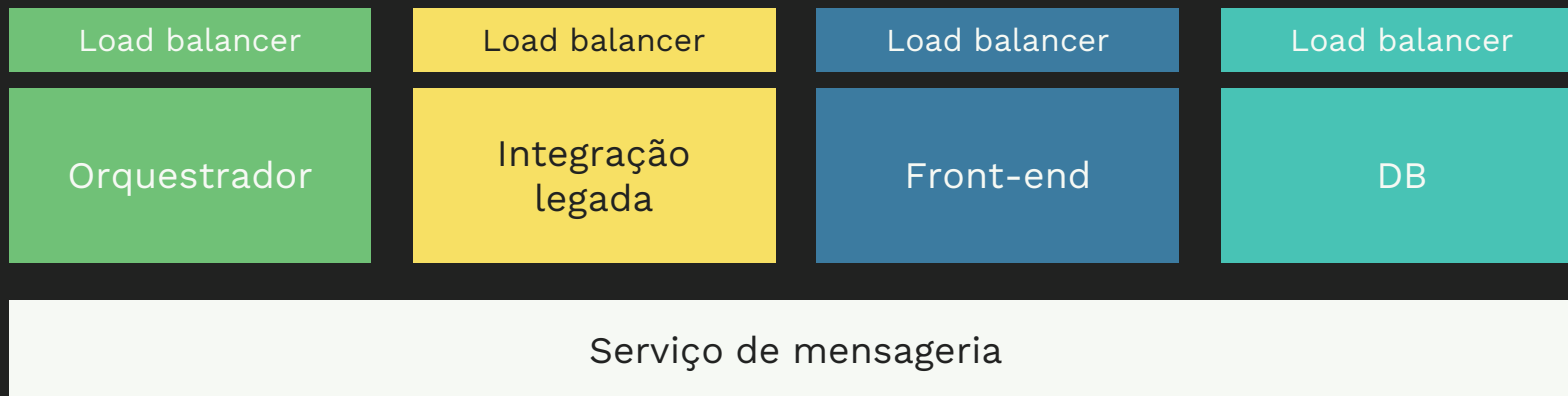
Front-end

DB

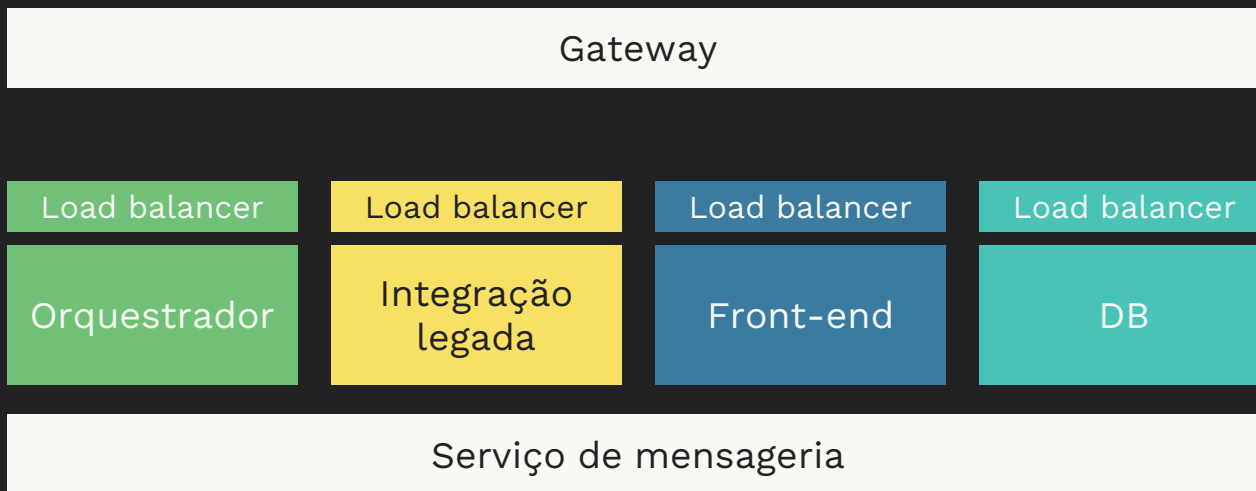
# Arquitetura Microserviços



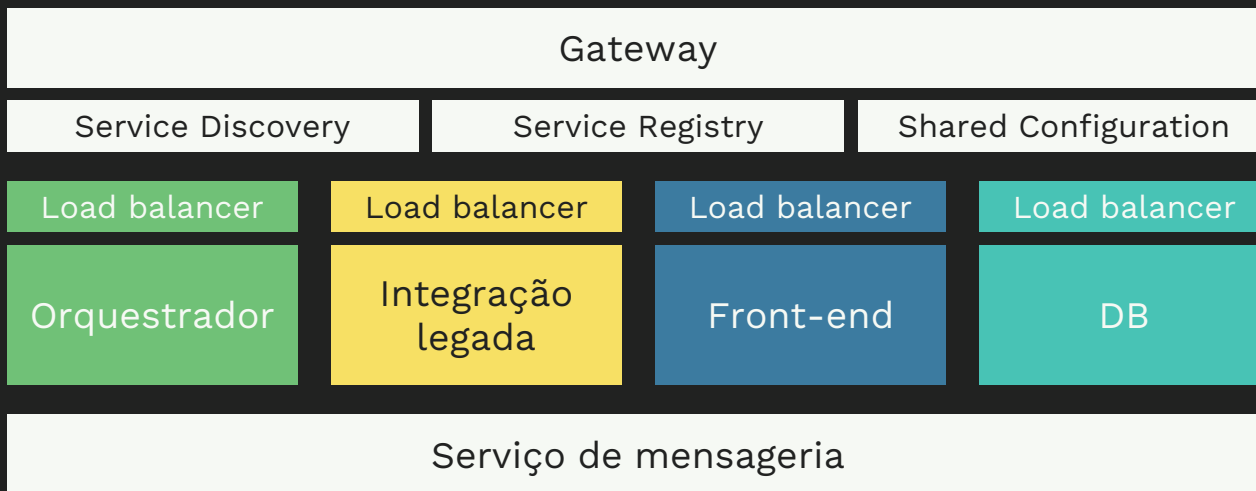
# Arquitetura Microserviços



# Arquitetura Microserviços

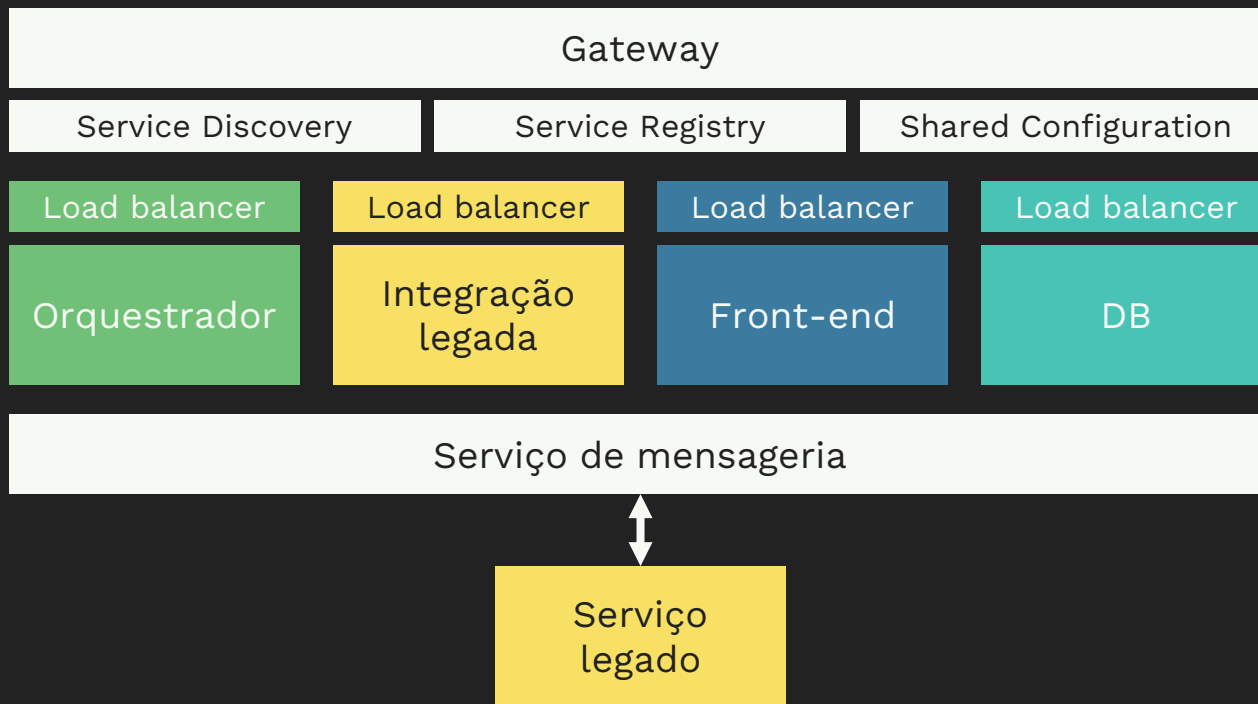


# Arquitetura Microserviços

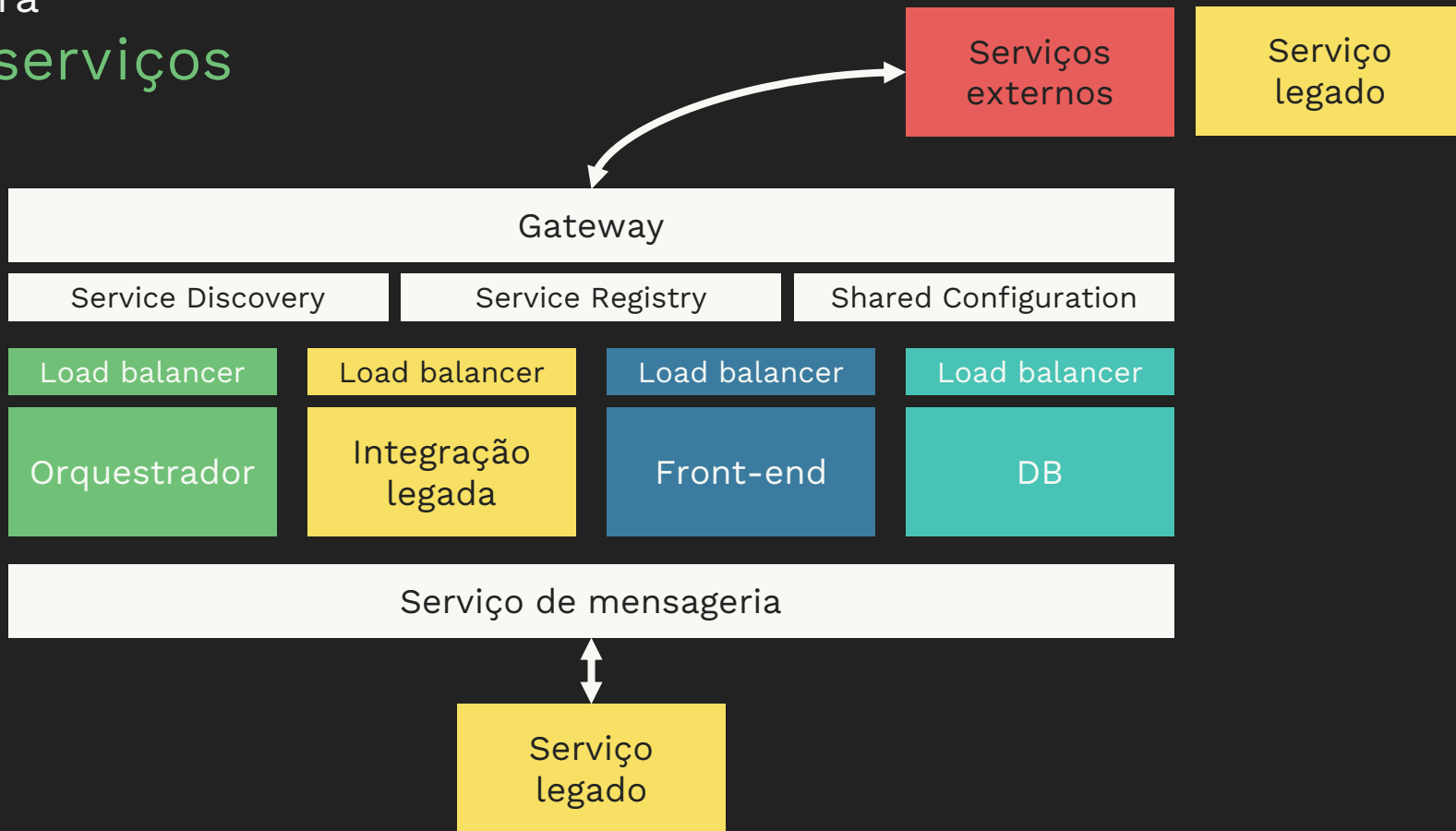


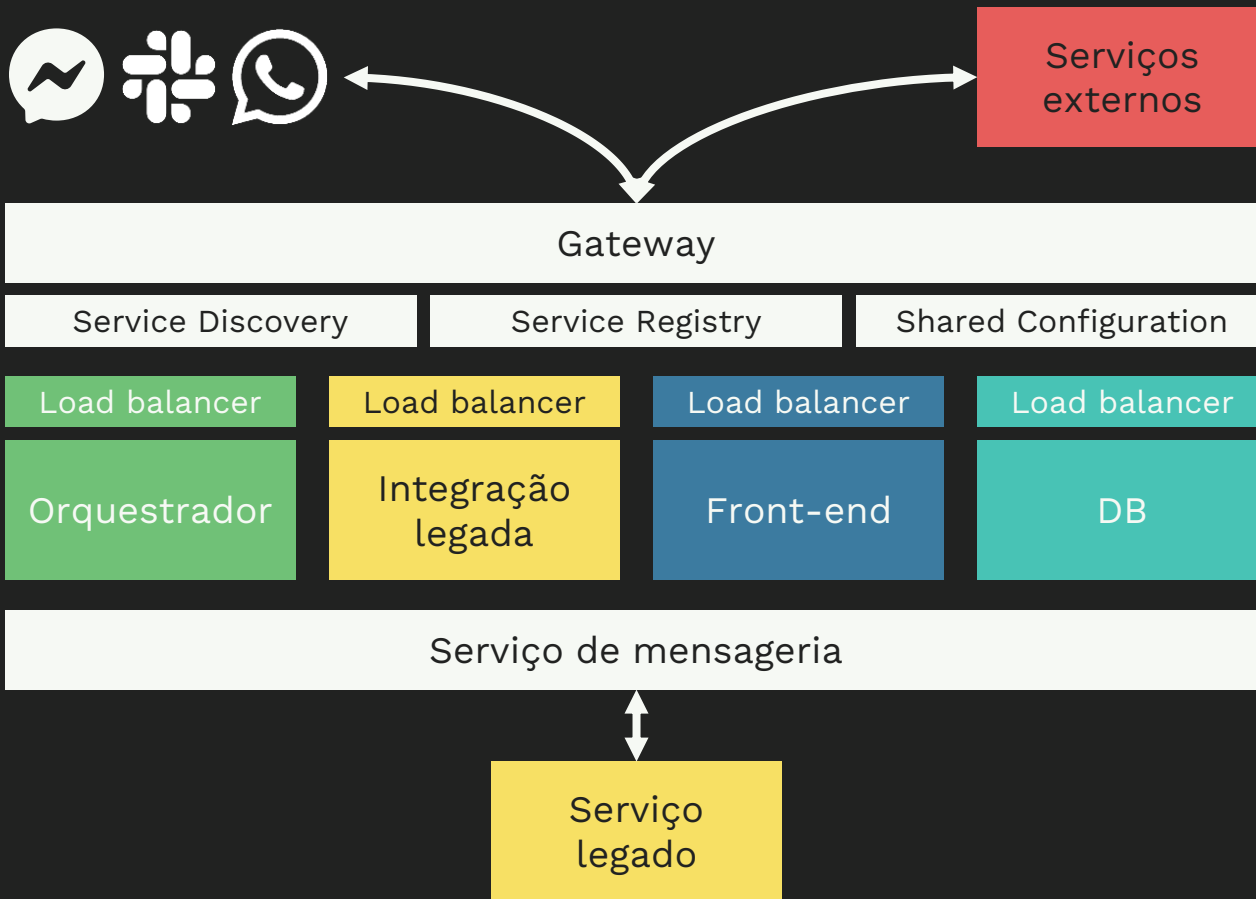


# Arquitetura Microserviços



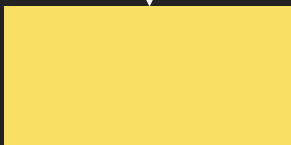
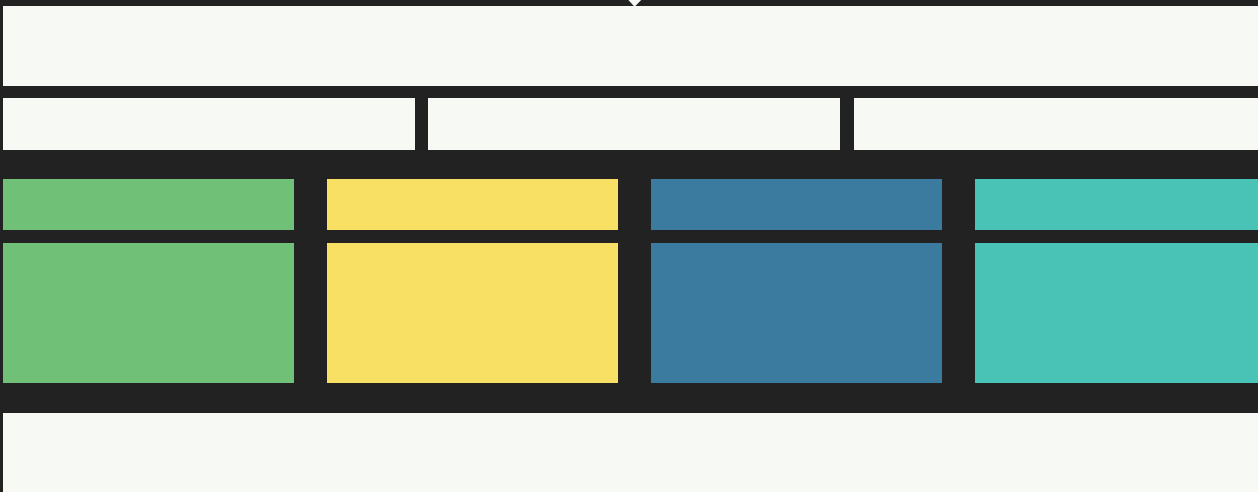
# Arquitetura Microserviços





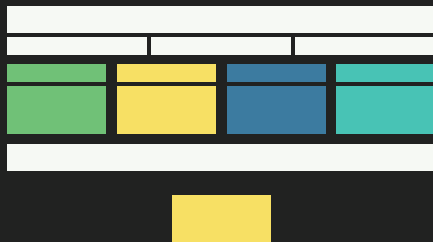


Serviços  
externos



Arquitetura

## Microserviços



- ✓ Preço
- ✓ Controle fino de escalabilidade com Kubernetes
- ✓ Pode ser usado em servidores próprios
- ✗ Arquitetura mais complexa

# Chatbots do futuro

- Muitas integrações
- Omnichannel
- Precisa de uma arquitetura escalável

# Chatbots



# Microserviços



*certsys*  
labs





*certsys*  
labs



[www.certsys.com.br](http://www.certsys.com.br)

Obrigado!



THE DEVELOPER'S  
CONFERENCE

