# From Traditional to Modern Data Warehouses

Jana Kostičová

Vybrané technológie analýzy dát, FMFI UK

3.10. / 10.10.2024

# Outline

1. Data analytics and business intelligence
2. BI - history and core principles
3. Traditional data warehouses - main components, architecture
4. Modern data warehouses

# Data analytics

= the process of examining data to uncover patterns, trends, and insights. It involves collecting, cleaning, transforming, and modeling data to extract meaningful information.

**Applications:**

- Business intelligence (BI) - providing insights to support business decision-making
- Science (bioinformatics, astronomy, climate research, …)
- Healthcare
- Social sciences
- …

# Four types of data analytics

1. **Descriptive analytics:**

   = to summarize and describe data. It is often used to identify trends and patterns in data.

   RETAILER: What have been our best-selling products in last 6 months? What has been the demographics of our customers in last year? (based on sales data)

2. **Diagnostic analytics**

   = to understand the cause of a problem. It is often used to identify root causes of issues.

   CAR MANUFACTURER: What has been the major cause of defects in SUV product line within last 3 years? (based on various data: test results, claims, product data, component data, supplier data, …)

# Four types of data analytics

3. **Predictive analytics:**

   = to predict future outcomes. It is often used to forecast sales or customer churn.

   BANK: What is the likelihood of customers defaulting on loans? (based on existing customer/loan data)

4. **Prescriptive analytics:**

   = to recommend a course of action. It is often used to optimize processes or make decisions

   TRANSPORTATION COMPANY: How can we optimize routes and reduce delivery times? (based on existing route data)

These four types of data analytics are often used in combination to provide a comprehensive understanding of data.

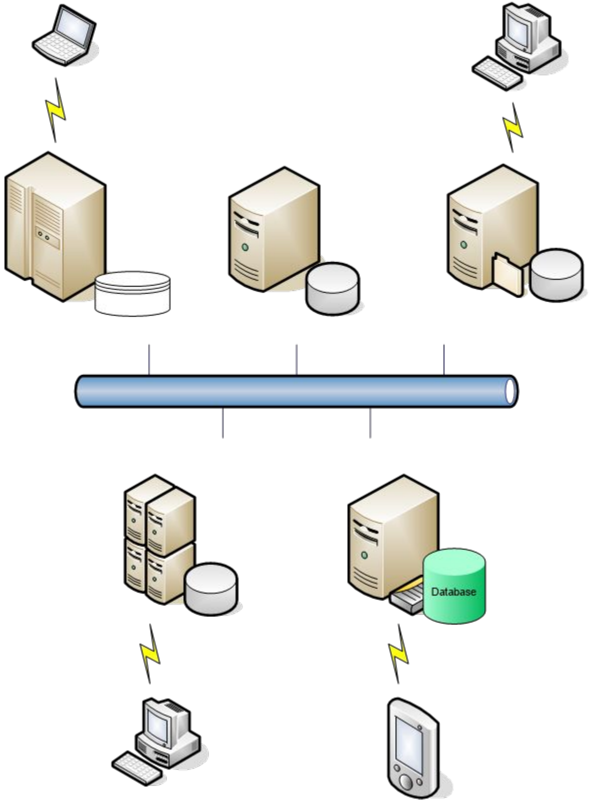
# Corporate IT architecture (1)

**Billing**

- Invoicing, payment processing, revenue collection

**HR - Human Resource System**

- Employees, positions, salaries, vacations, trainings, career progress, attendance, utilization, benefits,...



**Content Management System (CMS)**

- Intranet, information for employees
- Internet, web portal, information for customers

**Customer Relationship Management (CRM)**

- Interactions with customers / potential customers - sales, marketing, customer service

**… and more**

# Corporate IT architecture (2)

**Operational systems** (production systems, transaction systems)

- Stores data needed for common operations in organization
- Support “day-to-day” business
- OLTP (Online Transactional Processing)
  - Focused on transaction processing, optimized for real-time data writes / updates
  - Most common database type
- Also legacy systems
- Each operational system stores data from different point of view
  - Application-oriented data

# Analytical queries vs operational systems

- Did the effectivity of salespersons (the number of products sold per month) increased after a specific training?

  → HR system, CRM system

- What is the relationship between the length of employment of a given salesperson and the number of products sold per month?

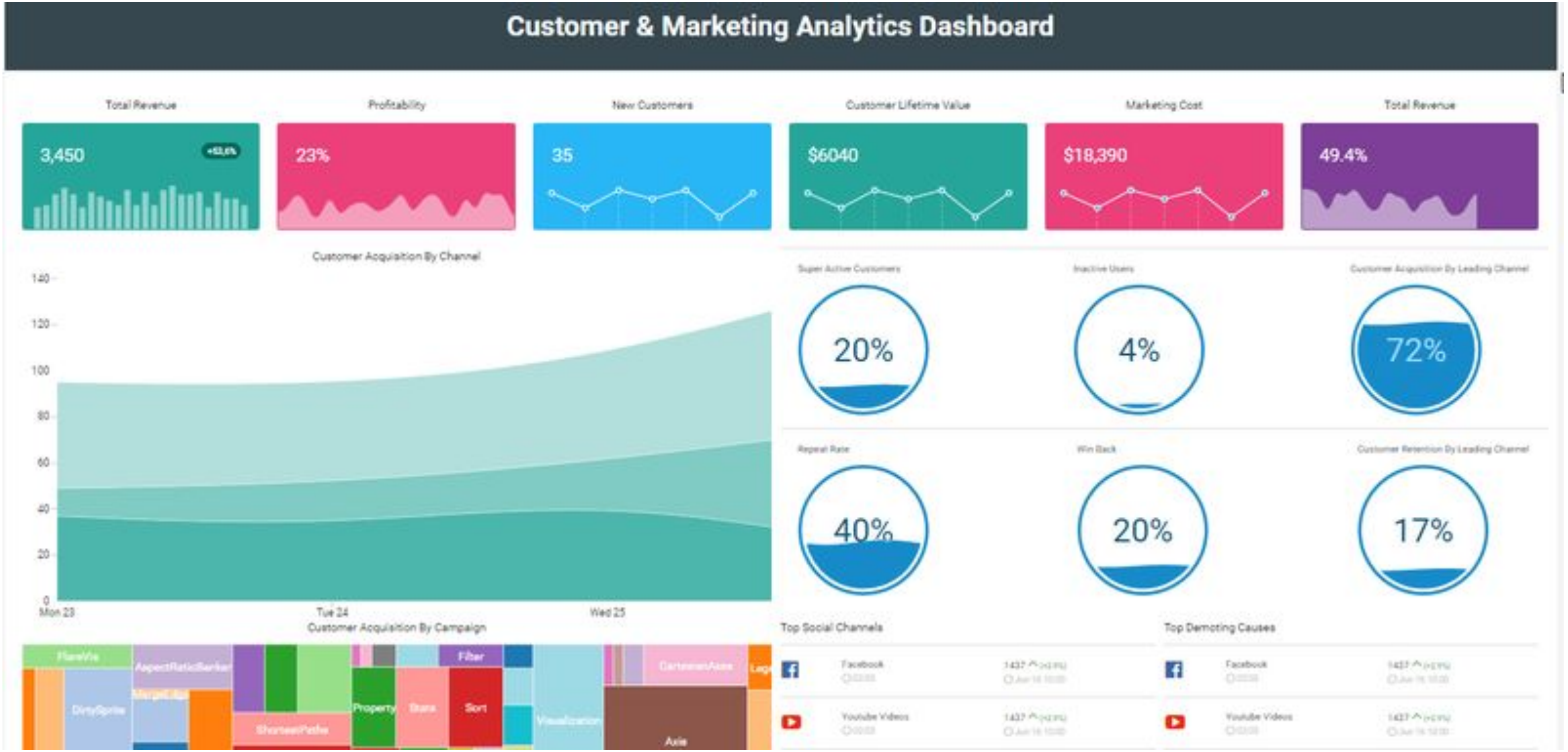
  → HR system, CRM system

- Which customer segment has the largest amount of unpaid invoices?

  → HR system, CRM system

Analytical queries generally need data from different operational systems.

# Analytics Dashboards



# Examples

- Google Analytics demo accounts: LINK
- Power BI demo: LINK

# Enterprise data

There is hidden wealth in enterprise data

- How to gain it?
- How to use it?

We need an unified view of the subjects (such as Customer, Product, Employee)

Enterprises can make better decisions on how to achieve its ultimate goals (mostly increasing the profit)

- Decision support systems (DSS)

# BI requirements

- Integrate data from different operational systems and other relevant data sources
- Allow to store large data volumes and provide scalability
- Provide historical data (months, years, …)
- Provide data designed for analytical queries
- Support efficient execution of complex queries (aggregations, filters, transformations, ..)

The core objectives remain similar also after the “Big data revolution”, With advancements in technology, the requirements will be more ambitious:

- The set of relevant data sources will be broader
- The more complex analytical queries will be required
- There will appear more strict requirements on data analytics speed (real-time scenarios)

# Operational databases and data analytics

Data in operational systems is not stored in accordance with requirements of BI.

- **Data volume**
  - Not optimized for storing and analyzing large amounts of historical data
- **Complexity**
  - Not optimized to execute complex analytical queries efficiently
- **Performance impact**
  - Running analytical queries can impact performance for transactional workloads
- **Data design**
  - Application-oriented data, it can be challenging to perform analytical queries (including joins across multiple tables)

# Before data warehouses

Ad-hoc methods used for data analytics:

**Manual methods:**

- E.g., using spreadsheets, manual data exports and integration
- Long lasting data preparation and less time for analysis itself
- Error-prone (human factor)

**Custom built-in applications:**

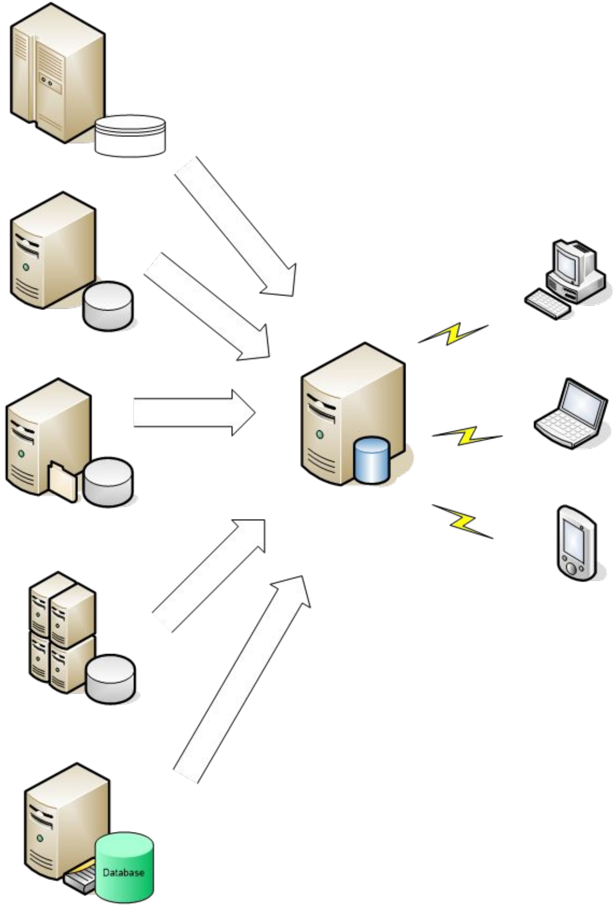
- Cobol / Fortran / C, …
- Inflexible, difficult to handle new or changed requirements (both changes driven by business requirements or by changes in data sources)

Issues arised especially with larger data volumes, more complex analyses and the need to integrate many systems.

** Designed by Wannapik*

# Data warehouse

- Systematic, modular approach
- Data preparation separated from the data analysis
- Data preparation is (mostly) automatic
  - Short-term process
  - The error rate is minimized
- Data analysts get pre-prepared data and they can focus solely on analysis



# Data warehousing era (1990s-2000s)

Data warehouse (**DWH**, DW, EDW) =

*“a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision making process”* (Inmon 1992)

- Bill Inmon: Building the Datawarehouse, 1992 (4th edition - 2005 [1])
- Ralph Kimball: The Data Warehouse Toolkit, 1996 (4th Edition - 2013 [2])

“Fathers” of datawarehousing

Early DWHs:

- Born in environments dominated by relational databases running on traditional servers.
- Primarily on-premise large-scale systems, built mostly by large corporations with significant IT budgets.
- Focused mostly on descriptive / diagnostic analytics

## Subject-oriented

- Data is organized around major entities of interests of an organization (e.g., Customer, Product, Supplier)

## Integrated

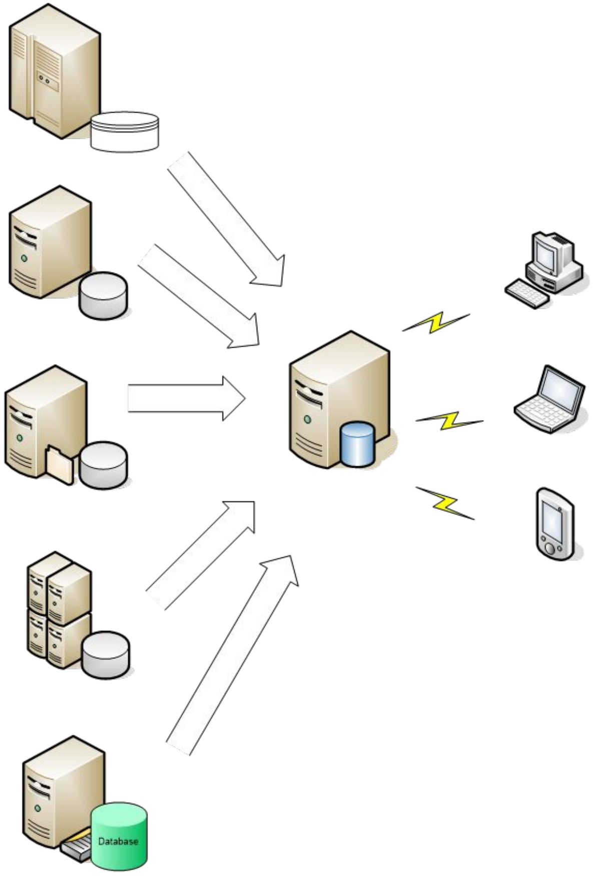
- Integrated data from various sources: operational databases, external databases, even meta-data

## Time-variant

- Multiple years of data (even > 10 years), data is identified by timestamp.

## Non-volatile

- Data is stable. Data is added, but usually not updated or deleted.



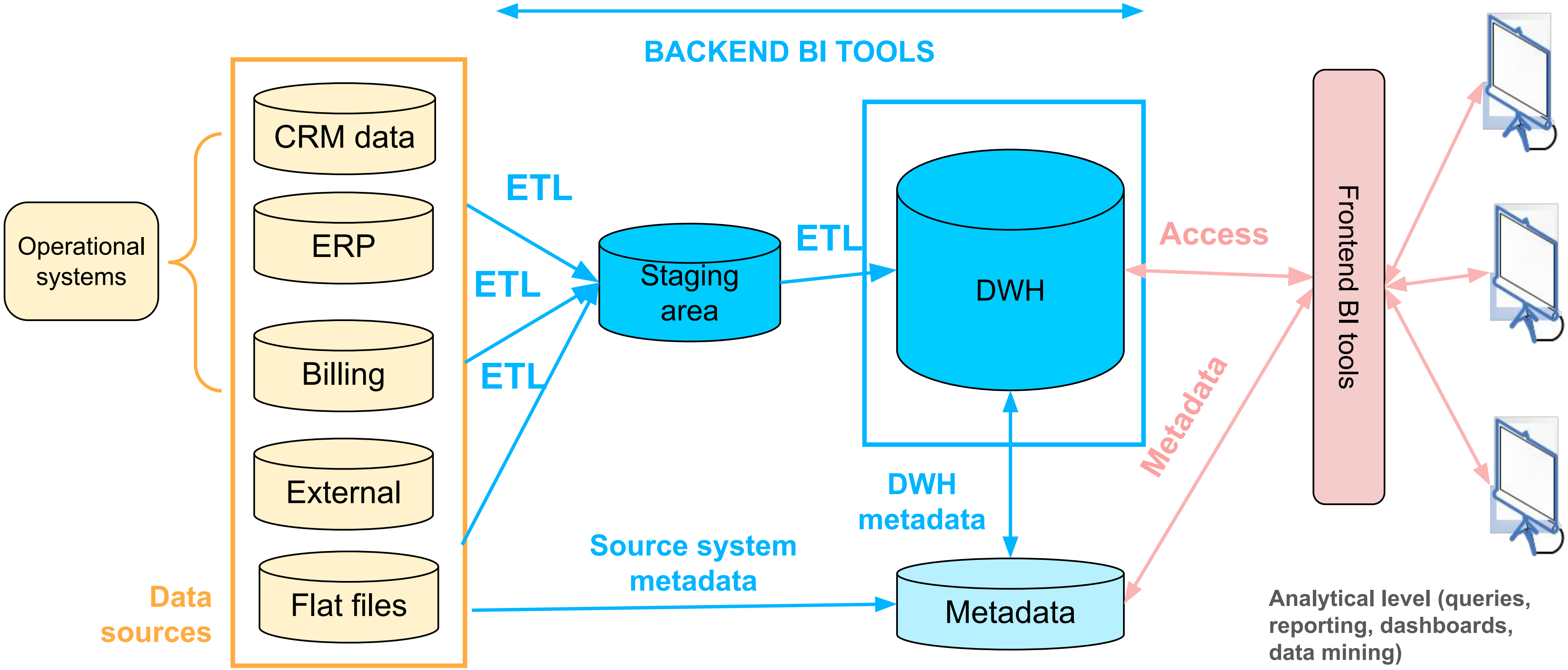
# Are data warehouses obsolete?

NO.

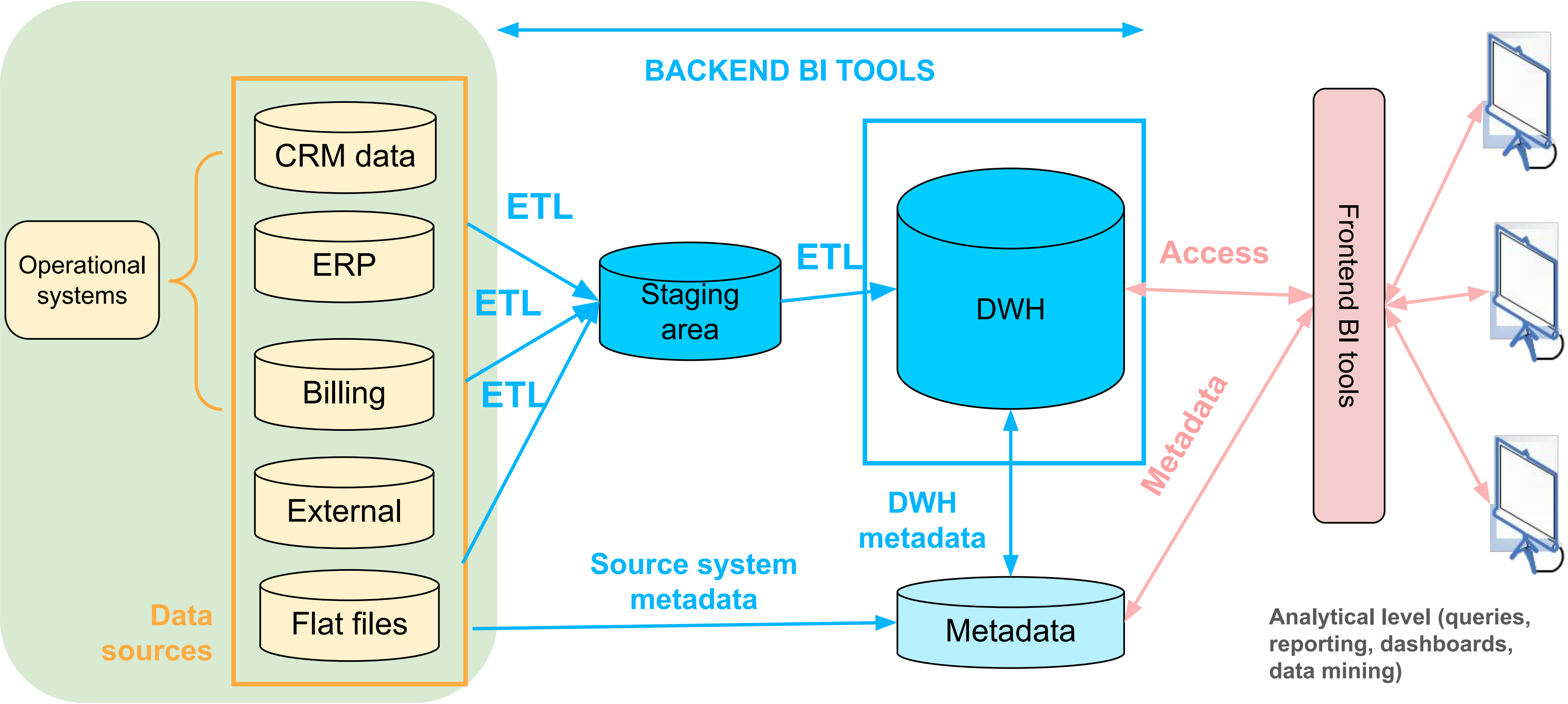
They remain a fundamental component of many organizations' data architectures.

But… they are evolving **to coexist (and cooperate)** with new technologies

# Traditional BI architecture



# Traditional BI architecture - data sources



# Data sources

- Contain enterprise data
- Various form: databases, files (mostly XML, CSV, …)
- Operational systems, external data

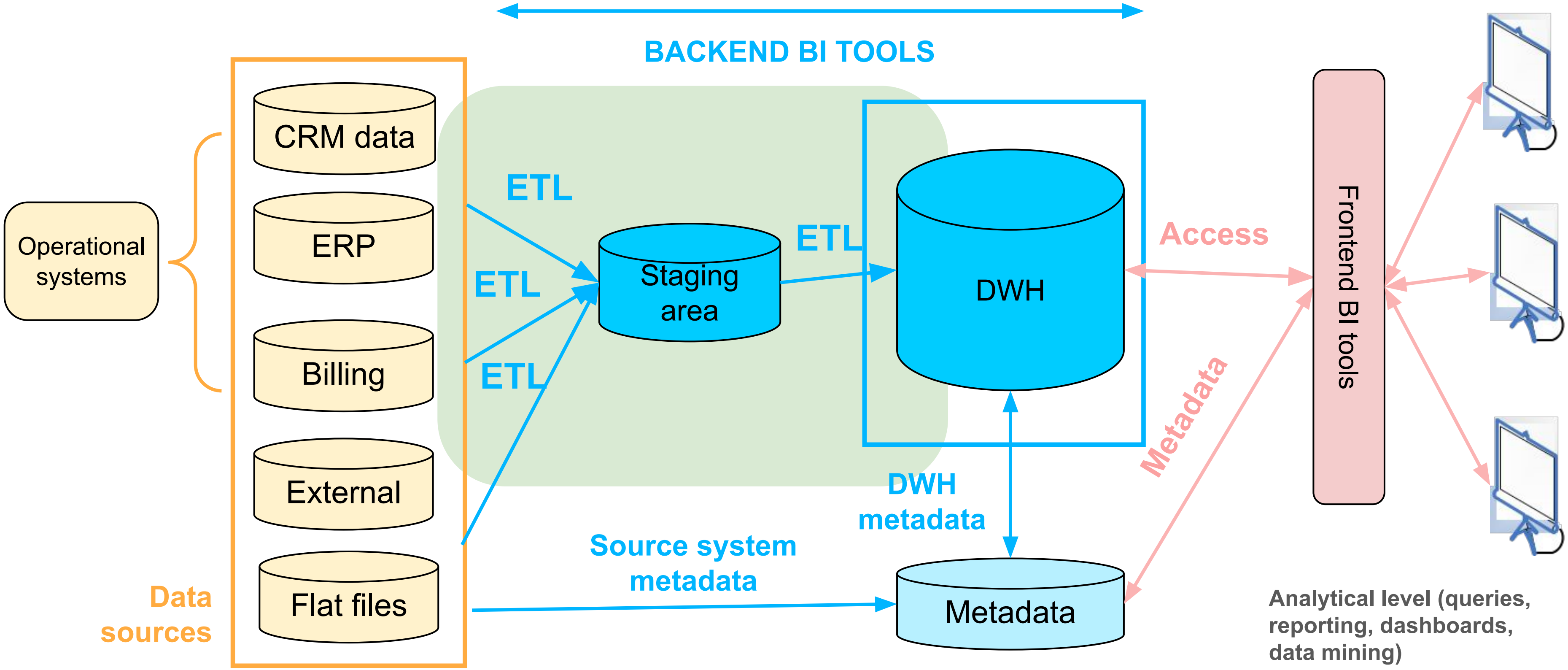
## Problems with operational systems

- They have to be up and running “all the time”
  - We can't burden them with data extraction at any time (usually ok during the night)
- Data is not stored in accordance with BI requirements

## External data

- External DBs or external files
- Examples of external files: territorial division (LINK), database of postal codes (LINK)

# Traditional BI architecture - ETL



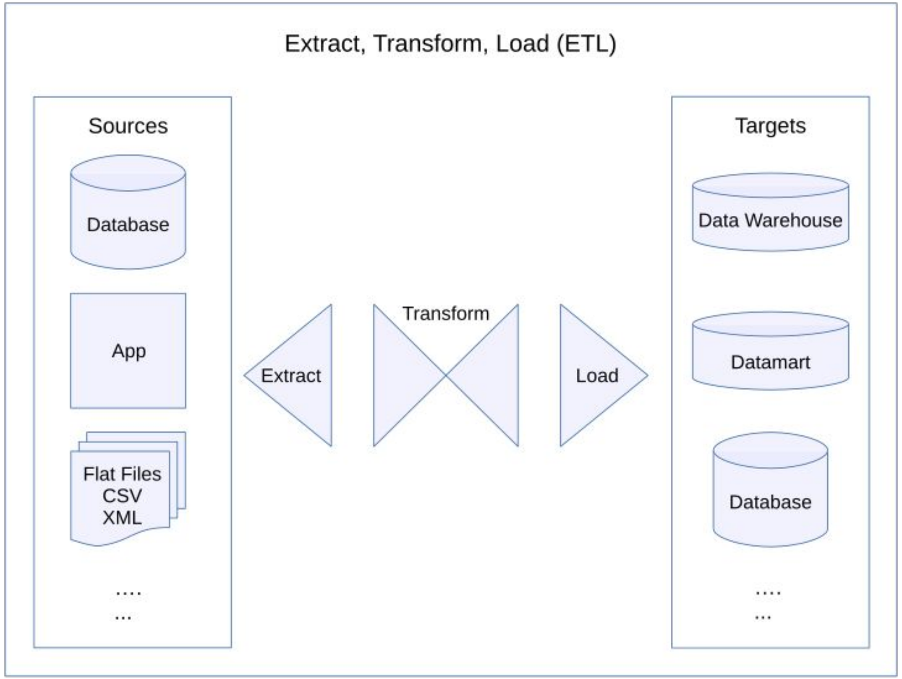
# ETL (Extract - Transform - Load)

= a process transporting (and transforming) data from the data sources to DWH

**Extract**: Data is extracted from source systems in its raw format.

**Transform:** Data is transformed to match the data warehouse schema and requirements.

**Load to DWH:** The transformed data is loaded into the final data warehouse tables.



## Extract

- Form of data storage (DB, CSV files, …)
- Form of access (DB native, ODBC, app, …)
- Protocols (HTTP, FTP, SFTP, ..)
- Full load vs incremental load (CDC - change data capture)
- Automatic loads vs manual loads

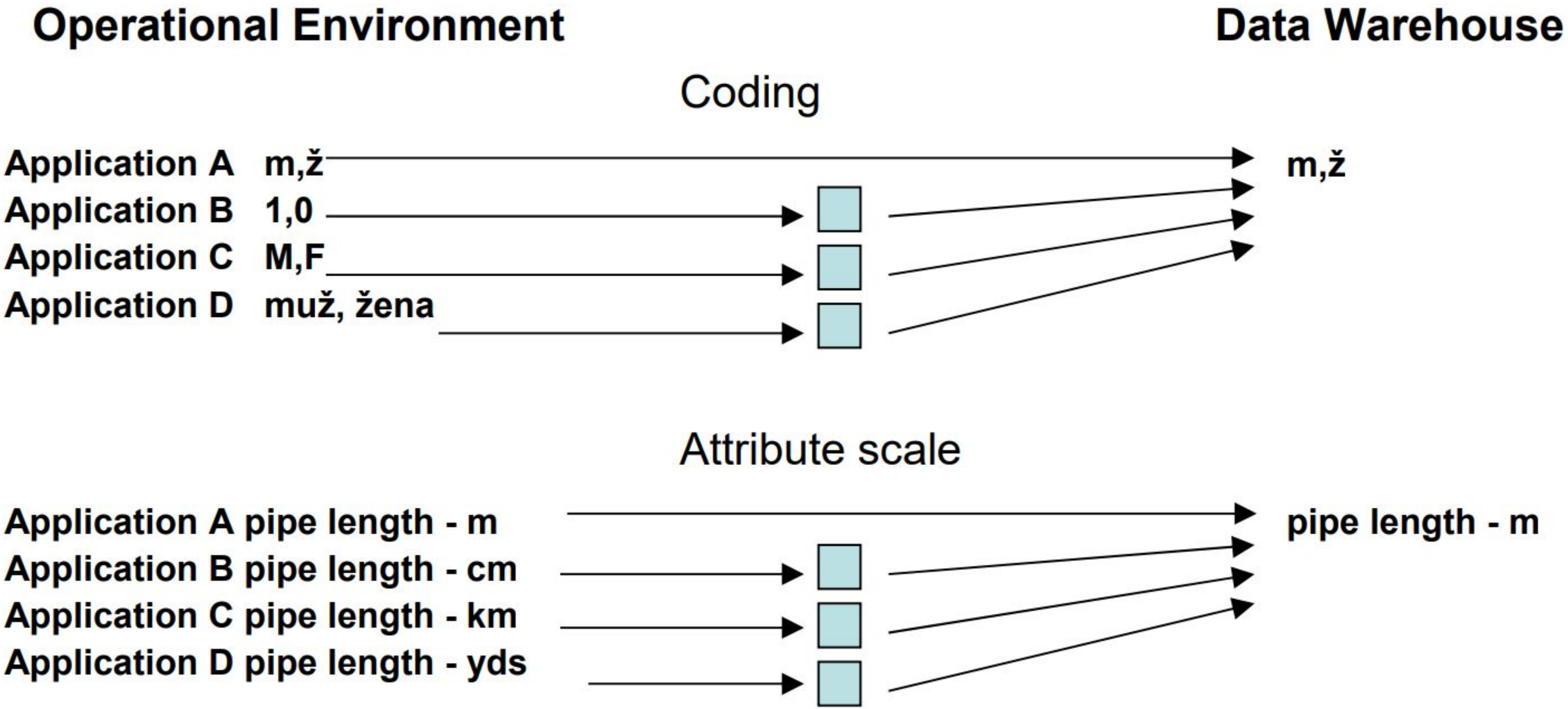
## Transform

- Data validation - checking for accuracy, completeness, consistency
- Data cleansing - handling missing values and inconsistencies, deduplication, etc.
- Data standardization - consistent format for dates, currencies, etc.
- Data masking - protecting sensitive data by replacing or obfuscating certain values
- Data enrichment - add calculated fields, derived attributes
- Data aggregations
- Data transformations - filtering, sorting, joining

## Load

- Target identification
- Loading mechanism: bulk for large datasets, incremental, CDC,...

# Example - data standardization



**Operational Environment**

**Data Warehouse**

Coding

**Application A m,ž**

**Application B 1,0**

**Application C M,F**

**Application D muž, žena**

**m,ž**

Attribute scale

**Application A pipe length - m**

**Application B pipe length - cm**

**Application C pipe length - km**

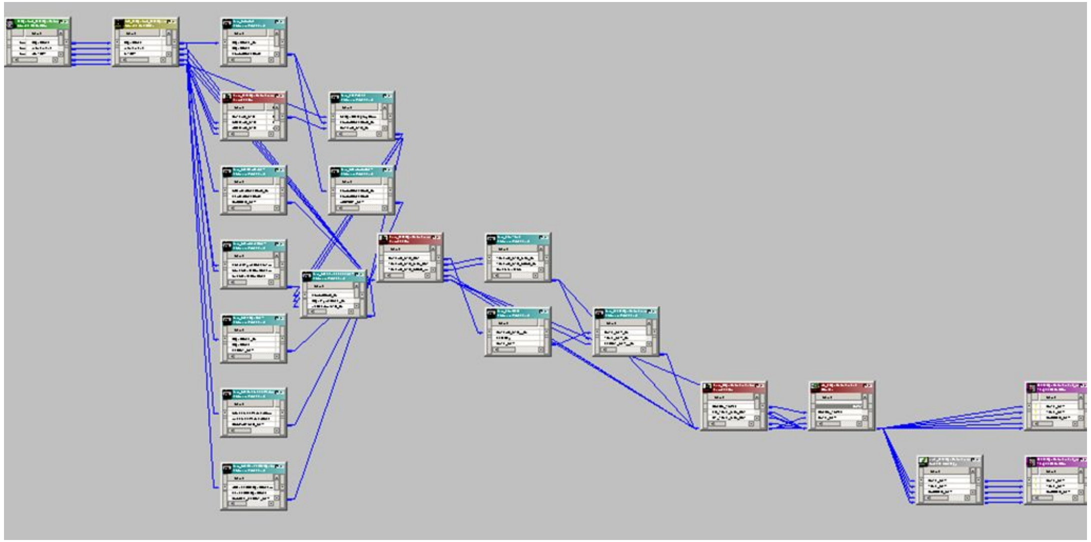
**Application D pipe length - yds**

**pipe length - m**

# ETL - implementation

- Hand-coded scripts vs ETL tool

- Scheduler: daily, weekly, monthly, event-driven, manual
- Exception handling
- Task recovery and restart
- Metadata
- Security (sensitive data)



Example: Sorting data in Informatica PowerCenter: LINK

# Staging area (stage, STG)

Stage is not necessarily needed in smaller DWHs

= temporary data storage, “snapshot” of source systems

**Main goal**

- to quickly load large volumes of raw data from source systems

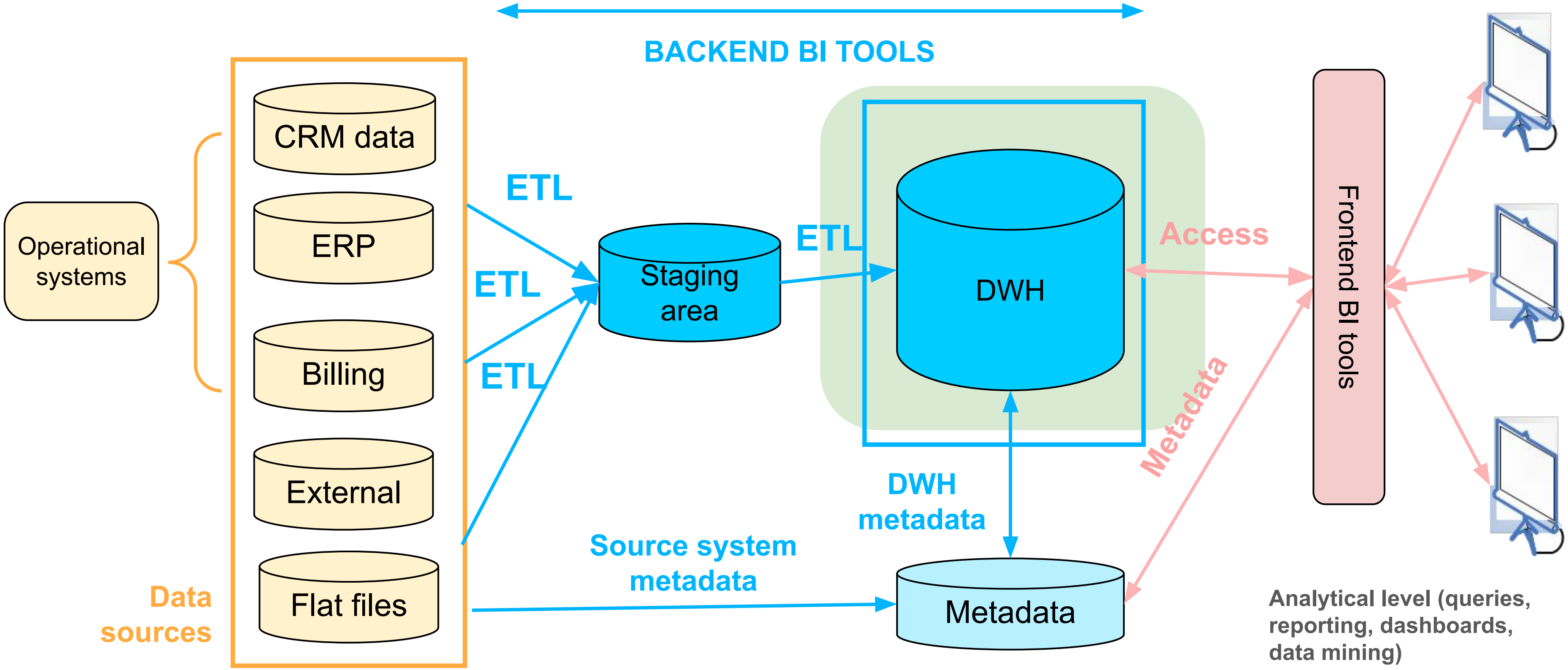
**Benefits**

- Handles inaccessibility of source systems
- Source systems are not burdened with complex operations
- Easy data re-loads

**Properties of STG tables**

- They should have the same structure as the tables in the source systems
- Typically no referential integrity
- Some lightweight transformations can be performed before loading data to stage such as
  - Data extraction and filtering, basic data cleansing, data masking

# Traditional BI architecture - DWH



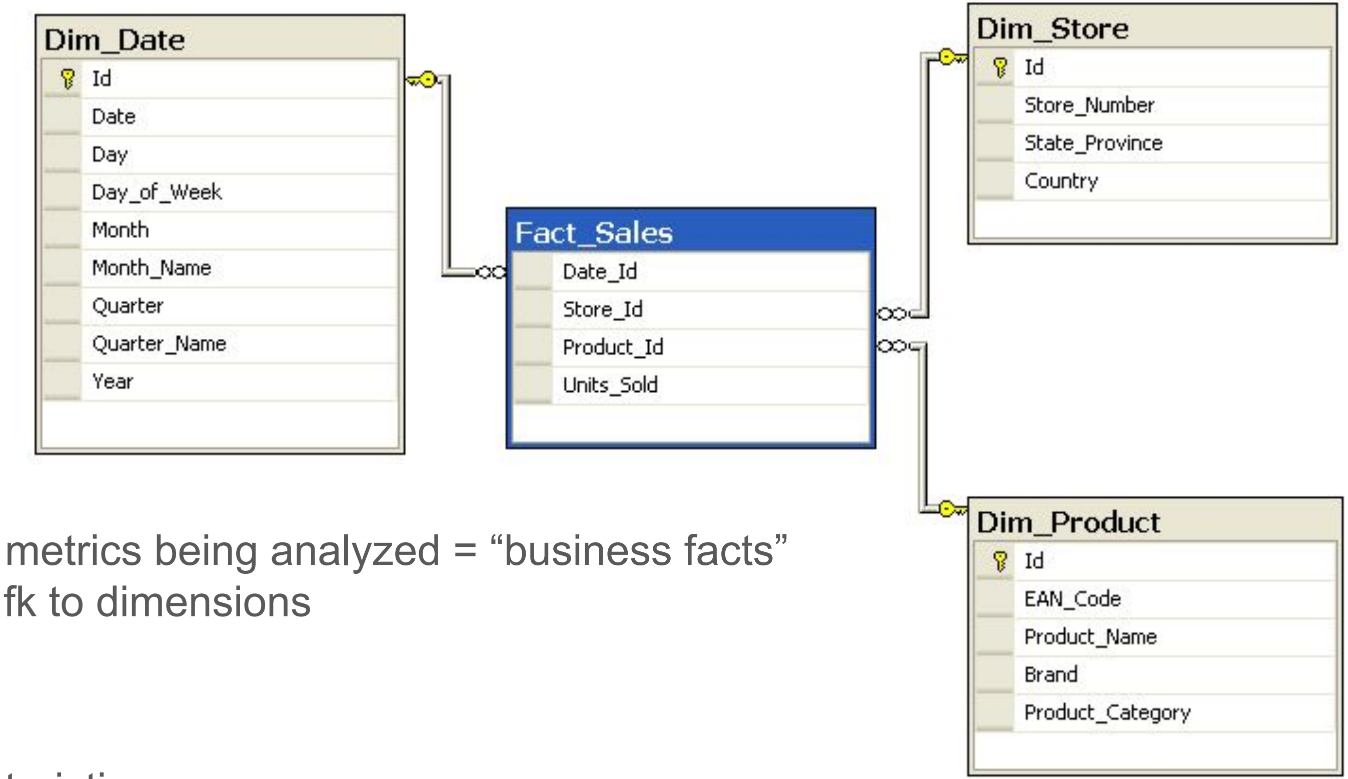
Analytical level (queries, reporting, dashboards, data mining)

# DWH

- Pre-prepared data for data analytics
  - Mostly dimensional model is used
- Table types (dimensional model)
  - Dimensions
  - Facts
  - Aggregation tables (pre-computed less detailed data for faster queries)
  - .. other
- Common data organization
  - Star schema, snowflake schema

# Star schema

A fact table at the center connected to multiple dimensions.



**Dim_Date**
- Id
- Date
- Day
- Day_of_Week
- Month
- Month_Name
- Quarter
- Quarter_Name
- Year

**Fact_Sales**
- Date_Id
- Store_Id
- Product_Id
- Units_Sold

**Dim_Store**
- Id
- Store_Number
- State_Province
- Country

**Dim_Product**
- Id
- EAN_Code
- Product_Name
- Brand
- Product_Category

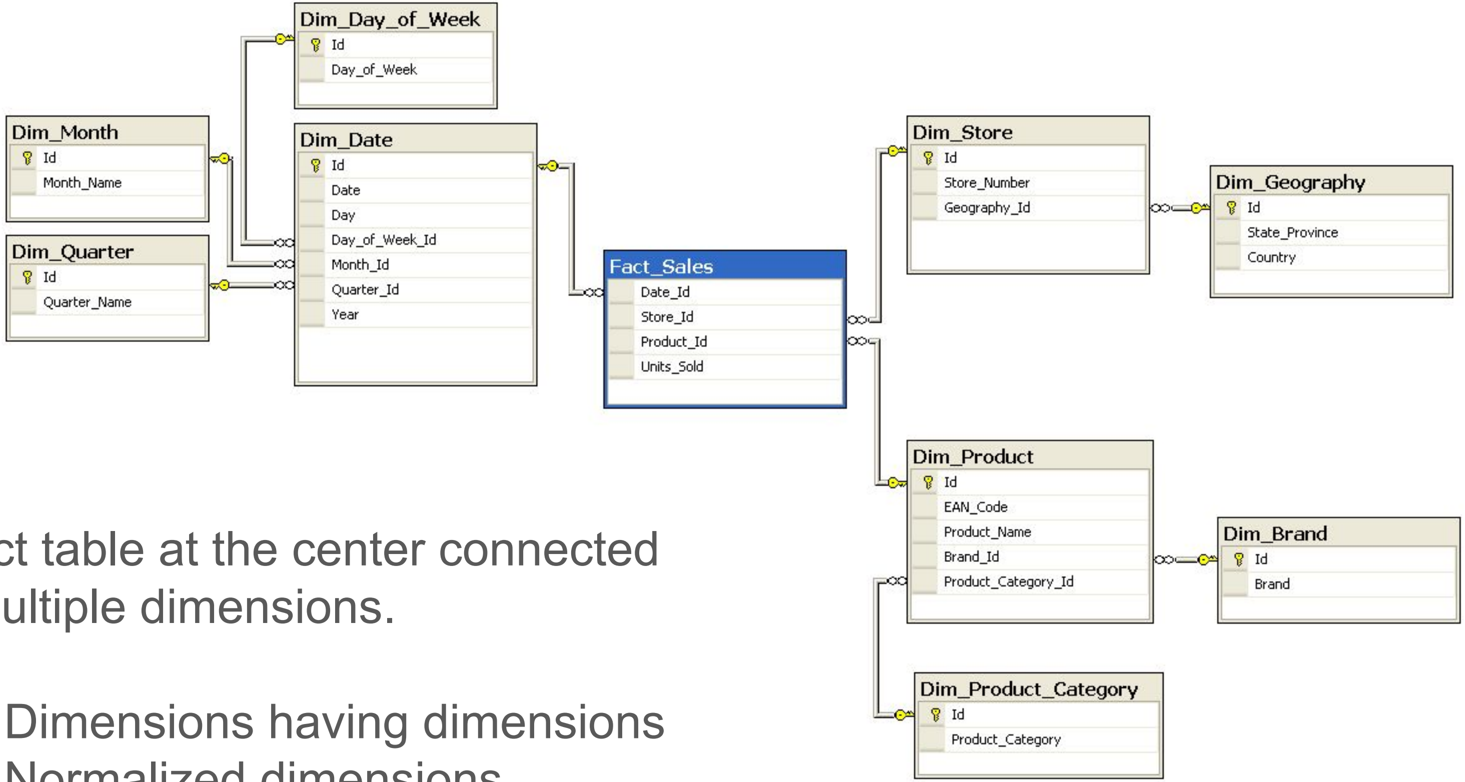
Fact table:

- Measurements or metrics being analyzed = “business facts”
- Numeric values + fk to dimensions
- Can grow fast

Dimension table:

- Attributes / characteristics
- Smaller number of records
- Can have many attributes
- Denormalized dimensions

# Snowflake schema



Dim_Day_of_Week
Id
Day_of_Week

Dim_Month
Id
Month_Name

Dim_Quarter
Id
Quarter_Name

Dim_Date
Id
Date
Day
Day_of_Week_Id
Month_Id
Quarter_Id
Year

Fact_Sales
Date_Id
Store_Id
Product_Id
Units_Sold

Dim_Store
Id
Store_Number
Geography_Id

Dim_Geography
Id
State_Province
Country

Dim_Product
Id
EAN_Code
Product_Name
Brand_Id
Product_Category_Id

Dim_Brand
Id
Brand

Dim_Product_Category
Id
Product_Category

A fact table at the center connected to multiple dimensions.

- Dimensions having dimensions
- Normalized dimensions

# Example

Google analytics dimensions and facts (metrics): LINK

# Data mart

- A subset of DWH data
- Focuses on a specific subject area or business department (e.g. finance, marketing, sales, management, ..)
- Designed to provide targeted data and analysis for a particular group of users, such as a sales department or a marketing team.

Two different principles

- Inmon’s vs Kimball’s approach

# Inmon’s approach

Top-down approach:

- First create a centralized DWH and then create data marts as needed

**CIF – Corporate Information Factory**

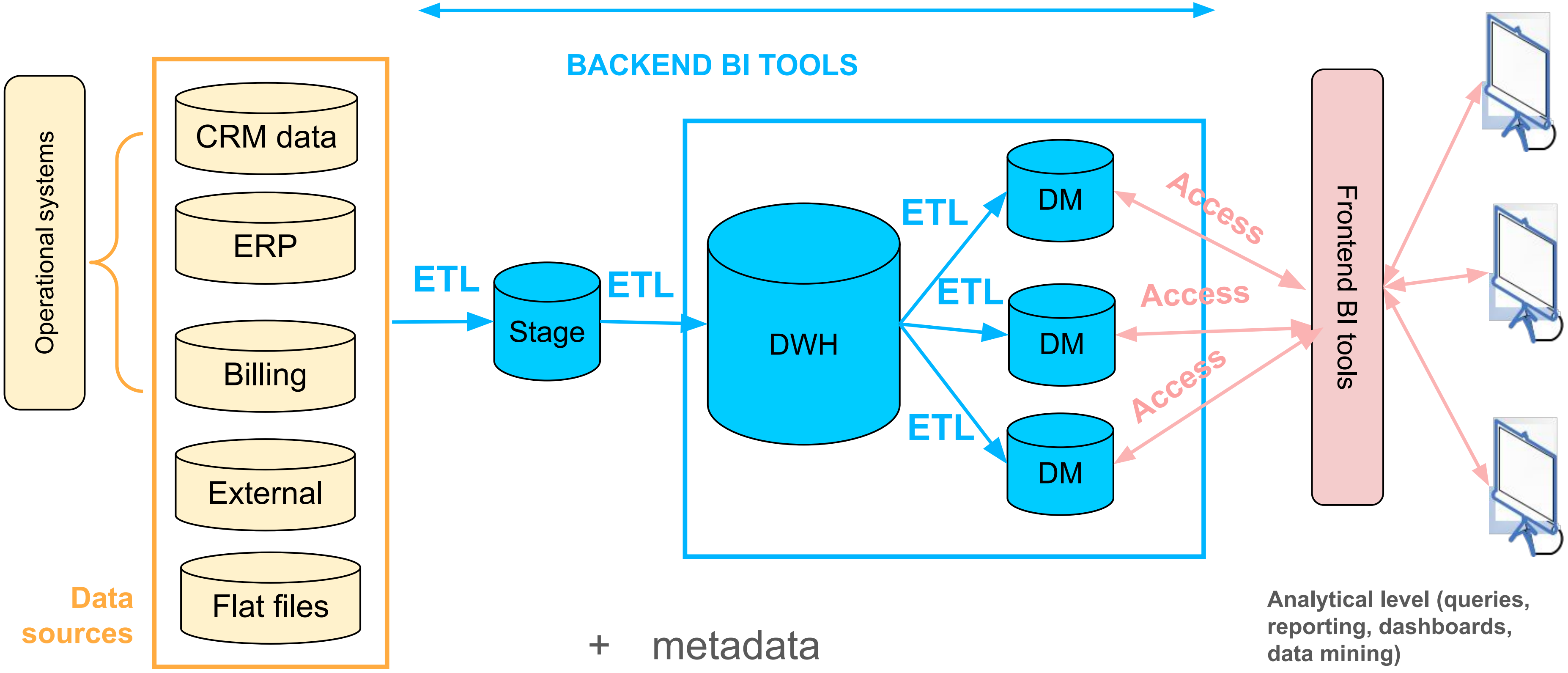
3-layer architecture

1. Staging area
2. Central data warehouse (normalized)
3. Data marts – independent, fed by the data from central DWH, each department has its own, customized data mart

(+) DWH = a single source of truth, consistent and accurate

(-) Time-consuming to build and maintain

# Inmon’s approach



+ metadata

# Kimball’s approach

Bottom-up approach:

- First create data marts for specific business needs and then integrate them into large DWH

**DWHBus – Data Warehouse Bus**

2-layer architecture

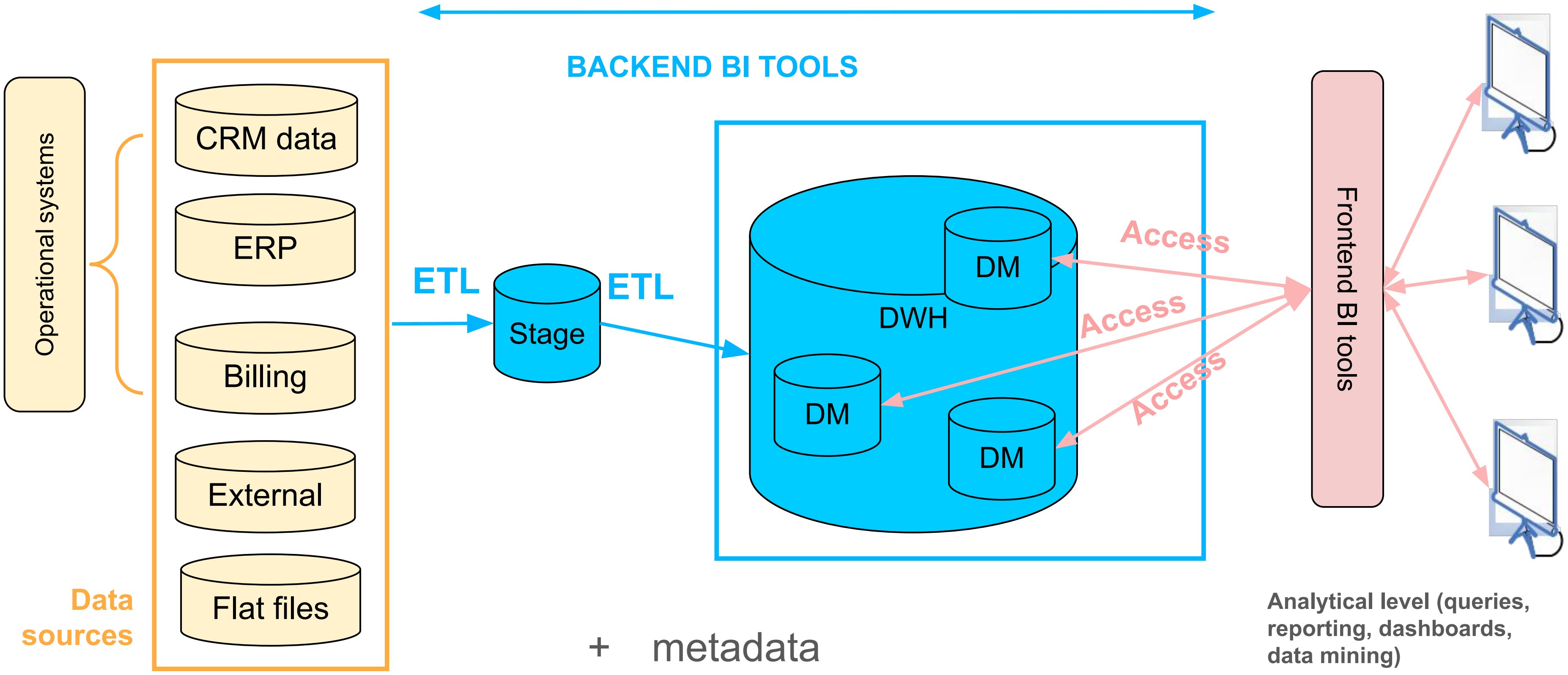
1. Staging area
2. DWH = collection of integrated data marts, conformed dimensions
    - Consolidated, shared dimensions
    - Data marts are often denormalized to improve query performance - star schema
    - Enteprise Data Warehouse Bus Matrix

(+) More agile approach

(-) DWH may become inconsistent as more and more DMs are added

(-) Data redundancy due to denormalization

# Kimball’s approach



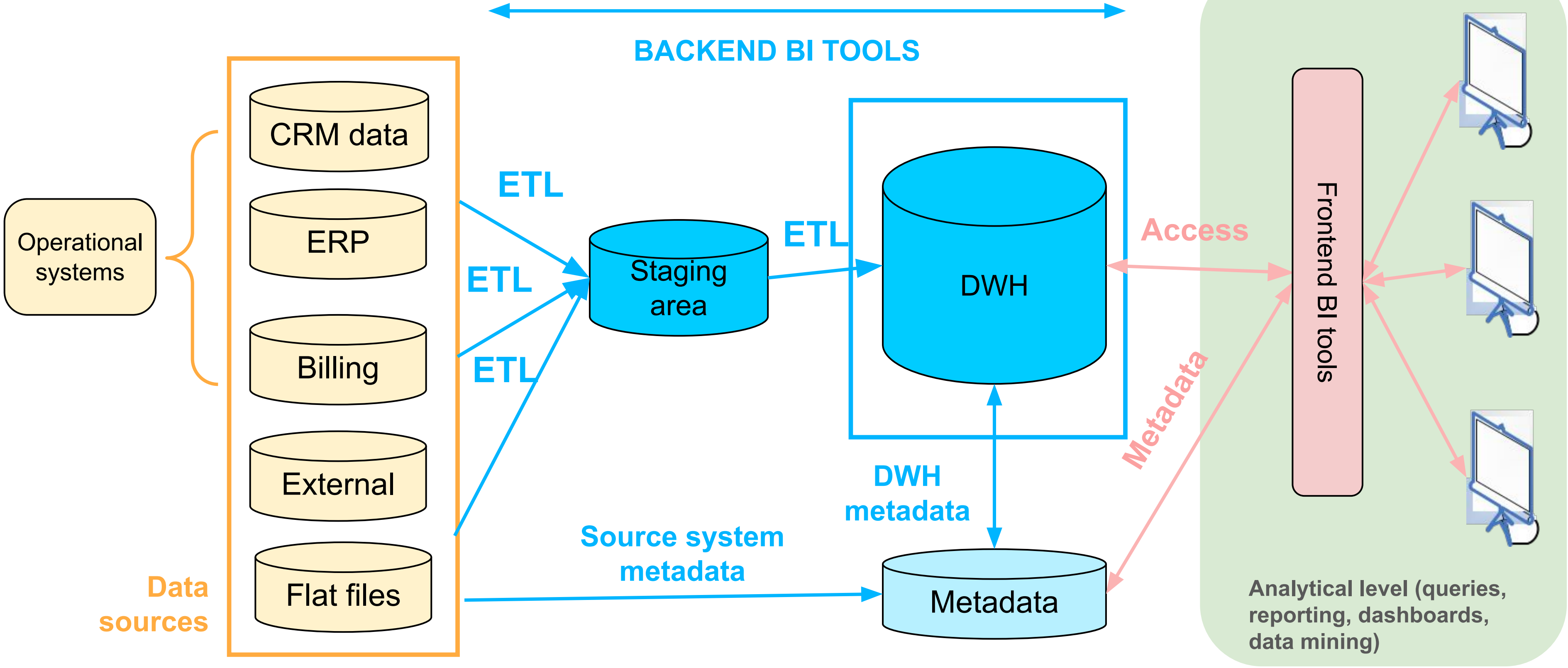
Analytical level (queries, reporting, dashboards, data mining)

+ metadata

# OLTP vs OLAP

| OLTP - Online Transactional Processing | OLAP - Online Analytical Processing |
|---|---|
| Current and recent data | Historical data |
| Detailed data | Both detailed and summarized data |
| Application oriented | Analytics oriented (Subject oriented) |
| Insert, update, delete | Read (including complex queries) |
| Optimized for fast transactions | Optimized for complex queries (dimensional model) |

# Traditional BI architecture - front-end BI tools



# Frontend BI tools

- Make information available to the users
- Tables + data visualizations: dashboards, charts, graphs
- Predefined reports vs ad-hoc (custom) reports

*Microsoft Power BI, SAP Business Objects, Tableau, Qlik Sense*

- Advanced users can access DWH / Data marts directly
  - They must know the data model and query language

- Data mining tools
  - Uncovering historical hidden trends and patterns, later also making predictions
  - Statistical methods, decision trees, neural networks,..
  - Later broader term "Data science" (+ new methods such as machine learning)

- Google Analytics demo accounts: LINK
- Power BI demo: LINK

# OLAP cube

OLAP ≠ OLAP cube !

= data structure for quick data analysis through multiple dimensions

Querying DWH / Data marts could be slow

Once created, OLAP cube allows for efficient:

- Slicing and dicing
- Drill-down, drill-up

Example: How many TV service sales did we have in Asia in Q1?

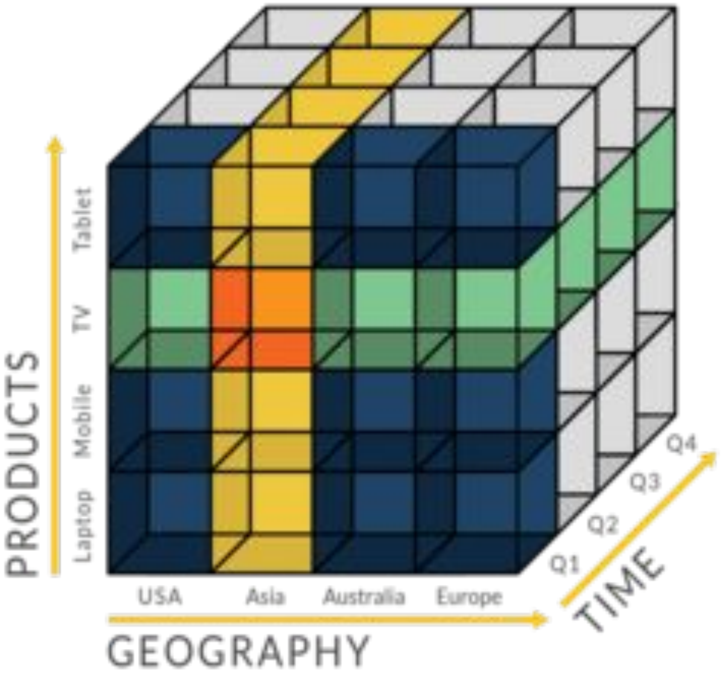
- “Asia” – **yellow cut**
- “TV“ – **green cut**
- “Q1” – **blue cut**

Creating OLAP cubes could be time-consuming

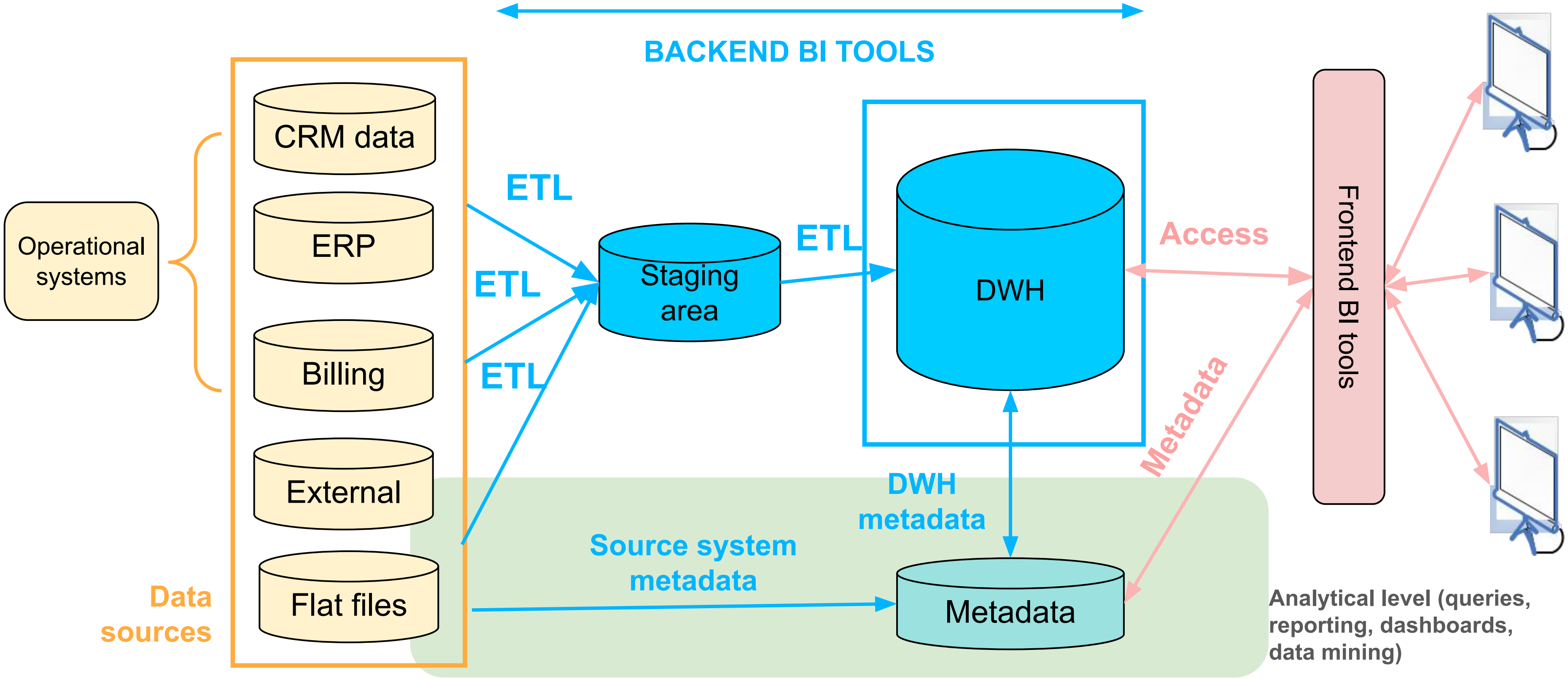
- Mostly implemented as materialized views

Less used in modern data warehouses

- New data storages allow for more efficient querying directly via DWH / Data marts



# Traditional BI architecture - metadata



# Metadata

= data describing data

**Technical**

- Table: creation date, author, column names, database name
- Column: data type, maximum length, table
- Report: type, creation date, data sources

**Business**

- Business definitions, business rules
  - Table name, table description, data preparation rules

**Process Execution**

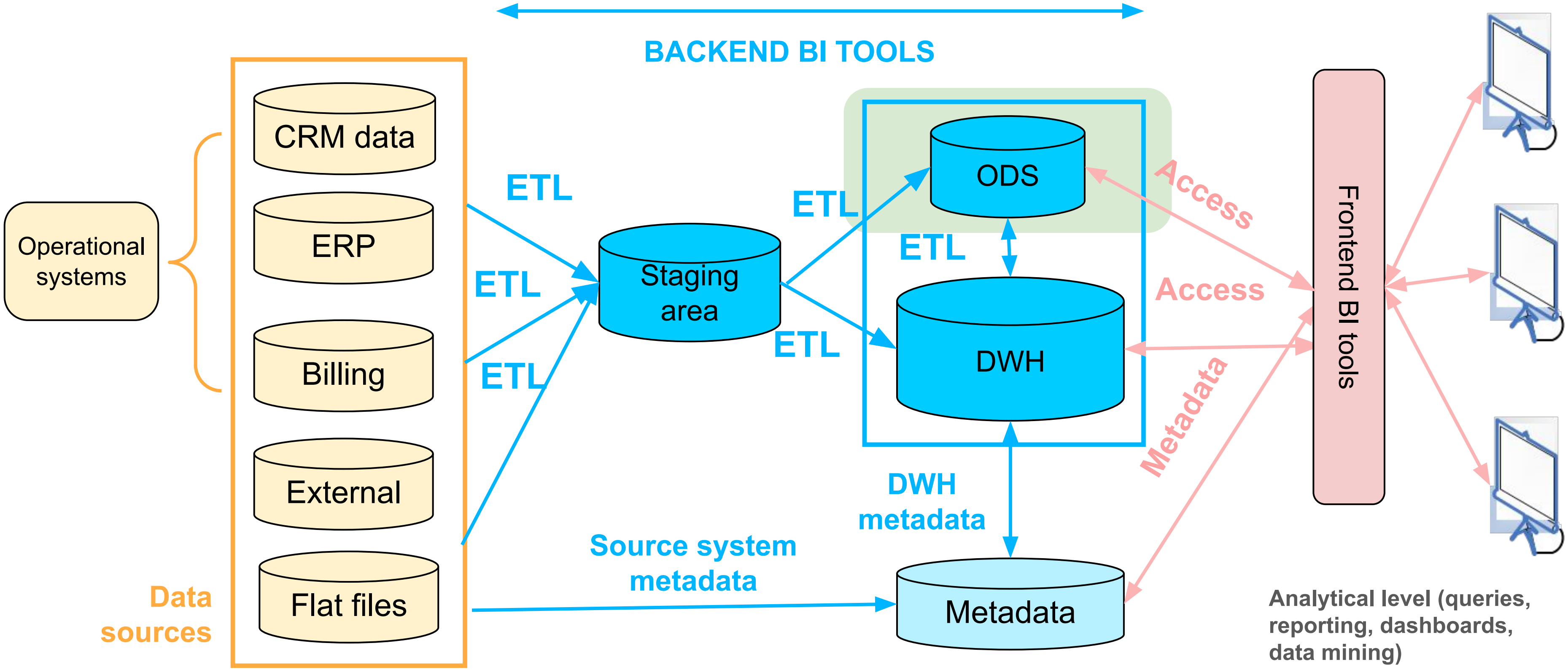
- Data “freshness”
- Length and result (success / fail) of data processing

Managed for each data storage / ETL separately or in centralized metadata management tool

- Automatic metadata extraction (if possible), metadata integration, visualizations

**Example:** Data lineage with QlikSense Nodegraph LINK

# Traditional BI architecture - ODS



# ODS (operational data store)

= a specialized analytical database designed for operational decision support

(DWH is used primarily for tactical and strategic decision support)

- Normalized format
- Often near-real-time updates
- Subject-oriented, partially integrated, but
  - Current-valued (little or no history is stored)
  - Volatile (data in ODS are updated following the updates in operational systems)
- Schema:
  - Mirror of operational systems, lightweight data processing (data cleansing, validations, data masking.. )
  - Some data computed based on DWH input (e.g. profile records)

**Examples:**

1. Insurance company: Incoming calls are routed automatically or manually to agents based on their current busyness and performance (source data, <= 15 minutes accepted latency)
2. Telco operator: Employees of a call center get customer profile record to be able to handle customer's requests properly (data from DWH)

# Back to BI definition

**BI = providing insights to support business decision-making**

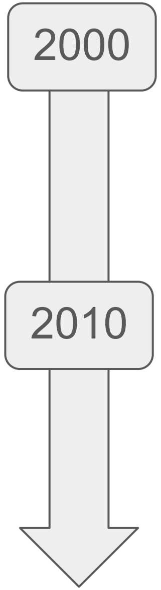
- It helps to optimize operations - the end product is separate from the analytics
- Used internally, customers / users don't see or interact with them
- Mostly a secondary function over data collected primarily for operational purposes

- BI system downtime:
  - Some issues can appear (delayed decision-making, missed opportunities for optimization, and less efficient operations)
  - The company usually continues its core operations because BI is not used for day-to-day business

# Data-driven products and services (2000 - )

Big tech companies: data are valuable not just for decision-making but also for building new revenue streams.

Data analytics used for BI **+ as a part of products / services**



2000

2010

**Amazon:** recommendation system, **Google:** targeted advertising

**Facebook, LinkedIn**: selling insights about their users (through ads)

**Google Maps:** real-time traffic / route suggestions based on mobile phone locations

**Netflix, Uber, Spotify:** machine learning used to offer personalized services, “data monetization”

**Tesla:** Autopilot feature relies on data analytics gathered from Tesla’s fleet of vehicles

**Airbnb:** predictive algorithms used to help hosts optimize pricing and availability

# New data and new requirements

## Original BI requirements still hold:

- Data analytics over integrated data
- Data analytics over large data volumes
- Data analytics over historical data
- Data designed for analytical queries
- Efficient execution of data analytics

## (New) extended requirements:

- Data analytics over **even larger (and increasing)** data volumes
- Data analytics over (a combination of) stored data and **data streams**
- Data analytics over (a combination of) structured, semi-structured and **unstructured** data
- Data analytics delivered in **“real-time”**
- **Predictive** & **prescriptive** data analytics

- Cost-effective data analytics

**New data sources:** social media data, IoT data, ..

- Some new data was generated solely for analytical purposes (interaction data, clickstreams, ..)
- Many of them in form **of data streams**
- New storage types and file formats

Typically one of the options:

- **real-time (RT)** ~ microseconds to milliseconds
- **near-real-time (NRT)** ~ seconds to minutes
- **micro-batch processing** ~ minutes to hours

(**batch processing** - higher latency)

# Issues with traditional methods

## Scalability of traditional RDBMSs

- Difficult to scale horizontally (ACID, data distribution complexity, centralized management, ..)
- Vertical scaling is possible, but more expensive and less flexible
- Storage and computation are coupled
  - Vertical scaling means upgrading the entire server

**Horizontal scaling (scaling out):** Adding more nodes to a cluster to increase capacity.

**Vertical scaling (scaling up):** Increasing the resources of an individual node, such as adding more CPU cores, memory, or storage.

## Batch processing

- DWH loads were triggered daily (weekly, monthly, ..)
- Time-consuming ETLs and analytical queries
- Latency in the analysis (  > 24 hrs)

## Not designed to process continuous data streams “on-the-fly”

## Limited data analytics capabilities

- Mostly descriptive and diagnostic analytics

# Modern data warehouses

- Architecture
  - Core principles of traditional architectures still relevant
  - Some new (and yet immature) concepts emerged:
    - Data lake / data lakehouse
- Technologies
  - New technologies used to implement the DWH architecture
    - Storage systems, etc.
    - To address new data and new requirements
  - New technologies ≈ “big data” technologies
  - Many of them - difficult to choose the proper technology stack
- Traditional BI tools adapted to work with new technologies
  - ETL tools, Frontend BI tools
- Many of new “Big Data” technologies have applications beyond data warehousing and data analytics !

# New technologies - storage systems & computation frameworks

- **Distributed file systems (DFS)**
  - Hierarchical structure similar to file system
  - Horizontal scalability, storage and compute decoupling, parallel processing
    *HDFS + MapReduce / Spark*
- **Cloud object storage systems**
  - Similar properties to DFS, but flat structure of individual objects identified by metadata
    *Google Cloud storage + Google Cloud compute engine, Amazon S3 + AWS Lambda / EC2*
- **NoSQL databases**
  - Many support both horizontal scalability and decoupling of storage and compute
    *Apache Cassandra, MongoDB, Amazon DynamoDB, Google Bigtable*
- **NewSQL databases**
  - Relational databases with support for horizontal scaling
    *Google Spanner, CockroachDB*
- **New file formats**
  - Intended for storing and processing large datasets
    *Parquet, ORC (Optimized row columnar), Avro*

Many of new technologies are primarily intended for cloud environment.

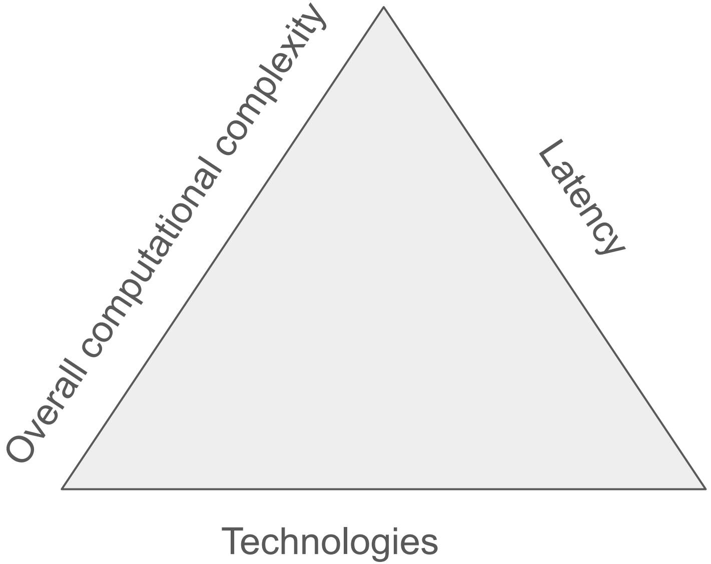
# New technologies - data streams & RT / NRT processing

Technologies for data stream processing (not necessarily RT / NRT):

- Stream processing technologies
  - *Apache Kafka*
  - *Apache Flink, Apache Storm, Spark Streaming*

Technologies for RT / NRT processing:

- New storage & compute systems
  - Parallel processing
  - In-memory data processing (*Redis*)
  - Columnar storage
- CDC “real-time” extraction



Higher overall computational complexity generally results in higher latency. This can be (partially) compensated by using more sophisticated technologies.

# New technologies - advanced analytics techniques

- Machine-learning, deep learning
  - Machine-learning / deep learning models integrated with DWH or data lake / lakehouse
  - Enhances the capabilities of all types of data analytics
- Natural Language Processing (NLP)
  - Analysis of unstructured data (e.g., customer reviews, social media posts)

# Cloud services

Cloud = a network of remote servers and data centers that are accessed over the internet

Highly scalable, both vertically and horizontally

- Usage of new technologies
  - All that have been mentioned (and even more)
  - Cloud-based technologies - optimized for cloud environments, some of them are vendor-specific (e.g., AWS -> AWS S3, AWS Lambda) - these cannot be used on-prem
- Managed services
  - Management and control of a customer's cloud resources (IaaS, PaaS, SaaS, ..)
  - Scalability, reliability, security
- Pre-configured data analytics solutions
  - Data warehouse / data lake / hybrid architectures
  - Cloud-based data analytics platforms: *Databricks, Snowflake, Amazon Redshift, …*
- Better affordability
  - Pay-as-you-go
  - Decoupling of storage and compute - cloud storage is generally cheaper than compute

# Data lake

- Centralized repository of data in its raw format
  - All data (including historical)
  - Structured / semi-structured / unstructured data

    *(CSV, JSON, Parquet, ORC, Avro, images, videos, texts, ..)*
  - Support for data streams
  - Schema on-read: schema is defined when the data is queried, data discovery driven by metadata
- Implementation
  - Distributed file system / cloud object storage system
- Criticism
  - Poorly maintained data lakes turn to data swamps
  - Strict metadata management is important

## Data lakehouse

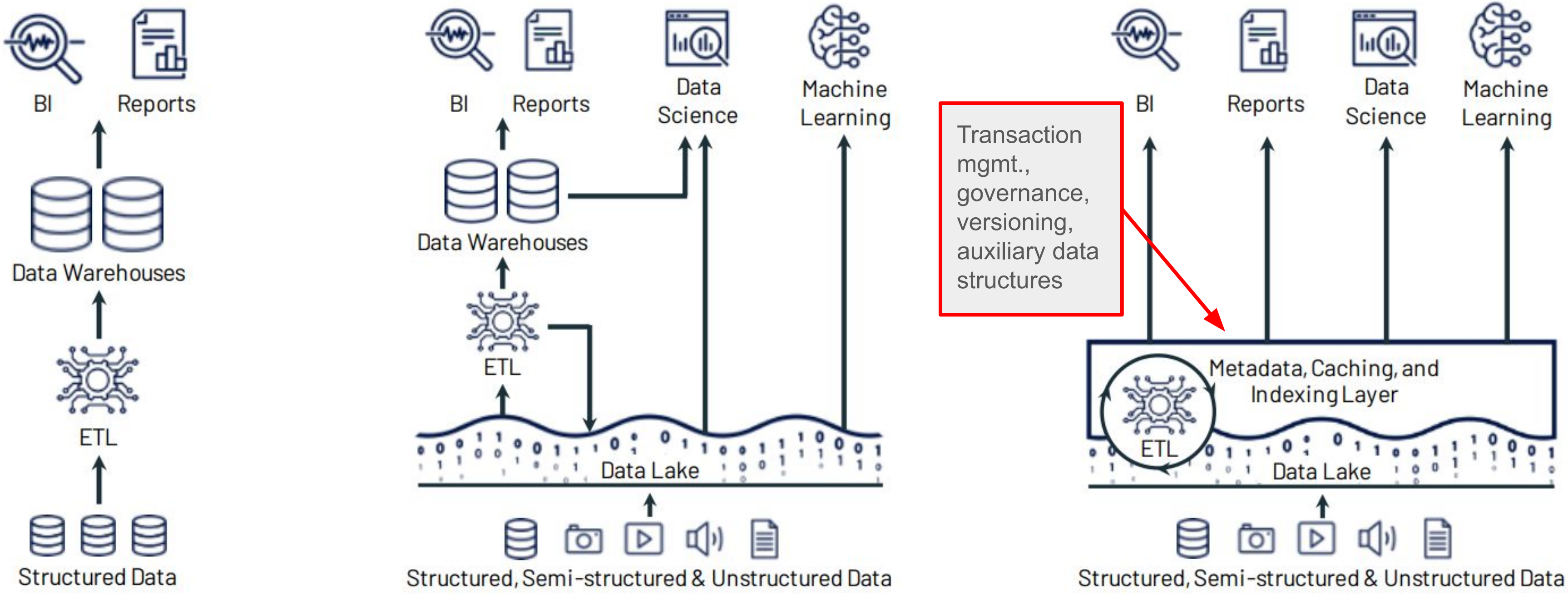
- Hybrid approach
- Data lake + management features
- Motivation
  - To prevent “data swamp” scenarios
  - To provide unified platform for data analytics

*Delta Lake, Apache Iceberg*

Data lake can be seen as an “extended” staging area available directly for data analysis

- History is kept
- Data are not modified during ingestion
- Schema on-read
- Highly scalable
- Stores all data types

# DWH vs Data lake vs Data lakehouse [8]



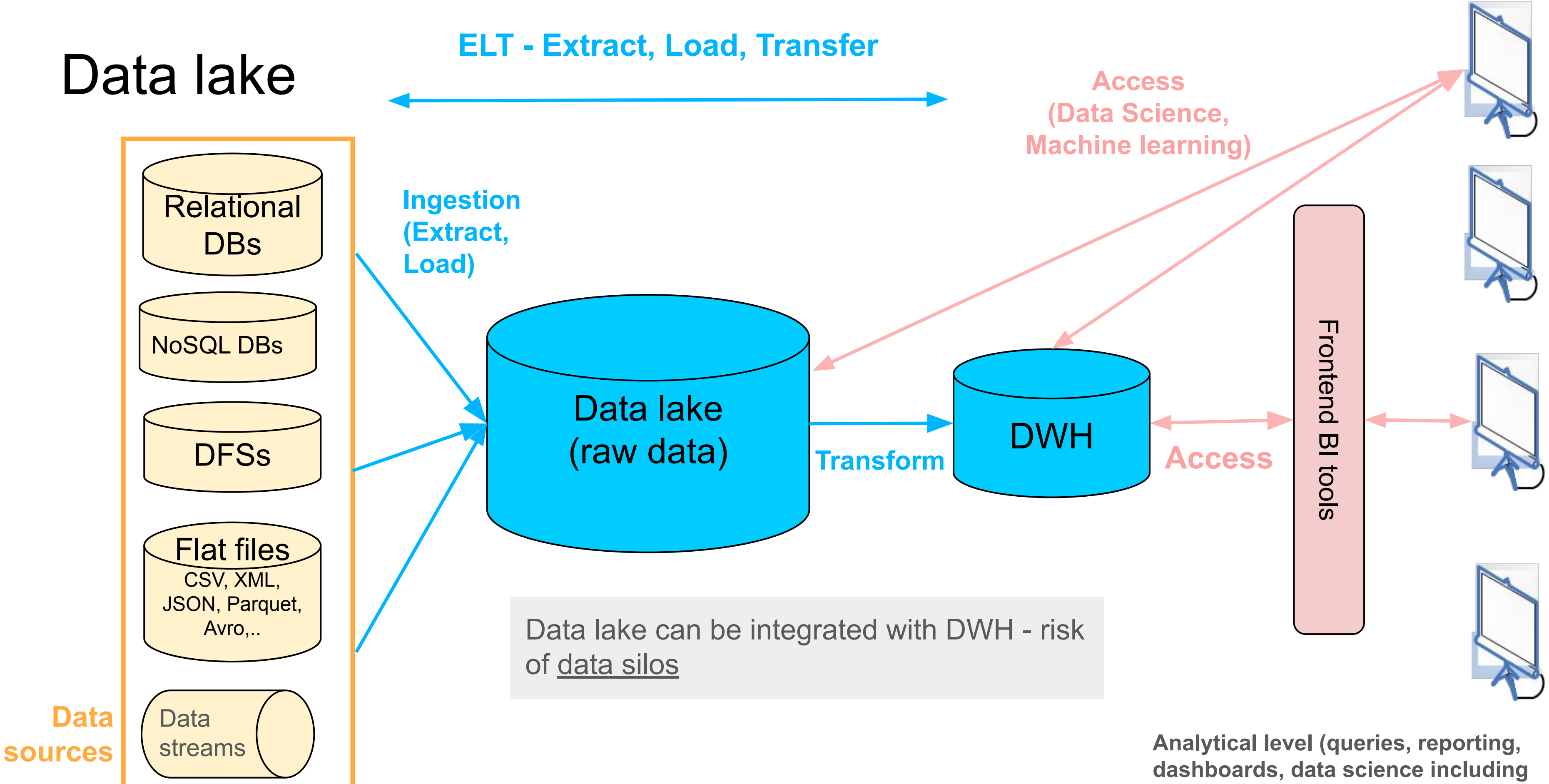
Transaction mgmt., governance, versioning, auxiliary data structures

**(a) First-generation platforms.**

**(b) Current two-tier architectures.**

**(c) Lakehouse platforms.**

# Data lake


ELT - Extract, Load, Transfer
Relational DBs
NoSQL DBs
DFSs
Flat files
CSV, XML, JSON, Parquet, Avro,..
Data streams
Data sources
Ingestion (Extract, Load)
Data lake (raw data)
Transform
DWH
Access (Data Science, Machine learning)
Access
Frontend BI tools

Data lake can be integrated with DWH - risk of data silos

Analytical level (queries, reporting, dashboards, data science including machine learning)

# Data lakehouse “zone architecture”

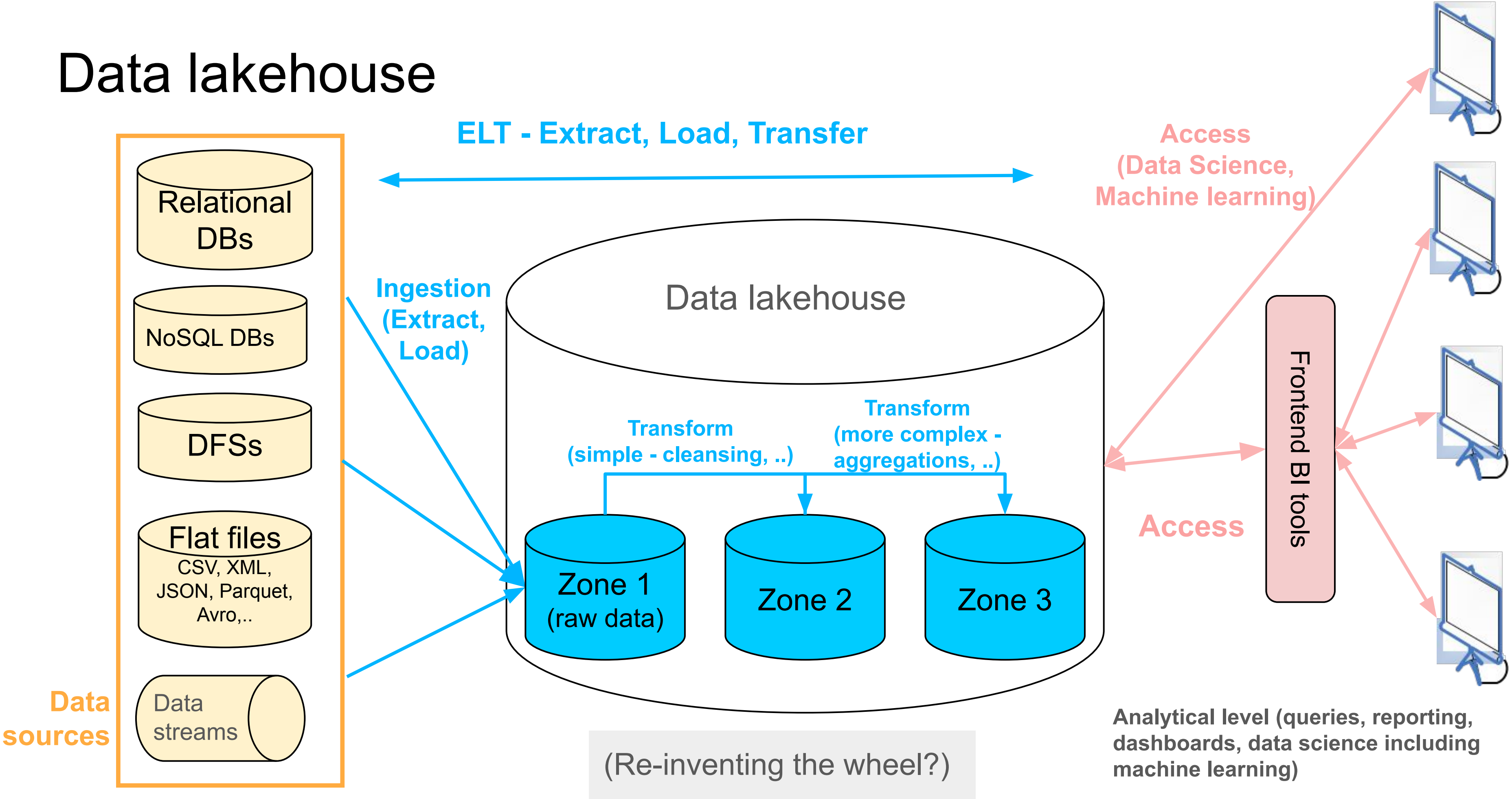
Initial data lake concept: only “raw data zone”

Today many zone models exist, but they share common principle:

- Data are assigned to a zone according to the degree of processing that has been applied to them
- More zones allow for direct access, but typically each of them is used for different use cases (type / purpose of data analytics, batch vs RT / NRT processing)

Example: Databricks / Medallion architecture

# Data lakehouse



ELT - Extract, Load, Transfer

Data sources

Relational DBs

NoSQL DBs

DFSs

Flat files
CSV, XML, JSON, Parquet, Avro,..

Data streams

Ingestion (Extract, Load)

Data lakehouse

Transform (simple - cleansing, ..)

Transform (more complex - aggregations, ..)

Zone 1 (raw data)

Zone 2

Zone 3

Access (Data Science, Machine learning)

Access

Frontend BI tools

Analytical level (queries, reporting, dashboards, data science including machine learning)

(Re-inventing the wheel?)

# Traditional vs modern data warehouses (1)

- Core traditional concepts still in use
  - Dimensional model, 2-layer / 3-layer architectures
- New architectural concepts
  - Data lake, data lakehouse - immature
- A shift in the use of data analysis in business
  - BI + part of products / services
- A significant shift in technologies (also when using traditional architectures)
  - Data storage
    - RDBMSs replaced or complemented by new storage systems
  - Data processing
    - ETL tools able to read from and write to new storage systems
    - ETL replaced or complemented by parallel computation mechanisms (MapReduce, Spark, …)
  - Integration of new data analytics methods

# Traditional vs modern data warehouses (2)

- Batch vs real-time data analytics
  - Mostly separate real-time data pipelines (Lambda architecture)
  - Some data pipelines cannot be accelerated inherently
- More affordable data analytics
  - Cloud solutions
  - More lightweight data analytics tools, embedded data analytics

- Adaptation of traditional DWH ?
  - Reduce pre-computations, migrate to new storage systems & computation frameworks, migrate to cloud, designing RT / NRT pipelines
  - Might get expensive
  - Might get complex for poorly documented DWHs
  - In some cases building new solution from scratch is more efficient

  -> Optimal solution strongly depends on actual requirements and the properties of the current data analytics solution

# References

[1] Inmon, W.H.: Building the Data Warehouse, 4th edition, 2005

[2] Kimball, R.: The Data Warehouse Toolkit, 4th Edition, 2013

[3] Song, IY. (2009). Data Warehouse. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA, 2009. https://doi.org/10.1007/978-0-387-39940-9_882

[4] What is a data lake?, aws.amazon.com

[5] Campbell, Chris. Top Five Differences between DataWarehouses and Data Lakes. Archived from Blue-Granite.com, 2015.

[6] What is a data lakehouse?, cloud.google.com

[7] Data Lakehouse, databricks.com

[8] Michael Armbrust Ali Ghodsi, Reynold Xin, Matei Zaharia: Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics, 2021, CC BY 3.0.

**Credit:**

- Slides 11, 14: Designed by Wannapik
- Slide 30: Example of a database star schema. A central fact table links records in dimension tables using unique identifiers to key related records.by SqlPac (talk), CC BY-SA 3.0
- Slide 31: Example of a Snowflake schema, by SqlPac (talk), CC BY-SA 3.0
- Slide 41: OLAP cube by Margaret Rouse, olap.com, Guru99, CC BY-SA 4.0
- Slide 50: Image by Vectorportal.com, CC BY
- Slide 56: Michael Armbrust Ali Ghodsi, Reynold Xin, Matei Zaharia: Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics, 2021, CC BY 3.0.