



Hochverfügbarkeit mit Replikation Logisch?

Sebastian Winkler
CarajanDB GmbH

- **Spezialversicherer mit europaweit verteilten Standorten**

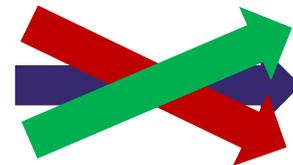
- Zentrale Datenbank zur Kunden- und Vertragsverwaltung



- **Umbau der „alten“ Hochverfügbarkeits-Umgebung**

- aus Enterprise Edition mit Data Guard soll eine Standard Edition One mit leistungsfähiger Hardware und Hochverfügbarkeitslösung werden

- veraltete Hardware
- kein Support
- wenig Leistung
- hohe Lizenzkosten

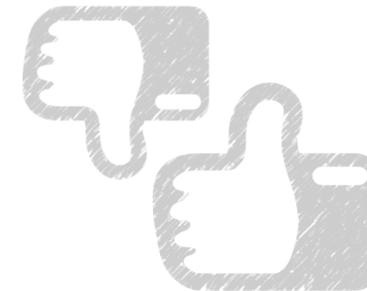


- neue Hardware
- mehr Leistung
- geringere Lizenzkosten
- verbessertes Backup

- **Experten mit über 20 Jahren Oracle Erfahrung**
- **Firmensitz in Erftstadt bei Köln**
- **Spezialisten für**
 - Oracle Datenbank Administration
 - Hochverfügbarkeit (RAC, Data Guard, Failsafe, etc.)
 - Einsatz der Oracle Standard Edition
 - Oracle Migrationen (HW, Unicode, Konsolidierung, Standard Edition)
 - Replikation (GoldenGate, SharePlex, Dbvisit)
 - Performance Tuning
- **Fernwartung**
- **Schulung und Workshops (Oracle, Toad)**



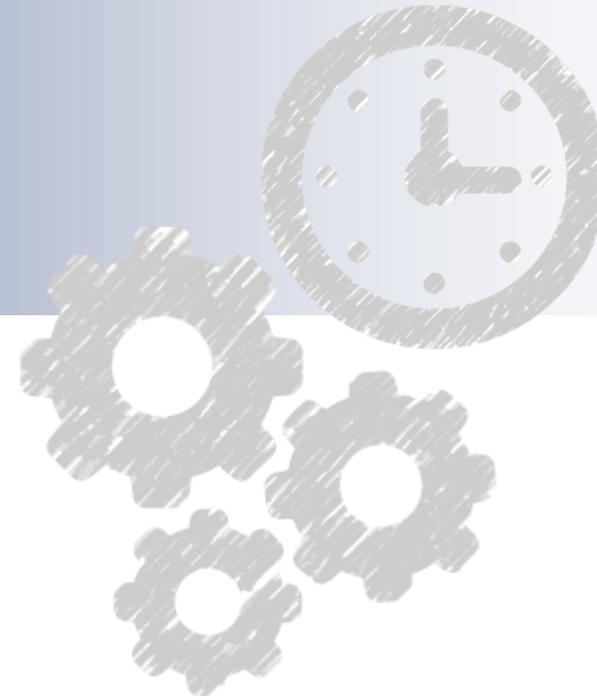
- **Hochverfügbarkeit**
 - Standard vs. Enterprise
- **Replikation**
 - Logisch vs. physikalisch
- **Projekt**
 - Herausforderungen und Bewältigung
- **Fazit**
 - Logisch ... oder doch nicht?





Hochverfügbarkeit

- Begriff
- Unterscheidung
- Lösungen SE + EE



Was bedeutet Hochverfügbarkeit?

- **Hochverfügbarkeit** (englisch high availability, HA)
 - Die Fähigkeit eines Systems, einen unterbrechungsfreien Betrieb einer Anwendung, auch im Fehlerfall (bspw. Ausfall einer Systemkomponente), mit einer möglichst hohen Wahrscheinlichkeit zu gewährleisten.
- **HA Lösungsansätze Oracle**
 - Real Application Clusters
 - Active Data Guard
 - GoldenGate



- **Klärung der Anforderungen**
 - Welche HA Lösung soll überhaupt zum Einsatz kommen?
- **Was soll abgesichert werden?**
 - Ausfall der Anwendung
 - Datenverlust (... Disaster Recovery)
- **Unterschiedliche Anforderungen bedeuten unterschiedliche Lösungen**
 - Technisch ... Data Guard vs. RAC
 - Monetär ... Data Guard vs. GoldenGate



- **Hochverfügbarkeit „nur“ mit Enterprise Edition?**
 - **Oracle Data Guard** is available only as a feature of Oracle Database Enterprise Edition. It is not available with Oracle Database Standard Edition. (11g Data Guard Documentation)



Primary DB Server



Secondary DB Server

- **Note:** It is possible to simulate a standby database environment with databases running Oracle Database Standard Edition. You can do this by manually transferring archived redo log files using an operating system copy utility or using custom scripts that periodically send archived redo log files from one database to the other. The consequence is that this configuration does not provide the ease-of-use, manageability, performance, and disaster-recovery capabilities available with Data Guard

- Archivelog Mode



- Gleiche Ordnerstruktur
- Files kopieren
- Recover Standby Database
- Mount Standby Database

- **Hochverfügbarkeit „nur“ mit Enterprise Edition?**
 - **Oracle Data Guard** is available only as a feature of Oracle Database Enterprise Edition. It is not available with Oracle Database Standard Edition. (11g Data Guard Documentation)
- **Oracle GoldenGate**
 - auch mit Standard Edition einsetzbar
 - plattformübergreifend, versionsübergreifend, bi-direktional
 - umfangreiche Features ← → Kostenfaktor
- **Drittanbieterlösungen**
 - SharePlex (Dell Software), Dbvisit Standby + Replicate, und weitere...



- Oracle Streams

- Streams als Replikationswerkzeug seit 11gR1 auch für Standard Edition
 - **Note:** If you are using Oracle Database 11g Standard Edition, then synchronous capture is the only Oracle Streams component that can capture database changes automatically. To use capture processes, you must have Oracle Database 11g Enterprise Edition. (11g Doc.)
 - **Streams und Advanced Replication mit 12cR1 DEPRECATED:**
 - Oracle hat die Entwicklung eingestellt
 - In 12c enthalten mit eingefrorenem Entwicklungsstand und Funktionsumfang von 11gR2
 - „deprecated“ ist nur eine Stufe vor „desupported“ = eine Frage der Zeit ...
- Laufen in 11gR2 wie gehabt weiter
 - 11gR2 Premier Support Ende: Januar 2015, Extended Support Ende: Januar 2018
 - Spätestens mit Upgrade auf 12c nur mit Einschränkungen nutzbar
 - Oracle empfiehlt: Umstieg auf GoldenGate



Replikation

- Begriff
- Unterscheidung
- Physikalische Replikation
- Logische Replikation



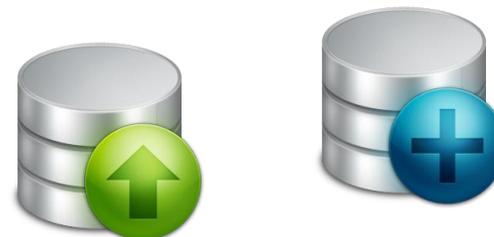
- **Replikation**

- Replikation oder Replizierung (lat. *replicare* „erwidern“, „wiederholen“) bezeichnet die mehrfache Speicherung derselben Daten an meist mehreren verschiedenen Standorten und die Synchronisation dieser Datenquellen.



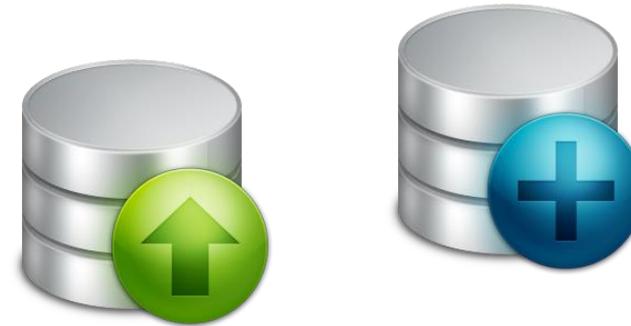
- **Standby**

- Der Bereitschaftsbetrieb oder Standby-Betrieb (auch Wartebetrieb) ist der Zustand eines technischen Gerätes, in dem die eigentliche Nutzfunktion temporär deaktiviert ist, aber jederzeit und ohne Vorbereitungen oder längere Wartezeiten wieder aktiviert werden kann.



Was ist Standby?

- Eine spezielle Art der Replikation
- Voraussetzung:
 - Quelle und Ziel sind identisch
 - Gleiche Endianness
 - Komplette Datenbank
- Eventuell eingeschränkte Nutzung
 - Physical Standby
 - Active DataGuard → Read Only
 - Logical DataGuard → Änderungen nur an „anderen“ Strukturen
- In der Regel Replikation auf anderen Server
- Direkter Zugriff auf Redolog-Informationen → REDO APPLY



... und was ist dann Replikation?

- Einzelnes Objekt, Objektgruppe oder gesamte Datenbank
- Auch innerhalb einer Datenbank möglich (z.B. Materialized Views)
- Zusätzliche Transformationen oder Filter möglich
- Quell- und Zieldatenbanken müssen getrennt verwaltet werden
 - Größere Flexibilität
 - Unterschiedliche Datenbankgrößen
- Zugriff auf Redolog-Informationen oder über Trigger
- SQL APPLY
 - Aus Redo Logs werden SQL Statements
 - Ziel-DB ist offen für Änderungen
 - Datentyp- und andere Restriktionen



- **Standby Lösung, weil „eigentliche Nutzfunktion temporär deaktiviert“**

- Physical Standby
- Active Data Guard

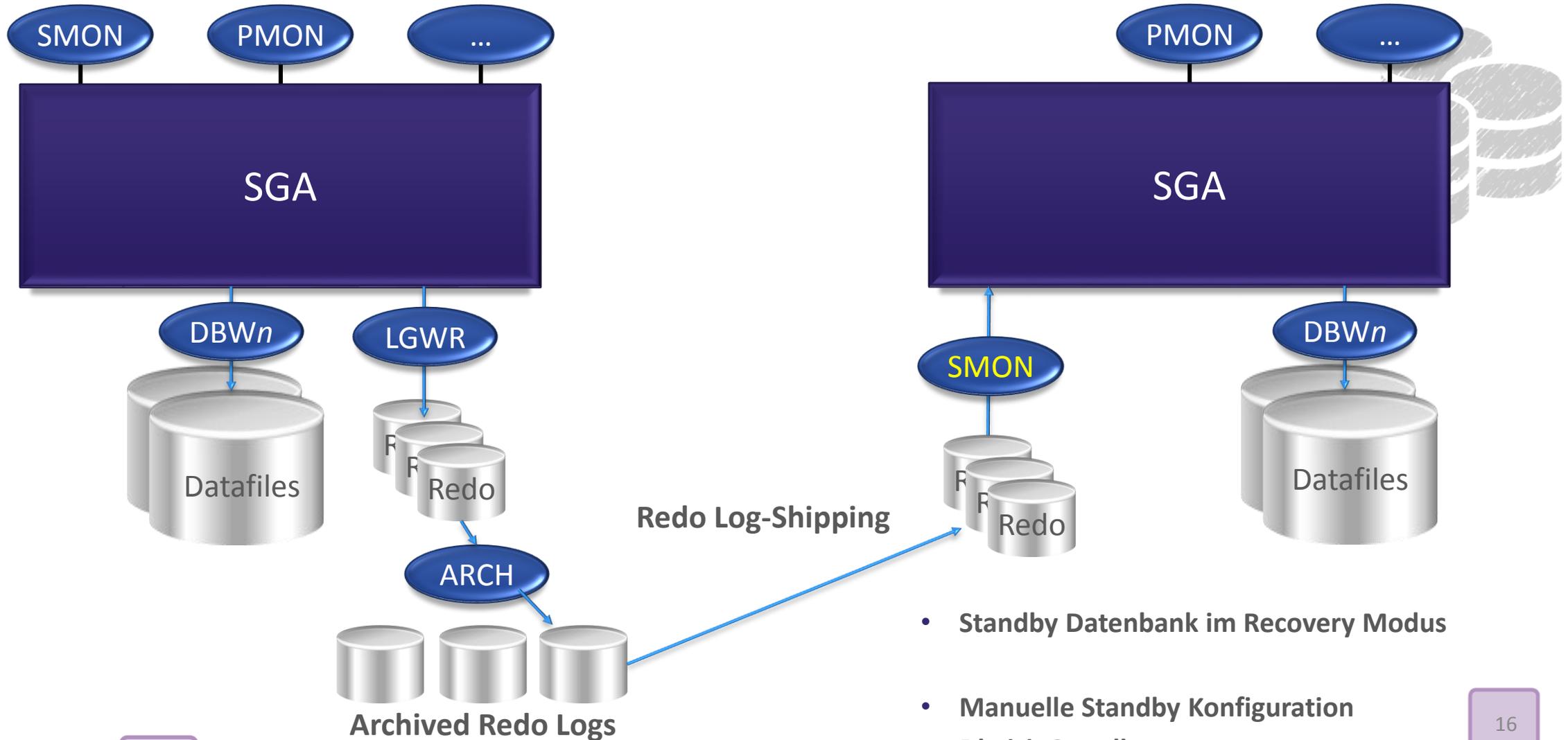


- **Replikationslösung, weil „mehrfache Speicherung derselben Daten an mehreren Standorten“**

- Logical Standby
- Active Data Guard

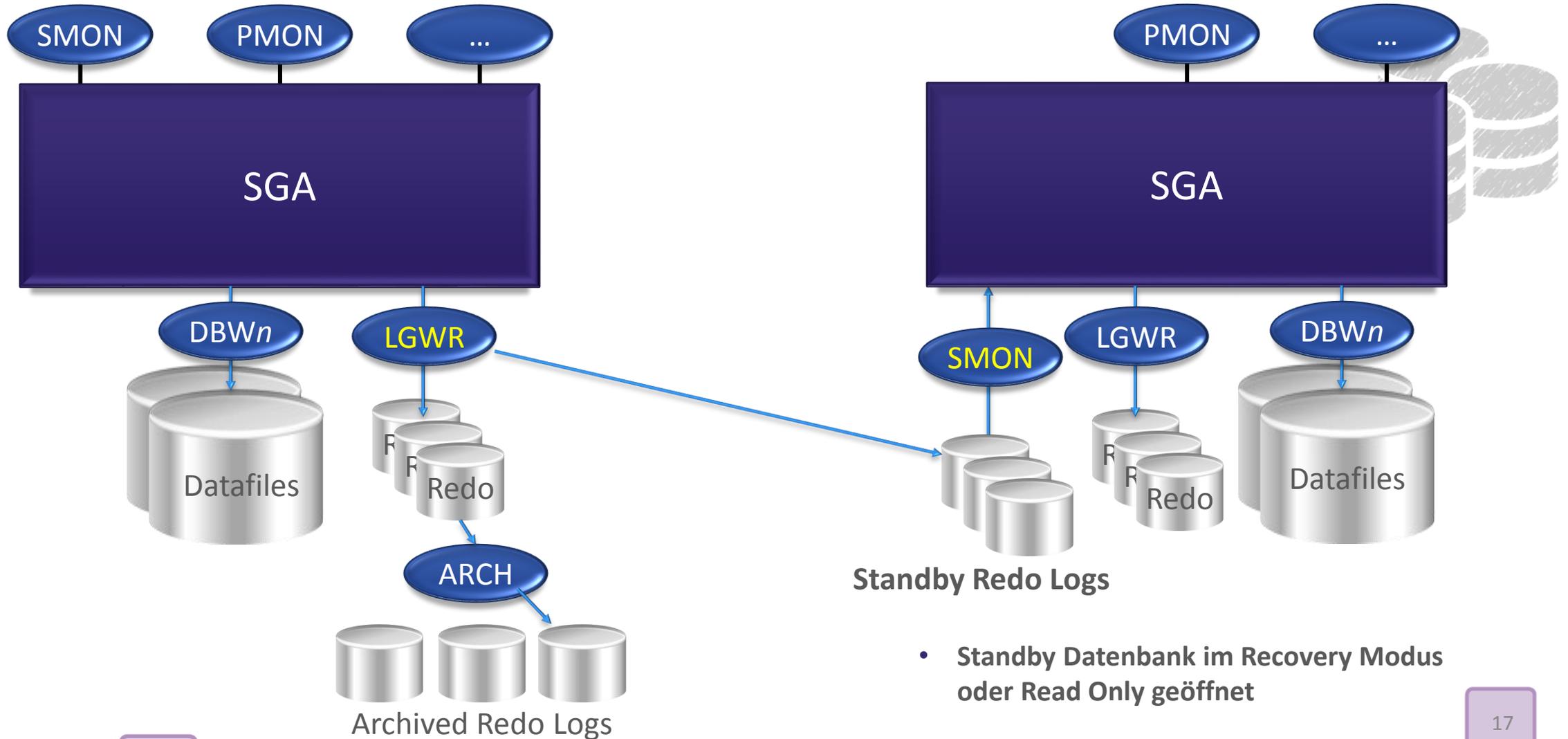


Redo Apply - Physical Standby



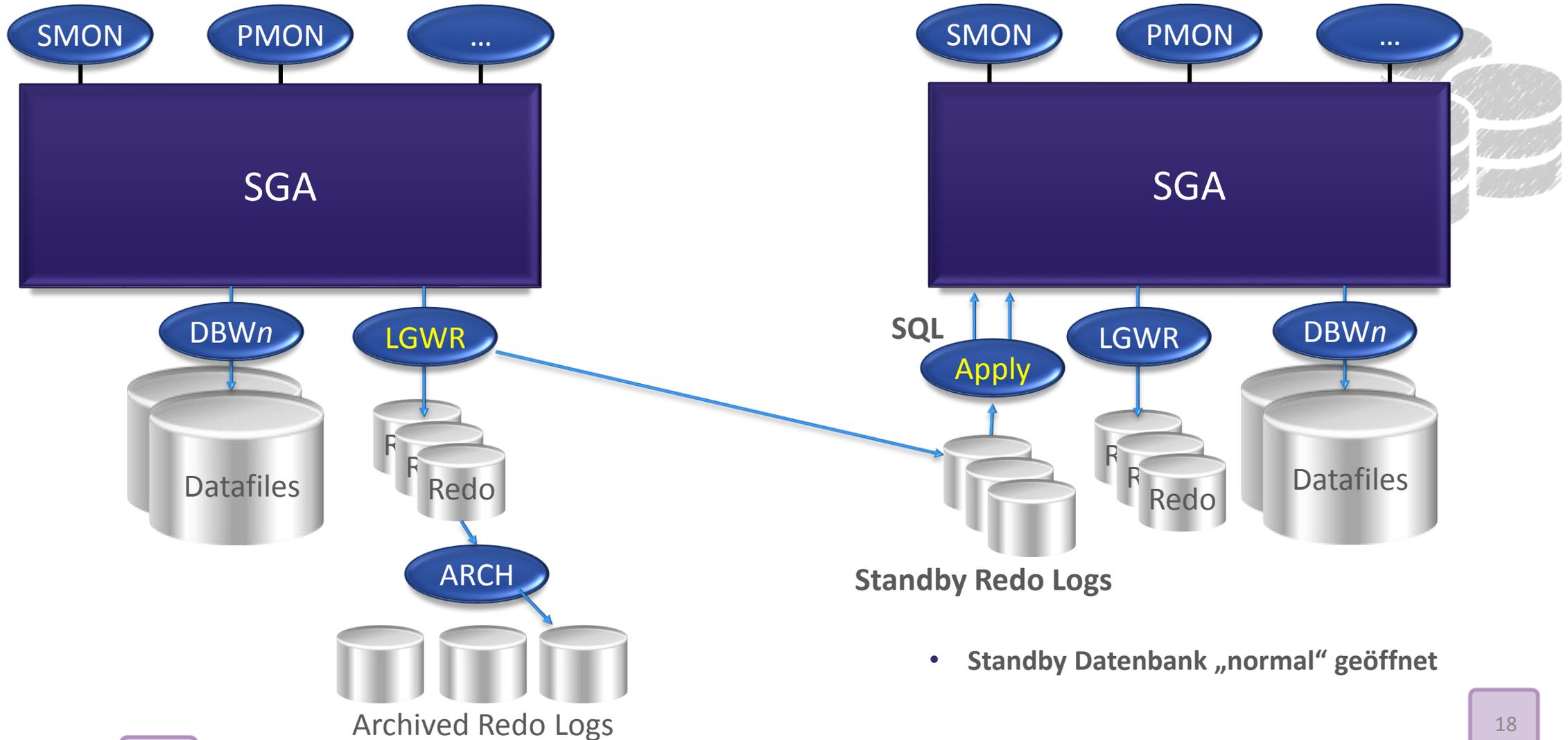
- Standby Datenbank im Recovery Modus
- Manuelle Standby Konfiguration
- Dbvisit Standby

Data Guard – Physical Standby



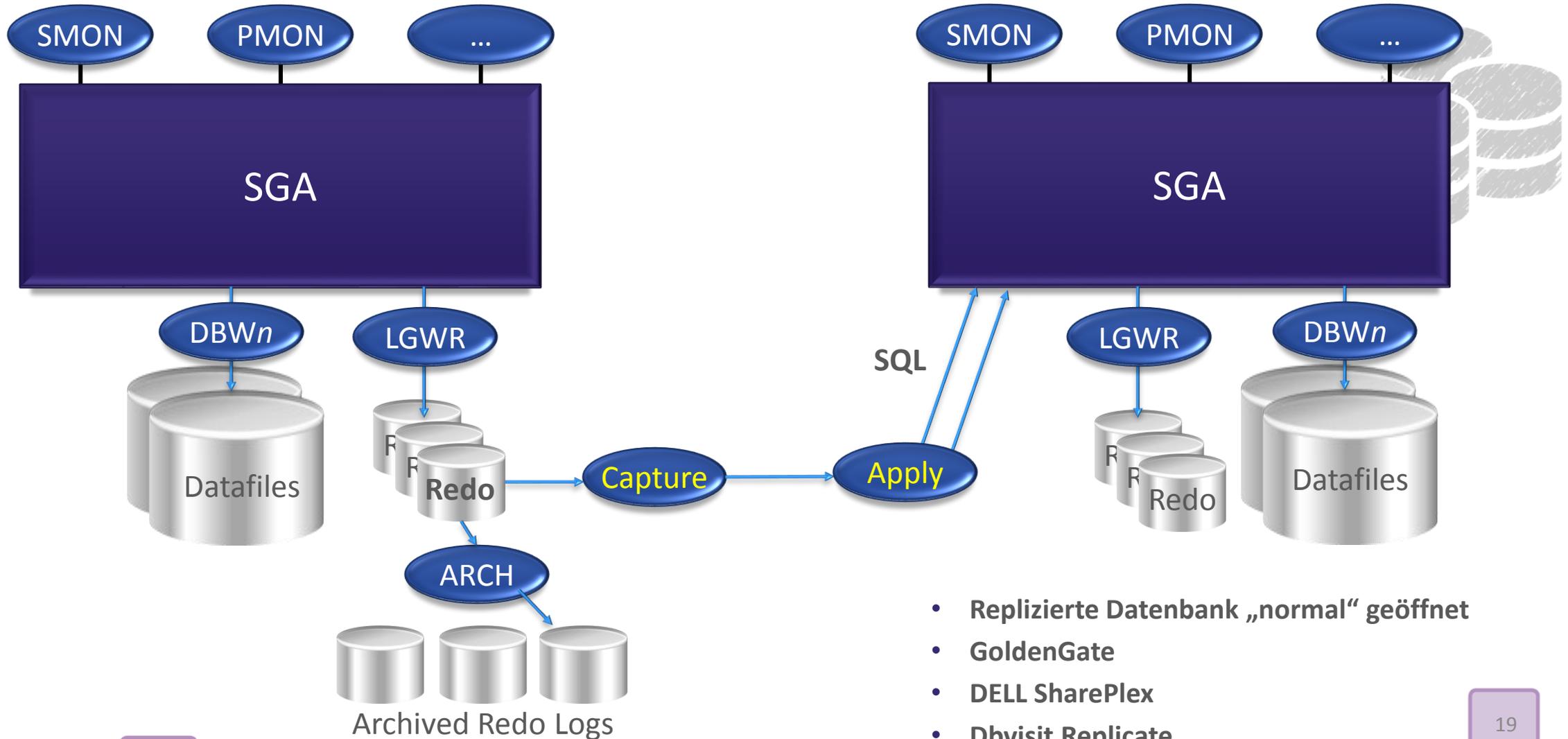
- Standby Datenbank im Recovery Modus oder Read Only geöffnet

Data Guard – Logical Standby



- Standby Datenbank „normal“ geöffnet

Logical Replication – SQL Apply



- Replizierte Datenbank „normal“ geöffnet
- GoldenGate
- DELL SharePlex
- Dbvisit Replicate

- **Unterschiede in der Nutzbarkeit des Replikats**
- **Standby = Redo Apply (physical)**
 - Aufbau ausschließlich für den Fehlerfall
 - Hochverfügbarkeit
 - Disaster Recovery
 - Fallback bei logischen Fehlern
- **Replikation = SQL Apply (logical)**
 - Volle Nutzbarkeit der Zieldatenbank möglich



Warum Replikation?

- **Data Offloading** → Aufbau einer Kopie für (hauptsächlich) lesende Zugriffe
- **Daten Konsolidierung** → Übernahme von Daten aus mehreren untergeordneten Lokationen
- **Daten Verteilung** → Aufbau von lokalen Kopien (oft nur Ausschnitte)
- **Lastverteilung** → Verteilung der Last auf mehrere Datenbanken
- **Migrationsen** → Temporärer Einsatz von Replikationen für die Minimierung der Downtime bei Migrationen
- **Change Data Capture** → Protokollierung von Datenänderungen
- **Hochverfügbarkeit** → 1:1 Kopie der Datenbank / des Schemas für Disaster Recovery



- **Enterprise Edition**

- mit Active Data Guard und Logical Standby gleich mehrere Lösungen für eine verlustfreie Replikation bei gleichzeitiger Nutzung des Replikats



- **Standard Edition**

- ohne Data Guard → Herausforderung ungleich höher
- Redo Log-Shipping basierende (Drittanbieter-) Lösungen
 - Vorteil: einfach zu realisierende HA-Umgebung
 - Nachteil: Datenverlust von praktisch 15 Minuten muss in Kauf genommen werden während das Replikat im „normalen“ Betrieb nicht genutzt werden kann

- ***Wieviel Datenverlust kann toleriert werden?***

- **Standby-Lösung**

- Ideal:
 - geringe Anzahl von Transaktionen
 - Redo Log-Daten alle 5 Minuten übertragen
- Vorsicht:
 - Batchläufe und andere größere Aktionen → Engpass
- Einschränkung:
 - bei maximalem Datenverlust unter 15 Minuten → physikalische Replikation nicht zu empfehlen



- **Logische Replikation**

- Stärke:
 - Datenverlust bei einem Failover minimieren

- **Kein Recovery**
 - im Gegensatz zur Standby-Lösung kein Recovery der archivierten Redo Logs
 - **Realtime-Apply** - DML- und DDL-Befehle werden direkt aus den Redo Log-Daten wiederhergestellt und auf der Standby ausgeführt
- **Wie sieht der minimale Datenverlust nun aus?**
 - beschränkt sich auf Transaktionsdaten, die bis zum Failover noch nicht von der Primärseite zur Replikationsseite übertragen wurden
 - Alles was in den Redo Log-Files gelandet ist und appliziert wurde ist auch auf der replizierten Datenbank



- **Konzeptionelle Unterschiede**

- **Optimistic Commit:** Datenübertragung vor dem Commit
 - *Oracle Streams, SharePlex, Dbvisit Replicate*
- **Pessimistic Commit:** Datenübertragung mit dem Commit
 - *GoldenGate*



- **Pessimistic Commit im „Fehlerfall“**

- ein einzelnes „Commit“ könnte noch durchgeführt worden sein, während größere Transaktionen, die als Gesamtheit übertragen wurden, wahrscheinlich nicht erfolgreich sind



Projekt

- Fragen
- Herausforderungen
- Umsetzung
- Problem und Lösung



Was sollte man beachten?



- **Replikation ist ein Projekt**
 - „Mal eben“ Daten zu replizieren wird funktionieren, in der Regel allerdings nicht sehr lange
- **Zeitraum für das Projekt → mind. 6 Monate bis Inbetriebnahme**

Fragen für das Projekt „Replikation“ (1)

- **Was soll repliziert werden?**
 - Einzelne Tabellen
 - Komplette Schemata
 - DDL
- **Was für Objekte werden repliziert?**
 - Gibt es besondere Datentypen, z.B. XMLTYPE, ADT
- **Auf wie viele Ziele soll repliziert werden?**
 - Ein Ziel ist einfacher zu beherrschen als 10 Ziele!
- **Was ist, wenn eines der Ziele nicht verfügbar ist?**
 - Was passiert mit den archivierten Redo Logs?
- **Wer sorgt dafür, dass Quelle und Ziel synchron sind (DDL)?**
 - DDL zu replizieren hat Vor- und Nachteile
 - Vorteil: Datenbanken sind identisch
 - Nachteil: Datenbanken sind identisch



Fragen für das Projekt „Replikation“ (2)

- **Wie setzt man die Replikation auf (laufender Betrieb)?**
 - Was passiert mit offenen Transaktionen?
 - Muss die Datenbank eventuell gesperrt werden?
- **Was passiert, wenn die Replikation nicht funktioniert?**
 - Was passiert mit der Quelldatenbank?
 - Läuft eventuell die Fast Recovery Area voll?
- **Wie wird neu synchronisiert?**
 - Wie lange dauert die Synchronisierung?
 - Können fehlerhafte Sätze „repariert“ werden?
- **Was passiert bei Konflikten?**
 - Wer hat Recht? Am besten: Überschreiben!
- **Was muss überwacht werden?**
 - Wie groß ist die Latenzzeit?



Fragen für das Projekt „Replikation“ (3)



- **Was ist mit Triggern?**
 - Können diese ausgeschaltet werden?
 - Erkennt die Replikationslösung Trigger?
- **Was ist mit Constraints?**
 - Werden „on delete cascade“ Constraints aufgelöst?
 - Muss eventuell ein Deferred Constraint Checking eingeschaltet werden?
- **Was ist mit Massenänderungen (Bulk Loads, Batch Jobs)?**
 - Wie schnell ist die Replikation?
 - Können Bulk Loads erkannt und entsprechend repliziert werden?
- **Was ist mit BFILES?**
 - Die werden nicht repliziert!!!
- **Wer sorgt dafür, dass auf der Zielseite nicht geändert wird bzw. wie ist die Strategie?**
 - Optimaler Weise wird nur auf einer Seite geändert!



Diese Fragen stellen sich in der Regel für Standby Lösungen nicht!

- **Umbau der HA-Umgebung**
 - aus Enterprise Edition mit Data Guard soll eine Standard Edition One mit leistungsfähiger Hardware und Hochverfügbarkeitslösung werden
- **Aufgabenstellungen**
 - Migration DBMS von Enterprise nach Standard Edition One
 - Migration Standby Technologie von integriertem Tool nach Drittanbieter Tool
 - Migration in neue Hardware-Umgebung
 - Migration in neue Backup-Umgebung
- **Anforderung: Ressourcenverteilung bei geringstmöglichem Datenverlust**
 - Wahl: Standard Edition One mit Dbvisit Replicate auf 2 Windows Servern



- **Replizierte Datenbank = eigenständige Datenbank**
 - Monitoring
 - Standby → i.d.R. einfaches Monitoring ausreichend
 - Separate Verwaltung
 - und bei produktiver Nutzung auch Sicherung
 - Eigenständiges SYS und SYSTEM Schema
 - Änderungen werden nicht erfasst und müssen bei Bedarf nachgezogen werden
 - wirkt sich auch auf ROWIDs aus
 - Spezielle Datentypen können von verschiedenen Lösungen nicht „logisch“ Repliziert werden bspw. XMLTYPE, ADT
 - BFILES werden nicht repliziert
 - Trigger „feuern“ unter Umständen zweimal
 - Sequenzen garantieren keine fortlaufende Nummerierung



- **SEQUENCE**

- ununterbrochen fortlaufende Zahlenfolge kann nicht garantiert werden
 - gilt auch ohne Replikation
 - Stichwort: Rollback
- Nummerierung: viele Anwendungen setzen trotzdem auf Sequences
 - bspw. für die Nummerierung von Rechnungen (Primary-Key)
 - garantiert eindeutig
 - aber nicht garantiert fortlaufend
- Bei einem Failover kann es hier zu Abweichungen kommen



- **CONSTRAINT**

- Konflikte können ein SQL Apply auf der Replikationsseite verhindern
- Problem: die Replikationsseite könnte von Dritter Seite verändert werden
- Zugriffsschutz
 - Konfliktvermeidung über einen entsprechenden Zugriffsschutz gewährleisten



- **Switchover und Failover**

- **BACKUP**

- Hochverfügbarkeit schützt nicht vor menschlichem Versagen
- Belastbare Backup-Strategie entwickeln

- **Kopie der Produktionsdatenbank**

- Komplette Einrichtung der neuen Umgebung auf zukünftiger Hardware
- Konfiguration und Test
- Durchspielen aller möglichen Szenarien – temporärer Ausfall, Switchover, Failover



- **Bereitstellen eines Datenbank-Klons**

- Bevor die Replikation beginnen kann muss sichergestellt werden, dass auf beiden Seiten
 - gleiche Datenbank-Struktur und
 - gleiche Datenbank-Inhalte bereitstehen.

- **1:1 DBMS Installation**

- Erstellen und Mounten einer Instanz auf der Standby-Seite



```
CMD> rman TARGET sys/password@PROD-QUELLE AUXILIARY sys/password@PROD-ZIEL
```

```
RMAN> DUPLICATE TARGET DATABASE TO ,prod` FROM ACTIVE DATABASE;
```

- **RMAN Duplicate**

- Relativ schnell und simpel umsetzbar
- Konfliktvermeidung
 - gewachsene Produktiv-Datenbank
 - mögliche Konflikte durch manuelles Aufsetzen der Datenbank und deren Strukturen
 - FROM ACTIVE DATABASE → Duplicate erfolgt nicht aus Backup heraus

- **Dbvisit Replicate**

- Installation auf beiden Seiten

- **Konfiguration**

- Festlegung der Schemas und/oder Tabellen granular möglich
- Neben der „einfachen“ Replikation eine 2-way, bi-directional replication möglich
 - absolute Konfliktvermeidung: 1-way replication
- Fehlerbenachrichtigung, Notification über Mail und Konsole
- Verwaltung und Logging über eigenes Schema „Dbvrep“ in der Datenbank
- Einrichtung eines MINE- (Quell-Seite) und eines APPLY-Prozesses (Ziel-Seite)



- Konfiguration in 4 Schritten
 - Quelle und Ziel konfigurieren



```
Dbvisit Replicate command console
D:\oracle\product\11.2.0\dbhome_1\network\admin
(TNS_ADMIN) - Please enter TNS configuration directory for this machine:
[D:\oracle\product\11.2.0\dbhome_1\network\admin]

Step 1 - Describe databases
=====
The first step is to describe databases used in the replication. There are
usually two of them (source and target).
Store SYSDBA and DBA passwords? Passwords only required during setup and
initialization? (yes/no) [yes]
Let's configure the database, describing it's type, connectivity, user names
etc.

What type of database is this? (Oracle): [Oracle]

Please enter database TNS alias: [] proda

Please enter SYSDBA user name: [SYS]

Please enter password for this user: [change_on_install] manager

Please enter user with DBA role: [SYSTEM]

Please enter password for this user: [manager]
```

- Konfiguration in 4 Schritten
 - Replikationsart festlegen



```
Dbvisit Replicate command console
Following databases are now configured:
1: Oracle proda, SYS/****, SYSTEM/****, dburep/****, USERS/TEMP, dburep/, ASM:NO,
TZ: +01:00
2: Oracle prodb, SYS/****, SYSTEM/****, dburep/****, USERS/TEMP, dburep/, ASM:NO,
TZ: +01:00

Enter the number of the database to modify it, or "add", or "done": [done]

Step 2 - Replication pairs
=====
The second step is to set source and targets for each replication pair.
Let's configure the replication pair, selecting source and target.
Following databases are described:
1: proda (Oracle)
2: prodb (Oracle)

Select target database: [2]

Will be DDL replication enabled? [yes]

Use fetcher to offload the mining to a different server? (yes/no) [no]

Would you like to encrypt the data across the network (yes or no) [no]

Would you like to compress the data across the network (yes or no) [no]
```

- Konfiguration in 4 Schritten
 - Replikationsquellen definieren



```
Dbvisit Replicate command console
load      : All replicated data is created and loaded automatically
none
(dp_networklink/dp_exp/exp/ddl_file/ddl_run/load/none) [dp_networklink]

Following replication pairs are now configured:
1: proda (Oracle) ==> prodb (Oracle), DDL: yes, fetcher: no, process suffix:
(no suffix), compression: no, encryption: no, network timeout: 60, prepare
type: single-sc, data load: dp_networklink

Enter number of replication pair to modify it, or "add", or "done": [done]

Step 3 - Replicated tables
=====
The third step is to choose the schemas and tables to be replicated. If the
databases are reachable, the tables are checked for existence, datatype
support, etc., schemas are queried for tables. Note that all messages are
merely hints/warnings and may be ignored if issues are rectified before the
scripts are actually executed.

Following tables are defined for replication pairs:
1: proda (Oracle) ==> prodb (Oracle), DDL: yes, suffix: (no suffix), prepare:
single-sc
   No tables defined.

Enter number of replication pair to modify it, or "done": [1]
```

- Konfiguration in 4 Schritten
 - Prozesse / Services konfigurieren



```
Dbvisit Replicate command console
2. Specify filtering conditions.Specify rename name or filter condition for any
of the specified schemas
(YES/NO): [no]

Following tables are defined for replication pairs:
1: proda <Oracle> ==> prodb <Oracle>, DDL: yes, suffix: <no suffix>, prepare:
single-scn
   BASIS<tables>, DEMO<tables>, BASIS<PLSQL>, DEMO<PLSQL>

Enter number of replication pair to modify it, or "done": [done]

Step 4 - Process configuration
=====
The fourth step is to configure the replication processes for each replication.

Following processes are defined:
1: MINE on proda
   Not configured.
2: APPLY on prodb
   Not configured.

Enter number of process to modify it, or "done": [1]
Fully qualified name of the server for the process (usually co-located with the
database, unless mine is offloaded using fetcher): [PROD-A]
```

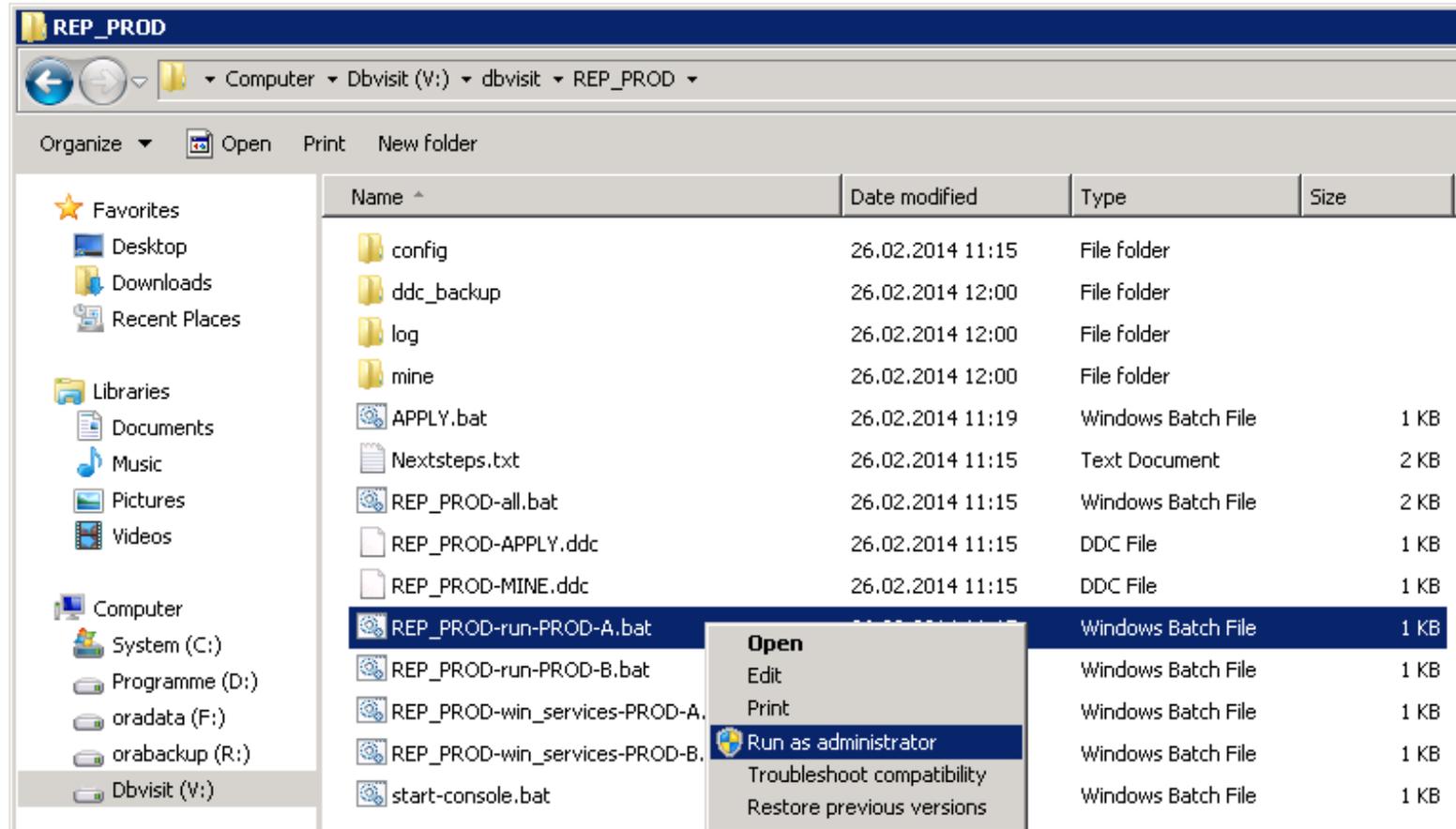
- Konfiguration in 4 Schritten
 - Fertig. Skript ausführen.



```
Dbvisit Replicate command console
Created file U:\dbvisit\REP_PROD\REP_PROD-APPLY.ddc.
Created file U:\dbvisit\REP_PROD\REP_PROD-MINE.ddc.
Created file U:\dbvisit\REP_PROD\config\REP_PROD-setup.dborep.
Created file U:\dbvisit\REP_PROD\config\REP_PROD-dbsetup_proda.sql.
Created file U:\dbvisit\REP_PROD\config\REP_PROD-dbsetup_prodb.sql.
Created file U:\dbvisit\REP_PROD\config\REP_PROD-grants_proda.sql.
Created file U:\dbvisit\REP_PROD\config\REP_PROD-grants_prodb.sql.
Created file U:\dbvisit\REP_PROD\config\REP_PROD-onetime.ddc.
Created file U:\dbvisit\REP_PROD\start-console.bat.
Created file U:\dbvisit\REP_PROD\REP_PROD-run-PROD-A.bat.
Created file U:\dbvisit\REP_PROD\REP_PROD-win_services-PROD-A.bat.
Created file U:\dbvisit\REP_PROD\REP_PROD-run-PROD-B.bat.
Created file U:\dbvisit\REP_PROD\REP_PROD-win_services-PROD-B.bat.
Created file U:\dbvisit\REP_PROD\Nextsteps.txt.
Created file U:\dbvisit\REP_PROD\REP_PROD-all.bat.
=====
Dbvisit Replicate wizard completed

Script U:\dbvisit\REP_PROD\REP_PROD-all.bat created. This runs all the above
created scripts. Please exit out of dborep, review and run script as current
user to setup and start Dbvisit Replicate.
=====
dborep>
```

- Replikation starten



The screenshot shows a Windows Explorer window titled "REP_PROD" with the address bar showing the path "Computer > Dbvisit (V:) > dbvisit > REP_PROD". The left sidebar shows the "Favorites" and "Libraries" sections. The main pane displays a list of files and folders:

Name	Date modified	Type	Size
config	26.02.2014 11:15	File folder	
ddc_backup	26.02.2014 12:00	File folder	
log	26.02.2014 12:00	File folder	
mine	26.02.2014 12:00	File folder	
APPLY.bat	26.02.2014 11:19	Windows Batch File	1 KB
Nextsteps.txt	26.02.2014 11:15	Text Document	2 KB
REP_PROD-all.bat	26.02.2014 11:15	Windows Batch File	2 KB
REP_PROD-APPLY.ddc	26.02.2014 11:15	DDC File	1 KB
REP_PROD-MINE.ddc	26.02.2014 11:15	DDC File	1 KB
REP_PROD-run-PROD-A.bat	26.02.2014 11:15	Windows Batch File	1 KB
REP_PROD-run-PROD-B.bat	26.02.2014 11:15	Windows Batch File	1 KB
REP_PROD-win_services-PROD-A...	26.02.2014 11:15	Windows Batch File	1 KB
REP_PROD-win_services-PROD-B...	26.02.2014 11:15	Windows Batch File	1 KB
start-console.bat	26.02.2014 11:15	Windows Batch File	1 KB

A context menu is open over the file "REP_PROD-run-PROD-A.bat", showing the following options:

- Open
- Edit
- Print
- Run as administrator
- Troubleshoot compatibility
- Restore previous versions

- Command Console
 - MINE + APPLY is running



```
C:\ Dbvisit Replicate command console
\ Dbvisit Replicate 2.6.00.3735(XTD edition) - Evaluation License expires in 30
days
MINE IS running. Currently at plog 75 and SCN 864430 (02/26/2014 12:21:35).
APPLY IS running. Currently at plog 75 and SCN 864414 (02/26/2014 12:21:31).
Progress of replication REP_PROD:MINE->APPLY: total/this execution (stale)
-----
0 tables listed.
dburep> _
```

- Command Console
 - MINE + APPLY is running



```
C:\> Dbvisit Replicate command console

! Dbvisit Replicate 2.6.00.3735(XTD edition) - Evaluation License expires in 24
days
MINE IS running. Currently at plog 414 and SCN 2047774 (03/02/2014 22:07:00).
APPLY IS running. Currently at plog 197 and SCN 1752389 (02/28/2014 16:14:23).
Progress of replication REP_PROD:MINE->APPLY: total/this execution
-----
DEMO.ADRESSEN:          100%  Mine:4000/2000      Unrecov:0/0      App
  lied:4000/2000  Conflicts:0/0      Last:04/03/2014 10:22:10/OK
DEMO.AUFTRAEGE:        100%  Mine:20000/10000   Unrecov:0/0      App
  lied:20000/10000 Conflicts:0/0      Last:04/03/2014 10:22:23/OK
DEMO.PERSONEN:         100%  Mine:2000/1000     Unrecov:0/0      App
  lied:2000/1000  Conflicts:0/0      Last:04/03/2014 10:22:10/OK
DEMO.POSITIONEN:       55%   Mine:250000/150000 Unrecov:0/0      App
  lied:138912/38912 Conflicts:14370/1781 Last:04/03/2014 10:29:20/OK
DEMO.TELEFONE:         100%  Mine:8000/4000     Unrecov:0/0      App
  lied:8000/4000  Conflicts:0/0      Last:04/03/2014 10:22:10/OK
-----
5 tables listed.
```

- Command Console
 - MINE is reachable, but NOT mining



```
Dbvisit Replicate command console
! Dbvisit Replicate 2.6.00.3735(XTD edition) - Evaluation License expires in 28
days
MINE is reachable, but NOT mining.
APPLY is reachable, but NOT applying, initialization NOT yet complete.
Progress of replication REP_PROD:MINE->APPLY: total/this execution
-----
DEMO.ADRESSEN:          100%  Mine:2000/2000      Unrecov:0/0      App
  lied:2000/2000  Conflicts:0/0      Last:28/02/2014  11:59:47/OK
DEMO.AUFTRAEGE:        100%  Mine:10000/10000   Unrecov:0/0      App
  lied:10000/10000 Conflicts:0/0      Last:28/02/2014  12:00:08/OK
DEMO.PERSONEN:         100%  Mine:1000/1000     Unrecov:0/0      App
  lied:1000/1000  Conflicts:0/0      Last:28/02/2014  11:59:47/OK
DEMO.POSITIONEN:      100%  Mine:100000/100000 Unrecov:0/0      App
  lied:100000/100000 Conflicts:12589/12589 Last:28/02/2014  12:09:22/OK
DEMO.TELEFONE:        100%  Mine:4000/4000     Unrecov:0/0      App
  lied:4000/4000  Conflicts:0/0      Last:28/02/2014  11:59:47/OK
-----
5 tables listed.

dburep> shutdown all
Dbvisit Replicate APPLY process shutting down.
Dbvisit Replicate MINE process shutting down.

dburep> _
```

- **Replikation beginnen**

- Nach dem starten der vorkonfigurierten Services läuft die Replikation bereits



- **MINE**

- Dbvisit generiert aus den Redo Logs „PLOGs“ auf der Quell-Seite
- Redo Logs → generierte PLOGs

- **APPLY**

- PLOGs werden über das Netzwerk auf die Ziel-Seite verschoben
- PLOGs → appliziert (SQL Apply)

- Dbvisit Replicate does not use Logminer, Oracle triggers or database queues for replication. Dbvisit Replicate uses its own internal log mining technology to capture the changes as they are written to the redo logs (or archived redo logs). As the changes appear in the redo log, they are written to an external file called a PLOG and transferred in real time to the target server to be applied to the target database.

(Dbvisit documentation)



- **Replikations-Lösung**

- Unterschiede zwischen verschiedenen Lösungen
- unterstützte Datentypen, commit-Ansatz, Ziel-Datenbanken, Plattformen, Features



- **Dbvisit Replicate**

- Eigenständige Datenbank = Monitoring
- SYS und SYSTEM werden nicht repliziert
- Nicht jedes DDL Kommando wird unterstützt
 - bspw. das anlegen eines Tablespaces erfolgt nicht automatisch
 - Trigger, Constraints, Sequences, ALTER DATABASE und ALTER SYSTEM Kommandos
 - müssen in einem eigenen Prozess nachgezogen werden

- **DB Compare**

- Unterschiede mithilfe von TOAD und einem DB Compare aufdecken und je nach Bedarf auf die andere Seite anwenden

- **Beispiel: Trigger feuert zweimal**
 - In früher Version von Replicate wurden Trigger-Kommandos noch repliziert
- **Folge**
 - ein durch einen Trigger ausgelöstes DML-Kommando auf der Primärseite
 - einmal durch die Replikation eingetragen
 - ein zweites Mal durch die Replikation auf der Ziel-Seite ausgeführt
- **Replikation steht**
 - Konflikt durch Constraint-Verletzung → gleicher Eintrag zweimal
 - standardmäßig „retry“ bis zur (manuellen) Auflösung des Konflikt



- **Conflict Handler**

- **Ignorieren** = Statement wird zurückgerollt
- **Überschreiben** = Zieldaten werden unter Missachtung der WHERE-Klausel überschrieben



- **Conflict Handling**

- Test-Umgebung: tolerabel
- Produktiv-Umgebung: Konflikte ausschließen
 - Keine Methode gibt Gewissheit über Datenkonformität

- **Lösung des Trigger-Problems**

- Da wir vollen Zugriff auf das Replikat haben: Trigger ausschalten

```
SQL> ALTER TABLE <table_name> DISABLE ALL TRIGGERS;
```

- **Vorgehen bei Produktions-Ausfall**
 - Hauptzweck unser HA-Umgebung der geringstmögliche Datenverlust
 - durch die Replikation realisiert
 - Replikat für eventuelle Produktion nutzbar machen
- **Umschalten, aber wie?**
 - Datenbankservices einrichten
 - Zugriff auf die Datenbanken steuern können
 - DMBS_SERVICE Package



```
SQL> exec dbms_service.create_service(service_name=>'produser', network_name=>'produser');
```

- **DBMS_SERVICE**
 - Zugriff Clients und Applikationen wird über diese Services realisiert
 - kann jederzeit gestoppt und gestartet werden
- **Produktionsausfall**
 - Replikationsseite freigeben, wenn produktionsrelevanten Daten bereit stehen



```
SQL> exec dbms_service.start_service('produser');
```

- **SEQUENCE**

- nicht repliziert
 - müssen an den aktuell höchsten Wert angepasst werden
 - zur Sicherheit empfiehlt es sich diesen Wert noch zu erhöhen



- **Problem**

- Forderung nach einer „unbedingt“ fortlaufenden eindeutigen Nummer für bspw. Buchhaltung und Rechnungsnummern

- **Lösung**

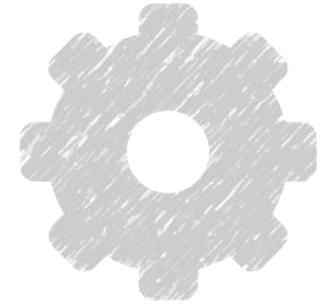
- Auftrag an den Entwickler:
Bitte eine (andere) Lösung finden, um eine fortlaufende Nummer zu garantieren.



- Die Replikation läuft sauber durch

... am Anfang

- mit laufender Produktion und über einen längeren Zeitraum treten weitere Probleme zutage ...
 - kleinere Bugs unter Windows
 - ... und ein größeres: CLOBs wurden nicht sauber der repliziert.
 - **Schritt für Schritt abarbeiten**
-
- **Replikation steht**
 - Replikation gerät durch Konflikte in Verzug
 - Teile müssen aus der Replikation genommen werden
-
- **Ziel verfehlt** ... solange die Probleme nicht gelöst sind.

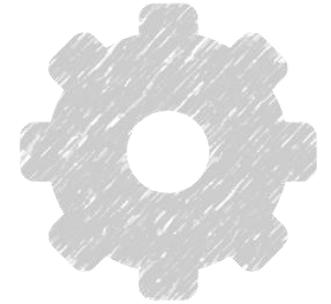


- **Aufbau Standby Datenbank**

- Paralleler Aufbau einer Standby Umgebung
- Performance? Speicher? ... kein Problem

- **Doppelstrategie**

- Verbindung aus Standby und logischer Replikation
- Standby garantiert eine strukturelle 1:1 Datenbank
 - Gesichertes Umschalten bei Ausfall oder Wartung
- Replikation sorgt für minimalen Datenverlust
 - fehlende Daten können bei Disaster aus replizierter Datenbank wiederhergestellt werden

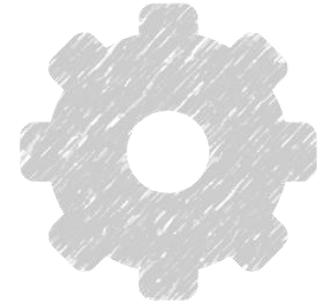


- **Graceful Switchover**
 - geplantes Umschalten

→ Umschalten auf die Standby Datenbank

- **Failover**
 - ungeplantes Umschalten
 - Voraussetzung: Verfügbarkeit der Online Redo Logs

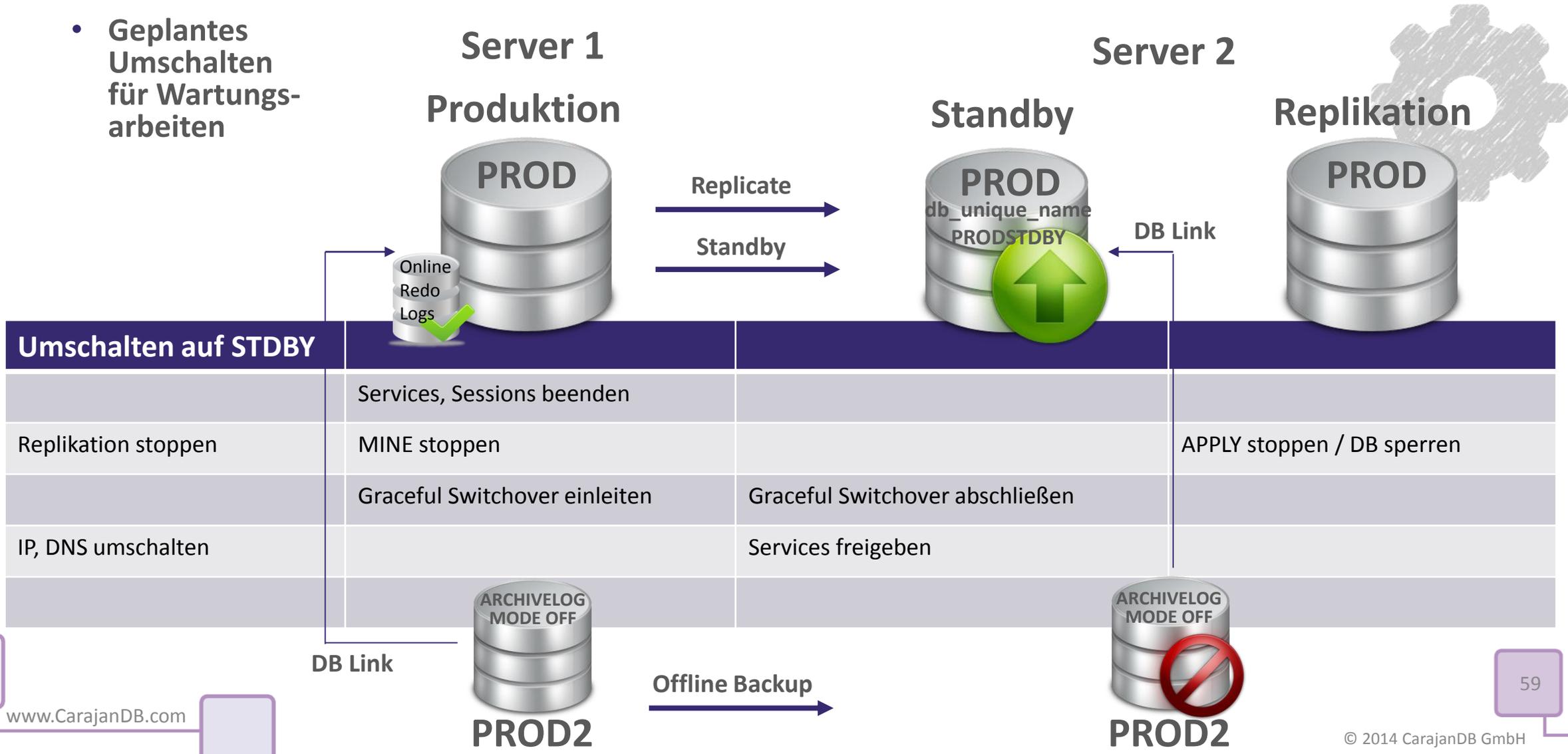
→ Umschalten auf die Standby Datenbank



Graceful Switchover

> geplantes Umschalten auf Standby

- Geplantes Umschalten für Wartungsarbeiten



Graceful Switchover

> geplantes Umschalten auf Standby

- Geplantes zurück-schalten

Server 1
Produktion



Replicate



Standby



Server 2

Standby



DB Link



Replikation



Zurückschalten

Sperren		Services, Sessions beenden	
Graceful Switchover	Graceful Switchover abschließen	Graceful Switchover einleiten	
Archive Logs	Erzeugte Archive Logs einfügen	Erzeugte Archive Logs kopieren	
Replikation	Replikation starten		Replikation starten
IP, DNS umschalten	Services frei		

DB Link



PROD2

Offline Backup



PROD2

Failover

> ungeplantes Umschalten auf Standby

- Ungeplanter Ausfall Produktionsserver
- Online Redo Logs auf SAN/NAS verfügbar

Server 1
Produktion



Replicate



Standby



Server 2

Standby



DB Link

Replikation



Umschalten auf STDBY

Check: Redo Logs verfügbar?

Online Redo Logs verfügbar!

Replikation stoppen

APPLY stoppen / DB sperren

Online Redo Logs

Online Redo Logs einspielen

Umschalten auf Standby

ACTIVATE STANDBY DATABASE

IP, DNS umschalten

Services freigeben

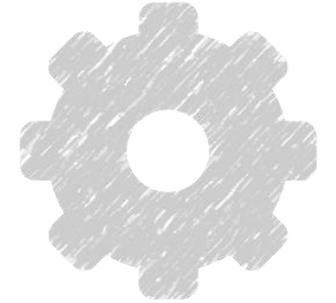
DB Link



Offline Backup



- **Worst Case Failover**
 - kompletter Serverausfall, Zugriff auf Online Redo Logs nicht mehr möglich
- Umschalten auf die Replikationsdatenbank



Worst Case Failover

> ungeplantes Umschalten auf Replikation

- Totalausfall Produktions-server + SAN/NAS

Server 1
Produktion



Replicate
Standby

Server 2

Standby



DB Link

Replikation



Umschalten auf REP

Check: Redo Logs verfügbar?	Online Redo Logs <u>nicht</u> verfügbar!		
Replikation stoppen			APPLY stoppen / DB sperren
DB Compare		DB Compare mit Standby	Check TRIGGER, GRANTS, SEQUENCES, XML-Types, Utlfile
IP, DNS umschalten			ARCHIVELOGs freigeben

DB Link



Offline Backup



TEST-Server als zwischen-Standby > nach Totalausfall eines Servers

- Standby Richtung Test-Server

Server 3
Zwischen-Standby



Server 2

Standby



Produktion
Replikation



Standby

DB Link

	Zwischen-Standby einrichten		Online Redo Logs verfügbar halten
			Replikation bleibt ausgesetzt
		Sicherung PROD2?	

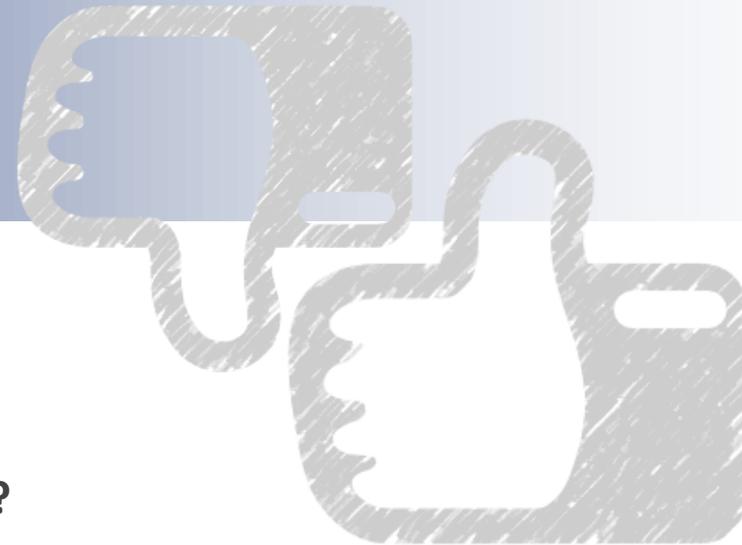


PROD2



Fazit

- Logisch ... ?
- ... oder doch nicht?

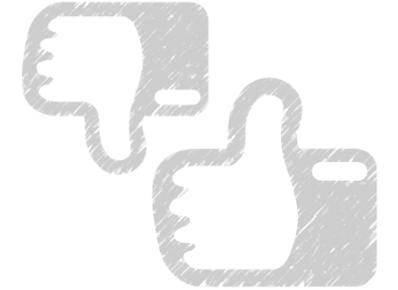


- **Standby Datenbank**

- Enge Integration in die Oracle Datenbank Architektur (DataGuard – Standby Redo Logs)
- Statisch
- Sehr robust (wenn es richtig aufgesetzt wird!)

- **Replikation**

- Eine Softwarelösung
- Muss projiziert werden
- Sehr flexibel
- Sensibel (Änderungen auf der Zielseite)



... oder doch nicht?

- **Logische Replikation setzt Grenzen**

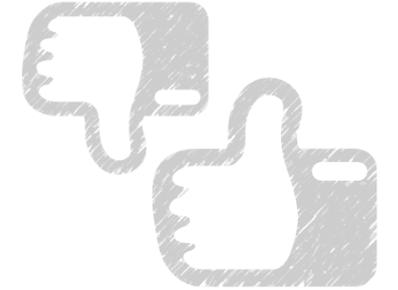
- Eine Reihe von Checks und Abläufen müssen geschaffen werden ...

- Monitoring
- Datenverwaltung
- Backup
- Conflicthandling
- Switchover
- Failover

- Je komplexer die Datenstruktur, desto weniger empfiehlt es sich allein auf eine logische Replikation zu setzen

- Bedienungsfreundlichkeit und Wartungsarmut kann eine logische Replikation nicht gewährleisten

- Vorher: Testen, Testen, Testen



Vielen Dank!



- Zeit für Ihre Fragen.