Replacing IBM IMS DB
–
Fully Automated and with Highest Security

The perfect Way to better Software

Delta Software Technology

For more than 45 years Delta stands for automated software development and maintenance.

We help organizations make proven applications fit for the future through…
… automated knowledge discovery,
… automated clean-up,
… automated migration and modernization.

We <u>know</u> COBOL, PL/I and Delta ADS!

Customers in Europe and beyond

Owner-managed family business (2nd generation)
CEO: Dr. Daniela Schilling

Why replacing IMS DB?

Reasons meantioned by our customers
- Parallel usage of different technologies
  - Effort and costs
  - Complexity and restrictions
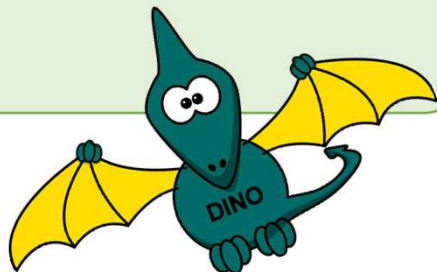- Limited availabilty of data
- Platform change
- Dwindling knowhow
- …

Two customer examples: DINO and GoBIMS

PROVINZIAL

DINO: Project facts

- 36.500 Sources (COBOL-programs + copybooks)
- 70 Mio Lines expanded code
- 16.000 database accesses
- 75 IBM IMS-databases
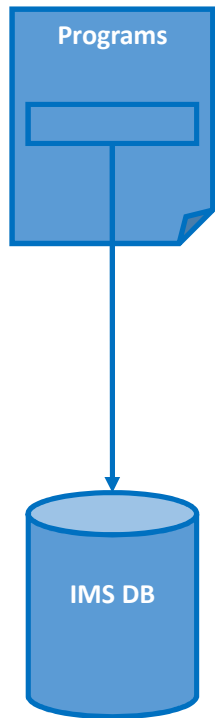
Pixabay.com

Gothaer

GoBIMS: Project facts

- 45.000 Sources (COBOL & Delta ADS)
- 50 Mio Lines expanded code
- 5.500 database accesses
- 447 IBM IMS-databases (incl. history)
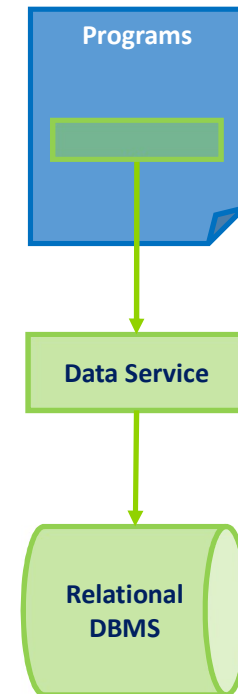 - 358 replaced by Delta

Challenges and Requirements

Programs

IMS DB

Automated Transformation

Programs

Data Service

Relational DBMS

New data model according to customer's requirements

Despite of paradigm change:
No change of application logic or datastructures

No restrictions of regular maintenance and daily business!
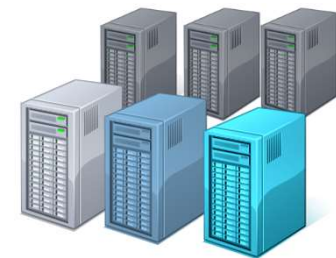
Automation Assumption

"Peopleware"

If you can explain the rules to somebody somewhere,

you should be able to define them for a computer.

If you don't know the rules,

you would better hesitate to do anything at all.

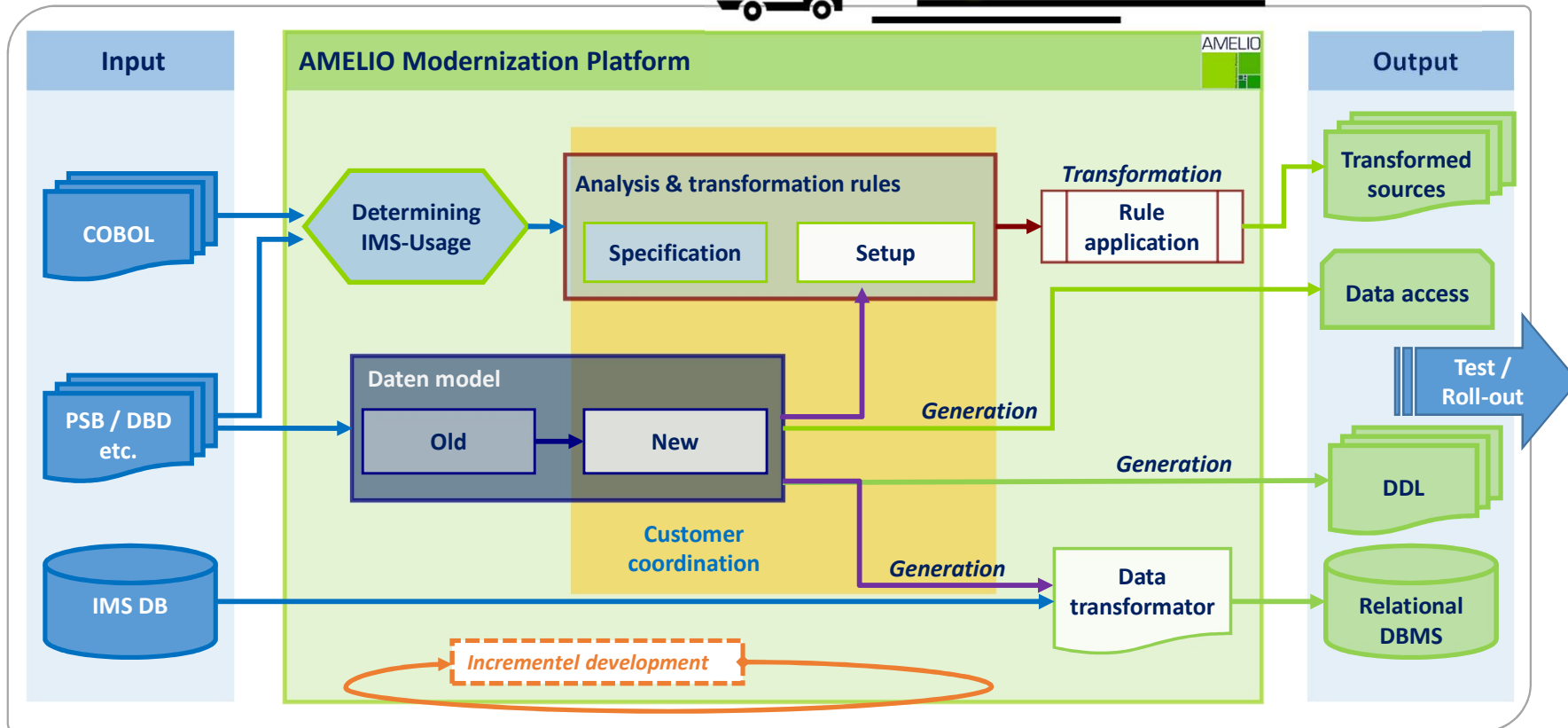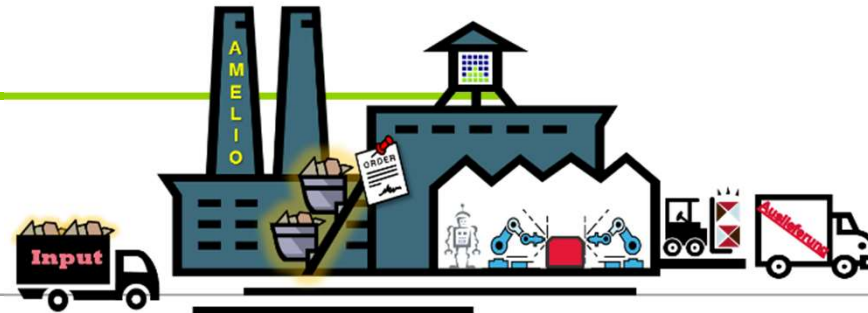Hard- and Software

AMELIO Modernization Platform

Fully automated modernization factory
- Individually made from prefabricated configurable components
- Clean Room-concept
  - Completely automated and controlled processes
- Model-driven and rule-based

The Factory

Input

- COBOL
- PSB / DBD etc.
- IMS DB

AMELIO Modernization Platform

Determining IMS-Usage

Analysis & transformation rules
- Specification
- Setup

Transformation

Rule application

Daten model
- Old → New

Generation

Customer coordination

Generation

Generation

Data transformator

Incrementel development

Output

- Transformed sources
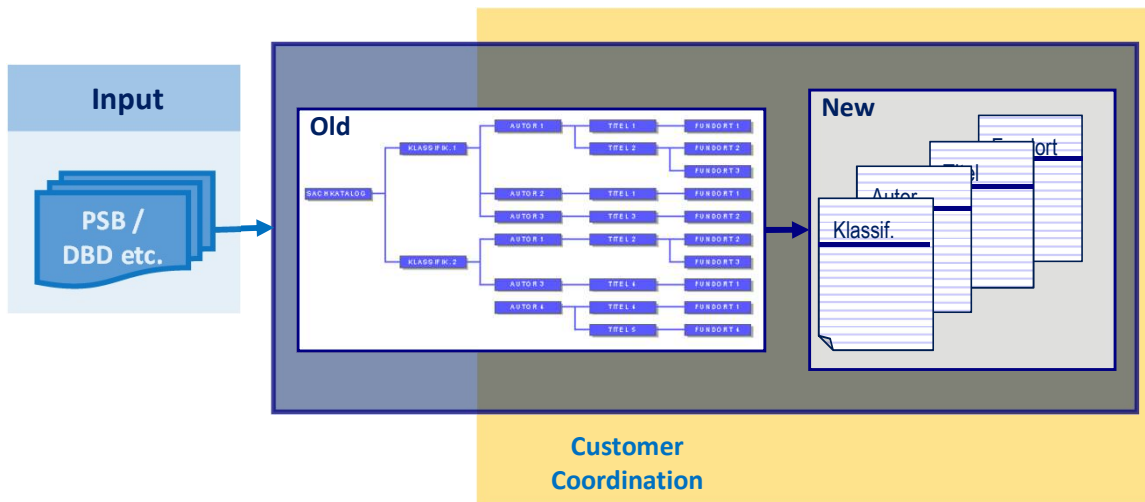- Data access
- Test / Roll-out
- DDL
- Relational DBMS

Defining the new Data Model

Defining the new data model and implementation rules

- Automated derivation of a proposal for a new data model using knowledge about
  - Existing database structures
  - Possible segment content
  - Demands for implementation of arrays, redefines, etc.
- Proposal will be discussed and adapted with customer

During the project

- Adaptation of the data model
  - Technical demands of application development
  - Performance optimization
  - …

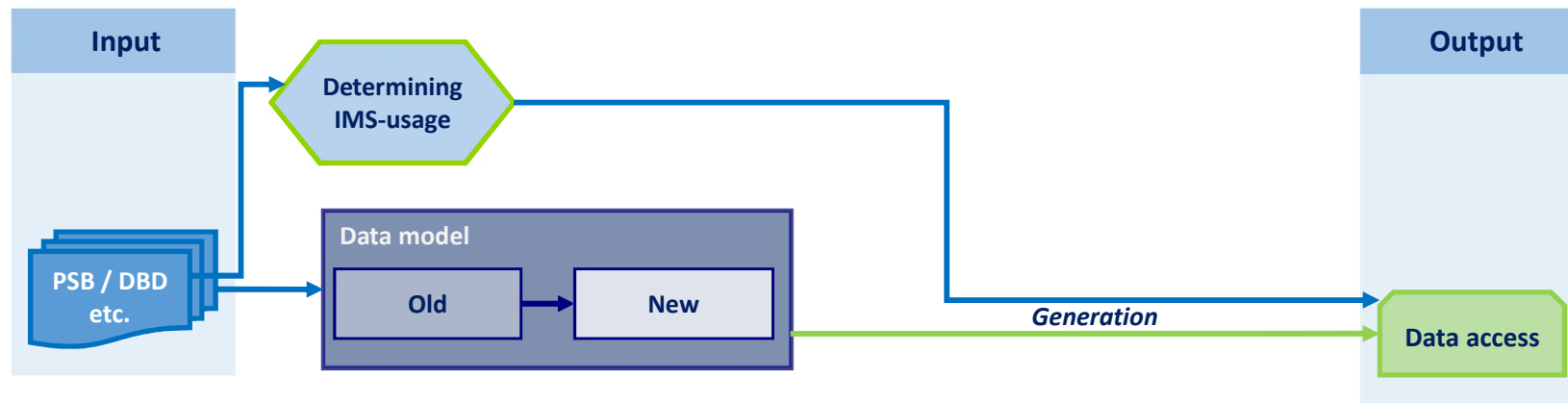# Application transformation
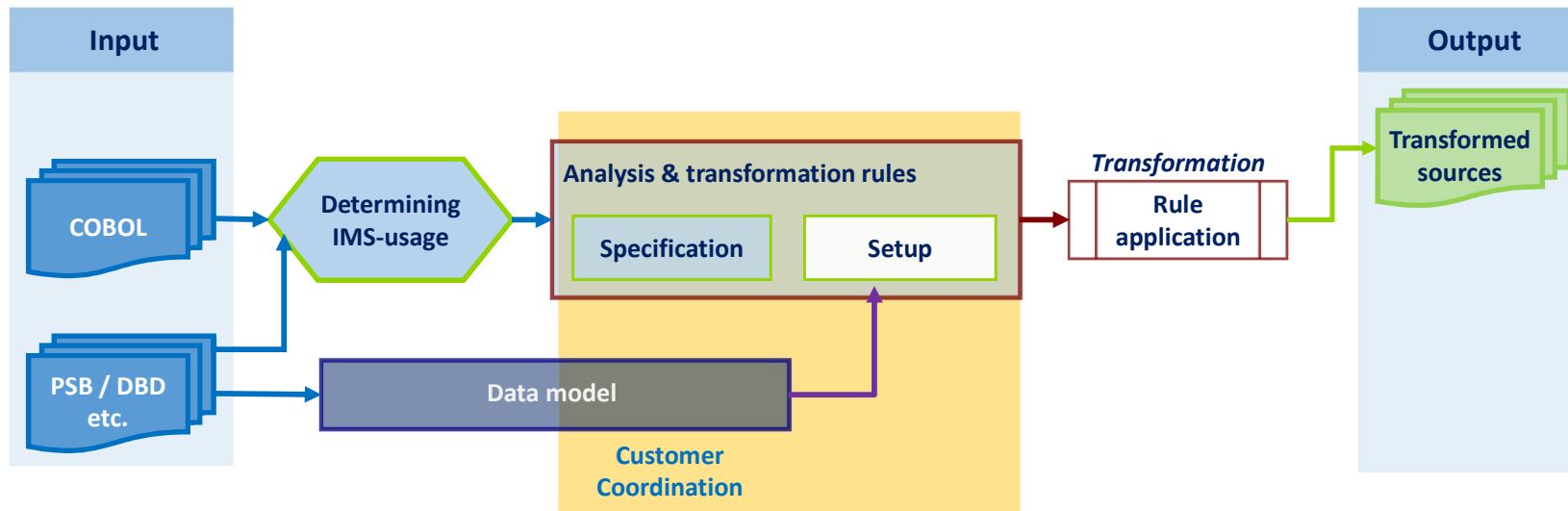
- Transformation of database accesses within programs and copybooks
  - Dependent on the context and the new data model

Generating database accesses

Challenges
- Paradigm change: from hierarchical to relational
- Structural characteristics as ARRAYS, REDEFINES,…
- ➢ 1:1-transformation not possible
- BUT: application logic and used data structures may not be changes!

Data services
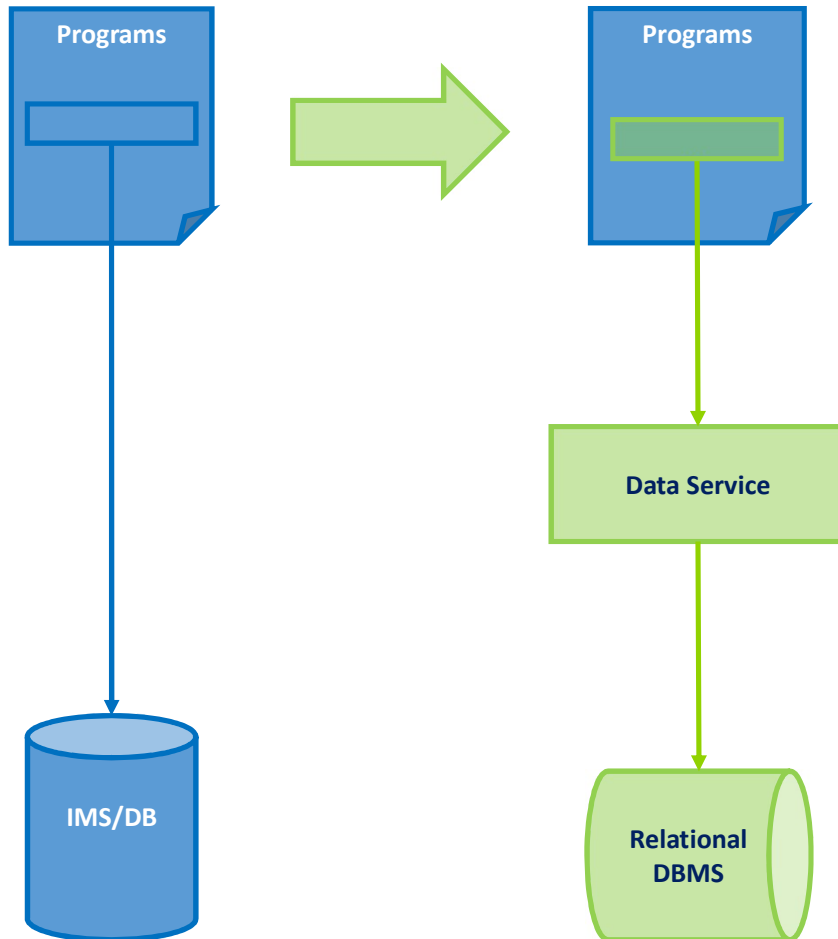
- Services handle data from relational database as expected by the program
  - Despite change of paradigm
- SQL-specific coding completely encapsulated within the service
- IMS-usage in programs determine accesses contained in the service
- IMS PCB informationen get emulated (e.g. ReturnCode, ConcatKey)
- Static SQL if possible
- Standardised error-handling
- Can be generated automatically, have to be maintainable manually, have to fullfill customer-specific coding standards

Optional: Data Migration
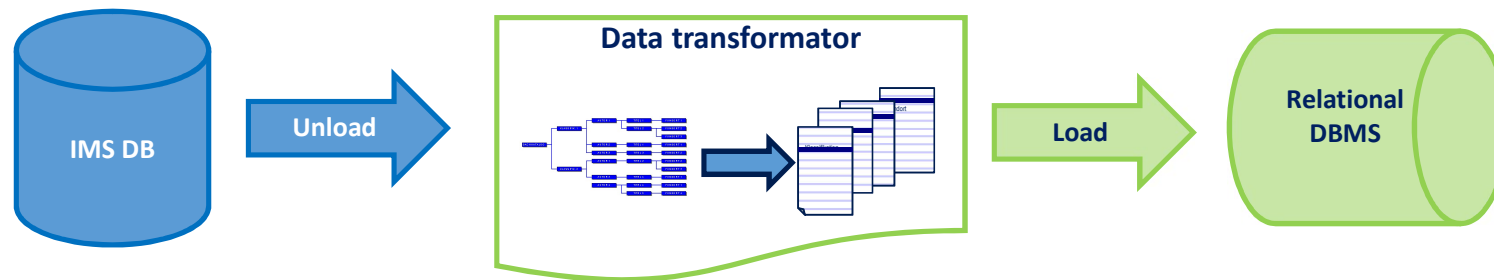
Program generation to transform the data sets
- Using existing mapping rules

Advantages
- Usage of the same data models for application and data transformation
  - Changes to the data model update the application and data transformation synchronously
- Documentation of data transformation
  - Numbers
  - Errors

The Factory

AMELIO Modernization Platform

Input:
- COBOL
- PSB / DBD etc.
- IMS DB

Determining IMS-Usage

Analysis & transformation rules
- Specification
- Setup

Transformation
- Rule application

Daten model
- Old → New

Customer coordination

Generation

Incrementel development

Data transformator

Output
- Transformed sources
- Data access
- Test / Roll-out
- DDL
- Relational DBMS

Setup the Factory

Development and Maintenance

Sources — Sources — Sources

Setup Basis-Factory → Train Factory → Transformation → (Meta-Level)-Test → Final Transformation → Roll-Out

Feedback

Analysis
Run-time-tests
Performance-tests

Setup a customized factory

- According to customer's requirements
- Parallel to ongoing development and maintenance
- Package-wise according to customer's demands
- Strategy change or adaptation possible at any time of the project

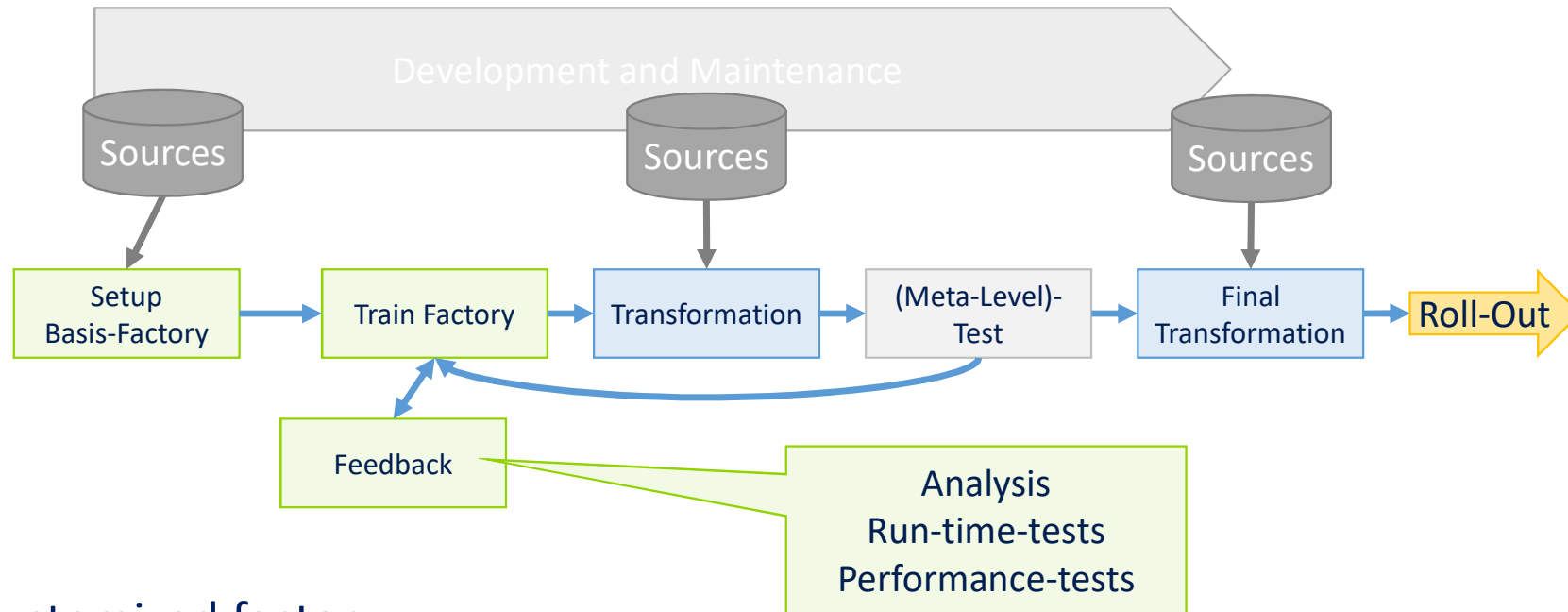# Test strategy: Meta-Level-Test

The principle:

> if an automaton applies a rule correctly once,
> *it will always apply it correctly*

Tests of all transformation rules
- … <u>not</u> of all changed pro...

Performed on a selecte...
- Automatically determin...
- Combination of all rules...

Factory performs coverage-analysis
- Which artifacts got transformed?
- Which transformation rule was applied to which artifact?

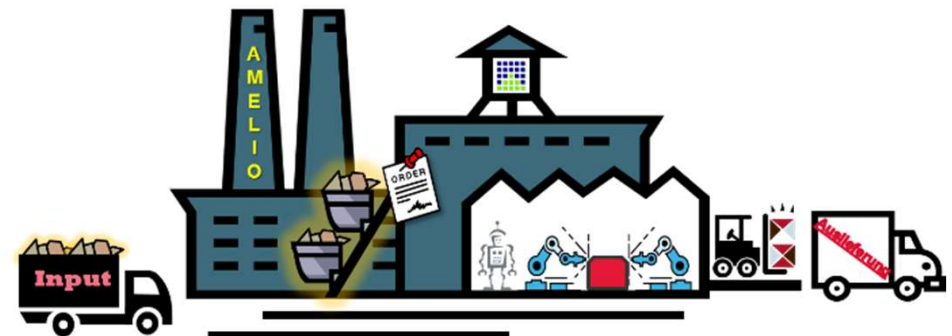| Name	Data Base	Function	Operation Name
Prog1	DBA	DLET	DELETEOBJECT
Prog1	DBA	GHU	GU-QU-PK-EQ-LOCK
	DBA	GHU	GU-QU-PK-EQ-LOCK
	DBA	GHU	GU-QU-PK-EQ-LOCK
	DBA	GN	GN-NQ
	DBA	GN	GN-NQ
	DBA	GN	GN-QU-SE3-EQ
	BA	GU	GU-NQ
	DBA	GU	GU-QU-PK-EQ-LOCK
Prog2	DBA	GU	GU-QU-PK-EQ-LOCK
Prog3	DBA	GU	GU-QU-PK-EQ-LOCK
Prog1	DBA	ISRT	INSERTOBJECT
Prog1	DBA	REPL	UPDATEOBEJCT

Quality and Security

Quality and Security

- By 100% automation
- Clean-Room-Concept
  - Completely automized and controlled processes
  - No „contamination" by manual intervention
- No personal style, no dependency of daily form
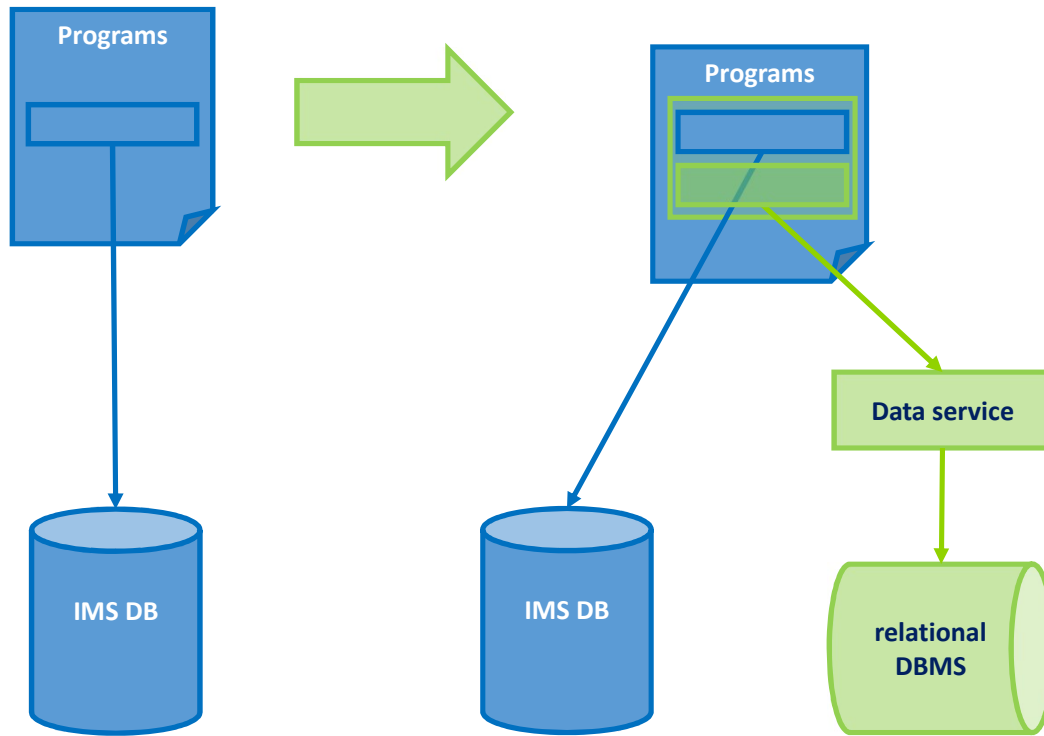- Reproducable, comprehensible

Security Mechanisms

- Documentation
- Optional: Switches and Verify

Documentation of Changes

Each performed code change can be automatically marked

```
*A_DOUBLE_VARNAME : C_CONFIRM: T_DOUBLE_VARNAME [RhHOyo_sU_113]    AM=REF
*        DISPLAY ' VNR PRUEFEN VERS.-SUMME ' BHS-0102-VNR-X        AM=UDL
         DISPLAY ' VNR PRUEFEN VERS.-SUMME ' BHS-0102-VNR-X OF     AM=UIN
         DBOE2100-1-G                                              AM=UIN
     END-IF.                                                       INS-C1C2
```

Name of the change rule

Replacing

```
*A_DLI_CALL_CBLTDLI : C_CONFIRM: T_CBLT [IhH7cnxsF_79]    AM=REF
*    CALL 'CBLTDLI' USING GHU-X BK2-NN2-PHY-PCB-G STATISTIK-G    AM=DEL
*                     SSA-BK2-Q-G                                AM=DEL
     MOVE 2 TO PROG-NOOFIMSCALL-D                                AM=INS
     MOVE 'STATISTIK' TO PROG-SECNAME-X                          AM=INS
```

Line(s) deleted

Line(s) inserted
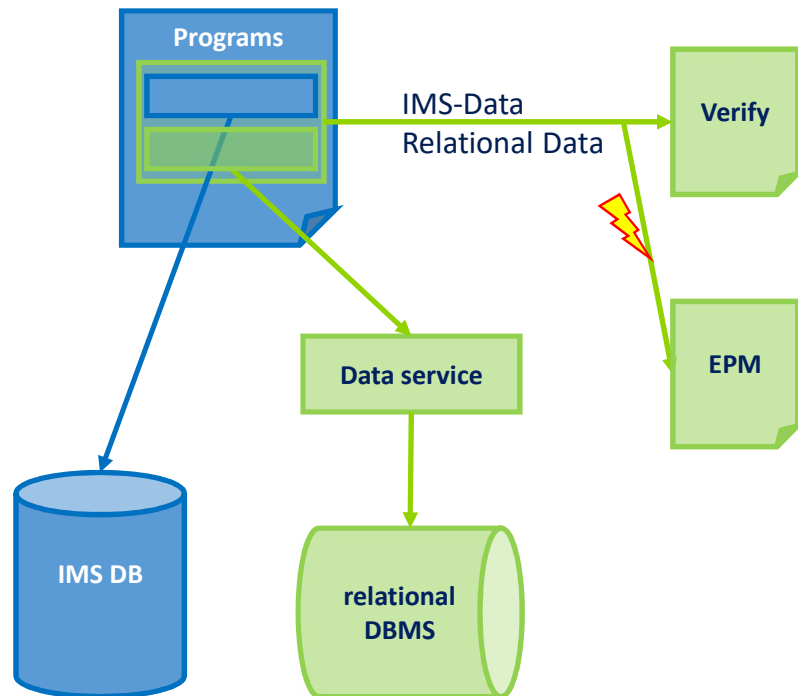
Optional: Switch

Switch

- Parallel access to IMS DB and relational database
- Switches (example)
  - 1: only accessing IMS DB
  - 2: accessing IMS DB and relational database,
    IMS/DB leading
  - 3: accessing IMS DB and relational database,
    relational database leading
  - 4: only accessing relational database
- One switch per IMS DB table

Optional: Verify and error protocol

Programs

IMS-Data
Relational Data

Verify

EPM

Data service

IMS DB

relational
DBMS

Verify
- While accessing IMS DB and relational database in parallel (switches 2 and 3)
- Automated comparison of the data
  - Incl. Consideration of special cases (e.g. Low-Value vs. Zero)

Error protocol (EPM)
- Complete and detailed documentation of revealed differences

Replacing IMS DB successfully

By automation
- ✓ Customized: adaptation of solution and processes to customer's requirements
- ✓ Flexibility: strategy changes with regard to solution and process possible at any time during the project
- ✓ Security: reproducable, testable and comprehensible (also for auditors)
- ✓ Performance: optimization possible together with the customer
- ✓ No restrictions to regular maintenance and daily business!

… and by close interaction with the customer!

The Perfect Way
to Better Software

Delta Software Technology GmbH
Eichenweg 16
57392 Schmallenberg, Germany

Daniela Schilling
CEO

phone (+49) 29 72 / 97 19-0
e-mail info@delta-software.com
internet www.delta-software.com