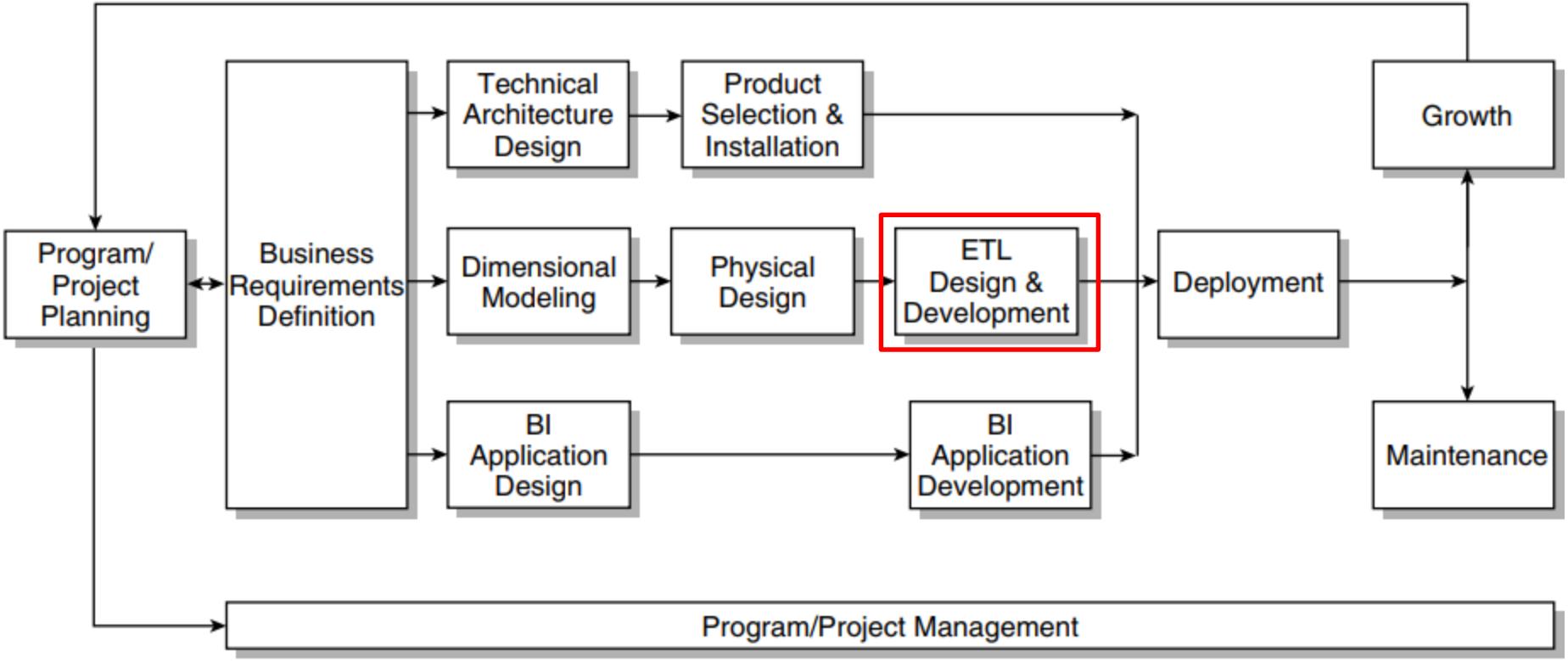
Data Warehousing and Business analytics

Lecture - 5

Designing and Developing the ETL System

Designing and Developing the ETL System

The Kimball Lifecycle diagram



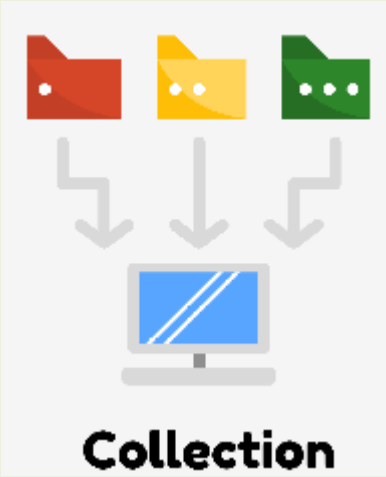
Note: Before you begin the ETL system design for a dimensional model, you should have completed the logical design, drafted your high level architecture plan, and drafted the source-to-target mapping for all data elements

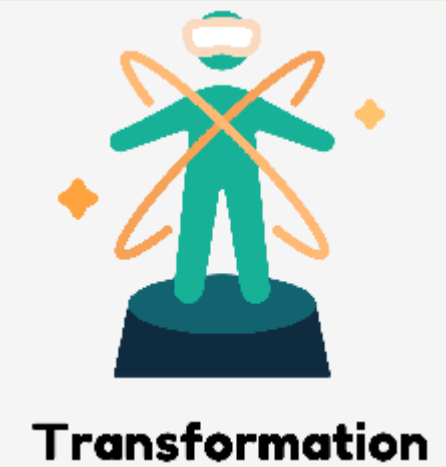
Data Architect

- Data Architectures
- Data Management Systems
- Data Processing Architectures

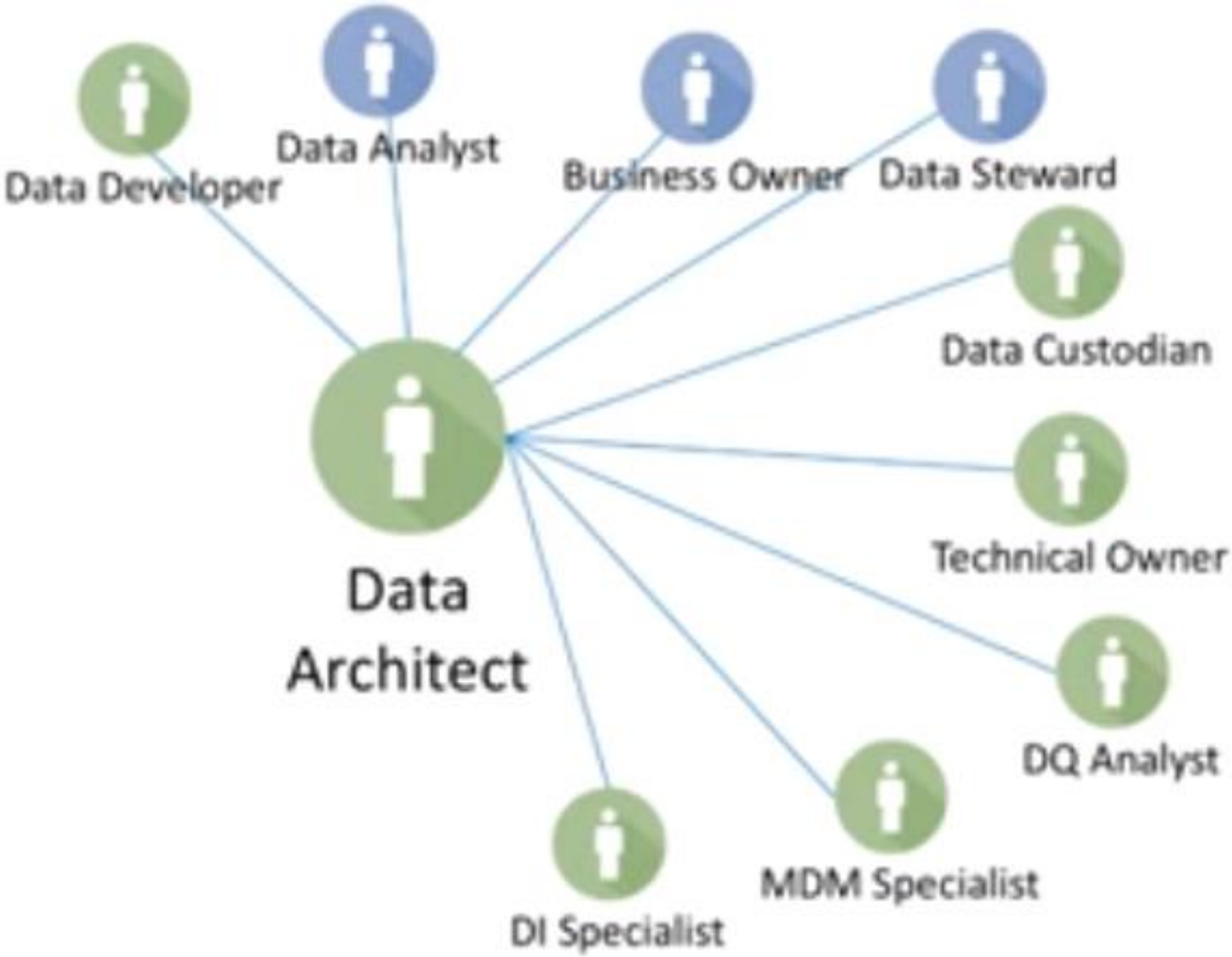
Data Architectures - Concept

- Data Architectures describe how data is managed (IBM)





Data Architecture roles



Benefits of data architectures

- Reducing redundancy
- Improving data quality
- Enabling integration
- Data lifecycle management

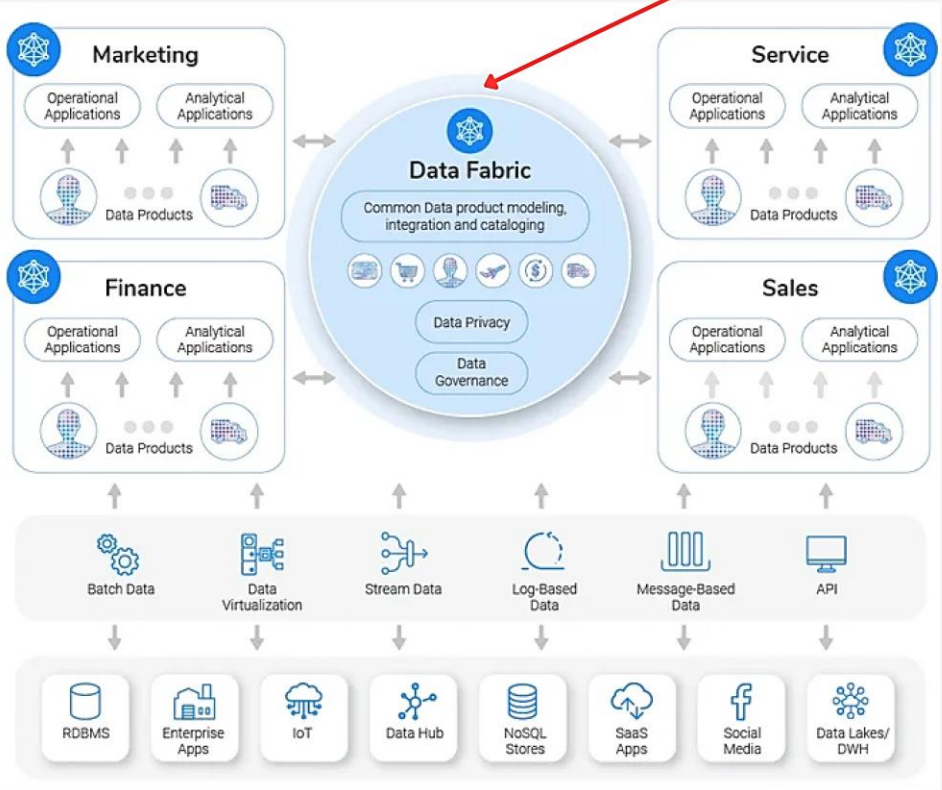
Types of data architectures

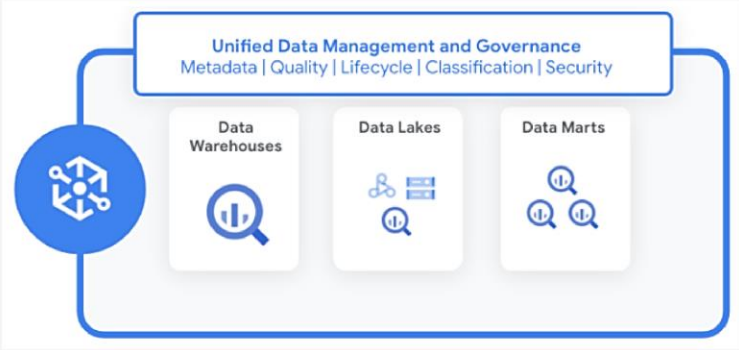
- **Data fabric**: Unify multiple and disjoint data sources in various environments.
  - ✓ Data sources: Data warehouses, Data marts, Data lakes
  - ✓ Environments: on-premis, cloud, edge
- **Data mesh:** Distribute data ownership to domain-specific teams. Each team manages, own and serves the data as products

Types of data architectures

A virtual layer to manage all data
(especially heterogeneous)

Data Fabric

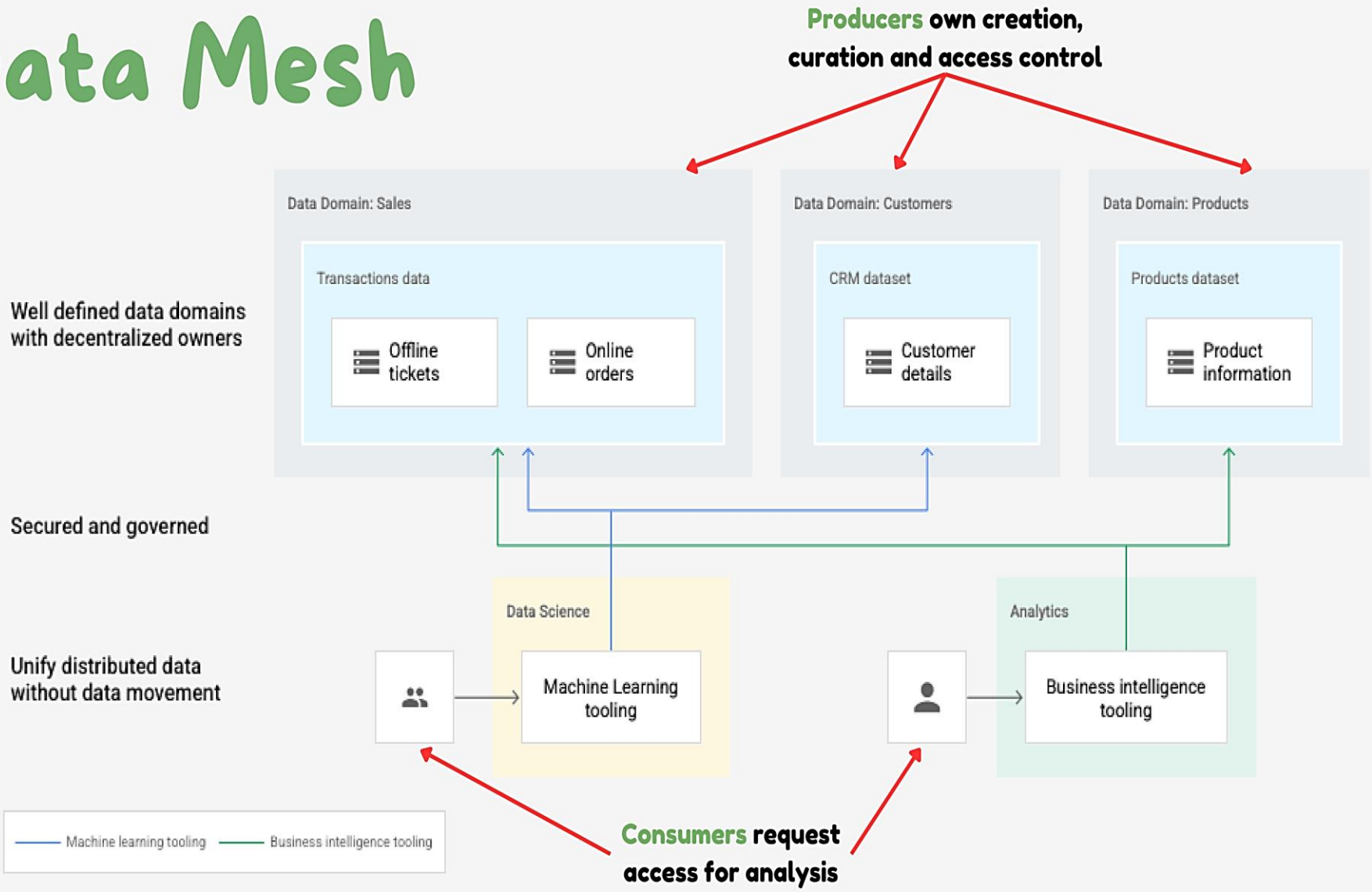




Dataplex - A GCP data fabric service

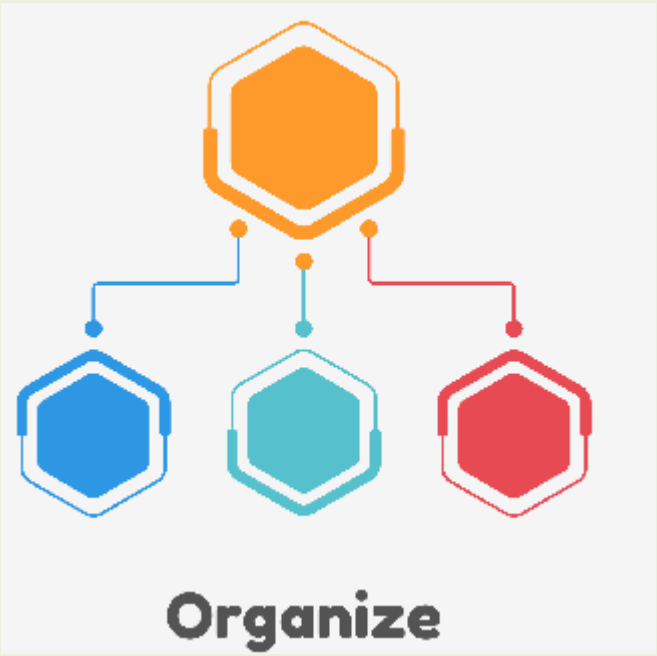
Types of data architectures

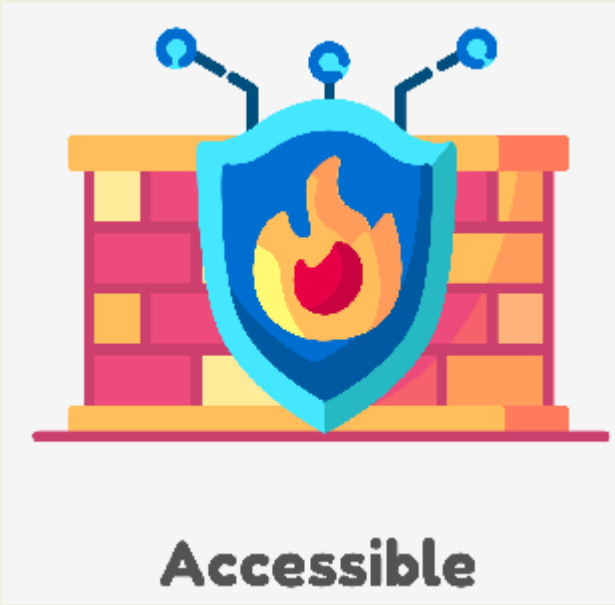
Data Mesh



Data Management Systems (DMS)

- Software systems to organize, secure and make data accessible for authorized users





Types of DMS

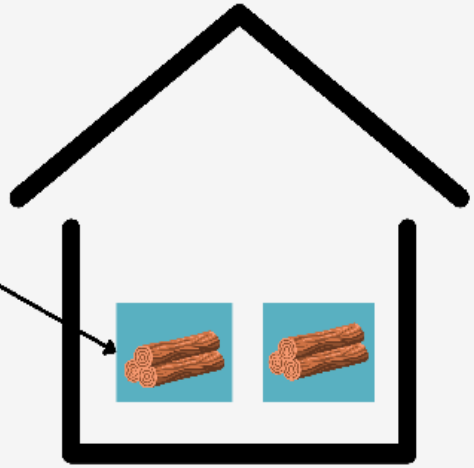
3. Data Lake

A central location for both structured and unstructured data in its raw form.

- More flexible and cost effective
- Hard for security and governance, slower query

2. Data Mart

A subset of a warehouse to serve a specific domain.
E.g., sales, accounting



1. Data Warehouse

A central data hub containing highly formatted and structured data for analytics
E.g., GCP BigQuery, AWS Redshift, Clickhouse

- Optimize for structured data, easier for data governance and security
- Only for structured data, high cost for maintenance, expensive scaling

4. Lake House

Add layers for data management, governance and query performance on top of Data Lake

- Address some of Data Lake's issues
- More complex than the others

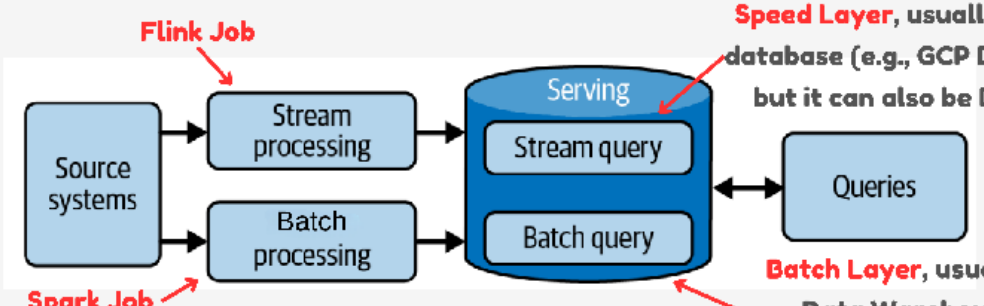
Types of DMS

- **Data warehouses:** aggregates data from different relational data sources across an enterprise into a single, central, consistent repository.
- **Data marts:** a focused version of a data warehouse that contains a smaller subset of data important to and needed by a single team or a select group of users within an organization
- **Data Lakes:** store raw data, typically petabytes of it. A data lake can store both structured and unstructured data

Data Processing Architectures

Types of architectures

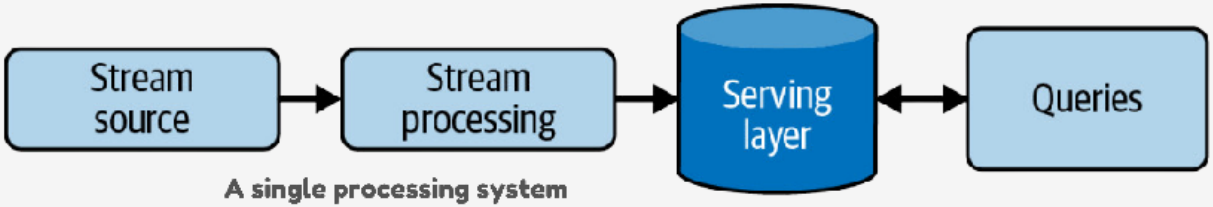
Lambda Architecture



Note:
- Unified serving layer
- Two-separate serving layer

Double infra, mismatched between 2 types of processing, 2 code base to maintain

Kappa Architecture

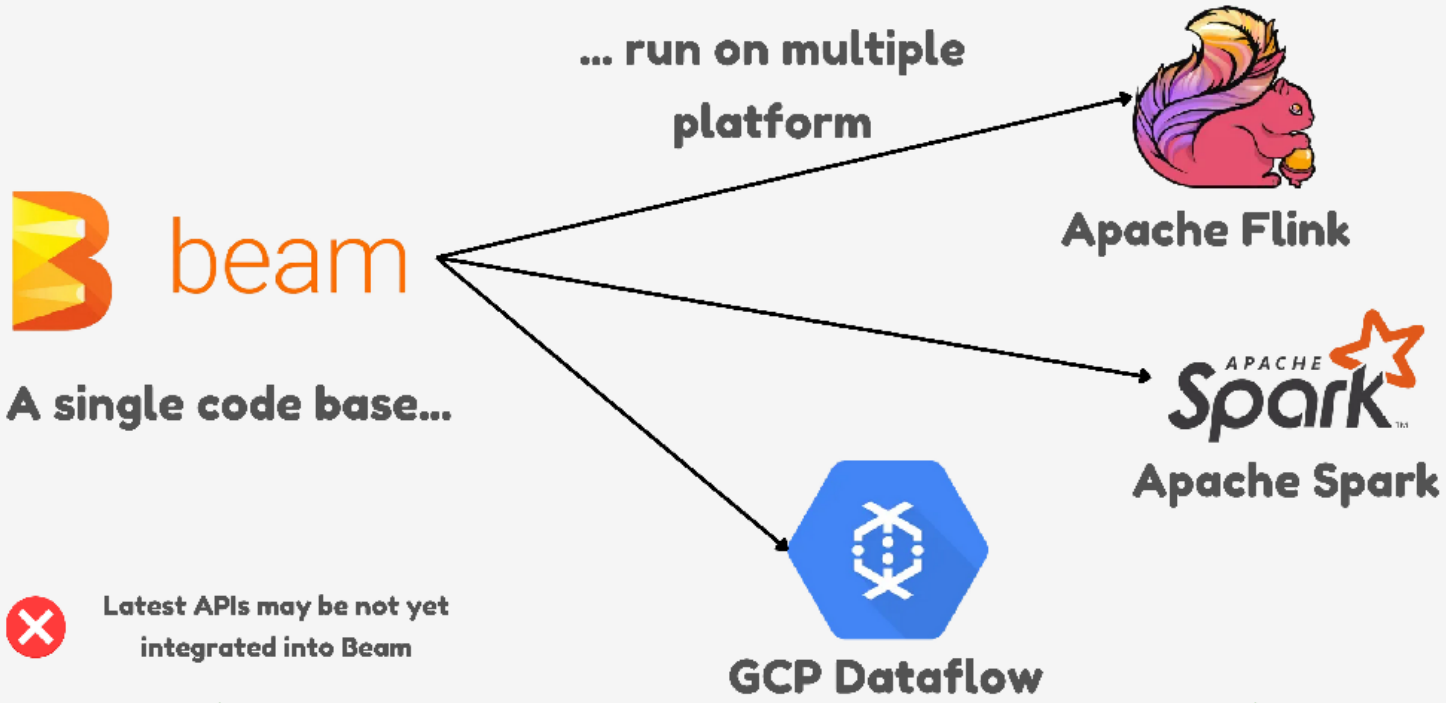


Single infra and code base

Not easy to implement, complex join (e.g., 30 tables), and expensive for huge volume data

DataFlow Model

Inspired from "batch is a special case of streaming" philosophy



Designing and Developing the ETL System

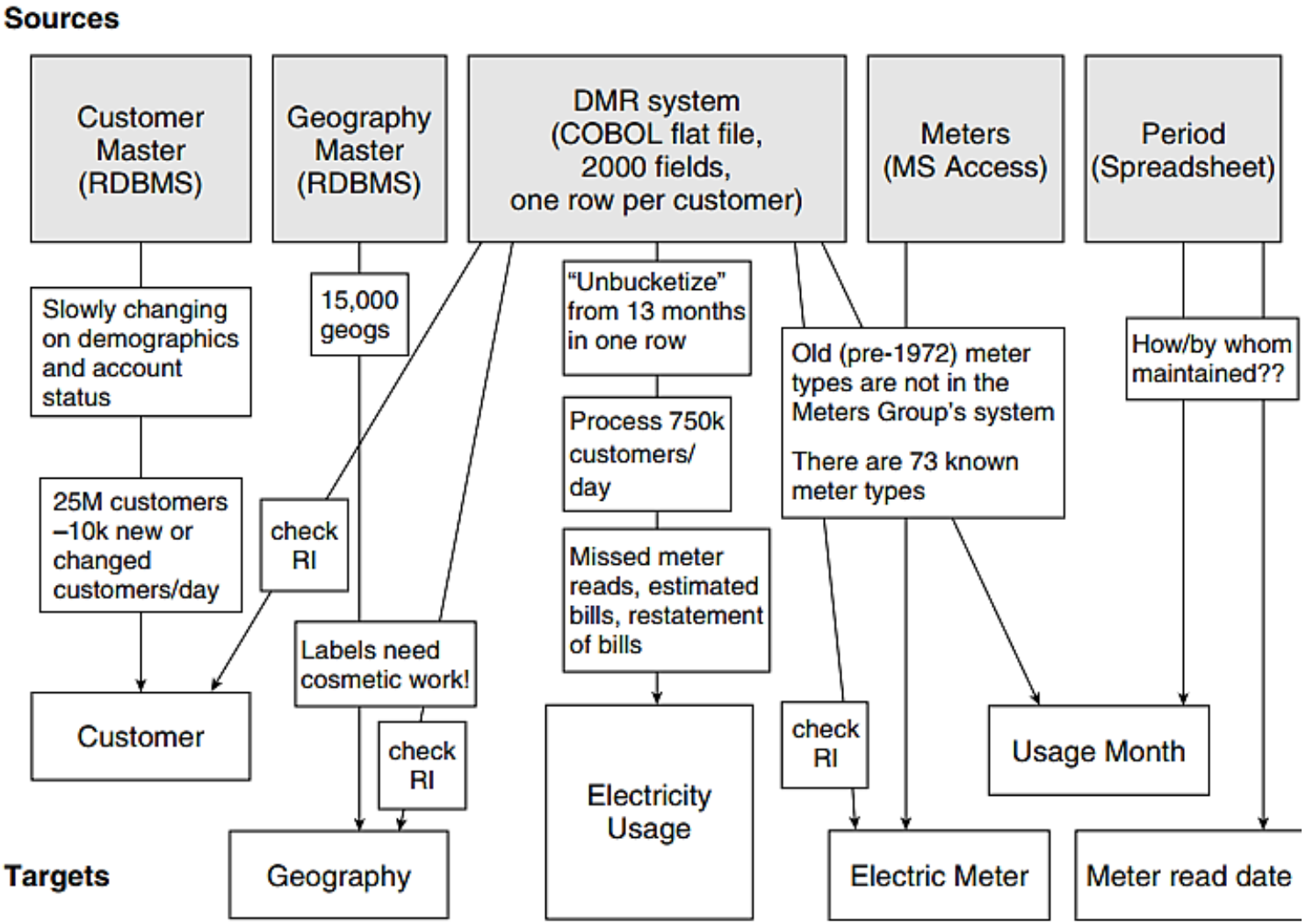
1. Plan
2. Develop One-Time Historic Load Process
3. Develop Incremental Load Process

Designing and Developing the ETL System

1. Draw the High Level Plan
2. Choose an ETL Tool
3. Develop Default Strategies
4. Drill Down by Target Table
5. Populate Dimension Tables with Historic Data
6. Perform the Fact Table Historic Load
7. Dimension Table Incremental Processing
8. Fact Table Incremental Processing
9. Aggregate Table and OLAP Loads
10. ETL System Operation and Automation

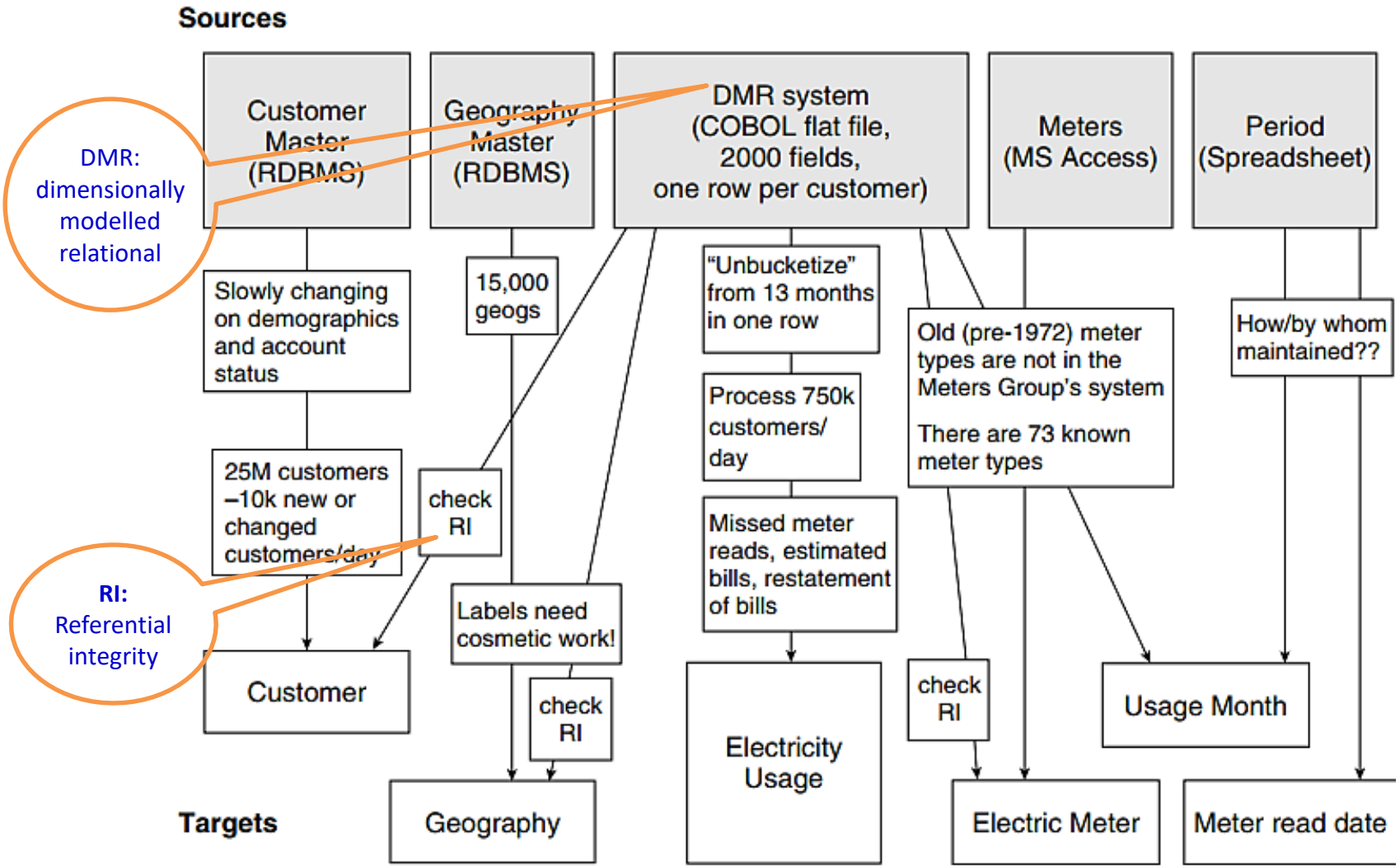
1. Draw the High Level Plan

Independent of any specific technology or approach

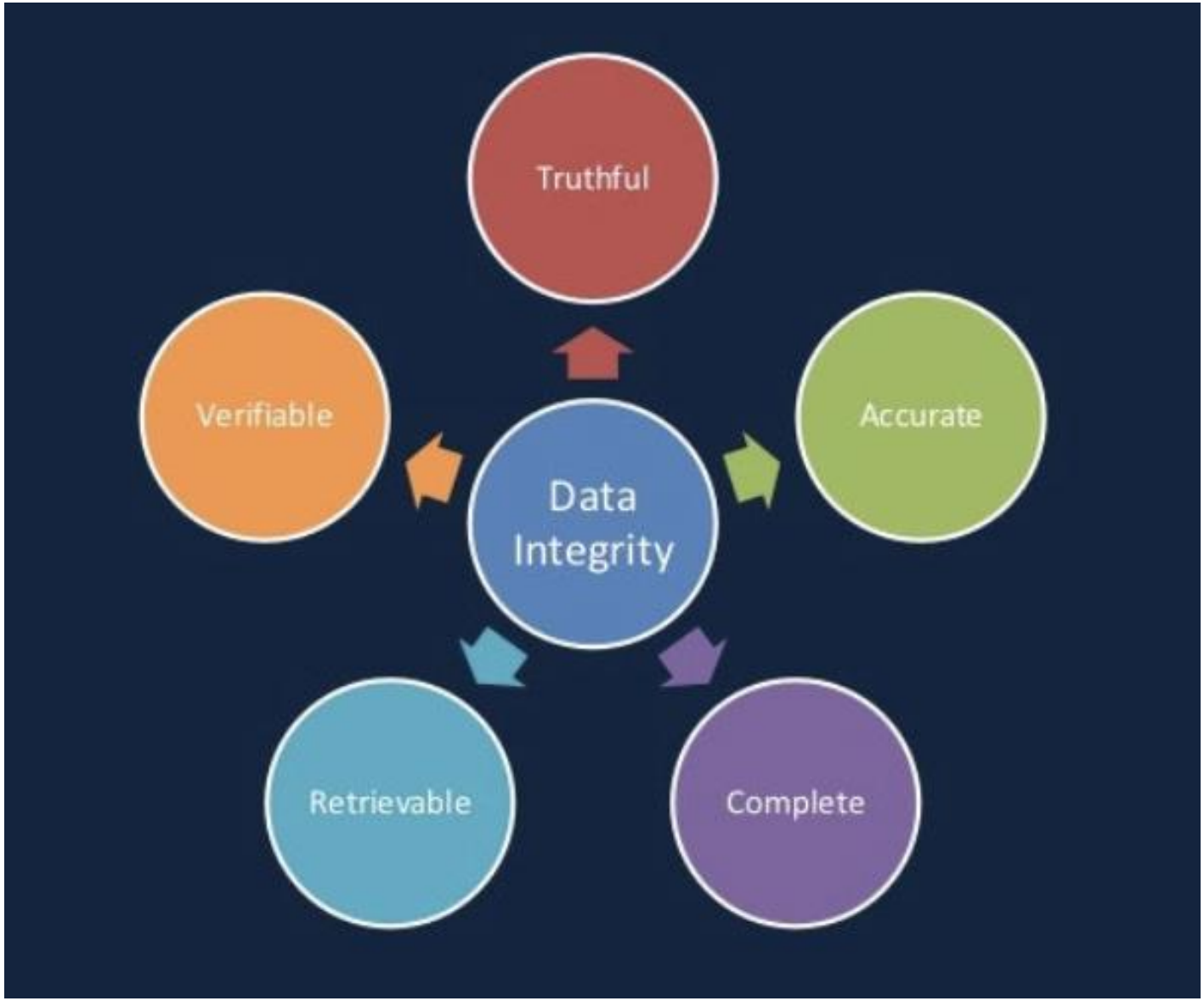


1. Draw the High Level Plan

Independent of any specific technology or approach



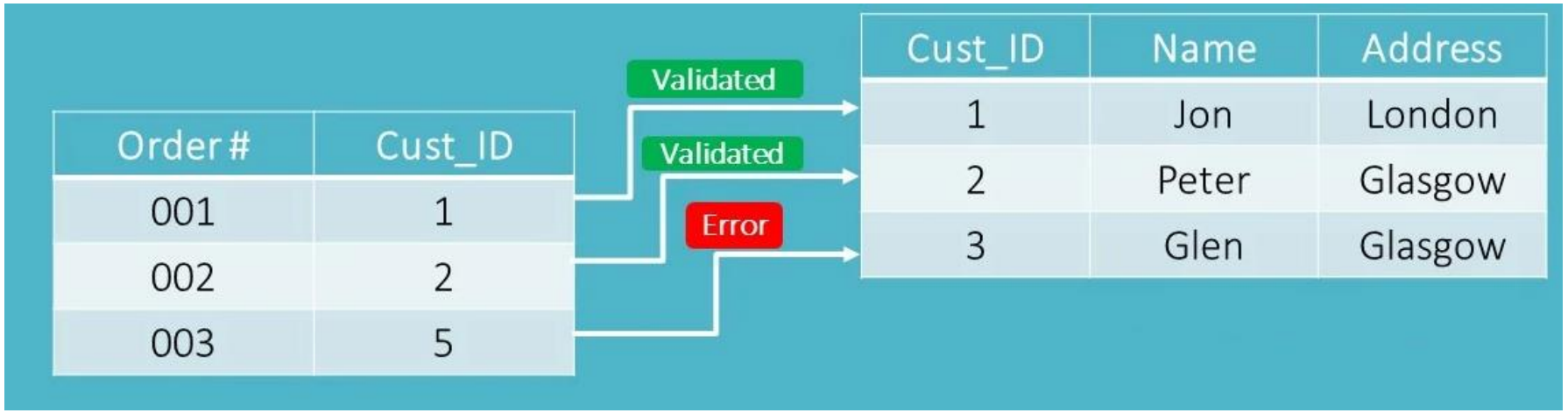
Data integrity



Types of Data integrity

- **Physical Integrity:** the rules and procedures that guarantee the accuracy of data
- **Logical Integrity:** ensures that the data accurately makes sense in a specific context. Logical integrity includes *Entity integrity*, *Domain integrity*, *Referential integrity*, *User-defined integrity*

Referential integrity



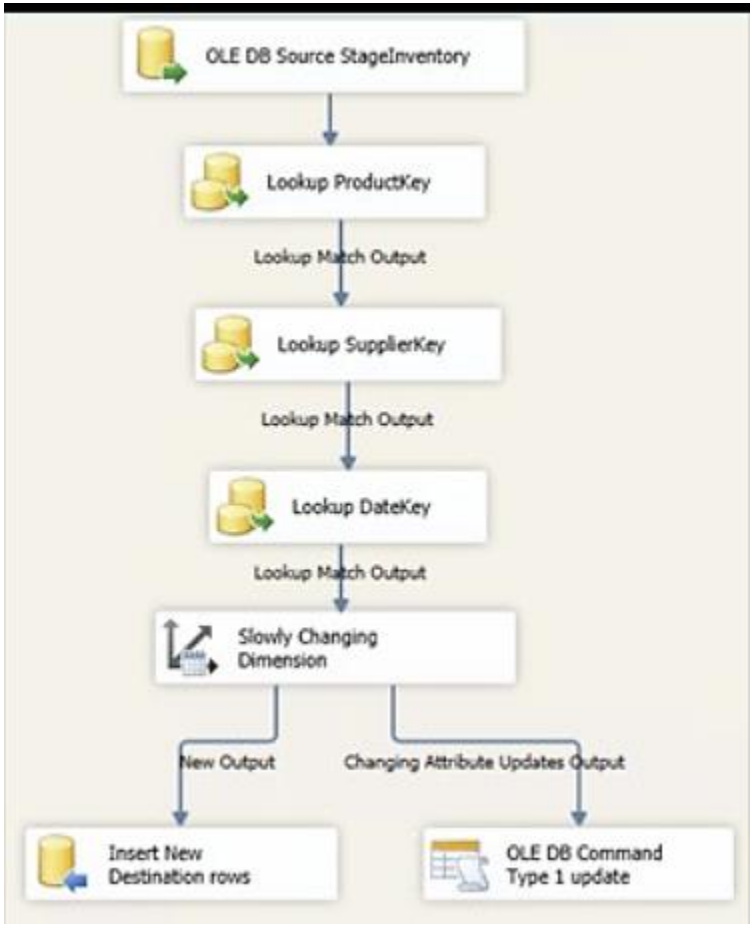
2. Choose an ETL Tool (1/3)

- ETL tools serve a variety of functions
- Self-documentation that comes from using a graphical tool.
- ETL tools offer advanced transformation logic
- Improved system performance at a lower level of expertise

2. Choose an ETL Tool (1/3)

```
46
47  ) AS
48  BEGIN
49      -- SET NOCOUNT ON added to prevent extra result sets from
50      -- interfering with SELECT statements.
51      SET NOCOUNT ON;
52      declare @id varchar(10)
53      declare @courseId varchar(10)
54      declare @keyid int
55      set @id = @term + '.' + @classNumber
56      set @courseId = @courseSubj + @courseNum
57      if (UPPER(@component) = 'LAB') set @courseTitle = (select 'LAB: ' + @courseTitle)
58
59      -- Courses table
60      if not exists(select * from dbo.Courses where
61          courseId=@courseId and courseTitle=@courseTitle)
62      begin
63          insert into dbo.Courses (courseId, courseTitle) values (@courseId, @courseTitle)
64      end
65
66      set @keyid = (select distinct keyId from Courses
67          where courseId=@courseId and courseTitle=@courseTitle)
68      --print 'keyid = ' + cast( @keyid as varchar(10))
69      -- Classes table
70      if exists(select * from dbo.Classes where id = @id)
71      begin
72          update dbo.Classes
73              set touched = 1
74              ,lastUpdate = GETDATE()
75          where id =@id
76      end
77      else
78      begin
79          insert into dbo.Classes ( id, scheduleSectionId, lastUpdate, touched)
80          values (@id, 0, GETDATE(), 1)
81      end
82
83      -- provClasses table
84      if exists(select * from dbo.provClasses where id = @id)
85      begin
86          update dbo.provClasses
87              set touched = 1
```

ETL code



ETL Tool

2. Choose an ETL Tool (3/3)

Tools for ETL subsystems: debenzium, rabbitmq, spark, great expectation, DBT

3. Develop Default Strategies (1/4)

- Extract from each major source system
- Archive extracted data
- Check data quality for dimensions and facts
- Manage changes to dimension attributes
- Ensure the data warehouse and ETL system meet the requirements for system availability
- Design the data auditing subsystem
- Organize the ETL staging area

3. Develop Default Strategies (2/4)

- **Extract from each major source system**: determine the default method for extracting data from each source system
- **Archive extracted data** for system auditability and recoverability. It's common to store the archived data in files rather than database tables.

3. Develop Default Strategies (3/4)

- **Check data quality for dimensions and facts:** Develop a general, standardized process for how you will check data quality, and what you will do if the data does not pass quality control
- **Manage changes to dimension attributes:** type 1 (overwrite history) and type 2 (track history)

3. Develop Default Strategies (4/4)

- **Ensure the data warehouse and ETL system meet the requirements for system availability**
- **Design the data auditing subsystem:** describes how the data entered the system
- **Organize the ETL staging area:** Most ETL systems will stage the data at least once or twice during the ETL process.

4. Drill Down by Target Table

- Start drilling into the detailed transformations needed to populate each target table in the data warehouse ata at least once or twice during the ETL process.
- Ensure that you know where the data is coming from for each column in each target table

Ensure Clean Hierarchies

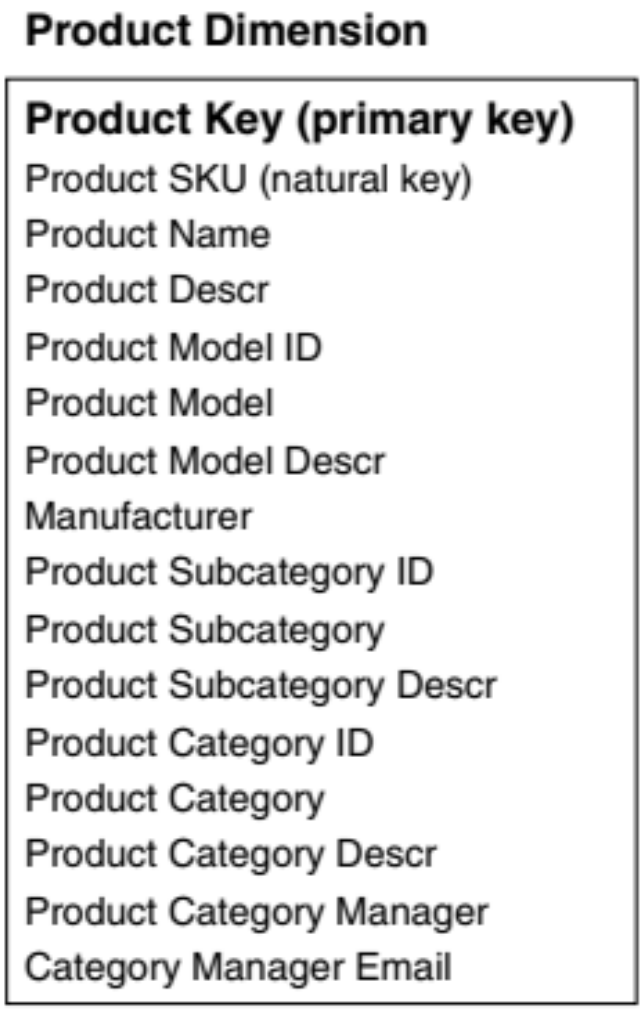
Develop Detailed Table Schematics

5.Populate Dimension Tables with Historic Data

- Dimension tables: type 1 (update in place) or type 2 (track history by adding new rows to the dimension table).
- Populate Type 1 Dimension Tables
- Dimension Transformations
- Dimension Table Loading
- Load Type 2 Dimension Table History
- Populate Date and Other Static Dimensions

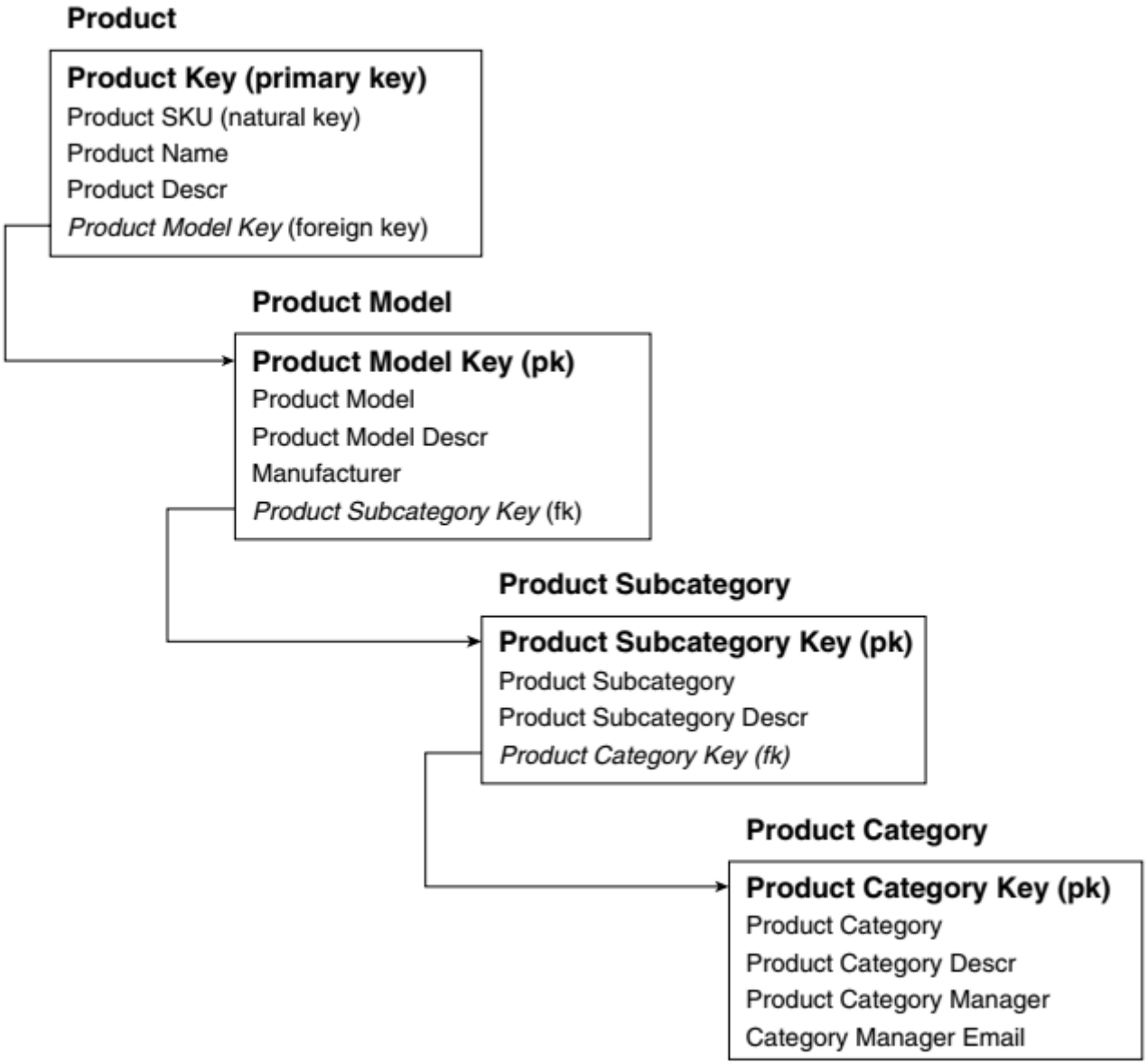
Hierarchical relationship

Product dimension table that includes a hierarchical relationship



Hierarchical relationship

Snowflaked hierarchical relationship in the product dimension



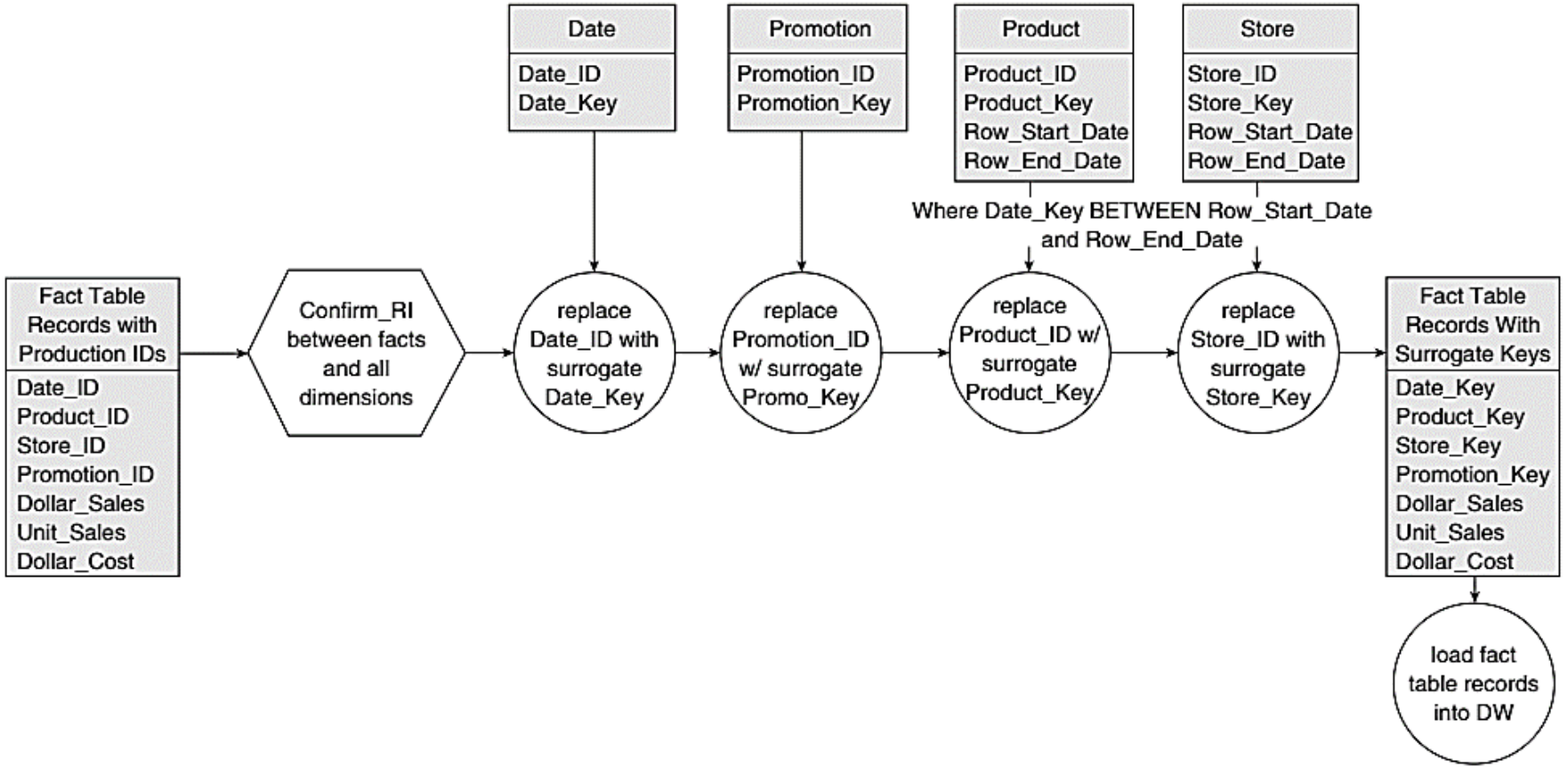
6. Perform the Fact Table Historic Load

- Historic Fact Table Extracts: make sure extracted records are useful for the data warehouse
- Fact Table Transformations: transformation of null values, pivoting or un-pivoting the data, and precomputing derived calculation
- Fact Table Loading: should have a perfectly clean set of data to load into the fact table.

Fact Table Loading process

- Disable foreign key (referential integrity) constraints between the fact table and each dimension table before loading data
- Drop or disable indexes on the fact table
- Load the data using fast loading techniques.
- Create or enable fact table indexes
- If necessary, perform steps to stitch together the table's partitions
- Confirm each dimension table has a unique index on the surrogate key column
- Enable foreign key constraints between the fact table and dimension tables

Historic surrogate key pipeline example



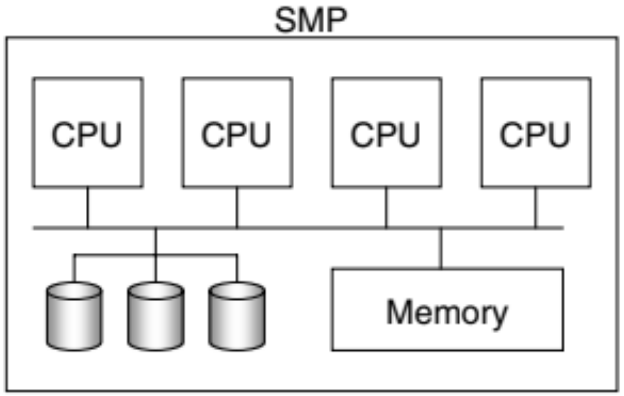
7. Dimension Table Incremental Processing

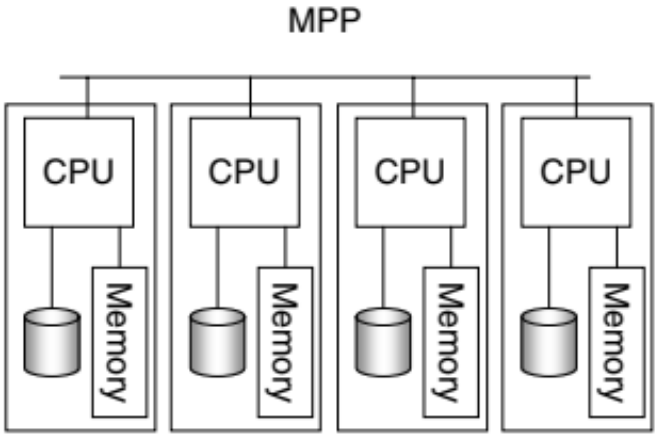
- Dimension Table Extracts
- Identify New and Changed Dimension Rows
- Process Changes to Dimension Attributes

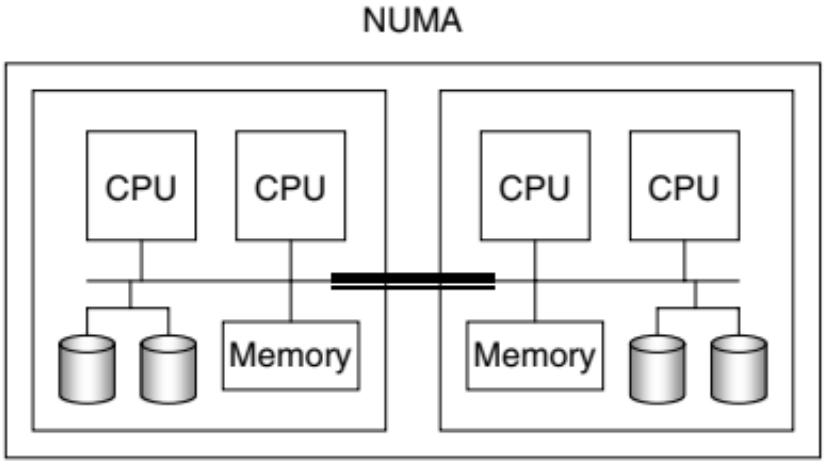
8. Fact Table Incremental Processing

- Fact Table Extract and Data Quality Checkpoint
- Fact Table Transformations and Surrogate Key Pipeline
- Late Arriving Facts and the Surrogate Key Pipeline
- Incremental Fact Table Load
- Speed Up the Load Cycle: More Frequent Loading or Parallel Processing

Basic parallel processing hardware



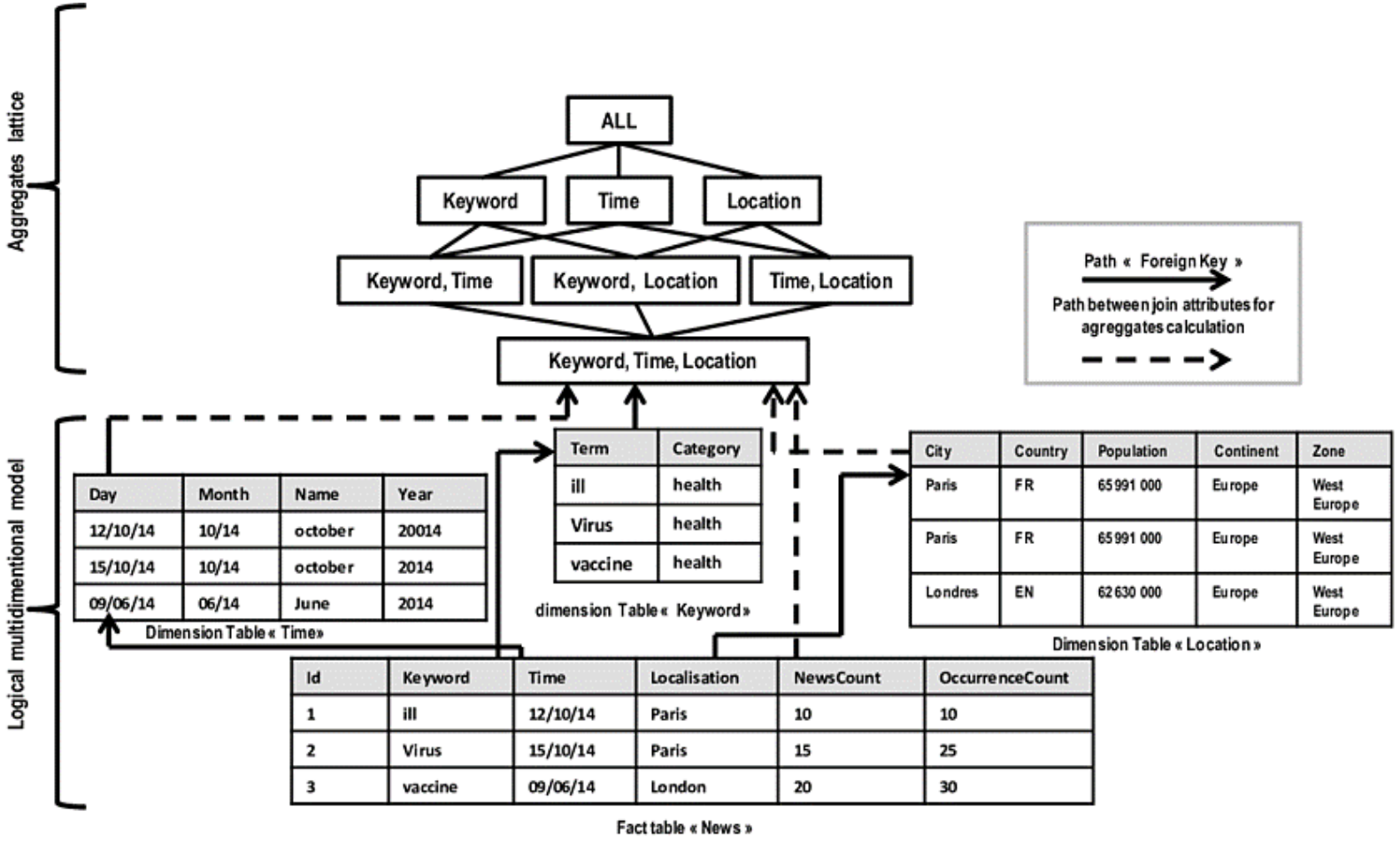




Symmetric Multiprocessing (SMP)
Massively Parallel Processing (MPP)
Non-Uniform Memory Architecture (NUMA)

9. Aggregate Table and OLAP Loads

- An aggregate table is *logically* easy to build
- Aggregate tables are often fairly easy to maintain



A pre-computed aggregate lattice in a R-OLAP system.

10. ETL System Operation and Automation

- Schedule Jobs
- Handle Predictable Exceptions and Errors Automatically
- Handle Unpredictable Errors Gracefully
- Maintain Database Objects
- Develop and Test ETL Automation


ANY
QUESTIONS?