UNIVERSIDADE FEDERAL
DE MATO GROSSO

Pós-graduação em DevOps

Desenvolvimento e Operações Integrados

Arquitetura de Micro(serviço)

João Paulo Delgado Preti
Professor Titular do IFMT
Departamento da Área de Computação (DCOM)
preti.joao@ifmt.edu.br

2024

Índice

Serviço? Micro?

Benefícios

Desafios

Reflexões

Serviço

Deixamos de adquirir um produto para obter um serviço:

- Telefonia (linha)
- Telefonia (modem)
- Telefonia (telefone)
- Música
- Livro
- Carro
- Casa
- ...

Na computação também:

- IaaS **I**nfrastructure **a**s **a** **S**ervice
- PaaS **P**latform **a**s **a** **S**ervice
- SaaS **S**oftware **a**s **a** **S**ervice
- ...

Arquitetura Orientada a Serviço (SOA)

É um **conceito** de arquitetura:

- Disponibilizar as funcionalidades de um sistema ou processos de negócio como serviço

Serviço - Definição

- Dicionário:
    - ação ou efeito de servir, de dar de si algo em forma de trabalho.
    - exercício e desempenho de qualquer atividade.
- Latim:
    - ETIM lat. servitĭum,ĭi 'condição de escravo, escravidão, jugo, obediência

Gartner Group

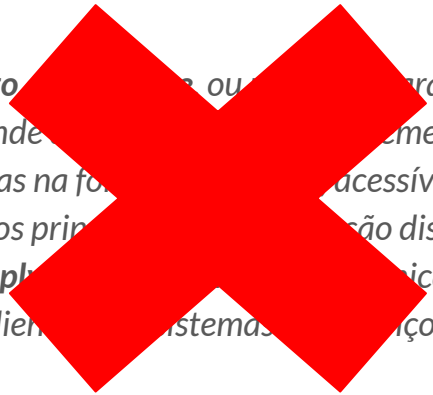
*SOA é uma abordagem arquitetural **corporativa** que permite a criação de serviços de negócio **interoperáveis** que podem facilmente ser **reutilizados** e **compartilhados** entre **aplicações** e **empresas**.*

OASIS

*Um **paradigma** para **organizar e utilizar capacidades distribuídas** que podem estar sob o controle de diferentes domínios de propriedade. Provê uma maneira uniforme de oferecer, descobrir, interagir e utilizar capacidades para produzir efeitos desejados consistentes com pré-condições e expectativas **mensuráveis**.*

Wikipedia

*...é um **padrão de projeto** [illegible] ou [illegible] arquitetura de software de baixo acoplamento, onde [illegible] mentadas nas aplicações devem ser disponibilizadas na fo[illegible] acessíveis normalmente via **web services**, é baseada nos prin[illegible]ção distribuída, que utiliza o **paradigma request/repl**[illegible]icação entre os sistemas clien[illegible]istemas [illegible]iços.*



Serviço

- Escopo bem definido
- Capacidades e restrições bem definidas
- Facilidade de localização (publicidade) e interação (pré-condições,...)
- **Auditável**
- **Mensurável**

Exemplo

- **Escopo:**
  - Pintura residencial interna e externa
- **Capacidades e restrições bem definidas**
  - 8h/dia
  - Alvenaria e madeira
  - Restrições: não realiza pintura eletrostática
- **Facilidade de localização (publicidade) e interação (pré-condições,...)**
  - Publicidade: Facebook, instagram, banners, rádio
  - Interação: Telefone, whatsapp, e-mail
  - Pré-condições: aquisição de material de pintura e pagamento de 50% da mão de obra.
- **Auditável**
  - Quem foi o(a) pintor(a)?
  - Quando o serviço foi realizado?
  - Quem era o(a) cliente?
  - Materiais utilizados?
- **Mensurável**
  - Quanto tempo levou?
  - Quantos m2?
  - Volume de tinta?
  - Vol/m2?
  - Tempo/m2?

Exemplo

- **Escopo:**
  - Atendimento ao Cliente
- **Capacidades e restrições bem definidas**
  - Realizado unicamente por meios digitais (chat, FAQ e e-mail)
  - das 08:00-18:00 horário de brasília em dias úteis para chat
- **Facilidade de localização (publicidade) e interação (pré-condições,...)**
  - Publicidade: Facebook, Instagram, site da empresa. e-mail marketing
  - Interação: site, whatsapp, e-mail
  - Pré-condições: número de protocolo e/ou ter cadastro como cliente
- **Auditável**
  - Quando ocorreu?
  - Quem era o(a) cliente?
  - Quem fez o atendimento?
- **Mensurável**
  - Tempo gasto?
  - Foi solucionado?
  - Grau de satisfação?

Exemplo

- **Escopo:**
  - Controle de despesa familiar
- **Capacidades e restrições bem definidas**
  - Não sincroniza com entidades financeiras
  - Permite o controle de gastos por mais de um membro da unidade familiar
  - Permite acompanhar a evolução da despesa por membro ou por unidade familiar de forma geral ou por classificações de despesa
- **Facilidade de localização (publicidade) e interação (pré-condições,...)**
  - Publicidade: repositório de serviços
  - Interação: CLI, API Rest e Whatsapp
  - Pré-condições: estar autenticado e autorizado
- **Auditável**
  - Quem foi que registrou a despesa?
  - Quando ocorreu a despesa?
  - Quem pagou?
  - Qual o tipo de despesa?
- **Mensurável**
  - Gasto diário, mensal, anual
  - Gasto por tipo
  - Gasto por membro da unidade familiar
  - Valor futuro a pagar

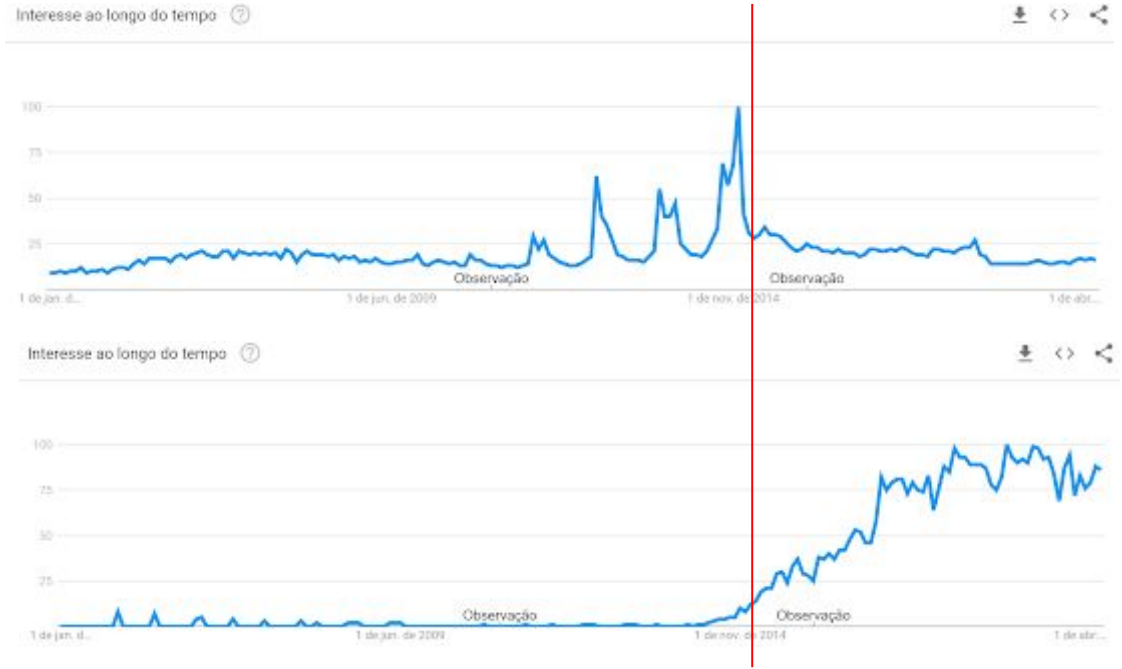
Resumindo

SOA não é uma tecnologia.

SOA não é uma metodologia.

SOA pode ser considerada uma filosofia arquitetural.

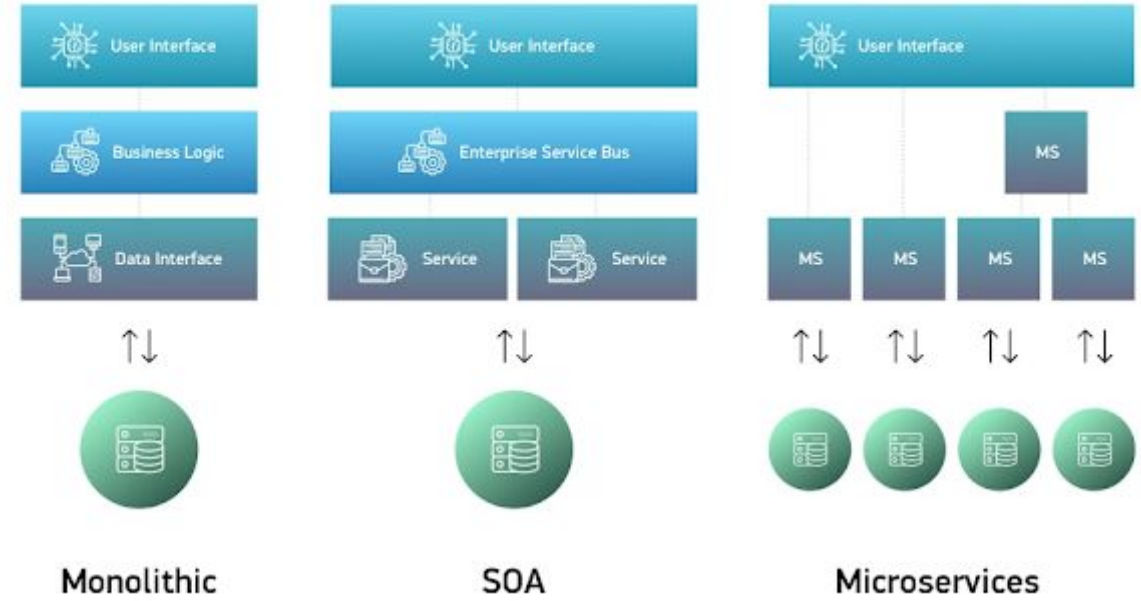
SOA vs Microservices



Fonte: Google Trends 16/08/2020

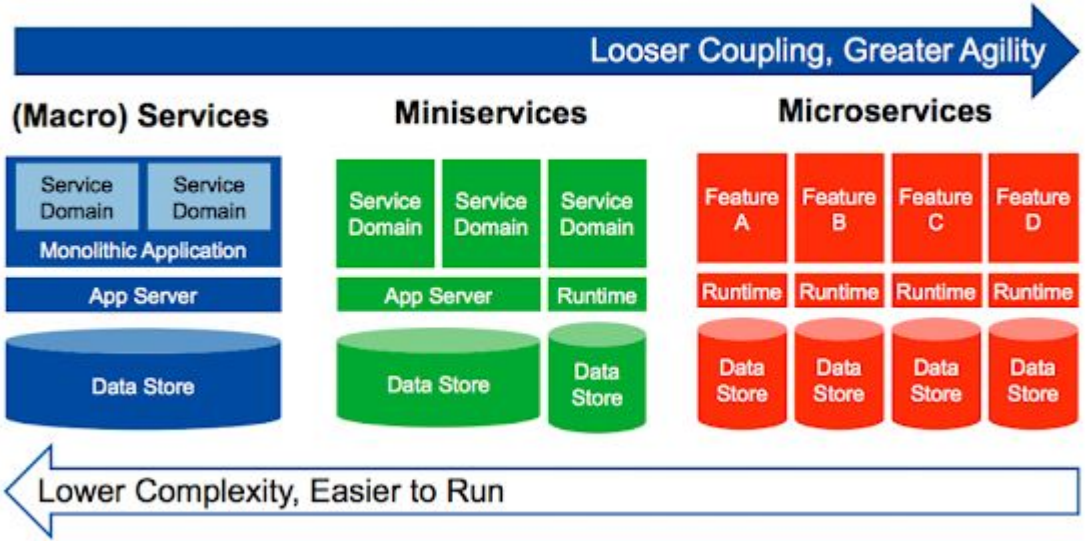
SOA vs Microservices

Microservices and web services have been instrumental in providing high-grade software. They are a good alternative to software solutions based on the monolithic architecture that made them so popular.

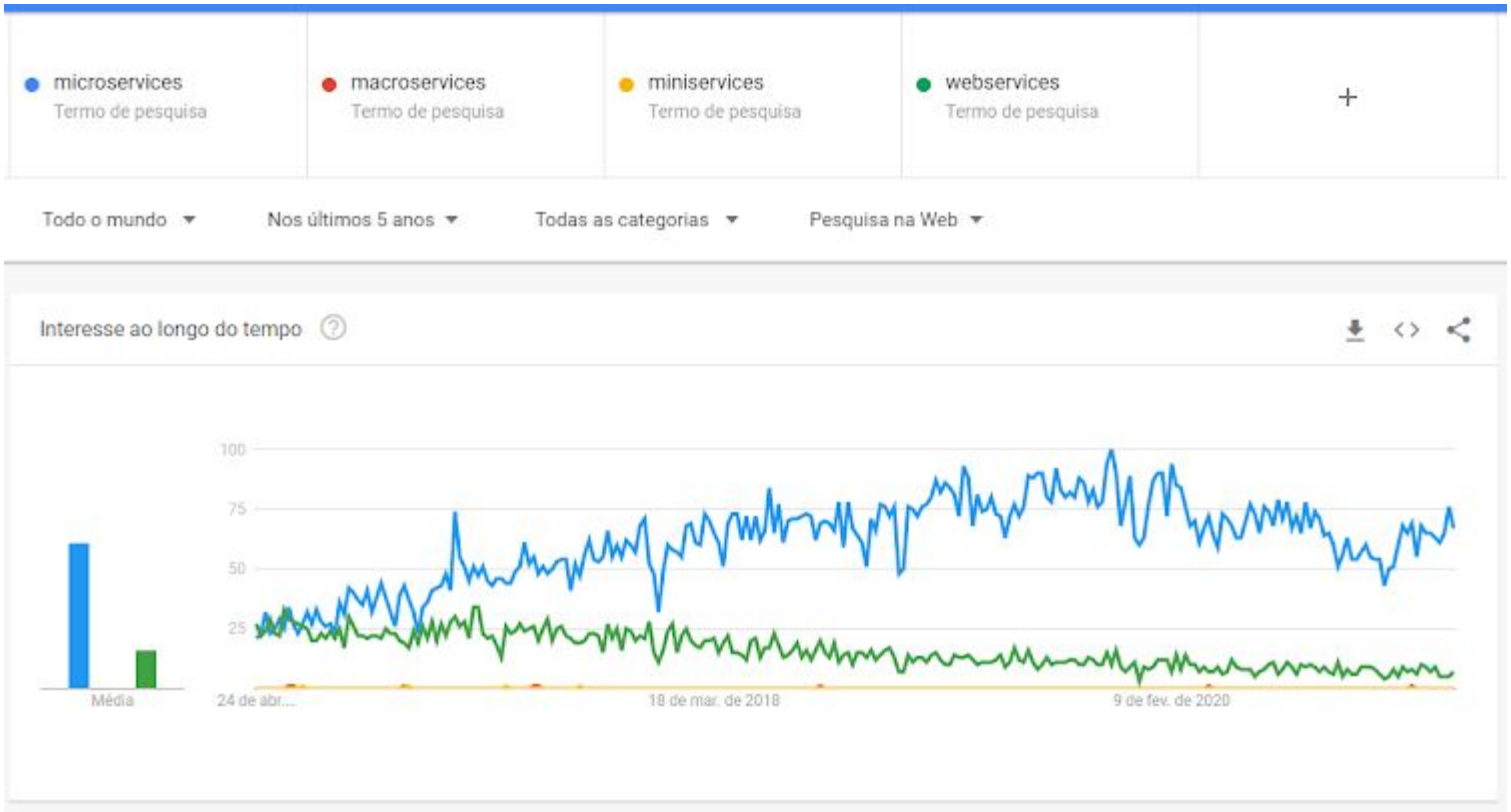


Fonte:https://www.velvetech.com/blog/microservices-vs-web-services/

...Services


Looser Coupling, Greater Agility
(Macro) Services
Miniservices
Microservices
Service Domain
Service Domain
Monolithic Application
App Server
Data Store
Service Domain
Service Domain
Service Domain
App Server
Runtime
Data Store
Data Store
Feature A
Feature B
Feature C
Feature D
Runtime
Runtime
Runtime
Runtime
Data Store
Data Store
Data Store
Data Store
Lower Complexity, Easier to Run

...Services



Fonte:
Google Trends
19/04/2021

Web Service vs Microservice

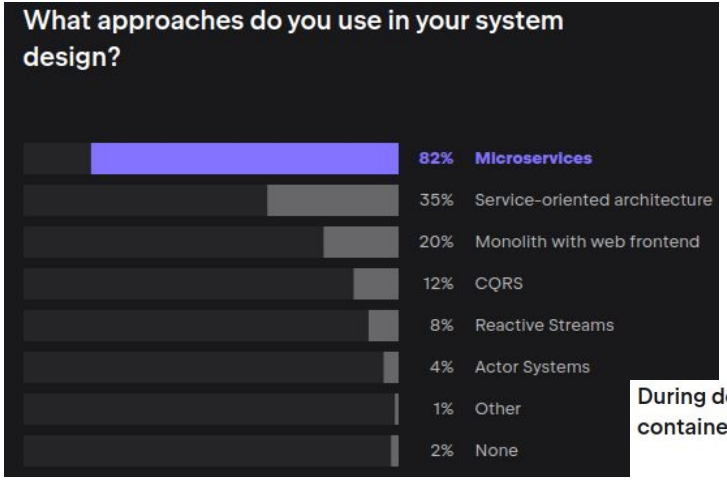
Um web service é uma **tecnologia** para prover serviços na web ou por meio do protocolo HTTP.

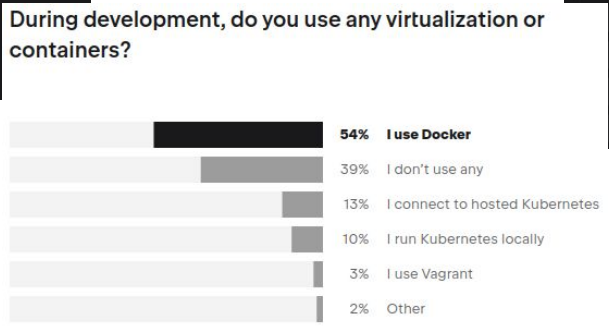
Um microserviço é uma **arquitetura de software**, que pode ser implementada com web services.

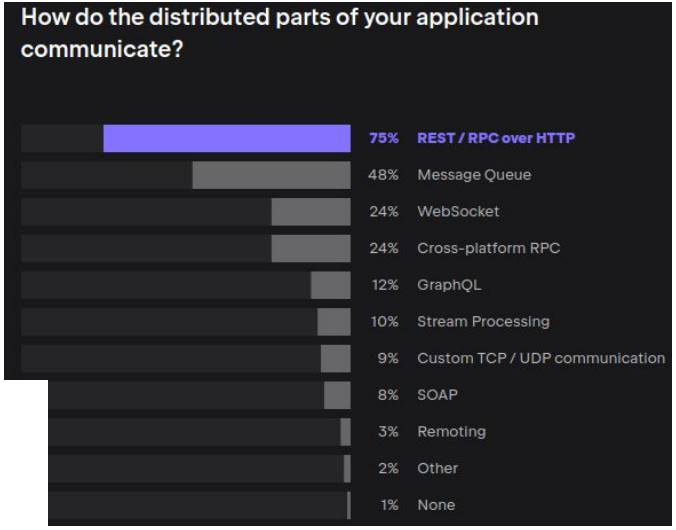
Um web service é uma estratégia para tornar os serviços de uma aplicação disponíveis para outras aplicações por meio de uma interface web.

Um microserviço é uma aplicação pequena e autônoma que realiza um serviço específico para a arquitetura de uma aplicação maior.

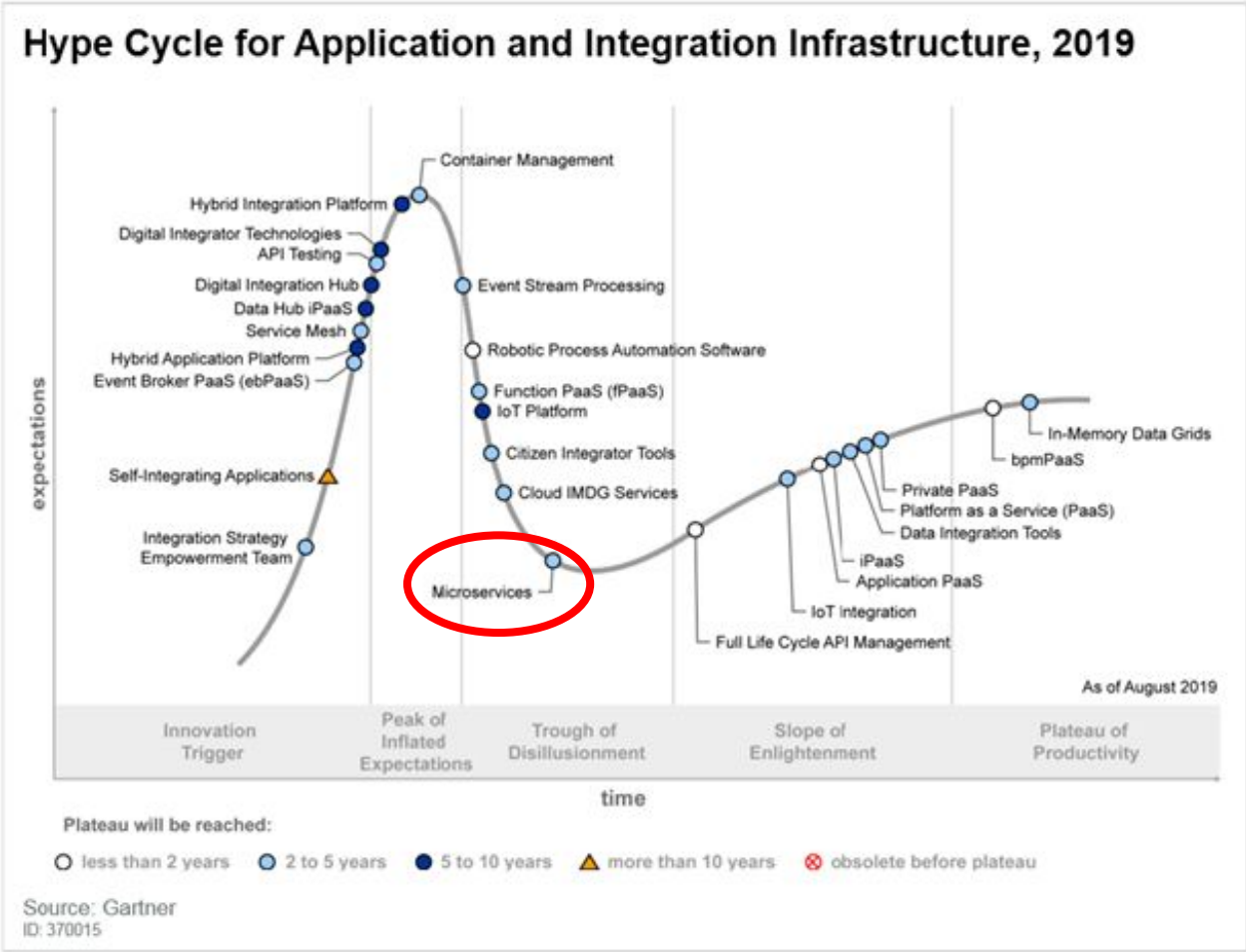
Jetbrains Report (Developer Ecosystem 2023)







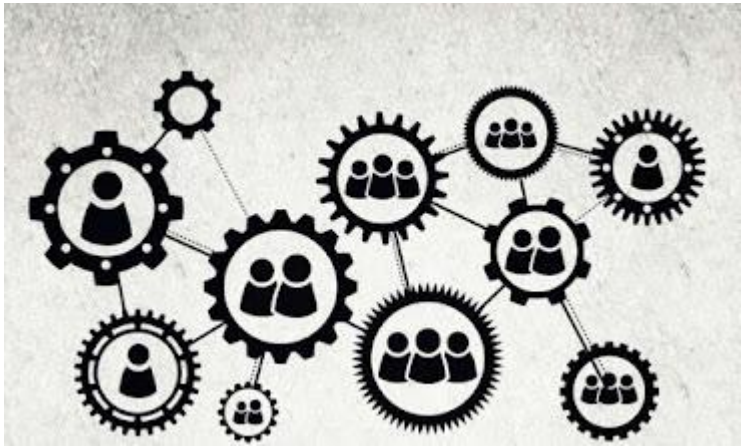
Fonte:https://www.jetbrains.com/lp/devecosystem-2023/


Hype Cycle for Application and Integration Infrastructure, 2019
Container Management
Hybrid Integration Platform
Digital Integrator Technologies
API Testing
Digital Integration Hub
Data Hub iPaaS
Service Mesh
Hybrid Application Platform
Event Broker PaaS (ebPaaS)
Event Stream Processing
Robotic Process Automation Software
Function PaaS (fPaaS)
IoT Platform
Citizen Integrator Tools
Cloud IMDG Services
Self-Integrating Applications
Integration Strategy Empowerment Team
Microservices
In-Memory Data Grids
bpmPaaS
Private PaaS
Platform as a Service (PaaS)
Data Integration Tools
iPaaS
Application PaaS
IoT Integration
Full Life Cycle API Management
As of August 2019
expectations
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Plateau will be reached:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau
Source: Gartner
ID: 370015

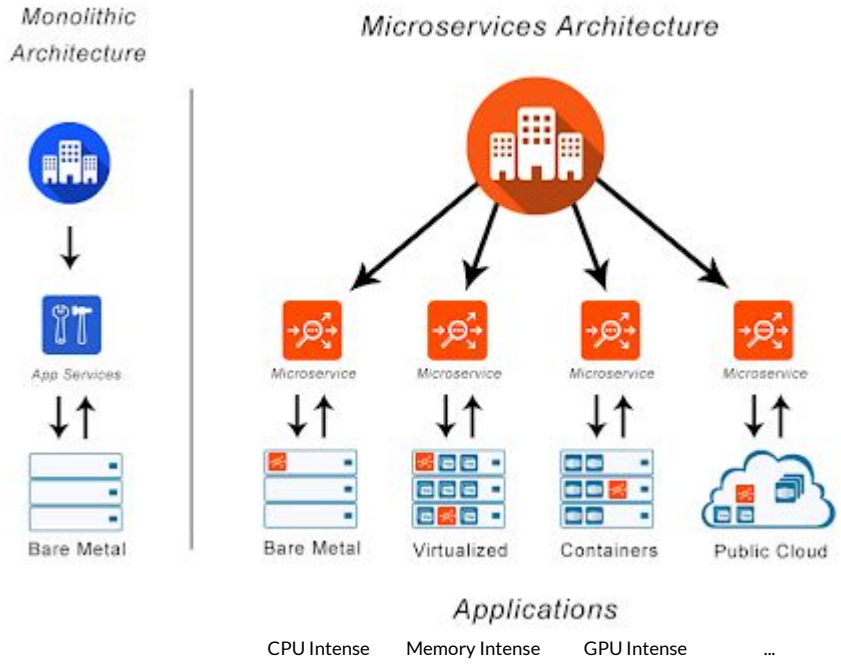
MICROSSERVIÇOS

Benefícios

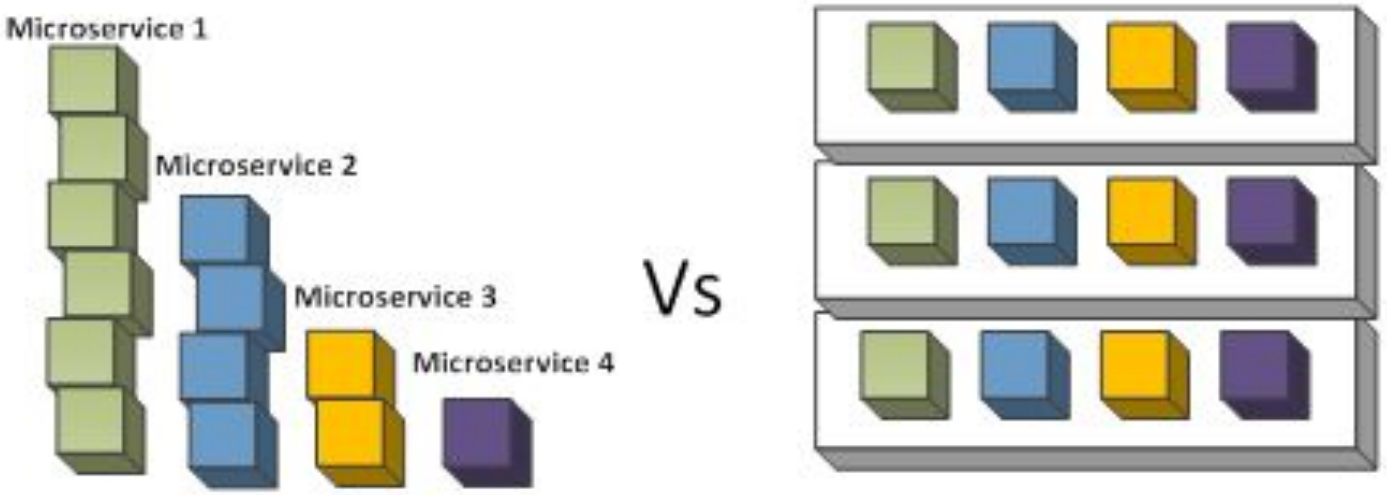
Ambiente Dev Light



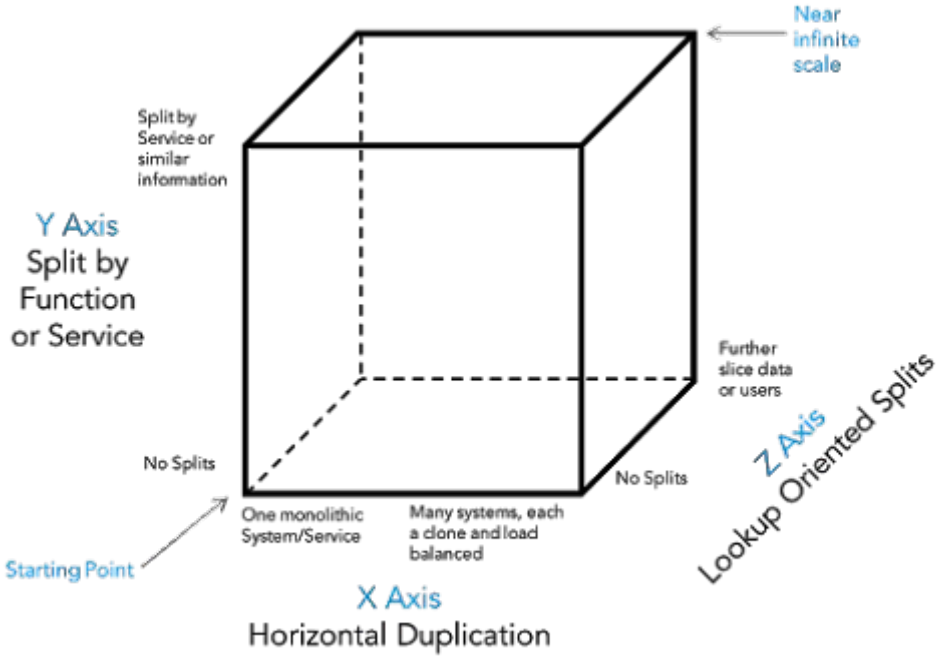
Uso + Adequado de Recursos



Escalabilidade



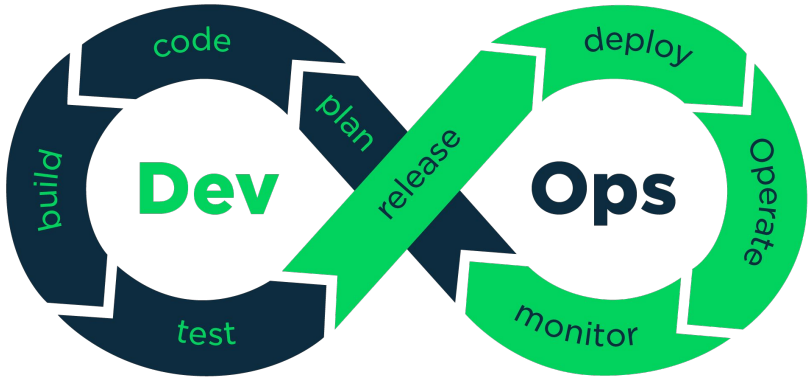
Escalabilidade

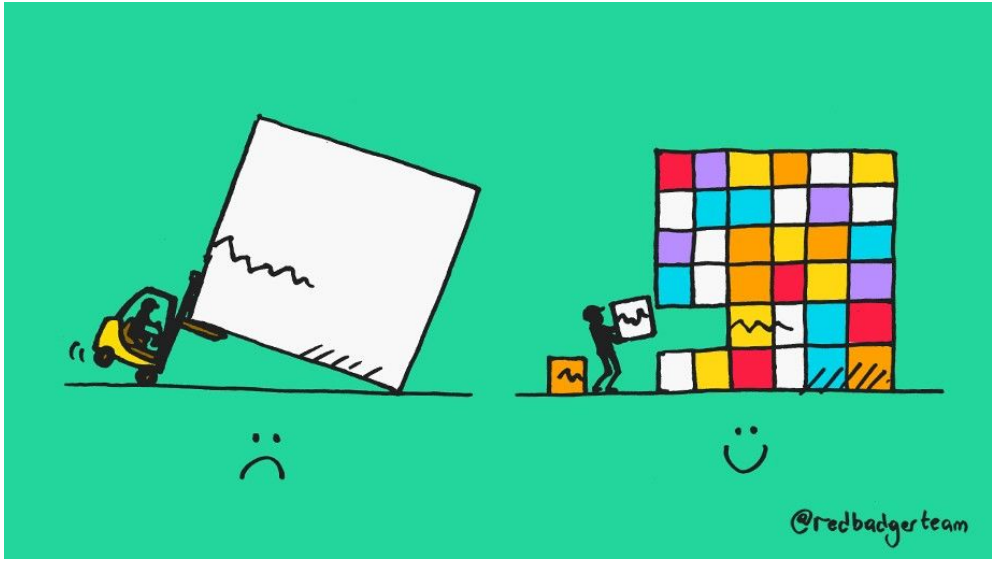


Fonte: livro “The Art of Scalability”

...

Agilidade e Disponibilidade





Pluralidade de Linguagem

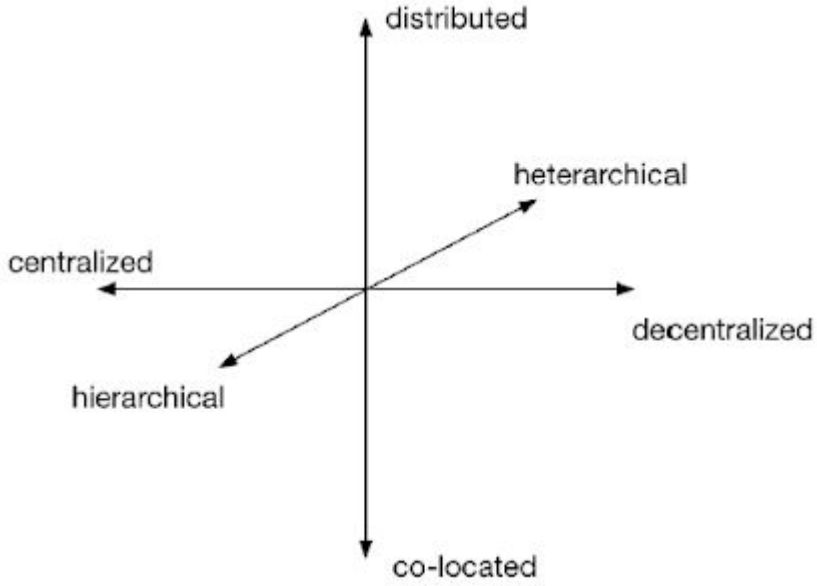
Desafios

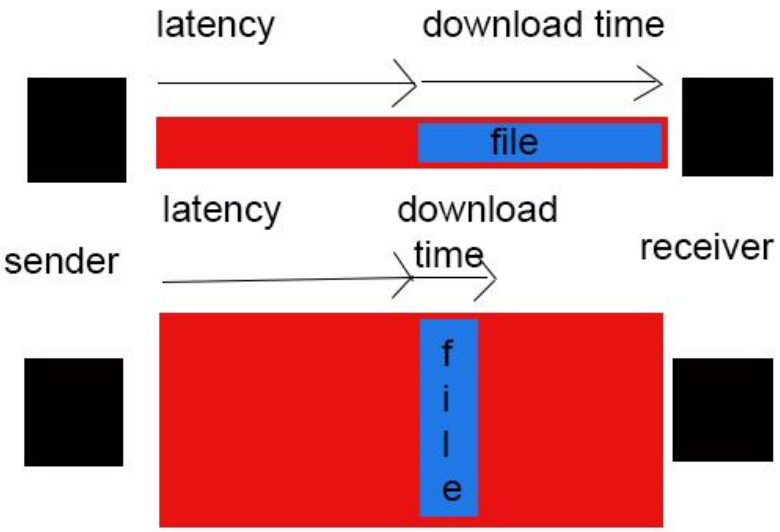
8 Falácias da Computação Distribuída

1. Rede é confiável
2. Latência é zero
3. Largura de banda é infinita
4. Rede segura
5. Topologia não muda
6. Há somente um administrador
7. Custo de transporte/tráfego é zero
8. Rede é homogênea

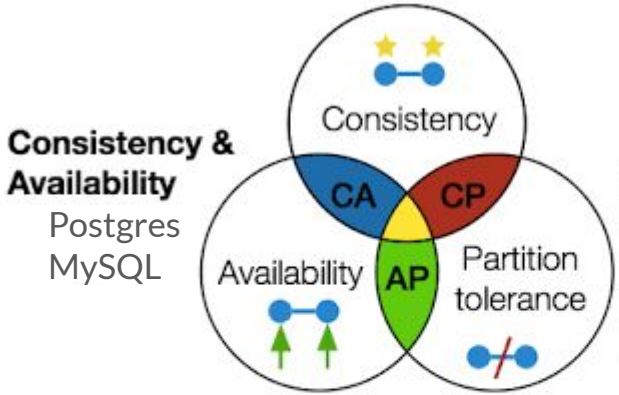
Em 1994 Peter Deutsch publicou uma lista de 7 premissas que arquitetos de software comumente assumem, mesmo que inconscientemente, e em 1997 James Gosling adicionou uma oitava premissa.

Sistema Distribuído





Teorema CAP



Consistency & Availability
Postgres
MySQL

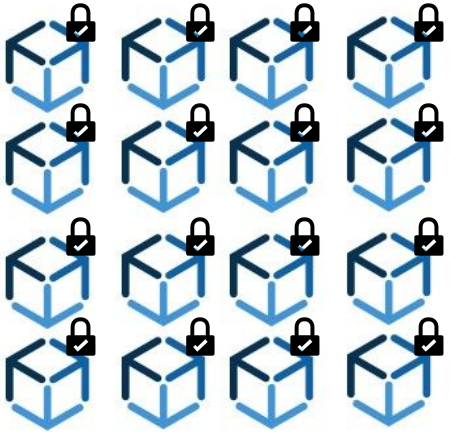
Consistency & Partition tolerance
MongoDB
Google BigTable

Availability & Partition tolerance
Cassandra
Voldemort

Fonte: https://medium.com/@lucascodejs/o-teorema-de-cap-utilizado-para-auxiliar-um-servidor-de-sistema-distribu%C3%ADdos-f73d6a4c5d9

https://www.objective.com.br/insights/nosql-saiba-quais-sao-os-tipos-e-suas-caracteristicas/

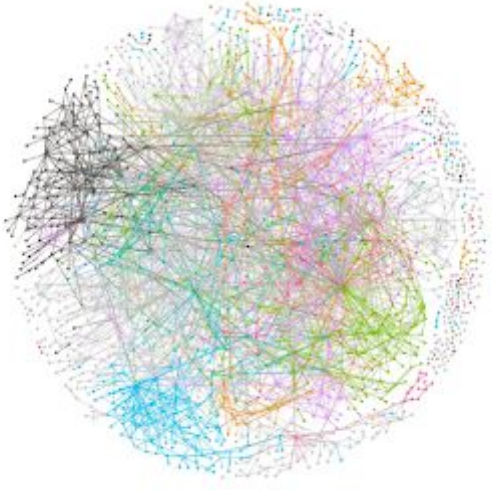
Segurança



Monitoramento

Log
- Debug
- Auditoria
- Monitoramento

Métricas de Aplicação
Métricas de Sistema
...



Fonte: https://monzo.com/blog/we-built-network-isolation-for-1-500-services

ACID -> BASE

Basically **A**vailable

Soft state

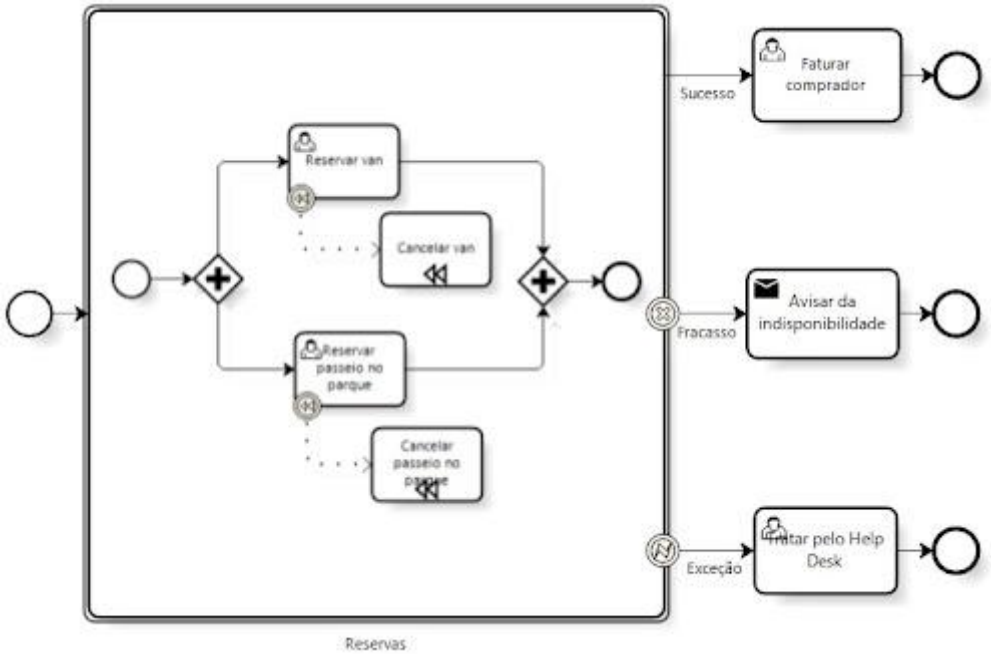
Eventually consistent

Disponibilidade e **Escalabilidade** são as maiores prioridades

Transações Distribuídas

Repensar Processos

Compensação de Tarefas



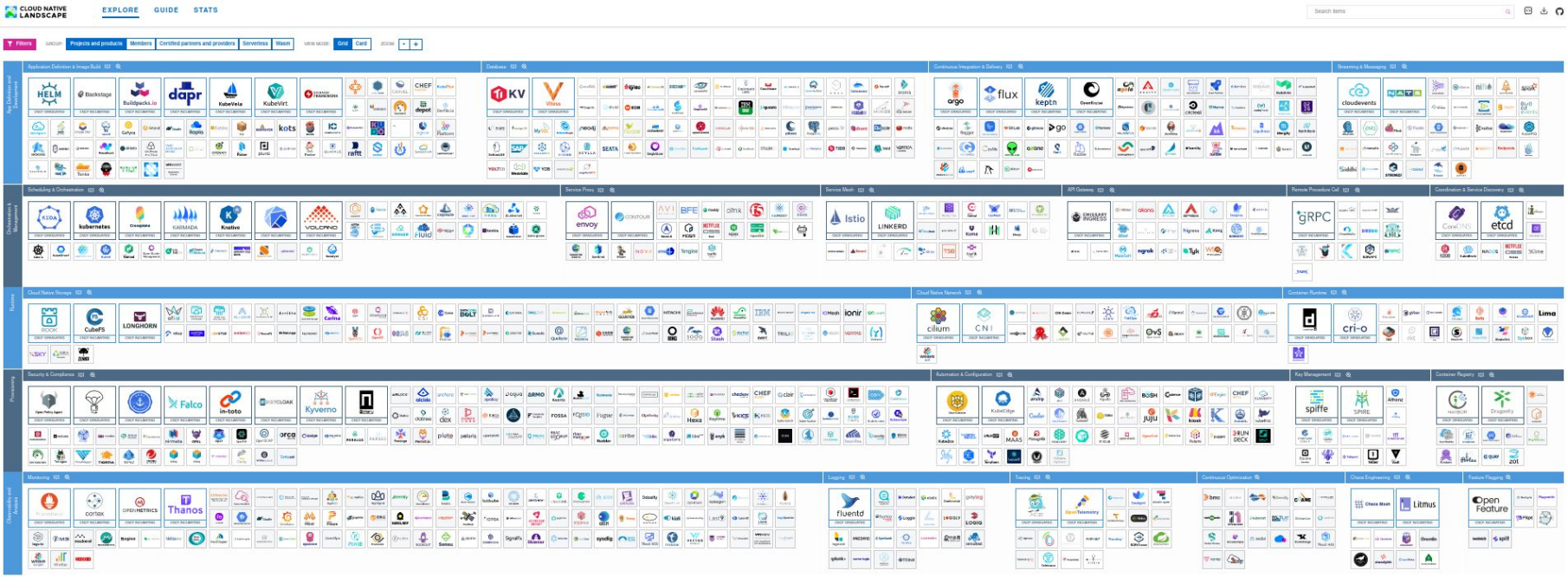
Versionamento

- URI versioning
- Header versioning
- Matriz de compatibilidade
- Versionamento para uma coleção de microsserviços

Serviço	Dependência e / ou Compatibilidade
A (1.0.0)	B (1.0.1) C (2.1.1)
A (1.1.0)	B (1.0.1) C (2.2.1) D (1.1.0)

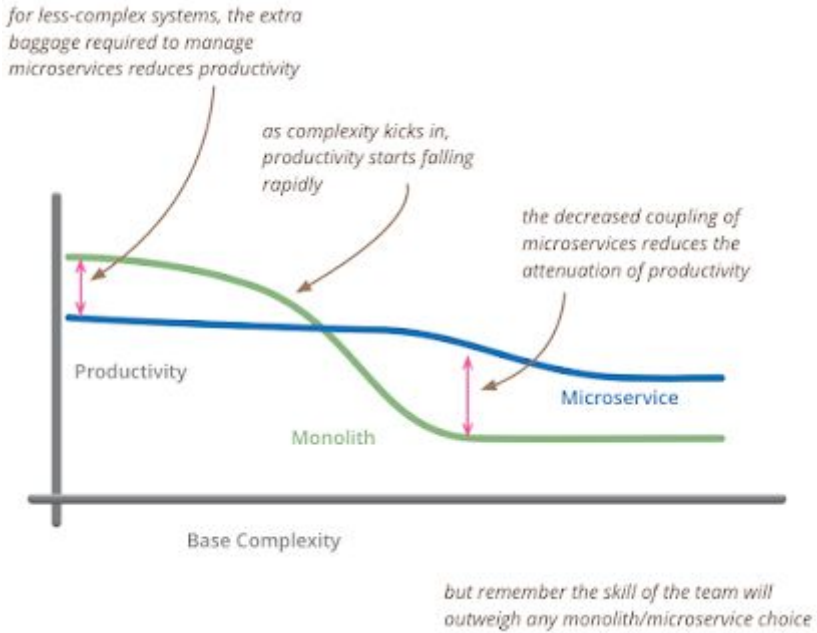
https://landscape.cncf.io/

Ferramentas de Apoio



Reflexões

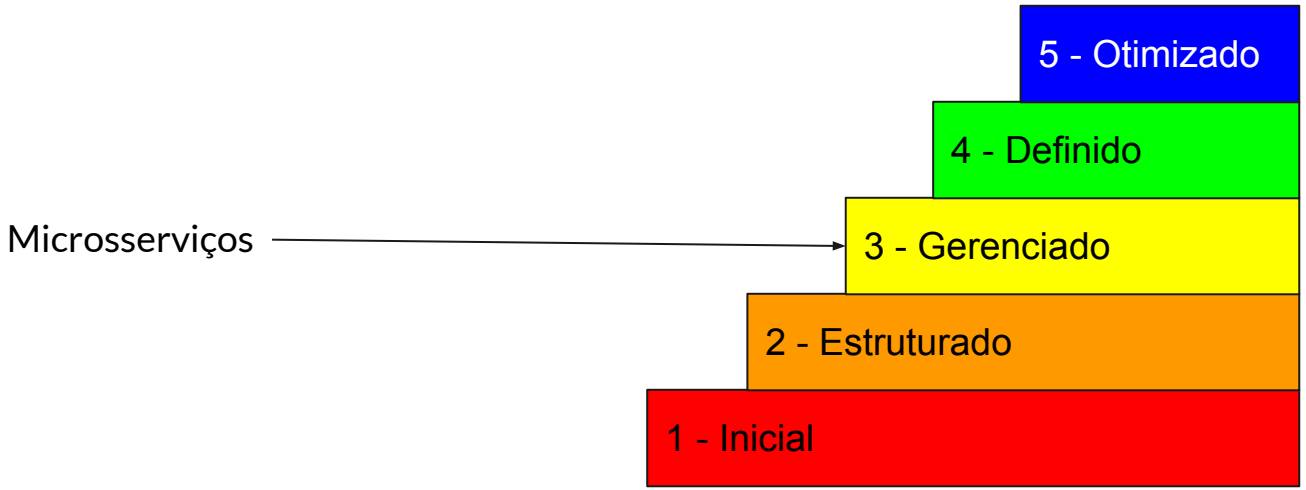
Produtividade



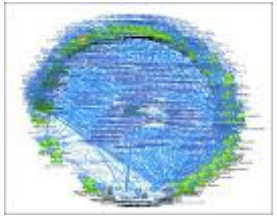
Complexidade do Software
=
Complexidade do domínio
+
Complexidade do legado
+
Complexidade da Solução Técnica

https://martinfowler.com/bliki/MicroservicePremium.html

Maturidade (DevOps)

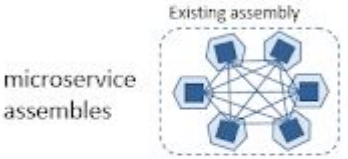


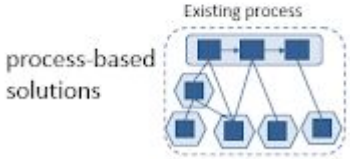
Modularidade



n * (n-1)/2

Escritório de Processos





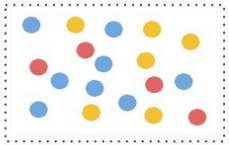
SA – Solution Architecture
CI – Continuous Integration

Domain-Driven Design starter modeling process

A starter process for beginners, not a rigid best-practice.
DDD is continuous, evolutionary and iterative design.

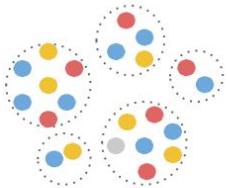
Align

Business model and user needs



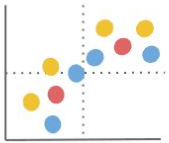
Discover

The domain visually & collaboratively



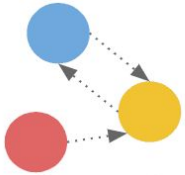
Decompose

The domain into sub-domains



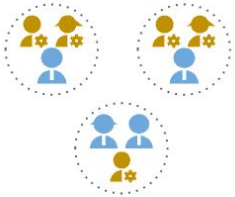
Strategize

Business differentiating core-domains



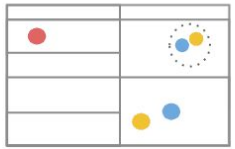
Connect

Sub-domains to form a loosely coupled architecture



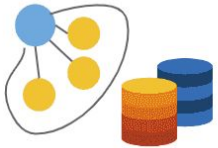
Organize

Teams around bounded contexts



Define

Roles & responsibilities for bounded contexts



Code

Your bounded context with tactical patterns

Domain Driven Designs

Strategic design

Ubiquitous Language

Domain

Sub-domain

Core domain

Domain model

Bounded Context

Context

Context Map

Generic Subdomain

Tactical patterns

Events

Entities

Respositories

Aggregates

Dependency Injection

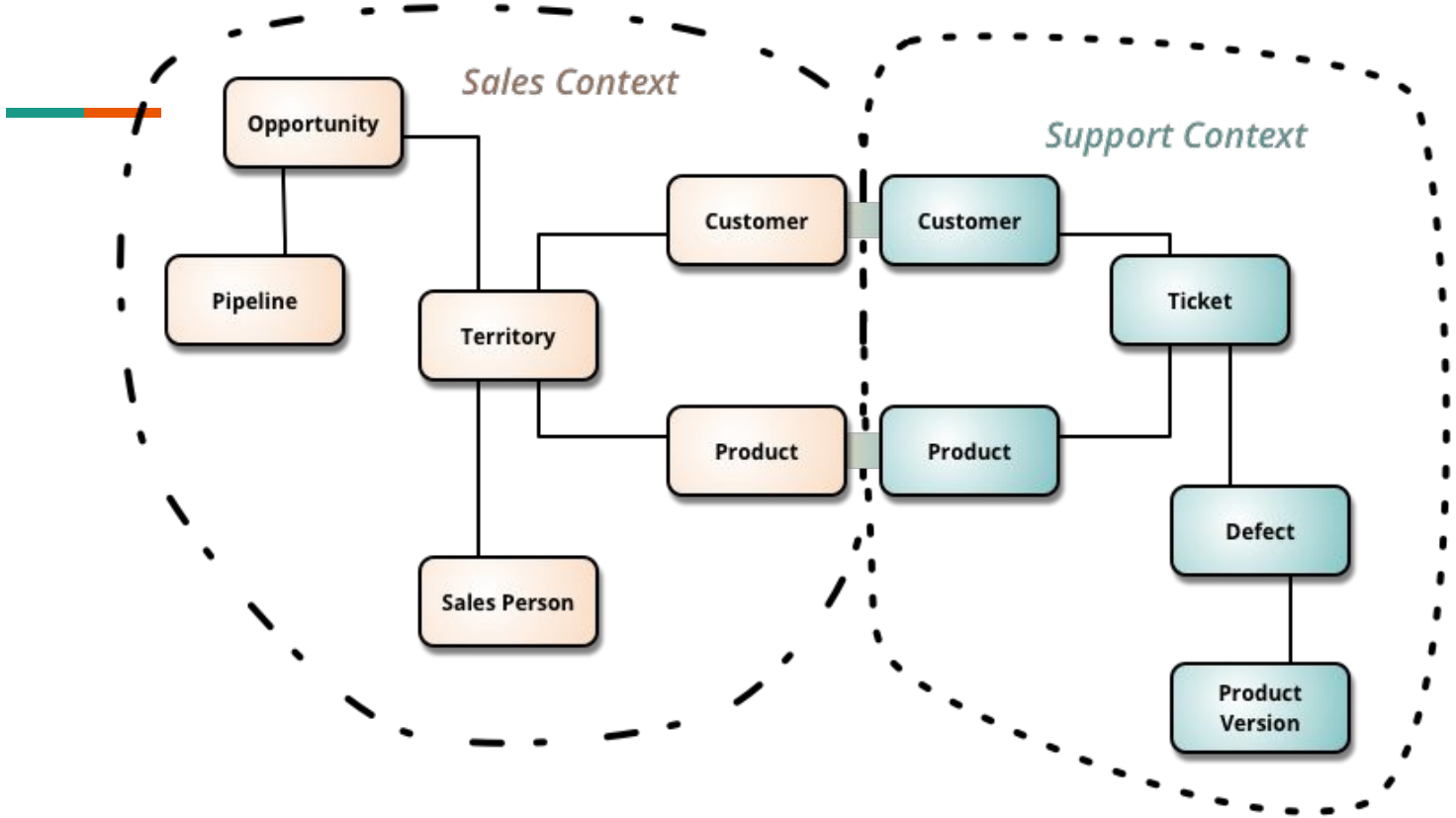
Factories

Layers

Services

Modules

DDD - Strategic Design - Bounded Context

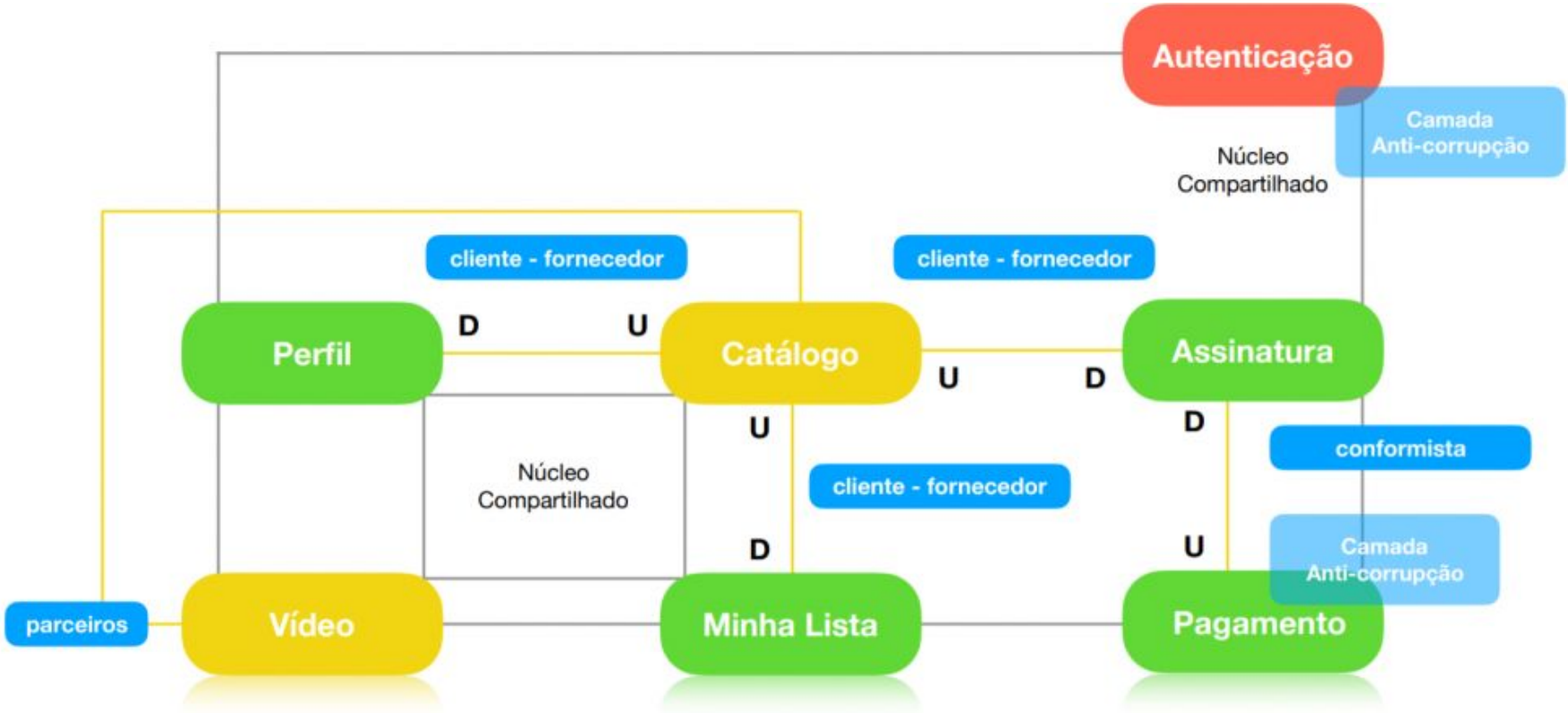


https://martinfowler.com/bliki/BoundedContext.html

DDD - Strategic Design - Tipos

https://fullcycle.com.br/domain-driven-design/

DDD - Strategic Design - Relações



https://fullcycle.com.br/domain-driven-design/

Name:

V5
github.com/ddd-crew/bounded-context-canvas

Purpose

What benefits does this context provide, and how does it provide them? Describe the purpose from a business perspective

Strategic Classification

Domain
- core
- supporting
- generic
- other?

Business Model
- revenue
- engagement
- compliance
- cost reduction

Evolution
- genesis
- custom built
- product
- commodity

Domain Roles

Role Types
- draft context
- execution context
- analysis context
- gateway context
- other

Inbound Communication

Collaborator

Messages

<Query>

<Command>

<Event>

Ubiquitous Language

Context-specific domain terminology

<Domain Term>
<definition>

Business Decisions

Key business rules, policies, and decisions

<Decision>

Outbound Communication

Messages

Collaborator

<Query>

<Command>

<Event>

Assumptions

Describe which currently unverified assumptions went into this bounded context design. Make those assumptions explicit by documenting them here

Verification Metrics

Describe metrics which can be used to (in)validate the current structure of this bounded context?

Open Questions

Escritório de Processos

Identificar processos, domínios e serviços

Mapear processos, domínios e serviços

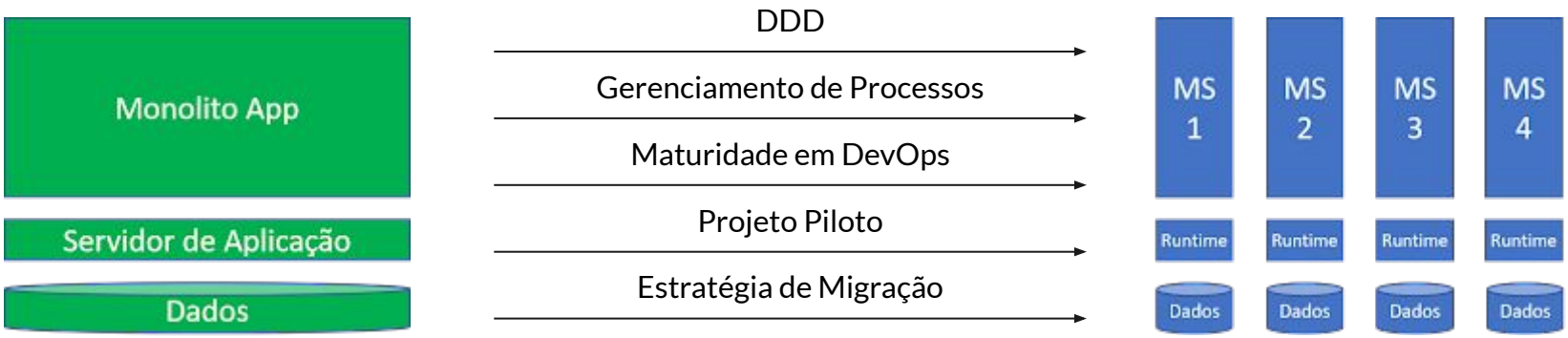
Promover Integração entre diferentes áreas (linguagem ubíqua)

Promover Inovação

Promover Adequações (otimizações)

Definir e acompanhar métricas

Migração


Monolito App
Servidor de Aplicação
Dados
DDD
Gerenciamento de Processos
Maturidade em DevOps
Projeto Piloto
Estratégia de Migração
MS 1
MS 2
MS 3
MS 4
Runtime
Runtime
Runtime
Runtime
Dados
Dados
Dados
Dados

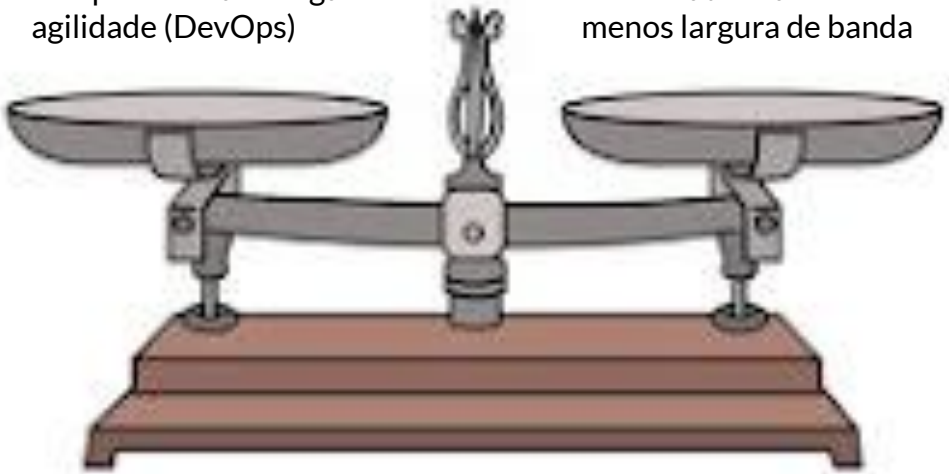
O que pesa mais?

Microserviço

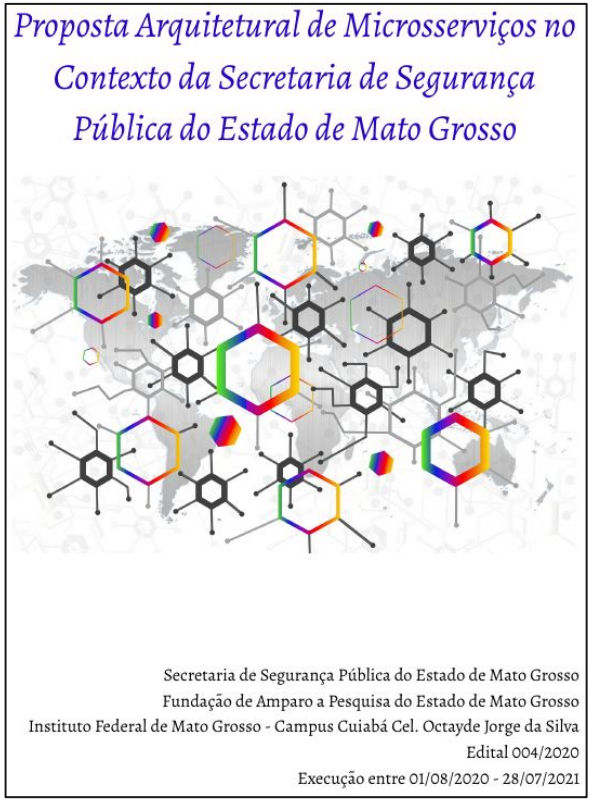
reuso/compartilhamento
maior disponibilidade
maior escalabilidade
equipes menores
independência do legado
agilidade (DevOps)

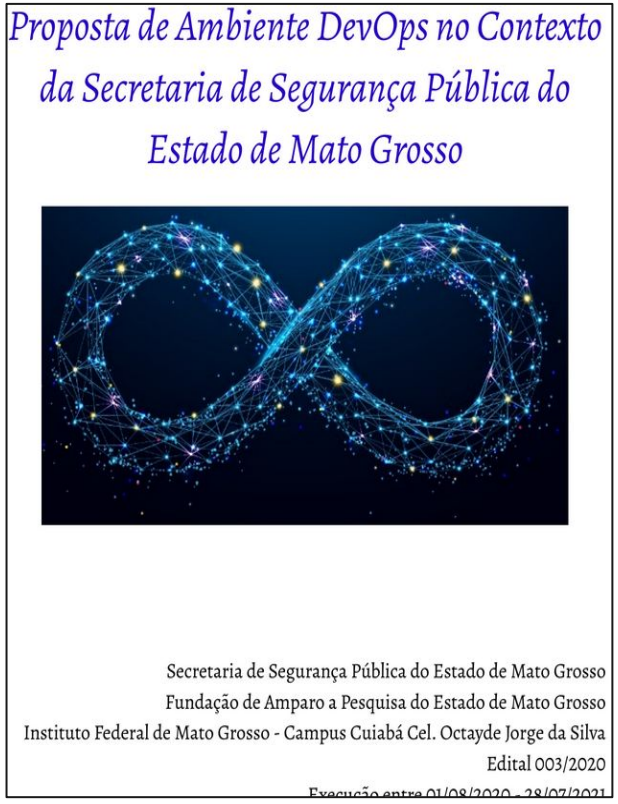
Monolito

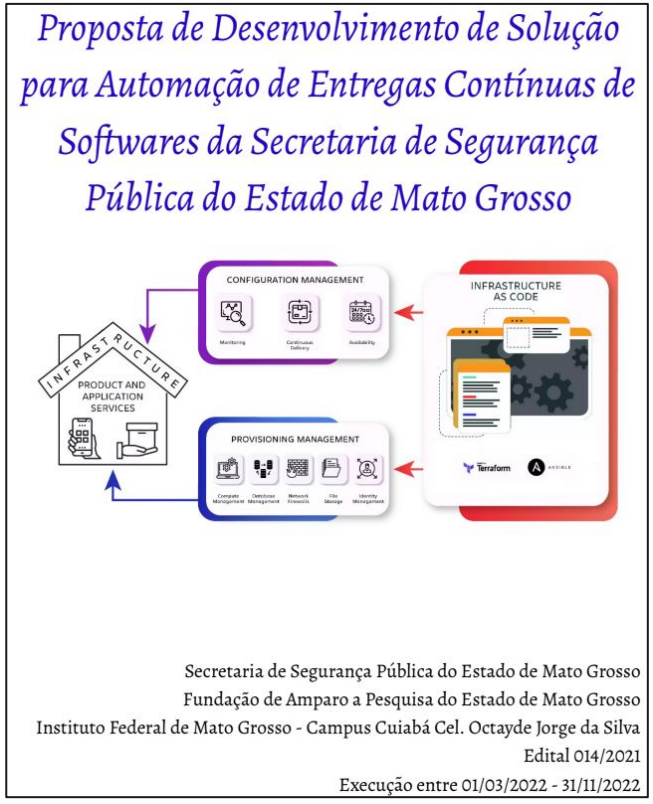
menor complexidade
facilidade de execução
facilidade de monitoramento
ACID
menor latência
menos largura de banda



PROTÓTIPO FUNCIONAL







ISBN: 978-65-00-70036-7

ISBN: 978-65-00-70037-4

ISBN: 978-65-00-70035-0

INSTITUTO FEDERAL
Mato Grosso

Campus Cuiabá
Cel. Octayde Jorge da Silva

3rd International Conference on Electrical, Communication and Computer Engineering

12 – 13 JUNE, 2021, KUALA LUMPUR, MALAYSIA

Proc. of the 3rd International Conference on Electrical, Communication and Computer Engineering (ICECCE), 12-13 June 2021, Kuala Lumpur, Malaysia

Monolithic to Microservices Migration Strategy in Public Security Secretariat of Mato Grosso

João Paulo Delgado Pires
Software Engineering Applied Research Group (ESA)
Federal Education Institute of Mato Grosso (IFMT)
Cuiabá-MT, Brazil
joao.pires@ifmt.edu.br

Adriano Nerys Araújo Souza
Computer Networks Research Group (GPRC)
Federal Education Institute of Mato Grosso (IFMT)
Cuiabá-MT, Brazil
ah.dsouza@gmail.com

Evandro Cleas Freiberger
Software Engineering Applied Research Group (ESA)
Federal Education Institute of Mato Grosso (IFMT)
Cuiabá-MT, Brazil
evandro.freiberger@cba.ifmt.edu.br

Tiago de Almeida Lacerda
Software Engineering Applied Research Group (ESA)
Federal Education Institute of Mato Grosso (IFMT)
Cuiabá-MT, Brazil
tiago.lacerda@cba.ifmt.edu.br

Public Safety Secretariat of Mato Grosso Microservice Environment

Abstract

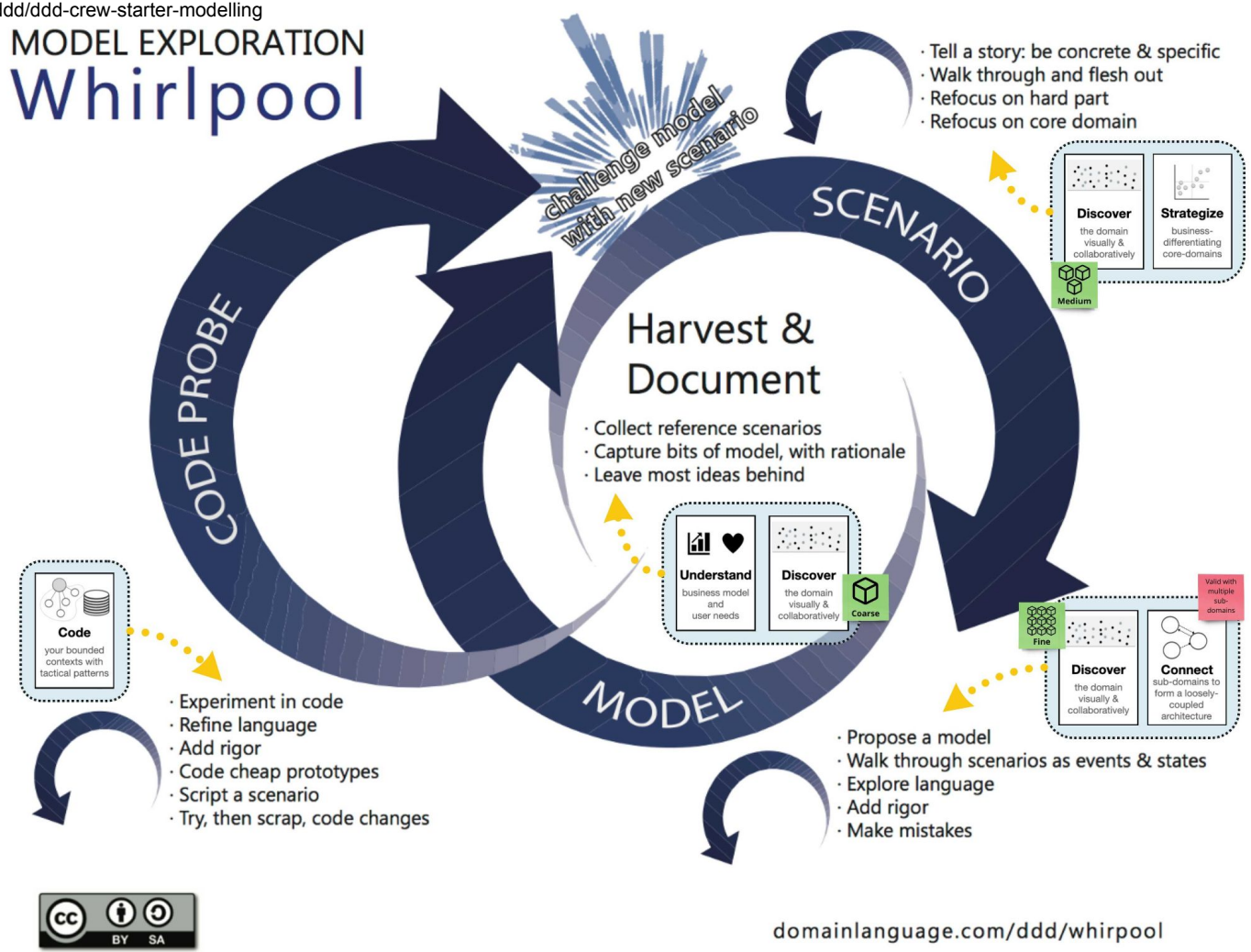
This paper presents the microservice environment of the Public Safety Secretariat of Mato Grosso (SESP-MT) which was conceived to allow a migration process from SESP-MT monoliths and to absorb new organizational agile requirements. Despite the hype of microservice oriented architecture, it's an architectural style, with some general principles and as the nature of

used to understand the organization of SESP-MT systems and the MoSCoW requirements prioritization technique was used for sprint planning.

From the second sprint on there were monthly deliveries of executable products and a tested environment with a demo presentation followed by the next sprint planning.

The activities addressed the following challenges:

https://virtualddd.com/learning-ddd/ddd-crew-starter-modelling

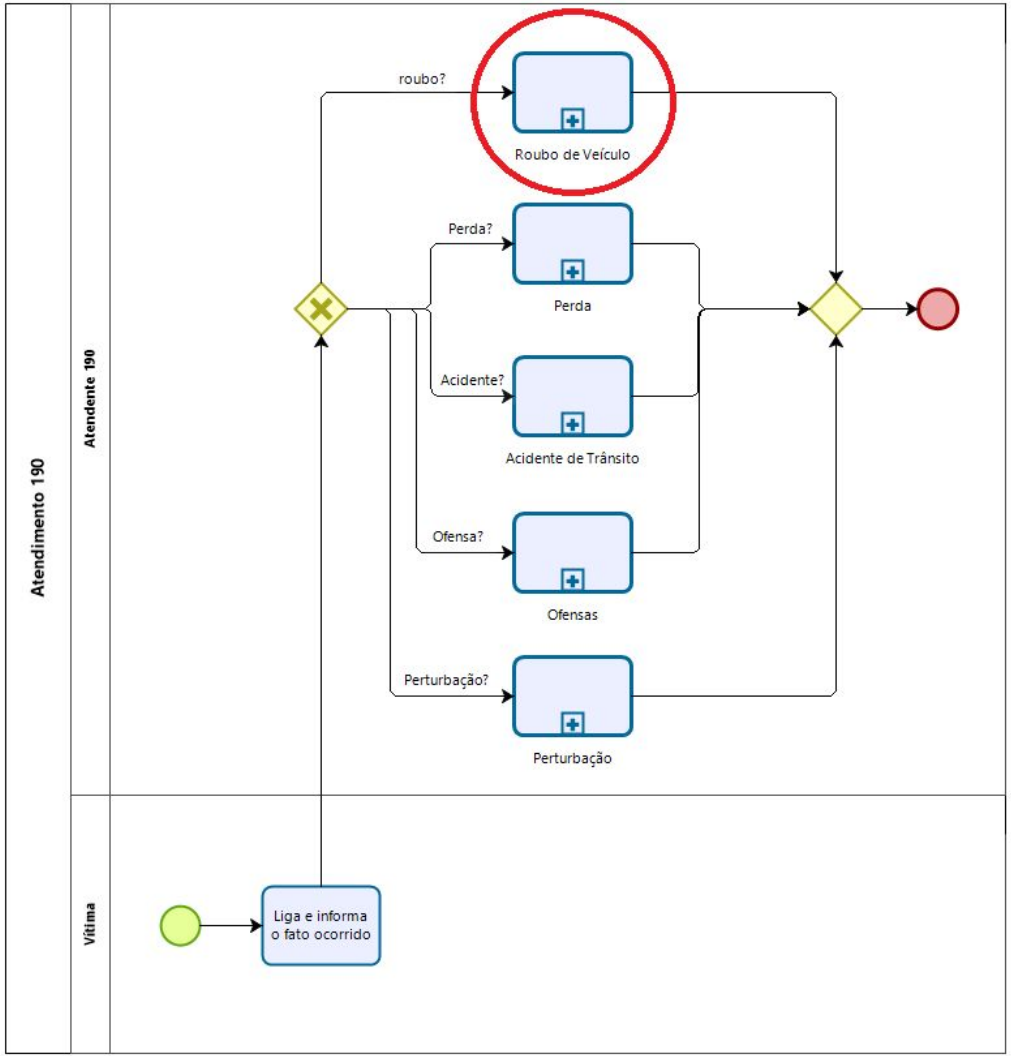


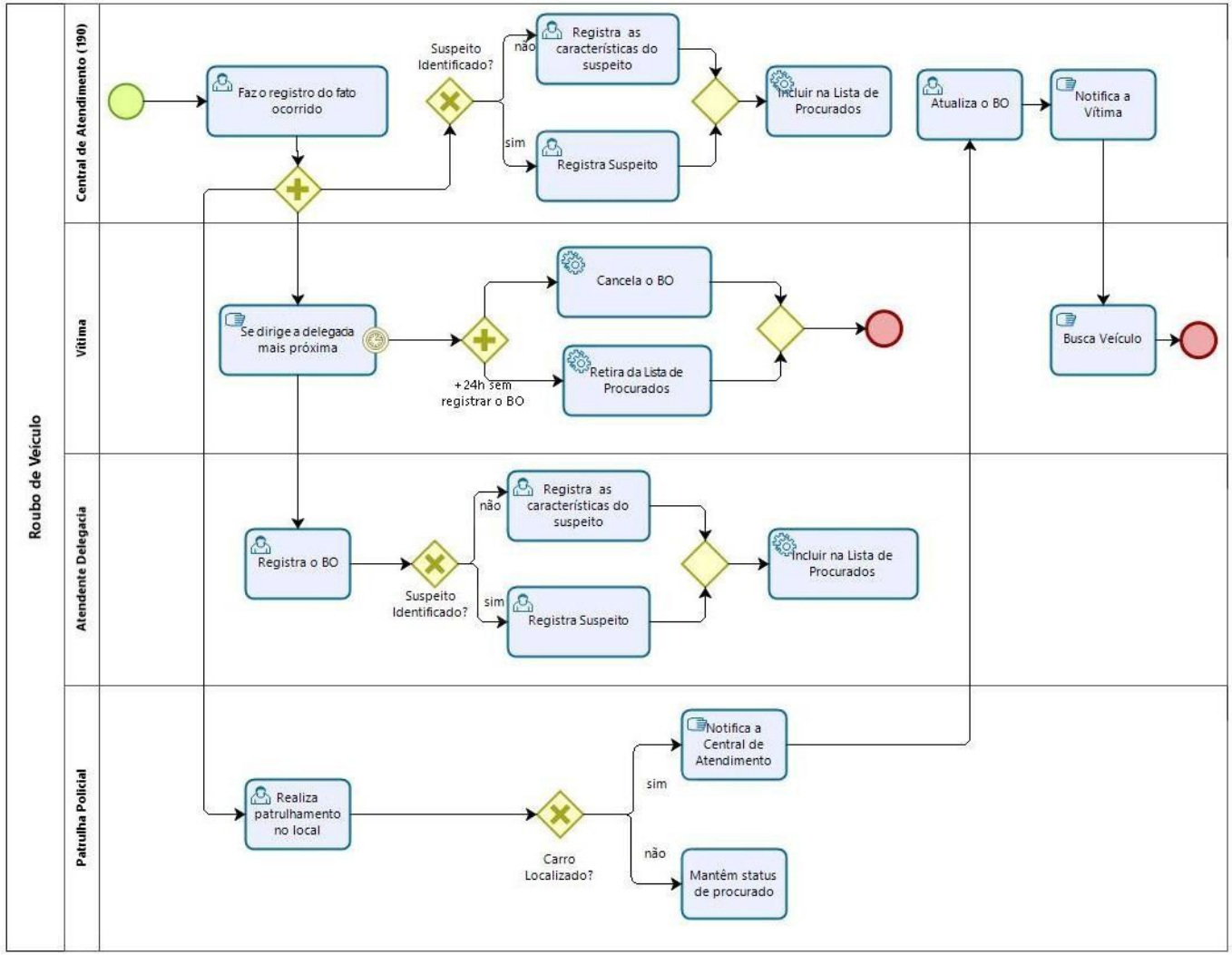
domainlanguage.com/ddd/whirlpool

Versões:

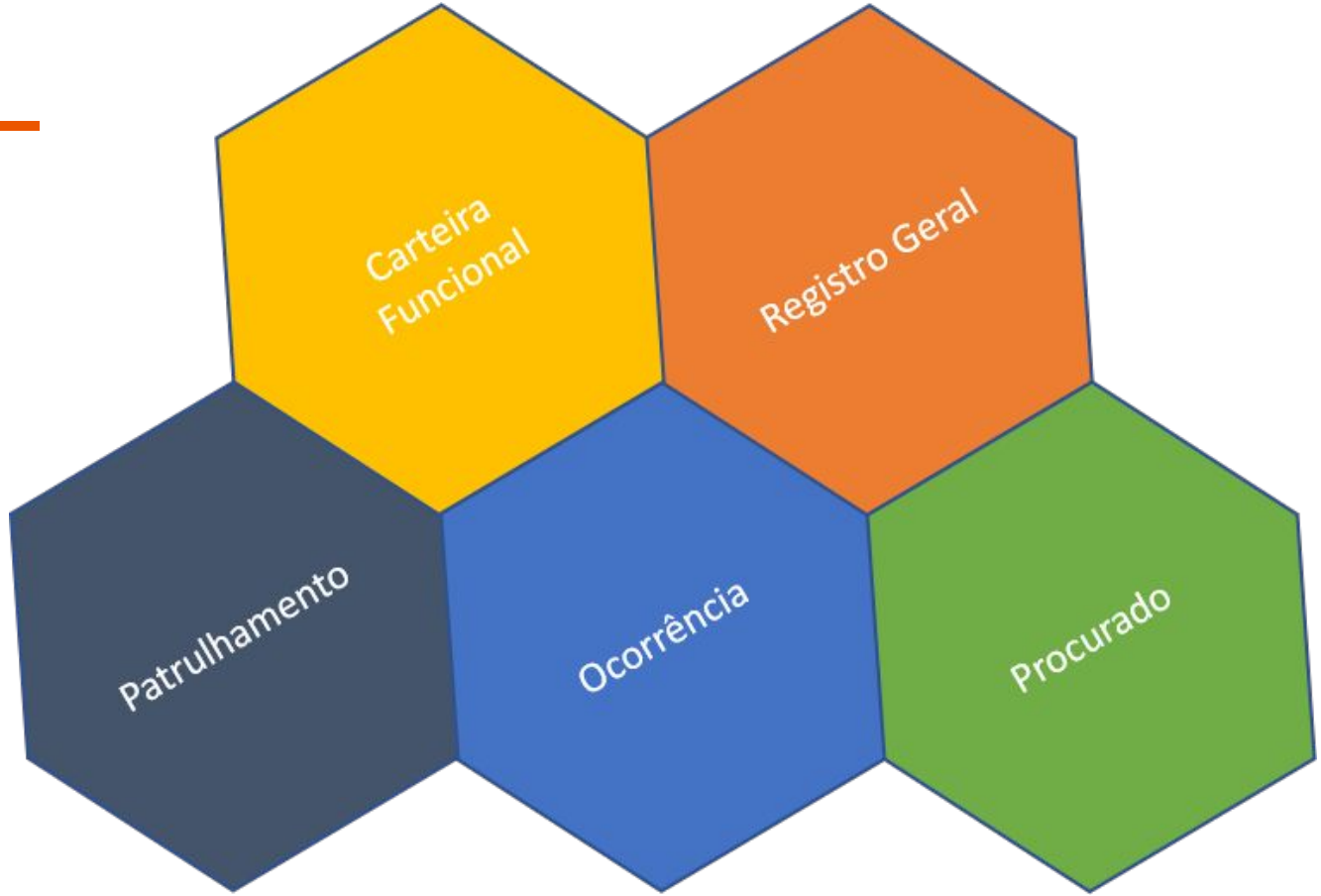
- v0.1 - Cenário e Ambiente Dev
- v0.2 - Autenticação
- v0.3 - Autorização
- v0.4 - Persistência
- v0.5 - Mensageria
- v0.6 - Log
- v0.7 - Monitoramento
- v0.8 - Transações Distribuídas
- v0.9 - Documentação Funcional
- v1.0 - Versão homologada

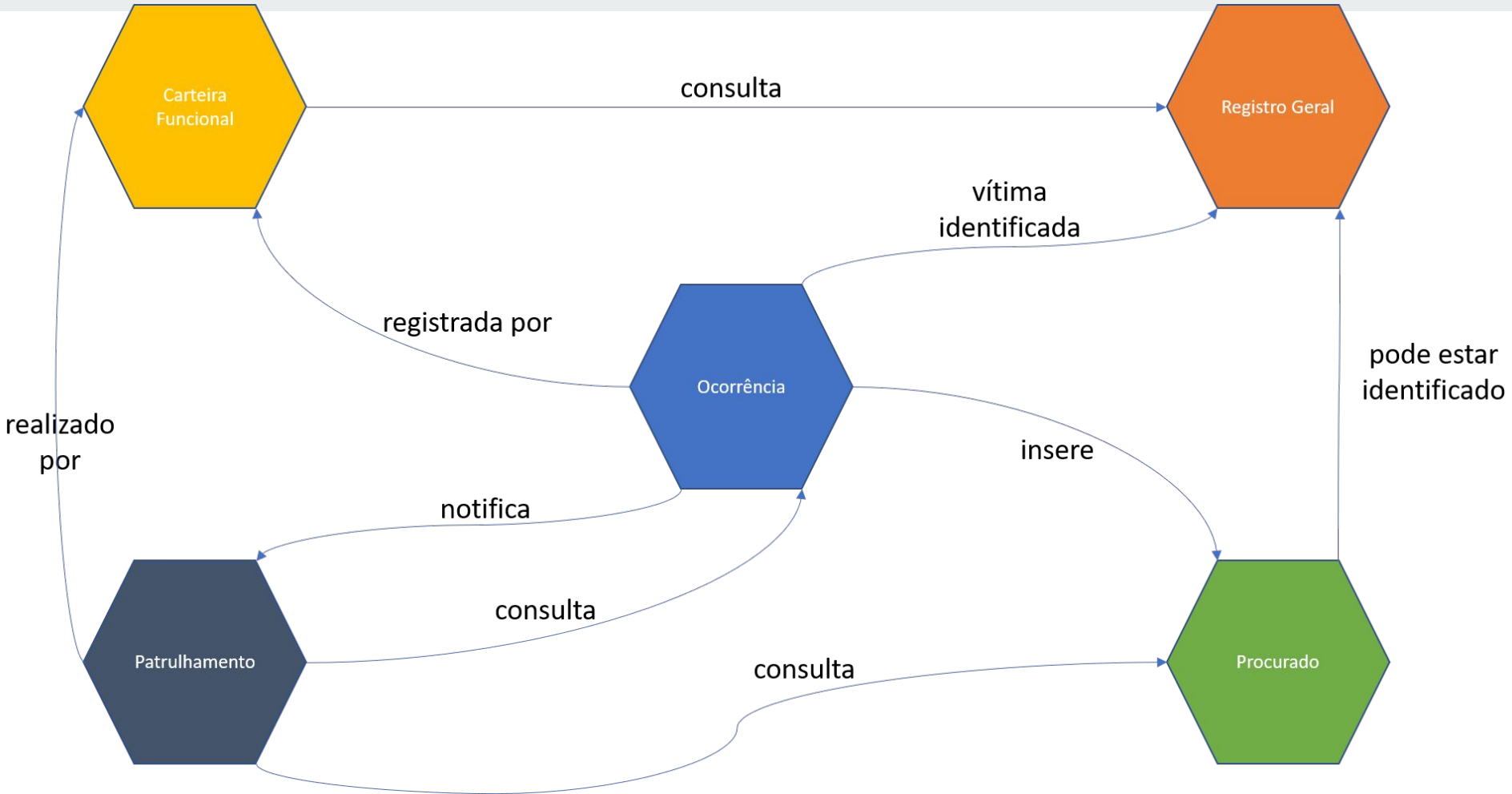
CENÁRIO


Atendimento 190
Atendente 190
Vítima
roubo?
Roubo de Veículo
Perda?
Perda
Acidente?
Acidente de Trânsito
Ofensa?
Ofensas
Perturbação?
Perturbação
Liga e informa o fato ocorrido


Roubo de Veículo
Central de Atendimento (190)
Faz o registro do fato ocorrido
Suspeito Identificado?
não
Registra as características do suspeito
sim
Registra Suspeito
Incluir na Lista de Procurados
Atualiza o BO
Notifica a Vítima
Vítima
Se dirige a delegacia mais próxima
+24h sem registrar o BO
Cancela o BO
Retira da Lista de Procurados
Busca Veículo
Atendente Delegacia
Registra o BO
Suspeito Identificado?
não
Registra as características do suspeito
sim
Registra Suspeito
Incluir na Lista de Procurados
Patrulha Policial
Realiza patrulhamento no local
Carro Localizado?
sim
Notifica a Central de Atendimento
não
Mantém status de procurado

PROJETO


Carteira Funcional
Registro Geral
Patrulhamento
Ocorrência
Procurado


Carteira Funcional
consulta
Registro Geral
vítima identificada
registrada por
Ocorrência
pode estar identificado
realizado por
insere
notifica
consulta
Patrulhamento
Procurado
consulta

AMBIENTE

10 repositórios Git
22 contêineres

gerenciamento-roubo

gerenciamento-ocorrencia

gerenciamento-patrulha

gerenciamento-procurado

registro-geral

carteira-funcional

autenticacao

ldap-admin

redis

keycloak

db-apps

rabbitmq

graylog

alertmanager

prometheus

grafana

springbootadmin

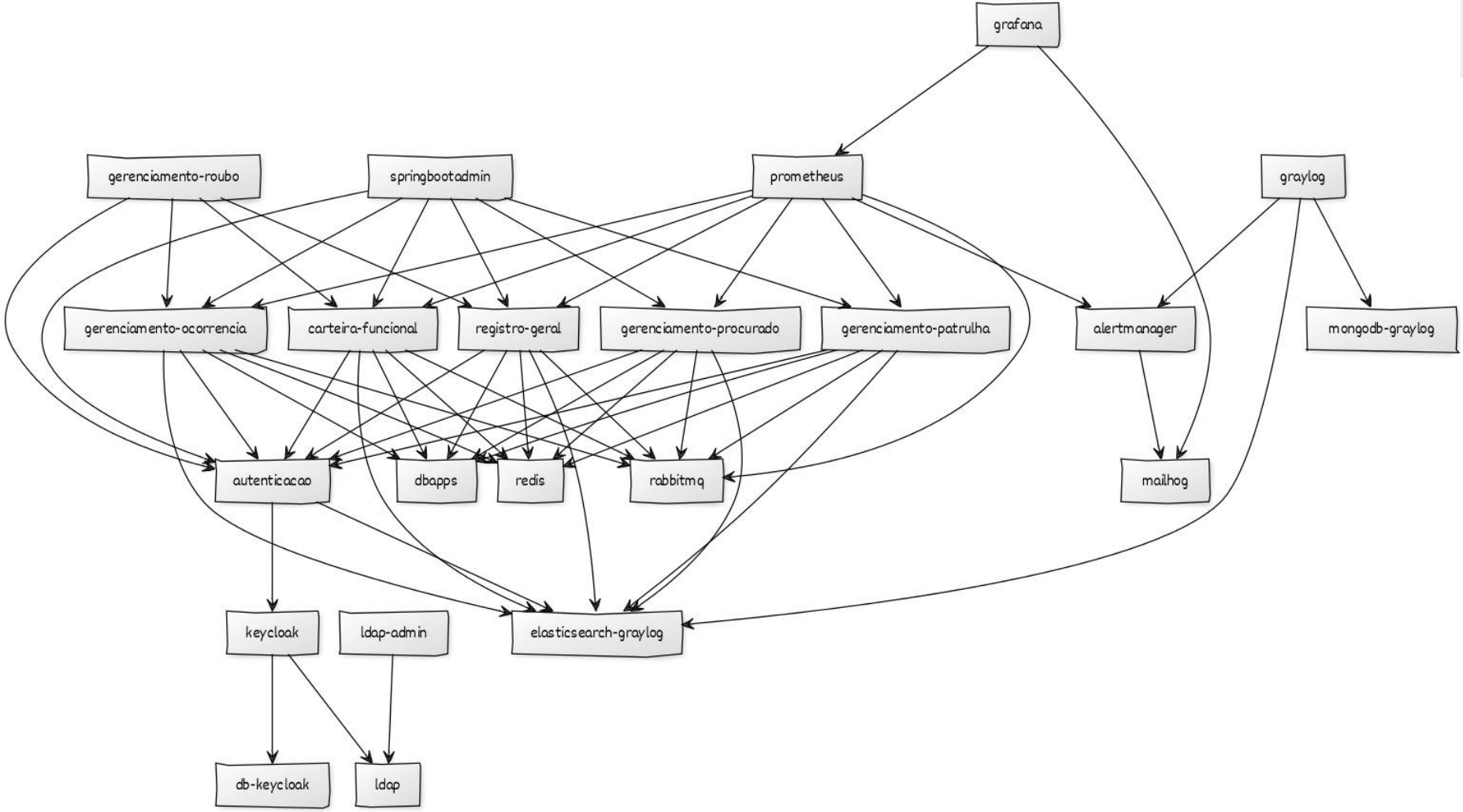
mailhog

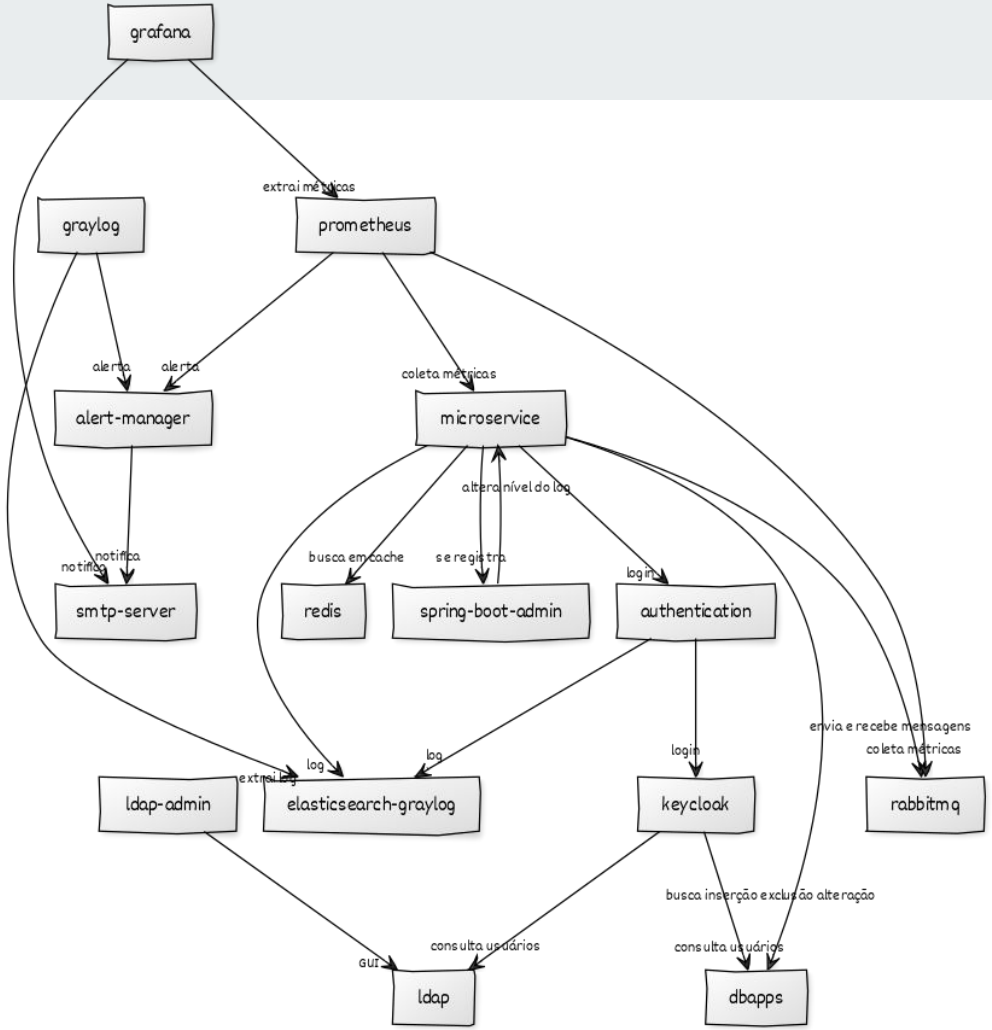
ldap

db-keycloak

mongodb-graylog

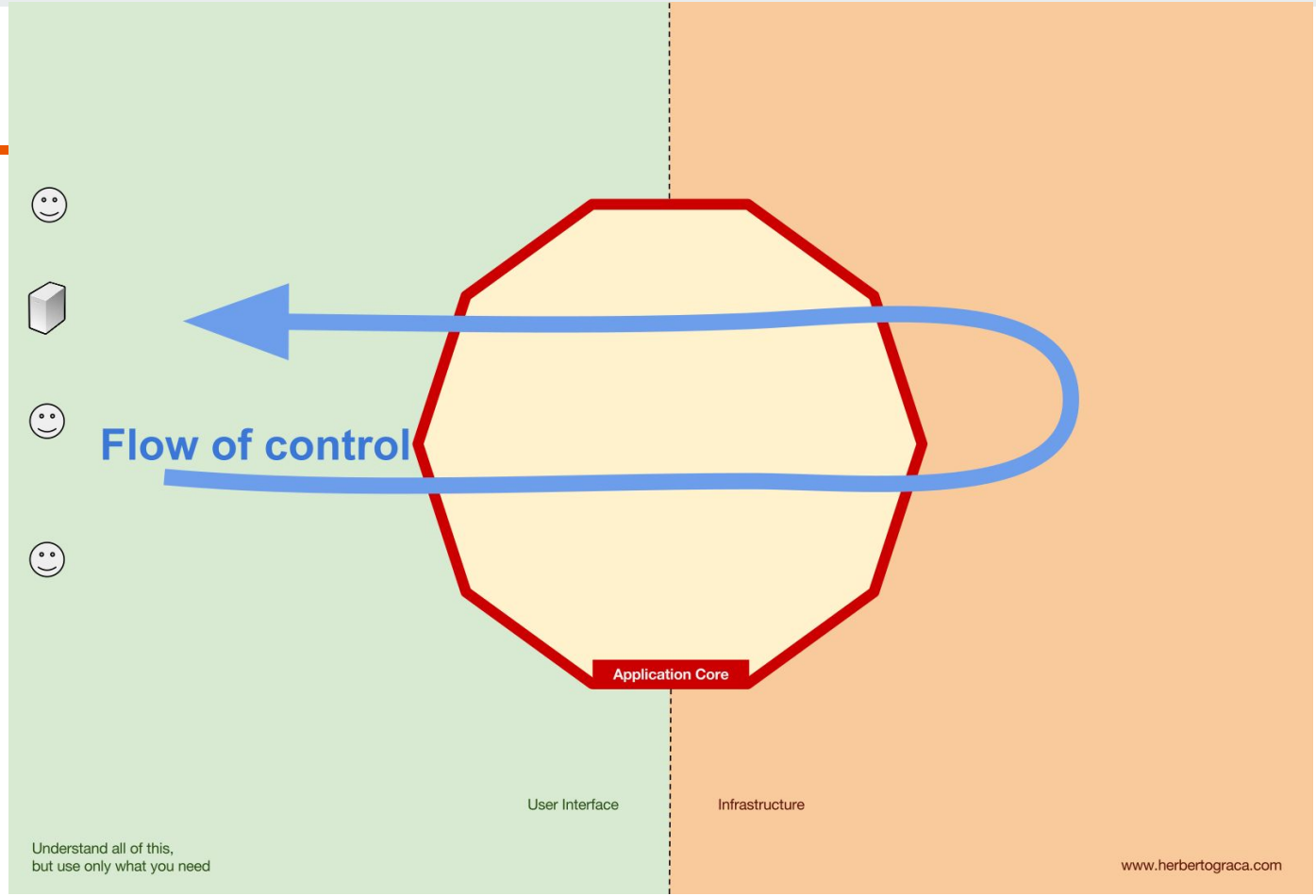
elasticsearch-graylog


grafana
gerenciamento-roubo
springbootadmin
prometheus
graylog
gerenciamento-ocorrencia
carteira-funcional
registro-geral
gerenciamento-procurado
gerenciamento-patrulha
alertmanager
mongodb-graylog
autenticacao
dbapps
redis
rabbitmq
mailhog
keycloak
ldap-admin
elasticsearch-graylog
db-keycloak
ldap
CREATED WITH YUML


grafana
graylog
extrai métricas
prometheus
alerta
alerta
alert-manager
coleta métricas
microservice
altera nível do log
notifica
notifica
smtp-server
busca em cache
redis
se registra
spring-boot-admin
login
authentication
envia e recebe mensagens
coleta métricas
log
extrai log
log
ldap-admin
elasticsearch-graylog
login
keycloak
rabbitmq
busca inserção exclusão alteração
consulta usuários
GUI
consulta usuários
ldap
dbapps
CREATED WITH YUML

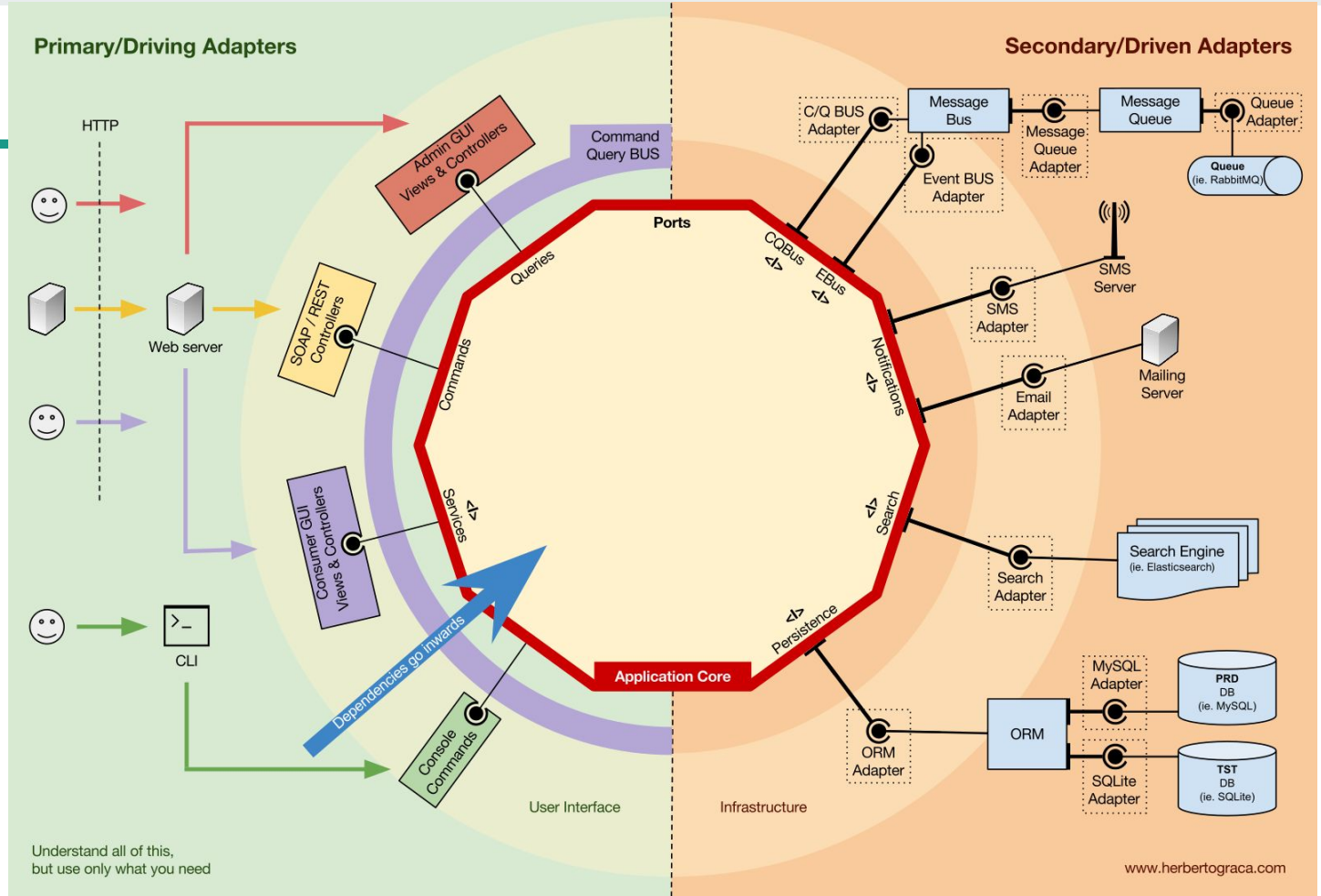
Arquitetura Base

Baseado na Arquitetura: Ports and Adapters

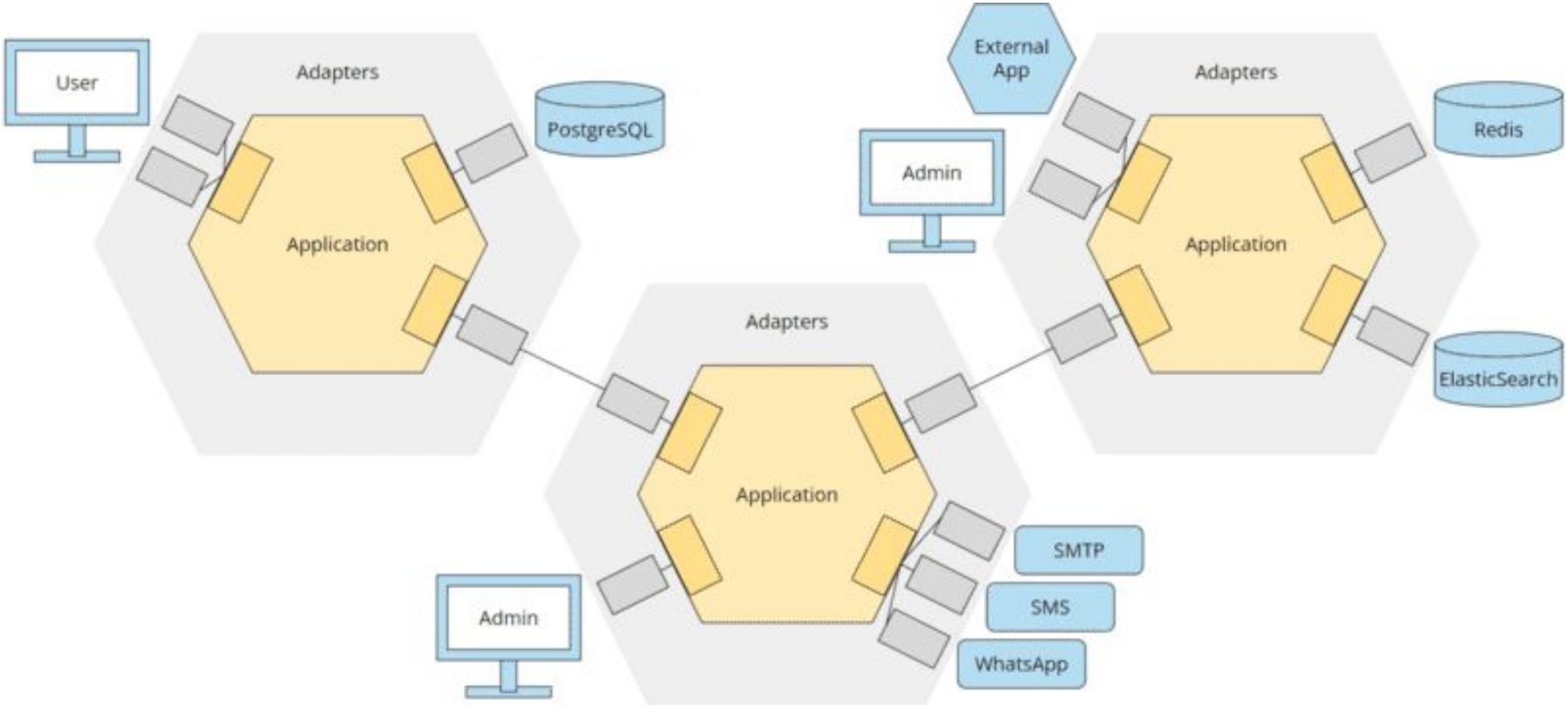


https://herbertograca.com/2017/11/16/explicit-architecture-01-ddd-hexagonal-onion-clean-cqrs-how-i-put-it-all-together/

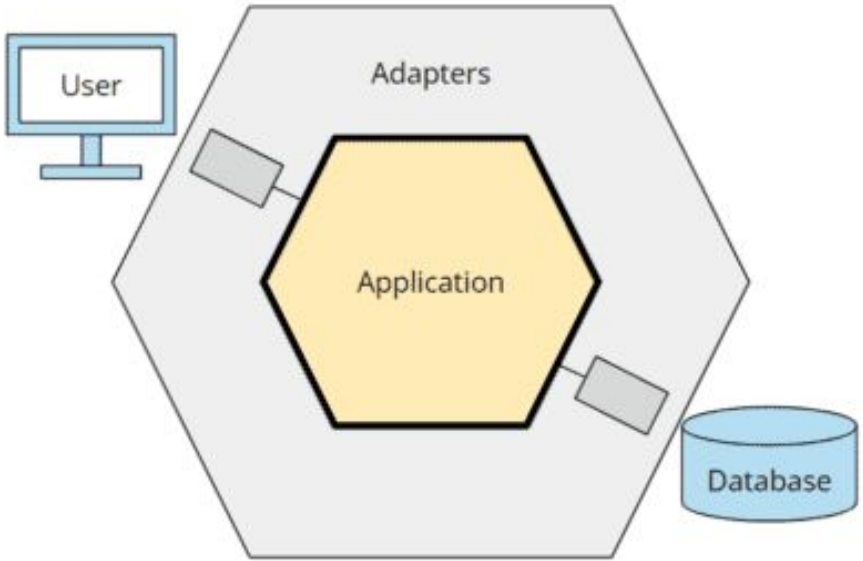
Adaptadores Primários e Secundários

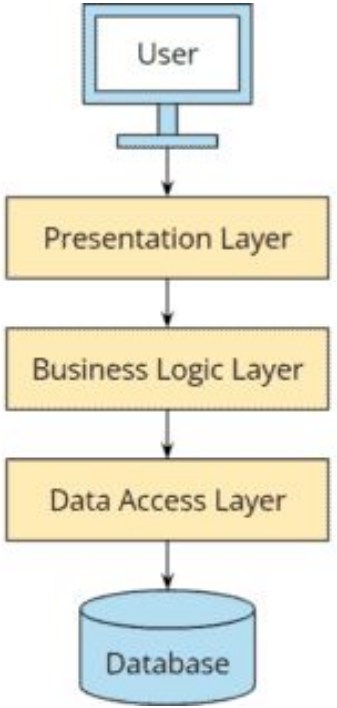


Arquitetura Hexagonal (Application é abstrata)

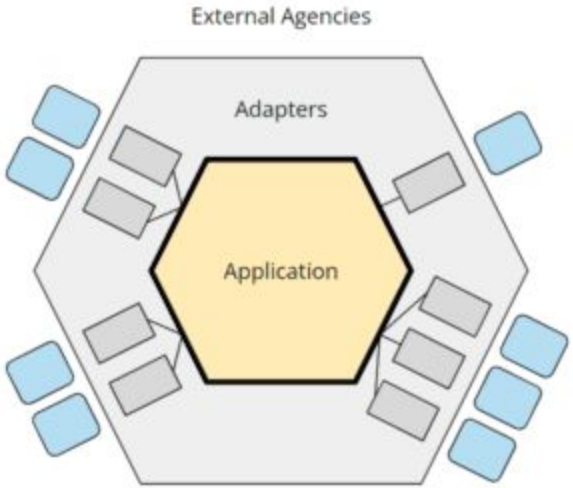


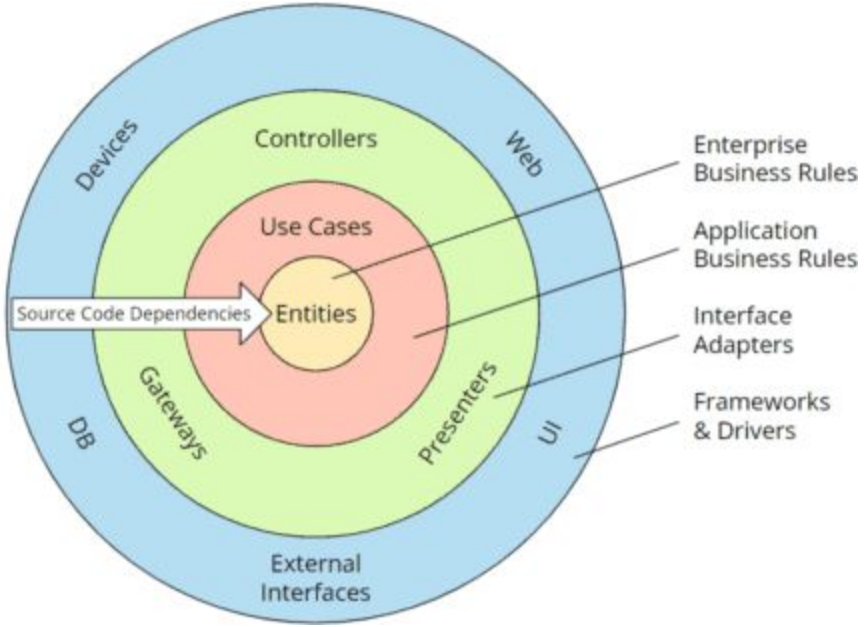
Hexagonal vs em Camadas (Layered)



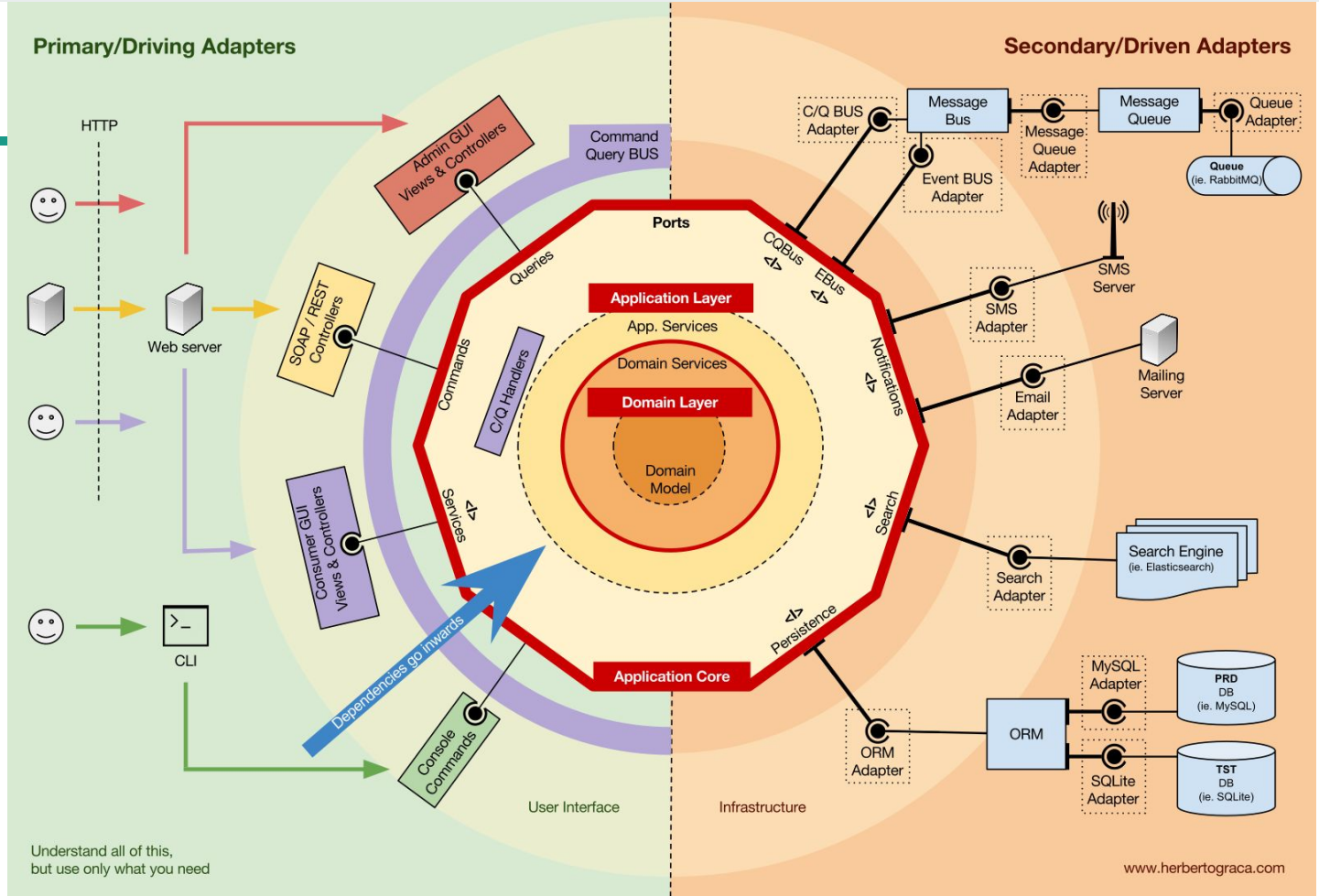


Hexagonal vs Arquitetura Clean

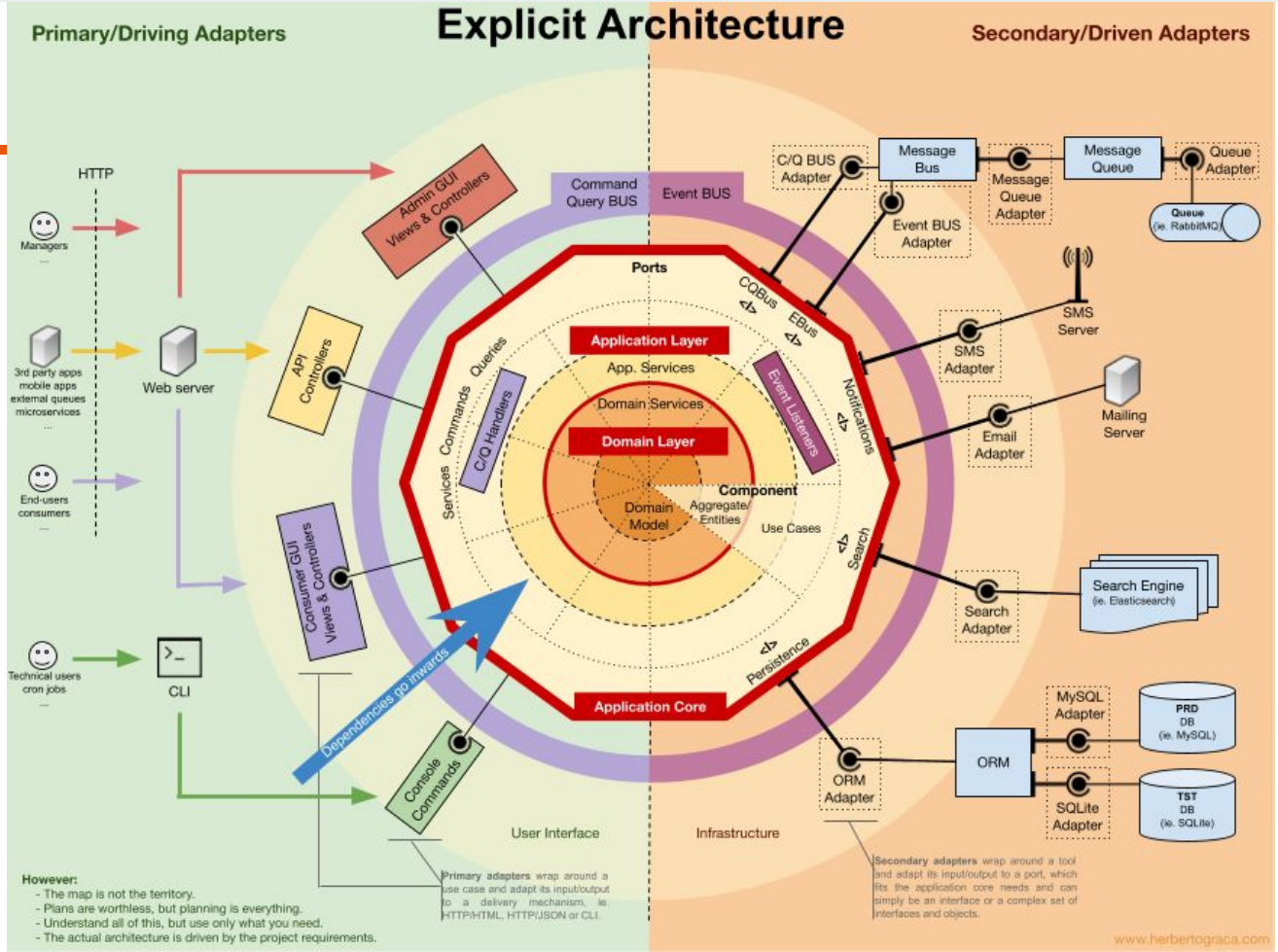


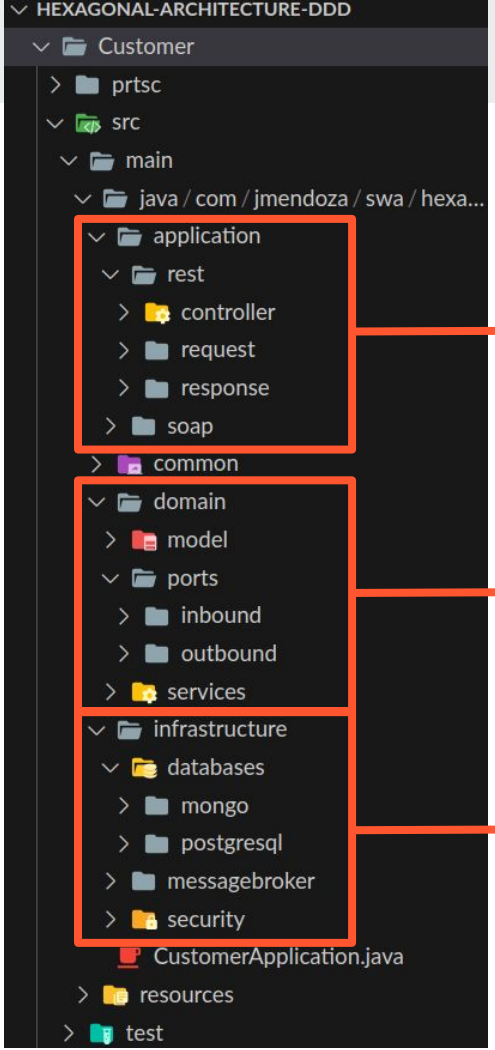


Organização do Núcleo em Camadas



Componentização para Organizar os Serviços





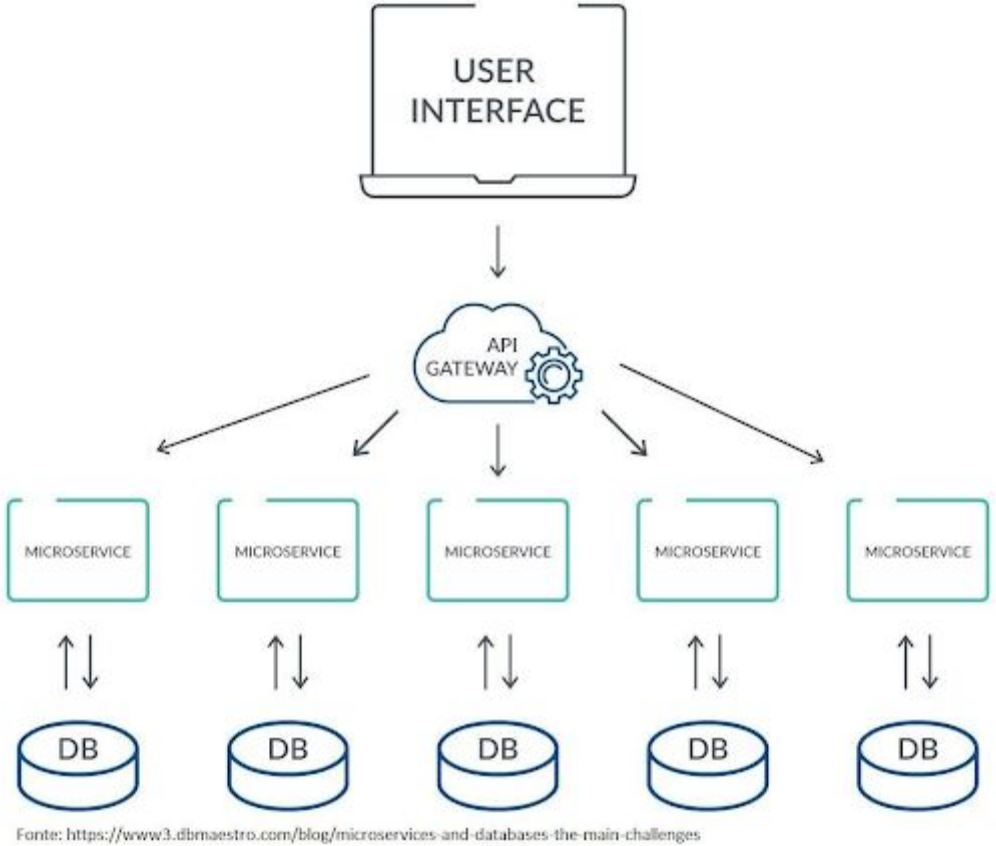
Primary Adapters / Application Adapters

Application Core

Secondary Adapters / Infrastructure Adapters

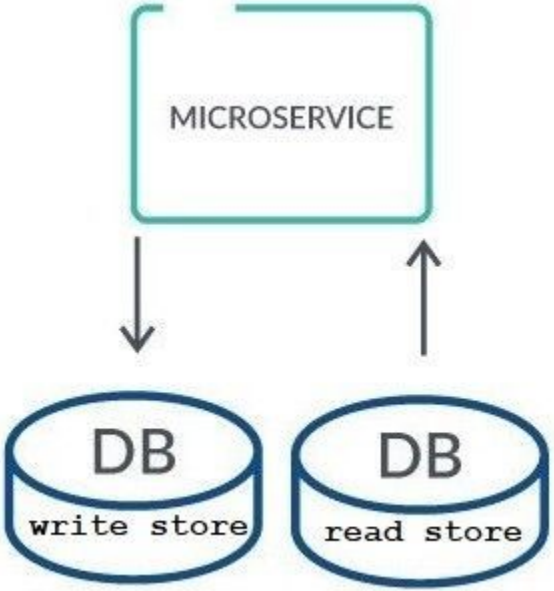
https://github.com/JonathanM2ndoza/Hexagonal-Architecture-DDD

PATTERN
Database Per Service

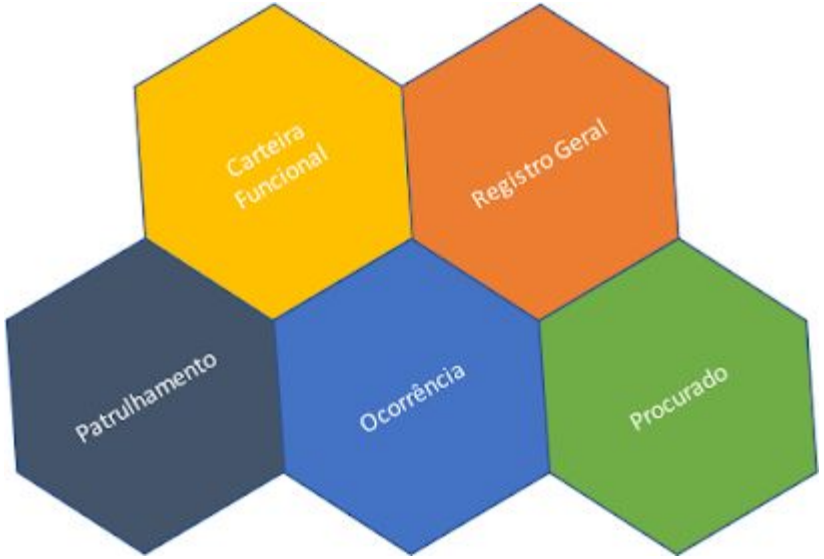


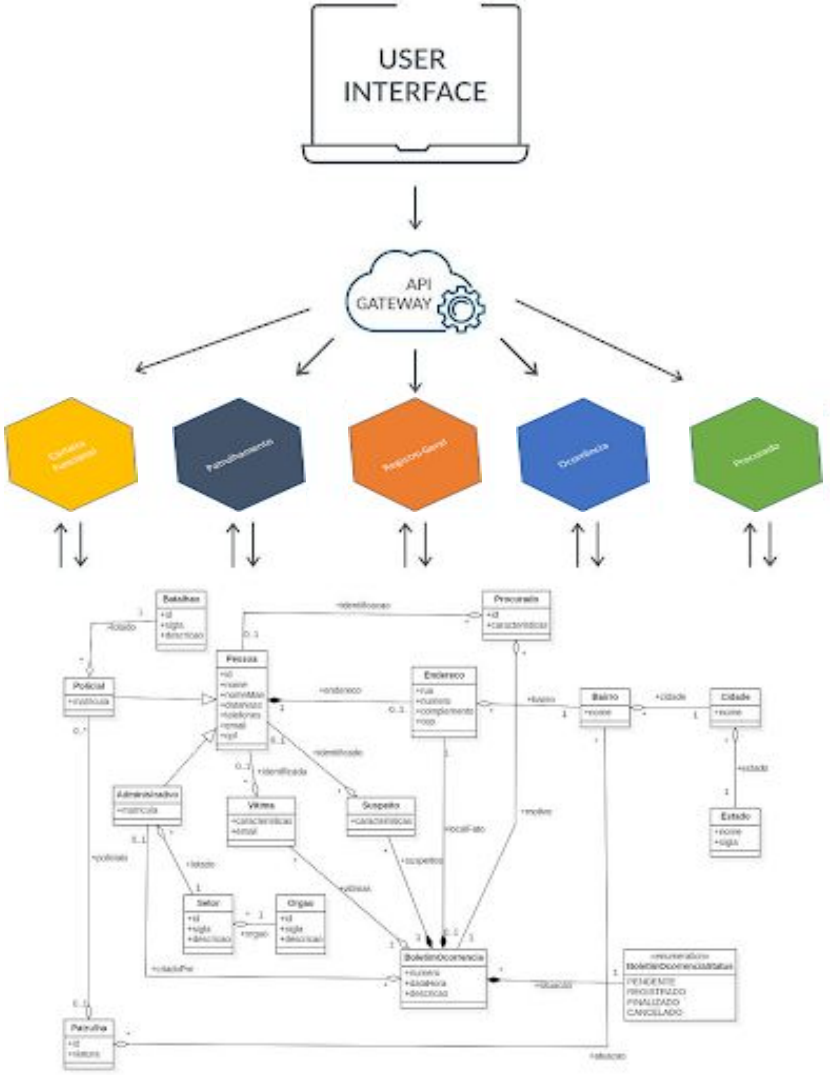
Fonte: https://www3.dbmaestro.com/blog/microservices-and-databases-the-main-challenges

Padrão CQRS

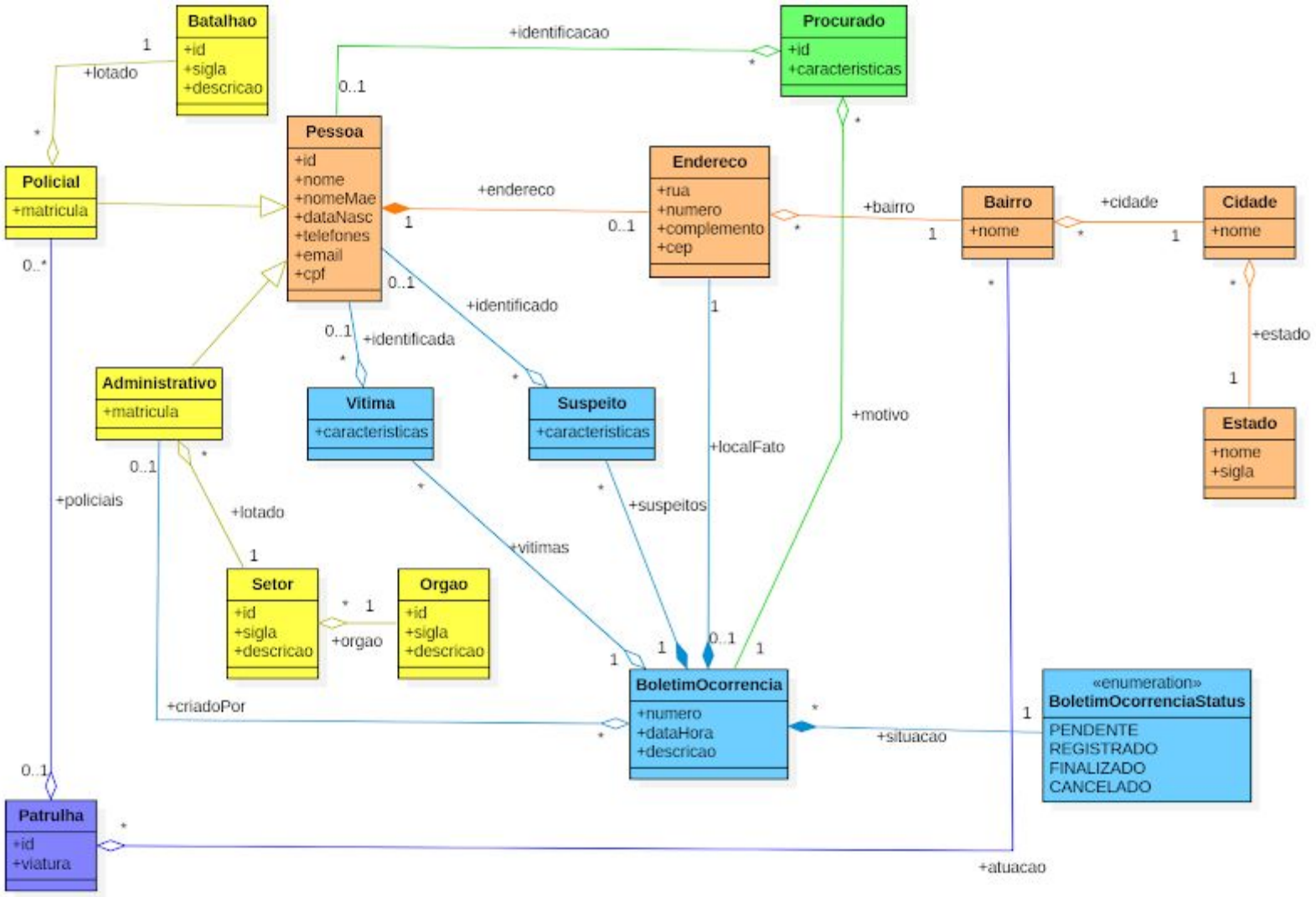


DESAFIO

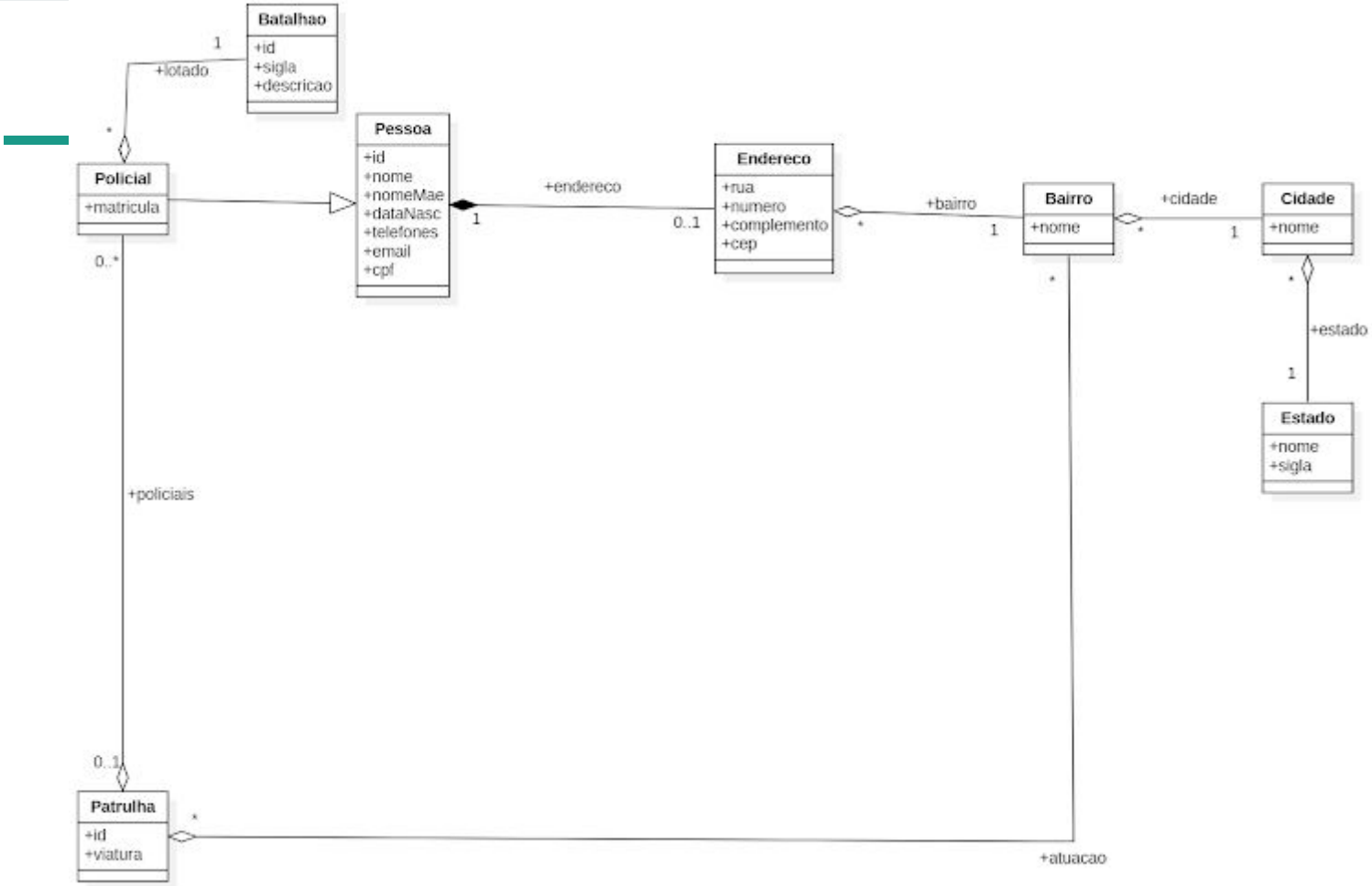

Carteira Funcional
Registro Geral
Patrulhamento
Ocorrência
Procurado


USER INTERFACE
API GATEWAY
Carteira Funcional
Patrulhamento
Registro Geral
Ocorrência
Procurado
Batalhao
Procurado
Pessoa
Policial
Endereco
Bairro
Cidade
Estado
Administrativo
Vitima
Suspeito
Setor
Orgao
BoletimOcorrencia
Patrulha

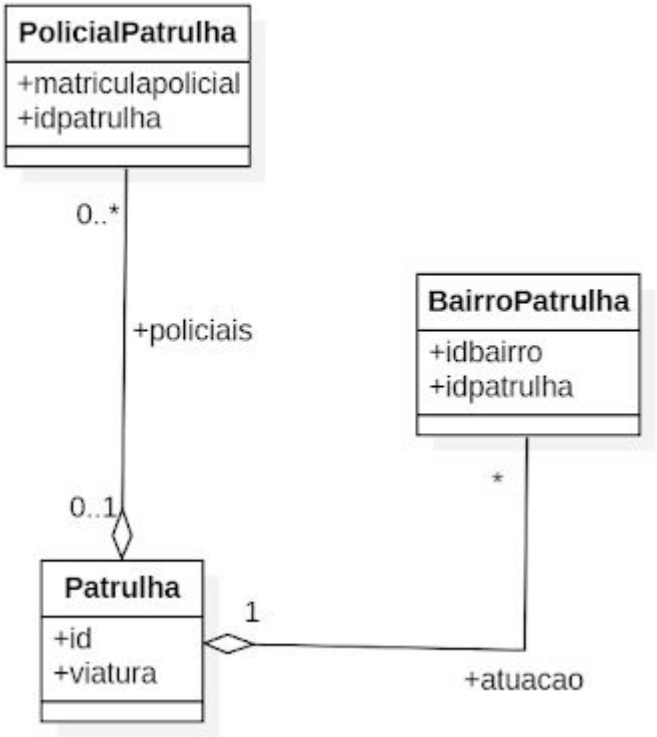
SOLUÇÃO


Batalhao
+id
+sigla
+descricao
Procurado
+id
+caracteristicas
Pessoa
+id
+nome
+nomeMae
+dataNasc
+telefones
+email
+cpf
Endereco
+rua
+numero
+complemento
+cep
Bairro
+nome
Cidade
+nome
Estado
+nome
+sigla
Policial
+matricula
Administrativo
+matricula
Vitima
+caracteristicas
Suspeito
+caracteristicas
Setor
+id
+sigla
+descricao
Orgao
+id
+sigla
+descricao
BoletimOcorrencia
+numero
+dataHora
+descricao
«enumeration»
BoletimOcorrenciaStatus
PENDENTE
REGISTRADO
FINALIZADO
CANCELADO
Patrulha
+id
+viatura
+identificacao
+lotado
+endereco
+bairro
+cidade
+estado
+identificado
+identificada
+localFato
+motivo
+policiais
+lotado
+vitimas
+suspeitos
+orgao
+criadoPor
+situacao
+atuacao

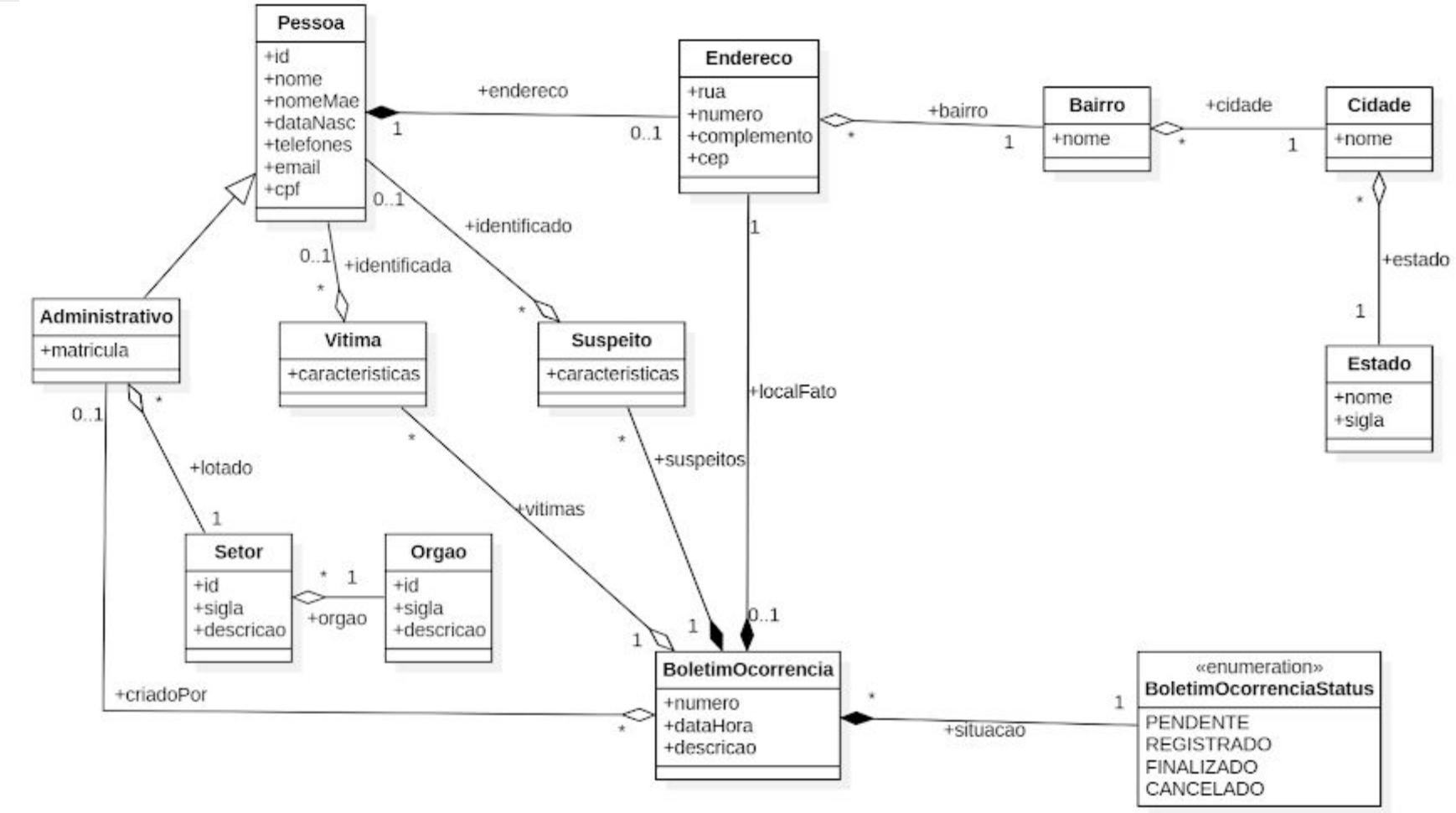
Gerenciamento Patrulha: Estratégia Permissiva



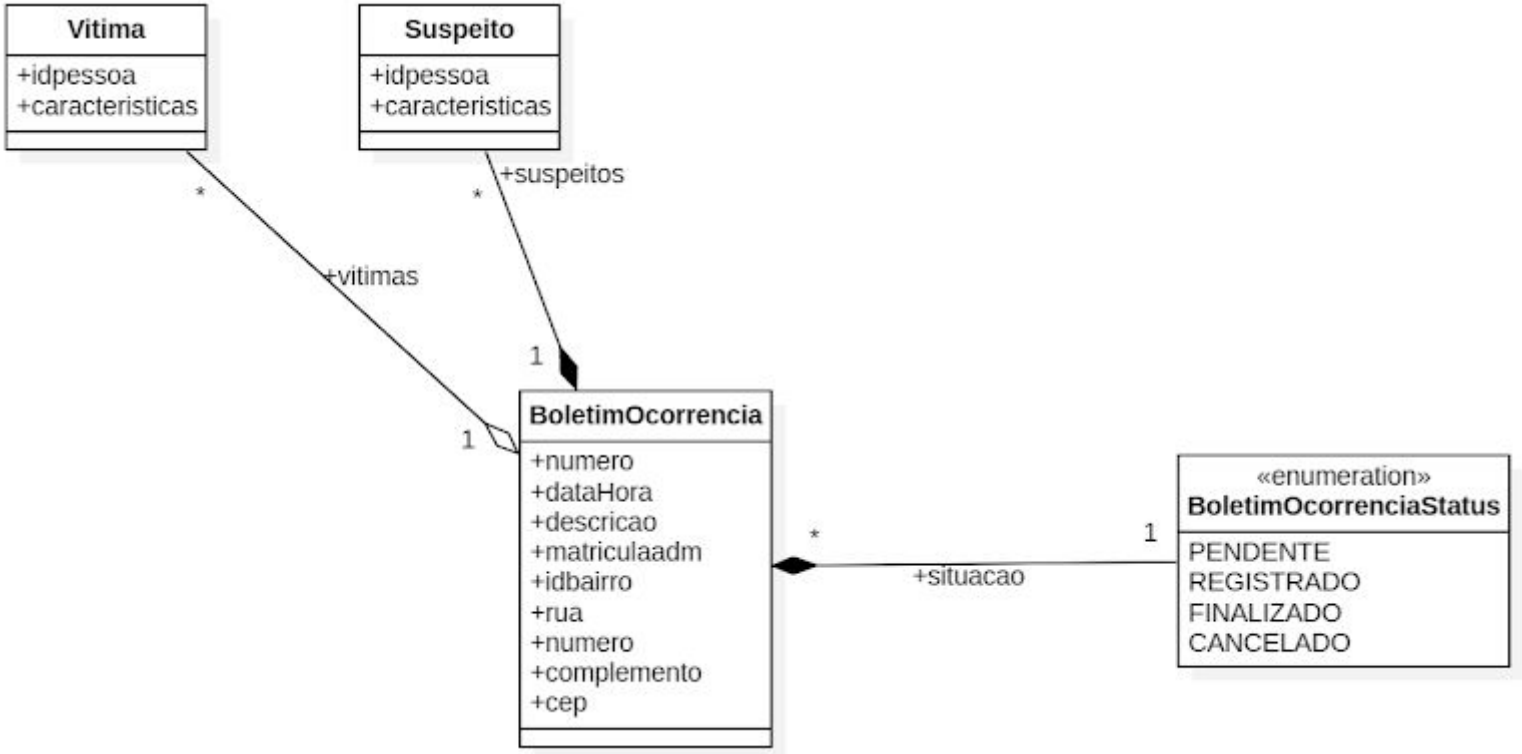
Gerenciamento Patrulha: Estratégia Restritiva



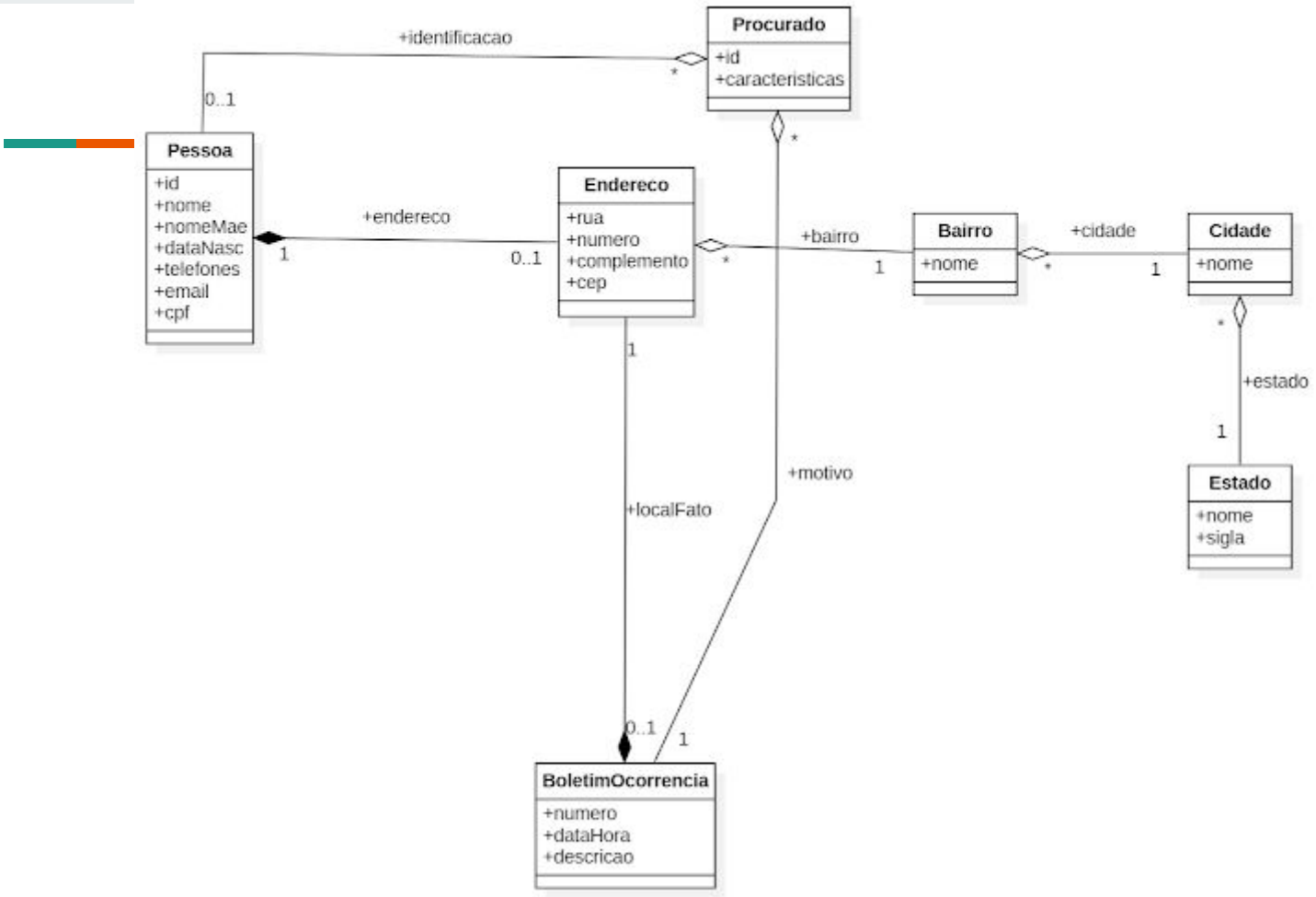
Gerenciamento Ocorrência: Estratégia Permissiva



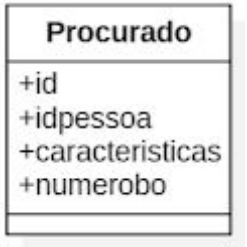
Gerenciamento Ocorrência: Estratégia Restritiva



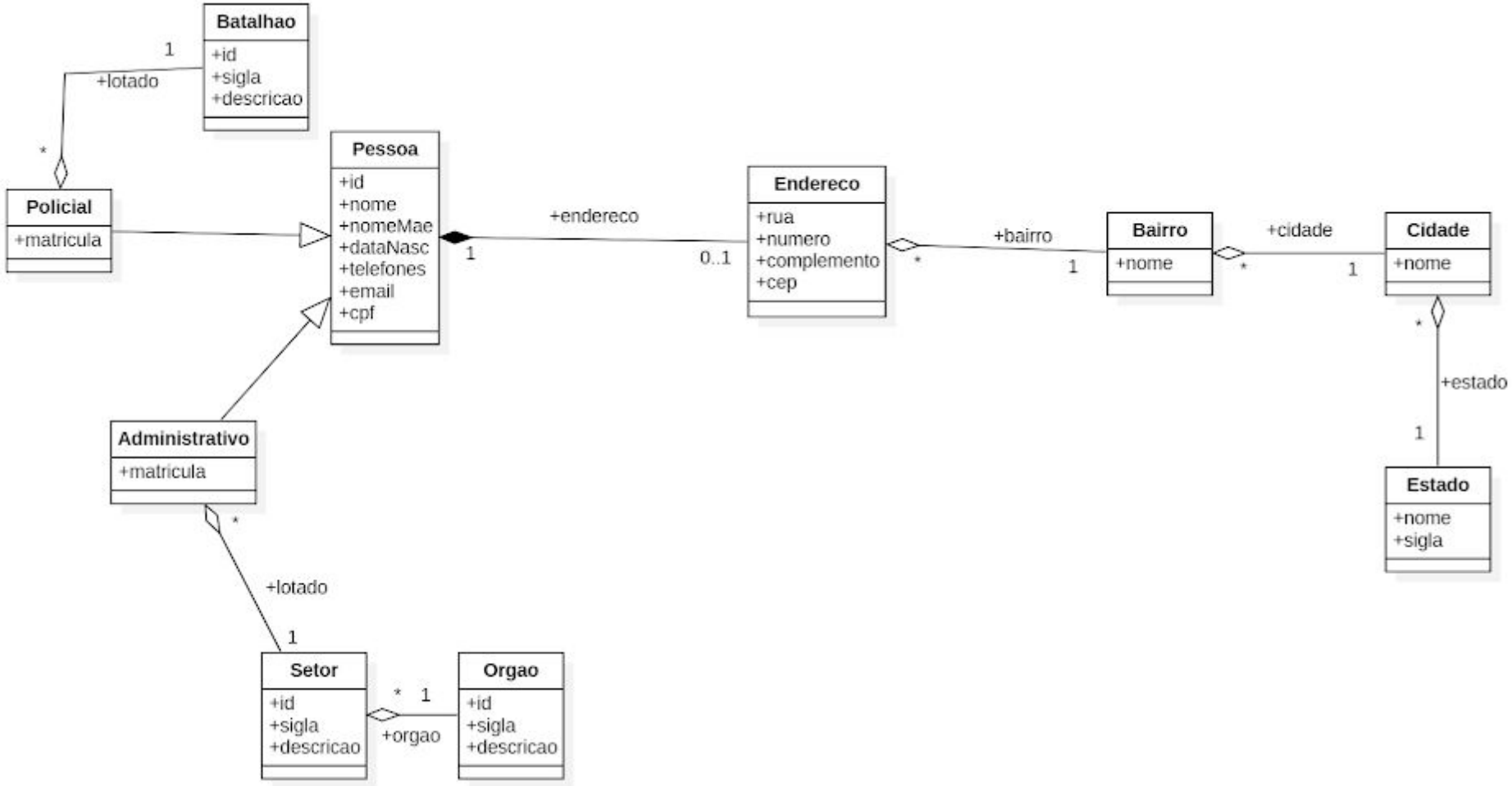
Gerenciamento Procurado: Estratégia Permissiva



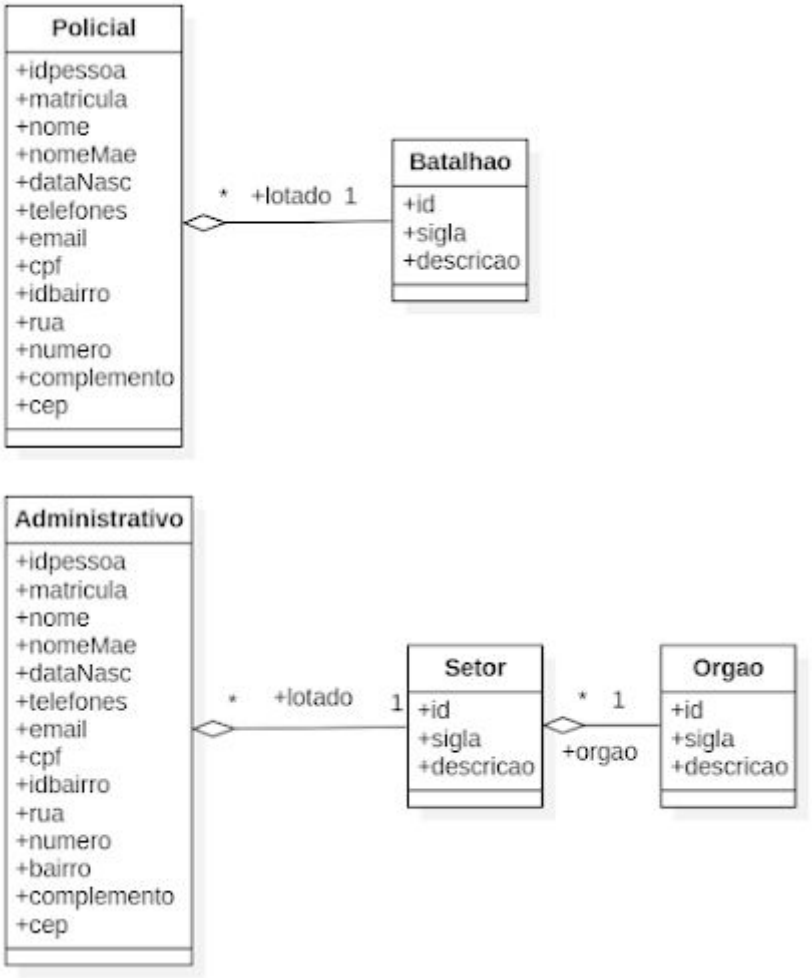
Gerenciamento Procurado: Estratégia Restritiva



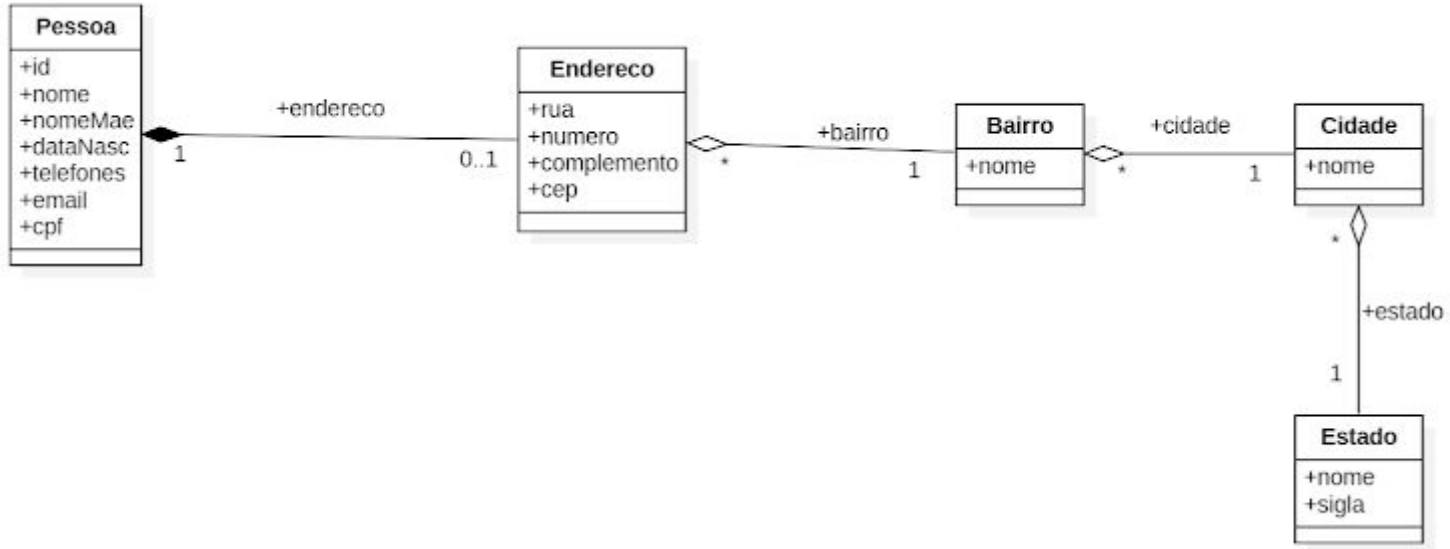
Carteira Funcional: Estratégia Permissiva



Carteira Funcional: Estratégia Restritiva



Registro Geral: Somente Estratégia Restritiva



Características	Estratégia	
	Permissiva	Restritiva
Performance	melhor	pior
Impactos de estrutura	maior	menor
Migração	dificultada	facilitada
Administração BD	dificultada	facilitada

```sql
CREATE USER registro_geral;
CREATE USER carteira_funcional;
CREATE USER gerenciamento_patrulha;
CREATE USER gerenciamento_procurado;
CREATE USER gerenciamento_ocorrencia;

ALTER USER registro_geral WITH PASSWORD 'registro_geral';
ALTER USER carteira_funcional WITH PASSWORD 'carteira_funcional';
ALTER USER gerenciamento_patrulha WITH PASSWORD 'gerenciamento_patrulha';
ALTER USER gerenciamento_procurado WITH PASSWORD 'gerenciamento_procurado';
ALTER USER gerenciamento_ocorrencia WITH PASSWORD 'gerenciamento_ocorrencia';

REVOKE ALL ON ALL TABLES IN SCHEMA "public" FROM registro_geral;
REVOKE ALL ON ALL TABLES IN SCHEMA "public" FROM carteira_funcional;
REVOKE ALL ON ALL TABLES IN SCHEMA "public" FROM gerenciamento_patrulha;
REVOKE ALL ON ALL TABLES IN SCHEMA "public" FROM gerenciamento_procurado;
REVOKE ALL ON ALL TABLES IN SCHEMA "public" FROM gerenciamento_ocorrencia;
```

```sql
GRANT SELECT, UPDATE, INSERT, DELETE ON batalhao TO carteira_funcional;
GRANT SELECT, UPDATE, INSERT, DELETE ON policial TO carteira_funcional;
GRANT SELECT, UPDATE, INSERT, DELETE ON orgao TO carteira_funcional;
GRANT SELECT, UPDATE, INSERT, DELETE ON setor TO carteira_funcional;
GRANT SELECT, UPDATE, INSERT, DELETE ON administrativo TO carteira_funcional;
```

```sql
GRANT SELECT, UPDATE, INSERT, DELETE ON vitima TO gerenciamento_ocorrencia;
GRANT SELECT, UPDATE, INSERT, DELETE ON suspeito TO gerenciamento_ocorrencia;
GRANT SELECT, UPDATE, INSERT, DELETE ON boletim_ocorrencia TO gerenciamento_ocorrencia;
```

Migração

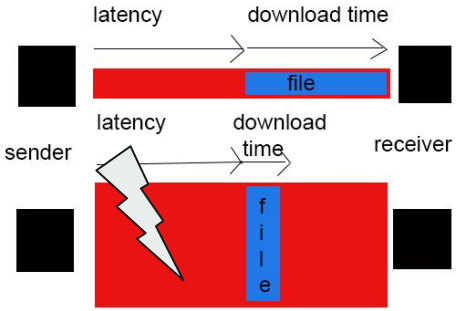
https://ieeexplore.ieee.org/document/9514268

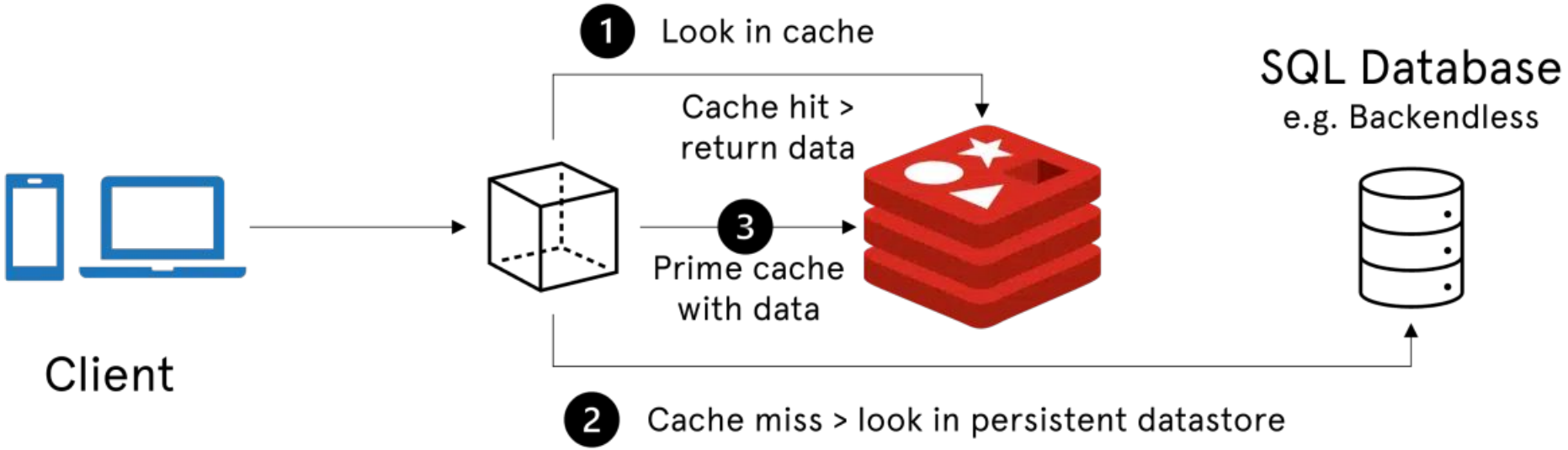
OBSERVAÇÕES IMPORTANTES

UUID (`0934b1f5-6aaf-4201-a54b-14e13dbd9b62`)

- Identifica unicamente uma informação em sistemas distribuídos
- Problema dos identificadores sequenciais:
  - Exige coordenação central
  - Dificulta a sincronização
  - Causa vazamento acidental de informação
  - Permite realização de estimativas para robôs ou ataques na segurança
  - Permite criar operações idempotentes

Caching


latency
download time
file
latency
download time
sender
receiver
f
i
l
e


1 Look in cache
Cache hit > return data
SQL Database
e.g. Backendless
3
Prime cache with data
Client
2 Cache miss > look in persistent datastore

Caching

Cache	Vantagem	Desvantagem
Embutido	Fácil de configurar e implantar. Baixa latência de acesso aos dados. Sem necessidade de manutenção pela equipe de operação. Impacto de disponibilidade local.	Gerenciamento não é flexível (escalonamento, backup). Limitado a tecnologia utilizada em cada microserviço (Java, python, ...). Dados armazenados e gerenciados junto da aplicação.
Cliente-Servidor	Dados armazenados e gerenciados separados da aplicação. Gerenciamento flexível (escalonamento, backup). Independe da linguagem de programação.	Demanda esforço da equipe de operação. Ajustes de configuração de rede necessários. Maior latência. Impacto de disponibilidade global.

Caching

Característica	Memcached	Redis
Baixa latência (sub-millisecond latency)	Sim	Sim
Facilidade de uso	Sim	Sim
Particionamento de dados	Sim	Sim
Suporte a extensa variedade de linguagens de programação	Sim	Sim
Estruturas de Dados Avançadas	-	Sim
Arquitetura Multithreaded	Sim	-
Snapshots	-	Sim
Replicação	-	Sim
Transações	-	Sim
Publish/Subscribe	-	Sim
Script Lua	-	Sim
Suporte Geospacial	-	Sim

Caching

```
@Cacheable("cidades")
public CidadeDTO buscar(Long id) {
  ...
}
```

```
@CacheEvict(cacheNames="cidades")
public CidadeDTO inserir(CidadeSaveDTO cidadeSaveDTO) {
  ...
}
```

```
@CachePut(value = "cidades", key = "#cidadeSaveDTO.id")
public CidadeDTO alterar(CidadeSaveDTO cidadeSaveDTO) {
   ...
}
```

Redis Leitura Complementar

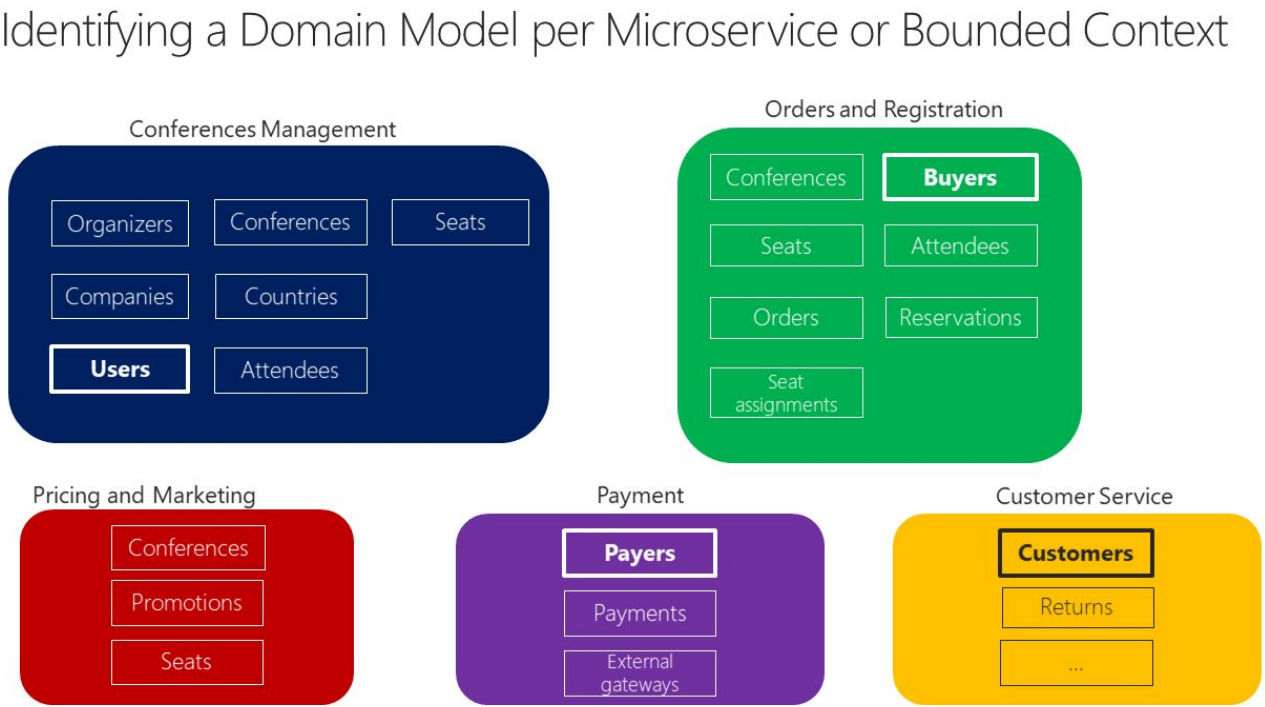
https://redis.com/ebooks/

https://redis.com/blog/redisjson-public-preview-performance-benchmarking/

https://www.atatus.com/blog/redis-metrics/

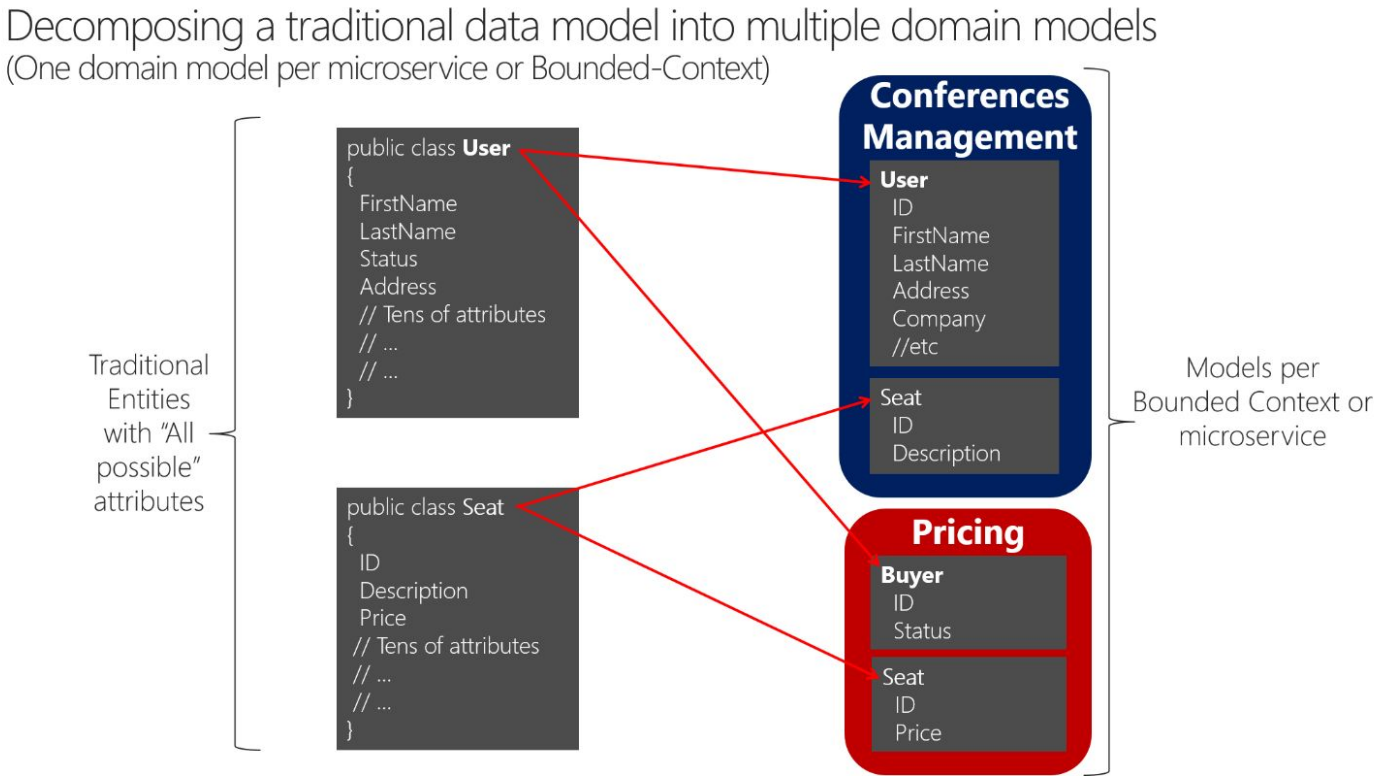
https://about.gitlab.com/blog/2022/11/28/how-we-diagnosed-and-resolved-redis-latency-spikes/

Decomposing Data Model -> Multiple Domains



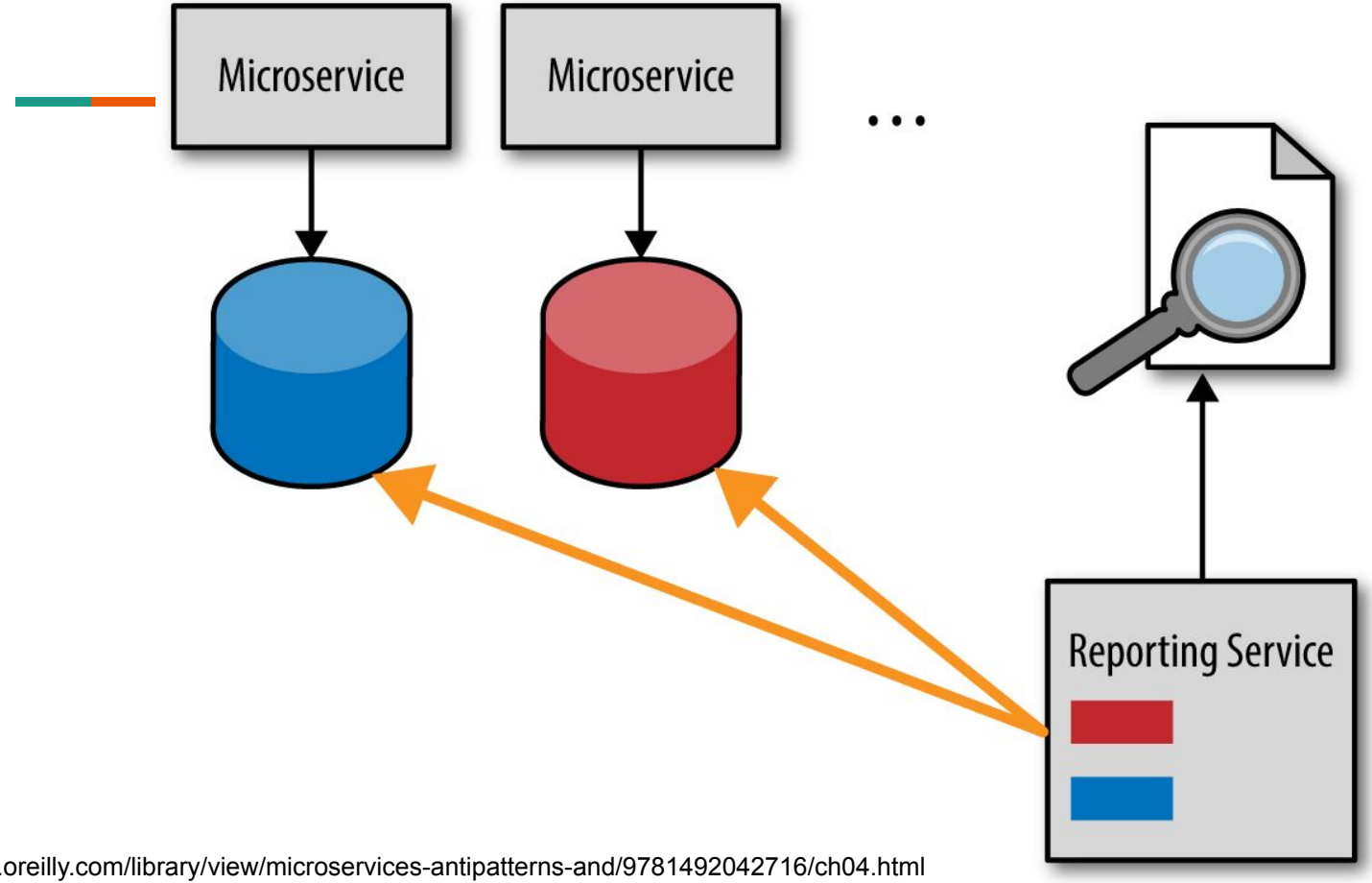
https://learn.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/identify-microservice-domain-model-boundaries

Decomposing Data Model -> Multiple Domains



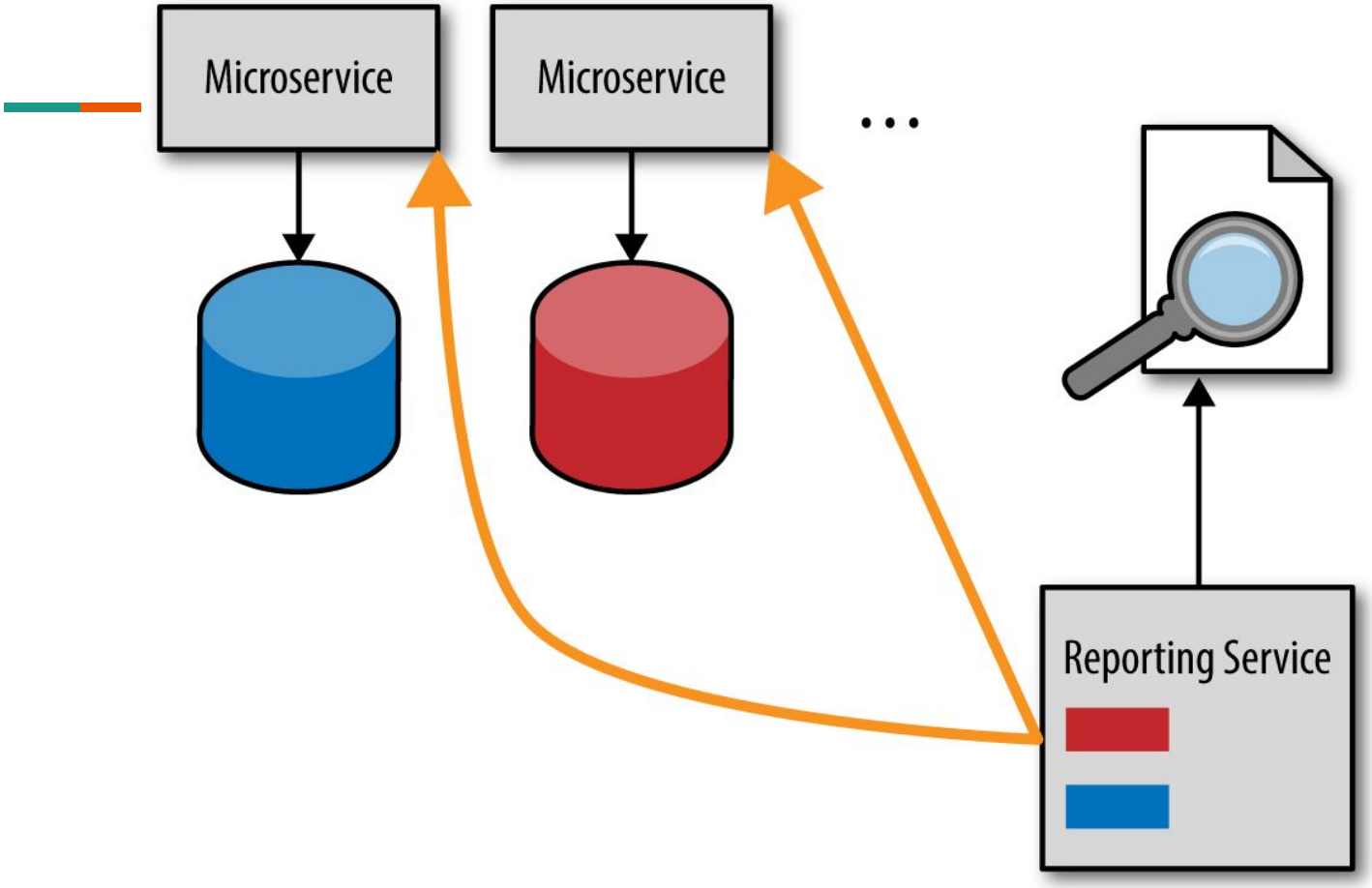
Relatórios

DB Pull

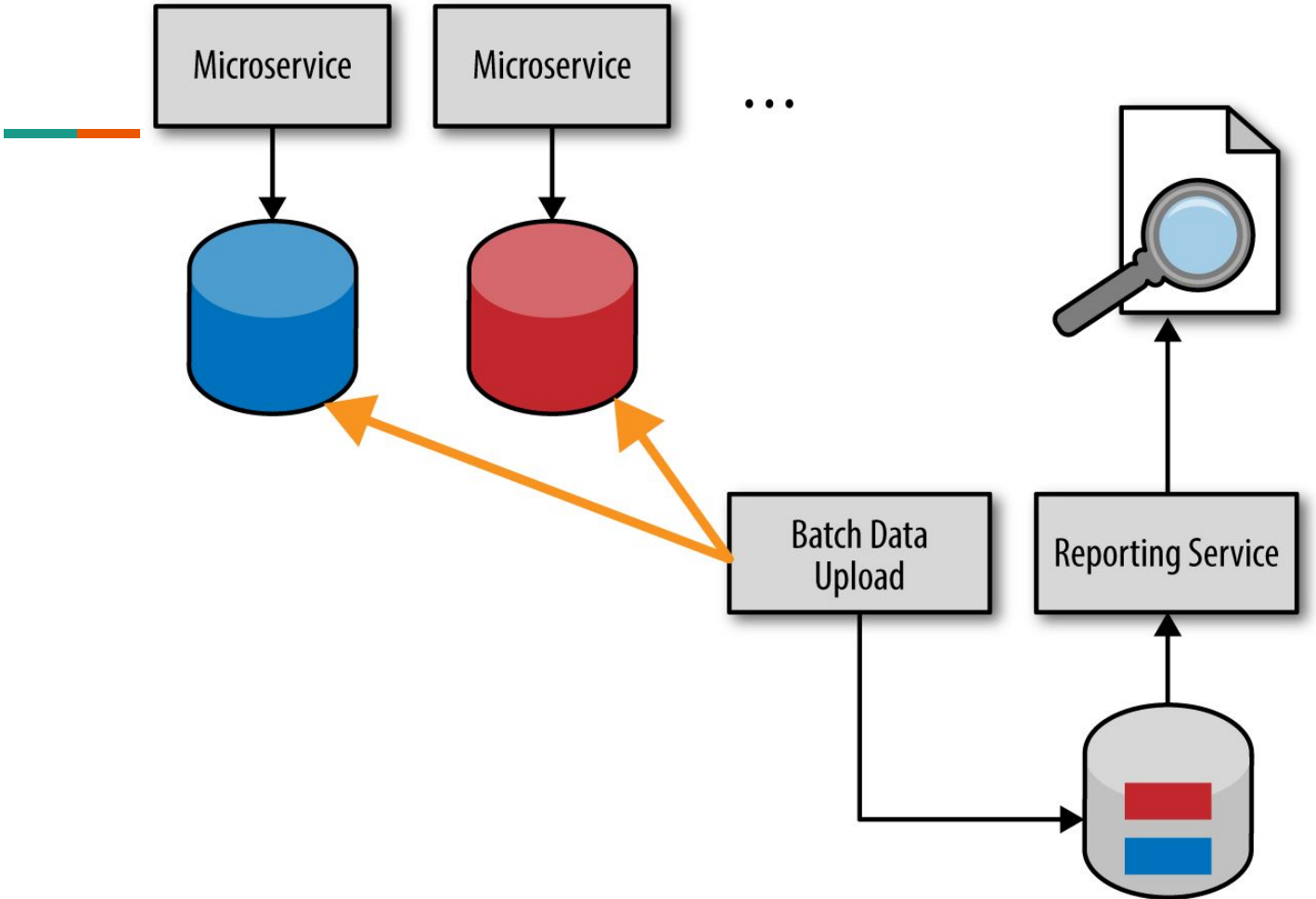


https://www.oreilly.com/library/view/microservices-antipatterns-and/9781492042716/ch04.html

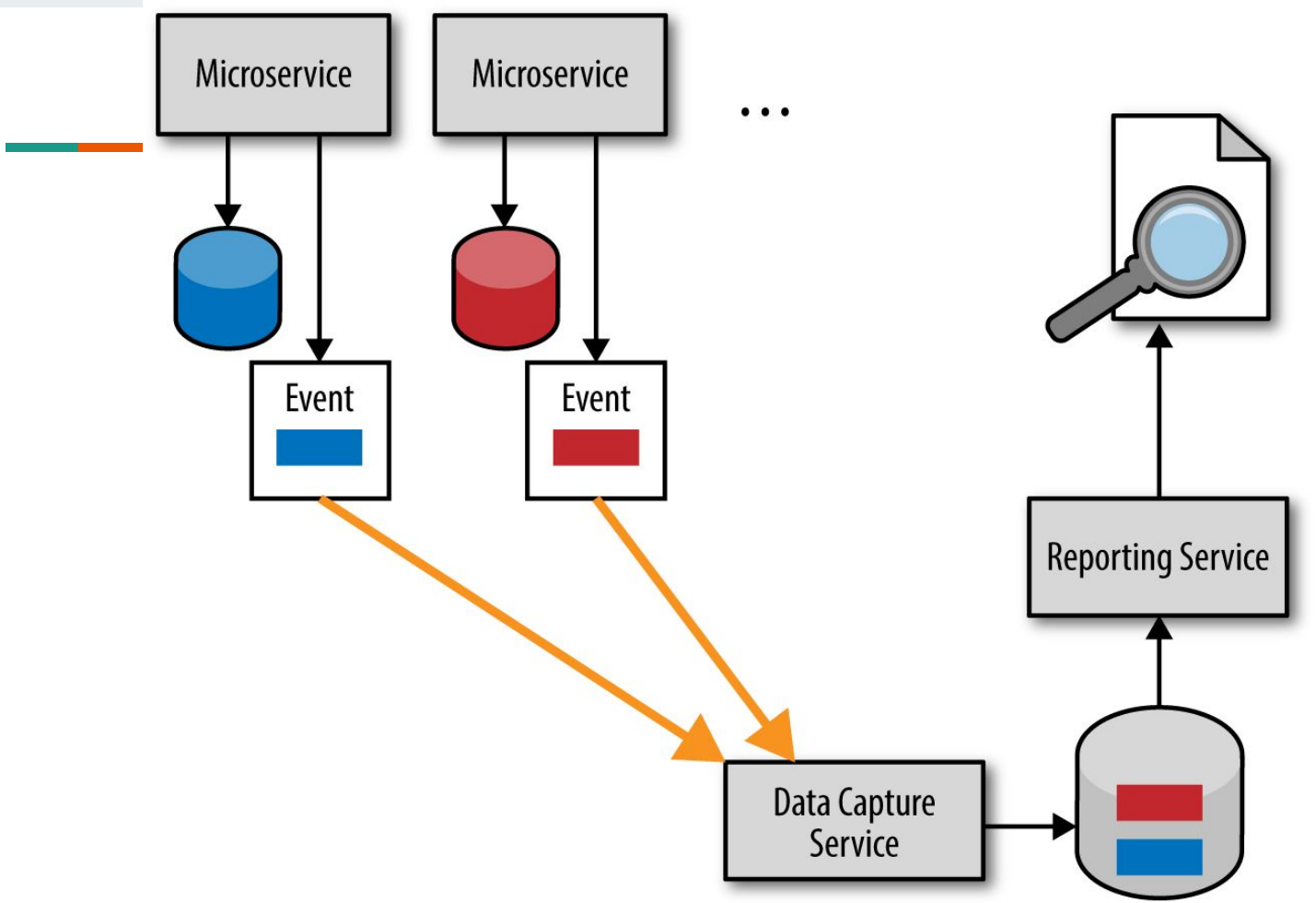
HTTP Pull

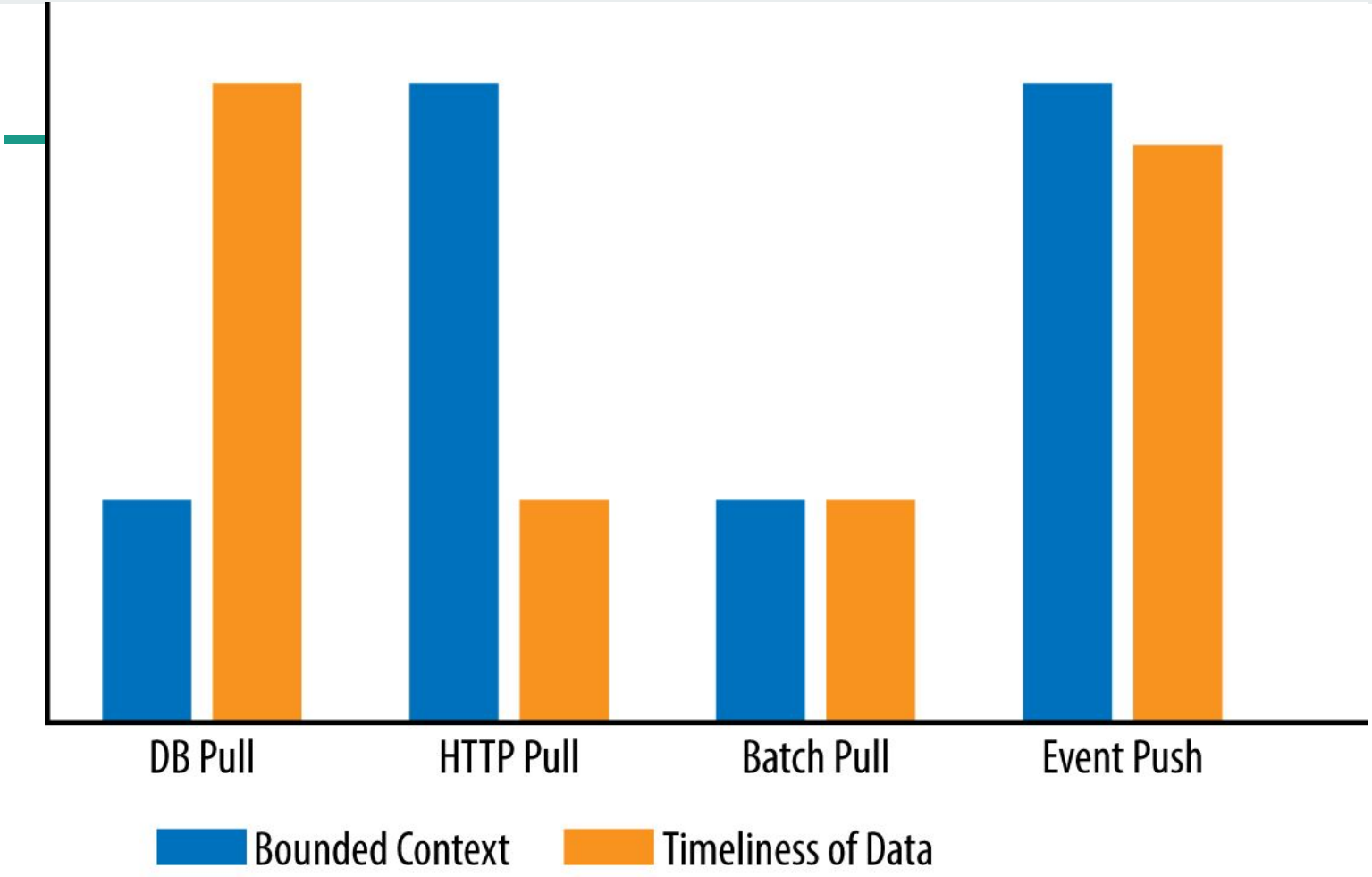


Batch Pull


Microservice
Microservice
Batch Data Upload
Reporting Service

Event Push


Microservice
Microservice
Event
Event
Reporting Service
Data Capture Service


DB Pull
HTTP Pull
Batch Pull
Event Push
Bounded Context
Timeliness of Data