

Plataforma de Comércio

uma plataforma para todos

Oswaldo Santana Neto

Plataforma Aberta

Democratizando a tecnologia

- Serviços Federados
- Protocolos abertos
- Sistemas distribuídos
- Software **Livre**
- Governança Pública

Camadas

web & mobile applications, 3rd-party solution providers

orders, shipment, fulfillment, payment, catalog, listings, channels, wms

api gateway, hosting, deployment, configuration, environments, logging, messaging

processing, storage, network, security

responsabilidades

SaaS

Varejo

Experiência

Core

shopify

aws api gateway, lambda, sam, cloudwatch, sns/sqs/eventbridge

aws ec2, vpc, s3, iam, rds/aurora, dynamo

benchmarks

Camadas

web & mobile applications, 3rd-party solution providers

orders, shipment, fulfillment, payment, catalog, listings, channels, wms

api gateway, hosting, deployment, configuration, environments, logging, messaging

processing, storage, network, security

responsabilidades

SaaS

Varejo

Experiência

Core

open platform

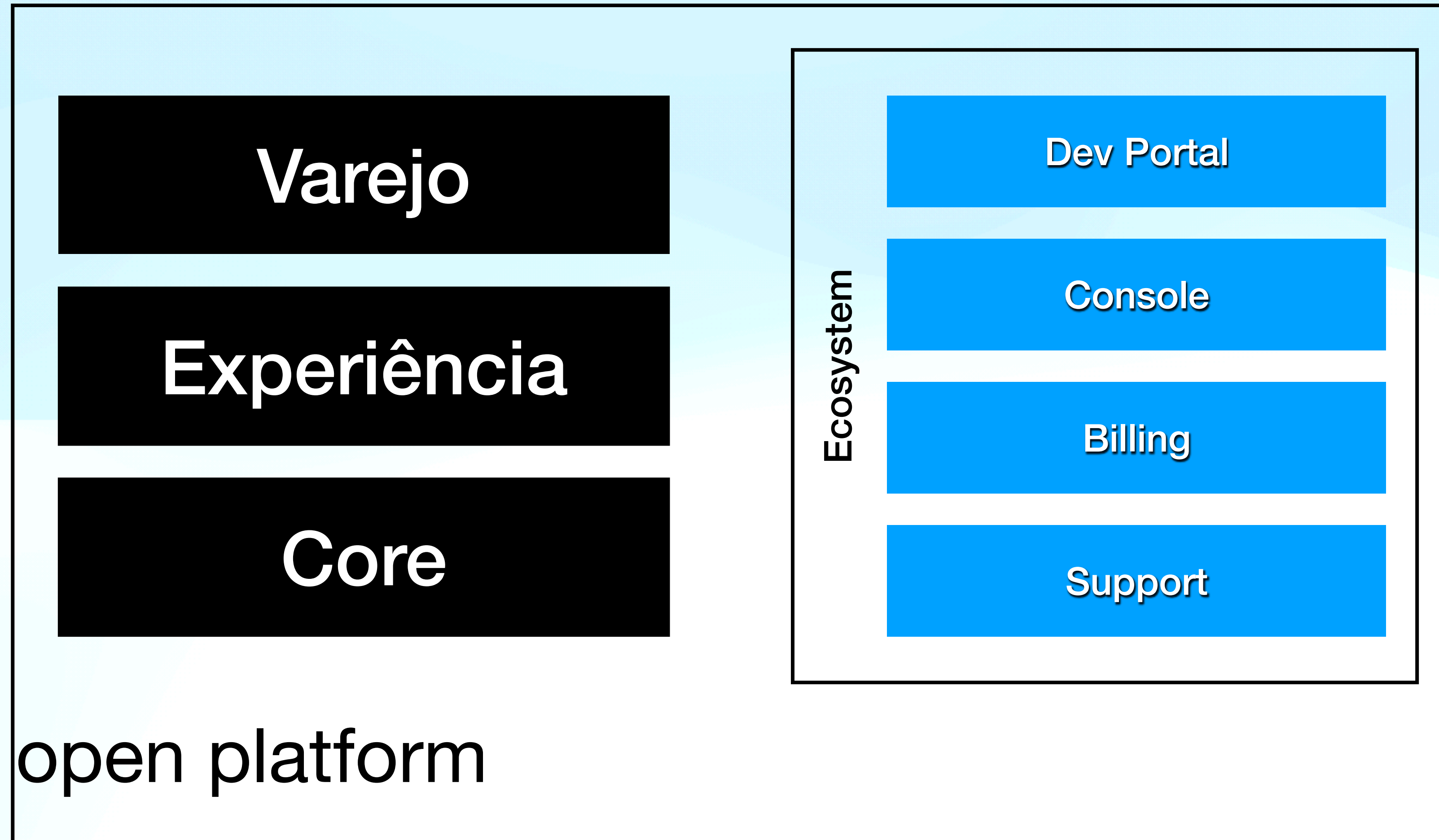
shopify

aws api gateway, lambda, sam, cloudwatch, sns/sqs/eventbridge

aws ec2, vpc, s3, iam, rds/aurora, dynamo

benchmarks

Plataforma Aberta



Core

fundações e conectividade

Infra

- Computação
- Rede
- Armazenamento
 - Armazenamento de Blob
 - Bancos de Dados
- Confiabilidade

Gestão de Identidade e Acesso

- Authentication & authorization
- Profiling for user and organizations
- Federation to enable other companies to use our Identity manager service

Experiência

blocos de construção

Diretrizes

- Diretrizes de Especificações e Desenvolvimento
- Modelo de Maturidade de Tecnologia
- Serviços e Ferramentas básicas para construção de APIs e Serviços para a plataforma

Hospedagem & Deployment

- Automated deployment system
 - Transparent deployment in serverless environments (eg. AWS Lambda)
 - API Management (eg. AWS Gateway)
- Event/Messaging system
 - Publish/Subscriber system, similar to SNS/SQS, integrated with identity management system
 - Support for system integrations through webhooks and queueing (eg. SQS)
 - Support for user communication through e-mail, sms, WhatsApp/telegram, etc
 - Support for user interaction through push notifications

Hospedagem & Deployment

- API Gateway
 - Transparent and automatic request routing to function-as-a-service
- Function as a Service (FaaS)
 - Respond to API Gateway requests
 - Consumes messages in queues
 - Integration with Storages, Monitoring & Logging
- Automated deployment system
 - Transparent deployment in serverless environments (eg. AWS Lambda with AWS SAM)
 - Support for Configuration and Environments

Eventos & Mensageria

- Publish/Subscriber system, similar to SNS, integrated with identity management system
- Subscription types:
 - Queue: (similar to SQS) with support for filtering messages on topic subscription
 - Push Notification: Webhook, Websocket and mobile notifications
 - Messaging: e-mail, SMS, WhatsApp/Telegram, etc
 - 3rd-party providers: Zapier, IFTTT, AWS Step Functions, etc
- Feed API with the latest messages (RSS/Atom Feed-like API)

Automação de Negócios

- Enable developers to create fast integrations with our new platform (AWS Step Functions, Zapier, IFTTT)
- Enable product managers to validate hypothesis quickly

Ecossistema

ajudando desenvolvedores

Plataforma para Desenvolvedores

- Website with documentation, community forums, etc
- Developer panel
- Billing system
- 3rd-parties products marketplace

Varejo

serviços para o varejo

Serviços para Varejo

- Platform Services (provided by UX Team)
- Catalog and Publication Services
- Fulfillment and Logistics Services
- Competitiveness Services
- Customer Relationship Services
- Financial and Tax Services

Catálogo & Publicação

viabilizando vendas



Catálogo & Publicação

Portfolio

Catalog

Distribution

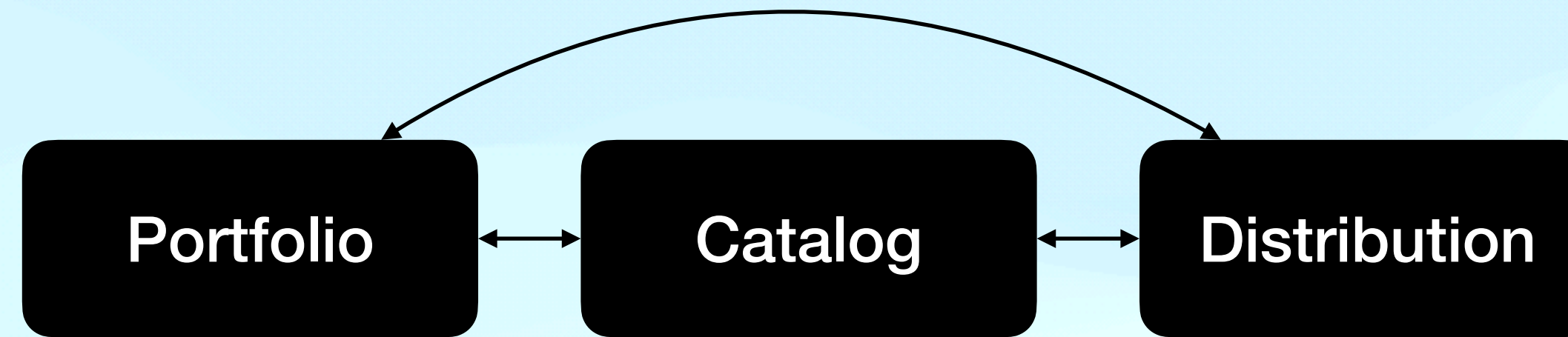
Catálogo & Publicação



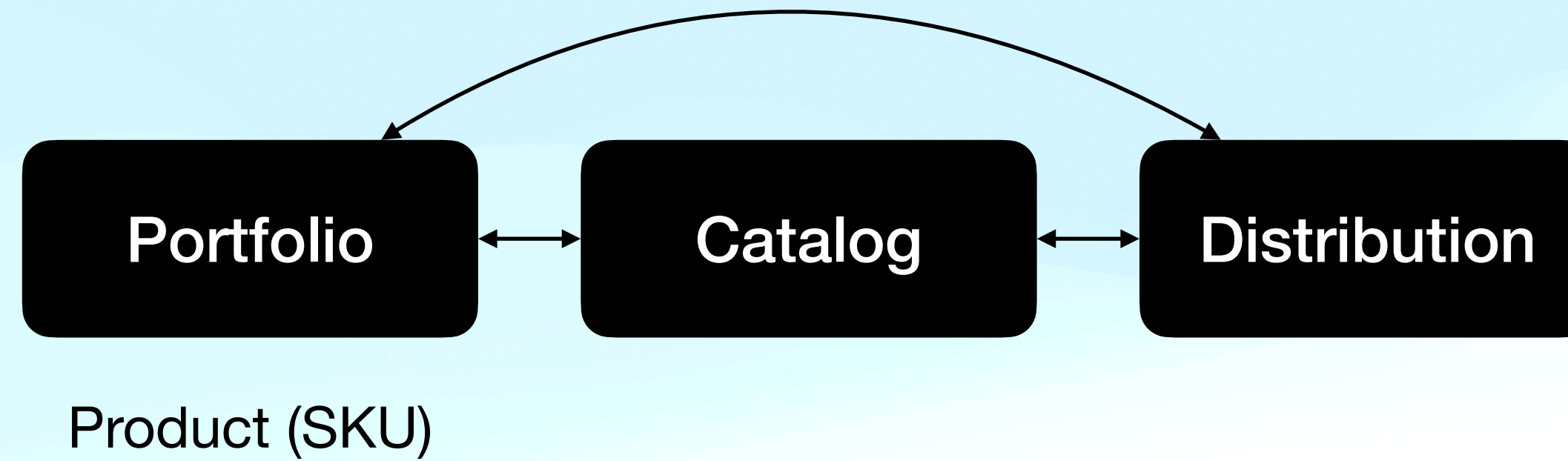
Catálogo & Publicação



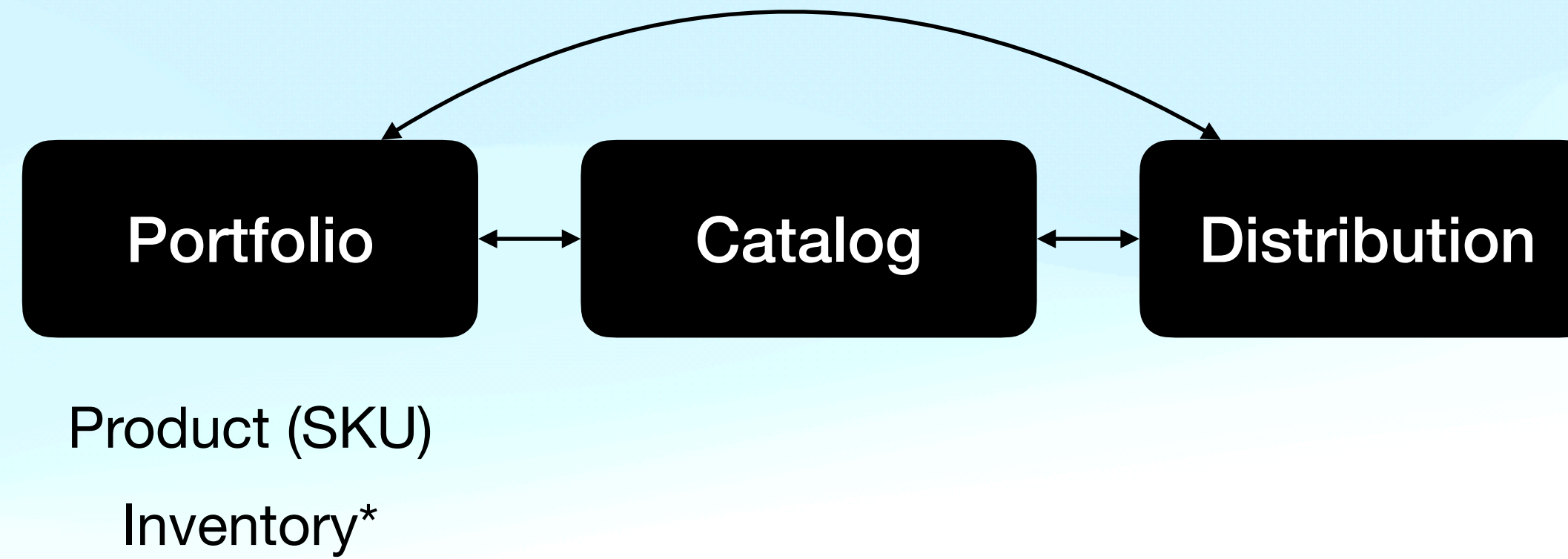
Catálogo & Publicação



Catálogo & Publicação

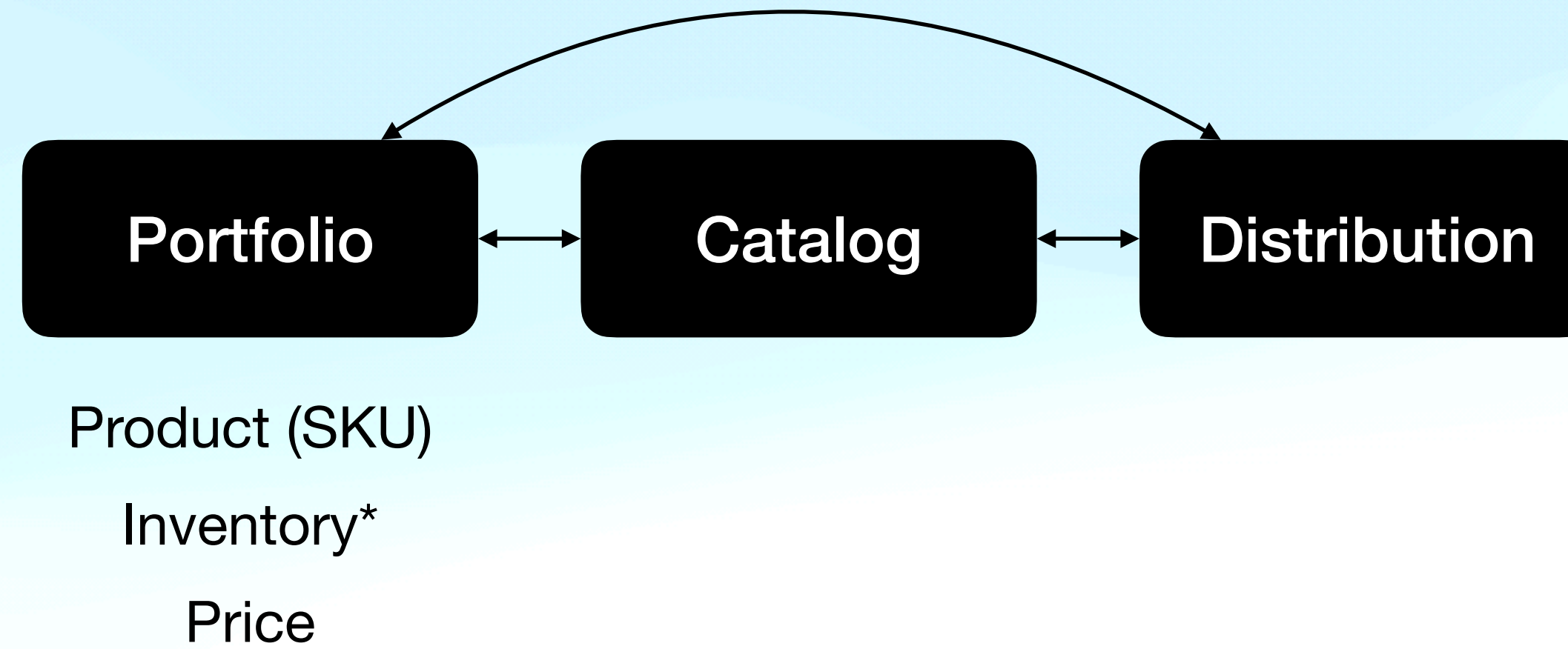


Catálogo & Publicação



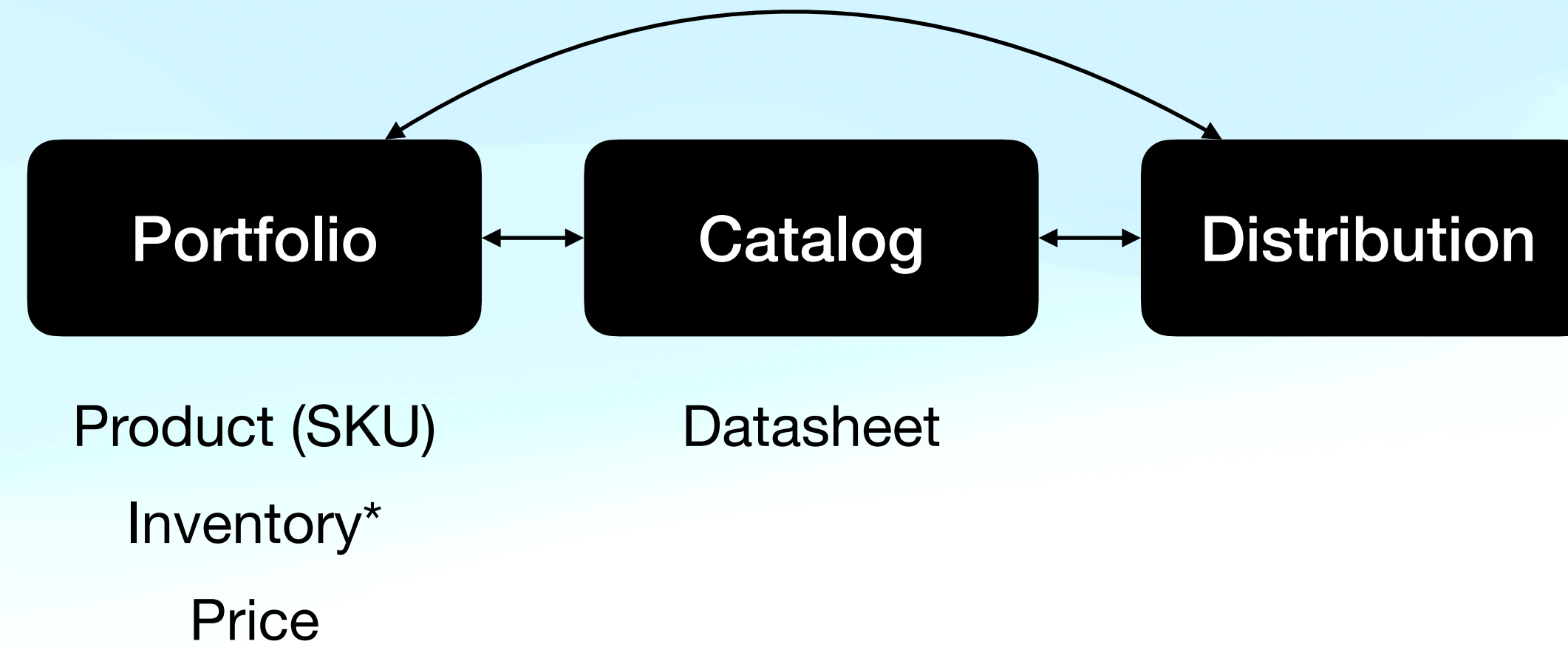
* Local non-transactional inventory report

Catálogo & Publicação



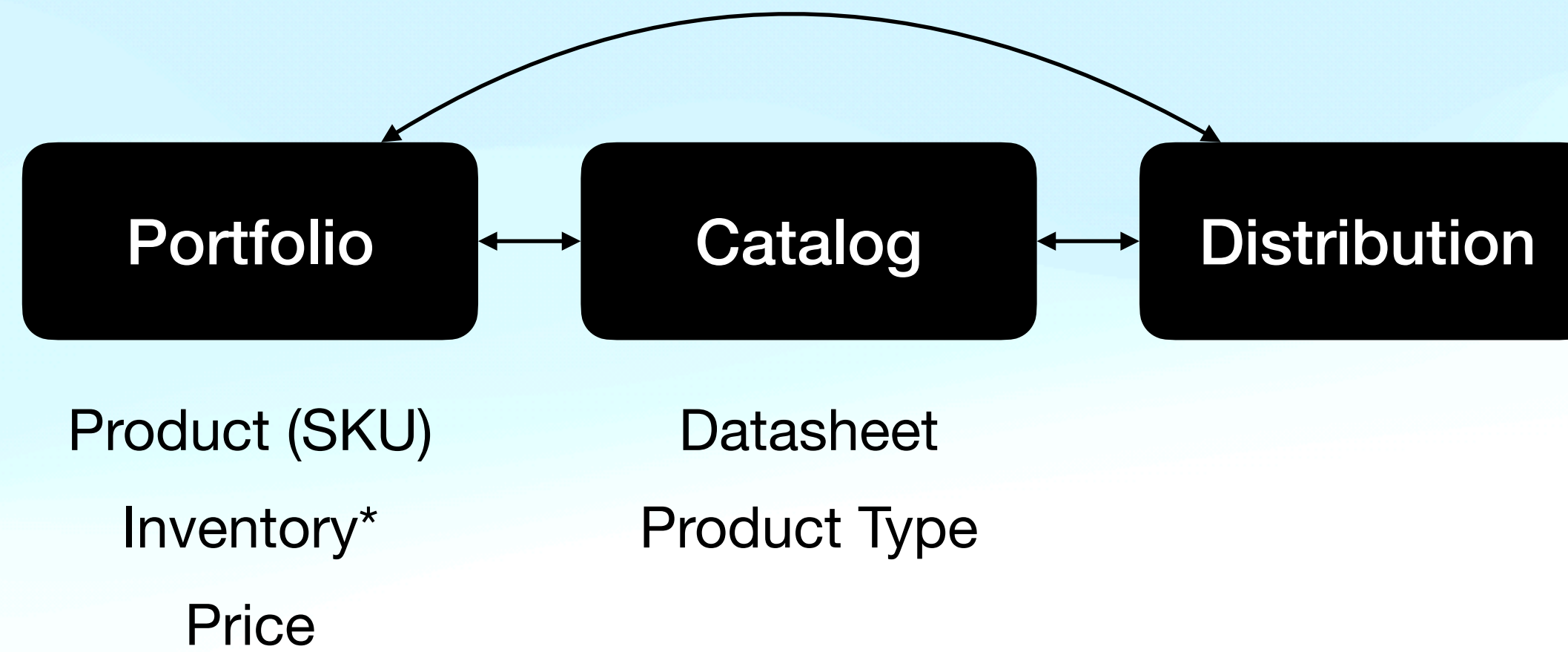
* Local non-transactional inventory report

Catálogo & Publicação



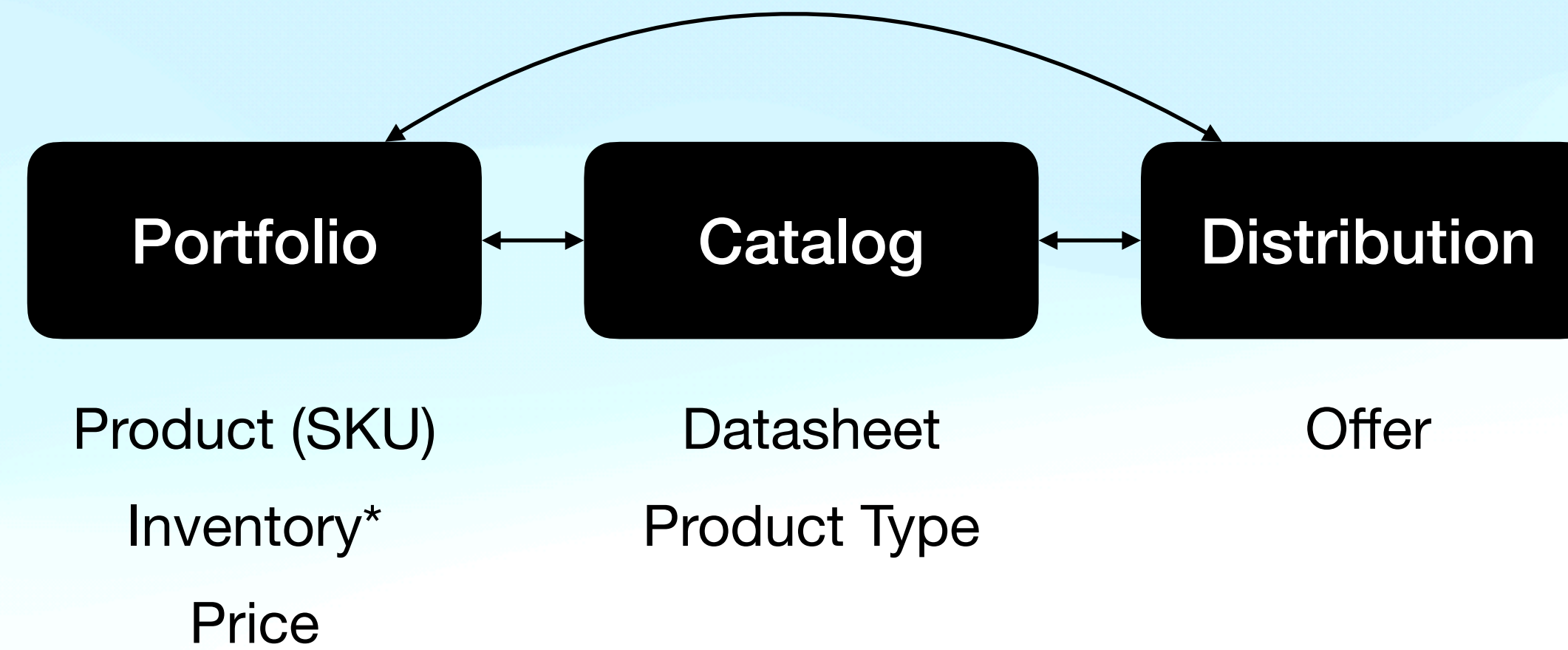
* Local non-transactional inventory report

Catálogo & Publicação



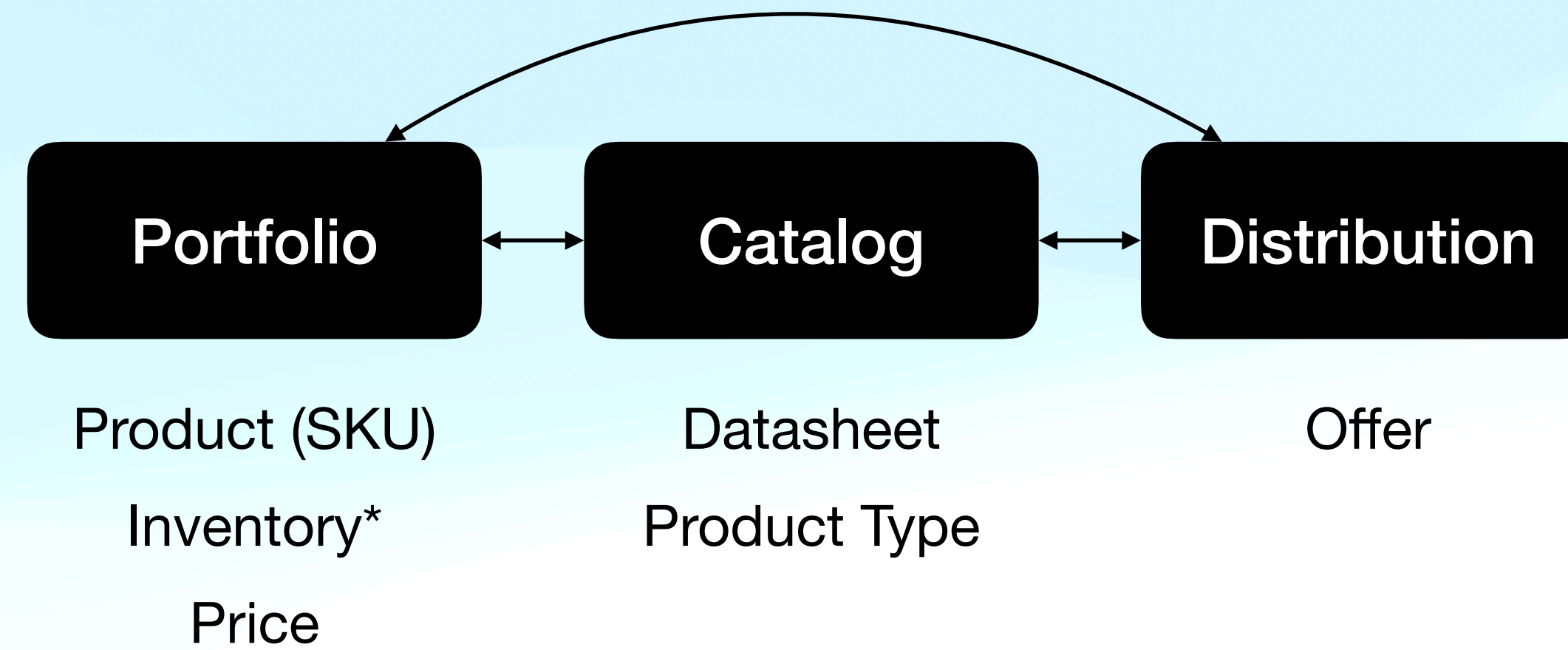
* Local non-transactional inventory report

Catálogo & Publicação



* Local non-transactional inventory report

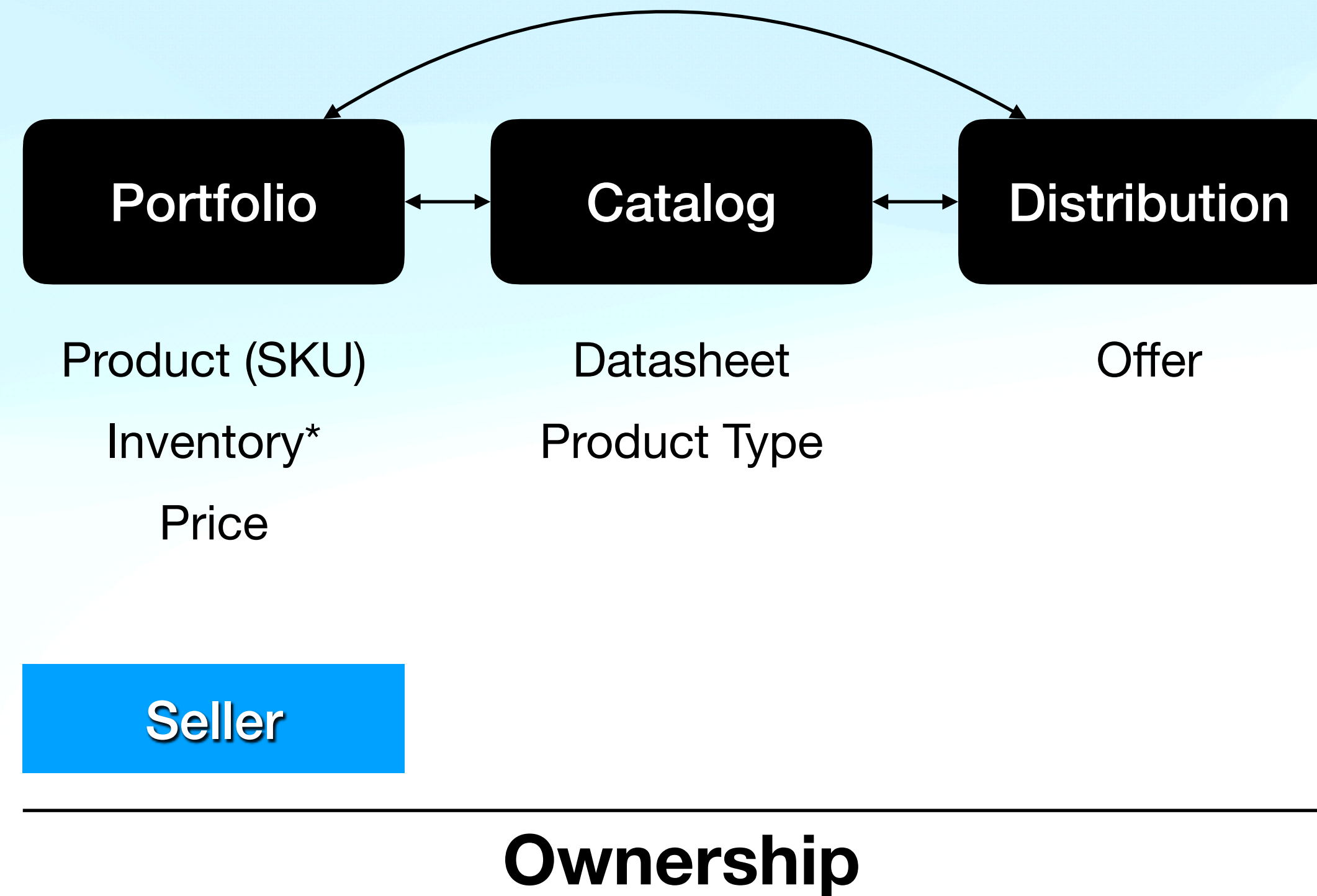
Catálogo & Publicação



Ownership

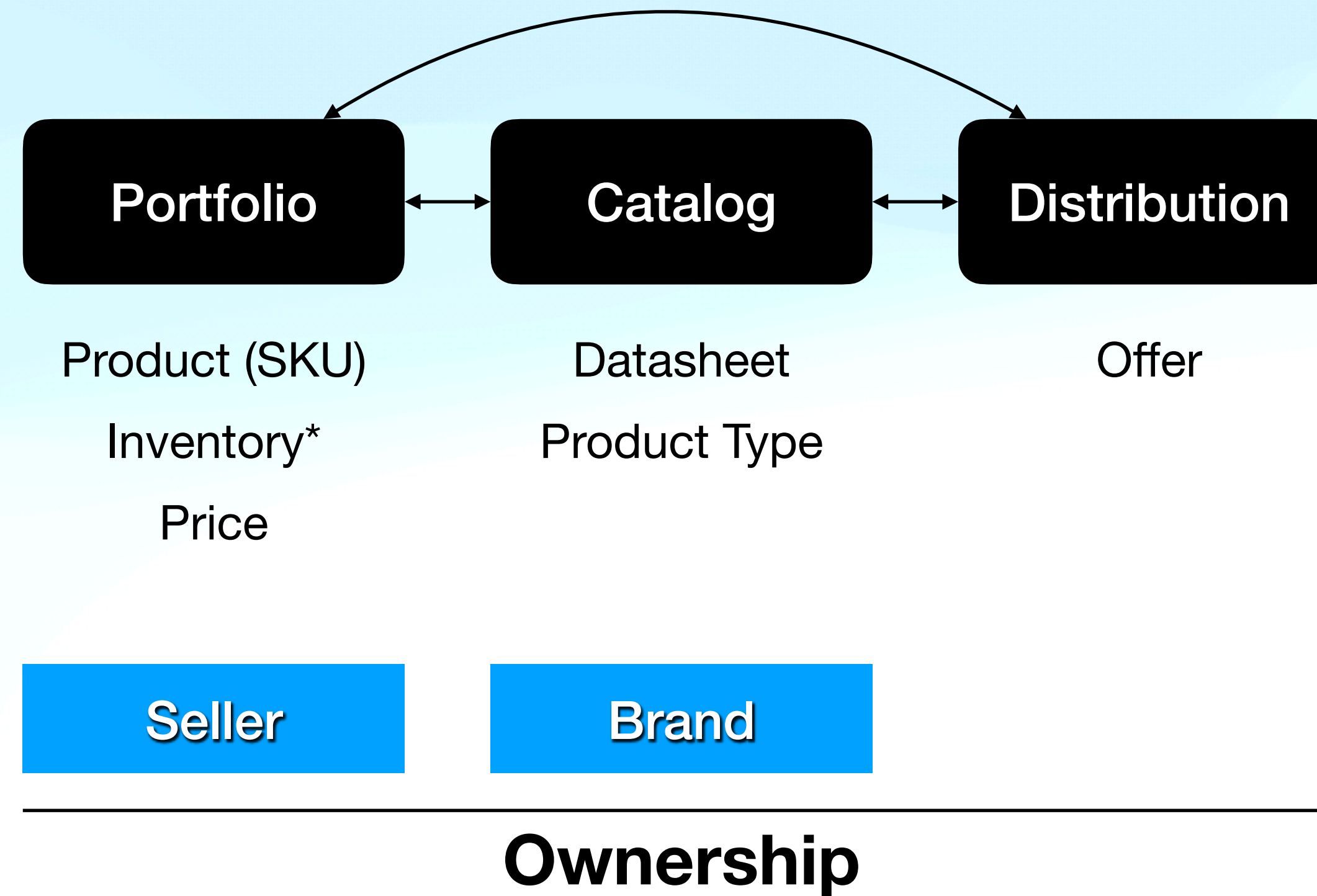
* Local non-transactional inventory report

Catálogo & Publicação



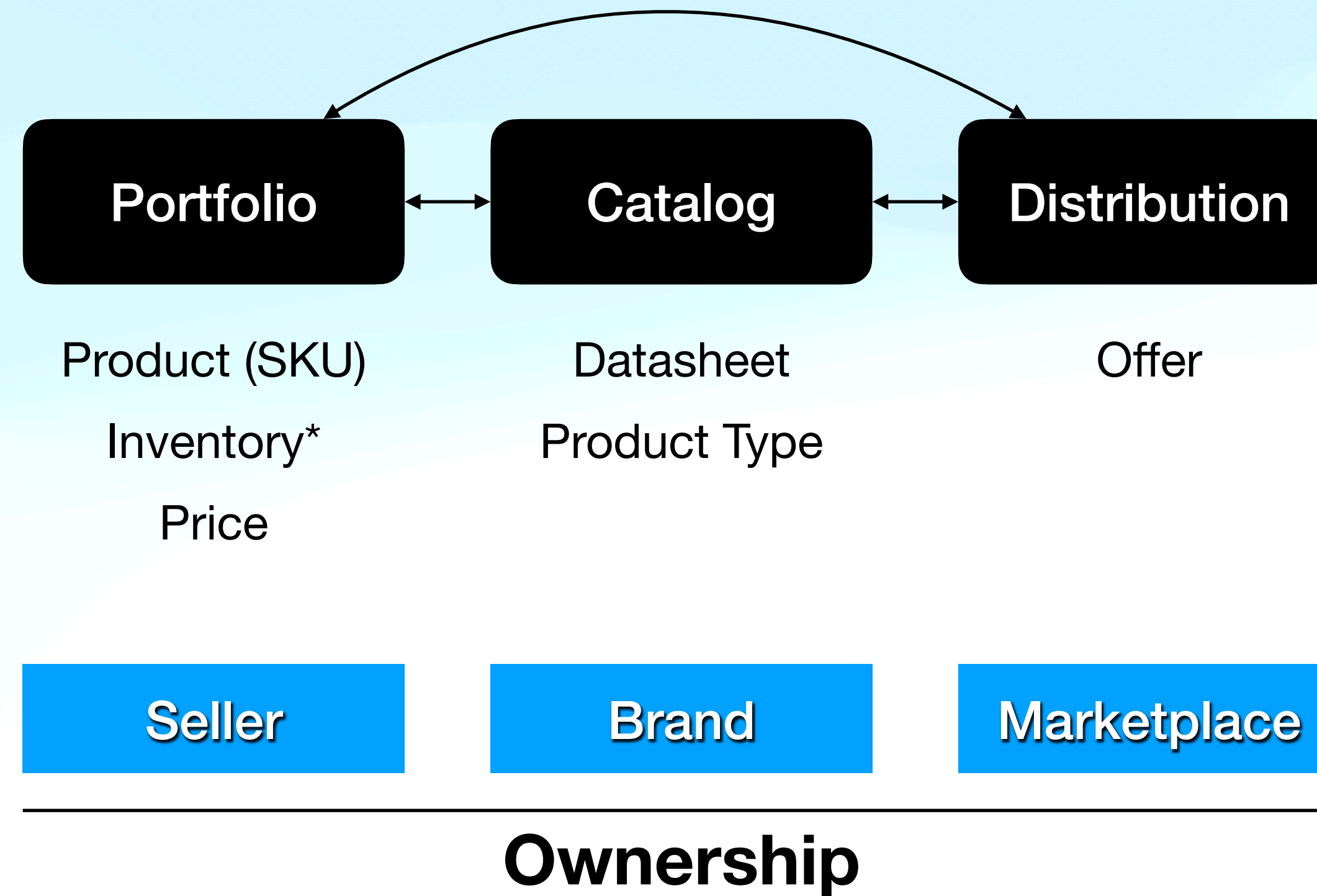
* Local non-transactional inventory report

Catálogo & Publicação



* Local non-transactional inventory report

Catálogo & Publicação



* Local non-transactional inventory report

Fulfillment & Logística

entregando com eficiência



Fulfillment & Logística

- Order Management (OMS)
- Order tracking
- Order bid and routing
- Invoice Management
- Invoice Generation
- Invoice Storage
- Invoice Validation

Fulfillment & Logística

- Warehouse Management (WMS)
- Shipping Service
 - Logistic operators management
 - Shipment bid and routing
 - Shipment Handling and Tracking
 - Reverse Logistics

Competitividade

vendendo mais

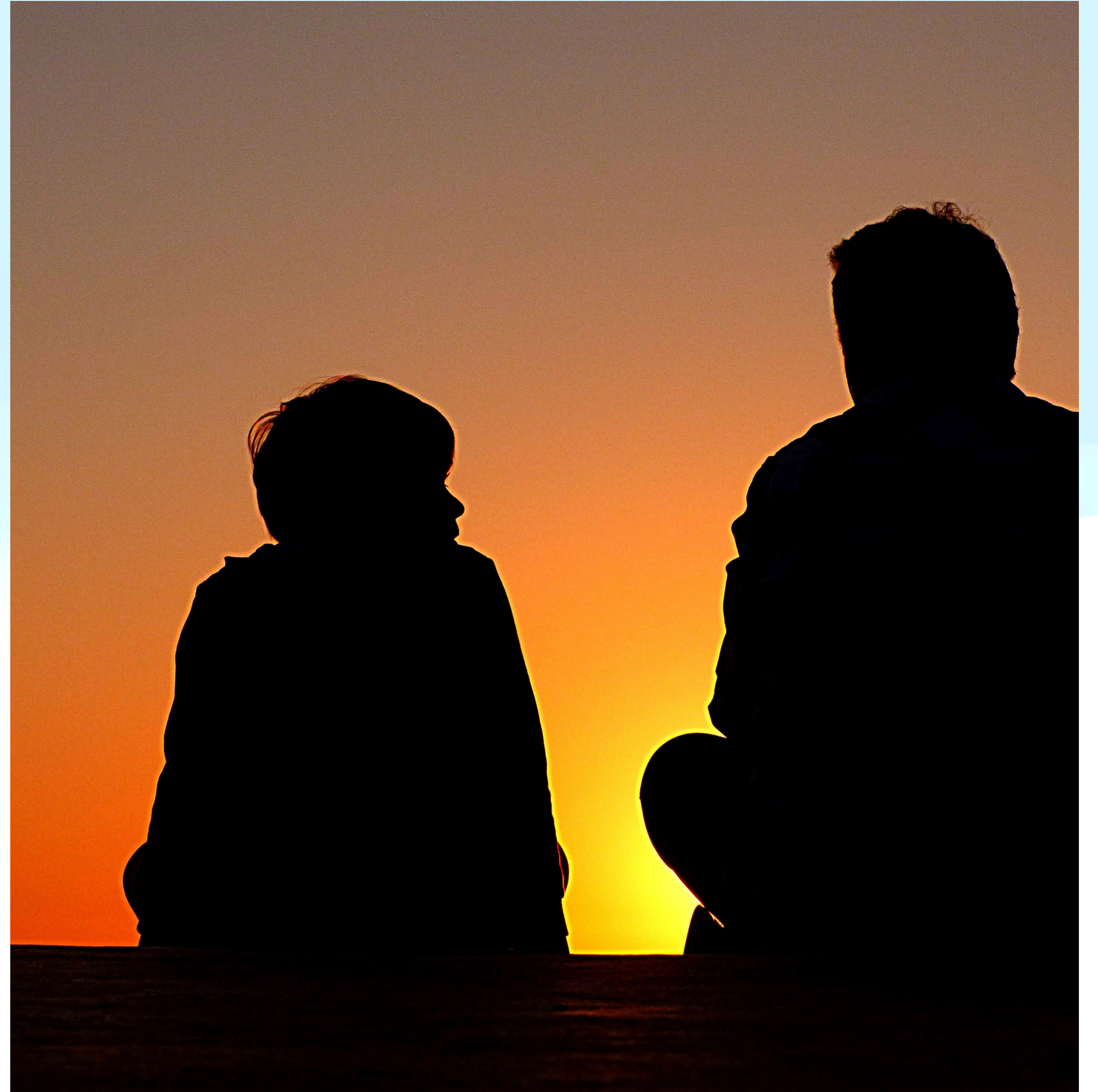


Competitividade

- Smart Pricing
 - Market monitoring
 - Pricing adjustments based on pre-defined rules
- Shipment optimization
 - Reduced freight prices with order routing
- Tools for analytics

Relacionamento com Clientes

falando com eles

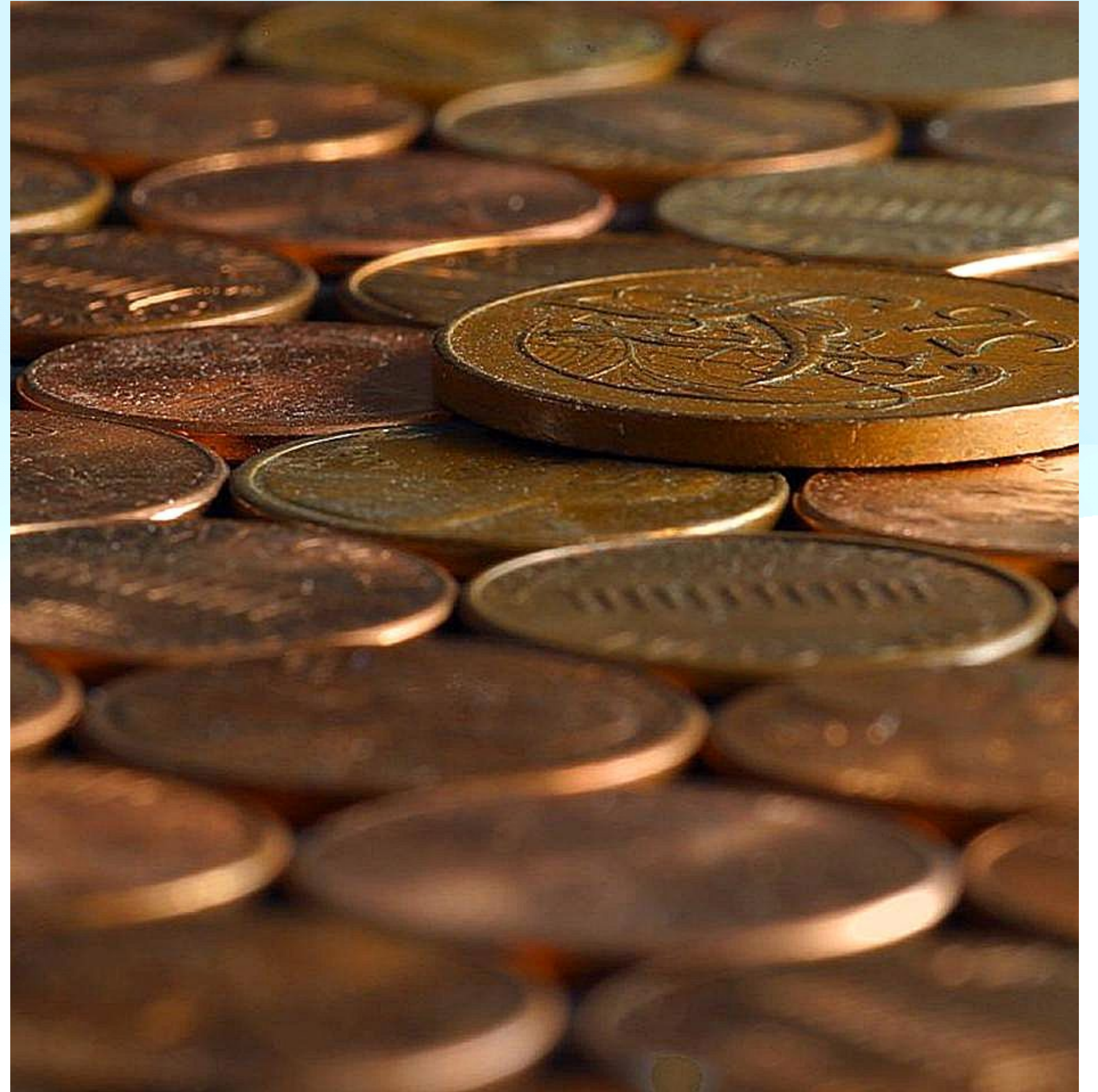


Relacionamento com Consumidores

- Pre-sale help-desk for buyers
 - Q&A for products
- Help-desk support for post-sale
- Product Quality Assurance
 - Recommendation analysis to deal with low-quality products or falsifications

Financeiro & Impostos

entregando valor



Financeiro & Impostos

- Payment management
- Subscription management
- Loan and payment anticipation
- Invoice Generation & Management

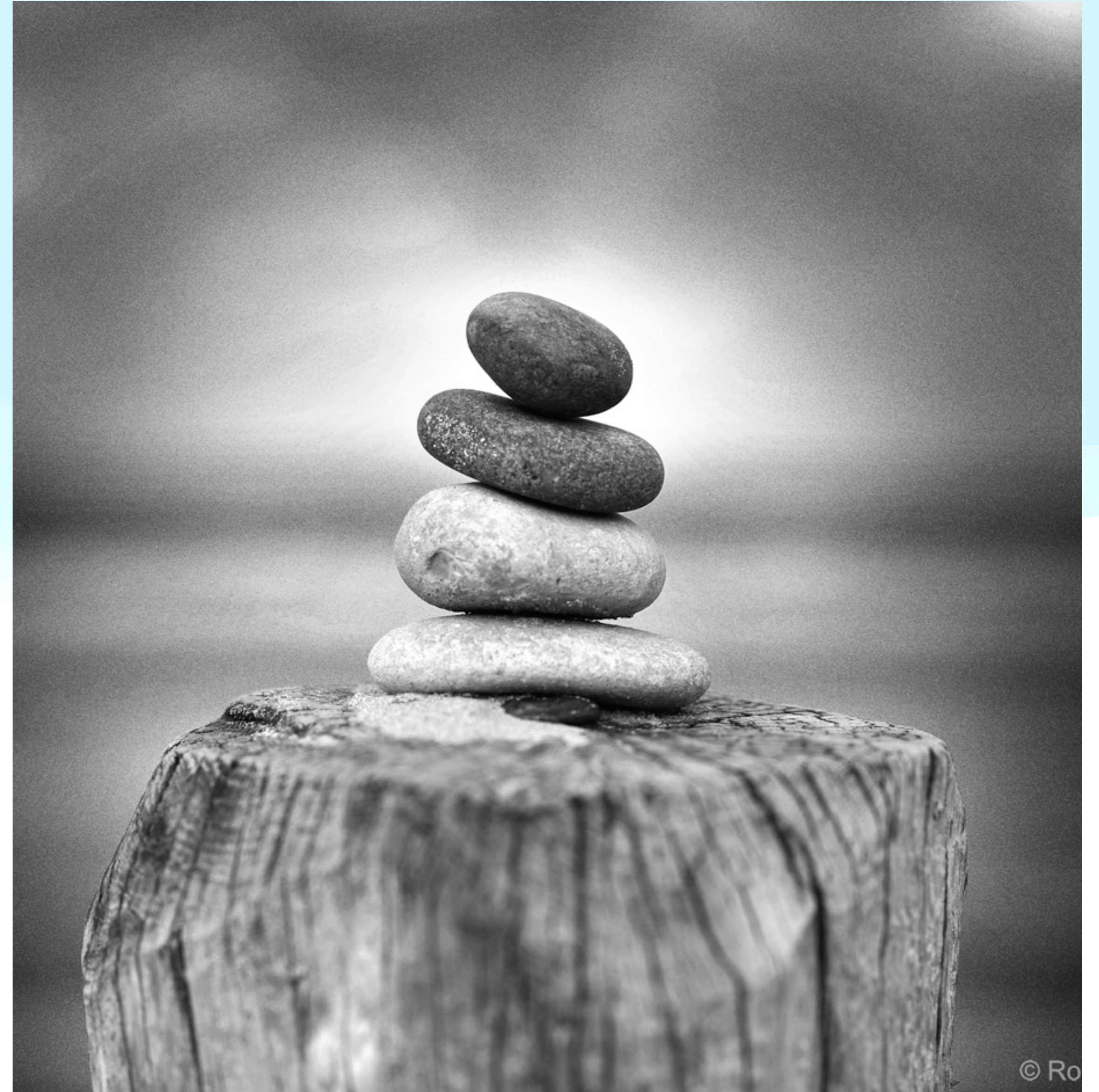
Implementação

construindo uma plataforma

Requisitos

o que faz uma boa plataforma

Simplicidade
Facilidade de uso e
interfaces uniformes



Simplicidade

- KISS - Keep It Simple Stupid
- Navalha de Occam - Princípio da parcimônia
- Menos é mais

Robustês

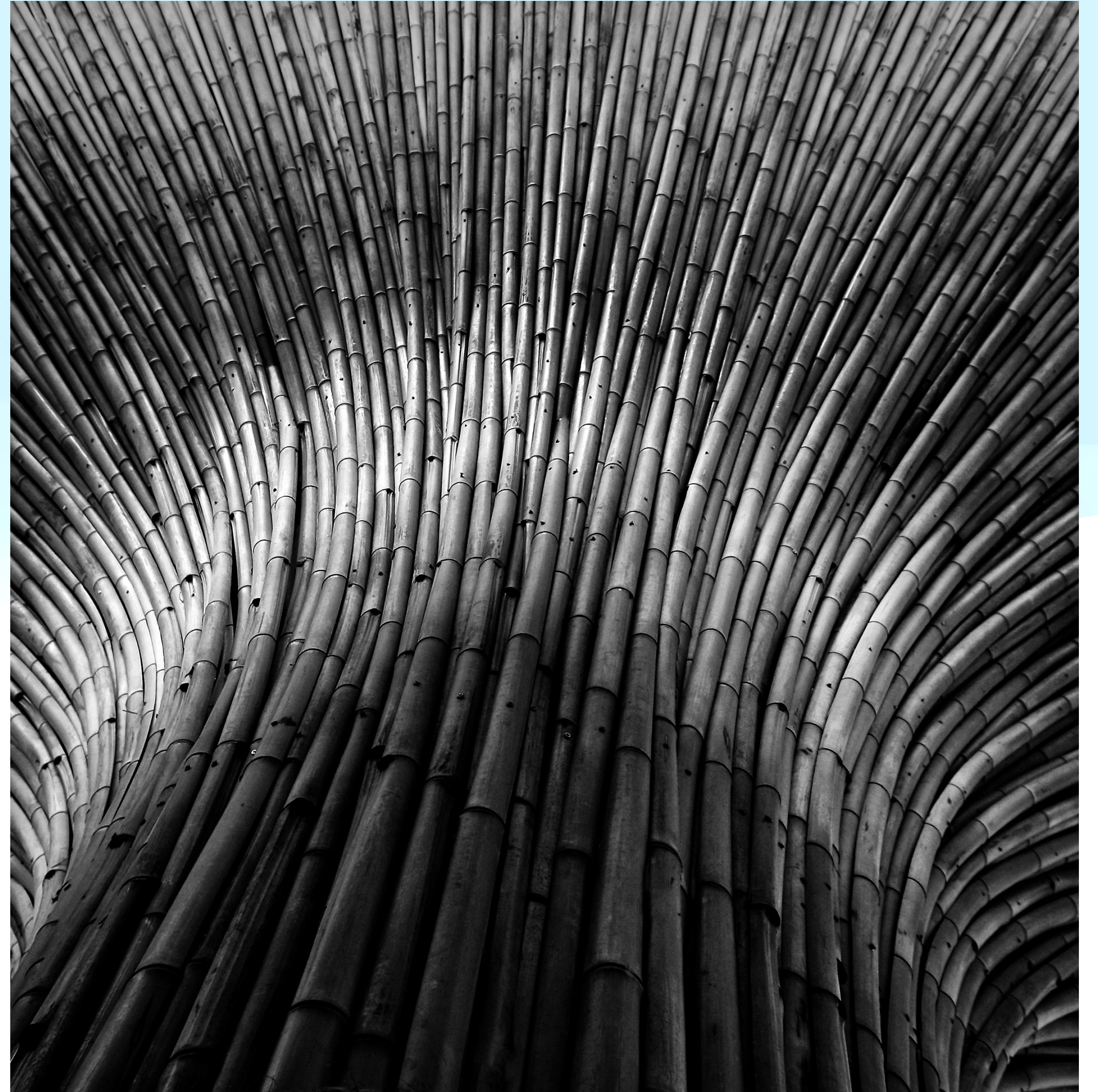
Escalabilidade e resiliência



Premissas

- Não importa se um **sistema** é interno ou externo, ele eventualmente...
 - ... sairá do ar...
 - ... quebrará...
 - ... ou vai alterar seu comportamento inesperadamente
- Lei de Postel
 - Seja conservador com o que você faz e seja liberal com o que você recebe dos outros
 - Também conhecido como princípio da robustês

Flexibilidade
Contemplar várias
necessidades

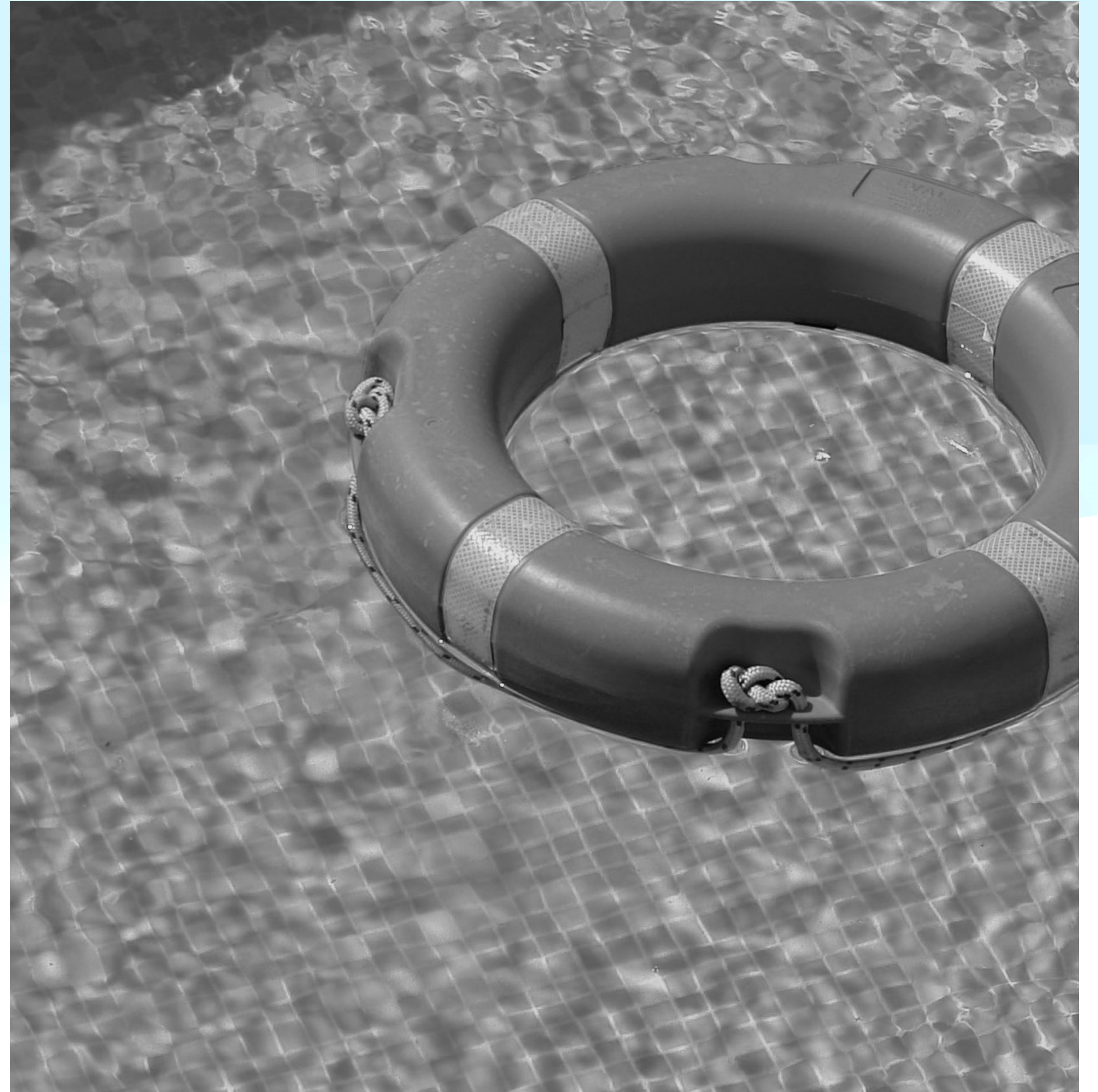


Flexibilidade

- Sistemas distribuídos
- Protocolos comuns
- Interfaces estáveis
- Regras de negócio customizáveis

Segurança

Segurança e confiabilidade



Segurança

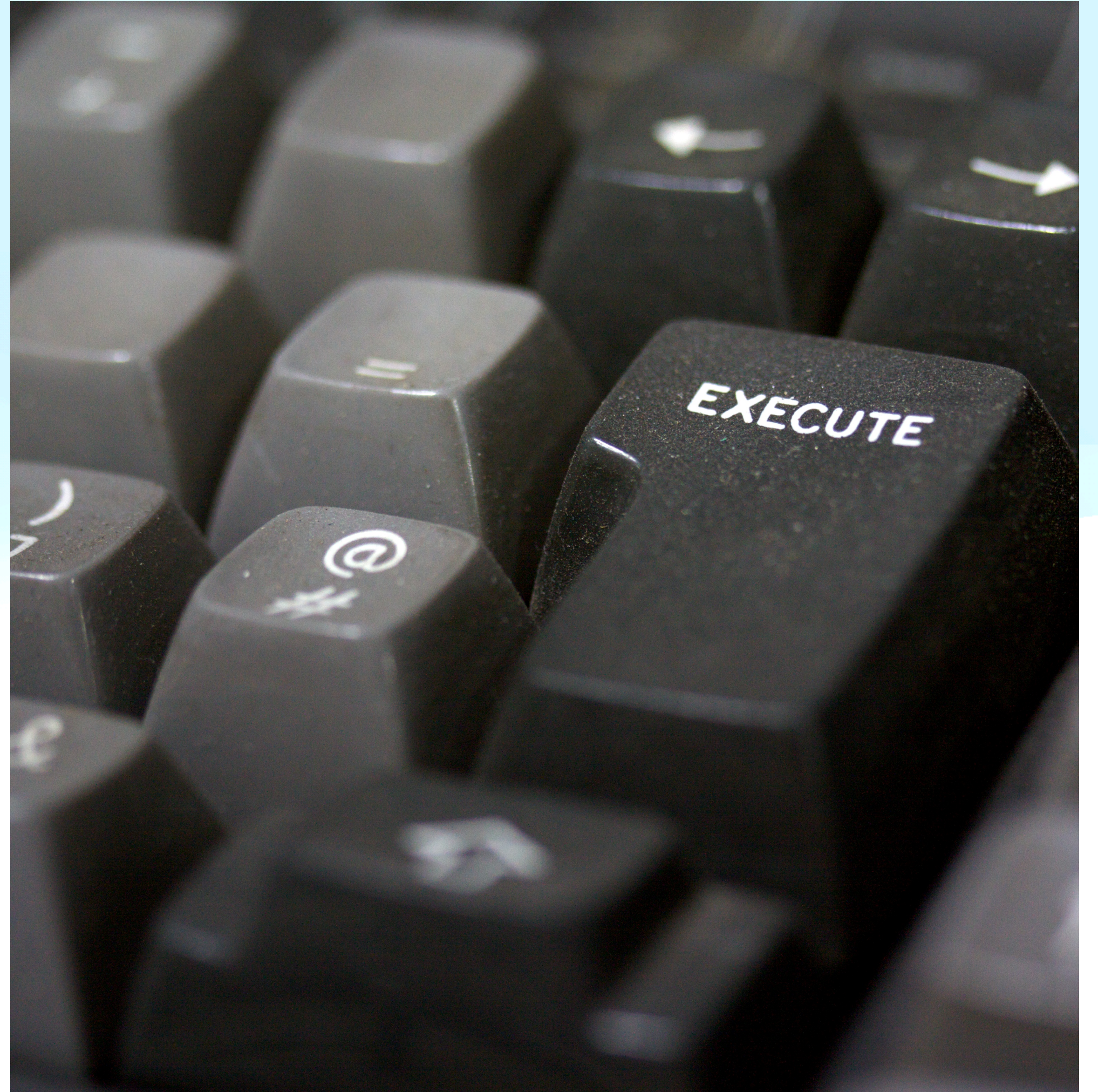
- ACID
- Redundância: Se você tem um você não tem nenhum
- Lei de Murphy

Arquitetura

eventos

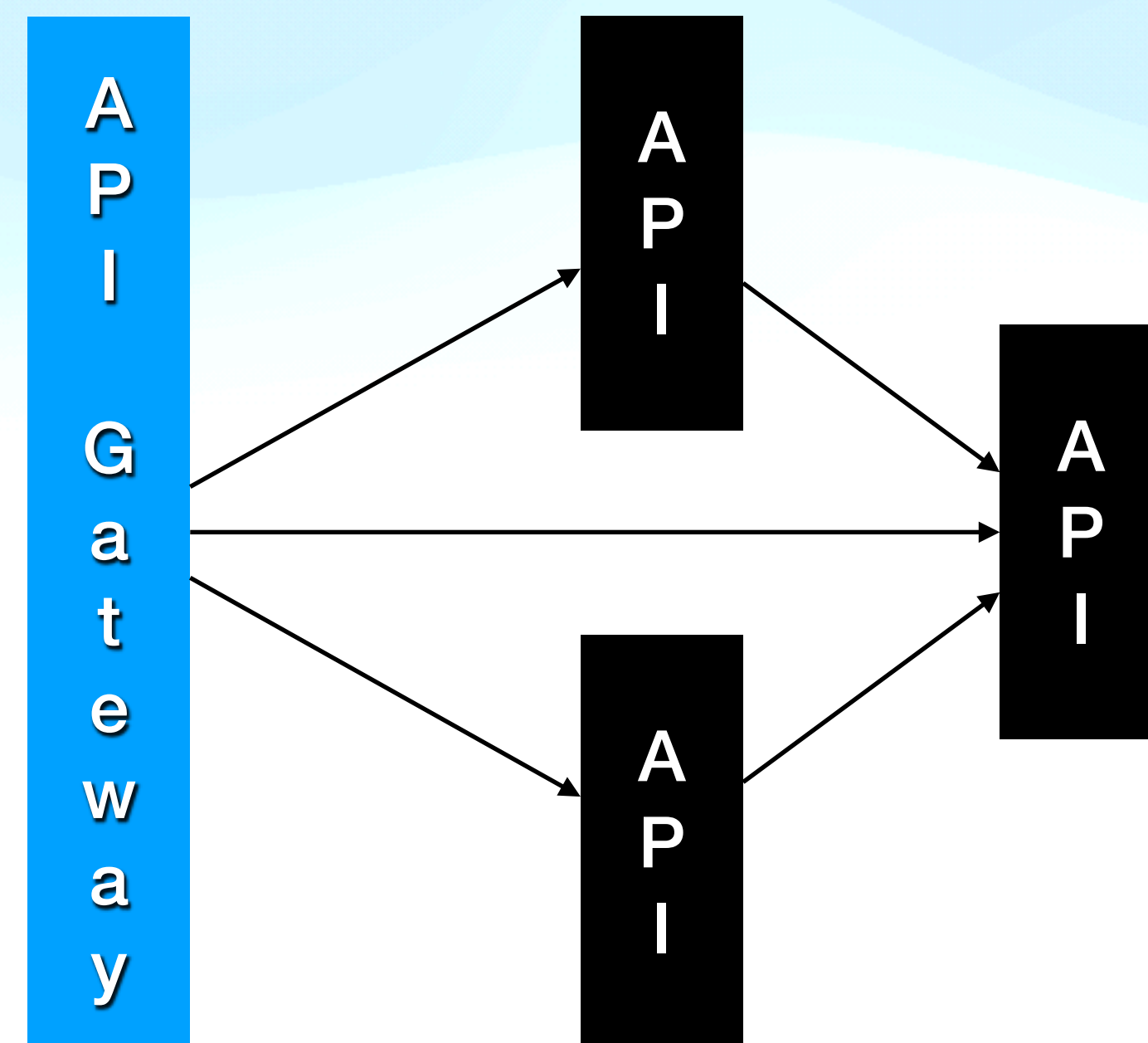
Comandos

"Faça"



Comandos Síncronos

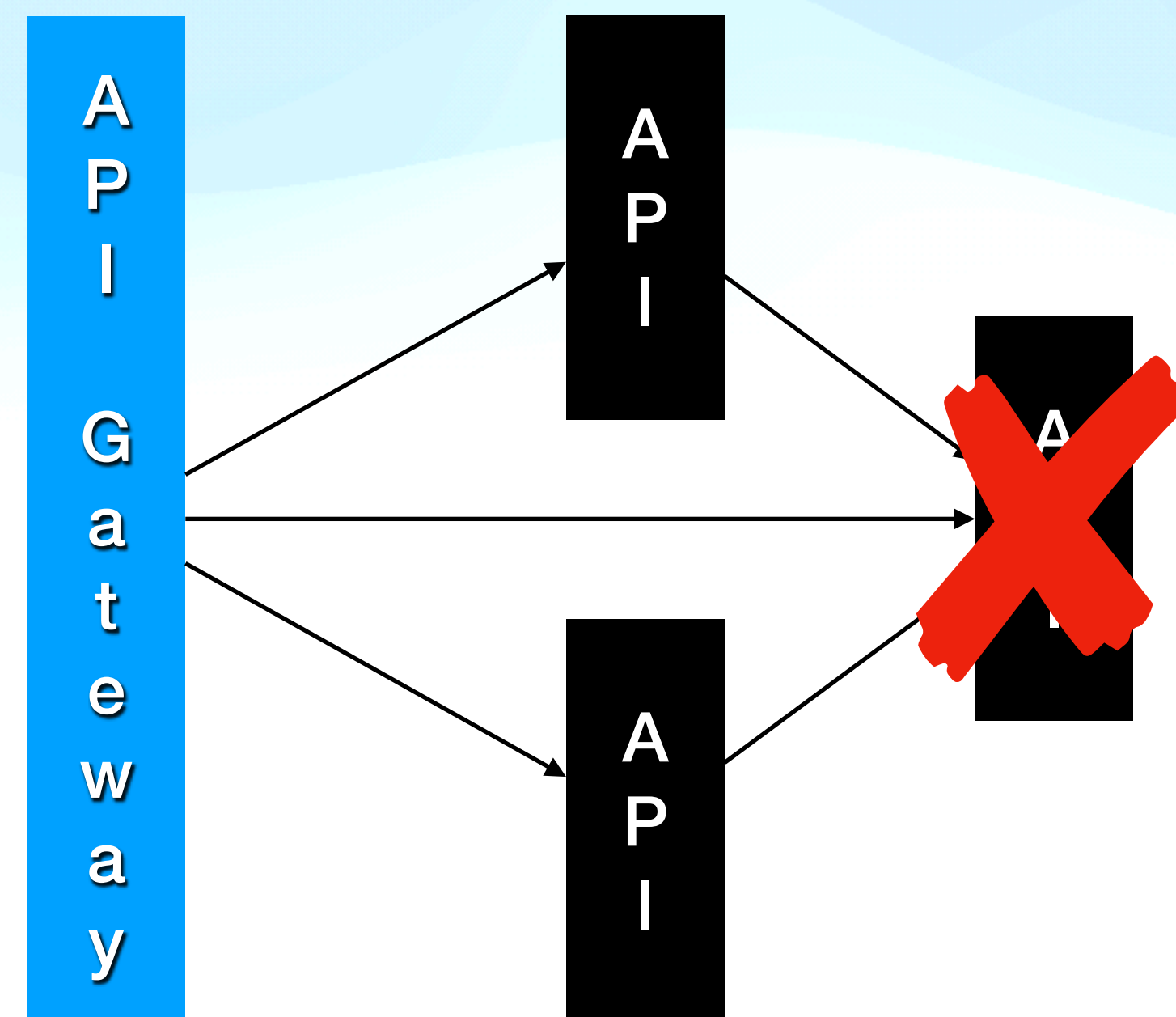
- Chamadas síncronas de procedimentos (RPC)
- Geralmente via requests HTTP ou RPC
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Falha em serviços causam indisponibilidade ou exigem a implementação de complexos sistemas de gestão de falhas (**baixa resiliência**)



HTTP Requests

Comandos Síncronos

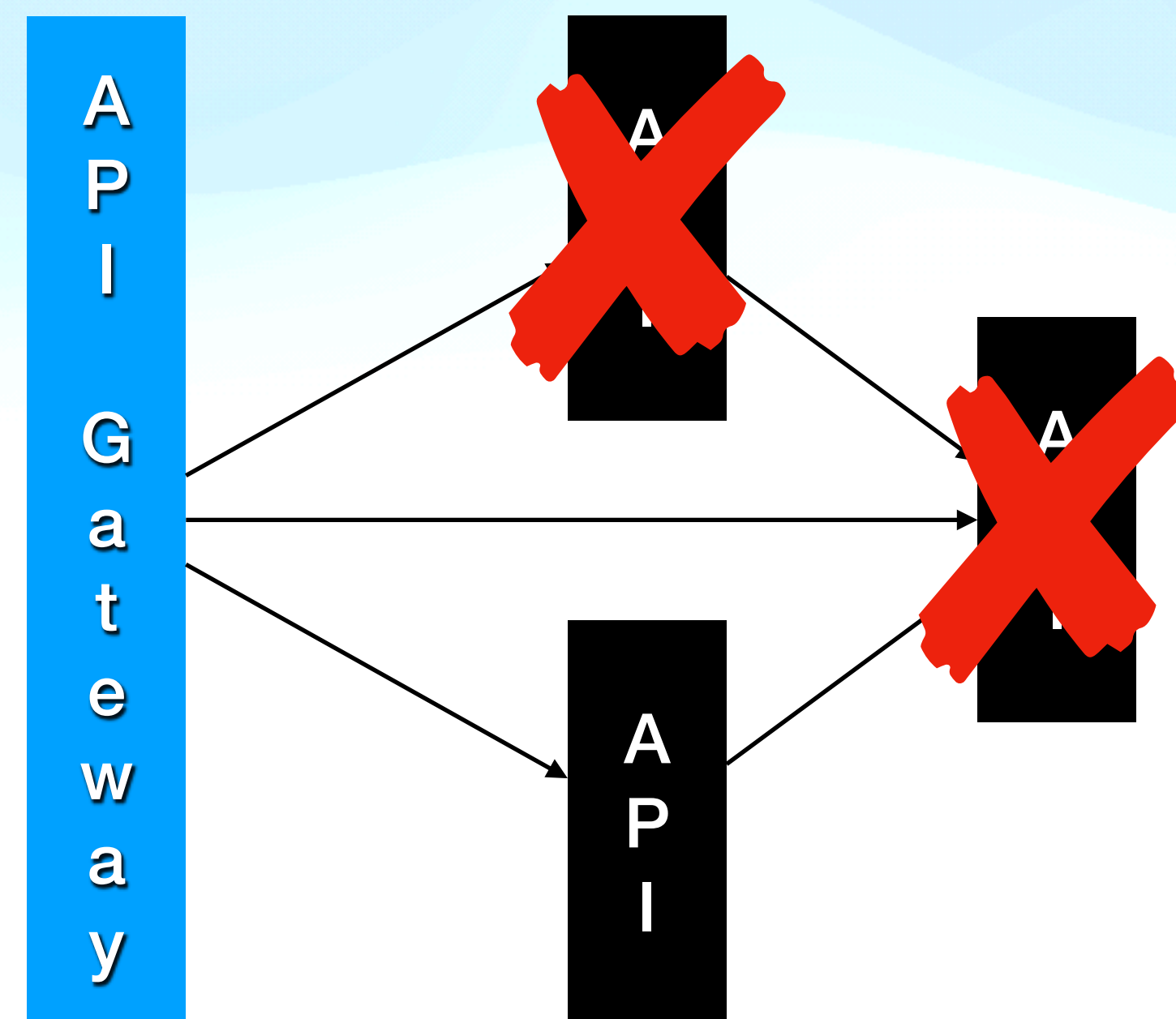
- Chamadas síncronas de procedimentos (RPC)
- Geralmente via requests HTTP ou RPC
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Falha em serviços causam indisponibilidade ou exigem a implementação de complexos sistemas de gestão de falhas (**baixa resiliência**)



HTTP Requests

Comandos Síncronos

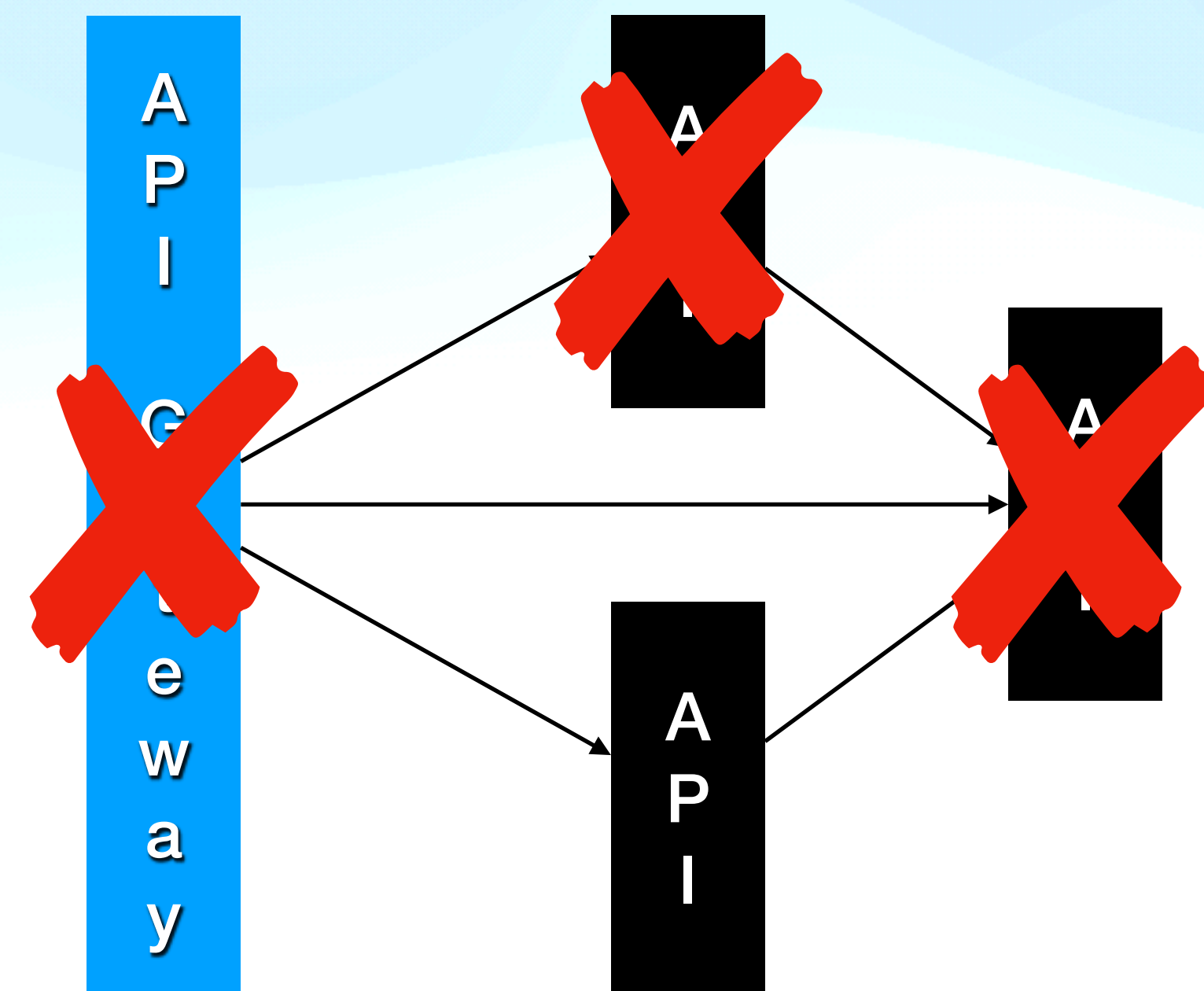
- Chamadas síncronas de procedimentos (RPC)
- Geralmente via requests HTTP ou RPC
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Falha em serviços causam indisponibilidade ou exigem a implementação de complexos sistemas de gestão de falhas (**baixa resiliência**)



HTTP Requests

Comandos Síncronos

- Chamadas síncronas de procedimentos (RPC)
- Geralmente via requests HTTP ou RPC
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Falha em serviços causam indisponibilidade ou exigem a implementação de complexos sistemas de gestão de falhas (**baixa resiliência**)



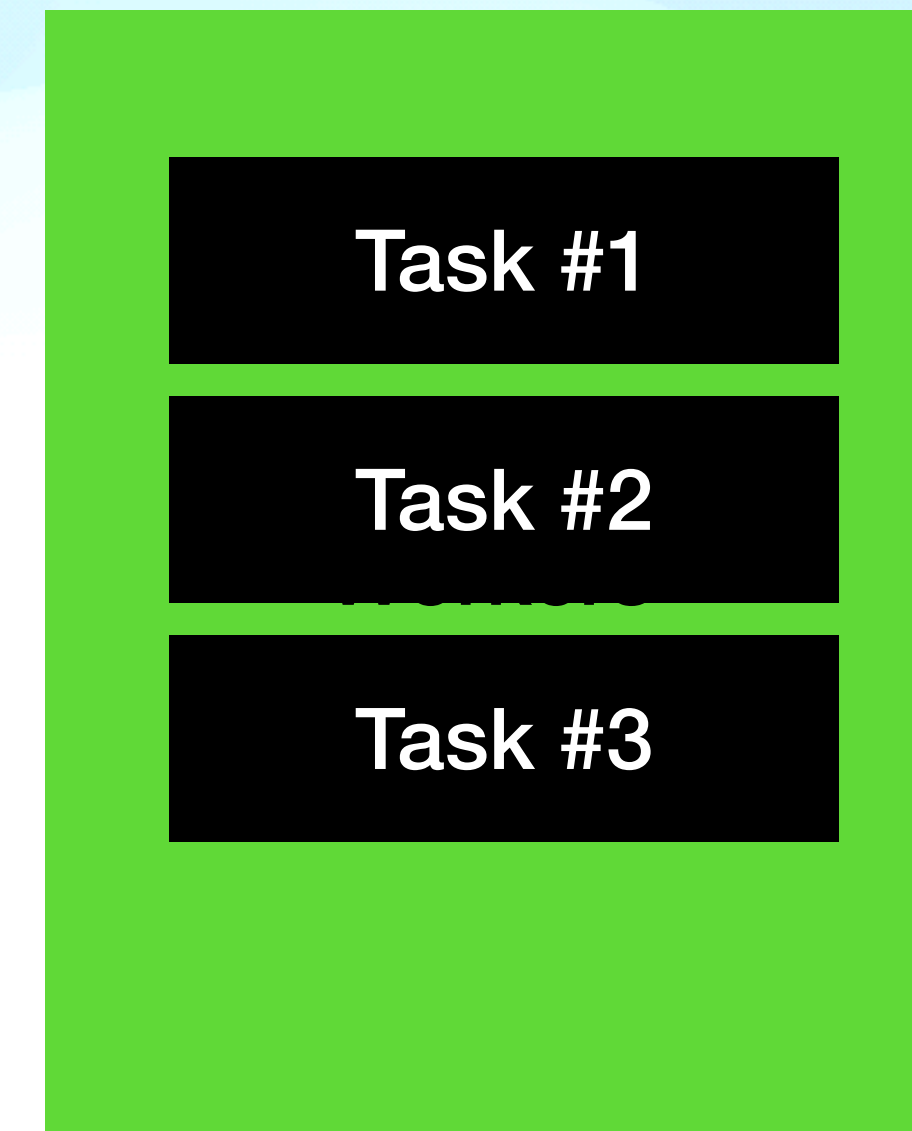
HTTP Requests

Comandos Assíncronos

- Chamadas assíncronas de procedimentos com uso de Workers (Async RPC)
- Geralmente usa execução de tarefas em filas
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Gestão de transações em cenários de falha é bastante complexo (**baixa resiliência e baixa confiabilidade**)

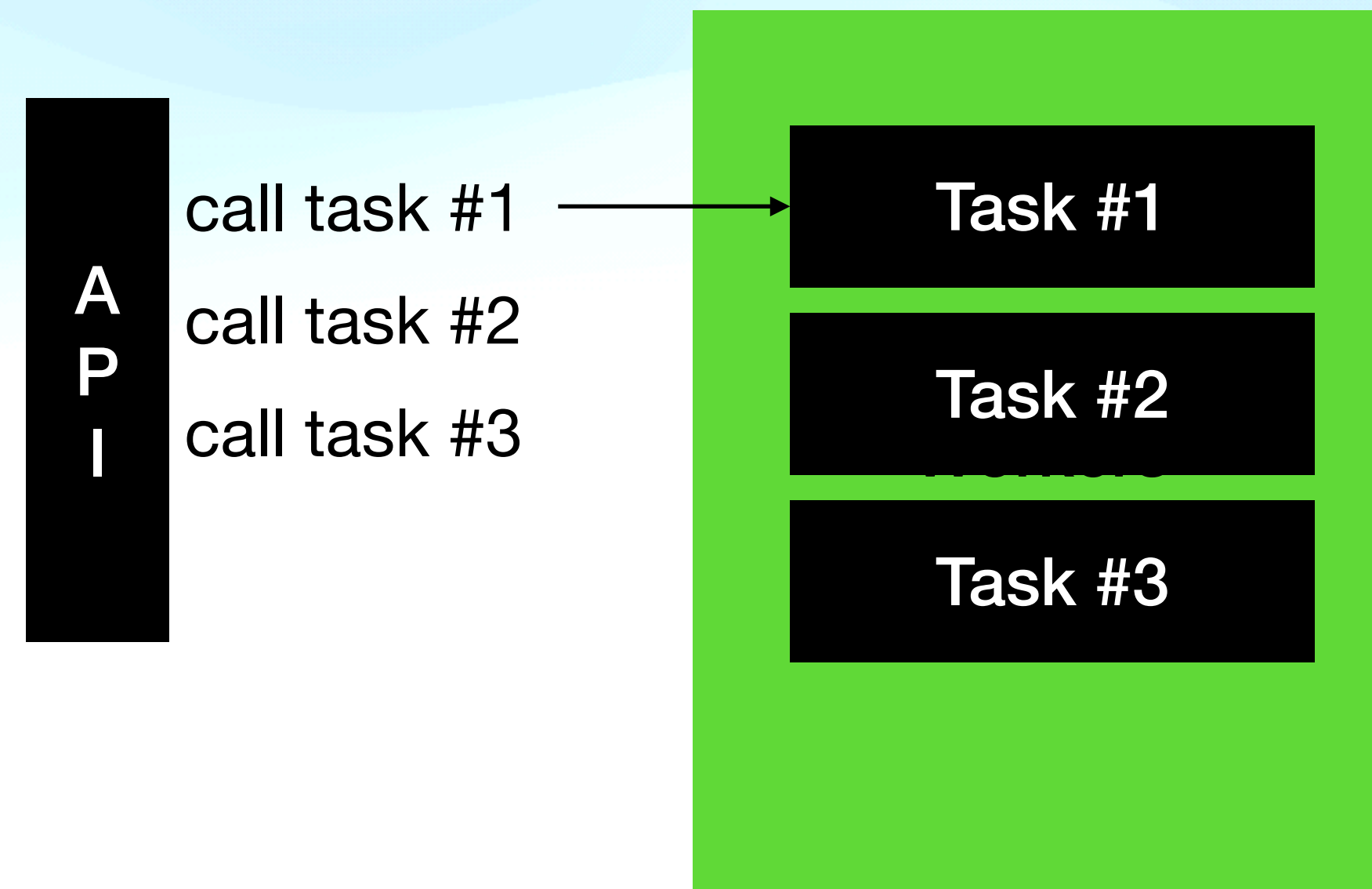
A
P
I

call task #1
call task #2
call task #3



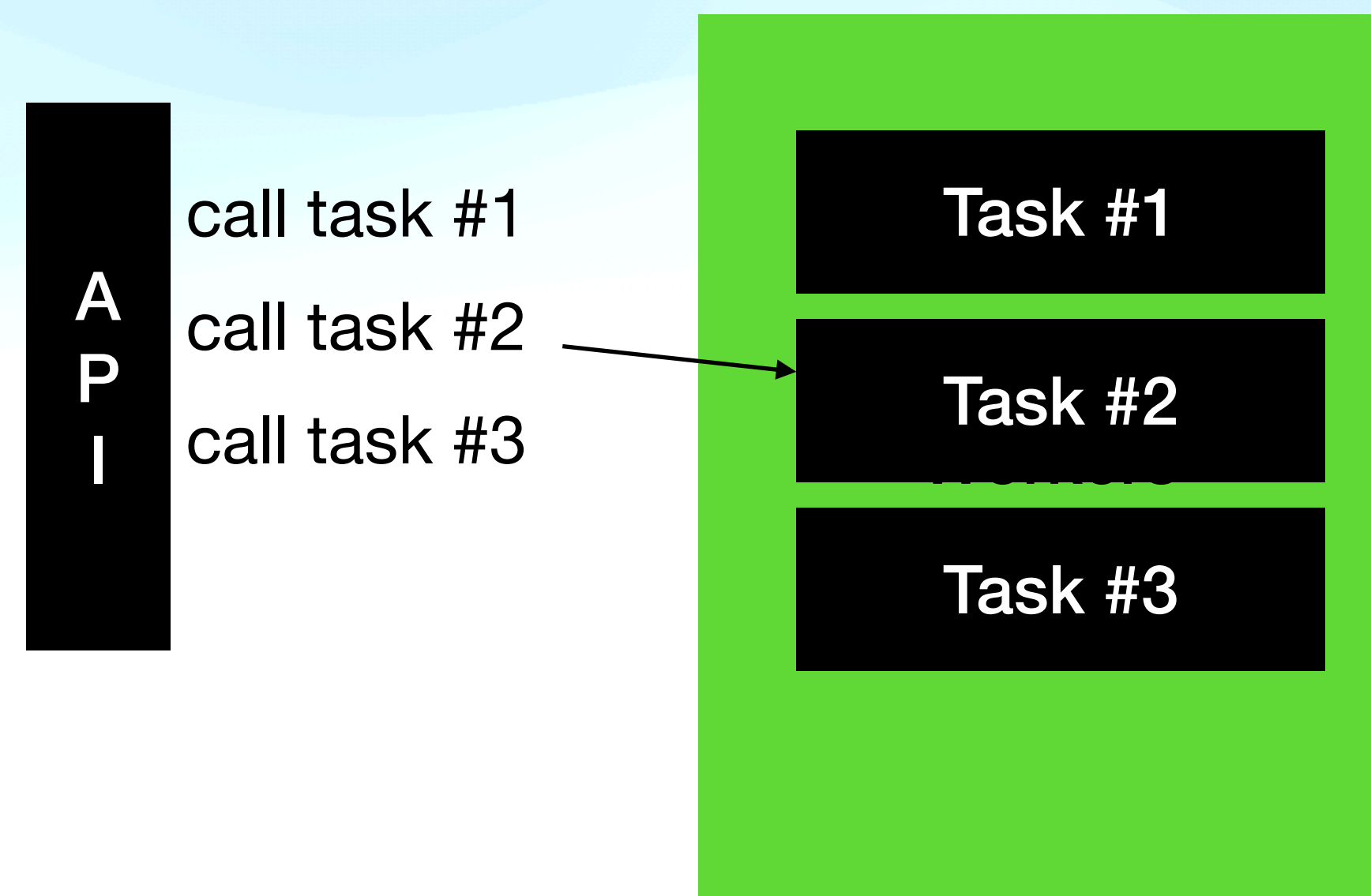
Comandos Assíncronos

- Chamadas assíncronas de procedimentos com uso de Workers (Async RPC)
- Geralmente usa execução de tarefas em filas
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Gestão de transações em cenários de falha é bastante complexo (**baixa resiliência e baixa confiabilidade**)



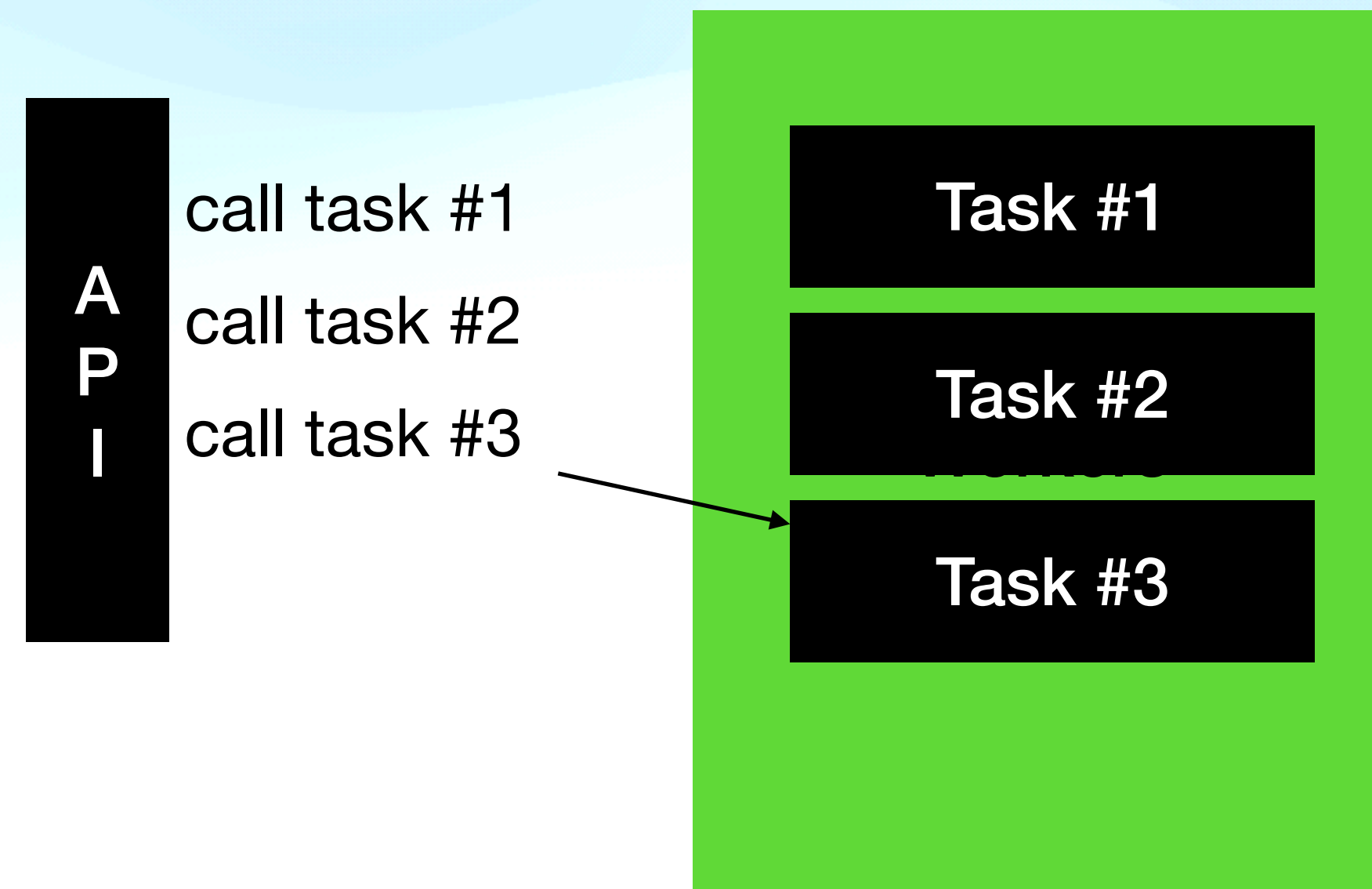
Comandos Assíncronos

- Chamadas assíncronas de procedimentos com uso de Workers (Async RPC)
- Geralmente usa execução de tarefas em filas
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Gestão de transações em cenários de falha é bastante complexo (**baixa resiliência e baixa confiabilidade**)



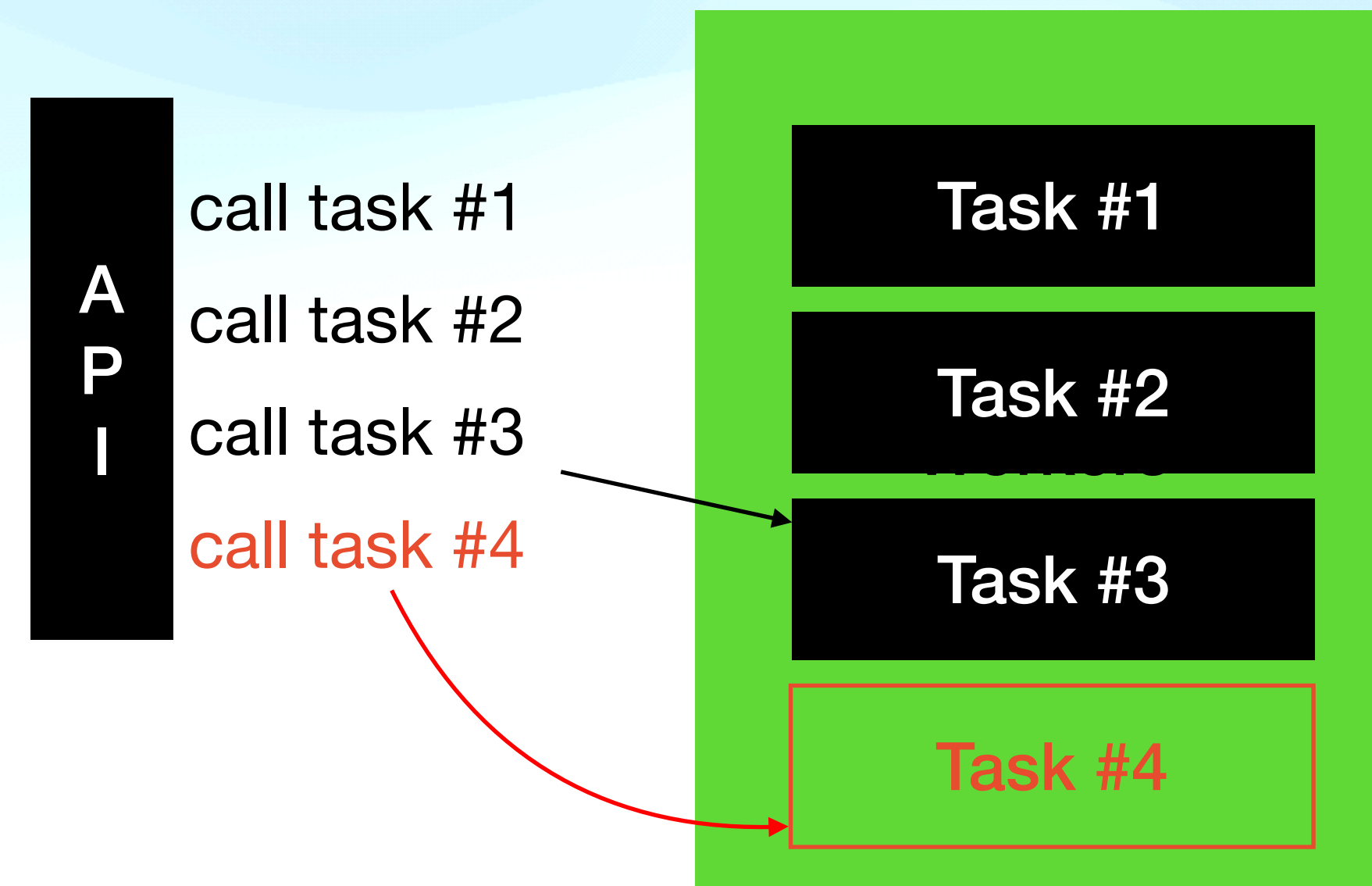
Comandos Assíncronos

- Chamadas assíncronas de procedimentos com uso de Workers (Async RPC)
- Geralmente usa execução de tarefas em filas
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Gestão de transações em cenários de falha é bastante complexo (**baixa resiliência e baixa confiabilidade**)



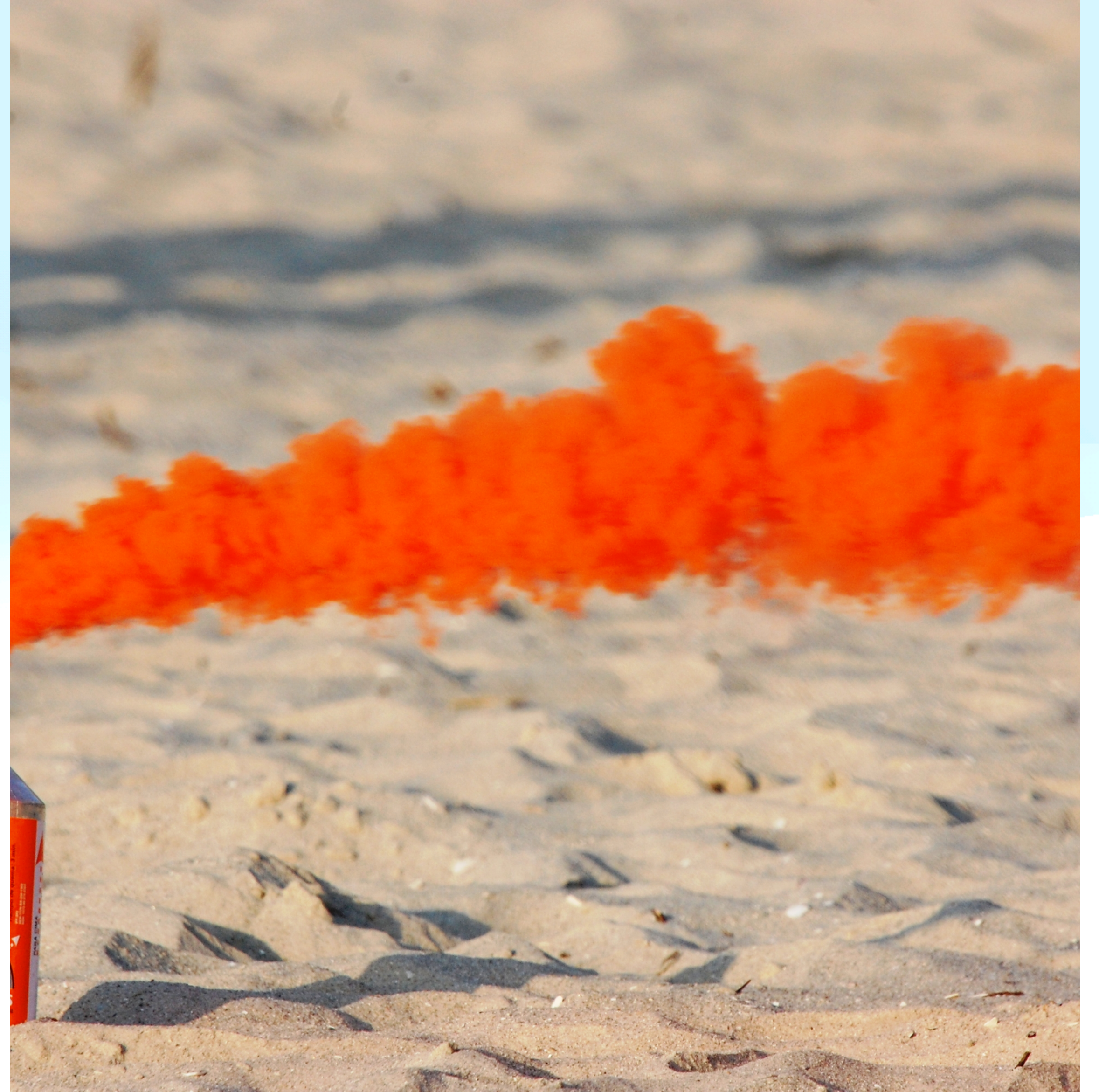
Comandos Assíncronos

- Chamadas assíncronas de procedimentos com uso de Workers (Async RPC)
- Geralmente usa execução de tarefas em filas
- Dependência explícita de serviços (**alto acoplamento**)
 - Novos procedimentos precisam ser adicionados na origem do comando
 - Gestão de transações em cenários de falha é bastante complexo (**baixa resiliência e baixa confiabilidade**)



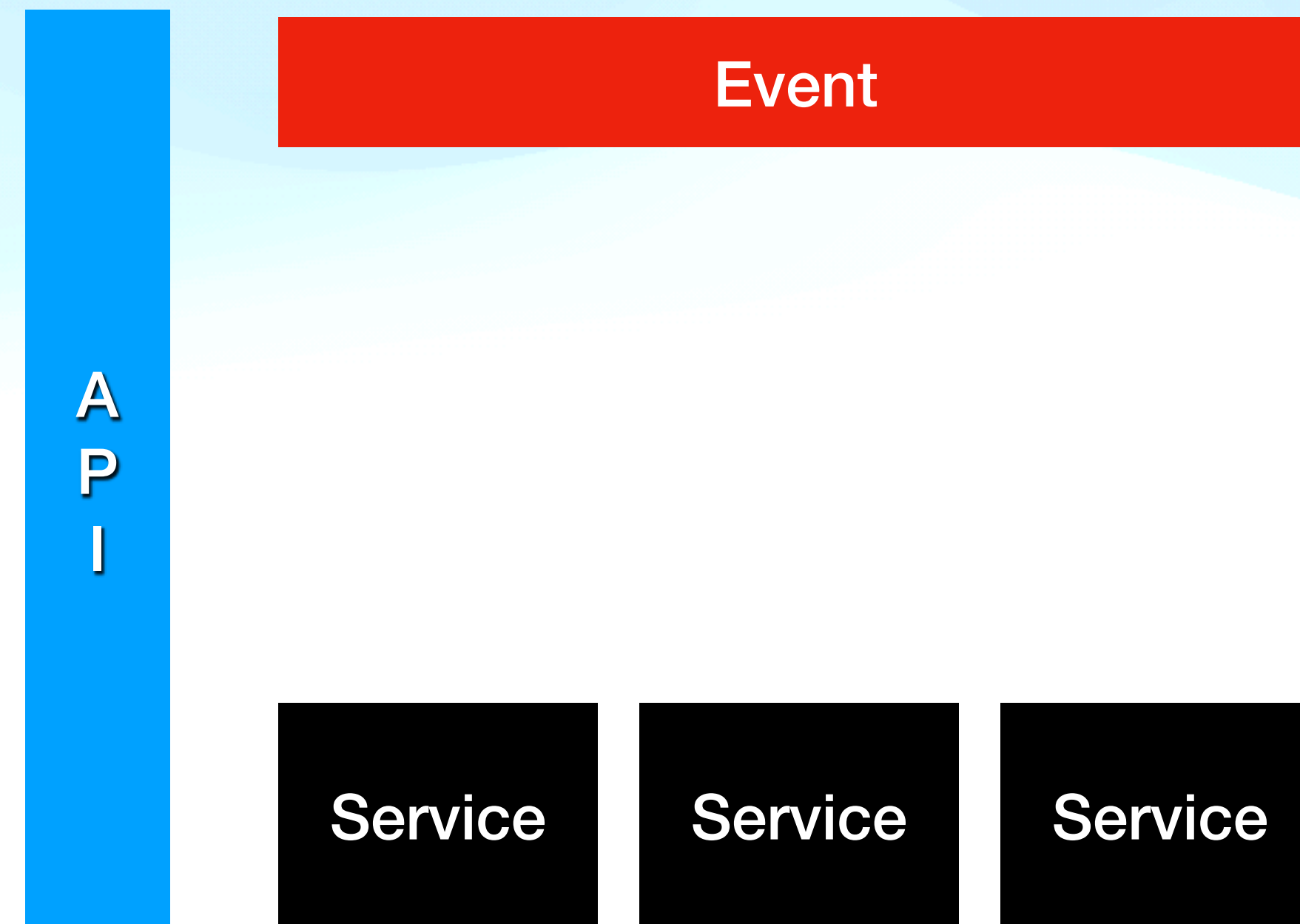
Eventos

"Fiz"



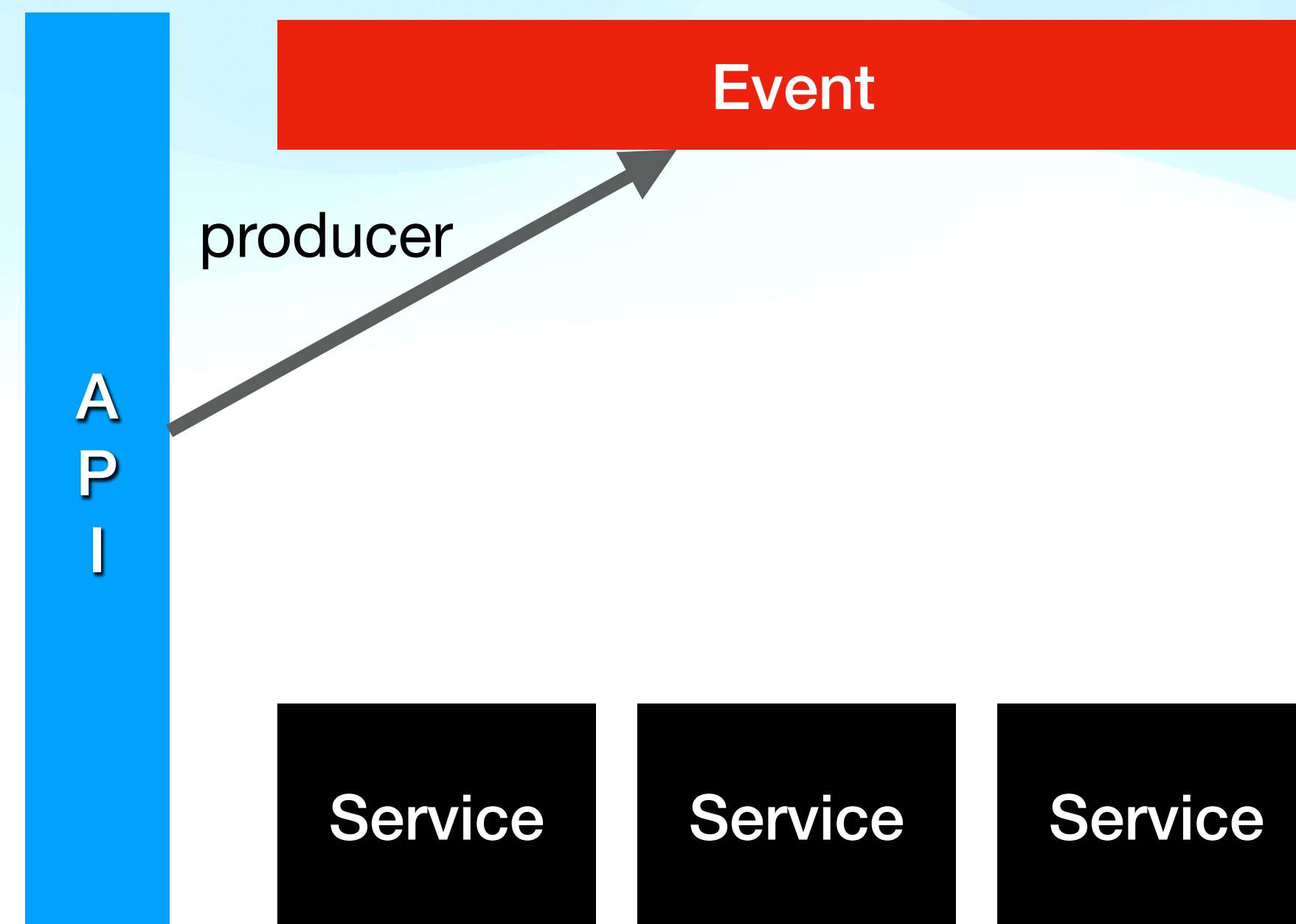
Eventos

- Serviço executa uma única operação e avisa que concluiu (Publisher/Producer)
- Serviços que precisam agir em função da operação se registram para receber esses eventos (Subscribers/Consumers)
 - Idealmente recebem esses eventos em filas persistentes (Queues)
 - Pode-se receber esses eventos via: Webhook, WebSocket, E-mail, Mobile Push Notification
- Publishers não tem conhecimento sobre Subscribers (alto desacoplamento)
- Eventos são mensagens contendo detalhes da operação e do sujeito da operação



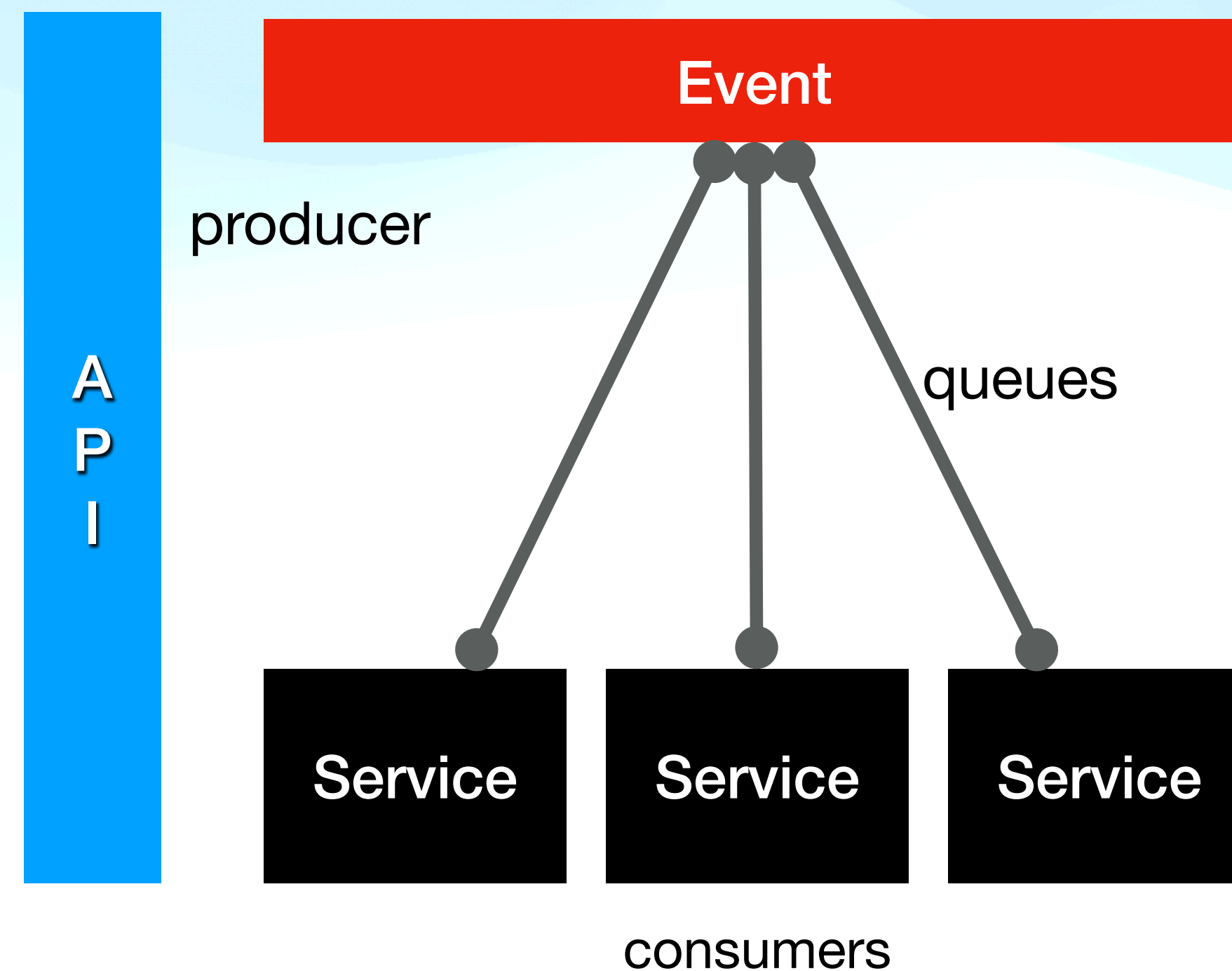
Eventos

- Serviço executa uma única operação e avisa que concluiu (Publisher/Producer)
- Serviços que precisam agir em função da operação se registram para receber esses eventos (Subscribers/Consumers)
 - Idealmente recebem esses eventos em filas persistentes (Queues)
 - Pode-se receber esses eventos via: Webhook, WebSocket, E-mail, Mobile Push Notification
- Publishers não tem conhecimento sobre Subscribers (alto desacoplamento)
- Eventos são mensagens contendo detalhes da operação e do sujeito da operação



Eventos

- Serviço executa uma única operação e avisa que concluiu (Publisher/Producer)
- Serviços que precisam agir em função da operação se registram para receber esses eventos (Subscribers/Consumers)
 - Idealmente recebem esses eventos em filas persistentes (Queues)
 - Pode-se receber esses eventos via: Webhook, WebSocket, E-mail, Mobile Push Notification
- Publishers não tem conhecimento sobre Subscribers (alto desacoplamento)
- Eventos são mensagens contendo detalhes da operação e do sujeito da operação



Arquitetura Baseada em Eventos

Elementos básicos



Especificações e Documentação

- Padrão para especificações de APIs e Serviços
- Produção de documentação de referência, uso, tutoriais, cursos, etc
- Guia para padronização de APIs e Serviços



ex. OpenAPI, Async API, Pactum, etc

Autenticação & Segurança

- Gerenciamento de identidade
- Gerenciamento de autenticação e credenciais
- Gerenciamento de autorização e permissões



ex. IAM, Cognito, Auth0

Mensageria

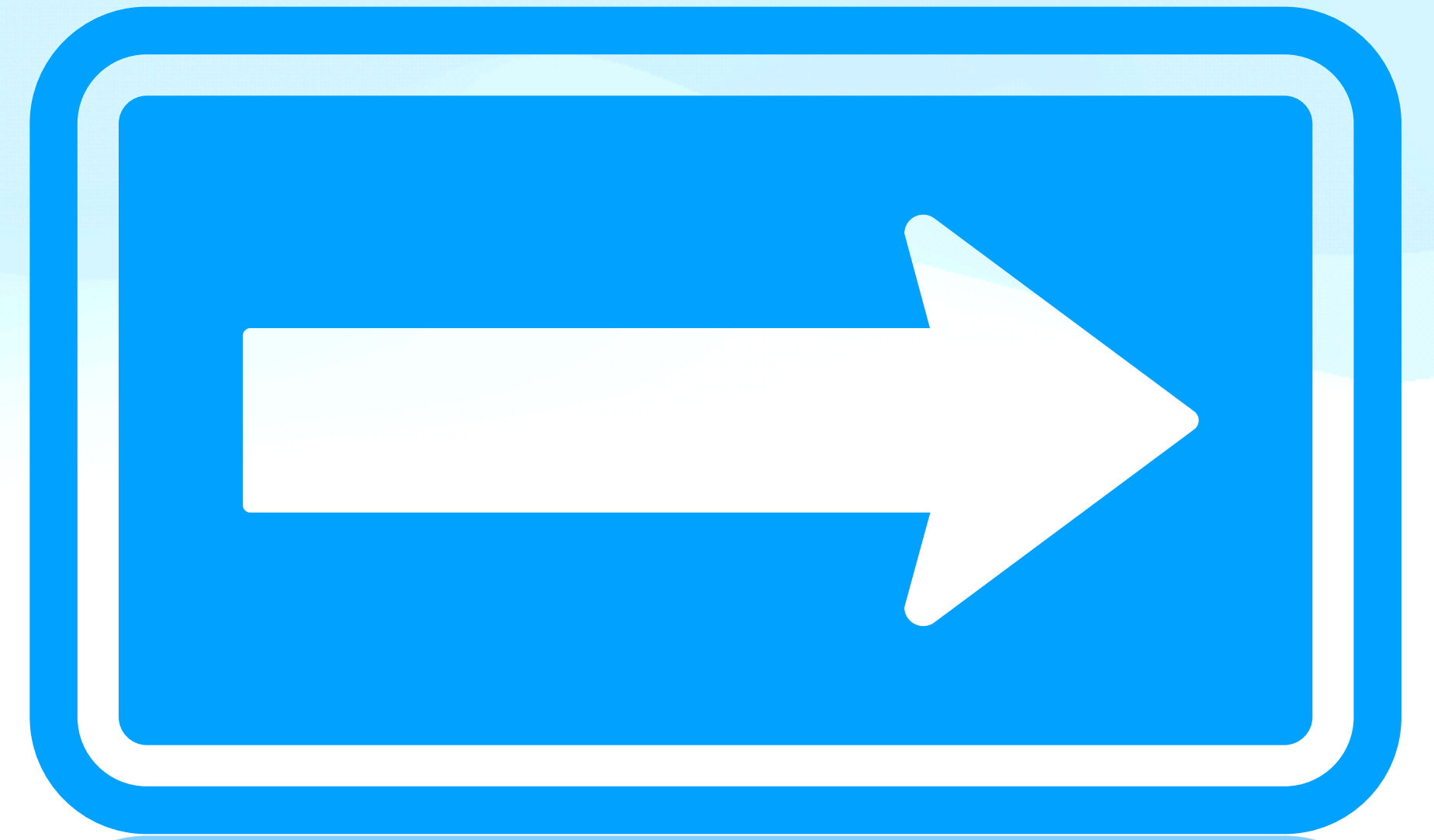
- Publisher/Subscribers (PubSub)
- Integração com sistema de autorização
- Interfaces (REST/gRPC) para:
 - Gerenciamento de Tópicos
 - Gerenciamento de Registro (subscription)
 - Registro condicional/filtro de registro
 - Publicação de Eventos
 - Backends para distribuição de eventos (fila, webhook, sms, e-mail, mobile, etc)
 - Feed com eventos distribuídos



ex. AWS SNS, Eventbridge, PubSub+

Filas Persistentes

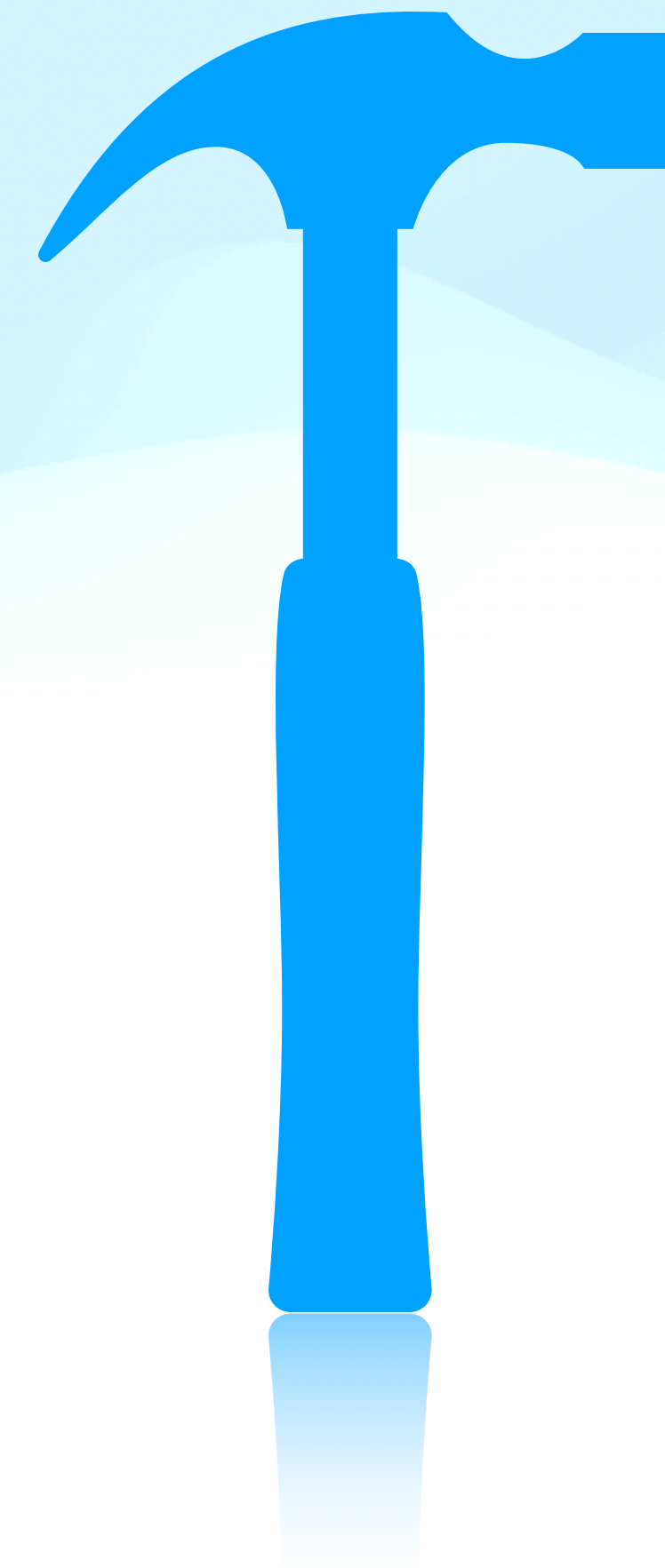
- Producer/Consumer
- Filas podem se registrar (subscribe) em tópicos no sistema de mensageira
- Interface (REST long-polling/gRPC) para:
 - Gerenciamento de filas
 - Inclusão/Leitura/Remoção de mensagens
 - Filas Dead-letter
 - Monitoramento



ex. AWS SQS

Webhook Trigger

- Disparador de requests
- Webhookers podem se registrar (subscribe) em tópicos no sistema de mensageira
- Interface (REST long-polling/gRPC) para:
 - Gerenciamento de webhooks
 - Sistema de política de Retry e recuperação de falhas
 - Sistema de circuit-breaker
 - Monitoramento



ex. AWS SNS

Funções

- Processamento de eventos e resposta à requisições
- Pode ter como entrada: requisição do API Gateway, mensagens em uma fila ou eventos em um tópico (função como subscriber).
- Sistema de configuração de deployment
- Suporte para as linguagens mais usadas: Python, Go, etc
- Interface (REST long-polling/gRPC) para:
 - Criação/Deployment de funções
 - Monitoramento



$f(x)$

ex. AWS Lambda + AWS SAM

Interface de Serviços

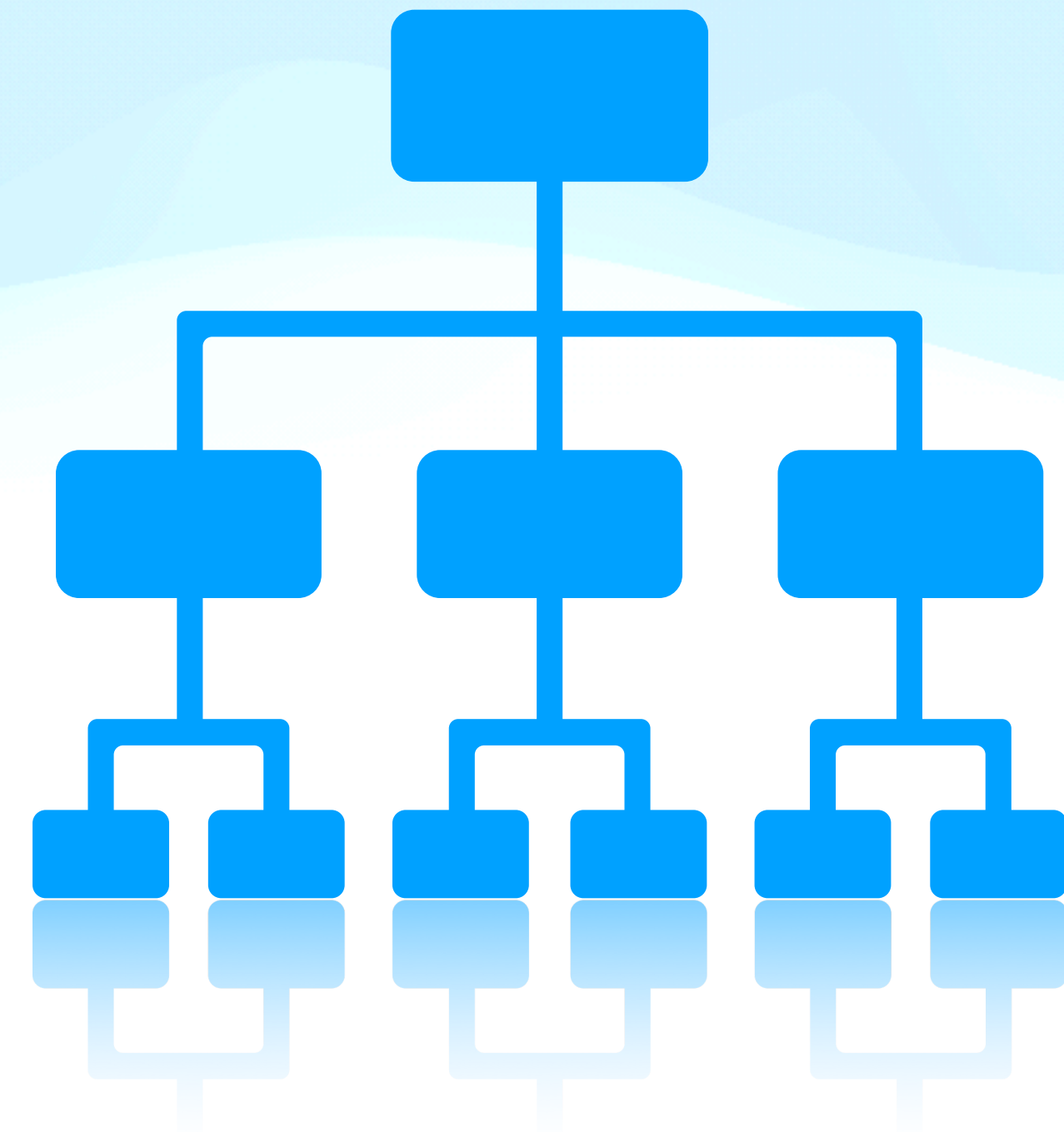
- Ponto de entrada para serviços e APIs
- Deve receber requisições (REST/gRPC) externas, autorizar no sistema de Authorization/Authentication e repassar para uma função ou publicar o requeste em um tópico
- Suporte para configurações via Deployment configuration
- Interface (REST long-polling/gRPC) para:
 - Gerenciamento de endpoints
 - Gerenciamento de autorizações para endpoints
 - Gerenciamento de conexões com funções e tópicos de eventos



ex. AWS API Gateway

Automação de Workflow

- Interface para automatizar conexões entre tópicos, filas e funções.
- Configuração de Workflow via API
- Criação/deploy/configuração de funções simples via API



ex. AWS Step Functions, Zapier, IFTTT

Infraestrutura

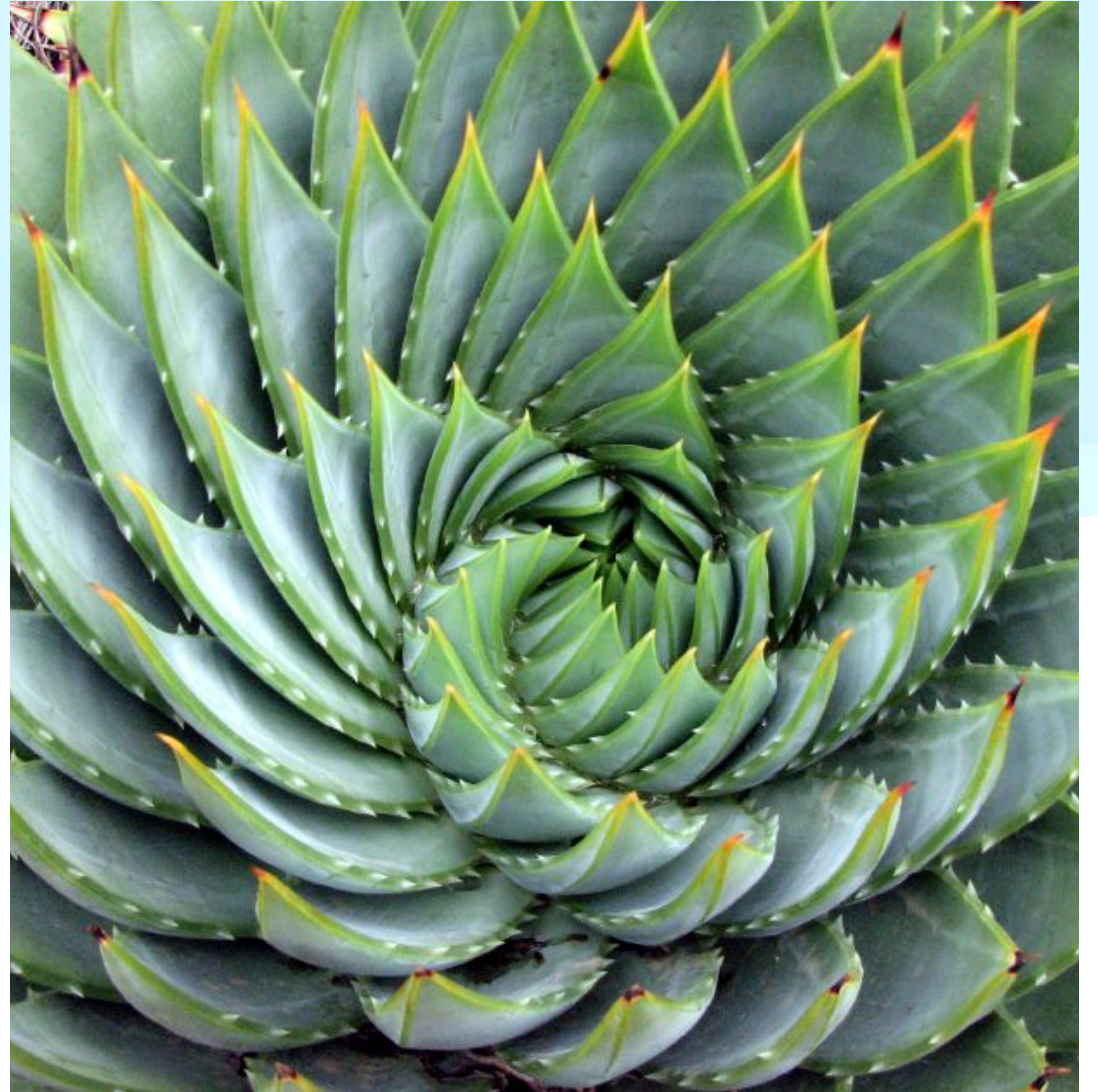
- Monitoramento
 - Alerta
- Logs
- Tracing
- Multi-environment/Account
- Billing
- Console Web/Mobile



ex. AWS CloudWatch

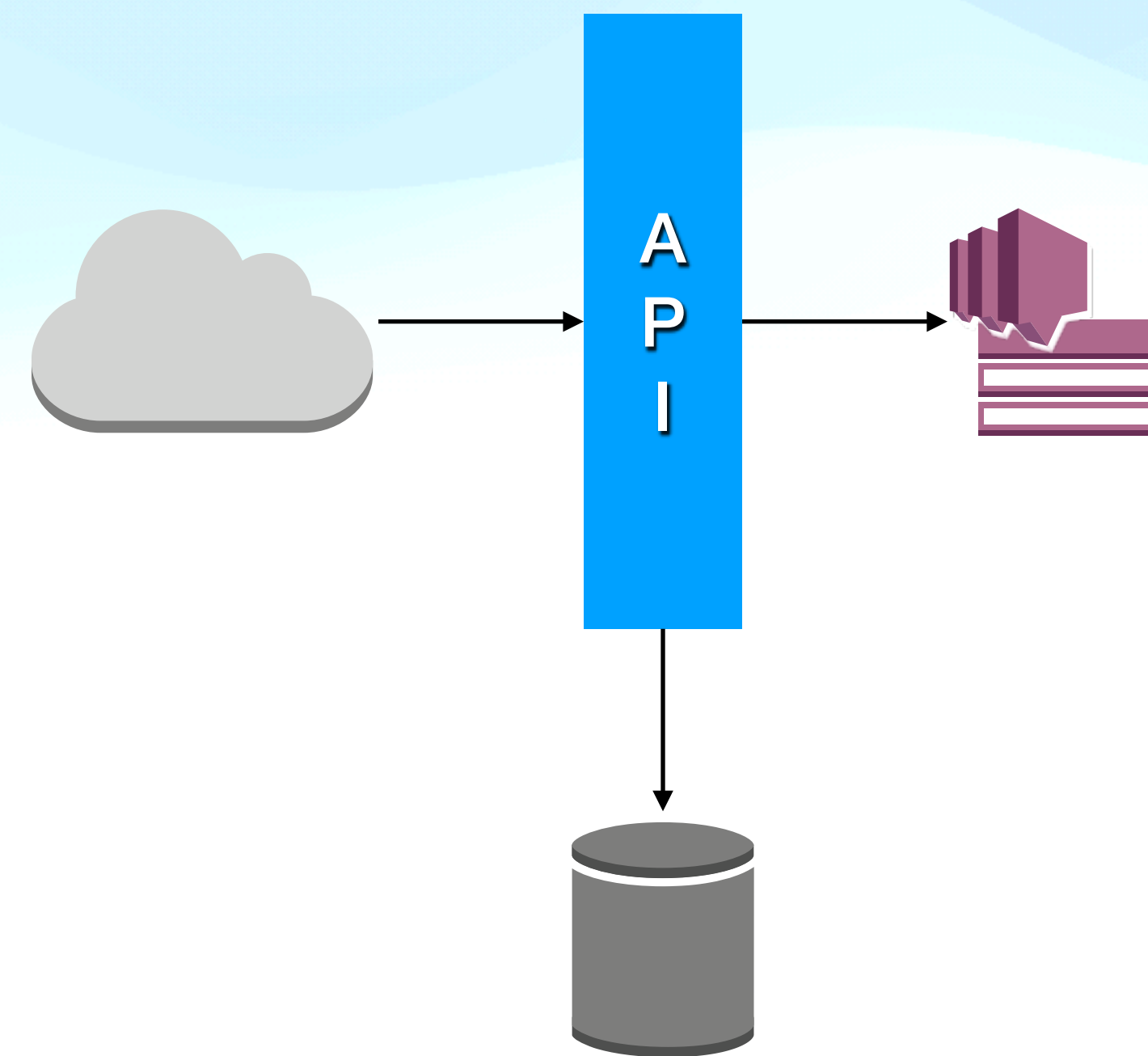
Arquitetura

Padrões (Patterns)



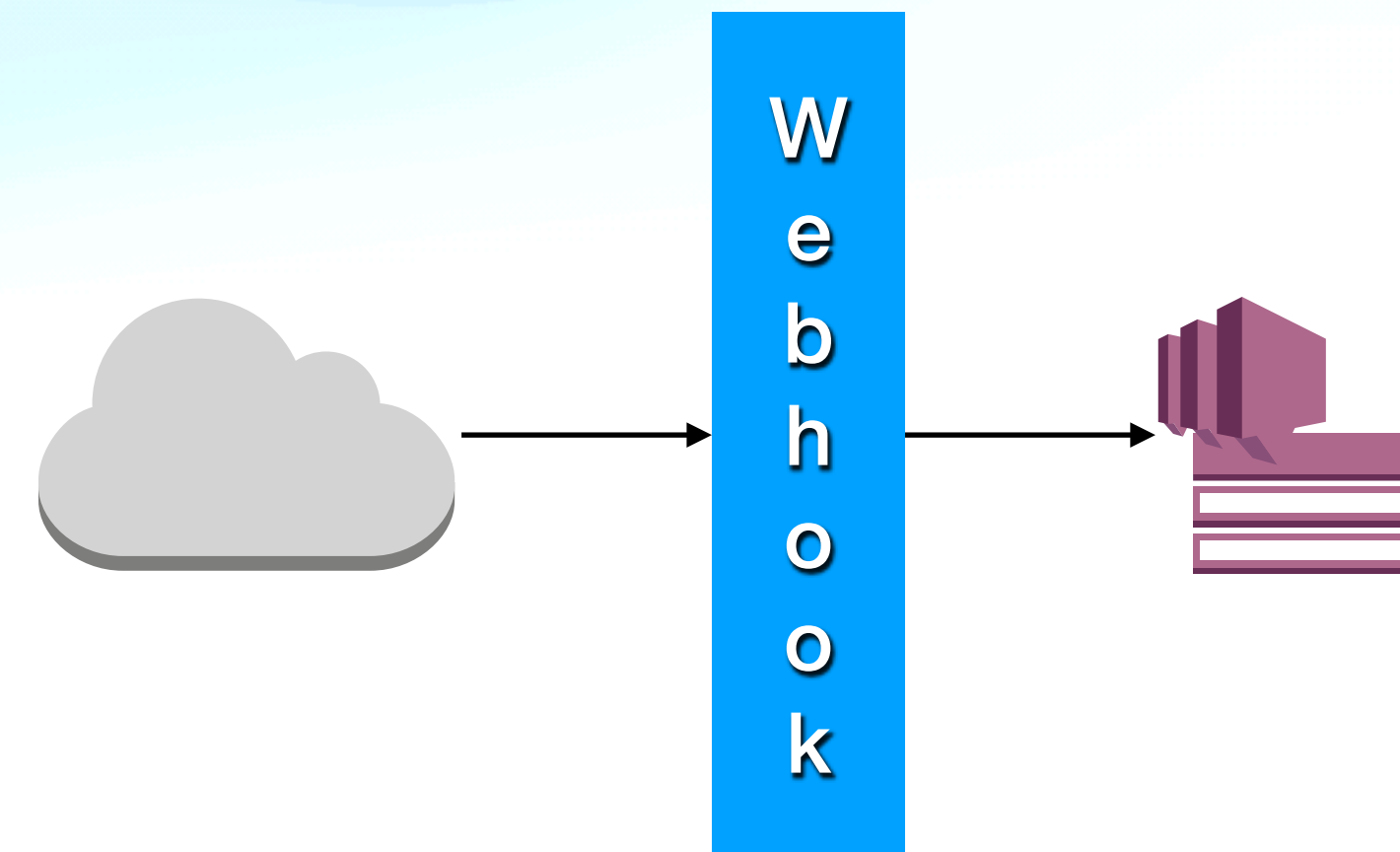
API

- Data Entry Point
- Data Validation
 - Workflow Management (eg. status/state machine)
- Data Persistence
- Event Triggering
- Idempotency handling (eg. discard duplicated requests returning a HTTP 304 Not Modified)



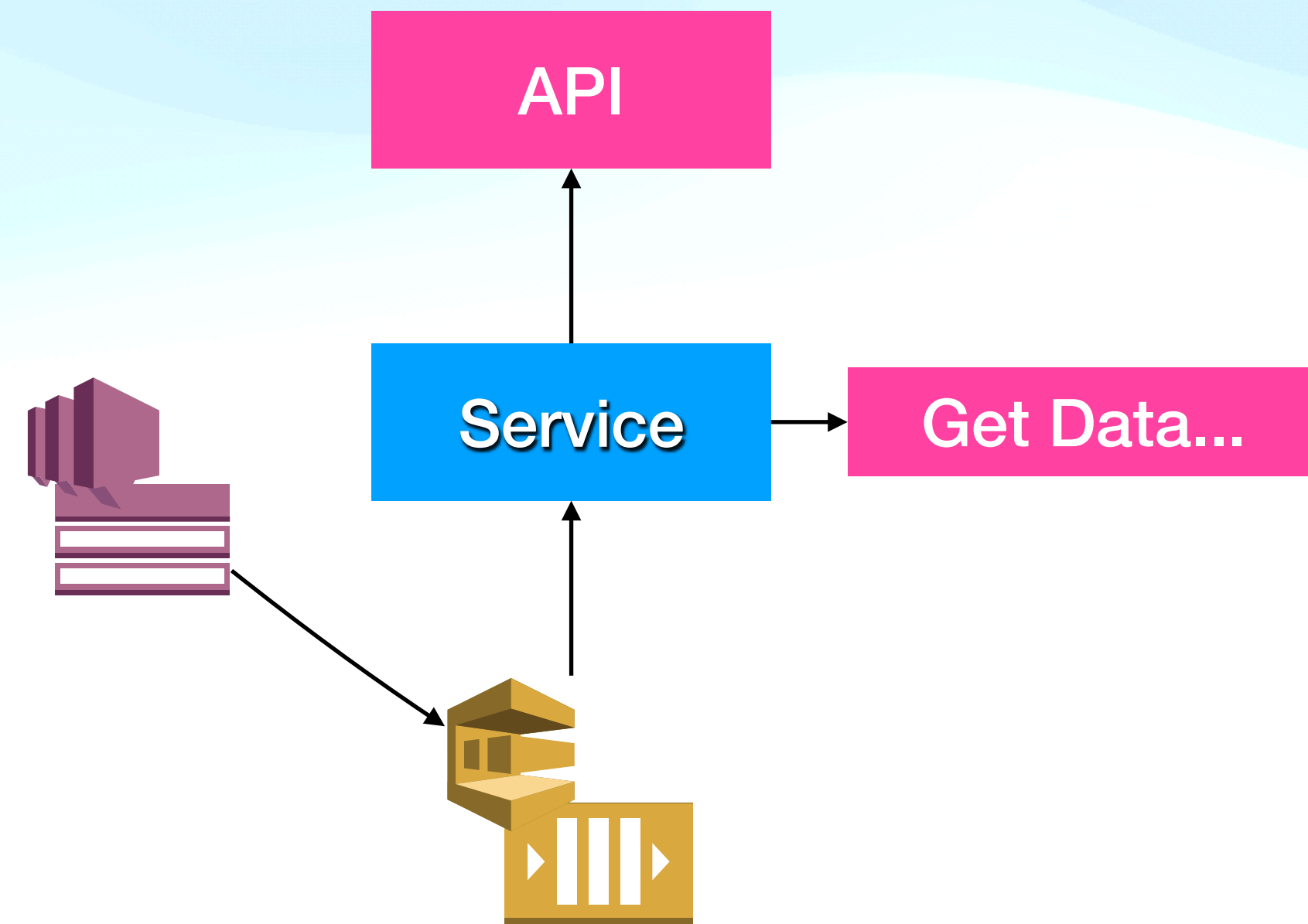
API (webhook)

- Data Entry Point
- No Data Persistence
- Proxy HTTP ➔ SNS
- Event Triggering



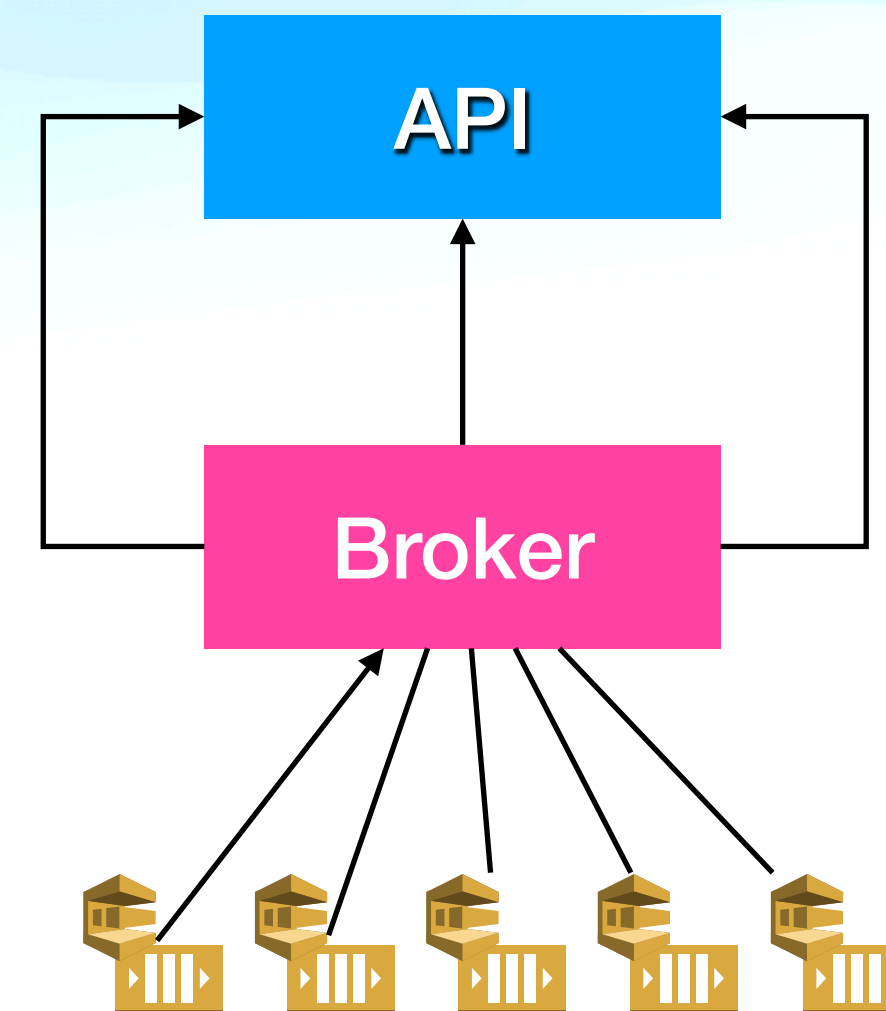
Service (consumer)

- Event Handling / Message Processing
- Business Logic



Service (broker)

- Event Handling / Message Processing
- Business Logic



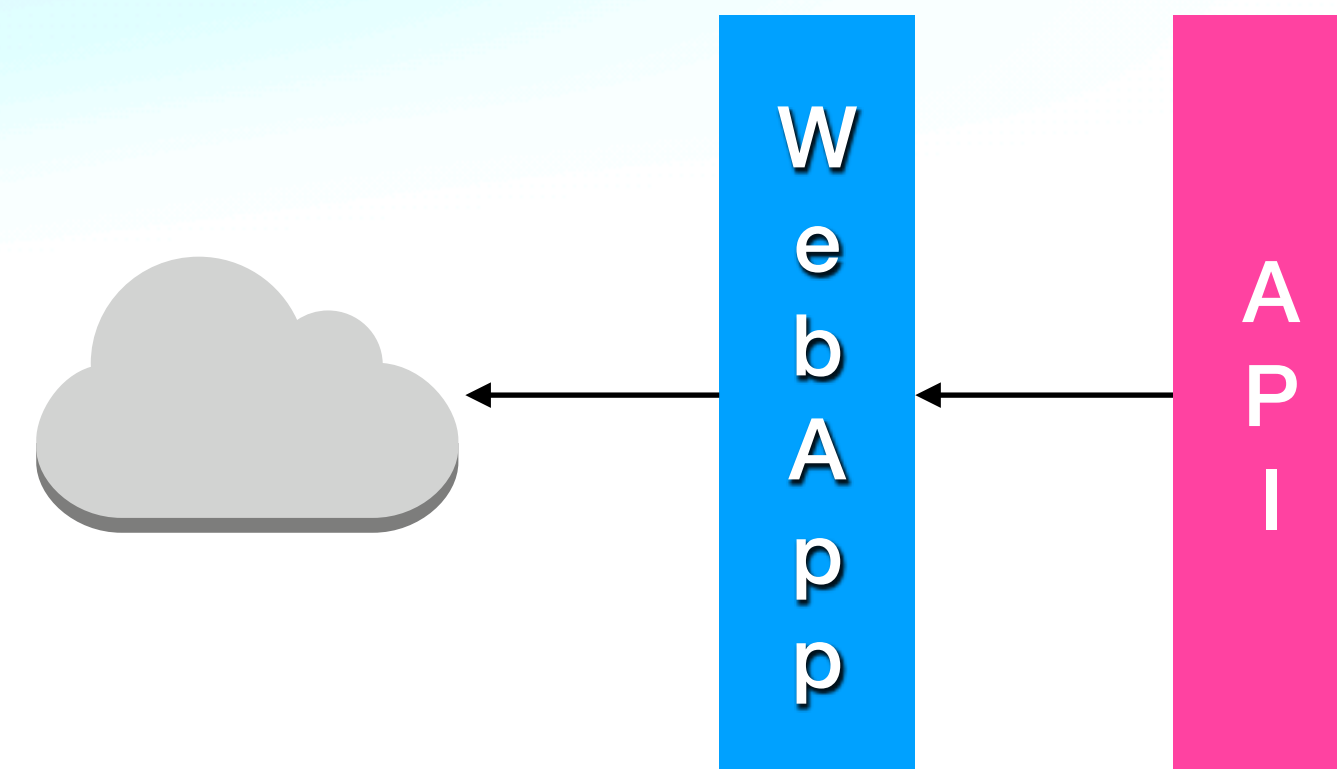
Service (job)

- Scheduled Job
- No Persistence
- Event Triggering



Client

- Web or Mobile Applications
- No Persistence (or basic persistence)
- Web presentation of APIs' resources



milestones

Diretrizes e arquitetura

- Update and maintain Technology Guidelines
 - <https://github.com/osantana/development-guidelines>
 - New API guidelines
 - Serverless architecture
 - Maturity Model for Microservices
- Commerce Services Design & Implementation Playbook
 - How to design, create, implement and launch a new service

Diretrizes e arquitetura

- Framework for service specification, documentation and style guide adherence (open-source)
- Asynchronous Framework for REST APIs (open-source)
 - Integrated with specification and documentation framework
 - Support for serverless environment
- Library for quick development of web clients (based on Pactum's specs)
 - <https://github.com/pactum-org/pactum>
 - Support for (distributed) retry, circuit breaking, throughput control, fail recovery, rich model and HATEOAS
- Improvements on library for message consumer services (eg. Loafer)
 - <https://github.com/georgeyk/loafer>

Sistema de gestão de identidade

- First service at platform
 - Created with Platform Playbook
- Focus on security and flexibility on profiles, roles, groups and permission.
- Support for Users, Profiles, Organizations, Organization Profiles and client credentials
 - Prepared for future support for multi-tenancy support
- SDK for fast adoption in current services

Sistema de mensageria

- Support for static topic subscribing (like AWS SNS)
- Multiple message publishing backends
 - SQS
 - Web hook
 - Push notification channels
- Support for individual components registration as messaging recipient