



# Plattform-Migration mit Data Guard

## Eine Migration innerhalb von Minuten

Martin Bracher, Senior Architect

OPITZ CONSULTING Schweiz GmbH



Baden, 25. März 2010

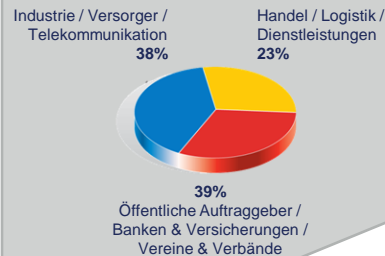


## Märkte

- Java
- SOA
- ORACLE
- BI/DWH
- Outtasking

## Kunden

- Branchen-  
übergreifend
- Über 600  
Kunden



## Leistungs- angebot

- IT-Strategie
- Beratung
- Implementierung
- Betrieb
- Training



## Fakten

- Gründung 1990
- 400 Mitarbeiter
- 8 Standorte in  
CH/D/P



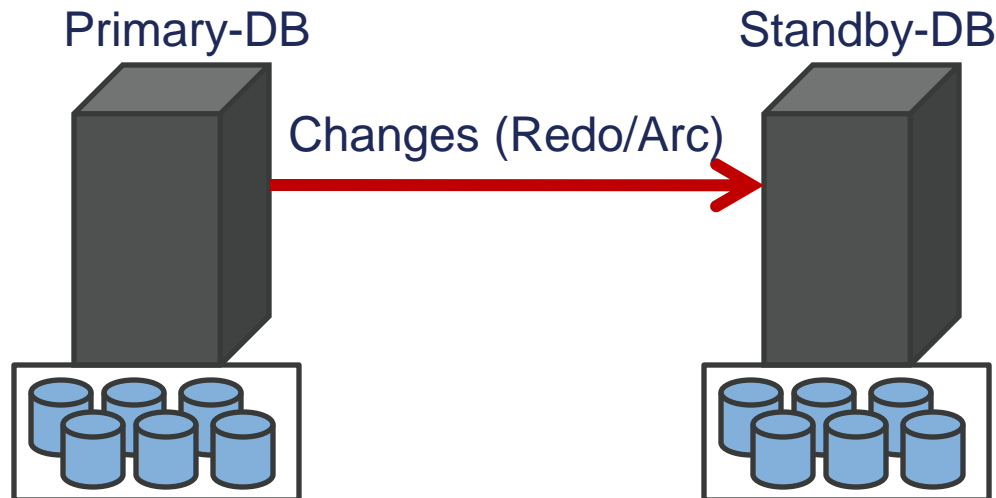
# Agenda

---

- 1. Einleitung: Standby Database, Data Guard**
- 2. Migration auf anderen Server: mögliche Varianten**
- 3. Migration auf anderen Server mit Data Guard**
  - Voraussetzungen, Möglichkeiten, Einschränkungen
  - Migration auf dieselbe Plattform, 32bit → 32bit
  - Migration auf dieselbe Plattform, 32bit → 64bit
  - Migration Windows 32bit → Linux 32bit
  - Migration Windows 32bit → Linux 64bit

# Einleitung

## ■ Was ist eine Standby Database (Data Guard) ?



- Eine Standby-Datenbank (Standby) ist eine Kopie einer Primärdatenbank, (Primary) die fortlaufend mit den Änderungen der Primary nachgeführt wird (synchron oder asynchron, mit oder ohne allfälligen Datenverlust)
- Wenn die Primary ausfällt, kann die Standby "aktiviert" werden, d.h. sie übernimmt bei einem Failover die Rolle der Primary
- Zu Wartungszwecken kann die Rolle auch getauscht werden ("switchover")

# Einleitung

---

## ■ Data Guard

- Automatisierung und Überwachung der Standby-Datenbank
  - Automatischer Log-Transfer, automatische Log-Applizierung
- Direkter Redo-Transfer, muss nicht auf Archivelog warten
- Bestandteil der Enterprise Edition
- (bei Standard-Edition nur Standby-Funktionalität ohne o.g. Automatismen)

## ■ Ziel von Data Guard ist Daten-Hochverfügbarkeit

- Eine vollständige Kopie der Datenbank steht auf unabhängiger Hardware zur Verfügung (eigener Server, eigener Storage)
- Die Systeme können weit auseinander stehen
  - relativ geringe Anforderungen an Bandbreite und Latenzzeiten (im Gegensatz zu RAC)
- Durch Switchover/Failover ist auch eine gewisse Service-Hochverfügbarkeit gegeben

# Migration auf einen anderen Server

## ■ Neue Datenbank aufbauen und Daten importieren

### ■ Oracle exp / imp Utility

- Vorteil: in allen Oracle Versionen verfügbar
- Vorteil: Plattform-unabhängig
- Nachteil: langsam (kein direct-path-load, keine Parallelisierung)
- Nachteil: nicht alle Datentypen unterstützt

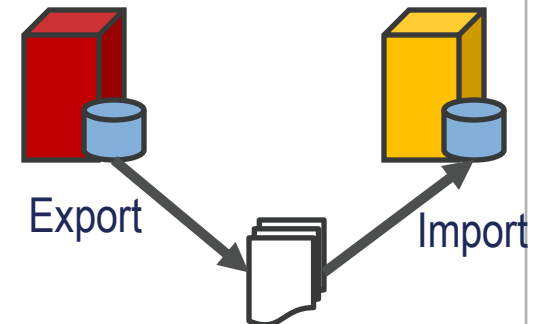
### ■ Oracle Datapump expdp / impdp Utility

- Vorteil: schneller als exp/imp (direct-path-load, Parallelisierung)
- Nachteil: erst ab Oracle 10g verfügbar
- Nachteil: nicht alle Datentypen korrekt unterstützt

### ■ Eventuell applikatorischer Datentransfer

### ■ Generelle Vor- und Nachteile

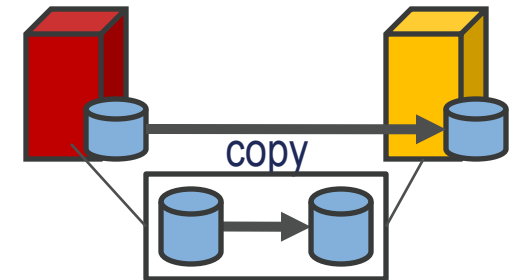
- Vorteil: Gleichzeitige Migration auf neuere Datenbank möglich
- Vorteil: Neu aufgebaute, reorganisierte Datenbank, Nutzung aller neuen Features möglich
- Nachteil: Langsam. Insbesondere der Aufbau von Indexten kann sehr lange dauern



# Migration auf einen anderen Server

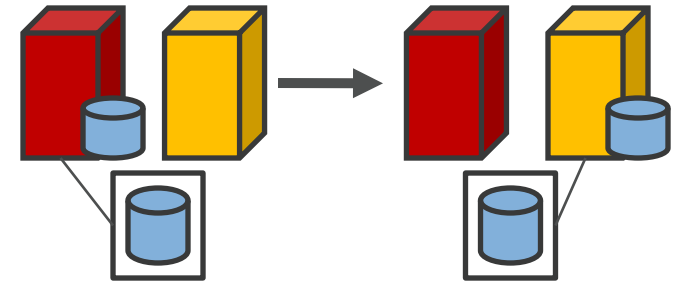
## ■ Kopieren / verschieben der bestehenden Datenbank auf einen neuen Server

- Kopieren via Netzwerk (NFS, SMB, scp, ftp, ...) oder via SAN
  - Vorteil: bei Misserfolg steht alte Umgebung noch zur Verfügung



## ■ Verschieben des bestehenden Storage auf neuen Server

- Physischer Transfer der Disks, oder
- umhängen der LUN's im SAN an neuen Server



## ■ Vorteile

- Technisch einfach
- Risikoarm, da keine Änderung an der Datenbank

## ■ Nachteile

- Lange Downtime für die Zeit des Kopierens/Verschiebens
- Nur innerhalb der gleichen Plattform
- Kein Upgrade der Oracle-Version
- Keine Reorganisation der Daten/Indices

# Migration auf einen anderen Server

---

## ■ Migration mit Transportable Tablespaces

- Tablespaces können an eine neue Datenbank "angehängt" werden
- Vorteil: funktioniert auch über Plattformgrenzen (rman convert)
- Vorteil: auch unterschiedliche Versionen möglich
- Nachteil: nicht möglich für SYSTEM Tablespace
- Nachteil: User, Prozeduren (deren Information in SYS-Tabellen liegen) müssen vorgängig erstellt werden

## ■ Migration mit Transportable Database

- Erfolgt über RMAN
- Vorteil: gewisse Plattform-Migrationen sind möglich
- Nachteil: Nur für Plattfomen desselben Endian-Formats

# Migration auf einen anderen Server

---

## ■ Reduktion der Downtime

- Kopieren der Datenbankfiles ist schneller als Export/Import der Daten
  - Keine Neu-Berechnung von Indices
- Ab Beginn des Kopierens muss die Datenbank gestoppt sein
  - Sonst gehen nachfolgende Änderungen verloren
- Gibt es weitere Möglichkeiten?

## ■ Data Guard für eine Plattform-Migration?

- Es werden auch Datafiles kopiert für die Standby, aber dies geschieht bei laufender Produktion (keine Downtime)
- Durch die Applizierung der Änderungen haben wir immer eine aktuelle Kopie, die schon bereit steht
- Im Folgenden werden wir uns die Möglichkeiten und Einschränkungen dieser Migrationsvariante anschauen

# Unterstützte Data Guard (DG) Konfigurationen

## ■ Metalink-Note 413484.1 (Teil 1)

- Jede DG-Konfiguration, wo die Primary und Standby auf derselben Oracle Plattform läuf

### ■ Oracle Plattform:

- Oracle-Software von denselben Medien installiert (gleiche CD, gleiche URL)
- Dieselbe Plattform-ID

```
SQL> select platform_id, platform_name from v$database;
```

```
PLATFORM_ID PLATFORM_NAME
```

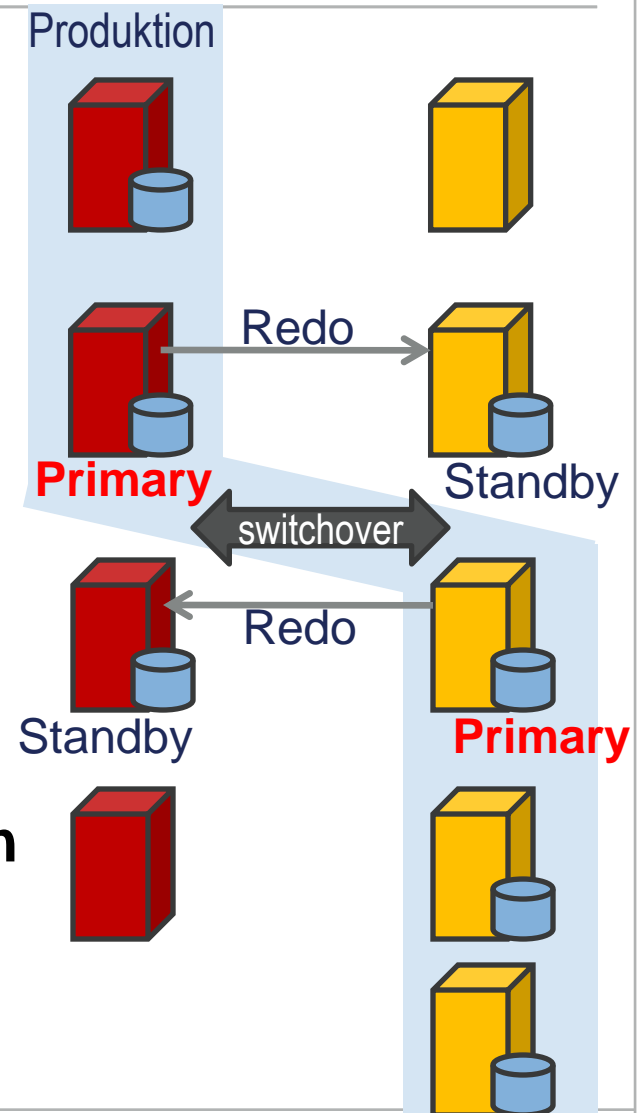
```
-----
```

```
10 Linux IA (32-bit)
```

- Unterschiede der Server von Primary- und Standby sind erlaubt, solange obenstehende Bedingungen erfüllt sind und der Server/OS zertifiziert ist
  - Unterschiedlicher Hersteller (z.B. HP / IBM)
  - Unterschiedliche CPU, unterschiedliche Speichergösse
  - Unterschiedliche OS-Distribution oder Version (RedHat/SuSE, OEL4/OEL5)

# Migration Server mit gleicher Oracle Plattform

- **Aufbau eines neuen Servers**
  - Betriebssystem (kann neuere Version sein)
  - Oracle-Software (Identische Version)
- **Aufbau einer Data Guard Umgebung zwischen dem alten und neuen Server**
  - Dies erfolgt ohne Downtime der Primary
- **Switchover auf neuen Server**
  - Kurzer Unterbruch im Minutenbereich
  - Danach arbeiten Clients auf neuem Server
- **Entfernen der Data Guard Konfiguration**
- **Entfernen des alten Servers**



# Migration von 32- auf 64-bit Server

## ■ Gemäss vorheriger Beschreibung keine unterstützte Konfig.

- Unterschiedliche Plattform-ID (z.B. 10=Linux 32bit, 11=Linux 64bit)

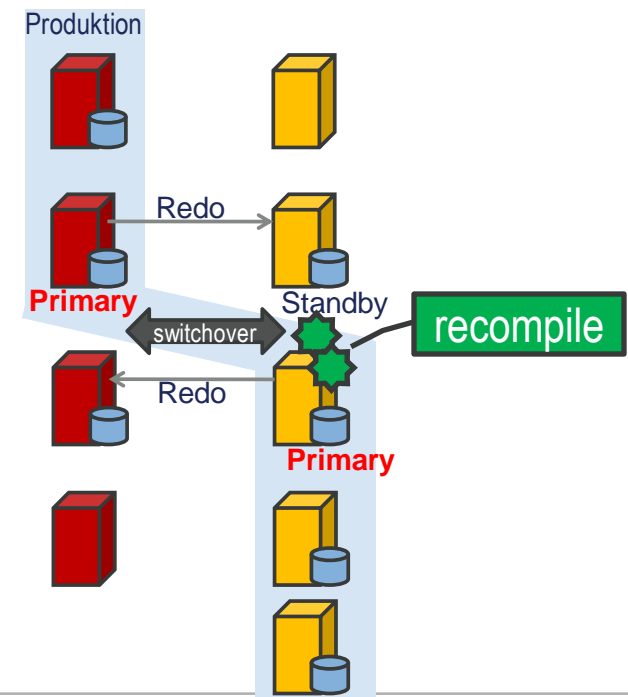
## ■ Ab Oracle 10g

- Zusätzlicher Support gewisser DG-Konfigurationen innerhalb derselben Betriebssystem-Familie (Linux x86 / Linux x86\_64, Windows 32bit / 64bit)

- Primary kann mit 32bit, Standby mit 64bit Oracle betrieben werden

- Die Version muss identisch sein
- Data Guard Broker funktioniert erst ab 11g, bei 10g Verwendung von sqlplus
- Nach Failover/Switchover müssen auf der neuen Primary alle PL/SQL Module invalidiert und danach neu kompiliert werden

- Metalink-Note: 414043.1

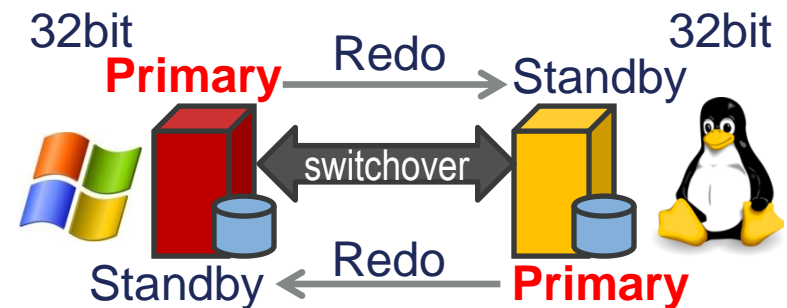


# Migration zu anderen Betriebssystemen

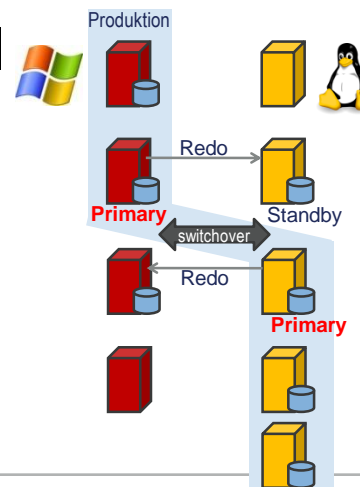
- Wie bisher gezeigt, sind nur (gewisse) Konfigurationen innerhalb derselben Betriebssystem-Familie supported

- **Ab Oracle 11g**

- Kombination von Windows und Linux DG-Konfiguration ist möglich
- Intel/AMD CPU, nicht Itanium
- Nur 32-Bit
- Metalink-Note 413484.1



- **Migration analog erstem Beispiel**



# Einige Data Guard Plattform Kombinationen

Standby Primary	Windows 32bit (ID=7)	Windows 64bit (ID=12)	Linux 32bit (ID=10)	Linux 64bit (ID=13)	HPUX PA-RISC (ID=3)	HPUX Itanium (ID=4)
Windows 32bit			11g			
Windows 64bit						
Linux 32bit	11g					
Linux 64bit						
HPUX PA-RISC						
HPUX Itanium						

 Kombination 32/64Bit, Neu-Kompilierung von Packages erforderlich

# Wieso Windows ↔ Linux nur ab 11g ?

## ■ Data Guard 10g funktioniert bis auf ein kleines Detail...

- Die ArchiveLogs können zwar transportiert, aber nicht appliziert werden ☹
- Eine DG-Broker Konfiguration funktioniert mit log\_apply=off

```
Errors in file /u00/app/oracle/admin/DB1020/bdump/db1020_p001_3061.trc:
```

```
ORA-00339: archived log does not contain any redo
```

```
ORA-00334: archived log: '/u00/oradata/DB1020/REDOG1M1DB1020.DBF'
```

```
ORA-00600: internal error code, arguments: [4502], [0], [], [], [], [], [], []
```

```
Sun Feb  8 12:55:30 2010
```

```
Errors in file /u00/app/oracle/admin/DB1020/bdump/migra102_p001_3061.trc:
```

```
ORA-10562: Error occurred while applying redo to data block (file# 1, block# 4559)
```

```
ORA-10564: tablespace SYSTEM
```

```
ORA-01110: data file 1: '/u00/app/oracle/product/10.2.0/dbs/D:U00ORADATADB1020SYSTEM01.DBF'
```

```
ORA-10561: block type 'TRANSACTION MANAGED DATA BLOCK', data object# 574
```

```
ORA-00600: internal error code, arguments: [4502], [0], [], [], [], [], [], []
```

# Wieso nur Windows ↔ Linux (11g)

## ■ Linux und Windows haben dasselbe Endian-Format

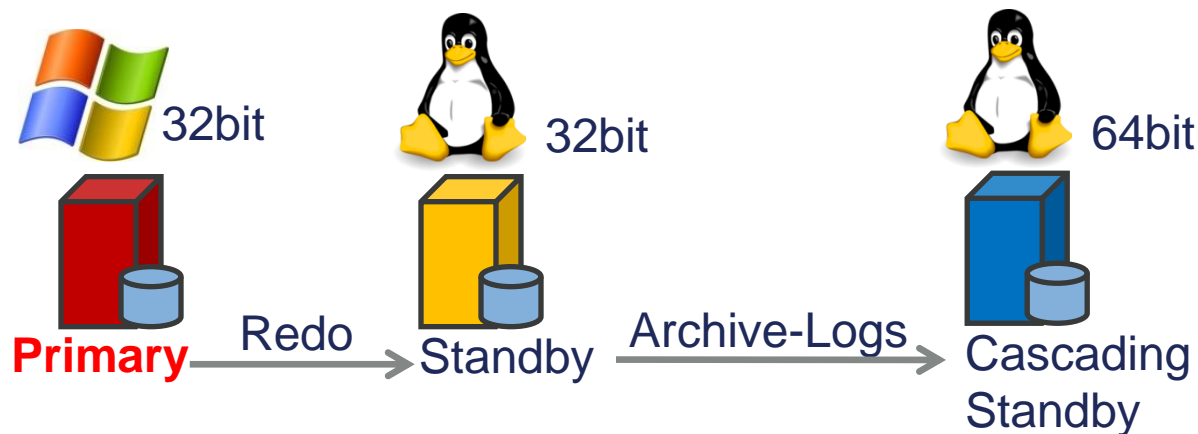
- Mit 11g Release 1 sind Windows und Linux die einzigen 32-Bit Plattformen mit Little-Endian (11.1 für Solaris x86 gab es nie)

```
SQL> select * from v$transportable_platform;
```

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
1	Solaris[tm] OE (32-bit)	Big
2	Solaris[tm] OE (64-bit)	Big
7	Microsoft Windows NT	Little
10	Linux IA (32-bit)	Little
6	AIX-Based Systems (64-bit)	Big
3	HP-UX (64-bit)	Big
5	HP Tru64 UNIX	Little
4	HP-UX IA (64-bit)	Big
11	Linux IA (64-bit)	Little
15	HP Open VMS	Little
8	Microsoft Windows IA (64-bit)	Little
9	IBM zSeries Based Linux	Big
13	Linux 64-bit for AMD	Little
16	Apple Mac OS	Big
12	Microsoft Windows 64-bit for AMD	Little
17	Solaris Operating System (x86)	Little

# Migration Windows 32bit → Linux 64bit

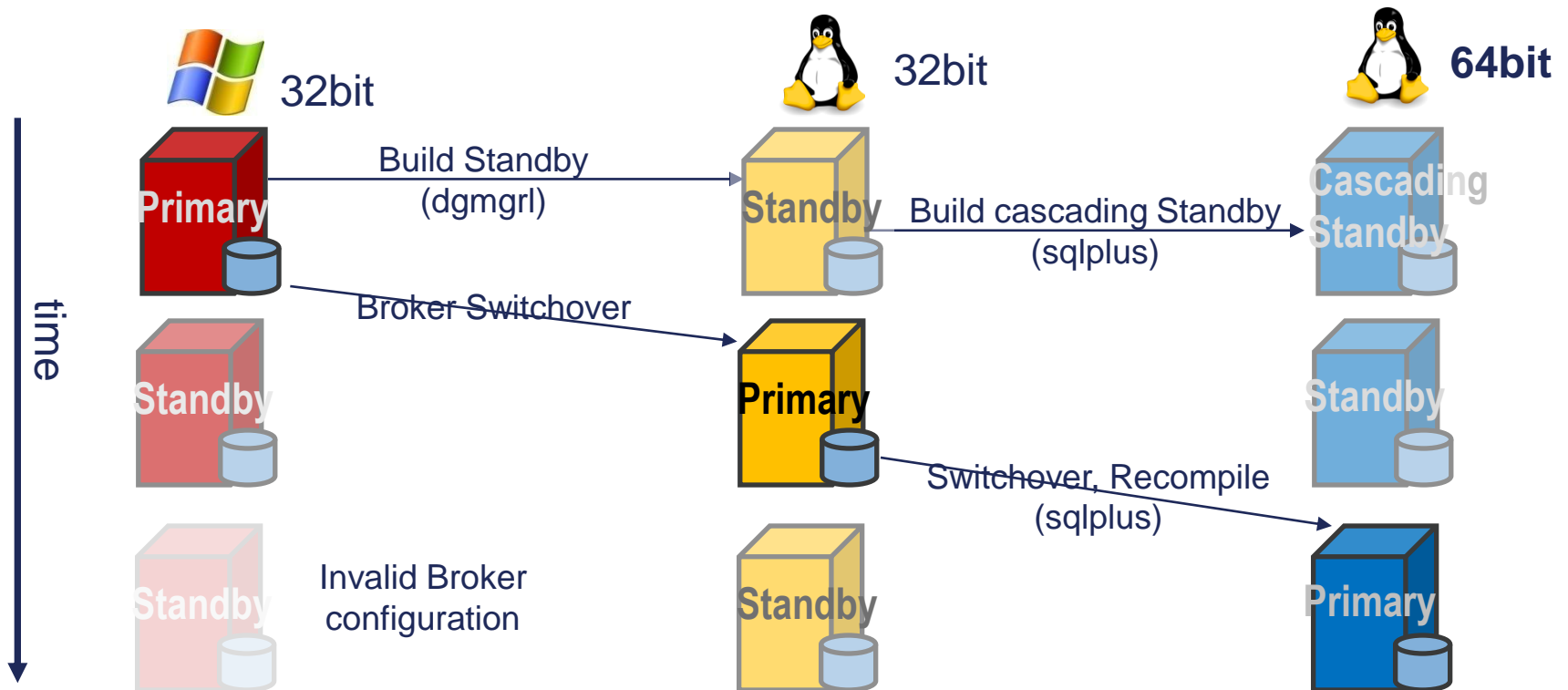
- Ist mit Data Guard eigentlich nicht möglich, aber:
  - Windows 32bit → Linux 32bit ist unterstützt
  - Linux 32bit → Linux 64bit ist unterstützt
  - ... und das lässt sich kombinieren mit einer kaskadierenden Standby Konfiguration (eine Standby, die ihre Logs von einer Standby kriegt)



- Konfiguration von Primary zu Standby kann mit DG-Broker gemacht werden
- Konfiguration der Cascading Standby erfolgt über sqlplus
  - kein synchroner Transfer mehr (Transport von ArchiveLog)

# Migration Windows 32bit → Linux 64bit

- Migration in 2 Schritten: 32bit → 32bit, 32bit → 64bit



# Pfadnamen unter Windows und Unix

---

- **Windows: Laufwerksbuchstaben (C:, D:)**
- **Unix: Mountpoints (/u00, /u01)**
- **Directory-Trennzeichen**
  - Windows: \
  - Unix: /
  - Das Windows-Trennzeichen "\" wird unter Unix als Escape-Sequenz interpretiert
    - z.B. C:\oradata\DB\system.dbf → C:oradataDBsystem.dbf
    - Ein Restore erzeugt unter Unix folgendes File:  
\$ORACLE\_HOME/dbs/**C:oradataDBsystem.dbf**

# Pfadnamen unter Windows und Unix

## ■ Verhindern solcher Umbenennungen

- Parameter `db_file_name_convert = 'quelle','ziel'`
- Parameter `log_file_name_convert = 'quelle','ziel'`
- 'ziel' ist die lokale Plattform
- Wichtig: Unix ist Case-Sensitiv. Die Schreibweise für die Quelle muss dem Wert entsprechen, wie er in `v$datafile` steht (nicht was der Explorer anzeigt)  
z.B. `C:\ORADATA\DB1120\SYSTEM01.DBF`

```
db_file_name_convert='C:\oradata\DB1110\','/u00/oradata/DB1110/'
```

- ergibt `/u00/app/oracle/product/11.1.0/dbs/C:ORADATADB1120SYSTEM01.DBF`

```
db_file_name_convert='C:\ORADATA\DB1110\','/u00/oradata/DB1110/'
```

- ergibt `/u00/oradata/SYSTEM01.DBF`

# Setup Standby mit Dataguard Broker

## ■ Erstellen der Standby Datenbank

- Erstellen/konfigurieren Directories, Passwort-File, listener.ora, tnsnames.ora
- Vorbereiten der Primary und erstellen des spfile für die Standby

```
sqlplus / as sysdba
alter database force logging;
alter system set log_archive_dest_1='location="C:\oracle\admin\DB1110\arch"';
alter system set db_unique_name='DB1120_W32' scope=spfile;
alter system set standby_file_management='AUTO';
alter system set db_file_name_convert='/u00/oradata/DB1110/', 'C:\oradata\DB1110\'
scope=spfile;
alter system set
log_file_name_convert='/u00/oradata/DB1110/', 'C:\oradata\DB1110\' scope=spfile;
...
create pfile='C:\tmp\init.ora' from spfile;

perl -p -i -e 's=\\=/=g' init.ora
perl -p -i -e "s=C:/oracle=/u00/app/oracle=gi" init.ora
perl -p -i -e "s=C:/oradata=/u00/oradata=gi" init.ora
perl -p -i -e \
"s-e_convert.*-e_convert='C:\\\\ORADATA\\\\DB1110\\\\', '/u00/oradata/DB1110/'-g" init.ora
perl -p -i -e "s/DB1110_W32/DB1110_L32/g" init.ora
```

# Setup Standby mit Dataguard Broker

## ■ Falls man die Konvertierung vergisst...

```
ls -l $ORACLE_HOME/dbs/D*
...
-rw-r----- 1 oracle dba      8389120 Feb  7 15:24 D:ORADATADB1110REDOG2M2.DBF
-rw-r----- 1 oracle dba      8389120 Feb  7 15:24 D:ORADATADB1110REDOG3M1.DBF
-rw-r----- 1 oracle dba      8389120 Feb  7 15:24 D:ORADATADB1110REDOG3M2.DBF
-rw-r----- 1 oracle dba 262152192 Feb  7 15:24 D:ORADATADB1110SYSTEM01.DBF
-rw-r----- 1 oracle dba 209723392 Feb  7 11:37 D:ORADATADB1110TEMP01.DBF
-rw-r----- 1 oracle dba 314580992 Feb  7 15:24 D:ORADATADB1110UNDOTS01.DBF
```

## ■ Duplizierung der Datenbank

```
rman target sys/manager@db1110_w32 auxiliary sys/manager@db1110_l32
run {
allocate channel w32 type disk;
allocate auxiliary channel l32 type disk;
duplicate target database for standby from active database dorecover nofilenamecheck;
}
```

# Setup Standby mit Dataguard Broker

---

## ■ Konfiguration wie üblich

```
create configuration 'DB1110' as
  primary database is 'DB1110_W32'
  connect identifier is 'DB1110_W32';
```

```
add database 'DB1110_L32' as
  connect identifier is 'DB1110_L32'
  maintained as physical;
```

```
edit database 'DB1110_W32' set property logxptmode='SYNC';
edit database 'DB1110_L32' set property logxptmode='SYNC';
edit database 'DB1110_W32' set property standbyfilemanagement='AUTO';
edit database 'DB1110_L32' set property standbyfilemanagement='AUTO';
edit database 'DB1110_W32' set property
StandbyArchiveLocation='C:\oracle\admin\DB1110\arch';
edit database 'DB1110_L32' set property
StandbyArchiveLocation='/u00/app/oracle/admin/DB1110/arch';
```

```
edit configuration set protection mode as maxperformance;
```

```
enable configuration;
```

# Setup Cascading Standby

---

## ■ Erstellung der Cascading Standby manuell oder mit RMAN

- Nur notwendig, wenn wir im gleichen Wartungsfenster auf 64bit migrieren wollen, ansonsten Aufbau nach Migration Win32→Linux32
- Vorbereitung der Umgebung wie vorher (Verzeichnisse, \*.ora, spfile, ...)

### ■ Manuell

- Stop der Standby (oder Stoppen des Log-Apply)
- Kopieren der Files (spfile nicht überschreiben!)
- Allfällig kopierte Broker-Files löschen

### ■ RMAN

- Duplizierung "from active database" ist nicht möglich

```
RMAN-03002: failure of Duplicate Db command at 03/08/2010 11:14:53
```

```
...
```

```
ORA-17627: ORA-12577: Message 12577 not found; product=RDBMS; facility=ORA
```

- Duplizierung ab RMAN-Backup funktioniert teilweise → siehe später

# Setup Cascading Standby

---

## ■ Konfiguration Data Guard (ohne Broker)

### ■ nur minimale Konfiguration, um Kopie aktuell zu halten

```
### Linux32
```

```
alter system set fal_server='DB1110_W32,DB1110_L64';  
alter system set fal_client='DB1110_L32';  
alter system set log_archive_config='dg_config=(DB1110_W32,DB1110_L32,DB1110_L64)';  
alter system set log_archive_dest_2='service="DB1110_L64", db_unique_name="DB1110_L64"  
valid_for=(STANDBY_LOGFILE,STANDBY_ROLE)';  
alter system set standby_archive_dest='/u00/app/oracle/admin/DB1110/arch';
```

```
### Linux64
```

```
alter system set fal_server='DB1110_L32';  
alter system set fal_client='DB1110_L64';  
alter system set log_archive_config='dg_config=(DB1110_W32,DB1110_L32,DB1110_L64)';  
alter system set standby_archive_dest='/u00/app/oracle/admin/DB1110/arch';  
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

# Switchover Teil 1: Windows 32bit → Linux 32bit

---

## ■ Migrationsdauer

- ist nicht von der Datenbankgrösse und Anzahl Objekte abhängig
- wird durch die Menge der noch zu applizierenden Änderungen beeinflusst

```
DGMGRL> switchover to 'DB1110_L32';  
Performing switchover NOW, please wait...  
New primary database "DB1110_L32" is opening...  
Operation requires shutdown of instance "db1110" on database "DB1110_W32"  
...  
Switchover succeeded, new primary is "DB1110_L32"
```

## ■ Fallback-Szenario

- Sollte die Applikation nicht korrekt funktionieren, kann wieder ein Switchover auf die alte Maschine zurück erfolgen

# Switchover Teil 2: Linux 32bit → Linux 64bit

## ■ Migrationsdauer

- ist nicht von der Datenbankgrösse abhängig
- wird durch die Menge der noch zu applizierenden Änderungen beeinflusst
- Die Dauer ist von der Anzahl und Grösse der Packages/Funktionen abhängig

## ■ Erstellung Broker-Konfiguration für die 64bit-Standby

- Ist optional, Switchover kann auch aus sqlplus heraus gemacht werden

```
add database 'DB1110_L64' as ...
```

```
edit database 'DB1110_L64' set property ...
```

```
DGMGRL> switchover to 'DB1110_L64';
```

```
Performing switchover NOW, please wait...
```

```
New primary database "DB1110_L64" is opening...
```

```
Operation requires shutdown of instance "db1110" on database "DB1110_L32"
```

```
...
```

```
Switchover succeeded, new primary is "DB1110_L64"
```

# Switchover Teil 2: Linux 32bit → Linux 64bit

## ■ "succeeded" mit Fehlern...

### ■ Die Datenbank ist noch nicht richtig benutzbar

- Einfache SELECT auf Tabellen funktionieren schon

- Code ist aber invalid

```
SQL> select object_name, object_type from all_objects where status='INVALID';
select object_name, object_type from all_objects where status='INVALID'
```

\*

ERROR at line 1:

```
ORA-06544: PL/SQL: internal error, arguments: [56319], [], [], [], [], [], [],
[]
```

```
ORA-06553: PLS-801: internal error [56319]
```

- Code muss invalidiert und neu kompiliert werden

```
shutdown immediate
```

```
startup migrate
```

```
@utlirp
```

```
shutdown immediate
```

```
startup
```

```
@utlirp ← dauert lange
```

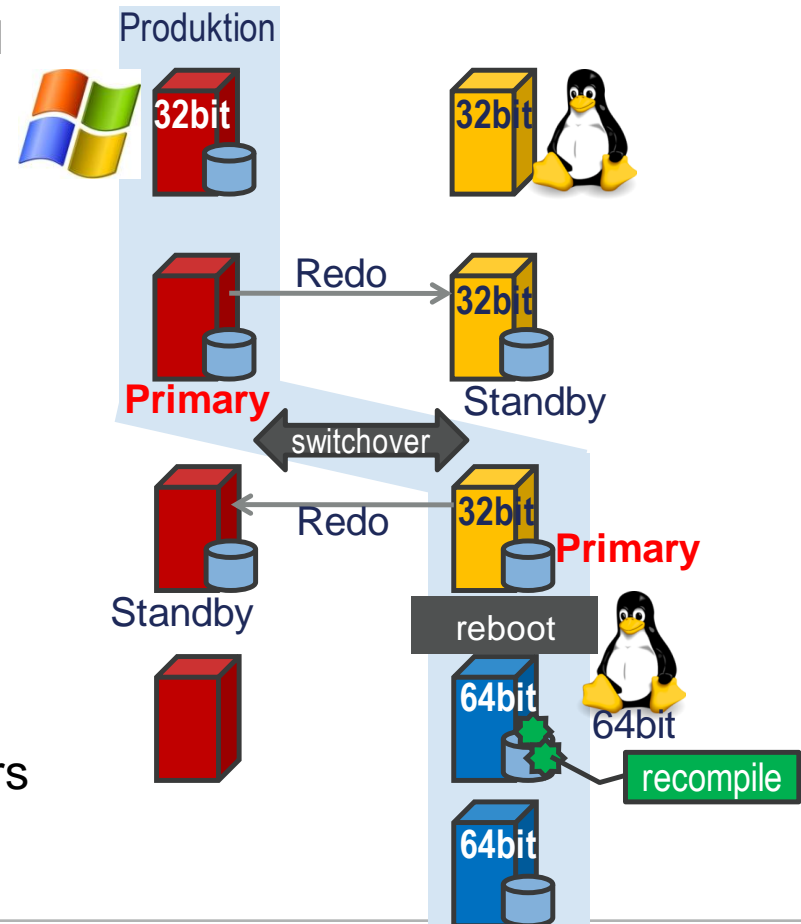
# Migration mit nur 2 Servern

## ■ Ziel-Server mit 32- und 64-bit Software

- Multi-Boot System mit 32- und 64-Bit Linux und Oracle installiert
- 32bit System booten und Migration Teil 1 (Win32 → Linux32) durchführen
- Reboot des Servers zu 64bit und Re-Kompilation der Packages
- Daten-Mountpoints bleiben unverändert

## ■ Alternativ

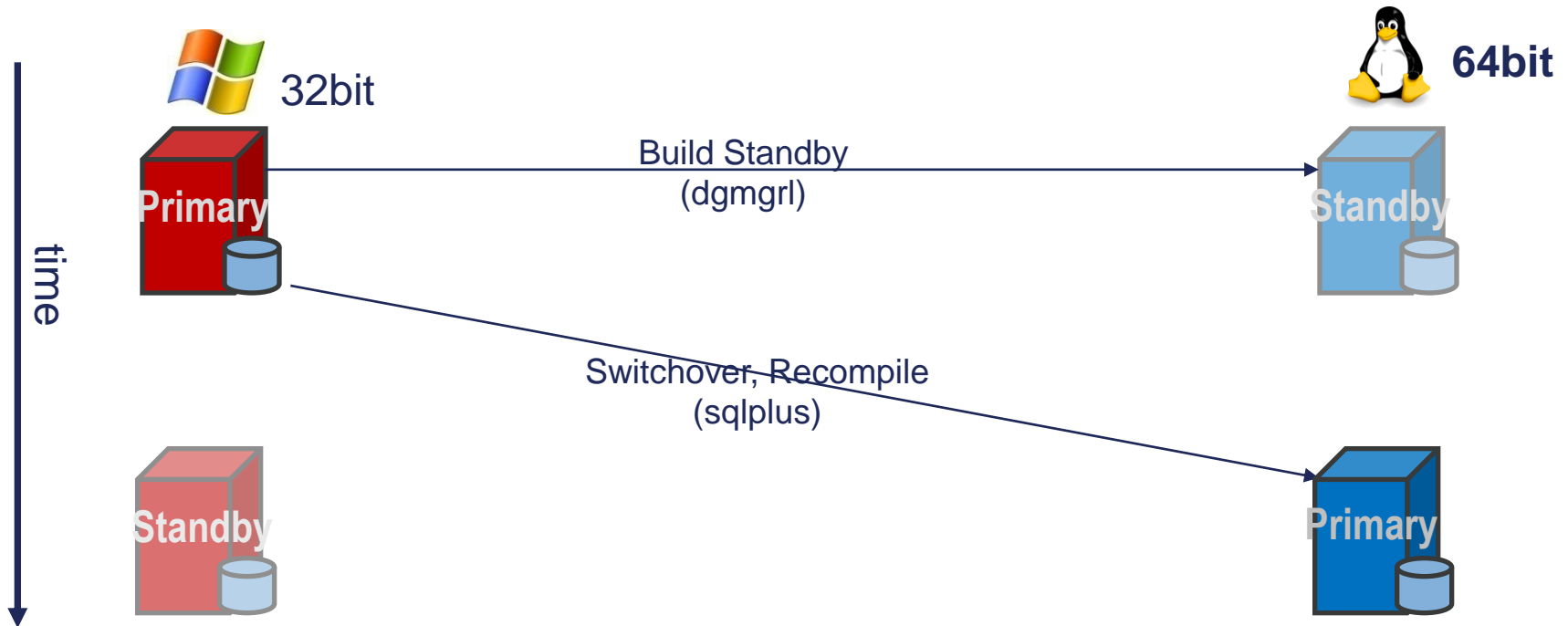
- Linux ist schon 64bit
- 2 Oracle-Home's, 32- und 64-bit
- Offiziell nicht supported 32bit Oracle auf 64bit Linux
- Spart die Zeit für den Reboot des Servers



# Direkte Migration Windows 32bit → Linux 64bit

## ■ Die direkte Migration ist nicht möglich ...

- aber es geht doch 😊
- getestet mit 11.1.0.7



# Direkte Migration Windows 32bit → Linux 64bit

## ■ Erstellung der Standby

### ■ Duplizierung mit RMAN "from active database" klappt nicht

```

RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of Duplicate Db command at 03/08/2010 11:14:53
RMAN-03015: error occurred in stored script Memory Script
RMAN-03009: failure of backup command on w32 channel at 03/08/2010 11:14:53
ORA-17627: ORA-12577: Message 12577 not found;  product=RDBMS; facility=ORA

```

### ■ Duplizierung mit RMAN ab Backupset funktioniert mit Einschränkungen

- Backupset wird unter Windows-Pfad gesucht
- Tapebackup kein Problem, Disk-Backups nach \$ORACLE\_HOME/dbs kopieren / umbenennen  
C:\bck\RMAN\_S1\_T713040503 → \$ORACLE\_HOME/dbs/C:BCKRMAN\_S1\_T713040503

# Direkte Migration Windows 32bit → Linux 64bit

- Duplizierung mit RMAN ab Backupset funktioniert mit Einschränkungen

- Befehl sieht gut aus...

contents of Memory Script:

```
{ ...
  set newname for datafile 1 to
  "/u00/oradata/DB1110/SYSTEM01.DBF";
...
  restore
  clone database
  ;
}
```

- aber das Ergebnis...

channel 164: specifying datafile(s) to restore from backup set

channel 164: restoring datafile 00001 to C:\U00\ORADATA\DB1110\SYSTEM01.DBF

- Es findet zwar eine Umwandlung statt, aber nicht die, die wir gern hätten ☹

# Direkte Migration Windows 32bit → Linux 64bit

## ■ Aufbau der Standby-Datenbank klappt nicht mit Broker

```
DG 2010-03-08-16:47:43      0 2 0 Connection to database DB1110_W32 returns ORA-01017.  
DG 2010-03-08-16:47:43      0 2 0 Please check database DB1110_W32 is using a remote password file,  
DG 2010-03-08-16:47:43      0 2 0 its remote_login_passwordfile is set to SHARED or EXCLUSIVE,  
DG 2010-03-08-16:47:43      0 2 0 and the SYS password is the same as this database.
```

- ... obwohl wir ein korrektes Passwort-File haben ☹
- Erzeugt mit "orapwd ... ignorecase=n" → "add database" klappt nicht
- Erzeugt mit "orapwd ... ignorecase=y" → nach "enable database" Fehler
  
- Lösung: Das Passwort-File muss kopiert und umbenannt werden.
  - Wieso haben wir das bei der 32bit-Migration nicht gemerkt?
  - "Duplicate ... from active database" kopiert das Passwort-File nochmals rüber.
  
- Wenn die Konfiguration "success" ist und die Datenbanken synchron sind, können wir die Migration mit "switchover to 'DB1110\_L64'" starten.
- Der Schritt der Re-Kompilierung ist analog der 2stufigen Migration

# Fazit

---

- **Migration mit Data Guard ist eine sehr schnelle Migrationsmöglichkeit**
- **Einfaches Fallback-Szenario**
  - Switchover/Failover auf die ursprüngliche Seite
- **Beschränkung auf gewisse Plattform-Kombinationen**
  - Mit 11g Release 2 noch mehr Plattformen
  - Solaris x86 ↔ Linux x86(\_64) klappt schon, vermutlich klappen alle x86 Plattformen, ist aber (noch) nicht dokumentiert

# Fragen?

---



OPITZ CONSULTING Schweiz GmbH  
Seestrasse 97, 8800 Thalwil  
044 721 11 22

[www.opitz-consulting.ch](http://www.opitz-consulting.ch)

[Martin.Bracher@opitz-consulting.com](mailto:Martin.Bracher@opitz-consulting.com)