Ten Principles of a good ETL & DWH Architecture

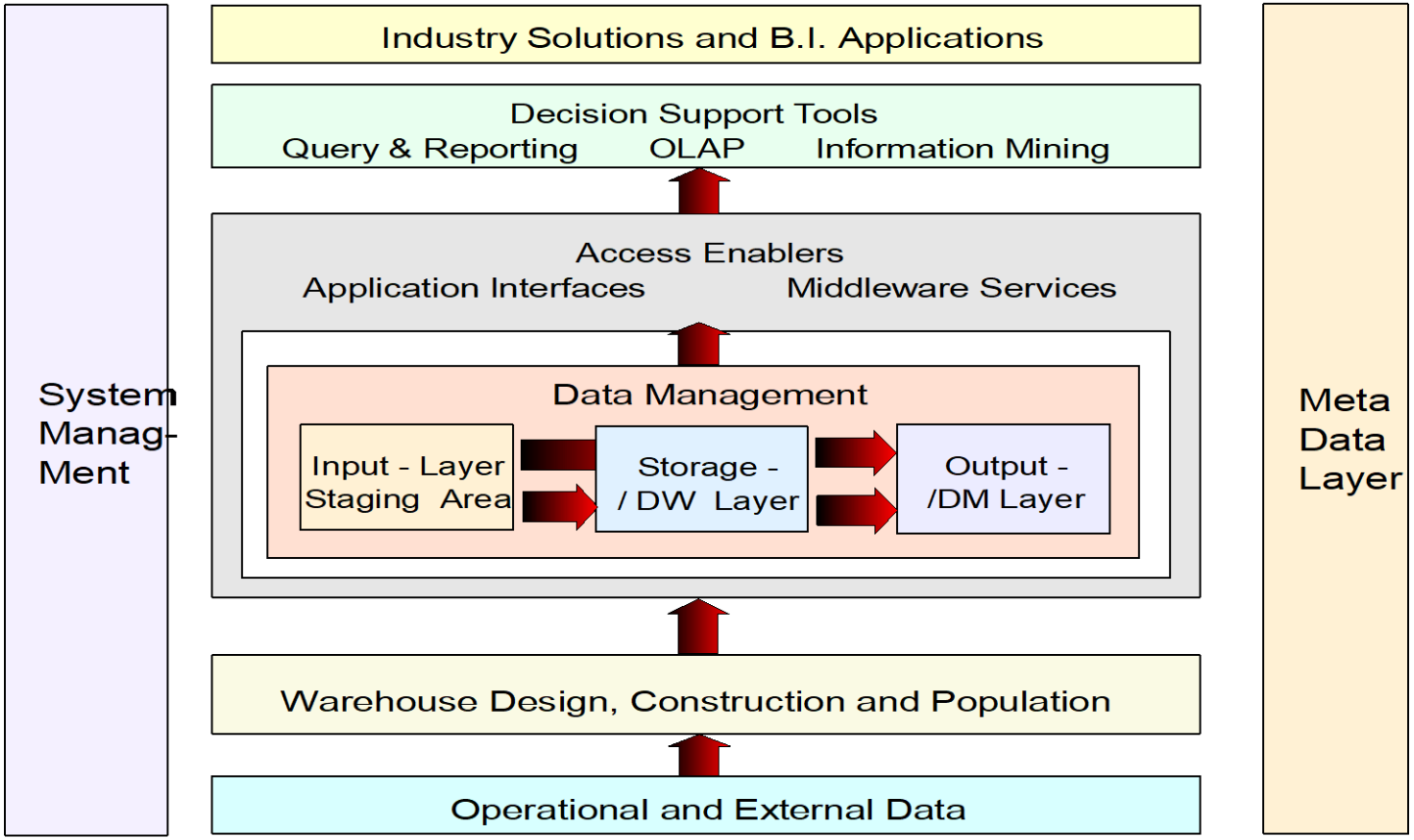
Hints & Tips for Building Data Warehouses

Dr. Hermann Völlinger
Consulting IT Architect
IBM SWG – Solution Architecture

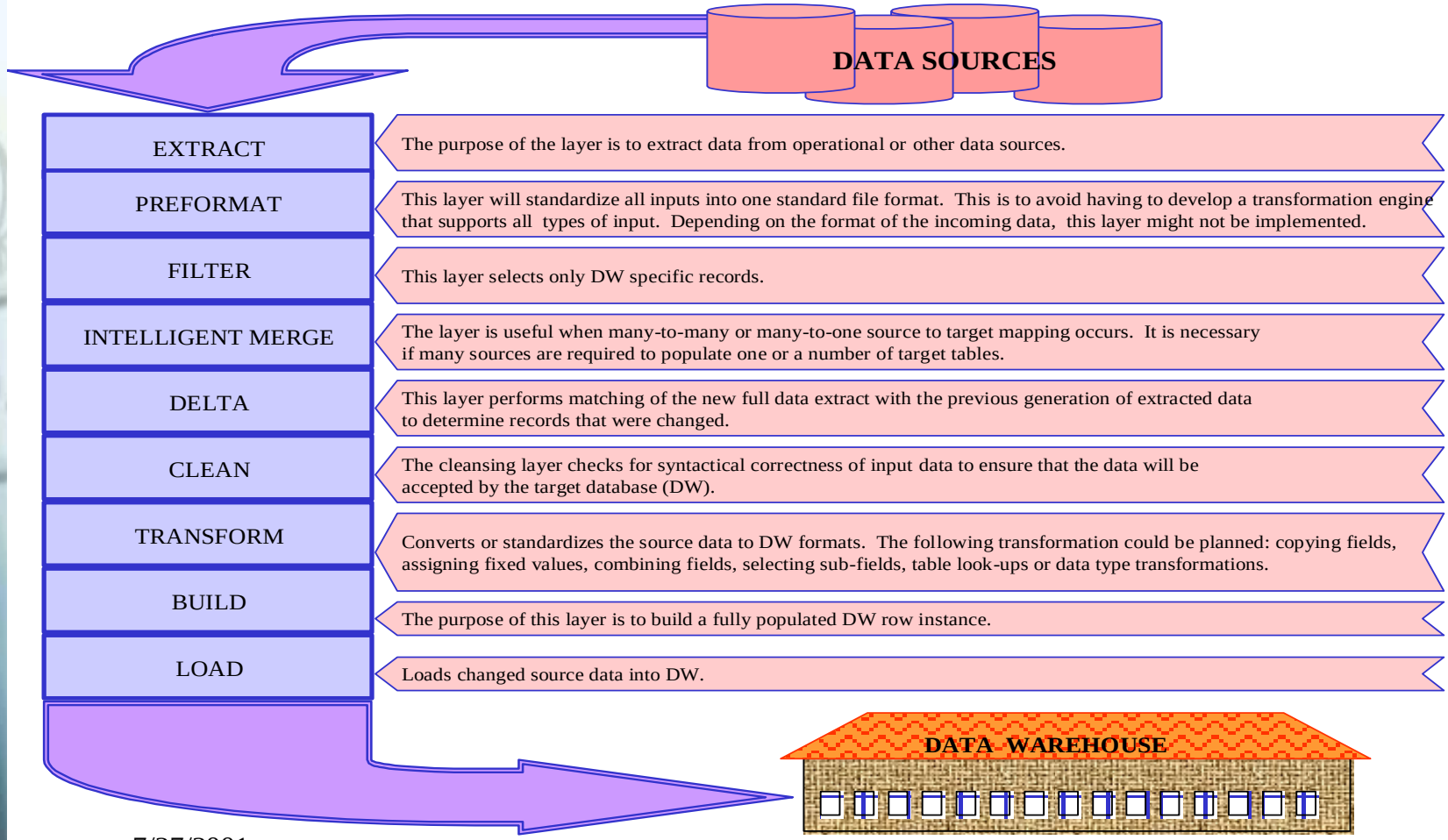
Ten Principles of a good ETL & DWH Architecture - Overview

1. **ETL & DWH Process Layer Concept**
2. **Framework / Control of Processes**
3. **Scalability & Parallel Processing**
4. **Central Metadata Repository**
5. **Integration of ETL and DB**
6. **Special Technique –** Piping
7. **Special Technique -** Piping & Parallel Load
8. **Historization**
9. **Delta Load**
10. **Key Attributes Transformation**

Process Layers of the DWH

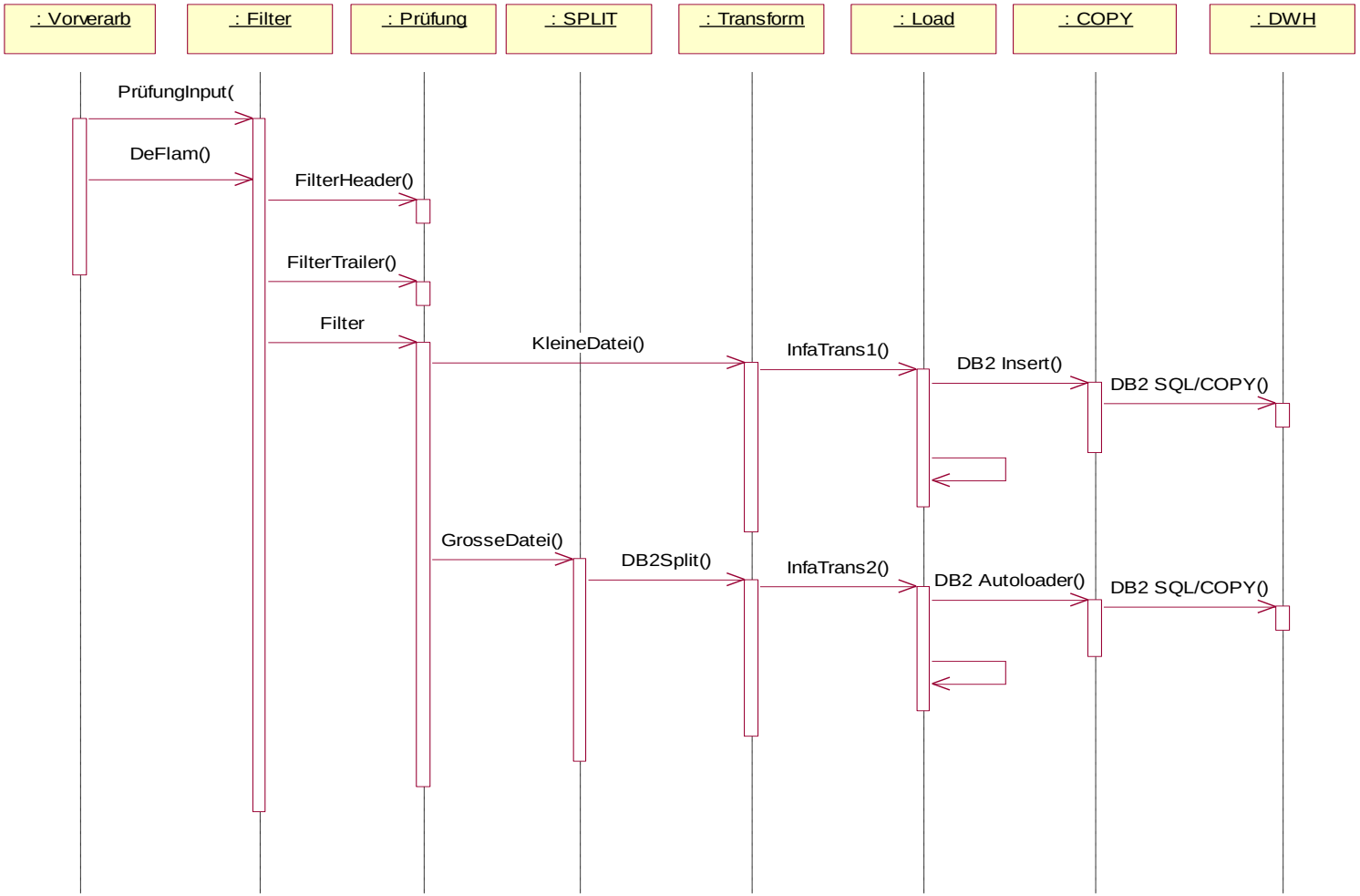


Generic ETL Process Layers

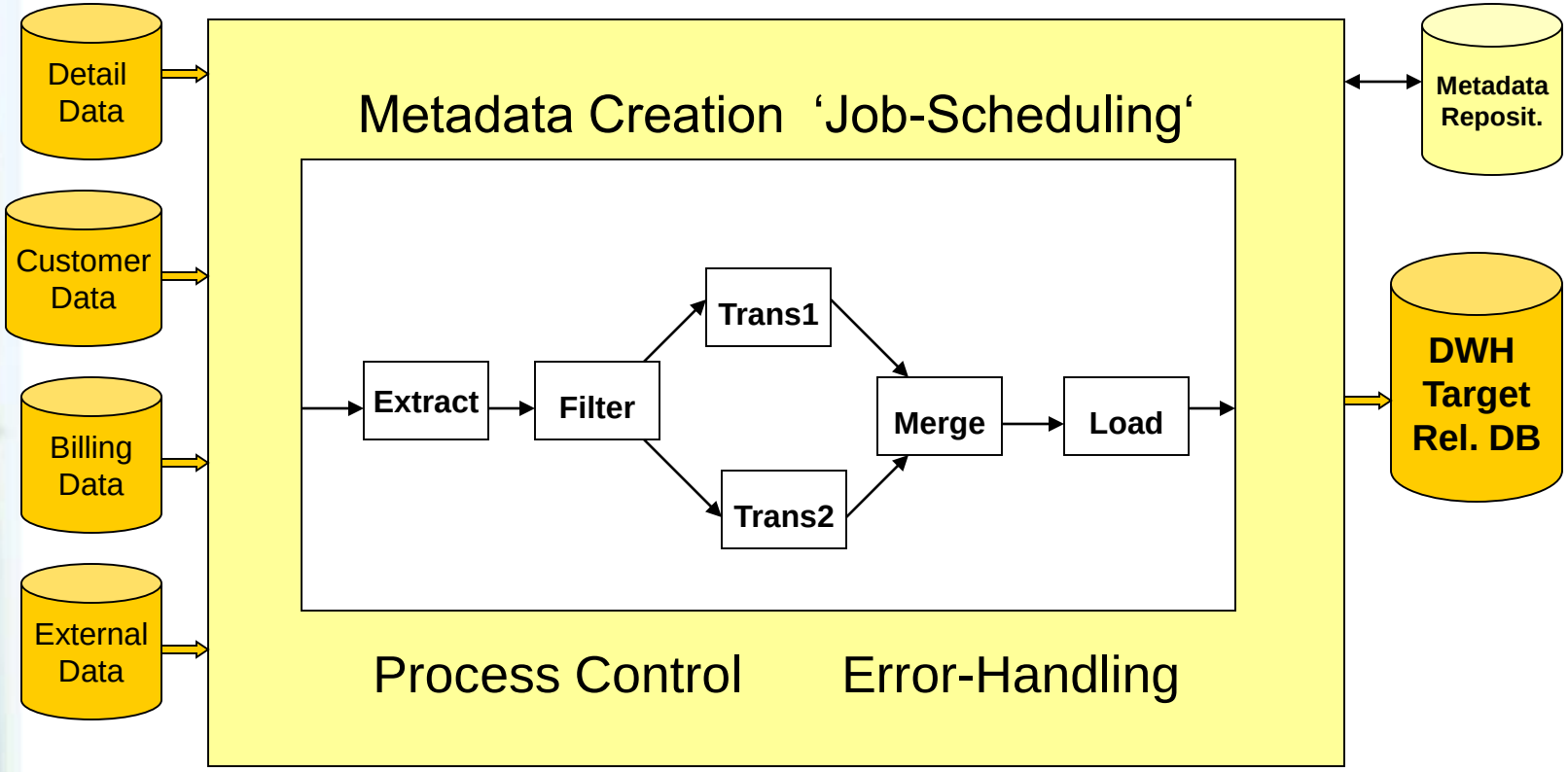


7/27/2001

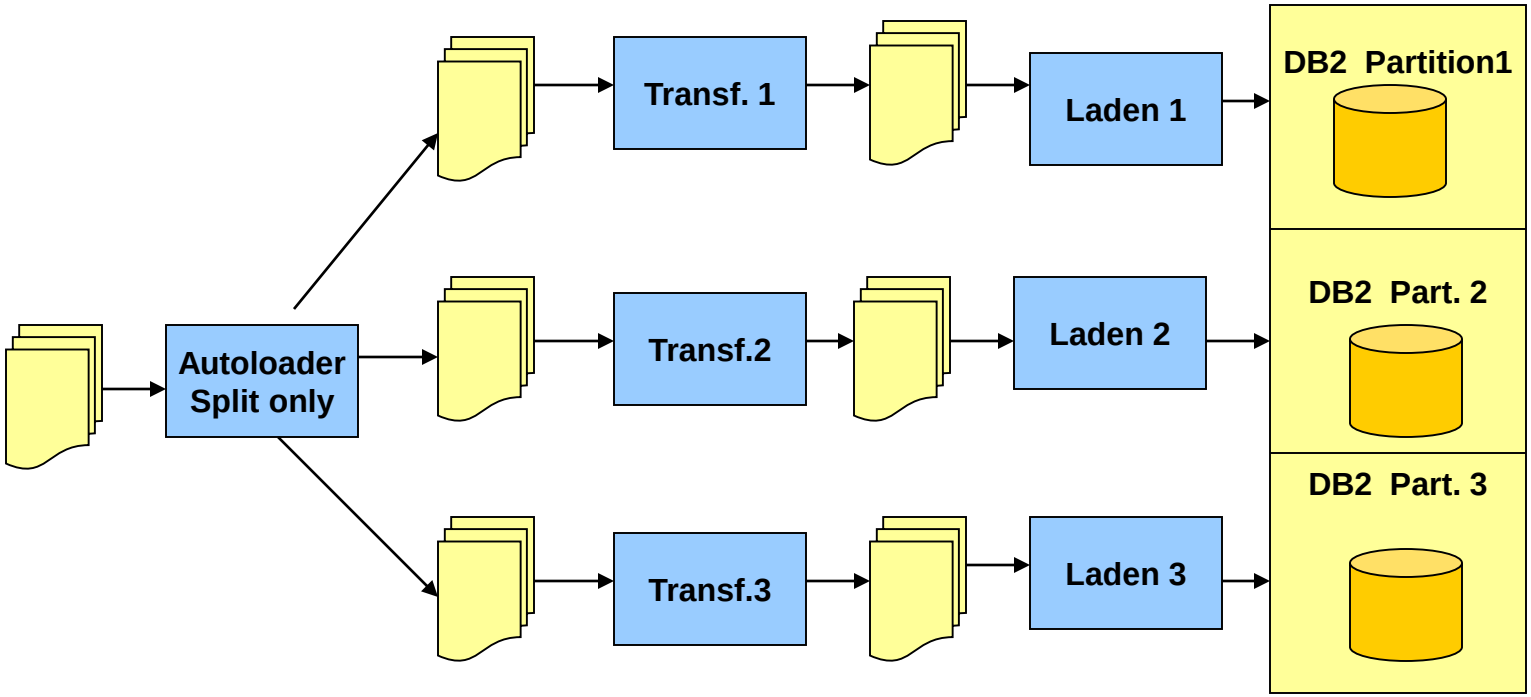
ETL Layer Concept (Example)



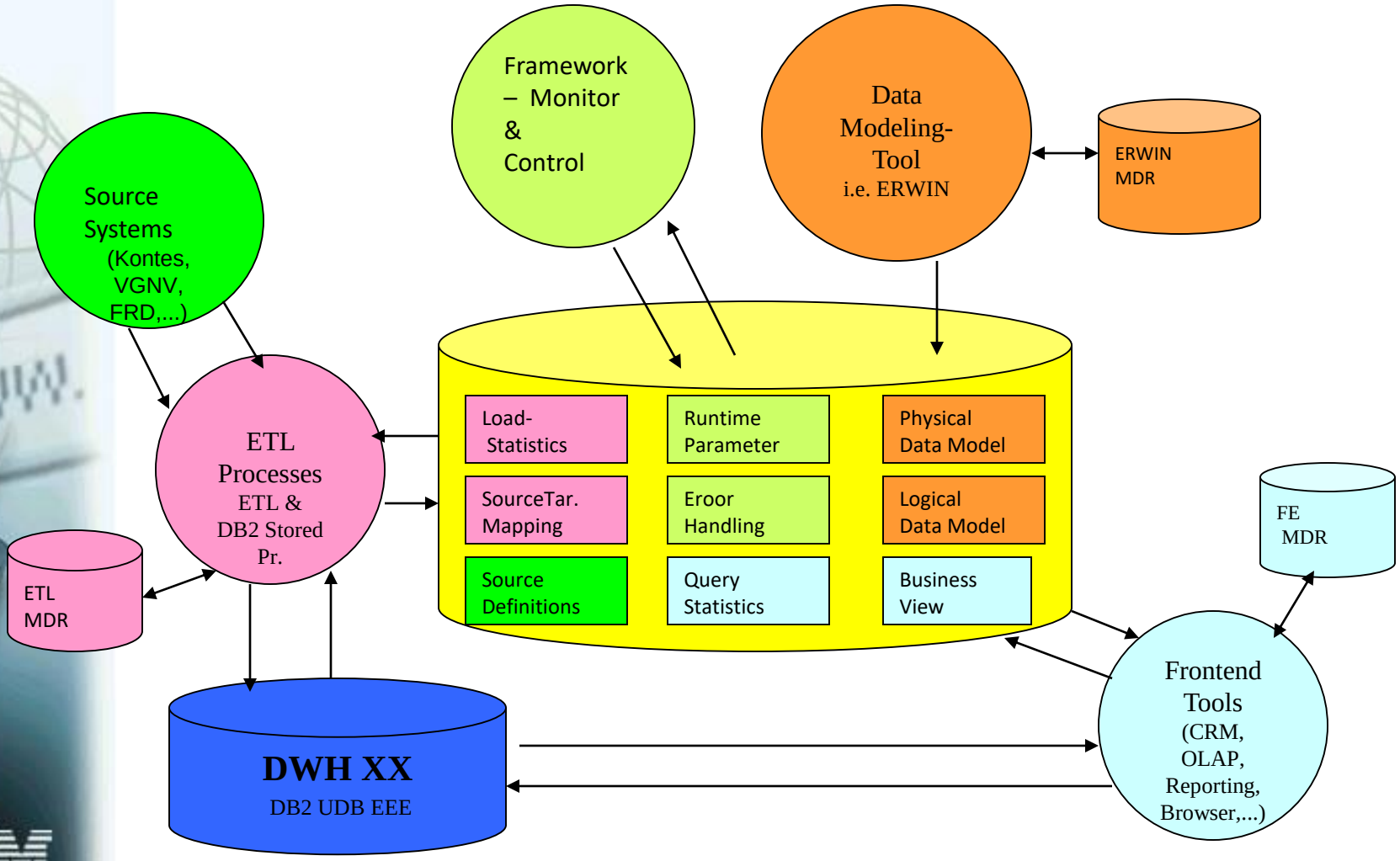
FRAMEWORK / Control of Processes



Scalability & Parallel Processing

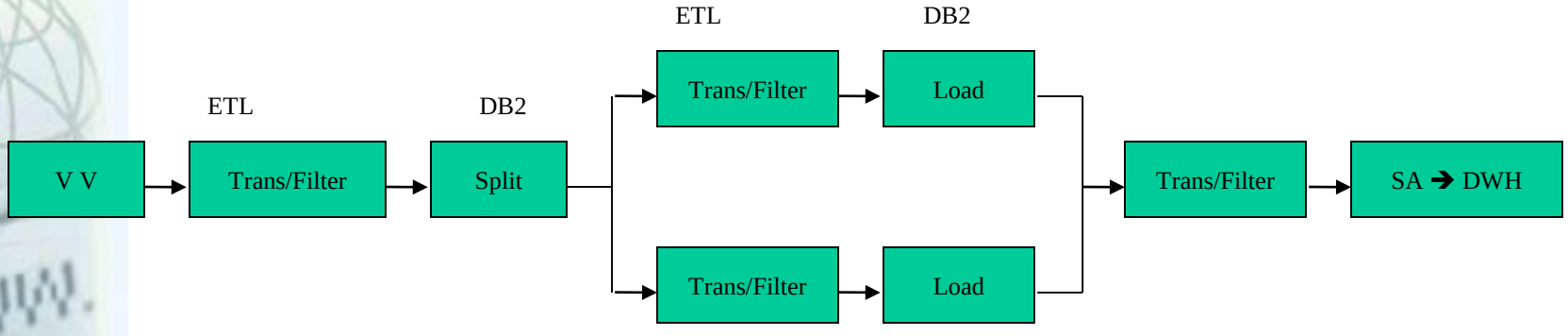


Concept of a Central Metadata Repository - cMDR



Integration of ETL & Database

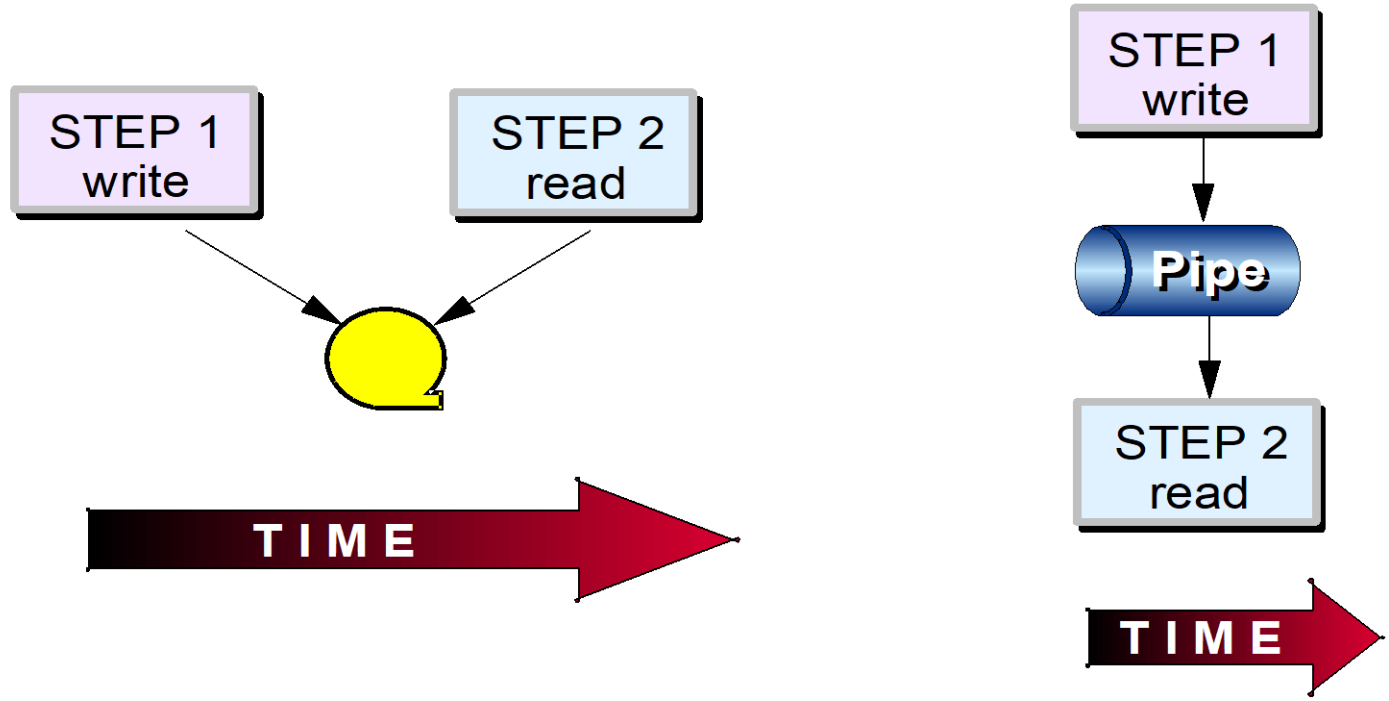
(Variante N)



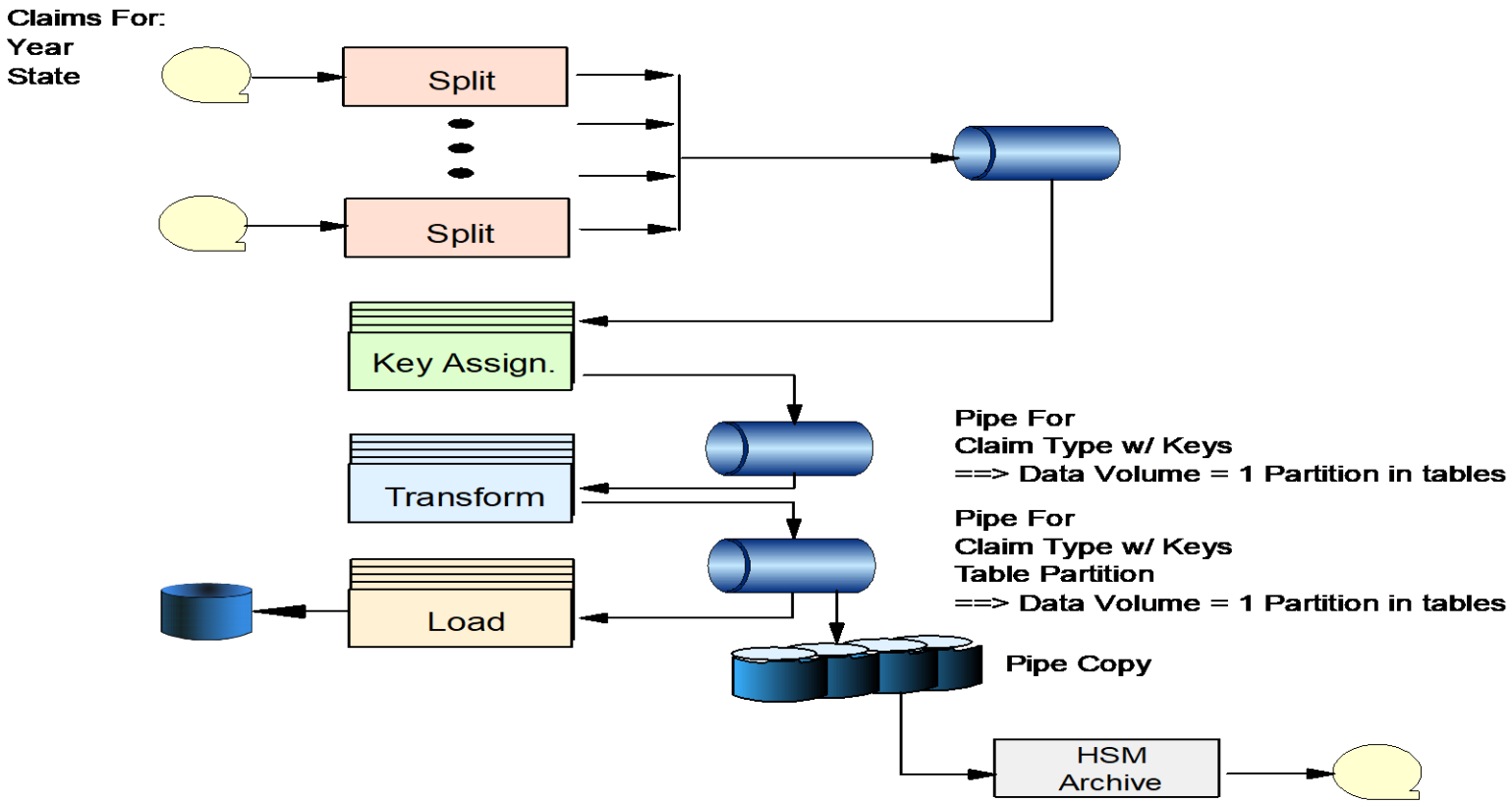
- Parallel Transformation, dependant from DB2 partitions (db2split)
- Performance: dependant from ETL & DB2 Load
- Piping versus temp. Flat Files
- ETL calls DB2 Autoloader (with Split Only)

ETL Technique – 'Piping'

- Manage workload, optimize data flow between parallel tasks
- Reduce I/Os

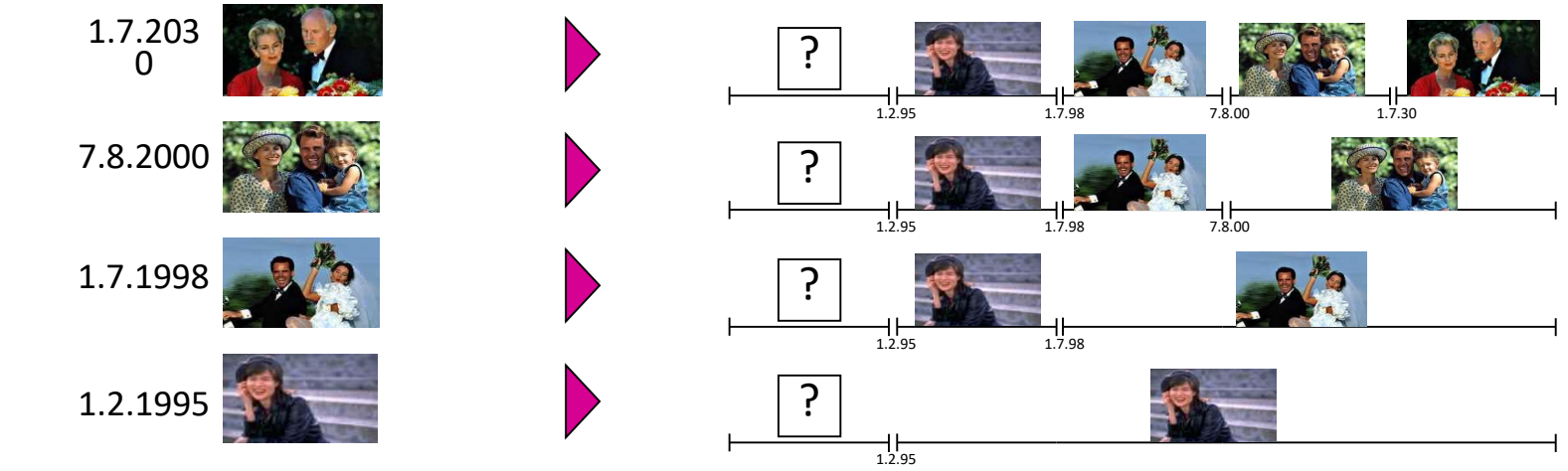


Example for the Combination of Techniques – ‘Piped Design‘ & Parallel Processing

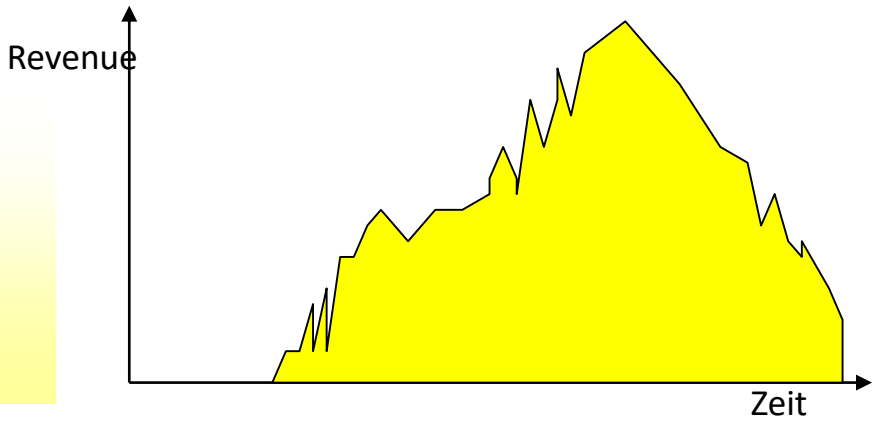


CPU Utilization Approximation

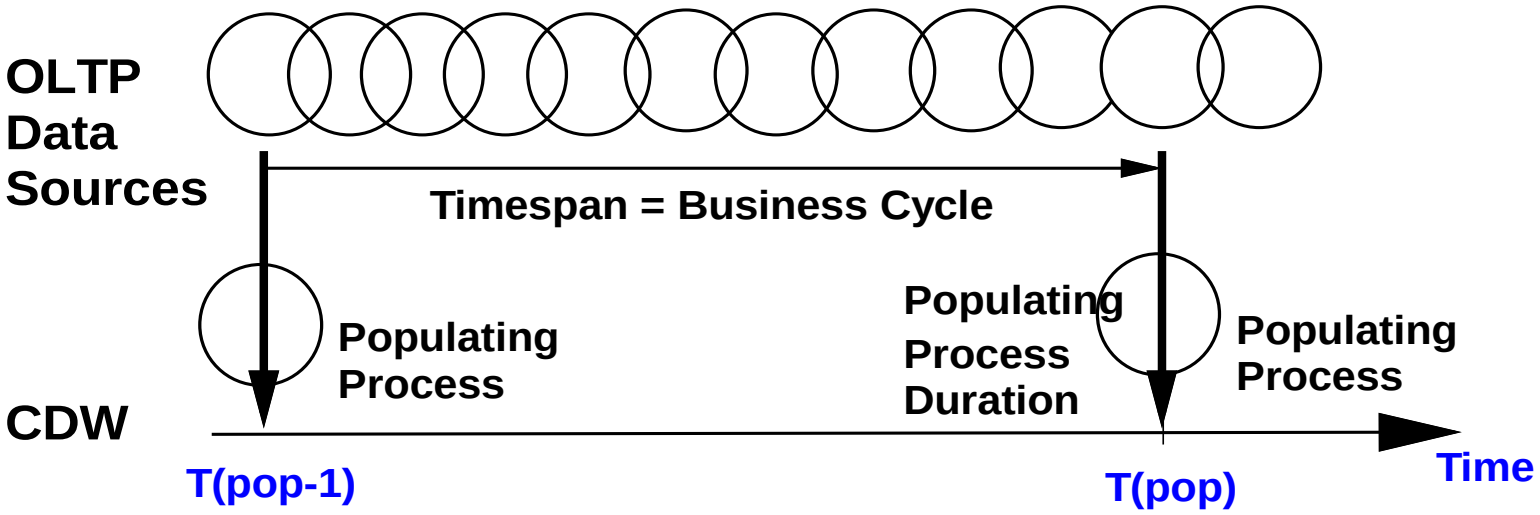
Motivation to Historization



- **Basis: Business Requirements**
- **Development of optimal Historization structures**
- **Efficient Storage of data ?**
- **Simple Actualization (ETL)?**
- **Simple Analysis?**

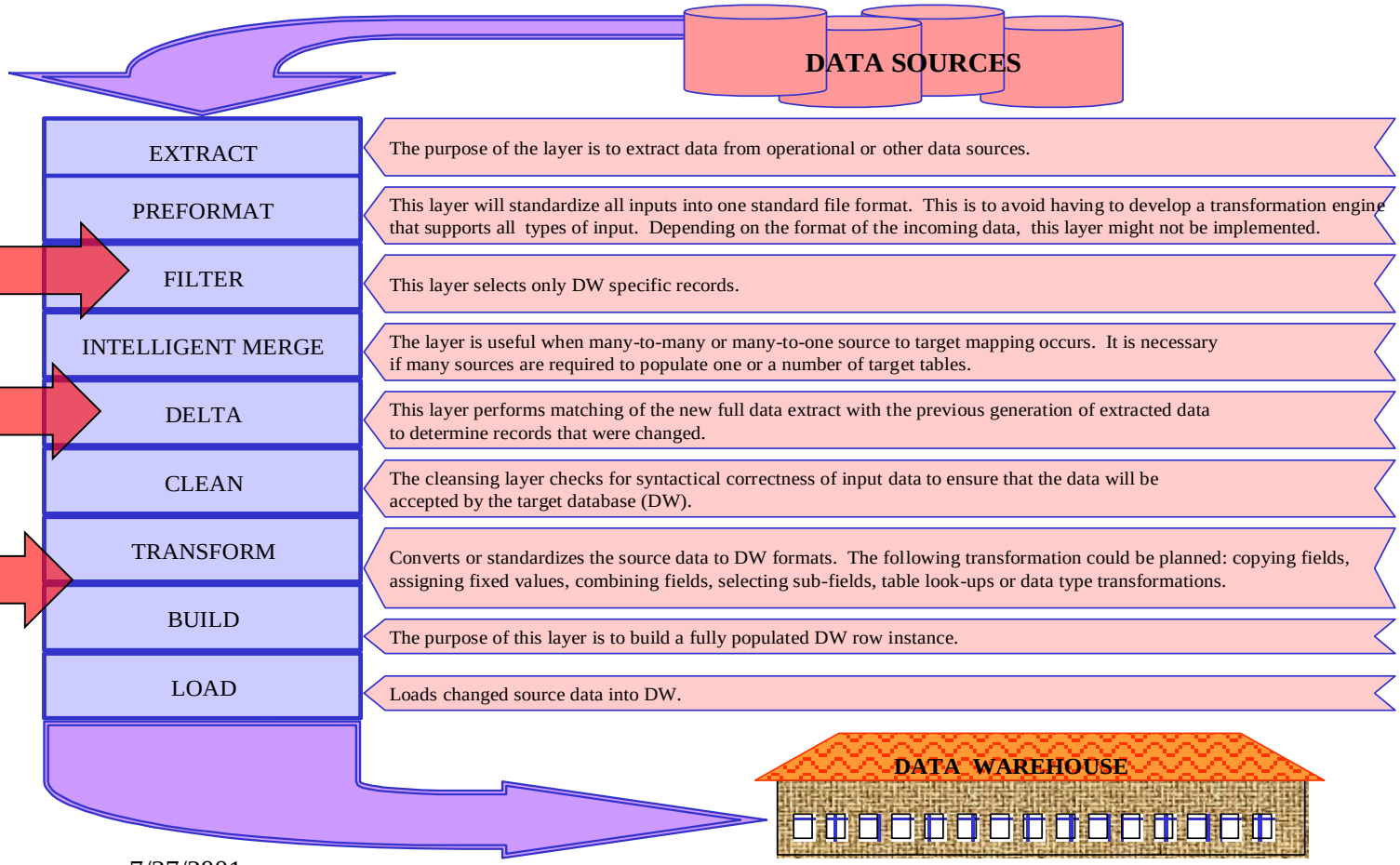


Delta Load - Motivation



- **OLTP data sources: continuously changing**
  - Database transactions <> Business transactions and processes
- **CDW must move from one business consistent state to the next**

Delta - Alternativen



7/27/2001

Delta - Discussion

- As sooner you ran the Delta, as smaller is the amount of data which needs to be transformed.
  - **But:** Changes in attributes with no business meaning can lead to an invalid Delta
  - **But:** Key attributes of a Delta must fit to key attributes of the table, instead you come to a wrong Delta
  - **But:** Delta before the Intelligent Merge makes the Merge more complex, since not all merge criteria's are known
- As later you ran the Delta, as exacter it is.
  - **But:** you invest much power in unnecessary transformations

Key Attributes Transformation Motivation

- Business Reasons
  - Operative Keys are not unique
    - over system limits
    - over the time
  - Operative keys didn't exist
- Technical Reasons
  - Operative keys „to long“
  - Candidate for partitioning

Challenges

- Position of key attribute Transformation
  - Beside or before the Merge?
  - Beside the Transformation?
  - During the build of a data record ?
- Algorithms of defining a key
  - Growing numbers
  - be a number by chance
- Parallel definition of keys
- Incomplete keys during reading or generating of a key transformation