

CSC12107 – Information Systems for Business Intelligence

Lesson 2: DATA WAREHOUSE

Giáo viên: Hồ Thị Hoàng Vy
TPHCM, 8-2021



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

Goals

- **After complete this chapter, students can:**
 - Present the basic definitions and concepts of data warehouses
 - Differentiate different types of data warehousing architectures; their comparative advantages and disadvantages
 - Describe the processes used in developing and managing data warehouses

Main topics

- ☐ Case study
- ☐ DW definition
- ☐ Characteristics of DW
- ☐ DW Architecture

Case study

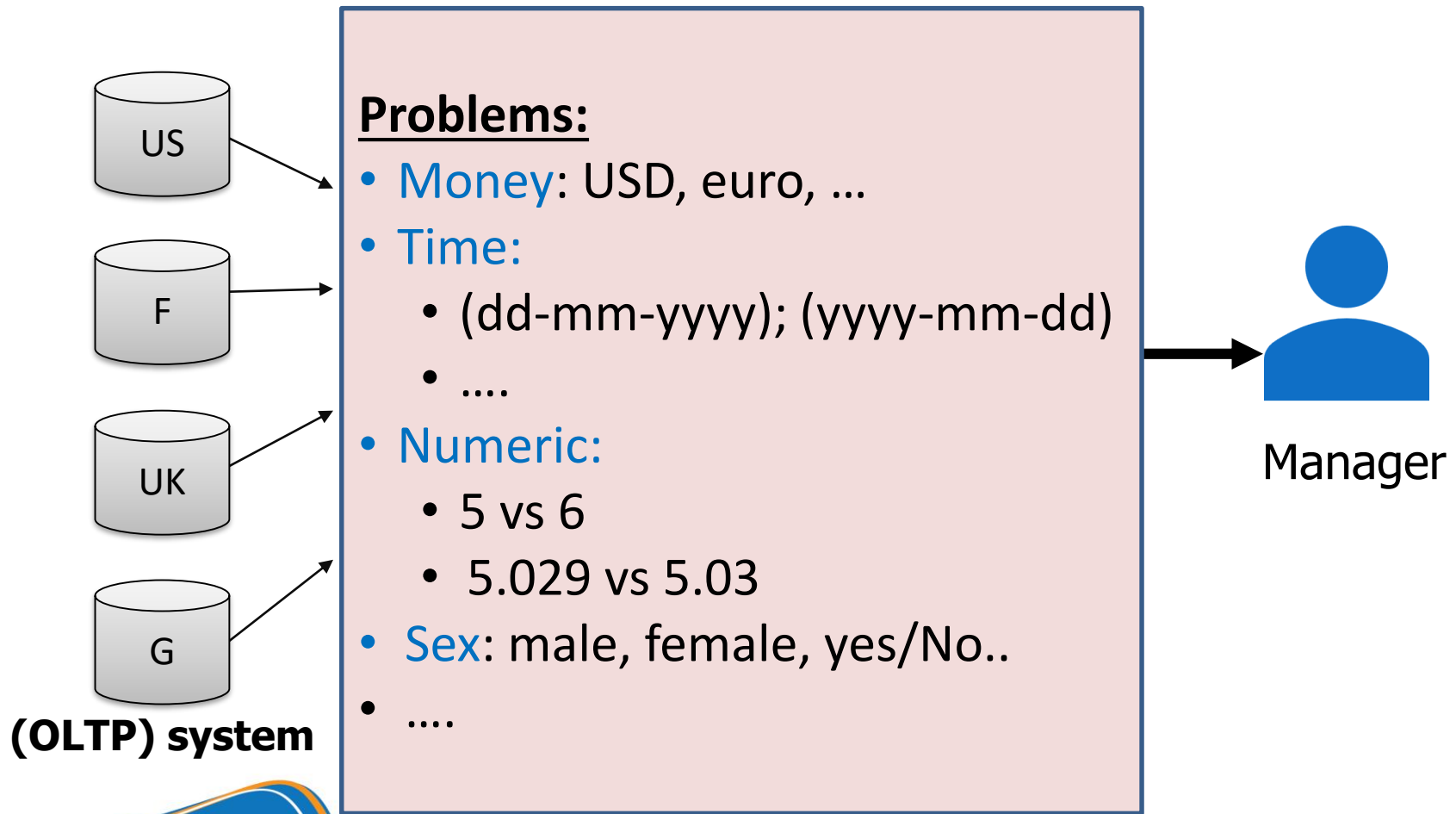


- Amadeus Entertainment is an entertainment retailer specializing in music, films, and audio books. It has eight online stores operating in the United States, Germany, France, the United Kingdom, Spain, Australia, Japan, and India. It has 96 offline stores operating in those countries as well.
- **Requirement:**
 - sale manager wants to make a daily/monthly/quarter report on sale revenue

Problem

- The Data is stored in many database with different platforms:
 - ▣ Sale processing, multimedia trading, sale order processing, ect.. All running in on an Oracle database
 - ▣ Business activities in the offline stores are manage in Java-based system
 - ▣ Products, inventory... are managed in DB2
 - ▣

Problem



Problem

- **Availability**

- Ex: a piece of data is available in one system but is not in the other system

- **Time ranges:**

- Ex: The same piece of data exists in different systems, but they have different time periods.

- **Definition**

- Ex: Sometimes the same data may contain different things

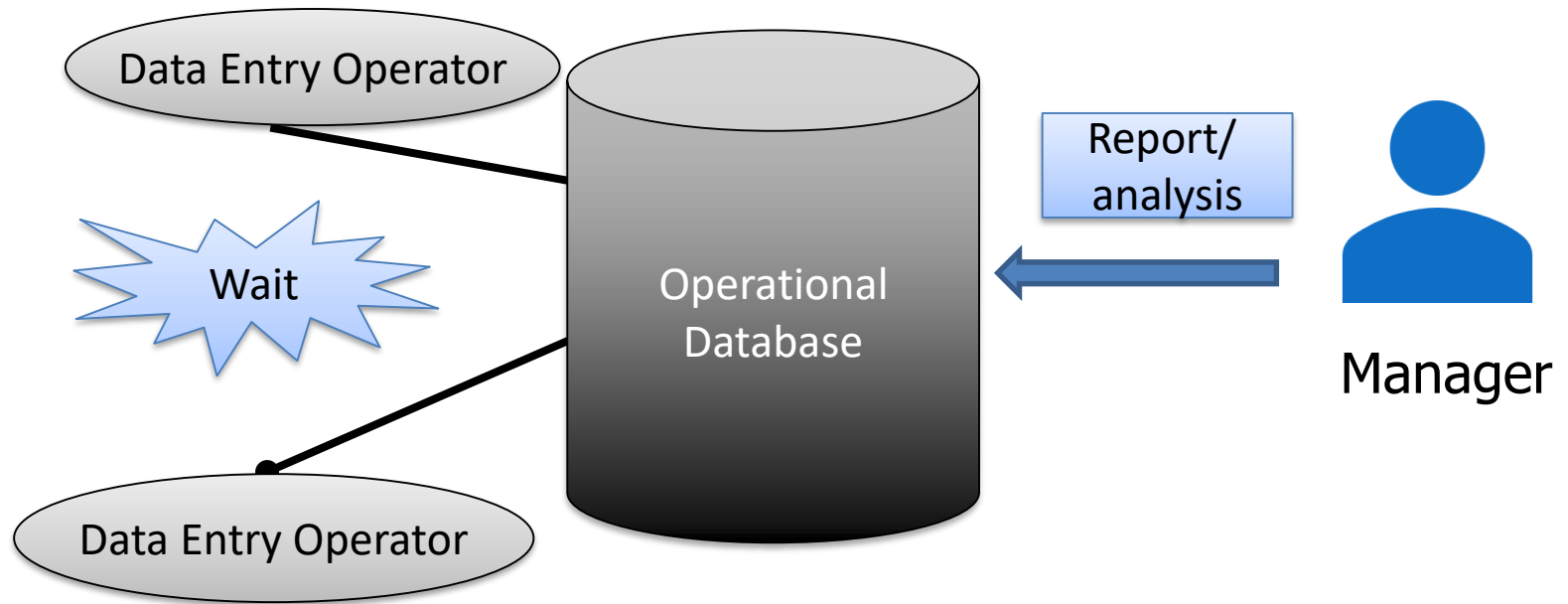
- **Conversion**

- Ex: the data in the source system is in different units of measure

- **Matching**

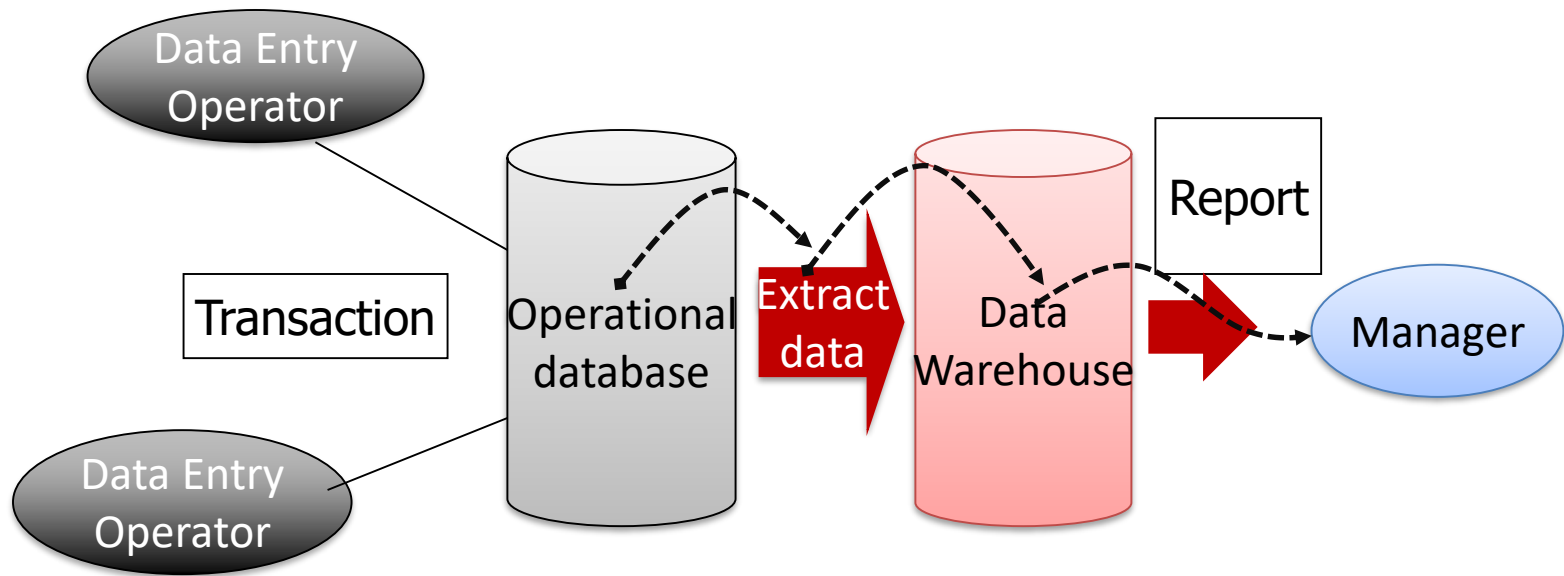
- is a process of determining whether a piece of data in one system is the same as the data in another system

Problem



- Reports and queries are time-consuming. Running directly on the database server can slow down performance
- Main purpose of OLTP system is to capture and store the business transactions

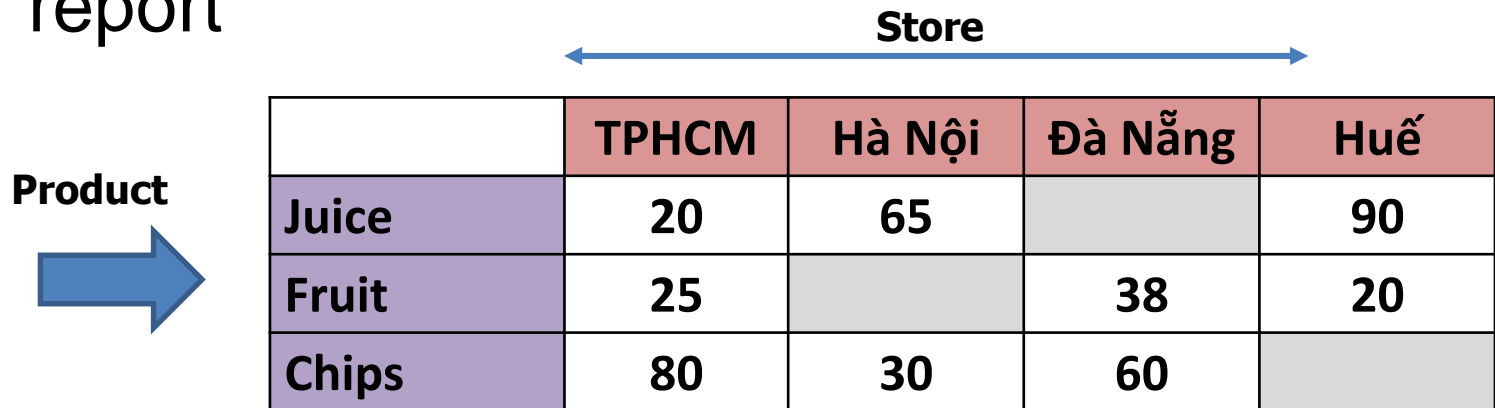
Problem



- Separates analytics processing from transactional databases, improving the performance of both systems
- Retrieving and transforming data from the source system and putting it into the data warehouse
- Update your data warehouse on a regular basis to ensure that the information derived from it is current

Problem

- Report format requirements are fairly simple and static, not flexible or interactive.
- If users want to swap a piece of data with another piece or want to view the data at a higher or lower level, we need to redesign the report



	Store			
	TPHCM	Hà Nội	Đà Nẵng	Huế
Juice	20	65		90
Fruit	25		38	20
Chips	80	30	60	

Problems

Location →

→ **Time**

Day 3
Day 2
Day 1

↑ **Products**

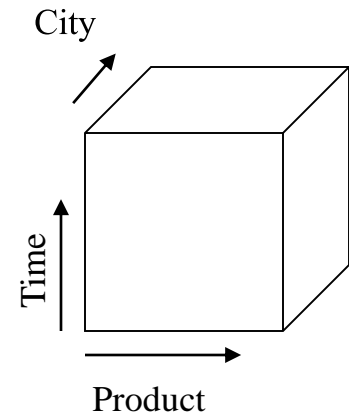
	TPHCM	Hà Nội	Đà Nẵng	Huế
Drinks	20	65		90
Fruits	25		38	20
snack	80	30	60	
snack	23	30	60	

- HCMC → on January 2, selling snack products: VND 53,000
- HCMC → what is best/least selling time?
- HCMC → snack consumption trend increased gradually over a period of time

Problems

- Indicate the quantity of goods in each city by month

		M1	M2	M3	M4
HCM	Bread			3	10
	Cheese	3	16	6	
	Milk	4	16	6	
Hà Nội	Bread			3	7
	Cheese	3			8
	Milk	4	9	15	



Main topics

- ☐ Case study
- ☐ **DW definition**
- ☐ Characteristics of DW
- ☐ DW Architecture

DW – Definition⁽¹⁾

- Defined in many different ways, but not rigorously
- **A data warehouse is a system:**
 - ❧ *retrieves data*
 - ❧ *consolidates data*

*periodically from the source systems into a **dimensional** or **normalized** data store*

❧ *It usually keeps years of **history** and is queried for business intelligence or other analytical activities. It is typically **updated in batches**, not every time a transaction happens in the source system*

(Vincent Rainardi - Building a Data Warehouse: With Examples in SQL Server)

- **Data warehousing:**
 - ❧ The process of constructing and using data warehouses

DW – Definition^(2,3)

(2) - “a data warehouse is a system that **extracts**, **cleans**, **conforms**, and **delivers** source data into a dimensional data store and then supports and implements querying and analysis for the purpose of decision making”

(Ralph Kimball)

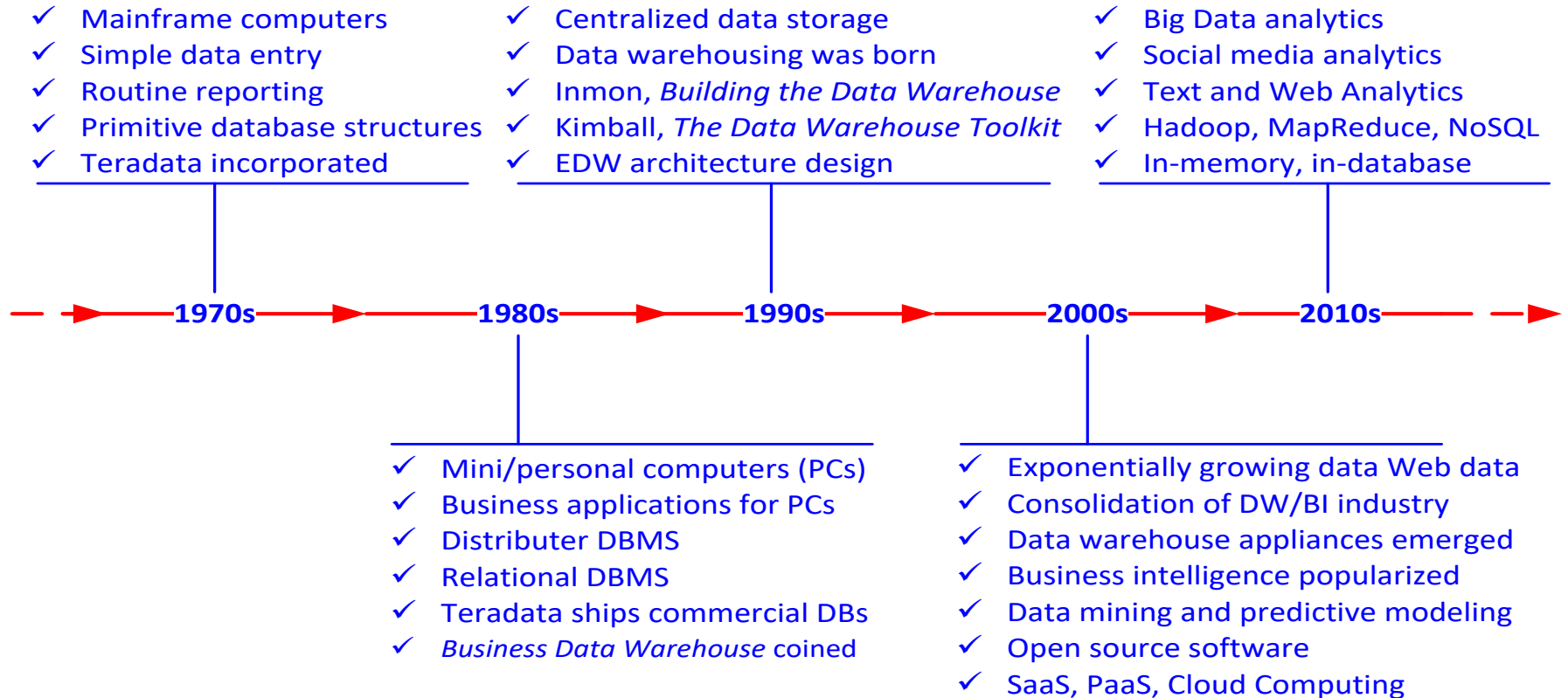
Building the Data Warehouse, Fourth Edition(John Wiley, 2005)

(3) - “a DW is a **subject -oriented** , **integrated** , **time-variant**, and **nonvolatile** collection of data in support of management’s decision making process”

(W.H. Inmon)

The Data Warehouse ETL Toolkit(John Wiley,2004)

DW - A Historical Perspective



Main topics

- ☐ Case study
- ☐ DW definition
- ☐ **Characteristics of DW**
- ☐ DW Architecture

Data warehouse - characteristics

□ Definition⁽¹⁾

- **Retrieves data** - The data retrieval is performed by a set of routines widely known as an ETL system
- **Consolidates Data**
 - Data availability
 - Time ranges
 - Matching
- **History**
 - Most transactional systems store some history, but data warehouse systems store very long history
- **Periodically**
 - The data retrieval and the consolidation do not happen only once; they happen many times and usually at regular intervals, such as daily or a few times a day
- **Updated in Batches**
 - users are not able to update or delete data in the data warehouse
 - DW data is updated using a standard mechanism called ETL at certain times

Data warehouse - characteristics

□ **Entity-oriented**⁽³⁾

- The data contained in a data warehouse are primarily concerned with the main entities of interest for the analysis

□ **Integrated**⁽³⁾

- The data originating from the different sources are integrated and **homogenized** as they are loaded into a data warehouse

□ **Time-variant**⁽³⁾

- Time-variant. All data entered in a data warehouse **are labeled with the time period to which they refer**

□ **Nonvolatile**⁽³⁾:

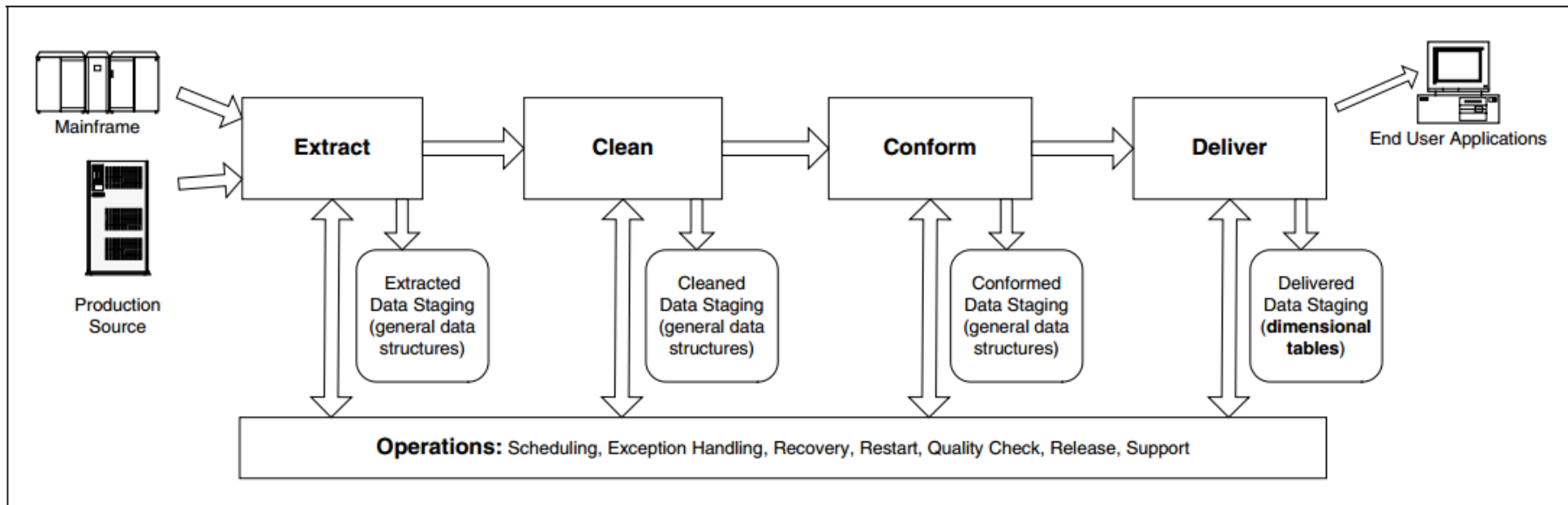
- After data are entered into a data warehouse, users cannot change or update the data

Data warehouse - characteristics

- ☐ Web based, relational/multi-dimensional
- ☐ Client/server, real-time/right-time/active...

Data warehouse - characteristics

□ Definition⁽²⁾



Main topics

- ☐ Case study
- ☐ DW definition
- ☐ Characteristics of DW
- ☐ **DW Architecture**

Data warehouse - architecture

□ Data architecture

- the higher-level view of how the enterprise handles its data, such as how it is categorized, integrated, and stored
- Provides guidelines for managing data from initial capture in source systems to information consumption by business people

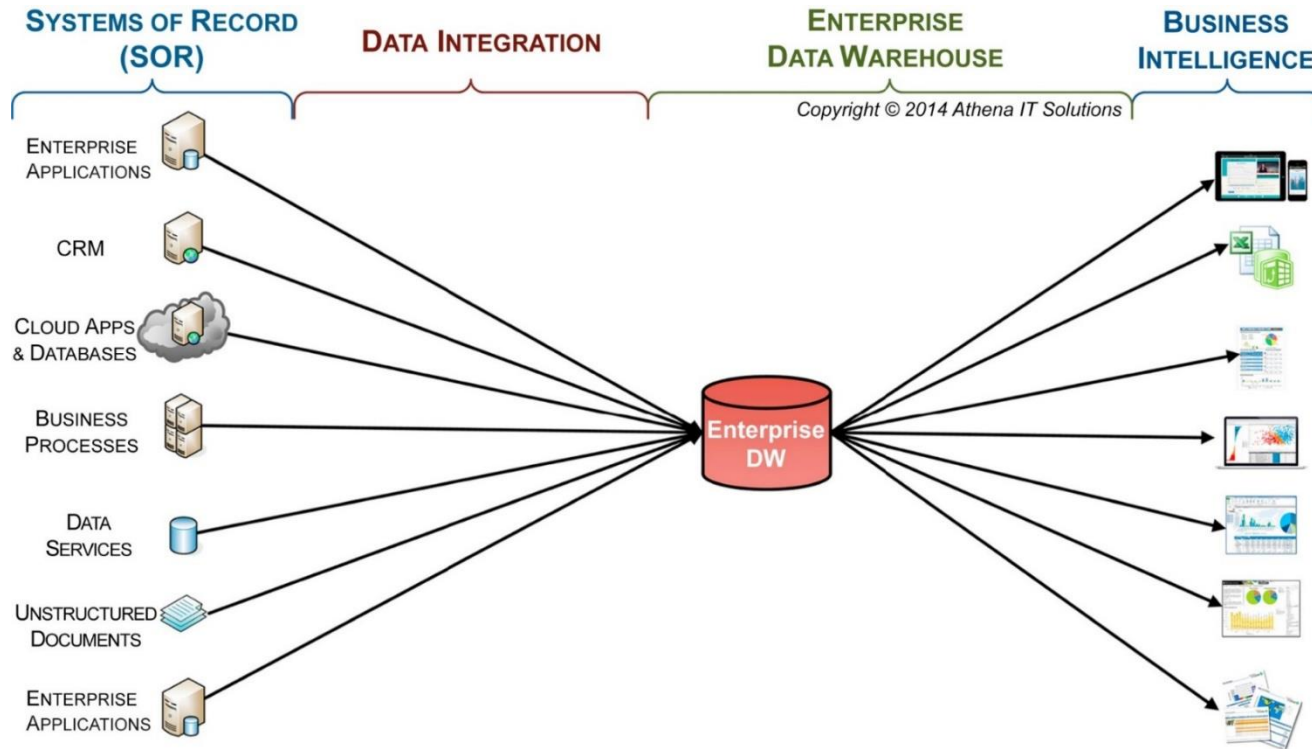
□ Data flow architecture

- is about how the data stores are arranged within a data warehouse and how the data flows from the source systems to the users through these data stores
- is about how the data flows are controlled, logged, and monitored, as well as the mechanism to ensure the quality of the data in the data stores

□ System architecture

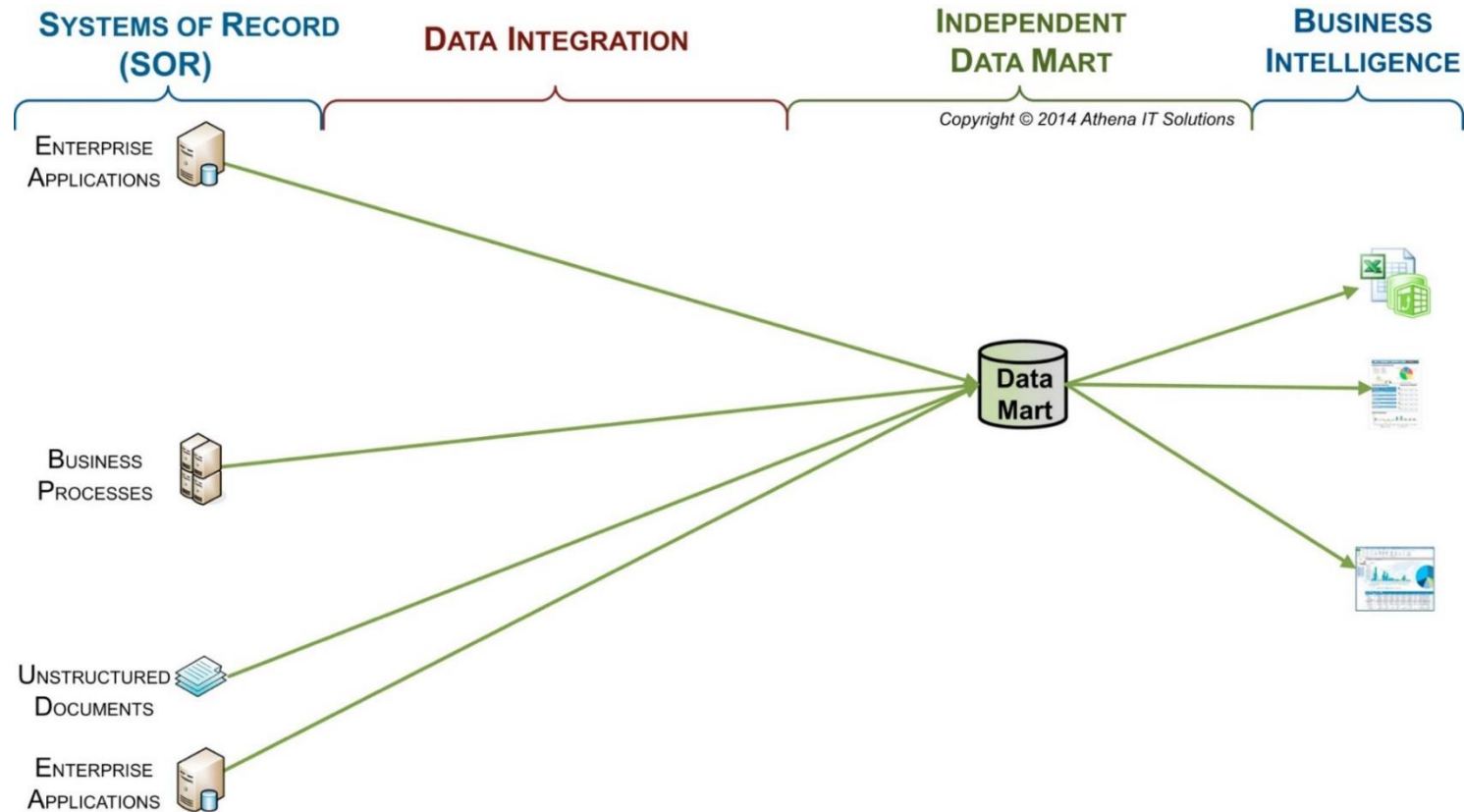
- is about the physical configuration of the servers, network, software, storage, and clients

Enterprise data warehouse (EDW)



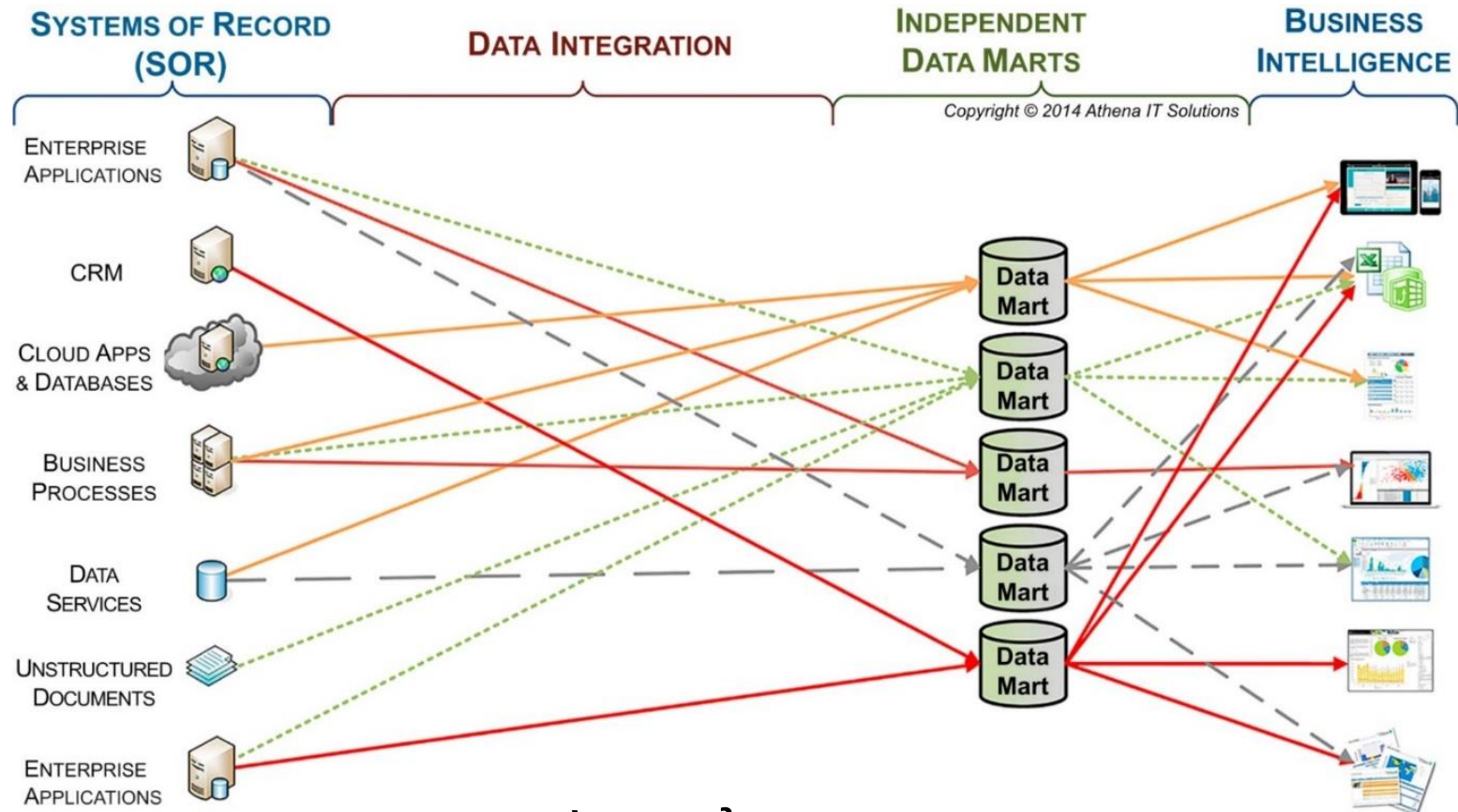
An EDW (Enterprise Data Warehouse) is an all-encompassing DW that covers all subject areas of interest to the entire organization.

Data mart



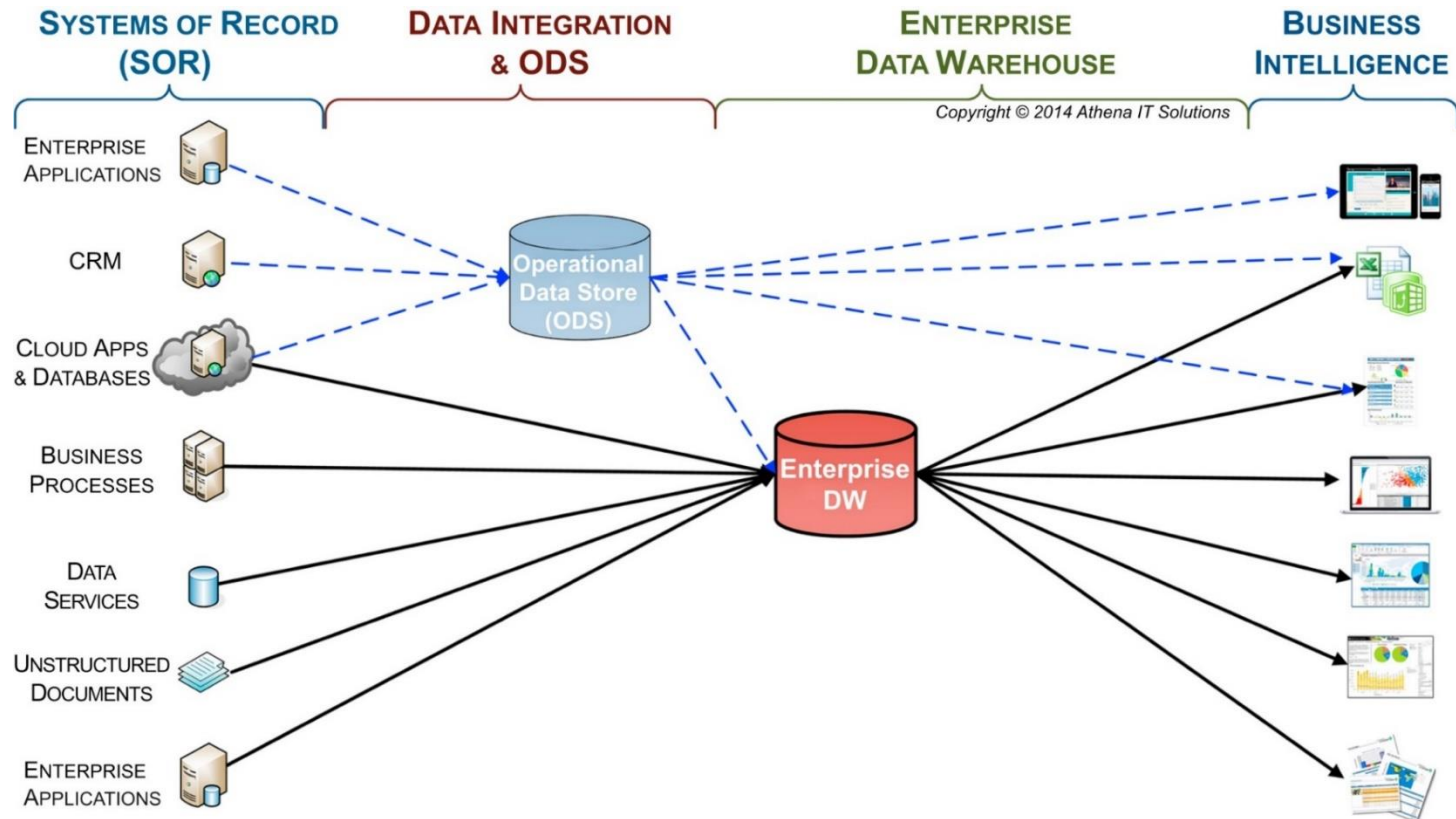
A data mart is a smaller DW designed around one problem, organizational function, topic, or other suitable focus area

Multiple independent data marts



- Ưu điểm?
- Khuyết điểm?

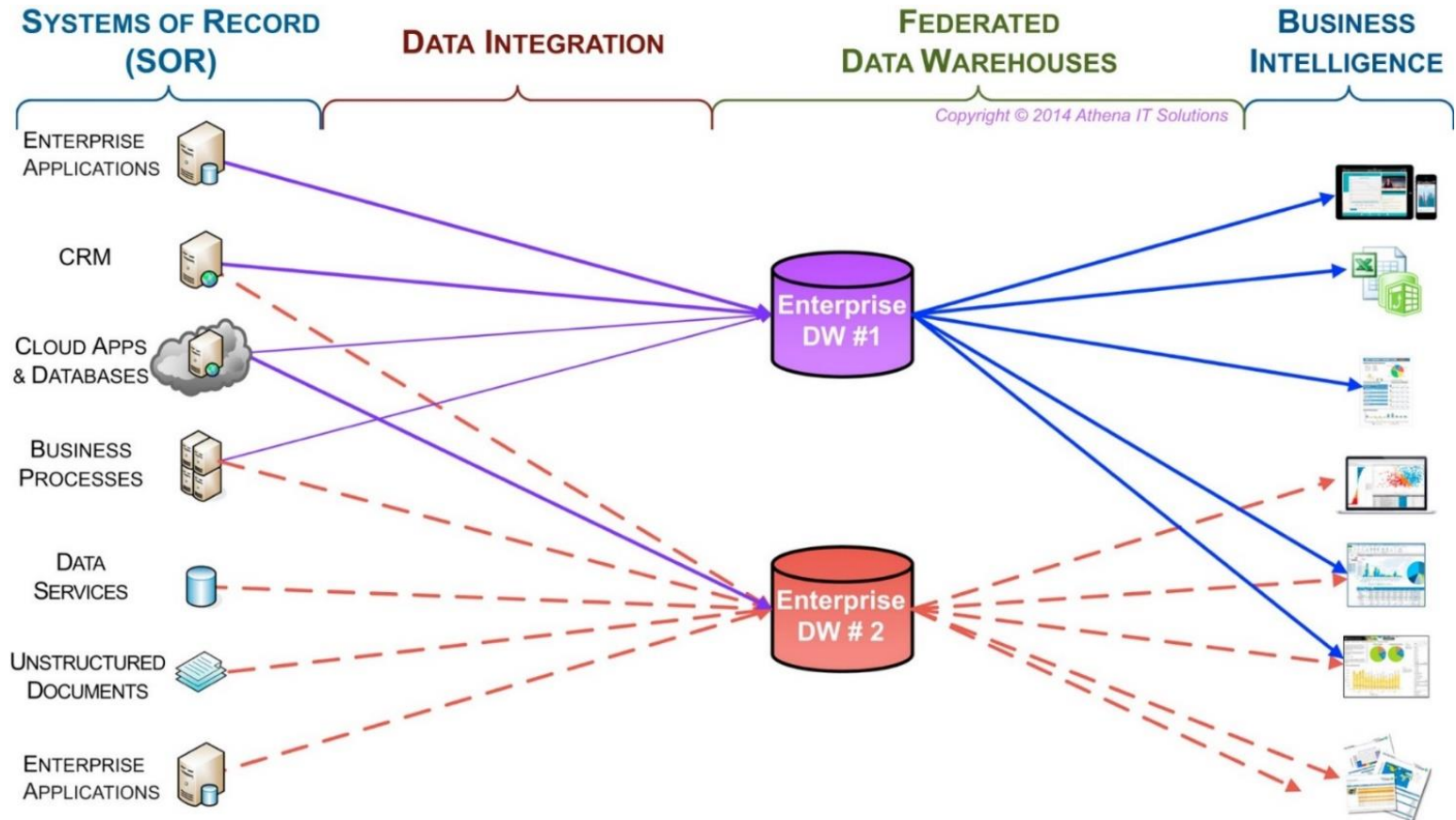
Operational data store



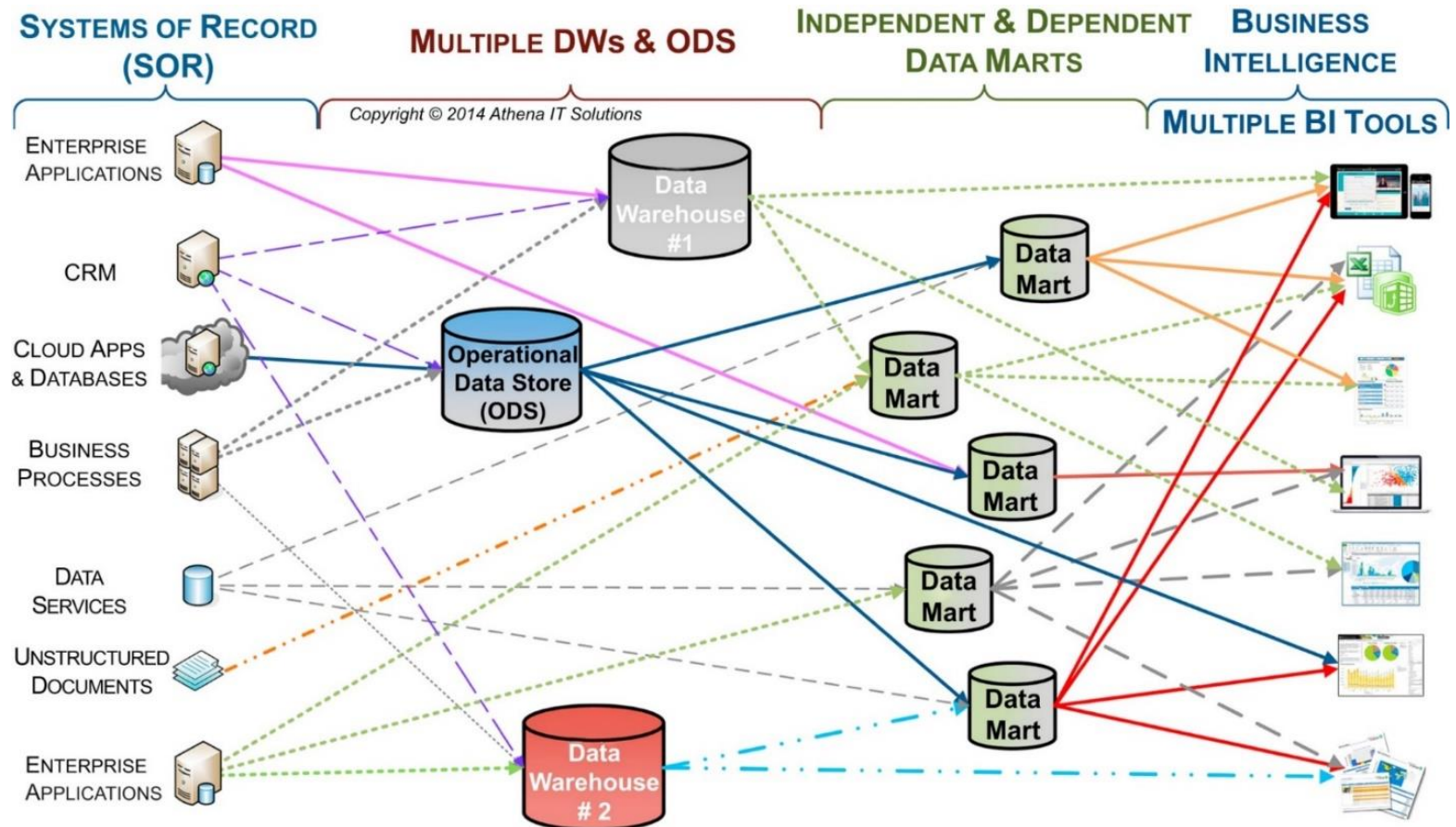
Enterprises with both an EDW and ODS could get their reports from:

- ODS only
- EDW only
- ODS and EDW

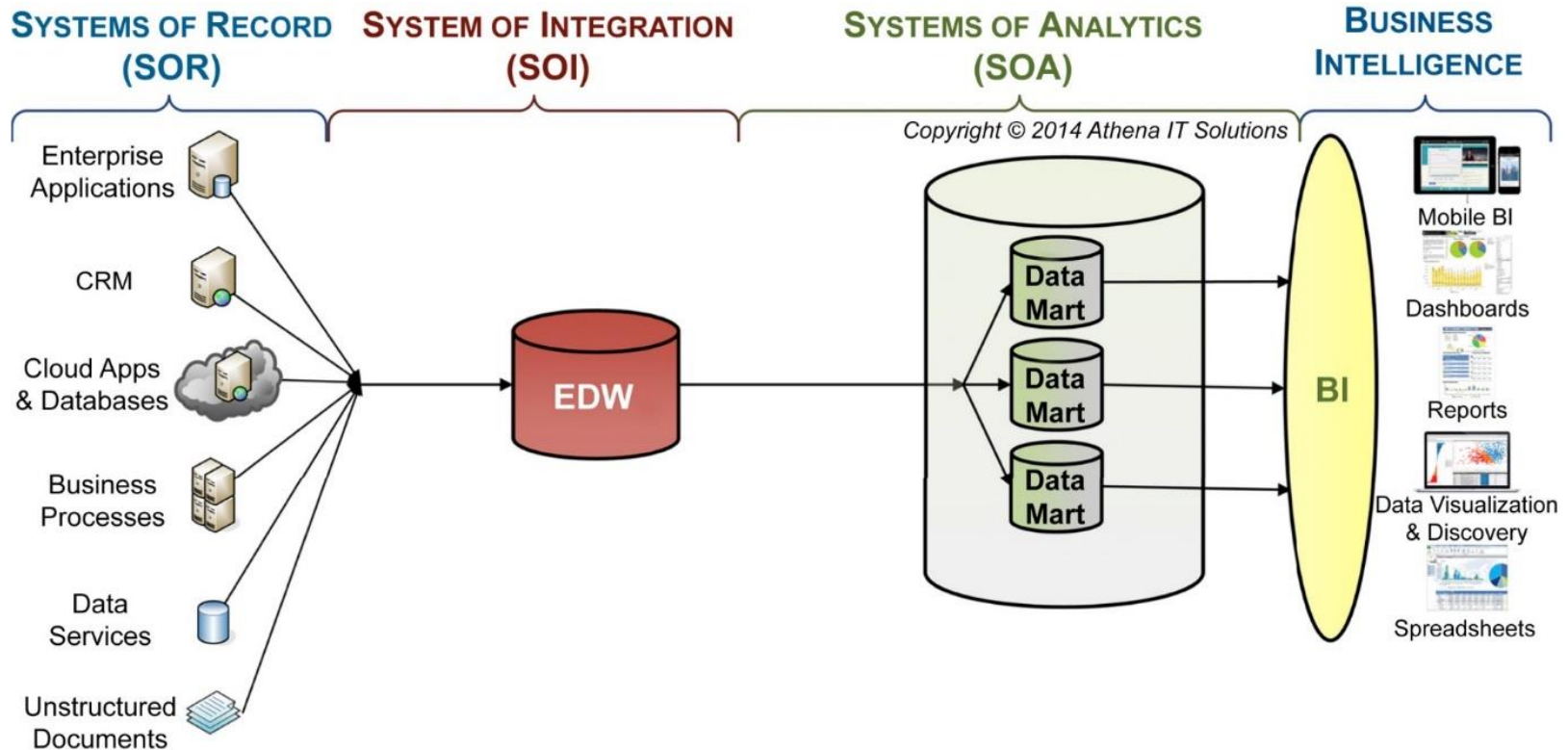
Federated data warehouses



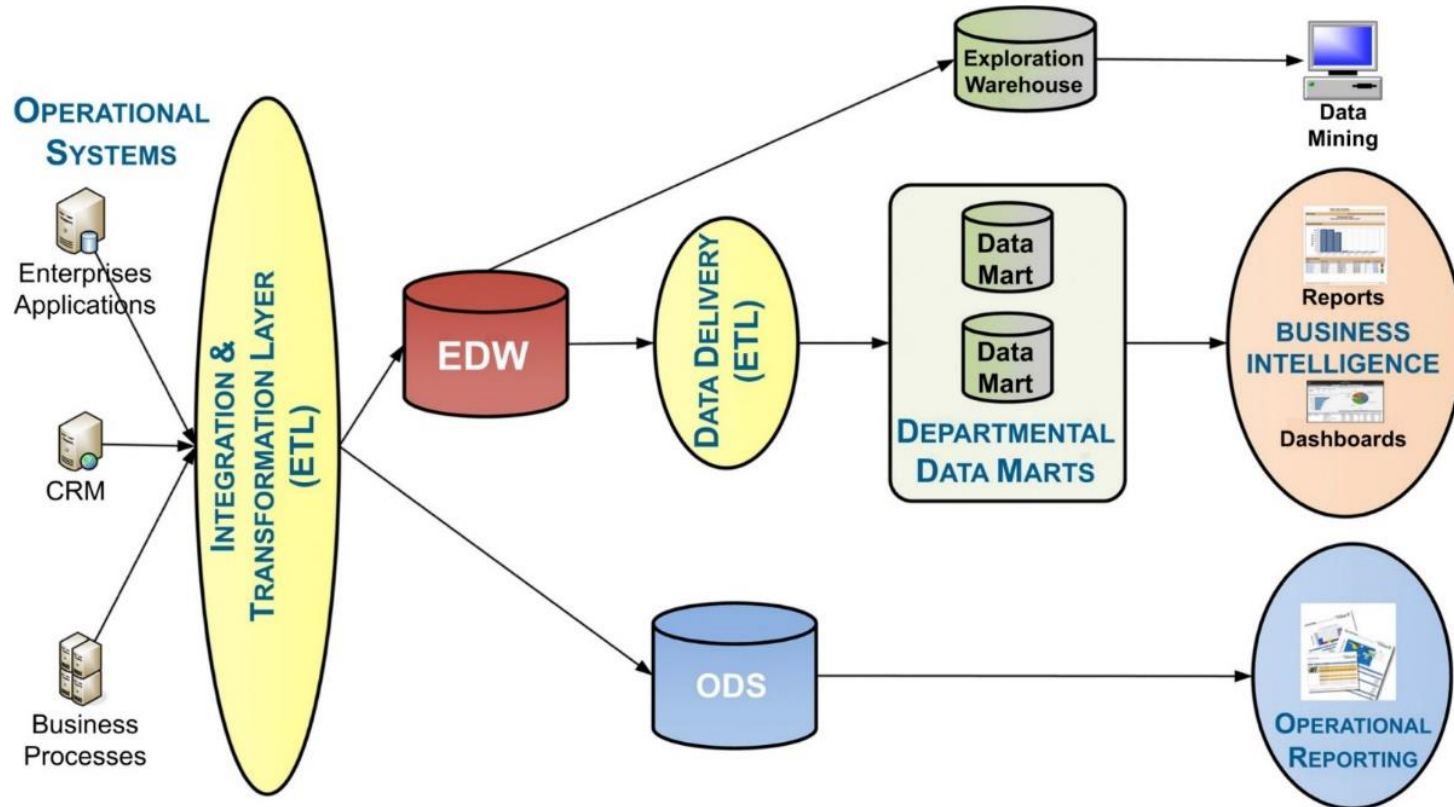
Multiple built BI silos & multiple BI tools



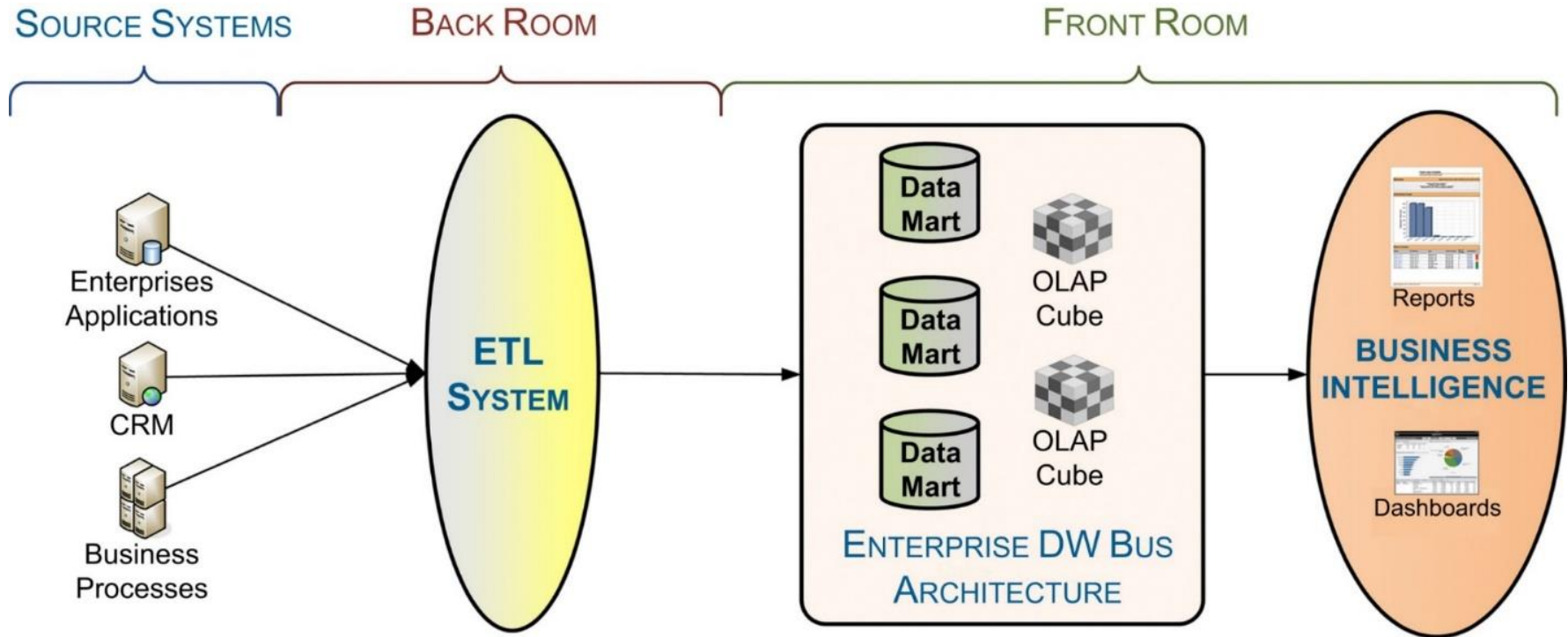
DI workflow – hub & spoke



Inmon's CIF architecture



Kimball's enterprise data bus architecture



Discussion

- ☐ Comparison of Inmon vs Kimball Architecture
- ☐ Which model is best?

Data warehouse - architecture

□ Data architecture

- the higher-level view of how the enterprise handles its data, such as how it is categorized, integrated, and stored
- Provides guidelines for managing data from initial capture in source systems to information consumption by business people

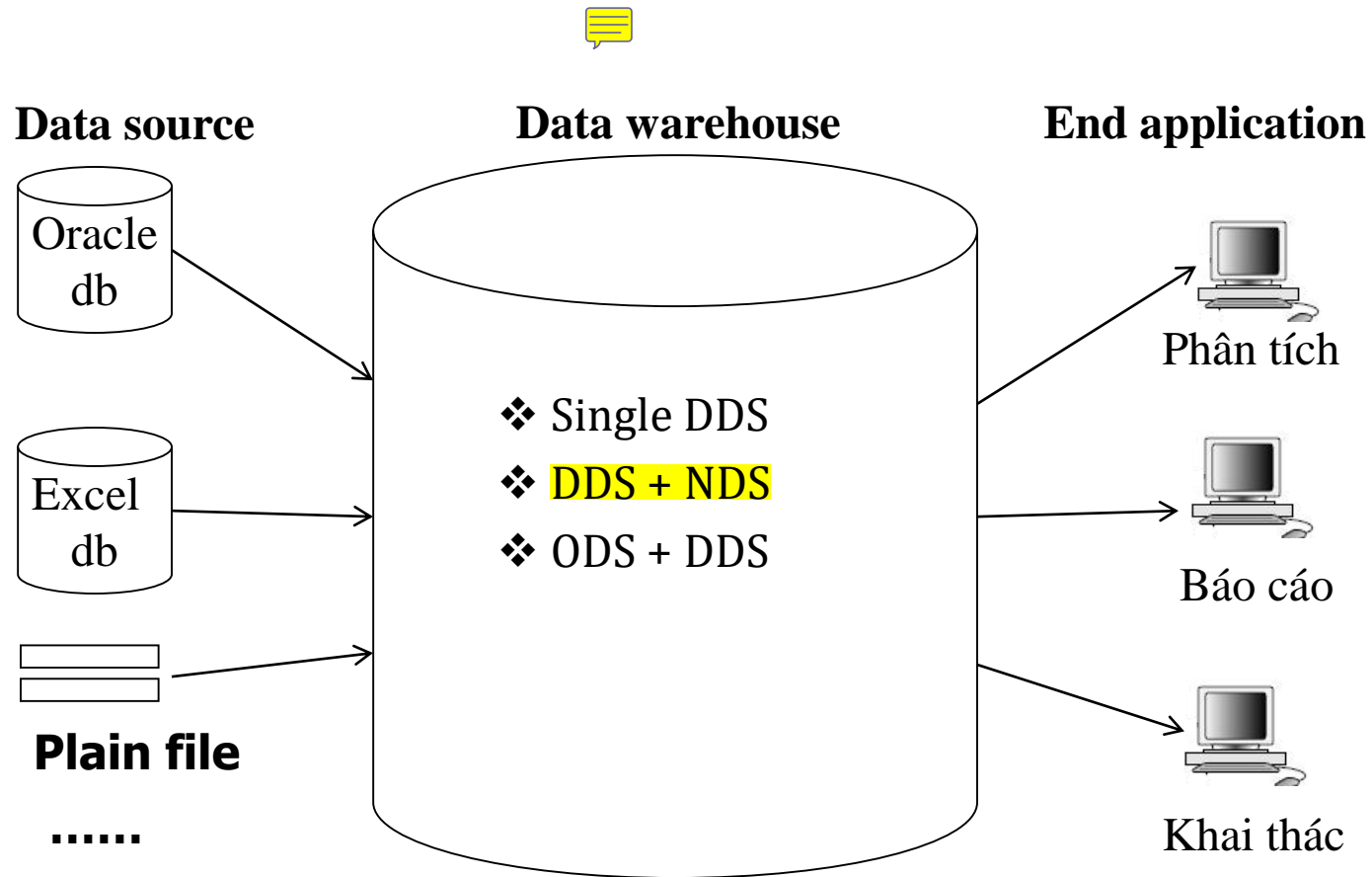
□ Data flow architecture

- is about how the data stores are arranged within a data warehouse and how the data flows from the source systems to the users through these data stores
- is about how the data flows are controlled, logged, and monitored, as well as the mechanism to ensure the quality of the data in the data stores

□ System architecture

- is about the physical configuration of the servers, network, software, storage, and clients

Data flow architecture



Data flow architecture

- **Data stores are important components of data flow architecture**
- **Data store**
 - is one or more databases or files containing data warehouse data, arranged in a particular format, involved in data warehouse processes
- **Data store classification:**
 - Based on the **user accessibility**
 - Based on the **data format**

Data store classification

□ Based on the user accessibility

□ A user-facing data store

- is a data store that is available to end users and is queried by the end users and end-user applications

□ An internal data store

- is a data store that is used internally by data warehouse components for the purpose of integrating, cleansing, logging, and preparing data, and it is not open for query by the end users and end-user applications

□ A hybrid data store

- is used for both internal data warehouse mechanisms and for query by the end users and end-user applications

Data store classification

□ Based on the data format

□ Stage

- is an internal data store used for transforming and preparing the data obtained from the source systems, before the data is loaded to other data stores in a data warehouse

□ Normalized data store(NDS)

- is an internal master data store in the form of one or more normalized relational databases for the purpose of integrating data from various source systems captured in a stage, before the data is loaded to a user-facing data store

□ operational data store(ODS)

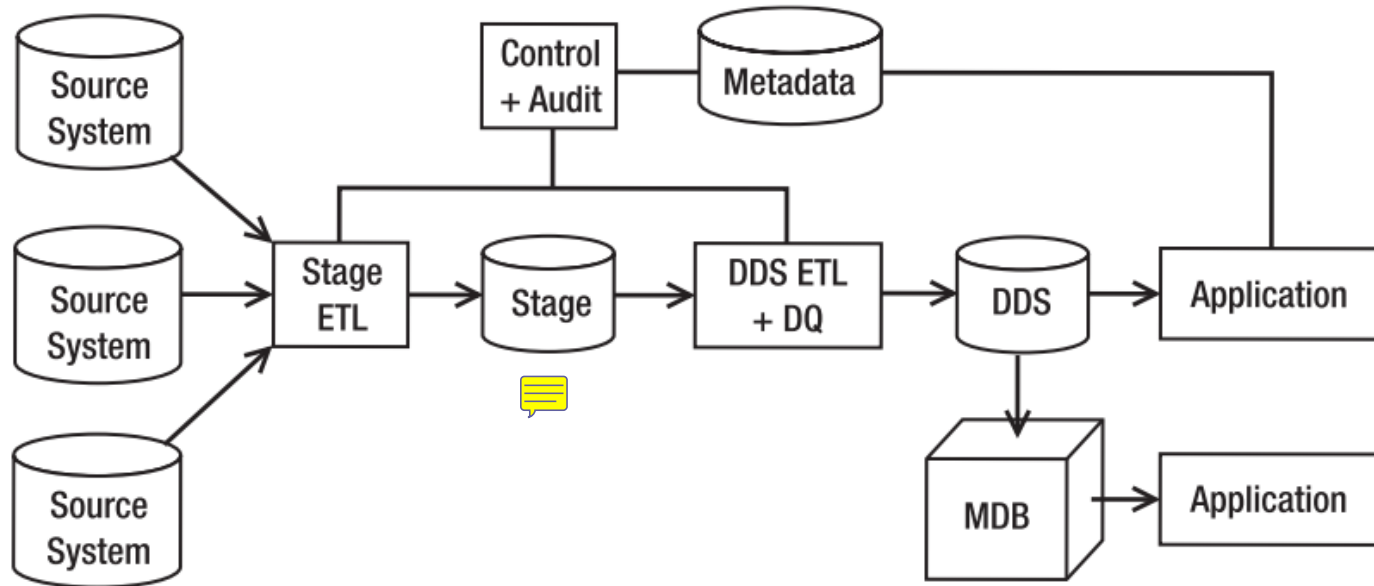
- is a hybrid data store in the form of one or more normalized relational databases, containing the transaction data and the most recent version of master data, for the purpose of supporting operational applications

□ dimensional data store(DDS)

- is a user-facing data store, in the form of one or more relational databases, where the data is arranged in dimensional format for the purpose of supporting analytical queries.

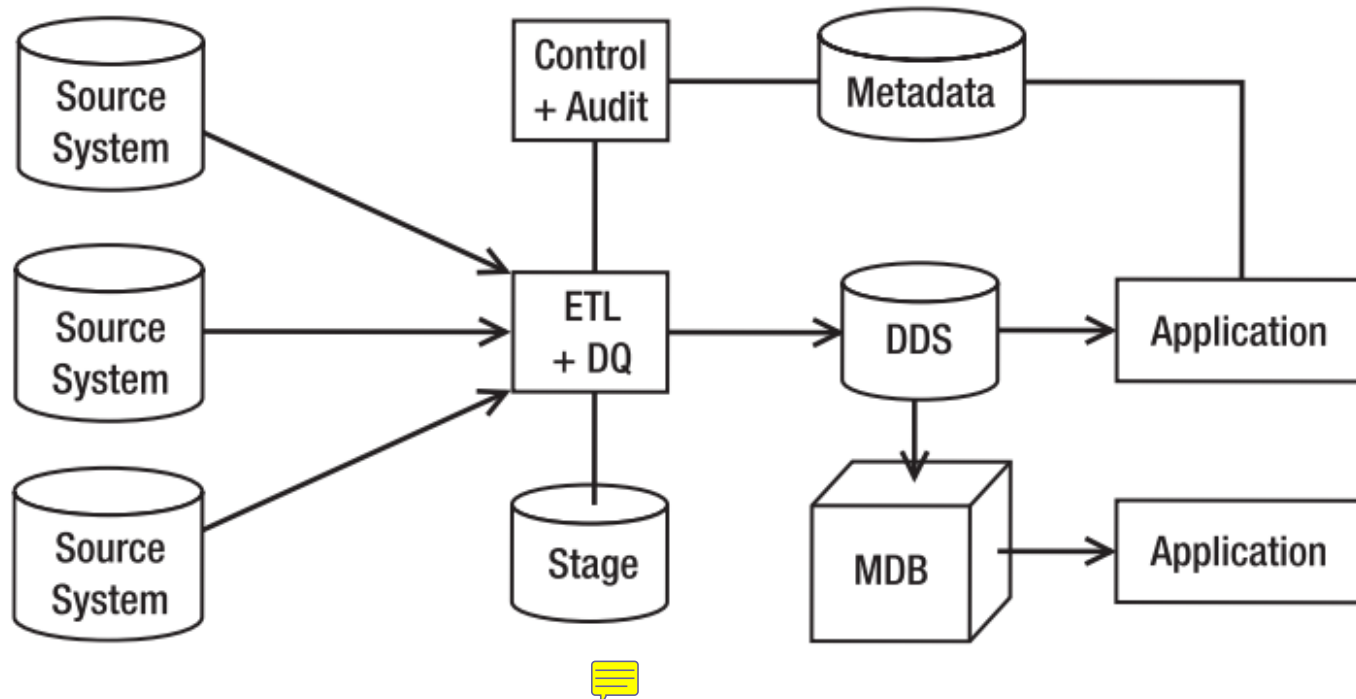


Single DDS



- ☐ **Single DDS** (dimensional data store) architecture has **stage** and **DDS** data stores
- ☐ **Stage**: is a place where you store the data you extracted from the store system temporarily, before processing it further.
- ☐ **Control + audit**: manage the ETL processes and log the ETL execution results
- ☐ **Meta data**: contains the definitions of data, related system, audit information...
- ☐ **DQ database**: **stores the bad data detected by the data firewall**, inform the people responsible for data quality (DQ)

Single DDS (cont)

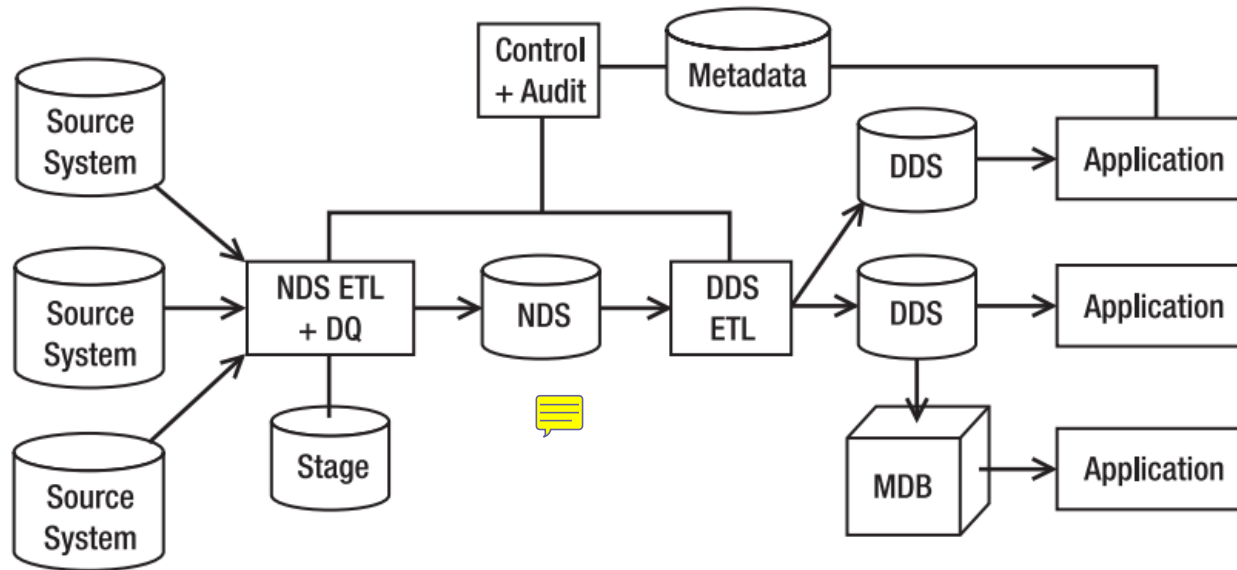


- the stage ETL, the DDS ETL, and the data quality processes are combined into one ETL

Single DDS

- The DDS in the single DDS architecture is the master data store. It contains a complete set of data in a data warehouse including all versions and all historical data.
- **Advantage:**
 - it is simpler than the next three architectures (NDS+DDS, ODS+DDS...)
 - the data from the stage is loaded straight into the dimensional data store, without going to any kind of normalized store first
- **Disadvantage:**
 - it is more difficult to create a second DDS.

NDS + DDS



- **The NDS** (normalized data store) + DDS architecture has stage, NDS, and DDS data stores
- The NDS is in third normal relational form or higher.

NDS + DDS (cont)

- NDS is the **master data store**, meaning NDS contains the complete data sets, including all historical transaction data and all historical versions of master data
- NDS is an **internal data store**, meaning it is not accessible by the end user or the end-user applications.
- There are two types of tables in the NDS:
 - ▣ **Transaction tables** (Eg: Sale order table)
 - contains a business transaction or business event
 - are the source of data for the DDS fact table (chapter 3)
 - ▣ **Master tables** (Eg: product table)
 - contains the people or objects involved in the business event
 - are the source of data for DDS dimension tables (chapter 3)

NDS+DDS (cont)

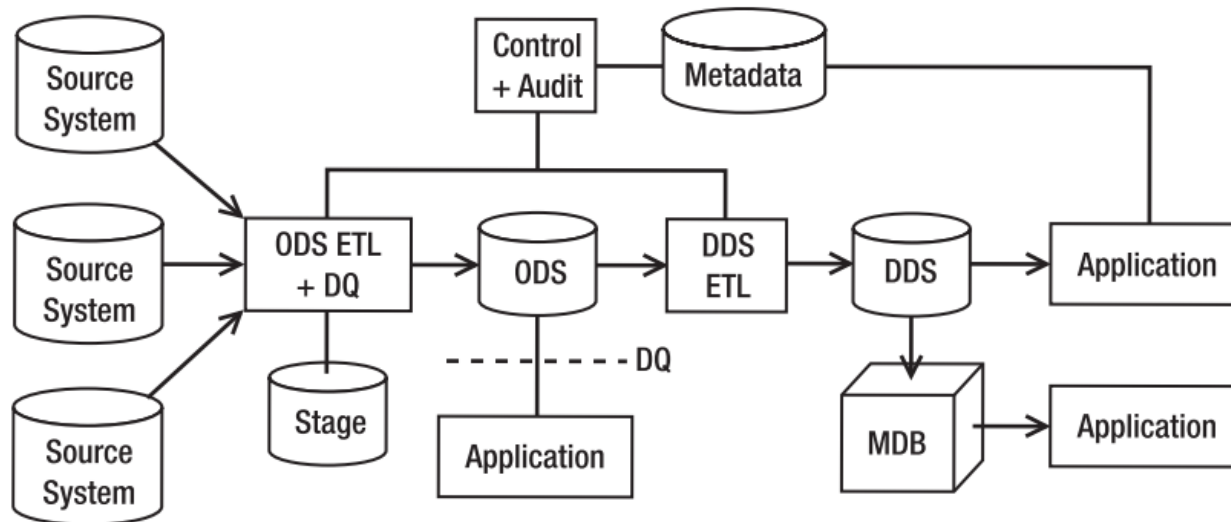
□ Advantage:

- can easily rebuild the main DDS
- can easily build a new, smaller DDS
- it is easier to maintain master data in a normalized store like the NDS and publish it from there because it contains little or no data redundancy
- flexibility for creating and maintaining data stores

□ Disadvantage:

- the data from the stage needs to be put into the NDS first before it is uploaded into the DDS → requires more effort compared to the single DDS architecture
- Need to build two ETL sets,
- Need to design three data stores

ODS + DDS



- **The ODS** (operational data store) + DDS architecture has stage, ODS, and DDS data stores

ODS + DDS (cont)

- ODS is in third normal form or higher
- Like NDS, ODS contains transaction tables and master table
- ODS's master tables contain only the current version of master data
- In an ODS + DDS architecture you have only one DDS
- ODS is a hybrid data store → is accessible by the end users and end-user application → ODS is updatable
- Eg: ODS is used to support a CRM customer support application → status and comments can be written on ODS directly, but all the customer data is still from the source system

ODS + DDS (cont)

□ Advantage

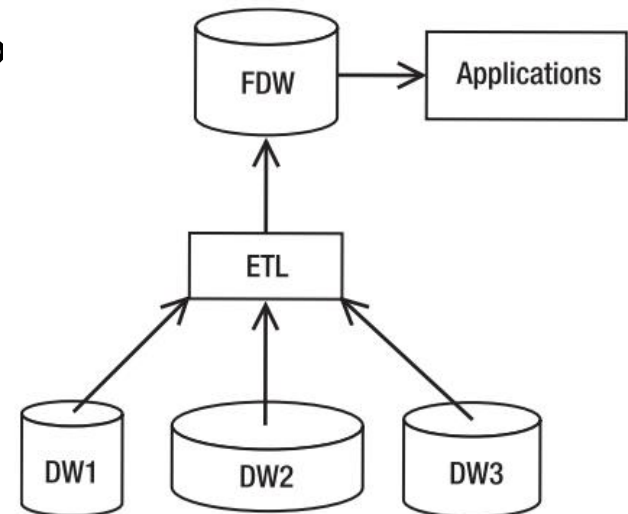
- it contains only current values → the performance of both ODS ETL and DDS ETL better than the ones in the NDS + DDS architecture
- is updatable by the end-user application

□ Disadvantage

- to build a new, small DDS you need to get it from the main DDS and cannot utilize the existing DDS ETL to do that

Fedarate data warehouse

- A federated data warehouse retrieves data from existing data warehouses using an ETL and loads the data into a new dimensional data store.
- The data in the source data warehouses may not arrive at the same frequency
- The FDW ETL needs to match the frequency of the source DWs
- ➔ The granularity of the FDW data?



FEDERATED DWS

- Federated DWs split the EDW into multiple physical DWs, using some of the following options to make the divisions:
 - ▣ Geographical regions or countries
 - ▣ Business functions such as finance, sales, marketing, or HR
 - ▣ Business entities such as divisions or subsidiaries

Data warehouse - architecture

□ Data flow architecture

- is about how the data stores are arranged within a data warehouse and how the data flows from the source systems to the users through these data stores
- is about how the data flows are controlled, logged, and monitored, as well as the mechanism to ensure the quality of the data in the data stores

□ System architecture

- is about the physical configuration of the servers, network, software, storage, and clients

DW - System architecture

□ Three-tier architecture

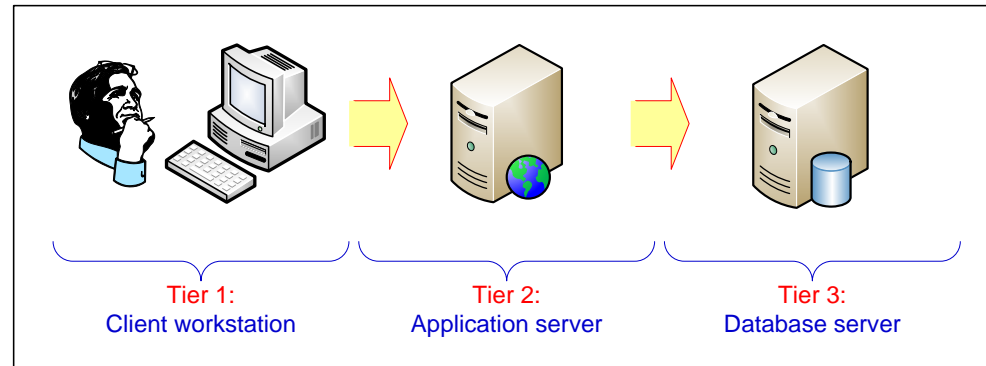
1. Data acquisition software (back-end)
2. The data warehouse that contains the data & software
3. Client (front-end) software that allows users to access and analyze data from the warehouse

□ Two-tier architecture

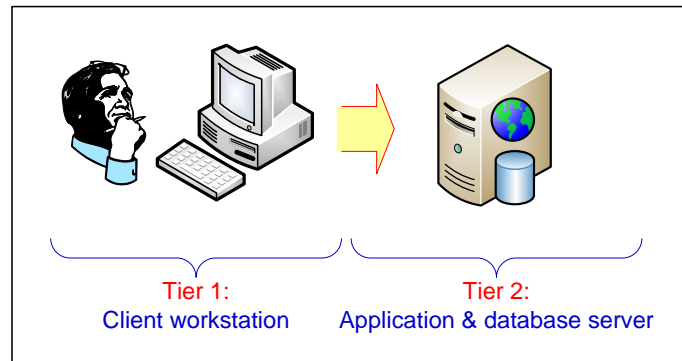
First two tiers in three-tier architecture is combined into one

DW - System architecture

3-tier architecture

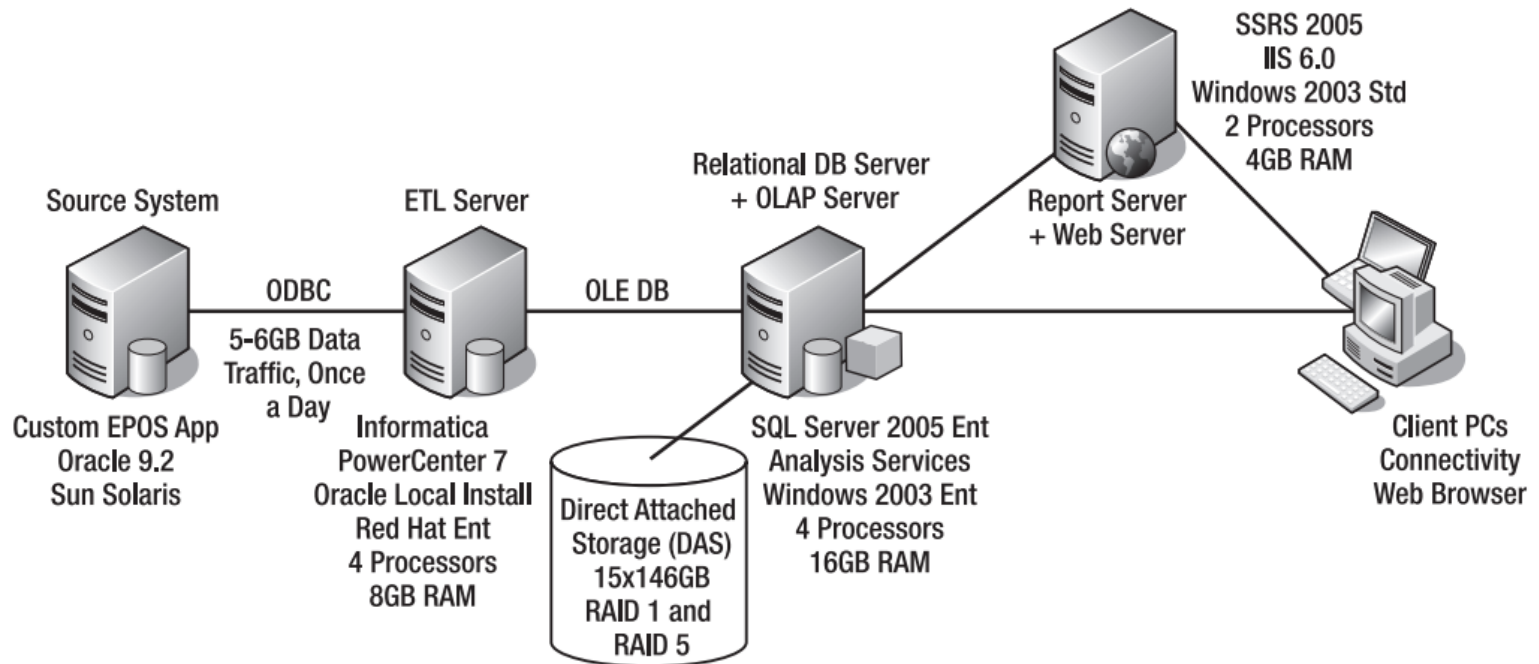


2-tier architecture



1-tier Architecture ?

DW - System architecture



- Once you have chosen a certain data flow architecture, you then need to design the system architecture, which is the physical arrangement and connections between the servers, network, software, storage system, and clients.

DW - System architecture

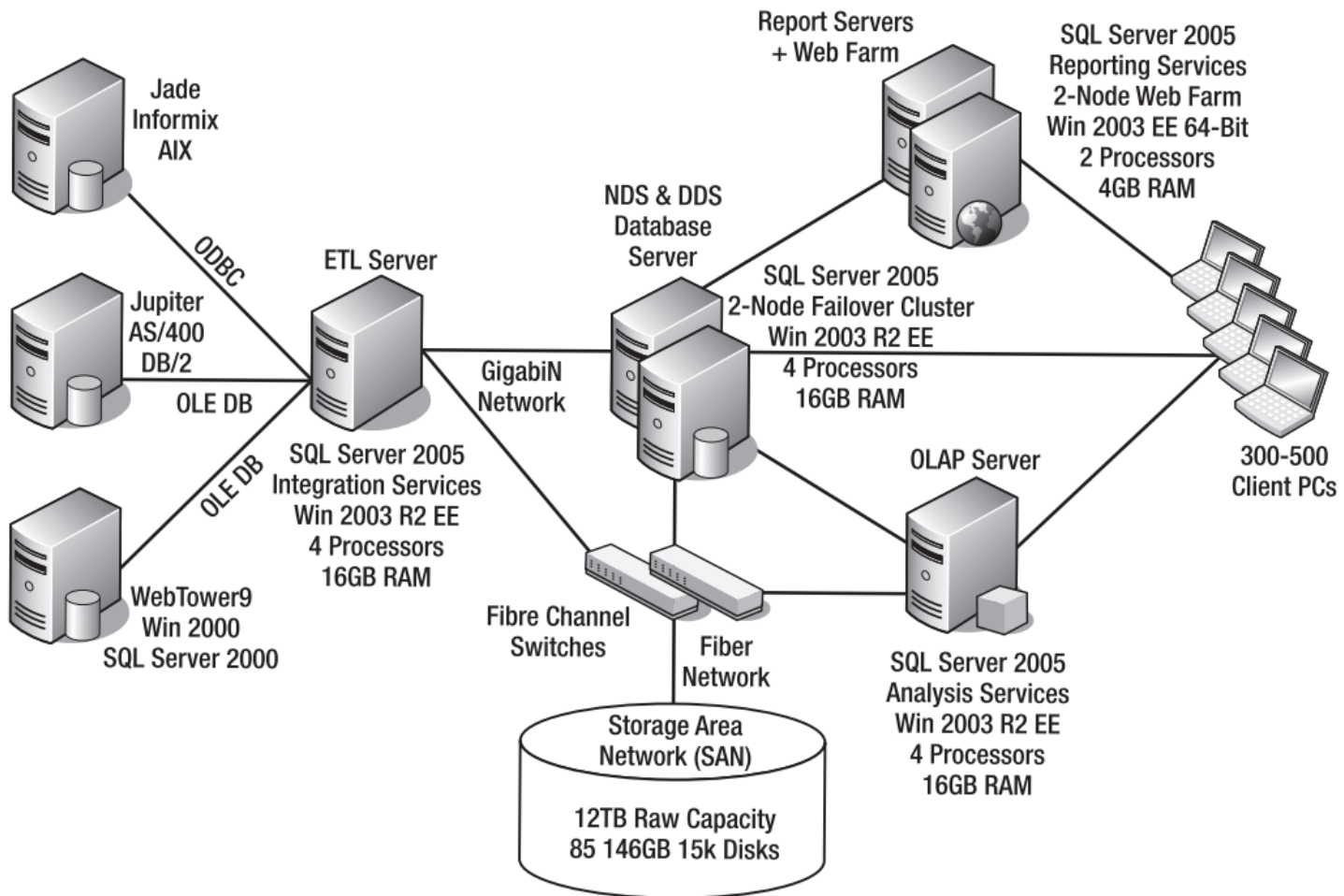


Figure 2-11. System architecture for the production environment for Amadeus Entertainment DW [1]

