Master Data

die logistische Koordination
der betrieblichen Datenbestände

Dr. Walter Kellerer, austriamicrosystems AG

Dipl.-Ing. Edmund-Gerhard Schrümpf, EDconsult

2002-10-10

austriamicrosystems

Inhalt

- Einleitung & Problemstellung
- “Datenlogistik” in Multisystemumgebungen
- Verfügbare Standardsoftware
- Beschreibung technischer Probleme und deren Lösung

austriamicrosystems AG

Design und Produktion von “mixed signal” ICs (ASICs, ASSPs)

3 Marktsegmente: Automotive, Communications, Industry

820 Mitarbeiter, 14 Standorte, Zentrale: Unterpremstätten/Graz

austriamicrosystems

austriamicrosystems AG - Firmenprofil

- 1981 Gründung austriamicrosystems (AMS) durch American Microsystems Inc. (AMI)
- 1993 AMS Börsengang (Wien)
- 2000 Spatenstich neue Fabrik (Graz)
- 2000 Übernahme von AMS durch internationale Investorengruppe (Schroders Investments aka PERMIRA)
- 2002 Produktionsstart in neuer Fabrik in Graz
- Multinationale Präsenz - 820 Beschäftigte, 14 Niederlassungen weltweit

austriamicrosystems - Marktsegmente

COMMUNICATION

Wireline Communications
Mobile Phones
RF Transceivers
Fiber Optronics

AUTOMOTIVE

Access Control&Mobilizing
Sensor&Micro Systems
Motor&Power Management
Bus Systems

INDUSTRY & MEDICAL

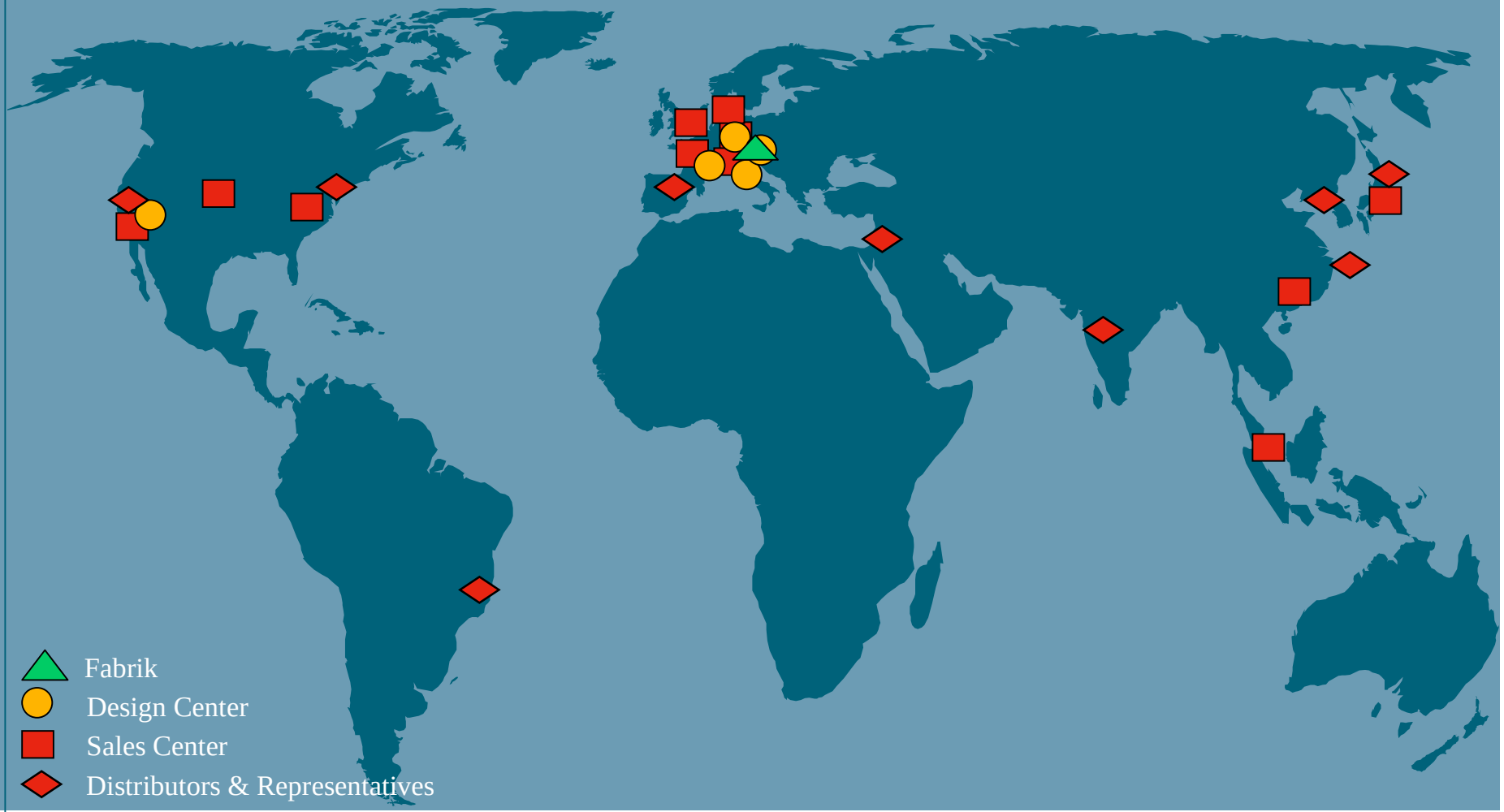
Solid State Metering
Sensors
Healthcare

FULL SERVICE FOUNDRY

Design Support
Process Characterization
Wafer Production
Backend Services

austriamicrosystems

austriamicrosystems - Standorte



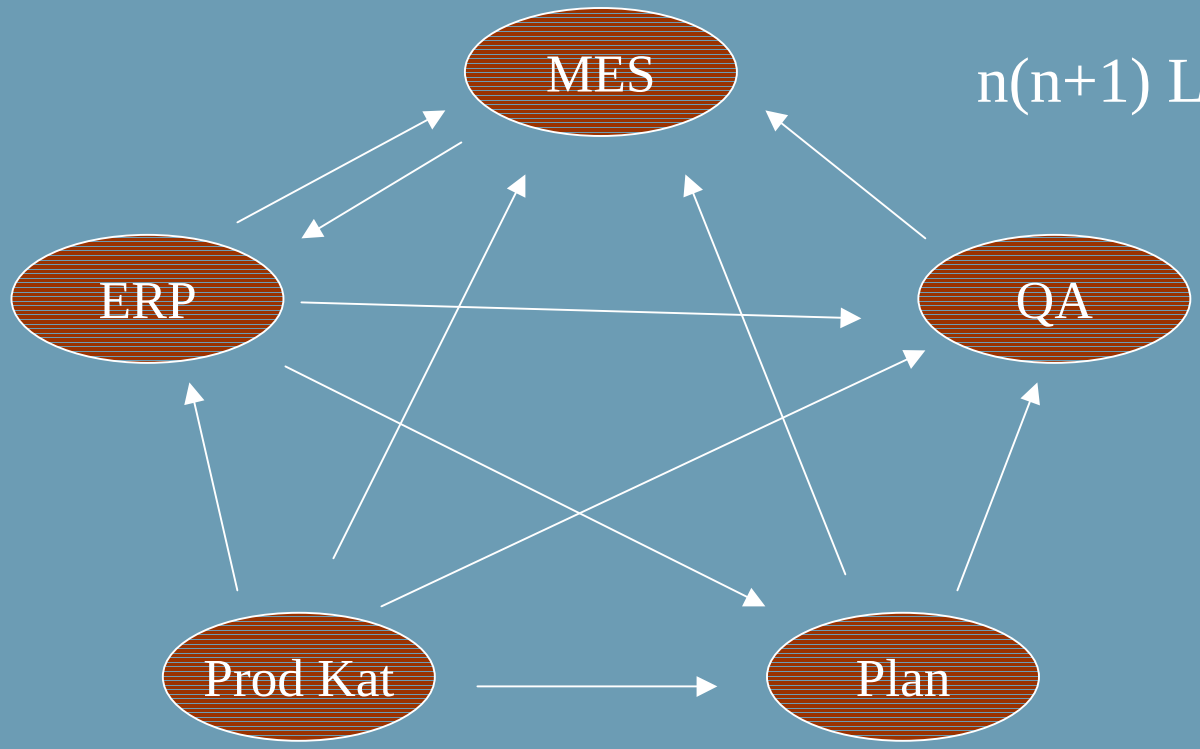
austriamicrosystems

Einleitung & Problemstellung

Enterprise Application Integration (EAI)

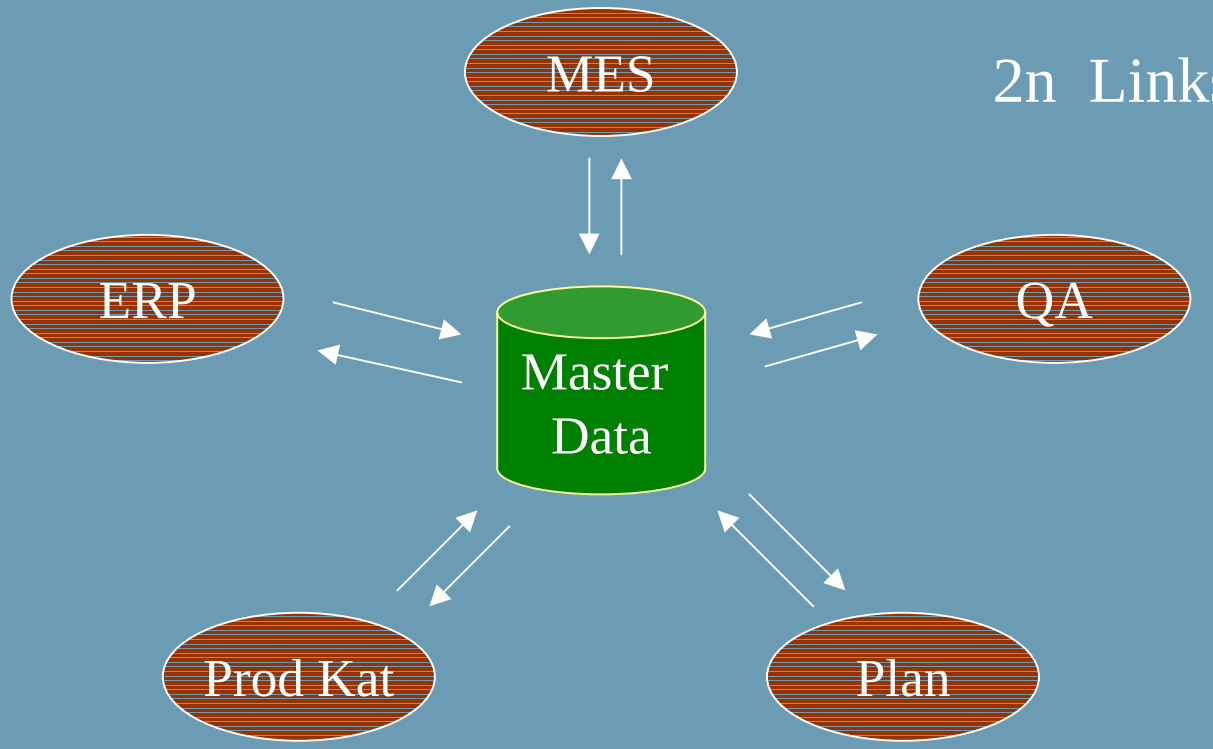
- Verbindung operationeller SW-Systeme
- Robustheit (Ausfallsicherheit)
- Skalierbarkeit
- Datentransformation
- Messaging

“Datenlogistik” in Multisystemumgebungen: Situation



n(n+1) Links

“Datenlogistik” in Multisystemumgebungen: Forderung



Verfügbare Standardsoftware (1)

The EAI Symposium & Expo
is a virtual exhibition for EAI and Web services
http://www.eaixpo.com/

COGNOS (read only)

TIBCO

Product TIBCO ActiveEnterprise
http://www.tibco.com/solutions/technology_solutions/eai/default.jsp?m=c15

Verfügbare Standardsoftware (2)

Vitria BusinessWare for EAI http://www.vitria.de

- **Metadata-driven design** dictates that all events are represented in metadata, so you can customize or change the underlying databases without programming
- **Create and maintain** graphical connection models without programming
- **Save and reuse** connection models
- **Decouple** business processes from method of application data exchange
- **Gain** high-level visibility and control through application connection management, allowing timely identification of problems
- **Mix-and-match** connection and transformation components to accommodate a wide range of business systems

Warum Projekt Master Data?

- ORACLE Snapshots nicht ausreichend, Snapshot-Manuals stellen Probleme bestens dar!
- Kritisches Standard-Produktionssystem (7*24) anzubinden, keine Unterstützung durch SWH
- Mehrere SW-Eigenentwicklungen anzubinden
- ORACLE MIX 7.3.4, 8.1.7 und 9i
- Vollständiges Änderungsjournal erforderlich
- EAI-SW damals noch unbekannt

Problem der Änderungshoheit

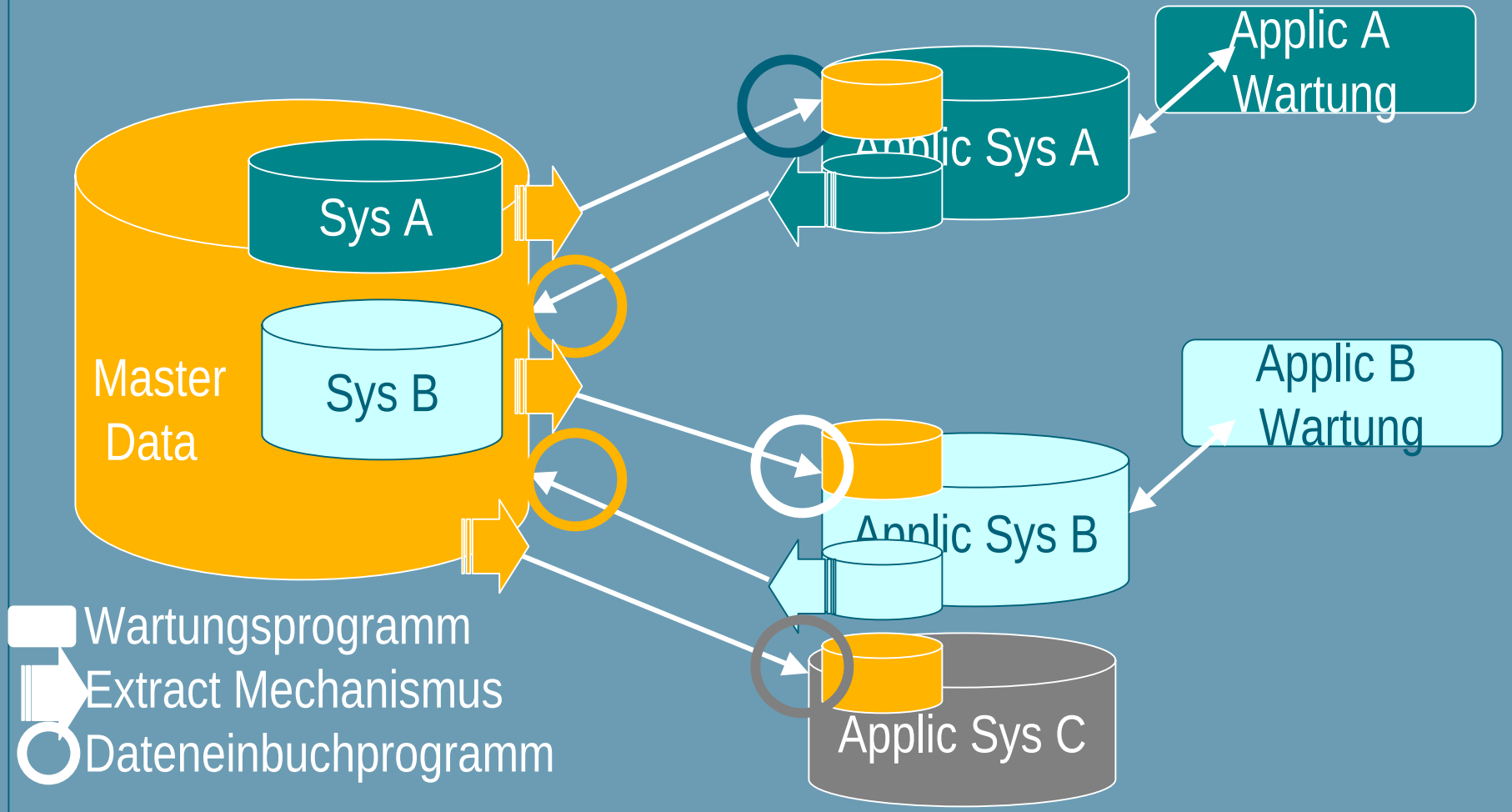
- Datenänderungen nur in einer DB
  oder in wechselnden DB
- Änderungshoheit auf Ebene Datenfeld
  oder Datensatz
- bei mehrseitiger Änderungshoheit muß der Konflikt gleichzeitiger Änderung geregelt sein durch Priorität oder manuellen Eingriff

➔ Welcher Applikation gehören die Stammdaten?

Kritische Punkte in der DFÜ-Logik

- die Versionierung (Protokollierung aller Datenänderungen)
- das Datenextraktionsprogramm
- die Bildung von konsistenten Datasets
- das Dateneinbuchprogramm

Master Data - Gesamtübersicht



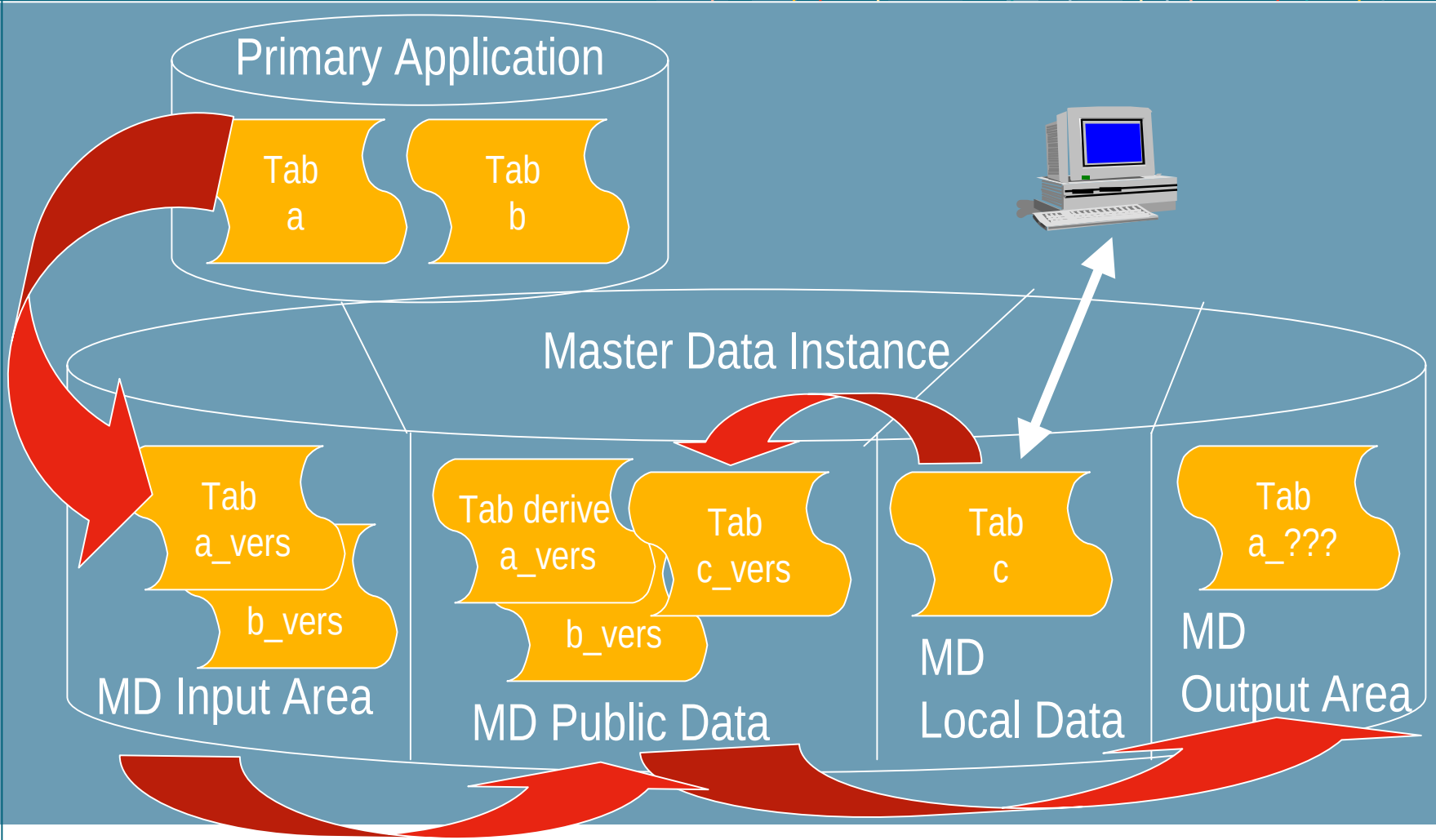
Versionierung: Aufgabenstellung

In einem Datenbestand werden im Laufe der Zeit Veränderungen vorgenommen.

Es sind

- alle Versionen der Datenbestände zu protokollieren
- für einen Zeitpunkt zusammenpassende (“konsistente”) Daten zu erzeugen.

Versionierung: Übersicht



Versionierung: Datenextraktion

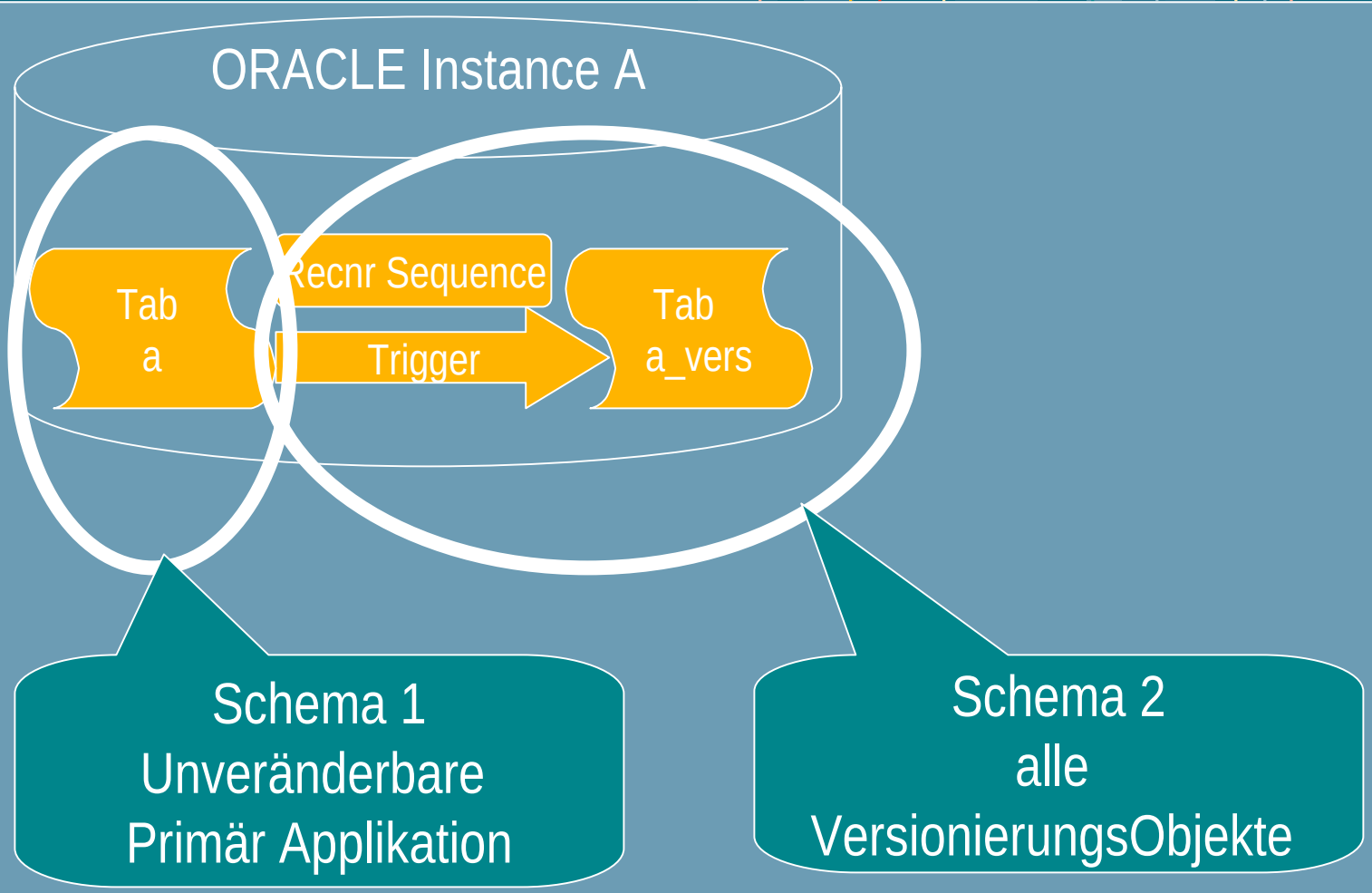
Ausgangsbasis:

Zukauf-Applikationen, daher keine Änderungen an der Datenstruktur möglich

Lösung

- zusätzliche Datenbank-Trigger, die Datenveränderungen in zusätzliche Tabellen (Journaltables) wegschreiben.
- Die Trigger müssen positiv enden
- keine „riskanten“ Remote-Zugriffe
- Trigger sollen die geringsmögliche Arbeit erledigen,
- Trigger dürfen Laufzeit der Transaktion kaum beeinflussen

Versionierung: Schema-Konzept



Aufzeichnungsgenauigkeit bei Datenveränderung A

A) Verfahren mit vollständiger Aufzeichnung aller Veränderungen:

- Während der Transaktion werden vollständige Datensätze mit allen Felder und Verwaltungsoverhead weggeschrieben.
- Der Extractionsmechanismus braucht dann nur mehr die betreffenden Datensätze auszuwählen und zu versenden.

Vorteil: Vollständiger Aufzeichnung aller Zustände des Datensatzes

Nachteil: Risiko durch Operationsvolumen in der Transaktion

Aufzeichnungsgenauigkeit bei Datenveränderung B

B) Zeitpunktorientiertes Verfahren:

- Es werden nur Schlüsselbegriffe (oder Rowids) während der Transaktion weggeschrieben.
- Zu einem bestimmten Zeitpunkt holt der Extractionsmechanismus die Daten zum Schlüsselbegriff und versendet diese Daten.

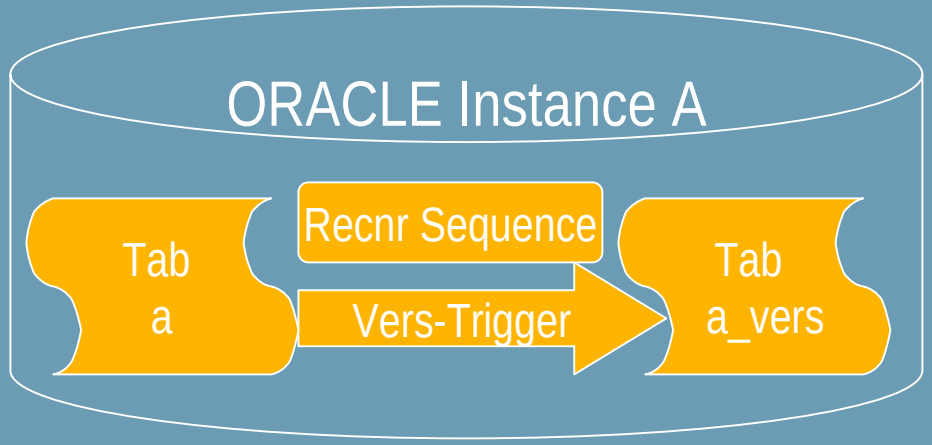
Dieses Verfahren wird zB von ORACLE bei Snapshots

Vorteil: Schnell, kaum Last in der Transaktion

Nachteil: Verlorene, nicht protokollierte Zustände des Datensatzes

Empfehlung: Es bietet sich an, das vollständige Verfahren A für Stammdaten (wichtige Daten, aber wenig Änderungen) und das rasche Verfahren B für Bewegungsdaten mit hohem Änderungsvolumen einzusetzen.

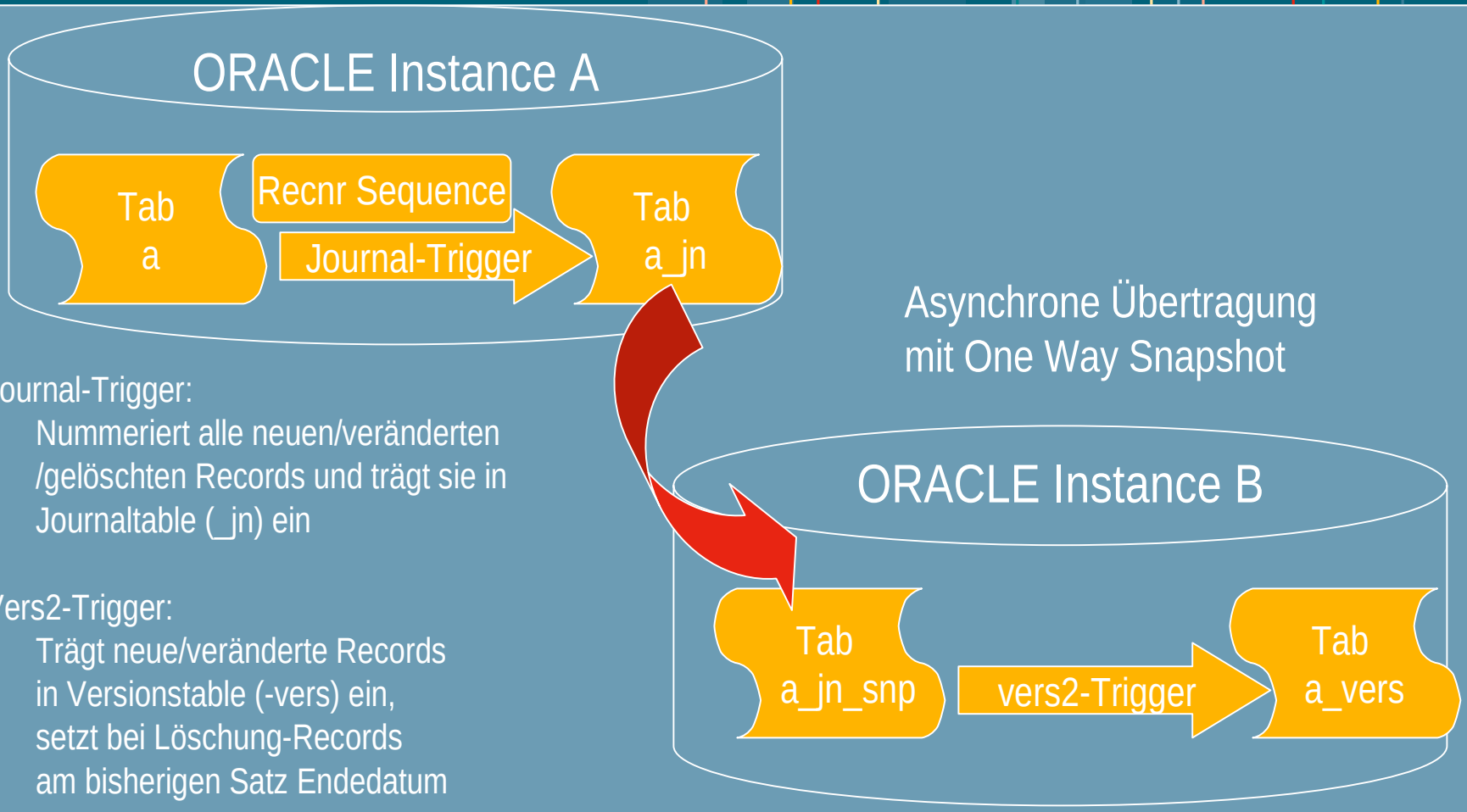
Versionierung: einstufig/synchron



Vers-Trigger:

- Nummeriert und trägt alle neue/veränderte Records in Versionstable (_vers) ein,
- setzt Endedatum am bisherigen Record bei Löschungen

Versionierung: zweistufig/asynchron(1)



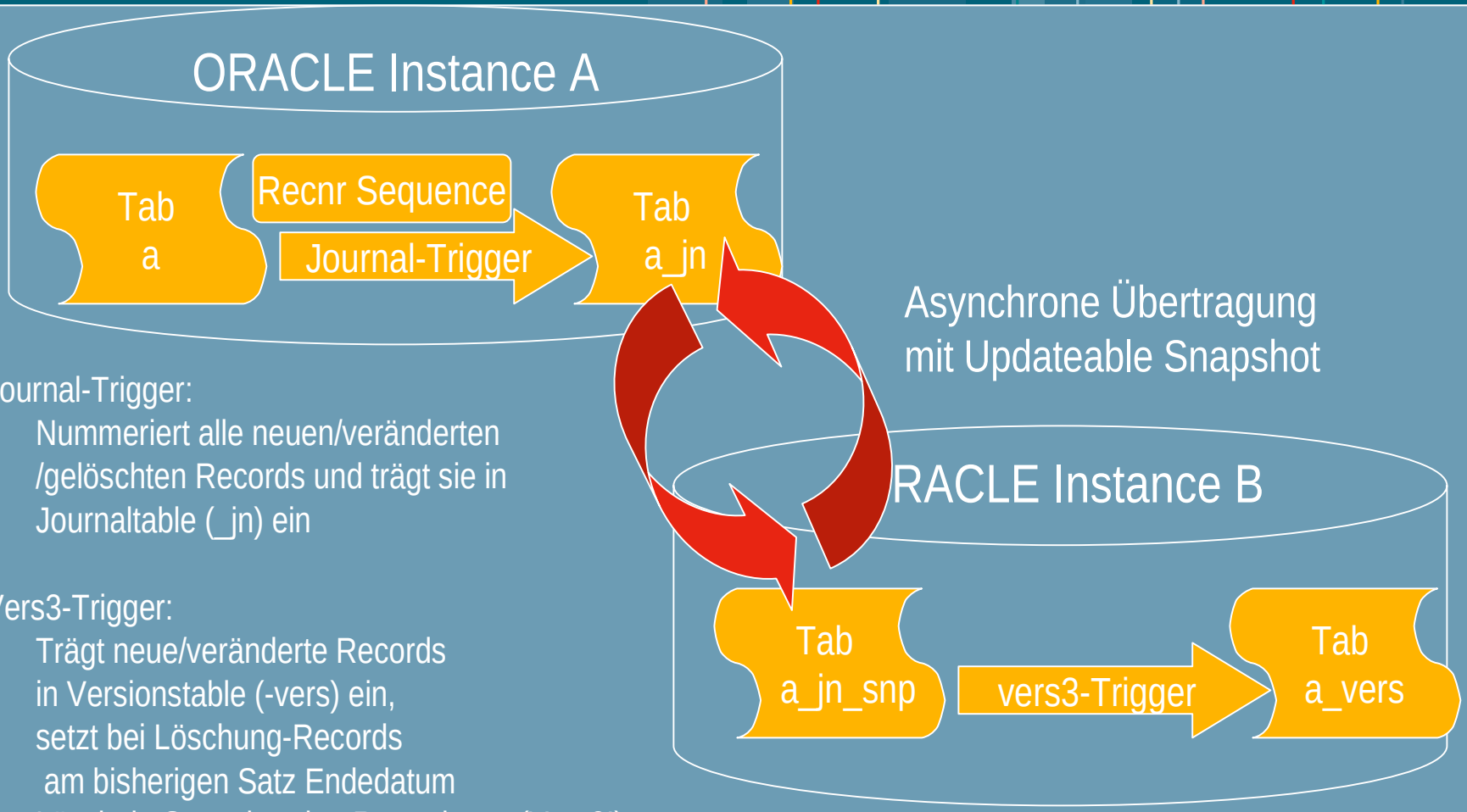
Journal-Trigger:

- Nummeriert alle neuen/veränderten /gelöschten Records und trägt sie in Journaltable (_jn) ein

Vers2-Trigger:

- Trägt neue/veränderte Records in Versionstable (-vers) ein,
- setzt bei Löschung-Records am bisherigen Satz Endedatum

Versionierung: zweistufig/asynchron(2)



Journal-Trigger:

- Nummeriert alle neuen/veränderten /gelöschten Records und trägt sie in Journaltable (_jn) ein

Vers3-Trigger:

- Trägt neue/veränderte Records in Versionstable (-vers) ein,
- setzt bei Löschung-Records am bisherigen Satz Endedatum
- Löscht in Snapshot den Record weg (Vers3!)

Zeitmessung in ORACLE

Ausgangsbasis (Oracle 7.3.4):

Datentyp DATE Genauigkeit eine Sekunde im Zeitraum
vom 1.1.4712 vor Christi bis 31.12.4712 nach Christi

mehrere Datensatzänderungen in einer Sekunde - DATE reicht nicht

Lösung/Erweiterung

Es wird daher eine Numerierung für Datenoperationen eingeführt. Die "Recordnummer" ist eine über alle beteiligten Tabellen einer Primären Applikation (Quellensystem) verwendete "ewige" Nummer in streng aufsteigender Reihenfolge. Die Vergabe erfolgt über eine "Ordered Sequence".

Hinweis: Messungen haben für ORACLE Version 8.1.7. gezeigt, dass die Klausel "ORDERED" keine zusätzliche Rechenzeit erfordert.

Lokale Zeit auf den Servern

Ausgangsbasis:

- Jeder Server hat eine lokale Systemzeit,
- Zeit muß nicht mit anderen Servern übereinstimmen
- Zeitsynchronisation des gesamten Netzes sehr sinnvoll, kann aber nicht garantiert werden.

Regeln für die Zeitverwendung:

- Zur Protokollierung/Versionierung nur die Maschinenzeit des Quellsystems verwenden
- Unter allen Umständen vermeiden, die Systemzeiten zweier Server zu vermischen oder zu vergleichen

Datasets: Supertransaktionen

Einzelne Versionen reichen oftmals nicht aus, es müssen daraus höherwertig-konsistente „Datasets“ gebildet werden. (z.B. vollständige Garnitur von Parameterdaten). Es sind Dataset-Typen prinzipiell zu unterscheiden:

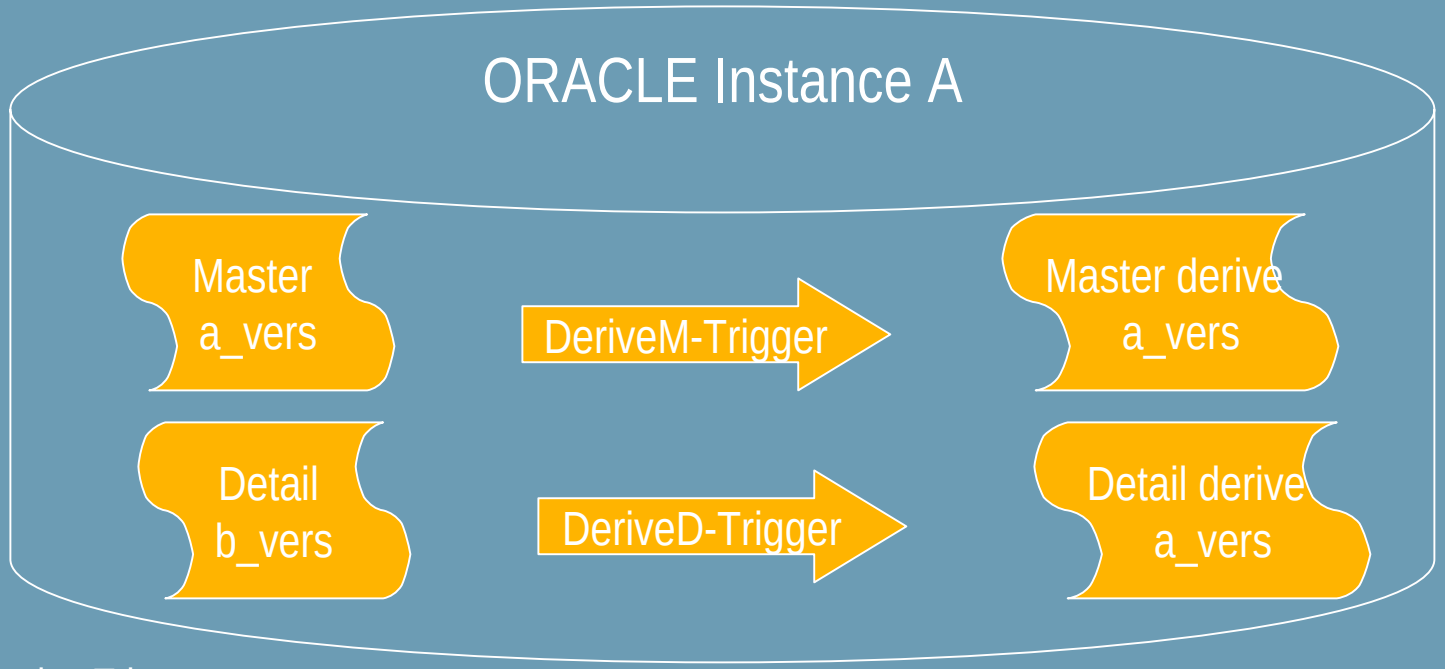
- Abgeschlossene Datasets: Keine Änderung mehr zulässig (eingefroren = frozen)
- Das aktuelle offizielle Dataset: dies ist das letzte abgeschlossene Dataset, das besonders markiert ist und aktuell verwendet wird. (Anmerkung auch hier ist keine Änderung mehr zulässig - frozen)
- Arbeits- oder Zukunfts-Dataset: Offenes, bearbeitbares Dataset, in dem alle Anpassungen vorgenommen werden können. Dieses Dataset stellt somit eine Supertransaktion über alle Arbeitstransaktionen zur Erstellung dieses Datasets dar.

Datasets: Methoden

Ausgangsbasis:

- Mit einem eigenen Mechanismus wird diese Arbeitsversion “in Betrieb genommen” oder “freigegeben” und dabei in die “Aktuelle offizielle Version” umgewandelt. Dabei werden die Daten eingefroren und eine neue Arbeitsversion erstellt.
- In der Praxis ist es fallweise erforderlich, abgeschlossene Versionen “nachzubearbeiten”. Der saubere Weg dazu ist ein “Unfreeze” der abgeschlossen Datasets vorzusehen.

Einbuch-Mechanismus: Einfaches Verfahren

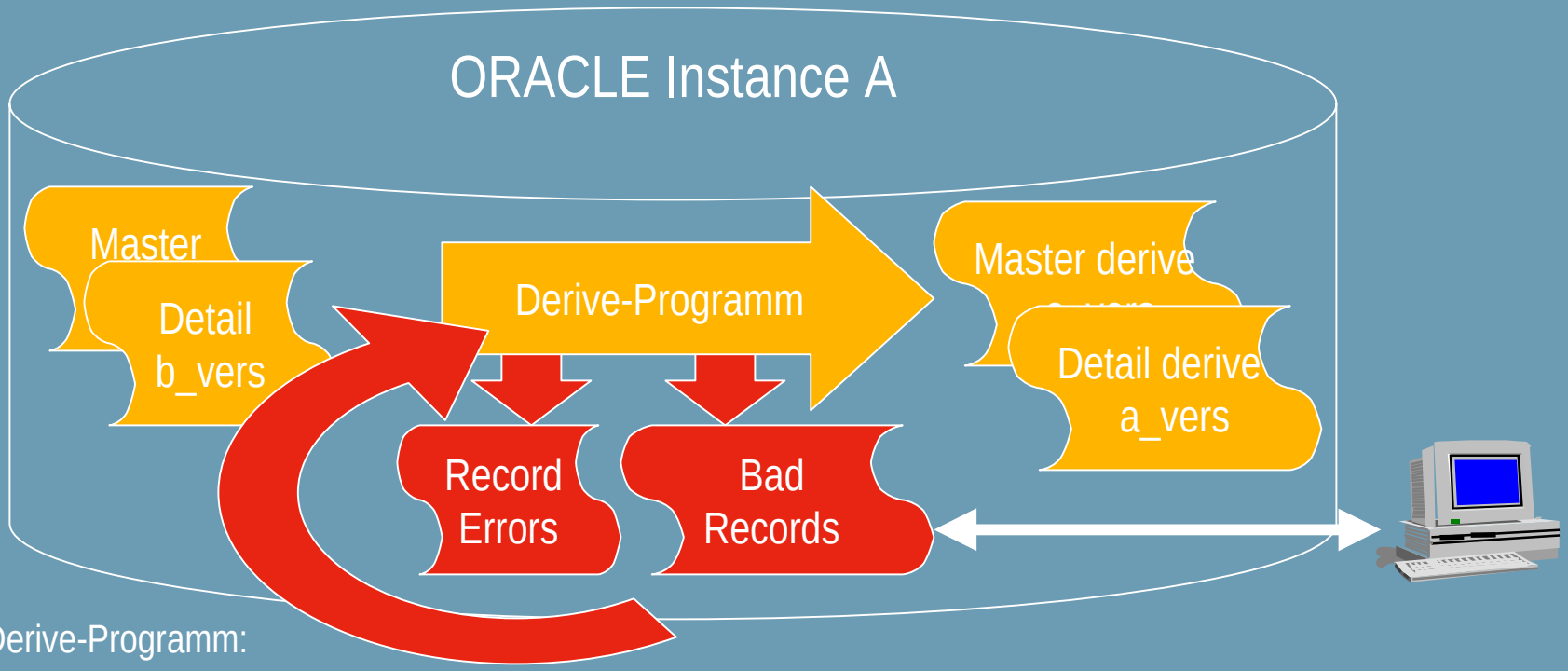


Derive-Trigger:

- Verändert Daten, passt an (kann nicht prüfen, da keine Ablehung möglich ist)
- trägt Datensatz in “derive” Tabelle ein

Keine Konsistenzprüfung möglich, alle Prüfungen abschalten!

Einbuch-Mechanismus: Verfahren mit QA



Derive-Programm:

- Verändert Daten, passt an
- prüft Integrity (Foreign Key Constraints, höhere Abhängigkeiten, etc.), vermerkt Fehler beim Datensatz
- trägt richtige Datensätze in _derive Tabellen ein
- stellt falsche Datensätze in Korrektur-Zyklus

Master Data Concept

Danke für die Aufmerksamkeit

austriamicrosystems