Data Warehouse e Data lake

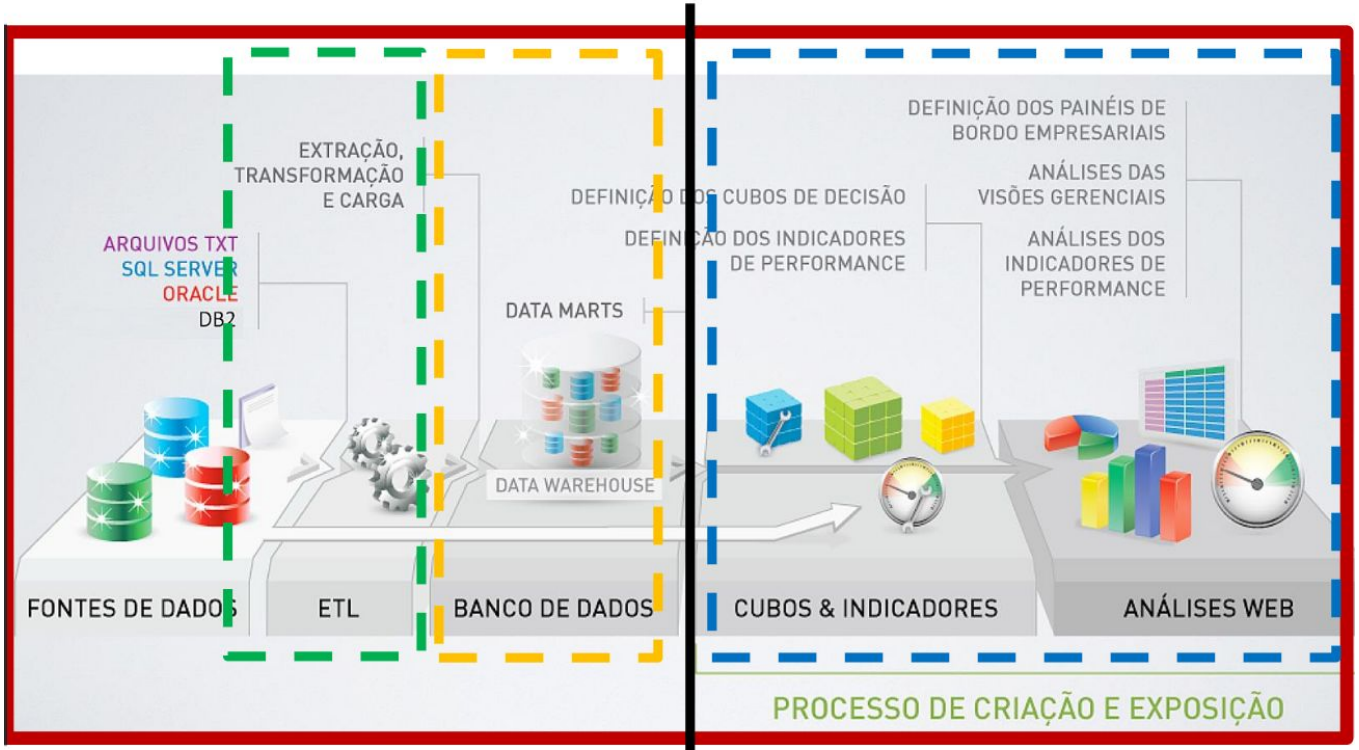
Aran Bey Tcholakian Morales
Richard Henrique de Souza
Saulo Popov Zambiasi

Sistemas Analíticos

O **objetivo** dos **sistemas analíticos** é definir regras e técnicas para a formação adequada dos **dados** da organização, visando transformá-los em depósitos de **informações** e **conhecimento** que atendam às necessidades dos processos de gestão e de decisão.

Em outras palavras, isso quer dizer transformar os **dados** em **informações** e **conhecimentos** relevantes para suprir as necessidades **gerenciais** de apoio aos **processos** de **gestão** e de **decisão**.

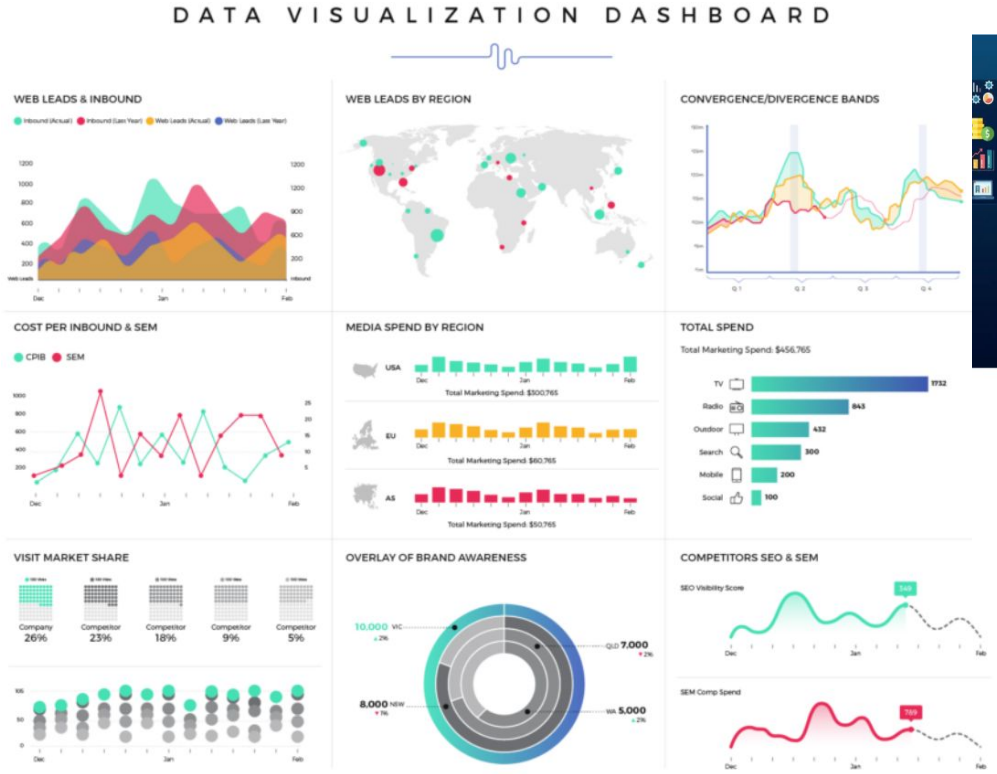
Sistemas Analíticos



Equipe técnica que desenvolve e oferece suporte ao sistema

Analistas de negócio

Sistemas Analíticos



Data Warehouse vs Data Lake

Data Warehouse - Data Lake

O **big data** fez que a **definição de dados analisáveis** mudar

Não existem mais apenas dados corporativos e estruturados (sistemas transacionais), mas todos os tipos de dados, internos e externos da organização.

Novos padrões arquitetônicos precisaram ser desenvolvidos para aproveitar o poder dos dados

Ou seja, **dar sentido aos dados** e **transformá-los em informação e conhecimento**.

Data Warehouse - Data Lake

Com as mudanças no paradigma de dados, surgiu o conceito: **Data Lake**.

Como a água no lago, os dados no Data Lake estão na forma mais pura possível e atende a **diferentes necessidades**.

Data Warehouse - Data Lake

Os **Data Warehouses** contêm **dados estruturados**, e a maneira como os dados são armazenados, de quais atributos estão disponíveis e até os formatos destes atributos é acordada com antecedência e o banco de dados segue essa estrutura.

A **consistência** e a **estabilidade** significam que os Data Warehouses podem atender a consultas de vários tipos de funções na organização.

Esse processo é muito **estruturado**, **previsível** e **eficiente**, mas também é **demorado, difícil** e tem um **custo alto** associado.

Data Warehouse - Data Lake

Os **Data Lakes** são outro meio de armazenar dados, **sem o esquema rigoroso** de um Data Warehouse.

No Data Lake, é muito mais fácil carregar os dados comparado aos Data Warehouse, mas as consultas são muito mais complexas para serem construídas, o que limita o uso dos Data Lakes, isto é, os **Data Lakes levam muito mais tempo para retornar resultados quando comparados aos Data Warehouses**.

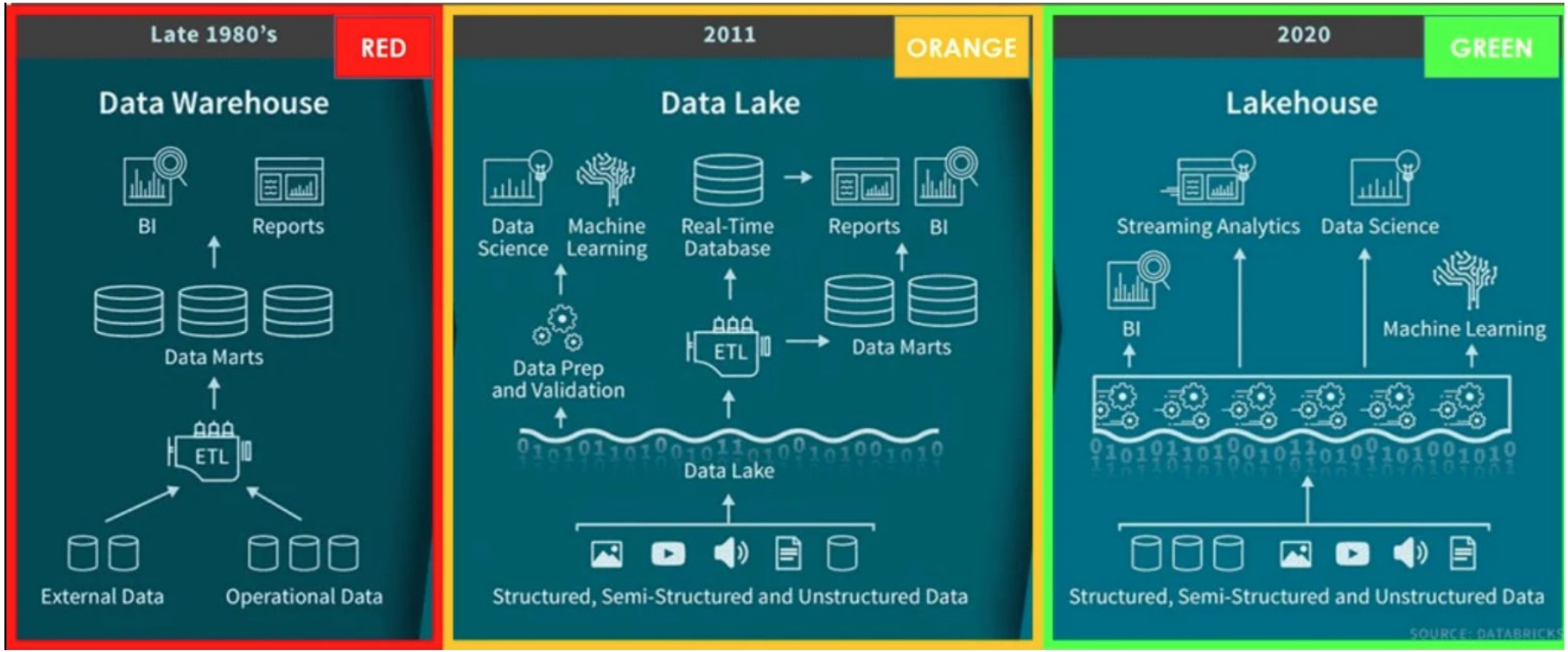
Data Warehouse vs Data Lake

Data Warehouse	Data Lake
Usa dados de diferentes fontes, predominantemente de sistemas transacionais.	Armazena dados estruturados e não estruturados no formato nativo (bruto).
Utilizado para relatórios e análises de dados “previamente conhecidos” para os negócios.	Não requer conhecimento prévio das análises que você acha que deseja realizar.
Representa uma imagem abstraída do negócio organizada por área ou assunto.	Todos os dados são carregados a partir de sistemas de origem.
Consistem de dados extraídos de sistemas transacionais e de métricas quantitativas e atributos que os descrevem.	Abrange todos os tipos de dados, independentemente da fonte e da estrutura.
São de fácil utilização, por serem estruturados e ter relatórios e análises previamente definidos e orientados.	São de difícil utilização, geralmente buscam análises novas (descoberta), que não estão definidas.

Data Warehouse vs Data Lake

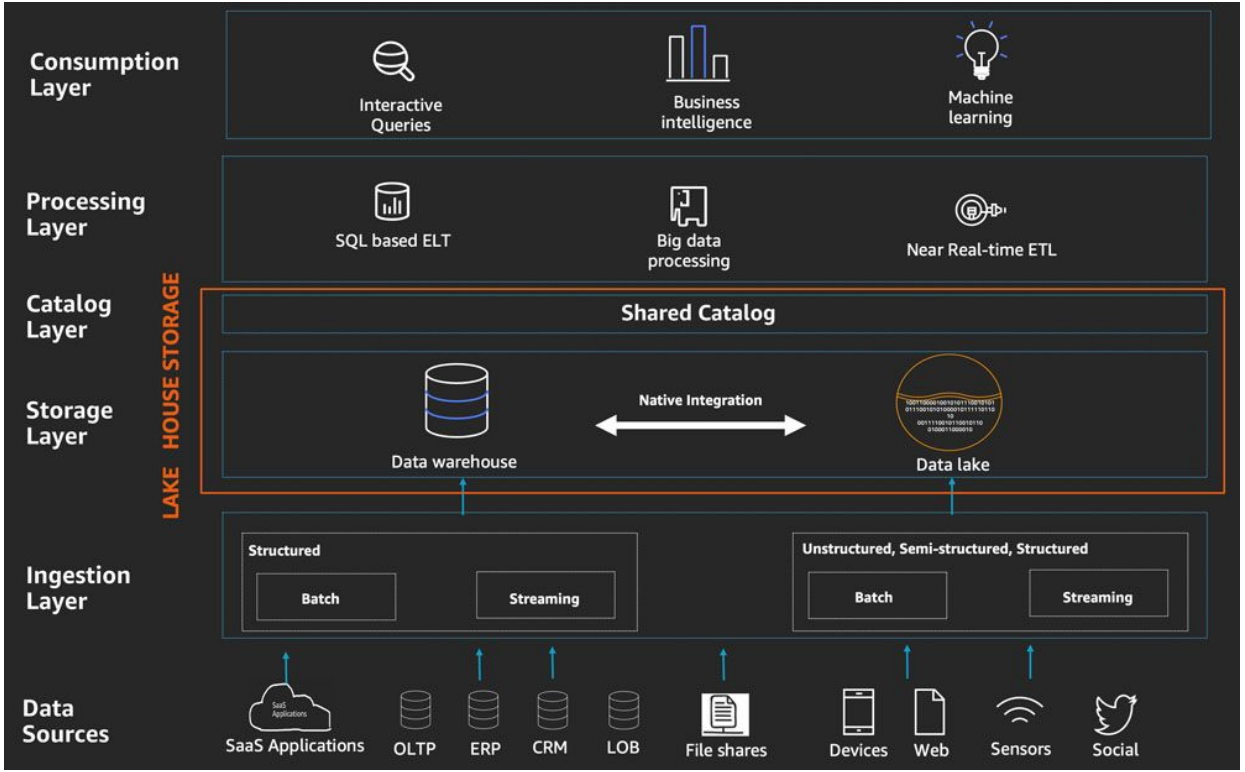
Data Warehouse	Data Lake
Dados limpos, transformados e integrados.	Dados brutos, sem integração nem consistência.
As tecnologias e as metodologias estão consolidadas e maduras (tem quase 30 anos).	As tecnologias e as metodologias estão em construção e são recentes.
Requer um investimento significativo com antecedência, mas, em troca, oferece a capacidade de análise de forma fácil.	Tem quase um potencial ilimitado, mas requer transformações antes de obter insights.
• Modelo de dados fácil de entender, os dados estão limpos, transformados e estruturados, mas leva tempo (custo) para construir. • Isto é, uma quantidade considerável de tempo é gasto analisando fontes de dados e compreendendo os processos de negócios: primeiro entender e depois carregar.	• Os dados são armazenados no nível nativo, em um estado não transformado. • Não existe “modelo de dados” definido. • A filosofia é: primeiro carregar para depois, entender, transformar e limpar.

Data Warehouse - Data Lake



REF: https://goodstrat.com/2020/05/16/why-i-called-bullshit-on-the-data-lakehouse-nonsense/

Data Warehouse - Data Lake



REF: https://aws.amazon.com/pt/blogs/big-data/build-a-lake-house-architecture-on-aws/

Tecnologias do Big Data

Tecnologias de armazenamento, processamento e análise de dados

As tecnologias que sustentam o Big Data, podem ser analisadas sob as seguintes óticas:

- **Infraestrutura** - que armazenam os dados: bancos NoSQL e NewSQL;
- **Processamento massivo de dados** - Hadoop e Spark (entre outras);
- **Análise** - ferramentas de análises, transformando os dados em valor para o negócio;

Armazenamento e de Processamento de Dados

NoSQLs - Não são relacionais e/ou não usam SQL

Google File System (GFS): sistema de arquivo distribuído entre vários servidores que dividia os arquivos em blocos (2003), para gravar grandes volumes de dados não estruturados (páginas HTML), que não cabiam em um único servidor

MapReduce: um framework que facilita extrair informação de dados armazenados no GFS, de forma paralela para aumentar a velocidade (2004)

Amazon iniciou o desenvolvimento do banco de dados comercializado como Dynamo DB (2004)

Armazenamento e de Processamento de Dados

O Google iniciou o desenvolvimento do BigTable (2006).

O LinkedIn iniciou o projeto Voldemort criando uma base de dados não SQL para gerenciar os seus 135 milhões de usuários (2007).

O Facebook baseou-se nas soluções da Amazon e do Google e desenvolveu a sua solução, que passou a ser distribuída como código aberto com o nome de Apache Cassandra (2008).

O Yahoo criou o Sherpa (2009).

Armazenamento e de Processamento de Dados

Key-value: Berkeley DB; Project Voldermort; MemcacheDB; SimpleBD

Orientadas a documentos: MongoDB, CouchDB, IBM Lotus Domino, Riak, RavenDB

Família de colunas (bigTable): BigTable, Hbase, Cassandra (Facebook)

Orientadas a grafos: Neo4J, InfoGrid, HyperGraphDB;

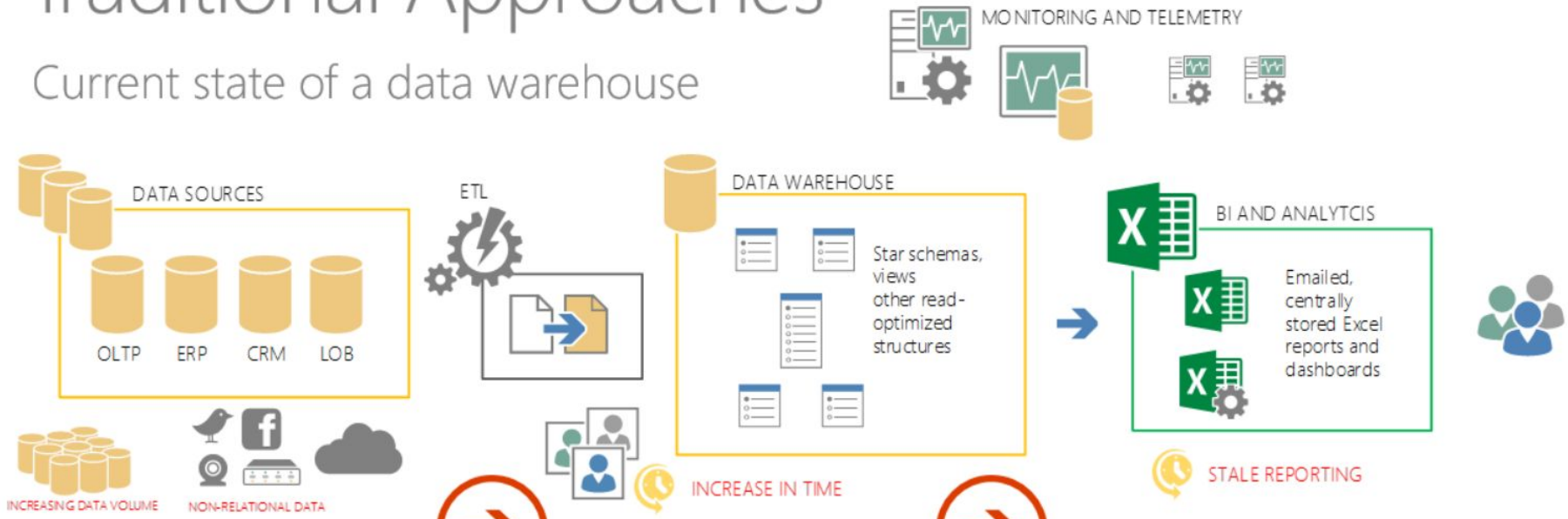
Orientadas a colunas*: MonetDB, Vertica , Infobright, LucidDB

Bancos de Dados Relacionais	Bancos de Dados NoSQL
Transações **ACID**: **Atomicidade**: a transação é executada totalmente ou não será executada; **Consistência**: o banco passará de uma forma consistente para outra forma consistente; **Isolamento**: a transação não será interferida por outra transação concorrente; **Durabilidade**: o que foi salvo não será perdido.	Transações **BASE**: **BA-Basicamente Disponível**, prioridade na disponibilidade dos dados (tolera falhas parciais); **S-Estado Leve**: o sistema não precisa ser consistente o tempo todo, isto é, a persistência não é necessariamente feita em tempo real; **E – Eventualmente consistente**: o sistema é consistente em momentos determinados.
Relacionado fisicamente (conceito ER).	Não relacionado fisicamente.
Dificuldade em armazenar e recuperar grandes volumes de dados não estruturados (e outros tipos de dados	Construído para tratar grandes volumes de dados não estruturados, escaláveis horizontalmente. heterogéneos).

Bancos de Dados Relacionais	Bancos de Dados NoSQL
Esquema definido (tabelas, chaves, tipos de dados, relacionamentos), estrutura rígida, sem redundância de dados, normalizada.	Não tem esquema definido (permite adicionar campos aos registros do banco livremente sem a necessidade de mudança na estrutura).
Escalabilidade vertical: mais CPU, mais memória, os dados não podem estar distribuídos.	Escalabilidade horizontal: servidores em paralelo, dados distribuídos em nodos diferentes.
Não existe Sharding: uma FK não pode apontar para uma tabela que está em outro nodo.	Sharding: divide os dados horizontalmente, partes diferentes dos dados em servidores diferentes.
Não existem algoritmos do tipo Map-reduce.	Map-reduce: algoritmo de gerenciamento em larga escala. Organiza o processamento, aproveitando múltiplas máquinas de um cluster.
Usados em sistemas locais, corporativos; segurança da informação; consistência dos dados.	Usados em sistemas de alta escalabilidade e performance na consulta/escrita (redes sociais).

Traditional Approaches

Current state of a data warehouse



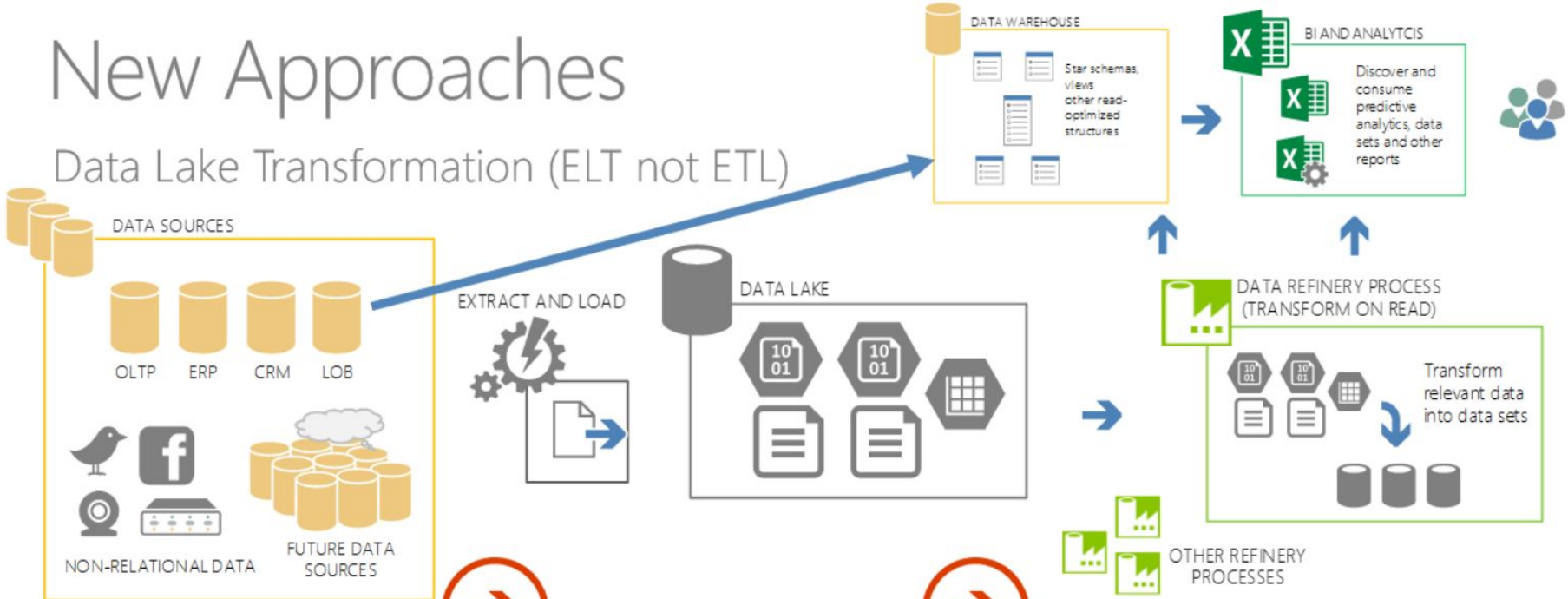
- Increase in variety of data sources
- Increase in data volume
- Increase in types of data
- Pressure on the ingestion engine

- Complex, rigid transformations can't longer keep pace
- Monitoring is abandoned
- Delay in data, inability to transform volumes, or react to new sources
- Repair, adjust and redesign ETL

- Reports become invalid or unusable
- Delay in preserved reports increases
- Users begin to "innovate" to relieve starvation

New Approaches

Data Lake Transformation (ELT not ETL)



All data sources are considered

Leverages the power of on-prem technologies and the cloud for storage and capture

Native formats, streaming data, big data

Extract and load, no/minimal transform

Storage of data in near-native format

Orchestration becomes possible

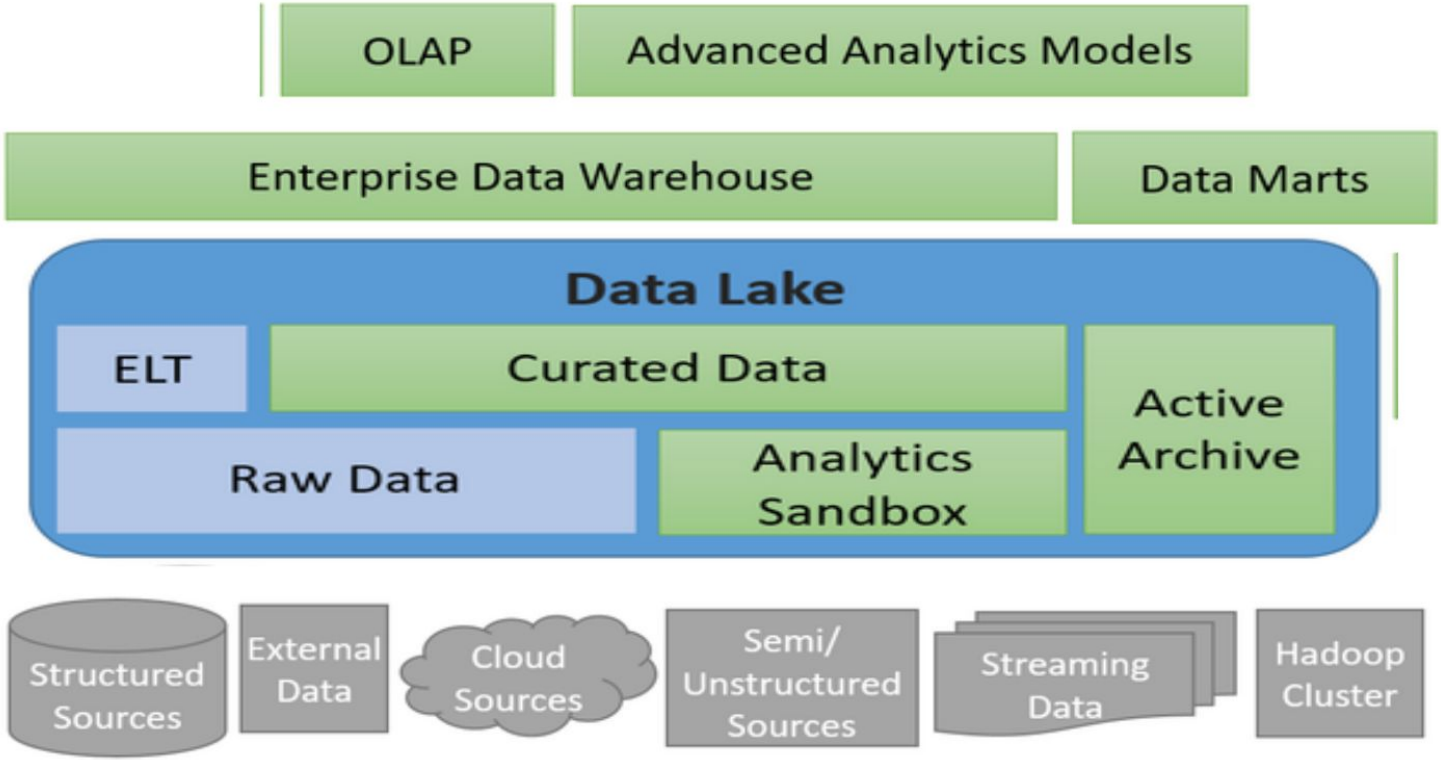
Streaming data accommodation becomes possible

Refineries transform data on read

Produce curated data sets to integrate with traditional warehouses

Users discover published data sets/services using familiar tools

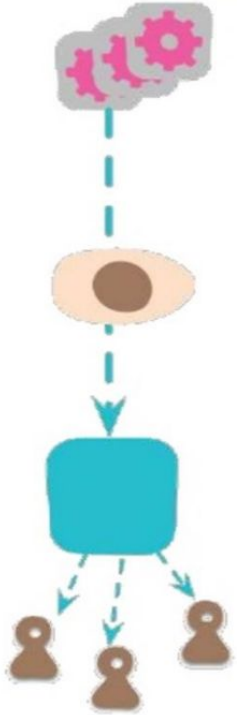
Data Warehouse e Data Lake



Data Warehouse e Data Lake

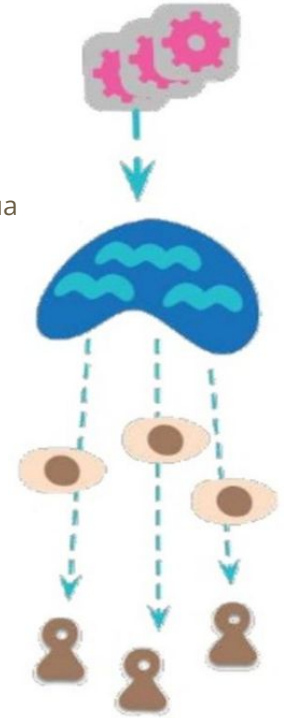
Com um **data warehouse**, os dados recebidos são limpos e organizados em um único esquema consistente antes de serem colocados no warehouse...

...a análise é feita diretamente nos dados curados da warehouse.



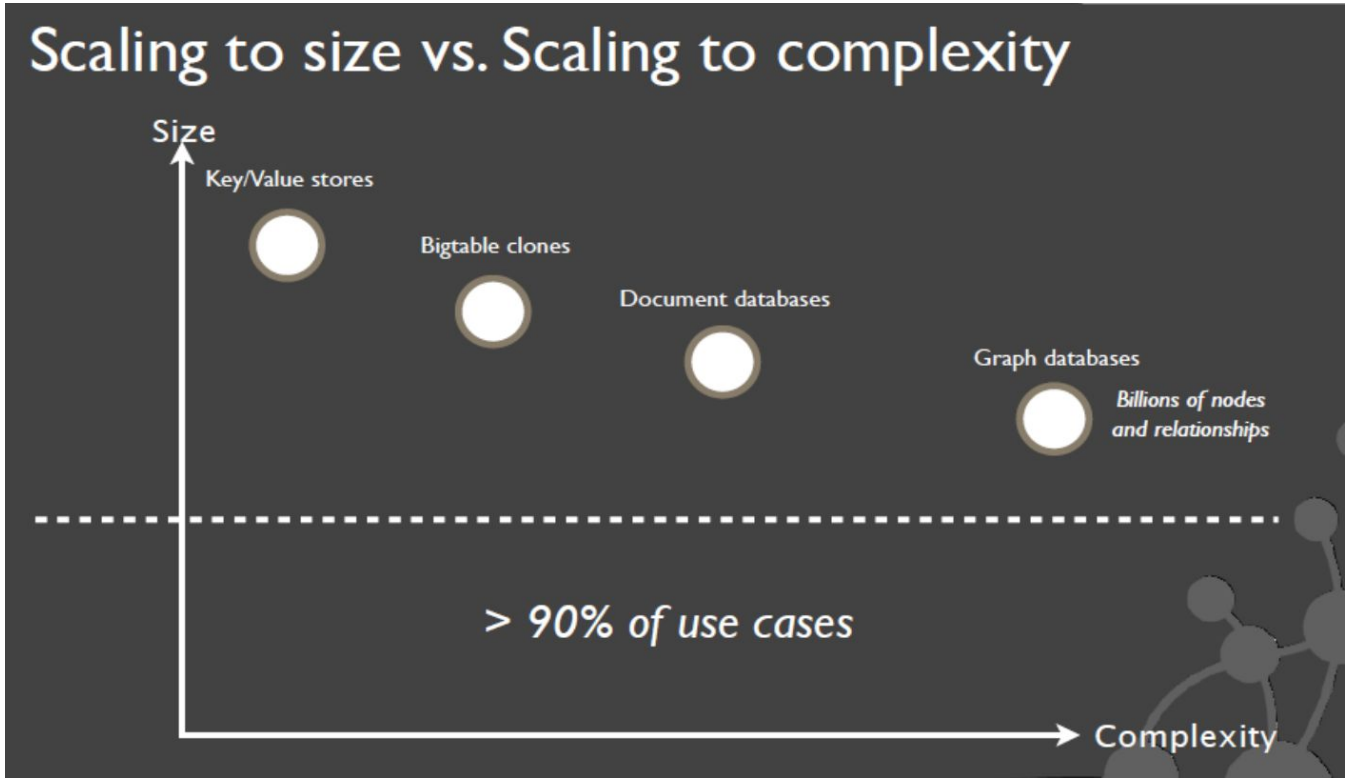
Como um data lake, os dados recebidos vão para o lake em sua forma de linha...

...selecionamos e organizamos dados para cada necessidade.



Fonte: Martin Fowler

Armazenamento e de Processamento de Dados



Ferramentas e Tendências em Armazenamento Colunar

Apache Drill é um mecanismo de consulta SQL distribuído, que acessa dados estruturados e não estruturados, bases em diversos formatos, sem esquemas, relacionais e não relacionais

Apache Parquet é um formato de armazenamento em coluna;

Apache Arrow é um formato de armazenamento em coluna em memória;

Ferramentas e Tendências em Armazenamento Colunar

Apache Calcite é um framework de gerenciamento de dados que contém uma API para analisar, planejar e otimizar consultas SQL;

Dremio é uma abordagem para análise de dados, é uma solução que “elimina” a necessidade de ETLs, tabelas de agregação ou cubos de dados, e permite a consulta a todas suas fontes de dados.

Apache Drill

Schema-free SQL Query Engine for Hadoop, NoSQL and Cloud Storage

DOWNLOAD NOW

Overview of Apache Drill Query Execution

Learning Apache Drill

News: Announcing Drill 1.21! (Charles Givre) | Drill 1.20.3 Released (James Turton)

Agility

Get faster insights without the overhead (data loading, schema creation and maintenance, transformations, etc.)

Flexibility

Analyze the multi-structured and nested data in non-relational datastores directly without transforming or restricting the data

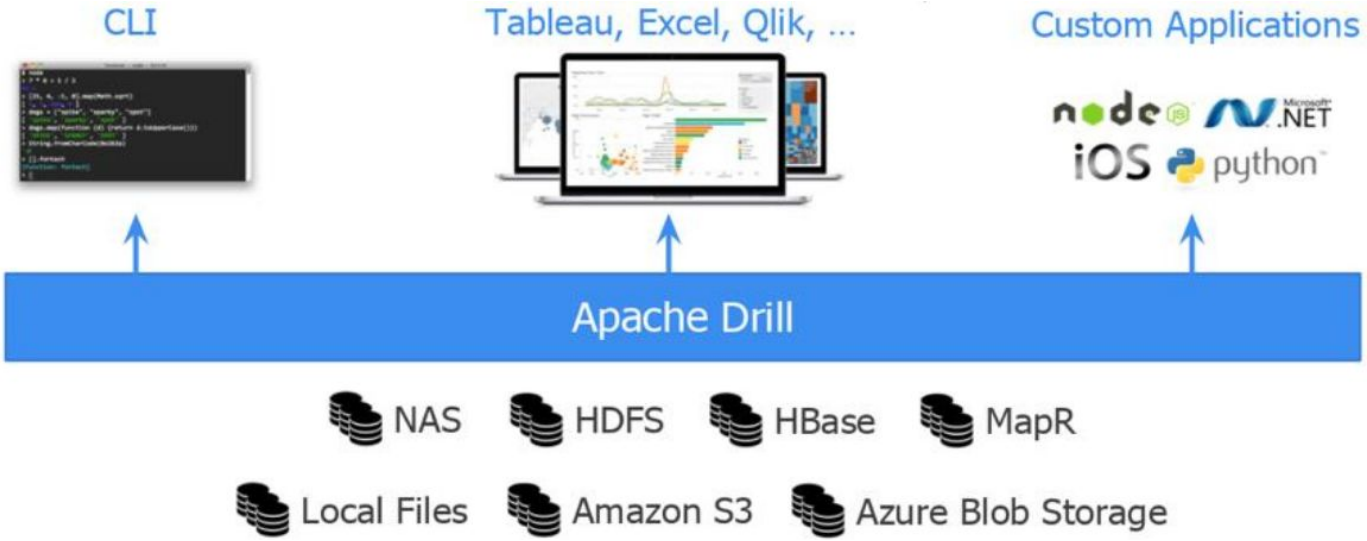
Familiarity

Leverage your existing SQL skillsets and BI tools including Tableau, Qlikview, MicroStrategy, Spotfire, Excel and more

https://drill.apache.org/

Apache Drill

Mecanismo de consulta SQL distribuído que acessa dados estruturados e não estruturados, bases em diversos formatos, sem esquemas, esquemas relacionais e não relacionais



Apache Drill

Executar: **sqlline.bat -u “jdbc:drill:zk=local”**

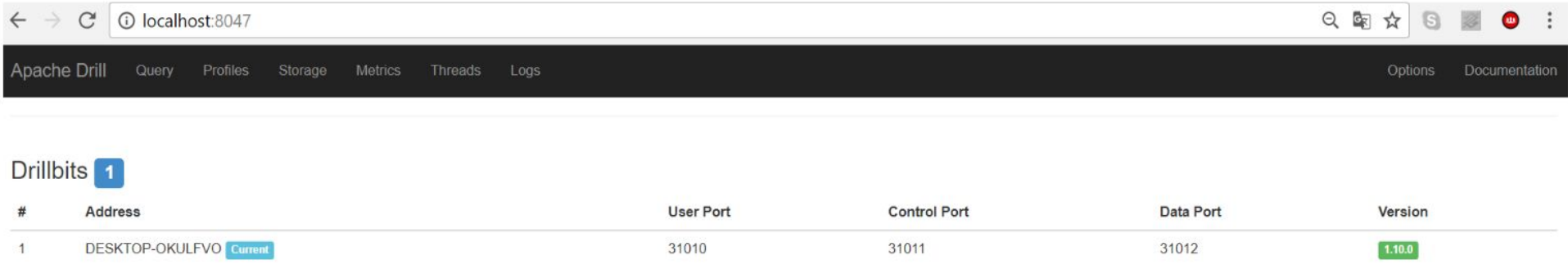
Aparecerá no prompt> **0: jdbc: drill: zk = local**, onde

0 é o número de conexões do Apache Drill;

JDBC é o tipo de conexão;

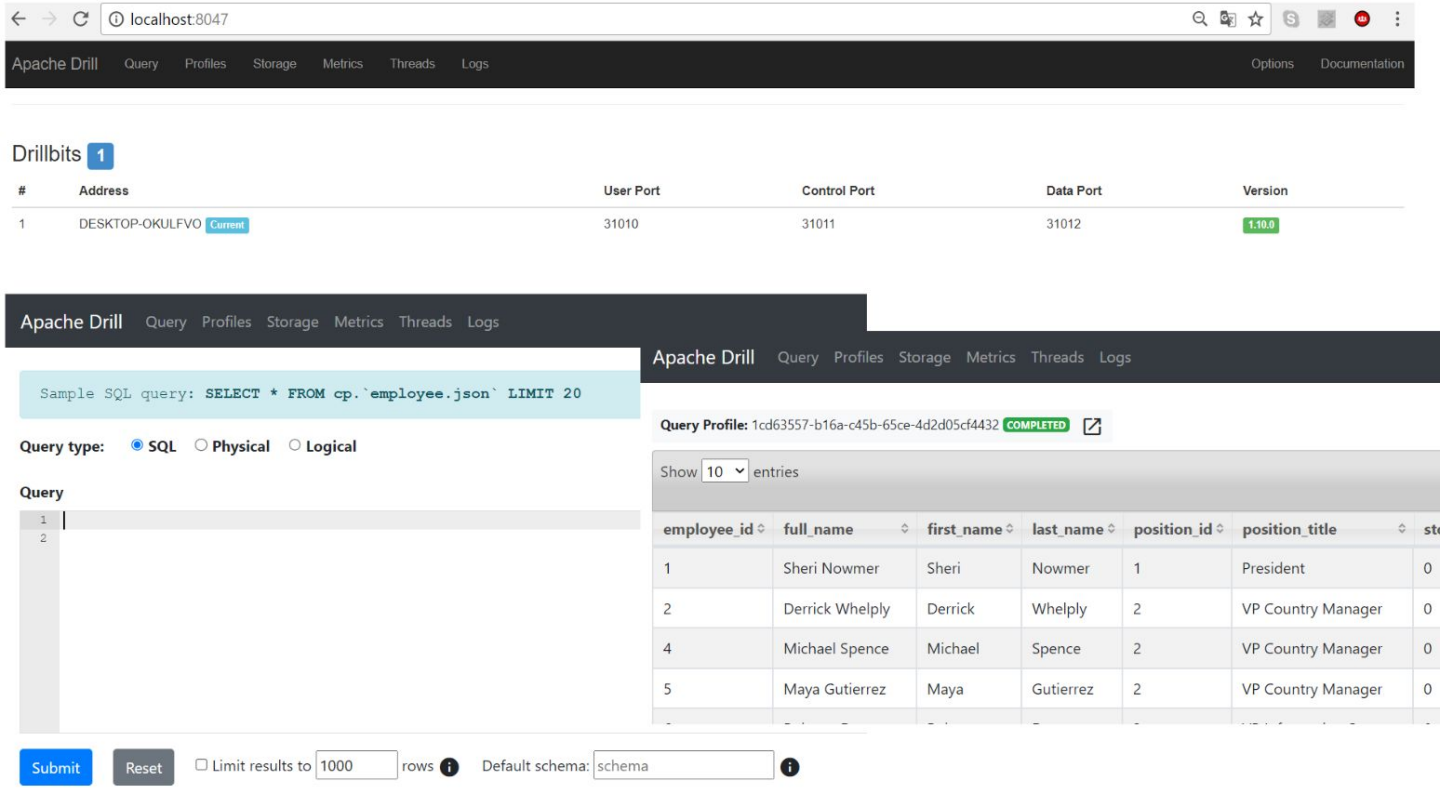
Zk =local, significa que o nó local substitui o nó do ZooKeeper (neste caso não precisa de instalação). O ZooKeeper é um API que permite construir sistemas distribuídos.

No navegador chamar: **localhost:8047**



Apache Drill

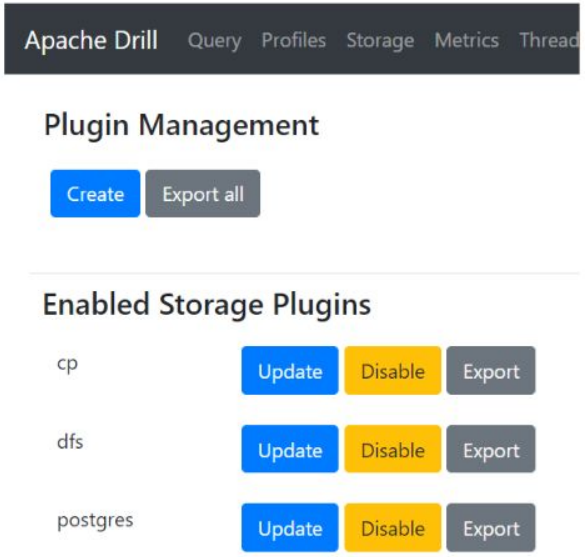
Executar: drill-embedded.bat e no browser chamar: localhost:8047

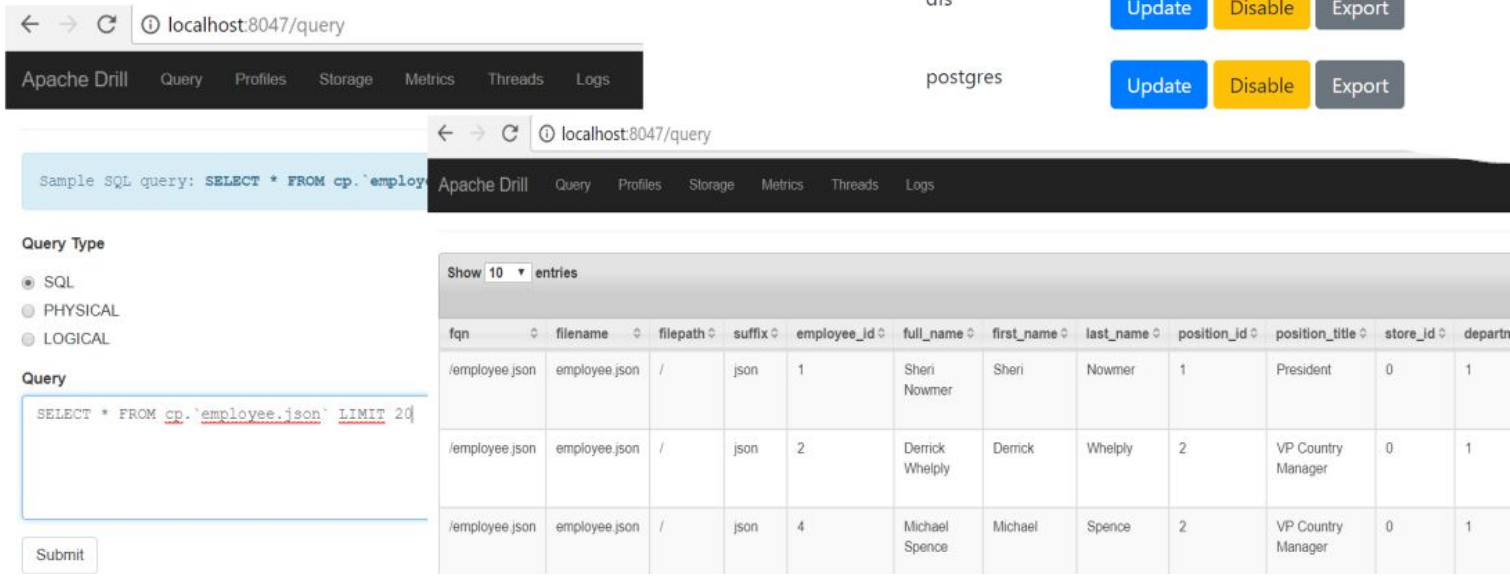


Apache Drill

cp, “aponta” para os arquivos JAR, onde estão embutidos alguns arquivos de exemplos;

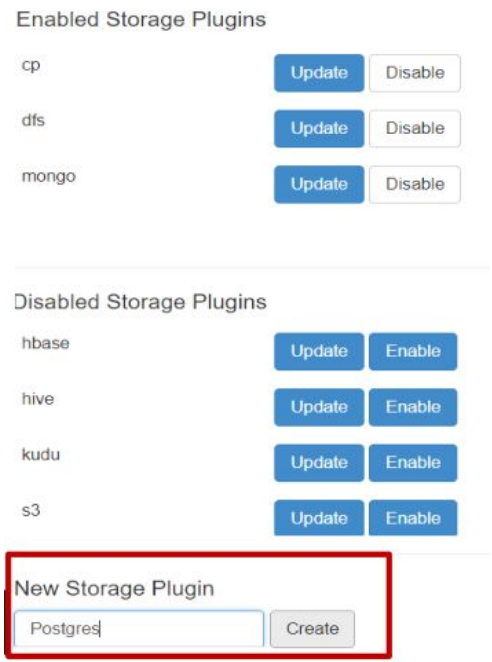
dfs, aponta para o sistema de arquivos local; podemos criar outras conexões através de driver JDBC;





Apache Drill

Criação de uma conexão com Postgres e com MonetDB



Configuration

```
{
  "type": "jdbc",
  "driver": "nl.cwi.monetdb.jdbc.MonetDriver",
  "url": "jdbc:monetdb://localhost:50000/demo",
  "username": "monetdb",
  "password": "monetdb",
  "enabled": true
}
```

Configuration

```
{
  "type": "jdbc",
  "driver": "org.postgresql.Driver",
  "url": "jdbc:postgresql://localhost:5433/cursopos",
  "username": "postgres",
  "password": "aran",
  "enabled": true
}
```

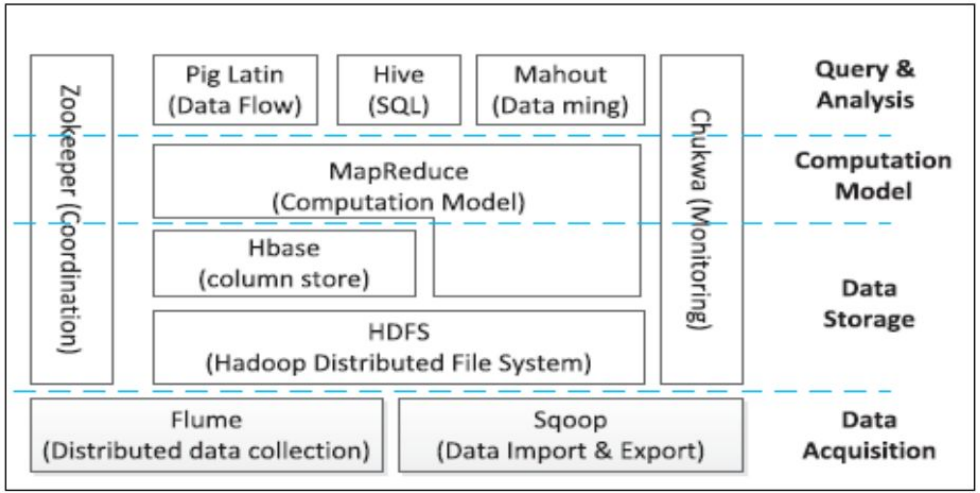
Hadoop e Spark

As tecnologias envolvidas no processamento do grande conjuntos de dados:

Hadoop e Spark (são as duas principais).

Hadoop é uma plataforma distribuída voltada para clusters e processamento de grandes massas de dados. Foi inspirado pelo MapReduce e GFS;

Spark é uma plataforma que trabalha em memória (e por isso muito mais rápida no processamento que o Hadoop) e indicada para dados em stream;

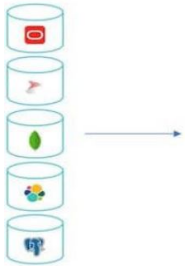


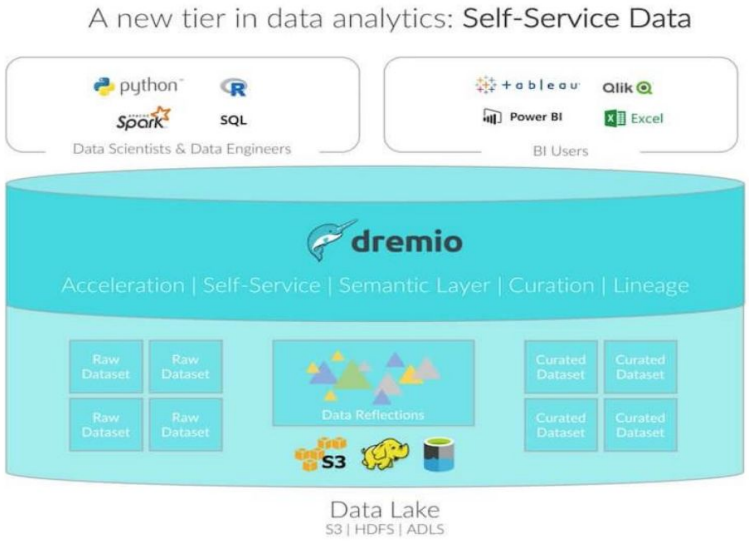
Dremio

O Dremio utiliza armazenamento e execução colunar alimentado pelo **Apache Arrow** (colunar na memória), e o **Apache Parquet** (colunar no disco) utiliza também o **Apache Calcite** para análise de SQL e otimização de consulta.

Formato de armazenamento em coluna em memória.

“Dremio combina uma arquitetura de **execução e aceleração colunar** para alcançar desempenho interativo em **qualquer volume de dados**, permitindo que **cientistas de dados** e **analistas de negócios** configurem os dados de acordo com as suas necessidades de análise”.

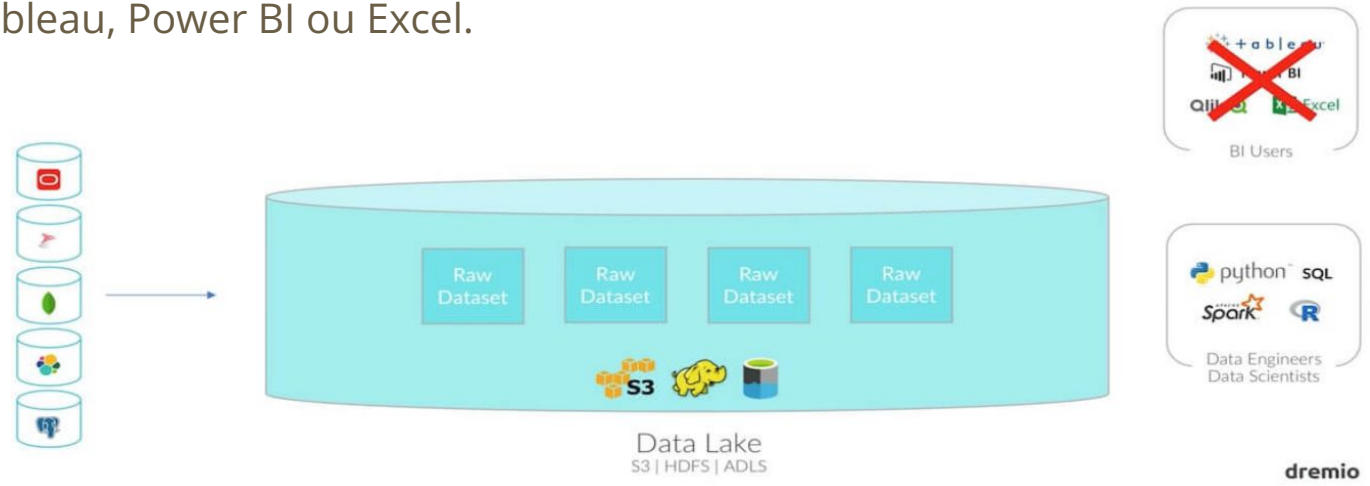




dremio

O Data Warehouse tem um custo alto de implementação: necessidade de levantar requisitos, entender os dados que atendem aos requisitos, extrair, transformar e carregar os dados, o que requer tempo e esforço.

O Data Lake é mais flexível, permite dados em sua forma bruta, em um local mais econômico e escalável. No entanto, a falta de rigor na estrutura dos dados dificulta o acesso a esses dados, em especial aos usuários do BI, que não podem usar ferramentas-padrão como Tableau, Power BI ou Excel.

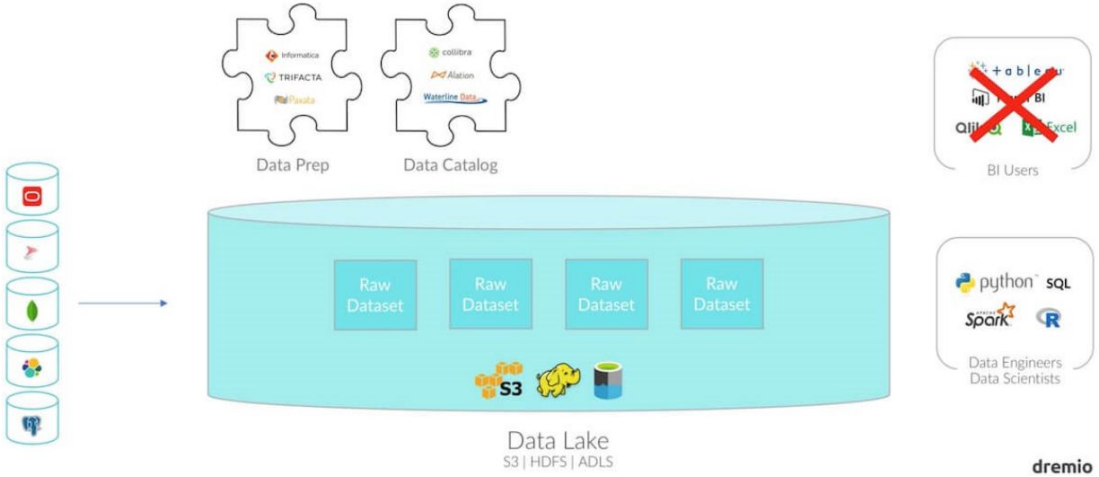


dremio

Precisamos de uma maneira de transformar os dados de sua forma bruta em algo de maior valor, mais limpo, mais normalizado e com a integridade necessária para suportar o BI. As ferramentas de ETL fazem isso há muitos anos.

À medida que começamos a reformular e refinar os dados, precisamos fazer um catálogo (inventário) para entender e administrar os dados do DL.

O Data Lake se torna um complexo quebra-cabeças

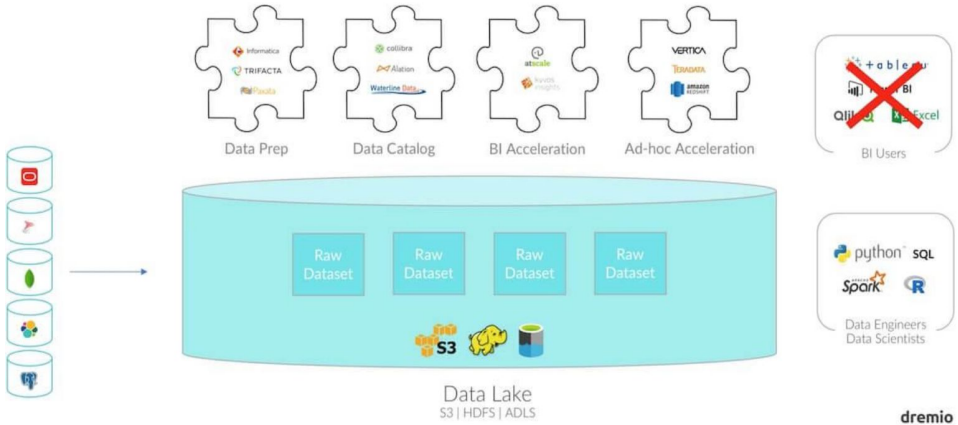


dremio

Um dos principais problemas do DL é a velocidade. E sem essa velocidade, é difícil para os usuários de BI aproveitarem os dados no DL. Assim, as empresas começam a olhar para as tecnologias de aceleração de BI, tecnologias de cubos de dados que possibilitam o acesso aos dados.

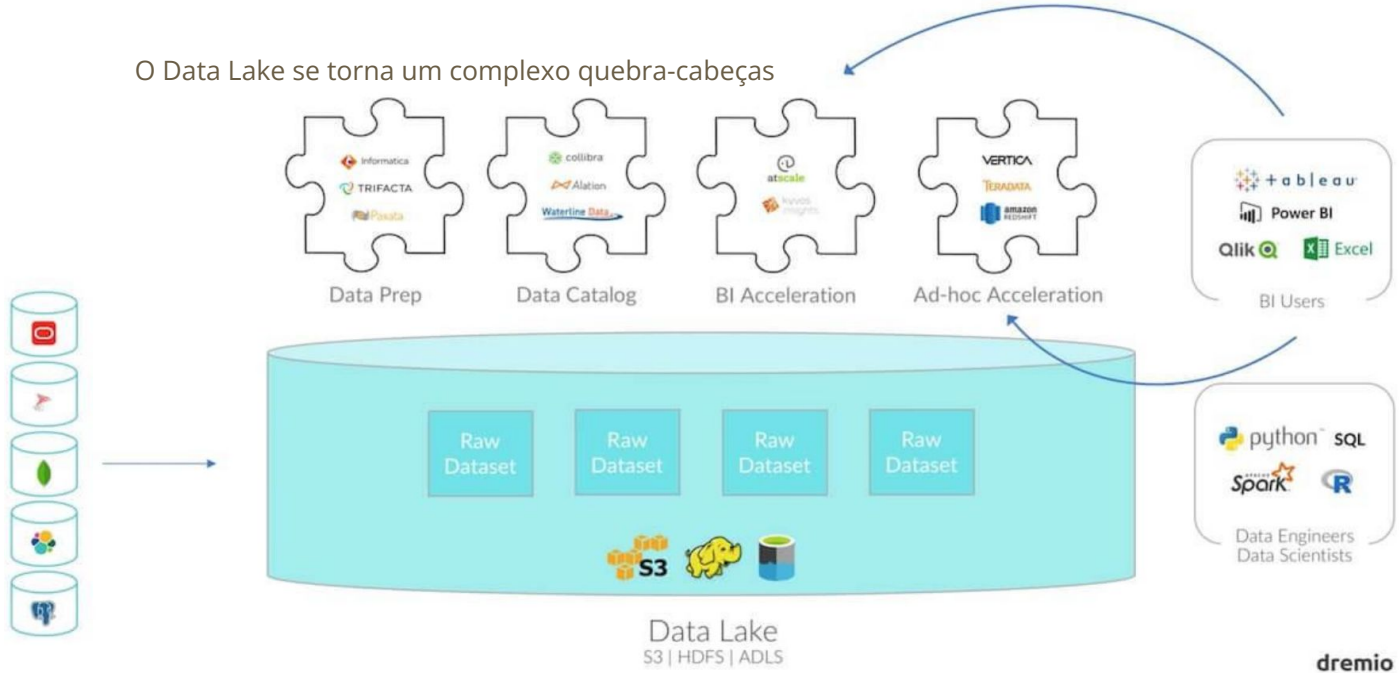
Mas ainda falta resolver as consultas ad hoc, e para isso as empresas tendem a utilizar um BD “tradicional”.

O Data Lake se torna um complexo quebra-cabeças



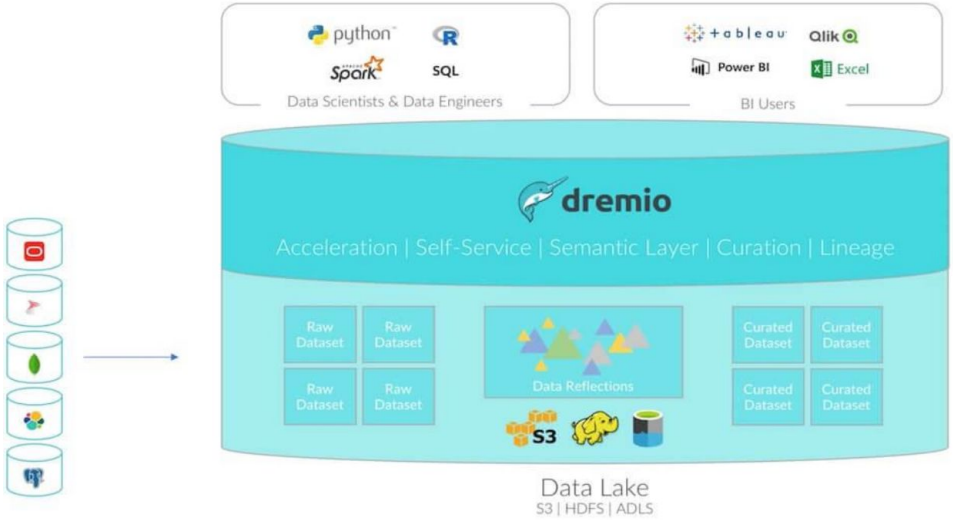
dremio

Quando as peças desse quebra-cabeça são montadas, os usuários de BI podem começar a executar o seu com os dados do **data lake**.



Em um alto nível, é isso que o Dremio é: um novo nível de análise de dados chamado de dados de autoatendimento, que fica entre as ferramentas dos cientistas de dados, engenheiros de dados e usuários de BI, e entre a infraestrutura e os dados do **data lake**. É executado diretamente nessa infraestrutura, fornecendo recursos de aceleração, de autoatendimento e uma camada semântica para os usuários finais descreverem os dados em seus próprios termos.

Uma nova camada na análise de dados: **Self-Service Data**

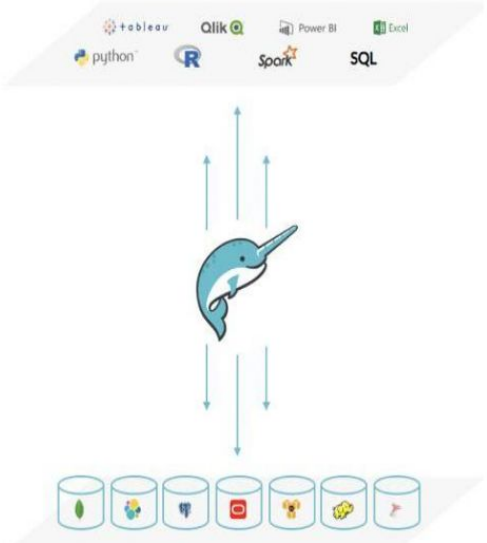


dremio

Cubes &
Aggregation Tables

Data Warehouse

Data Staging



Data Warehouse e Data lake

Aran Bey Tcholakian Morales
Richard Henrique de Souza
Saulo Popov Zambiasi