FIAP

MBA+

Data Science & Artificial Intelligence

MBA+

Data Architecture, Integration and Ingestion

Prof.: Ivan Gancev
Email: profivan.gancev@fiap.com.br

Data Architecture, Integration and Ingestion

(O que vamos explorar?)

Aula 1 – 12/abr (qua)

- Pilares de arquitetura: persistência, integração e consumo
- Estratégias de arquitetura
- Tipos de tratamentos e arquiteturas

Aula 2 – 19/abr (qua)

- Exemplos de Bancos, diferenças e usos:
  - Bancos Relacionais
  - Bancos Colunares

Aula 3 – 26/abr (qua)

- Exemplos de Bancos, diferenças e usos:
  - Bancos de documentos
  - Bancos chave-valor
  - Bancos de Grafos

Aula 4 – 03/mai (qua)

- Ingestão de dados, tratamentos e manipulações
- Pipeline de dados, governança e qualidade
- Integração de dados
  - Cargas batch, ETL, vantagens e desvantagens

Aula 5 – 10/mai (qua)

- Eventos, APIs, NRT e casos de uso
- Arquiteturas para analytics
- Boas práticas, recomendações e cuidados

Eventos, APIs e NRT

Eventos de dados

Eventos de dados são comportamentos ou ações atômicas que podem ser coletadas e tratadas.

Diferente da visão de entidades ou lotes de dados, os eventos estão atrelados normalmente a alguma operação executada com granularidade singular.

Eventos são frequentemente usados em análises de comportamentos, que demandam de decisões individualizadas e em tempo real.

Tipos de comunicação:

- ✓ **Síncrona:** Quando o evento é enviado e aguarda a resposta para seguir (entregue em mãos)
- ✓ **Assíncrona**: Quando o evento é enviado e a etapa do fluxo é concluída independente das etapas posteriores (entregue na caixa de correio)

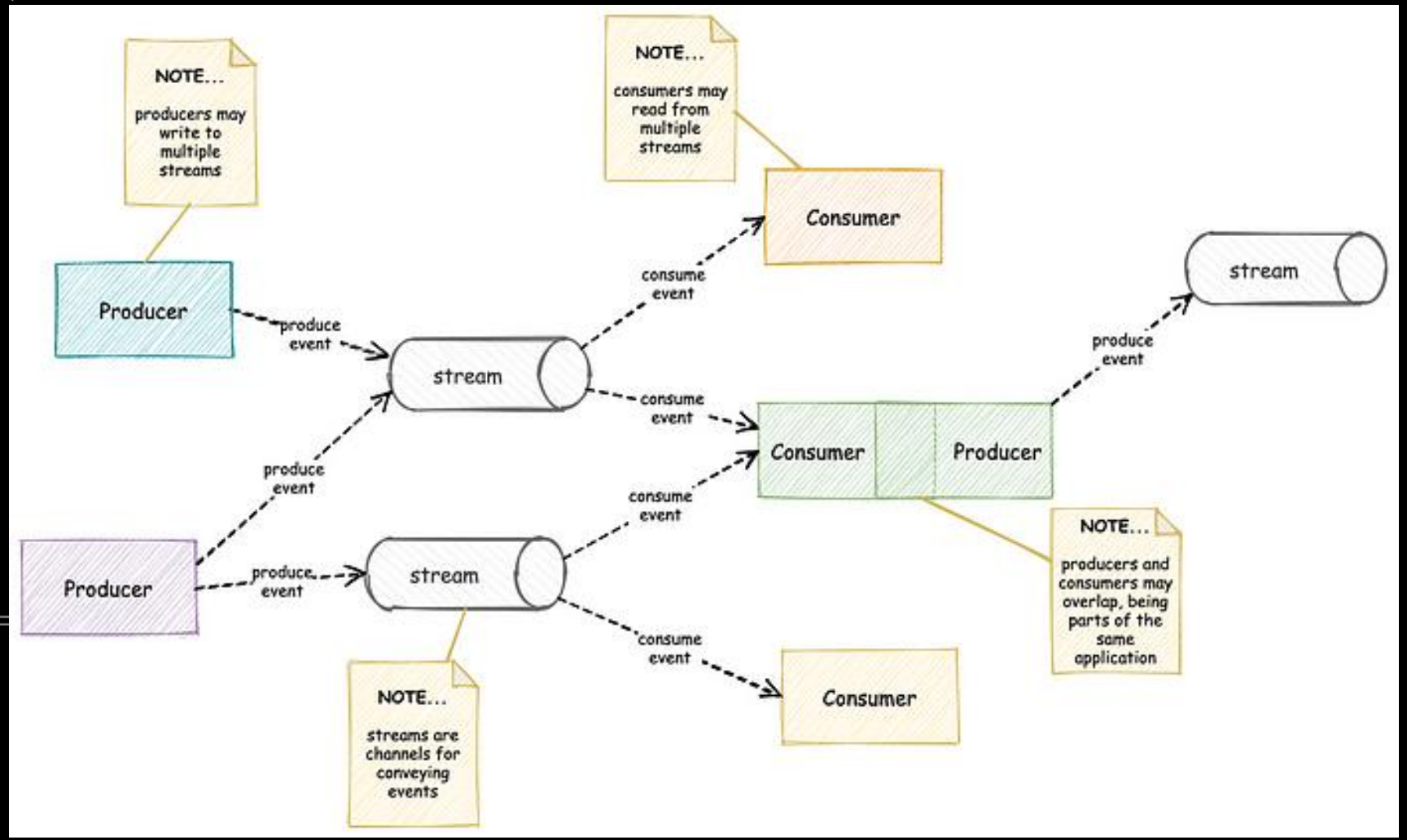
Exemplos de eventos

- ✓ Alteração de cadastro do cliente
- ✓ Confirmação de uma venda
- ✓ Alteração de uma cotação, temperatura ou movimento
- ✓ Comentário em um post
- ✓ Navegação em um App ou site

... Essas características exigem arquiteturas voltadas à eventos...

Arquitetura orientada a eventos

(Event Driven Architecture – EDA)



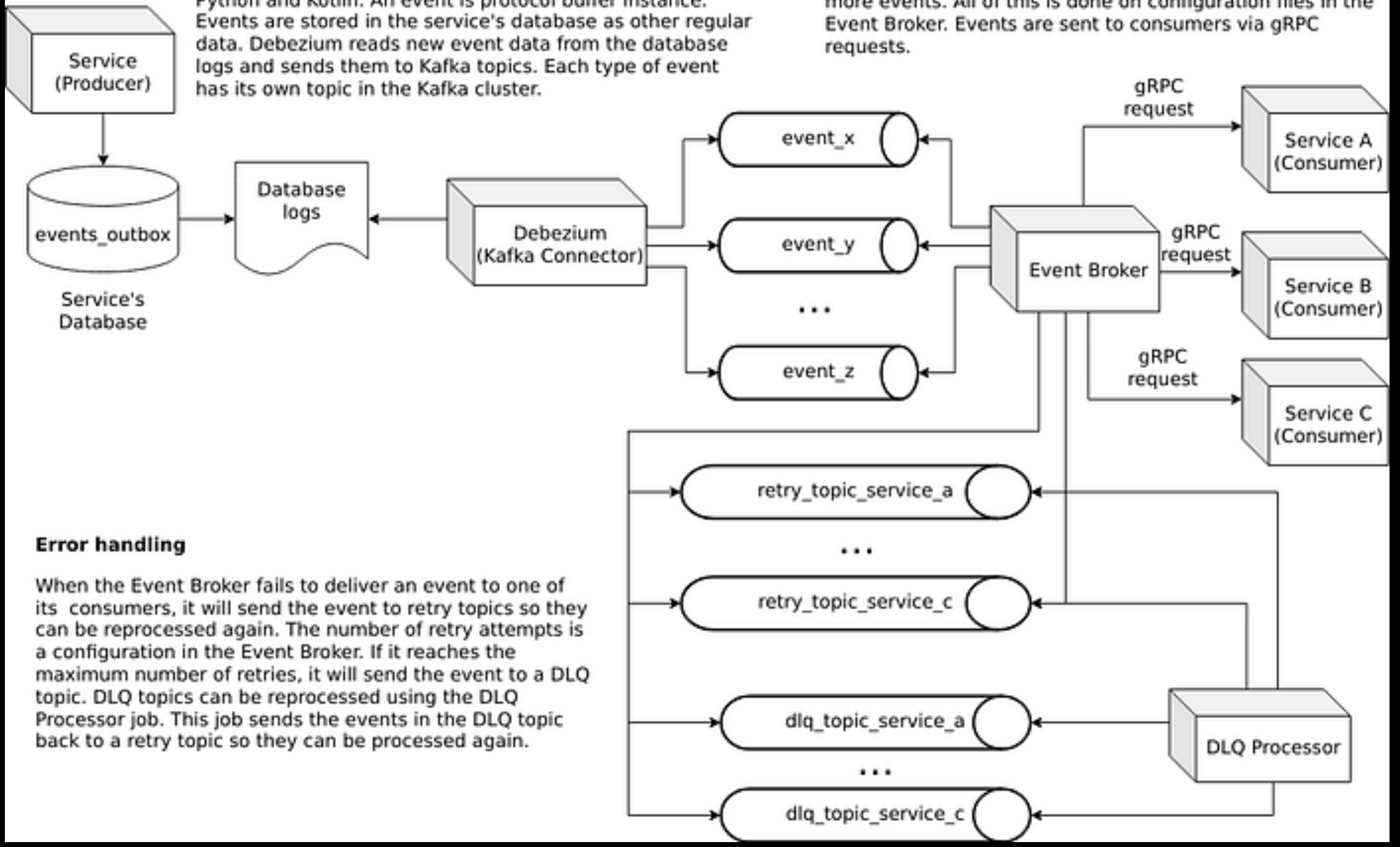
Ref.: https://medium.com/microservicegeeks/introduction-to-event-driven-architecture-e94ef442d824

Producing events

Events are produced using libs that are avaliable for both Python and Kotlin. An event is protocol buffer instance. Events are stored in the service's database as other regular data. Debezium reads new event data from the database logs and sends them to Kafka topics. Each type of event has its own topic in the Kafka cluster.

Consuming events

Once events are sent to their Kafka topics, a microservice called Event Broker will consume them and send events to registered consumers. A consumer may subscribe to one or more events. All of this is done on configuration files in the Event Broker. Events are sent to consumers via gRPC requests.

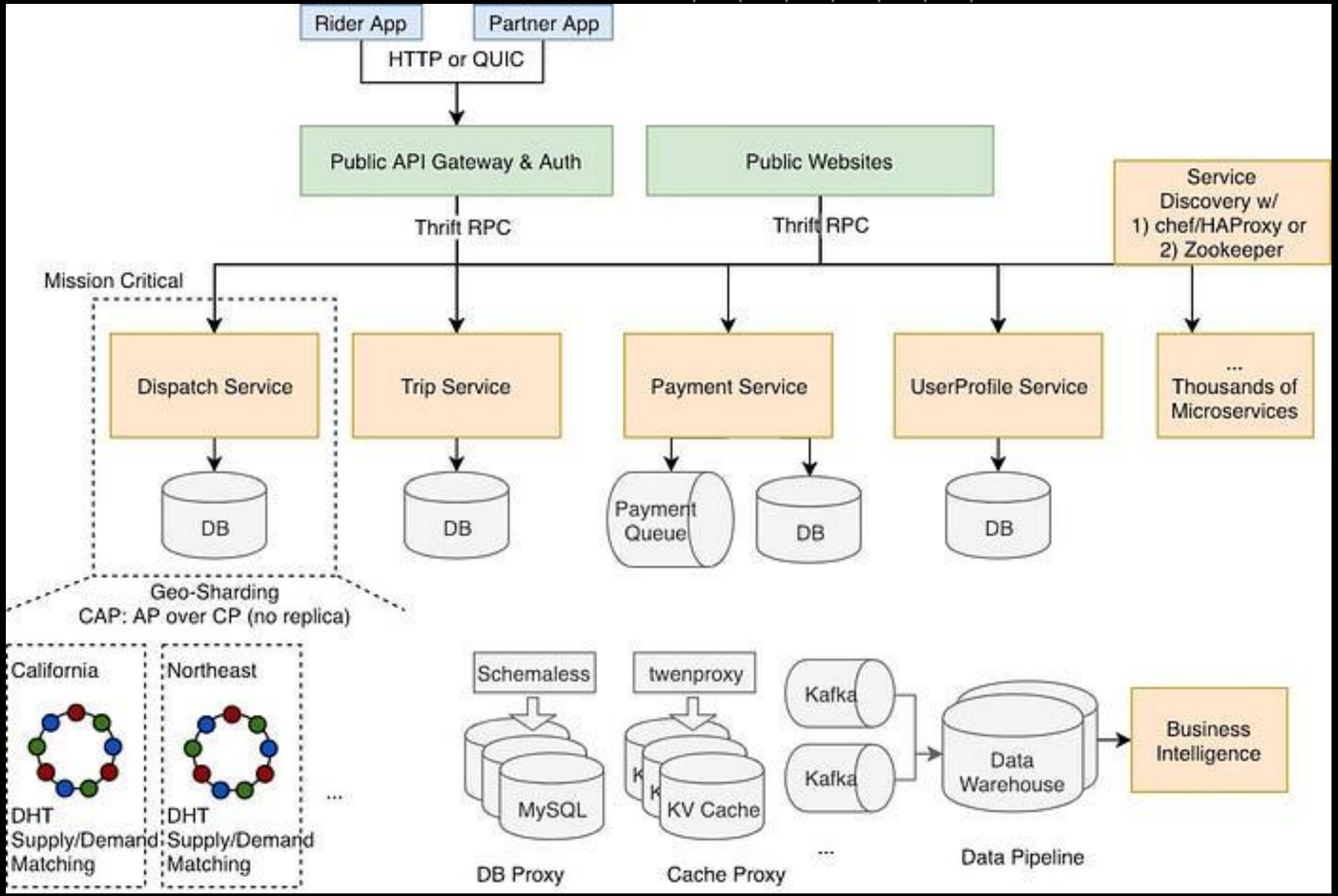


Error handling

When the Event Broker fails to deliver an event to one of its consumers, it will send the event to retry topics so they can be reprocessed again. The number of retry attempts is a configuration in the Event Broker. If it reaches the maximum number of retries, it will send the event to a DLQ topic. DLQ topics can be reprocessed using the DLQ Processor job. This job sends the events in the DLQ topic back to a retry topic so they can be processed again.

Exemplo arquitetura de eventos Loggi

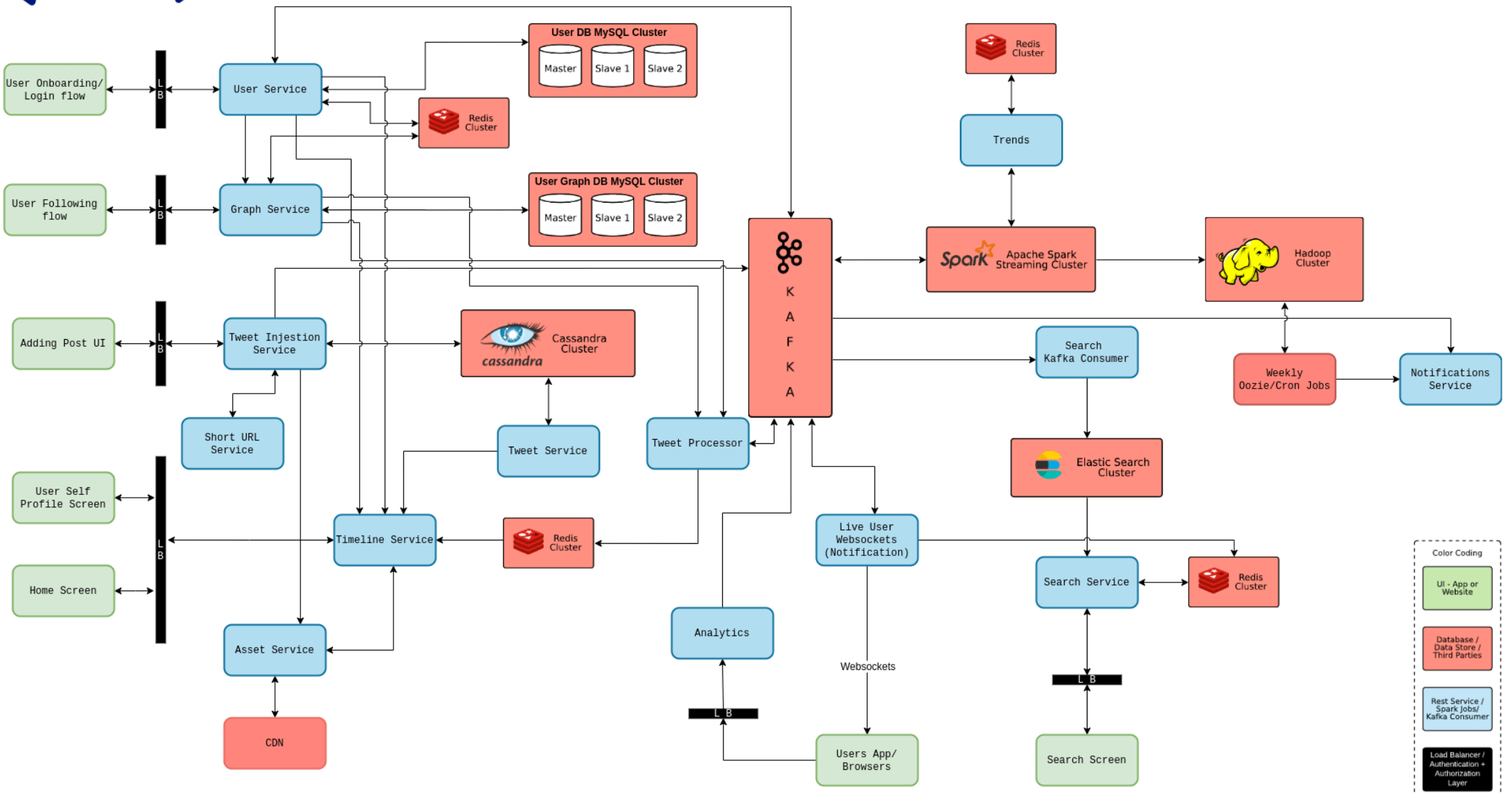
Ref.: https://partiu.loggi.com/designing-loggis-event-driven-architecture-fca8333263dd



Exemplo arquitetura de eventos Uber

Ref.: https://interviewnoodle.com/uber-system-architecture-40201134aaea

Twitter System Design



Ref.: https://www.codekarle.com/system-design/Twitter-system-design.html

Application Programming Interface (*API*)

São aplicações que contém um conjunto de regras com o objetivo de se comunicar com outra aplicação. Possibilita a comunicação/integração entre diferentes sistemas e empresas. Implementam controle e segurança em suas comunicações.

Funcionam da seguinte forma:

Protocolos de APIs

SOAP (Simple Object Access Protocol): Construído com XML, o SOAP permite que os endpoints enviem e recebam dados através de SMTP e HTTP. As APIs SOAP facilitam o compartilhamento de informações entre aplicativos ou componentes de software executados em ambientes diferentes ou escritos em idiomas diferentes.

RPC (Remote Procedure Call): O protocolo de chamada de procedimento remoto (RPC) é um meio simples de enviar vários parâmetros e receber resultados. As APIs RPC invocam ações ou processos executáveis, enquanto as APIs REST trocam principalmente dados ou recursos, como documentos. O RPC pode empregar duas linguagens diferentes, JSON e XML, para codificação; essas APIs são denominadas JSON-RPC e XML-RPC, respectivamente.

REST (Representational State Transfer): REST é um conjunto de princípios de arquitetura de API da Web. APIs REST—também conhecidas como API RESTful)—são APIs que aderem a certas restrições arquitetônicas REST. É possível construir APIs RESTful com protocolos SOAP, mas os dois padrões geralmente são vistos como especificações concorrentes.

Exemplos de APIs

OpenWeatherMap API: Uma API que fornece informações sobre o clima em todo o mundo. Endpoint: https://api.openweathermap.org/data/2.5/weather

Twitter API: A API do Twitter permite que os desenvolvedores acessem e interajam com os dados do Twitter em tempo real. Endpoint: https://api.twitter.com/1.1

Google Maps API: A API do Google Maps permite que os desenvolvedores integrem mapas interativos em seus aplicativos e serviços. Endpoint: https://maps.googleapis.com/maps/api

NASA API: A API da NASA fornece acesso a dados de observação da Terra, imagens de satélite e muito mais. Endpoint: https://api.nasa.gov/

Ferramentas de manuseio de APIs

Near Real Time
(*NRT*)

Tipos de processamento	Quando usar	Exemplos
Real-time	Quando você precisa de informações processadas imediatamente	Caixa eletrônico Sistema de radar
Near real-time	Quando a velocidade é importante, mas você não precisa dela imediatamente	Monitoramento de sistemas Sensores
Batch	Quando você pode esperar dias (ou mais) pelo processamento	Folha de pagamento Previsão de produção

Exercício: APIs com nifi

Bitcoin rate

Data
USD
GBP
EUR

RAW

Iniciem os dockers necessários

1. Abrir o prompt de comando
2. Abrir o diretório: C:\docker\5dts
   - *C:*
   - *cd \docker\5dts*
3. *Verificar a pasta "bigdata_docker"*
   - *dir*
4. *Acessar a pasta "bigdata_docker"*
   - *cd bigdata_docker*
5. *Iniciar os dockers*
   - *docker-compose up -d nifi*
   - *docker-compose up -d mongo*
   - *docker-compose up -d database*

```
{
  "time": {
    "updated": "Apr 21, 2023 22:28:00 UTC",
    "updatedISO": "2023-04-21T22:28:00+00:00",
    "updateduk": "Apr 21, 2023 at 23:28 BST"
  },
  "disclaimer": "This data was produced from the CoinDesk Bitcoin Price Index (USD). Non-USD currency data converted using hourly conversion rate from openexchangerates.org",
  "chartName": "Bitcoin",
  "bpi": {
    "USD": {
      "code": "USD",
      "symbol": "&#36;",
      "rate": "27,384.5367",
      "description": "United States Dollar",
      "rate_float": 27384.5367
    },
    "GBP": {
      "code": "GBP",
      "symbol": "&pound;",
      "rate": "22,882.2998",
      "description": "British Pound Sterling",
      "rate_float": 22882.2998
    },
    "EUR": {
      "code": "EUR",
      "symbol": "&euro;",
      "rate": "26,676.5369",
      "description": "Euro",
      "rate_float": 26676.5369
    }
  }
}
```


API do exercício

Endpoint:
https://api.coindesk.com/v1/bpi/currentprice.json

Esta API retorna o Bitcoin Price Index (BPI) atualizada a cada minuto

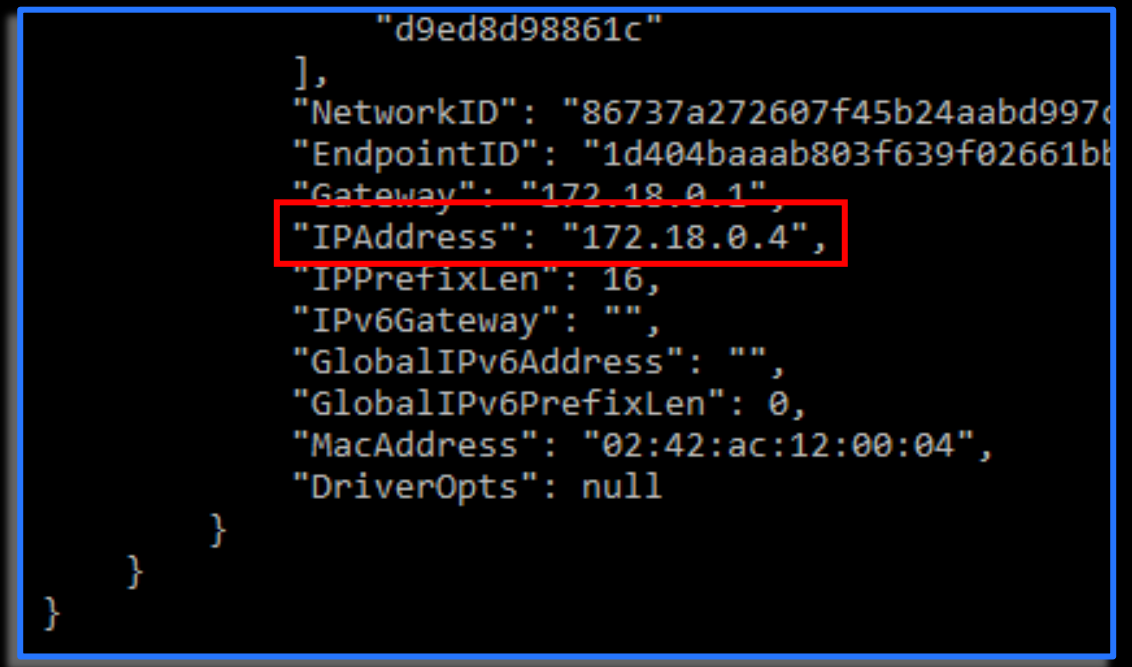
Dica: https://jsonformatter.org/

Exercício NRT

Preparar o mongo para receber os dados brutos:

1. Acessar o container do mongo
2. Acessar o mongo com o usuário root
3. Acessar o database dbAula
4. Criar uma collection chamada bitcoin_raw
5. Sair da CLI do mongo
6. Sair do docker mongo
7. Verificar qual o IP do docker mongo (*)

Script disponibilizado com os comandos



Exercício NRT

Criar fluxo para carga dos dados brutos no mongo usando o nifi:

Script disponibilizado com os comandos

8. Adicionar um processor InvokeHTTP (*)
9. Configurar a API de cotação de bitcoin
10. Adicionar processor UpdateAttribute
11. Configurar processor UpdateAttribute
12. Adicionar processor PutMongoRecord
13. Configurar o processor PutMongoRecord
14. Inicie os processors do nifi
15. Verificar os dados acessando o mongo
16. Acessar o database dbAula
17. Consultar a collection bitcoinraw (*)

Todo processor possui na aba “Scheduling” uma propriedade chamada “Run Schedule”. É o intervalo entre execuções. Zero significa que as execuções não terão intervalo entre elas.

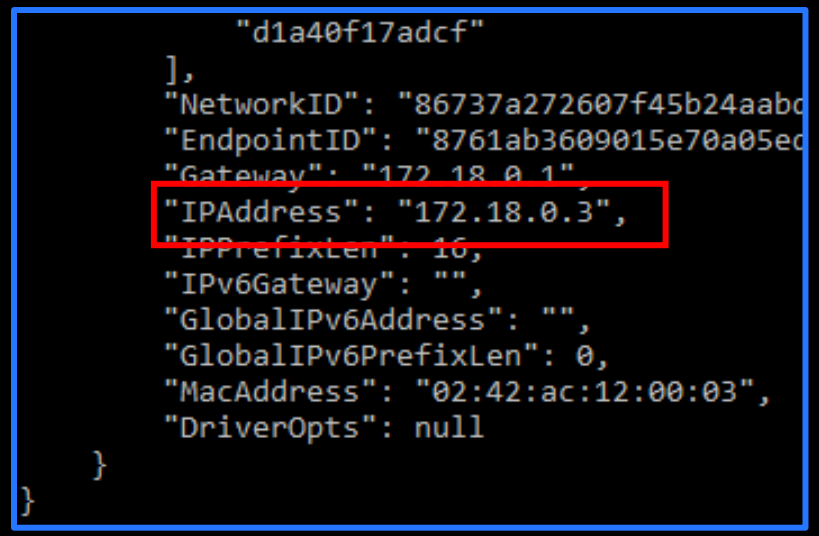
A camada de dados raw também poderia ser um diretório no HDFS

Exercício NRT

Preparar o MySQL para receber os dados tratados:

Script disponibilizado com os comandos

18. Acessar o container database
19. Acessar o mysql com o usuário root
20. Criar um database chamado dbAula
21. Acessar o database dbAula
22. Criar uma tabela chamada bitcoin_rate
23. Sair da CLI do mysql
24. Sair do docker database
25. Verificar qual o IP do docker database (*)
26. Baixar o arquivo "*mysql-connector-j-8.0.33.jar*" para sua máquina local
27. Copiar o arquivo do driver para o nifi



Exercício NRT

Script disponibilizado com os comandos

Evoluir fluxo para carga dos dados prontos no mysql usando o nifi:

28. Adicionar um processor EvaluateJsonPath (*)
29. Configurar o processor para extrair os campos data, usd, gbp e eur
30. Adicionar um processor AttributesToJson
31. Configurar processor AttributesToJson para gerar um novo json
32. Adicionar um processor PutSQL
33. Configurar a conexão do processor
34. Configurar instrução SQL para o PutSQL

Observe que na aba “Settings” todos os fluxos que não tiverem um direcionamento precisam ter marcada a opção de auto-terminar

Exercício NRT

Verificar os dados no mysql:

35. Acessar o docker database
36. Acessar o mysql com o usuário root
37. Acessar o database dbAula
38. Fazer um select na tabela bitcoin_rate

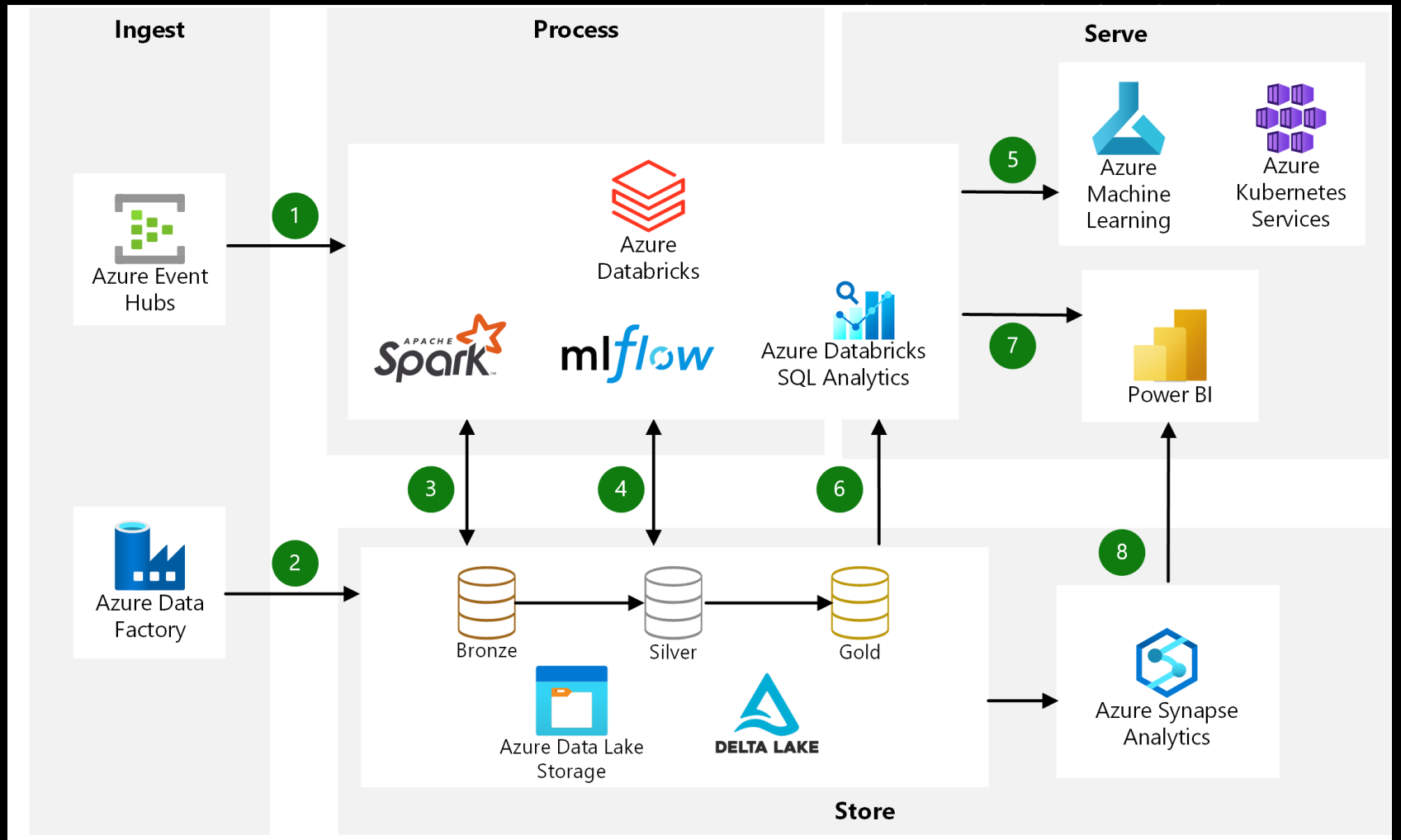
Script disponibilizado com os comandos

Questão

Neste exemplo de pipeline de dados, podemos tirar ideias práticas para nosso dia a dia?

Arquiteturas para analytics

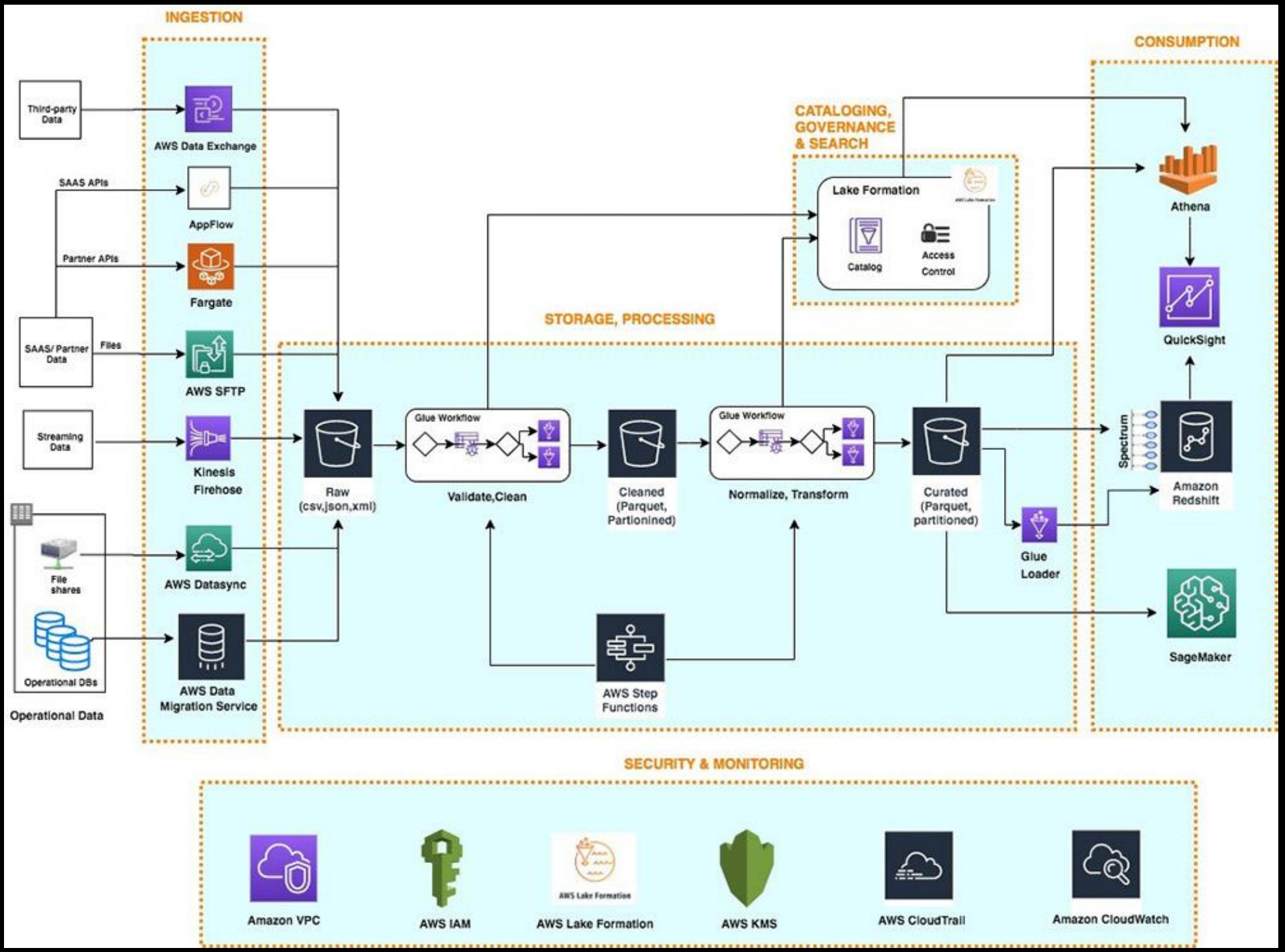
Landscape



Azure

Ref.: https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/azure-databricks-modern-analytics-architecture

Landscape

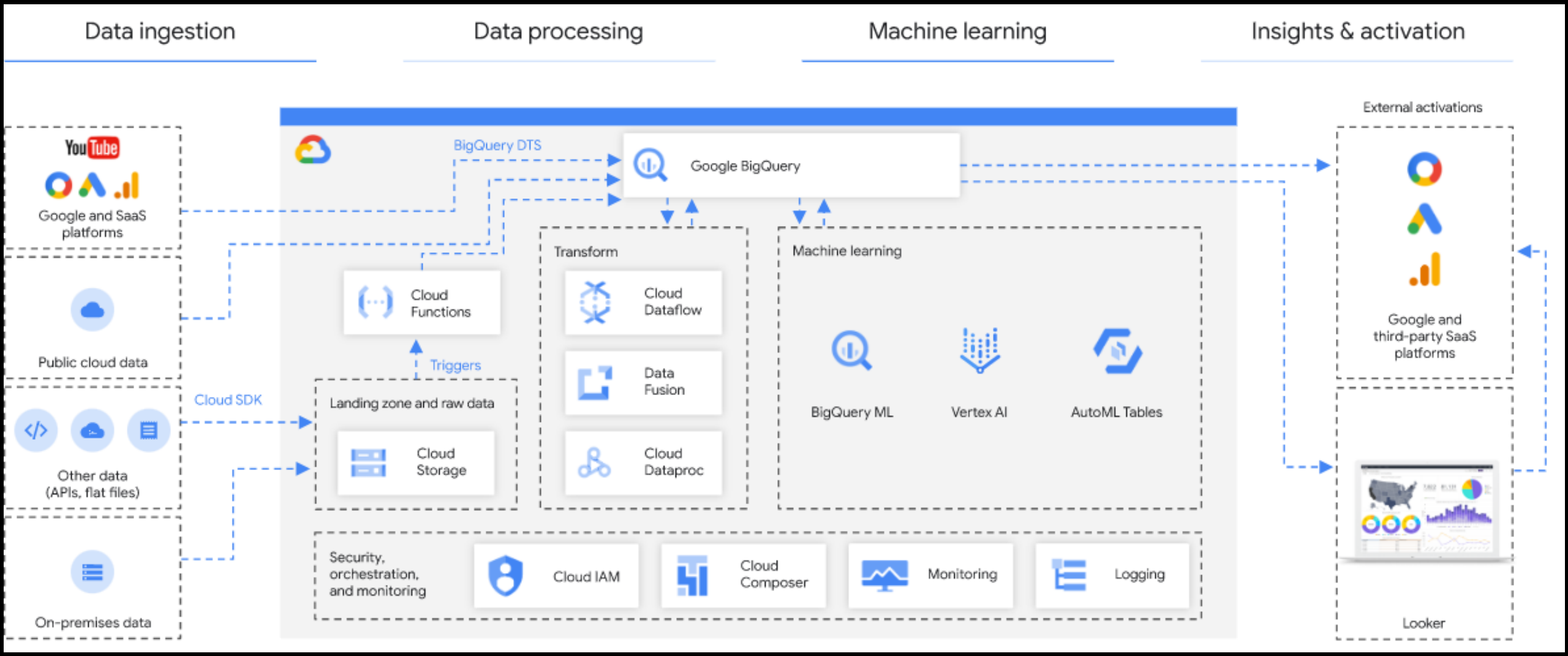


Ref.: https://aws.amazon.com/pt/blogs/big-data/aws-serverless-data-analytics-pipeline-reference-architecture/

Landscape

Google Cloud

FIAP



Ref.: https://cloud.google.com/architecture/marketing-data-warehouse-on-gcp?hl=pt-br

Real time analytics

Real-time analytics é uma técnica de processamento de dados que permite a coleta, análise e interpretação de informações em tempo real.

Diferente da análise com base em histórico e grandes volumes massificados, nesta abordagem cada evento é analisado individualmente para tomadas de decisões granulares.

Quando usar

- ✓ **Quando a necessidade de negócio demanda respostas em tempo real**
- ✓ **Análises de comportamentos e tendências do cliente**
- ✓ **Ofertar ao cliente o produto que ele necessita no momento certo**
- ✓ **Análises de sensores e dispositivos de missão crítica**

Exemplos de uso

Sugerir ao usuário a melhor rota com base no trânsito local

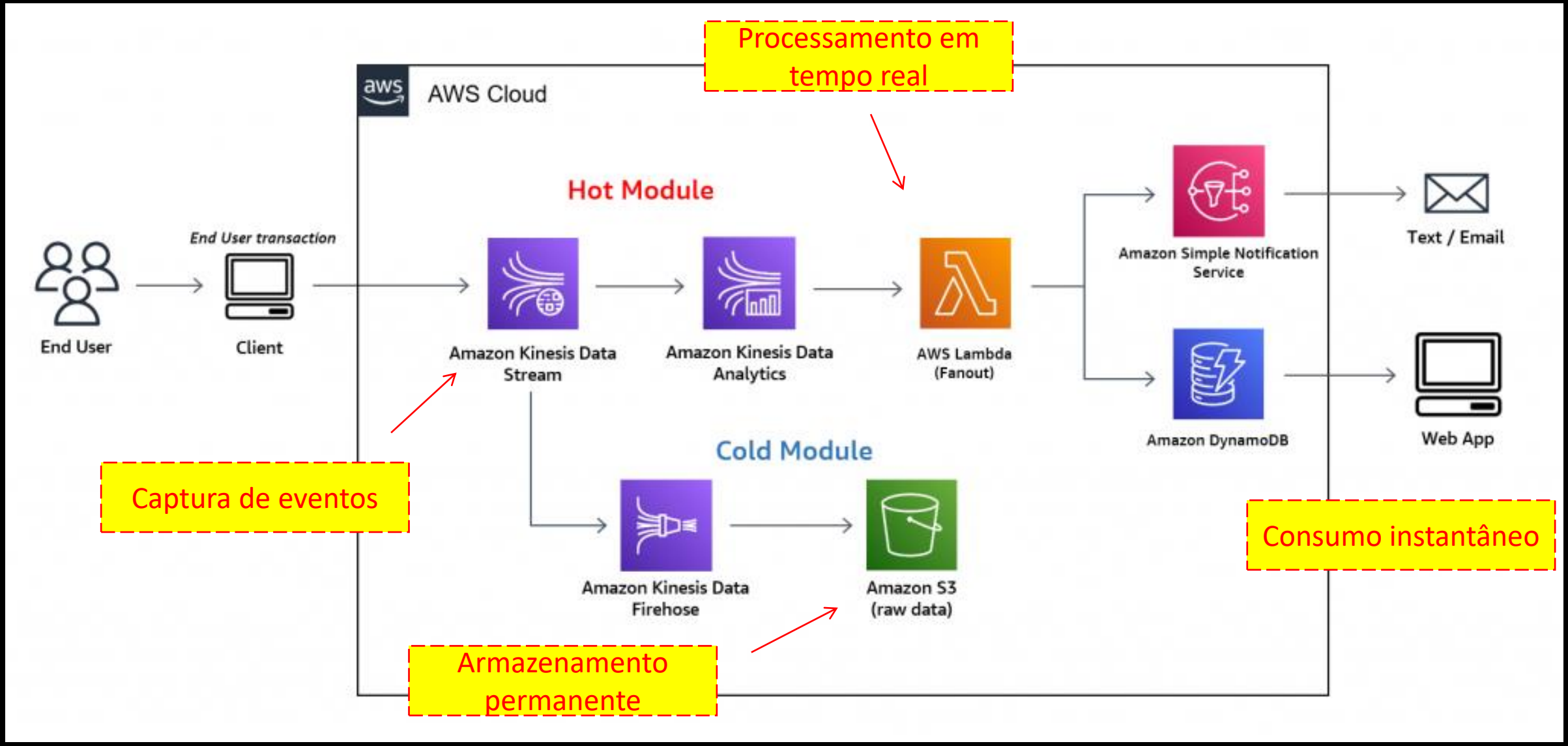
Sugerir compras que o usuário possa ter interesse com base na sua localização

Compartilhamento de viagens para otimizar despesas de deslocamento

Acionar equipes de segurança a partir da leitura de um detector de movimento

Oferecer crédito para um cliente que teve uma compra recusada

Arquitetura



Recomendações

- ✓ Vimos a importância de uma arquitetura de referência para dados, contendo regras de como os dados devem ser tratados dentro da empresa.
- ✓ Vimos que cada tecnologia possui especificidades para cada necessidade, portanto é importante saber qual tecnologia usar em cada situação
- ✓ As arquiteturas são bastante flexíveis para atender qualquer tipo de negócio desde grandes processamentos massivos até pequenos eventos em tempo real
- ✓ As tecnologias em Cloud trouxeram uma nova gama de produtos e possibilidades para explorar dados com maior velocidade, precisão e eficiência. Devemos estar atentos ao surgimento de novas tecnologias
- ✓ Verifique as práticas adotadas na empresa em que trabalha, identifique e sugira potenciais evoluções

Cuidados

- ✓ Na escolha de ferramentas, o melhor a se fazer é testar diferentes abordagens em diferentes ferramentas antes de decidir por qual seguir. Principalmente se houverem custos de aquisição e instalação importantes.
- ✓ Lock-in: Quando falamos de grandes volumes de dados, não é trivial movê-los de uma plataforma para outra. Portanto, esteja atento a ferramenta/tecnologia escolhida, pois sempre há um lock-in.
- ✓ Atenção aos níveis de controle implementados em cada processo. Quanto maior o controle é maior a segurança, contudo também é maior complexidade, esforço e custo de construção/operação. Implemente controles na medida necessária.
- ✓ Falamos sobre rastreabilidade e linhagem do dado, práticas de logging, aprovações de novos dados, auditorias e outras se utilizam desta rastreabilidade.

Conteúdos Adicionais para estudos

Integrações internas e externas

Presentation Layer

Presentation layer

- Relatórios de vendas
- Relatórios contábeis
- Indicadores departamentais
- Dados monetizáveis
- Dados para canais

Catalogação

Consumidores **internos** da empresa

Consumidores **externos** a empresa

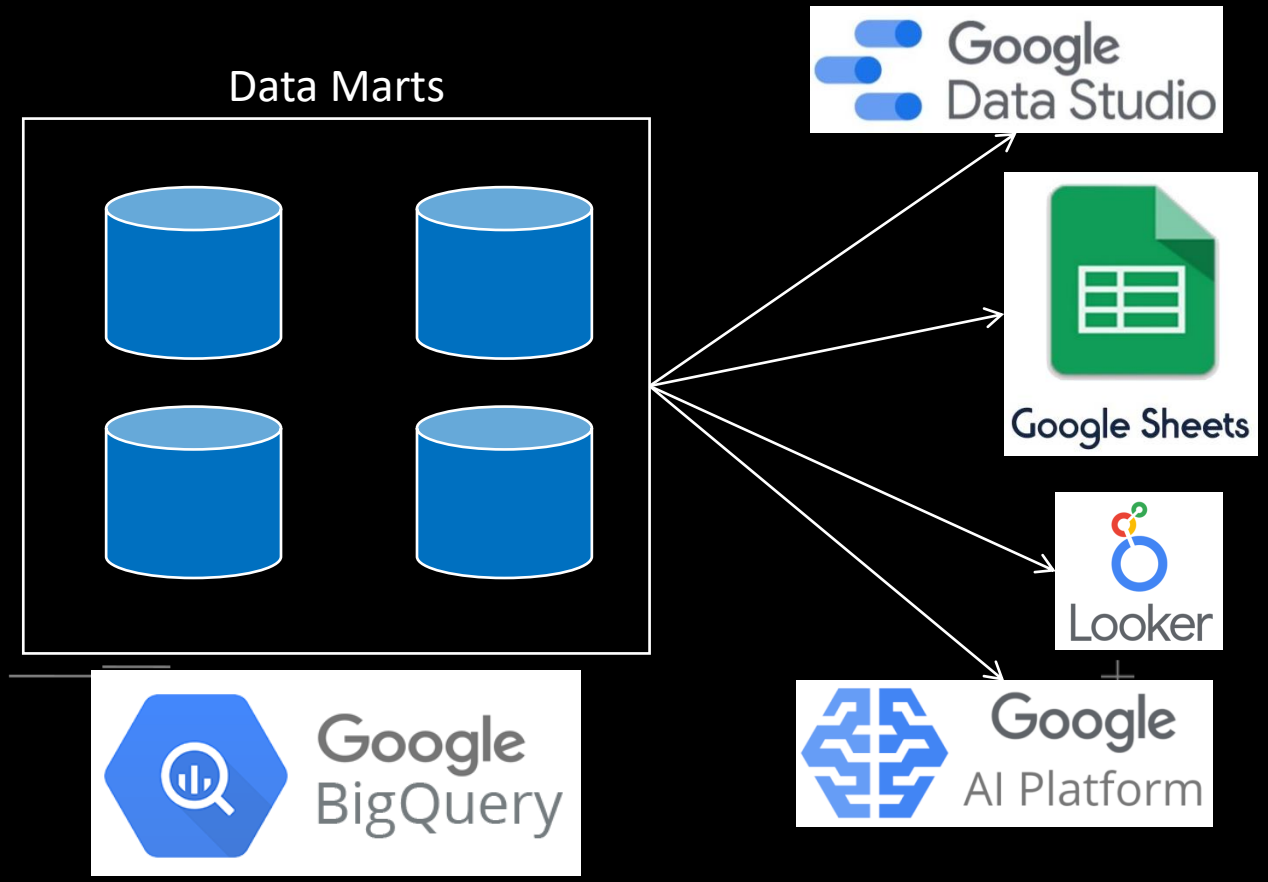
txt, avro,
csv, parquet

Canais de consumo de dados

- Site
- Apps
- SaaS
- Integrações (*)

Presentation Layer

Alguns exemplos de ferramentas para consumos de BI

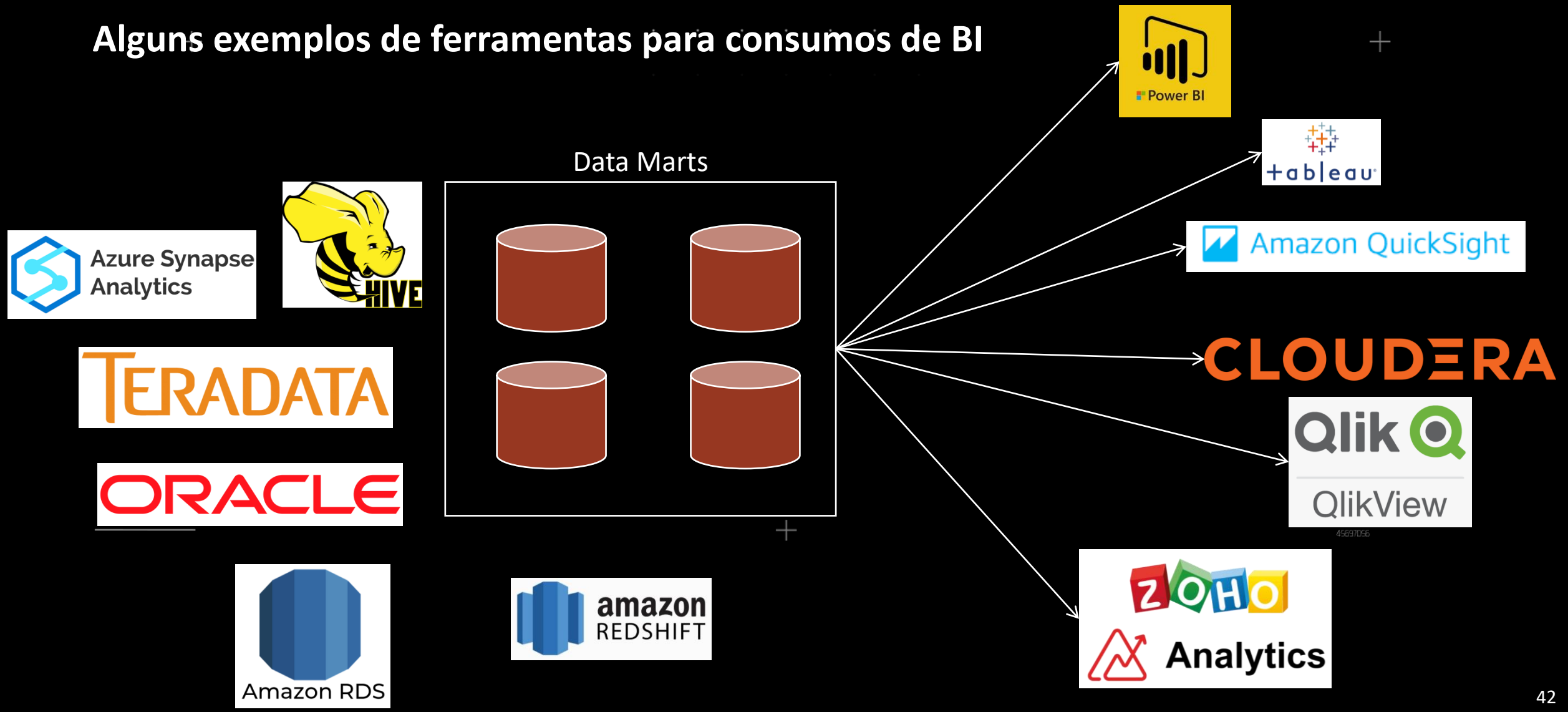


Características

- Dados em formato de indicadores para serem consumidos
- Catálogo detalhado disponível para os consumidores
- Usuários de negócio usando como base para tomada de decisões desde o nível estratégico até o tático e operacional

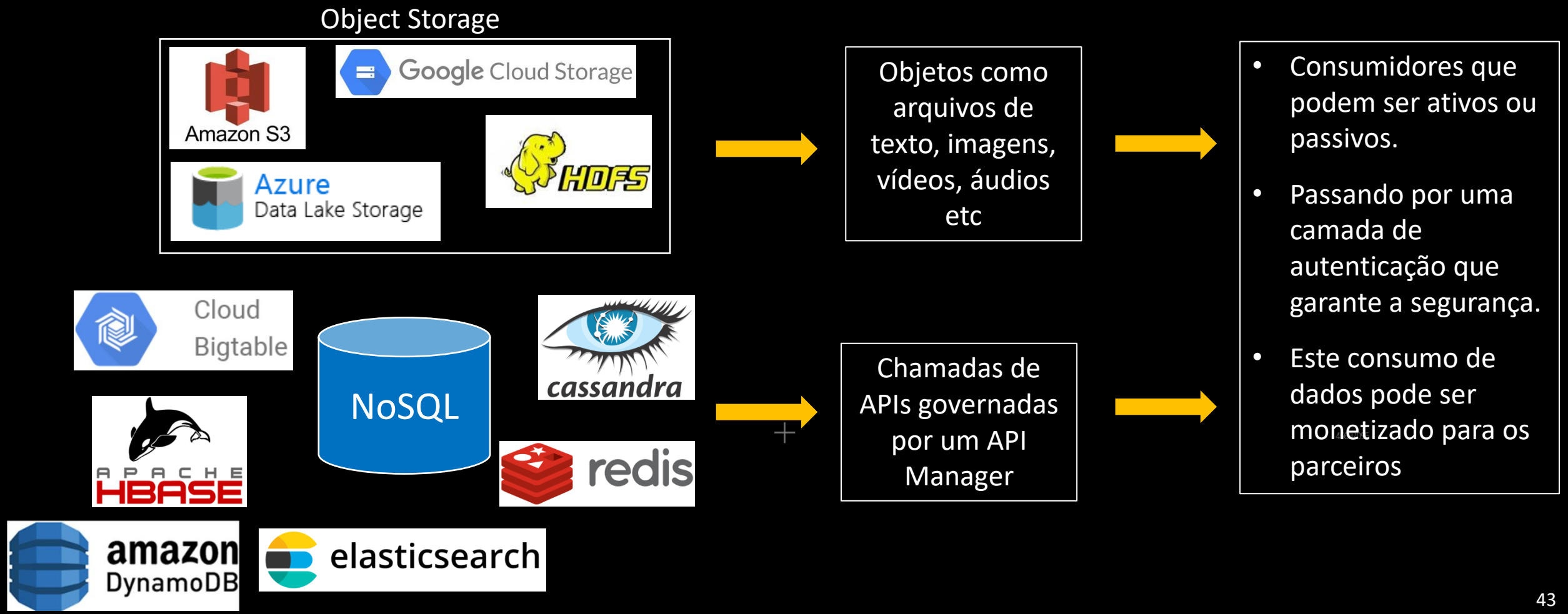
Presentation Layer

Alguns exemplos de ferramentas para consumos de BI



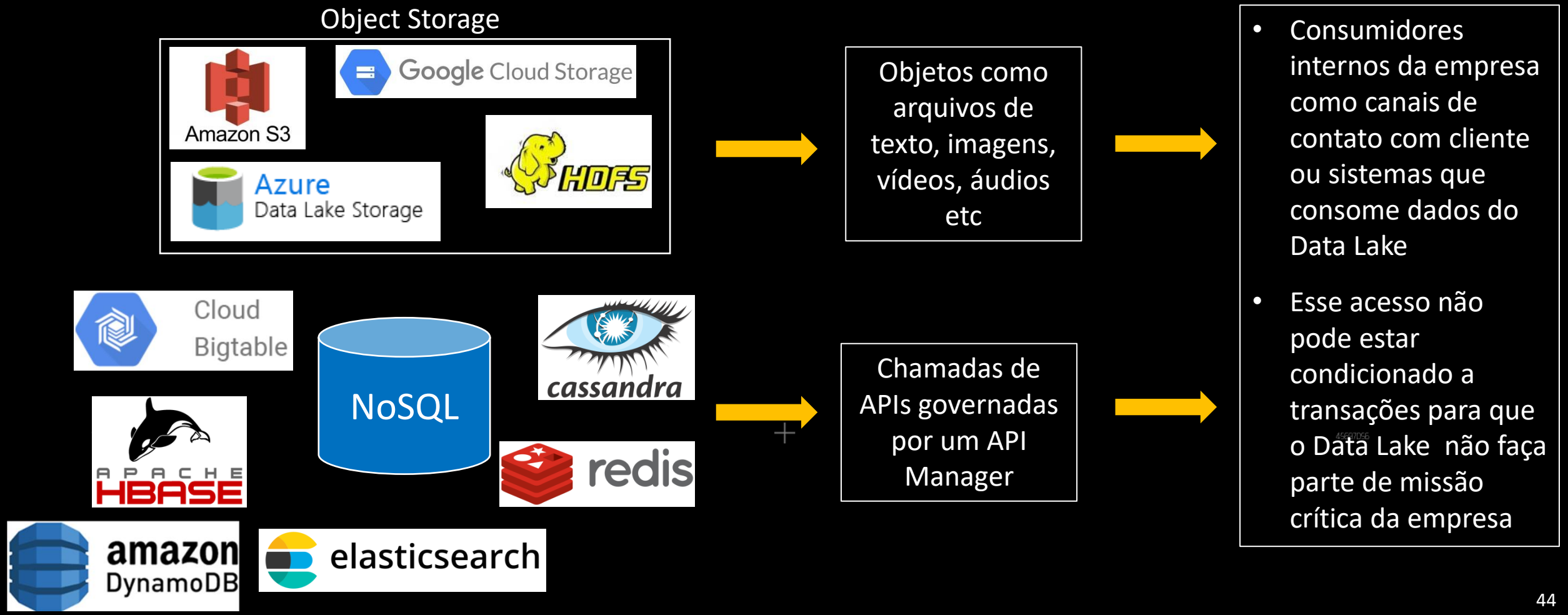
Presentation Layer

Alguns exemplos de ferramentas para consumos externos



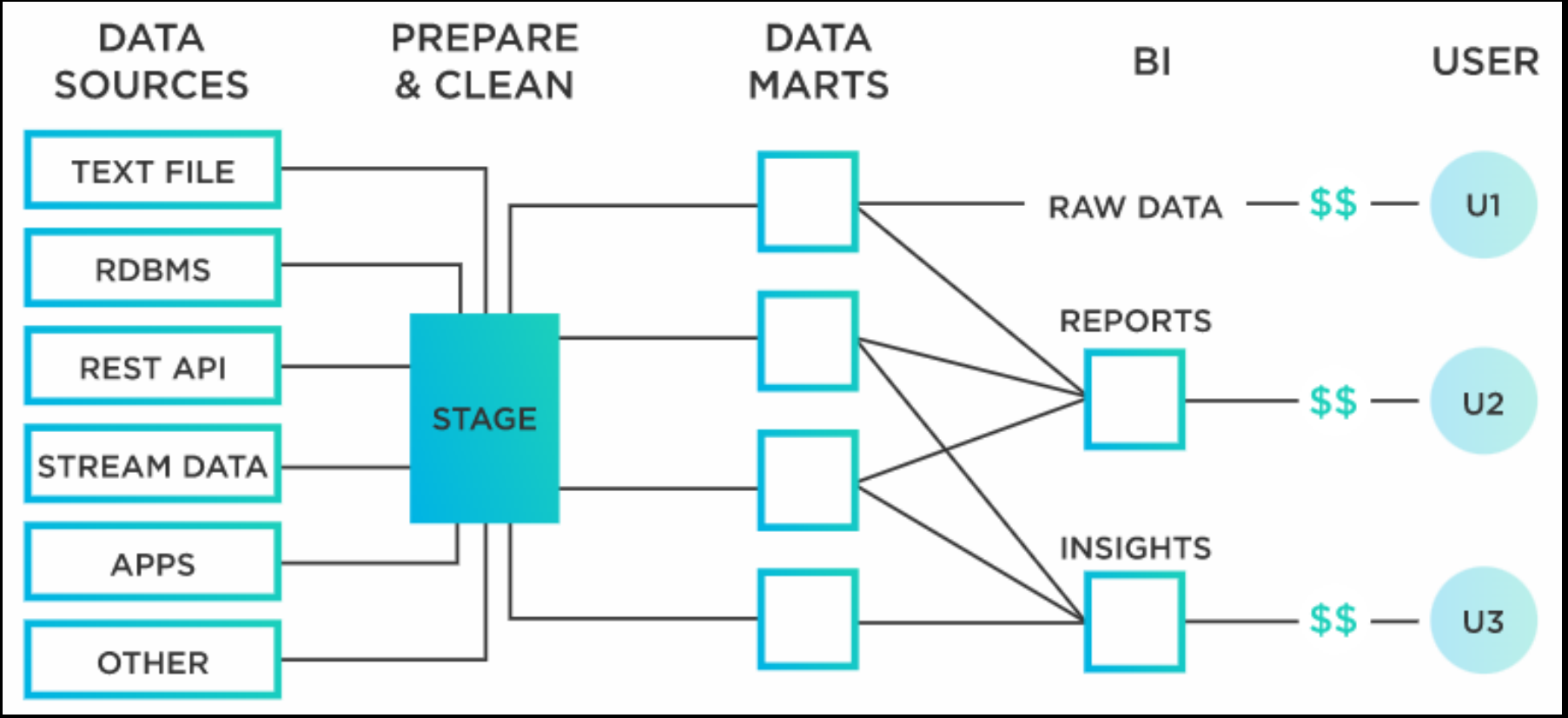
Presentation Layer

Alguns exemplos de ferramentas para consumos em canais



Definição de monetização

Data Monetization consiste no uso dos dados para obter benefícios econômicos para a empresa.



Ref.: https://www.tibco.com/reference-center/what-is-data-monetization

Exemplos

Uso interno entre os departamentos para compor margem de lucro e rateio de despesas

Exemplo: Custos para operar plataformas de dados podem ser rateados de acordo com o volume que cada consumidor utiliza

Uso externo com clientes e parceiros que tenham interesse em dados brutos ou transformados

Exemplo: Venda de imagem dos produtos fabricados para parceiros

Pontos de atenção

- **Garanta a legalidade do uso do dado para monetização**
- **Garanta o isolamento e segurança dos acessos monetizados**
- **Considere a venda dos dados como uma fonte de renda da empresa, identificando novas oportunidades onde pode ser aplicada**
- **É mais um diferencial competitivo no mercado, onde as empresas que estiverem a frente neste tema, terão maiores resultados**

Áreas de sandbox

Sandbox para cientista de dados

São **áreas** habilitadas para experimentação, teste, inovação, estudos de modelos e dados.

São importantes para que o cientista de dados tenha um espaço seguro para experimentações. Reduzindo risco de perda de dados e de impacto em sistemas operacionais da companhia.

A existência deve ser **temporária**, com objetivo **claro e específico**. Uma vez atingido o objetivo da experimentação, os **resultados finais** são armazenados e a área de sandbox deve ser **removida**.

Ciclo de vida de um sandbox

CRIAÇÃO

- ✓ Identificação das necessidades de negócio
- ✓ Autorização dos responsáveis pelos dados
- ✓ Coleta dos dados necessários
- ✓ Disponibilização de recursos necessários

USO

- ✓ Alimentação de dados complementares
- ✓ Testes e simulações de cenários previstos
- ✓ Validação dos resultados parciais
- ✓ Revisão dos cenários e variáveis para confirmação dos resultados

RESULTADOS

- ✓ Execução de piloto com dados reais
- ✓ Apresentação para o negócio do estudo de caso com resultados observados e comprovados
- ✓ Aprovação para evolução do piloto
- ✓ Em caso de falha: confirmação e registro das lições aprendidas

REMOÇÃO

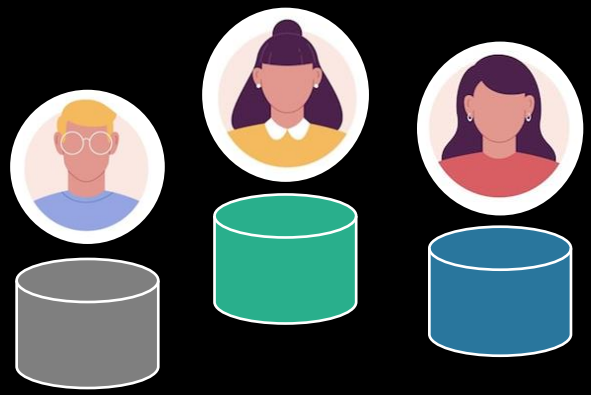
- ✓ Armazenamento dos estados e dados que forem necessário
- ✓ Remoção dos acessos ao ambiente
- ✓ Exclusão dos dados obsoletos e demais recursos que foram disponibilizados

Ciclo de vida

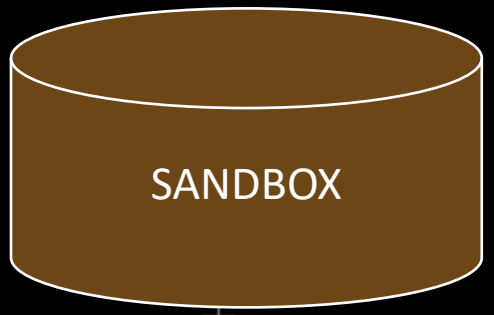
CRIAÇÃO

- ✓ Identificação das necessidades de negócio
- ✓ Autorização dos responsáveis pelos dados
- ✓ Coleta dos dados necessários
- ✓ Disponibilização de recursos necessários

Problema ou Hipótese



Dados iniciais de bases de sistemas legados



Bibliotecas e recursos necessários

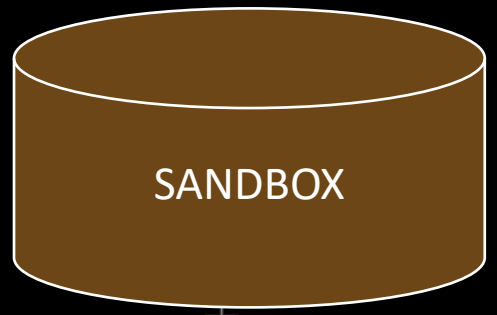
Ciclo de vida

USO

- ✓ Alimentação de dados complementares
- ✓ Testes e simulações de cenários previstos
- ✓ Validação dos resultados parciais
- ✓ Revisão dos cenários e variáveis para confirmação dos resultados

Fontes de dados externas

Simulação de modelos com diferentes variáveis, controles e técnicas



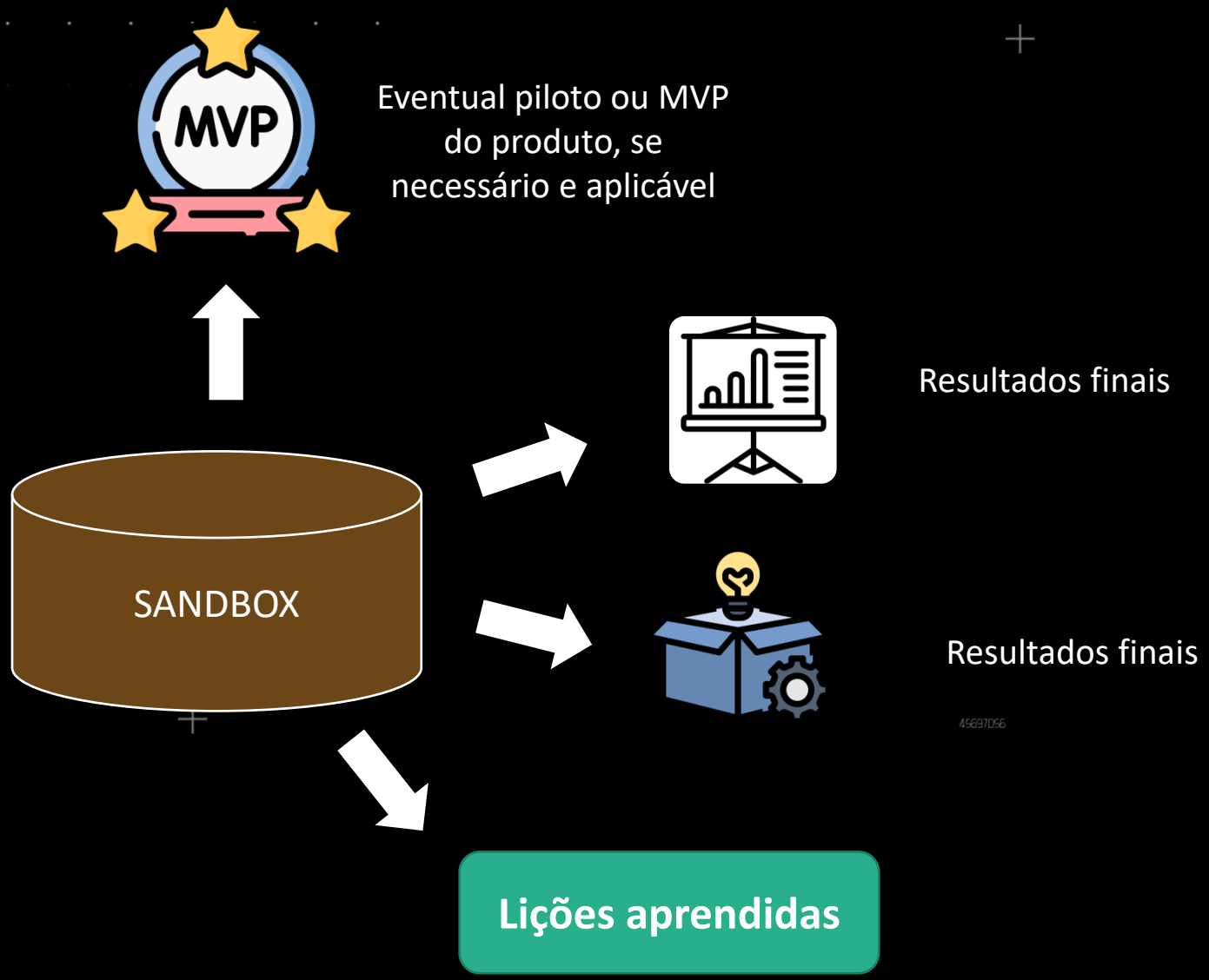
Resultados parciais

Resultados obtidos precisam ser validados e confrontados para aumentar sua segurança e qualidade

Ciclo de vida

RESULTADOS

- ✓ Execução de piloto com dados reais
- ✓ Apresentação para o negócio do estudo de caso com resultados observados e comprovados
- ✓ Aprovação para evolução do piloto
- ✓ Em caso de falha: confirmação e registro das lições aprendidas

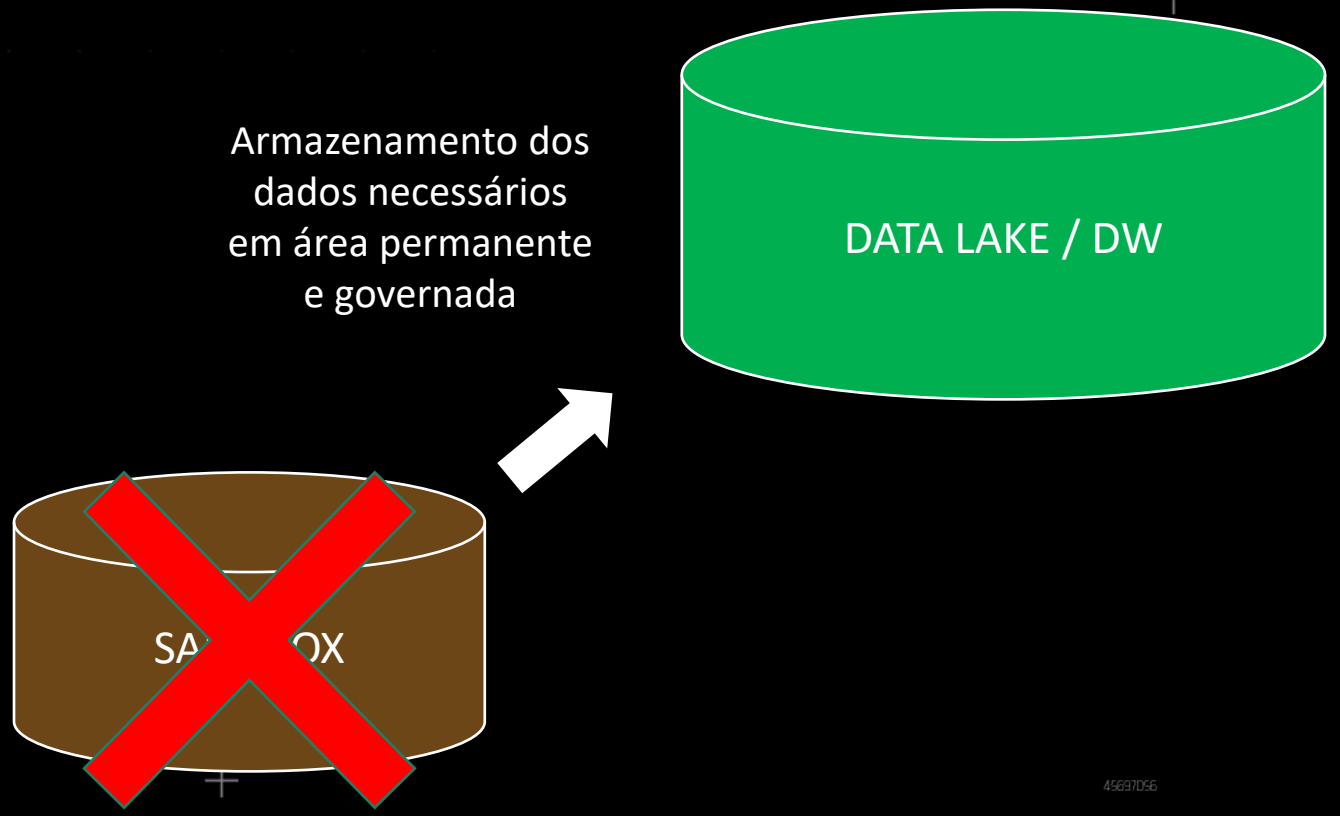


Ciclo de vida de um sandbox

REMOÇÃO

- ✓ Armazenamento dos estados e dados que forem necessário
- ✓ Remoção dos acessos ao ambiente
- ✓ Exclusão dos dados obsoletos e demais recursos que foram disponibilizados

Armazenamento dos dados necessários em área permanente e governada



Sandbox para cientista de dados

Características:

- ✓ Isolado de sistemas e bases corporativas
- ✓ Governança e restrições flexíveis para dar autonomia ao cientista de dados
- ✓ Agilidade para inovações e hipóteses baseadas em dados
- ✓ Melhor *Time to Market* e competividade para a empresa
- ✓ Ambiente segregado e escalável, provisionado e removido via automação

Ciclo de vida do dado

Ciclo de vida do dado

Criação

Fontes de dados. Podem surgir dentro ou fora da empresa em qualquer uma de suas operações, análise, relatórios ou produzido a partir de dados anteriores.

Como vimos antes, a empresa não necessariamente terá interesse em todos os dados e não precisará estabelecer um ciclo de vida para todos eles.

Armazenamento

Coleta e ingestão do dado. Uma vez dentro do Data lake, o dado estará sujeito as políticas de segurança, uso, armazenamento e tratamento dos dados. Como vimos, os dados podem ser não-estruturados, semi-estruturados ou estruturados.

Ciclo de vida do dado

Consumo, compartilhamento e aplicação dos dados às necessidade de negócio da companhia.
Como vimos nas transformações, novos dados podem ser criados através da análise de dados anteriores.

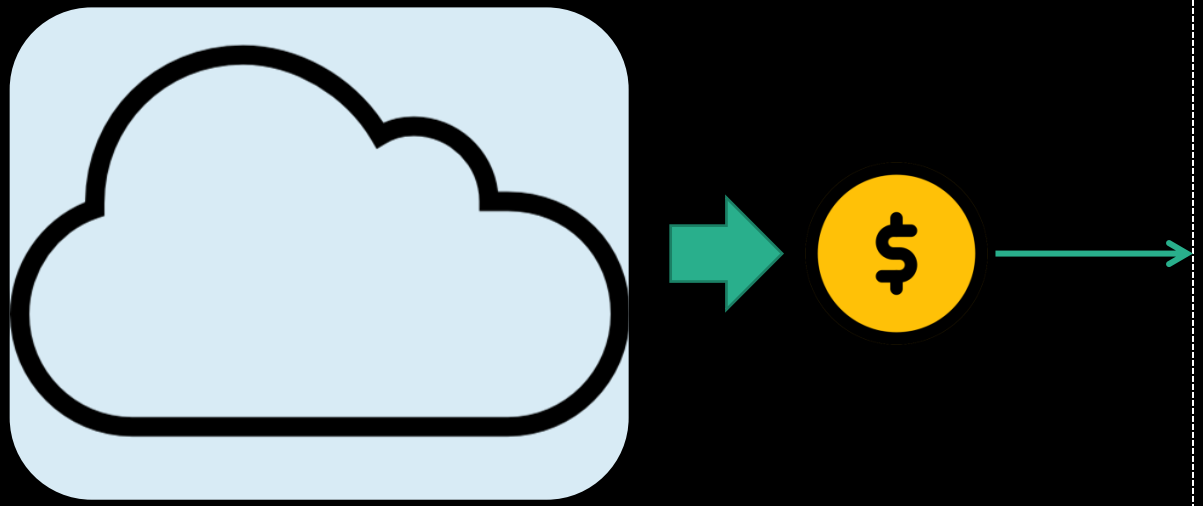
É importante ter uma política de arquivamento para os dados que são utilizados com baixíssima frequência pela empresa, alinhada à estratégia e necessidade do negócio. O uso mais comum é o de arquivamento para segurança ou por regulamentação do negócio.

Passado o período de arquivamento necessário por regulamentação ou por interesse de armazenamento dos dados por parte da empresa. A remoção dos dados reduz custos de armazenamento.

Ambientes on-premises

- Rateio não exato das despesas
- Dificuldade nos estudos de casos de negócio
- Afeta diferentemente a margem de cada produto da empresa
- Tempo elevado de provisionamento

Ambientes Cloud Pública



- Cobrança conforme o uso
- Pagamentos distribuídos por serviço
- Licenças, espaço, equipes e equipamentos inclusos no custo do serviços
- Provisionamento de ambientes imediato
- Infra como código implementada em qualquer lugar do mundo
- Disponibilidade assegurada pelo provedor de Cloud

- **Mas isso não é o suficiente...**

5 elementos de FinOps

Responsabilidade e Capacitação

- Estabelecer e treinar equipes multidisciplinares para definição e implementação de governança no gerenciamento dos gastos financeiros

Medição e realização

- Criação de indicadores de sucesso da adesão da cultura
- Visão de valor para o negócio

Otimização de custos

- Dedicação para identificar otimizações de recursos, preços e arquitetura dos consumos.

Planejamento e estimativa

- Os times podem facilmente perder o controle de gastos na nuvem, é essencial planejar os custos e verificar se estão dentro do previsto

Ferramentas e aceleradores

- Uso de ferramentas de gerenciamento de custos em cloud
- Painéis e relatórios de prestação de contas
- Automação com monitoramento e alertas dos custos

MBA+

profivan.gancev@fiap.com.br
profleandro.mendes@fiap.com.br