VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

OLAP and DW Architectures

Lecturer : Dr. Pavle Mogin

SWEN 432
Advanced Database Design and Implementation

Plan for OLAP & DW Architectures

- Common sources of Data Warehouse building failures
- A classification of OLAP architectures
- Actual OLAP&DW architectures
- Good Architectures are aimed for avoiding failures
- An approach to Data Warehouse design
- OLAP and Cloud Databases
- *Reading:*
  - *Chaudhuri, Dayal : An Overview of Datawarehousing and OLAP Technologies*
  - *Mimo, P.R.* : *“Mistakes to Avoid in Building Data Warehouses”, Cutter IT Journal, Vol.12, No6, June 1999, pp 36-50*

OLAP & DW Building Failures

- Common sources of Data Warehouse building failures:
  - Missing business drivers
  - Use of a wrong architecture
  - “Dirty” source data
  - Top down development
  - Neglecting scalability and performance issues

Missing Business Drivers

- The Data Warehouse should be built to solve a recognized and well defined business problem
- Examples of such problems are:
  - Customers are moving to competitors,
  - Management has little insight and control over costs
  - Promotions are failing for unknown reasons
  - There is a high turnover of goods and high cost of inventory
  - The organization has an inadequate understanding of customer needs

A Certain Cause of a DW Project Failure

- If the business drivers are missing and:
  - A business manager decides to build a DW because the others have it, or
  - An IT manager decides to build a DW hoping the business managers will use it
- Such a Data Warehouse project is likely to fail

Common OLAP Characteristics

- OLAP systems contain six common architectural characteristics:
  - Advanced support to data management,
  - User interface adopted to the user knowledge and needs,
  - Multidimensional data structures,
  - Techniques of multi dimensional data analysis, and
  - Metadata repository

OLAP Server Architectures

- An OLAP system can be implemented using a:
  - Traditional relational database server
  - Specialized SQL server
  - ROLAP server, or
  - MOLAP server
- Although traditional relational servers are not aimed at supporting OLAP queries and Gbyte databases efficiently, they may be used to accomplish these tasks to some extent

Specialized SQL Servers

- The objective of a specialized SQL server is to provide an advanced query language and query processing support for OLAP queries over relational multidimensional structures
- SQL is extended with appropriate commands (`MATERILIAZED VIEW, CREATE DIMENSION, CUBE, ROLLUP , WINDOW, OPTIMIZE,` …)
- Query processing engine is enhanced to support and utilize:
  - Functional dependencies,
  - Materialized views, and
  - New kinds of indices

  in an intelligent way

ROLAP Servers

- ROLAP servers posses OLAP tools and are built as intermediate servers between a relational back end server and a client front end
- Relational back end stores and manages data
- ROLAP server is used to optimize OLAP specific queries for relational back end by:
  - Identifying views to be materialized,
  - Rephrasing user queries to use materialized views, and
  - Generating multi-statement SQL for the back end server (e.g. to execute a pivot operation defined by `CUBE` clause using multiple `SELECT … GROUP BY` statement and storing intermediate results in temporary tables)

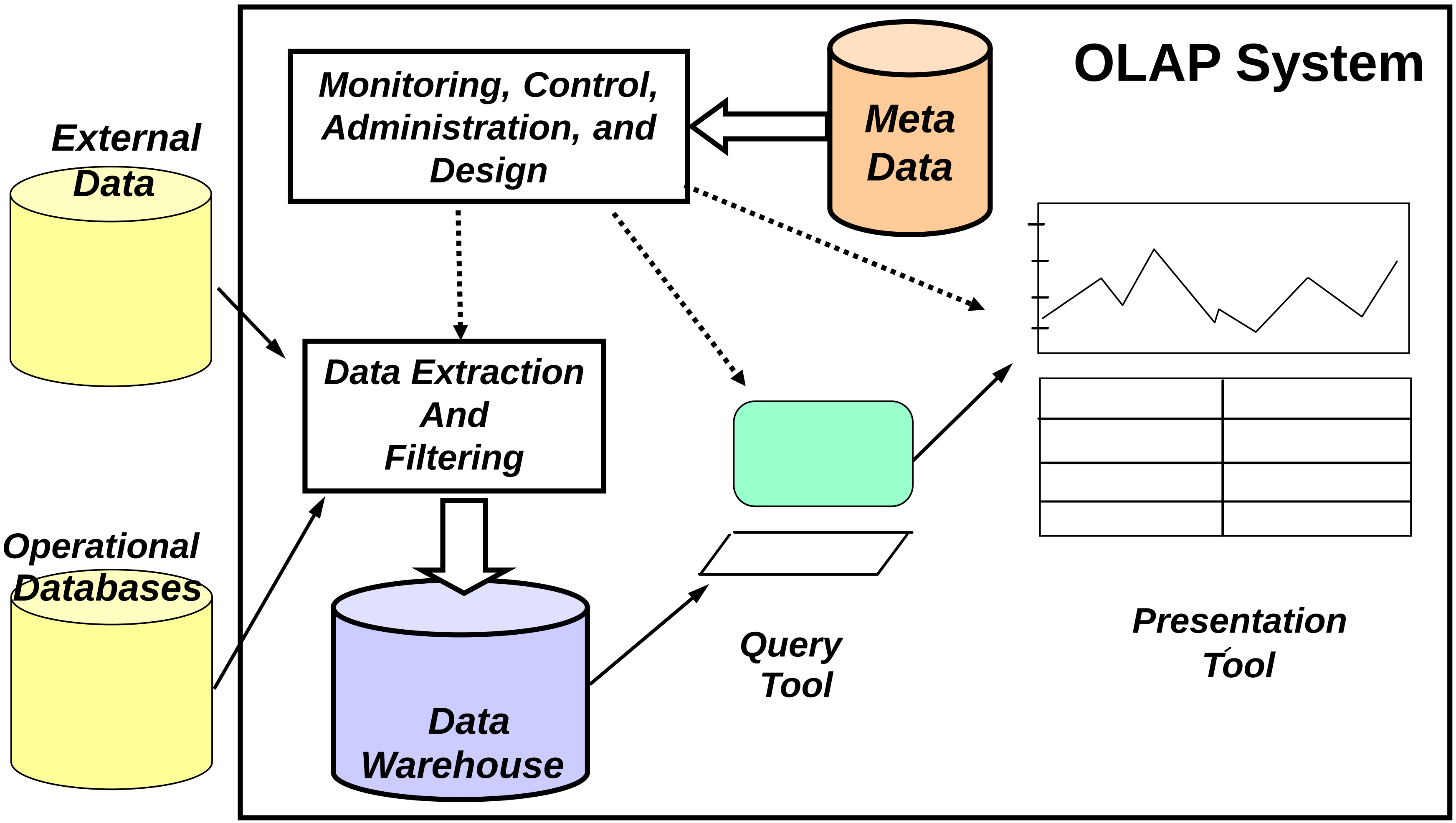
MOLAP Servers

- Directly support multidimensional view of data through a multidimensional storage engine
- Use arrays to build hyper cubes
- Execute multidimensional front end queries directly against hyper cubes
- MOLAP is a specialized system that efficiently supports:
  - Queries involving aggregate and group by operators,
  - Complex boolean functions,
  - Various statistical functions, and
  - Time related queries

OLAP Basic Architectures

- OLAP systems can use data from operational databases to execute data analysis queries, but
- Very often they poses tools for building their own multidimensional Data Warehouse from operational databases
- Also, a separate specialized software can be used for data extracting, filtering, and integration of operational data into Data Warehouse
- Data analysis (against multidimensional and operational data) is done by OLAP front end components

A Basic OLAP System Structure



Basic OLAP Distributed Architectures

- In principle, an OLAP database may be implemented using:
  - A centralized Data Warehouse, or
  - A federation of Data Marts
- A centrally controlled Data Warehouse may be distributed for:
  - Load balancing,
  - Availability (higher reliability), and
  - Scalability (better performance)

  reasons, but will still retain the control over the metadata repository
- Federation of Data Marts is cheaper, faster, and easier to implement, but separate and independent metadata repositories may lead to disintegration

Advanced Support to Data Management

- It is often stressed that OLAP systems contain an advanced support to data management
- Advanced support to data management pertains to :
  - Ability to access:
    - Operational databases organized using different DBMS's,
    - Conventional files, and
    - Data Warehouse
  - Support to management of very large databases, and
  - Possessing an own **metadata repository**

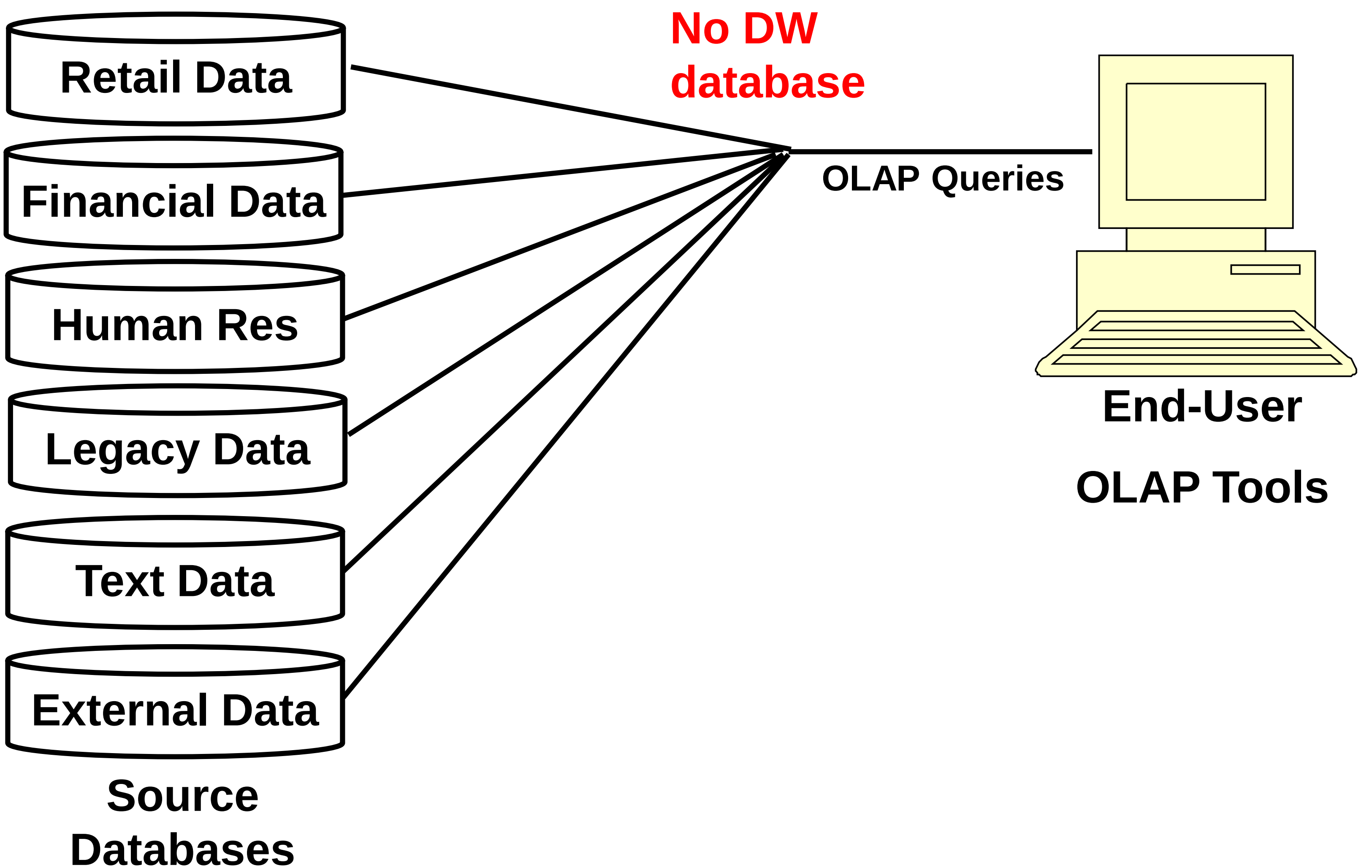
Wrong DW Architecture

- Many Data Warehouse projects fail due to the selection of an architecture that is incapable to meet business requirements
- A desire to build a Data Warehouse quickly and cheaply often leads to selecting a wrong architecture
- There exist architectures that are generally considered to be wrong:
  - “Virtual” Data Warehouse,
  - “Data Mart in a Box”,
  - “Stovepipe” Data Marts

“Virtual” Data Warehouse Architecture

- No Data Warehouse database
- Business analysts access operational databases using simple OLAP front-end tools
- Popular because:
  - Requires minimum investment in additional hardware and software
  - No extra IT personal needed
  - No extracting, cleaning and loading burden
  - The front-end data access and analysis tools simplify access to legacy database systems on mainframes, and allow multidimensional queries on views and drill-down operations on operational data
  - No extra end user skills needed

"Virtual" Data Warehouse Architecture



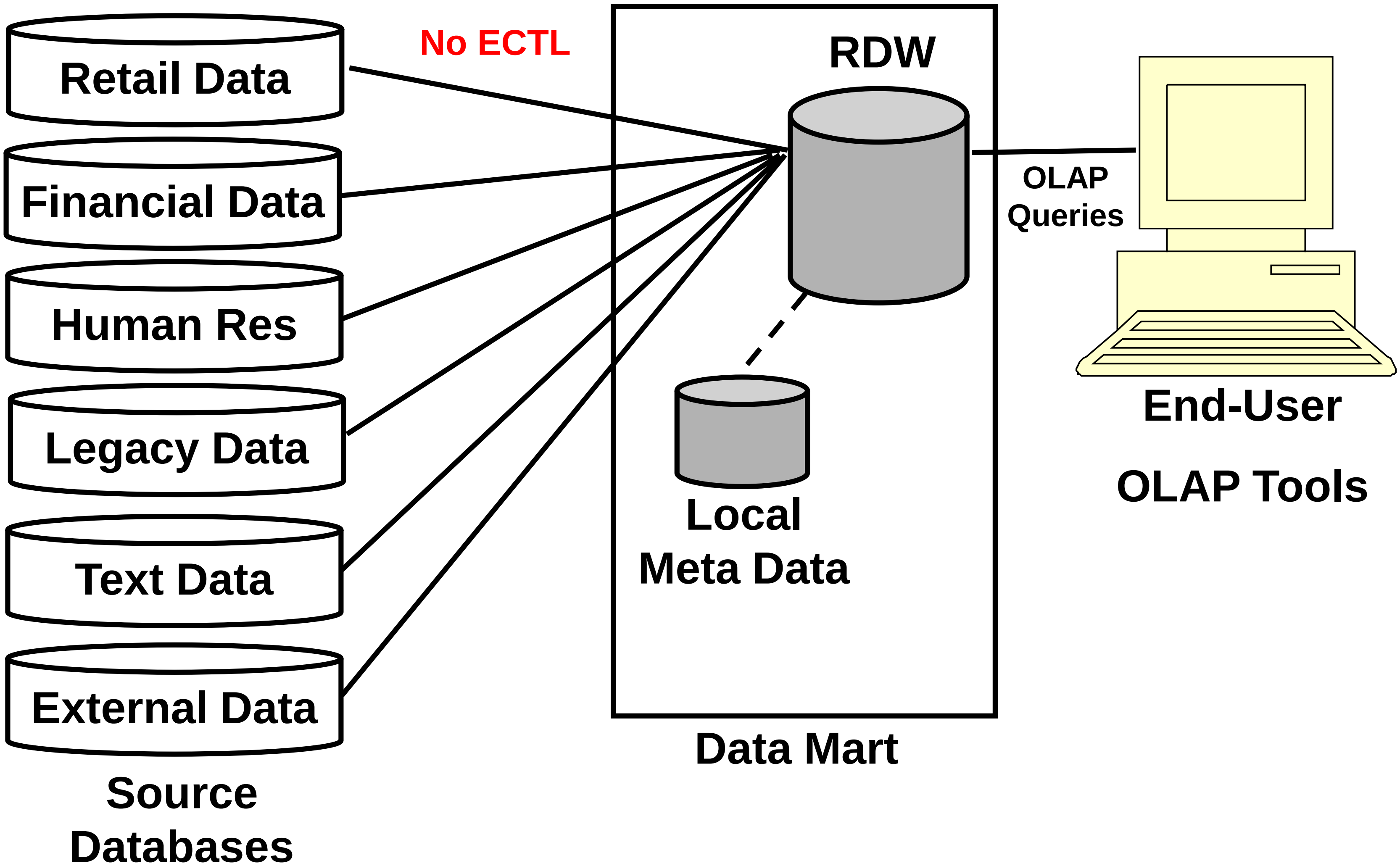
Limitations of "Virtual" DW

- Since no true DW database is built, there is no:
  - Historical data,
  - Summarized and aggregated data,
  - Central meta data repository with enterprise wide definitions of the business data semantics
  - Cleaning and transforming operational data to suit the decision making processes
- OLAP queries and OLTP transactions compete for the same resources
- A "virtual" DW can be considered as a really short term temporary solution

Data_Mart_in_a_Box Architecture

- A packaged product that allows:
  - Building a Data Warehouse database that supports needs of an individual business unit using data from various sources
  - Accessing DW database using user friendly data access and analysis tools
  - Building a local meta data repository with data definitions in business terms

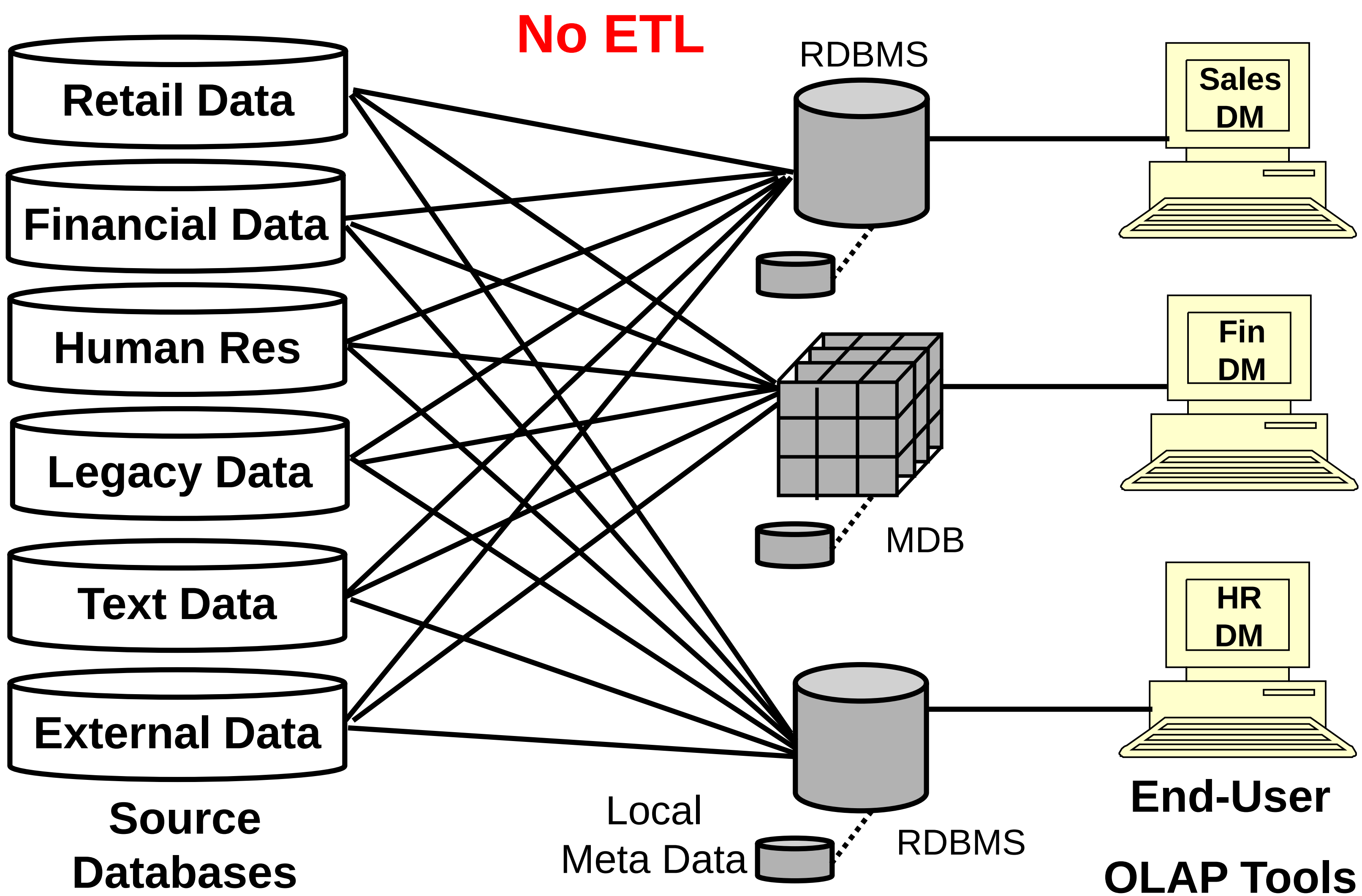
"Data_Mart_in_a_Box" Architecture



Advantages and Disadvantages

- The data_mart_in_a_box architecture eliminates the interference of OLAP operations with OLTP
- But it retains some of the old and introduces some new problems:
  - This architecture tends to proliferate in an uncontrolled manner leading to multiple, non integrated, independent, local data marts, purchased from different vendors
  - Lack of support for common business rules, semantics, and data definitions across business areas (although every data mart maintains its own meta data repository)
  - Population of data marts with “dirty” source data
  - Data inconsistency across various data marts

Independent Data_Marts_in_a_Box



The Dirty Data Problem

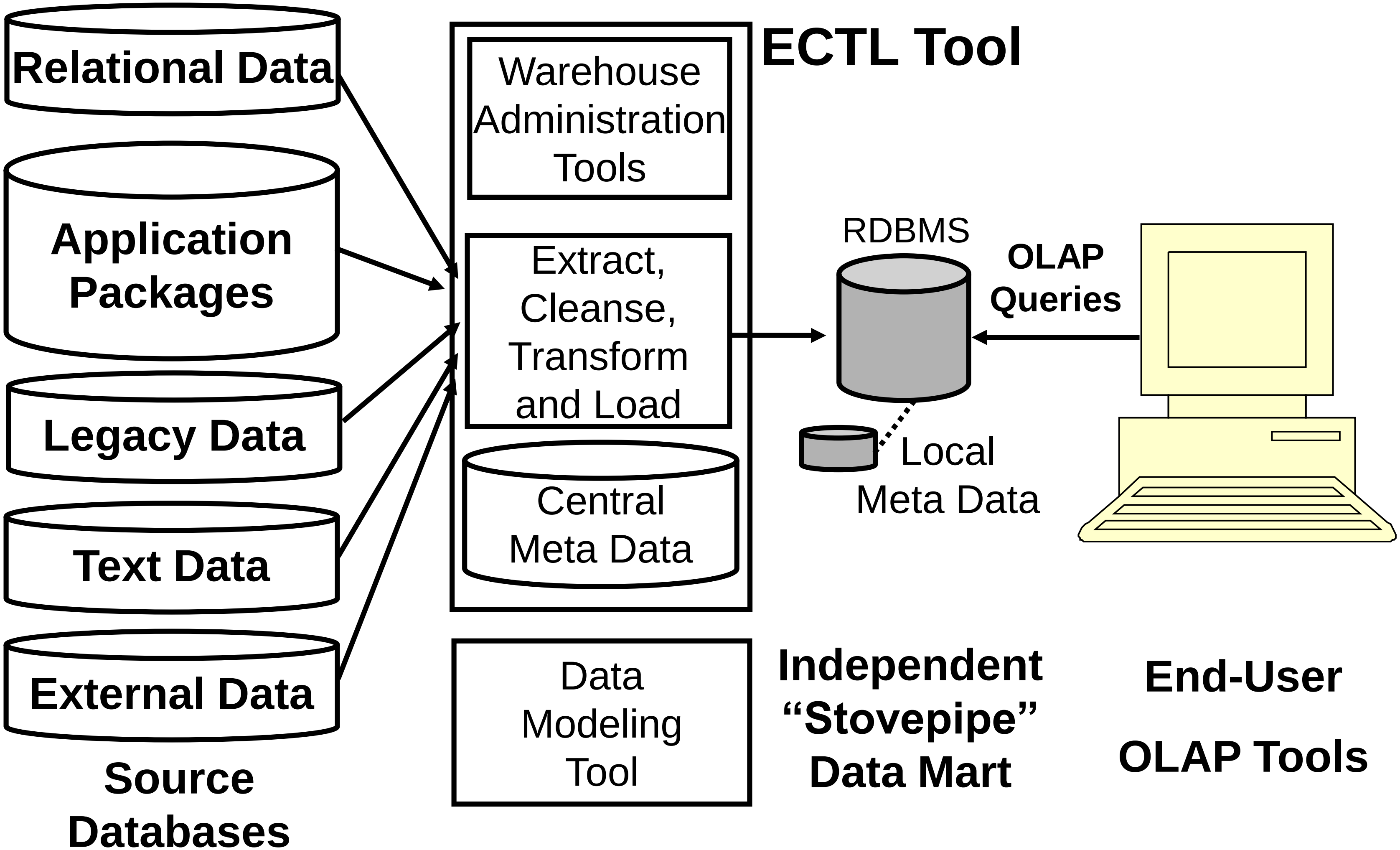
- Data stored in the legacy databases have high percentage of:
  - missing,
  - erroneous, or
  - inconsistent

  data values
- Examples of “dirty” data are:
  - multiple attribute values in one field,
  - one attribute value across two or more fields,
  - different spellings of the same attribute value,
  - inconsistent names for legal entities,
  - incorrect use of codes across records.
- Up to 20% of fields can contain such “dirty” data

Independent (Stovepipe) Data Marts

- The data_mart_in_a_box architecture is enhanced by introducing a single (central) Extraction, Cleansing, Transformation and Loading software package (ETL tool, also called Data Staging)
- Besides ECTL functions, the ECTL tool:
  - Generates and maintains a centralized meta data repository,
  - Offers data warehouse administration facilities,
  - Performs summarizations and aggregations,
  - Loads cleansed, transformed, and reorganized data into the target data marts,
  - Contains an interface to a data modeling tool
- This architecture is often called ***stovepipe*** data mart

"Stovepipe" Data Mart Architecture



ECTL Tool

- ECTL tool eliminates (or at least significantly tempers) the “dirty” data problem
- ECTL tool acts as a single central point that provides coordinated access to source data
- ECTL tool generates and maintains central meta data repository

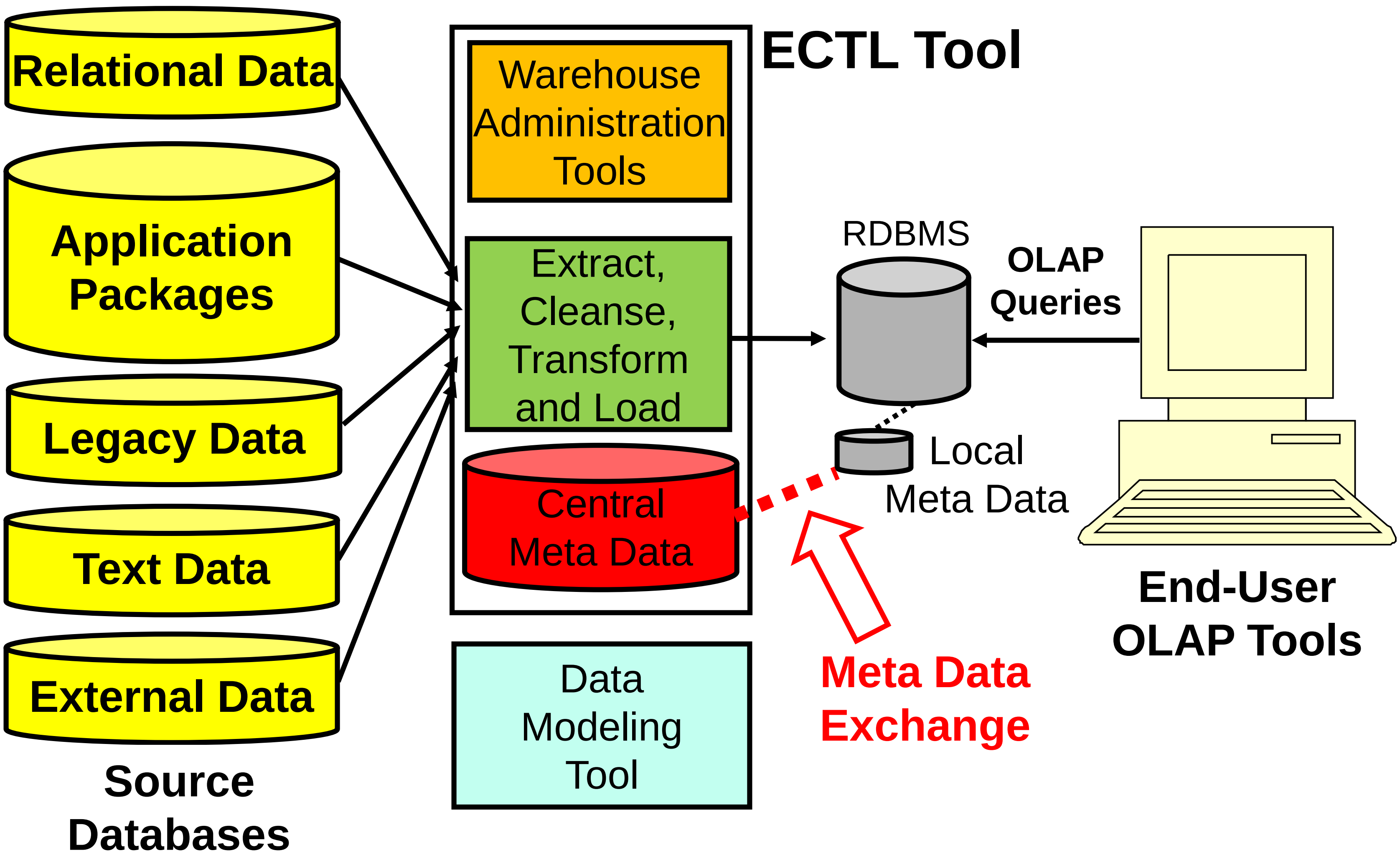
The “Stovepipe” Data Mart Problem

- The main disadvantage of the stovepipe architecture is lack of integration between the central and local meta data repositories
- Even worse, many stovepipe vendors do not provide any means to establish that link
- That way, there are many mutually independent data marts developed in a company
- These data marts support needs of individual business units, but can’t support corporate level needs

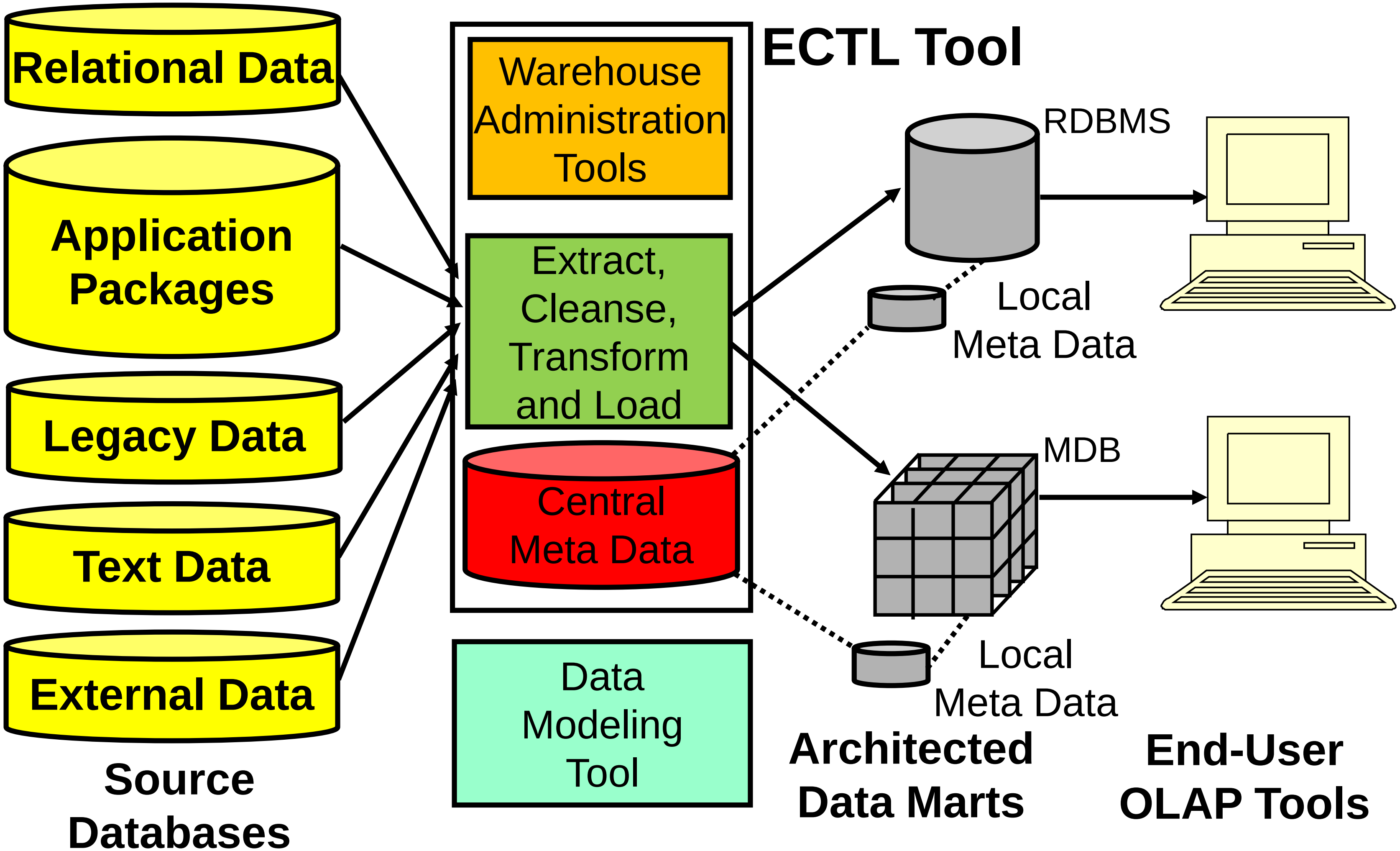
Architected Data Mart

- To avoid the “stovepipe” data mart problem a new component – meta data exchange software should be added
- The meta data exchange component conforms local meta data repositories with the central one
- That way, the central meta data repository becomes the hart of the Data Warehouse

Architected Data Mart



Architected Data Marts



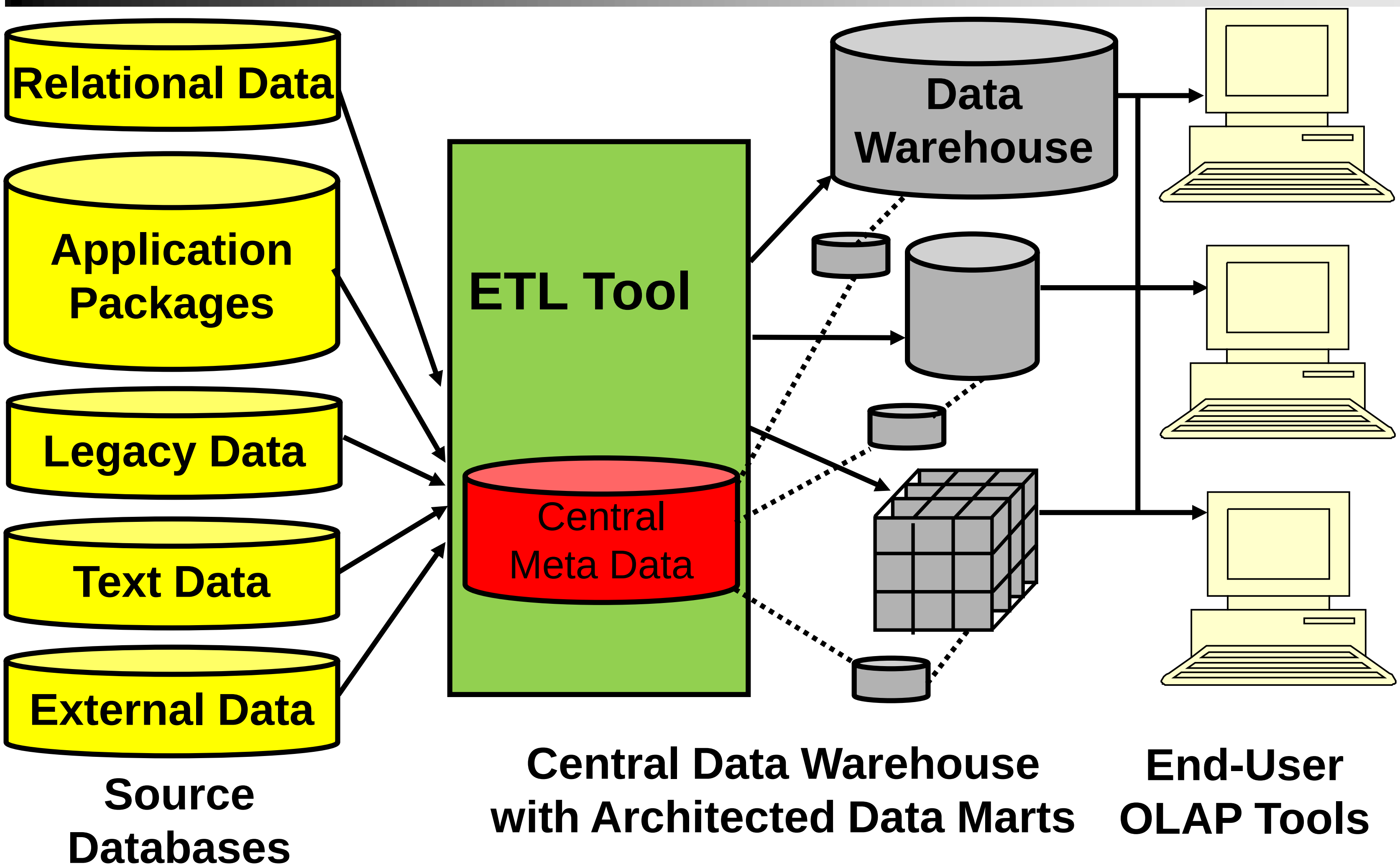
The Central Meta Data Repository

- The central meta data repository provides a “single version of the truth”
- It contains:
  - Enterprise wide source data definitions,
  - Business data semantics,
  - Logical and physical data models for the target databases,
  - Data sources descriptions,
  - Source to target data mappings,
  - Data cleansing rules,
  - Data transformation rules,
  - Procedures to generate summary and aggregate data
- Unfortunately, there is no industry wide accepted meta data repository standard

Enterprise DW Architecture

- Multiple data sources
- Off-the-shelf ETL tool
- Central Meta Data Repository (CMDR)
- Meta data exchange component
- Central Data Warehouse
- Multiple architected data marts
- Central Data Warehouse coordination and management (through CMDR)
- Data access and analysis tools
- Web access

Enterprise Data Warehouse Architecture



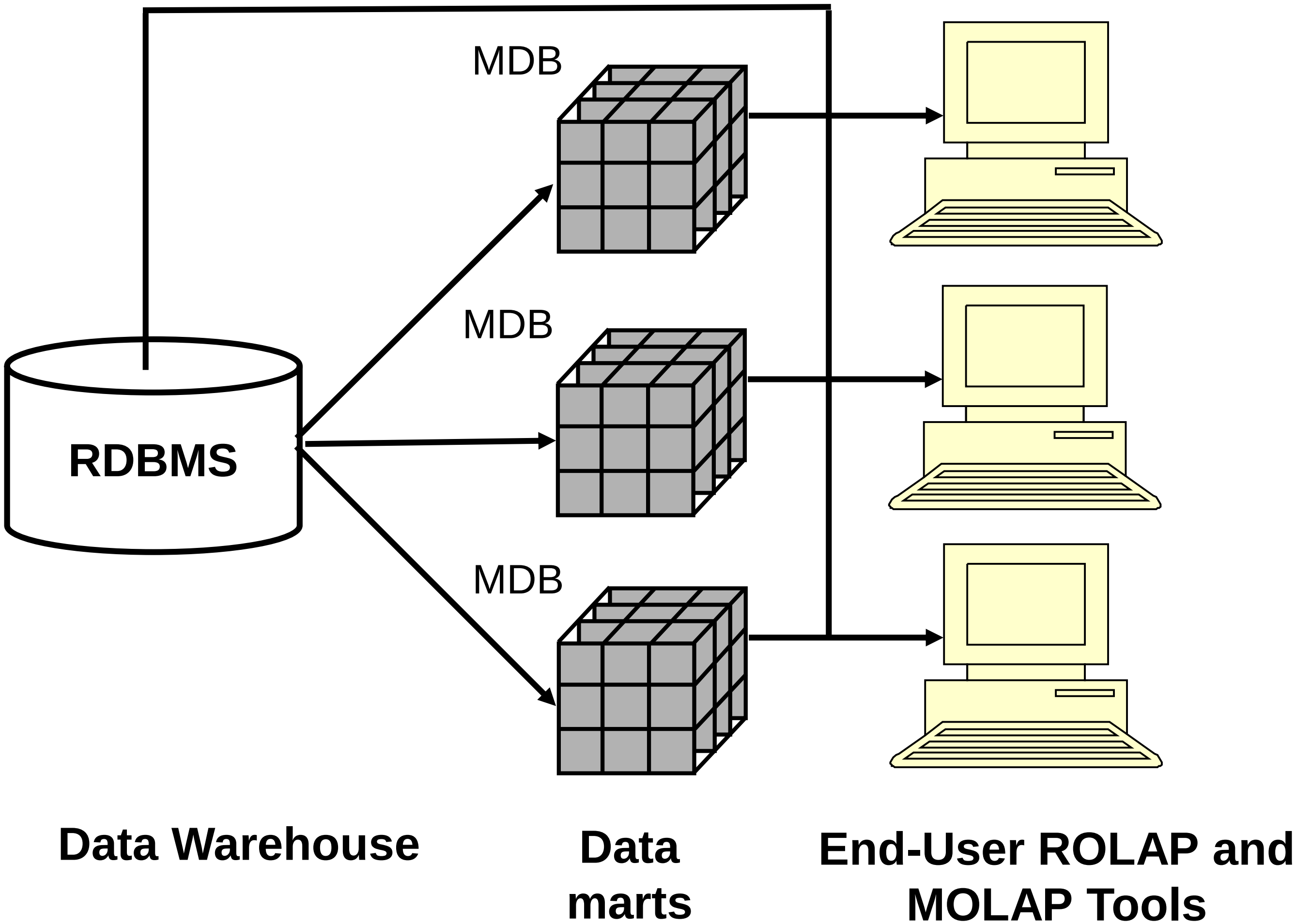
Central Data Warehouse

- The central Data Warehouse stores detail data (atomic transactions)
- It represents the enterprise wide source of consolidated data
- Data analysts use it to execute queries against detailed data (drill-down)
- It is used for enterprise wide data analysis and reporting
- But it is a separate database, **not the operational one**

Hybrid OLAP Data Mart Architecture

- Hybrid OLAP (HOLAP) combines elements of MOLAP and ROLAP data structures
- HOLAP keeps DW basic data in relational tables, and stores aggregated data in MOLAP structures
- Rational behind this approach is that relational structures are faster for large data volumes, and multidimensional structures are faster for small and medium data volumes
- HOLAP is implemented in Enterprise DW Architecture

Hybrid OLAP Data Mart Architecture



Shift in the DW Definition

- At the beginning of the 21st century the view on Data Warehousing started to shift
- Bill Inmon defined DW in 1993 as:
  - non-volatile,
  - time series,
  - subject oriented,
  - integrated data copies
  - used primarily for decision making
- The new view on DW describes it as:
  - near real-time,
  - event oriented,
  - business process oriented,
  - central source data
  - for use everywhere inside and outside an enterprise

Near Real - Time

- The time frame for doing business has decreased dramatically
- Instead of weekly or monthly updates, near real-time is requested to allow insight in the current status of the business
- So, the same transactions that update the operational databases are used to update the central detailed DW database

Event Data

- Traditionally, DSS were assumed to be based on summary information
- To day, when doing decision making, managers also need detailed, transaction level information

Business Process Oriented

- Early Data Warehouse idea was to build it using dimension data and facts, as summary data
- To day, data covering the whole business process are requested
- So, everyday business transactions are added to the Data Warehouse

Centralized Source Data

- The new Data Warehouse should contain all source transaction data for ad hoc reporting and auditing
- So, DW is the place to store integrated basic enterprise source data

For Use Inside and Outside the Company

- Initially, the Data Warehouse was aimed for use by managers
- Now, the Data Warehouse is considered as a source of data for everyone inside the enterprise and for the key business partners, as well
- The major philosophical change is that instead of using operational databases for querying and reporting, Data Warehouse, as a central source data repository should be used
- That way, all but data entry is removed from OLTP operational databases

OLAP on Cloud

- The ever increasing volume of data is the primary driver of implementing OLAP DBMSs and databases on a shared – nothing network architecture, since it scales the best
- Infrequent batch writes eliminate the need for complex distributed locking and commit protocols that makes **A**, **C**, and **I** of ACID easy to obtain
  - Batch writes are made by data that satisfy integrity constraints
- Data security is still an issue, but sensitive detailed data can be anonymized or encrypted
- Efficient processing of complex aggregate queries is also still an issue
- DW is considered as a good candidate application for implementation on cloud

An Outline of a DW Design Approach

- The basic principles:
  - Top – down design
    - The architecture of a whole future Data Warehouse
  - Bottom – up development and implementation
    - Step – by – step (data mart at a time) with gradual integration

Steps of the Top – Down Design Phase

- Design the long – term enterprise Data Warehouse architecture (on paper):
  - Use projected needs of multiple business units for DW facilities
  - Foresee a central Data Warehouse database used to store detailed, transaction level data
  - Foresee multiple data marts used to store detailed and aggregated data for individual business units
  - Foresee separate DW databases to avoid “virtual” DW problem (contention between OLTP and OLAP)
  - Foresee an ECTL tool to avoid “dirty” data problem
  - Foresee a meta data exchange architecture to avoid “stovepipe” data mart problem

Steps of the Bottom – Up Development

- Choose a business unit to develop the first data mart (pilot project):
  - Identify business drivers
  - Identify functional requirements:
    - Dimensions
    - Facts
    - Time granularity
  - Identify data sources
  - Choose an off-the-shelf ECTL tool
  - Choose a data_mart_in_a_box tool capable of supporting meta data exchange architecture
  - Model logical and physical structures of the data mart
  - Provide data models for both detailed and aggregated data
  - Choose the necessary hardware components
  - Provide for web access to the DW
  - Set the pilot project duration to 90 – 120 days

Steps of the Bottom – Up Development

- The pilot project is a proof of the concept
- So, it should prove viability of all decisions made in the planning phase:
  - From the data mart architecture to
  - The functionality of each single hardware and software component
- Lessons learned by the pilot project could also initiate some corrective actions

Steps of the Bottom – Up Development

- Following the completion of the pilot project, additional business units that require data marts may be identified
- The development procedure of these data marts should follow the procedure outlined for the first one with the exception that possible sharing of dimensions can be considered
- Finally, after completion of all individual data marts, expansion to an enterprise Data Warehouse can be made by moving all detailed data from data marts in a large central Data Warehouse (OLER)

Summary

- Some of the most frequent mistakes in building a DW are:
  - Missing business drivers,
  - Wrong Data Warehouse architectures, and
  - Top – down development of an enterprise Data Warehouse
- There are some Data Warehouse architectures that are considered wrong:
  - “Virtual” Data Warehouse,
  - Data Mart in a box (“dirty” data problem), and
  - “Stovepipe” data mart (lack of integration problem)
- Answers to theses problems are:
  - Multiple architected HOLAP data marts with a central detailed DW database, and
  - Top – down design with bottom – up development methodology
- OLAP is a classic db application that is considered to be candidate for a successful implementation on cloud