Dr. Gernot Schreib, b.telligent GmbH & Co.KG

DATENFLÜSSE IM DWH

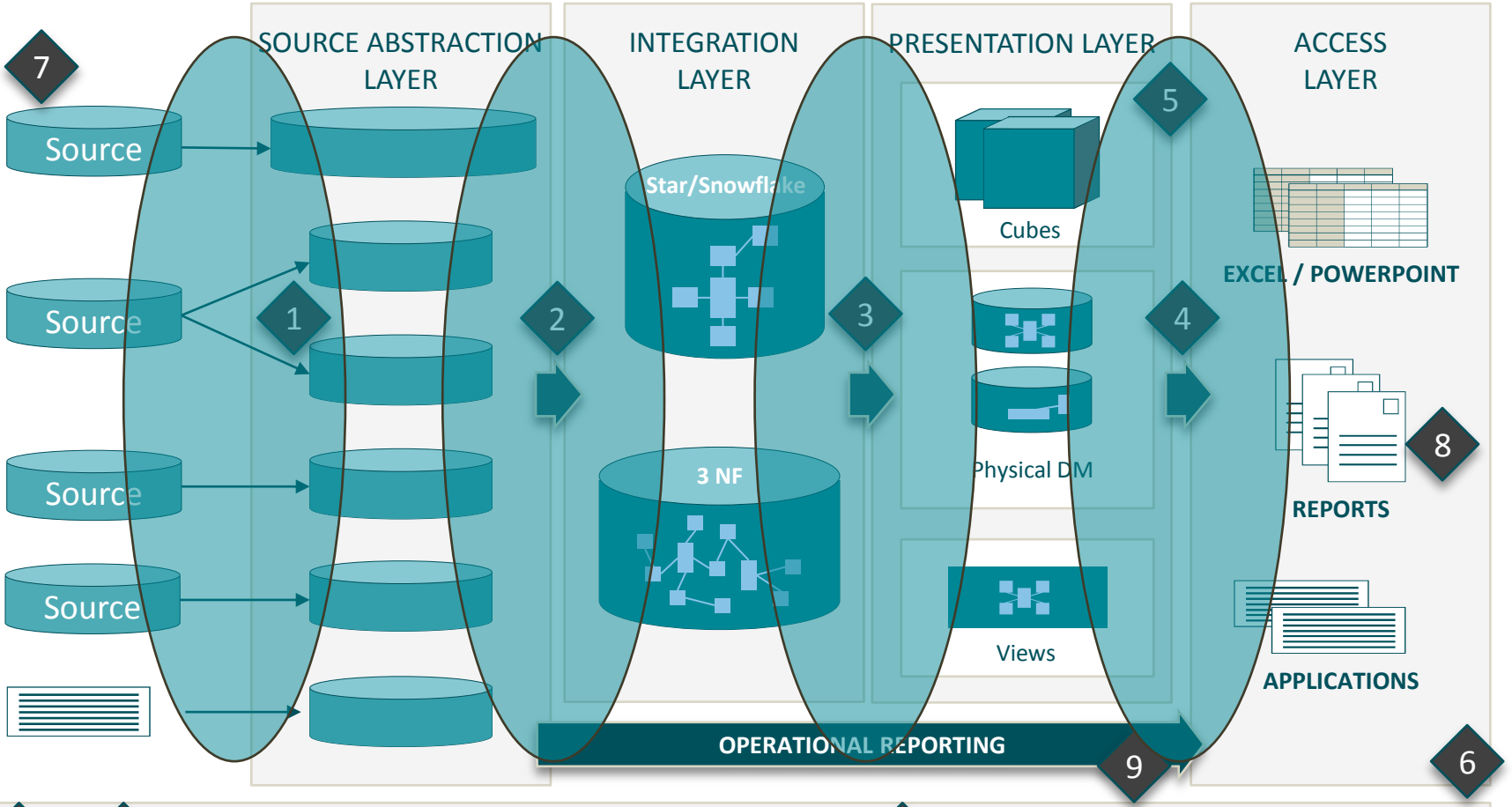
EINSATZ VON 3RD-PARTY SOFTWARE

b.telligent

Agenda

- DATENFLUSS IN DWH-LANDSCHAFTEN, DWH-ARCHITEKTUR
- ETL VS. ELT IN DER DATENBEWIRTSCHAFTUNG FÜR DEN PRESENTATION LAYER
- VOR- UND NACHTEILE VON ETL UND ELT
- BEISPIELE: BI-MODELLE UND DUBLETTENPRÜFUNG IN DER DWH-LANDSCHAFT

BI-DWH Architektur



1 – 4	Data flow
5	Presentation Layer
6	BI system stack
7	Operational Systems as Sources
8	Reports
9	BI-DWH Monitoring

Standard DWH-Architektur / Software

Tools		ETL	ELT
Oracle Warehouse Builder (OWB) / PLSQL	1 2		X
Oracle Data Integrator (ODI)	1 2	X	X
Informatica	1 2	X	(X)
Talend	1 2	X	(X)
usw.			

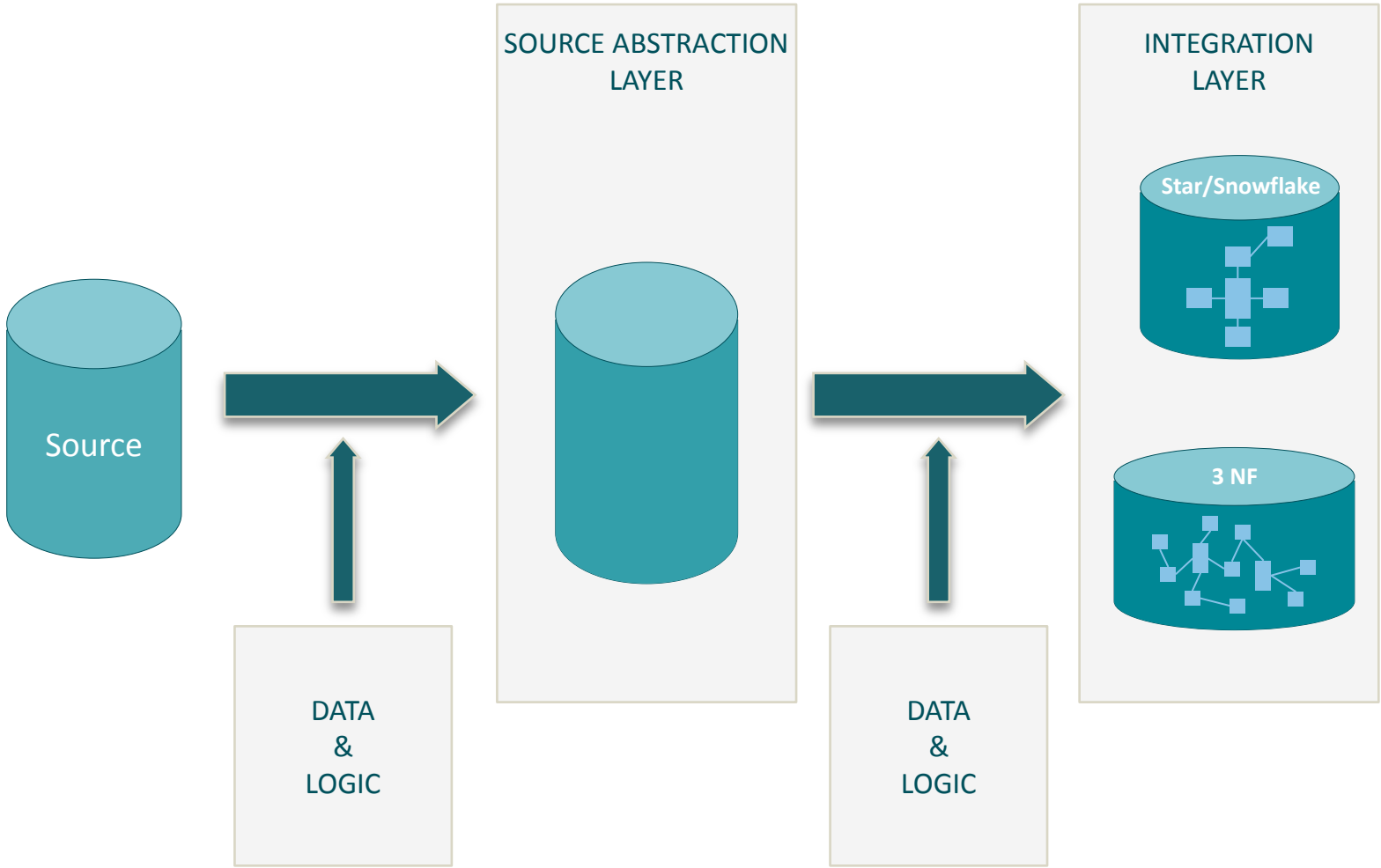
Prozesse		ETL	ELT
Data Marts / Cubes / Reporting	3	X	X
BI-Modelle wie Regression, Segmentierung, …	3	X	(X)
Adressprüfung / Cleansing	3	X	
Steuerberechnung	3	X	
usw.			

Agenda

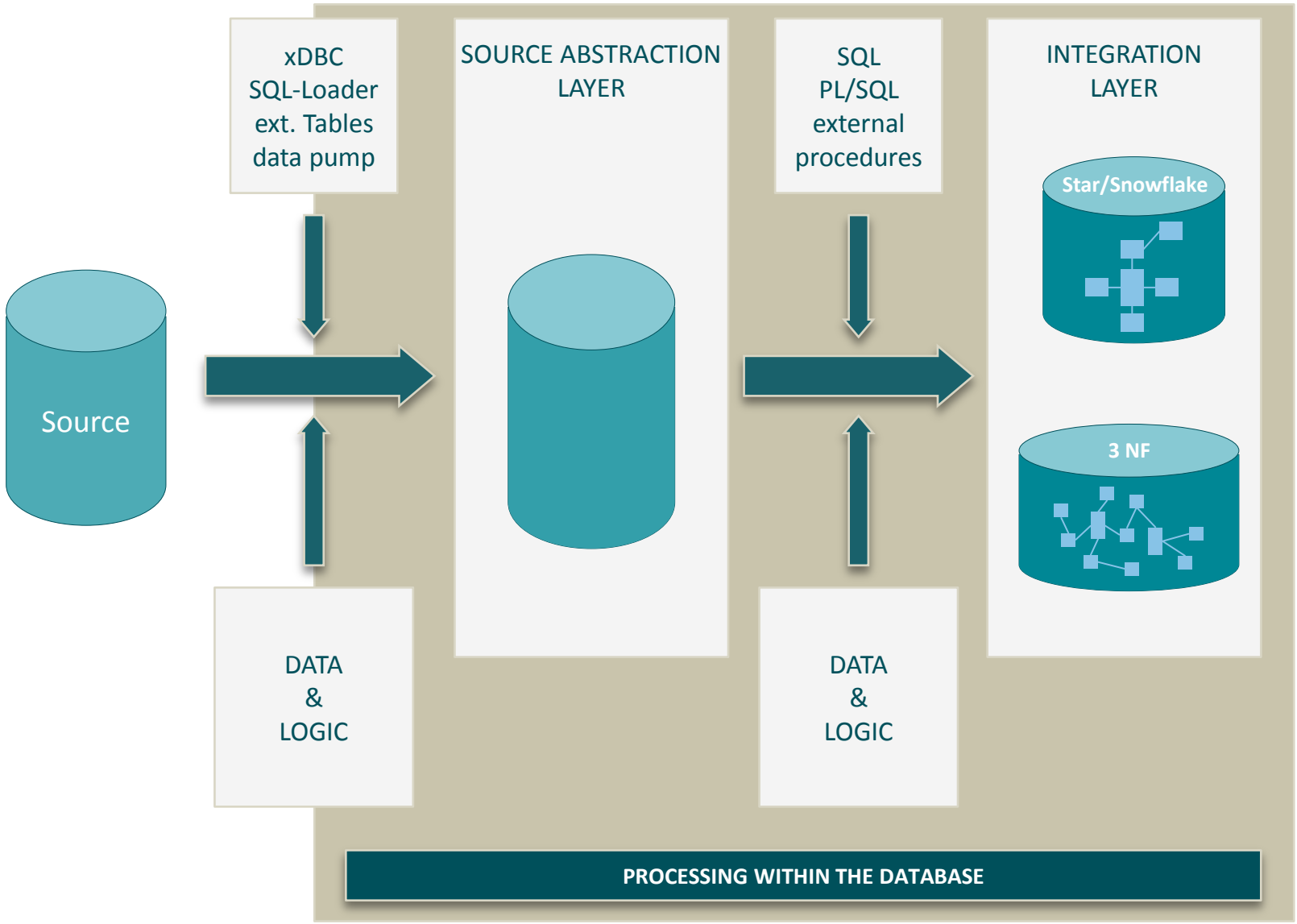
- DATENFLUSS IN DWH-LANDSCHAFTEN, DWH-ARCHITEKTUR
- **ETL VS. ELT IN DER DATENBEWIRTSCHAFTUNG FÜR DEN PRESENTATION LAYER**
- VOR- UND NACHTEILE VON ETL UND ELT
- BEISPIELE: BI-MODELLE UND DUBLETTENPRÜFUNG IN DER DWH-LANDSCHAFT

Literatur:
http://it.toolbox.com/blogs/infosphere/so-what-is-better-etl-or-elt-13572

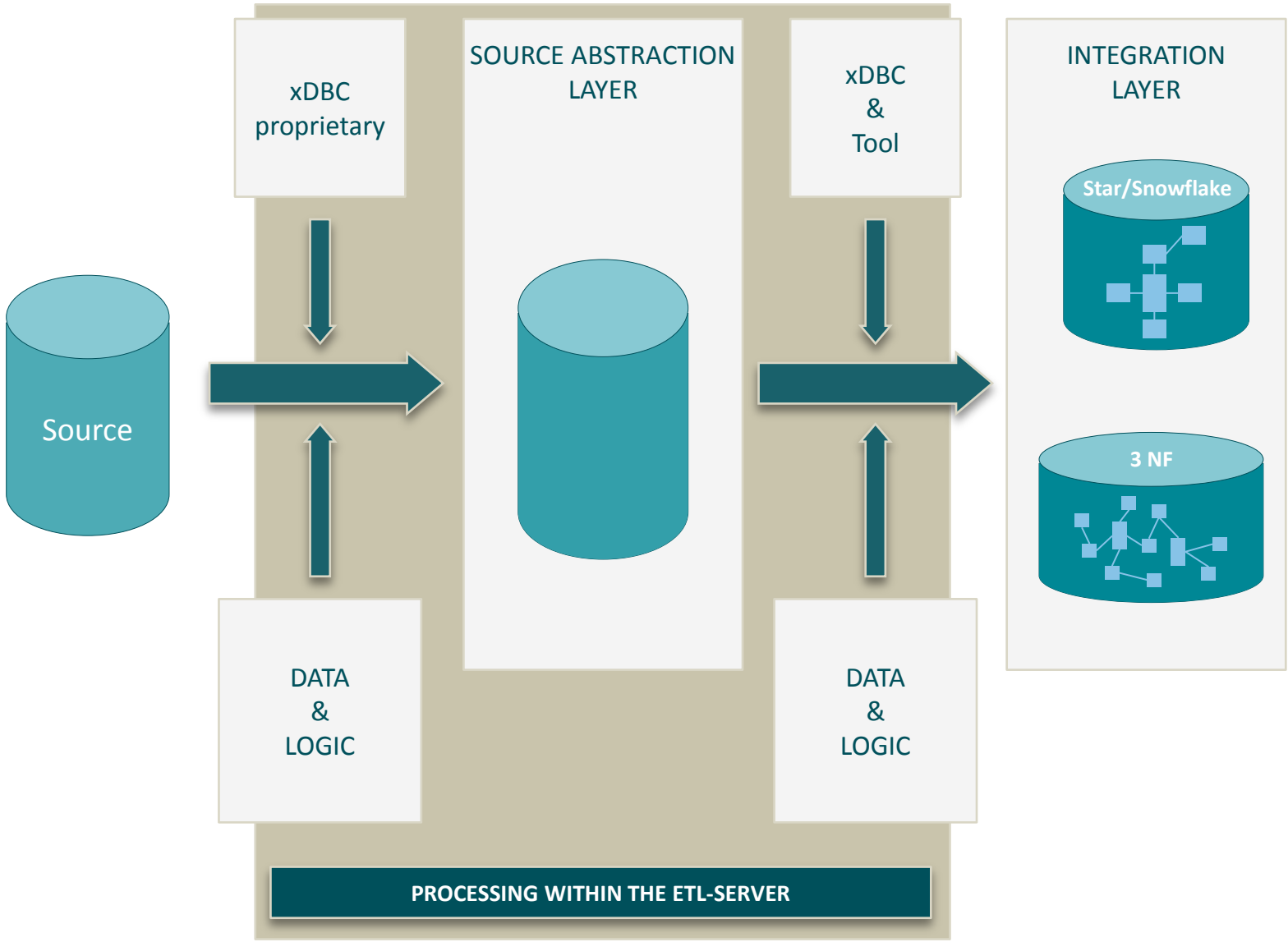
ETL vs. ELT


SOURCE ABSTRACTION LAYER
INTEGRATION LAYER
Star/Snowflake
3 NF
Source
DATA & LOGIC
DATA & LOGIC

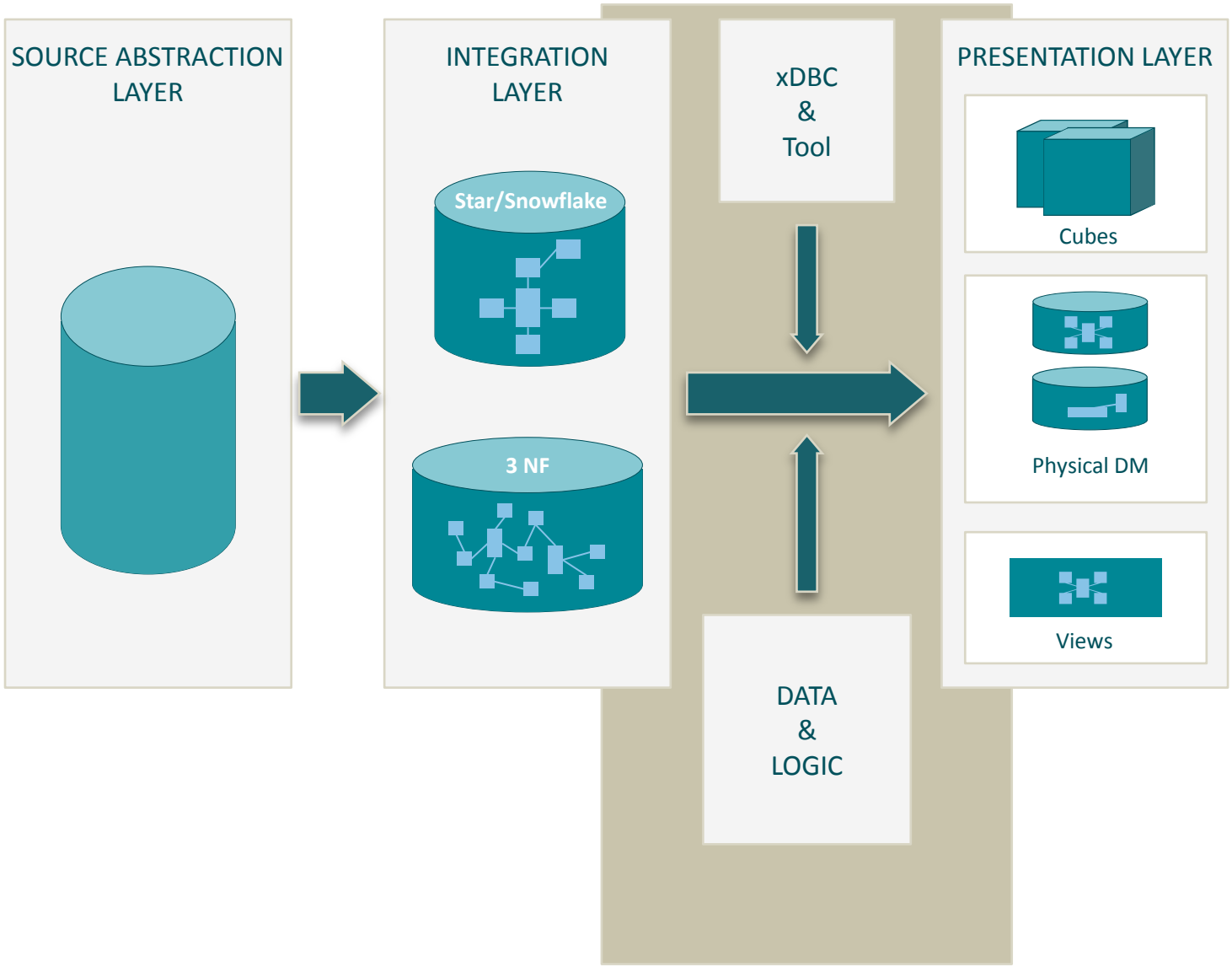
ExtractionLoadTransformation (ELT)



ExtractionTransformationLoad (ETL)

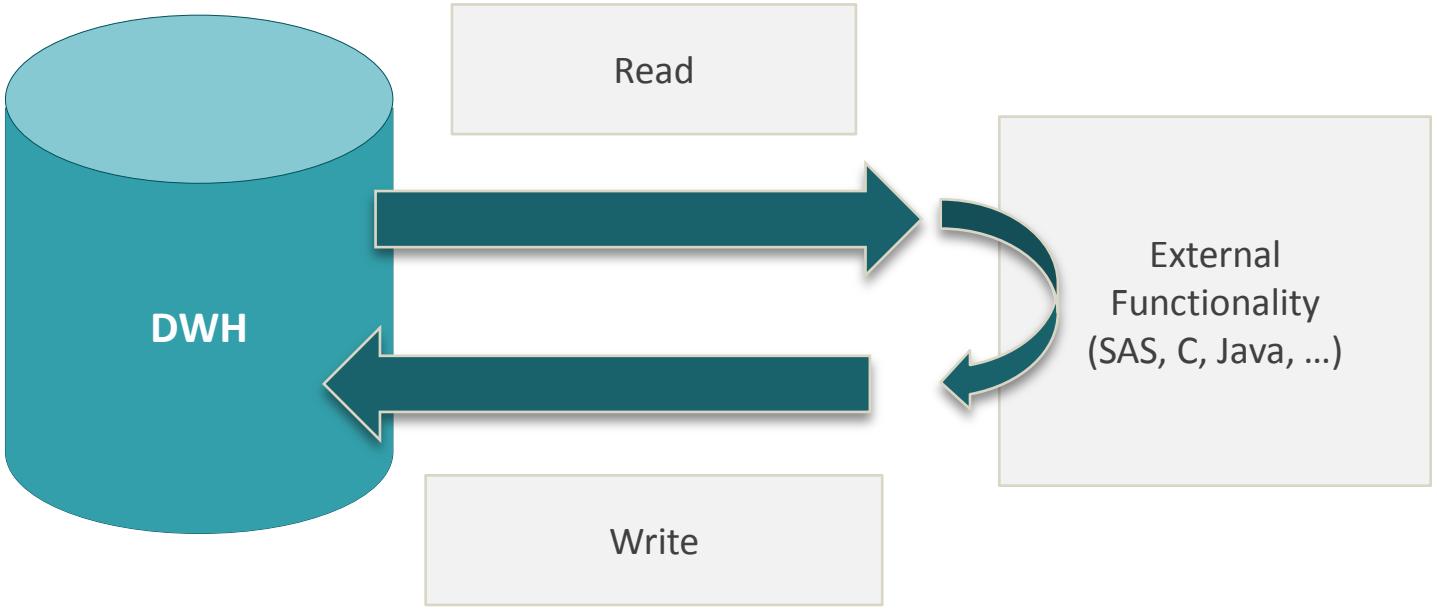


ETL vs. ELT for presentation layer



b.telligent

Externe Datenverarbeitung (ETL)



Externe Datenverarbeitung (ETL)

Externes System (3rd party) startet Datenfluss außerhalb der DB

- Verbindung des Systems/Programms zur Datenbank
- Öffnen eines Cursors zum Lesen der Daten im 3rd party Umfeld
- Datenverarbeitung im Kontext des externen Systems / Programms
- Zurückschreiben in die Zieltabelle (zeilenweise oder bulk write, falls möglich und unterstützt)

Agenda

- DATENFLUSS IN DWH-LANDSCHAFTEN, DWH-ARCHITEKTUR
- ETL VS. ELT IN DER DATENBEWIRTSCHAFTUNG FÜR DEN PRESENTATION LAYER
- **VOR- UND NACHTEILE VON ETL UND ELT**
- BEISPIELE: BI-MODELLE UND DUBLETTENPRÜFUNG IN DER DWH-LANDSCHAFT

Extraction**T**ransformation**L**oad (ETL)

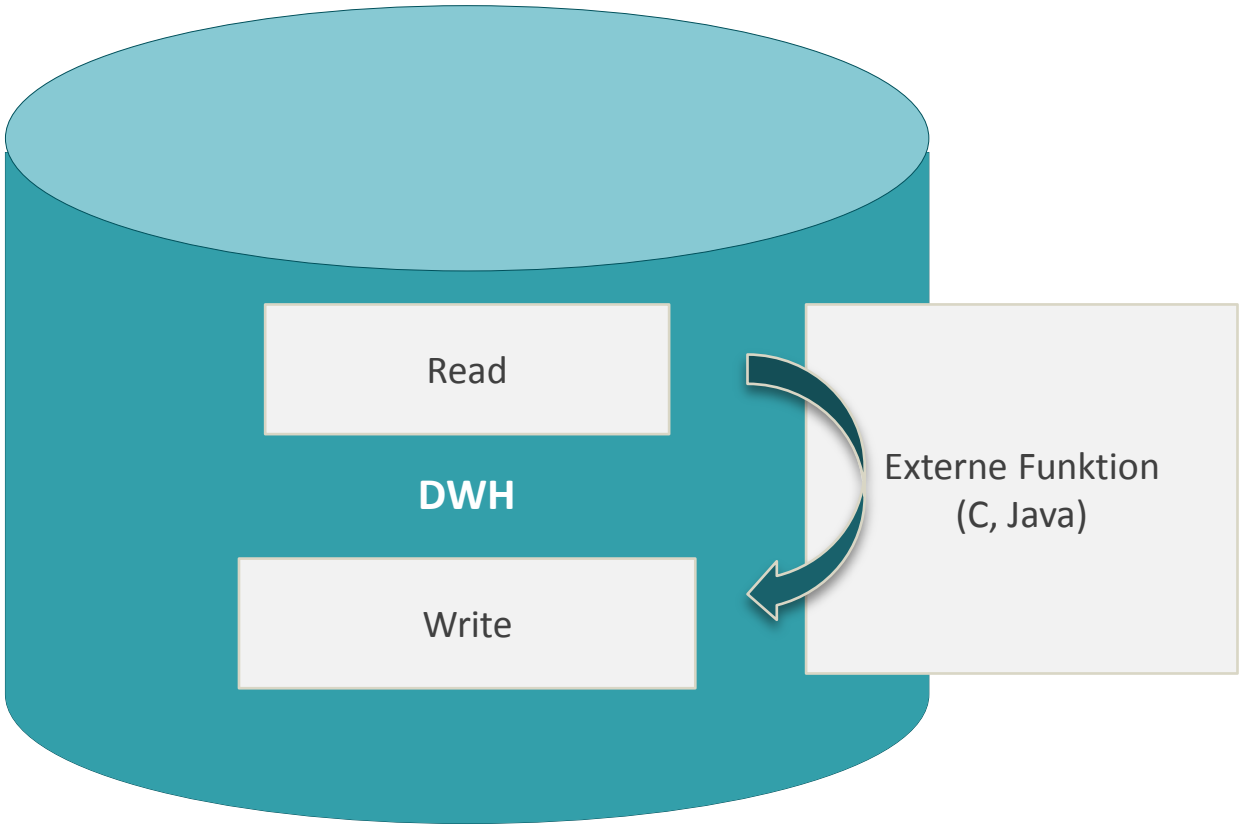
VORTEILE

- Unabhängig von Datenbank (nur definierte Schnittstellen wie z. B. ODBC, JDBC Connection notwendig), wenn kein embedded SQL
- Stärken der Programmiersprache im externen Tool voll ausnutzbar
- kein tiefes Datenbank know how notwendig
- Durch Systemtrennung Entlastung von DB-Server möglich
- Redundanz

NACHTEILE

- Verbindung zur DB (User / Passwort)
- größere Datenmengen müssen aus der Datenbank extrahiert werden, extern verarbeitet werden und ggf. wieder geschrieben werden
- i.d.R. sequentielle Verarbeitung der Daten
- ggf. embedded SQL notwendig (proprietär, wartungsunfreundlich, fehlende Versionssicherheit)
- hohe Komplexität und ggf. Kosten aufgrund Einsatz zweier Systeme (DB + 3rd party)
- Performanceprobleme, falls
  - 3rd party System und/oder DB nicht parallelisieren können
  - Datenmenge den verfügbaren Hauptspeicher übersteigt

Interne Datenverarbeitung (ELT)



Interne Datenverarbeitung (ELT)

Die Verarbeitungslogik wird in der Datenbank aufrufbar zur Verfügung gestellt:

- SQL (z. B. select / insert) wird gestartet
- Cursors zum Lesen der Daten wird durch DB-Server geöffnet (intern)
- Datenverarbeitung im Kontext des SQL-Servers
- Berechnung wird durchgeführt
- Zurückschreiben in die Zieltabelle durch DB-Server

ExtractionLoadTransformation (ELT)

VORTEILE

- Daten verlassen die DB nicht, d.h. die Verarbeitung der großen Datenmengen (Cursor) findet in der Datenbank statt
  - Speicherverwaltung der DB wird benutzt
  - Parallelisierungsoptionen der DB wird benutzt
  - Transaktionssystem der DB wird benutzt
- Reine Rechenoperationen können in (externen) Funktionen / Prozeduren sehr schnell und effizient ausgeführt werden
- Ideal für kontext-/nebenläufigfreie Rechenoperationen auf Spalten einer einzelnen Zeile (z. B. Scoring, decision tree, regression, …)
- Kein zweites Systemumfeld (Komplexitätsreduktion, Kosten)

NACHTEILE

- Logik muss in SQL / PLSQL codiert werden
- DB muss externe Funktionen unterstützen
- Bei externen Funktionen i. d. R. nur Operationen innerhalb einer Zeile möglich; keine zeilenübergreifenden Operationen
- DB-Server wird belastet (CPU-Last)
- keine Redundanz, da nur 1 System

Agenda

- DATENFLUSS IN DWH-LANDSCHAFTEN, DWH-ARCHITEKTUR
- ETL VS. ELT IN DER DATENBEWIRTSCHAFTUNG FÜR DEN PRESENTATION LAYER
- VOR- UND NACHTEILE VON ETL UND ELT
- **BEISPIELE: BI-MODELLE UND DUBLETTENPRÜFUNG IN DER DWH-LANDSCHAFT**

Datenfluss in DWH-Landschaften

Datenfluss planen:

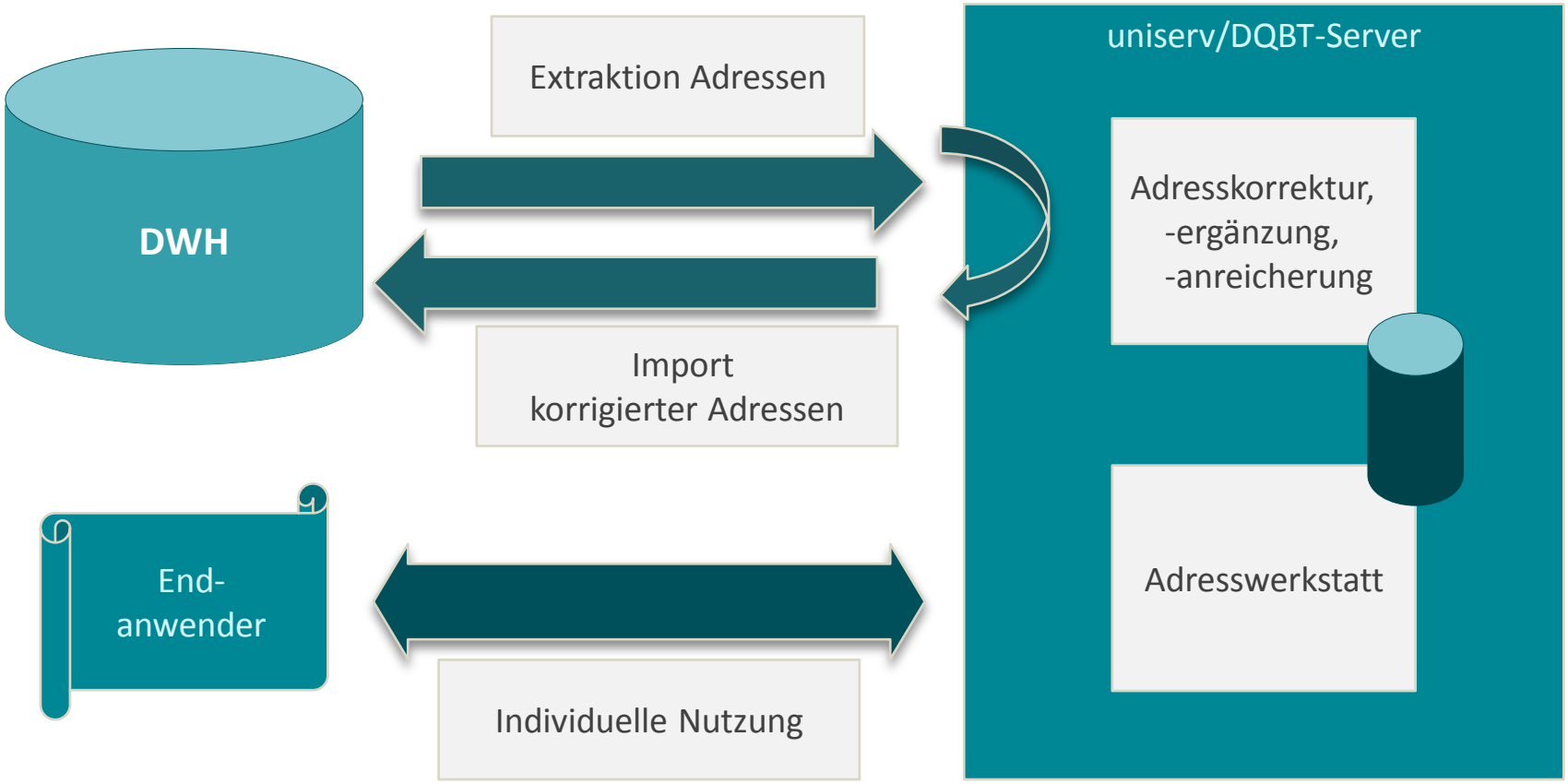
- Welche Datenmengen werden bewegt?
- Wie häufig ist eine Logikänderung notwendig? Sind die Anforderungen zeitkritisch?
- Möchte der Fachbereich selbst Parametrisierungen / Logikänderungen durchführen?
- Unterstützt das externe Tool ELT-Prozesse (Logiktransport zum DB-Server)
- Ist Redundanz gewünscht oder soll sie gerade vermieden werden?
- Wie hoch sind die Maintenance-Kosten und wie viel Aufwand muss davon die interne IT erbringen?
- …

Prozessindividuell Lösungen sind gut und notwendig,

aber nicht alles was machbar ist muss auch gemacht werden.

Praxisbeispiel 1: ETL-Prozess

Adress- und Dublettenprüfung mit uniserv data qualtiy batch suite



Praxisbeispiel 2: BI-Modelle

Erstellung BI-Modelle mit SAS Enterprise Miner (EM)

Produktivsetzung

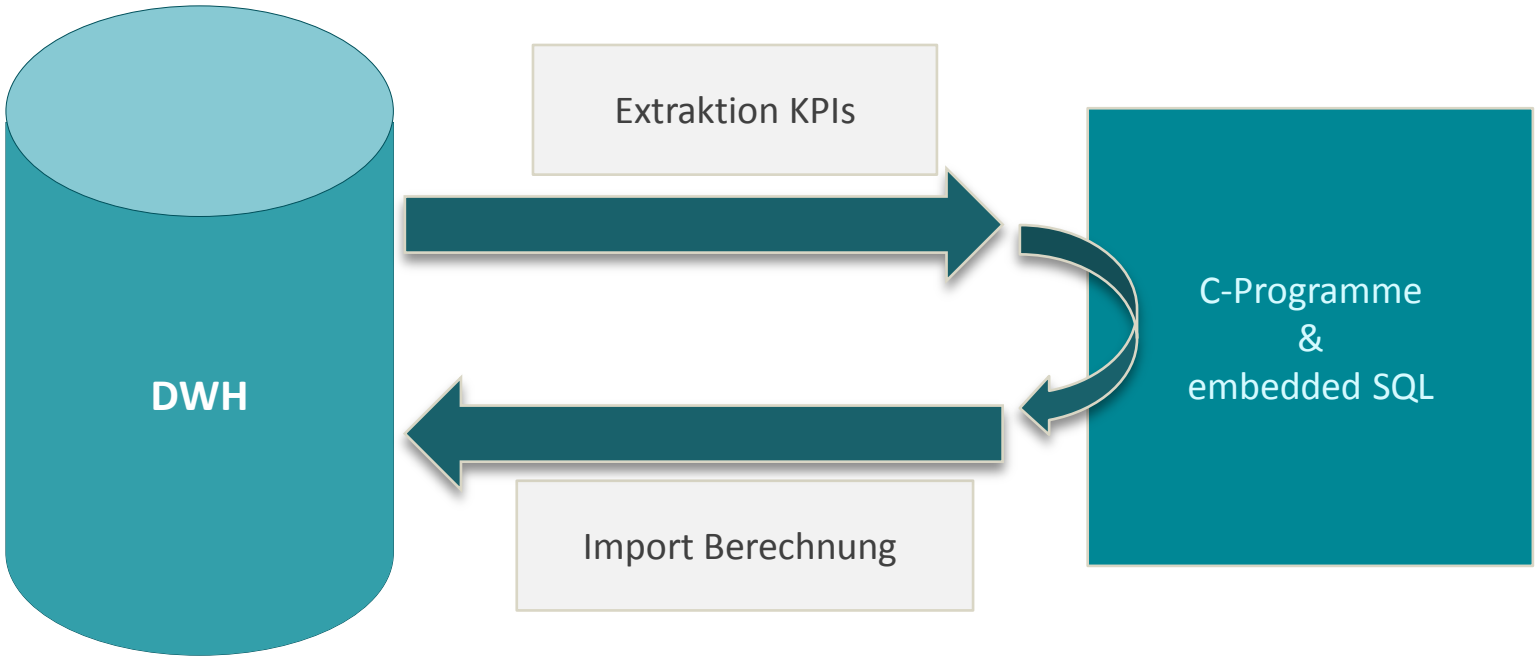
Export Modell durch SAS-EM

- C-Programm
- XML-Metadaten

Build-Prozess: Anbindung DB

- C-Rahmenprogramm (embedded SQL)
- Tabellen (Source/Target)
- Null-Value handling
- Datentypkonvertierungen

Praxisbeispiel 2: BI-Modelle (ETL-Variante)



Praxisbeispiel 2: Vom ETL zum ELT

Erstellung BI-Modelle mit SAS Enterprise Miner (EM)

Produktivsetzung

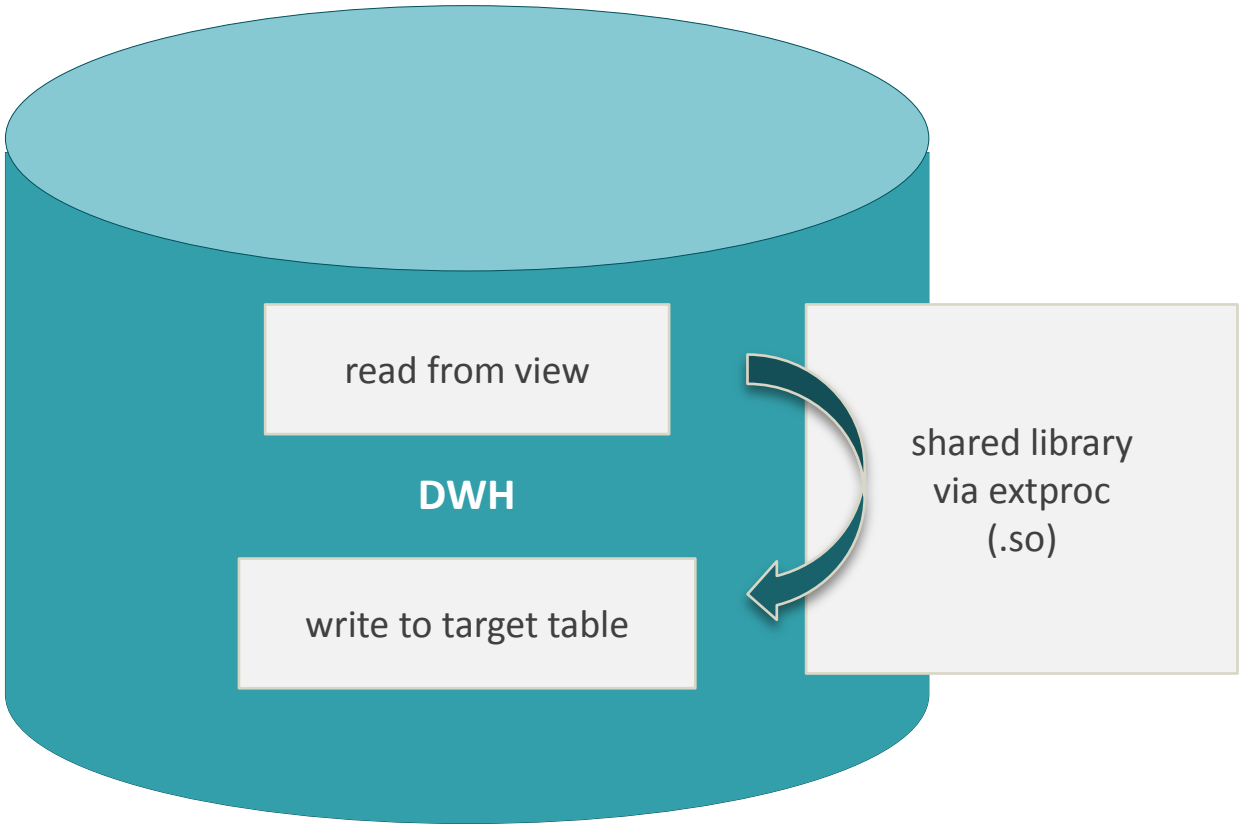
Export Modell durch SAS-EM

- C-Programm
- XML-Metadaten

Build-Prozess: Shared Library

- C-Rahmenprogramm (für PL/SQL –Aufruf)
- PL/SQL-Wrapper
- View

Praxisbeispiel 2: Vom ETL zum ELT



Fazit

ETL vs. ELT wird nach wie vor intensiv diskutiert.

Es gibt Gründe sowohl ETL- als auch ELT-Techniken einzusetzen. Eine generelle Entscheidung ist nicht sinnvoll.

Bei der Datenbewirtschaftung der Applikationen (presentation layer) wird der Datenfluss oft nicht betrachtet. Performance- und Prozessprobleme sind oft die Folge.

Die geplante Entscheidung für ETL- oder ELT-Datenbewirtschaftung bei der Versorgung der Applikationen in der Design-Phase vermeidet Folgeprobleme. Kosten (z. B. Lizenzgebühren) lassen einsparen.

Aus dem Blickwinkel der Datenbewirtschaftung in der DWH-Landschaft eröffnen sich in dem Prozessschritt der Datenversorgung des presentation layers oft unentdeckte Möglichkeiten an Optimierungspotential.

MÜNCHEN **HAMBURG** **DÜSSELDORF** **ZÜRICH**

b.telligent GmbH & Co. KG
Georg-Brauchle-Ring 54
80992 München

b.telligent

Code-Beispiel: C-Rahmenprogramm

```
/*#include <oci.h>*/
#define OCIInd signed short
#define OCI_IND_NOTNULL (OCIInd)0                                   /* not NULL */
#define OCI_IND_NULL (OCIInd)(-1)                                   /* NULL */
//#define OCI_IND_BADNULL (OCIInd)(-2)                              /* BAD NULL */
//#define OCI_IND_NOTNULLABLE (OCIInd)(-3)                          /* not NULLable */

int c_function_in_shared_library(
/* in */
   double V1
  ,OCIInd _V1
  ,char*  V2
  ,OCIInd _V2
  ,…
/* out */
  ,char* R1
  ,double* R2
  ,…
)
{
  …
  return(0);
}
```

Code-Beispiel: PLSQL-Wrapper

```
FUNCTION function_to_call(
/* in */
     V1   in BINARY_DOUBLE
    ,V2   in VARCHAR2
    ,…
/* out */
    ,R1   out VARCHAR2
    ,R2   out BINARY_DOUBLE
    ,…
)
  RETURN BINARY_INTEGER
  AS LANGUAGE C
  LIBRARY C_SHARED_LIB NAME "c_function_in_shared_library"
  PARAMETERS (
     V1
    ,V1 INDICATOR
    ,V2
    ,V2 INDICATOR
    ,…
    ,R1
    ,R2
    ,…
    ,RETURN);
```