



Batch e Streaming

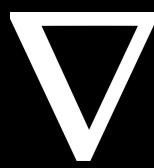
Explorando métodos
de processamento e
suas aplicações
práticas.

Apresentação por:
Victor Zaneti
@vzanetii

Índice

- Introdução
- Definições: Batch e Streaming
- Processamento Batch
- Processamento Streaming
- Arquiteturas Híbridas:
Lambda e Kappa
- Casos de Uso Práticos
- Comparação Batch x Streaming
- Desafios e Soluções
- Conclusão

Introdução



Introdução

A crescente demanda por processamento de dados eficientes na engenharia de dados exige métodos que lidem com grandes volumes e forneçam insights em tempo real.

Dois métodos principais se destacam nesse cenário:
Batch, que processa dados em grandes lotes acumulados, e Streaming, que processa dados em tempo real conforme eles chegam.

Compreender as diferenças e as aplicações práticas de cada abordagem é essencial para escolher a solução adequada para diferentes cenários.

Definições



Definições

O que é?

Batch

Batch refere-se ao processamento de grandes volumes de dados acumulados, geralmente em intervalos regulares.

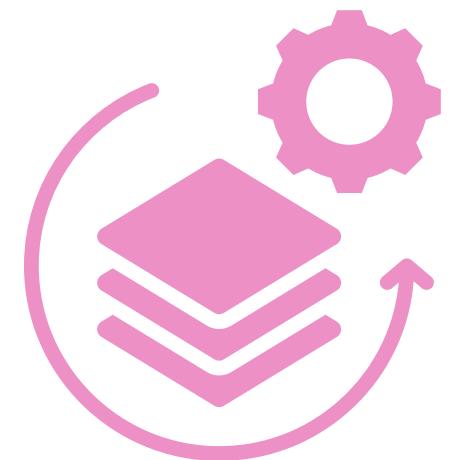
Esse método é ideal para aplicações que não demandam tempo real, como relatórios mensais ou anuais.

Streaming

Streaming envolve o processamento contínuo de dados em tempo real, permitindo análises e respostas imediatas a eventos. É amplamente utilizado em sistemas como monitoramento de fraudes ou redes sociais.

Ambos os métodos têm aplicações práticas distintas e desafios únicos, que abordaremos nos próximos slides.

Processamento Batch



Batch

O processamento Batch é caracterizado pela análise de grandes volumes de dados armazenados ao longo do tempo. Isso o torna ideal para tarefas que exigem precisão histórica, mas não são sensíveis ao tempo, como cálculos financeiros e análises de tendências.

Vantagens

- Eficiência para processar grandes conjuntos de dados.
- Consistência e confiabilidade nos resultados.

Desvantagens

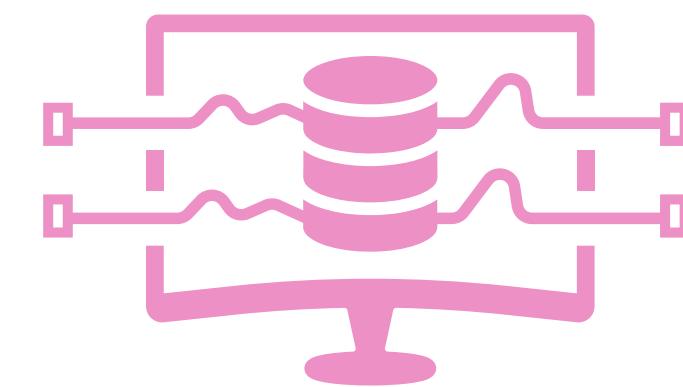
- Alta latência, inadequado para respostas rápidas.
- Requer infraestrutura robusta para armazenar e processar grandes volumes.

Exemplo:

Um sistema de relatórios de desempenho anual que consolida dados de diversas fontes para gerar insights históricos.



Processamento Streaming



Streaming

O processamento Streaming trabalha com dados em movimento, processando-os assim que chegam. Isso permite ações quase instantâneas, como ajustar campanhas de marketing em tempo real ou identificar transações fraudulentas.

Vantagens

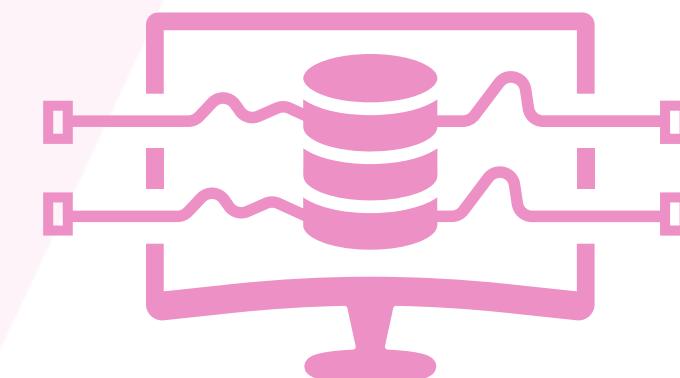
- Permite baixa latência e insights imediatos.
- Ideal para sistemas críticos que exigem monitoramento constante.

Desvantagens

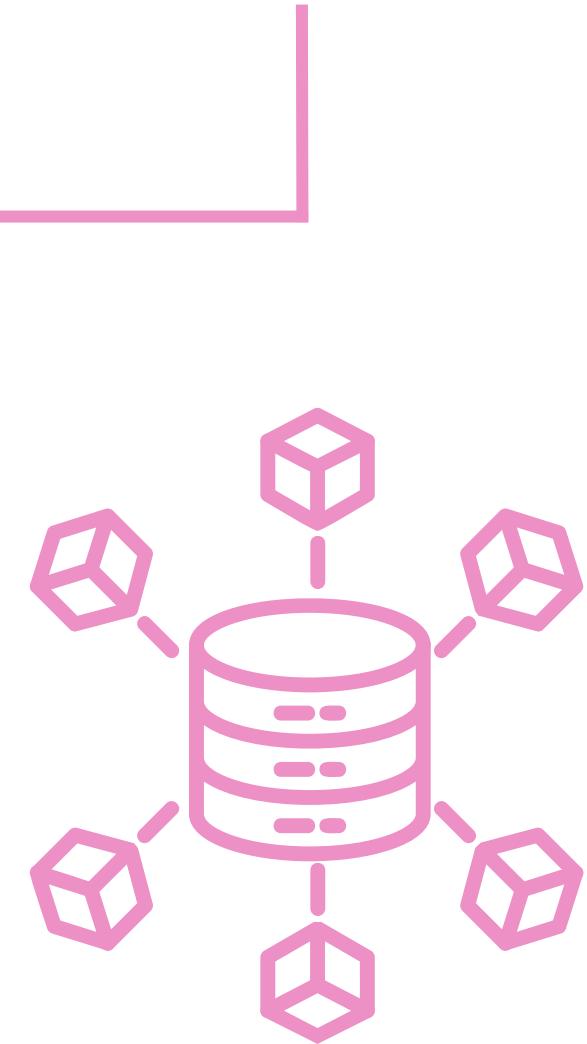
- Complexidade na implementação e manutenção.
- Necessidade de recursos computacionais mais elevados.

Exemplo:

Monitoramento de tráfego em redes sociais para detectar tendências em tempo real.



Arquiteturas Híbridas - Lambda e Kappa



Lambda Architecture

A Lambda Architecture é uma abordagem que combina processamento em batch e streaming para fornecer análises consistentes e em tempo real.

Como Funciona?

Possui três camadas principais:

Camada Batch: Processa grandes volumes de dados históricos para análises precisas e detalhadas.

Camada de Streaming: Trabalha em tempo real para gerar insights imediatos.

Camada de Merging: Combina os resultados de batch e streaming para apresentar dados completos e atualizados.

Vantagens

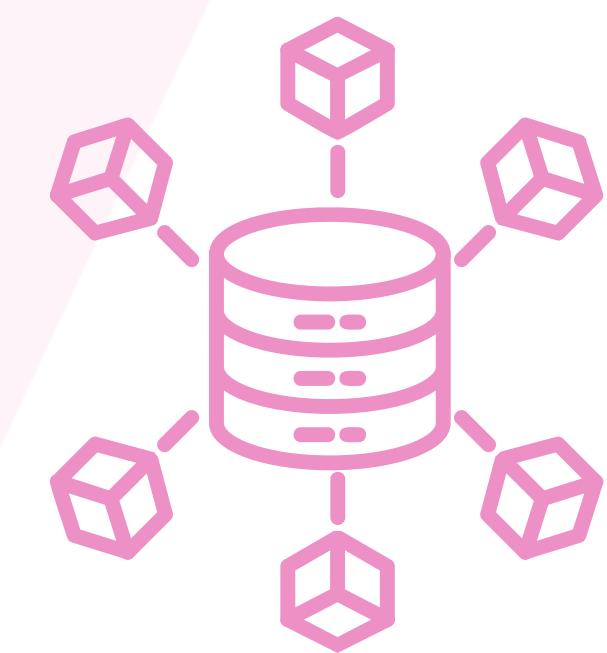
- Análises históricas completas e precisas.
- Respostas rápidas para eventos em tempo real.
- Alta flexibilidade para aplicações que demandam consistência e baixa latência.

Desvantagens

- Complexidade: Requer manutenção de dois pipelines distintos, o que aumenta custos e esforço de desenvolvimento.
- Sincronização: Garantir que batch e streaming estejam alinhados é desafiador e pode gerar inconsistências temporárias.

Exemplo:

Redes sociais que precisam mostrar métricas históricas (número total de curtidas) e insights em tempo real (engajamento ao vivo).



Kappa Architecture

A Kappa Architecture simplifica o processamento ao tratar tudo como streaming, inclusive dados históricos, que são processados como um fluxo de dados armazenado.

Como Funciona?

Não possui uma camada separada de batch.

Utiliza ferramentas de streaming avançadas para processar tanto dados históricos quanto em tempo real.

Vantagens

Simplicidade: Um único pipeline é mais fácil de gerenciar e manter.

Escalabilidade: Perfeito para sistemas de alta performance que precisam lidar com grandes volumes de dados.

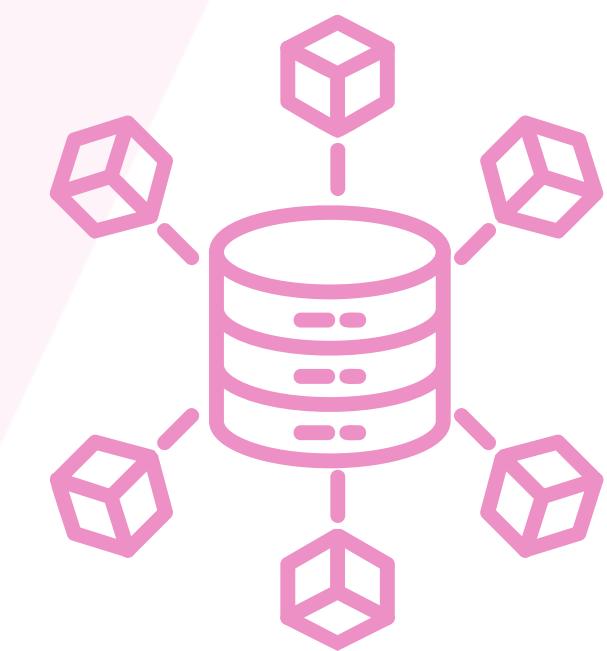
Desvantagens

Limitações em Precisão Histórica:
Sem a camada batch, análises históricas podem ser menos detalhadas.

Dependência de Ferramentas de Streaming: Exige tecnologias robustas como Kafka, Flink ou Spark Streaming.

Exemplo:

Sistemas de monitoramento de saúde em tempo real que analisam tanto o histórico do paciente quanto mudanças momentâneas nos sinais vitais.





Qual escolher?

Lambda

Projetos que exigem análises históricas detalhadas combinadas com respostas rápidas.

Empresas que possuem equipes especializadas capazes de lidar com a complexidade do sistema.

Exemplos: Bancos, redes sociais e plataformas de mídia.

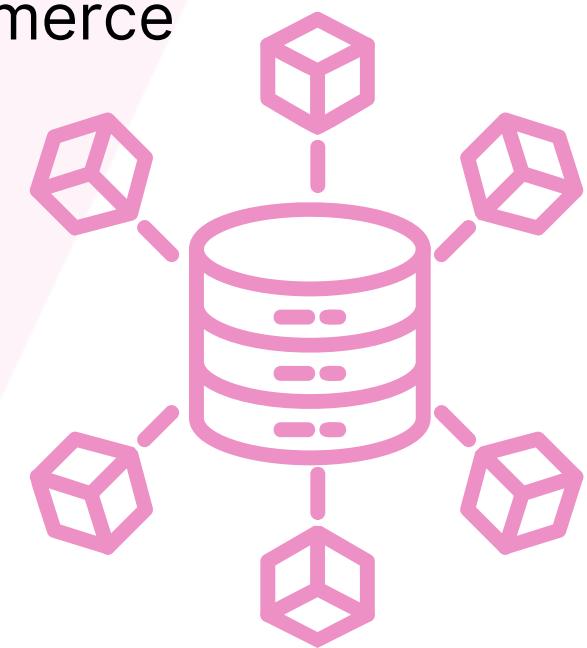
A Lambda Architecture é ideal para cenários onde a consistência histórica é fundamental, mas exige esforços adicionais para gerenciar a complexidade. Já a Kappa Architecture oferece simplicidade e desempenho, mas pode comprometer análises históricas detalhadas. A escolha entre as duas depende das prioridades do projeto, como latência, precisão e custo operacional.

Kappa

Projetos que priorizam simplicidade e eficiência operacional.

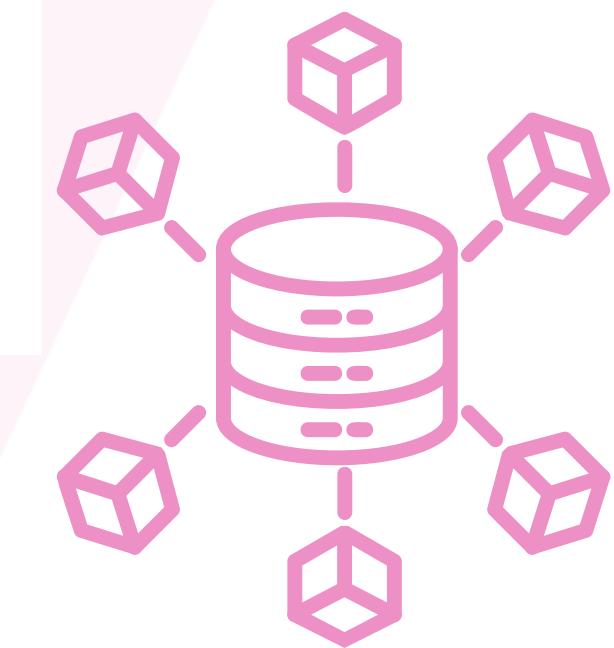
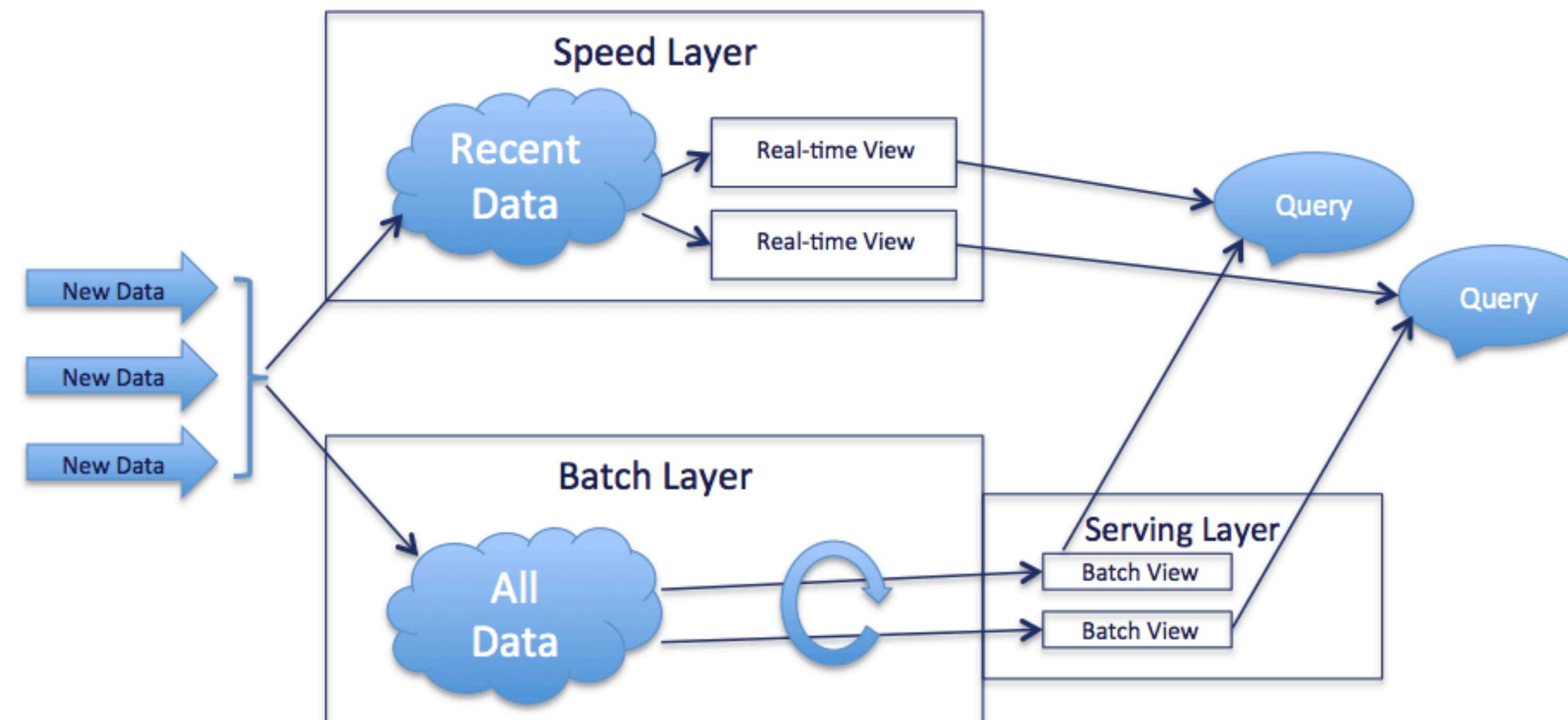
Aplicações que processam grandes volumes de dados continuamente, mas não exigem alta precisão histórica.

Exemplos: Monitoramento de IoT, sistemas de análise de comportamento em e-commerce



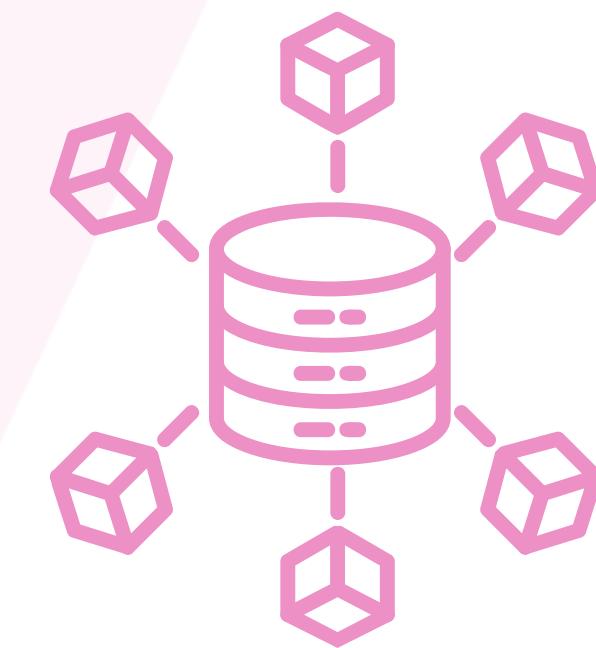
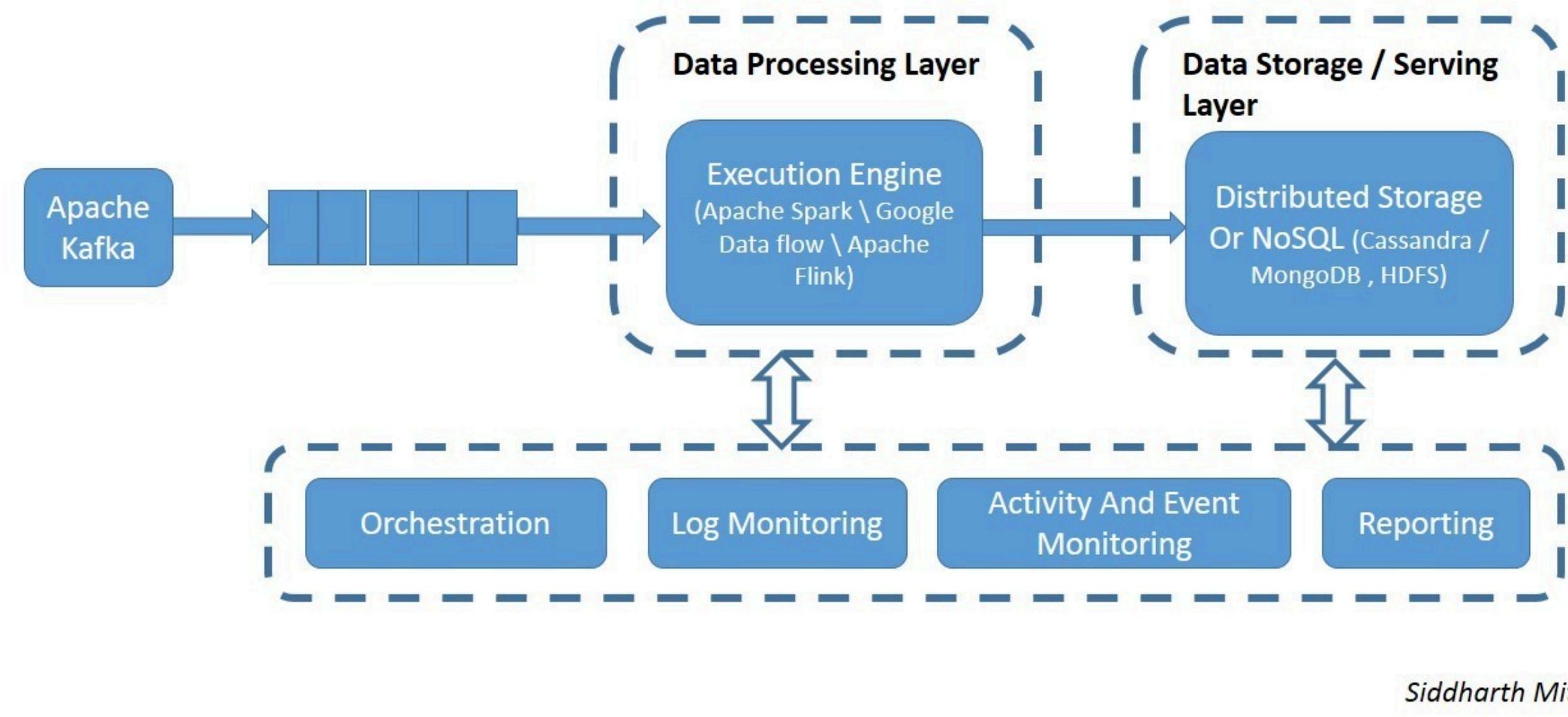
Exemplo Lambda Architecture

The Lambda Architecture

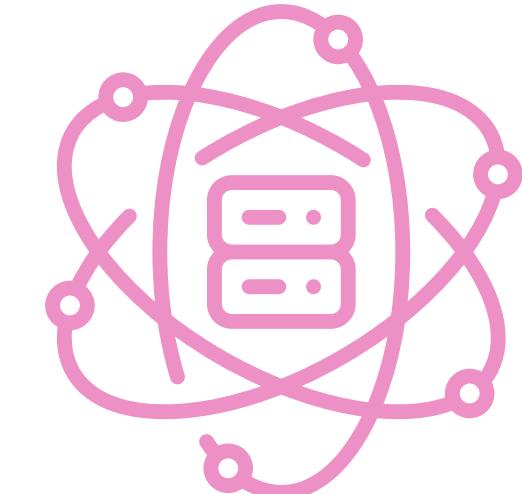


Exemplo Kappa Architecture

Kappa Architecture



Casos de uso Práctico



Casos de Uso: Processamento em Batch

Geração de relatórios financeiros históricos:

Compilação de dados financeiros ao longo de semanas, meses ou anos para criar relatórios detalhados usados em auditorias ou planejamento estratégico.

Por que usar Batch?

Esses dados não exigem análise em tempo real, permitindo processamento em horários otimizados para economia de recursos computacionais.

Análises de dados de vendas mensais:

Avaliação de vendas consolidadas para identificar tendências de consumo e otimizar estoques e campanhas de marketing.

Por que usar Batch?

Os dados são coletados diariamente e analisados periodicamente, o que torna o Batch eficiente para esse propósito.



▽ Casos de Uso: Processamento em Streaming

Detecção de fraudes em transações bancárias:

Monitoramento contínuo de transações financeiras para identificar atividades suspeitas, como padrões incomuns de transferências ou compras.

Por que usar Streaming?

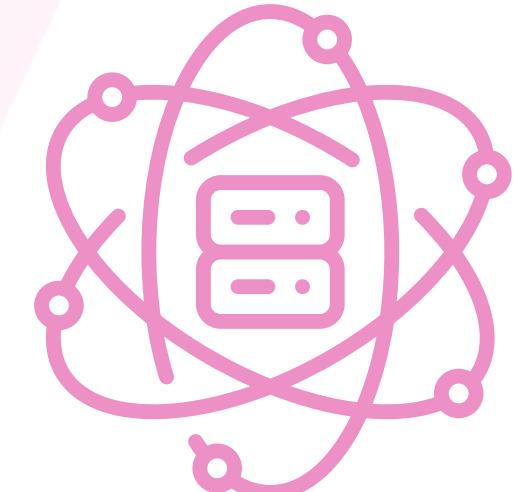
A detecção precisa ser instantânea para prevenir perdas financeiras ou ataques cibernéticos.

Monitoramento de redes sociais em tempo real:

Análise contínua de postagens e interações para detectar tendências, sentimentos públicos ou crises de marca.

Por que usar Streaming?

A relevância das informações depende da rapidez com que são captadas e respondidas.



Ferramentas Utilizadas

Tanto em Batch quanto em Streaming, a escolha das ferramentas é fundamental para garantir a eficiência e a escalabilidade das soluções. Algumas ferramentas amplamente utilizadas incluem:

Apache Kafka
Intermediário poderoso para o fluxo de dados em tempo real e integração com múltiplos sistemas.

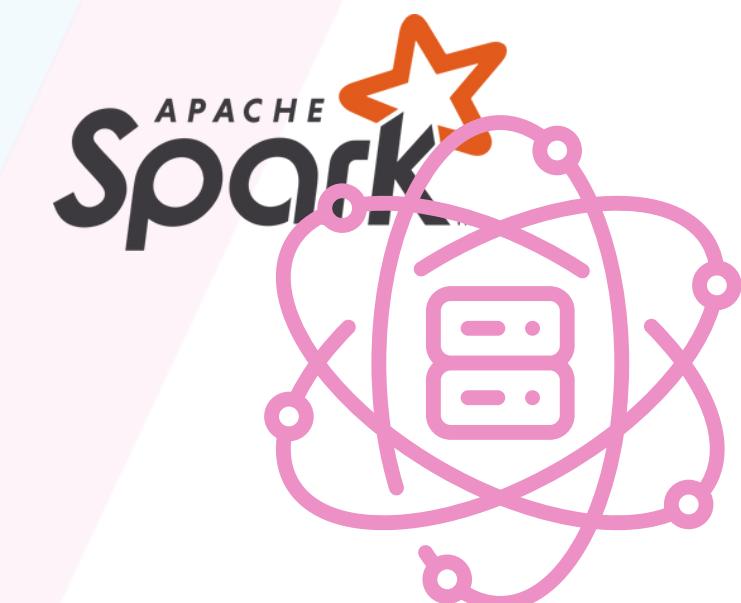


Apache Flink
Especialista em processamento de dados em Streaming e análise em tempo real.

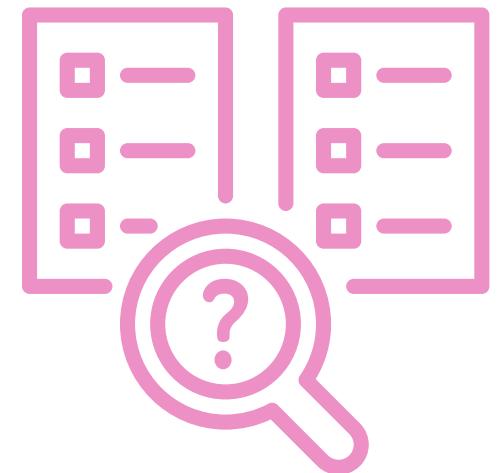


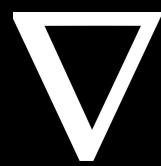
Apache Flink

Apache Spark
Processamento híbrido para cargas em Batch e Streaming, com capacidade de escalabilidade massiva.

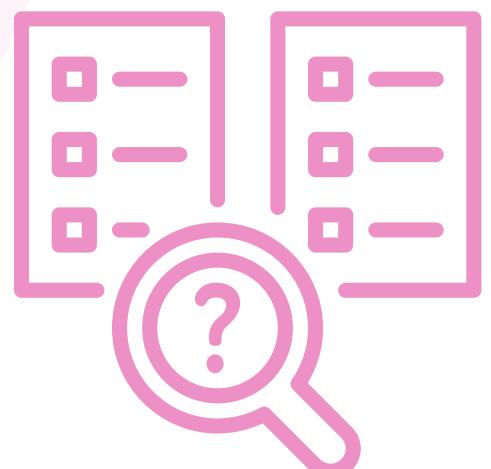


Comparação Batch x Streaming

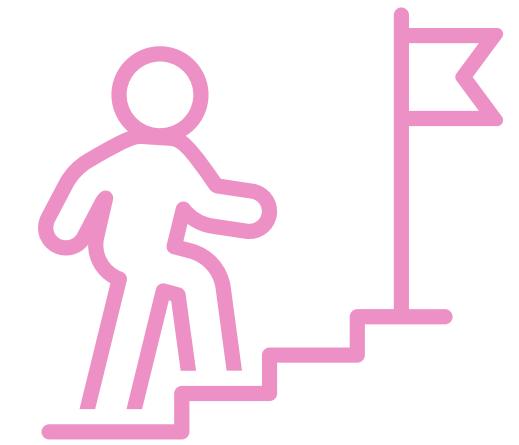




	BATCH	STREAMING
Latência	Alta	Baixa
Vol. de dados	Grandes lotes	Fluxo Contínuo
Aplicações	Históricas	Tempo Real
Armazenamento de Dados	Armazenamento prévio em bancos de dados ou sistemas de arquivos.	Requer buffers ou filas para gerenciar o fluxo contínuo de dados.
Custo Computacional	Menor, processa em horários específicos.	Maior, precisa rodar continuamente



Desafios e Soluções





1) Drift de Modelos no Streaming

O que é o problema?

Drift ocorre quando os padrões nos dados mudam ao longo do tempo, tornando os modelos de aprendizado de máquina menos precisos ou mesmo obsoletos. Em sistemas de streaming, isso é um grande problema, pois os dados são processados continuamente e a adaptação precisa ser imediata.

Por que é um desafio?

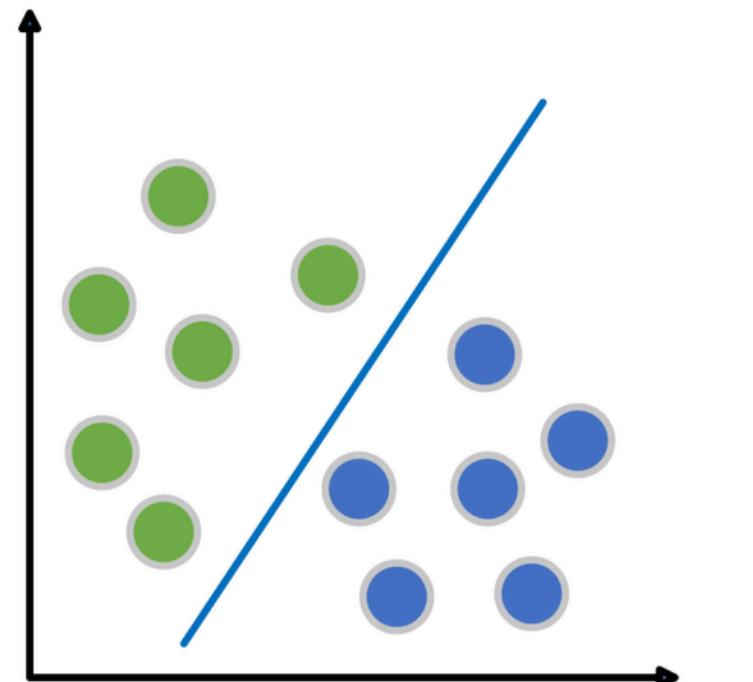
Detectar o drift em tempo real é complexo, já que ele pode ser sutil ou gradual.

Modelos desatualizados podem gerar decisões erradas, como falhas em detectar fraudes ou prever eventos críticos.

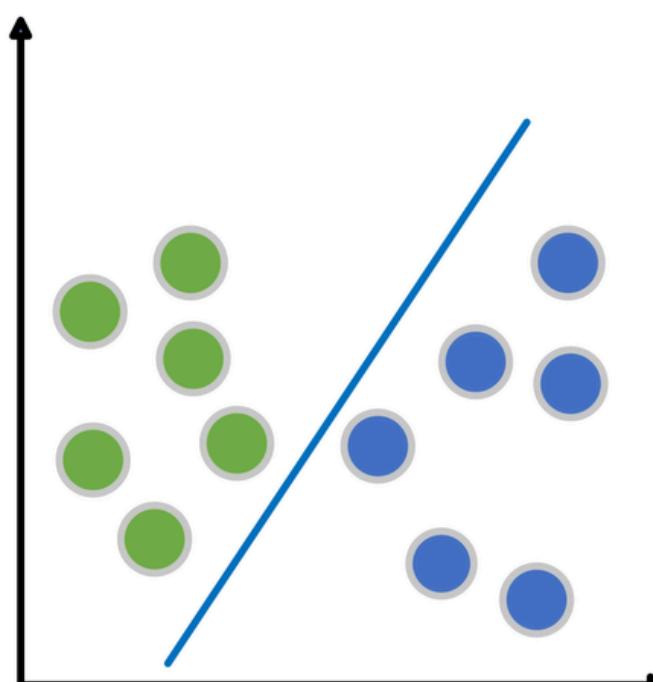
Atualizar modelos continuamente demanda infraestrutura robusta e bem projetada.

Impacto:

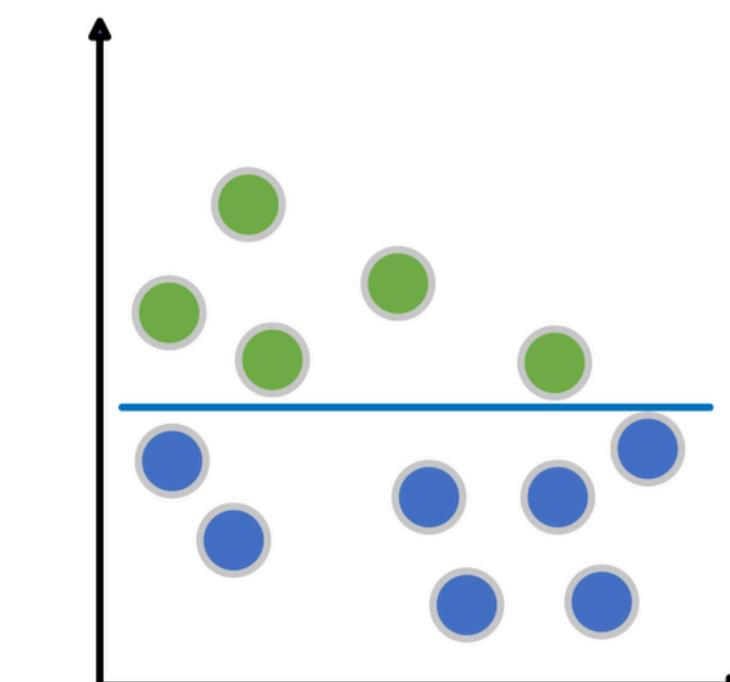
Redução na eficácia dos sistemas e aumento no custo de operação por retrabalho ou erro de análise.



Original Data Distribution



Virtual Drift



Real Drift



Implementação de MLOps para Monitoramento e Ajustes

O que é?

MLOps (Machine Learning Operations) integra práticas de DevOps com aprendizado de máquina para gerenciar modelos em produção. Ele automatiza a detecção de drift, o retrabalho dos modelos e o monitoramento de desempenho.

Por que essa solução funciona?

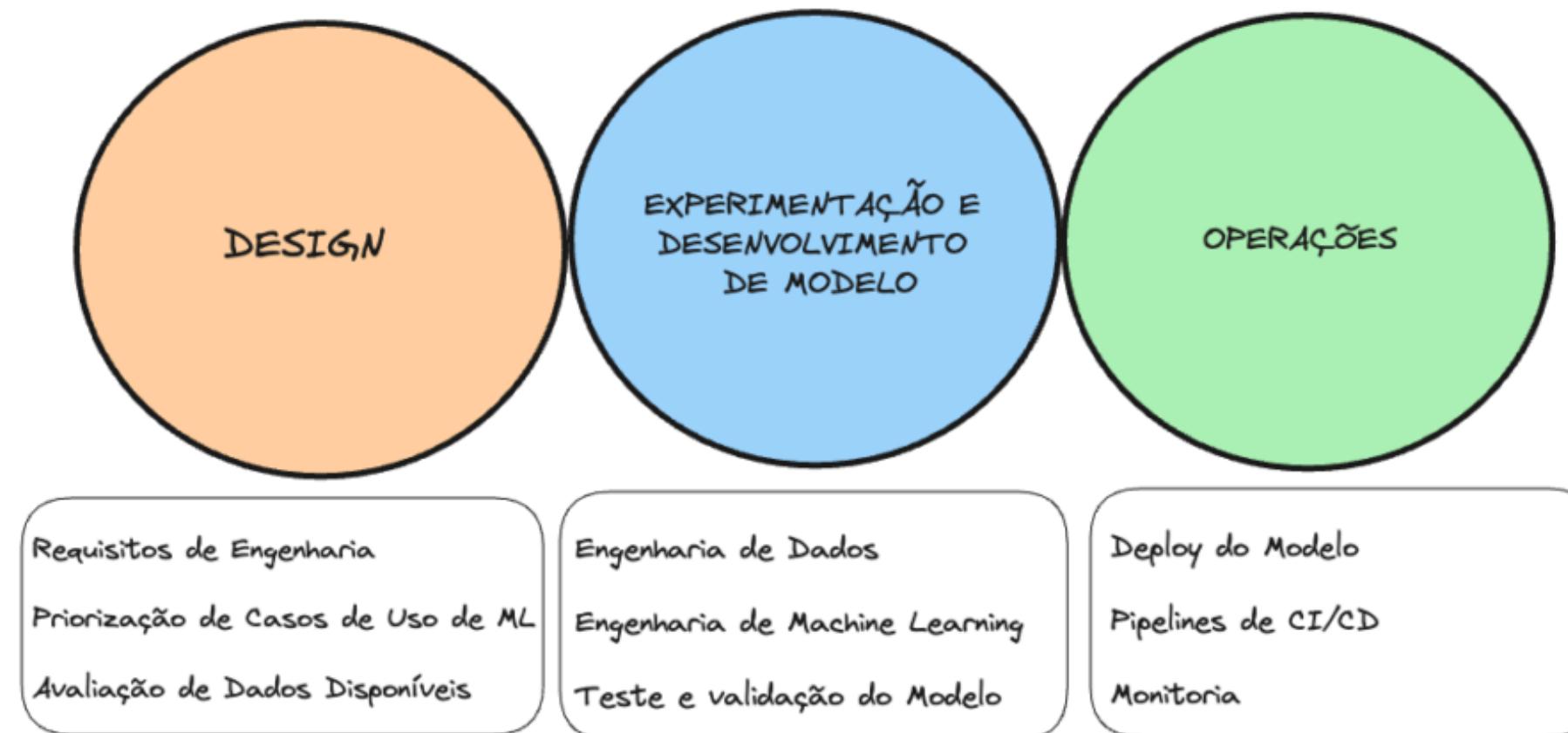
Monitora continuamente os dados e modelos em produção, identificando problemas de forma proativa.
Automatiza o processo de atualização de modelos, garantindo consistência e reduzindo o trabalho manual.

Ferramentas utilizadas:

Apache Kafka: para ingestão e processamento contínuo.
River: para aprendizado incremental e adaptação em tempo real.

Benefícios:

Redução de erros, aumento da eficiência operacional e maior confiabilidade dos sistemas.



Conclusão

Conclusão

Batch e Streaming são métodos complementares, cada um com suas forças.

A escolha depende do caso de uso e dos objetivos do projeto.

Ferramentas modernas tornam possível integrar ambas as abordagens para melhores resultados.

**Obrigado pela
Atenção!**