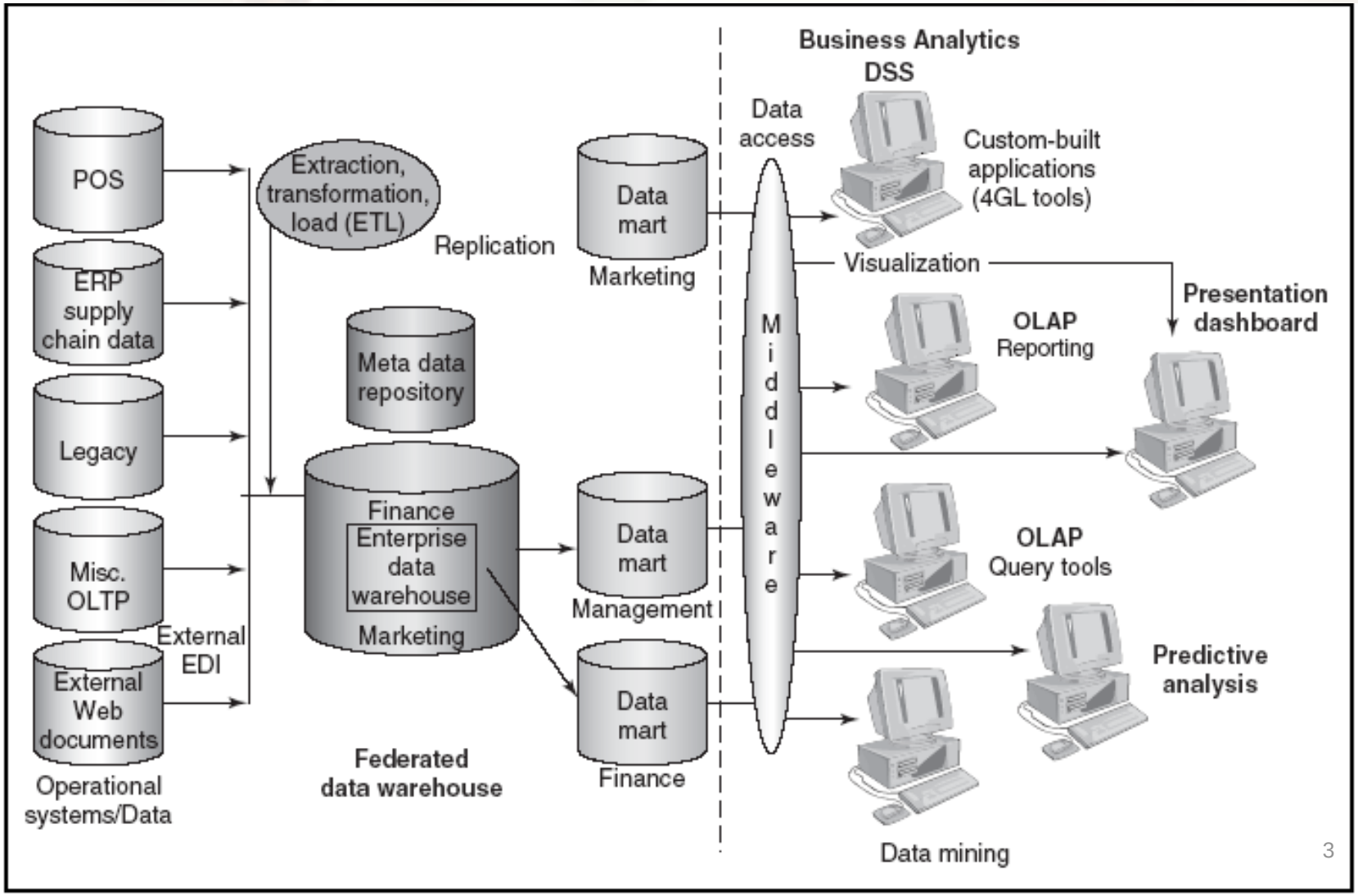
Data Warehouses

Outline

- Data Warehouse Definition and Characteristics
- Data Warehouse Components
- Functionality of Data Warehouse Components
- Data Warehouse Development Cycle

The role of DW



The role of DW

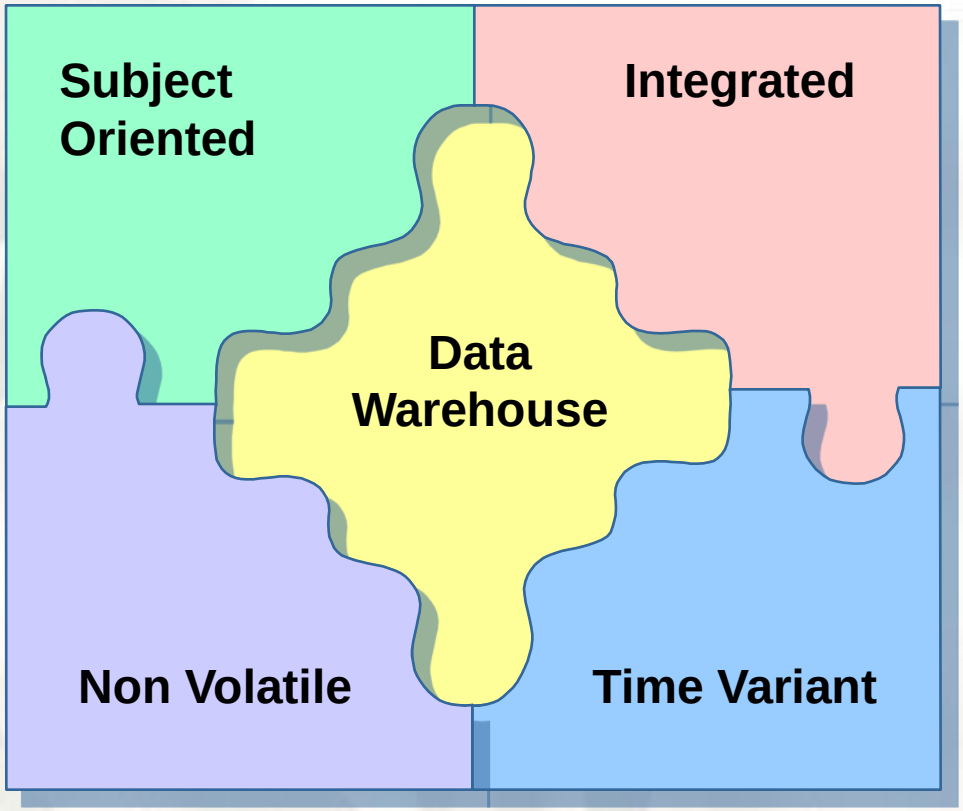
- Operational systems are “black holes”: information goes in but never comes back; hard for end-user ad-hoc queries, reports; everything should be programmed by IT staff
- Data flows from operational systems (e.g., CRM, ERP) to a DW
- DW is a special database prepared to support decision-making applications ranging from those for simple reporting and querying to complex optimization

What is Data Warehouse?

"A data warehouse is simply a single, complete, and consistent store of data obtained from a variety of sources and made available to end users in a way they can understand and use it in a business context."

-- Barry Devlin, *IBM Consultant*

DW Characteristics

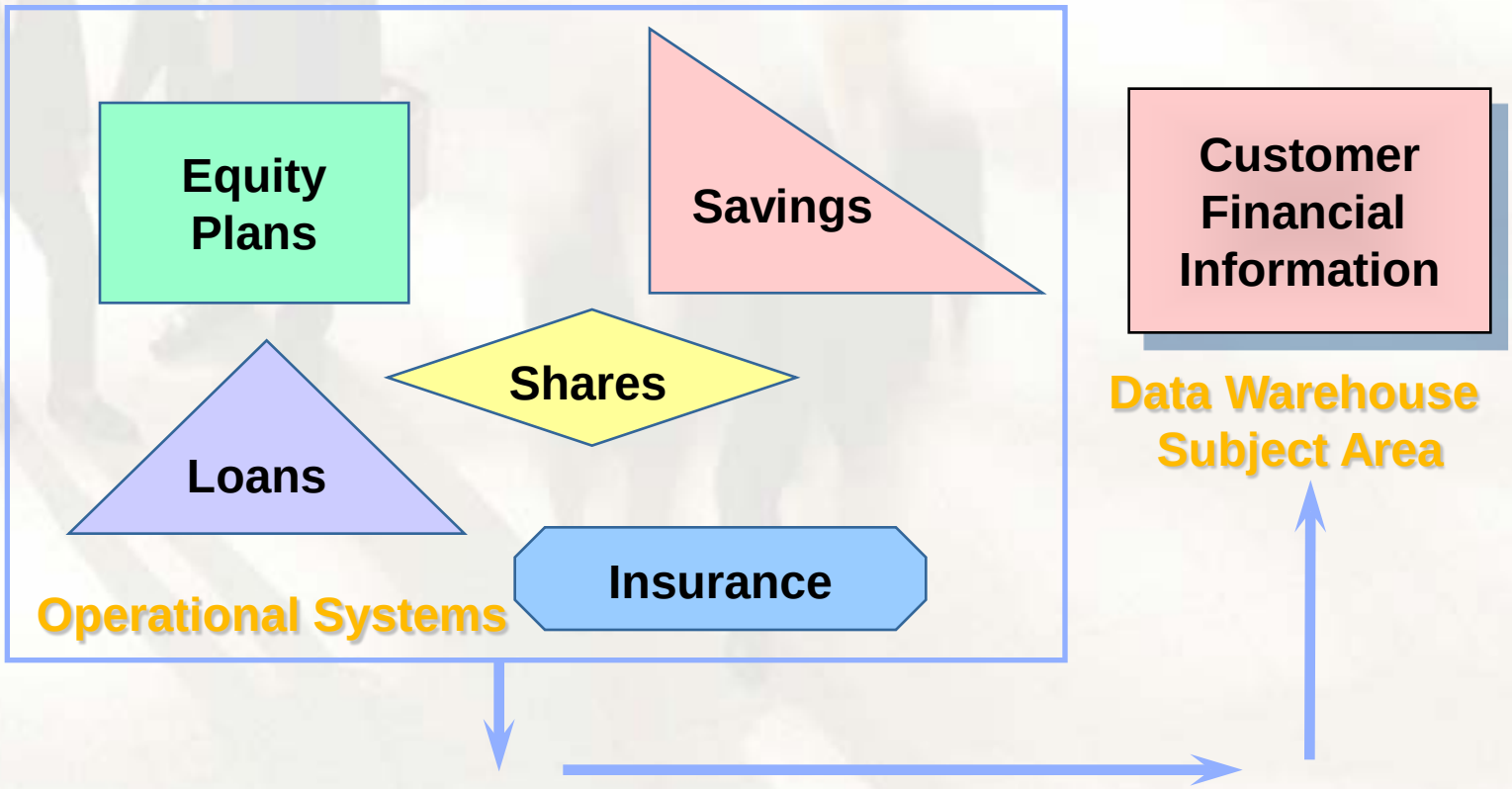


- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."

  —*W. H. Inmon, the father of the term 'data warehouse'*

DW: Subject-Oriented

Data is categorized and stored by business subject rather than by application

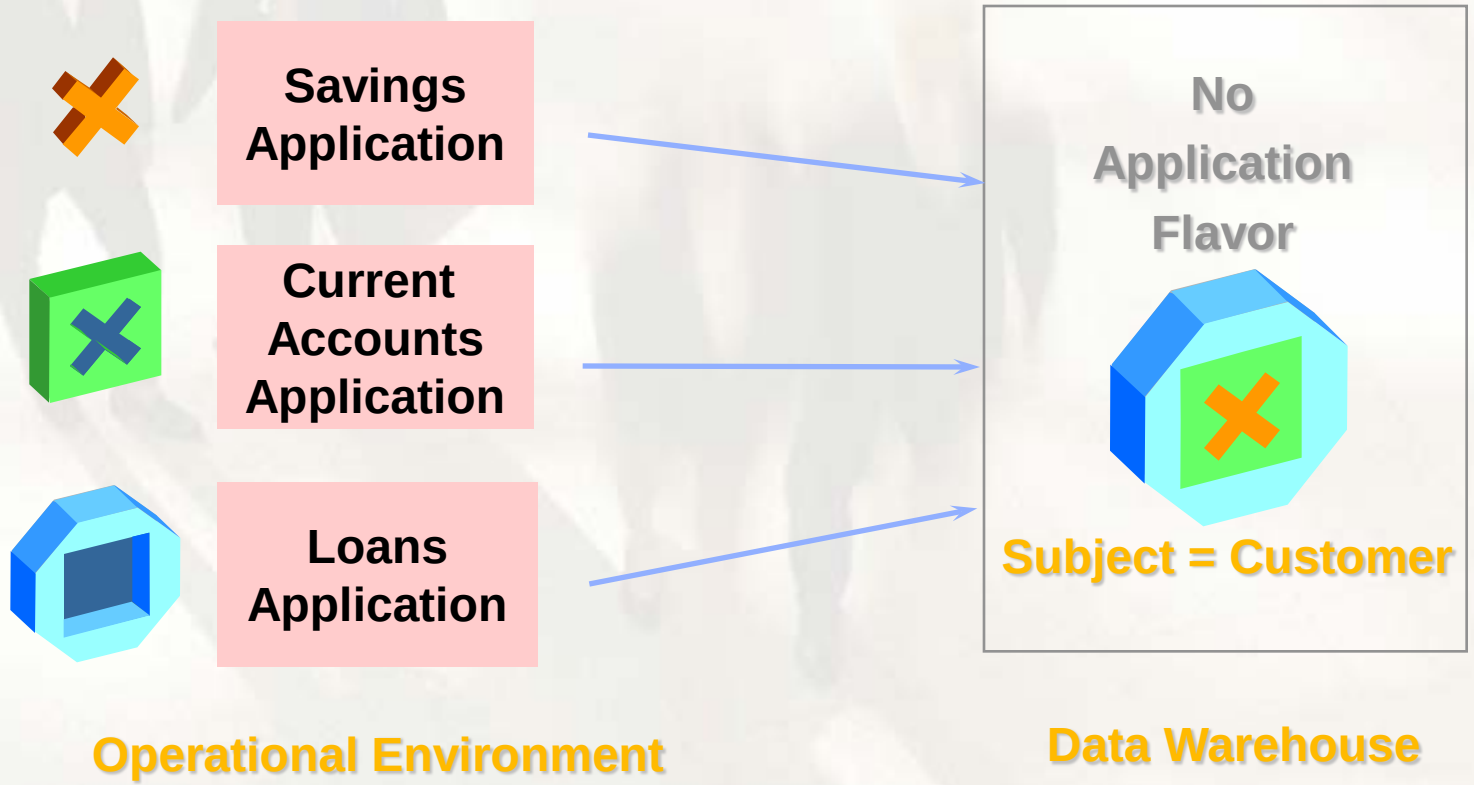


DW: Subject-Oriented

- Organized around major subjects, such as customer, product, sales.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
- Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

DW: Integrated

Data on a given subject is defined and stored once



DW: Integrated

- Constructed by integrating multiple, heterogeneous data sources
  - relational databases, flat files, on-line transaction records
- One set of consistent, accurate, quality information
- Standardization
  - Naming conventions
  - Coding structures
  - Data attributes
  - Measures

DW: Integrated

- Data cleaning and data integration techniques are applied.
  - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
  - When data is moved to the warehouse, it is converted

DW: Time Variant

Data is stored as a series of snapshots, each representing a period of time

Time	Data
01/97	Data for January
02/97	Data for February
03/97	Data for March

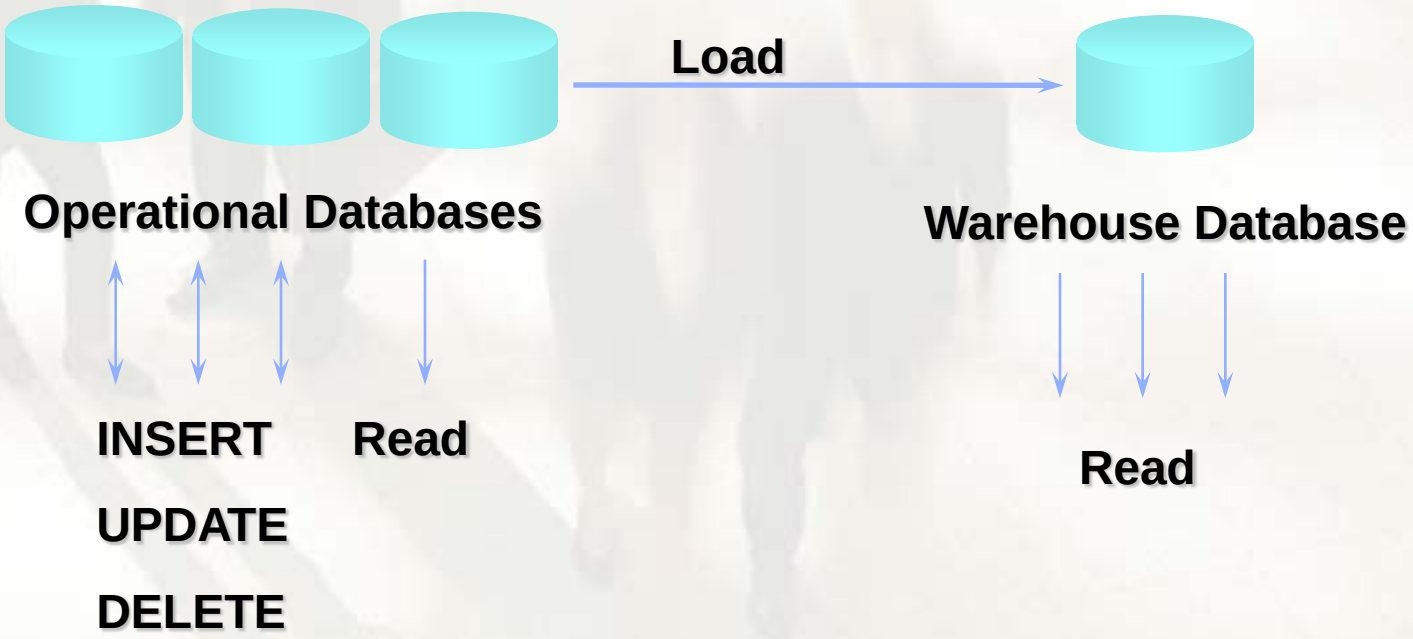
Data Warehouse

DW: Time Variant

- The time horizon for the data warehouse is significantly longer than that of operational systems
  - Operational database: current value data
  - Data warehouse data: provide information from a historical perspective (e.g., past **5-10** years)
- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly
  - But the key of operational data may or may not contain "time element"

DW: Non-Volatile

Typically data in the data warehouse
is **not** deleted



DW: Non-Volatile

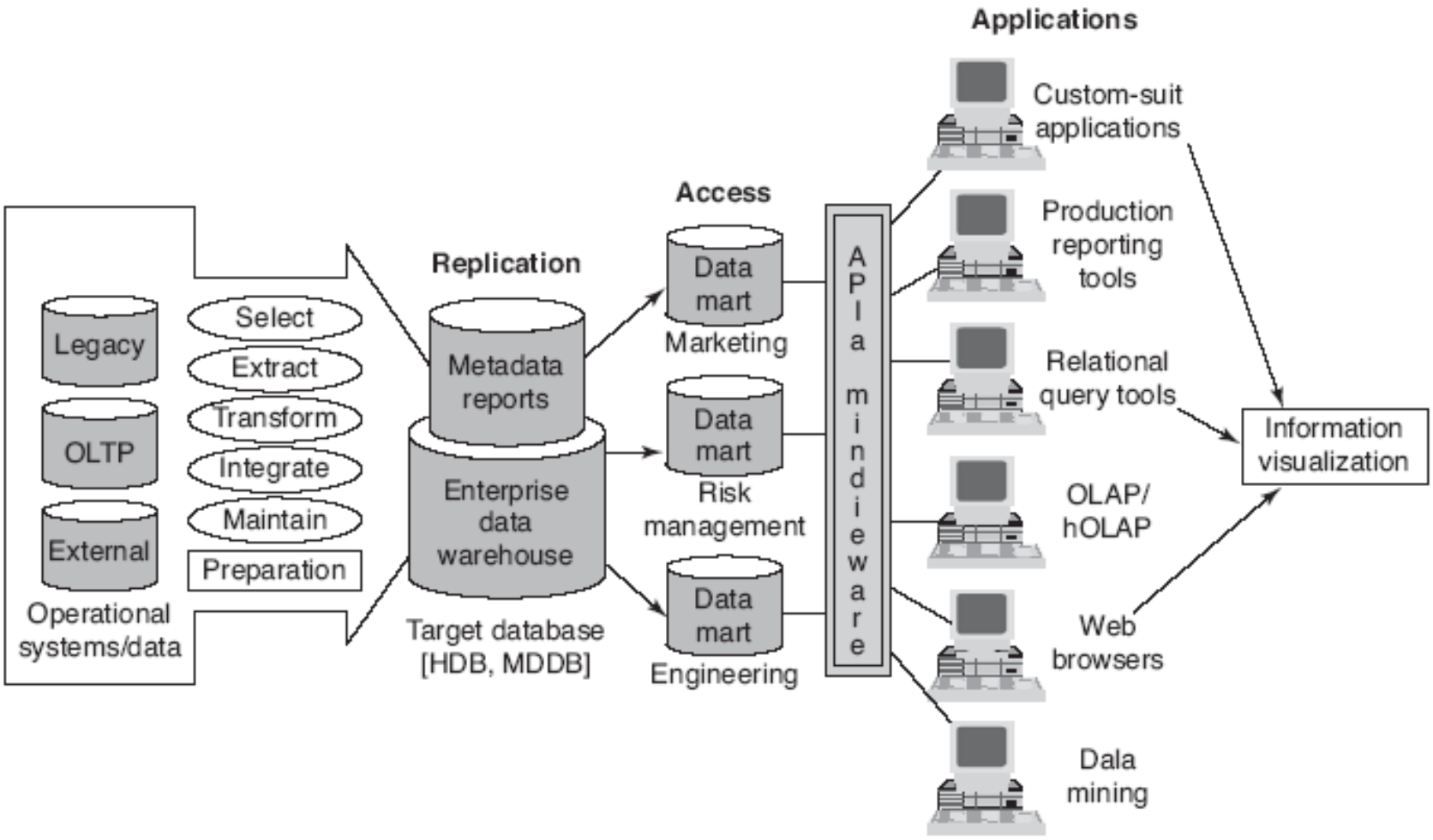
- A physically separate store of data transformed from the operational environment
- Operational update of data does not occur in the data warehouse environment
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*

Outline

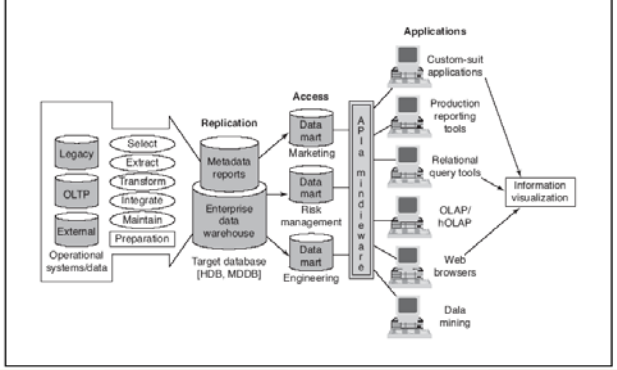
- Data Warehouse Definition and Characteristics
- Data Warehouse Components
- Functionality of Data Warehouse Components
- Data Warehouse System Architectures

Data Warehousing

Components

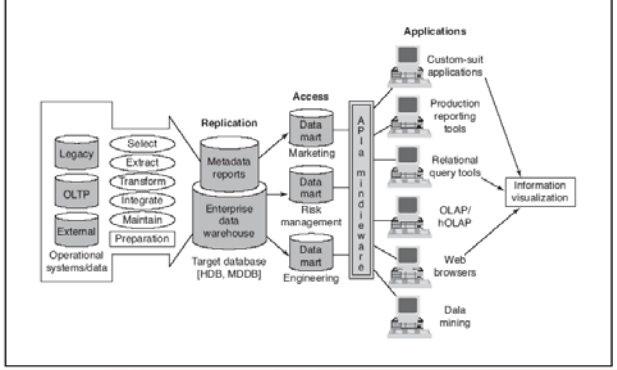


Data Warehousing
Components



- **Enterprise data warehouse (EDW)**
  A centralized repository for the entire enterprise build using a top-down, normalized data model
- **Bottom-up data warehousing**
  Data marts are first created and integrated (bus architecture) to a data warehouse

Data Warehousing
Components



- **Data mart**

  A departmental data warehouse that stores only relevant data

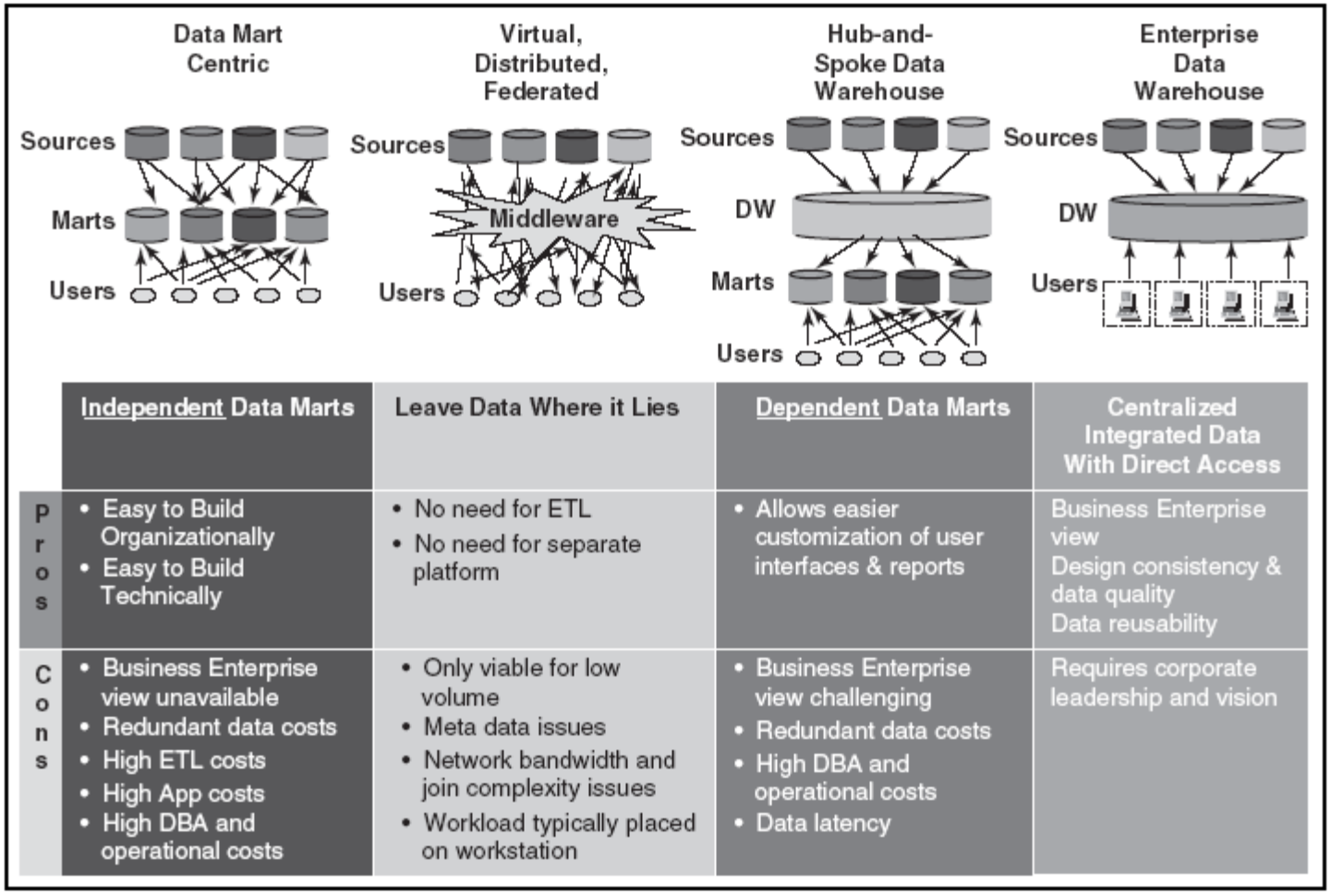
  – **Dependent data mart**

    A subset that is created directly from an enterprise data warehouse

  – **Independent data mart**

    A small data warehouse designed for a strategic business unit or a department (bottom-up approach)

Data Warehousing Process Overview

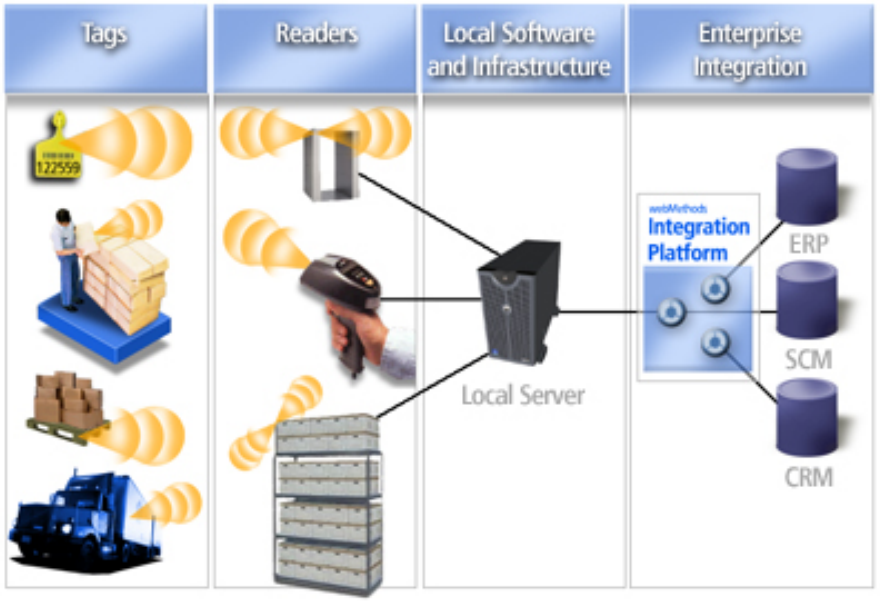


	Independent Data Marts	Leave Data Where it Lies	Dependent Data Marts	Centralized Integrated Data With Direct Access
Pros	• Easy to Build Organizationally • Easy to Build Technically	• No need for ETL • No need for separate platform	• Allows easier customization of user interfaces & reports	Business Enterprise view Design consistency & data quality Data reusability
Cons	• Business Enterprise view unavailable • Redundant data costs • High ETL costs • High App costs • High DBA and operational costs	• Only viable for low volume • Meta data issues • Network bandwidth and join complexity issues • Workload typically placed on workstation	• Business Enterprise view challenging • Redundant data costs • High DBA and operational costs • Data latency	Requires corporate leadership and vision

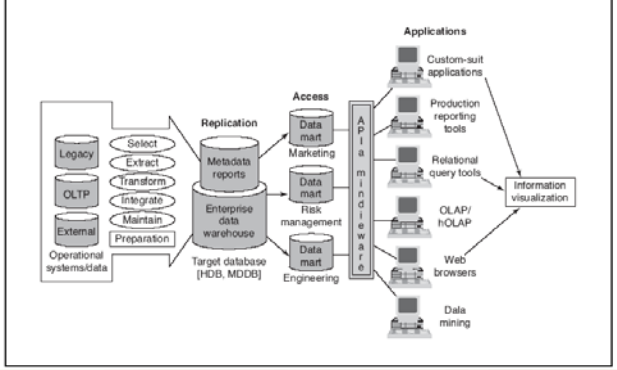
More info: http://en.wikipedia.org/wiki/Data_warehouse

Real-time DW

- Traditional BI systems use periodically updated data
- Attaining real-time, on-demand BI
  - New data-generating technologies, e.g., RFID, is accelerating the need for real-time BI



Data Warehousing Components



- Meta data is the data defining warehouse objects:
  - Description of the structure of the warehouse
    - schema, view, dimensions, hierarchies, data mart locations and contents
  - Operational meta-data
    - data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails)
  - The algorithms used for summarization (measures, gran, etc)
  - The mapping from operational environment to the data warehouse
  - Data related to system performance
    - warehouse schema, view and derived data definitions
  - Business data
    - business terms and definitions, ownership of data, charging policies

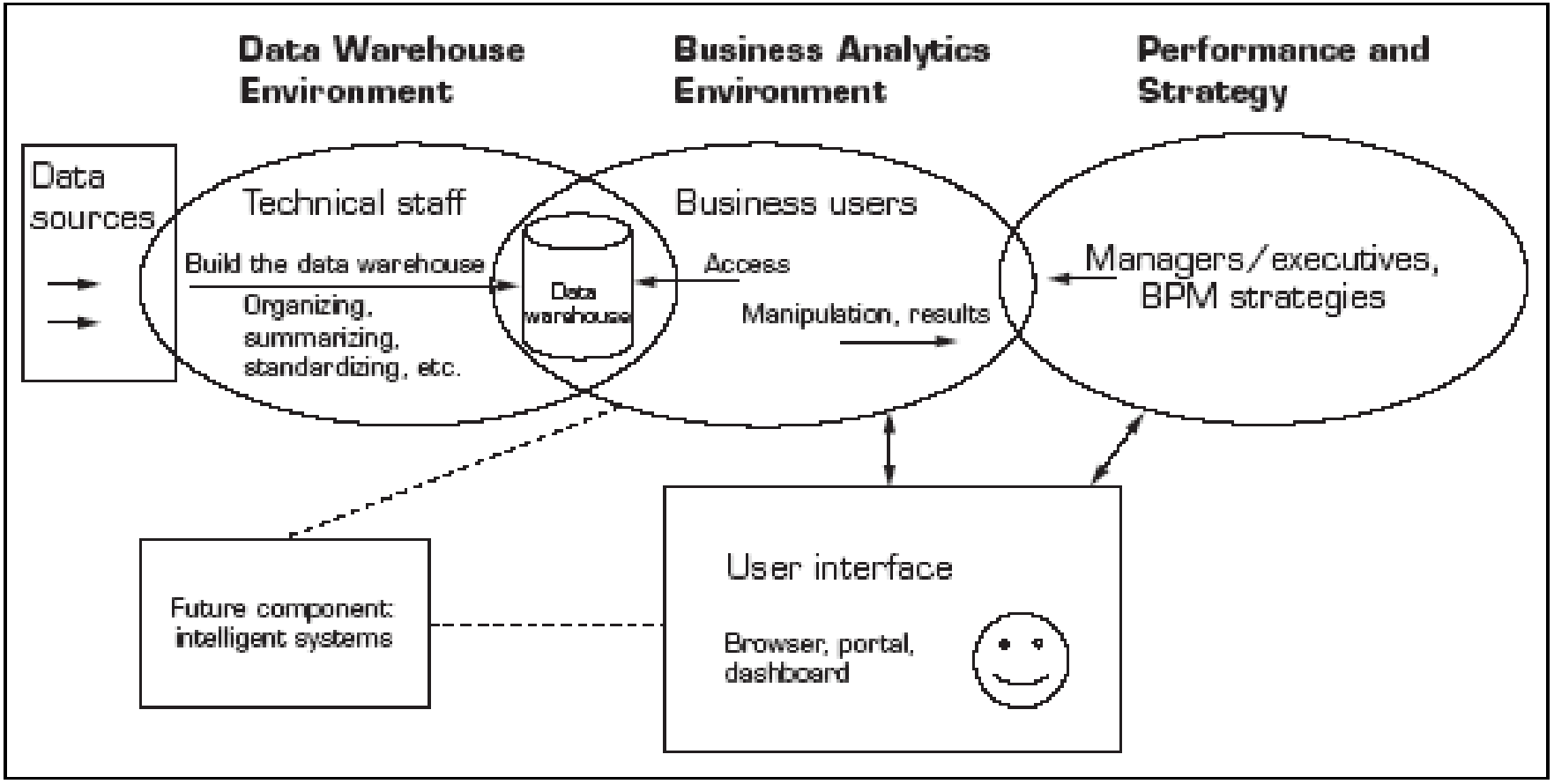
Outline

- Data Warehouse Definition and Characteristics
- Data Warehouse Components
- Functionality of Data Warehouse Components
- Data Warehouse Development Cycle

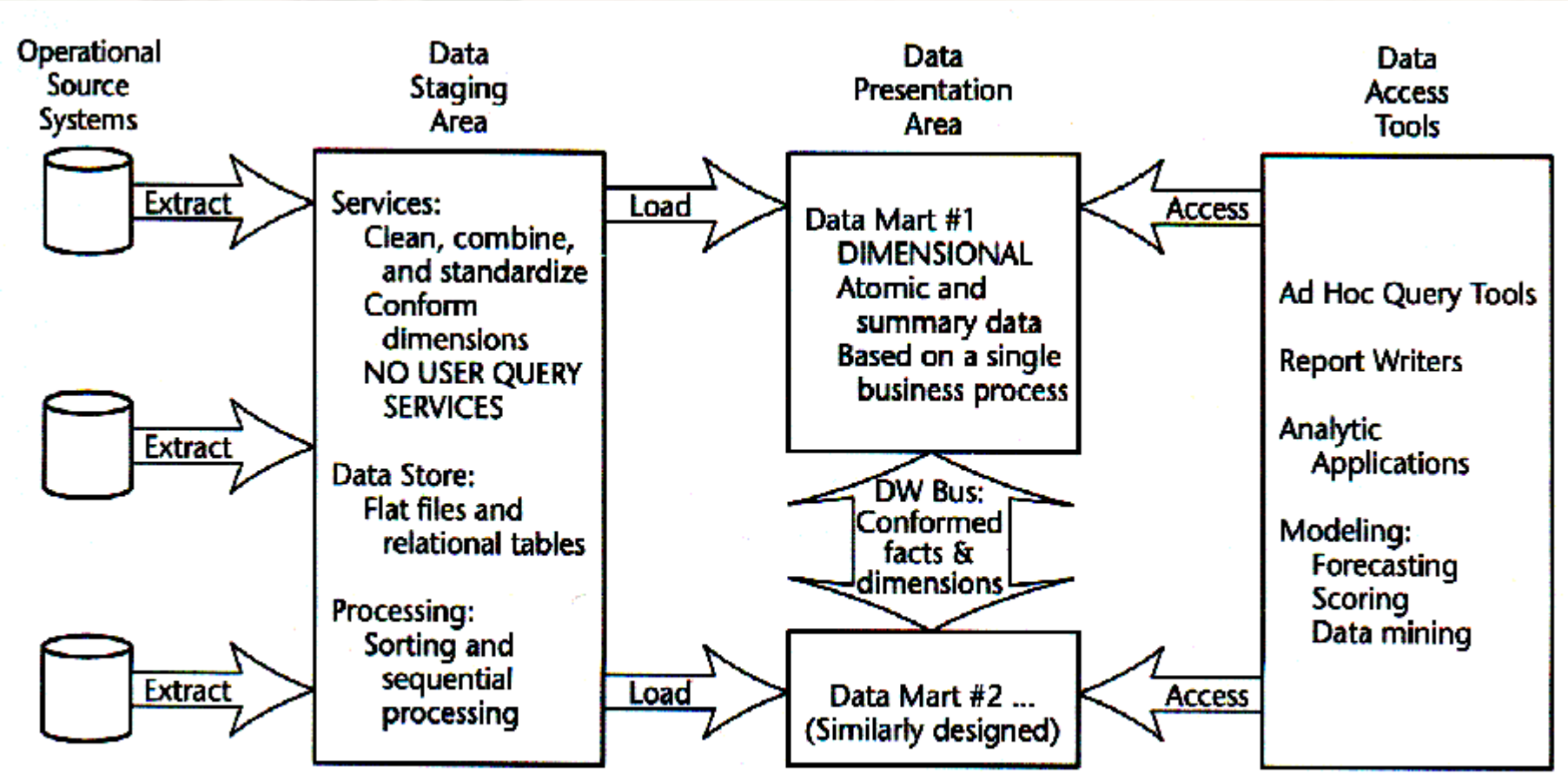
DW users

	Types of Users					
Functionality	***IT***	***Power Users***	***Executives***	***Functional Managers***	***Occasional Information Consumers***	***Extranet: Partners and Customers***
Number of users	Few	Dozens	Dozens	Dozens to hundreds	Hundreds to thousands	Hundreds to thousand
BI tools and functions	Developer, administrator, metadata, security, data management	Ad hoc query, OLAP reports, data mining, advanced analysis	Dashboard, scorecard, reports, CPM	Reports, spreadsheets, OLAP view, business activity monitoring (BAM), corporate performance management (CPM)	Reports, spreadsheets	Reports
Strategic value		High	High	Medium	Low	High
← ..... Intranet ..... →						
←Extranet→						

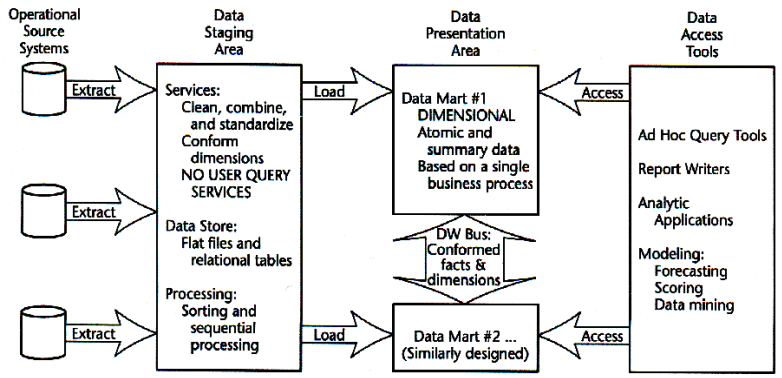
The role DW users



Functionality of DW
Components

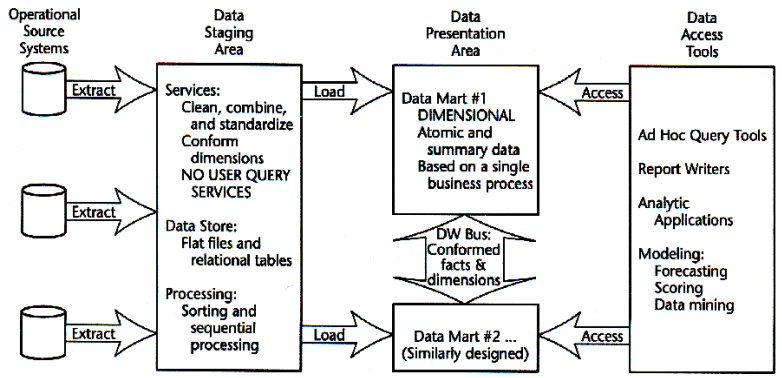


Operational Source Systems



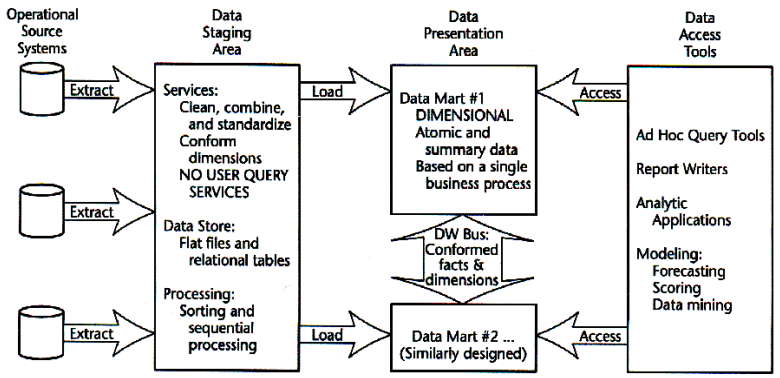
- Each source system is a stovepipe application, little investment to sharing common data (e.g., product, customer)
- Reengineering with consistent view would be great
  - Enterprise Application Integration (EAI) effort will make the pass to DW more easy

Data staging area



- It is off-limits to the business users
- Does *not* provide query and presentation services
- Normalization is not the end goal
  - Normalized databases are excluded from the presentation area, so no need to normalize in data staging area

Data presentation



- Where the data are organized, stored, made available for querying
- This *is* the data warehouse for business community (remember, they can't see data staging area)
- Series of integrated data marts
- A data mart presents the data from a single business process
  - Business processes cross the boundaries of organizational functions

Data presentation

- Data must be stored and accessed in *dimensional schemas*
  - No normalization (3NF) should be used
  - Dimensional schemas are simple and intuitive for business users; Normalized schemas are difficult to grasp by them
- Data must be atomic (at lower granularity)
  - Not only summarized – they don't allow for arbitrary, complex queries
- Data marts must be build on *dimensions* and *facts* that are conformed
  - Otherwise, data marts are stovepipes
  - Conformation leads to bus architecture – data marts can cooperate

Additional Material:
Normalization Example

"every non-key attribute must provide a fact **about** the key, the **whole** key, and **nothing but** the key"

OrderId	ItemId1	ItemId2	…
1	100	101	

Unnormalized

OrderId (PK)	ItemId (PK)	OrderDate	…
1	100	2009-01-01	
1	101	2009-01-01	

1 NF

Orders		
OrderId (PK)	OrderDate	…
1	2009-01-01	

Order_Items		
OrderId (PK)	ItemId (PK)	…
1	100	
1	101	

2 NF

Additional Material:
Normalization Example

Orders			
OrderId (PK)	OrderDate	CustomerName	CustomerCity
1	2009-01-01	John Smith	Chicago

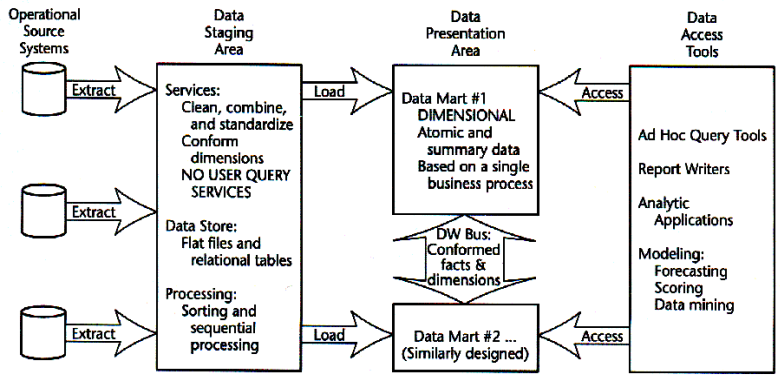
2 NF
but not 3 NF

Orders		
OrderId (PK)	OrderDate	CustomerId (FK)
1	2009-01-01	100

Customers		
CustomerId (PK)	Customer Name	…
100	John Smith	

3 NF

Data access tools



- Ad hoc, complex queries are targeted to small percentage of business users
- 80-90% of the potential users will be served by ‘canned’ applications
  - Canned: pre-build parameter-driven analytic applications for reporting

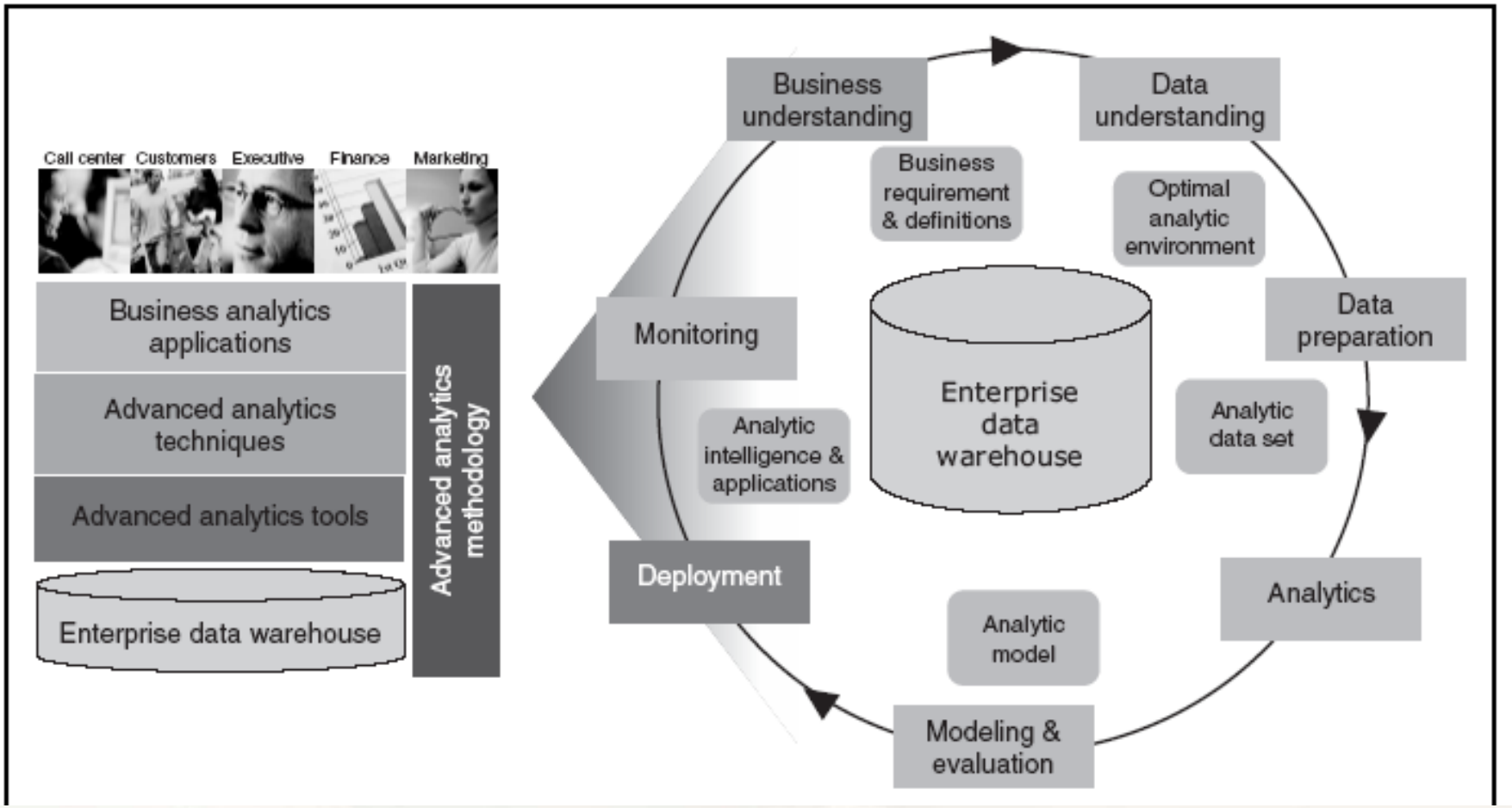
Outline

- Data Warehouse Definition and Characteristics
- Data Warehouse Components
- Functionality of Data Warehouse Components
- Data Warehouse Development Cycle

DW development process

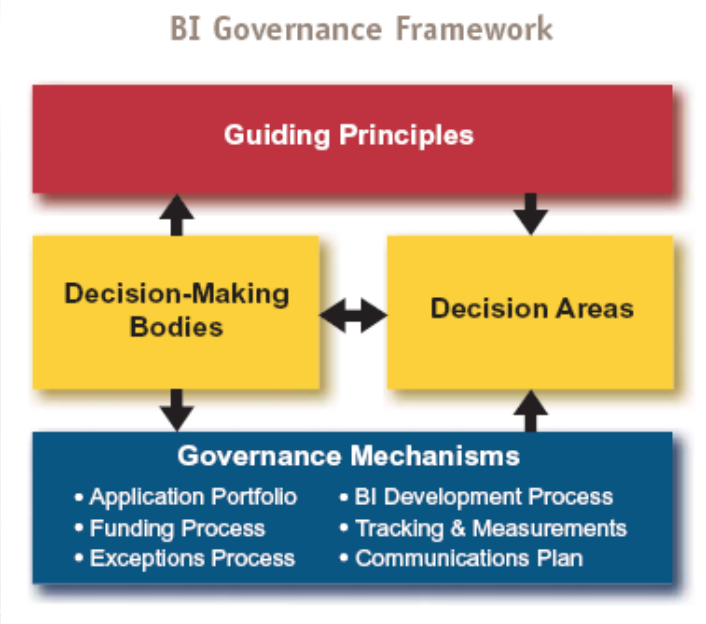
- Developing vs. acquiring support systems
- Justification and cost-benefit analysis
  - Prioritizing the steps of DW
- Security and protection of privacy
  - BI handles critical information
- Integration of systems and applications
  - CRM, ERP, e-commerce, legacy
- The Web in DSS/BI implementation
  - Information portals, pervasive BI

DW Development Cycle



BI Governance

- BI governance:
  - Project management during BI development process
- Four components of good governance:
  - guiding principles, decision-making bodies, decision areas and decision rights, and governance oversight mechanisms



BI Governance best practices

- Actively design BI Governance.
- Position BI Governance within a big picture view of the corporation.
- Establish joint business and IT decision-making and accountability.
- Adopt portfolio management with controlled decision-making to prioritize investments and allocate resources.

BI Governance best practices

- Separate the roles of governance and management in order to align decisions, activities, inputs, and deliverables at the right level in the governance structure.
- Enable closed loop feedback for continuous program improvement.
- Implement a communications plan.
- Measure performance and publish value achieved.

Acceptance of BI

- Upper managers must build enthusiasm for the intra-organizational sharing of BI best practices
- BI acceptance criteria will be about:
  - data consistency, report completeness, authorization, accessibility, report lay-out and performance
  - These criteria help in accepting BI
  - They have to be made specific and measureable in order to be tested