Do Data Warehouse ao Data Lake

Os **data warehouses** introduziram a necessidade de um modelo de dados abrangente para integrar diferentes áreas temáticas dentro de uma empresa. Para atingir esse objetivo, foi amplamente adotada a técnica de modelagem dimensional. Esse modelo facilita a análise de processos específicos de negócios, como vendas, oferecendo uma estrutura organizada e eficiente. No entanto, apesar de sua eficácia inicial, os data warehouses enfrentaram sérias **limitações**, especialmente com o avanço do big data e os desafios relacionados aos **quatro Vs**:

- **Volume**: As arquiteturas de data warehouse tradicionais não foram projetadas para lidar com o crescimento exponencial dos dados. O armazenamento de petabytes de informações tornou-se caro e difícil de escalar, já que essas soluções não utilizam tecnologias modernas, como processamento paralelo e em memória. Isso resultou em dificuldades para acompanhar a escalabilidade exigida.
- **Velocidade**: Os data warehouses não oferecem suporte adequado para processar dados em tempo real, sendo incapazes de lidar com a velocidade de fluxos de dados contínuos (streaming). A execução de processos de ETL em janelas reduzidas acaba comprometendo a infraestrutura com o aumento da carga.
- **Variedade**: Enquanto são excelentes para dados estruturados, os data warehouses falham ao armazenar e consultar dados semiestruturados e não estruturados, como logs, imagens ou dados de sensores, que são cada vez mais comuns em ambientes de big data.
- **Veracidade**: A rastreabilidade e a confiabilidade dos dados são pontos fracos nos data warehouses. Metadados são limitados ao esquema e pouco focados na qualidade e na linhagem dos dados, o que compromete análises baseadas em dados confiáveis.

Além disso, os **formatos proprietários** dessas arquiteturas restringem a integração com ferramentas modernas, como as de **ciência de dados** e **aprendizado de máquina**, que demandam maior flexibilidade e escalabilidade. Esses fatores tornam os data warehouses não apenas caros de construir e manter, mas também inadequados para responder às rápidas mudanças do cenário atual de negócios e tecnologia. Diante dessas limitações, surgiu o conceito de **data lake**.

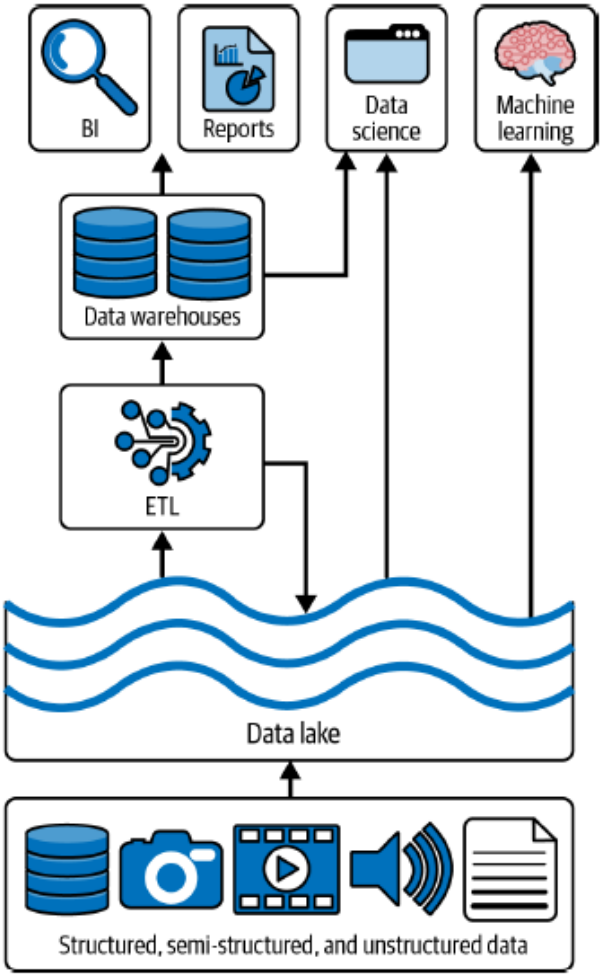
Essa nova abordagem foi projetada para lidar com os desafios do big data, oferecendo uma solução moderna e flexível. Diferentemente dos data warehouses, os data lakes permitem armazenar todos os tipos de dados estruturados, semiestruturados e não estruturados em um único repositório centralizado, com custos mais baixos e escalabilidade quase ilimitada. Essa arquitetura também permite maior liberdade para integrar ferramentas avançadas de análise e processamento, atendendo às demandas de empresas modernas que buscam se adaptar ao ritmo acelerado de crescimento e inovação no uso de dados.

Data Lake

Entre as arquiteturas mais populares que emergiram na era do big data, destaca-se o **data lake**. Em vez de impor limitações rígidas à estrutura dos dados, a proposta do data lake é simples: **consolidar todos os dados estruturados, semiestruturados e não estruturados em qualquer escala em um único repositório centralizado**.

O termo “data lake” vem da analogia com um rio ou lago real, que retém a água, ou neste caso, os dados, com vários afluentes que fluem para o lago em tempo real. Essa

abordagem prometeu transformar o acesso e a utilização de dados, funcionando como uma força democratizadora ao permitir que empresas explorem, de maneira flexível, uma fonte inesgotável de informações para tomadas de decisão e inovação.



O conceito de data lake teve início com o HDFS (Hadoop Distributed File System). Com o avanço e a popularização da computação em nuvem, os data lakes evoluíram para soluções baseadas em armazenamento de objetos na nuvem, caracterizadas por custos extremamente baixos e uma capacidade de expansão praticamente ilimitada. Diferentemente de um **data warehouse** monolítico, onde o armazenamento e o processamento estão fortemente acoplados, o data lake permite centralizar grandes volumes de dados de qualquer tamanho ou tipo, sem restrições rígidas de estrutura.

Quando há necessidade de consultar ou transformar esses dados, é possível aproveitar o poder computacional escalável da nuvem, criando clusters sob demanda e selecionando a ferramenta de processamento mais adequada para cada tarefa. Tecnologias como **MapReduce**, **Spark**, **Ray**, **Presto** e **Hive**, entre outras, tornam essa abordagem flexível e eficiente, atendendo a uma ampla gama de casos de uso e necessidades analíticas.

Componentes dos Data Lakes

1. **Armazenamento**: Data lakes utilizam sistemas de armazenamento escaláveis e duráveis, geralmente baseados na nuvem. Esses sistemas permitem a ingestão de grandes volumes de dados em lote ou em fluxo contínuo (ex.: IoT, streaming). A separação entre armazenamento e computação possibilita escalabilidade independente e ajustes sob demanda, tornando a solução mais eficiente.

2. **Computação**: O processamento de dados é realizado por mecanismos como o **Apache Spark**, que utilizam clusters de computação para lidar com tarefas de análise e transformação.
3. **Formatos de Dados**: Os data lakes adotam formatos abertos e otimizados para diferentes casos de uso, como **Parquet** (eficiência analítica), **Avro** (serialização), **JSON** e **CSV**, garantindo compatibilidade e flexibilidade.
4. **Metadados**: Metadados fornecem informações contextuais sobre os dados, como timestamps (registro de uso), esquemas (estrutura dos dados) e tags (proprietário e classificação). Esses dados adicionais tornam a gestão e a análise mais eficientes.

Apesar de toda a promessa e o hype, o conceito de data lake enfrentou **sérias limitações**:

- Embora projetados para análises OLAP, data lakes armazenam dados brutos que frequentemente são desorganizados e difíceis de consultar diretamente. Isso exige ferramentas adicionais de processamento e análise, além de etapas de transformação e carga para data warehouses, prolongando o tempo necessário para gerar valor. Esse modelo híbrido de **data lake + warehouse**, amplamente adotado, tem perdido espaço com o surgimento dos **lakehouses**.
- Apesar do armazenamento em nuvem ser barato, construir e manter um data lake eficiente requer habilidades especializadas, elevando os custos de pessoal ou consultoria.
- Operações básicas de SQL, como atualizar ou excluir dados, são muito limitadas em data lakes e, geralmente, exigem recriar tabelas inteiras, o que torna o processo lento e ineficiente. Além disso, como não há garantias transacionais, atualizações simples podem exigir reescritas completas dos dados já armazenados, aumentando custos e esforço,
- Com a abordagem de "esquema na leitura", os data lakes permitem ingerir dados sem um esquema predefinido, mas essa flexibilidade pode resultar em **problemas de qualidade**, transformando o repositório em um verdadeiro **"pântano de dados".**

Diante dessas limitações dos data lakes de primeira geração, diversas empresas buscaram maneiras de superar esses desafios e liberar todo o potencial dessa abordagem. Um exemplo marcante foi a introdução do conceito de **data lakehouse** pela **Databricks**, combinando o melhor dos mundos dos data lakes e data warehouses em uma solução mais eficiente e integrada.

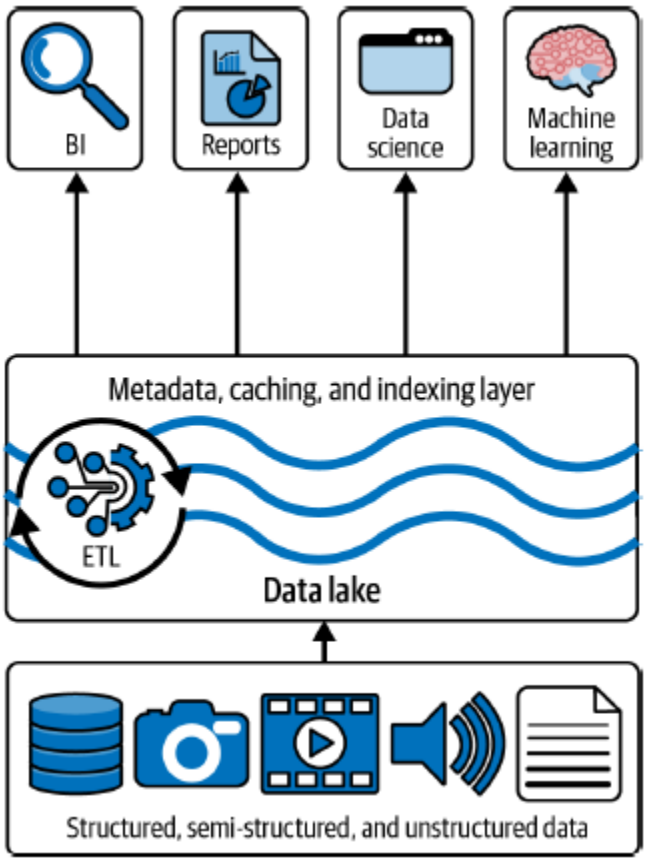
Data Lakehouse

O **lakehouse incorpora os controles, a gestão de dados e as estruturas de dados** encontradas em um data warehouse, enquanto ainda armazena dados em armazenamento de objetos e suporta uma variedade de mecanismos de consulta e transformação. Em particular, o data lakehouse suporta transações com **atomicidade, consistência, isolamento e durabilidade (ACID)**, uma grande mudança em relação ao data lake original, onde os dados eram apenas despejados, sem a possibilidade de atualizá-los ou excluí-los. O termo "data lakehouse" sugere uma convergência entre data lakes e data warehouses.

- **Atomicidade:** Garante que uma transação seja tratada como uma única unidade indivisível. Isso significa que ou toda a transação é concluída com sucesso, ou nenhuma de suas partes é aplicada.

- **Consistência**: Uma transação cria um novo estado válido dos dados ou retorna todos os dados ao seu estado anterior em caso de falha.
- **Isolamento**: Garante que as transações simultâneas sejam executadas como se fossem independentes, sem interferir umas nas outras.
- **Durabilidade**: Garante que, uma vez confirmada, uma transação será permanentemente registrada no banco de dados, mesmo em caso de falhas no sistema ou desligamento.

Assim como os data lakes, a arquitetura de lakehouse utiliza sistemas de armazenamento em nuvem de baixo custo, com a flexibilidade e escalabilidade horizontal inerentes a esses sistemas. O objetivo de um lakehouse é utilizar formatos de dados de alto desempenho já existentes, como o Parquet, enquanto também permite transações ACID e outros recursos.



A Camada de Metadados, Cache e Indexação transforma o Data Lake em um Lakehouse, adicionando governança, transações ACID, e otimizações que tornam o armazenamento pronto para análises avançadas, sem a necessidade de manter cópias redundantes de dados no data lake e em um data Warehouse.

Para que um lakehouse ofereça governança de dados, transações ACID, e outros controles avançados, é necessário utilizar sistemas ou frameworks adicionais que são responsáveis por implementar essas funcionalidades. Frameworks como Delta Lake, Apache Iceberg ou Apache Hudi são adicionados como uma camada de gerenciamento sobre o data lake.

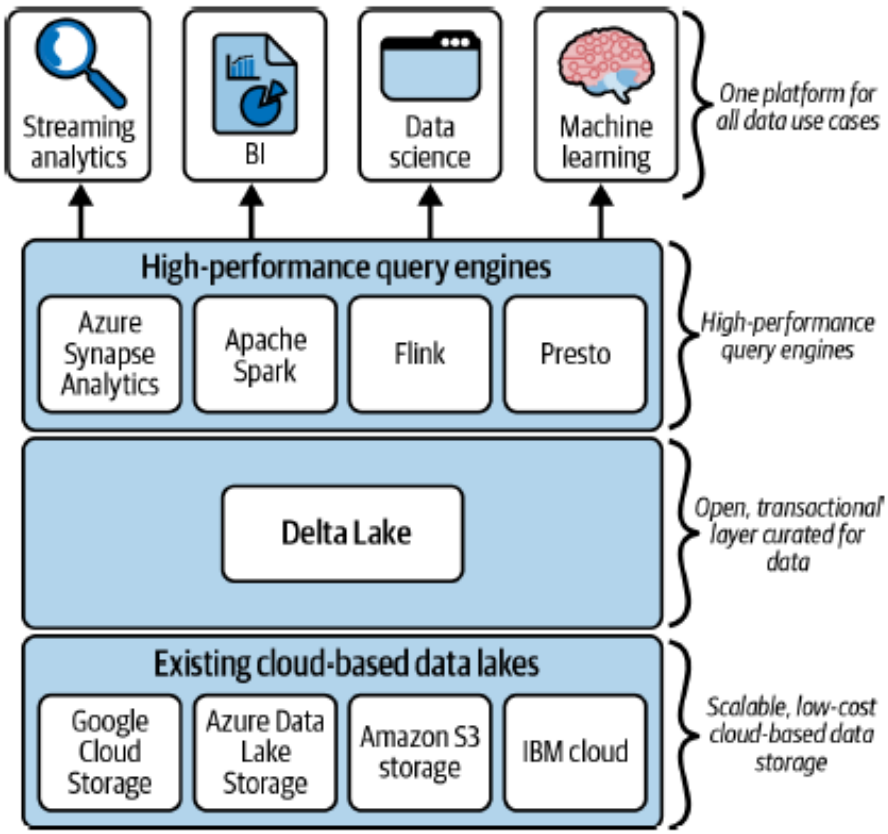
As três tecnologias (Delta Lake, Apache Iceberg e Apache Hudi) são as principais "camadas de tabela" (table formats) que permitem essa transformação. Cada uma tem suas particularidades:

- **Delta Lake:**
  - Desenvolvido pela Databricks, muito integrado com Spark
  - Fornece suporte completo para transações ACID, versionamento e governança de dados.
- **Apache Iceberg:**
  - Criado pela Netflix, projetado para análise em larga escala, é altamente eficiente para consultas analíticas distribuídas.
  - Oferece transações ACID e controle de esquemas.
- **Apache Hudi:**
  - Criado pelo Uber, focado em operações transacionais incrementais (atualizações e deleções).
  - Ideal para fluxos de dados contínuos e dados de mudança (CDC).

Esses frameworks atuam como o "motor" que transforma um data lake simples em um lakehouse robusto e funcional. Sem eles, a arquitetura lakehouse perde muitos de seus benefícios e se aproxima mais de um data lake tradicional. Esses frameworks são chamados de "camadas de tabela" porque organizam e tratam os dados armazenados no data lake bruto (em arquivos como Parquet ou JSON) como se fossem tabelas estruturadas, oferecendo funcionalidades que antes eram exclusivas de bancos de dados ou data warehouses.

Delta Lake

O **Delta Lake** é um formato de tabela aberto que se sobrepõe ao armazenamento de data lakes existentes (como Amazon S3, ADLS ou GCS). Ele combina a flexibilidade e escalabilidade dos **data lakes** com funcionalidades transacionais e analíticas avançadas típicas de um **data warehouse**. Essa fusão possibilita a implementação da arquitetura **lakehouse** com mais eficiência, confiabilidade e capacidade de análise.

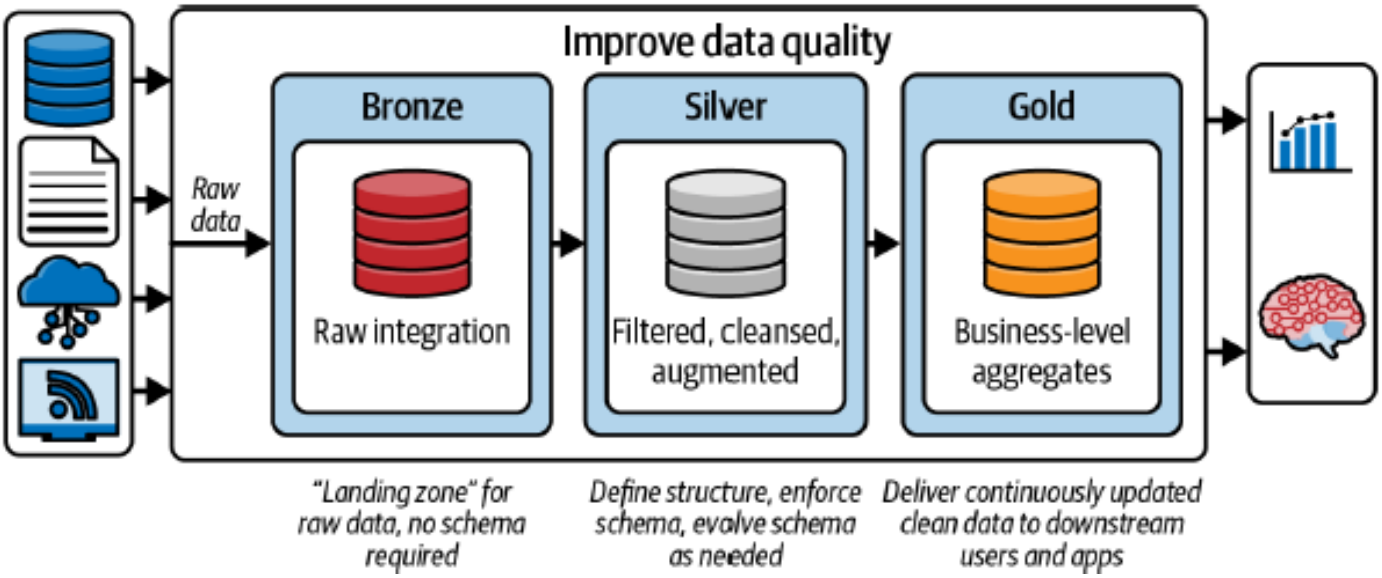


Uma arquitetura lakehouse é estruturada em três camadas principais:

- **Camada de Armazenamento:** A camada base do lakehouse utiliza tecnologias de armazenamento em nuvem padrão, como Azure Data Lake Storage (ADLS), Google Cloud Storage (GCS) ou Amazon S3, para fornecer uma infraestrutura escalável e econômica. Esses serviços permitem o armazenamento de grandes volumes de dados em um formato bruto ou semiestruturado, garantindo alta disponibilidade e custo reduzido.
- **Camada Transacional:** A camada intermediária é alimentada por tecnologias como o Delta Lake, que introduz garantias ACID ao lakehouse. Isso possibilita transformar dados brutos em tabelas prontas para análises avançadas. Além disso, o Delta Lake oferece: Suporte para operações DML (como UPDATE, DELETE, MERGE). Processamento eficiente e escalável de metadados. Integração com fluxos de dados em streaming. Um log de auditoria detalhado, que permite rastrear todas as alterações nos dados, além de oferecer versionamento e recursos como "time travel".
- **Camada de Consulta de Alto Desempenho:** Na camada superior, encontram-se os mecanismos de consulta e processamento, responsáveis por análises rápidas e eficientes. Esses mecanismos aproveitam os recursos de computação em nuvem subjacentes e são compatíveis com diversas ferramentas, incluindo:
  - Apache Spark: Para processamento distribuído e análise em larga escala.
  - Apache Hive: Para consultas SQL em grandes volumes de dados.
  - Presto e Trino: Para execução de consultas SQL interativas e análises rápidas.

Arquitetura Medallion

O padrão arquitetural conhecido como **Medallion Architecture**, com as camadas **Bronze**, **Silver** e **Gold**, é amplamente utilizado em lakehouses para organizar e estruturar os dados. Embora seja apenas uma das várias abordagens possíveis, essa arquitetura é ideal para soluções como data warehouses modernos, data marts e outras iniciativas de análise de dados.



A Medallion Architecture organiza os dados em camadas lógicas que refletem diferentes estágios de processamento e qualidade dos dados. Essa abordagem incremental facilita a limpeza, organização e preparação dos dados para consumo final. Cada camada tem um propósito específico:

- **Camada Bronze**: É a zona de aterrissagem para os dados ingeridos dos sistemas de origem. Os dados são armazenados "como estão" (dados brutos), sem processamento inicial, mas podem ser enriquecidos com metadados, como data e hora de carregamento ou identificadores de processamento. O formato da fonte de dados é mantido: arquivos CSV são armazenados como CSV, dados JSON são armazenados como JSON, e dados extraídos de tabelas de banco de dados geralmente chegam à camada Bronze como arquivos Parquet ou AVRO. O objetivo do processo de ingestão é armazenar rapidamente e de forma simples os dados da fonte na camada Bronze, com auditoria e metadados mínimos para permitir rastreabilidade e reprocessamento.

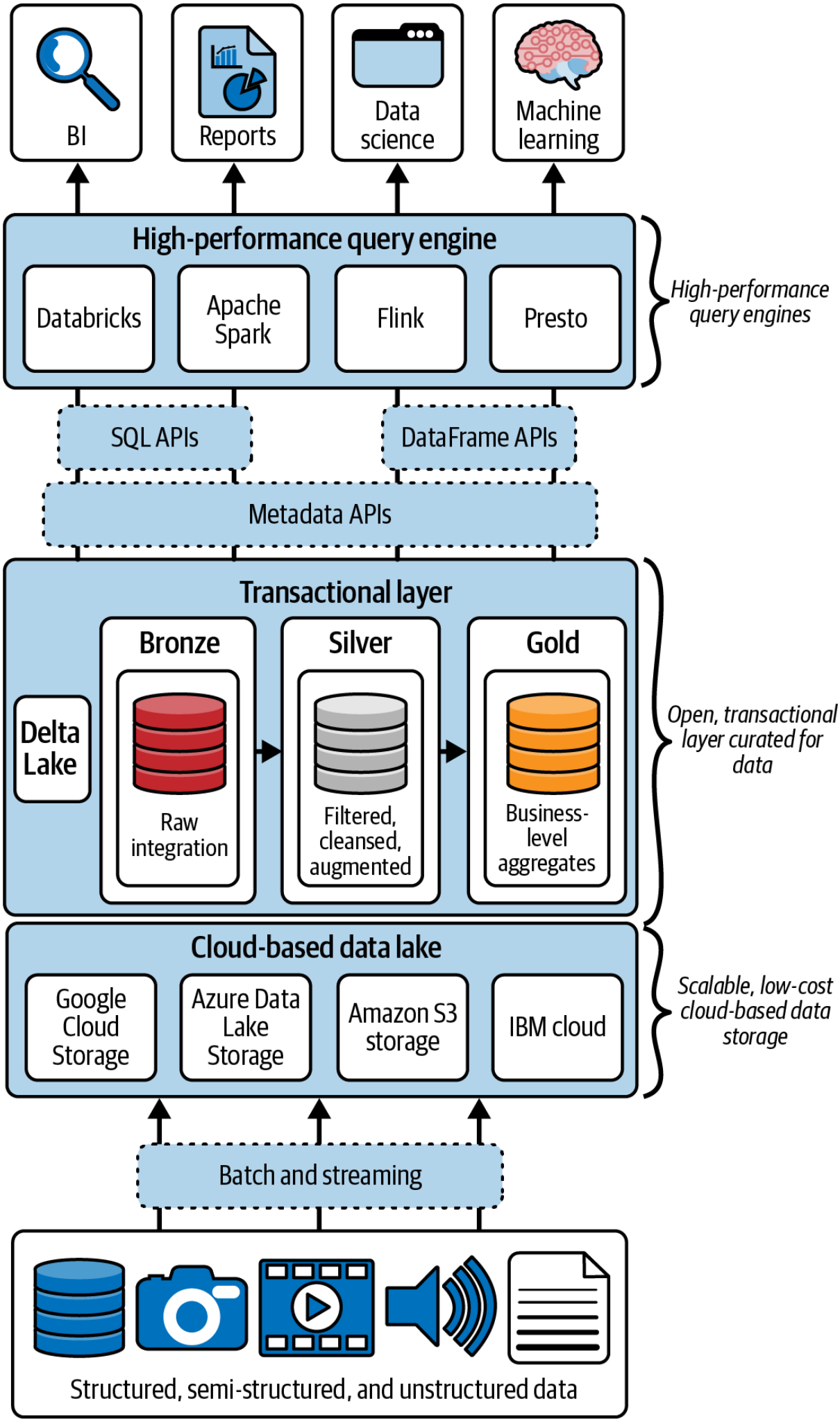
- **Camada Silver**: Aqui, os dados da camada Bronze passam por processos de limpeza, normalização e conformação. É nessa camada que a visão corporativa dos dados em diferentes áreas temáticas começa a se formar. É nessa camada que começamos a aplicar esquemas (schemas), permitindo que eles evoluam nas etapas subsequentes. Por ser a primeira camada onde a qualidade dos dados é garantida e a visão empresarial é criada, a camada Silver serve como uma fonte de dados útil para a empresa, especialmente para análises de autoatendimento e relatórios ad hoc. Além disso, a camada Silver é uma excelente fonte de dados para casos de uso de aprendizado de máquina e inteligência artificial. Esses algoritmos geralmente funcionam melhor com os dados "menos refinados" da camada Silver, em vez dos formatos prontos para consumo encontrados na camada Gold.

- **Camada Gold**: Contém dados prontos para consumo, otimizados para análises e relatórios. Os dados podem estar em um formato de esquema em estrela (com tabelas de dimensões e fatos) ou qualquer outro modelo adaptado ao caso de uso. Nesta camada final, são aplicadas as transformações de dados e as regras de qualidade de dados finais, resultando em dados de alta qualidade e confiáveis que servem como a única fonte de verdade dentro da organização. Essa camada é especialmente ajustada para consumo otimizado por ferramentas de **BI**, relatórios, aplicativos e usuários de negócios. Torna-se a camada principal para leitura de dados utilizando motores de consulta de alto desempenho.

Resumo da Arquitetura Medallion:

Camada	Bronze	Silver	Gold
Valor de negócios	- Auditoria exata do que foi recebido da fonte. - Capacidade de reprocessar sem voltar à fonte.	- Primeira camada útil para o negócio. - Permite descoberta de dados, autoatendimento, relatórios ad hoc, análises avançadas e ML.	- Dados em um formato fácil para os usuários de negócios navegarem. - Altíssimo desempenho.
Propriedades	- Sem regras de negócios ou transformações de qualquer tipo. - Deve ser rápido e fácil trazer novos dados para esta camada.	- Prioriza velocidade para o mercado e desempenho de escrita, apenas transformações mínimas. - Espera-se dados de qualidade.	- Prioriza casos de uso de negócios e experiência do usuário. - Transformações pré-calculadas e específicas para negócios. - Pode ter visões separadas dos dados para diferentes casos de consumo.
Como é feito	- Deve incluir uma cópia do que foi recebido. - Normalmente, os dados são armazenados em pastas com base na data de recebimento.	- Mesclagem Delta (**Delta merge**). - Pode incluir modelagem leve (3NF, Vaulting). - Devem ser incluídas verificações de qualidade de dados.	- Prioriza modelos de dados desnormalizados e otimizados para leitura. - Totalmente transformado. - Agregado.

Ilustração do Lakehouse completo:


BI
Reports
Data science
Machine learning
High-performance query engine
Databricks
Apache Spark
Flink
Presto
High-performance query engines
SQL APIs
DataFrame APIs
Metadata APIs
Transactional layer
Delta Lake
Bronze
Raw integration
Silver
Filtered, cleansed, augmented
Gold
Business-level aggregates
Open, transactional layer curated for data
Cloud-based data lake
Google Cloud Storage
Azure Data Lake Storage
Amazon S3 storage
IBM cloud
Scalable, low-cost cloud-based data storage
Batch and streaming
Structured, semi-structured, and unstructured data