

# 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

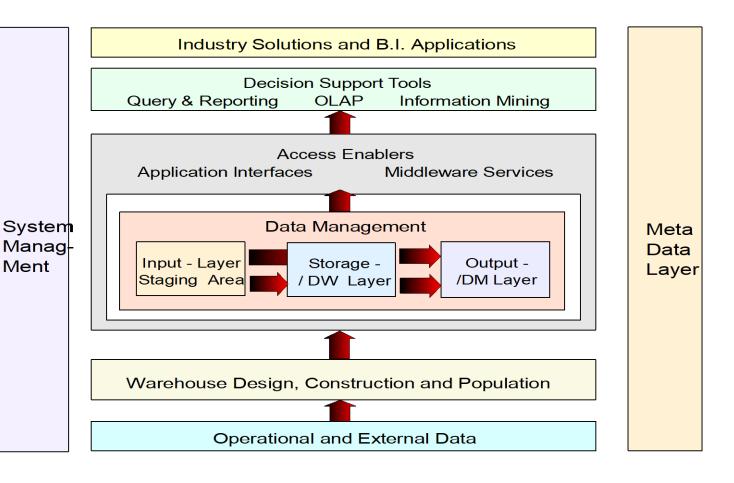


- 1. ETL & DWH Process Layer Concept
- 2. Framework / Control of Processes
- 3. Scalability & Parallel Processing
- 4. Central Metadata Repository
- 5. Integration of ETL and DB
- 6. Special Technique Piping
- 7. Special Technique Piping & Parallel Load
- 8. Historization
- 9. Delta Load
- 10. Key Attributes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



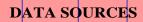
## **Process Layers of the DWH**







#### **Generic ETL Process Layers**



**EXTRACT** 

**PREFORMAT** 

**FILTER** 

INTELLIGENT MERGE

**DELTA** 

**CLEAN** 

**TRANSFORM** 

**BUILD** 

LOAD

The purpose of the layer is to extract data from operational or other data sources.

This layer will standardize all inputs into one standard file format. This is to avoid having to develop a transformation engine that supports all types of input. Depending on the format of the incoming data, this layer might not be implemented.

This layer selects only DW specific records.

The layer is useful when many-to-many or many-to-one source to target mapping occurs. It is necessary if many sources are required to populate one or a number of target tables.

This layer performs matching of the new full data extract with the previous generation of extracted data to determine records that were changed.

The cleansing layer checks for syntactical correctness of input data to ensure that the data will be accepted by the target database (DW).

Converts or standardizes the source data to DW formats. The following transformation could be planned: copying fields, assigning fixed values, combining fields, selecting sub-fields, table look-ups or data type transformations.

The purpose of this layer is to build a fully populated DW row instance.

Loads changed source data into DW.

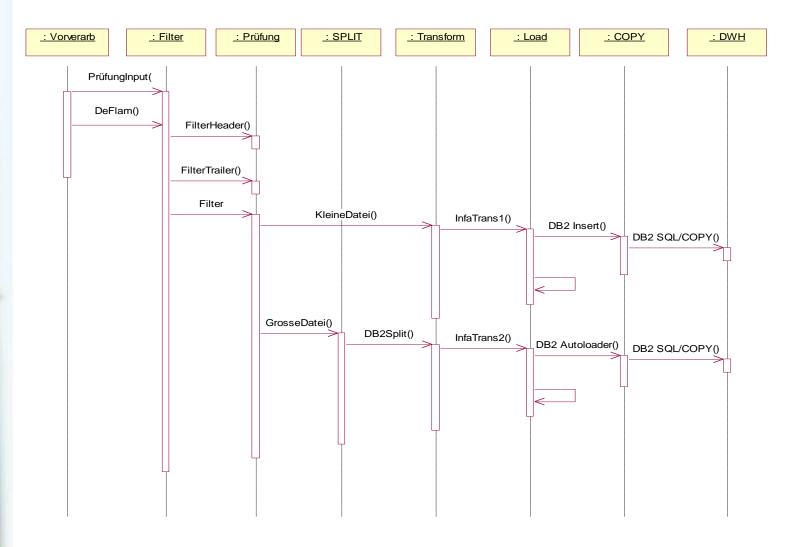


7/27/2001





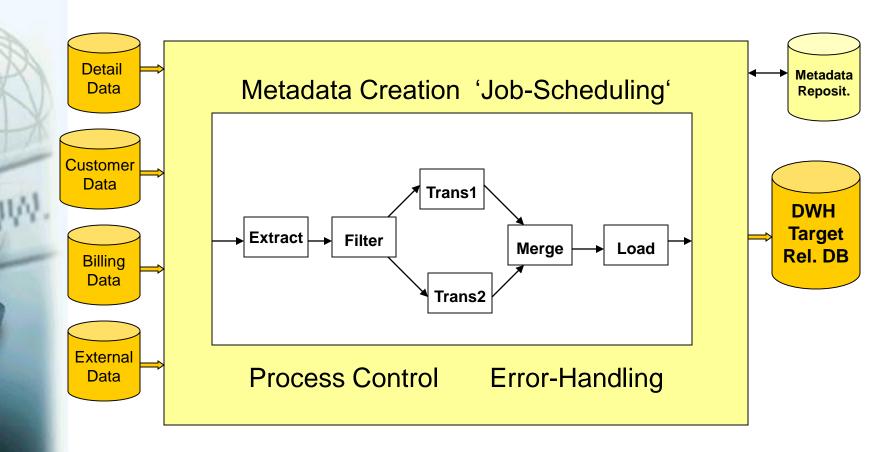
## **ETL Layer Concept (Example)**







#### FRAMEWORK / Control of Processes

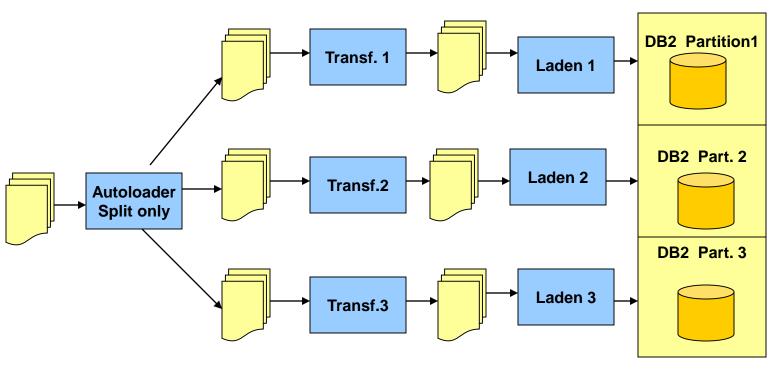






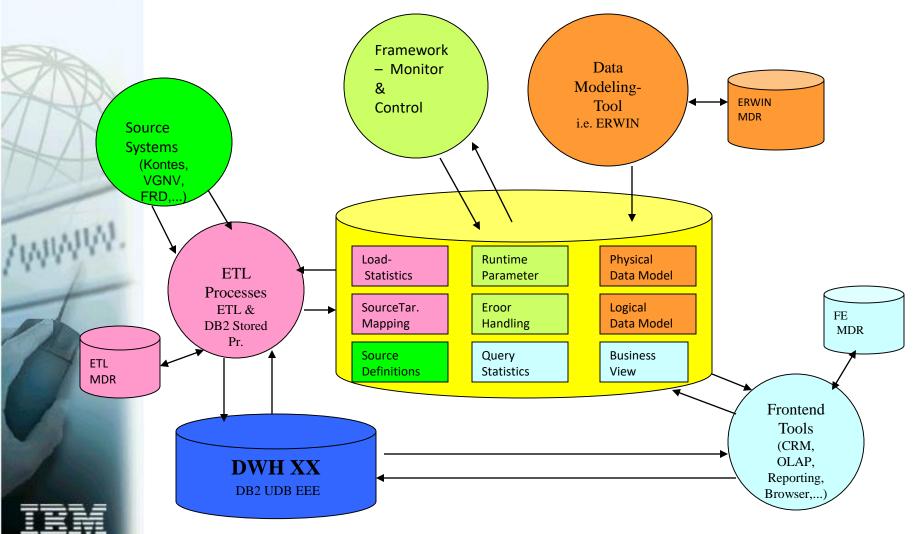
## **Scalability & Parallel Processing**







### **Concept of a Central Metadata Repository - cMDR**

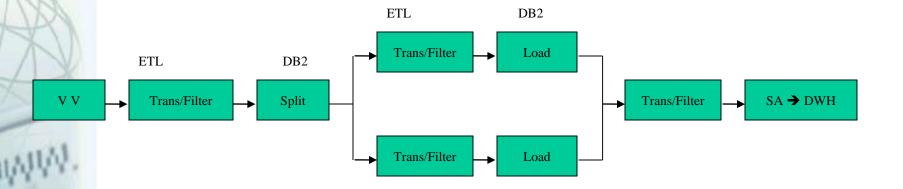






### Integration of ETL & Database

( Variante N )



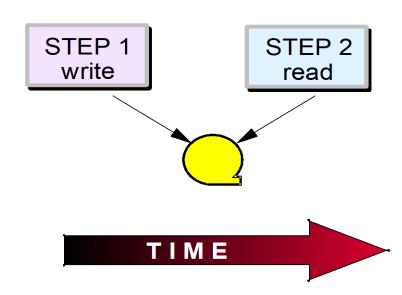
- Parallel Transformation, dependant from DB2 partitions (db2split)
- Performance: dependant from ETL & DB2 Load
- Piping versus temp. Flat Files
- ETL calls DB2 Autoloader (with Split Only)

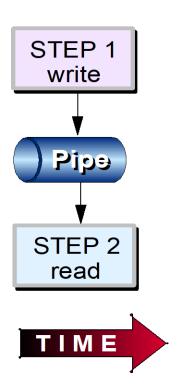




## ETL Technique – 'Piping'

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





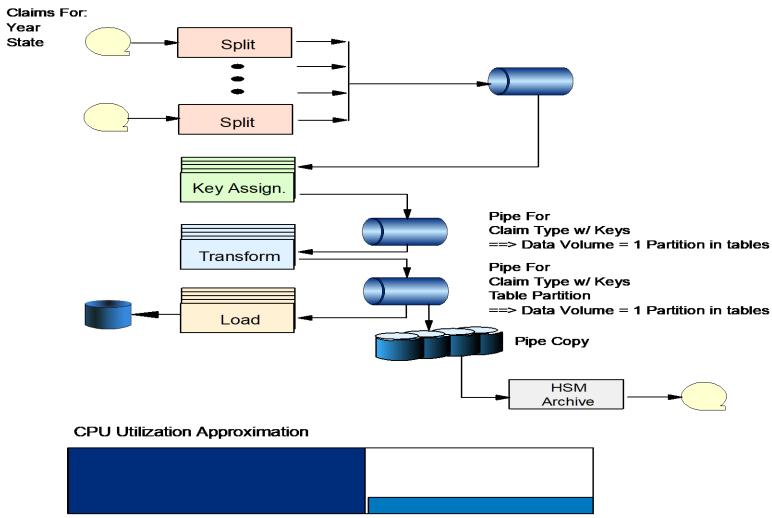


Folie: 10

Dr. H. Völlinger, IBM SWG



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







#### **Motivation to Historization**

Changes

1.7.203

7.8.2000

7.8.2000

1.7.1998

1.2.1995

1.2.95

1.2.95

1.2.95

1.2.95

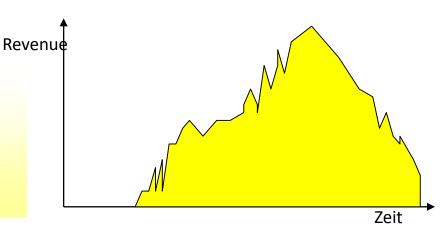
1.2.95

1.2.95

1.2.95



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

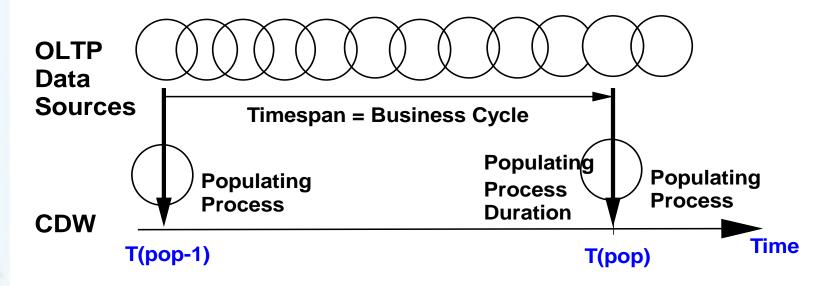




Folie: 12 Dr. H. Völlinger, IBM SWG



### **Delta Load - Motivation**



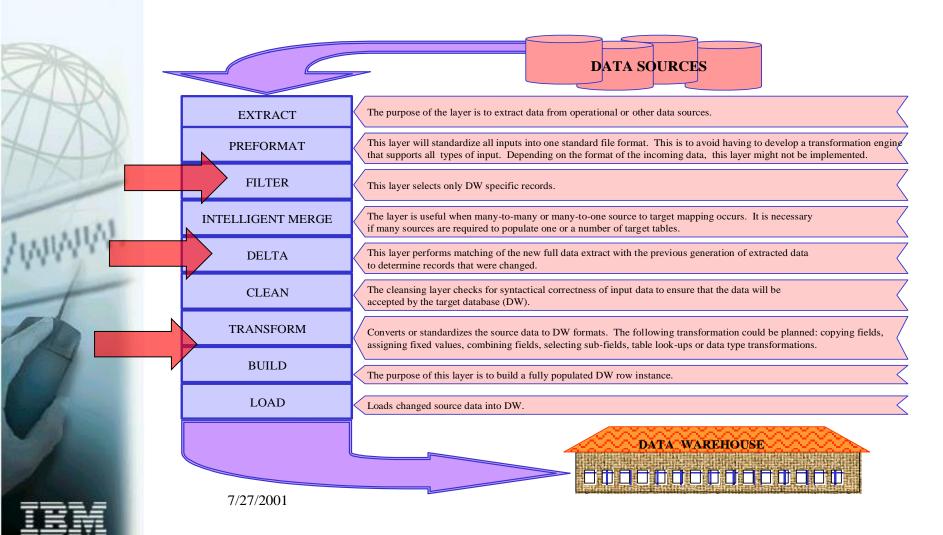
- OLTP data sources: continuously changing
  - Database transactions <> Business transactions and processes
- CDW must move from one business consistent state to the next



Folie: 13



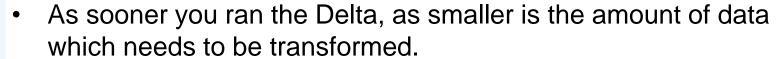
#### **Delta - Alternativen**



Folie: 14



#### **Delta - Discussion**



- 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**



- 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

