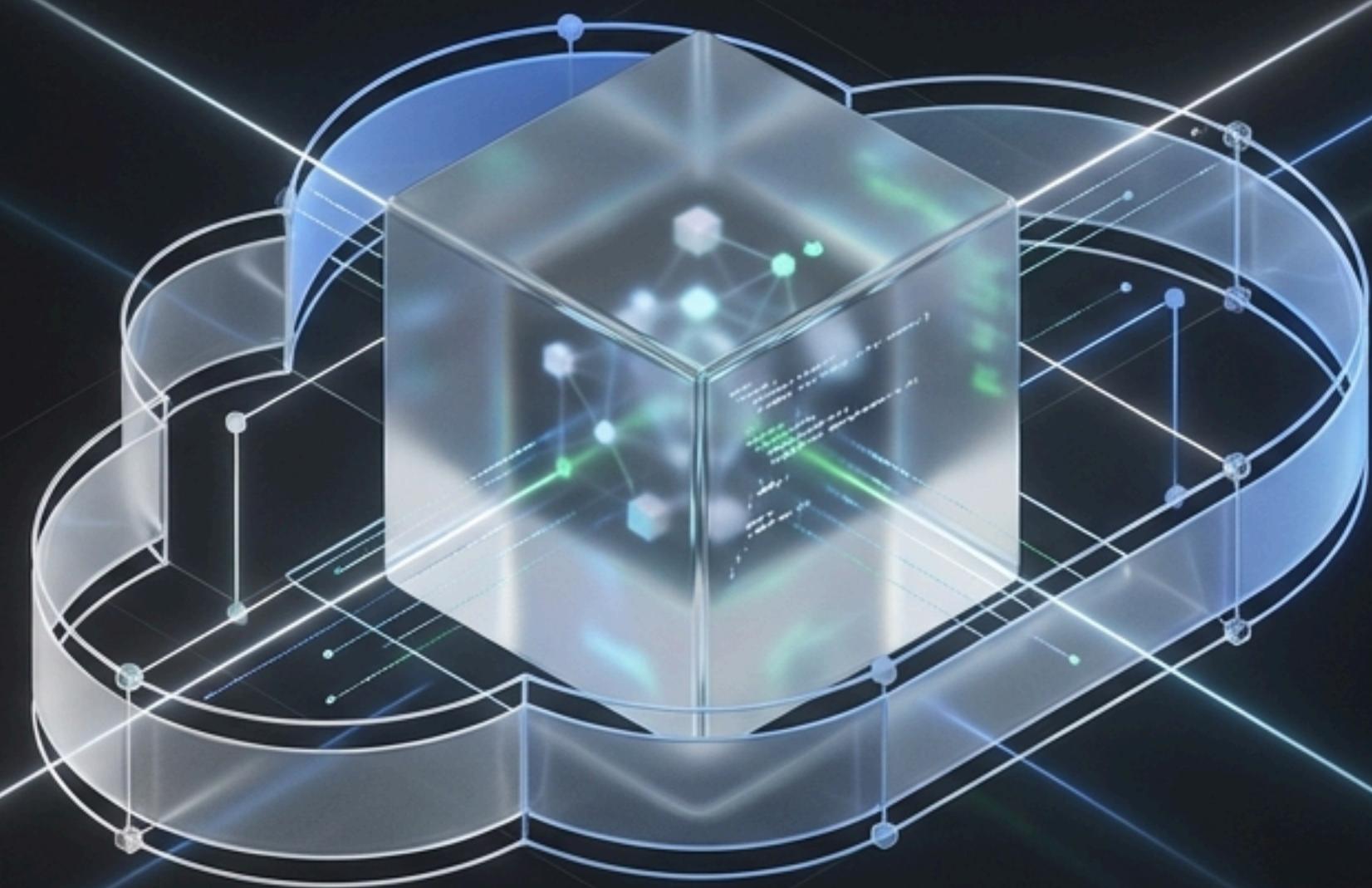


# Arquitetura Cloud Native & DevOps

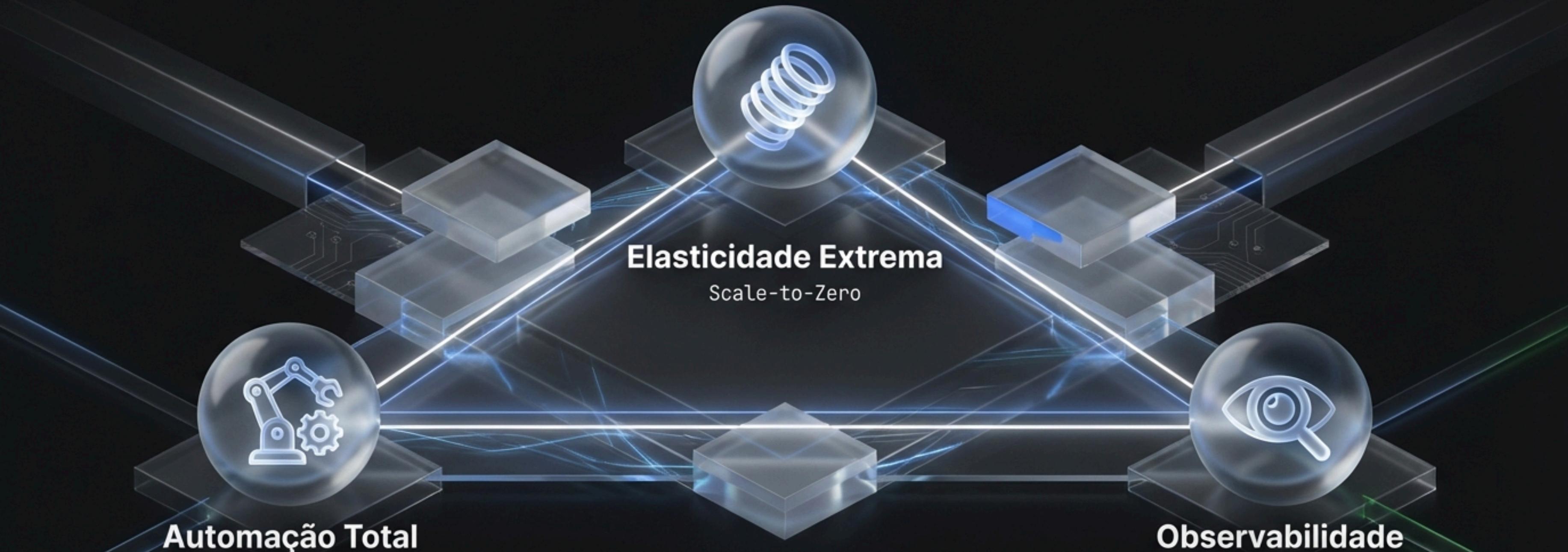
Engenharia de Microsserviços, Containerização e Automação Contínua



JetBrains Mono

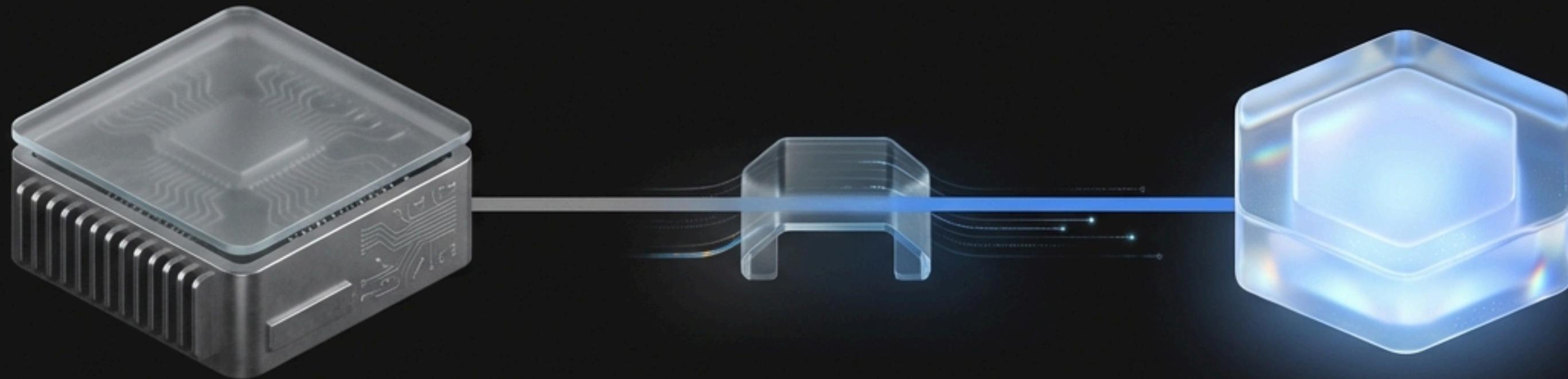
PROJETO INTEGRADOR - RESIDÊNCIA EM TIC 20 | JANEIRO, 2026

# O Triângulo de Tensão da Engenharia



O desafio: Criar um sistema que “dorme” para economizar recursos, acorda instantaneamente sob demanda e relata sua própria saúde sem falhas.

# A Decisão pelo PaaS: Abstração e Foco no Artefato



## IaaS / EC2

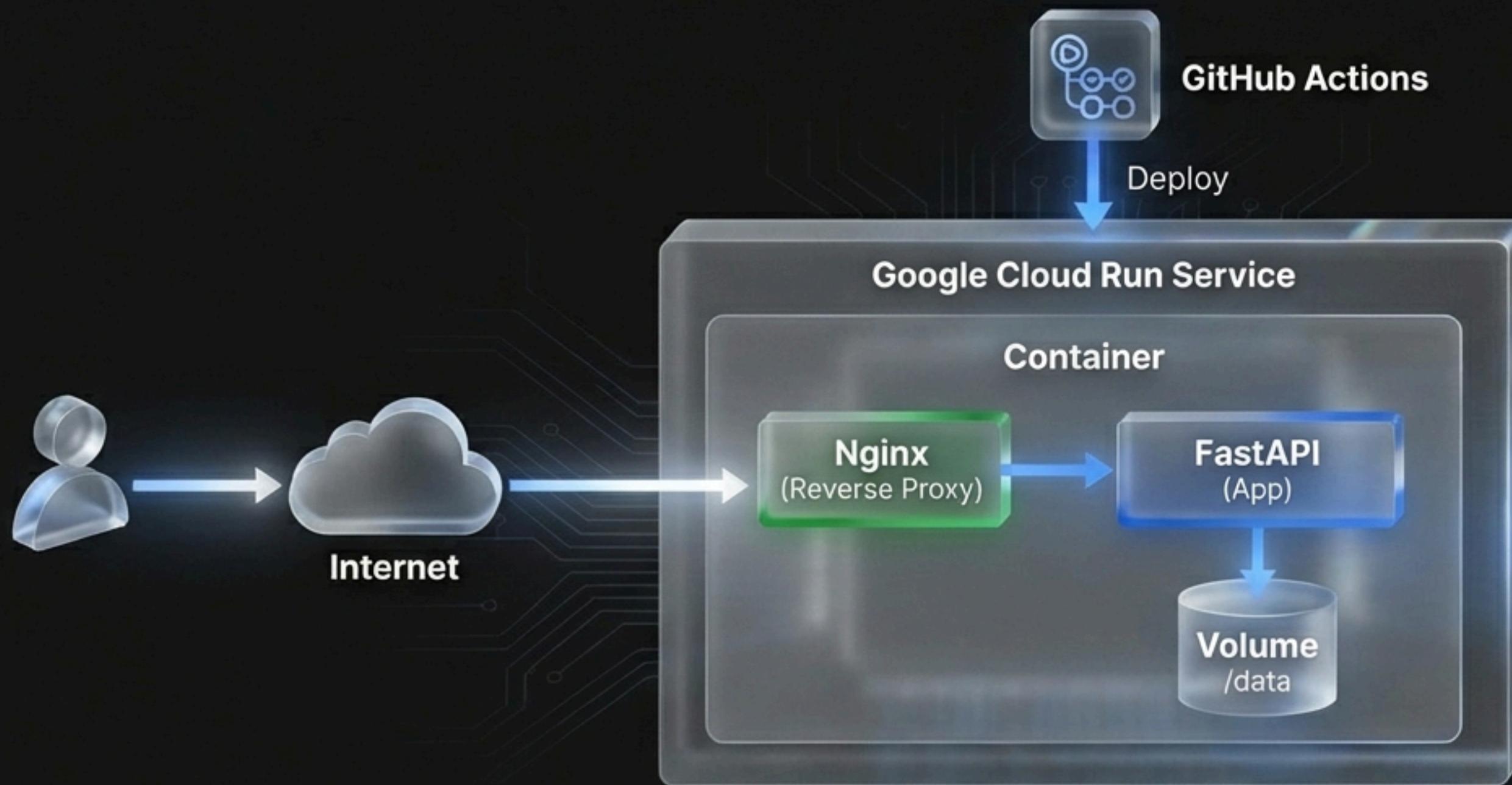
- Alta Gestão de OS
- Patching Manual
- Custo por Instância Ligada

## Google Cloud Run

- Abstração de Infraestrutura
- HTTPS Gerenciado
- Cobrança por Segundo (Pay-as-you-go)

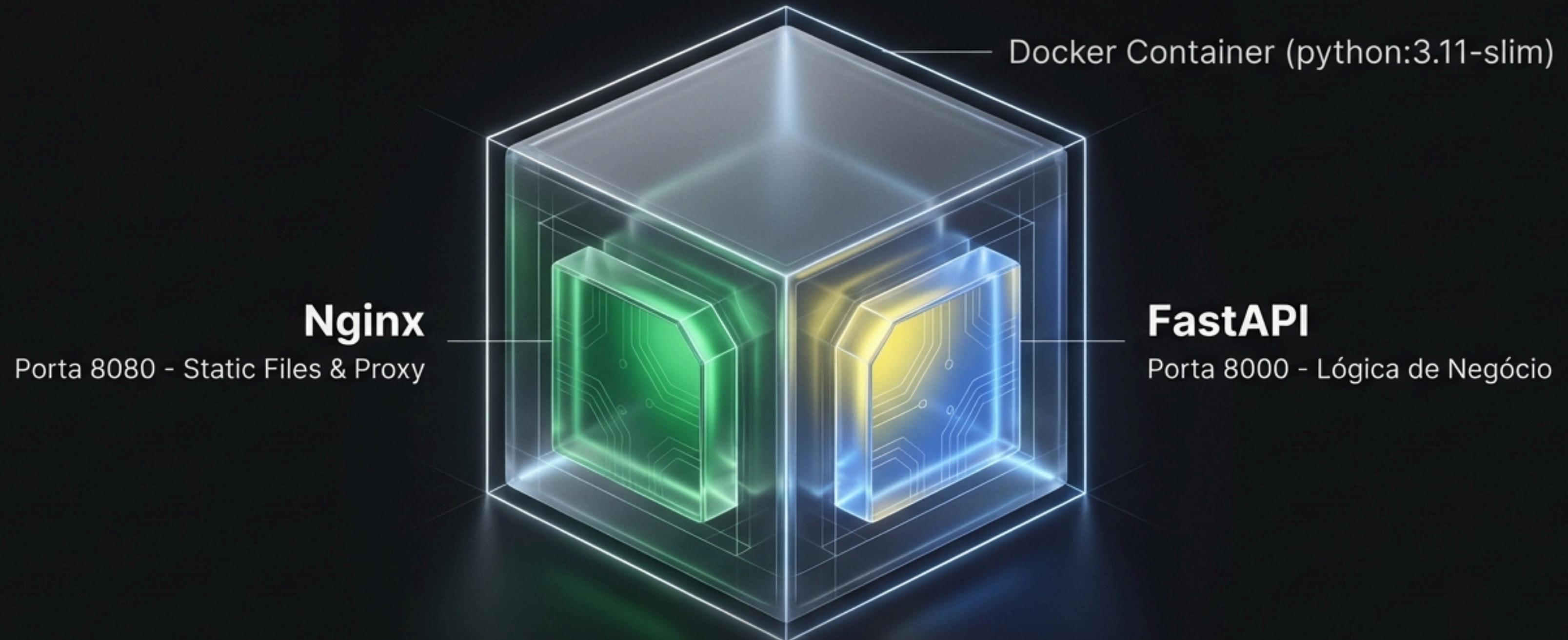
**A escolha do Cloud Run elimina o gerenciamento de hardware e SO, permitindo foco exclusivo no container Docker e na lógica de negócio**

# O Fluxo de Requisição e Componentes



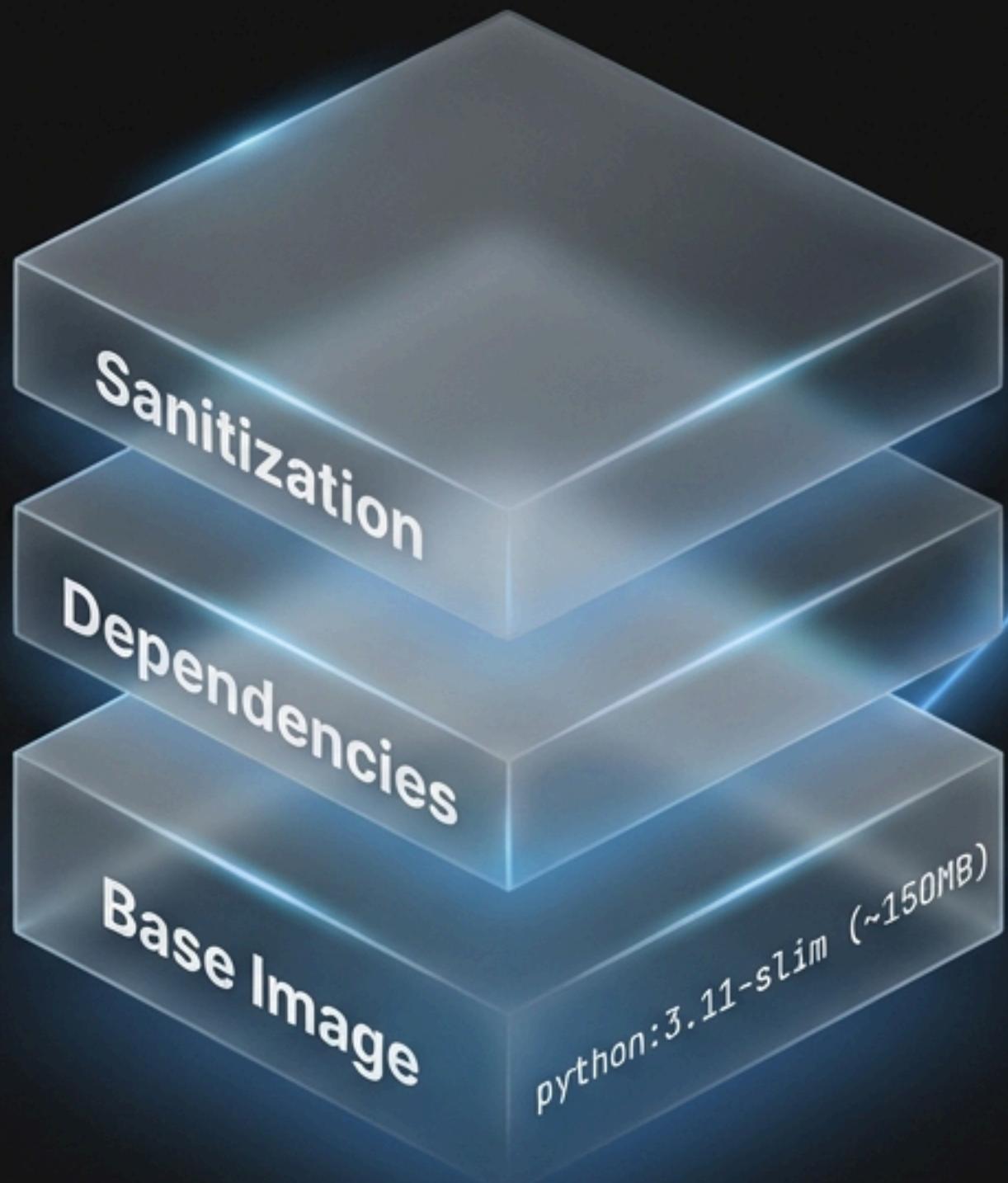
**Uma arquitetura encapsulada onde a simplicidade externa esconde uma orquestração interna robusta.**

# Anatomia do Container Híbrido



Coexistência estratégica para redução de latência de rede interna  
(loopback) e simplificação da unidade de deploy.

# Engenharia do Dockerfile e Camadas



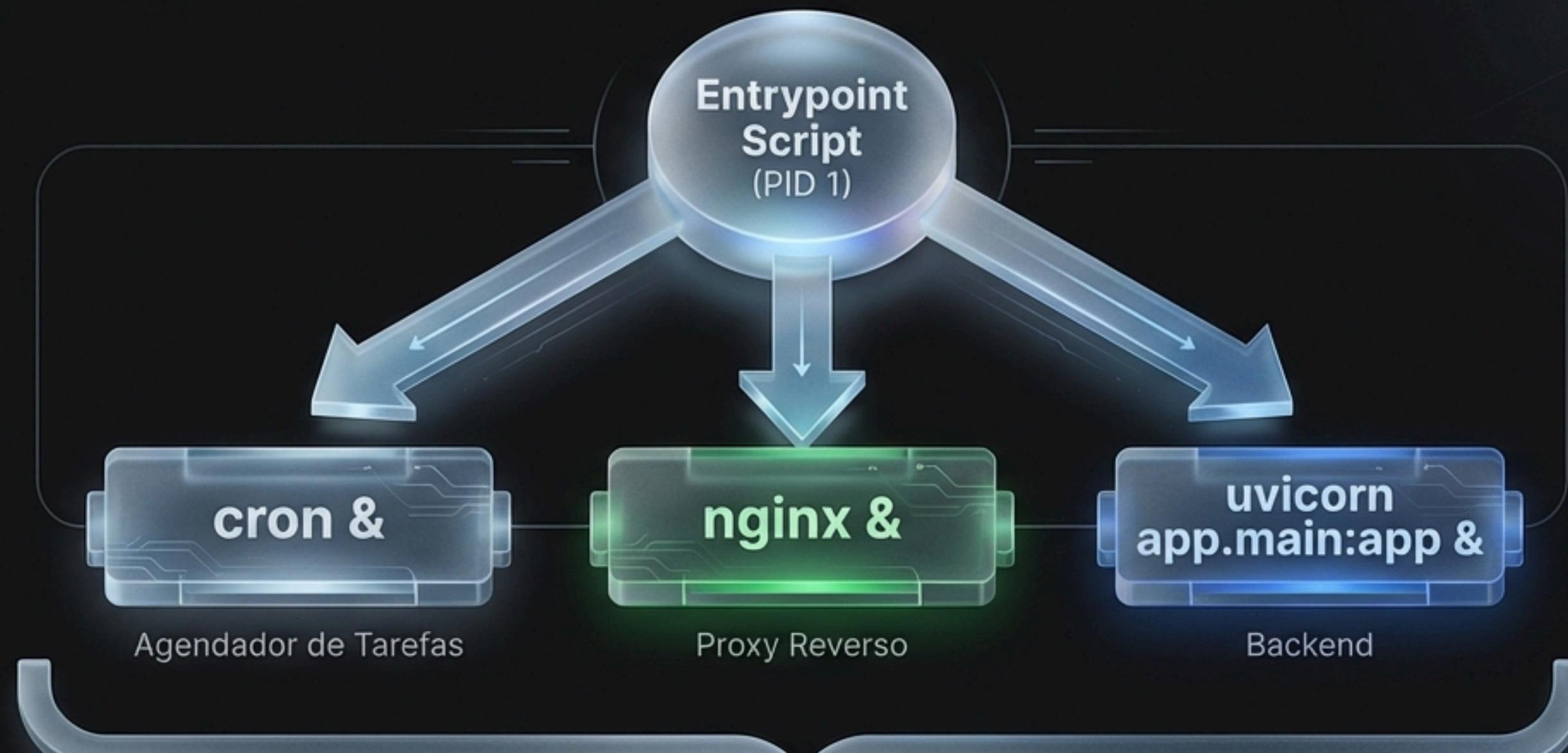
Correção de EOL (Windows CRLF -> Linux LF)

```
RUN sed -i 's/\r$//' /entrypoint.sh
```

Instalação Nginx + Cron + Requirements

Robustez Cross-Platform: Prevenção de erros de 'bad interpreter' entre ambientes Windows e Linux.

# O Maestro dos Processos: Entrypoint Script



**⚠ Fail-Safe Mechanism**

Se o Backend falhar, o container reinicia automaticamente.

# Inter Performance e Validação no Backend



**FastAPI**



## Alta Performance

Framework ASGI moderno, escolhido pela velocidade de execução.



## Segurança de Dados

Validação automática de tipos e schemas via Pydantic.



## Interatividade

Documentação automática (Swagger UI) e endpoints de gestão.

# O Paradoxo da Persistência: Stateless vs. Stateful

## Ambiente Local (Docker)



Persistência Garantida.  
O volume mapeia o disco  
do host.

## Ambiente Cloud Run



Stateless / Efêmero.  
O disco é volátil e se  
perde no reinício.

Conformidade com "Twelve-Factor App": Dados locais são tratados como temporários, exigindo backup externo.

# Observabilidade “White-Box”

## Health Check

● **ONLINE**

Latência: 45ms

</api/healthcheck>

## Metrics



</api/metrics>

## System Logs

[2024-01-27 10:15:02][INFO] Health check successful.

[2024-01-27 10:15:05][INFO] Metrics endpoint accessed.

[2024-01-27 10:15:08][DEBUG] Disk space check initiated...

</api/logs>

O sistema monitora a própria saúde, permitindo detecção proativa de falhas de disco ou rede.

# Pipeline de Entrega Contínua (CI/CD)



Automação total via Service Accounts com princípio de menor privilégio.

# Estratégias de Robustez e Resiliência



## Sanitização de EOL

Limpeza automática de scripts shell no build.



## Backup Automático

Cron Job para simulação de Disaster Recovery.



## Fail-Safe

Verificação de integridade antes de operações de arquivo.

# Maturidade Técnica Alcançada



**Custo Otimizado:** Modelo Pay-per-use (Scale-to-Zero).



**Segurança:** HTTPS automático e gestão de segredos.



**Portabilidade:** Compatibilidade total Windows/Linux.



**Velocidade:** Cache de Build e resposta rápida (ASGI).

# Teste a Elasticidade em Tempo Real



**<https://portfolio-docker-cloudrun-728889819893.us-central1.run.app>**

Acesse para verificar o 'Cold Start' e a performance  
do container híbrido na prática.