

# Brückentechnologie – Minimal Downtime Plattform-Migration/Upgrade 9.2 ⇒ 11.2

Christian Ballweg  
OPITZ CONSULTING Essen GmbH  
Essen

## Schlüsselworte:

Minimal Downtime Platform Migration 9i 11g, DataGuard, GoldenGate, Shareplex, Fallback

## Einleitung

Es gibt viele Gründe für eine Datenbankmigration. Diesen werden jedoch von den unterschiedlichen Beteiligten nicht immer die gleiche Bedeutung und Dringlichkeit beigemessen. Die Akzeptanz kann daher höchst unterschiedlich sein. Sobald die Dringlichkeit jedoch hoch genug ist, geht an einer Migration kein Weg vorbei.

Hier findet sich häufig die erste Herausforderung. Beteiligte sind nicht allein die technischen Abteilungen, sondern von den Nutzern der Datenbanken über die Entwickler eigener Applikationen, externen Dienstleistern bis hin zur Geschäftsführung alle Nutzer der Geschäftsprozesse, welche die Datenbank verwenden und letztlich Entscheider, welche die für eine Migration benötigten Mittel bereitstellen müssen.

Es folgt nun die Gegenüberstellung der Risiken einer Migration mit den Risiken auf sie zu verzichten und die Ermittlung von Anforderungen und Möglichkeiten der Migrationsdurchführung.

Eine Bewertung der ermittelten Optionen sollte im Abschluss immer die Auswirkung eines möglichen Scheiterns beinhalten. Diesem Scheitern kann durch Planung von Fallback-Varianten ein Teil des Schreckens genommen werden. Der Rückschritt im Katastrophenfall (freundlich: K-Fall) ist dann zwar mit Kosten, nicht jedoch mit Informationsverlust verbunden, was unter dem Stichwort *Business Continuity* den Fortbestand des Dienstes, manchmal sogar der ganzen Unternehmung sichert.

Die Absicherung des Fallbacks im K-Fall war die grundlegende Motivation für die „Minimal Downtime“ bei der Migration im vorgestellten Projekt. Da die Methodik einer verlustfreien Umschaltung nach der Migration ebenfalls während der Migration genutzt werden konnte, war es so gleichzeitig möglich die Downtime für den Dienst während der Migrationsarbeiten zu begrenzen.

## Projektbeschreibung

Das Projekt „Migration der ELSTER Clearingstellen“ beim Rechenzentrum der Finanzverwaltung des Landes Nordrhein-Westfalen hatte die Migration zweier replizierter 9.2.0.8 Datenbanken mit jeweils ca. 1,5 TB Daten zum Ziel.

## ELSTER - Elektronische SteuerErklärung

ELSTER bietet Arbeitnehmern, Arbeitgebern, Unternehmern und anderen Verwaltungen die Möglichkeit, Daten elektronisch über das Internet an die Finanzverwaltung zu übermitteln und

auszutauschen. Es ermöglicht damit eine medienbruchfreie Verarbeitung, beginnend beim Steuerbürger.

Die sogenannten „Clearingstellen“ in Düsseldorf und München (siehe *Abbildung 1*) fungieren dabei für alle 16 Bundesländer als „Briefkasten“. Sie sind Annahmestelle (für Steuerbürger, Arbeitgeber, Behörden, Banken usw.) und sorgen anschließend für die Verteilung in die Bundesländer. Des Weiteren dienen sie als Auskunftssystem für länderübergreifende Suchen in ELSTER.

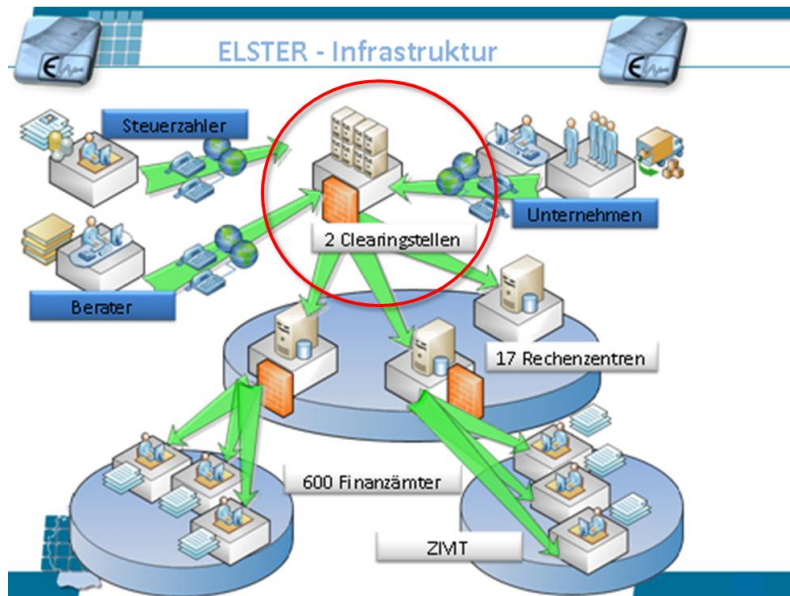


Abb. 1: Clearingstellen in ELSTER Infrastruktur

Die beiden Rechenzentren – eines in Düsseldorf, das andere in München – werden für ELSTER von den Ländern Nordrhein-Westfalen und Bayern gemeinsam betrieben.

Zur Lastverteilung und Ausfallsicherheit, z.B. bei Stromausfall oder geplanten Wartungsarbeiten in einem Rechenzentrum, sowie durch die große Entfernung auch im K-Fall, können Grunddienste von der anderen Clearingstelle übernommen werden.

### Technik

Zur Datenverteilung zwischen den Clearingstellen wird ORACLE Advanced Replication genutzt. Bei der *Multimaster Replication* werden die in der als *Master-Site* bezeichneten Datenbank gespeicherten Daten zur anderen Seite repliziert (*peer-to-peer*). Eine logische Trennung von Objekten der Datenbank lässt sich mittels Replikationsgruppen vornehmen. In einer Replikationsgruppe können Objekte eines oder mehrerer Schemata enthalten sein. Ein Replikationsobjekt darf jedoch nur einer einzigen Replikationsgruppe zugehören.

Die Datenbank, auf der die Definition und Administration der Replikationsgruppe erfolgt, wird *Master Definition Site* genannt und ist im vorliegenden Fall die Clearingstelle Düsseldorf.

Der Datentransfer findet je Replikationsgruppe statt und kann synchron oder asynchron erfolgen. Bei der synchronen Übertragung ist die Datenkonsistenz (durch *2PhaseCommit*) zwar immer gewährleistet, es müssen dafür jedoch auch alle *Master Sites* erreichbar sein. Die (hier verwendete, 5 minütliche) asynchrone Übertragung hat dagegen den Vorteil, dass lokale Transaktionen auch ohne die Erreichbarkeit der anderen *Master Site* erfolgen können. Dadurch mögliche Konflikte werden durch (vorgefertigte oder angepasste) Konfliktauflösungsroutinen gelöst (Beispiel: jüngste Transaktion gewinnt).

Datenänderungen werden auf der Quelle mittels interner Trigger erfasst und als verzögerte Transaktionen (*deferred transactions*) an das Ziel weitergeleitet. Die durch diese Trigger aufgerufenen entfernten Prozeduren (*remote procedure calls*) werden standardmäßig im Tablespace des Schemas SYSTEM zwischengespeichert und von dort aus mittels Datenbankjob auf die andere *Master Site* übertragen.

Hier wird erkennbar, dass dieser Replikationstechnik ein besonderes Augenmerk bei der Migration der Datenbanken der Clearingstellen zukommen muss. Außerdem ergab sich dadurch der Wunsch nach einer externen Lösung, die möglichst wenig in die Datenbankinterna eingreift.

Als Vertreter einer externen Replikationslösung sind die Produkte GoldenGate von ORACLE und Shareplex von QUEST evaluiert worden. Beide zeichnen sich dadurch aus, dass sie lediglich mittels COMMIT festgeschriebene Änderungen aus den Redologs (ggf. Archivelogs) auslesen, in proprietären Dateien puffern und so eine lose Kopplung zwischen der Quelle und dem Ziel erlauben. Der Vortrag beschäftigt sich mit der Einrichtung beider Produkte zum Zweck der Migration und Fallback-Absicherung. Des Weiteren wird die Durchführung unter Verwendung von Quest Shareplex beschrieben.

## Migration

### Migrationsgründe

Die Gründe für die Migration einer Datenbank sind vielfältig. Einige augenfällige seien hier genannt (hier zutreffendes Fett hervorgehoben):

- **Desupport der „alten“ Version**
- **neue Hardware / Plattform**
  - **bessere Hardware (hier: Storage)**
  - **Betriebssystem-Modernisierung (hier: Solaris Zones)**
  - (strategischer) Plattformwechsel
  - Applikations- / RDBMS- / Server-Konsolidierung
- Anforderungen von Dritt-Anbietern (Support, Zertifikation)
- **bessere Performance / Skalierbarkeit**
- neue benötigte Features

Neben diesen Gründen sind Änderungen im Datenbankumfeld wie

- Portierung / Migration von Applikationen
- Konvertierungen, z.B. Partitionierung
- **Wachstum (Größe der Datenbank)**
- Änderungen im Workload (Anzahl der Anwender)
- neue Architektur bzgl. Verfügbarkeit
- ... usw.

ein Grund für Migrationen.

Dabei müssen die Business-Anforderungen

- schnelle Einführung
- niedrige Kosten
- keine (geringe) Risiken für den Betrieb

erfüllt werden, wobei es dabei Hemmnisse gibt wie

- „Never change a running system“-Attitude
- Sorge um die Stabilität der Applikation

- Schlechte Erfahrungen in der Vergangenheit
- Hoher Testaufwand respektive Unvorhersehbarkeit von Auswirkung und Risiko einer Änderung
- Qualitative und umfassende Test sind zu kostenintensiv
- ...

Alle diese Punkte haben ihrerseits Einfluss auf die Bewertung des Migrationsrisikos und damit die Priorität einer Fallbackabsicherung.

Herausforderung allein die neue Datenbankversion betreffend: aktuell gibt es knapp **350** sichtbare von mehr als 2500 verborgenen (teils undokumentierten) Initialisierungs-Parametern (**11gR2**), gegenüber **258** von 871 (**9iR2**) – also nun fast 100 sichtbare sowie über 1500 „hidden parameters“ mehr, welche alle wiederum mehrere mögliche Werte besitzen können!

Weitere Felder, die betrachtet werden können: DB-Storage (Tabelle/Index: initrans, freelists, ...), OS-Einstellungen (I/O-Optionen, Memory-/Prozessor-/Netzwerk-Parameter) und Storage Subsystem Spezifika (Filesysteme, Arraykonfiguration, LVM- und ASM-Parametrisierung).

### Planung

Angenommen, man hat sich aus einem der genannten Gründe (hier Supportende) für eine Migration entschieden, sollten nun die weiteren Schritte geplant werden.

- 1) Ermittlung und Prüfung von **Migrationsoptionen**
- 2) Wahl der **Migrationsmethode**
- 3) Testen der Auswirkungen: **Testplan!**
- 4) Evaluation möglicher **Risiken!**
- 5) Verbesserungen nötig & möglich? Modifikation und **Iteration der Schritte 1-4...**
- 6) Planung der produktiven Umstellung (betrifft verschiedene Fachbereiche): **Checkliste!**
- 7) **Fallback?**  
Wichtig:
  - Darf das Projekt „Migration“ nicht behindern!
  - Reicht das Budget?
  - Wie lange soll ein Fallback nach der Migration garantiert werden?
  - Sind Kriterien für den Erfolg der Migration festgelegt?

### Migrationsoptionen

Bevor über das weitere Vorgehen entschieden werden kann, müssen die gangbaren Wege untersucht werden. Im Fall der Migration von 9iR2 (Final-Release: 9.2.0.8) nach 11gR2 (zum Zeitpunkt der Umstellung: 11.2.0.2) können sie aus der *Abbildung 2* entnommen werden.

#	MIGRATIONSOPTION	ORACLE 9iR2	ORACLE 10gR2	ORACLE 11g	ENDIAN CONVERSION
1	DATA PUMP EXPORT/IMPORT	X	😊	😊	😊
2	RMAN CLONE	X	😊	😊	X
3	RMAN RESTORE	😊	😊	😊	X
4	RMAN CONVERT DATABASE	X	😊	😊	X
5	RMAN CONVERT TABLESPACE	X	😊	😊	😊
6	RMAN CONVERT DATAFILE	X	😊	😊	😊
7	TRANSPORTABLE TABLESPACES (TTS)	😊	😊	😊	X
8	CROSS PLATFORM TTS (XTTS)	X	😊	😊	😊
9	TRANSPORTABLE DATABASE (TDB)	X	😊	😊	X
10	EXPORT/IMPORT	😊	😊	😊	😊

Abb. 2: Migrationsoptionen

Da im vorgestellten Migrationsprojekt der ELSTER-Clearingstellen keine Änderung der Plattform bezüglich der “endianness” (Byte Reihenfolge: Angabe wo im Speicher das höchstwertige Byte steht) stattfand, konnten theoretisch die Verfahren RMAN Restore (3), Transportable Tablespace (7) und Export/Import (10) Anwendung finden. Erstes bietet den Transfer der Daten in Form eines RMAN-Backupsets an, welches ohne Installation der alten Version direkt mit dem neueren Oracle Release der Zieldatenbank als Grundlage für ein Upgrade verwendet werden kann (Details: MOS Note 369644.1). Die anderen beiden importieren die Datenfiles (9i) mit ihren Metadaten (7) bzw. alle Daten (10) in eine neu aufgesetzte Zieldatenbank (11g).

Nachteil ist in allen Fällen, dass die Zeit zwischen Backup und Upgrade bzw. Stopp der Applikation(en) und Einspielung der Daten als Downtime zu verzeichnen ist. Diese liegt zwar im günstigsten Fall im kleinen Stundenbereich, im ungünstigsten jedoch deutlich höher.

### Downtime Minimierung

Wie können wir nun die Downtime bei der Migration reduzieren? **Die Zeit der Migration ist nicht zwingend gleich der Downtime!**

Hier kommen Replikationstechniken ins Spiel, mit deren Hilfe sich die während der Migration anfallenden Modifikationen puffern lassen um sie anschließend in die neue Version einzuspielen. Somit teilte sich die Migration im Groben in folgende Teile auf (Details in *Abbildung 6*):

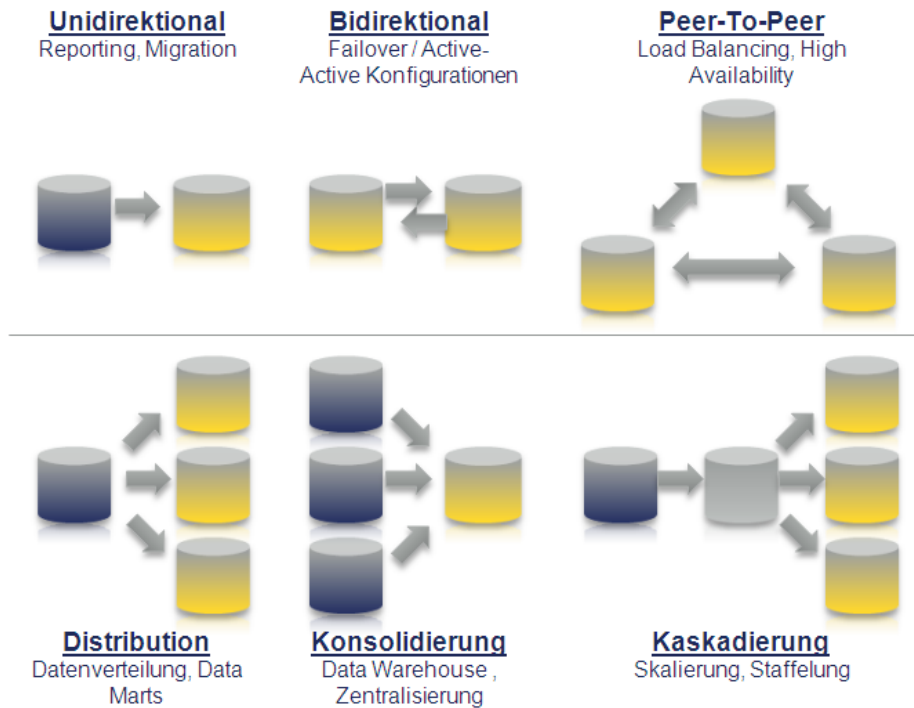
- 1) Datentransfer: z.B. Oracle DataGuard (bei laufender Datenbank) – ansonsten Punkt 2 vorziehen
- 2) Replikation: Einrichtung der externen Pufferung
- 3) Migration: Aktivierung der Standby-DB und InPlace-Migration nach 11gR2 & ASM (Primary online!) sowie Einrichtung von Full & Incremental Backups (11g)
- 4) Synchronisierung der Ziel-DB mit der Quelle: Einspielung der gepufferten Daten
- 5) **Downtime:** „Cutover“ – letztes inkrementelles Backup (11g), Umstellung der Application Server und Aktivierung der Fallback-Replikation (puffernd)
- 6) Backup der 9i-Instanz vor dem Apply der neuen Daten (Fallback-Absicherung)

Im vorliegenden Fall wurde nach einer externen Replikationstechnik gesucht, bei der alle erforderlichen Schritte (Konfiguration, Capture, Queueing, Transfer und Apply-Steuerung) außerhalb

der Datenbank stattfinden – und somit versionsunabhängig und möglichst frei von negativen Einflüssen durch/auf die Migration sind.

### Realisierung

Zur Realisierung wurden die Produkte ORACLE GoldenGate und QUEST Shareplex in die engere Wahl gezogen. Beide bieten, über die gewünschte Fallback-Funktionalität (Uni- & Bidirektionaler Aufbau) hinaus, die in *Abbildung 3* gezeigten Möglichkeiten.



*Abb. 3: Replikationsarten*

Von den genannten Replikationsarten kommt für das beschriebene Projekt lediglich die Variante Unidirektionaler Transfer, sowie abschließendem Ab- und (vorbereitetem) Neuaufbau mit Richtungsumkehr, zum Einsatz. Denkbar wäre in anderen Szenarien auch durchaus der Einsatz einer dauerhaften bidirektionalen Failover-Konfiguration oder der Transfer (von Teilen: Distribution) der Daten über ein DMZ in ein anderes Netz mittels einer Kaskadierung oder eine Mischung der dargestellten Replikationsarten.

In *Abbildung 4* sind die Prozesse (Rot) und Komponenten (Blau) der Software GoldenGate veranschaulicht.

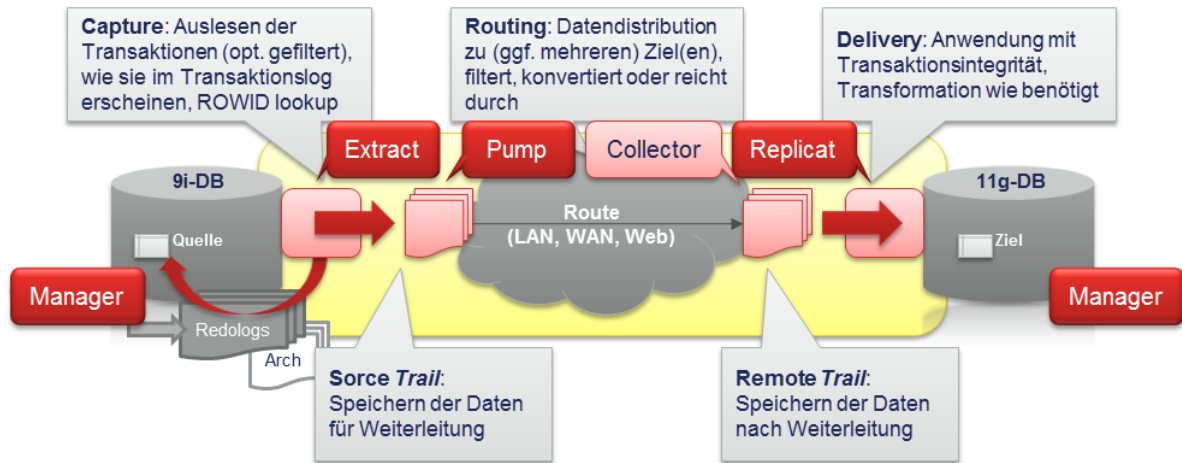


Abb. 4: Prinzip ORACLE GoldenGate

Vergleichend sehen Sie in Abbildung 5 die Prozesse (Rot) und Komponenten (Blau) der Software Shareplex.

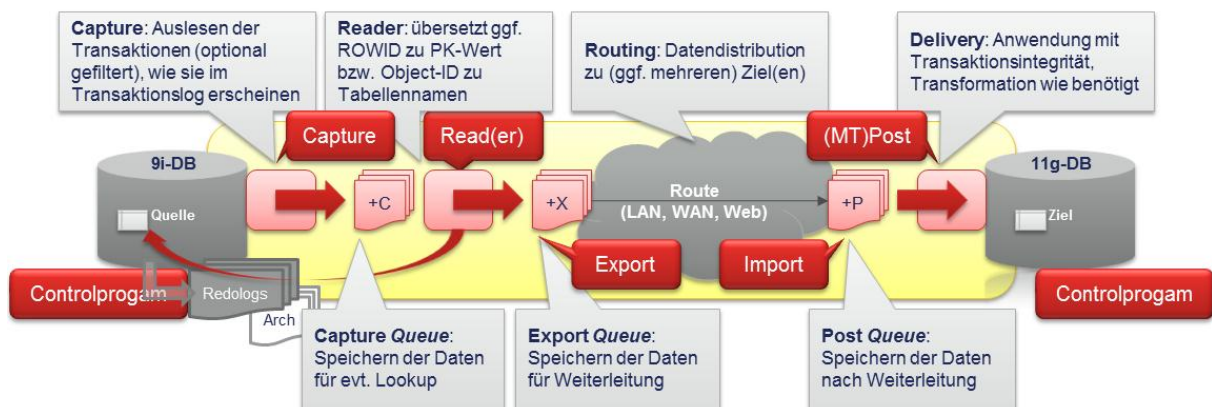


Abb. 5: Prinzip QUEST Shareplex

Es ist unschwer zu erkennen, dass beide ein sehr ähnliches Bild abgeben. Doch wo liegen nun die Unterschiede? Und welchen Einfluss haben diese auf ihren Einsatz im Migrationsszenario?

Zusammenfassend kann hier gesagt werden, dass beide Produkte die an sie gestellten Anforderungen erfüllen. Die Hauptunterschiede liegen in der Bedienung. Während GoldenGate für nahezu jedes denkbare Szenario konfigurierbar ist und durch seine Heterogenität besticht, ist das Setup gewöhnungsbedürftig und setzt das Verständnis des Grundkonzeptes voraus. Shareplex ist hingegen sehr einfach einzurichten, hat für alle in *Abbildung 3* dargestellten Replikationsarten eine vorgefertigte Syntax und besticht im Gegensatz zu GoldenGate sogar durch zusatzkostenfreie Features zur Überwachung und Korrektur. Heterogene Umgebungen scheiden jedoch hier aus, da das Produkt auf ORACLE Datenbanken beschränkt ist.

Auf die konkreten Unterschiede in Konfiguration und Bedienung wird im Vortrag genauer eingegangen.

In *Abbildung 6* sind die erkannten Vor- und Nachteile zusammengefasst.

## Oracle GoldenGate

### ■ Vorteile

- heterogen (auch MySQL, DB2,...)
- Unterstützung neuester Oracle-Datentypen und Features
- Fokus auf Mapping (flexible Trailverknüpfung)
- Strategisches ORACLE-Produkt!

### ■ Nachteile

- Komplex im Setup
- Überprüfung & Korrektur
  - €: Veridata (vs. compare & repair)
- Monitoring & Alerting
  - €: Mgmt Pack for OGG (vs. Foglight)

## Quest Shareplex

### ■ Vorteile

- Einfach im Setup
- Überprüfung & Korrektur: **enthält**
  - compare & repair (vs. Veridata)
- Monitoring & Alerting: **enthält**
  - Foglight (vs. Mgmt Pack for OGG)

### ■ Nachteile

- homogen: nur ORACLE
- wird immer etwas *hinterherhinken*
- Konkurrenzprodukt!

Abb. 6: Vergleich

### Migrationsszenario

Im vorliegenden Migrationsszenario kam die Software Shareplex zur Anwendung. In *Abbildung 7* ist der Einsatz vor und während der Migration in Blau gekennzeichnet und die Downtime sowie die Umkehr der Shareplex-Replikation in Rot. Beschrieben ist der Ablauf in Düsseldorf, gleiches gilt für die zweite Clearingstelle in München.

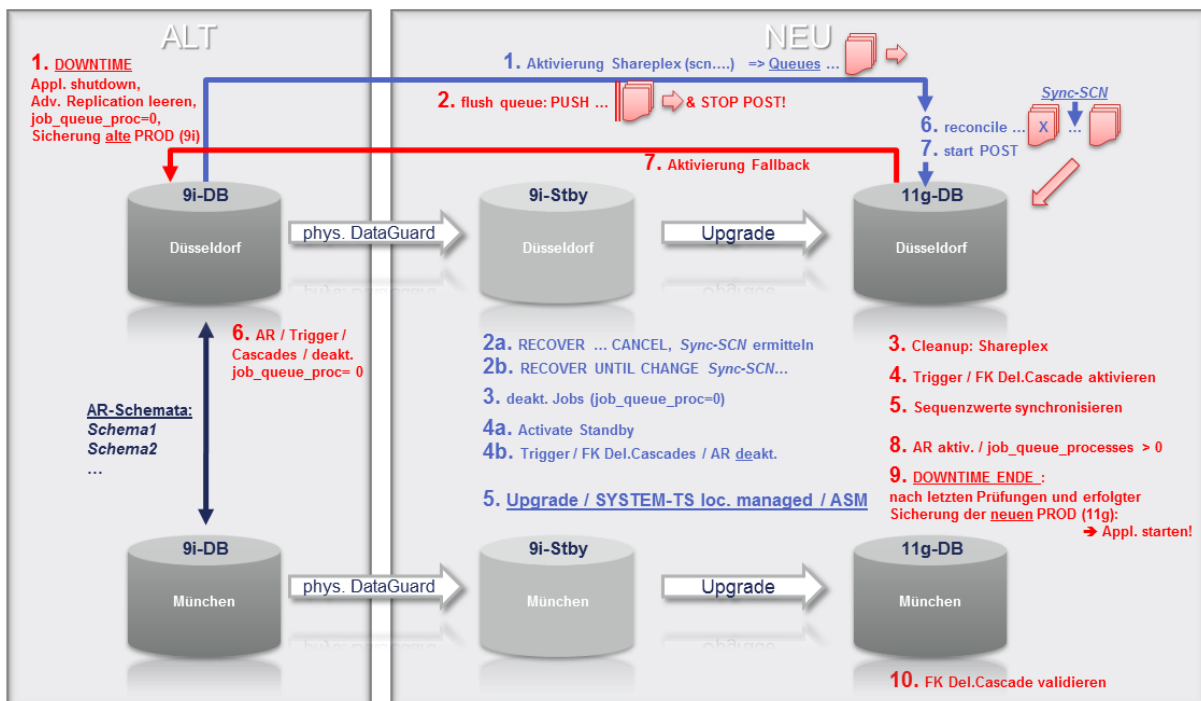


Abb. 7: Migrationsszenario



Das vorgestellte Migrationsszenario hatte den Vorteil dass die Migration selbst zu keinem Zeitpunkt Einfluss auf die fortdauernde Verfügbarkeit der alten Produktivumgebung hatte. Vielmehr kann die neue Datenbank in geöffnetem Zustand parallel – für Tests nur lesend, oder mittels Flashback-Funktionalität sogar schreibend – betrieben werden.

### **Fazit**

Nach guter Planung konnte Shareplex die Migration um eine kurze Unterbrechungszeit bereichern. Außerdem bot es gleichzeitig die Möglichkeit eines Fallbacks.

**Die Downtime beginnt also erst mit dem Stopp & Switch der Applikation!**

**Die Fallbackmöglichkeit garantiert, dass die nach dem GoLive anfallenden Änderungen in der alten Datenbank verfügbar sind. So kann selbst zu einem späteren Zeitpunkt ein Rückschritt auf die alte Umgebung durchgeführt werden!**

Kontaktadressen:

Christian Ballweg, Senior Consultant Service Engineering  
OPITZ CONSULTING Essen GmbH  
Altendorfer Straße 3  
D-45127 Essen

Telefon: +49 (0) 201-892994 1718  
Fax: +49 (0) 201-892994 4700  
E-Mail [christian.ballweg@opitz-consulting.com](mailto:christian.ballweg@opitz-consulting.com)  
Internet: [www.opitz-consulting.com](http://www.opitz-consulting.com)

Anja Albrecht, Sachgebietsleiterin Datenbankadministration  
Rechenzentrum der Finanzverwaltung des Landes Nordrhein-Westfalen  
Roßstraße 131  
D-40408 Düsseldorf

Telefon: +49 (0) 211-4572 2193  
Fax: +49 (0) 211-4572 1202  
E-Mail [Anja.Albrecht@rzf.fin-nrw.de](mailto:Anja.Albrecht@rzf.fin-nrw.de)  
Internet: [www.rzf.de](http://www.rzf.de)