DATA WAREHOUSES - part4

1. **Introduction to the DWH discipline**
2. **Brief history** (Inmon &Linstedt,2015) **and Data Architecture** (Kimball&Ross,2016)
3. **Dimensional Modeling Fundamentals** (Kimball&Ross,2016)
4. **Technical Architecture Considerations** (Kimball&Ross,2016)
5. **Extract Transform Load and Data Quality** (Kimball&Ross,2016)
6. **DWH Lifecycle** (Kimball&Ross,2013)
7. **Trends in the evolution of DWH:**

   Extended RDBMS Architecture (Kimball&Ross,2013)**.**

   Pushing into the Future (Reeves,2009)**.**

   DWH 1.0 vs. 2.0 (Krishnan,2013)

Assessment

Type of activity	10.1 Assessment criteria	10.2 Assessment methods	10.3 Share of final grade
Developing support components for a DWH application prototype	Real-world application, complexity, validity and originality	Six face-to-face lab presentations of the homework about: the design of DWHcubes & DM models implemented, the support MDX & DMX queries and the code execution behind .NET forms in weeks No.: 3, 5, 7, 9, 11 and 13.	40% (6 * 6.66%)
Theoretical presentations during lecture hours (see those 14 themes on domains at section 8.1, pp.2)	Format, consistent pro-or-cons arguments, originality of comments and conclusions	Theoretical presentation and responses to questions	20%
Theoretical exam	Knowledge about DWH theory, real-world scenarios and personal implementations	Final theoretical test with at least three open questions	40%

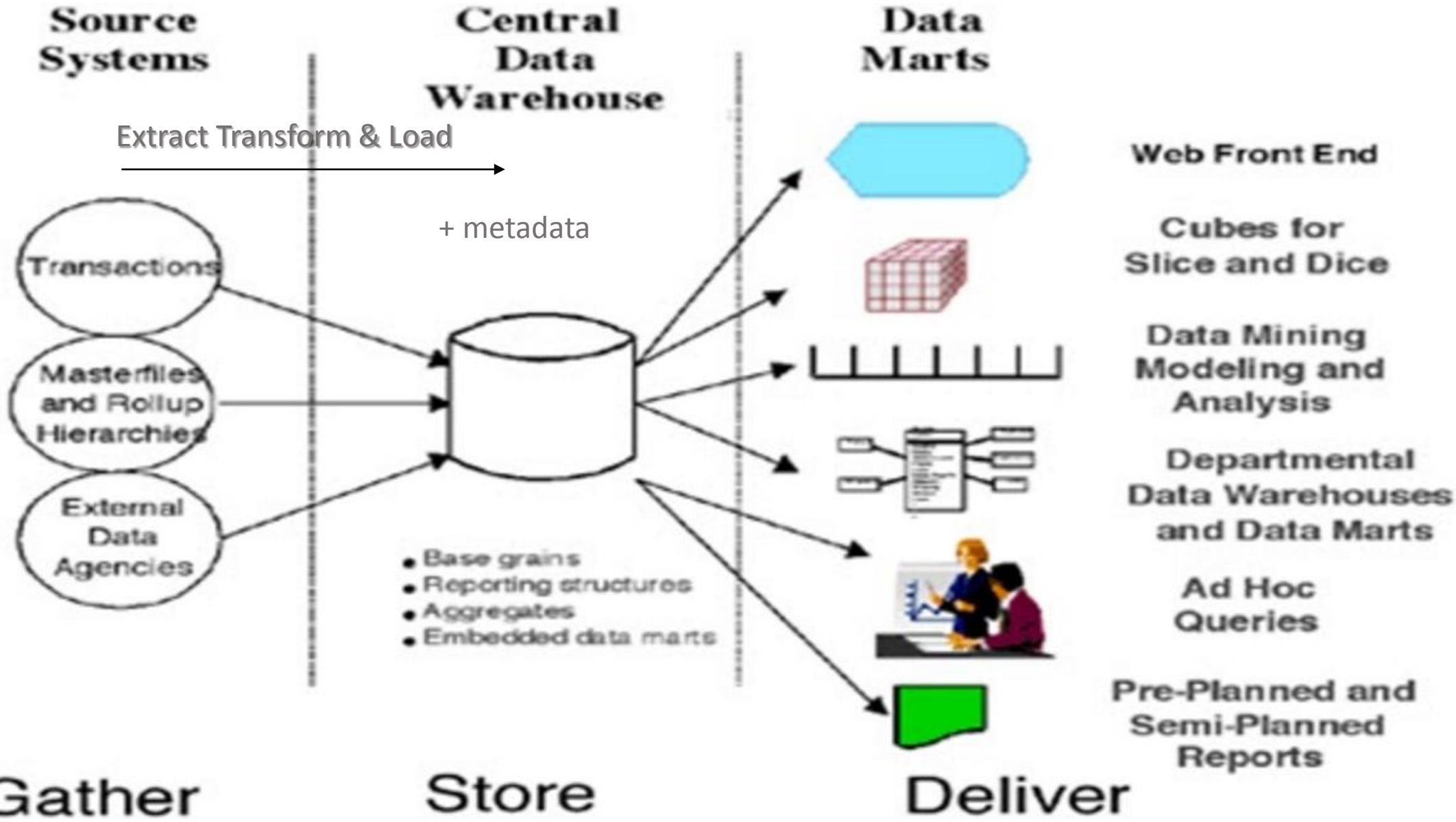
10.6 Minimum performance standard

- Design and implement a DWH cube, a DM model, two interfaces to programmatically connect to and query them by using MDX and DMX queries, alternative examples (user mode) with Microsoft Excel add-ins;
- Each master student must create, present and answer questions for a part of those at least 10 specific presentations (14 themes on domains - section 8.1, pp.2) during lecture hours;
- The average grade for all those six face-to-face lab presentations (section 8.2, pp.3) of the homework is grater than or equal to 5;
- The grade for the final theoretical test must be greater than or equal to 5.

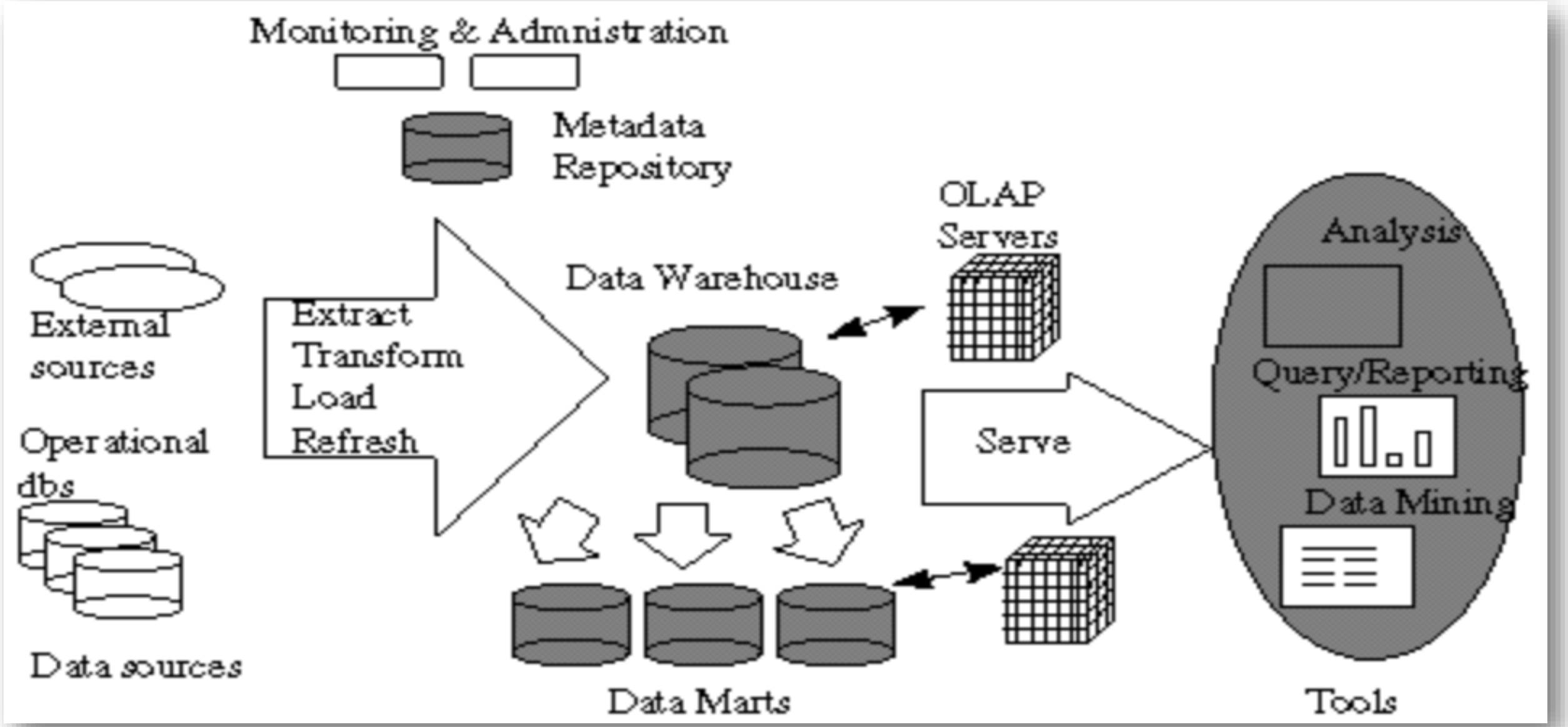
Planning those at least 10 presentations (course hours, 20% in final grades) of master students (teams) about:
(1.) retail sales, (2.) inventory, (3.) procurement, (4.) order management, (5.) accounting, (6.) CRM, (7.) HRM,
(8.) financial services, (9.) telecommunications,
(10.) transportation, (11.) education, (12.) healthcare,
(13.) e-commerce, (14.) insurance
details: **Kimball&Ross,2013**.
by **e-mail**: dan.homocianu@gmail.com
(**subject**: *DWH course pres. theme TITLE.. on DATE..*
content: *team: MEMBER1, MEMBER2, MEMBER3*)
Each master student (team member) must present a part of his team's support presentation for at least 5-10 minutes
and answer questions.

DWH – overview

- *Data mart* has been replaced with *business process dimensional model*, *business process subject area*, or just *subject area*, depending on the context. (Kimball&Ross,2016)



DWH – overview



http://redbook.cs.berkeley.edu/redbook3/lec28.html

DWH & SOA

Kimball &Ross,2016

The service oriented architecture (SOA) movement has captured the imagination, if not the budgets, of many IT departments. In a nutshell, organizing your environment around SOA means identifying reusable services, and implementing these services as centralized resources typically accessed over the web. The appeal comes from the promised cost savings of implementing a service only once in a large organization, and making the service independent of specific hardware and operating system platforms because all communications take place via a neutral communications protocol, most often WSDL-SOAP-XML.

DWH & SOA

Kimball &Ross,2016

Well, pretty much all of these advantages of SOA can be realized, but early SOA pioneers have learned some valuable lessons that give one pause for thought. The names of these lessons are data quality, data integration, and governance. To make a long story short, SOA initiatives fail when they 1) sit on a platform of poor-quality data, 2) attempt to share data that is not integrated across the enterprise, and 3) are implemented with insufficient thought given to security, compliance, and change management. SOA architects have also learned to avoid the overly detailed use case. Services meet the goals of SOA architecture when they are simple, conservative in their scope, and not dependent on complex business rules of an underlying application.

DWH & SOA

Kimball &Ross,2016

So, does SOA have anything to offer data warehousing? Can we identify abstract services in the data warehouse world, commonly recognized, simply stated, and independent of specific data sources, business processes, and BI deployments? I think we can.

Consider the relationship between the dimension manager and fact provider. Remember that a dimension manager is the centralized resource for defining and publishing a conformed dimension to the rest of the enterprise. A master data management (MDM) resource is an ideal dimension manager. but few of us are lucky enough to have a functioning MDM resource More likely, the data warehouse team is a kind of downstream MDM function that gathers incompatible descriptions of an entity such as customer and publishes the cleaned, conformed, and deduplicated dimension to the rest of the data warehouse community. The subscribers to this dimension are almost always owners of fact tables who want to attach this high quality conformed dimension to their fact tables so that BI tools around the enterprise can perform drill-across reports on the conformed contents of the dimension.

DWH & SOA

Kimball &Ross,2016

Every dimension manager publisher needs to provide the following services to their fact table subscribers. For these services, *fetch* means that the fact table provider pulls information from the dimension manager, and *alert* means that the dimension manager pushes information to the fact providers.

- Fetch specific dimension member. We assume in this and the following steps that the dimension record has a surrogate primary key, a dimension version number, and that the information transmitted is consistent with the security and privacy privileges of the requester.
- Fetch all dimension members.
- Fetch dimension members changed since specific date/time with designated SCD types 1, 2, and 3.
- Fetch natural key to surrogate key correspondence table unique to fact table provider.
- Alert providers of new dimension release. A major dimension release requires providers to update their dimensions because type 1 or type 3 changes have been made to selected attributes.
- Alert providers to late-arriving dimension members. This requires fact table providers to overwrite selected foreign keys in fact tables.

DWH & MDM

Kimball &Ross,2016

It's not enough to keep track of the correct current value of an attribute. Analytic needs and compliance reporting demand tracking of historical changes to the MDM function. This requirement makes MDM more complex but doesn't fundamentally change what needs to happen, or how.

There are three common approaches to creating MDM systems:

1. **Let the data warehouse do it.** This approach is the least disruptive to existing systems, but it has hidden costs and limitations.
2. **Create an operational MDM to integrate data from multiple sources on a transactional basis.** This approach surfaces the hidden costs and improves timing and accessibility to the master data, but it still leaves multiple, disparate copies of data in various transaction systems.
3. **Create an enterprise MDM system of record for transaction systems data and facilitate data sharing as needed.**

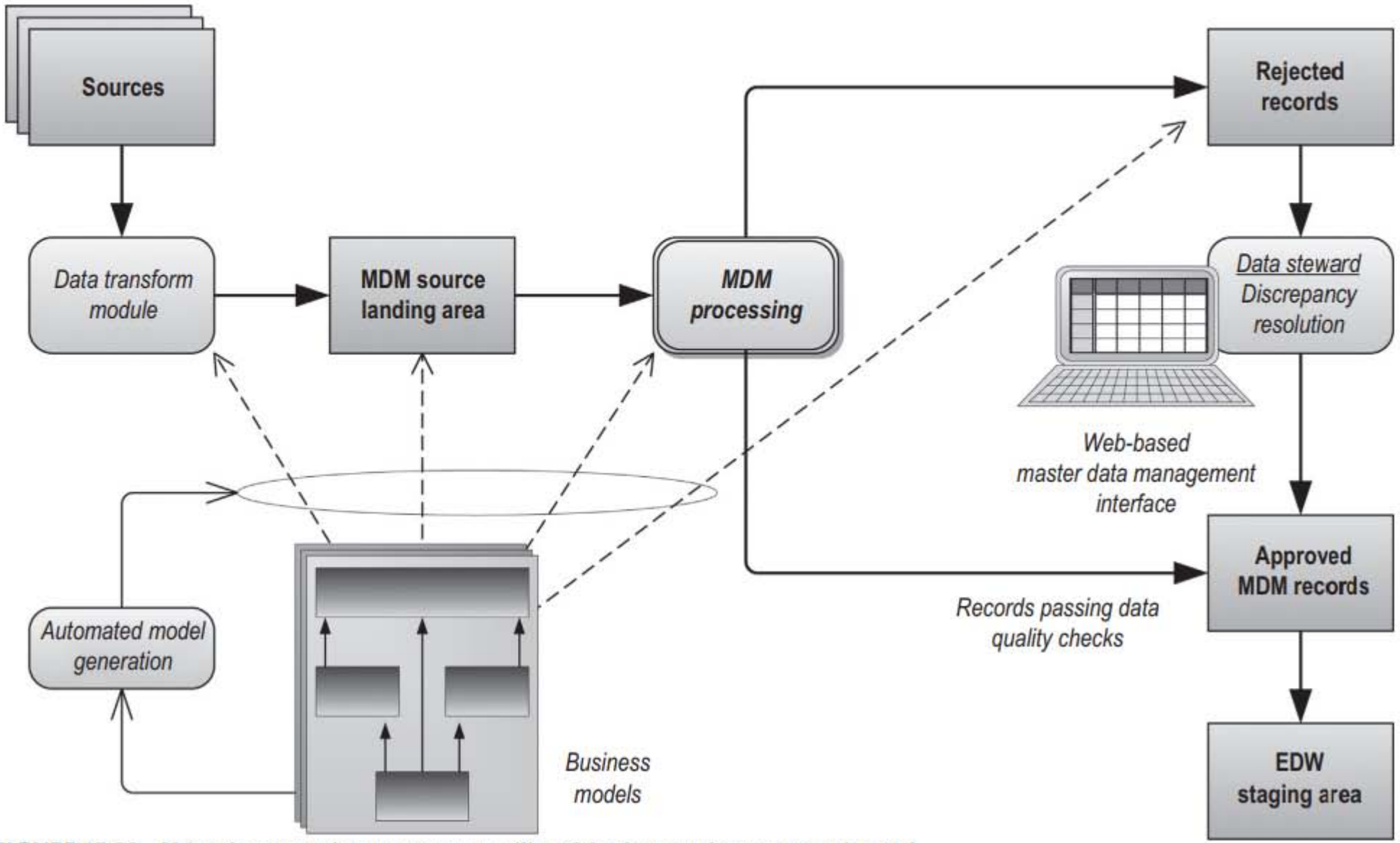


FIGURE 15.22 Using the master data management utility of the data warehouse automation tool.

DWH & MDM

Kimball &Ross,2016

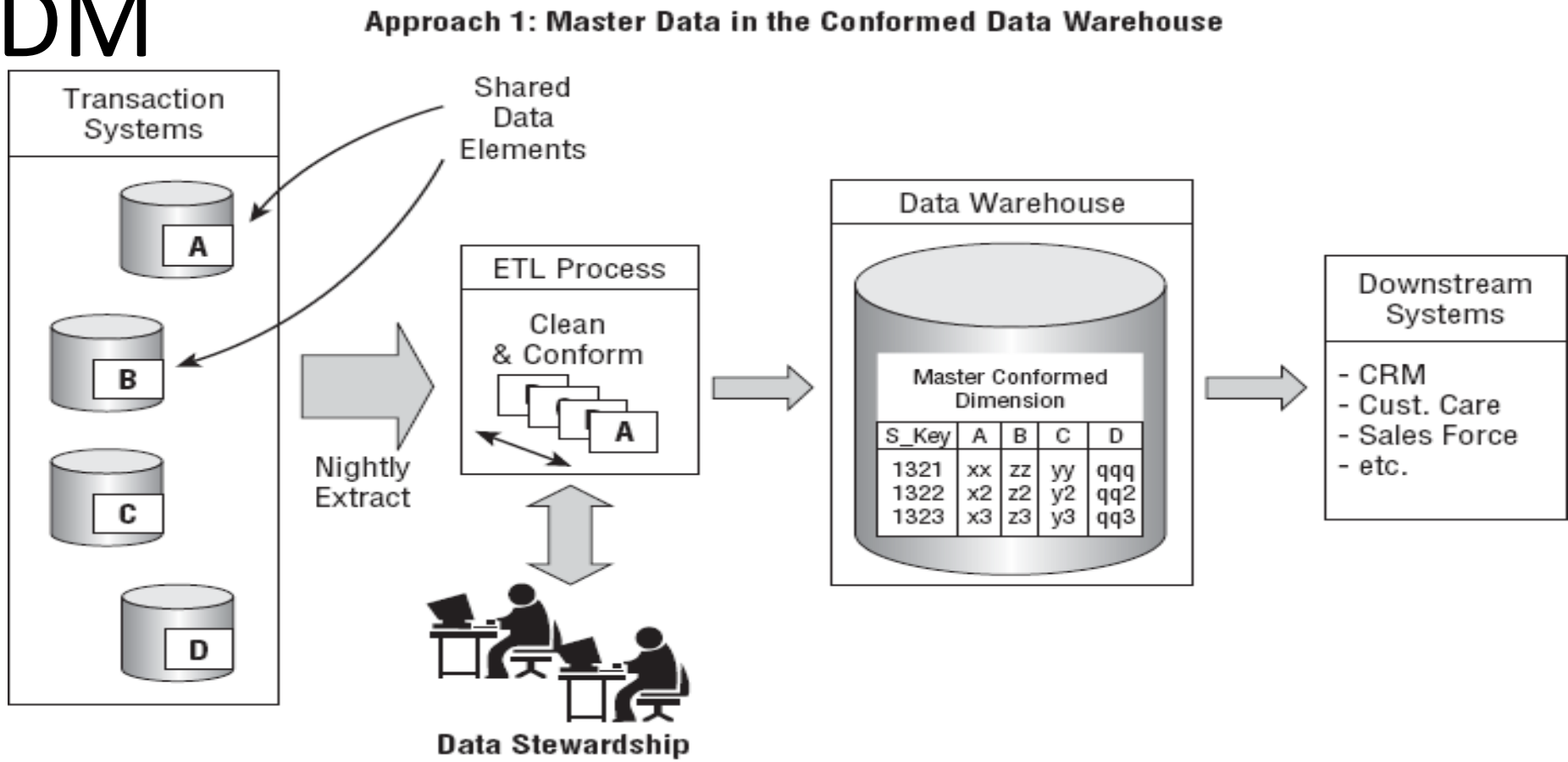


Figure 12-1: Approach 1: Managing master data in the data warehouse ETL.

The goal of creating a true enterprise understanding of the business has forced the data warehouse to deal with disparate sources. In the 1990s, most transaction systems managers did not care about data integration because it was not a business priority. Their top priority was meeting the specific transaction needs as quickly as possible. The idea that the transaction systems should create and manage a centralized master data set was considered a grand vision at best, but not practical; more often, it was considered a good joke.

DWH & MDM

Kimball &Ross,2016

The master conformed dimension contains attributes that are common across transaction systems, such as customer name, and attributes that are unique to individual systems, such as ship-to address. The master conformed dimension maps the source system keys to a single master surrogate key for each entry in the entity. This mapping is the Rosetta stone that ties data from one source system to another.

Once the master dimension is in place, the effort does not end. Changes and additions to the shared attributes from different source systems must be entered into the dimension. Also, the dimension management system is usually called upon to identify the best version of a shared attribute out of all possible versions that exist. This determination, known as *survivorship*, involves complex business rules and comparisons during the ETL pipeline.

To many, having the data warehouse create and manage master data feels wrong because it flies in the face of the fundamental tenets of quality. In the 1950s, Edward Deming and his counterparts taught us that you must fix quality problems at their source or you will be doomed to fix them forever. Every night, the ETL process will push its Sisyphean rock up to the top of the hill, only to have it knocked back down by the transaction systems the next day. Unfortunately, a majority of organizations have been unwilling to address the problem any other way, so we had to do it in the data warehouse.

DWH & MDM

Kimball &Ross,2016

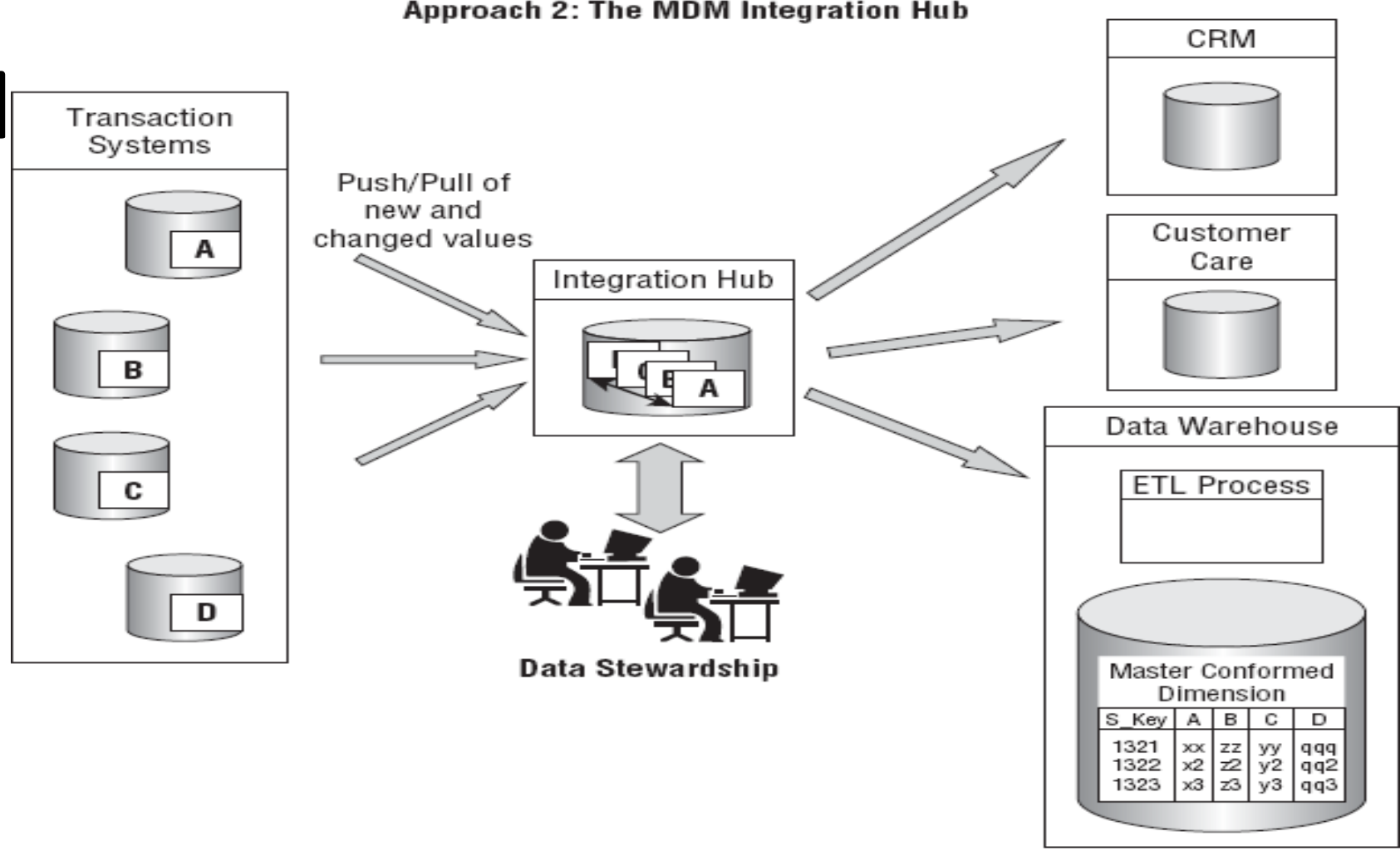


Figure 12-2: Approach 2: Implementing MDM via an integration hub.

Most organizations have multiple customer touch points, and they often have a separate customer relationship management (CRM) system and a web-based customer interface that keeps some of its own customer data. Companies that want to address the problem of disparate data might begin by leaving each silo transaction data store in place while building a centralized master data store. A copy of all the attributes is placed in the MDM system, and all future updates are replicated into the MDM system from each source. As a result, the MDM system will contain multiple versions of the same attributes as they are collected by the various source systems.

DWH & MDM

Kimball &Ross,2016

This MDM approach usually generates the same valuable outputs as the data warehouse version: the source key mapping table, and the best, or master, value for each attribute (equivalent to the survivorship function in the ETL process) based on complex rules that assess the accuracy and validity of each source. These rules include an assessment of the relative accuracy of the different sources, how recently the source has been updated, where the update came from, and so on. The data steward is responsible for creating and maintaining these rules through the MDM interface.

The integration hub approach does not attempt to fix the problem at the source; it simply moves the cleaning, aligning, standardizing, and integration upstream a bit. The integration hub is better than the data warehouse approach because it can operate at transaction service levels, letting some of the operational systems make calls to the integration hub to get the most current value of an attribute. It also becomes a data provider for downstream systems, like the data warehouse. At last, the data warehouse can stop pushing the rock up the hill. Applying this approach to customer data is essentially what customer data integration (CDI) is all about.

DWH & MDM

Kimball &Ross,2016

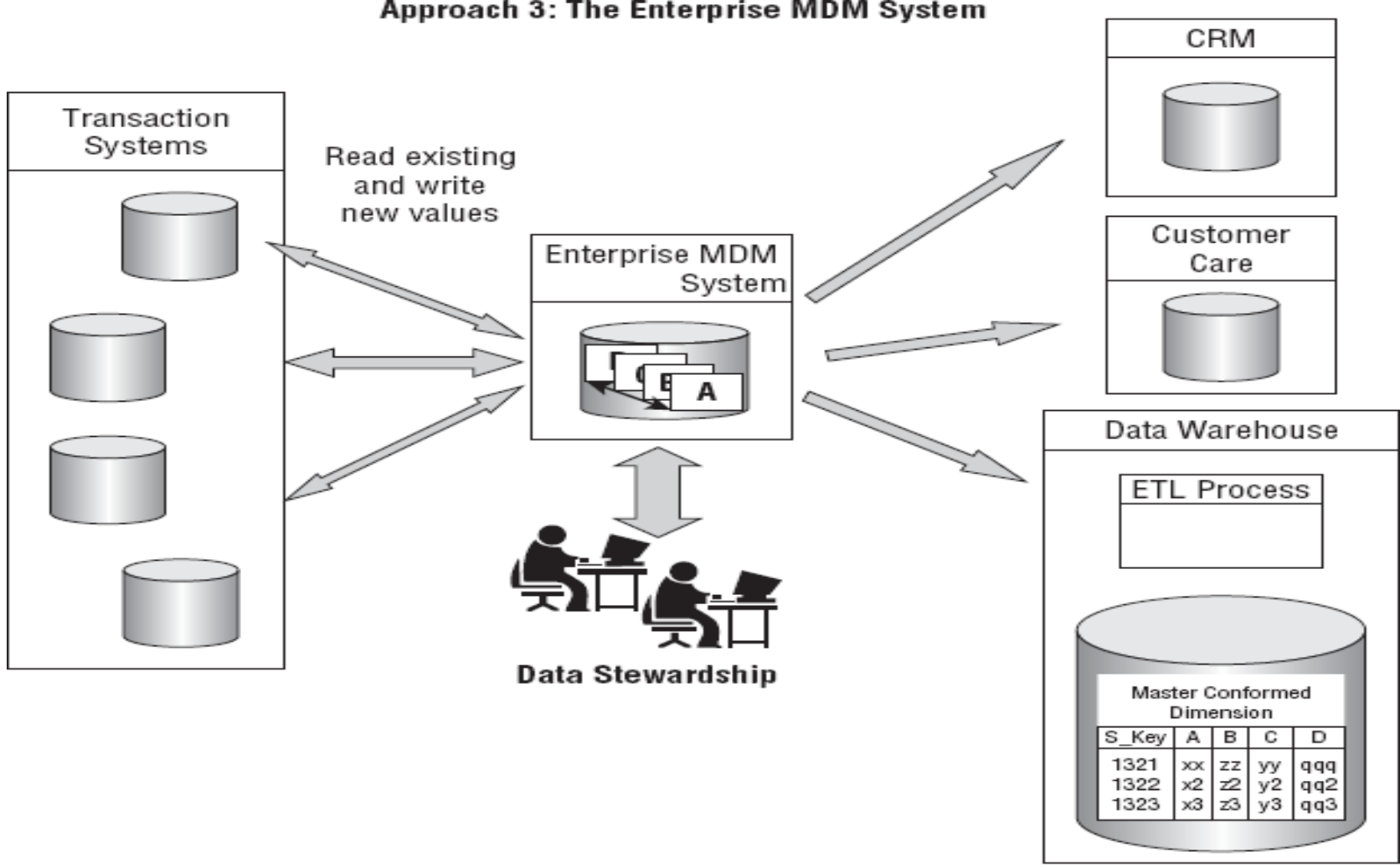


Figure 12-3: Approach 3: The enterprise MDM system.

The third approach, shown in Figure 12-3, is to create a centralized database that holds the master version of all attributes of a given object along with the interface functions needed to tie this master data back to the transaction systems. In other words, the master database serves as the data store of record for all participating transaction processes. Rather than having each system maintain its own customer data, transaction systems all create, query, and update attributes in the centralized MDM.

DWH & MDM

Kimball &Ross,2016

The user-facing portion of each system must first identify the exact customer or product involved in a given transaction and obtain its unique key from the MDM. The MDM must keep all attributes that are (or might be) required by more than one system and must also allow each system to define its own custom attributes. In addition, master data systems must have standard APIs available to all client systems.

The enterprise MDM does not solve the problem of disparate data by itself. One system can define an attribute that is the equivalent of an already existing attribute. The data steward must be responsible for each master data object, and have cross-system responsibility for uniquely defining each attribute and monitoring the usage of those attributes. As the Kimball Group has long espoused, strong, centralized data governance is the mandatory foundation for successfully creating information that's conformed across the enterprise.

This single, centralized database is obviously the ideal solution, but it requires every transaction system that shares data to give up ownership of shared data sets. This is not trivial because it requires rewriting core parts of each system, or at least redirecting the data requests to the external MDM system. On the other hand, the enterprise MDM is the ideal place to start a service oriented architecture (SOA) deployment because a large number of clients could take advantage of the obvious reusability of the service. From what we've seen, far fewer than 10 percent of organizations have an enterprise MDM system in place.

Web-intensive DWH

Kimball &Ross,2016

A web-intensive data warehouse must be:

- **Fully web deployed**, with every user and administrative capability usable through a standard web browser from any location.
- **Historically accurate up to the moment**, so that the longer view of history extends seamlessly to include the order a customer placed five minutes ago.
- **Profoundly distributed**, so that internal business processes, external data sources, and the databases of far-flung suppliers and customers are all part of the "enterprise" data warehouse.
- **Dynamically changing**, so that as we venture into new lines of business, and as we initiate contracts with new suppliers, the data warehouse smoothly accommodates the new data types and interfaces.

DWH & ERPs

Kimball &Ross,2016

In spite of ERP systems' basic constructive approach, until now they have had a minor, disappointing effect on the data warehouse world, for several reasons:

- ERP systems have been far more concerned with transaction processing than decision support. Data warehousing has been a late arrival and an afterthought for most ERP vendors. Many ERP vendors have a back room culture that doesn't validate the needs of front room business users.
- ERP systems' primary database schemas are absurdly complex, including thousands of database tables. Although it is possible to extract data from an ERP system into a data warehouse, it requires specialized knowledge and powerful extraction software.
- ERP vendors have introduced data warehouse environments within their product offerings, but have not until now convincingly demonstrated their willingness to openly import and export data when that is what the customer desires.

DWH & ERPs

Kimball &Ross,2016

Any enterprise data warehouse we implement within an ERP environment can only be successful if the ERP vendor makes a full commitment to *open data warehousing* with:

- Competitive level of query performance, equal to the best performance available from an extracted, independent, and aggregate-aware set of data warehouse tables
- Competitive level of user interface excellence, and ease of use if the ERP vendor expects the application to use its own tools
- Full data import interfaces that encourage the easy integration of foreign data sources, including syndicated data suppliers, legacy applications, packaged applications, and competitor ERP systems
- Full data export interfaces that let the customer extract the ERP data to an independent data warehouse if that is what the customer wants to do
- Flexible attachment of third party query and reporting tools directly to the ERP vendor's data warehouse tables
- Flexible attachment of external transaction processing applications and other ERP systems to the vendor's operational and/or data warehouse data through extensible markup language (XML) interfaces, thereby sending the clear message that it is OK for us to view the ERP vendor's system as part of a larger system
- Seamless connection of current status reporting to historical record reporting
- Seamless transition between analysis (data warehousing) and update (transaction processing)
- Seamless integration of standardized facts and dimensions with local, user-defined facts and dimensions

DWH & SMART Apps

Kimball &Ross,2016

A better long term approach is to build on an enterprise data warehouse, an information infrastructure that:

- Is focused on the business users' requirements
- Is enterprise wide, containing integrated, conformed information from multiple business processes
- Contains data at the finest granularity possible
- Defines standard attributes, hierarchies, and structures that are used across multiple business processes
- Includes attributes, such as customer income levels, that can be used to predict behavior
- Implements change-tracking techniques on key attributes to associate behaviors such as purchases with the attribute's value at the time the behavior occurred
- Identifies data quality problems and works to fix them at the source or in the ETL process
- Delivers standard reports and analytic applications to the enterprise
- Includes a software, hardware, and infrastructure environment that can support all this data, transformation, reporting, and analysis at the enterprise level

DWH & Big Data

Kimball &Ross,2016

In the coming 2010s decade, the analysis of big data will require a technology or combination of technologies capable of:

- Scaling to easily support petabytes (thousands of terabytes) of data
- Being distributed across thousands of processors, potentially geographically unaware, and potentially heterogeneous
- Subsecond response time for highly constrained standard SQL queries
- Embedding arbitrarily complex user-defined functions (UDFs) within processing requests
- Implementing UDFs in a wide variety of industry-standard procedural languages
- Assembling extensive libraries of reusable UDFs crossing most or all of use cases
- Executing UDFs as "relation scans" over petabyte-sized data sets in a few minutes
- Supporting a wide variety of data types growing to include images, waveforms, arbitrarily hierarchical data structures, and data bags
- Loading data to be ready for analysis, at very high rates, at least gigabytes per second
- Integrating data from multiple sources during the load process at very high rates (GB/sec)
- Loading data before declaring or discovering its structure
- Executing certain "streaming" analytic queries in real time on incoming load data
- Updating data in place at full load speeds
- Joining a billion row dimension table to a trillion row fact table without pre-clustering the dimension table with the fact table
- Scheduling and execution of complex multi-hundred node workflows
- Being configured without being subject to a single point of failure
- Failover and process continuation when processing nodes fail
- Supporting extreme mixed workloads including thousands of geographically dispersed online users and programs executing a variety of requests ranging from ad hoc queries to strategic analysis, and while loading data in batch and streaming fashion

DWH & Extended RDBMS architecture

Kimball &Ross,2016

All of the major relational database management system vendors are adding features to address big data analytics from a solid relational perspective. The two most significant architectural developments have been the overtaking of the high end of the market with massively parallel processing (MPP), and the growing adoption of columnar storage. When MPP and columnar storage techniques are combined, a number of the system requirements in the preceding list can start to be addressed, including:

- Scaling to support exabytes (thousands of petabytes) of data
- Being distributed across tens of thousands of geographically dispersed processors
- Subsecond response time for highly constrained standard SQL queries
- Updating data in place at full load speeds
- Being configured without being subject to a single point of failure
- Failover and process continuation when processing nodes fail

DWH & Extended RDBMS architecture

Kimball &Ross,2016

Additionally, RDBMS vendors are adding some complex user-defined functions (UDFs) to their syntax, but the kind of general-purpose procedural language computing required by big data analytics is not being satisfied in relational environments at this time.

In a similar vein, RDBMS vendors are allowing complex data structures to be stored in individual fields. These kinds of embedded complex data structures have been known as "blobs" for many years. It's important to understand that relational databases have a hard time providing general support for interpreting blobs since blobs do not fit the relational paradigm. An RDBMS indeed provides some value by hosting the blobs in a structured framework, but much of the complex interpretation and computation on the blobs must be done with specially crafted UDFs, or BI application layer clients. Blobs are related to "data bags" discussed earlier.

DWH & Extended RDBMS architecture

Kimball &Ross,2016

MPP implementations have never satisfactorily addressed the "big join" issue where an attempt is made to join a billion row dimension table to a trillion row fact table without resorting to clustered storage. The big join crisis occurs when an ad hoc constraint is placed against the dimension table resulting in a potentially very large set of dimension keys that must be physically downloaded into every one of the physical segments of the trillion row fact table stored separately in the MPP system. Since the dimension keys are scattered randomly across the separate segments of the trillion row fact table, it is very hard to avoid a lengthy download step of the very large dimension table to every one of the fact table storage partitions. To be fair, the Hadoop architecture has not been able to address the big join problem either.

DWH & Extended RDBMS architecture

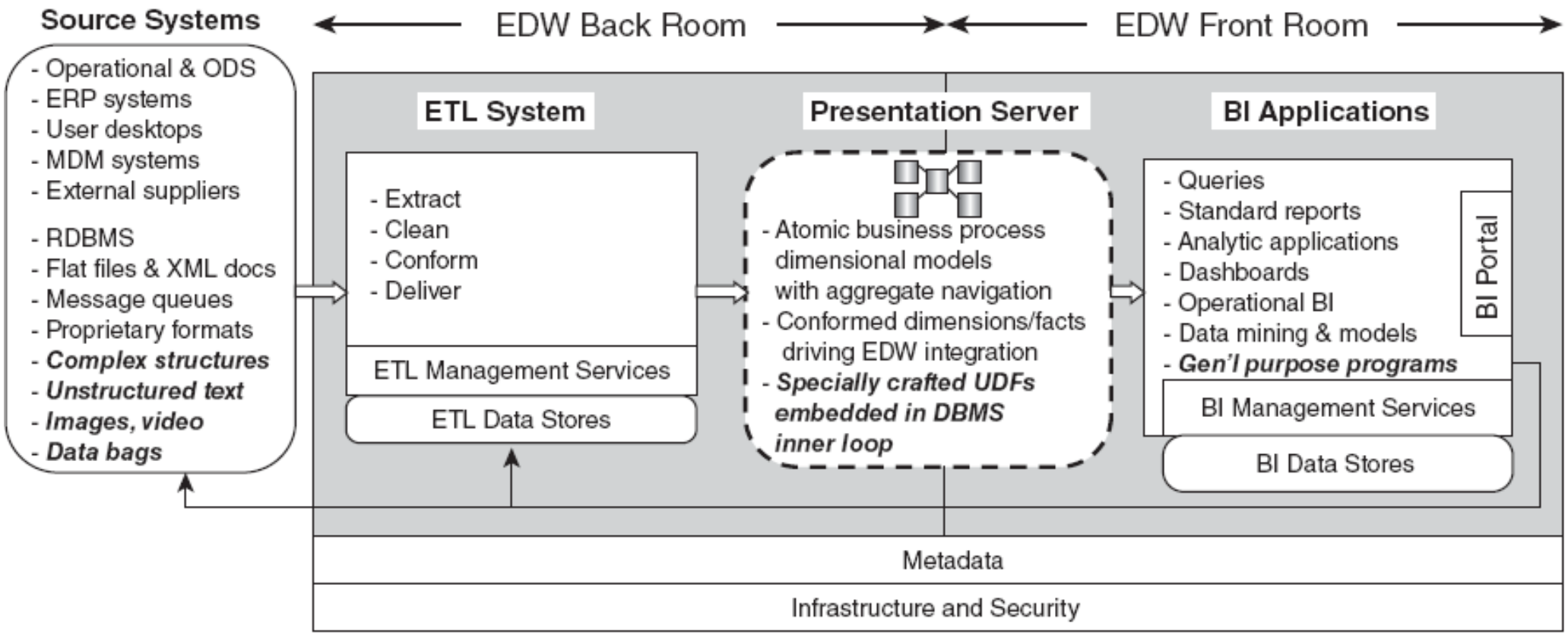
Kimball &Ross,2016

Columnar data storage fits the relational paradigm, and especially dimensionally modeled databases, very well. Besides the significant advantage of high compression of sparse data, columnar databases allow a very large number of columns compared to row-oriented databases, and place little overhead on the system when columns are added to an existing schema. The most significant Achilles' heel, at least in 2015, is the slow loading speed of data into the columnar format. Although impressive load speed improvements are being announced by columnar database vendors, they have still not achieved the gigabytes-per-second requirement.

DWH & Extended RDBMS architecture

Kimball &Ross,2016

The extended RDBMS architecture to support big data analytics preserves the familiar data warehouse architecture with a number of important additions, shown in Figure 12-4 with **bold text**:



user-defined functions (UDFs)

Figure 12-4: The extended RDBMS–based architecture for an enterprise data warehouse.

DWH & Extended RDBMS architecture

Kimball &Ross,2016

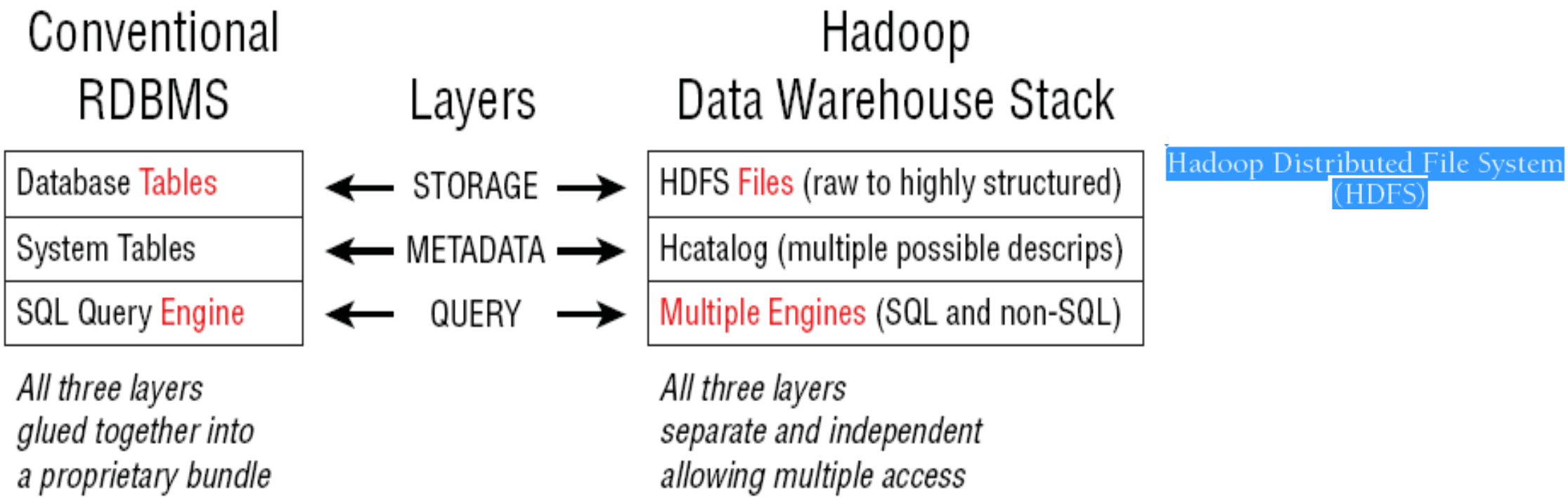


Figure 12-5: Comparison of the storage, metadata, and query layers between conventional RDBMSs and the Hadoop data warehouse stack.

DWH & Extended RDBMS architecture

Kimball &Ross,2016

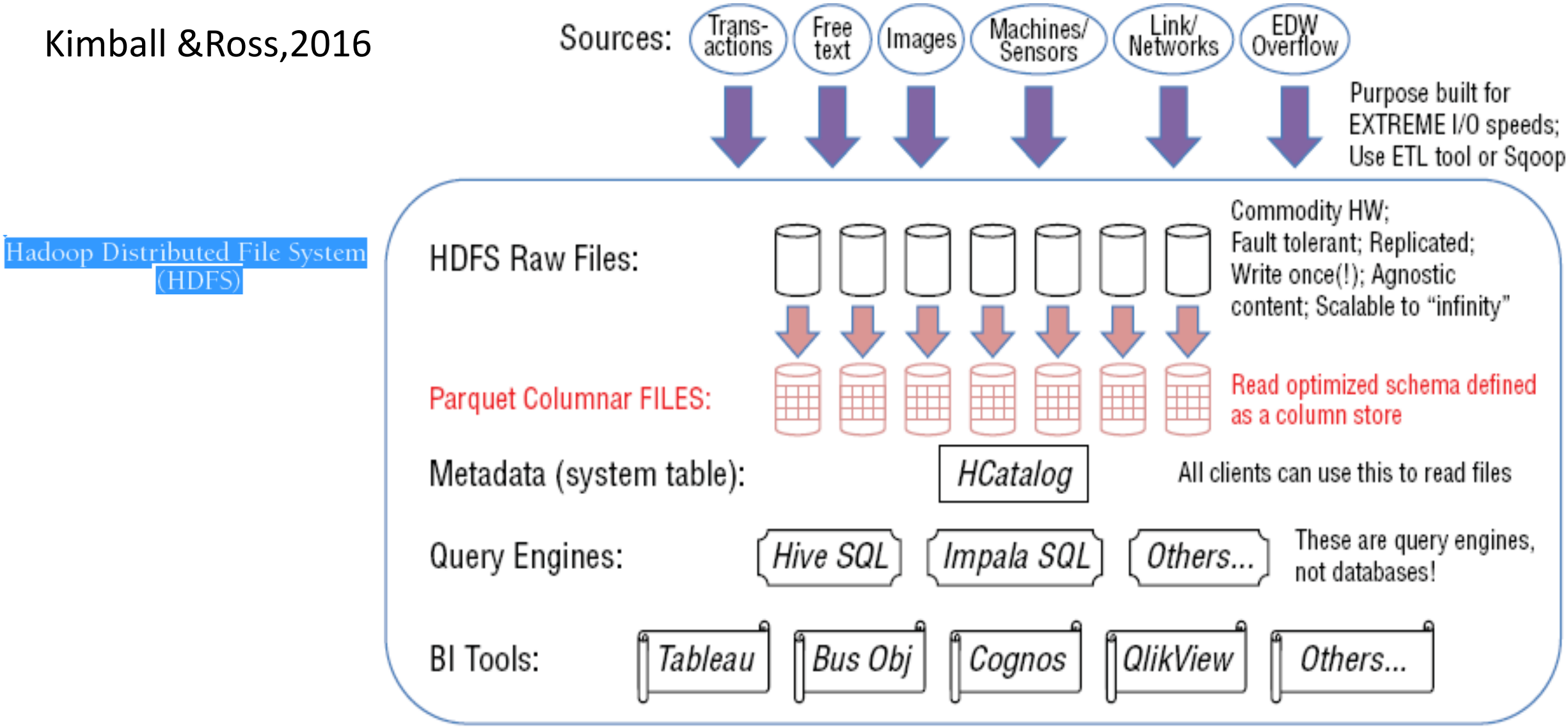


Figure 12-6: HDFS viewed as a data warehouse environment.

The Zachman Framework for Enterprise Architecture™

The Enterprise Ontology™

Version 3.0

Classification Names / Audience Perspectives	What	How	Where	Who	When	Why	Classification Names / Model Names
	Composite Integrations →		← Alignment →		← Composite Integrations		
Executive Perspective (Business Context Planners)	Inventory Identification e.g. List: Inventory Types	Process Identification e.g. List: Process Types	Distribution Identification e.g. List: Distribution Types	Responsibility Identification e.g. List: Responsibility Types	Timing Identification e.g. List: Timing Types	Motivation Identification e.g. List: Motivation Types	Scope Contexts (Scope Identification Lists)
Business Mgmt Perspective (Business Concept Owners)	Inventory Definition Business Entity Business Relationship	Process Definition Business Transform Business Input/Output	Distribution Definition Business Location Business Connection	Responsibility Definition Business Role Business Work Product	Timing Definition Business Interval Business Moment	Motivation Definition Business End Business Means	Business Concepts (Business Definition Models)
Architect Perspective (Business Logic Designers)	Inventory Representation System Entity System Relationship	Process Representation System Transform System Input /Output	Distribution Representation System Location System Connection	Responsibility Representation System Role System Work Product	Timing Representation System Interval System Moment	Motivation Representation System End System Means	System Logic (System Representation Models)
Engineer Perspective (Business Physics Builders)	Inventory Specification Technology Entity Technology Relationship	Process Specification Technology Transform Technology Input /Output	Distribution Specification Technology Location Technology Connection	Responsibility Specification Technology Role Technology Work Product	Timing Specification Technology Interval Technology Moment	Motivation Specification Technology End Technology Means	Technology Physics (Technology Specification Models)
Technician Perspective (Business Component Implementers)	Inventory Configuration Tool Entity Tool Relationship	Process Configuration Tool Transform Tool Input /Output	Distribution Configuration Tool Location Tool Connection	Responsibility Configuration Tool Role Tool Work Product	Timing Configuration Tool Interval Tool Moment	Motivation Configuration Tool End Tool Means	Tool Components (Tool Configuration Models)
Enterprise Perspective (Users) The Enterprise	Inventory Instantiations Operations Entities Operations Relationships	Process Instantiations Operations Transforms Operations In/Outputs	Distribution Instantiations Operations Locations Operations Connections	Responsibility Instantiations Operations Roles Operations Work Products	Timing Instantiations Operations Intervals Operations Moments	Motivation Instantiations Operations Ends Operations Means	Operations Instances (Implementations) The Enterprise
Audience Perspectives / Enterprise Names	Composite Integrations →		← Alignment →		← Composite Integrations		
	Inventory Sets	Process Flows	Distribution Networks	Responsibility Assignments	Timing Cycles	Motivation Intentions	

*Horizontal integration lines are shown for example purposes only and are not a complete set. Composite, integrative relationships connecting every cell horizontally potentially exist.

FIGURE 4.6 Zachman framework adapted for an enterprise data warehousing program. *Adapted from the Zachman Framework, 2001 version, as presented in [DMDict] and [Session 2007].*

Bibliography

-Airinei, D., Depozite de date, Polirom, Iaşi, 2002
portal.feaa.uaic.ro/Master/sia/Pages/default.aspx
-Airinei, D., Dospinescu, O., Huiban, A., Aplicatii practice cu sisteme OLAP si Depozite de date, Editura Sedcom Libris, Iaşi, 2008
-Homocianu, D., Sistemele de asistare a deciziilor in contextual societatii cunoasterii, Editura UAIC, Iasi, 2009 (ssrn.com/abstract=2384380)
-Inmon,W.H.,Linstedt, D., Data Architecture: A primer for the data scientist, MK, MA, 2015 (tinyurl.com/h7dcq66)
-Kimball, R., Ross, M., The Data Warehouse Toolkit Third Edition. The Definitive Guide to Dimensional Modeling, John Wiley & Sons, New York, 2013 (tinyurl.com/jogo5uy)
-Kimball, R., Ross, M., The Kimball Group Reader. Relentlessly Practical Tools for Data Warehousing and Business Intelligence – Remastered Collection, Second Edition, Wiley, New York, 2016 (tinyurl.com/johox4v)
-Krishnan, K, Data Warehousing in The Age of Big Data, Morgan Kaufmann (MK), MA, 2013 (tinyurl.com/zrjo75j)
-Reeves, L.L., A Manager's Guide to Data Warehousing, Wiley, New York, 2009 (tinyurl.com/htsslk8)
-Sarka, D., et. al., Implementing a Data Warehouse with Microsoft SQL Server 2012. Training Kit, O'Reilly Media, Sebastopol, 2012 (tinyurl.com/jk6lclk)
-Sheldon, B., et. al., Professional Visual Basic 2012 and .NET 4.5 Programming, John Wiley & Sons, Indianapolis, 2013 (tinyurl.com/hrb6j9u)