Apex und Datenbanklinks ï Einsatz in Produktivumgebungen

Sven-Uwe Weller
syntegris information solutions GmbH
Neu-Isenburg

Schlüsselworte

Apex, Datenbanklinks, brown field projects

Einleitung

Der folgende Vortrag fasst Erfahrungen zusammen, die im Laufe verschiedener Projekte bei verschiedenen Kunden entstanden sind. Hierbei hat sich gezeigt das Apex als Werkzeug hervorragend geeignet ist, um schnell neue Funktionalitäten im Rahmen von Applikationen bereitzustellen, die auf bereits existierenden Datenbanken aufsetzen.

Aufgrund der vorhandenen Systemlandschaft ist jedoch zunächst die Hürde vorhanden, dass oft am laufenden System nichts geändert werden darf. Apex auf einer extra Datenbank und der Zugriff mittels DB-Links ist eine Alternative, die diese Einstiegshürde gering hält.

Der erste kurze Teil des Vortrags stellt typische Szenarien vor, für die so eine Systemlandschaft genommen werden kann.

Im zweiten Teil werden die Probleme näher beleuchtet, die durch den Einsatz von Datenbanklinks in Kombination mit Apex entstehen. Und es werden Wege aufgezeigt, wie diese Probleme vermieden werden können.

Integration von Apex Anwendungen in vorhandene Umgebungen

Die meisten Applikationen entstehen heute in den Unternehmen nicht auf der grünen Wiese, sondern erweitern bereits existierende Systeme, sogenannte Ăbrown field projectsñ. Sehr viele dieser Systeme haben eine Oracle DB im Backend. Immer wenn es in Unternehmen Anforderungen gibt, die mit den vorhandenen Lösungen nur schwer oder sehr aufwendig umgesetzt werden können, dann ist eine Überlegung, ob mit einem RAD Framework wie Apex diese Erweiterungen schnell und anwenderfreundlich umgesetzt werden können.

Gründe für den Einsatz von Apex sind häufig

- Moderne Oberfläche (browserbasiert
- Existierende Kenntnisse (SQL, Datenmodell, etc.) können weiterverwendet werden, neue Sprachen sind (im ersten Moment) nicht nötig.
- Schnell vorzeigbare Ergebnisse
- Geringer Deploymentaufwand (keine Client Installation, einfaches Provisioning)
- Administrationskenntnisse im Oracle Umfeld sind bereits vorhanden.
- Potential für Mobile Apps / läuft auch auf dem Tablett

Nachteile bzw. Hürden für den Einsatz sind häufig

- Änderungen an existierenden System sind nicht gestattet
- DBAs haben keine Erfahrung mit APEX
- Alles soll in einer einheitlichen integrierten Umgebung laufen

Szenario 1 ï Reporting Applikation

Für ein OLTP-Fremdsystem eines Softwareherstellers sollen zusätzliche Reports und Statistiken erstellt werden. Eine Erweiterung innerhalb des Fremdsystems ist sehr teuer oder dauert sehr lange. Die Applikationsbetreuer und Entwickler verfügen aber über solide Kenntnisse des zugrundeliegenden Datenmodels. Erweiterungen im Schema sind aber aus lizenzrechtlichen Gründen nicht gestattet.

Lösung: Die nötigen Reports werden in einer Apex Applikation zusammengefasst. Diese greift mittels read-only DB Links auf das existierende Schema zu. Lokal werden praktisch keine Daten gehalten.

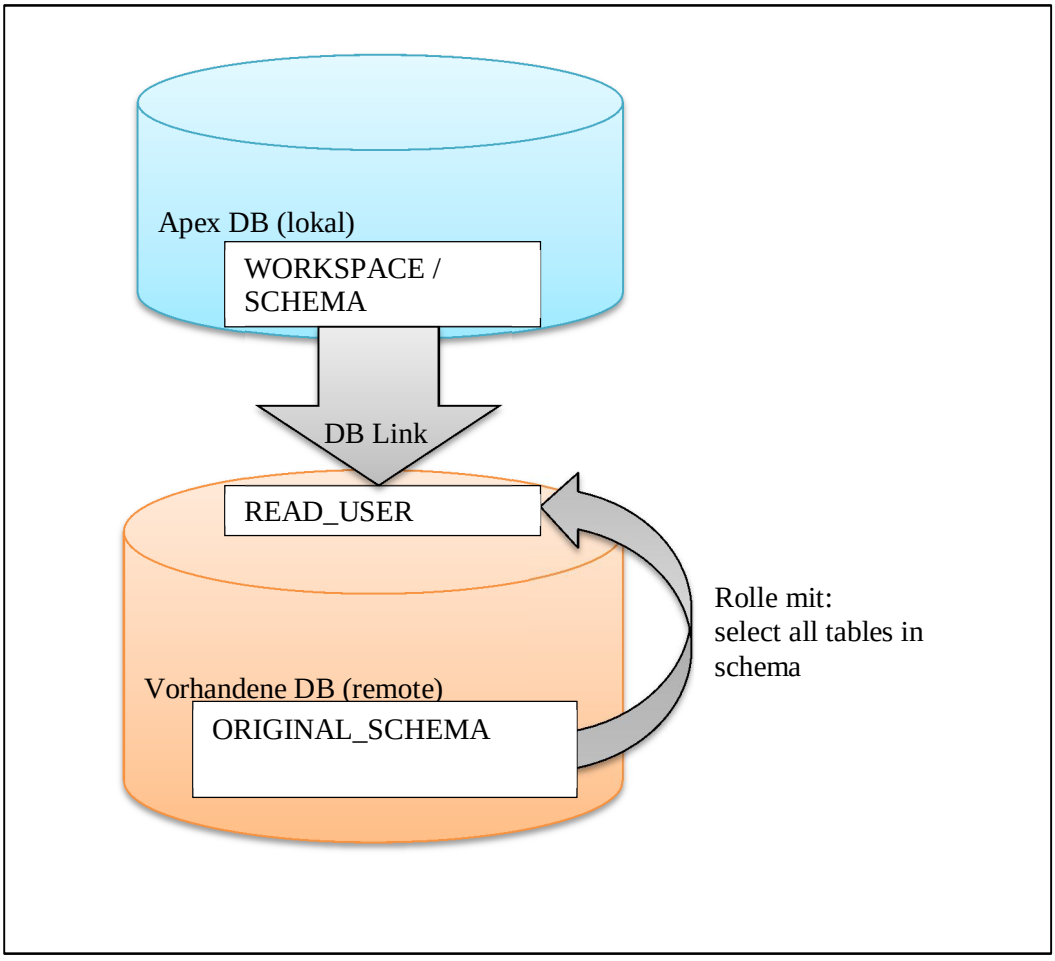


Abb. 1: Verbindung zw. Apex und existierenden DBs

Szenario 2 ï Vergleich Test und Produktiv DB

Eine extra DB auf der mit Apex gearbeitet werden kann eröffnet schnell Optionen, die es sonst in der Form nicht gibt. Ein Beispiel ist der Datenvergleich zwischen Test- und Produkttiefsystemen. Mittels geeigneter Mechanismen kann der gleiche Report sowohl für den Zugriff auf eine Test Instanz, als auch für den Zugriff auf eine Produktiv Instanz verwendet werden. Dies ist insbesondere bei Erweiterungen am vorhandenen System oder bei Vorabuntersuchungen sehr praktisch. Es gibt auch die Möglichkeit Reports zu erstellen, die gleichzeitig die Daten beider Systeme anzeigen.

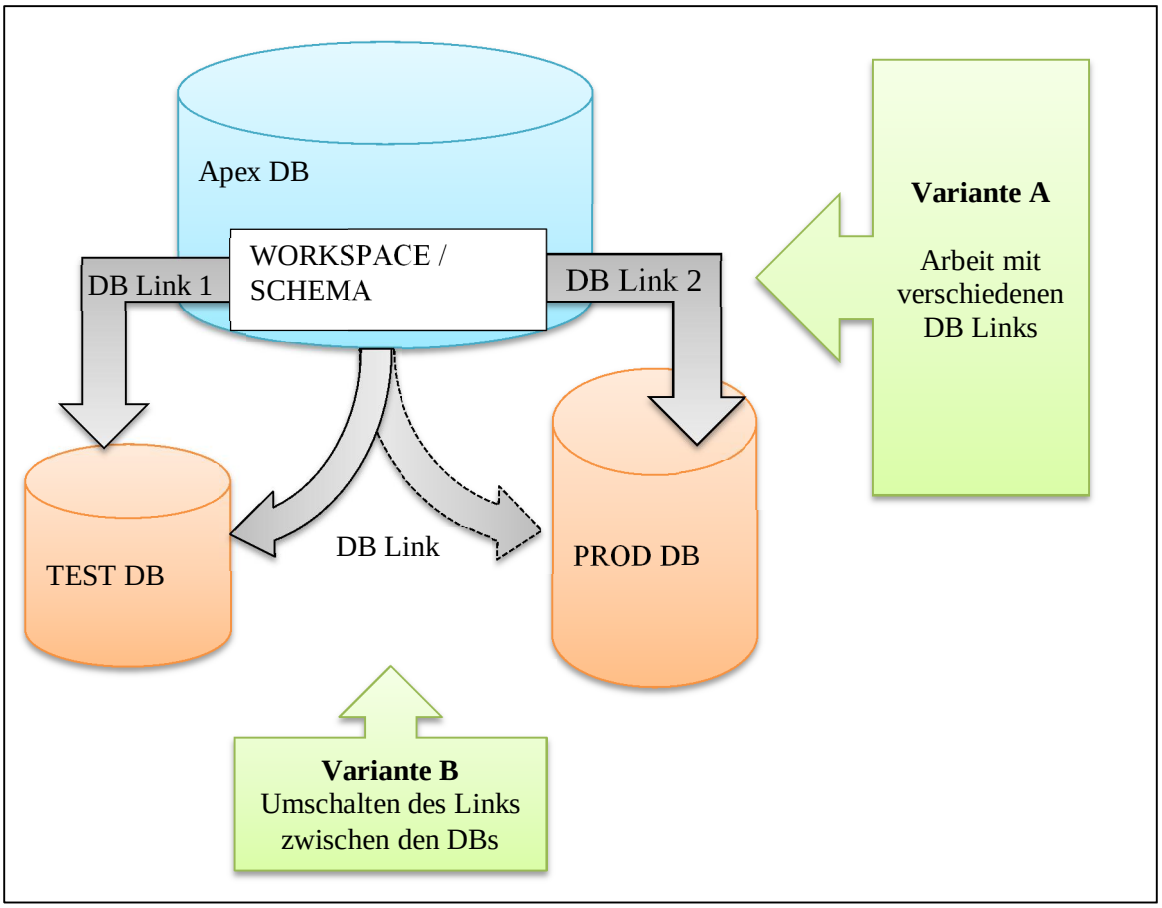


Abb. 2: Zugriff auf mehrere Remote DBs

Beispiel für Reports mit Daten aus beiden remote-DB-Instanzen.

```
select 'TEST' source, t.* from emp@testdb t
UNION ALL
select 'PROD' source, p.* from emp@proddb p
;
```

Tipp: Immer UNION ALL verwenden. Niemals nur UNION.

Beispiel für Report, der zwischen den Instanzen umschaltbar ist.

```
create synonym empremote on emp@testdb;
select e.* from empremote e;

Daten aus der Instanz 1# werden angezeigt.
```

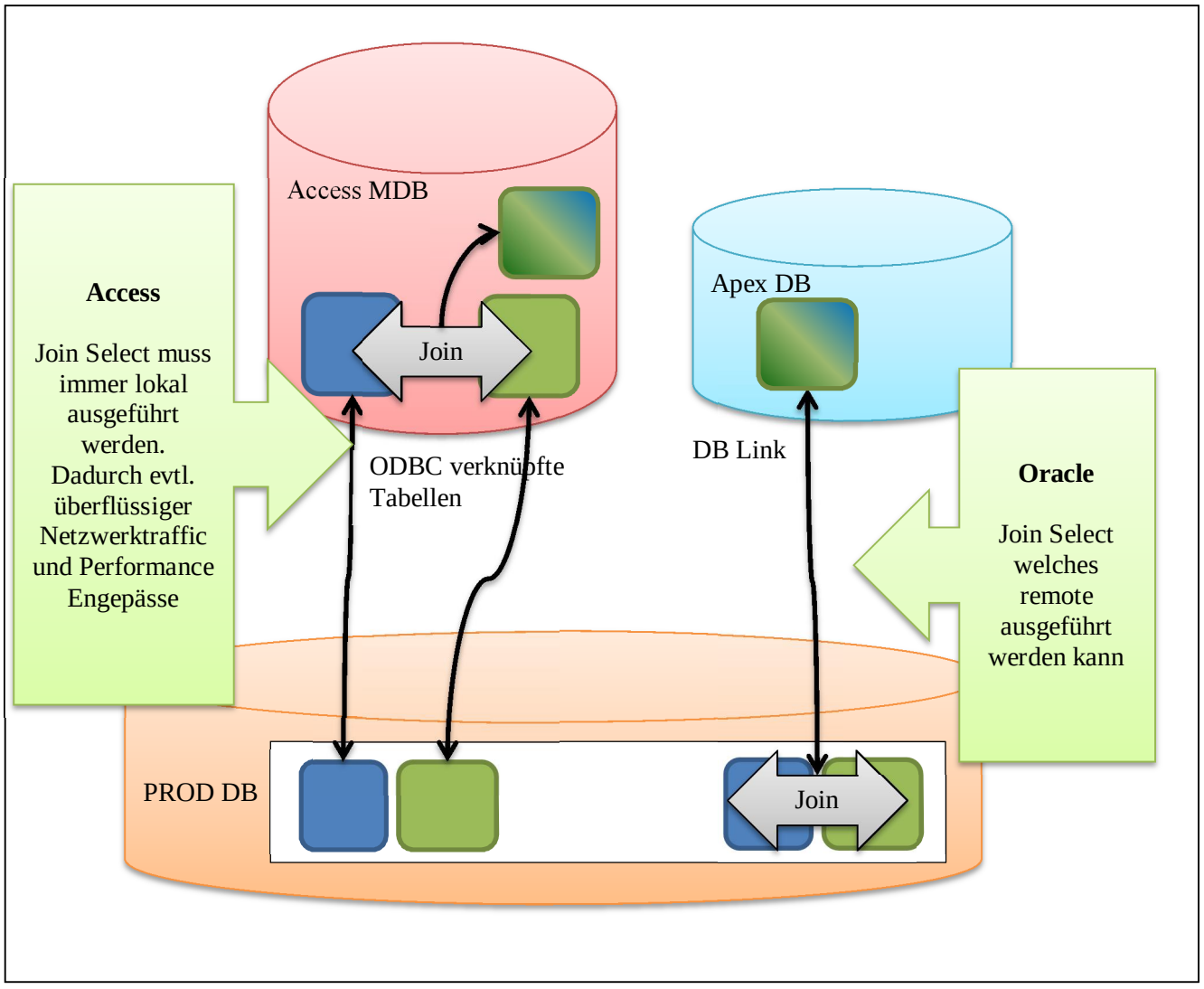
➔ Danach Switch des DB links oder alternativ umschalten eines Synonyms auf den zweiten Link

```
drop synonym empremote:
create synonym empremote on emp@proddb;

select e.* from empremote e;

Nun kommen die Daten aus der Instanz 2#!
```

Szenario 3 ï Ersatz von Access Zugriffen



Beispiel für Report, der die Abfrage remote ausführt.

```
-- Ermittle die Anzahl aller Angestellten, je Department
select /*+ driving_site( e ) */
       d.dept_name Department
```

```
     , count(e.empno) Anzahl
from deptremote d
left join empremote e on d.deptno = e.deptno;
```

Fallstricke und Lösungsmöglichkeiten bei der Verwendung von DB Links mit Apex

Im Folgenden werden einzelne Problemfelder beschrieben, die bei der Verwendung von DB Links in Apex auftreten können. Die Symptome und die Ursachen für diese Probleme werden erklärt. Danach erfolgt die Vorstellung von Lösungsansätzen oder Strategien zur Vermeidung des Problems.

Join Performance

Problem: Die Performance eines Joins hängt stark davon ab, ob die Daten lokal oder remote vorliegen. Bei Remote Abfragen sollte so viel wie möglich auf der remote DB ñgejointò werden und erst das Ergebnis über den DB Link zur lokalen DB kopiert werden. Oracle führt Standardmäßig ein Select immer lokal aus, selbst wenn alle beteiligten Tabellen remote liegen.

Lösung: Verwendung des driving_site hints.

Verwendung der Apex Wizards

Problem: Die Apex Wizards können mit remote Tabellen nichts anfangen. Auch nichts mit Synonymen.

Lösung: Jede Tabelle oder View auf die mittels DB Link zugegriffen wird, bekommt ein lokales Synonym und eine View die auf dieses Synonym zeigt.

Das Synonym dient nur dazu den DB Link aus den Apex Seiten fern zu halten. Es kommt doch häufig vor, dass der Name des Links sich ändert. Das Synonym bietet eine extra Abstraktionsschicht für minimalen Aufwand.

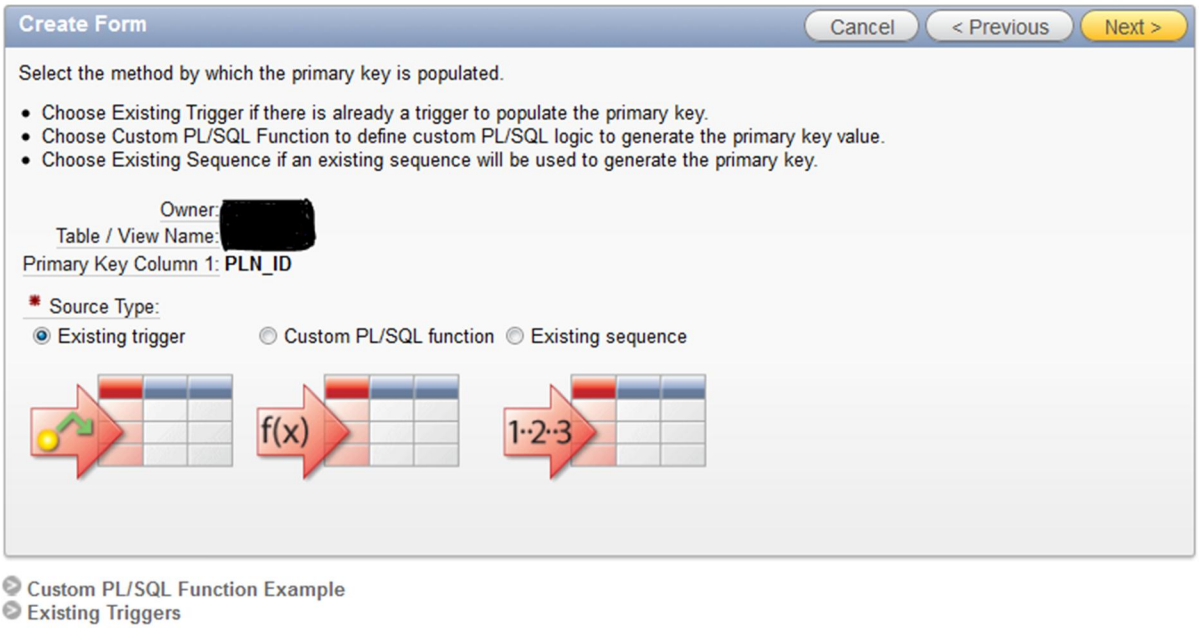
Die View dient dazu, den Apex Wizards ein lokales Objekt zu geben, auf dem der Wizard aufbauen kann. Sie dient als Abstraktionsschicht und bildet Meta Daten im lokalen Data Dictionary. Diese Meta Daten werden an einigen Stellen durch Apex verwendet. Am offensichtlichsten ist es bei den Report und Form Wizards, aber auch andere Stellen funktionieren nicht oder schlecht, wenn diese Metadaten fehlen.

Probleme mit ID Spalten

Sequenzen und die Returning Clause

Identity Spalten (kurz IDs) werden normalerweise mittels einer Oracle Sequence befüllt. Diese Sequence wird üblicherweise in einem BEFORE-ROW-INSERT Trigger auf der jeweiligen Tabelle aufgerufen und befüllt damit die ID Spalte (PK).

Apex kann damit sehr gut umgehen. Bei Verwendung des Form Wizards, wird man gefragt, wie die Werte für den PK bestimmt werden sollen.



Problem: Apex verwendet die RETURNING clause um den Wert der ID Spalte nach einem Insert in das Feld zu laden. Dies tut Apex auch, wenn das Feld RETURNING gar nicht verwendet wird. Meiner Meinung nach ein Bug. Die returning clause funktioniert aber nicht für remote Tabellen.

```
ORA-22816: Nicht unterstützte Funktion mit RETURNING-Klausel
ORA-06512: in Zeile x
22816. 00000 -  "unsupported feature with RETURNING clause"
*Cause:    RETURNING clause is currently not supported for object type
           columns, LONG columns, remote tables, INSERT with subquery,
           and INSTEAD OF Triggers.
*Action:   Use separate select statement to get the values.
```

Lösung 1: Apex 4.1 kann anstelle von PK Spalten auch mittels ROWID den jeweiligen Datensatz bestimmen. Dies ist teilweise ein Workaround. Scheint aber nicht in allen Fällen zu funktionieren.

Lösung 2: Ein extra insert process. Der normale DML Prozess wird nur noch für UPDATE und DELETE genommen. Fürs Insert gibt es eine eigene Umsetzung ohne die returning clause.

Lösung 3: Befüllen des Feldes aus der Sequence mittels einer Computation, kurz bevor das Insert gemacht wird. Diese Variante halte ich für sehr schwach. Dafür ist ein extra remote Zugriff nötig und die Seqeunce muss auch bekannt sein.

Das ID Beispiel wird im Vortrag ausführlicher gezeigt und die unterschiedlichen Lösungsstrategien näher beleuchtet. Ob und in welcher Form der Fehler auftritt hängt stark von der eingesetzten Apex Version ab.

Audit Spalten und DB Trigger

Problem: Ein DB Trigger schreibt den aktuell angemeldeten User in eine Spalte ñEINGEFUEGT_VONò. Der Trigger liegt auf der Remote DB und kann deshalb nicht auf v(óAPP_USERô) zugreifen.

Lösung1: Spalte über Apex selbst füllen. Hidden Column, Default Wert aus :APP_USER setzen Einfach, Übersichtlich, Nachvollziebar, aber extra Programmieraufwand bei jeder Form.

Lösung2: Einen Applikations Kontext für die Remote DB setzen. Der remote DB-Trigger wertet dann diesen Kontext aus. Apex macht das sogar automatisch für die lokale DB. Für die Remote DB ist es etwas schwieriger.

Speicherung von Lobs

Problem: Lobs können nicht über DB Links übertragen werden.

Die obige Aussage stimmt zum Glück nicht zu 100%. Es ist nur recht schwierig vernünftig mit LOBs umzugehen. Ein einfaches Select liefert aber einen Fehler, wenn lob Spalten vorhanden sind.

```
select clobColumn from myTable@remoteDb;

ORA-22992: LOB-Locators können nicht von Remote-Tabellen verwendet werden
22992. 00000 -  "cannot use LOB locators selected from remote tables"
*Cause:    A remote LOB column cannot be referenced.
*Action:   Remove references to LOBs in remote tables.
```

Lösung: Es wird lokal eine Kopie der remote Tabelle angelegt. Mittels geeigneter Mechanismen (insert.. select é statements) wird diese lokale Kopie synchronisiert (teilweise repliziert).

Sonstiges

Die Installation einer weiteren Oracle DB muss auch unter dem Gesichtpunkt der Lizenzkosten sorgfältig abgewägt werden. Für einige Szenarien ist der Einsatz einer Oracle XE Version denkbar. In vielen Großunternehmen ist jedoch die Installation einer weiteren Oracle DB praktisch lizenzkostenfrei. Entweder weil die entsprechenden Server bzw. VMs bereits eine weitere Oracle DB betreiben oder es gibt sogar eine unlimited license.

Sonstige Themen, wie z.B. Security Aspekte werden nicht näher beleuchtet.

Kontaktadresse:
Sven-Uwe Weller
syntegris information solutions GmbH
Hermannstraße 54-56
D-63263 Neu-Isenburg

Telefon:	+49 (0) 160 7565015
Fax:	+49 (0) 6102 558806
E-Mail	sven.weller@syntegris.de
Internet:	www.syntegris.de