DATA WAREHOUSES - part5

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%

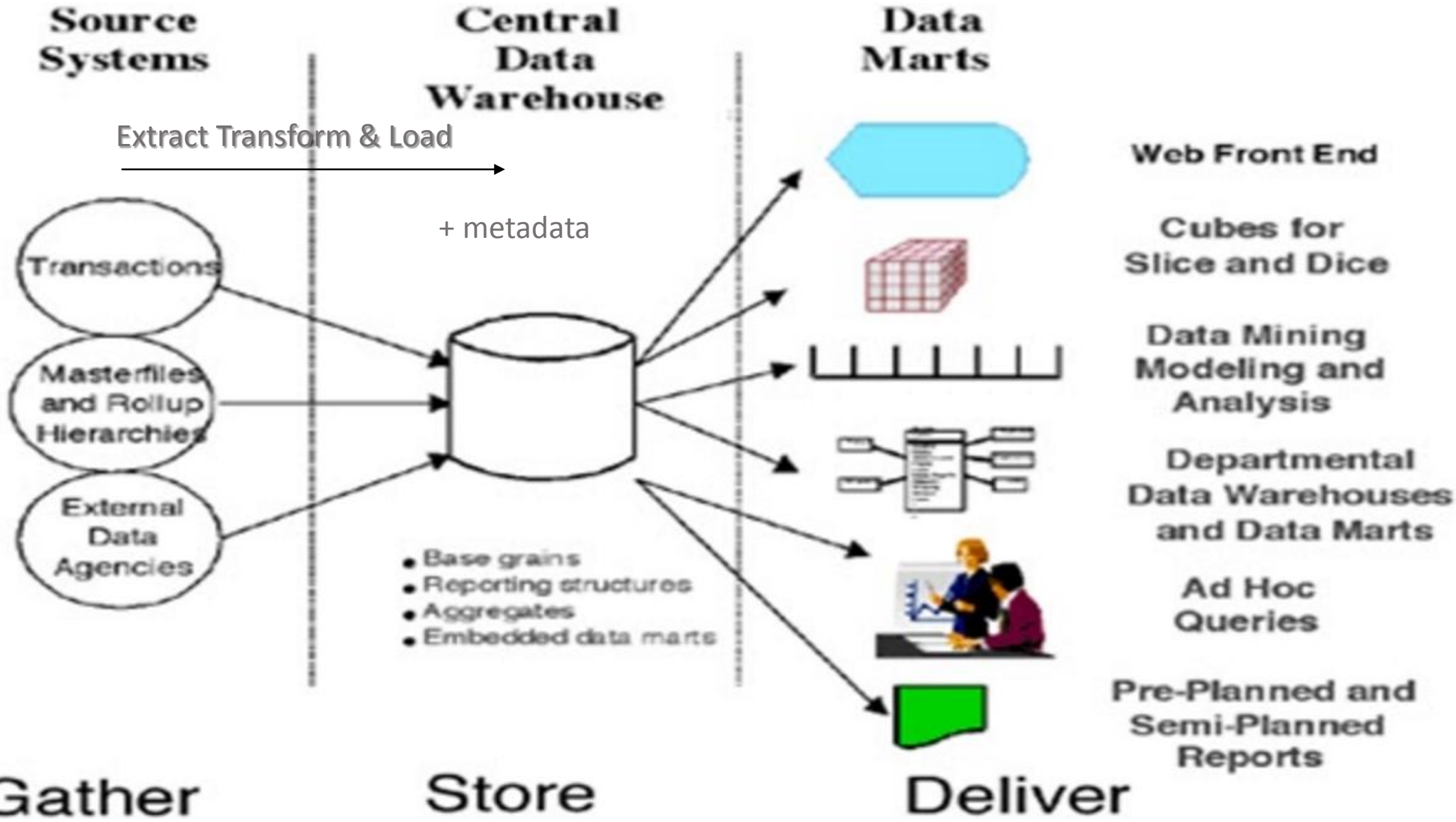
10.6 Minimum performance standard

- 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.

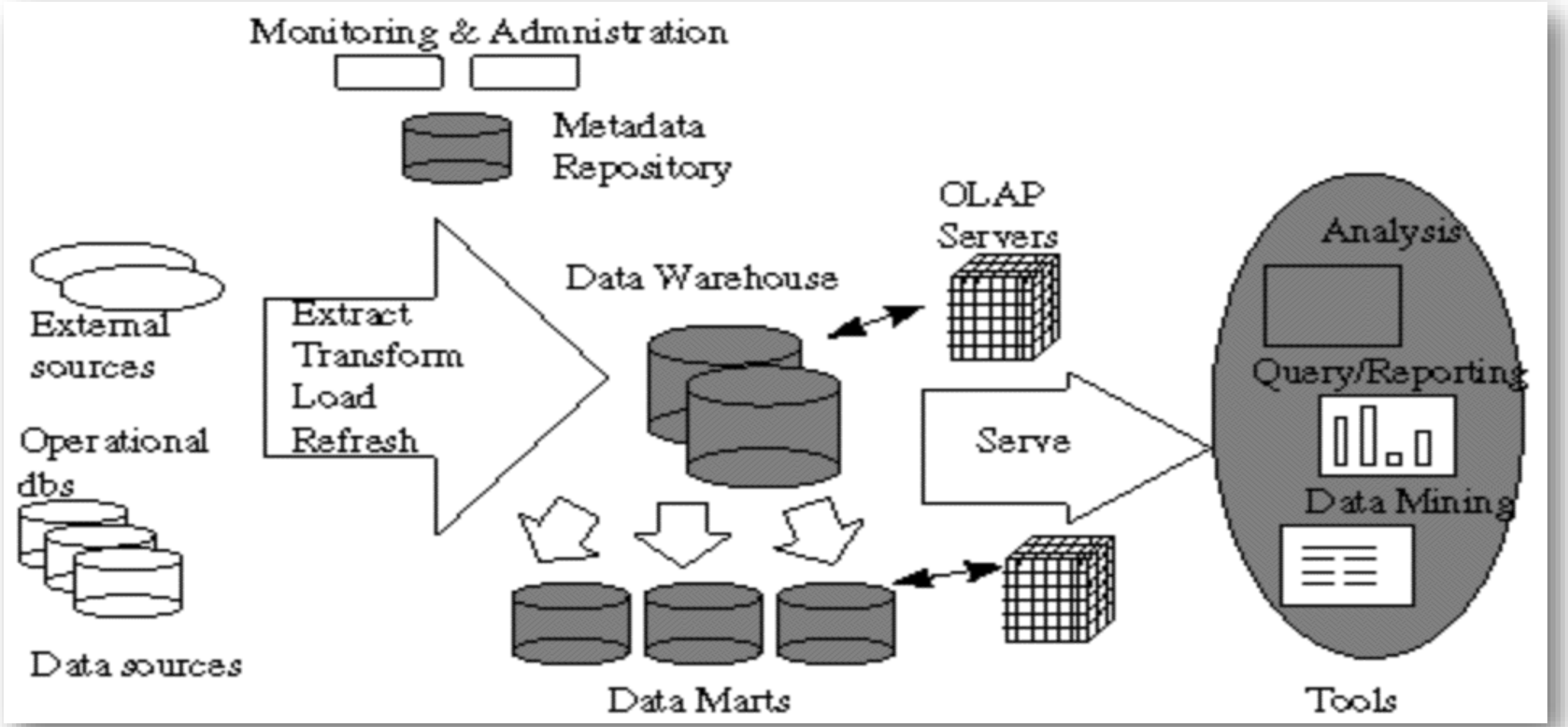
Planning those at least 10 presentations (course hours, 20% in final grades) of master students (teams) about:
(1.) retail sales, (2.) inventory, (3.) procurement, (4.) order management, (5.) accounting, (6.) CRM, (7.) HRM,
(8.) financial services, (9.) telecommunications,
(10.) transportation, (11.) 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

Extract Transform and Load

Kimball &Ross,2016

Ideally, the design of your ETL system begins with one of the toughest challenges: surrounding the requirements. By this we mean gathering in one place all the known requirements, realities, and constraints affecting the ETL system. The list of requirements is pretty overwhelming, but it's essential to lay them on the table before launching a DW/BI project.

The requirements are mostly things you must live with and adapt your system to. Within the framework of your requirements, you'll have many places where you can make your own decisions, exercise your judgment, and leverage your creativity, but the requirements are just what they're named. They are *required*.

Extract Transform and Load

Kimball &Ross,2016

Business Needs

In many cases, the original interviews with the business users and the original investigations of possible sources don't fully reveal the data's complexities and limitations. The ETL team often makes significant discoveries that affect whether the user's business needs can be addressed as originally hoped for. And, of course, the ETL team often discovers additional capabilities in the data sources that expand the users' decision making capabilities. The lesson here is that even during the most technical back room development steps of building the ETL system, you must maintain a dialog among the ETL team, data warehouse architects, business analysts, and business users. In a larger sense, the business needs and the content of the data sources are both moving targets that constantly need to be reexamined and discussed.

Extract Transform and Load

Kimball &Ross,2016

Compliance

Some of the financial reporting issues will be outside the scope of the data warehouse, but many others will land squarely within its scope. Typical due diligence requirements for the data warehouse include the following:

- Saving archived copies of data sources and subsequent stagings of data
- Proof of the complete transaction flow that changed any data results
- Fully documented algorithms for allocations, adjustments, and derivations
- Proof of security of the data copies over time, both online and offline

Extract Transform and Load

Kimball &Ross,2016

Data Quality via Data Profiling

As Jack Olson explains so clearly in his book *Data Quality: The Accuracy Dimension* (Morgan Kaufmann, 2003), data profiling is a necessary precursor to designing any kind of system to use that data. As he puts it: Data profiling "employs analytic methods for looking at data for the purpose of developing a thorough understanding of the content, structure, and quality of the data. A good data profiling [system] can process very large amounts of data, and with the skills of the analyst, uncover all sorts of issues that need to be addressed."

Data profiling is a systematic examination of the quality, scope, and context of a data source to allow an ETL system to be built. At one extreme, a very clean data source that has been well maintained before it arrives at the data warehouse requires minimal transformation and human intervention to load directly into final dimension tables and fact tables.

And at the other extreme, if data profiling reveals that the source data is deeply flawed and can't support the business' objectives, the data warehouse effort should be cancelled.

Extract Transform and Load

Kimball &Ross,2016

Data Integration and the 360 Degree View

Data integration is a huge topic for IT because, ultimately, it aims to make all systems work together seamlessly. The "360 degree view of the customer" is a familiar name for data integration. In many cases, serious data integration must take place among the organization's primary transaction systems before any of that data arrives at the data warehouse. But rarely is that data integration complete, unless the organization has settled on a single enterprise resource planning (ERP) system, and even then it's likely that other important transaction processing systems exist outside the main ERP system.

Data integration usually takes the form of conforming dimensions and conforming facts in the data warehouse. *Conforming dimensions* means establishing common dimensional attributes across separated databases so that "drill across" reports can be generated using these attributes. *Conforming facts* means making agreements on common business metrics such as key performance indicators (KPIs) across separated databases so that these numbers can be compared mathematically by calculating differences and ratios.

Extract Transform and Load

Kimball &Ross,2016

Data Latency

The data latency requirement describes how quickly the data must be delivered to the business users. Data latency obviously has a huge effect on the architecture and system implementation. Up to a point, more clever processing algorithms, parallel processing, and more potent hardware can speed up most of the traditional batch-oriented data flows. But at some point, if the data latency requirement is sufficiently urgent, the ETL system's architecture must convert from batch-oriented to streaming.

Archiving and Lineage

We recommend staging the data at each point that a major transformation has occurred. These staging points occur after all four steps: extract, clean, conform, and deliver. So, when does staging (writing the data to disk) turn into archiving (keeping the data indefinitely on some form of permanent media)?

Our simple answer is a conservative answer. *All staged data should be archived unless a conscious decision is made that specific data sets will never be recovered in the future.* It's almost always less of a headache to read the data back in from permanent media than it is to reprocess the data through the ETL system at a later time. And, of course, it may be impossible to reprocess the data according to the old processing algorithms if enough time has passed.

And while you are at it, each staged/archived data set should have accompanying metadata describing the origins and processing steps that produced the data.

Extract Transform and Load

Kimball &Ross,2016

BI User Delivery Interfaces

In general, the ETL team and data modelers need to work closely with the application developers to determine the exact requirements for the final data handoff. Each BI tool has certain sensitivities that should be avoided and certain features that can be exploited if the physical data is in the right format. The same considerations apply to data prepared for OLAP cubes.

Available Skills

Some of the big design decisions when building an ETL system must be made on the basis of available resources to build and manage the system. You shouldn't build a system that depends on critical C++ processing modules if those programming skills aren't in-house and you can't reasonably acquire and keep those skills. You may be much more confident in building your ETL system around a major vendor's ETL tool if you already have those skills in-house and you know how to manage such a project.

ETL subsystems

Kimball &Ross,2016

Extracting: Getting Data into the Data Warehouse

To no surprise, the initial subsystems of the ETL architecture address the issues of understanding your source data, extracting the data, and transferring it to the back room environment where the ETL system can operate on it independent of the operational systems. The extract related ETL subsystems include:

1. **Data profiling system.** Column property analysis including discovery of inferred domains; and structure analysis including candidate foreign key/primary key relationships, data rule analysis, and value rule analysis
2. **Change data capture system.** Source log file readers, source date and sequence number filters, and record comparisons based on cyclic redundancy checksum (CRC) algorithms
3. **Extract system.** Source data adapters, push/pull/dribble job schedulers, filtering and sorting at the source, proprietary data format conversions, and data staging after transfer to the ETL environment

ETL subsystems

Kimball &Ross,2016

Cleaning and Conforming Data

The five major subsystems in the cleaning and conforming step include:

4. **Data cleaning and quality screen handler system.** Typically a dictionary-driven system for complete parsing of names and addresses of individuals and organizations, possibly also products or locations. "Surviving" using specialized data merge logic that preserves specified fields from certain sources to be the final saved versions. Maintains back references (such as natural keys) to all participating original sources. Inline ETL tests applied systematically to all data flows checking for data quality issues
5. **Error event handler.** Comprehensive system for reporting and responding to all ETL error events. Includes branching logic to handle various classes of errors, and real-time monitoring of ETL data quality
6. **Audit dimension assembler.** Assembly of metadata context surrounding each fact table load in such a way that the metadata context can be attached to the fact table as a normal dimension
7. **Deduplication system.** Includes identification and removal, usually of individuals and organizations, possibly products or locations. Often uses fuzzy logic
8. **Data conformer.** Identification and enforcement of special conformed dimension attributes and conformed fact table measures as the basis for data integration across multiple sources

ETL subsystems

Kimball &Ross,2016

Delivering: Preparing for Presentation

9. **Slowly changing dimension (SCD) processor.** Transformation logic for handling dimension attribute time variance: type 1 (overwrite), type 2 (create new record), and type 3 (create new field)
10. **Surrogate key creation system.** Robust mechanism for producing surrogate keys, independently for every dimension. Independent of database instance, able to serve distributed clients
11. **Hierarchy dimension builder for fixed, variable, and ragged hierarchies.** Data validity checking and maintenance system for all forms of many-to-one hierarchies in a dimension. Variable hierarchy dimension builder. Data validity checking and maintenance system for all forms of ragged hierarchies of indeterminate depth
12. **Special dimension builder.** Creation and maintenance of dimensions consisting of miscellaneous low cardinality flags and indicators found in most production data sources
13. **Fact table loader for transaction, periodic snapshot, and accumulating snapshot grains.** System for updating fact tables including manipulation of indexes and partitions. Transaction grain: normally append mode for most recent data. Periodic snapshot grain: includes frequent overwrite strategy for incremental update of current period facts. Accumulating snapshot grain: includes manipulation of indexes and partitions, and updates to both dimension foreign keys and accumulating measures
14. **Surrogate key pipeline.** Pipelined, multithreaded process for replacing natural keys of incoming data with data warehouse surrogate keys
15. **Multi-valued dimension bridge table builder.** Creation and maintenance of associative bridge table used to describe a many-to-many relationship between dimensions. May include weighting factors used for allocations and situational role descriptions
16. **Late-arriving data handler.** Insertion and update logic for fact records and/or dimension records that have been delayed in arriving at the data warehouse
17. **Dimension manager system.** Administration system for the dimension manager who replicates conformed dimensions from a centralized location to fact table providers
18. **Fact table provider system.** Administration system for the fact table provider who receives conformed dimensions sent by the dimension manager. Includes local key substitution, dimension version checking, and aggregate table change management
19. **Aggregate builder.** Creation and maintenance of physical aggregate database structures that are used in conjunction with a query rewrite facility to improve query performance. Includes standalone aggregate tables and materialized views
20. **Multidimensional cube builder.** Creation and maintenance of star schema foundation for loading multidimensional (OLAP) cubes, including special preparation of dimension hierarchies as dictated by the specific cube technology
21. **Data integration manager.** System for transfer of large data sets between production applications and to the data warehouse

ETL subsystems

Kimball &Ross,2016

Managing the ETL Environment

22. **Job scheduler.** System for scheduling and launching all ETL jobs. Able to wait for a wide variety of system conditions including dependencies of prior jobs completing successfully. Able to post alerts
23. **Backup system.** Backup data and metadata for recovery, restart, security, and compliance requirements
24. **Recovery and restart system.** Common system for resuming a job that has halted, or for backing out a whole job and restarting. Significant dependency on backup system
25. **Version control system.** Consistent "snapshotting" capability for archiving and recovering all the metadata in the ETL pipeline. Check-out and check-in of all ETL modules and jobs. Source comparison capability to reveal differences between different versions
26. **Version migration system from development to test to production.** Move a complete ETL pipeline implementation out of development, into test, and then into production. Interface to version control system to back out a migration. Single interface for setting connection information for entire version. Independence from database location for surrogate key generation
27. **Workflow monitor.** Dashboard and reporting system for all job runs initiated by the job scheduler. Includes number of records processed, summaries of errors, and actions taken
28. **Sort system.** Standalone high performance sort package
29. **Lineage and dependency analyzer.** Display the ultimate physical sources and subsequent transformations of any selected data element, chosen either from the middle of the ETL pipeline or on a final delivered report (lineage). Display all affected downstream data elements and final report fields affected by a potential change in any selected data element, chosen either in the middle of the ETL pipeline or in an original source (dependency).
30. **Problem escalation system.** Automatic plus manual system for raising an error condition to the appropriate level for resolution and tracking. Includes simple error log entries, operator notification, supervisor notification, and system developer notification
31. **Parallelizing/pipelining system.** Common system for taking advantage of multiple processors or grid computing resources, and common system for implementing streaming data flows. Highly desirable (eventually necessary) that parallelizing and pipelining be invoked automatically for any ETL process that meets certain conditions, such as not writing to the disk or waiting on a condition in the middle of the process
32. **Security system.** Administer role-based security on all data and metadata in the ETL pipeline.
33. **Compliance reporter.** Comply with regulatory statutes to prove the lineage of key reported operating results. Prove that the data and the transformations haven't been changed. Show who has accessed or changed any such data.
34. **Metadata repository manager.** Comprehensive system for capturing and maintaining all ETL metadata, including all transformation logic. Includes process metadata, technical metadata, and business metadata

ETL – where should we correct data?

Kimball &Ross,2016

The key decision is where to correct the data. Clearly, the best solution is to have the data captured accurately in the first place. Of course, this isn't always the case, but nonetheless, in most cases, the data should be corrected back in the source system. Unfortunately, it is inevitable that poor quality data will reach the ETL system. In this event, there are three choices:

- Halting the entire load process
- Sending the offending record(s) to a suspense file for later processing
- Merely tagging the data and passing it through

The third choice is by far the best choice, whenever possible. Halting the process is obviously a pain because it requires manual intervention to diagnose the problem, restart or resume the job, or abort completely. Sending records to a suspense file is often a poor solution because it is not clear when or if these records will be fixed and reintroduced to the pipeline. Until the records are restored to the data flow, the overall integrity of the database is questionable because records are missing.

ETL – how quickly must source data be available via DWH?

Kimball &Ross,2016

Be careful: Asking business users if they want "real-time" delivery of data is an open invitation for trouble. Of course, most business users will respond positively to having data updated more frequently, regardless of whether they understand the impact of their request. Clearly this kind of data latency requirement could be dangerous. We recommend dividing the real-time challenge into three categories: daily, frequently, and instantaneous. You need to get your end users to describe their data latency requirements in similar terms and then design your ETL solution appropriately to support each set of requirements:

- **Instantaneous** means that the data visible on the end user's screen represents the true state of the source transaction system at every instant. When the source system status changes, the online screen must also respond instantly.
- **Frequently** means that the data visible to the end user is updated many times per day but is not guaranteed to be the absolute current truth as of this instant.
- **Daily** means that the data visible on the screen is valid as of a batch file download or reconciliation from the source system at the end of the previous working day.

ETL – failing to embrace a metadata strategy

Kimball &Ross,2016

Unfortunately, many ETL implementation teams do not embrace metadata early in the development process, putting off its capture to a future phase. This compromise typically is made because the ETL team does not "own" the overall metadata strategy. In fact, in the early stages of many new implementation efforts, it's not uncommon for there to be no designated owner of the metadata strategy.

At a minimum, the ETL team should strive to capture the business metadata created during the data-modeling and source-to-target mapping processes. Most organizations find it valuable to focus initially on capturing, integrating, flowing, and, ultimately, surfacing the business metadata through their BI tool; other metadata can be integrated over time.

ETL – pay attention to aggregations & dimensional modelling

Kimball &Ross,2016

Aggregations are like indexes: They are specific data structures meant to improve performance. Aggregations are a significant distraction in the back room. They consume processing resources, add complexity to the ETL applications, and take up a lot of storage. But aggregations remain the single most potent tool in the arsenal of the data warehouse designer to improve performance cost effectively.

the most widely used back room trade-off made for the benefit of the business user is the practice of dimensional modeling. A dimensional model is a second normal form version of a third (or higher) normal form model. Collapsing the snowflakes and other complex structures of the higher normal form models into the characteristic flat dimension tables makes the designs simple, symmetrical, and understandable. Furthermore, database vendors have focused their processing algorithms on this well understood case to make dimensional models run really fast.

ETL – advantages and disadvantages

Kimball &Ross,2016

There are numerous advantages associated with using an ETL tool:

- **Visual flow and self-documentation.** The single greatest advantage of an ETL tool is that it provides a visual flow of the system's logic. Each tool presents these flows differently, but even the least appealing of these user interfaces compare favorably to custom systems consisting of stored procedures, SQL and operating system scripts, and a handful of other technologies. Ironically, some ETL tools have no practical way to print the otherwise attractive self-documentation.
- **Structured system design.** ETL tools are designed for the specific problem of populating a data warehouse. Although they are only tools, they do provide a metadata-driven structure to the development team. This is particularly valuable for teams building their first ETL system.
- **Operational resilience.** Many of the home-grown ETL systems I've evaluated are fragile: They have too many operational problems. ETL tools provide functionality and practices for operating and monitoring the ETL system in production. You can certainly design and build a well-instrumented hand-coded ETL application, and ETL tool operational features have yet to mature. Nonetheless, it's easier for a DW/BI team to leverage the management features of an ETL tool to build a resilient system.

ETL – advantages and disadvantages

Kimball &Ross,2016

- **Data lineage and data dependency functionality.** We would like to be able to right-click on a number in a report and see exactly how it was calculated, where the data was stored in the data warehouse, how it was transformed, when the data was most recently refreshed, and what source system or systems underlay the numbers. Dependency is the flip side of lineage: We'd like to look at a table or column in the source system and know which ETL modules, data warehouse tables, OLAP cubes, and user reports might be affected by a structural change. In the absence of ETL standards that hand-coded systems could conform to, we must rely on ETL tool vendors to supply this functionality, though unfortunately, few have done so to date.
- **Advanced data cleaning functionality.** Most ETL systems are structurally complex with many sources and targets. At the same time, requirements for transformation are often fairly simple, consisting primarily of lookups and substitutions. If you have a complex transformation requirement, for example, if you need to deduplicate your customer list, you should use a specialized tool. Most ETL tools either offer advanced cleaning and deduplication modules (usually for a substantial additional price) or integrate smoothly with other specialized tools. At the very least, ETL tools provide a richer set of cleaning functions than are available in SQL.
- **Performance.** You might be surprised that performance is listed last under the advantages of the ETL tools. It's possible to build a high performance ETL system whether you use a tool or not. It's also possible to build an absolute dog of an ETL system whether you use a tool or not. I've never been able to test whether an excellent hand-coded ETL system outperforms an excellent tool-based ETL system; I believe the answer is that it's situational. But the structure imposed by an ETL tool makes it easier for an inexperienced ETL developer to build a quality system.

ETL – advantages and disadvantages

Kimball &Ross,2016

- **Software licensing cost.** The greatest disadvantage of ETL tools in comparison to handcrafted systems is the licensing cost for the ETL tool software. Costs vary widely in the ETL space, from several thousand dollars to hundreds of thousands of dollars.
- **Uncertainty.** We've spoken with many ETL teams that are uncertain and sometimes misinformed about what an ETL tool will do for them. Some teams undervalue ETL tools, believing they are simply a visual way to connect SQL scripts together. Other teams unrealistically overvalue ETL tools, imagining that building the ETL system with a tool will be more like installing and configuring software than developing an application.
- **Reduced flexibility.** A tool-based approach limits you to the tool vendor's abilities and scripting languages.

ETL – document the ETL system

Kimball &Ross,2016

Whether you use an ETL tool or hand code your ETL system, it's a piece of software like any other and needs to be documented. As your data warehouse evolves, the ETL system evolves in step; you and your colleagues need to be able to quickly understand both the entire system architecture and the gritty details.

There's a widespread myth that ETL tools are self-documenting. This is true only in comparison with hand-coded systems. Don't buy into this myth: You need to develop an overall, consistent architecture for your ETL system. And, you need to document that system. Yes, that means writing a document.

The first step in building a maintainable ETL system is to stop and think about what you're doing. How can you modularize the system? How will those modules fit together into an overall flow? Develop your system so that you use a separate package, flow, module (or whatever your tool calls it) for each table in the data warehouse. Write a document that describes the overall approach; this can be a few pages, plus a screenshot or two.

Design a template module and group like activities together. The template should clearly identify which widgets are associated with extracts, transformations, lookups, conformation, dimension change management, and final delivery of the target table. Then, document this template flow in painstaking detail, including screenshots.

ETL – new directions

Kimball &Ross,2016

Extreme Integration

Most organizations are realizing that they have dozens, if not hundreds of potential data sources, especially those that are customer facing. The established best practices of manual data conforming are becoming unscalable. A new class of software startups are offering statistical data conforming that can provide usable matching of entities, such as customers, that approach the accuracy of manual data conforming, but at drastically higher speeds. The standard notion of a human dimension manager may give way to a "robot dimension manager!"

Extreme Variety

The big data revolution has trumpeted the four V's: volume, velocity, variety, and value. In my opinion, the most interesting and challenging V is variety. Standard relational databases and standard ETL pipelines are ill-equipped to handle unstructured text, hyper-structured machine data, graph relationships (think Facebook and LinkedIn), or images. As the value of these data types grows, ETL must change with new pipelines and new logic.

ETL – new directions

Kimball &Ross,2016

Huge Volumes

The lid has been off of data volumes for some time but it has reached a ridiculous point where even normal Main Street organizations want to access petabytes of data. Even when this data consists of conventional text and numbers (think log data), the data is a whale trapped in a swimming pool. You don't dare move or copy the data to a new place for processing.

Real-Time Delivery

Reports and ad hoc queries remain important but the new phrase is operational analytics, which combine high performance data ingestion, real-time data quality checking and data conforming, and finally sophisticated analytics. All of this requires forgoing conventional batch processing and slow periodic updates of the available data.

Rise of the Analyst and Monetization of Data Insights

Data scientist is the new name for analysts who mine data for insights and propose experiments in marketing strategies (often in real time) that can affect revenue, profitability, and customer satisfaction. These data scientists, frequently working in business departments, are often skilled at communicating directly with senior management, effectively bypassing IT. The challenge is whether IT can become a participant in this process, understanding the ETL pipelines, and creating a stable data infrastructure beyond the prototypes built by the data scientists.

New Analytic Tools

Data scientists and others are employing advanced analytic tools that can take several forms. Some of these tools are statistical algorithms found in advanced packages such as MadLib that can be loaded into some DBMS systems. Others are custom user defined functions (UDFs) typically programmed in C. Finally, others are separate BI tools that consume the data warehouse data.

ETL – new directions

Kimball &Ross,2016

Columnar Data Stores and In-Memory Databases

The sweet spot for high performance dimensional databases is a combination of columnar data stores and in-memory processing. Columnar data stores are typically excellent at processing many simultaneous joins against a fact table and tolerating very wide dimension tables. As the cost of memory continues to decline, it becomes feasible to stand up terabytes of RAM in a distributed shared-nothing environment. Taking advantage of this physical RAM is still a work in progress, and this balancing is a key aspect of the ETL pipeline architecture. The MapReduce processing framework typically found in Hadoop clusters addresses this problem for certain kinds of "relation-scan" analyses, but MapReduce relies on a shuffle step that moves the data from node to node to achieve balancing. Expect to see much more progress that will affect ETL in order to take advantage of the huge performance advantages offered by columnar in-store architectures.

Data Virtualization on Steroids

Finally, the venerable approach of defining simple SQL views to present tables in more usable formats, has given way to data virtualization, which now has become a major tool that in some cases replaces conventional ETL. In its basic form, data virtualization replaces physical data transformations with equivalent computations every time the data is accessed. Data virtualization becomes a classic trade-off between query performance and speed of deployment. Data virtualization is great for prototyping and data exploration, and when the prototyping phase is over, data virtualization can be replaced with true conventional ETL where the data is permanently moved and transformed.

ETL – share findings

Kimball &Ross,2016

Once you have an idea of the data quality issues you face, and the analytic problems they will cause, you need to educate the business people. Ultimately, they will need to redefine the data capture requirements for the transaction systems and allocate additional resources to fix them. They won't do this unless they understand the problems and associated costs.

The first major chance to educate on data quality problems is as part of the opportunity prioritization session with senior management. You should show examples of data quality problems, explain how they are created, and demonstrate their impact on analytics and project feasibility. Explain that you will document these in more detail as part of the modeling process, and at that point you can reconvene to determine your data quality strategy. Set the expectation that this is work and will require resources.

The dimensional modeling process is the second major education opportunity. All of the issues you identify during the modeling process should be discussed as part of documenting the model, and an approach to remedying the problem should be agreed upon with key business folks.

At some point, you should have generated enough awareness and concern to establish a small scale data governance effort, which will become the primary research and education channel for data quality.

ETL – have you built your audit dim. yet?

Kimball &Ross,2016

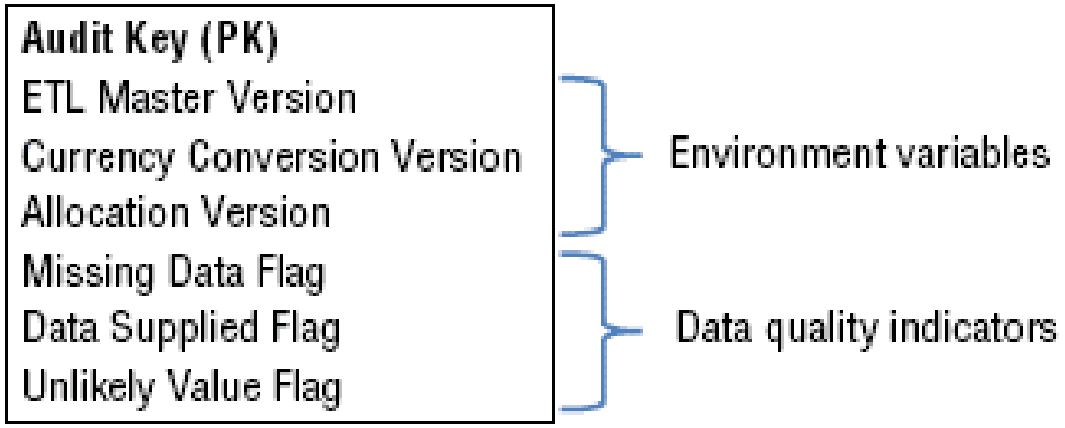


Figure 11-9: A simple audit dimension.

One of the most effective tools for managing data quality and data governance, as well as giving business users confidence in the data warehouse results, is the audit dimension. We often attach an audit dimension to every fact table so that business users can choose to illuminate the provenance and confidence in their queries and reports. Simply put, the audit dimension elevates metadata to the status of ordinary data and makes this metadata available at the top level of any BI tool user interface.

The secret to building a successful audit dimension is to keep it simple and not get too idealistic in the beginning. A simple audit dimension should contain environment variables and data quality indicators, shown in Figure 11-9.

ETL – working in web time

Kimball &Ross,2016

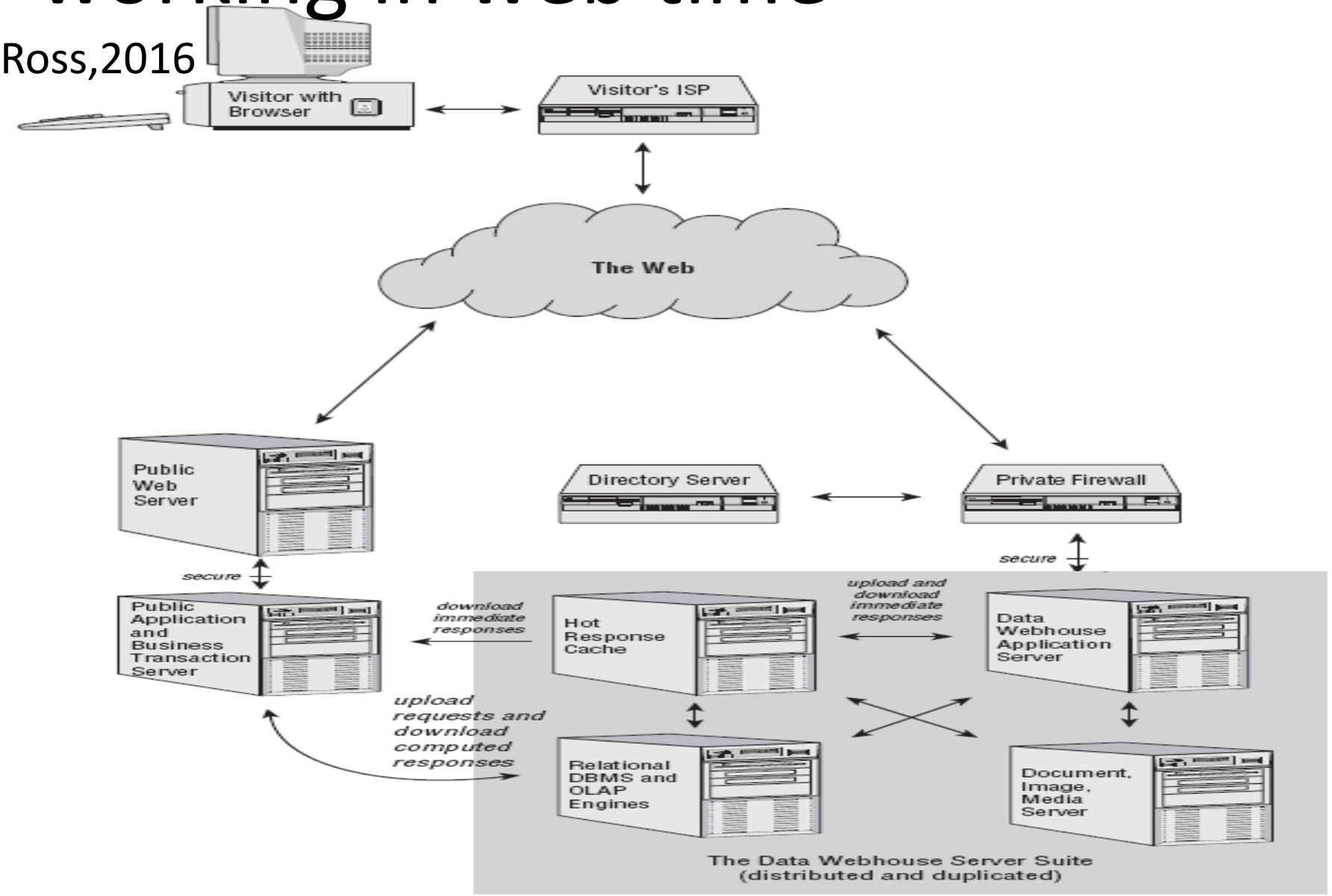


Figure 11-21: The web-intensive data warehouse architecture, showing the hot response cache and the application server that creates its contents.

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)
-Building an ETL pipeline from scratch in 30 minutes (Google Cloud Platform-Data Flow Python SDK)
-Getting Started with Amazon Web Services Glue ETL
-ETL demo (written in C#)