

# Red Stack

Magazin

DOAG

SOUG  
swiss oracle  
user group

AOUG  
AUSTRIAN ORACLE USER GROUP

inklusive BUSINESS NEWS



# APEX

## Aus der Praxis

Datentransfer mit Oracle GoldenGate Microservices



## Im Interview

Mathias Weber und Markus Geis, LMU Klinikum, München

## Business News

Nachhaltige Geschäftsmodelle

# *APEX* *connect* by DOAG

22.04. - 24.04.2024

VAN DER VALK AIRPORTHOTEL  
DÜSSELDORF



[apex.doag.org](https://apex.doag.org)



Carolin Krüztzmann

DOAG Themenverantwortliche APEX

## Liebe Mitglieder, liebe Leserinnen und Leser,

derzeit erfreut sich das Thema Low-Code-Entwicklung zunehmender Beliebtheit und Oracle hat sich zweifellos mit APEX eine herausragende Position in diesem Bereich erarbeitet. In dieser Ausgabe des Red Stack Magazins freue ich mich sehr darüber, dass wir eine Vielzahl herausragender Beiträge präsentieren können, die verschiedene Aspekte dieses Themas beleuchten.

Unsere Ausgabe bietet eine breite Palette von Inhalten und Interviews, in denen die Anwendung und Nutzung von APEX an der Universität München erläutert wird, bis hin zu technischen Fachbeiträgen, die aufzeigen, wie die Entwicklung mit APEX schneller und effizienter gestaltet werden kann.

Besonders hervorzuheben ist unser exklusives Interview mit Marc Sewtz, der als „Senior Director of Software Development“ bei Oracle für die Weiterentwicklung von APEX verantwortlich ist. In diesem Gespräch gewährt er spannende Einblicke in die aktuelle Entwicklung und gibt einen Ausblick auf neue Funktionen, auf die man sich in den kommenden Versionen freuen kann.

Da APEX eng mit Oracle-Datenbanken zusammenarbeitet, finden Sie selbstverständlich auch Artikel aus anderen Bereichen, die sich mit Themen wie dem Cloud-Betrieb und alternativen Entwicklungsansätzen neben der traditionellen PL/SQL-Entwicklung beschäftigen.

Besonders in Zeiten, in denen Umweltschutz und Nachhaltigkeit immer mehr an Bedeutung gewinnen, möchte ich an dieser Stelle noch auf unsere „Business News“ hinweisen, in denen wir gleich drei Artikel zum Thema „Nachhaltige Geschäftsmodelle“ präsentieren können. Hier erfahren Sie, wie die IT-Community aktiv dazu beitragen kann, diese drängenden Umweltfragen anzugehen und Lösungen zu finden.

Ein großes Dankeschön geht an unsere Autorinnen und Autoren, deren wertvolle Beiträge das Red Stack Magazin zu dem machen, was es heute ist. Wir schätzen es sehr, dass ihr euer Fachwissen und eure Expertise mit unserer Gemeinschaft teilt.

Ich wünsche viel Spaß beim Durchblättern und Lesen dieser Ausgabe.

Carolin Krüztzmann



Ausgabe Nr. 6/2023  
auf Abruf!

## DOAG WEBSESSION

Die DOAG WebSessions\* bieten Ihnen in regelmäßigen Abständen spannende Online-Vorträge und -Diskussionen zu einer Vielzahl von Themenbereichen aus den jeweiligen DOAG Communities.

Freuen Sie sich auf WebSessions rund um die Themen Datenbank, Data Analytics und NetSuite oder beteiligen Sie sich bei den DOAG DevTalks an interessanten Gesprächsrunden zu aktuellen Development-Themen!



[www.doag.org/go/websessions](http://www.doag.org/go/websessions)



\*Die Buchung der WebSessions erfolgt ganz einfach über unseren Shop. Mitglieder erhalten im Buchungsprozess automatisch **100 % Rabatt.**



08  
Interview mit  
Mathias Weber  
und  
Markus Geis



18

APEX-Performance –  
Irgendwie ist meine  
Applikation langsam



46

Datenbank-Migration in  
die Oracle Cloud mit dem  
Oracle ZDM Tool

## Einleitung

- 3 Editorial
- 6 Timeline
- 8 „Vor allem die APEX Community ist hervorragend. Die Feedbackzeit bei Fragen, wenn man mal ein Problem hat, ist extrem kurz.“  
*Interview mit Mathias Weber und Markus Geis*
- 12 APEX-Low-Code oder was?  
*Günther Stürner*

## APEX

- 14 Flows for APEX 23.1  
*Moritz Klein*
- 18 APEX-Performance – Irgendwie ist meine Applikation langsam  
*Jörg Doppelreiter*
- 26 APEX-Item-Plugin-Entwicklung für Formulare und Interactive Grids  
*Michael Schmid*
- 34 Daten im Wandel der Zeit: Aufbau eines Historisierungs-Framework  
*Wolf G. Beckmann*
- 44 „Ohne das langjährige, aktive Engagement der Community, denke ich nicht, dass es APEX heute noch gäbe.“  
*Interview mit Marc Sewtz*

## Cloud

- 46 Datenbank-Migration in die Oracle Cloud mit dem Oracle ZDM Tool  
*Marc Wagner*

## Datenbank

- 74 Datentransfer mit Oracle Golden Gate Microservices leichtgemacht  
*Gerald Hacke*
- 82 Eine Einführung in Container Management, Kubernetes und Stateful Applications  
*Martin Bach*

## Engineered Systems

- 88 Die Qual der Wahl: Exadata Database Machine On-Premises oder Cloud Service?  
*Oliver Röniger*

- 98 Automatisiertes Patching der Oracle Exadata  
*Pavel Tolstorukov*

## BUSINESS NEWS

### Nachhaltige Geschäftsmodelle

- 56 Mit prozessorientierten Informationssystemen zu nachhaltigen Geschäftsmodellen  
*Andreas Fritsch*
- 62 Nachweis ökologischer Nachhaltigkeit in Geschäftsmodellen der Luftfahrt – Herausforderungen meistern durch Einsatz geeigneter IT-Lösungen  
*Dr. Thomas Mützel-von Schwartz und Florian Lösch*
- 70 Was ist ESG Reporting?  
*Chris Neubauer*



56

Leitartikel | Mit prozessorientierten Informationssystemen zu nachhaltigen Geschäftsmodellen



88

Die Qual der Wahl: Exadata Database Machine On-Premises oder Cloud Service?



62

Nachweis ökologischer Nachhaltigkeit in Geschäftsmodellen der Luftfahrt – Herausforderungen meistern durch Einsatz geeigneter IT-Lösungen



70

Was ist ESG Reporting?

### Intern

- 109 Neue Mitglieder + Termine
- 110 Impressum + Inserenten

### News

- 54 Event-Flatrate der DOAG
- 108 Best of DOAG Online

# TIMELINE

## 17. August 2023

Die Regio Hamburg trifft sich in der Hansestadt. Dieses Mal gibt es Vorträge aus den beiden Streams APEX und Oracle-Datenbank. Mit dabei ist auch der Direktor der Softwareentwicklung der Oracle Corp. in New York Marc Sewtz.

## 22. August 2023

Das Regionaltreffen NRW findet in Dortmund statt. Auf dem Programm stehen zwei Vorträge zu den Themen "Die Kraft der Einfachheit: Höchstleistung mit Oracle spielend erreichen" und "Backup Strategien in Zeiten von Verschlüsselungs-Trojanern. Warum Paranoia keine schlechte Strategie ist ...". Es moderiert der DOAG Themenverantwortliche Dev-Ops Axel vom Stein.

## 24. August 2023

Im DevTalk Summer Special erfahren die Teilnehmerinnen und Teilnehmer alles rund um das Thema APEX 23.x.

## 1. September 2023

In der DOAG WebSession heißt das Thema "Oracle Support: AHF – Autonomous Health Framework".

## 8. September 2023

In der DOAG WebSession mit Christian Pfundtner geht es um das Thema "DBA\_INDEXE\_USAGE – Speed up your DML".

## 12. September 2023

Der Java-Treff der Java User Group Ingolstadt findet als Veranstaltung mit Vorträgen, Austausch und Diskussion statt. Mit dabei ist diesmal Natan Yellin, der über "The Robusta Monitoring Platform" referiert.

## 20. September 2023

Beim Regionaltreffen Rhein-Main in Wiesbaden können sich die Teilnehmerinnen und Teilnehmer auf zwei Vortragsthemen freuen. "Docker container in a nutshell (for DBAs)" und "K8s (Kubernetes für den DBA)".

## 21. September 2023

Im DevTalk mit Carsten Czarski, Jürgen Menge und Frank Hoffmann steht das Thema "Oracle Forms" auf der Agenda.

## 26. und 27. September 2023

Aufgrund der hohen Nachfrage beim ersten Noon2Noon-Workshop "Datenbank-Security" im März gibt es einen Wiederholungstermin in Hannover. Die DOAG Datenbank Community lud die zahlreichen Teilnehmerinnen und Teilnehmer zu diesem Hacking-Event ein.

## 27. und 28. September 2023

Das vom pfälzischen Datenbank-Experten Randolph Eberle-Geist gehaltene Berliner Expertenseminar konzentriert sich auf B\*Tree und Bitmap-Indizes und vermittelt an zwei Tagen alles, was Server- und Datenbank-Administrierende über Oracle-Datenbank-Indexing wissen müssen.

## 13. Oktober 2023

In der DOAG WebSession mit Stefan Oehrli geht es um das Thema "Oracle 23c DB Nest im praktischen Einsatz".



NEUE LOCATION!

# Javaland

www.javaland.eu

## AM NÜRBURGRING

09. -

11.04. 2024

Early Bird bis 15.02.2024



### JETZT TICKETS SICHERN!





*„Vor allem die APEX-Community ist hervorragend. Die Feedbackzeit bei Fragen, wenn man mal ein Problem hat, ist extrem kurz.“*

Martin Meyer, Redaktionsleiter des Red Stack Magazins, unterhielt sich mit Mathias Weber und Markus Geis, beide vom LMU Klinikum, Institut für Notfallmedizin und Medizinmanagement (INM), über die Oracle-Datenbank, die Entwicklung von APEX zum zentralen Entwicklungstool für Applikationen im Institut, den vielfältigen Einsatz und die APEX Community sowie zukünftige Entwicklungstrends.



## Bitte stellen Sie sich beide kurz unseren Lesern vor. Wer sind Sie und mit was beschäftigen Sie sich?

**Mathias:** Wir arbeiten beide am Institut für Notfallmedizin und Medizinmanagement (INM) am Klinikum der Universität München. Markus im Bereich IT und ich im Bereich Systemanalyse und Prozessmanagement (SysPro).

**Markus:** Ich bin DBA für einen Oracle-RAC-Cluster und für die Installation sowie den grundsätzlichen Betrieb der APEX-Applikationen verantwortlich. Mathias arbeitet in der APEX- und DB-Entwicklung.

## Wie kamen Sie zur Oracle-Datenbank und schließlich zu APEX?

**Markus:** Ich habe mich bereits im Studium mit SQL und relationalen Datenbanken beschäftigt. Die Oracle-RDBMS-Version war damals 4.0.11 auf Olivetti XT. Nach dem Studium habe ich bei unterschiedlichen Firmen sowohl als Oracle-DBA als auch im Bereich der Entwicklung gearbeitet. Hier ging es vor allem darum, bestehende BS2000-Systeme in die Oracle-Welt zu überführen.

**Mathias:** Mein erster Kontakt mit Oracle war im Bereich der Geoinformatik. Ich habe mit ESRI-Produkten gearbeitet und die Datenhaltung erfolgte in ArcSDE auf einer Oracle-Datenbank. Im INM haben wir unsere Daten schon ziemlich früh auf einer Oracle-Datenbank gehalten. Allerdings wurden Auswertungen zunächst in Excel und Access durchgeführt, auch wegen der Möglichkeiten, Grafiken für gedruckte Gutachten zu generieren. Mit der Zeit bekam die Oracle-Datenbank jedoch eine immer zentralere Rolle und wurde nicht mehr nur zur Datenhaltung und -sicherung verwendet, sondern auch zur Berechnung von Auswertungen, Analysen und so weiter.

**Markus:** Die Ergebnisse unserer Auswertungen wurden relativ früh schon in einer Web-Applikation zur Verfügung gestellt. Umgesetzt wurde das ganze zunächst mit JavaServer-Pages, dann in Java. Aufgrund der immer kürzer werdenden Entwicklungszyklen haben wir uns dann nach neuen Entwicklertools umgesehen und sind schließlich auf APEX gestoßen. Angefangen haben wir mit der Version 3.2.

**Mathias:** APEX war ideal für uns. Die erforderlichen Kenntnisse, SQL, PL/SQL, HTML und JavaScript waren bei uns im Institut vorhanden. Wir konnten also sofort loslegen, da die grundlegenden Strukturen in der Datenbank bereits vorhanden waren. Wir benötigten somit kein neues System und konnten mit unseren Kenntnissen unmittelbar datenbankzentrierte Web-Applikationen entwickeln. Im Laufe der Jahre entwickelte sich APEX zum zentralen Entwicklungstool für Applikationen im Institut. Es wurde in unterschiedlichen Projekten für unsere Auftraggeber, aber auch für die Entwicklung interner Applikationen eingesetzt.

## Wo und wie setzen Sie APEX ein?

**Markus:** Das INM als interdisziplinäre Einrichtung verfolgt das zentrale Ziel, die Qualität der akutmedizinischen und rettungsdienstlichen Versorgung der Bevölkerung durch Dienstleistungen, Forschung und Lehre kontinuierlich zu optimieren. APEX setzen wir unter anderem bei der Umsetzung großer Portale für uns, unsere Kunden und Auftraggeber ein.

**Mathias:** Das sind zum Beispiel die Datendienste Bayern, die wir im Rahmen der Trend- und Strukturanalyse des Rettungsdienstes in Bayern (TRUST I – IV) für das Bayerische Staatsministerium und die Sozialversicherungsträger in Bayern entwickelt haben und seit mehreren Jahren betreiben. In diesem Portal erfassen wir alle Strukturen des bayerischen Rettungsdienstes – Rettungsdienststandorte, Notarztstandorte, die Vorhaltungen an den einzelnen Standorten, das heißt, wann ist wo ein Fahrzeug im Dienst, Krankenhäuser und die für die Notfallrettung relevanten Einheiten und vieles mehr.

**Markus:** Zudem werten wir seit dem Jahr 2000 die Daten aller bayerischen Leitstellen aus und analysieren diese Daten. Die aufbereiteten Ergebnisse werden den Auftraggebern und allen am bayerischen Rettungsdienst Beteiligten – den Zweckverbänden, den durchführenden Hilfsorganisationen, den Leitstellen, ärztlichen Leitern und Weiteren – zur Verfügung gestellt. Die Ergebnisse spielen auch eine Rolle, wenn es darum geht, Standorte zu verschieben, neue Standorte zu planen oder die Zeiten, in denen Rettungs- oder Transportmittel zur Verfügung stehen müssen, anzupassen.

**Mathias:** Ergänzt werden die tabellarischen und grafischen Darstellungen in vielen Bereichen durch kartographische Darstellungen, da die räumlichen Zusammenhänge hierüber viel besser zu visualisieren sind und erfasst werden können.

**Markus:** Wir verfügen somit über eine Datenbank, in der alle Struktur-, Einsatz- und Abrechnungsdaten des Rettungsdienstes in Bayern umfassend dokumentiert sind. Sie ist die Grundlage für Analysen der Versorgungssituation und der Gutachten zur Standortstruktur des Rettungsdienstes in Bayern.

Bei notwendigen Veränderungen oder Anpassungen in der Rettungsmittelvorhaltung oder der räumlichen Verteilung von Rettungs- oder Notarztstandorten arbeiten wir mittlerweile auch mit Simulationen, die das Rettungsdienstgeschehen auf Basis der Realdaten auswerten.

**Mathias:** Das SQR-BW-Portal, das wir für die Stelle zur trägerübergreifenden Qualitätssicherung im Rettungsdienst Baden-Württemberg (SQR-BW) entwickelten, ist ein weiteres Portal, das wir mit APEX realisiert haben und seit mehreren Jahren betreiben. Auch hier werden rettungsdienstliche Strukturdaten dargestellt und Analysen auf Basis von Einsatzdaten und Protokollen zur Verfügung gestellt. Ein Bereich des Portals ist der so genannte „Gestufte Dialog“. Es handelt sich hierbei um ein Verfahren, in dem rechnerisch auffälligen Indikatoren nachgegangen wird. Qualitätsdefizite werden so erkannt und geeignete Maßnahmen zur Beseitigung können eingeleitet werden. Die Beteiligten an diesem Dialog haben dabei unterschiedliche Sichten auf die einzelnen, teilanonymisierten Daten.

Das Besondere am SQR-BW-Portal ist, dass die Mitarbeiter des SQR-BW die Analysen, die im Portal dargestellt werden können, selbst über das Portal definieren können. Sie legen das entsprechende SQL auf die Daten, können definieren, ob in der Applikation eine tabellarische, grafische oder kartographische Darstellung angezeigt werden soll, und können bei einer grafischen Darstellung sogar zwischen verschiedenen Diagrammtypen wählen. Sie entwickeln praktisch ihre Analysen in einer APEX-Applikation, können das Ganze testen und dann in die Live-Applikation übernehmen. Zudem können sie über die Applikation auch die Zugriffsrechte auf die Analysen in Kombination mit

der Analyse-Ebene definieren. So kann festgelegt werden, dass zum Beispiel ein Rettungsdienstleiter Auswertungen für einen ganzen Bereich, ein Standortleiter jedoch nur Auswertungen für seinen Standort sehen kann. Das SQL für die Darstellung der Daten ist jedoch immer gleich.

**Markus:** Außerdem gibt es verschiedene kleine Portale, die wir im Rahmen von Projekten in den Bereichen Rettungsdienst und Notfallrettung entwickelt haben. Intern verwenden wir APEX auch. Hier zum Beispiel für das interne Bestell-, Auftrags- und Rechnungswesen.

Zu Beginn der Corona-Pandemie haben wir für die Koordinationsgruppe Münchner Kliniken (KMK) im Auftrag des Referats für Gesundheit und Umwelt der Landeshauptstadt München (RGU) in wenigen Tagen eine Applikation zur Verwaltung der Covid-Patienten im Hinblick auf freie Betten und Beatmungskapazitäten entwickelt.

### Können Sie Beispiele für Anwendungen in der Klinik beschreiben?

**Markus:** Im INM verwenden wir APEX-Applikationen für die Zeiterfassung und Aufgabenverwaltung. Aber auch für das interne Bestell-, Auftrags- und Rechnungswesen.

**Mathias:** Zurzeit arbeiten wir unter anderem an einer Kanban-Board-Applikation, die wir in APEX bauen. Hier werden wir Schnittstellen in verschiedene Applikationen realisieren, sodass unsere Kunden uns Wünsche, Aufgaben oder Bugs zu ihren Applikationen mitteilen können und wir diese Rückmeldungen gleich als Task in den Projektlisten des Kanban-Boards übernehmen können.

**Markus:** Innerhalb des KUM setzen wir APEX-Applikationen in der Verwaltung von Kursen und Fortbildungsveranstaltungen ein.

### Gibt es eine APEX-Anwendung, die besonders bedeutsam ist – und warum?

**Mathias:** Da fällt mir spontan die bereits erwähnte Covid-Applikation ein. Wir haben mit einer Handvoll Leuten in kürzester Zeit eine Datenbank, ein Datenbank-Schema inklusive Userverwaltung und Zugriffskontrolle sowie eine Internet-Applikation mit APEX entwickelt. Das ist ein perfektes Beispiel für ein Low-Code-Projekt.

### APEX ist ja ein Teil der Oracle-Datenbank: Was außer APEX setzen Sie noch alles ein?

**Markus:** Wie bereits erwähnt, betreiben wir einen Oracle Real-Application-Cluster (RAC) mit einer Enterprise Edition. Fast alle Applikationen, die wir betreiben, benötigen eine User-Authentifizierung. Hierfür nutzen wir das Oracle Unified Directory (LDAP V3). Unsere User können sich so mit einer Kennung und dem gleichen Passwort an allen für sie freigeschalteten APEX-Applikationen anmelden. Zudem ist die Administration der User wesentlich einfacher. Für die Autorisierung der Daten nutzen wir das Virtual Private Database (VPD). Über Policies und Policy-Funktionen werden die Rechte, die ein User auf die Daten hat, durchgesetzt. Dies bezieht sich zum einen auf den räumlichen Aspekt, also etwa Rettungsdienstbereiche und die dazugehörigen ret-

tungsdienstlichen Strukturen, wie Rettungswachen oder Notarztstandorte, zum anderen auf Analysen auf unterschiedlichen Ebenen. Virtual Private Database setzen wir in allen Applikationen ein. Der Vorteil liegt vor allem darin, dass die Rechtestruktur in der Datenbank abgebildet und ohne Anpassungen der Applikation geändert werden kann.

**Mathias:** Um die großen Datenmengen, wir arbeiten unter anderem mit Tabellen, die mehr als 53.000.000 Zeilen umfassen, performant zu verarbeiten, nutzen wir das Partitioning, eine Option der Enterprise Edition.

Für die APEX-Umgebung nutzen wir den Oracle Restful Data-service (ORDS) auf einem Tomcat. Die Applikationen werden von einem Apache-Server nach außen präsentiert.

Im APEX-Umfeld betreiben wir einige Applikationen, in denen wir APEX Office Print, das wir auf einem eigenen Server betreiben, verwenden. AOP nutzen wir für die Erstellung von PDF-Dokumenten, die sich die User bei Bedarf selbst erstellen und downloaden können oder auch für die Erstellung von klinikinternen Bestellformularen im PDF-Format. Früher wurden diese Dokumente am Rechner ausgefüllt, gedruckt, unterschrieben, gescannt und schließlich per E-Mail an den Einkauf versendet. Jetzt werden alle Daten in einer APEX-Applikation erfasst und mit AOP in das Bestellformular gedruckt, das wiederum in der Datenbank gespeichert wird. Die Beteiligten können die Formulare mit einer digitalen Unterschrift freigeben und letztendlich wird das Formular aus der Datenbank direkt an den Einkauf versendet. Ausgedruckt wird jetzt nichts mehr.

**Markus:** Als Ersatz für Microsoft Sharepoint werden wir zukünftig APEX Office Edit einsetzen. Hier sind wir jedoch gerade am Anfang der Projektphase. Zudem haben wir Flows für APEX in einem APEX-Projekt in Verwendung. Hier sogar in Kombination mit Oracle Text und APEX Office Print.

Die räumlichen Daten der rettungsdienstlichen Strukturdaten speichern wir ebenfalls in der Oracle-Datenbank. Früher hatten wir für die Darstellung von kartographischen Daten auch den Oracle MapViewer in Verwendung.

### Wie viele User verwenden Ihre Apps?

**Mathias:** Das ist unterschiedlich, je nachdem, ob es sich um eine interne Applikation oder eine Plattform, die nach draußen geht, handelt. Die internen Applikationen werden von 30 bis 50 Usern genutzt. Auf die großen Portale, wie zum Beispiel die Datendienste Bayern, können circa 800 User zugreifen.

### Welche Rolle spielt die Oracle- und APEX Community für Sie?

**Markus:** Vor allem die APEX-Community ist hervorragend. Die Feedbackzeit bei Fragen, wenn man mal ein Problem hat, ist extrem kurz.

**Mathias:** Der Austausch in den Communities ist wichtig. Man kann sich fachlich weiterentwickeln, schauen, wie setzt ein anderer Entwickler eine Sache um, es entstehen neue Ideen und Projekte. Und letztendlich ist ein Netzwerk auch wichtig für den Austausch bei Problemen.

**Markus:** Mathias und ich haben im Laufe der Zeit schon verschiedene Vorträge, auch für die DOAG, gehalten. Hier in

München unterstützen wir aktiv die Oracle-MeetUp-Gruppe in München, die sich auch bei uns im INM immer wieder trifft. Es ist ein Geben und Nehmen und, wenn jeder versucht, möglichst viel beizutragen, haben am Ende alle mehr davon.

### Welche Trends sehen und welche Entwicklungen erwarten Sie für die Zukunft hinsichtlich APEX und Development?

**Markus:** Ich denke, dass Oracle für die Reports weitere Funktionen zur Verfügung stellen wird, die einfach und schnell zu nutzen sind. Ich denke da beispielsweise an die Möglichkeiten, Trends in großen Datenmengen einfach zu erkennen und einfach visualisieren zu können oder auch Vorhersagen treffen zu können.

**Mathias:** Ich bin auf die in der APEX-Roadmap vorgestellte Visualisierung der Anwendungslogik gespannt. Dies wird vor allem bei komplexen Seiten hilfreich sein. Zudem wird die Einbindung von KI auch vor der Applikationsentwicklung nicht Halt machen und diesen Bereich sicherlich, vor allem, was die Entwicklungszeiten und die Reproduzierbarkeit von Code angeht, verändern.



#### MARKUS GEIS

Schon während des Studiums hat sich Markus Geis für relationale Datenbanken interessiert und ist sehr frühzeitig mit Oracle-RDBMS in Kontakt gekommen. Mit SQL-Forms und Oracle Pro-C\*-Compiler hat er am Anfang die DB-Applikationsentwicklung durchgeführt. Später war er hauptsächlich mit Migrationsprojekten von „alter-Welt“ auf RDBMS beschäftigt. Er arbeitet seit 20 Jahren im Institut für Notfallmedizin und Medizinmanagement des Klinikums der Universität München und ist dort für den Betrieb der Oracle-Umgebung (RAC, ORDS, OUD, APEX) zuständig. Wie Mathias hat er auch frühzeitig die Möglichkeiten von APEX erkannt und die Umstellung der Entwicklungsumgebung angestrebt.



#### MATHIAS WEBER

Mathias Weber arbeitet seit über 20 Jahren im Bereich Systemanalyse und Prozessoptimierung am Institut für Notfallmedizin und Medizinmanagement des Klinikums der Universität München. Nach mehreren Jahren in der Entwicklung von Geoinformationssystemen wechselte er in die Oracle-APEX-Entwicklung. Seit der APEX-Version 3.2 realisiert er mit einem kleinen Team Low-Code-Applikationen und die entsprechenden Datenbankmodelle.



# AUS DER FERNE BETRACHTET:

## APEX-LOW-CODE ODER WAS?

Kürzlich bin ich über eine Analyse von Gartner Inc. zum Thema **Low-Code Application Platforms** gestolpert (vom 31.12.2022). Siebzehn Systeme wurden analysiert und im berühmten Gartner Magic Quadrant dargestellt. Mit einem Blick kann man die Gewinner und die Verlierer erkennen oder im Gartner-Sprachgebrauch die Leader, die Niche Player, die Challengers und die Visionaries. Oder im Klartext: Unten links ist schlecht, Finger weg davon, oben rechts ist gut, da macht man keinen Fehler, die beiden anderen Quadranten, mal sehen was aus denen wird.

Nun, ich bin kein Kenner der Low-Code-Welt. Das einzige System, das ich kenne, mit dem ich auch arbeite und das sich der Kategorie „Low-Code-Plattform“ selbst zuordnet, ist APEX (<https://apex.oracle.com/de/plattform/low-code/>).

Wie OutSystems, Retool, Mendix, MS Power Apps, Zoho oder die anderen aufgeführten Plattformen funktionieren oder wie weit sie verbreitet sind, weiß ich nicht. Auch kenne ich keinen Entwickler, der mit diesen Systemen arbeitet. Aber das will nichts heißen. Man kann nicht alles wissen und nicht auf allen Hochzeiten tanzen.

Überraschend war jedoch die Bewertung von APEX. Ich hatte vermutet, dass sich APEX mit Sicherheit im oberen rechten Quadranten wiederfindet, bei den Guten, bei den Wichtigen und Richtigen. Nicht unbedingt ganz oben rechts – obwohl das meine persönliche Wahl gewesen wäre –, aber zumindest bei den Top Drei.

Gelandet ist APEX jedoch nur im linken-oberen Quadranten, der die „Herausforderer“ die „Challengers“ versammelt, mit einem nicht unerheblichen Abstand zu den fünf Bestbewerteten.

Als Fan ist das enttäuschend und als Hardcore-Fan geht man auf Ursachenforschung und die beginnt bei dem Textteil der Gartner-Analyse. Man braucht nicht allzu viel Fantasie, um zu merken, dass sich die Autoren und Autorinnen nicht auf eigene Erfahrung mit den Produkten, sondern auf die Webpages der Anbieter und im besten Fall auf Interviews oder sonstige Aussagen von Firmenvertretern verlassen haben. Die finanzielle Kapazität von Oracle, das Commitment zu APEX, die 80.000 Sales Reps, die auf den Verkauf von APEX trainiert sind, und die weltweite Präsenz sowie die große Community werden als Stärken von APEX gelobt. Bis auf die auf APEX-trainierten Sales-Reps, die es nicht gibt und noch nie gab, sind es banale Aussagen, die nichts aber auch gar nichts zu APEX, der APEX-Architektur, den APEX-Fähigkeiten oder anderem aussagen.

Das soll jetzt kein Gartner-„Bashing“ werden, obwohl ich durch eigene Erfahrung geprägt, ein eher distanziertes Verhältnis zu diesen Analytikern habe. Viele der Analysen von Gartner und Co. bleiben eher im Ungefähren, aber solange Gartner mit seinen Magic Quadrants einen großen Einfluss und eine große Reichweite hat, muss man sich als Anbieter entsprechend verhalten. Erklären, zeigen, versuchen zu überzeugen. Zwar ist es oft mühsam und kann auch frustrierend sein, muss aber professionell durchgeführt werden. Daher ist es eine Aufgabe für die Besten eines Teams und sollte auf keinen Fall als zweitrangige Aktion betrachtet werden.

Eine andere Möglichkeit wäre, wenn man sich als APEX aus dem Low-Code-Umfeld löst und eine neue, eine eigene Kategorie erfindet. Low-Code ist ja nicht unbedingt ein Begriff, den man

unmittelbar mit etwas Gutem, etwas Innovativen verbindet. Eher kommen einem Begriffe wie Low-Quality, Low-Budget, Low Profile und ähnliche, nicht eben positiv geprägte Begriffe, in den Sinn. „Hallo, guten Tag, ich bin der neue Low-Code- Entwickler“. Klingt schon schräg.

Oracle hat einige geniale Marketing-Leute. Es sollte doch mit dem Teufel zugehen, wenn denen nichts Besseres als Low-Code einfällt.

Vielleicht sollten sie sich ein Beispiel an meinem Heimat-Bundesland nehmen. Wir nennen uns ja auch nicht mehr Baden-Württemberg – und stehen in Konkurrenz mit Bayern, Schleswig-Holstein und anderen –, sondern nennen uns kurz und bündig, mit internationalem Flair, „The Länd“.

Was Kretschmann kann, sollte auch Larry hinbekommen.

PS: In eigener Sache

Wolfgang Taschner hat die Rubrik: „Aus der Ferne betrachtet“ 2016 „erfunden“. Seine Idee war, dass Personen, die nicht unmittelbar im Tagesgeschäft der IT stecken, zu unterschiedlichen Themen ihre Meinung kundtun können. Die Distanz der Autoren und Autorinnen, so die Idee, könnte zusätzliche oder neue Aspekte einbringen und den Diskurs erweitern. Ob dieses Ziel erreicht wurde, kann ich nicht beurteilen, aber es war für mich stets eine persönliche Bereicherung an einem Text zu arbeiten. Ich hoffe, dass der ein oder die andere die wenigen Minuten Lesezeit als sinnvoll investierte Zeit betrachtet hat. In diesem Sinne wünsche ich dem Red Stack Magazin weiterhin viel Erfolg und viele interessierte Leserinnen und Leser.



GÜNTHER STÜRNER

# European NetSuite User Days



## IN NÜRNBERG

[netsuite.doag.org](https://netsuite.doag.org)

DOAG



# Flows for APEX 23.1

Moritz Klein, MT GmbH

Flows for APEX ist mittlerweile seit circa 3 Jahren öffentlich verfügbar. Die aktuelle Version 23.1 enthält wieder mehrere spannende Funktionen, von denen einige im folgenden Artikel nach einer kurzen allgemeinen Einführung erklärt werden.

## Was ist Flow for APEX?

Flows for APEX ist eine BPMN 2.0 konforme Workflow Engine mit hoher Integration in Oracle APEX. Alle Funktionen sind mit den Fähigkeiten der Oracle-Datenbank abgebildet, es muss also kein zusätzlicher Server betrieben werden. Die Software steht unter MIT-Lizenz auf GitHub für jeden zur Verfügung und kann ohne weitere Lizenz-

kosten direkt verwendet werden. Wer sich vorab nochmal grundlegend informieren will, dem empfehle ich folgende Ressourcen:

- Die offizielle Webseite von Flows for APEX [flowsforapex.org](https://flowsforapex.org)
- Den Red-Stack-Artikel „Flows for APEX“ in Ausgabe 06/2022 [1]
- Den "Flows for APEX"-YouTube-Kanal [youtube.com/flowsforapex](https://youtube.com/flowsforapex)

## Wer steckt eigentlich hinter Flows for APEX?

Flows for APEX wird durch eine Gruppe von sechs Enthusiasten weiterentwickelt und betreut. Alle Mitglieder des Kernteams sind langjährig mit Oracle-Technologie verbunden und (zumindest mittlerweile) leidenschaftliche APEX-Entwickler. Besonders hervorzuheben ist dabei, dass die jeweiligen Arbeitgeber die

**USER TASK**

**General** ● >

**Documentation** >

**Task Type** >

**APEX Page** >

**Variable Expressions** >

**Assignment** ▾

▾ Potential Users

Expression Type

▾ Potential Groups

Expression Type

▾ Excluded Users

Expression Type

**Scheduling** ▾

▾ Priority

Expression Type

▾ Due On

Expression Type

Abbildung 1: Assignment & Scheduling (Quelle: Moritz Klein)

Arbeit an Flows for APEX unterstützen und somit auch für Stabilität in der Weiterentwicklung sorgen.

Was ist in Release 23.1 enthalten?

Flows for APEX 23.1 setzt minimal Oracle Database Server 19c und APEX 20.2 vor-

aus. Der Release beinhaltet eine Menge neuer Funktionen, sodass es nicht möglich ist, alle im vollen Umfang hier darzustellen. Als kleine Anekdote aus dem Entwicklungsprozess der initiale Plan für den Release 23.1: „Release 23.1 soll generell ein Maintenance-Release werden. Nicht zu viel neue Funktion, dafür bestehende erweitern und verbessern.“ [2] Inwiefern wir uns an diesen Plan ge-

halten haben, kann der geneigte Leser gerne mit Hilfe der folgenden Auflistung entscheiden.

- Neue Funktionen:
  - REST API auf Basis ORDS
  - Automatische Erstellung und Aggregation von Workflow-Statistiken
  - Automatische Archivierung von beendeten Prozessinstanzen
  - Task Assignments für User Tasks
  - Bind-Variablen für PL/SQL, Variable Expressions & Element Settings
  - Übersetzungen für Italienisch, Koreanisch, vereinfachtes & traditionelles Chinesisch
  - Unterstützung von Message Flows (experimentell)
- Verbesserungen:
  - Modernisierter BPMN Modeler
  - Einfachere Anzeige und Modellierung von Subprozessen
  - Fälligkeiten & Prioritäten für Prozesse & User Tasks
  - Debug Log für den BPMN-Parser
  - Unzählige interne Optimierungen

Zum Zeitpunkt als dieser Artikel geschrieben wurde, war der Release noch nicht veröffentlicht, daher kann die finale Änderungsliste etwas unterschiedlich aussehen. Im Folgenden werden zuerst die aktuell verfügbaren „Expression Types“ und danach noch eine der neuen Funktionen detaillierter erklärt.

## Expression Types

Expression Types sind die unterschiedlichen Arten, mittels denen Prozess-Variablen und Element-Attribute gesetzt werden können. Die meisten sind durch vorhandene Optionen aus APEX inspiriert und sollten demnach vertraut sein:

- Static -> statischer Wert
- Process Variable -> Wert einer Prozess-Variable übernehmen
- SQL Query (single value) -> Abfrage, die exakt 1 Zeile zurückliefert
- SQL Query (colon delimited list) -> Mehrzeilen-Abfrage, wird mit „:“ in einem Wert zusammengefasst
- Expression -> PL/SQL-Ausdruck
- Function Body -> PL/SQL-Funktion
- Interval -> Zeitraum im ISO 8601 oder Oracle Day-To-Second-Format

- Scheduler Expression -> Zeitraum im Oracle-Scheduler-Format

## Task Assignment & Scheduling für User Tasks

Mittels der Attributgruppe „Task Assignment“ kann für User Tasks definiert werden, welche Benutzer oder Gruppen für diesen Task vorgesehen oder ausgeschlossen sind. Als weitere Attributgruppe wurde „Scheduling“ hinzugefügt. Mittels der Attribute „Priority“ und „Due On“ können die Wichtigkeit und ein Fälligkeitsdatum für die Aufgabe definiert werden. Die Attribute können durch verschiedenste der oben genannten „Expression Types“ gesetzt werden. Der große Vorteil für beide Attributgruppen ist, dass die entsprechenden Werte durch die Flows-Engine gesetzt werden und über die entsprechenden Views wie zum Beispiel FLOW\_TASK\_INBOX\_VW bereitgestellt werden. Somit liegt die Definition, wer einen Task ausüben darf, an der gleichen Stelle wie die eigentliche Task Definition und sie sind nicht mehr voneinander getrennt. Ein typischer Anwendungsfall für „Task Assignment“ ist die Zuordnung von einem oder mehreren Genehmigern für eine Anfrage. So würde man typischerweise den direkten Vorgesetzten und seine Vertreter für die Genehmigung eines Urlaubs eintragen. Wird der Urlaub jedoch durch einen Vertreter des entsprechenden Vorgesetzten beantragt, sollte dieser nicht als Genehmiger in Betracht kommen. Diesen Anwendungsfall kann man mittels Task Assignment leicht lösen:

- Potential Users -> Vorgesetzter des Antragstellers + Vertreter
- Excluded Users -> Antragssteller

Wie in den „Expression Types“ aufgelistet könnte hier etwa eine PL/SQL-Funktion für „Potential Users“ in Frage kommen und „Excluded Users“ wird per Prozess-Variable gesetzt, in welcher wir zu Anfang des Prozesses den Antragssteller zwischengespeichert haben.

## Wo kann ich mehr über Flows for APEX erfahren?

Wer mehr über Flows for APEX erfahren möchte, startet am besten auf der offiziellen Homepage [flowsforapex.org](https://flowsforapex.org). Von dort aus kann man auch weiter zum GitHub Repository finden ([github.com/flowsforapex/apex-flowsforapex](https://github.com/flowsforapex/apex-flowsforapex)). Dort können Diskussionen gestartet oder Issues erstellt werden, um Fragen, Feature Requests, oder Bugs einzustellen. Gerne kann auch der Autor unter den untenstehenden Kontaktdaten direkt kontaktiert werden. Das ganze Team freut sich über jede Rückmeldung zu Flows for APEX.

## Quellen

- [1] Red Stack Magazin 06/2022 - Niels de Bruijn (2022): Flows for APEX.
- [2] Flows for APEX-Kernteam – Während der Planung für 23.1

## Über den Autor

Moritz Klein arbeitet als Senior Principal Consultant bei der MT GmbH. Seit 2001 arbeitet er mit der Oracle-Datenbank. Ab 2004 kam dann APEX (HTMLDB) mit dem ersten öffentlichen Release dazu. Aktuell beschäftigt er sich mit Datenmodellierung, PL/SQL, APEX und natürlich Flows for APEX sowie den Möglichkeiten der Oracle Cloud.



Moritz Klein  
moritz.klein@mt-itsolutions.com



INFODAYS

# Java 21 LTS

powered by JavaSPEKTRUM

15. NOVEMBER 2023 | ONLINE-KONFERENZ

## DIE THEMEN IM ÜBERBLICK

Unsere Expert:innen erklären Ihnen in ihren praxisnahen Vorträgen die wichtigsten neuen Erkenntnisse zum neuen Java 21 LTS Release:

Was sind die **wichtigsten Features** seit Java 17?

Wo kann das **neue Pattern Matching** helfen?

Wie geht nun eine **bessere nebenläufige Programmierung** mit Java vonstatten?

Was bietet **Project Amber** noch so alles?

**Project Panama:** Wie sich künftig nativer Code besser in Java einbetten lässt?

JETZT  
ANMELDEN!



[INFODAYS.DE/JAVA-LTS](https://infodays.de/java-lts)

## SPRECHER

Wolfgang Weigend | Oracle  
Christian Schuster | mgm technology  
Falk Sippach | embarc

Merlin Bögershausen | adesso  
Dennis Makogon | Oracle



# *APEX-Performance – Irgendwie ist meine Applikation langsam*

Jörg Doppelreiter, solicon IT

Warum ist meine Applikation langsam? Mit welchen Tools kann ich meine Prozesse analysieren? Wie sind die Ergebnisse dieser Tools zu interpretieren? Ist das Problem im Front- oder Backend? Zentrale Fragen zu Beginn eines Optimierungsprozesses, zu dessen Beginn oftmals die Aussage „Es ist langsam“ steht. In diesem Artikel werden Bordinstrumente aufgezeigt, mit denen diese Fragen, im Kontext einer Oracle-APEX-Applikation, beantwortet werden sollen.



Der erste Eindruck einer Web-Applikation hat großen Einfluss auf Akzeptanz und Erfolg. Je nach Anwendungsgebiet kann bereits der erste Ladevorgang entscheidend sein. Benutzer verlassen eventuell die Seite, ohne eine weitere Aktion auszuführen, man spricht von der so genannten Bounce-Rate, der Absprungs-Rate [1].

Im Unternehmensumfeld führt eine hohe Ladezeit häufig zu Frust der Anwender und im Anschluss zu schlechter Datenqualität. Langsame Applikationen werden häufig nur im minimal notwendigen Umfang genutzt.

Es gibt im Lebenszyklus einer Applikation bestimmte Faktoren, die Lade-,

Reaktions- und Bearbeitungszeiten beeinflussen. Neue Applikationen werden prototypisch umgesetzt, besitzen einen geringen Funktionsumfang und arbeiten mit „kleinen“ Datenmengen, die schnell verarbeitet werden. Mit fortlaufender Lebensdauer werden neue Features hinzugefügt und die Datenbasis wird größer, was zur Erhöhung der genannten Zeiten führen kann.

Eine APEX-Applikation besteht aus verschiedenen Komponenten: SQL für das Laden und Bearbeiten der Daten, PL/SQL für die Programmlogik sowie HTML, CSS und JavaScript für die Präsentation. In allen Teilbereichen kann es zu einem unerwünschten Verhalten kommen. Ein simples SQL-Statement kann im Laufe der Zeit zu unerwünscht hohen Ladezeiten führen, der Browser kann mit der steigenden Menge an JavaScript und Seitenelementen überfordert werden. Beispiele lassen sich viele finden, doch wie kann man vorgehen, wenn man vor einer frustrierten Nutzerbasis steht und versucht, eine Applikation unter Zeitdruck zu optimieren oder ein bestimmtes Problem zu analysieren?

Ein wichtiger Faktor zur effizienten Optimierung ist, das Problem schnell einzugrenzen. Dazu stehen in der Datenbank, der APEX-Umgebung und auch im Browser verschiedene Bordmittel zur Verfügung.

Wie in einem binären Suchbaum kann die Ursache des Problems untersucht werden (siehe Abbildung 1). Ein weiterer Aspekt besteht in der Einschätzung, in welchem Bereich das größte Optimierungspotential steckt.

Alle Screenshots in diesem Artikel wurden mit Oracle APEX Version 23.1.2 erstellt.

## APEX Page Timing

Ein Blick in die „APEX Page Timings“ liefert erste Informationen zum Ablauf eines Prozesses. APEX visualisiert Ausführungszeiten und Abfolgen von Aktionen, die während des Ladevorgangs oder anschließend ausgeführt werden. Die Timings können über die Developer Toolbar aufgerufen werden (siehe Abbildung 2).

In den folgenden Abbildungen werden einzelne Einträge des Trackingvorgangs näher erläutert.

**Page Reload** kennzeichnet die Gesamtdauer des Ladevorgangs vom Aufruf bis zu dem Zeitpunkt, an dem die Seite vollständig im Browser visualisiert und bedienbar ist. **Request** stellt die initiale Ladezeit der Seite dar, bis die erste Response der APEX-Seite an den Browser übertragen wurde (siehe Abbildung 3). Anschließend werden APEX-spezifische Dateien (CSS/JS) geladen und verarbeitet (siehe Abbildung 4).

**Processing** steht für Aktionen nach der initialen Antwort, die im Frontend-Bereich der Applikation ausgeführt werden (JavaScript). Im weiteren Verlauf des Ladevorgangs findet sich das Event **DOM Content Loaded** (siehe Abbildung 5). Der DOM-Baum wird auf Basis der Response erstellt.

Das folgende (gekürzte) Beispiel in Abbildung 6 bildet den gesamten Ladevorgang ab.

Der Seitenladevorgang wurde mit einer Gesamtdauer von sechs Sekunden gemessen. Aus der ersten Analyse dieses Vorgangs können bereits zwei Erkenntnisse abgeleitet werden:

- **Request:** Der initiale Ladevorgang innerhalb der Datenbank benötigt mehr als drei Sekunden. Eine weitere Analyse kann durch APEX Debug oder mittels SQL-Trace erfolgen.
- **Processing:** Nach der Response startet ein Processing-Block mit einer Gesamtzeit von 3 Sekunden. Der nächste Schritt der Analyse kann über die Browser DevTools erfolgen.

Im nachfolgenden Abschnitt wird zunächst der clientseitige Teil des Problems mit Hilfe der Browser-Entwicklerwerkzeuge weiter analysiert.

## Browser DevTools

Ein nützlicher Werkzeugkasten für Web-Applikationsentwickler, der stetig erweitert wird: die Entwicklertools der Browser!

Je nach Browser können diese unterschiedlich aussehen sowie im Umfang variieren, jedoch hat nahezu jeder Hersteller bestimmte Tools in den Browser inkludiert. Die folgenden Screenshots wurden unter Verwendung des Google Chrome Browsers (Version

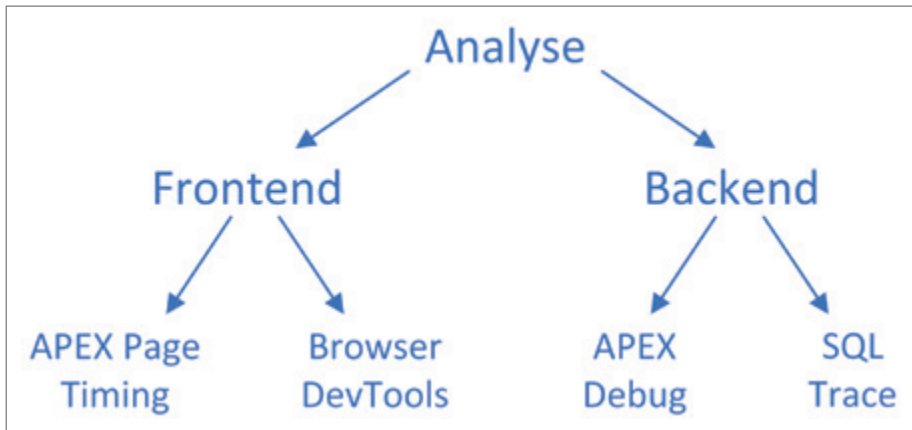


Abbildung 1: Analysebaum (Quelle: Jörg Doppelreiter)

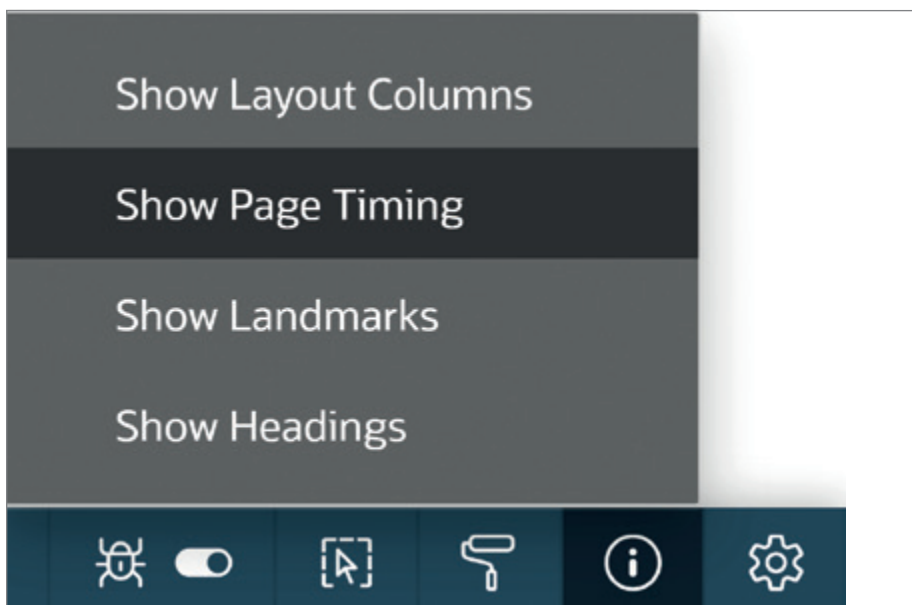


Abbildung 2: APEX Page Timing – Developer Toolbar (Quelle: Jörg Doppelreiter)

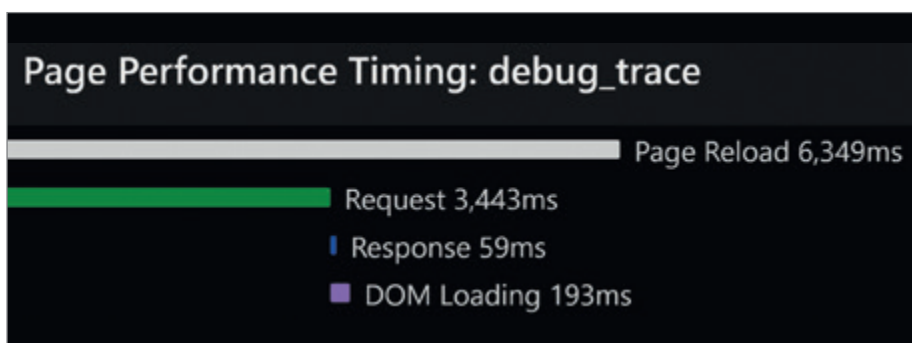


Abbildung 3: APEX Page Timing – Page Reload/Request (Quelle: Jörg Doppelreiter)

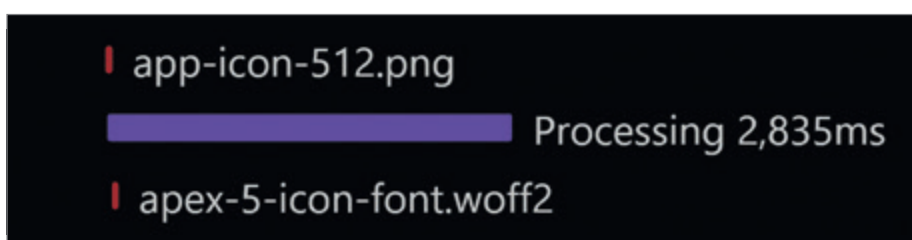


Abbildung 4: APEX Page Timing – Processing (Quelle: Jörg Doppelreiter)

115.0.5790.110) erstellt. Das Öffnen der Entwicklerwerkzeuge kann auf verschiedene Arten, beispielsweise über das Options-Menü oder durch den Shortcut F12, erfolgen (siehe Abbildung 7).

Ein elementares Werkzeug ist der so genannte „Leistung“-Tab. Hier lässt sich der Ladevorgang der Seite oder einer anderen Aktion, wie einem Ajax-Call, analysieren. Um hier erste Informationen zu erhalten, muss nicht jedes Detail betrachtet beziehungsweise genau verstanden werden. Ein guter Einstiegspunkt ist die Funktion „Profilerstellung starten und Seite aktualisieren“ (Ctrl-Shift-E) (siehe Abbildung 8).

Der Browser lädt die aktuelle Seite neu und zeichnet den gesamten Ladevorgang auf. Um die Analyse-Möglichkeiten des Browsers bestmöglich zu nutzen, kann für die APEX-Seite zuvor der Debug-Modus aktiviert werden. Durch diese Aktivierung wird von sämtlichen APEX-Sourcen (CSS/JS) die „Unminified“-Version geladen. Das bedeutet höhere Ladezeiten durch größere Datenmengen, bietet jedoch Vorteile für die weitere Auswertung. Drei der zahlreichen Informationsblöcke werden hier nun näher betrachtet: das **CPU-Chart**, die **Zusammenfassung** sowie die **Von-unten-nach-oben-Analyse** (siehe Abbildung 9).

Das **CPU-Chart** bietet einen guten Überblick. Der weiße/grau Bereich zu Beginn stellt die „Wartezeit“ des Browsers auf die Antwort (Response) dar. Der orange Scripting-Block deutet auf eine zeit- und kostenintensive JavaScript-Ausführung hin.

**Achtung:** Die Profilerstellung beeinflusst die APEX Page Timings in bestimmten Bereichen wie der Ausführung von JavaScript. Ein erneuter Blick in die Page-Timings, nach einer Profilerstellung, zeigt für denselben Ladevorgang nun eine **Processing-Zeit** von 7,9 Sekunden an!

Die **Zusammenfassung** innerhalb des Leistung-Tabs bietet aggregierte Informationen (siehe Abbildung 10).

Der **Von-unten-nach-oben**-Bereich ermöglicht eine Überprüfung der „Top-Consumer“ der Browser-Prozesse (siehe Abbildung 11).

In bestimmten Fällen gibt es oftmals eine geringe Anzahl an Aktionen, die den größten Teil der Zeit benötigen. Durch die Aktivierung des APEX-Debug-Modus kann

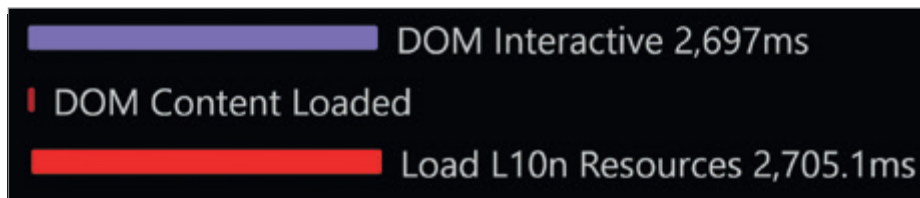


Abbildung 5: APEX Page Timing – DOM Content Loaded (Quelle: Jörg Doppelreiter)

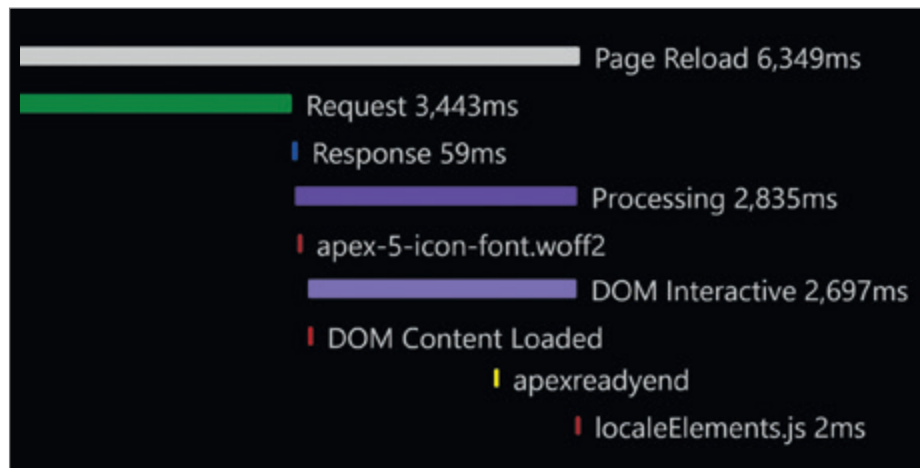


Abbildung 6: APEX Page Timing – gekürzter Ablauf (Quelle: Jörg Doppelreiter)

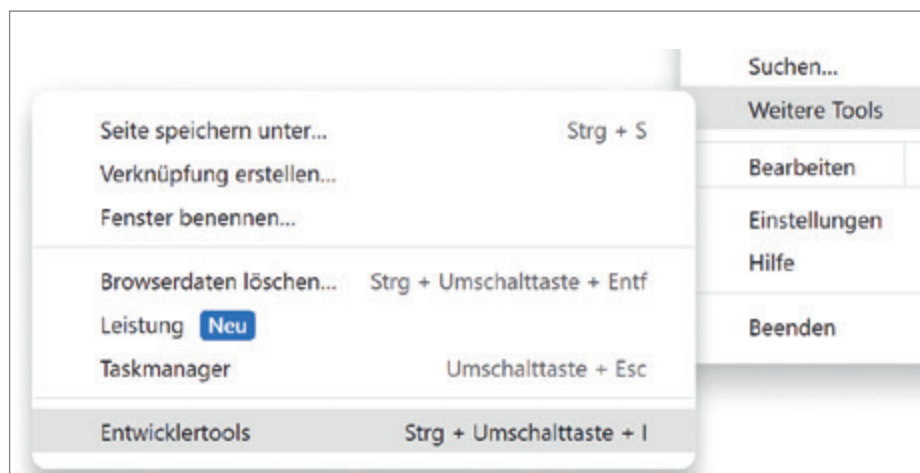


Abbildung 7: Chrome-Entwicklertools (Quelle: Google Chrome)

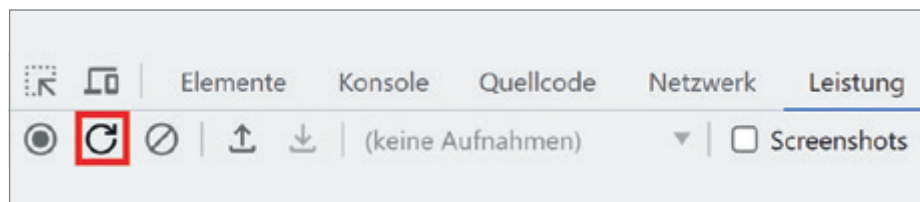


Abbildung 8: Chrome Performance – Start Profilerstellung (Quelle: Google Chrome)

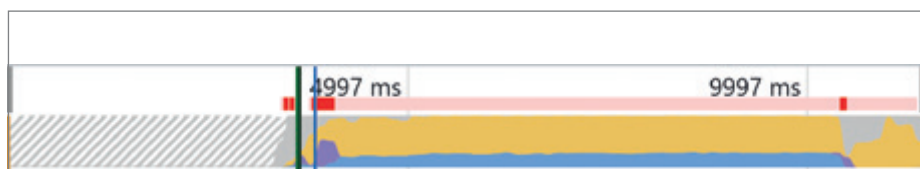


Abbildung 9: Chrome Leistung – CPU (Quelle: Google Chrome)

der Browser eine weiterführende Analyse innerhalb der JavaScript-Bibliotheken durchführen. Chrome bietet hier bereits einen Link in der entsprechenden Zeile an. Durch diesen gelangt man direkt an die Stelle im Code, in der die meiste Zeit konsumiert wird (siehe Abbildung 12).

Das JavaScript-Fragment zeigt in diesem Fall eine „After Refresh“-Dynamic-Action als Hauptverursacher des Problems. In weiterer Folge kann mit der Optimierung der Funktion begonnen werden.

Für die Problemlösung können verschiedene Ansätze gewählt werden, unter anderem:

- **Rework** der JavaScript-Logik: Optimieren der Funktionslogik.
- **Redesign** Lade-Prozess: Änderung des Ladeprozesses beziehungsweise Einsatz anderer Methoden/Technologien.
- **Async/Defer**: Entfernung des Prozesses aus dem Hauptthread des Browsers durch Einsatz von JavaScript-Technologien Async beziehungsweise Defer. Es muss jedoch darauf Rücksicht genommen werden, dass die Ergebnisse/Aktionen zu einem bestimmten Zeitpunkt eventuell noch nicht abgeschlossen sind. Google bietet einen Überblick verschiedener Möglichkeiten in einem spezifischen Artikel zu diesem Thema, der über Shortcuts aus dem Chrome-Lighthouse-Bericht abrufbar ist: „Minimize main thread work“ [2].

### APEX Debug/Trace

Für die Analyse von Datenbank-Prozessen, bietet Oracle APEX ein eigenes Debugging-Werkzeug. Es gibt verschiedene Möglichkeiten der Aktivierung:

- APEX Developer Toolbar
- URL – Parameter
- APEX-Session: Aktivierung durch Workspace Admin
- APEX – Session: Aktivierung durch API-Call (APEX\_SESSION.SET\_DEBUG)

Nach der Aktivierung können entsprechende Log-Information eingesehen werden. APEX unterscheidet verschiedene Log-Stufen [3]. In Abbildung 13 sieht man einen Auszug der verfügbaren Level.

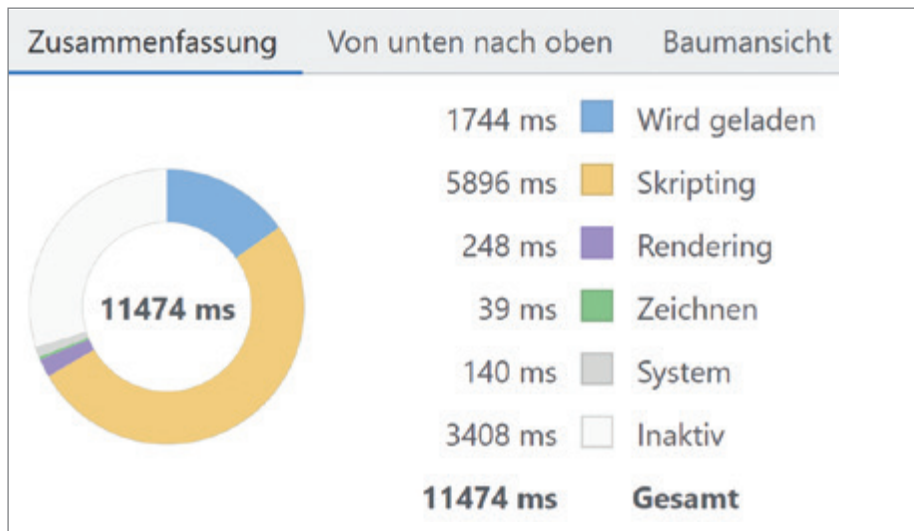


Abbildung 10: Chrome-Leistung – Zusammenfassung (Quelle: Google Chrome)

Zeit für Einzelakti...	Gesamtzeit	Aktivität
1724.0 ms 444.8 %	1858.1 ms 479.4 %	▶ HTML parsen
1248.9 ms 322.2 %	6323.3 ms 1631.4 %	▶ javascriptFunction debug_trace?debug=LEVEL...
707.1 ms 182.4 %	1487.9 ms 383.9 %	▶ buildFragment jquery-3.6.4.js?v=23.1.2:5042:23

Abbildung 11: Chrome-Leistung – Von unten nach oben (Quelle: Google Chrome)

Liniennummer	Dauer (ms)	Code
715	1.0 ms	{ "name": "After Refresh - JS
716		let row = \$("#DEVTOOLS_REPORT
717	0.4 ms	for(let i=0; i<10000; i++) {
718	3.6 ms	\$(row).after('<tr class="a
719	686.2 ms	console.log('row added');
720	5.4 ms	\$("#DEVTOOLS_REPORT tr.add
721		}

Abbildung 12: Chrome-Leistung – Quellcode-Ausführungszeiten (Quelle: Google Chrome)

```

c_log_level_info constant t_log_level := 4;
  -- default level if debugging is enabled
  -- (for example, used by apex_application.debug)
c_log_level_app_enter constant t_log_level := 5;
  -- application: messages when procedures/function
c_log_level_app_trace constant t_log_level := 6;
  -- application: other messages within procedures,
c_log_level_engine_enter constant t_log_level := 8;
  -- APEX engine: messages when procedures/function
c_log_level_engine_trace constant t_log_level := 9;
    
```

Abbildung 13: Oracle APEX Log Level (Auszug) (Quelle: Jörg Doppelreiter)

Für eine initiale Analyse eignet sich Log Level 4. Wird das höchste Log-Level (9) aktiviert, verlängert sich oftmals die Ausführungszeit in großem Umfang durch die Steigerung der Log-Einträge. Es werden jedoch weitere Informationen für die Analyse bestimmter Situationen ausgegeben. Darunter zählen bei Ausführungen von Select-Statement die **sql\_id** sowie die **child number** der Ausführung. Gleichzeitig werden die Debug-Outputs mit Information aus DBMS\_XPLAN (.DISPLAY\_CURSOR) angereichert (siehe Abbildung 14).

Es handelt sich hier zunächst nicht um die tatsächliche Ausführung des Statements, sondern um die „Einschätzung“ des Optimizers. Für eine genauere Analyse eignet sich der Oracle Hint **/\*+ gather\_plan\_statistics \*/**, welcher in der SQL-Abfrage hinzugefügt werden kann. Ab diesem Zeitpunkt werden weitere Informationen zur Ausführung gesammelt (siehe Abbildung 15).

Eine Ausführung und Überprüfung der Debug-Ergebnisse retourniert nun eine neue **sql\_id** beziehungsweise **child-number**.

Die zusätzlich gesammelten Informationen können über einen entsprechend berechtigten User der Datenbank mit Hilfe des Packages DBMS\_XPLAN eingesehen werden (siehe Listing 1 und Abbildung 16).

Durch Angabe eines zusätzlichen Parameters mit dem Wert **ALLSTATS LAST** wird der tatsächliche Ausführungsplan des Statements, inklusive Estimated- (E) und Actual-Rows (A), ausgegeben. Diese Ausgabe bietet einen ersten Ansatzpunkt für die mögliche Optimierung des Statements und kann selbst ohne Oracle-Datenbank-Tracing aus APEX extrahiert werden. In dem angeführten Beispiel sticht eine große Diskrepanz zwischen Estimated- und Actual-Rows heraus.

Nicht alle Situationen können über die APEX-Debug-Möglichkeiten ausreichend untersucht werden. Ein Beispiel hierfür können PL/SQL-Funktionen darstellen, die in verschiedenen Bereichen eines Statements verwendet werden. Durch den Ausführungsplan des Statements kann unter Umständen das Hauptproblem, der Aufruf der Funktion, ausgemacht werden, jedoch nicht die Ursache innerhalb der Funktion. Dafür benötigt es eine Analyse des Funktionscodes beziehungsweise den Einsatz der Datenbank-Trace-Möglichkeiten.

Elapsed	Execution	Message
		SQL_ID 0wbn5nfnjty04, child number 1 ----- select /*+qb_name(apex\$2850_1120)*/ from(select a.*,row_number()over(order by null)apx\$rownum from(select "PRODUCT_ID","PRODUCT_NAME",SUM("DEVIATION_AVG")

Abbildung 14: APEX Debug – Suche sql\_id (Quelle: Jörg Doppelreiter)

**Source**

Location: Local Database

Type: SQL Query

SQL Query

```
select /*+ gather_plan_statistics */
o.order_id
, o.order_date
, oi.product_id
```

Abbildung 15: APEX Debug – Hint gather\_plan\_statistics (Quelle: Jörg Doppelreiter)

Id	Operation	E-Rows	A-Rows	A-Time
0	SELECT STATEMENT		51	00:00:27.78
* 1	VIEW	664	51	00:00:27.78
* 2	WINDOW NOSORT STOPKEY	664	51	00:00:27.78
3	VIEW	664	51	00:00:27.78
4	WINDOW BUFFER	664	51	00:00:27.78
5	HASH GROUP BY	664	288	00:00:27.78
6	VIEW	664	25643	00:00:27.72
7	WINDOW BUFFER	664	25643	00:00:00.09

Abbildung 16: DBMS\_XPLAN – Auszug Ausführungsplan (Quelle: Jörg Doppelreiter)

Elapsed	Execution	Message																																																	
2.18200	0.00500	<table border="1"> <thead> <tr> <th>Id</th> <th>Operation</th> <th>Name</th> <th>Rows</th> <th>Bytes</th> <th>Cost (%CPU)</th> <th>Time</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SELECT STATEMENT</td> <td></td> <td></td> <td></td> <td>4 (100)</td> <td></td> </tr> <tr> <td>* 1</td> <td>VIEW</td> <td></td> <td>2</td> <td>78</td> <td>4 (25)</td> <td>00:00:01</td> </tr> <tr> <td>* 2</td> <td>WINDOW NOSORT STOPKEY</td> <td></td> <td>2</td> <td>52</td> <td>4 (25)</td> <td>00:00:01</td> </tr> <tr> <td>3</td> <td>VIEW</td> <td></td> <td>2</td> <td>52</td> <td>4 (25)</td> <td>00:00:01</td> </tr> <tr> <td>4</td> <td>SORT ORDER BY</td> <td></td> <td>2</td> <td>6</td> <td>4 (25)</td> <td>00:00:01</td> </tr> <tr> <td>5</td> <td>TABLE ACCESS FULL</td> <td>MY_TABLE</td> <td>2</td> <td>6</td> <td>3 (0)</td> <td>00:00:01</td> </tr> </tbody> </table>	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	0	SELECT STATEMENT				4 (100)		* 1	VIEW		2	78	4 (25)	00:00:01	* 2	WINDOW NOSORT STOPKEY		2	52	4 (25)	00:00:01	3	VIEW		2	52	4 (25)	00:00:01	4	SORT ORDER BY		2	6	4 (25)	00:00:01	5	TABLE ACCESS FULL	MY_TABLE	2	6	3 (0)	00:00:01
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time																																													
0	SELECT STATEMENT				4 (100)																																														
* 1	VIEW		2	78	4 (25)	00:00:01																																													
* 2	WINDOW NOSORT STOPKEY		2	52	4 (25)	00:00:01																																													
3	VIEW		2	52	4 (25)	00:00:01																																													
4	SORT ORDER BY		2	6	4 (25)	00:00:01																																													
5	TABLE ACCESS FULL	MY_TABLE	2	6	3 (0)	00:00:01																																													

Abbildung 17: DBMS\_XPLAN – Ausführungsplan my\_view (Quelle: Jörg Doppelreiter)

## SQL-TRACE

Die SQL-Trace-Umgebung generiert Statistiken für alle SQL-Statements innerhalb einer Datenbanksession.

Dazu zählen unter anderem die Ausführungsanzahl von Parse-, Execute-

und Fetch- Operationen sowie die damit verbundene Ausführungszeit und Anzahl der verarbeiteten Zeilen [4]. SQL-Trace kann über Prozeduren im Package DBMS\_MONITOR für die Oracle-Datenbank-Session aktiviert und deaktiviert werden (siehe Listing 2).

Diese Funktionen sind hilfreich, wenn die Ursache bereits auf eine bestimmte Aktion eingegrenzt werden konnte. Weitere Möglichkeiten bestehen darin, das Tracing für die gesamte Datenbank oder einzelne APEX-Sessions zu aktivieren.

Trace-Files werden auf dem Datenbank-Server abgelegt und können in das TKPROF-Format überführt werden. Dabei werden Informationen aus dem Trace-File zusammengeführt. Die Ausgabe enthält jedes ausgeführte Statement sowie die Anzahl der Ausführungen, der verbrauchten Ressourcen und verarbeiteten Zeilen [5].

Das nachfolgende Beispiel soll den Einsatz von TKPROF verdeutlichen:

Für die Berechnung eines Wertes innerhalb eines SQL-Statements wird eine Funktion verwendet (siehe Listing 3).

Innerhalb der Funktion get\_calculated\_value wurde eine Loop-Anweisung eingefügt.

Die Ausführung dieser Logik ist durch die Möglichkeiten von APEX-Debug nicht ersichtlich (siehe Abbildung 17).

Der Debug-Output zeigt für die Abfrage von zwei Zeilen eine Elapsed-Time von 2,18 Sekunden. Weitere Informationen sind in diesem Beispiel noch nicht ersichtlich.

Die Aktivierung von SQL-Trace für die APEX-Session kann über das APEX-Backend (Monitor Activity → Active Sessions → Session) oder das API-Package APEX\_SESSION durchgeführt werden (siehe Abbildung 18).

Der Speicherort der Trace-Dateien kann über die Dictionary View **v\$sqldiag\_info** ausgelesen werden.

Die Dateinamen der erstellten Trace-Files beinhalten die APEX-Session-ID und sind somit leicht zuordbar (siehe Abbildung 19).

Das TKPROF-Tool überführt diese Daten in ein besser lesbares Format (siehe Abbildung 20).

Innerhalb der TKPROF-Datei befindet sich die Ausführung der eingebetteten SQL-Loop-Anweisung (siehe Abbildung 21).

Hier ist nun eine Ausführungszeit von 1,92 Sekunden sowie die Ausführungsanzahl ersichtlich. Das Tracing sollte nach der Extraktion der notwendigen Informationen umgehend wieder deaktiviert werden, um die Erzeugung großer Datenmengen zu vermeiden.

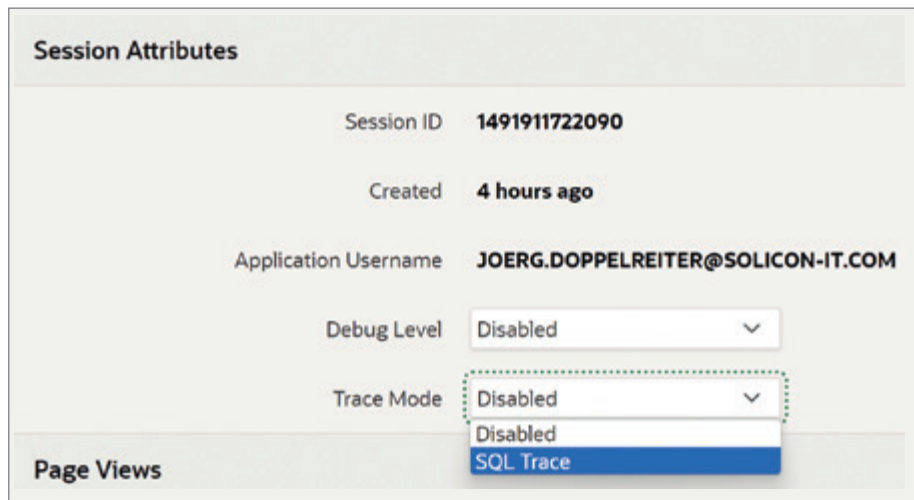


Abbildung 18: SQL-Trace – Aktivierung für eine APEX-Session (Quelle: Jörg Doppelreiter)

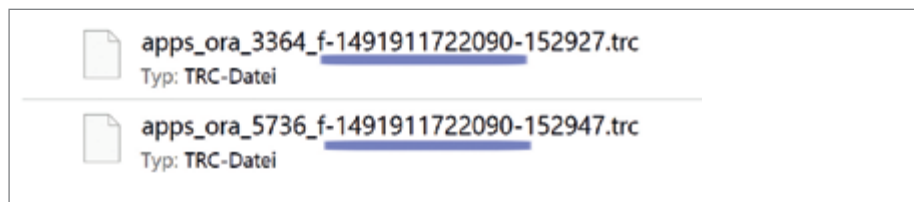


Abbildung 19: SQL-Trace – Beispieldateien (Quelle: Jörg Doppelreiter)

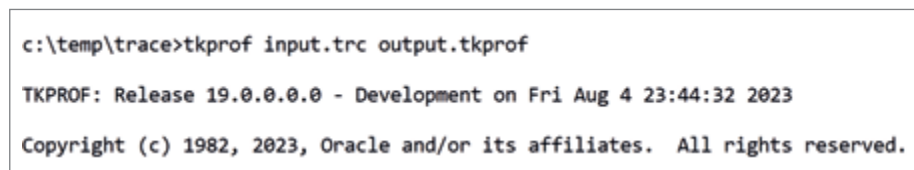


Abbildung 20: TKPROF-Aufruf (Quelle: Jörg Doppelreiter)

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	199998	1.67	1.61	0	0	0	0
Fetch	199998	0.27	0.30	0	0	0	199998
<b>total</b>	<b>399997</b>	<b>1.95</b>	<b>1.92</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>199998</b>

Abbildung 21: TKPROF-Inhalt (Quelle: Jörg Doppelreiter)

```
select *
  from table(
    dbms_xplan.display_cursor(
      sql_id => '0wbn5nfnjty04',
      cursor_child_no => 1,
      format => 'ALLSTATS LAST'
    )
  );
```

Listing 1: DBMS\_XPLAN – Aufruf Ausführungsplan

## Fazit

Unabhängig der technischen Möglichkeiten ist auch das fachliche Verständnis der Applikation oftmals ein entscheidender Faktor bei der Lösung eines Problems. Eine Optimierung ohne Kenntnisse über die Abläufe und den Zweck der Anwendung kann mit einem erhöhten Zeitaufwand beziehungsweise schlechterem Ergebnis einhergehen.

Um eine Performanceoptimierung effizient durchführen zu können, ist es wichtig, das Problem möglichst schnell zu eruieren, beziehungsweise einzugrenzen. Dazu sind Kenntnisse zur Systemumgebung erforderlich:

- Welche Tools stehen zur Verfügung: Browser DevTools/APEX Debug/SQL-Trace?
- In welchem Bereich liegt das Problem: Client- oder Serverseitig?
- Wo befindet sich das größte Optimierungspotential: Zeitaufwand für erwartbares Ergebnis?

Anhand der vorgestellten Tools kann, ohne zusätzliche Systemumgebungen, jeder Aspekt der APEX-Applikation durchleuchtet werden.

## Quellen

- [1] Google Support, 2023: Absprungrate <https://support.google.com/analytics/answer/1009409?hl=de>
- [2] Google Chrome Developers, 2023: Minimize main thread work <https://developer.chrome.com/docs/lighthouse/performance/main-thread-work-breakdown>
- [3] Oracle APEX Online Dokumentation, 2023: APEX\_DEBUG [https://docs.oracle.com/en/database/oracle/apex/23.1/aeapi/APEX\\_DEBUG.html](https://docs.oracle.com/en/database/oracle/apex/23.1/aeapi/APEX_DEBUG.html)
- [4] Oracle-Online-Dokumentation, 2023: Overview of the SQL Trace Facility <https://docs.oracle.com/en/database/oracle/oracle-database/19/tgsql/performance-application-tracing.html#GUID-374ABE48-C63F-4A7B-B80E-906A51F9246C>
- [5] Oracle-Online-Dokumentation, 2023: Overview of TKPROF <https://docs.oracle.com/en/database/oracle/oracle-database/19/tgsql/performance-application-tracing.html#GUID-A1F41137-03E2-43AD-98E4-AD49760C4C35>



```
-- Tracing aktivieren
DBMS_MONITOR.SESSION_TRACE_ENABLE (
  session_id  IN  BINARY_INTEGER DEFAULT NULL,
  serial_num  IN  BINARY_INTEGER DEFAULT NULL,
  waits       IN  BOOLEAN DEFAULT TRUE,
  binds       IN  BOOLEAN DEFAULT FALSE,
  plan_stat   IN  VARCHAR2 DEFAULT NULL);
-- Tracing deaktivieren
DBMS_MONITOR.SESSION_TRACE_DISABLE (
  session_id  IN  BINARY_INTEGER DEFAULT NULL,
  serial_num  IN  BINARY_INTEGER DEFAULT NULL);
```

Listing 2: Oracle-Dokumentation - DBMS\_MONITOR

```
create or replace view my_view
as
select id, get_calculated_value(id) as val
from my_table;
/
```

Listing 3: View-Definition my\_view

```
create or replace function get_calculated_value(p_id number)
return number
as
  l_dummy date;
begin
  for rec in ( select level
               from dual
               connect by level < 100000 )
  loop
    select sysdate into l_dummy from dual;
  end loop;
  ...
```

Listing 4: Auszug Funktion get\_calculated\_value

```
-- SQL Trace für eine APEX-Session aktivieren
begin
  apex_session.set_trace (
    p_session_id => 1491911722090,
    p_mode       => 'SQL' );
  commit;
end;
/
```

Listing 5: SQL-Trace aktivieren

```
-- SQL Trace für eine APEX-Session deaktivieren
begin
  apex_session.set_trace (
    p_session_id => 1491911722090,
    p_mode       => null );
  commit;
end;
/
```

Listing 6: SQL-Trace deaktivieren

```
SELECT VALUE FROM V$DIAG_INFO WHERE NAME = 'Diag Trace';
```

Listing 7: Speicherort für Trace-Dateien

## Über den Autor

Jörg Doppelreiter ist Senior Consultant und APEX Solution Expert für die solicon IT GmbH. Seit mehr als 10 Jahren beschäftigt er sich hauptsächlich mit der Konzeption und Umsetzung von APEX-Applikationen.



Jörg Doppelreiter  
joerg.doppelreiter@solicon-it.com



# APEX-Item-Plugin- Entwicklung für Formulare und Interactive Grids

Michael Schmid, Accenture

Plugins sind in APEX ein mächtiges Werkzeug, um modulare, wiederverwendbare und gut wartbare Komponenten zu entwickeln. Dabei erfreuen sich insbesondere Item-Plugins einer großen Beliebtheit. In diesem Artikel wird der Aufbau der Schnittstellen (Callbacks) vorgestellt, die für die Implementierung eines Item-Plugins von Bedeutung sind. Damit lassen sich Plugins erstellen, die sowohl in klassischen Einzelsatzformularen funktionieren, aber auch im Rahmen von Interactive Grids eingesetzt werden können. Zudem steht ein nützliches Beispiel-Plugin zur Erstellung von QR-Codes [1] zum Download bereit.

Mit Hilfe der Item-Plugins können wir unsere eigenen Item-Typen bauen, die sich nahezu genauso verhalten wie die eingebauten Item-Typen: Sie lassen sich unter anderem genauso auf der Seite platzieren wie üblich, können einen Item-Wert haben oder auf einer Datenbankspalte basieren und vieles mehr. Darüber hinaus besitzen Item-Plugins noch folgende vom Plugin-Entwickler festgelegte Eigenschaften:

- Einen eindeutigen internen Namen.
- Den Typ des Plugins wie „Item“, „Region“, „Process“ usw. In unserem Fall ist der Typ „Item“.
- Die Angabe, ob das Item-Plugin für die üblichen Page-Items oder (auch) innerhalb von Spalten in Interactive Grids verwendet werden kann.
- Die Definition, welche der „normalen“ Item-Attribute („Standard Attributes“) im Page Designer pflegbar sein sollen (siehe Abbildung 1).
- Bis zu 15 eigene, sogenannte „Custom Attributes“: Diese Attribute lassen sich über den Page Designer pflegen und erlauben zum Beispiel die Eingabe von Zeichenketten, Zahlen, HTML-Fragmenten, (statischen) Auswahllisten, bis hin zu SQL- und PL/SQL-Statements (siehe Abbildung 2).

- Hochgeladene, unterstützende Dateien, die automatisch bei Verwendung des Plugins in die Seite eingebunden werden. Meistens handelt es sich dabei um JavaScript- oder CSS-Dateien (siehe Abbildung 3). Diese Dateien liegen im Quelltext vor und können zu Review-Zwecken oder Ähnlichem auch betrachtet werden.
- Vier PL/SQL-Callbacks, deren Code beim Plugin (unter „Source“) oder als Routinen in der Datenbank abgelegt werden und die die Verhaltensweise des Plugins steuern.

## Die PL/SQL-Callbacks

Die Callbacks bilden den entscheidenden Kern eines Item-Plugins. Erst durch den Code der Callbacks werden die Standardbeziehungswise Customer Properties usw. mit Leben gefüllt. Wir können vier PL/SQL-Callback-Routinen angeben, die jeweils unterschiedliche Aufgaben erfüllen (siehe Abbildung 4):

1. Render-Callback
2. Metadaten-Callback
3. AJAX-Callback
4. Validation-Callback

5. Darüber hinaus ist für Item-Plugins für Interactive Grid-Spalten noch ein JavaScript-Callback interessant; dazu weiter unten mehr.

Die Dokumentation zu diesen Callbacks, ihre Funktion und die verwendeten Datentypen ist leider nicht vollständig und über eine Reihe von Stellen verstreut. So findet man dazu Informationen in der App Builder User's Guide [2] (insbesondere im Kapitel 19.2.2.2 zu den möglichen Standard-Attributen), in der Onlinehilfe im App Builder und in der API-Dokumentation der Packages APEX\_PLUGIN [3].

## Der Render-Callback

Der Render-Callback hat die Aufgabe, mit Hilfe von Aufrufen von SYS.HTP den für das Item benötigte HTML zu erzeugen. Unsere eigene Prozedur muss sich dabei exakt an den von Oracle vorgegebenen Aufbau halten (siehe Listing 1).

Die Render-Prozedur wird von APEX automatisch aufgerufen, sobald ein Item, das auf dem Plugin basiert, auf der Seite erzeugt werden soll. Dabei werden, wie auch bei den anderen drei Callbacks, zunächst in Form von zwei PL/SQL-Records,

```

procedure <name>(
  p_item in apex_plugin.t_item,
  p_plugin in apex_plugin.t_plugin,
  p_param in apex_plugin.t_item_render_param,
  p_result in out nocopy apex_plugin.t_item_render_result);

type t_item_render_param is record(
  value_set_by_controller boolean default false,
  value varchar2(32767), -- legacy
  session_state_value wwv_flow_session_state_api.t_value,
  is_readonly boolean default false,
  is_printer_friendly boolean default false);

subtype t_data_type is pls_integer range 1..11;
type t_value is record(
  data_type t_data_type,
  varchar2_value varchar2(32767),
  clob_value clob);

type t_item_render_result is record(
  is_navigable boolean default false,
  navigable_dom_id varchar2(255),
  item_rendered boolean default true);

```

Listing 1: Struktur des Render-Callbacks und darin verwendete Datentypen

```

type t_item is record(
  id number,
  name varchar2(4000)
  session_state_name varchar2(4000),
  component_type_id number,
  region_id number,
  form_region_id number,
  data_type varchar2(30),
  label varchar2(4000),
  /* weitere Properties; siehe APEX_PLUGIN package doc */
  attribute_01 varchar2(32767),
  /* attribute_02 bis attribute_14 */
  attribute_15 varchar2(32767));

type t_plugin is record(
  name varchar2(45),
  file_prefix varchar2(4000),
  attribute_01 varchar2(32767),
  /* attribute_02 bis attribute_14 */
  attribute_15 varchar2(32767));

```

Listing 2: Struktur der in jedem Callback verwendeten Record-Typen

eine Reihe von Informationen über das momentan zu erzeugende Item und seine Eigenschaften als auch Plugin-Einstellungen übergeben (siehe Listing 2).

Die Felder des Records P\_ITEM entsprechen dabei direkt den Einstellungen, die der Verwender des Plugins im Page Designer vorgenommen hat. Die Werte

eigener Custom Attributes finden sich in den generischen Feldern ATTRIBUTE\_01 bis ATTRIBUTE\_15 wieder.

Über den Record P\_PLUGIN haben wir die Möglichkeit, die applikationsweiten Attribute des Plugins auszulesen (ATTRIBUTE\_01 bis ATTRIBUTE\_15). Diese Attribute werden in der Applikation einmal

gepflegt (in den Shared Components unter Component Settings). Dies kann zum Beispiel zur zentralen Definition von Konfigurationseinstellungen wie Server-URL und so weiterverwendet werden.

Im Record P\_PARAM erhalten wir schließlich die Information, welchen Wert unser Item gerade besitzt, und, ob das Item in einer Nur-Lese-Variante gerendert wird. (Dass hier für den Wert des Items nicht mehr nur ein Legacy-VARCHAR2-Feld vorgesehen ist, sondern auch ein generischer Behälter-Record, könnte ein Hinweis darauf sein, dass Items zukünftig konsequent „richtige“ Datentypen erhalten.)

Schließlich können wir mit Hilfe des Ausgabe-Records P\_RESULT Informationen zurückgeben. Meist ist dies nicht nötig, da die Defaults passen.

Eine sehr einfache Implementierung einer Render-Prozedur zeigt Listing 3.

Bei normalen Items erscheint das durch die Render-Prozedur erzeugte HTML direkt an dem Platz, an dem das Item im Page Designer platziert wurde. In Abbildung 5 ist die entsprechende Ausgabe des QR-Code-Plugins [1] zu sehen.

Anders verhält es sich, wenn wir das Plugin bei Interactive Grid-Spalten einsetzen. Dort wird das erzeugte HTML erst dann in der Zelle angezeigt, wenn sich die Zelle im Edit-Modus befindet, das heißt, wenn der Benutzer den Zellenwert verändern möchte. Wie die Zelle ansonsten dargestellt wird, werden wir weiter unten beim JavaScript-Callback betrachten.

## Der Metadaten-Callback

Der Metadaten-Callback ist optional und wird von APEX ausschließlich dann aufgerufen, wenn unser Plugin in Interactive Grid-Spalten verwendet wird. Die erforderliche Struktur wird in Listing 4 gezeigt.

Man erhält wieder die üblichen Informationen über das Item, beziehungsweise die Spalte, und das Plugin. Der Parameter P\_PARAM ist im Moment nur ein Dummy. Von Interesse sind dabei innerhalb des Rückgabe-Record P\_RESULT die folgenden Felder:

- ESCAPE\_OUTPUT  
Gibt an, ob APEX den vom JavaScript-Callback (siehe weiter unten) erzeugten Anzeigetext escapen soll. Meist ist dies

**Supported for**

Component Type  Page Items  Interactive Grid Columns

---

**Standard Attributes**

Attributes:

<input checked="" type="checkbox"/> Is Visible Widget	<input checked="" type="checkbox"/> Standard Form Element	<input checked="" type="checkbox"/> Session State Changeable	<input checked="" type="checkbox"/> Has Read Only Attribute
<input type="checkbox"/> Has Escape Special Characters Attribute	<input checked="" type="checkbox"/> Has Quick Pick Attributes	<input checked="" type="checkbox"/> Has Source Attributes	<input type="checkbox"/> Format Mask Date Only
<input type="checkbox"/> Format Mask Number Only	<input checked="" type="checkbox"/> Has Element Attributes	<input checked="" type="checkbox"/> Has Width Attributes	<input checked="" type="checkbox"/> Has Height Attribute
<input type="checkbox"/> Has Element Option Attribute	<input checked="" type="checkbox"/> Has Placeholder Attribute	<input type="checkbox"/> Has Icon Attribute	<input checked="" type="checkbox"/> Has Encrypt Session State Attribute
<input type="checkbox"/> Has List of Values	<input type="checkbox"/> Has LOV Display Null Attributes	<input type="checkbox"/> Has Cascading LOV Attributes	<input type="checkbox"/> Join LOV for Column Display
<input type="checkbox"/> Has Filter	<input type="checkbox"/> Has Link	<input type="checkbox"/> Has "Initialization JavaScript Code" Attribute	<input type="checkbox"/> Session State Supports CLOB

Abbildung 1: Die einstellbaren Standard-Attribute des QR-Code-Item-Plugins (Quelle: Michael Schmid)

**Custom Attributes** Add Attribute

Substitute Attribute Values

Label	Scope ↑	Attribute	Sequence	Type	Required	Default Value	Depending on
Output Format	Component	1	10	Select List	Yes	SVG	
Error Correction	Component	2	20	Select List	Yes	M	
Cell Size	Component	3	30	Integer	No	2	
Margin	Component	4	40	Integer	No	8	
Before HTML	Component	5	50	HTML	No		
After HTML	Component	6	60	HTML	No		

Abbildung 2: Einige Custom Attributes des QR-Code-Item-Plugins (Quelle: Michael Schmid)

**Files** Delete All Files Download as Zip Create File

File Prefix

Name ↑	Mime Type	File Size	Reference	File	Updated	Updated By
qr-plugin.css	text/css	956	#PLUGIN_FILES#qr-plugin.css	Download	7 weeks ago	apex
qr-plugin.js	text/javascript	22KB	#PLUGIN_FILES#qr-plugin#MIN#.js	Download	5 weeks ago	apex
qr-plugin.min.js	text/javascript	19KB	#PLUGIN_FILES#qr-plugin#MIN#.js	Download	5 weeks ago	apex

Abbildung 3: Die unterstützenden JavaScript- und CSS-Dateien des QR-Code-Item-Plugins (Quelle: Michael Schmid)

**Callbacks**

Render Procedure/Function Name

Meta Data Procedure Name

AJAX Procedure/Function Name

Validation Procedure Name

Abbildung 4: Die möglichen PL/SQL-Callbacks eines Item-Plugins (Quelle: Michael Schmid)

**Simple Form on EMP**

Ename  
SMITH

Job  
CLERK




Abbildung 5: HTML-Ausgabe des QR-Code-Plugins mit Input-Element zur Eingabe des Wertes und Darstellung des entsprechenden QR-Codes als SVG (Quelle: Michael Schmid)




		Search: All Text Columns	Go	Actions ▾	Edit	Save	Add Row
<input type="checkbox"/>	☰	Ename	Job	Mgr	VCard	Hiredate	
<input type="checkbox"/>	☰	CLARK	MANAGER2 	7839		6/9/1981	
<input checked="" type="checkbox"/>	☰	KING	PRESIDENT			11/17/1981	

Abbildung 6: HTML-Ausgabe des QR-Code-Plugins mit Input-Element zur Eingabe des Wertes und Darstellung des entsprechenden QR-Codes als SVG (Quelle: Michael Schmid)

nicht wünschenswert, da man bewusst HTML zur Anzeige erzeugt und nicht nur einfachen Text.

- **CONTAINER\_CSS\_CLASSES**  
Eine mit Leerzeichen separierte Liste von CSS-Klassennamen, die zur Gestaltung der Zelle verwendet werden kann. In vielen Fällen ist der Metadaten-Callback so einfach wie in *Listing 5* aufgebaut (und zum Beispiel das QR-Code-Plugin bildet hierbei keine Ausnahme).

## Der AJAX-Callback

Dieser Callback ist ebenfalls optional und kann genutzt werden, um – ohne Neuladen der Seite – Daten aus der Datenbank beziehungsweise in diese zu transferieren. Die dazugehörige Datenbank-Logik wird in diesem Callback ausprogrammiert und muss den Aufbau, wie in *Listing 6* gezeigt, aufweisen.

Vorhanden sind neben den bekannten Parametern P\_ITEM und P\_PLUGIN im Moment nur die beiden Dummies P\_PARAM und P\_RESULT. Es stellt sich hier die Frage, wie unser Callback an seine Aufruf-Parameter kommt und, wie er sein Ergebnis zurückgeben soll.

Um festzustellen, was zu tun ist, kann der Callback auf Item-Werte im Session State zugreifen und auch die Werte von APEX\_APPLICATION.G\_X01, ... G\_X10

```

procedure my_render(
    p_item      in          apex_plugin.t_item,
    p_plugin    in          apex_plugin.t_plugin,
    p_param     in          apex_plugin.t_item_render_param,
    p_result    in out nocopy apex_plugin.t_item_render_result)
as
begin
    if p_param.is_readonly then /* Nur-Lese-Variante? */
        apex_plugin_util.print_hidden_if_readonly(p_item, p_param);
        htp.prn('<span id="" || p_item.name || \'_DISPLAY\' class="apex-item-display-only">' ||
            apex_escape.html(p_param.value) ||
            '</span>'); /* Zeige Wert als span an. */
    else /* Rendere einfaches Text-Input-Feld mit momentanem Wert */
        htp.prn('<input type="text" class="apex-item-text" value="" ||
            apex_escape.html_attribute(p_param.value) || \'>');
    end if;
end;

```

Listing 3: Sehr einfache, aber funktionale Render-Prozedur

```

procedure <name>(
    p_item      in          apex_plugin.t_item,
    p_plugin    in          apex_plugin.t_plugin,
    p_param     in          apex_plugin.t_item_meta_data_param,
    p_result    in out nocopy apex_plugin.t_item_meta_data_result);

type t_item_meta_data_param is record (
    dummy boolean /* wird i.M. noch nicht genutzt */);

type t_item_meta_data_result is record(
    is_multi_value          boolean default false,
    display_lov_definition varchar2(32767),
    return_display_value   boolean default true,
    escape_output          boolean default true,
    container_css_classes  varchar2(32767));

```

Listing 4: Struktur des Metadaten-Callbacks und darin verwendete Datentypen

```

procedure metadata(
    p_item      in          apex_plugin.t_item,
    p_plugin    in          apex_plugin.t_plugin,
    p_param     in          apex_plugin.t_item_meta_data_param,
    p_result    in out nocopy apex_plugin.t_item_meta_data_result)
as
begin
    p_result.escape_output := false;
end;

```

Listing 5: Üblicher Aufbau eines Metadaten-Callbacks, der das Escaping deaktiviert

auslesen. Die JavaScript-Funktion `apex.server.plugin`, die man im Browser verwendet, um die Kommunikation mit der Datenbank zu initiieren, ermöglicht es, die benötigten Werte dorthin zu übertragen [4]. Eine beispielhafte Verwendung von `apex.server.plugin` in eigenem JavaScript, bei der die Werte von zwei Seiten-Items in den Session State übertragen werden

und der Wert von `APEX_APPLICATION.G_X01` gesetzt wird, zeigt Listing 7.

Wichtig ist dabei, dass man den benötigten Wert für den AJAX-Identifizierer vorher (meist im Render-Callback) mit Hilfe der PL/SQL-Funktion `APEX_PLUGIN.GET_AJAX_IDENTIFIERER` [4] erzeugt und auf die Seite einbindet, um den Wert in JavaScript zur Verfügung zu haben.

In der Datenbank wird APEX dann unseren AJAX-Callback aufrufen. Dieser sendet seine Antwort per `SYS.HTP`-Aufrufen zurück. Als recht praktikabel hat es sich erwiesen, dabei das JSON-Format zu benutzen. Listing 8 zeigt den Code eines einfachen AJAX-Callbacks. Natürlich könnte der Callback auch umfangreiche Abfragen oder Transaktionen durchführen.

## Der Validation-Callback

Der letzte PL/SQL-Callback, den wir verwenden können, ist optional und dient zur serverseitigen Prüfung der Werte für unser Item. Dabei handelt es sich um eine implizite Prüfung, das heißt, es gibt hier kein explizites Validation-Objekt. Die Prüfung erfolgt, wenn vorhanden, automatisch im Rahmen des Page Processing. Listing 9 zeigt den Aufbau eines Validation-Callbacks.

Neben den bekannten Parametern `P_ITEM` und `P_PLUGIN` finden wir im Parameter `P_PARAM` den momentanen Wert, um ihn zu prüfen. Unser Prüfungsergebnis speichern wir im Ausgabeparameter `P_RESULT`. Dabei speichern wir bei einem positiven Ergebnis `NULL` im Feld `MESSAGE`; bei einem negativen speichern wir dort hingegen die Fehlermeldung, die dem Benutzer angezeigt werden soll. Falls eine Meldung angezeigt werden soll, können wir durch das Füllen der Felder `DISPLAY_LOCATION` und `PAGE_ITEM_NAME` festlegen, wo beziehungsweise wie die Fehlermeldung erscheinen soll. Für `DISPLAY_LOCATION` sind im Package `APEX_PLUGIN` die folgenden Konstanten definiert:

```

procedure <name>(
  p_item   in          apex_plugin.t_item,
  p_plugin in          apex_plugin.t_plugin,
  p_param  in          apex_plugin.t_item_ajax_param,
  p_result in out nocopy apex_plugin.t_item_ajax_result);

type t_item_ajax_param is record(
  dummy boolean /* wird i.M. noch nicht genutzt */ );

type t_item_ajax_result is record(
  dummy boolean /* wird i.M. noch nicht genutzt */ );

```

Listing 6: Struktur des AJAX-Callbacks und darin verwendete Datentypen

```

apex.server.plugin(ajaxIdentifier,
  { pageItems: ["P1_DEPTNO", "P1_EMPNO"], x01: myValue },
  { success: function(data) {
    /* data enthält die Serverantwort */
    console.log(data.status, data.dbsession);
  }
});

```

Listing 7: Ein Aufruf des AJAX-Callbacks mit apex.server.plugin

```

procedure ajax(
  p_item   in          apex_plugin.t_item,
  p_plugin in          apex_plugin.t_plugin,
  p_param  in          apex_plugin.t_item_ajax_param,
  p_result in out nocopy apex_plugin.t_item_ajax_result)
as
  l_json varchar2(4000 byte);
begin
  select json_object(
    'status' value upper(apex_application.g_x01),
    'dbsession' value sys_context('userenv', 'sid')
    returning varchar2)
  into l_json
  from dual;
  --
  htp.prn(l_json);
end;

```

Listing 8: Ein AJAX-Callback, der unter Verwendung eines Eingabeparameters eine JSON-Antwort erzeugt und zurückschickt

- **c\_inline\_with\_field**  
Die Meldung wird nur beim Item beziehungsweise der Zelle dargestellt.
- **c\_inline\_in\_notification**  
Die Meldung erscheint nur in der Hinweisbox (rechts oben).
- **c\_inline\_with\_field\_and\_notif**  
Die Meldung wird sowohl beim Item/Zelle als auch in der Hinweisbox angezeigt. Dies ist der Default.

Schließlich wäre es mit dem Füllen von PAGE\_ITEM\_NAME noch möglich, dass das durch die Validierung betroffene Item (um)

zuzufügen. Im Listing 10 sehen wir einen Validation-Callback, der prüft, ob der Wert unseres Items eine gerade Anzahl von Zeichen umfasst. Falls dies nicht der Fall ist, wird eine Fehlermeldung mit Nennung des Namens des Items erzeugt, die nur in der Fehler-Hinweisbox angezeigt wird.

## Der JavaScript-Callback

Insbesondere für Item-Plugins, die im Interactive Grid eingesetzt werden sollen, ist der JavaScript-Callback wichtig. Dieser ist, wie die

Bezeichnung schon vermuten lässt, nicht in PL/SQL zu implementieren, sondern in JavaScript. Strenggenommen handelt es sich nicht nur um einen Callback, den wir implementieren können, sondern um mehrere, die von APEX in bestimmten Situationen im Browser aufgerufen werden. Die JavaScript-API-Routine apex.item.create [6] erlaubt uns die Definition einer ganzen Reihe von Callbacks. Ein Beispiel ist in Listing 11 zu sehen.

Der erste Parameter (hier: itemName) enthält dabei den Namen des Items beziehungsweise der Interactive Grid-Spalte, für die wir Callbacks definieren möchten. Dann folgt ein JavaScript-Objekt, das als Eigenschaften die Callbacks enthält. Die relevanten Eigenschaftsnamen können wir der (ziemlich guten) Dokumentation von apex.item.create entnehmen. In Listing 11 sehen wir die üblichen Verdächtigen, wobei hier die JavaScript-Variablen item eine Referenz auf das Input-DOM-Objekt (zum Beispiel ein einfaches Textfeld) enthält.

Der Callback displayValueFor ist aber im Interactive Grid ganz besonders wichtig. Dieser Callback wird von APEX aufgerufen, um den Text beziehungsweise viel mehr das HTML zu erhalten, das in der Zelle dargestellt werden soll, wenn sich die Zelle nicht im Editiermodus befindet (was ja meistens der Fall ist). Und da die Darstellung fast immer vom momentanen Wert abhängt, bekommt unser Callback diesen Wert als Parameter übergeben. In Listing 11 wird eine Hilfsroutine mit dem Namen getContainer aufgerufen, die ein mehr oder weniger komplexes HTML-Objekt erzeugt. Ob es sich dabei um einen einfachen Span oder einen Table handelt, ist an sich nicht relevant. APEX erwartet als Rückgabewert einen einfachen String, deswegen wird im Listing 11 noch die Funktion outerHTML aufgerufen.

Wenn wir im Metadaten-Callback ESCAPE\_OUTPUT nicht auf false gesetzt haben, wird APEX in der zurückgegebenen Zeichenkette als HTML-Sonderzeichen escapen. Dadurch wäre keine Darstellung von HTML möglich. Da dies aber in den meisten Fällen sinnvoll und gewünscht ist, müssen wir einen Metadaten-Callback implementieren und darin ESCAPE\_OUTPUT auf false setzen – dann funktioniert es auch mit der Anzeige von HTML.

Abbildung 6 zeigt ein Interactive Grid im Editiermodus, das für die Job- und VCard-Spalte das QR-Code-Plugin [1] verwendet. Die Job-Spalte in Zeile 2 wird gerade editiert



```

procedure <name>(
  p_item in          apex_plugin.t_item,
  p_plugin in       apex_plugin.t_plugin,
  p_param in        apex_plugin.t_item_validation_param,
  p_result in out nocopy apex_plugin.t_item_validation_result);

type t_item_validation_param is record (
  value          varchar2(32767), -- Legacy
  session_state_value wwv_flow_session_state_api.t_value);

type t_item_validation_result is record(
  message          varchar2(32767),
  display_location varchar2(40),
  page_item_name   varchar2(255));

```

Listing 9: Struktur des Validation-Callbacks und darin verwendete Datentypen

```

procedure validation(
  p_item in          apex_plugin.t_item,
  p_plugin in       apex_plugin.t_plugin,
  p_param in        apex_plugin.t_item_validation_param,
  p_result in out nocopy apex_plugin.t_item_validation_result)
as
begin
  if mod(length(p_param.value), 2) != 0 /* OK? */
  then
    p_result.message := 'Error in item ' || p_item.name;
    p_result.display_location := apex_plugin.c_inline_in_notification;
  end if;
end;

```

Listing 10: Ein Validation-Callback mit Anzeige der Meldung nur in der Hinweisbox

```

const colItem = apex.item.create(itemName, {
  getValue: function() { return item.val(); },
  setValue: function(value) { item.val(value); },
  disable: function() { item.prop("disabled", true); },
  enable: function() { item.prop("disabled", false); },
  displayValueFor: function(value, state) {
    const c = getContainer(value);
    return (c.outerHTML);
  }
});

```

Listing 11: Die Definition einer Reihe von JavaScript-Callbacks

und deshalb wird hier das Input-Element angezeigt, das der Render-Callback erzeugt hat. In den restlichen Zellen der beiden Spalten wird hingegen das HTML mit dem QR-Code angezeigt, das der JavaScript-Callback `displayValueFor` zurückgegeben hat.

## Fazit

APEX bietet mit den Item-Plugins ein mächtiges, aber auch komplexes Werk-

zeug an, um individuelle, wiederverwendbare Komponenten zu implementieren. Leider ist die Dokumentation dazu über einige Stellen verstreut und zum Teil lückenhaft. Wenn man aber diese Elemente gefunden, zusammengefügt und die Lücken durch entsprechende Experimente gefüllt hat, ergibt sich ein schlüssiges Gesamtbild und das Zusammenspiel der Komponenten wird klar. Zudem kann man durch das Erforschen bestehender Plugins, wie zum Beispiel des QR-Code-Plugins, weitere Er-

kennnisse und Anregungen sammeln. Dann steht der Implementieren von eigenen Item-Plugins nichts mehr im Wege.

## Quellen

- [1] Beispiel-QR-Code-Plugin: <https://sourceforge.net/projects/qr-code-plugin/>
- [2] App Builder User's Guide, 19.2 Implementing Plug-ins: <https://docs.oracle.com/en/database/oracle/apex/23.1/htmldb/implementing-plug-ins.html>
- [3] Oracle APEX API Reference, Package APEX\_PLUGIN: [https://docs.oracle.com/en/database/oracle/apex/23.1/aeapi/APEX\\_PLUGIN.html](https://docs.oracle.com/en/database/oracle/apex/23.1/aeapi/APEX_PLUGIN.html)
- [4] Oracle APEX JavaScript API Reference, Namespace: apex.server: <https://docs.oracle.com/en/database/oracle/apex/23.1/aeajs/apex.server.html#plugin>
- [5] Oracle APEX API Reference, Package APEX\_PLUGIN: [https://docs.oracle.com/en/database/oracle/apex/23.1/aeapi/GET\\_AJAX\\_IDENTIFIER-Function.html](https://docs.oracle.com/en/database/oracle/apex/23.1/aeapi/GET_AJAX_IDENTIFIER-Function.html)
- [6] Oracle APEX JavaScript API Reference, Namespace: apex.item: <https://docs.oracle.com/en/database/oracle/apex/23.1/aeajs/apex.item.html#create>

## Über den Autor

Michael Schmid ist Associate Manager, Principal Consultant und Trainer bei der Accenture GmbH. Seit über 25 Jahren beschäftigt er sich mit Entwicklungsthemen im Oracle-Umfeld. Er unterstützt Kunden aus verschiedenen Branchen bei den Themen Requirements Engineering, Prozess- und Datenmodellierung und Datenbankdesign sowie in den Technologien SQL, PL/SQL, APEX und Web-Entwicklung.



Michael Schmid  
m.a.schmid@accenture.com



# *Daten im Wandel der Zeit: Aufbau eines Historisierungs- Frameworks*

Wolf G. Beckmann, TEAM GmbH

In der schnelllebigen Welt der Daten ist es entscheidend, nicht nur den aktuellen Zustand zu erfassen, sondern auch seine Entwicklung im Laufe der Zeit zu verstehen. Daten können sich kontinuierlich ändern und oft ist es unerlässlich, auf frühere Datenkonstellationen zurückzugreifen. Die Datenhistorisierung ermöglicht es uns, vergangene Zustände jederzeit wiederherzustellen und so wertvolle Erkenntnisse zu gewinnen. In diesem Artikel werde ich die Bedeutung dieser Herangehensweise genauer untersuchen und Methoden aufzeigen, wie Organisationen ihre Datenbestände erfolgreich historisieren können.

## Arten der Historisierung

Am Anfang der Historisierung sollte man sich erst einmal die Frage stellen, was konkret das Ziel der Historisierung ist. Geht es um die Protokollierung aller Änderungen, um die Reproduzierbarkeit von Datenzuständen und Konstellationen oder um eine organisierte Historisierung in Revisionen beziehungsweise Revisionierung?

Abhängig vom Ziel der Datenhistorisierung ist die Vorgehensweise unterschiedlich. Wenn es um die Protokollierung von Änderungen und die Reproduzierbarkeit von Datenzuständen geht, empfiehlt es sich, mit Triggern zu arbeiten. Diese erfassen automatisch jede Änderung und speichern den vorherigen Zustand ab. Dadurch wird die Rekonstruktion vergangener Zustände oder zumindest die Auflistung der Änderungen ermöglicht.

Bei einer Revisionierung hingegen wird die Historisierung gezielt durch den Benutzer aufgerufen. Dies geschieht in dem Moment, in dem eine bestimmte Revision erreicht werden soll. Hierbei werden nur bestimmte Zustände historisiert, die über mehrere Transaktionen hinweg entstanden sind.

Der Fokus dieses Artikels liegt auf der Möglichkeit, einen beliebigen Datenbestand zu reproduzieren, und das sowohl über beliebige Tabellen als auch über ganze Tabellenstrukturen hinweg.

In diesem Artikel möchte ich Schritt für Schritt anhand eines einfachen Beispiels die Konzepte hinter der automatisierten Historisierung erklären, die jeden Zustand restaurieren lässt.

Oracle bietet grundsätzlich ein Feature dafür an, das sich Flashback Query Archive nennt. Allerdings ist dieses Feature nicht in der Autonomous Database integriert. Deshalb stelle ich dar, wie grundsätzlich ein Historien-Framework aufgebaut werden kann. Das Konzept kann auch auf andere Datenbanken übertragen werden.

## Aufbau von Historien-Daten

Um dies zu veranschaulichen, nehmen wir eine typische Tabelle mit den dazugehörigen Standard-Trigger wie in *Listing 1* dargestellt.

Wenn man Tabellen historisieren möchte, benötigt man in erster Linie zwei Informationen: Den Zeitpunkt, ab dem eine Datenzeile gültig ist, und den Zeitpunkt, ab dem sie nicht mehr gültig ist. Ein weiterer Aspekt ist, dass in einer Tabelle, die historische Informationen aufnimmt, der Primärschlüssel erweitert werden muss, da er nur noch in Kombination mit dem Gültigkeitszeitraum eindeutig ist. Alternativ kann der

ehemalige Primärschlüssel entfernt oder zu einem Index umgewandelt werden, vorausgesetzt, die Daten ändern sich nicht allzu häufig.

Bei der Historisierung speichere ich typischerweise auch, welche konkrete Transaktion die Zeile geändert hat, also ob sie hinzugefügt, geändert oder gelöscht wurde.

Als Zusatzinformation kann ich auch noch speichern, wer die Aktion durch-

```
CREATE SEQUENCE hd_customers_seq;

CREATE TABLE hd_customers (
  hd_customer_id NUMBER DEFAULT ON NULL
    hd_customers_seq.NEXTVAL
    CONSTRAINT hd_customers_id_pk PRIMARY KEY,
  customer_name VARCHAR2(100 CHAR),
  contact_number VARCHAR2(15 CHAR),
  created DATE NOT NULL,
  created_by VARCHAR2(255 CHAR) NOT NULL,
  updated DATE NOT NULL,
  updated_by VARCHAR2(255 CHAR) NOT NULL
);

CREATE OR REPLACE TRIGGER hd_customers_biu
BEFORE INSERT OR UPDATE ON hd_customers
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    :new.created := SYSDATE;
    :new.created_by :=
      COALESCE(SYS_CONTEXT('APEX$SESSION', 'APP_USER')
        , USER);
  END IF;
  :new.updated := SYSDATE;
  :new.updated_by :=
    COALESCE(SYS_CONTEXT('APEX$SESSION', 'APP_USER')
      , USER);
END hd_customers_biu;
```

Listing 1: Erstellen einer Demo-Tabelle

```
CREATE TABLE hd_customers_hist (
  hd_customer_id NUMBER,
  customer_name VARCHAR2(100 CHAR),
  contact_number VARCHAR2(15 CHAR),
  created DATE,
  created_by VARCHAR2(255 CHAR),
  updated DATE,
  updated_by VARCHAR2(255 CHAR),
  hist_trans VARCHAR2(1 CHAR),
  valid_from TIMESTAMP,
  valid_until TIMESTAMP,
  invalidated_by VARCHAR2(255 CHAR),
  invalidation_trans VARCHAR2(255 CHAR)
);
```

Listing 2: Erstellen einer Tabelle für die Historisierung mit Gültigkeitszeitraum

```

CREATE OR REPLACE TRIGGER HD_CUSTOMERS_BUD_HIST
BEFORE UPDATE OR DELETE ON HD_CUSTOMERS
FOR EACH ROW
DECLARE
    v_hist_trans VARCHAR2(1);
BEGIN
    IF UPDATING THEN
        v_hist_trans := 'U';
    END IF;

    IF DELETING THEN
        v_hist_trans := 'D';
    END IF;

    INSERT INTO HD_CUSTOMERS_HIST
        (hd_customer_id, customer_name, contact_number, created,
        created_by, updated, updated_by, hist_trans, valid_from,
        valid_until, invalidated_by, invalidation_trans)
    VALUES
        (:old.hd_Customer_ID, :old.Customer_Name, :old.Contact_Number,
        :old.created, :old.created_by, :old.updated, :old.updated_by,
        v_hist_trans,
        NVL(:old.updated, :old.created),
        SYSDATE,
        NVL(SYS_CONTEXT('APEX$SESSION', 'APP_USER'), USER),
        v('APP_PAGE_ID') || ' (' || v('APP_PAGE_ALIAS') || ')');
END;
/

```

Listing 3: Erstellen eines Triggers für eine Historisierungstabelle mit Gültigkeitszeitraum

```

CREATE OR REPLACE PACKAGE TEAM_HIST AS
    PROCEDURE SET_HIST_DATE(p_date IN TIMESTAMP);
    FUNCTION GET_HIST_DATE RETURN TIMESTAMP;
END TEAM_HIST;
/

```

Listing 4: Erstellen eines Package-Headers zum Setzen und Lesen eines Gültigkeitsdatums

```

CREATE OR REPLACE PACKAGE BODY TEAM_HIST AS
    g_hist_date TIMESTAMP;

    PROCEDURE SET_HIST_DATE(p_date IN TIMESTAMP) IS
    BEGIN
        g_hist_date := p_date;
    END SET_HIST_DATE;

    FUNCTION GET_HIST_DATE RETURN TIMESTAMP IS
    BEGIN
        RETURN coalesce(g_hist_date, sysdate+1);
    END GET_HIST_DATE;

END TEAM_HIST;
/

```

Listing 5: Erstellen eines Package-Bodies zum Setzen und Lesen eines Gültigkeitsdatums

geführt hat und in welchem Rahmen die Transaktion stattgefunden hat. Um zu verhindern, dass die Applikation beim Arbeiten mit den aktuellen Daten beeinflusst wird, halte ich diese historischen Daten üblicherweise in einer eigenen Tabelle.

Somit würde die „Customers“-Tabelle für die historischen Daten wie in *Listing 2* aussehen.

Des Weiteren benötige ich einen Trigger, der genau diese historische Tabelle befüllt. Es ist anzumerken, dass der Trigger nur für Updates und Deletes zuständig ist, da ich den Create-Fall ausschließlich in den aktuellen Daten verwalte. Darauf werde ich später näher eingehen. In diesem Fall möchte ich das Framework im APEX-Umfeld einsetzen. Daher nehme ich als invalidation\_trans die APEX-Applikation und -Seite (*siehe Listing 3*).

Um die Daten aus der Tabelle unter Berücksichtigung eines bestimmten Zeitpunkts abzufragen, benötige ich einen View. Dieser View soll extern gesteuert werden können, um den Zeitpunkt festzulegen, zu dem die Daten sichtbar sein sollen.

Um das Testen zu vereinfachen, erstelle ich ein kleines Package, in dem ich eine Variable für das Datum setzen und auslesen kann (*siehe Listing 4 und 5*).

Der View sieht wie in *Listing 6* aus.

Wenn ich jetzt Beispieldaten einfüge (*siehe Listing 7*) liefert der Customers View folgende Daten zurück (*siehe Listing 8*).

Dann ändere ich bei Kunde A die Daten (*siehe Listing 9*) und lösche ihn nach einer kurzen Wartezeit gänzlich, wie in *Listing 10* dargestellt.

Jetzt liefert der Customers View folgende Daten zurück (*siehe Listing 11*).

Setze ich aber das Abfrage-Datum auf „19.07.2023 13:44:24“, ändert sich das Ergebnis, wie in *Listing 12* dargestellt.

Zu dem Zeitpunkt gab es noch nicht den Kunden C und A hatte noch die Nummer 123456789.

Das VALID\_UNTIL-Datum und die HIST\_TRANS haben sich geändert. Diese geben an, dass der Datensatz nicht mehr aktuell ist, sondern durch ein Update invalidiert wurde.

So sieht das Ergebnis kurz vor dem Delete (*siehe Listing 13*) aus.

Ich sehe, dass die CONTACT\_NUMBER 111111111 ist und dieser Satz wiederum durch ein Löschen invalidiert wurde.

```

CREATE OR REPLACE VIEW V_HD_CUSTOMERS_HIST AS
SELECT *
FROM (
  SELECT t.*,
         'A' AS hist_trans,
         NVL(t.updated, t.created) AS valid_from,
         to_date('01.01.4000','dd.mm.yyyy') AS valid_until,
         NULL AS invalidated_by,
         NULL AS invalidation_trans
  FROM HD_CUSTOMERS t
 UNION ALL
  SELECT *
  FROM HD_CUSTOMERS_HIST
)
WHERE TEAM_HIST.GET_HIST_DATE >= valid_from
      AND TEAM_HIST.GET_HIST_DATE < valid_until;

```

Listing 6: Erstellen eines Views um historische Daten anzuzeigen

```

INSERT INTO HD_CUSTOMERS (CUSTOMER_NAME, CONTACT_NUMBER)
VALUES ('Kunde A', '123456789');

INSERT INTO HD_CUSTOMERS (CUSTOMER_NAME, CONTACT_NUMBER)
VALUES ('Kunde B', '987654321');

INSERT INTO HD_CUSTOMERS (CUSTOMER_NAME, CONTACT_NUMBER)
VALUES ('Kunde C', '555555555');

```

Listing 7: Beispieldaten

CUSTOMER_NAME	CONTACT_NUMBER	HIST_TRANS	VALID_FROM	VALID_UNTIL
Kunde A	123456789	A	19.07.2023 13:44:21	01.01.4000
Kunde B	987654321	A	19.07.2023 13:44:24	01.01.4000
Kunde C	555555555	A	19.07.2023 13:44:28	01.01.4000

Listing 8: Ausgabe der Beispieldaten

```

UPDATE HD_CUSTOMERS
SET CONTACT_NUMBER = '111111111'
WHERE CUSTOMER_NAME = 'Kunde A';

```

Listing 9: Ändern der Beispieldaten

```

DELETE FROM HD_CUSTOMERS
WHERE CUSTOMER_NAME = 'Kunde A';

```

Listing 10: Ändern der Beispieldaten

CUSTOMER_NAME	CONTACT_NUMBER	HIST_TRANS	VALID_FROM	VALID_UNTI
Kunde B	987654321	A	19.07.2023 13:44:24	01.01.4000
Kunde C	555555555	A	19.07.2023 13:44:28	01.01.4000

Listing 11: Ausgabe der Beispieldaten

```

/* Der nächste Befehl muss in einer Zeile stehen!
Er passte hier nur nicht ein eine Zeile! */
exec team_hist.set_hist_date(
    to_date('19.07.2023 13:44:24','dd.mm.yyyy hh24:mi:ss')
);

```

CUSTOMER_NAME	CONTACT_NUMBER	HIST_TRANS	VALID_FROM	VALID_UNTI
Kunde A	123456789	U	19.07.2023 13:44:21	19.07.2023
Kunde B	987654321	A	19.07.2023 13:44:24	01.01.4000

Listing 12: Ausgabe der Beispieldaten

```

exec team_hist.set_hist_date(
    to_date('19.07.2023 14:07:36','dd.mm.yyyy hh24:mi:ss')
);

```

CUSTOMER_NAME	CONTACT_NUMBER	HIST_TRANS	VALID_FROM	VALID_UNTI
Kunde A	111111111	D	19.07.2023 14:07:36	19.07.2023
Kunde B	987654321	A	19.07.2023 13:44:24	01.01.4000
Kunde C	555555555	A	19.07.2023 13:44:28	01.01.4000

Listing 13: Ausgabe der Beispieldaten

```

CREATE SEQUENCE hd_transaction_seq;

CREATE TABLE hd_transaction (
    transaction_num    NUMBER,
    transaction_trans  VARCHAR2(255),
    created            DATE NOT NULL,
    created_by        VARCHAR2(255) NOT NULL,
    updated           DATE NOT NULL,
    updated_by        VARCHAR2(255) NOT NULL
);

CREATE OR REPLACE TRIGGER hd_transaction_biu
BEFORE INSERT OR UPDATE
ON hd_transaction
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        :new.created := sysdate;
        :new.created_by :=
            COALESCE(sys_context('APEX$SESSION', 'APP_USER'),
                user);
    END IF;
    :new.updated := sysdate;
    :new.updated_by :=
        COALESCE(sys_context('APEX$SESSION', 'APP_USER'),
            user);
END;

```

Listing 14: Erstellen einer Transaktionstabelle

```

ALTER TABLE HD_CUSTOMERS ADD transaction_num NUMBER;

DROP TABLE HD_CUSTOMERS_HIST;

CREATE TABLE HD_CUSTOMERS_HIST AS
SELECT t.*,
    CAST(NULL AS VARCHAR2(1)) AS hist_trans,
    CAST(NULL AS TIMESTAMP) AS valid_from,
    CAST(NULL AS TIMESTAMP) AS valid_until,
    CAST(NULL AS VARCHAR2(255)) AS invalidated_by,

```

```
CAST(NULL AS VARCHAR2(255)) AS invalidation_trans,
CAST(NULL AS NUMBER) AS until_transaction_num
FROM HD_CUSTOMERS t
WHERE 1=0;
```

Listing 15: Erweitern der Tabellen um die Transaktion

```
/* Globale Variablen */
g_transaction_id VARCHAR2(255);
g_transaction_num NUMBER;

FUNCTION GET_TRANSACTION_NUM RETURN NUMBER IS
BEGIN
IF DBMS_TRANSACTION.local_transaction_id IS NULL OR
NVL(g_transaction_id,'#*') !=
DBMS_TRANSACTION.local_transaction_id
THEN
g_transaction_id :=
DBMS_TRANSACTION.local_transaction_id;
INSERT INTO hd_transaction (transaction_num,
transaction_trans)
VALUES (hd_transaction_seq.NEXTVAL,
v('APP_PAGE_ID')
|| ' (' || v('APP_PAGE_ALIAS') || ')')
RETURNING transaction_num INTO g_transaction_num;
END IF;
RETURN g_transaction_num;
END;
```

Listing 16: Funktion zum Ermitteln der Transaktionsnummer

Es sieht gut aus, aber was passiert, wenn eine Transaktion über mehrere Zeilen oder Tabellen hinweg etwas länger dauert?

Beispielsweise verändern wir in einer Tabellenstruktur eine neue Bestellung (eine Tabelle ORDERS) und erst in der nächsten Sekunde wird eine Position (eine Tabelle ORDER\_ITEMS) gelöscht, jedoch wird beides gemeinsam committed.

Nutze ich dann das VALID\_FROM-Datum der Bestellung und setze es als Hist-Datum, würde die Position immer noch angezeigt werden.

Tatsächlich handelt es sich um ein komplexes Problem, da Oracle das Datum, an dem das Commit stattgefunden hat, nicht ohne weiteres preisgibt. Die eigentliche Frage ist, was benötigt wird. In der Praxis geht es selten um den sekundengenauen Commit-Zeitpunkt, sondern eher um konsistente Daten und dabei kann uns Oracle relativ einfach helfen. Ich kann herausfinden, welche Daten in einer Transakti-

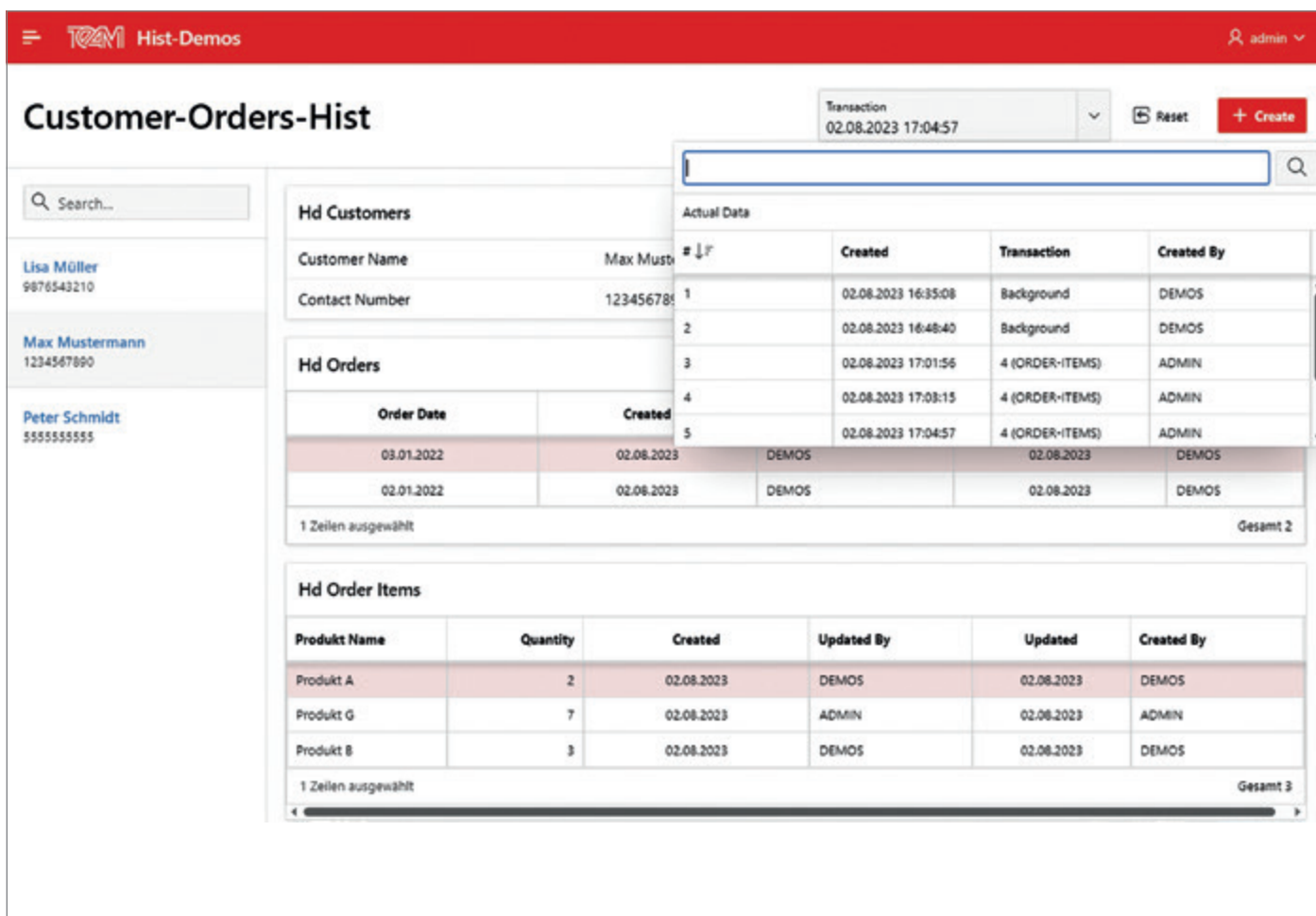


Abbildung 1: APEX-Demo mit Transaktionsauswahl (Quelle: TEAM GmbH)

```

CREATE OR REPLACE TRIGGER HD_CUSTOMERS_BIUD_HIST
BEFORE INSERT OR UPDATE OR DELETE ON HD_CUSTOMERS
FOR EACH ROW
DECLARE
    v_hist_trans VARCHAR2(1);
BEGIN

    IF INSERTING OR UPDATING THEN
        :NEW.transaction_num := team_hist.GET_TRANSACTION_NUM;
    END IF;

    IF UPDATING THEN
        v_hist_trans := 'U';
    END IF;

    IF DELETING THEN
        v_hist_trans := 'D';
    END IF;

    IF NOT INSERTING THEN
        INSERT INTO HD_CUSTOMERS_hist
            (HD_CUSTOMER_ID, CUSTOMER_NAME, CONTACT_NUMBER,
            CREATED, CREATED_BY, UPDATED, UPDATED_BY,
            TRANSACTION_NUM,
            hist_trans, valid_from, valid_until,
            invalidated_by, invalidation_trans,
            until_transaction_num)
        VALUES
            (:old.HD_CUSTOMER_ID, :old.CUSTOMER_NAME,
            :old.CONTACT_NUMBER, :old.CREATED, :old.CREATED_BY,
            :old.UPDATED, :old.UPDATED_BY,
            :old.TRANSACTION_NUM,
            v_hist_trans,
            NVL(:old.updated, :old.created),
            SYSDATE,
            NVL(SYS_CONTEXT('APEX$SESSION', 'APP_USER'), USER),
            v('APP_PAGE_ID') || ' (' || v('APP_PAGE_ALIAS')
            || ')',
            team_hist.GET_TRANSACTION_NUM);
    END IF;
END;

```

Listing 17: Trigger der Hist-Tabelle mit Transaktionsnummer

on gemeinsam bearbeitet wurden. Das funktioniert über die Funktion DBMS\_TRANSACTION.local\_transaction\_id (in PostgreSQL wäre übrigens das Äquivalent die txid\_current(), falls man das Konzept übertragen möchte). Wenn ich mich in einer Transaktion befinde, also wenn Daten in der aktuellen Session geändert wurden, wird hier eine ID gefunden, die so lange gleichbleibt, bis ein Commit erfolgt.

Um das für mich auszunutzen, gehe ich wie folgt vor:

1. Ich lege eine Sequence für Transaktionsnummern an.
2. Ich erweitere die eigentlichen Datentabellen um eine Transaktionsnummer.
3. In meinem Package erstelle ich eine Funktion, die die aktuelle Transaktionsnummer liefert. Diese zieht nur eine neue Nummer, wenn sich die local\_transaction\_id geändert hat.
4. Bei jeder Änderung speichere ich die Transaktionsnummer, mit der die Daten erzeugt wurden und mit welcher sie invalidiert wurden.
5. Um eine Übersicht über alle Transaktionen zu bekommen, legen wir noch eine Transaktionstabelle an, in der die Transaktionsnummern und das Datum, zu dem sie erzeugt wurden, hinterlegt sind

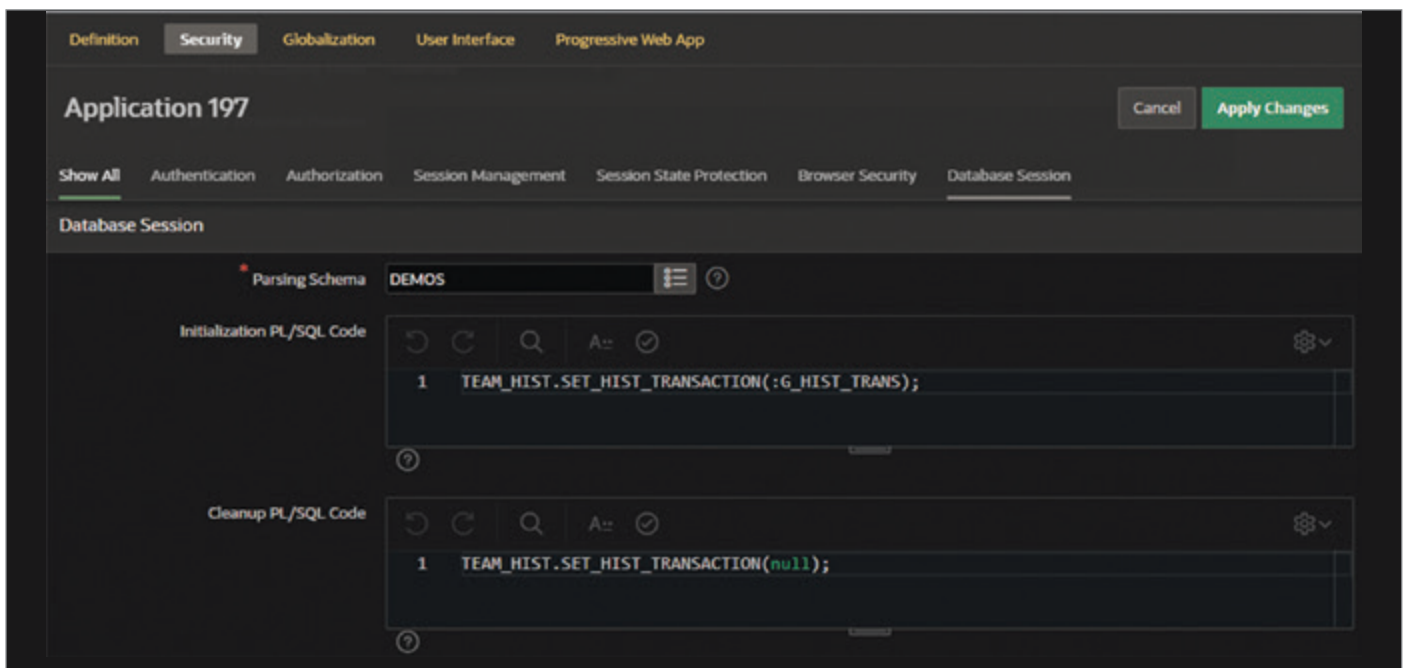


Abbildung 2: Setzen der Hist-Trans aus APEX heraus (Quelle: TEAM GmbH)



```

/* Globale Variablen */
g_hist_transaction_num NUMBER;

PROCEDURE SET_HIST_DATE(p_date IN TIMESTAMP) IS
BEGIN
  g_hist_date := p_date;
  g_hist_transaction_num := NULL;
END SET_HIST_DATE;

PROCEDURE SET_HIST_TRANSACTION(p_transaction_num IN NUMBER) IS
BEGIN
  g_hist_date := NULL;
  g_hist_transaction_num := p_transaction_num;
END SET_HIST_TRANSACTION;

FUNCTION GET_HIST_TRANSACTION_NUM RETURN NUMBER IS
BEGIN
  RETURN g_hist_transaction_num;
END;

```

Listing 18: Hist-Package um Abfrage der Transaktion erweitern

```

CREATE OR REPLACE VIEW V_HD_CUSTOMERS_HIST AS
SELECT *
FROM (
  SELECT t.*,
         'A' AS hist_trans,
         NVL(t.updated, t.created) AS valid_from,
         TO_DATE('01.01.4000', 'dd.mm.yyyy') AS valid_until,
         NULL AS invalidated_by,
         NULL AS invalidation_trans,
         1.0E125 AS until_transaction_num
  FROM HD_CUSTOMERS t
  UNION ALL
  SELECT *
  FROM HD_CUSTOMERS_HIST)
WHERE (   TEAM_HIST.GET_HIST_TRANSACTION_NUM IS NULL
        AND TEAM_HIST.GET_HIST_DATE >= valid_from
        AND TEAM_HIST.GET_HIST_DATE < valid_until)
OR (   TEAM_HIST.GET_HIST_TRANSACTION_NUM IS NOT NULL
      AND TEAM_HIST.GET_HIST_TRANSACTION_NUM >=
          transaction_num
      AND TEAM_HIST.GET_HIST_TRANSACTION_NUM <
          until_transaction_num)

```

Listing 19: View zum Abfragen historischer Werte mit Transaktionsnummer

```

BEGIN
  -- Transaktion 1
  INSERT INTO HD_CUSTOMERS (CUSTOMER_NAME, CONTACT_NUMBER)
  VALUES ('Kunde C', '555555555');
  COMMIT;
  DBMS_SESSION.SLEEP(5);

  -- Transaktion 2
  INSERT INTO HD_CUSTOMERS (CUSTOMER_NAME, CONTACT_NUMBER)
  VALUES ('Kunde A', '123456789');
  DBMS_SESSION.SLEEP(5);
  INSERT INTO HD_CUSTOMERS (CUSTOMER_NAME, CONTACT_NUMBER)
  VALUES ('Kunde B', '987654321');
  COMMIT;
END;

```

Listing 20: Erstellung der Testdaten

6. Abschließend wird der View zum Abfragen der historischen Daten um die Transaktionsnummer erweitert. So kann man nach Datum oder Transaktionsnummer abfragen.

Zuerst die Transaktionstabelle (siehe Listing 14).

Dann die Customers-Tabellen erweitern/passend erstellen (siehe Listing 15).

Dann erweitere ich das Package, um die Transaktionsnummer zu füllen (siehe Listing 16).

Und ich passe den Trigger an, der jetzt auch bei einem Insert aktiv wird, um die TRANSACTION\_NUM zu füllen (siehe Listing 17).

Zum Abfragen erweitere/ändere ich erneut das Package (siehe Listing 18).

Für die Abfragen gehe ich somit entweder über das Datum oder die Transaktionsnummer. Damit komme ich zum View (siehe Listing 19).

Jetzt teste ich das Ganze mit folgendem Script (siehe Listing 20).

Ich schaue mir die Customers an, bevor ich sie lösche (siehe Listing 21).

Wenn ich jetzt den Zustand der Datenbank über das Datum setze, passiert Folgendes (siehe Listing 22).

Tatsächlich war dieser Zustand in der Datenbank für niemanden sichtbar, da ich Kunde A und B in derselben Transaktion committet habe.

Über die Transaktionsnummer kann ich jetzt konsistente Zustände abrufen (siehe Listing 23).

Über die Hist-Views kann ich jetzt komplette Datenstrukturen genauso abfragen, wie mit den Original-Tabellen. Der typische Weg ist einen Query mit den Tabellen zu erstellen und Aliase zu verwenden. Anschließend werden in dem Query die Tabellen durch die Views ersetzt.

Durch das Setzen der Hist-Transaktionsnummer oder des Hist-Datums im Package kann ich die Datenkonstellation zu einem beliebigen historischen Zeitpunkt wiederherstellen (siehe Abbildung 1).

## Einbinden des Frameworks in APEX

In APEX besteht die Herausforderung darin, dass das Setzen einer Package Variable wenig Nutzen hat, wenn ich die Variable nicht innerhalb eines Requests setze

```
select customer_name,contact_number,transaction_num,
       to_char(valid_from,'dd.mm.yy hh24:mi:ss') valid_from
  from v_hd_customers_hist
 order by valid_from;
```

CUSTOMER_NAME	CONTACT_NUMBE	TRANSACTION_NUM	VALID_FROM
Kunde C	555555555	61	25.07.23 15:45:06
Kunde A	123456789	62	25.07.23 15:45:11
Kunde B	987654321	62	25.07.23 15:45:16

```
delete from hd_customers;
commit;
```

```
select ...
```

Keine Zeilen ausgewählt

Listing 21: Abfragen der historischen Stände

```
exec team_hist.set_hist_date(
  to_date('25.07.2023 15:45:11','dd.mm.yyyy hh24:mi:ss')
);
```

```
select * from V_HD_CUSTOMERS_HIST
```

CUSTOMER_NAME	CONTACT_NUMBE	TRANSACTION_NUM	VALID_FROM
Kunde C	555555555	61	25.07.23 15:45:06
Kunde A	123456789	62	25.07.23 15:45:11

Listing 22: Abfragen mit Zeitstempel

```
exec team_hist.set_hist_transaction(62);
```

```
select * from V_HD_CUSTOMERS_HIST
```

CUSTOMER_NAME	CONTACT_NUMBE	TRANSACTION_NUM	VALID_FROM
Kunde C	555555555	61	25.07.23 15:45:06
Kunde A	123456789	62	25.07.23 15:45:11
Kunde B	987654321	62	25.07.23 15:45:16

Listing 23: Abfragen mit Transaktionsnummer

und sofort verwende. Dies liegt daran, dass ORDS bei jedem Zugriff des Frontends auf die Datenbank eine beliebige Datenbankverbindung aus einem Pool auswählt.

Glücklicherweise bietet APEX einen Mechanismus, mit dem Sessions für einen Request vorbereitet und anschließend bereinigt werden können.

Das Vorgehen besteht daher darin, ein Application Item zu erstellen, in das die History-Transaktion eingetragen wird. Jeder Request wird dann vorbereitet und anschließend bereinigt.

Gelöst wird das ganze dadurch, dass die Hist-Transaktionsnummer oder das

Hist-Datum in ein globales Item geschrieben wird und dieses bei der Session-Behandlung verwendet wird.

Die Konfiguration für die Session-Behandlung findet man unter Shared Components > Security > Security Attributes (siehe Abbildung 2).

Damit man nicht alles selbst abtippen muss, stelle ich das Package mit der Demo-Applikation über folgendes Git-Repository zur Verfügung: [https://github.com/Team-wb/TEAM\\_HIST](https://github.com/Team-wb/TEAM_HIST).

Das Erstellen der History-Views und Trigger ist je nach Größe des Datenmodells aufwendig, kann aber hervorragend durch dynamisches SQL automatisiert

werden. Ich habe die TEAM\_HIST Package daher um Prozeduren erweitert, die automatisch die Hist-Tabelle, -Trigger und -Views erstellen.

## Transferleistung

Das Konzept der globalen Variable, die Views unterschiedlicher Daten zurückliefern lässt, ist grundlegend und kann für verschiedene Zwecke verwendet werden. Ich habe mit diesem Konzept beispielsweise eine VPD-Lösung (Virtual Privat Database) „für Arme“ entwickelt, um eine Mandantenfähigkeit umzusetzen.

## Über den Autor

Wolf G. Beckmann ist Bereichsleiter des Software- und Consulting-Teams und Consultant bei der TEAM GmbH sowie leidenschaftlicher Entwickler.



Wolf G. Beckmann  
wb@team-pb.de

# Die Oracle- Anwenderkonferenz

2023  
**DOAG**  
Konferenz + Ausstellung

21. - 24.  
Nov. 2023  
Nürnberg



Eventpartner:

**AUG**  
AUSTRIAN ORACLE USER GROUP

**SOUG**

swiss oracle  
user group

[anwenderkonferenz.doag.org](https://anwenderkonferenz.doag.org)



*„Ohne das langjährige, aktive Engagement der Community, denke ich nicht, dass es APEX heute noch gäbe.“*

Carolin Krüztmann und Arne Hattendorf unterhielten sich mit Marc Sewtz über APEX und das APEX Team, die Bedeutung der APEX Community sowie neue und bevorzugte Features.

**Das APEX Team ist in den letzten Jahren stark gewachsen. Wie viele Mitglieder umfasst das Team zu diesem Zeitpunkt?**

Insgesamt hat unser Team jetzt etwa 100 Mitarbeiter, davon arbeiten über 70 in der Entwicklung, plus Product Management, QA und Doc.

**Das starke Wachstum des Teams deutet darauf hin, dass APEX stark an Bedeutung für Oracle gewonnen hat. Wie ist es dazu gekommen?**

Unser Wachstum ist klar auf die wachsende Bedeutung von APEX bei Oracle zurückzuführen. Gründe dafür gab es mehrere. Angefangen hat es mit Oracles Cloud Services, wovon APEX

ein wichtiger Bestandteil war und ist. Zunächst als Teil des Schema Services und Exadata Express. Danach dann mit der Autonomous Database (ADB) und dem APEX Development Service. Und natürlich auch als Teil des Database Cloud Service (DBCS).

Ein weiterer Schub kam dann mit unseren Covid-Projekten, da Oracle sich gleich zu Beginn dazu verpflichtet hatte, der US-Regierung bei der Bekämpfung der Pandemie zu helfen. Und APEX auf der Oracle Cloud war dazu die ideale Plattform. Sehr schnell hatten wir eine Anwendung zur Analyse der Wirksamkeit existierender Medikamente entwickelt. Später kamen dann Anwendung für die Vaccination-Tests sowie für die Distribution der Impfstoffe hinzu. Und schließlich dann v-Safe, das jedem Einwohner in den USA zur Verfügung stand, um seine Impfung zu registrieren. Aus diesen Pandemie-Apps sind später neue Teams in Oracles General Business Units entstanden, die sich jetzt alle auf APEX standardisiert haben, und Industrielösungen in den verschiedensten Bereichen anbieten. Seit letztem Jahr liegt unser Fokus darauf, bei der Modernisierung des Gesundheitswesens im Zuge der Cerner-Übernahme zu helfen. Es gibt also genug zu tun – und APEX spielt in vielen dieser Bereiche eine zentrale Rolle.

**Bei der diesjährigen KScope-Konferenz habt ihr erstmals die Community über fünf Ideen aus der APEX Ideas-App abstimmen lassen und versichert, dass diese in die nächsten Releases einfließen werden. Warum habt ihr euch für diesen Schritt entschieden und welche Rolle spielt die Community für euch als Entwicklungsteam?**

APEX ist nur dahin gekommen, wo es heute ist, dank der APEX Community. Ich bin seit 2001 dabei und bis vor ein paar Jahren gab es von Oracle selbst so gut wie keine Unterstützung dafür, APEX besser zu vermarkten. Ohne das langjährige, aktive Engagement der Community, denke ich nicht, dass es APEX heute noch gebe. Wir wollen also an den Features arbeiten, die von der Community gewünscht werden. Aber auch mit einem jetzt sehr viel größeren Team, ist es nicht immer einfach, diese Features auch wirklich zu liefern, denn häufig haben eben die oben genannten Projekte Priorität. Daher versuchen wir auf diesem Weg, mit der Ideas-App und mit Events wie dem auf der Kscope, den Community-Ideen mehr Gewicht zu geben.

**Dieses Jahr habt ihr als APEX Team auch viele Schulungsunterlagen veröffentlicht. Kannst du uns ein paar Insights dazu geben? Wie werden sie angenommen? Wie lange hat die Produktion gedauert?**

Wir haben vor ein paar Monaten das APEX Foundations Training veröffentlicht und jetzt auch das APEX Developer Professional Training – was auch eine Zertifizierung anbietet. Beide wurden sehr gut angenommen, mit über 10.000 Teilnehmern bislang für Foundations und mehr als 5.000 für Professional. Zu dem Erfolg beigetragen hat auch, dass diese Trainings-Angebote nicht nur über Oracle, sondern auch Udemy und LinkedIn Learning verfügbar sind.

**Welches bisher veröffentlichte Feature von APEX verwendest du am häufigsten? Was ist hierfür aus deiner Sicht ein spannender Anwendungsfall?**

Am längsten verwendet habe ich sicher Classic Reports und Tabular Forms, da ich der ursprüngliche Entwickler dieser Features war. Dieser Tage verwende ich häufig Faceted Search zusammen mit Cards und Maps – und zumeist verwende ich dazu auch noch REST Data Sources. Jeder, der in den letzten Jahren eine meiner Präsentationen auf einer Konferenz oder online gesehen hat, ist jetzt sicher sehr vertraut mit dem Schulsystem in New York City: Dank Faceted Search war ich in der Lage, die öffentlich verfügbaren Schuldaten, Statistiken, Umfragen, Demographie-Infos, und viele mehr detailliert auszuwerten und die Ergebnisse für die Schulwahl meiner Kinder zu verwenden – und das Ganze in meiner Demo häufig vorzuführen.

**Welches Feature ist aus deiner Sicht das meistunter-schätzte Feature und warum sollten die Leser dieses sofort ausprobieren?**

Das sind aus meiner Sicht die Template Components. Die sind brandneu – gerade erst mit APEX 23.2 herausgekommen. Aber ich denke, wir haben noch nicht ausreichend das Potential dieser Komponenten erklärt. Template Components sind insbesondere darum so spannend, weil sie es Entwicklern ermöglichen, APEX zu erweitern und neue Daten-basierte Komponenten zu entwickeln, die dann im Create Page Wizard sowie im Page Designer neben den nativen Komponenten zur Verfügung stehen.

**Du hast in deinem Interview im Podcast „Devs On Tape“ über ein neues Feature gesprochen, welches den Entwicklungsprozess in APEX maßgeblich beeinflussen könnte. Kannst du uns darüber etwas mehr erzählen?**

Aus meiner Sicht werden die Working Copies ein „Game Changer“ für APEX werden – damit ist es Entwicklern möglich, sehr einfach eine „Working Copy“ einer Anwendung zu erstellen, das heißt, eine Kopie, auf die nur sie Zugriff haben und die sie beliebig ändern können, ohne anderen Entwicklern auf die Füße zu treten. Wenn sie dann mit ihren Änderungen fertig sind und diese sorgfältig getestet haben, können sie diese mit der Main-Anwendung vergleichen (diffen) und die Änderungen dann sauber in die Main-Anwendung mergen.



Marc Sewtz  
marc.sewtz@oracle.com



# *Datenbank-Migration in die Oracle Cloud mit dem Oracle ZDM Tool*

Marc Wagner, dbi services

Wir sprechen immer häufiger über die Migration von Datenbanken in die Oracle Cloud und Exadata-Lösungen. Aber wie kann ich meine Produktionsdatenbank auf diese Oracle-Lösungen migrieren, ohne dass mein Service durch zu lange Ausfallzeiten beeinträchtigt wird? Oracle hat ein kostenloses Tool namens ZDM entwickelt, um Datenbanken auf OCI- (Oracle Cloud Infrastructure) und Exadata-Lösungen zu migrieren. ZDM steht für Zero Downtime Migration. In diesem Artikel werden wir, nach einer kurzen Einführung in die Oracle Cloud Solutions, die Funktionalitäten dieses Tools in Bezug auf die Oracle-Datenbankmigration behandeln.

Soll ich in die Cloud migrieren oder nicht?

Der Wechsel in die Cloud bringt einige zu-

sätzliche Vorteile mit sich, wie zum Beispiel Analysefunktionen, Skalierbarkeit und Flexibilität sowie Kosteneinsparungen (wobei wir davon ausgehen können,

dass die CapEx-Einsparungen höher sind als der erwartete Anstieg der OpEx). Keine dedizierte Hardware und die damit verbundene Wartung, keine Wartung für

Patches, keine Sorgen um HA, DR oder Datensicherheit. Jedoch wird der Wechsel in die Cloud einige Ängste hervorrufen. Wir werden uns sicherlich fragen, wie die neue Performance in der Cloud sein wird und vor allem, wie hoch die Latenz sein wird. Möglicherweise haben wir auch einige sensible Daten, die wir nicht in der öffentlichen Cloud speichern möchten, da die Unternehmensrichtlinien dies nicht zulassen. Wir werden unsere neuen Kosten überwachen müssen, um sicherzustellen, dass wir nicht mehr Geld ausgeben als erwartet. Das Lock-in-Risiko könnte ein weiterer zu berücksichtigender Punkt sein: Wie hoch sind die Ausstiegskosten? Und nicht zuletzt möchten wir wissen, wie hoch die Ausfallzeiten bei der Migration sein werden. Für diesen letzten Punkt wird uns das Oracle ZDM-Tool eine gute Auskunft geben. ZDM ist ein kostenloses Tool, das ohne Lizenz verwendet werden kann, um Datenbanken auf OCI- und Exadata-Lösungen zu migrieren.

## Exadata Machine on On-Premises

Diese Lösung ist die beste Wahl, wenn es darum geht, sensible Daten zu schützen und hohe Anforderungen an Leistung und Verfügbarkeit zu erfüllen. Die Exadata wird im Rechenzentrum des Kunden installiert. Die Daten werden lokal hinter der Firewall des Kunden gespeichert. Wir verwalten alle Komponenten einschließlich der Hardware. Es sind keine Cloud Services verfügbar und die Lösung basiert auf einem kaufbasierten Modell. Die aktuelle Lösung der Exadata-Familie ist die Exadata X9M. Für Exadata X9M haben wir 2 Modelle. Das X9M-2 mit 2 Sockeln und das X9M-8 mit 8 Sockeln. Ein Half Rack X9M-8 wird beispielsweise mit 2 DB-Servern mit insgesamt 384 Kernen, 3 Storage-Servern mit insgesamt 98 Kernen, 648 TB Plattenspeicher, 77 TB Flash-Speicher und 4,5 TB persistentem Speicher geliefert. Das Oracle-Datenblatt für Exadata zeigt alle technischen Details.

## Exadata Cloud at Customer (ExaCC)

Bei dieser Lösung wird die Exadata weiterhin im Rechenzentrum des Kunden in-

stalliert. Die Daten befinden sich also weiterhin auf Kundenseite hinter der Firewall. Aber der Kunde ist nur noch für die Verwaltung der Exadata VMs (DomU) und der Datenbanken verantwortlich. Oracle wird seinerseits die Exadata HW, Firmware und Hypervisor verwalten. Im Falle eines Automomous-Database-Service-Abonnements verwaltet Oracle den gesamten Stack, also auch die VMs und die DB. Wir sind in diesem Fall nur für die Daten verantwortlich. In dieser Variante können wir von Cloud Services profitieren. Die Datenbanken werden über die Oracle Cloud zur Verfügung gestellt. Alle Daten werden verschlüsselt und sind dann für Oracle nicht sichtbar. Die Kosten basieren auf dem stündlichen OCPU-Verbrauch und dem Shape Sizing. Diese Lösung beruht auf einem abonnementbasierten Modell.

## Exadata Cloud Service (ExaCS)

Bei dieser Lösung sind die Exadata und die Daten des Kunden in der Public Cloud von Oracle installiert. Wir sind in diesem Fall, wie bei ExaCC, für die Verwaltung der Exadata VMs (DomU) und der Datenbanken verantwortlich, während Oracle seinerseits die Exadata HW, Firmware und den Hypervisor verwaltet. Im Falle des Abonnements des Automomous Database Service wird alles ähnlich wie bei ExaCC sein. Natürlich können wir auch von Cloud Services profitieren. Die Kosten basieren auf einem Pay-per-Use-Modell. Diese Lösung beruht ebenfalls auf einem abonnementbasierten Modell.

## Oracle Cloud Infrastructure

Mit der OCI können wir Datenbanken verwenden, die auf VM (Virtual Machine) oder BM (Bare Metal) laufen. Wir werden auch die Möglichkeit haben, Automomous Database zu verwenden. Dies ist eine vollständige Cloud-Lösung.

## Wie kann ich meine Datenbanken in die OCI migrieren?

Es gibt mehrere Faktoren, die zu berücksichtigen sind:

- Datenbankversion
- Betriebssystem und Version
- Zeichensatz
- Verwendungszweck der Datenbank
- Verwendete Indizes-Typen
- Datentypen in der Datenbank
- Speicherverwaltung
- Netzwerk-Bandbreite
- Dauer der möglichen Ausfallzeit
- Architektur CDB oder Nicht-CDB
- Endian-Format(ViewV\$TRANSPORTABLE\_PLATFORM)

Die OCI-Datenbank verwendet die Linux-Plattform Little Endian.

Es stehen mehrere Migrationsmethoden zur Verfügung, die wir verwenden können:

- DataPump (Export/Import, Transportable, ...)
- Remote Cloning PDB oder Non-CDB
- RMAN Cross-Platform Transportable PDB Tablespace
- RMAN Duplicate from Active Database
- RMAN Convert Transportable Tablespace
- SQL Developer
- Plug/Unplug PDB oder Non-CDB

Und natürlich das Oracle-Tool ZDM (Zero Downtime Migration).

## Einführung in Oracle ZDM

ZDM ist die bevorzugte Lösung von Oracle für die vereinfachte und automatisierte Migration von Oracle-Datenbanken. Sie bietet „null“ bis vernachlässigbare Ausfallzeiten für das Produktionssystem. Das heißt, dass während der Migration die Produktionsdatenbank nur minimal bis gar nicht beeinträchtigt wird. Dieses Tool reduziert die menschlichen Fehler. Die aktuelle Version ist Oracle ZDM 21.4.

Mit ZDM können wir zu folgenden Lösungen migrieren (und nur zu einer):

- Oracle Exadata On-Premises
- Oracle Exadata Cloud at Customer (ExaCC)
- Oracle Exadata Cloud Service (ExaCS)
- Oracle Database Cloud Infrastructure (BM and VM)
- Oracle Cloud Autonomous Database (OTP and ADW)

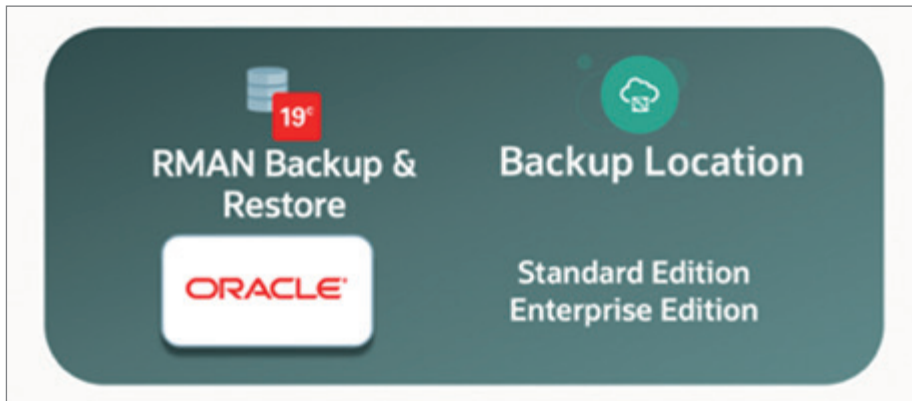


Abbildung 1: Physical Offline Migration (Quelle: Oracle)

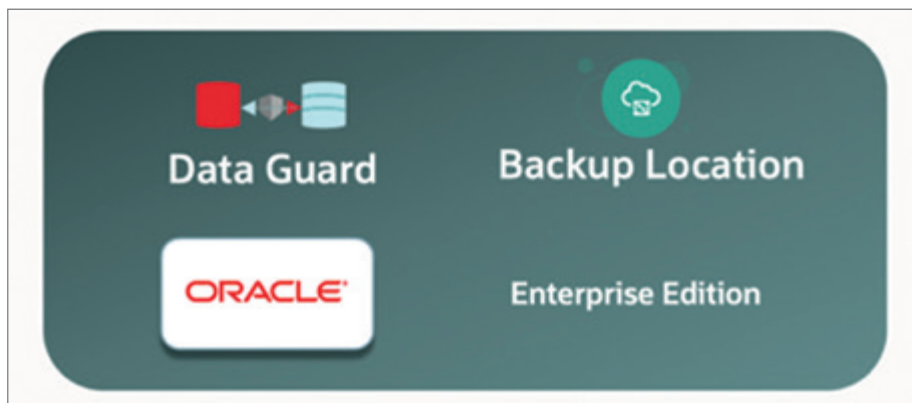


Abbildung 2: Physical Online Migration mit Backup-Verzeichnis (Quelle: Oracle)

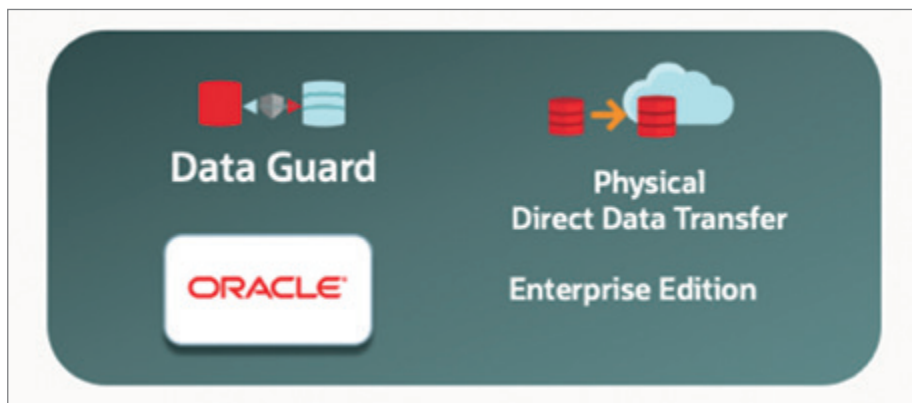


Abbildung 3: Physical Online Migration mit Direct Data Transfer (Quelle: Oracle)

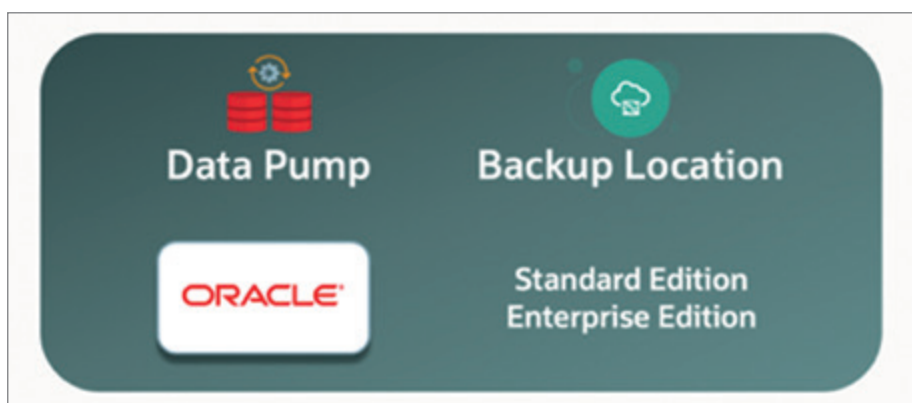


Abbildung 4: Logical Offline Migration mit Backup Location (Quelle: Oracle)

Mit ZDM können wir hauptsächlich migrieren von:

- On-Premises
- Oracle-Datenbanken auf AWS RDS (Relational Database Service)
- Oracle Public Cloud Gen 1
- Oracle Cloud Infrastructure (OCI)
- Datenbanken auf Linux, AIX und Solaris OS
- Datenbank kann SI, RAC one Node oder RAC sein
- Oracle Database Enterprise und Standard Edition
- Non-CDB oder CDB einer oder mehreren PDB

Unterstützte Source-Datenbank-Versionen sind: 11.2.0.4, 12.1.0.2, 12.2.0.1, 18c, 19c und 21c.

ZDM bietet die folgenden Vorteile:

- Einfaches und effizientes Arbeiten mit dem automatisierten Workflow von Oracle ZDM, der es ermöglicht, Datenbanken zu migrieren, ohne die verwendete Technologie zu kennen. Es stehen Migrationsvorlagen zur Verfügung und es ist keine manuelle Konfiguration oder Bedienung erforderlich. So werden menschliche Fehler vermieden.
- Hochverfügbar durch den Einsatz von Technologien wie Oracle Data Guard und Oracle GoldenGate, die eine Migration ohne Ausfallzeiten und ohne Auswirkungen auf die Produktion gewährleisten.
- Flexibel, da wir von verschiedenen Oracle-Quellen mit verschiedenen Technologien migrieren können.
- Validierung mit Prüfungen vor und nach der Migration, Möglichkeit zum Anhalten und Wiederaufnehmen der Migrationsaufgaben bei Bedarf sowie Integration einer Evaluierungsoption, um etwaigen Problemen während der Datenbankmigration vorzubeugen.
- Kosteneffektiv, da ZDM ohne zusätzliche Kosten verfügbar ist.

Schließlich folgt ZDM den Best Practices der Oracle MAA (Maximum Availability Architecture) mit den Lösungen GoldenGate und Data Guard. Der Online-Migrationsworkflow wird mit Recovery Manager, Data Pump und DB Links durchgeführt.



## Oracle ZDM-Migrationsabläufe

Es gibt 7 Migrationsworkflows, die in 4 Migrationstypen unterteilt sind:

- Physisch Offline
- Physisch Online
- Logisch Offline
- Logisch Online

Der physische Migrations-Workflow setzt voraus, dass Quell- und Zielta-

tenbank dieselbe Datenbankversion haben, während der logische Migrations-Workflow plattform- und versionsübergreifende Migrationen unterstützt (In-Flight-Upgrade bei der Migration in die Oracle Cloud). Die Migration von On-Premises-Datenbanken zu Oracle Autonomous wird nur über logische Workflows unterstützt. Das ZDM-Tool ermöglicht auch die Migration von Nicht-CDBs zu PDBs.

Wie wir sehen werden, verwenden einige der Workflows Backups. Falls

Backups verwendet werden, wird das erste Backup der Quelle mit RMAN komprimiert. Das Positive an ZDM ist, dass die Lizenzen für Oracle Advanced Compression und Oracle Advanced Security, während der Migration mit ZDM und nur für den Zweck der Migration, kostenlos genutzt werden können. Der Vorteil der möglichen Nutzung von Advanced Compression ist, dass die Größe des Backups, das heißt die Datenmenge, die in die Cloud transportiert werden muss, reduziert wird.

Wie RMAN-Backups genutzt werden, kann angepasst werden. Im Falle einer Migration zu On-Premise, das heißt einer Migration zur Exadata-Machine oder ExaCC, können wir vorhandene Backups verwenden, und es müssen keine über ZDM erstellt werden. Bei Verwendung der Zero Data Loss Recovery Appliance (ZDLRA) kann die Wiederherstellung auch direkt von dieser erfolgen, ohne dass Backups erstellt werden müssen. Bei allen anderen Zielplattformen (OCI, ExaCS oder Autonomous Database) muss ZDM als Teil des Workflows die Sicherungen selbst erstellen. Der Sicherungsort kann entweder ein Objektspeicher, ein NFS-Dateisystem oder eine ZDLRA (Oracle Zero Data Loss Recovery-Appliance) sein.

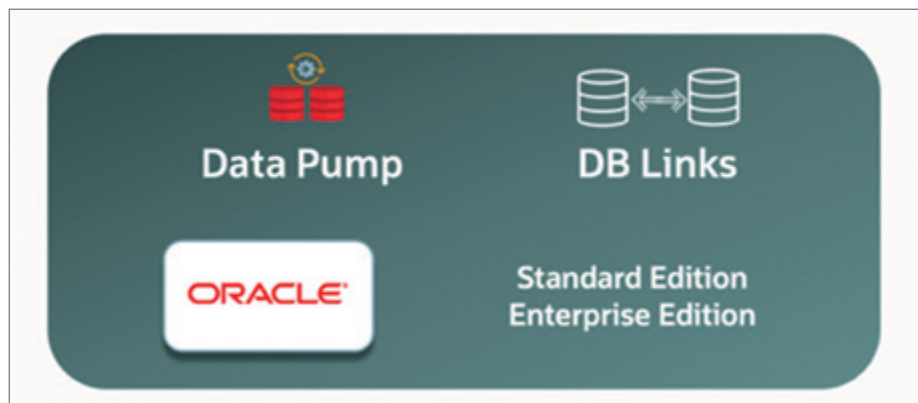


Abbildung 5: Logical Offline Migration mit DB Links (Quelle: Marc Oracle)

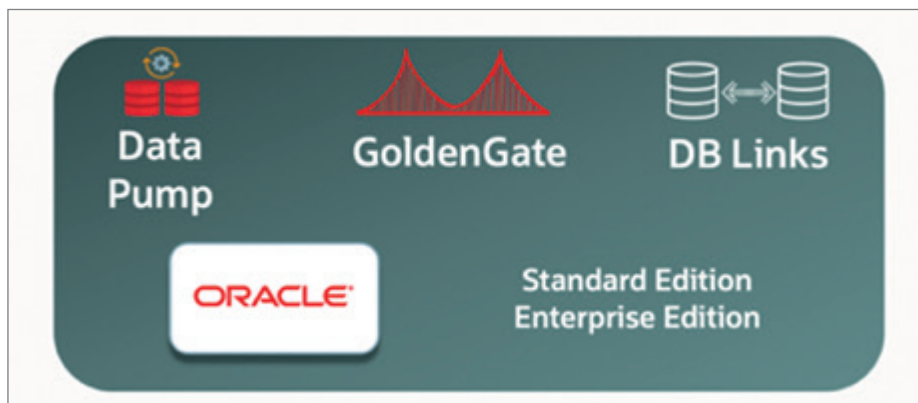


Abbildung 6: Logical Online Migration mit Backup Location (Quelle: Oracle)

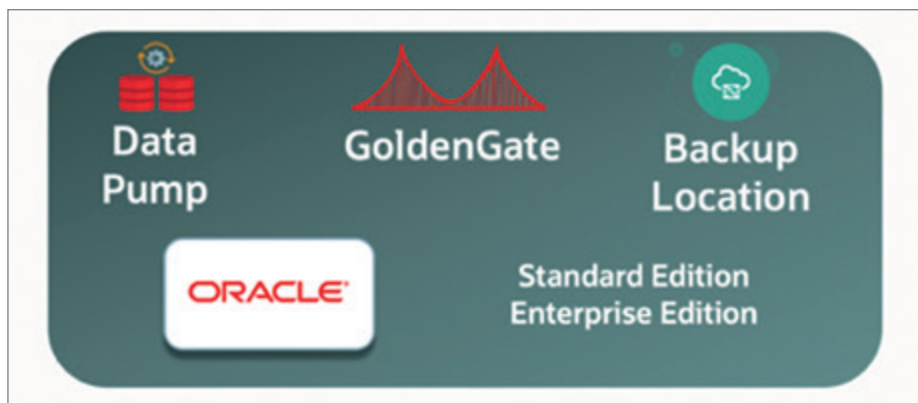


Abbildung 7: Logical Online Migration mit DB-Links (Quelle: Oracle)

## Physical Offline Migration

Dieser Workflow unterstützt die SE2- und EE-Datenbankmigration (siehe Abbildung 1). Er verwendet die Sicherung und Wiederherstellung durch RMAN mit einem Dump-Verzeichnis.

Im Folgenden sind die wichtigsten Schritte des Workflows aufgeführt:

- Download der ZDM-Software und Konfiguration
- Start der Datenbankmigration
- Zugriff der Quelldatenbank auf das Dump-Verzeichnis erstellen
- Sicherung der Datenbank
- Übertragung der Sicherungsdateien der Datenbank auf das Ziel
- Instanziierung einer Zieldatenbank mit einem Recovery unter Verwendung der Sicherungsdateien
- Umstellung auf die Zieldatenbank
- Abschluss des Migrationsprozesses

## Physical Online Migration mit Backup-Verzeichnis

Dieser Workflow unterstützt nur die EE-Datenbankmigration, da er die Data-Guard-Technologie verwendet (siehe *Abbildung 2*). Es werden RMAN und Data Guard verwendet. Die Standby-Datenbank wird mit Hilfe eines Backups erstellt.

Im Folgenden werden die wichtigsten Schritte des Workflows beschrieben:

- Download der ZDM-Software und Konfiguration
- Start der Datenbankmigration
- Verbindung der Quelldatenbank mit dem Backup-Verzeichnis
- Sicherung der primären Datenbank
- Übertragung der Sicherungsdateien der Datenbank auf das Ziel
- Instanziierung einer Standby-Datenbank unter Verwendung der Sicherungsdateien
- Synchronisierung von Primär- und Standby-Datenbank
- DG-Switchover (Tausch der Rollen)
- Abschluss des Migrationsprozesses

## Physical Online Migration mit Direct Data Transfer

Dieser Workflow unterstützt nur die EE-Datenbankmigration (siehe *Abbildung 3*). Es werden RMAN und Data Guard verwendet und die Standby-Datenbank wird diesmal nicht mit Hilfe eines Backups erstellt, sondern mit aktiver Datenbankduplikation (für Version 11.2) oder Wieder-

herstellung aus dem Service (für Version 12.1, 12.2, 18.X oder 19.X).

Nachfolgend sind die wichtigsten Schritte des Workflows aufgeführt:

- Download der ZDM-Software und Konfiguration
- Start der Datenbankmigration
- Verbindung zu den Quell- und Zieldatenbanken herstellen
- Instanziierung einer Standby-Datenbank mit aktiver Datenbankduplikation oder Wiederherstellung aus dem Service
- Synchronisation von Primär- und Standby-Datenbank
- DG-Switchover (Tausch der Rollen)
- Abschluss des Migrationsprozesses

Für beide Arbeitsabläufe der physischen Online-Migration gilt, dass die einzige kurze Ausfallzeit während des Switchover-Schrittes auftritt, bei dem die Primär- und Standby-Rolle getauscht werden. Es ist auch wichtig zu beachten, dass nur diese beiden physischen Online-Workflows ein Fallback unterstützen. Die OCI-Datenbank wird jedoch mit TDE (Transparent Tablespace Encryption) verschlüsselt. Jedes Redo, das für einen verschlüsselten Tablespace erzeugt wird, wird ebenfalls verschlüsselt. Das bedeutet, dass die On-Prem-Datenbank (Quelle der Migration) in der Lage sein muss, das Redo zu entschlüsseln, bevor es angewendet werden kann. Die Oracle-Advanced-Security-Lizenz kann nur auf dem Weg zur Cloud Solution und während der Migration kostenlos genutzt werden. Im Fallback-Sze-

nario ist dies anders und kann auf zwei Arten geschehen:

Entweder wir haben eine Lizenz für die Advanced-Security-Option oder wir müssen das Redo entschlüsseln, um es auf unverschlüsselte Tablespaces auf der On-Prem-Seite anwenden zu können. Dies wird mit dem Instanzparameter `tablespace_encryption` konfiguriert, der entweder den Wert `decrypt_only` (Konfiguration auf der On-Premise-Datenbank) oder `auto_enable` (Konfiguration auf der OCI-Datenbank) annehmen kann.

Für jeden anderen Workflow muss das Fallback-Szenario unabhängig vom ZDM-Tool manuell konfiguriert werden.

## Logical Offline Migration mit Backup Location

Dieser Arbeitsablauf unterstützt die SE2- und EE-Datenbankmigration (siehe *Abbildung 4*). Er wird Data Pump mit einem Sicherungsort verwenden.

Im Folgenden werden die wichtigsten Schritte des Workflows beschrieben:

- Download der ZDM-Software und Konfiguration
- Validierungen (z.B. Verzeichnis-Export/Import)
- Verbindung mit dem Dump-Verzeichnis
- Data-Pump-Export aus der Quelldatenbank in das Dump-Verzeichnis
- Data-Pump-Import aus dem Dump-Verzeichnis in die Zieldatenbank
- Instanziierung der Zieldatenbank

```
[zdmuser@zdm-host ~]$ /u01/app/zdmhome/bin/zdmcli -h
Performs Zero Downtime Migration operations for the cloud.
Usage:
    zdmcli abort           Perform abort operation for a migration job.
    zdmcli add             Adds an image type or user actions.
    zdmcli audit          Perform audit operation for a migration job.
    zdmcli delete         Deletes a job, an image type or an user action.
    zdmcli migrate        Performs migration operation for the cloud.
    zdmcli modify         Modifies user actions.
    zdmcli query          Gets information of a job, image type, user actions and audit.
    zdmcli resume         Perform resume operation for a migration job.
    zdmcli suspend        Suspend the migration job.

For detailed help on each command use:
    zdmcli <command> <object> -help
```

Listing 1: ZDMcli Online-Leitfaden

```

[zdmuser@zdm-host migration]$ cat zdm_logical_off_to_adb.rsp
MIGRATION_METHOD=OFFLINE_LOGICAL
DATA_TRANSFER_MEDIUM=OSS
TARGETDATABASE_OCID=ocid1.autonomousdatabase.oc1.eu-zurich-1.an5heljrv*****
*****lmq
TARGETDATABASE_ADMINUSERNAME=ADMIN
SOURCEDATABASE_ADMINUSERNAME=orcl_user
SOURCEDATABASE_CONNECTIONDETAILS_HOST=srv-source-zdm-01
SOURCEDATABASE_CONNECTIONDETAILS_PORT=1521
SOURCEDATABASE_CONNECTIONDETAILS_SERVICENAME=pdb1
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_TENANTID=ocid1.tenancy.oc1..aaaaaaay*****
*****62na
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_USERID=ocid1.user.oc1..aaaaaaad1*****
*****6sjiq
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_FINGERPRINT=2f:**:**:**:**:**:**:**:**:**:**:**:**:**:*:da
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_PRIVATEKEYFILE=/u01/app/zdmhome/.oci/oci_api_key.pem
OCIAUTHENTICATIONDETAILS_REGIONID=eu-zurich-1

#TARGETDATABASE_CONNECTIONDETAILS_BASTIONDETAILS_PORT=22
#SOURCEDATABASE_CONNECTIONDETAILS_BASTIONDETAILS_PORT=22
#SOURCECONTAINERDATABASE_CONNECTIONDETAILS_BASTIONDETAILS_PORT=22
GOLDENGATESETTINGS_REPLICAT_MAPPARALLELISM=4
GOLDENGATESETTINGS_REPLICAT_MINAPPLYPARALLELISM=4
GOLDENGATESETTINGS_REPLICAT_MAXAPPLYPARALLELISM=50
GOLDENGATESETTINGS_ACCEPTABLELAG=30

DATAPUMPSETTINGS_JOBMODE=SCHEMA
DATAPUMPSETTINGS_DELETEDUMPSINOS=TRUE
DATAPUMPSETTINGS_FIXINVALIDOBJECTS=TRUE
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_ENCRYPTION=ALL
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_TABLEEXISTSACTION=TRUNCATE
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_IMPORTPARALLELISMDEGREE=1
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_EXPORTPARALLELISMDEGREE=1
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_EXCLUDETYPELIST=user
DATAPUMPSETTINGS_DATA_BUCKET_NAMESPACENAME=zrk0iitcogu9
DATAPUMPSETTINGS_DATA_BUCKET_BUCKETNAME=ZDMBucket
DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_NAME=DB_EXP_DR
DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_PATH=/u01/app/oracle/export
DATAPUMPSETTINGS_CREATEAUTHOKEN=FALSE
DATAPUMPSETTINGS_OMITENCRYPTIONCLAUSE=TRUE
DATAPUMPSETTINGS_SECUREFILELOB=TRUE
DATAPUMPSETTINGS_SKIPDEFAULTTRANSFORM=FALSE
EXCLUDEOBJECTS-1=owner:ORCL_USER, objectName:*
EXCLUDEOBJECTS-2=owner:PDBADMIN, objectName:*

DUMPTRANSFERDETAILS_PARALLELCOUNT=3
DUMPTRANSFERDETAILS_RETRYCOUNT=3
DUMPTRANSFERDETAILS_RSYNCAVAILABLE=FALSE
DUMPTRANSFERDETAILS_SOURCE_USEOCICLI=FALSE
DUMPTRANSFERDETAILS_TARGET_USEOCICLI=FALSE

```

Listing 2: Vorlage, Beispiel einer logischen Offline-Migration nach Autonomous-Datenbank

```
[zdmuser@zdm-host ~]$ /u01/app/zdmhome/bin/zdmcli migrate database -rsp /home/zdmuser/migration/zdm_logical_off_to_adb.rsp -sourcenode srv-source-zdm-01 -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/ssh-key-VM-YAK-ace.key -srcarg3 sudo_location:/bin/sudo -sourcedb CDB01
zdm-host.sub03101110460.vcnxchange.oraclevcn.com: Audit ID: 36
Enter source database administrative user "orcl_user" password:
Enter target database administrative user "ADMIN" password:
Enter Data Pump encryption password:
Operation "zdmcli migrate database" scheduled with the job ID "7".

[zdmuser@zdm-host ~]$ /u01/app/zdmhome/bin/zdmcli query job -jobid 7
zdm-host.sub03101110460.vcnxchange.oraclevcn.com: Audit ID: 39
Job ID: 7
User: zdmuser
Client: zdm-host
Job Type: "MIGRATE"
Scheduled job command: "zdmcli migrate database -rsp /home/zdmuser/migration/zdm_logical_off_to_adb.rsp -sourcenode srv-source-zdm-01 -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/ssh-key-VM-YAK-ace.key -srcarg3 sudo_location:/bin/sudo -sourcedb CDB01"
Scheduled job execution start time: 2023-03-15T17:51:43Z. Equivalent local time: 2023-03-15 17:51:43
Current status: SUCCEEDED
Result file path: "/u01/app/zdmhome/chkbase/scheduled/job-7-2023-03-15-17:51:51.log"
Metrics file path: "/u01/app/zdmhome/chkbase/scheduled/job-7-2023-03-15-17:51:51.json"
Excluded objects file path: "/u01/app/zdmhome/chkbase/scheduled/job-7-filtered-objects-2023-03-15T17:52:13.614.json"
Job execution start time: 2023-03-15 17:51:51
Job execution end time: 2023-03-15 17:56:08
Job execution elapsed time: 4 minutes 16 seconds
ZDM_VALIDATE_TGT ..... COMPLETED
ZDM_VALIDATE_SRC ..... COMPLETED
ZDM_SETUP_SRC ..... COMPLETED
ZDM_PRE_MIGRATION_ADVISOR ..... COMPLETED
ZDM_VALIDATE_DATAPUMP_SETTINGS_SRC .... COMPLETED
ZDM_VALIDATE_DATAPUMP_SETTINGS_TGT .... COMPLETED
ZDM_PREPARE_DATAPUMP_SRC ..... COMPLETED
ZDM_DATAPUMP_ESTIMATE_SRC ..... COMPLETED
ZDM_PREPARE_DATAPUMP_TGT ..... COMPLETED
ZDM_DATAPUMP_EXPORT_SRC ..... COMPLETED
ZDM_UPLOAD_DUMPS_SRC ..... COMPLETED
ZDM_DATAPUMP_IMPORT_TGT ..... COMPLETED
ZDM_POST_DATAPUMP_SRC ..... COMPLETED
ZDM_POST_DATAPUMP_TGT ..... COMPLETED
ZDM_POST_ACTIONS ..... COMPLETED
ZDM_CLEANUP_SRC ..... COMPLETED
```

Listing 3: Logical-Offline-Migration-Ausführung

```
[zdmuser@zdm-host scheduled]$ pwd
/u01/app/zdmhome/chkbase/scheduled

[zdmuser@zdm-host scheduled]$ cat job-7-2023-03-15-17:51:51.log
zdm-host: 2023-03-15T17:51:52.056Z : Starting zero downtime migrate operation ...
zdm-host: 2023-03-15T17:51:52.063Z : Executing phase ZDM_VALIDATE_TGT
...
...
...
zdm-host: 2023-03-15T17:56:06.876Z : Executing phase ZDM_POST_ACTIONS
zdm-host: 2023-03-15T17:56:06.887Z : Execution of phase ZDM_POST_ACTIONS completed
zdm-host: 2023-03-15T17:56:06.906Z : Executing phase ZDM_CLEANUP_SRC
zdm-host: 2023-03-15T17:56:07.137Z : Cleaning up ZDM on the source node srv-source-zdm-01 ...
zdm-host: 2023-03-15T17:56:08.093Z : Execution of phase ZDM_CLEANUP_SRC completed
```

Listing 4: Die Log-Datei der Logischen Offline-Migration

- Umstellung auf die Zieldatenbank
- Abschluss des Migrationsprozesses

## Logical Offline Migration mit DB Links

Dieser Workflow unterstützt die SE2- und EE-Datenbankmigration (siehe Abbildung 5). Er verwendet Data Pump und DB-Links. Dieser Workflow kann nur für Datenbanken mit einer Größe von weniger als 100 GB verwendet werden. Es erfolgt ein direkter Import in die Zieldatenbank von der Quelldatenbank über das Netzwerk.

Im Folgenden werden die wichtigsten Schritte des Workflows beschrieben:

- Download der ZDM-Software und Konfiguration
- Überprüfungen
- Verbindung von Quell- und Zieldatenbank über DB-Link
- Import der Datenpumpe von der Quell- in die Zieldatenbank über das Netzwerk
- Validierung der Zieldatenbank
- Umstellung auf die Zieldatenbank
- Abschluss des Migrationsprozesses

## Logical Online Migration mit Backup Location

Dieser Workflow unterstützt die SE2- und EE-Datenbankmigration (siehe Abbildung 6). Er verwendet Data Pump mit einem Sicherungsort und GoldenGate. Dieser Workflow bietet keine Ausfallzeit für die Migration.

Im Folgenden werden die wichtigsten Schritte des Workflows beschrieben:

- Download der ZDM-Software und Konfiguration
- Start der Datenbankmigration
- Konfiguration der Quell-, Ziel- und Backup-Verbindungen
- GoldenGate-Konfiguration und Erfassen von Quelltransaktionen
- Data-Pump-Export von der Quelldatenbank zum Sicherungsort
- Data-Pump-Import von der Sicherungsdatenbank in die Zieldatenbank
- GoldenGate-Konfiguration und Übernahme der Änderungen
- Umstellung auf die Zieldatenbank
- Abschluss des Migrationsprozesses

## Logical Online Migration mit DB Links

Dieser Workflow unterstützt die SE2- und EE-Datenbankmigration (siehe Abbildung 7). Er verwendet Data Pump, GoldenGate und DB-Links. Dieser Workflow bietet keine Ausfallzeit für die Migration.

Im Folgenden sind die wichtigsten Schritte des Workflows aufgeführt:

- Download der ZDM-Software und Konfiguration
- Start der Datenbankmigration
- Quell- und Zieldatenbankverbindungen über DB-Link
- GoldenGate-Konfiguration und Erfassen von Quelltransaktionen
- Import von Data Pump aus der Quelldatenbank in die Zieldatenbank
- GoldenGate-Konfiguration und Übernahme der Änderungen
- Umschalten auf die Zieldatenbank
- Abschluss des Migrationsprozesses

## ZDM cli

Die Verwendung des Tools ist recht einfach. Es gibt nur einen Befehl, den man kennen muss: `zdmcli`. Das CLI verfügt über verschiedene Optionen, die es beispielsweise ermöglichen, die Ausführung eines Workflows zu überprüfen, eine Migration abzubrechen, auszusetzen oder wieder aufzunehmen und so weiter (siehe Listing 1).

Die Option `-eval` des Cli erlaubt es, die Migration zu testen und zu überprüfen, ohne eine echte Migration durchzuführen. Es wird empfohlen, sie vor jeder Migration zu verwenden. In Listing 2 finden Sie ein Beispiel für eine logische Offline-Vorlage, die für die Migration einer lokalen Datenbank in eine Autonomous Database in OCI verwendet wird.

Listing 3 zeigt, wie ZDM für eine logische Offline-Migration unter Verwendung der vorherigen Vorlage ausgeführt wird. Die Option `Query Job` bestätigt, dass alle Schritte erfolgreich ausgeführt wurden.

Wir können auch die generierte Protokolldatei überprüfen (siehe Listing 4), die das letzte Protokoll „Execution of phase ZDM\_CLEANUP\_SRC completed“ enthält, welches bestätigt, dass die Migration erfolgreich durchgeführt wurde.

## Quellen

- [1] <https://www.oracle.com/database/zero-downtime-migration/>
- [2] <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.4/>
- [3] <https://www.oracle.com/docs/tech/oracle-zdm-technical-brief.pdf>
- [4] <https://apexapps.oracle.com/pls/apex/dbprm/r/livelabs/view-workshop?wid=850>
- [5] <https://www.oracle.com/a/ocom/docs/engineered-systems/exadata/exadata-x9m-8-ds.pdf>
- [6] <https://www.oracle.com/a/ocom/docs/engineered-systems/exadata/exadata-x9m-2-ds.pdf>

## Über den Autor

Marc Wagner arbeitet bei dbi services und ist seit mehreren Jahren als Oracle Senior Consultant tätig, wobei er sich hauptsächlich mit ODA, Oracle Database Administration und Hochverfügbarkeitsumgebung mit Data Guard und Dbvisit beschäftigt. Er hat mehrere Vorträge auf der SOUG, der DOAG und UKOUG zu diesen Themen gehalten und ist ACE Pro. Neben seiner Tätigkeit als Senior Consultant führt er Schulungen durch und ist auch Referent für zwei Workshops, dem Oracle Database Appliance Workshop und dem Data Guard Workshop innerhalb des Unternehmens. Marc Wagner ist auch Teil des Teams der DMK-dbi-tool-Entwickler.



Marc Wagner  
marc.wagner@dbi-services.com

1

**APEX connect**  
by DOAG

1050 EUR <sup>1)</sup>

2

**DOAG**  
Datenbank  
mit Exaday

**2023**

950 EUR <sup>1)</sup>

3

**CloudLand**  
2023

2060 EUR <sup>1)</sup>

## Event-Flatrate

[shop.doag.org/genera](https://shop.doag.org/genera)



1950 EUR



<sup>1)</sup> Originalpreis 2023 (zzgl. MwSt). Leichte Anpassung der Veranstaltungspreise 2024 möglich. DOAG-Mitglieder erhalten Rabatte: 30 % in 2023, 15 % in 2024.

<sup>2)</sup> Für DOAG-Bestandskunden bleibt der Preis in 2024 gleich. Neukunden zahlen ab dem 02.01.2024 für die PremiumCard 2250 EUR (zzgl. MwSt.).

<sup>3)</sup> Voraussetzung für den Erwerb der PremiumCard ist eine aktive DOAG-Mitgliedschaft (ab 120 EUR).

te der DOAG

/shop/#premiumcard

2023  
DOAG  
M CARD

4

2023  
European  
NetSuite  
User Days  
850 EUR <sup>1)</sup>

5

2023  
DOAG  
Konferenz + Ausstellung  
KI Navigator 2023  
2880 EUR <sup>1)</sup>

EUR <sup>1) 2) 3)</sup>

## Die Vorzüge der DOAG PremiumCard

Sie gilt für alle Konferenzen und eintägigen Veranstaltungen inkl. WebSessions. Sie kostet **1950 EUR zzgl. Mwst.** Eine aktive DOAG-Mitgliedschaft (ab 120 EUR) ist Voraussetzung für den Erwerb der PremiumCard. Sie gilt immer für das laufende Kalenderjahr.

Ein Vergleich: Der Besuch sämtlicher hier dargestellter 2023er-Events kostet die Teilnehmenden insgesamt 7790 EUR zzgl. Mwst. Mit einer aktiven Mitgliedschaft (= 30 % Rabatt) sinkt dieser Preis auf 5453 EUR zzgl. Mwst.

Mit Erwerb der Mitgliedschaft und der PremiumCard spart das Mitglied beim Besuch aller hier dargestellten Events über 3383 EUR (ohne Mwst.) gegenüber dem Mitgliederpreis OHNE PremiumCard.

Der Vorteil liegt also klar auf der Hand: Die PremiumCard lohnt sich für alle DOAG-Mitglieder, die neben einem großen Event noch ein zweites oder sogar drittes ins Auge fassen.

Ein weiterer Bonus: PremiumCard-Besitzer erhalten auch rückwirkend kostenfreien Zugriff auf alle verfügbaren Vortragsunterlagen der Events des laufenden Jahres und, soweit vorhanden, auf die Vortragsaufzeichnungen. Das Abonnement der Red Stack inkl. Business News gibt es gratis dazu.

Vom Angebot ausgeschlossen sind: JavaLand, DOAG Berliner Expertenseminare, sonstige mehrtägige Veranstaltungen und Schulungstage bzw. Schulungen.



## NACHHALTIGE GESCHÄFTSMODELLE



# Mit prozessorientierten Informationssystemen zu nachhaltigen Geschäftsmodellen

Andreas Fritsch, Karlsruher Institut für Technologie

*Der Wandel hin zu nachhaltigen Geschäftsmodellen erfordert es, die Auswirkungen der eigenen Unternehmenstätigkeit auf Menschen und Umwelt zu untersuchen und zu steuern. Insbesondere müssen negative ökologische und soziale Auswirkungen erkannt und vermieden werden. Eine tiefe Integration von Nachhaltigkeitsdaten in die Informationssysteme eines Unternehmens eröffnet die Möglichkeit, Verantwortlichkeiten und Verbesserungspotentiale zu identifizieren. Dies kann dazu beitragen, wirksame Veränderungen in Richtung mehr Nachhaltigkeit im Unternehmen anzustoßen. Unternehmen stehen dabei jedoch vor konzeptuellen und technischen Herausforderungen. Im vorliegenden Beitrag wird daher ein Vorgehensmodell skizziert, das bei der Bewältigung dieser Herausforderungen als Orientierung dienen kann. Das skizzierte Vorgehen verknüpft Konzepte des Geschäftsprozessmanagements mit denen der Nachhaltigkeitsanalyse und unterstützt so den Entwurf von Informationssystemen, die sowohl für die Berichterstattung von Nachhaltigkeitskennzahlen als auch für die Steuerung von Verbesserungsmaßnahmen eingesetzt werden können. Das Vorgehensmodell wird anhand von praktischen Beispielen aus der Treibhausgasbilanzierung in der Elektronikindustrie illustriert.*

## Einleitung

Die Forderung nachhaltige Geschäftsmodelle zu verfolgen, wird aktuell von verschiedenen Seiten an Unternehmen herangetragen, sei es von der Politik, Nichtregierungsorganisationen, Kundinnen und Kunden oder der eigenen Belegschaft. Viele Unternehmen haben daher bereits erste Schritte unternommen, indem sie Nachhaltigkeitsinitiativen gestartet oder Nachhaltigkeitsberichte erstellt haben. Die Suche nach Wegen zu mehr Nachhaltigkeit wird jedoch gelegentlich durch unterschiedliche Interpretationsmöglichkeiten des Begriffs erschwert. Zu Beginn dieses Artikels soll daher eine kurze Abgrenzung und Begriffsklärung vorgenommen werden. Die Kernanliegen der Nachhaltigkeitsdiskussion im Sinne einer globalen nachhaltigen Entwicklung sind der Schutz der Umwelt und soziale Gerechtigkeit [1]. Das bedeutet, aus Unternehmenssicht geht es um potenzielle (negative) Auswirkungen des unternehmerischen Handelns auf Menschen (soziale Gerechtigkeit) und Umwelt (Umweltschutz). Manchmal werden unternehmerische Themen mit dem Etikett „Nachhaltigkeit“ versehen, die nicht dem hier vertretenen Begriffsver-

ständnis entsprechen: Beispielsweise, wenn von „nachhaltigen Unternehmensvorteilen“ oder „nachhaltigem Unternehmenswachstum“ gesprochen wird und dabei lediglich ein langfristiger finanzieller Erfolg des Unternehmens gemeint ist. Diese Anliegen sind aus unternehmerischer Sicht valide und es gibt durchaus Zusammenhänge und Synergien zwischen beispielsweise Umweltschutz und dem finanziellen Erfolg eines Unternehmens. Doch aus Sicht einer globalen nachhaltigen Entwicklung, wie sie im weiteren Verlauf dieses Artikels zugrunde gelegt wird, handelt es sich dabei um Nebenschauplätze [2]. Der Fokus liegt im Folgenden auf den Themen soziale Gerechtigkeit und Umweltschutz sowie darauf, wie ein Unternehmen seinen Beitrag dazu verstehen und verbessern kann.

Eine grundlegende Konsequenz aus der hier vertretenen Auffassung des Nachhaltigkeitsbegriffs ist, dass sich der Verantwortungsbereich für unternehmerisches Handeln erweitert [3]. Es reicht nicht aus, nur die unmittelbaren Auswirkungen innerhalb der Unternehmensgrenzen zu betrachten, wie zum Beispiel Löhne an die eigene Belegschaft oder Schadstoffemis-

sionen auf dem Betriebsgelände. Vielmehr müssen auch die Auswirkungen der durch die Unternehmensaktivitäten ausgelösten Stoff-, Energie- und Produktströme berücksichtigt werden. Zum Beispiel trägt ein Unternehmen, das einen Server betreibt, auch Verantwortung für die Treibhausgasemissionen, die durch die Erzeugung der für den Betrieb des Servers benötigten Energie in einem Kohlekraftwerk entstehen. Ebenso trägt ein Batteriehersteller Verantwortung für den Lithium-Abbau und die damit einhergehende Kontamination der Wasservorkommen lokaler Gemeinschaften [4]. Dieser Gedanke ist ein zentraler Bestandteil international anerkannter und standardisierter Methoden der Nachhaltigkeitsanalyse [5]–[7]. Er findet sich auch in der aktuellen Nachhaltigkeitsgesetzgebung wie dem Lieferkettengesetz, das von Unternehmen verlangt, Rechenschaft über soziale und ökologische Themen entlang ihrer Lieferkette abzulegen.

Dabei stellt sich die Frage, wie diese Auswirkungen identifiziert und gesteuert werden können, insbesondere da sie sich auch außerhalb der direkten Kontrolle des Unternehmens befinden und somit

im Allgemeinen kein direkter Einfluss ausgeübt werden kann. Ein erster Schritt für viele Unternehmen ist die Nachhaltigkeitsberichterstattung, bei der beispielsweise Treibhausgasemissionen bilanziert werden. Unternehmen, die von sich behaupten, „nachhaltige Geschäftsmodelle“ zu betreiben, müssen jedoch proaktiv vorgehen und darüber nachdenken, wie sie nicht nur gesetzliche Berichtspflichten erfüllen, sondern auch die systemischen Nachhaltigkeitsauswirkungen ihrer Unternehmensaktivitäten effektiv managen können. Die bereits erwähnten standardisierten Methoden der Nachhaltigkeitsanalyse helfen dabei, belastbare Nachhaltigkeitsdaten zu erheben. Diese Nachhaltigkeitsdaten müssen schließlich aggregiert, analysiert und den Entscheidungsträgern auf verschiedenen Ebenen des Unternehmens zur Verfügung gestellt werden, um eine effektive Steuerung zu ermöglichen. Dabei können Konzepte und Methoden des Geschäftsprozessmanagements unterstützend wirken. Das hier vorgestellte Vorgehensmodell baut auf diesen Erkenntnissen auf und verbindet die Konzepte und Prinzipien des Geschäftsprozessmanagements mit denen der Nachhaltigkeitsanalyse. Das Vorgehensmodell ist bewusst offen gestaltet, damit es für Unternehmen mit unterschiedlichem Reifegrad bezüglich ihres Nachhaltigkeitsmanagements anwendbar ist. Es bietet

Orientierung bei der Entwicklung von Informationssystemen für das Nachhaltigkeitsmanagement auf Basis der Prinzipien, die auf dem eingangs eingeführten Nachhaltigkeitsverständnis beruhen. Ziel ist es, Nachhaltigkeitsdaten nicht nur für die Berichterstattung zu verwenden, sondern sie zur effektiven Gestaltung der Unternehmensaktivitäten nutzbar zu machen.

**Vorgehensmodell**

Abbildung 1 zeigt das Vorgehensmodell, das aus vier Phasen besteht: (1) Geschäftsprozessidentifikation, (2) Spezifikation des Nachhaltigkeitskonzepts, (3) Datenintegration, (4) Implementierung oder Anpassung des Informationssystems. Über die rückführenden Pfeile wird angezeigt, dass das Vorgehensmodell nicht streng sequenziell zu verstehen ist und, dass Ergebnisse einer Phase auch Überarbeitungen in der vorangegangenen Phase auslösen können. Der große rückführende Pfeil zeigt an, dass das Vorgehensmodell iterativ gedacht ist. Es ist also möglich, zunächst nur einen eng abgegrenzten Anwendungsfall zu unterstützen und das Informationssystem dann Schritt für Schritt zu erweitern, beziehungsweise zu verbessern. Entlang des rückführenden Pfeils sind den einzelnen Phasen Merkmale (Prozessumfang, Granularität, ...) zugeordnet, die in einer Iteration im Sinne eines effektiveren Nachhaltigkeitsmanagements

verbessert werden können. Im Folgenden werden die einzelnen Phasen anhand von Beispielen erläutert. Dabei wird auch darauf eingegangen, wie das resultierende Informationssystem iterativ verbessert werden kann.

Die erste Phase besteht darin, die zu unterstützenden Geschäftsprozesse zu identifizieren. Grundsätzlich kann jeder Geschäftsprozess in Frage kommen, sei es ein Produktionsprozess oder ein Dienstleistungsprozess, ein Kernprozess des Unternehmens oder ein unterstützender Prozess. Um geeignete Geschäftsprozesse auszuwählen, kann zunächst eine einfache Abschätzung der potenziellen Nachhaltigkeitsauswirkungen der Geschäftsprozesse erfolgen. Die Geschäftsprozesse mit hohen zu erwartenden Nachhaltigkeitsauswirkungen sind gute Kandidaten für eine erste Iteration. Alternativ kann die Auswahl auch rein pragmatisch erfolgen, wenn ein Geschäftsprozess beispielsweise bereits gut dokumentiert in einem Prozessmanagementsystem des Unternehmens vorliegt. Unternehmen, die bereits einen hohen Reifegrad in ihrem Geschäftsprozessmanagement erreicht haben, können an dieser Stelle möglicherweise die Verantwortlichen für einzelne Prozessschritte benennen. Das Prozessmodell kann eventuell auch schon Auskunft über geschäftsrelevante genutzte Ressourcen, Inputs und Outputs des Prozes-

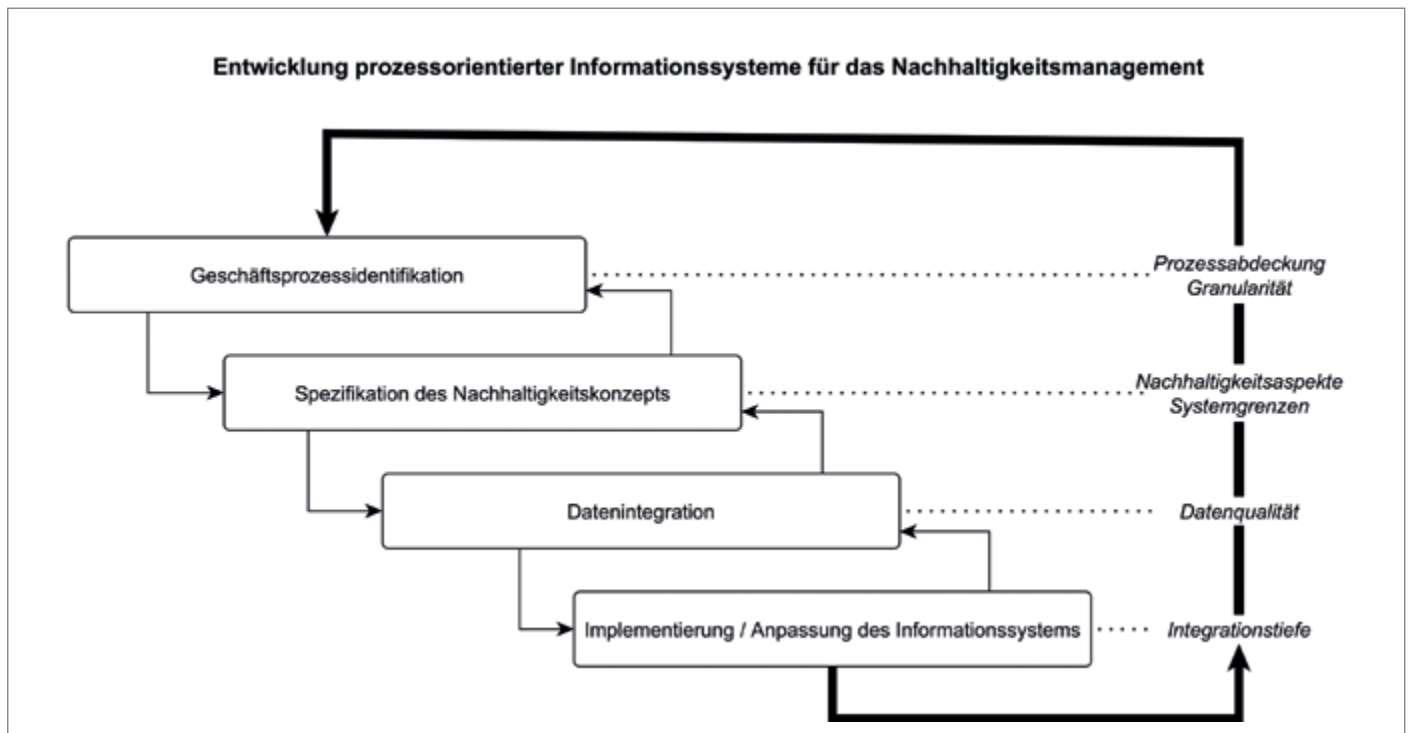


Abbildung 1: Job Level (@viadee angelehnt an Ash Maurya Running Lean [3])

ses geben. Je nach Anwendungsfall kann die Betrachtung der Geschäftsprozesse in unterschiedlichem Umfang und auf unterschiedlichen Granularitätsstufen erfolgen (Verbesserungsmerkmale Prozessabdeckung und Granularität). Für ein möglichst effektives Nachhaltigkeitsmanagement sollte eine hohe Abdeckung und feingranulare Betrachtung der Geschäftsprozesse angestrebt werden. Es kann aber sinnvoll sein, zunächst nur die wichtigsten Prozessschritte grob abzubilden und die Betrachtung später zu verfeinern. Um ein Beispiel aus der Elektronikindustrie zu nennen, könnte ein produzierendes Unternehmen einen Kernprozess wie die Leiterplattenbestückung als Kandidaten auswählen. Dieser Prozess könnte detailliert unter Beachtung diverser Teilschritte (wie Bestücken, Löten) betrachtet werden oder auf einer höheren Abstraktionsebene. Verschiedene geschäftsrelevante Inputs (wie Bauteile, Kondensatoren) sowie Prozessverantwortliche und Ressourcen (wie Klebe- und Lötanlagen) könnten bereits mit Abschluss dieser Phase benannt werden.

Nachdem die zu unterstützenden Geschäftsprozesse identifiziert und gegebenenfalls modelliert wurden, folgt als zweite Phase die Spezifikation des Nachhaltigkeitskonzepts. Dabei ist insbesondere der eingangs erwähnte erweiterte Verantwortungsbereich über die eigenen Unternehmensgrenzen hinaus zu beachten. Zur Orientierung können verschiedene Standards der Nachhaltigkeitsanalyse herangezogen werden. Da diese Standards in vielen Fällen aufeinander aufbauen und miteinander synchronisiert sind, können wie im Folgenden beschrieben, allgemeingültige Prinzipien daraus extrahiert werden. Zu den zu treffenden Entscheidungen gehört, welche Nachhaltigkeitsaspekte betrachtet werden sollen. Im Beispiel des Elektronikunternehmens könnte der Fokus auf Treibhausgasemissionen und damit den potenziellen negativen Auswirkungen der Produktion auf den Klimawandel liegen. Weitere Umweltaspekte wie Wasserverbrauch, Biodiversität, sowie soziale Aspekte wie Arbeitsbedingungen und Menschenrechte könnten ebenfalls berücksichtigt werden. Ein weiterer wichtiger Schritt ist die Festlegung der Systemgrenzen der Nachhaltigkeitsanalyse. Dabei stellt sich die Frage, ob nur unternehmensinterne Nachhaltigkeitsauswirkungen betrachtet werden oder, ob auch die Lieferkette miteinbe-

zogen wird. Das Treibhausgasprotokoll [7] definiert dazu beispielsweise drei „Scopes“ (Scope 1: direkte Emissionen im Unternehmen, Scope 2: Emissionen aus dem Bezug von Energie, Scope 3: Emissionen, die außerhalb des Unternehmens, zum Beispiel in der Lieferkette, anfallen). Ähnliche Abgrenzungen sind auch für andere Nachhaltigkeitsaspekte vorzunehmen. Es ist nicht immer notwendig oder sinnvoll, den vollen Scope zu betrachten, aber in jedem Fall muss er möglichst klar abgegrenzt werden, um eine gewisse Vergleichbarkeit und Nachvollziehbarkeit von Nachhaltigkeitskennzahlen zu erreichen. Sobald die zu betrachtenden Nachhaltigkeitsaspekte und Systemgrenzen festgelegt sind, können die nachhaltigkeitsrelevanten Inputs und Outputs des Geschäftsprozesses innerhalb dieser Abgrenzung ermittelt werden. Im Fall eines Elektronikunternehmens könnten dies beispielsweise die Energie sein, die für die verschiedenen Teilschritte der Leiterplattenbestückung benötigt wird (Scope 2) sowie sonstige Komponenten und Materialien, die zur Herstellung eingekauft werden (insbesondere für Scope 3 relevant).

Die dritte Phase ist die Datenintegration. Hierbei müssen zunächst geeignete Datenquellen für die Bewertung der in der zweiten Phase ermittelten Inputs und Outputs identifiziert werden. Je nach Anwendungsfall und Datenverfügbarkeit können Datenquellen unterschiedlicher Qualität akzeptiert werden. In der Literatur zu Nachhaltigkeitsanalysen wird oft zwischen Primär- und Sekundärdaten unterschieden [8]. Primärdaten umfassen direkte Messungen an den betrachteten Prozessen, wohingegen Sekundärdaten nicht direkt aus den betrachteten Prozessen stammen. Stattdessen handelt es sich um generische Daten, etwa aus kommerziellen Ökobilanzierungsdatenbanken, die herangezogen werden können, wenn Primärdaten fehlen, was oft der Fall ist. Für das betrachtete Beispiel könnten zur Bestimmung der Treibhausgasemissionen einzelner verarbeiteter Elektronikkomponenten Daten aus einer Ökobilanzierungsdatenbank herangezogen werden (Sekundärdaten). Alternativ könnten die Treibhausgasemissionen direkt beim Lieferanten angefragt werden (Primärdaten). In jedem Fall ist sicherzustellen, dass die erhobenen Daten korrekt interpretiert werden. Es sollte geklärt werden, ob der Lieferant selbst für die Erhebung seiner Treibhausgasemissionen Scope 1, 2 oder 3

betrachtet. Um die Interpretation und Berechnung auch softwaregestützt durchführen zu können, ist schließlich ein geeignetes Datenmodell zu erstellen. Ein Beispiel dafür ist das Pathfinder-Framework [9], das ein Datenformat für den Austausch von Treibhausgasdaten entlang der Lieferkette festlegt und Standards sowohl für die eigentlichen Emissionsdaten als auch für begleitende Metadaten bezüglich der angesprochenen Themen wie Systemgrenzen und Datenqualität vorgibt.

In der vierten Phase erfolgt die Implementierung oder Erweiterung des Informationssystems, das den Geschäftsprozess unterstützt. Dabei sollte eine möglichst hohe Integrationstiefe angestrebt werden. Damit ist gemeint, dass Nachhaltigkeitsdaten möglichst tief in die Steuerungs- und Managementsysteme des Unternehmens integriert sein sollten. Bestehende Informationssysteme für das Nachhaltigkeitsmanagement haben oftmals eine eher geringe Integrationstiefe, da sie lediglich die Erstellung von jährlichen Nachhaltigkeitsberichten unterstützen und eine manuelle Dateneingabe erfordern. Demgegenüber steht das Idealbild eines Informationssystems, das Prozessverantwortlichen im Einkauf, in der Produktion oder in anderen Abteilungen tagesaktuelle Nachhaltigkeitsdaten zuführt und so auch im Unternehmensalltag als Entscheidungsunterstützung dienen kann. Eine Umsetzung dieses Idealbildes kann mit prozessorientierten Informationssystemen gelingen. Sie bieten die Möglichkeit, erhobene Nachhaltigkeitsdaten direkt den modellierten Aktivitäten des Geschäftsprozesses zuzuordnen und die gewonnenen Informationen unmittelbar zur Analyse, Steuerung und Verbesserung der Geschäftsprozesse zu nutzen. Dieser Ansatz ermöglicht eine nahtlose Integration von Nachhaltigkeitsdaten in die bestehenden Geschäftsabläufe und eröffnet somit neue Wege für eine ganzheitliche Betrachtung der Nachhaltigkeitsleistung des Unternehmens.

Wie zu Beginn erwähnt, kann die Anwendung des Vorgehensmodells dazu dienen, das Nachhaltigkeitsmanagement eines Unternehmens schrittweise zu verbessern. Dieser Ansatz wird durch das Konzept der Verbesserungsmerkmale im Vorgehensmodell unterstützt. Über eine Verbesserung des Informationssystems bezüglich dieser Merkmale kann eine effektivere Steuerung der Nachhaltigkeitsauswirkungen des Un-



ternehmens erreicht werden. Die Verbesserungsmerkmale wurden in den vorangegangenen Erläuterungen bereits erwähnt und werden hier zur Übersicht noch einmal abschließend zusammengefasst. Die Reihenfolge der Verbesserungsmerkmale in *Abbildung 1* spielt dabei keine Rolle. In jeder Iteration kann das betreffende Informationssystem bezüglich eines oder mehrerer Merkmale verbessert werden:

- Hinsichtlich des Merkmals Prozessabdeckung ist das Ziel, nicht nur einzelne Geschäftsprozesse, sondern möglichst alle Unternehmensprozesse in das Nachhaltigkeitsmanagement einzubeziehen.
- Die Granularität der Betrachtung kann dabei variieren, sollte aber möglichst hoch sein. Beispielsweise kann es in einem ersten Schritt genügen, die Treibhausgasemissionen auf Basis von Einkaufsdaten und Treibhausgas-Faktoren für das gesamte Unternehmen oder Teilbereiche abzuschätzen.
- Eine schrittweise Ausweitung der betrachteten Nachhaltigkeitsaspekte ist ebenfalls möglich. Beginnend mit dem Thema der Treibhausgasemissionen, das aufgrund seiner Dringlichkeit naheliegender ist, könnten nach und nach weitere Nachhaltigkeitsaspekte hinzugezogen werden, je nach Anwendungsfall.
- Eine weitere Verbesserungsdimension betrifft die Systemgrenzen. Eine Nachhaltigkeitsbetrachtung, die über das Unternehmen hinausgeht (beispielsweise Scope 3 des Treibhausgasprotokolls), bringt weit höhere Herausforderungen bei der Datenerhebung mit sich als eine Betrachtung innerhalb der Unternehmensgrenzen. Im ersten Schritt kann ein Unternehmen daher auch mit Scope 1 und Scope 2 starten. Dabei muss jedoch stets transparent sein, dass damit eine aus Nachhaltigkeitssicht sehr unvollständige Betrachtung erfolgt.
- Bezüglich der Datenqualität wäre ein anzustrebendes Ideal die Integration von Echtzeitdaten zu jedem Geschäftsprozess des Unternehmens. Wo diese nicht verfügbar sind, kann zunächst auch mit der Integration von generischen Daten aus Ökobilanzierungsdatenbanken begonnen werden.
- Schließlich sollte eine möglichst hohe Integrationstiefe der Nachhaltigkeitsdaten mit den Steuerungs- und Managementsystemen des Unternehmens angestrebt



## Dir ist wichtig, spannende Projekte zu betreuen?

### Werde IT-Berater:in und Web-Entwickler:in für Oracle APEX.

Wir suchen dich als IT-Berater:in und Web-Entwickler:in (w/m/d) für Oracle APEX am Standort Berlin, Erfurt, Frankfurt oder als „Wo du willst-Job“ aus dem mobilen Standort deiner Wahl.

#### Deine Aufgaben:

- Als IT-Berater:in/Web-Entwickler:in mit Datenbankerfahrung und dem Schwerpunkt Oracle APEX hast du den maximalen Kundennutzen im Blick
- Du bist zuständig für Beratung und Erstellung von Geschäftsprozessen und deren iterative Umsetzung von der Idee bis zur produktiven Webanwendung

#### Dein Profil:

- Basis deines Erfolgs ist ein abgeschlossenes IT-orientiertes Studium oder du hast eine IT-orientierte Berufsausbildung absolviert und verfügst über einschlägige Berufserfahrung

#### Deine Vorteile:

- Freifahrten
- Homeoffice
- Freizeit sponsored by Bahn
- Attraktive Konditionen



Jetzt unter der Ausschreibungsnummer 221214 bewerben:  
**db.jobs/it-apex**

Was ist dir wichtig?

werden. Prozessorientierte Informationssysteme erscheinen hier als vielversprechender Ansatz.

### Fazit

Das aufgezeigte Vorgehensmodell gibt Orientierung für den Entwurf von nachhaltigkeitsorientierten Informationssystemen. Dabei können je nach Branche und betrachtetem Nachhaltigkeitsaspekt spezifische Herausforderungen auftreten, die mit dem vorgestellten Modell noch nicht vollständig gelöst, aber dennoch eingeordnet werden können. Im Projekt „Scope3transparent“ werden spezifische Lösungen und Software-Werkzeuge im Bereich Treibhausgasemissionen entwickelt. Interessierte Unternehmen können unter [www.scope3transparent.de](http://www.scope3transparent.de) Unterstützungsangebote finden. Es wird unter anderem ein Praxisleitfaden für die Bilanzierung von Scope 3-Treibhausgasemissionen und eine Übersicht zu verfügbaren Softwarewerkzeugen bereitgestellt. Diese Angebote sind auf Unternehmen in der Elektronikbranche zugeschnitten, können aber auch Anhaltspunkte für Unternehmen anderer Branchen bieten. Unabhängig von der Branche sollten Unternehmen, die ein nachhaltiges Geschäftsmodell verfolgen, die tiefe Integration von Nachhaltigkeitsdaten in ihre Informationssysteme anstreben. Eine umfangreiche, prozessorientierte Steuerung und Verbesserung ihrer Nachhaltigkeitsauswirkungen sollten dabei im Fokus stehen.

### Förderung

Das Projekt Scope3transparent wird im Rahmen der Nationalen Klimaschutzinitiative vom Bundesministerium für Wirtschaft und Klimaschutz aufgrund eines Beschlusses des Deutschen Bundestages gefördert.

### Quellen

- [1] United Nations General Assembly, Report of the world commission on environment and development: Our common future (A/42/427). 1987.
- [2] S. Connelly, „Mapping Sustainable Development as a Contested Concept“, Local Environment, Bd. 12, Nr. 3, S. 259–278, 2007, doi: 10.1080/13549830601183289.
- [3] E. Heiskanen, „The institutional logic of life cycle thinking“, Journal of Cleaner Production, Bd. 10, Nr. 5, S. 427–437, Okt. 2002, doi: 10.1016/S0959-6526(02)00014-8.
- [4] Deutschlandfunk, „Kehrseite der Energiewende: Lithium-Abbau in Südamerika“. 2019. [Online]. Verfügbar unter: <https://www.deutschlandfunk.de/>

*lithium-abbau-in-suedamerika-kehrseite-der-energiewende-100.html*, abgerufen am 08.08.2023

- [5] ISO, „ISO 14040:2006 Environmental management - Life cycle assessment - Principles and framework“. 2006.
- [6] UNEP und SETAC, Guidance on organizational life cycle assessment. Paris, France: UNEP, 2015.
- [7] WBCSD und WRI, „The greenhouse gas protocol: A corporate accounting and reporting standard“. 2004.
- [8] W. Klöpffer und B. Grahl, Ökobilanz (LCA): Ein Leitfaden für Ausbildung und Beruf. Weinheim: Wiley-VCH, 2009.
- [9] WBCSD, „Pathfinder Framework: Guidance for the Accounting and Exchange of Product Life Cycle Emission“. 2021.

**Andreas Fritsch**  
[andreas.fritsch@kit.edu](mailto:andreas.fritsch@kit.edu)

Andreas Fritsch ist wissenschaftlicher Mitarbeiter am Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) am Karlsruher Institut für Technologie (KIT). Er forscht und lehrt zu Themen an der Schnittstelle zwischen Informationstechnologie und Nachhaltigkeit mit Schwerpunkten im Bereich nachhaltigem Geschäftsprozessmanagement, Software Engineering und Datenbanksystemen.



# Nachweis ökologischer Nachhaltigkeit in Geschäftsmodellen der Luftfahrt – Herausforderungen meistern durch Einsatz geeigneter IT-Lösungen

Von Dr. Thomas Mützel-von Schwartz, SkyTender AG und Florian Lösch, Prociris consulting GmbH

*Nachhaltigkeit nachzuweisen ist nicht einfach, denn sie ist vielschichtig und bietet Spielraum für Interpretationen. Bereits das Verständnis einer messbaren Größe zum allgemeinen Ziel höherer Nachhaltigkeit ist nicht einheitlich. Dennoch wird es mehr und mehr zu einem wichtigen Kriterium bei der Kaufentscheidung von Endkunden in gesättigten und umkämpften Kundenmärkten. Dies wiederum hat einen Einfluss auf die Kaufentscheidungen im Geschäftskundenumfeld entlang der vorangelagerten Lieferkette. Einen hinsichtlich Nachhaltigkeit häufig kritisch beäugten Industriezweig stellt hierbei die Luftfahrt dar. Dennoch hat es sich SkyTender Solutions mit ihrer Produktpalette zum Ziel gesetzt, eine disruptive Revolution im Luftfahrt-Catering in Gang zu setzen, um umweltschädliche Einflüsse signifikant zu reduzieren. Um hierbei Entscheidungsträger zu überzeugen, sind neben kommerziellen Aspekten auch der Nachweis der Nachhaltigkeit des neuen Produktes und des damit verbundenen Geschäftsmodells notwendig. Dies wurde im vorliegenden Fall mit einer vergleichenden LCA-Studie durchgeführt. Der nachfolgende Bericht zeigt die Herausforderungen der Studiendurchführung und die Hilfestellung, die moderne IT-Lösungen hierbei bieten können.*



### Einführung in die Formen der Nachhaltigkeit und in nachhaltige Geschäftsmodelle

Nachhaltigkeit und nachhaltiges Handeln rückten in den letzten Jahren mehr und mehr in den Fokus der Öffentlichkeit. Dies wurde unterstützt durch Aktionsbündnisse wie Fridays for Future, welche die gesellschaftliche Aufmerksamkeit durch friedliche Protestaktionen auf sich ziehen oder die Vielzahl der verschiedenen Ökolabels auf Produkten und deren kritische Diskussion. Dieser nachhaltige Diskurs findet fortschreitend Einzug in politische Debatten und Gesetzesinitiativen und wird somit zunehmend verbindlicher für Unternehmen.

Dabei ist die Definition von Nachhaltigkeit nicht trivial. Die erste allgemein aner-

kannte Formulierung stellt die im Brundtland Report 1987 veröffentlichte dar. Sie beschreibt Nachhaltigkeit als ein Handeln, welches gewährleistet, dass die Bedürfnisse der heutigen Generation befriedigt werden, ohne die Fähigkeit künftiger Generationen zu gefährden, ihre eigenen Bedürfnisse zu befriedigen [1]. Seitdem wurden allein bis 2007 etwa 300 verschiedene Definitionen von Nachhaltigkeit und nachhaltiger Entwicklung gezählt, wobei die Tendenz stetig steigend ist [2]. Durchgesetzt hat sich hierbei die Definition, dass Nachhaltigkeit aus drei Dimensionen besteht – der ökonomischen, der ökologischen und der sozialen Nachhaltigkeit.

Die Dimension der Ökonomie setzt sich in diesem Kontext mit der Sicherung eines langfristigen und kontinuierlichen ökonomischen Wachstums auseinander. Dieses soll das Einkommen aller durch den sorgfältigen Umgang mit bestehenden Ressourcen sicherstellen. Es ist jedoch essenziell, dass ein Wachstum ermöglicht wird.

Soziale Nachhaltigkeit beschäftigt sich mit dem Menschen als Individuum und soziales Wesen. Hierbei wird der Fokus auf die Erfüllung der Grundbedürfnisse der Menschen gelegt, sodass Gleichberechtigung und faire Behandlung aller sichergestellt ist. Hierzu zählen auch die Achtung der Menschenrechte, die Bereitstellung von Bildungsmöglichkeiten für jedermann und die allgemeine Sicherheit sowie eine allgemeine Gesundheitsversorgung.

Zur ökologischen Nachhaltigkeit gehört nicht nur die ökonomische Notwendigkeit, das vorhandene Naturkapital zu erhalten, sondern auch die Sicherung der ökologischen Bedingungen für das menschliche Überleben im Allgemeinen. Giftige Stoffe, die zum Beispiel durch eine verschwenderische Produktion in die Umwelt gelangen, spielen eine wichtige Rolle bei der Erzeugung von Risiken für die menschliche Gesundheit und der Gefährdung der biologischen Vielfalt, die eng mit der Bedrohung des menschlichen Lebens und der Umwelt verbunden sind [3]. Wie aus *Abbildung 1* entnommen werden kann, die die von Menschen verursachten Treibhausgasemissionen nach Sektoren eingeteilt ausgibt, stellen Industrieemissionen ein Fünftel der Gesamtemissionen dar. Um diese zu vermindern sind nachhaltige Geschäftsmodelle von Nöten.]

Diese zeichnen sich dadurch aus, dass sie sich mittelfristig selbst tragen müssen, für

Mensch und Natur einen positiven Beitrag leisten und ebenfalls die weiteren Stakeholder des Unternehmens begünstigen. Dies kann beispielsweise durch die Minimierung des ökologischen Fußabdrucks und die gleichzeitige Übernahme von sozialer und wirtschaftlicher Verantwortung erfolgen. Es lässt sich somit sagen, dass durch neuartige Produkte oder umgestaltete Wertschöpfungsketten ein positiver Beitrag für Viele erzeugt werden muss [5].

### Nachhaltigkeit messbar machen

Um diese Optimierung jedoch gestalten zu können, ist es notwendig, sich des Status quo bewusst zu werden. Hier haben sich in den vergangenen Jahren verschiedene Formen der Nachhaltigkeitsbewertung herausgebildet. Diese basieren entweder auf wissenschaftlichen Methoden, auf gesetzlichen oder öffentlichen Initiativen.

*Abbildung 2* gibt eine Übersicht über verschiedene Bewertungsmethoden, hat aber nicht den Anspruch alle verfügbaren Ansätze aufzulisten und wissenschaftlich fundiert einzuteilen. Vielmehr verdeutlicht sie, dass es mit den Unterscheidungen zwischen produkt- oder organisationsbasiert und zwischen sozialem oder ökologischem Fokus zwei verschiedene Klassifizierungen gibt. Die Unterscheidung zwischen Produkt und Organisation legt hierbei den Schwerpunkt auf das Bewertungsobjekt und ist zumeist sehr eindeutig, ob etwas tendenziell eher für die Bewertung eines Produktes oder einer Dienstleistung herangezogen werden kann oder, ob eine Organisation als Ganzes betrachtet wird. Lediglich die UN Sustainable Development Goals (SDG) finden auf Grund ihrer Gestaltung als Meta-Kriterien zum Teil in beiden Bereichen Anwendung. Die Differenzierung zwischen sozialen und ökologischen Bewertungssystemen ist nicht derart klar möglich. Viele der Standards berücksichtigen beide Gebiete, legen hierbei aber unterschiedliche Schwerpunkte. Die einzelnen Konventionen lassen sich wie folgt kurz beschreiben:

- CSR: Corporate Social Responsibility beschreibt die Verantwortung von Unternehmen gegenüber sozialen, ökologischen und ökonomischen Aspekten.
- CSRD: Die Corporate Sustainability Reporting Directive der EU-Kommission definiert die Anforderungen und die Anwender der Nachhaltigkeitsberichterstattung ab 2024.

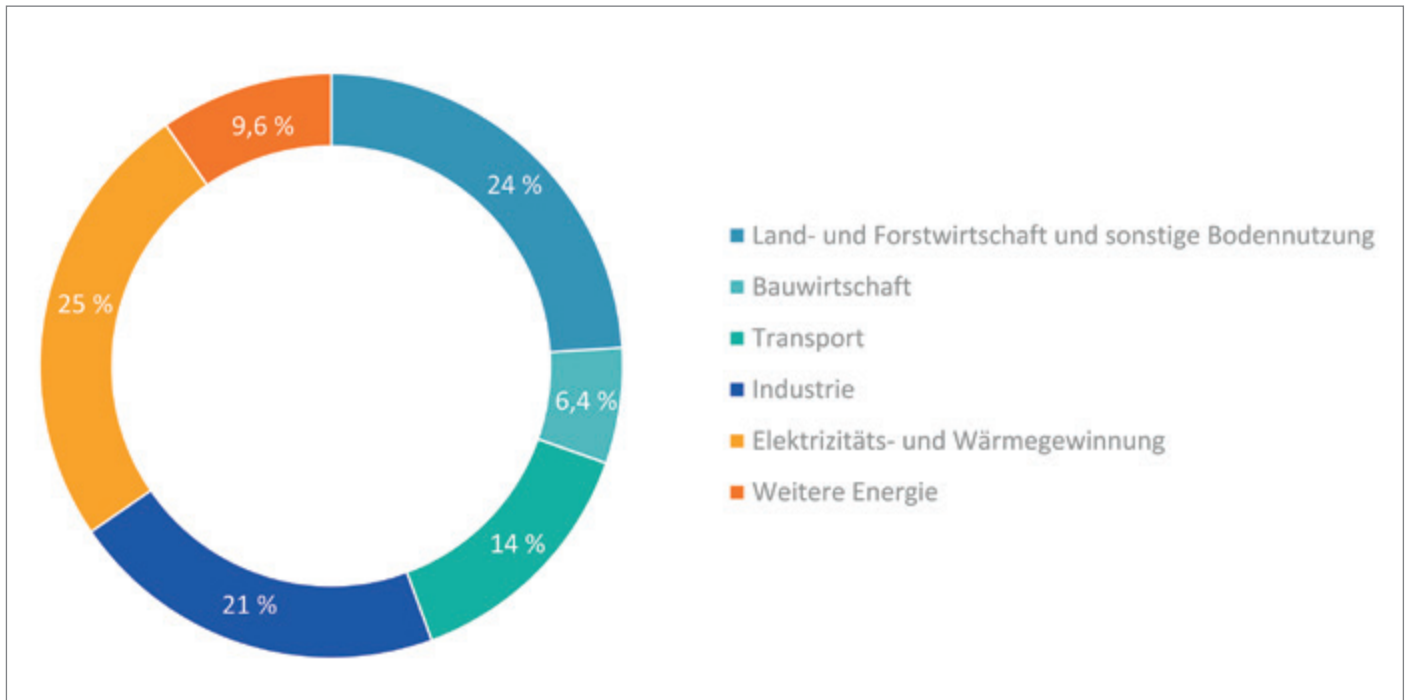


Abbildung 1: Greenhouse Gas Emissionen nach Sektoren angelehnt an [4]

- ESG: ESG steht für Environmental, Social, Governance und definiert einen Kriterienkatalog, welchen nachhaltige Unternehmen berücksichtigen müssen.
- ESRS: Die EU Sustainability Reporting Standards sind die für alle EU-Unternehmen geltenden Standards der Nachhaltigkeitsberichterstellung.
- GHG Protocol: Das Greenhouse Gas Protocol stellt einen international anerkannten Standard zur Ermittlung von Treibhausgasemissionen dar.
- GRI: Die Global Reporting Initiative ist ein Anbieter von international anerkannten Richtlinien zur Publizierung von Nachhaltigkeitsberichten.
- LCA: Die Life-Cycle-Analysis wird eingesetzt, um die ökologische Nachhaltigkeit eines Produktes nachzuweisen.
- OEF: Der Organisation Environmental Footprint basiert auf dem PEF und dient zur Bewertung von Organisationen.
- O-LCA: Die Organisational-LCA basiert auf der LCA-Methode und dient zur Bewertung von Organisationen.
- PEF: Der Product Environmental Footprint baut auf der Methode der LCA auf und wird bis Ende 2024 als EU-Standard für die nachhaltige Bewertung von Produkten festgelegt.
- SDG: Die Sustainable Development Goals setzen sich aus 17 Zielen zusammen, die die Agenda 2030 der UN beschreiben.

- SLCA: Die Social-LCA beschäftigt sich anders als die LCA mit der sozialen Nachhaltigkeit von Produkten und berücksichtigt hierbei auch Bewertungskriterien, welche die Gesamtorganisation betreffen.

### SkyTender und die Wahl der passenden Methode

Die SkyTender AG ist ein in Deutschland ansässiges Unternehmen, welches mit dem Ziel gegründet wurde, das Catering in der Luftfahrt zu revolutionieren. Am Standort Herborn wird für diesen Zweck ein Produktportfolio entwickelt und produziert, das aus vier Kernelementen besteht, welche entlang der kompletten Wertschöpfungskette des Getränkercatering-Prozesses eingesetzt werden:

- Getränkedispenser (mobil und fest installiert)
- Isolierter Wassertank
- Wasseraufbereitungs- und -abfüllanlage
- IT-Backbone zur Bereitstellung verschiedener Informationen

Der Getränkedispenser besteht aus einem Flugzeug-Trolley, der in der Luftfahrt weit verbreiteten Standardklasse ATLAS, welcher die für die Herstellung von heißen und kalten Softdrinks sowie karbonisiertem oder stillem, kaltem und heißen Wasser notwendigen technologischen Elemente beinhaltet.

Das allgemeine Funktionsprinzip dieses Getränkewagens basiert auf der Logik eines Postmix-Getränkebereiters. Diese Systeme verwenden Wasser und Sirupkonzentrat unterschiedlichen Geschmacks, um diese in einem bestimmten Verhältnis zu einem Getränk zu mischen. Das Wasser kann vorab mit Kohlensäure versetzt werden (Limonaden), kann aber auch als stilles Wasser prozessiert werden (Säfte).

Der Trolley verfügt dafür über einen bis zu vier austauschbaren, isolierten Wassertanks mit zehn Litern Fassungsvermögen. Für aromatisierte Getränke (Coca-Cola etc.) fasst das System bis zu acht austauschbare Ein-Liter-Beutel mit Sirupkonzentrat. Für kohlenensäurehaltige Getränke ist das System mit einer austauschbaren Druckflasche sowie einem Karbonisierungssystem ausgestattet.

Getränke werden ausgegeben, indem der Benutzer das gewünschte Getränk über ein Touch-Display auswählt. Bei der Bestätigung des gewünschten Getränks wird das Wasser aus dem Tank angesaugt und von einer Pumpe durch einen Durchflusskarbonator verarbeitet. Gleichzeitig mit dem Beginn des Wasserflusses öffnet ein Ventil die CO<sub>2</sub>-Leitung und liefert einen definierten CO<sub>2</sub>-Volumenstrom, der in den Inline-Karbonisator gelangt, wo er mit dem Wasser vermischt wird. Dadurch entsteht kohlenensäurehaltiges Wasser, bevor es durch einen Auslass gegossen wird.



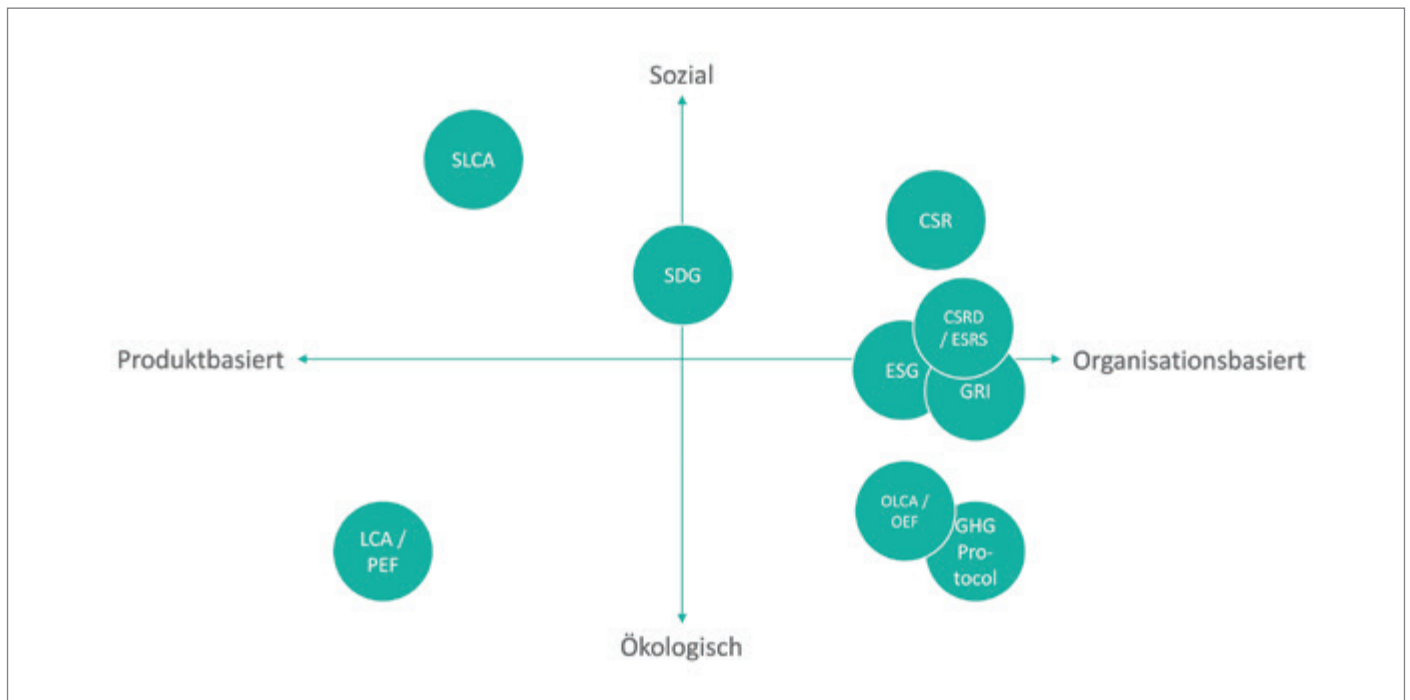


Abbildung 2: Klassifizierung von Methoden zur Nachhaltigkeitsbewertung (© PROMATIS software GmbH)

Bei stillen Getränken wird der Karbonator umgangen, sodass die Pumpe das Wasser aus dem Wassertank direkt zum Auslass fördert. Das System kann Getränke in der heute üblichen Menge von 150 Milliliter pro Becher abfüllen, ist aber in der Ausgabemenge grundlegend flexibel einstellbar.

Der Ausgabekopf kann vollständig im Trolley verstaut werden, um eine ordnungsgemäße Aufbewahrung in der Bordküche zu ermöglichen.

Die Bereitstellung des vorgekühlten Wassers erfolgt über eine Wasseraufbereitungs- und -abfüllanlage, die im Cateringbereich eines Flughafens stationiert ist. Abhängig von der vor Ort verfügbaren Wasserqualität umfasst die sogenannte Water Supply Station (WSS) die Prozessstufen Filtrierung, Aufbereitung, Kühlung und Abfüllung. Die bereits erwähnten Zehn-Liter-Wassertanks können vor dem Befüllen in der Station zusätzlich gereinigt werden.

Um ein effizientes und sicheres Zusammenspiel der drei Komponenten Trolley, Tank und WSS zu gewährleisten, werden sie über ein IT-Backbone und eine entsprechende Managementinfrastruktur miteinander verbunden. Dabei können einerseits funktionale Daten im Sinne der präventiven Wartung insbesondere für Trolley und WSS verarbeitet, aber auch Konsumdaten gemanagt werden, die dazu führen, dass nach einer initialen Verbesserung des Catering-Prozesses ein kontinuierliches Optimum

für die Versorgung der jeweiligen Flotte gewährleistet werden kann.

Ein Sirupbeutel ersetzt grob 28 Getränkedosen des Volumens 0,33 Liter. Diese Reduzierung bewirkt eine Gewichtseinsparungen im Flugzeug, aber zusätzlich deutliche Aufwandsreduzierung im Catering- und Handling-Prozess am Boden sowie in der Luft. Hinzu kommt die grundlegende Einsparung an Overcatering, da das System die Mitnahme von einer für die Gesamtnachfrage notwendige Menge Wasser für alle nichtalkoholischen Getränke ermöglicht und die Schwankungen in der Nachfrage nach Softdrinks über deutlich leichtere und konzentrierte Sirupe abdeckt. Zuletzt erspart die Versorgung mit lokalem Trinkwasser aufwändige und umweltschädliche Transportprozesse und Lagerstufen.

Im kontinuierlichen Nutzungsprozess kann über die Systemverbesserung bei Einführung der mobilen Getränkelösung kontinuierlich am Optimum der Versorgung gearbeitet werden. Von Standardbeladung bis hin zu Milk-Run ähnlicher Versorgung mit frischen Wassertanks und Sirupen im Vorfeld kann Stück für Stück ein Kosten-Nachhaltigkeits-Optimum erarbeitet werden. Hierzu bieten die Nutzungsinformationen aus den IOT-fähigen Geräten eine entsprechende Ausgangsbasis.

Die Luftfahrtbranche kann hinsichtlich ihrer Innovationsbereitschaft als konservativ klassifiziert werden. Dies ist in erster Li-

nie auf die hohen Sicherheitsanforderungen und den damit enormen Kosten für die Entwicklung, Zulassung und Produktion neuer Produkte zurückzuführen. Um in einem solchen Umfeld eine disruptive Veränderung voranzutreiben, müssen die Argumente klar und belastbar sein.

Während die Kostenvorteile der vorgestellten Getränkelösung auf Basis eines herkömmlichen Business Cases mit Vergleichsdaten aus dem heutigen Getränke-Catering verschiedener Airlines gerechnet werden können, und damit über Zweifel erhaben sind, gab es für den Bereich Nachhaltigkeit keine solche Berechnungsgrundlage.

Die Einsparung lässt sich qualitativ mit dem Verhältnis 1:28 Beutel:Dose zwar leicht erläutern, jedoch bedarf es für ein klares Verständnis der Vorteile und eine umfassende Analyse aller Systemkomponenten, die zur Bereitstellung eines Getränkes heute und mit der Lösung von SkyTender benötigt werden.

Insofern war es notwendig, für den Vergleich ein wissenschaftliches ebenso einwandfreies wie umfassendes Vorgehen zu nutzen, dass durch die Unabhängigkeit des Anbieters auch auf höchsten Entscheidungsebenen Bestand hat. Dementsprechend fiel die Wahl auf das produktbasierte Bewertungsverfahren einer vergleichenden Life Cycle Analysis (LCA), mit der der herkömmliche Prozess der Getränkebereitstellung im Flugzeug mit dem disruptiven Ansatz von SkyTender verglichen wurde.



Abbildung 3: SkyTender AG Produktvision (© SkyTender AG)

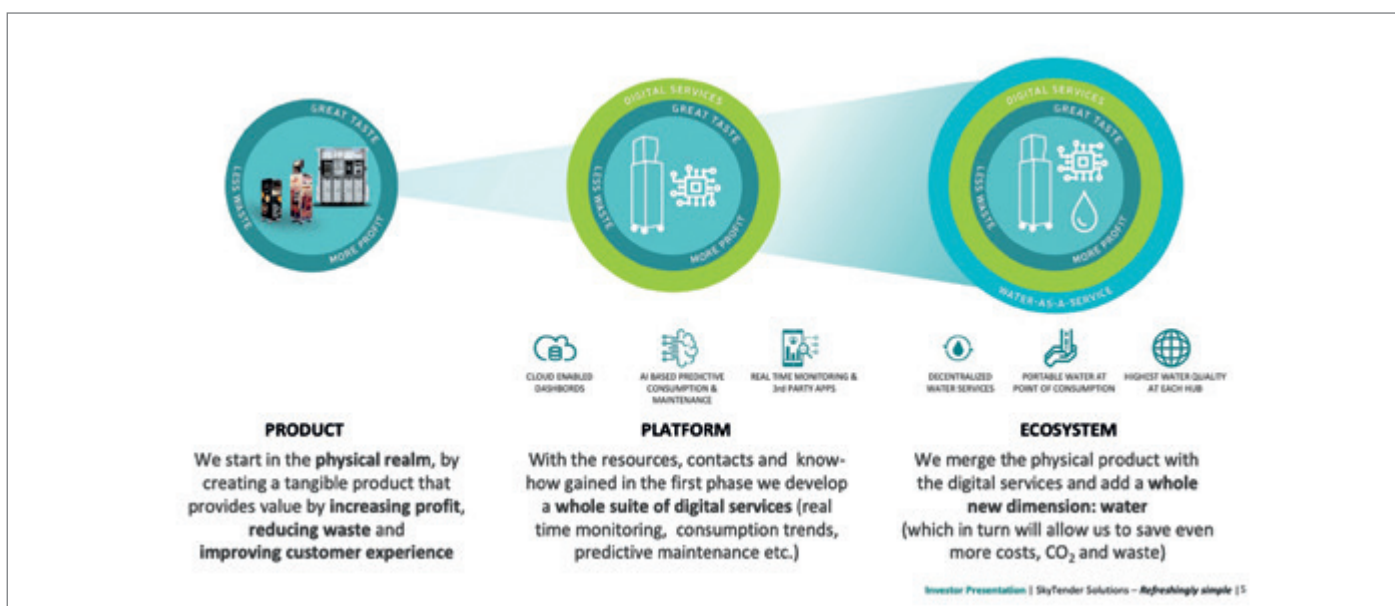


Abbildung 4: Das skalierbare SkyTender Ecosystem (© SkyTender AG)

### Einführung in das LCA-Vorgehen

Die LCA beziehungsweise deren artverwandter Product Environmental Footprint (PEF) werden in diversen EU-Direktiven wie der Waste Directive, dem European Green Deal, dem Circular Economy Action Plan, der EU Taxonomy und der in Entwicklung befindlichen Green Claims Directive aufgeführt. Sie stellt somit eine sowohl wissenschaftlich als auch legislativ anerkannte Methode zur Bewertung von ökologischer Nachhaltigkeit von Produkten und Dienstleistungen dar. Hierbei wird der gesamte Lebenszyklus von der Beschaffung bis zur

endgültigen Entsorgung oder idealerweise der nochmaligen Nutzung abgebildet. Das Vorgehen zur Erstellung einer LCA besteht aus den nachfolgenden vier Prozessschritten und wird in der ISO-Norm 14040 eindeutig beschrieben:

- Goal Definition & Scoping
- Inventory Analysis
- Impact Assessment
- Interpretation

Hierbei stellt die Definition des Ziels und des Umfangs der Studie die erste Phase dar. In

diesem Schritt wird festgelegt, welche Produkte bewertet und welche Prozesse entlang der Wertschöpfungskette betrachtet werden. Im Fall von SkyTender wurde hierzu der Prozess des Getränketerings vom Beschaffen des Grunderzeugnisses Wasser inklusive der verschiedenen Geschmacksrichtungen mit der Erzeugung, Verpackung sowie dem Handling und dem Transport, bis es im Flugzeug ist, die Nutzung im Flieger bis zur Entsorgung oder Wiederverwendung des nicht genutzten Getränkes und der verwendeten Verpackung als sogenannte funktionale Einheit berücksichtigt.

Zugleich wurde definiert, dass der Prozess des Caterings von verschiedenen Flugscenarien mit Kurzstrecken-, Mittelstrecken- und Langstreckenflügen zwischen dem von einer großen europäischen Airline genutzten System mit der SkyTender-Lösung verglichen wird. Hierbei musste sichergestellt werden, dass die Passagiere der auf den jeweiligen Strecken eingesetzten und vollbesetzten Flugzeuge eine einheitliche Getränkeversorgungsqualität erhalten. Somit lässt sich von einem normativen Szenario sprechen [6, 7, 8].

Die Inventory Analysis der LCA wird häufig als die arbeits- und zeitintensivste Phase der Studiererstellung dargestellt. Dies ist der Fall, da in verschiedenen Datenquellen die Rohdaten der Analyse gesucht, extrahiert und abschließend aufbereitet werden müssen, um das analysierte System innerhalb seiner vorab definierten Grenzen sauber zu beschreiben. Wenn hierbei nicht ausreichend Daten zur Verfügung stehen, müssen die Wissenslücken durch die Nutzung von LCA-Datenbanken geschlossen werden. Dies können öffentliche, nationale oder regionale Datenbanken, aber auch spezielle industriebezogene oder von Beratungshäusern in Verbindung mit LCA-Software angebotene Datenbanken sein [6]. Das Ergebnis sind Inventartabellen, welche die Inputs aus der Umwelt und die Outputs in die Umwelt sowie deren Mengen in Bezug auf die funktionale Einheit auflisten [7, 8].

Im beschriebenen Beispiel kam für die Analyse des SkyTender-Systems in dessen Entstehungs- und Entsorgungsprozess das PLM-System des Unternehmens zum Einsatz, um hieraus die Stücklisten der Pro-

dukte zu erfahren. Mithilfe der im DMS abgelegten Produkt-Datenblätter wurden die spezifischen Inhaltsstoffe der genutzten Einzelkomponenten ergänzt. Zusätzlich diente das Enterprise-Resource-Planing-System (ERP) dazu, die Lieferanten und die damit verbundenen Lieferwege zu identifizieren sowie durch die Analyse von Rechnungsdaten die Energienutzung im Montageprozess zu bewerten. Die Daten für die In-Flight-Bewertung stammen von verschiedenen Testflügen. Die Vergleichsdaten des bestehenden Systems bestehen aus ERP- und BDE-Daten der Airline zur Warenbeschaffung sowie über Waren-Handling- und Entsorgungsdienstleistungen. Diese wurden durch manuelle Zählungen während Testflugvorbereitungen ergänzt.

Das anschließend durchzuführende Impact Assessment umfasst die Modellierung der funktionalen Einheit, um die Bewertung der anzuwendenden Wirkungskategorien zu erstellen. Wirkungskategorien sind beispielsweise das Ausmaß des Abbaus der Ozonschicht, die Ökotoxizität des Produktes – also das Ausmaß des Ausstoßes von für die Umwelt giftigen Substanzen – oder dessen Humantoxizität, – das heißt, der Umfang der Belastung eines individuellen Menschen durch das Produkt. In dieses Modell werden die Daten der Inventory Analysis in normierter Form eingegeben, um das Ergebnis für einen konkreten Fall zu erhalten [7, 8]. Im Anwendungsbeispiel von SkyTender kam an dieser Stelle die GABI-Software zum Einsatz. Dabei wurden zwei Systeme modelliert, sodass deren Ergebnisse unabhängig voneinander erhoben werden konnten. Der anschließende Vergleich

dieser erfolgte auf Basis der Wirkungskategorien. Hierfür wurden alle in einer LCA vorgesehenen Kategorien genutzt.

Die erhaltenen Ergebnisse und die diesen zu Grunde liegenden Entscheidungen und Annahmen werden in der anschließenden Phase interpretiert und auf ihre Plausibilität geprüft. Hieraus werden Schlussfolgerungen gezogen, einschließlich einer Bewertung der Ergebnisse im Hinblick auf ihre Konsistenz und Vollständigkeit sowie einer Analyse der Ergebnisse im Hinblick auf ihre Robustheit. Schließlich werden Schlussfolgerungen und Empfehlungen formuliert. Dies ist notwendig, da Ergebnisse von LCA-Studien im Allgemeinen aufgrund der subjektiv zu treffenden Annahmen mit vielen Unsicherheiten behaftet sein können. Um diese Unsicherheit zu reduzieren, empfiehlt es sich, geeignete Maßnahmen wie die Anwendung anerkannter wissenschaftlicher oder praktischer Standards zu nutzen, möglichst vollständige Daten bereitzustellen und eine konsistente Datenbasis für das Füllen der Lücken zu nutzen [7, 8].

Im vorliegenden Fall wurde die Interpretation in Workshops zwischen der wissenschaftlichen Unterstützung und dem operativen Auftraggeber durchgeführt. Zusätzlich erfolgte die Verifikation durch ein Peer-Review bestehend aus drei erfahrenen LCA-Anwendern aus Wissenschaft und Forschung.

### Ergebnisse der Studie und Nutzen für SkyTender

Zusammenfassend lassen sich die Ergebnisse wie in *Abbildung 5* darstellen.

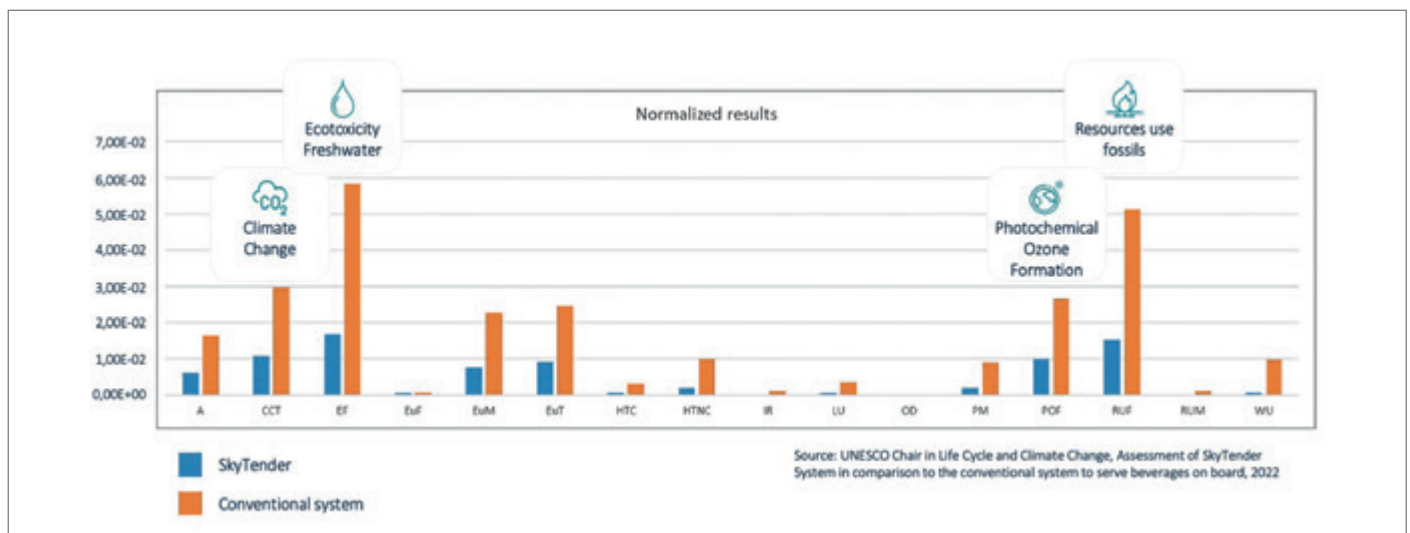


Abbildung 5: Normierte Ergebnisse der relevantesten Einflussfaktoren (© SkyTender AG)



Abbildung 6: Die relevantesten Einsparungen für Kunden der Luftfahrtindustrie (© SkyTender AG)

Das SkyTender-System erzielt dementsprechend in allen untersuchten Kategorien bessere Werte als konventionelle Catering-Modelle.

Diese lassen sich aufgrund der wissenschaftlichen Vorgehensweise und durch fünf unabhängige Prüfer bestätigten Ergebnisse gut im notwendigen Change Prozess als Argumentationsbasis nutzen.

Abbildung 6 zeigt die aufbereiteten Informationen, mit denen SkyTender einerseits seinen Unternehmenszweck bestätigt sieht und andererseits seit Finalisierung der Studie auf deutlich gestiegenes Interesse im Zielmarkt Aviation trifft. Dies verdeutlicht sich dadurch, dass mit Hilfe der Studienergebnisse die Akzeptanz der Lösung am Markt derart gesteigert werden konnte, dass innerhalb eines Jahres nach Veröffentlichung dieser erste Airlines den Einsatz der neuen Technologie im Linienflugverkehr umgesetzt haben.

### Herausforderungen und Optimierungsansätze durch moderne IT

Die Herausforderungen, die einem beim Erstellen einer Life-Cycle-Analyse begegnen, sind vielseitig und im Fall von SkyTender aufgrund des vergleichenden Aufbaus der Studie nochmal speziell. Hier lassen sich, neben dem schwierigen Vergleich einer Dienstleistung mit einem Produkt, Punkte wie die aufwändige Datenbeschaffung bei verschiedenen Partnern entlang der heutigen und zukünftigen Wertschöpfungskette, fehlende Sekundärdaten in bestehenden LCA-Datenbanken oder auch das manuelle

Modellieren und Befüllen der Berechnungssoftware nennen. Moderne IT-Systeme können hierbei in manchen, jedoch nicht in allen Bereichen Abhilfe schaffen. In an das Projekt anschließend stattfindenden Lessons Learned Workshops wurden vor allem die Wiederholbarkeit der Studie, die Optimierung der Datenqualität und des Datenerhebungsaufwands sowie die Automatisierung der Inventardatenverarbeitung hin zum Endergebnis identifiziert.

Im Bereich der Wiederholbarkeit der Studiererstellung hat sich auf Grund der positiven Rückmeldungen des Marktes aus der Studie der Bedarf ergeben, ähnliche Studien für weitere Anwendungsfälle durchzuführen. Dies betrifft zum einen das Anwenden des bewerteten Produktes auf Use Cases anderer Airlines und zum anderen die Erhebung der Nachhaltigkeitskennzahlen der anderen SkyTender-Produkte. Um die Datenerhebung für weitere potenzielle Kunden zu vereinfachen, bestehen zwei Ansatzpunkte: Zum einen bringt Process Mining die Möglichkeit mit sich, Unternehmensabläufe mit der Process Mining Discovery aus bestehenden Datenpunkten zu erfassen und zu visualisieren. Sofern dies an die bestehende LCA-Software angeschlossen wird, verringert es den Modellierungs- und Diskussionsaufwand, welcher sowohl im Scoping als auch in der Inventory Analysis mit erheblichen Anstrengungen verbunden ist. Zum anderen ermöglichen standardisierte Objektmodelle, welche die benötigten Attribute eindeutig identifizieren, dem Studienteilnehmer und potenziellen Kunden seine Attribute

zielgerichtet aus seinen Bestandssystemen bereitzustellen. Dies kann durch Extraktionstemplates oder direkten Schnittstellen zu Enterprise-Performance-Management-Lösungen (EPM) weiter unterstützt und automatisiert werden. Idealerweise ist hierbei ein vorausschauender Ansatz zu wählen, bei dem die im bestehenden ERP-System fehlenden Attribute frühzeitig hinzugefügt werden, um eine ganzheitliche Datenerfassung zu ermöglichen.

Bezüglich der Optimierung der Datenqualität und der Verringerung des Datenerhebungsaufwandes besteht durch die Integration des Product Lifecycle Management-Systems (PLM) mit dem ERP- und dem Dokumentenmanagementsystem (DMS) bereits ein erster Ansatzpunkt, um Daten kanalisiert von einer Quelle zu beziehen. Jedoch kann die Integration der in Datenblättern vorhandenen Produktinformationen in die ERP-Objektmodellstruktur die Extraktion der Daten nochmals optimieren. Hier stellt Optical Character Recognition (OCR) eine mögliche Hilfe dar, um Datenblätter auslesen zu können. Allerdings benötigt OCR auf Grund der geringen Standardisierung im Aufbau von Produktdatenblätter ein hohes Maß an manueller Kontrolle, um eine ausreichende Datenqualität sicherzustellen. Wobei an dieser Stelle der gezielte Einsatz künstlicher Intelligenz perspektivisch Abhilfe schaffen kann. Darüber hinaus wird empfohlen, das Attributmodell von ERP-Objekten, wie dem Artikel oder dessen Verbindung zum Lieferanten, zu verfeinern, um neben der reinen Ergänzung von weiteren Produktmerkmalen zusätzlich The-

men wie die Distanz zum Lieferanten oder die Aufschlüsselung nach Lieferanten zu erkennen. Auch an dieser Stelle machen standardisierte Extraktions-Templates oder eine zwischengeschaltete EPM-Lösung Sinn.

Um die Automatisierung der Inventardatenverarbeitung hin zum Endergebnis zu optimieren, sollten Möglichkeiten geschaffen werden, um die vorab beschriebenen Endprodukte aus einer optimierten Wiederholbarkeit der Studie sowie deren Datenerhebung und -qualität für die LCA-Software nutzbar zu machen. Hierzu besteht der Bedarf, Prozessmodelle und Inventardaten zielgerichtet importieren zu können. Besonders im Bereich des Datenimports liefern einige LCA-Softwareanbieter bereits weitreichende Möglichkeiten in ihren Lösungen mit.

### Fazit

Somit lässt sich festhalten, dass die LCA-Methode einen in die Lage versetzt, Produkte und Dienstleistungen auf den Grad ihrer Auswirkungen auf die Umwelt zu untersuchen und die damit verbundene ökologische Nachhaltigkeit dieser systematisch und nach anerkannten Standards zu bewerten und zu vergleichen. Sie dient zusätzlich als Grundlage für den Product Environmental Footprint, welcher aktuell als Standard für das von der EU anerkannte produktbasierte Umweltzertifikat eingeführt wird. Dementsprechend kann die LCA-Methode ebenfalls herangezogen werden, um ökologisch nachhaltige Produkte und Geschäftsmodelle zu identifizieren. Beim Beispiel von SkyTender konnte ebenso das große Einsparpotential unter anderem beim Wasserverbrauch, der CO<sub>2</sub>-Emission und dem Freisetzen von für Mensch und Umwelt toxischen Materialien aufgezeigt werden. Der Prozess der Studiererstellung ist jedoch mit enormem manuellem Aufwand verbunden. Hier können mit verschiedenen IT-Lösungen Verbesserungen erzeugt werden. Ganzheitlich wird sich der Prozess jedoch nicht automatisieren lassen, da eine kritische Diskussion der Annahmen und Ergebnissen stets von Nöten ist, um Greenwashing zu vermeiden.

### Literatur

- [1] Gro Harlem Brundtland. „Our Common Future – Call for Action“. In: Environmental Conservation 14.4 (1987), pp. 291–294
- [2] Paul Johnston, Mark Everard, David Santillo, and Karl-Henrik Robèrt. „Reclaim-

ing the Definition of Sustainability“. In: Environmental Science and Pollution Research International 14.1 (2007)

- [3] Cristina Romanelli, David Cooper, Diarmid Campbell-Lendrum, Marina Maiero, William B. Karesh, Danny Hunter, and Christopher D. Golden. Connecting Global Priorities: Biodiversity and Human Health: A State of Knowledge Review. World Health Organisation / Secretariat of the UN Convention on Biological Diversity, 2015.
- [4] Valérie Masson-Delmotte, Panmao Zhai, Anna Pirani, Sarah L. Connors, Clotilde Péan, Sophie Berger, Nada Caud, Yang Chen, Leah Goldfarb, Melissa I. Gomis, Mengtian Huang, Katherine Leitzell, Elisabeth Lonnoy, J. B. Robin Matthews, Thomas K. Maycock, Tim Waterfield, Ozge Yelekçi, Rong Yu, and Baiquan Zhou. IPCC Climate Change 2021: The Physical Science Basis. Cambridge: Cambridge University Press, 2021.
- [5] UN Global Compact Office (2015) Supply chain sustainability: a practical guide for continuous improvement, 2. Aufl. United Nations Global Compact and BSR, New York
- [6] Göran Finnveden, Michael Z. Hauschild, Tomas Ekvall, Jeroen Guinée, Reinout Heijungs, Stefanie Hellweg, Annette Koehler, David Pennington, and Sangwon Suh. „Recent Developments in Life Cycle Assessment“. In: Journal of Environmental Management 91.1 (2009), pp. 1–21.
- [7] Hans de Bruijn, Robbert van Duin, Mark A. J. Huijbregts, Jeroen B. Guinee, Marieke Gorree, Reinout Heijungs, Gjalt Huppes, Renée Kleijn, Arjan de Koning, Laurant van Oers, Anneke Wegener Sleeswijk, Sangwon Suh, and Helias A. Udo de Haes, eds. Handbook on Life Cycle Assessment: Operational Guide to the ISO Standards. Vol. 7. EcoEfficiency in Industry and Science. Dordrecht: Springer Netherlands, 2002.
- [8] ISO 14040: 2006 - Environmental Management – Life Cycle Assessment – Principles and Framework. July 2006.



**Dr. Thomas Mützel-von Schwartz**  
thomas.muettel@skytender.com

Dr. Thomas Mützel-von Schwartz trat als Leiter der Entwicklung und Produktion der SkyTender AG bei und ist heute im Vorstand der SkyTender AG zuständig für die Bereiche Entwicklung, Produktion, Wartung, Materialwirtschaft und Technical Sales. Er ist Experte im Bereich Operational Excellence und hat langjährige Berufserfahrung als Führungskraft in der Automobil- und Luftfahrtindustrie.



**Florian Lösch**  
florian.loesch@promatis.de

Florian Lösch ist als Senior Principal Consultant Digitized Processes und Vice President BU NetSuite in der PROMATIS Unternehmensgruppe in verschiedenen geschäftsprozessorientierten Implementierungen Oracle-basierter Unternehmenssoftware aktiv. Er ist Experte für Geschäftsprozessmanagement und lehrt und forscht in diesem Zusammenhang die Themen Nachhaltigkeit und Geschäftsprozessmanagement



# Was ist ESG Reporting?

Chris Neubauer, BARC GmbH

*Das Thema Nachhaltigkeit ist mit der Festlegung der Sustainable Development Goals (SDGs) der Vereinten Nationen (UN) im Jahr 2015 ein zentrales Diskussionsthema in Politik, Wirtschaft und Gesellschaft geworden. Die in der zugehörigen Agenda verankerten Nachhaltigkeitsziele sind untergliedert in ökonomische, ökologische und soziale Aspekte. Auch für Unternehmen wird ESG (Environment, Social, Governance) ein immer wichtigeres Thema. Künftig müssen sich aufgrund strengerer Regularien auch viele kleinere Unternehmen in Deutschland und der EU der Herausforderung der ESG-Berichtspflicht stellen. In diesem Beitrag wird ESG, das zugehörige Reporting sowie ein ESG-Glossar vorgestellt. Dieses erläutert zentrale und regulatorische Begriffe und benennt die ESG-Standards.*

## Was ist ESG Reporting?

ESG Reporting ist als Teil der Nachhaltigkeitsberichterstattung ein Prozess, bei dem Unternehmen ihre Leistungen in Bezug auf Nachhaltigkeit und Corporate Social Responsibility (CSR) messen, bewerten und offenlegen. Die Berichterstattung umfasst

Umweltbelange, soziale Aspekte und verantwortungsvolle Unternehmensführung.

Diese Form der Berichterstattung hat in den letzten Jahren nicht nur durch regulatorische Pflichten an Bedeutung gewonnen [1]. Unternehmen achten auch aufgrund der eigenen Reputation und Außendarstellung

gegenüber Kunden, Mitarbeitenden und Investoren vermehrt darauf, wie sie ihre Geschäfte betreiben und wie diese sich auf die Gesellschaft und Umwelt auswirken.

ESG Reporting soll Transparenz schaffen und Unternehmen zur Rechenschaftspflicht bewegen, um den Stakeholdern Informati-

onen über die Nachhaltigkeit der Geschäftspraktiken zur Verfügung zu stellen. Das Ziel von ESG Reporting besteht darin, eine umfassende Bewertung der Nachhaltigkeitsleistung eines Unternehmens zu ermöglichen. Dazu werden verschiedene Faktoren berücksichtigt wie zum Beispiel die Auswirkungen des Unternehmens auf die Umwelt, die Art und Weise, wie das Unternehmen mit Mitarbeitenden und anderen Interessensgruppen umgeht sowie Governance-Aspekte [2] und Compliance-Praktiken [3].

Das ESG Reporting lässt sich, wie der Name bereits verrät, in die drei Teilbereiche Environment, Social und Governance untergliedern. Die Komplexität der Berichterstattung wird deutlich, wenn man einen Blick auf die ersten Entwürfe des European Sustainability Reporting Standards (ESRS) wirft. Hier ist von insgesamt 70 datengetriebenen und weniger-datengetriebenen Kennzahlen die Rede, die sich den verschiedenen Bereichen zuordnen lassen.

### E: Environmental

32 dieser 70 Key Performance Indikatoren (KPI) sind innerhalb des Bereichs „Environmental“ zu verordnen. Die relevanten Umweltthemen lassen sich am Beispiel des ESRS einteilen in

- Klimawandel,
- Umweltverschmutzung,
- Wasser und maritime Ressourcen,
- Biodiversität und Ökosysteme sowie
- Ressourcennutzung und Kreislaufwirtschaft.

Hierbei geht es beispielsweise um Schadstoff-Emissionen, den CO<sub>2</sub>-Fußabdruck des Unternehmens, Abfall- und Wassermanagementpraktiken sowie Bemühungen zur Förderung und Nutzung erneuerbarer Energien. Zentraler Bestandteil des Bereichs „Environment“ ist die Erfassung unternehmensinterner und -externer Daten sowie die Berechnung/Schätzung von Performance-Indikatoren mittels datengetriebener Unternehmenskennzahlen.

### S: Social

32 KPI sind gemäß ESRS im Bereich „Soziales“ anzusiedeln. Zentrale Fragestellungen sind unter anderem die Mitarbeiterzufriedenheit, Gleichberechtigung und gemeinnützige/soziale Unternehmenspraktiken. Thematisch lässt sich der Bereich nach ESRS beispielsweise untergliedern in

- eigene Arbeitskräfte,
- Arbeitskräfte in der Wertschöpfungskette,
- betroffene Gemeinschaften sowie
- Kunden und Endverbraucher.

### G: Governance

Der Bereich Governance widmet sich dem Bereich der verantwortungsvollen Geschäftspraktiken. Es geht hierbei primär um Fragen der Unternehmensführung wie zum Beispiel die Diversität des Vorstands, Korruptionsbekämpfung und die Einhaltung von Gesetzen und Vorschriften. Das Teilsegment Governance ist am wenigsten von datengetriebenen Kennzahlen geprägt, was sich auch in der Menge der hier identifizierten KPI bestätigt. Der ESRS ordnet dem Bereich lediglich sechs Kennzahlen zu.

Branchenspezifische ausgewählte Key Performance Indikatoren der Bereiche E, S und G ermöglichen in der Folge die Nach-



## Zukunftssicherheit: Ihr Partner für Digitale Intelligenz

### Nachhaltige Technologien für Unternehmen von morgen

Sie haben die Daten, wir die Vernetzung zum Erfolg – individuell für Ihre Bedürfnisse. Mit innovativen Cloud-Lösungen und Oracle-Applikationen, die eine praxisnahe Umsetzung garantieren. Sicher. Einfach. Überall.

Jetzt informieren: [www.promatis.de](http://www.promatis.de)

haltigkeitsbewertung des wirtschaftlichen Handelns von Unternehmen und die Vergleichbarkeit untereinander. Dieser Bewertungsmaßstab wird auch als ESG Performance bezeichnet.

### Markt und Lösungen

Es gibt im ESG-Markt bereits einen intensiven Wettbewerb zwischen Software-Anbietern und viele konkurrierende Lösungen. BARC sieht den dringlichsten Bedarf an Software- und Reportinglösungen im Bereich Environment, da dies der datenintensivste Bereich ist. Eine aktuelle Einordnung des Gesamtmarktes werden wir in einem der kommenden Beiträge dieser Reihe beleuchten. Im Segment „Governance“ gibt es einige bestehende Tools und Plattformen.

Abschließend lässt sich festhalten, dass die Wichtigkeit von ESG Reporting nicht nur aufgrund regulatorischer Pflichten und der angestrebten positiven Außendarstellung von Unternehmen an Bedeutung gewinnt: Als wirksames Instrument zur Überwachung und Verbesserung der unternehmerischen Nachhaltigkeitsleistung kann ESG Reporting das Risikomanagement optimieren und langfristig zum wirtschaftlichen Erfolg des Unternehmens beitragen.

### ESG Glossar

Das ESG-Glossar dient der Einführung in die ESG-Thematik, indem es zentrale Begriffe auflistet und erläutert. Das ESG-Glossar enthält allgemeine und regulatorische Begriffe sowie ESG-Standards.

### Allgemeine Begriffe

#### ESG

ESG ist die Abkürzung für „Environmental, Social and Governance“ und bezeichnet die drei Hauptbereiche der unternehmerischen Nachhaltigkeitsverantwortung und -berichterstattung. Die Bereiche evaluieren, wie ein Unternehmen ökologische und soziale Faktoren sowie Aspekte der Unternehmensführung in seine Strategie und Tätigkeiten integriert und über diese berichtet.

#### Environmental

Der Bereich „E“ (Environmental) umfasst alle Aspekte des unternehmerischen Umweltmanagements und beurteilt, inwieweit ein Unternehmen ökologische Kriterien berücksichtigt und nachhaltige Praktiken implementiert. Dies beinhaltet z. B. die Position des Unternehmens in Bezug auf den

Klimawandel, seine Ziele und Maßnahmen zur Reduktion von CO<sub>2</sub>-Emissionen.

#### Social

Der Bereich „S“ (Social) bezieht sich auf soziale und gesellschaftliche Aspekte der unternehmerischen Tätigkeit und bezieht neben den eigenen Arbeitskräften auch Kunden und Endverbraucher mit ein. Die Bewertung der sozialen Performance eines Unternehmens umfasst Aspekte wie z. B. Mitarbeitergesundheit, Einhaltung von Arbeitsrechten, Sicherheit am Arbeitsplatz und faire Entlohnung.

#### Governance

Der Bereich „G“ (Governance) bezieht sich auf die Unternehmensführung einschließlich aller Prozesse und Vorgehensweisen, die für die Leitung und Kontrolle eines Unternehmens notwendig sind. Dieser Teilbereich der ESG-Kriterien bewertet die verantwortungsvolle Unternehmensführung.

#### CSR

Corporate Social Responsibility (CSR) basiert auf dem Prinzip, dass Unternehmen positive wie auch negative Auswirkungen auf die Gesellschaft und Stakeholder haben. CSR umfasst ökologische, ökonomische und soziale Dimensionen, die sich gegenseitig bedingen und gleichermaßen berücksichtigt werden sollten. Während CSR qualitativ geprägt ist, legt ESG einen stärkeren Fokus auf quantitative Bewertungen und schließt explizit das Kriterium der Corporate Governance ein.

#### ESG Ratings

ESG Ratings werden von spezialisierten Rating-Agenturen erstellt und bewerten die ESG-Konformität und Performance eines Unternehmens. Sie dienen u. a. dazu, potenzielle ESG-Risiken und -Chancen für Unternehmen und Investoren zu identifizieren, um diese in Entscheidungsprozesse einfließen lassen zu können.

#### Doppelte Wesentlichkeit

Der Begriff steht für eine doppelte Perspektive in der Nachhaltigkeitsberichterstattung von Unternehmen. Zum einen wird die Inside-Out-Perspektive betrachtet: Hier geht es darum, wie sich die Aktivitäten des Unternehmens auf Umwelt und Gesellschaft auswirken, etwa durch die Emission von Treibhausgasen oder die Einhaltung von Arbeitsrechten. Zum anderen

wird die Outside-In-Perspektive betrachtet: Hier wird analysiert, wie Umwelt- und Gesellschaftsfaktoren das Unternehmen beeinflussen können, zum Beispiel durch klimabedingte Risiken, Gesetzesänderungen oder die öffentliche Meinung. Beide Perspektiven zusammen ergeben ein ganzheitliches Bild der Nachhaltigkeitsleistung eines Unternehmens.

### Standards, Frameworks & Richtlinien

#### CSRD

Die Corporate Sustainability Reporting Directive (CSRD) ist eine Überarbeitung und Erweiterung der Non-Financial Reporting Directive (NFRD). Durch die CSRD werden künftig deutlich mehr Unternehmen über ihre Nachhaltigkeitsleistung berichten müssen, als dies unter der NFRD der Fall war. Die CSRD konkretisiert dabei die Anforderungen an die ESG-Berichterstattung und legt Art, Inhalt und Umfang der Berichterstattung weitgehend fest. Die Anwendung erfolgt differenziert nach Unternehmensgröße, Unternehmenstyp und weiteren Faktoren. Die Nachhaltigkeitsberichte müssen sich an den European Sustainability Reporting Standards (ESRS) der European Financial Reporting Advisory Group (EFRAG) orientieren.

#### DNSH

„Do No Significant Harm“ (DNSH) ist ein Grundsatz, der im Rahmen der EU-Taxonomie für die Bewertung von Unternehmensaktivitäten beachtet werden muss. Er besagt, dass keine Unternehmensaktivität eines der sechs in der EU-Taxonomie festgelegten Umweltziele erheblich beeinträchtigen darf.

#### EFRAG

Die European Financial Reporting Advisory Group (EFRAG) unterstützt die Europäische Kommission bei der Überleitung der International Financial Reporting Standards (IFRS) in EU-Standards. Sie bietet technische Beratung und ausgearbeitete Entwürfe für EU-Standards zur Nachhaltigkeitsberichterstattung.

#### ESRS

Die European Sustainability Reporting Standards (ESRS) sind EU-Berichtsstandards, die für alle Unternehmen gelten, die von der CSRD betroffen sind. Sie umfassen sektorunabhängige und sektorspezifische



Standards und werden von der EFRAG entwickelt und veröffentlicht.

### EU-Taxonomie

Die EU-Taxonomie ist ein Klassifizierungssystem für ökologisch nachhaltige Wirtschaftsaktivitäten. Sie bietet Unternehmen, Investoren und politischen Entscheidungsträgern eine gemeinsame Definition dafür, welche Aktivitäten als ökologisch nachhaltig angesehen werden können. Dies soll unter anderem Sicherheit für Investoren schaffen, Greenwashing verhindern und Unternehmen dabei unterstützen, klimafreundlicher zu werden. Eine wirtschaftliche Aktivität gilt dann als ökologisch nachhaltig, wenn sie einen signifikanten Beitrag zu mindestens einem der sechs Umweltziele leistet und dabei sowohl dem DNSH-Prinzip entspricht als auch bestimmte Mindestanforderungen erfüllt.

### GRI

Die Global Reporting Initiative (GRI) ist eine unabhängige Organisation, die weltweit anerkannte Standards für die Berichterstattung über die wirtschaftlichen, ökologischen und sozialen Auswirkungen von Organisationen entwickelt hat.

### IASB

Das International Accounting Standards Board (IASB) ist ein unabhängiges Gremium, das für die Entwicklung und Veröffentlichung der International Financial Reporting Standards (IFRS) verantwortlich ist. Diese Standards dienen als Leitlinien für die Finanzberichterstattung und sollen eine konsistente und transparente Darstellung der finanziellen Leistung von Unternehmen weltweit sicherstellen. Das IASB ist eine Organisation der IFRS Foundation.

### IFRS

Die International Financial Reporting Standards (IFRS) sind internationale Rechnungslegungsstandards, die von der IFRS Foundation entwickelt wurden. Sie sollen leicht umsetzbar, durchsetzbar und weltweit anerkannt sein. Bei der Entwicklung der Standards sind die beiden Gremien IASB und ISSB beteiligt.

### ISSB

Das International Sustainability Standards Board (ISSB) ist ein der IFRS Foundation angehöriges Gremium, das die IFRS-Standards für die Nachhaltigkeitsberichterstattung

(IFRS Sustainability Disclosure Standards) entwickelt.

### LkSG

Das ab 2023 geltende deutsche Lieferkettensorgepflichtengesetz zielt darauf ab, Menschenrechtsverletzungen und Umweltzerstörungen entlang der Lieferkette zu minimieren. Die Europäische Kommission hat zudem bereits ein EU-Lieferkettengesetz vorgelegt, das deutlich über die Vorgaben des LkSG hinausgeht.

### NFRD

Mit der Non-Financial Reporting Directive (NFRD) wurden die Anforderungen an die jährliche Finanzberichterstattung großer Unternehmen um die Offenlegung nicht-finanzieller Informationen erweitert. Die NFRD wird in Zukunft zunehmend durch die CSRD ersetzt werden.

### SASB

Das Sustainability Accounting Standards Board (SASB) erstellt branchenspezifische Rechnungslegungsstandards. Die Standards ermöglichen es Unternehmen, finanzrelevante Nachhaltigkeitsdaten zu erfassen, zu verwalten und zu kommunizieren.

### SFDR

Die Sustainable Finance Disclosure Regulation (SFDR) zielt auf eine erhöhte Transparenz von Finanzinstituten und Marktteilnehmern in Bezug auf Nachhaltigkeit ab. Sie fördert nachhaltige Anlageoptionen und verlangt die Kennzeichnung aller in Europa vertriebenen Investmentfonds.

### TCFD

Die Task Force on Climate-related Financial Disclosures (TCFD) wurde gegründet, um die Berichterstattung über klimabezogene Finanzinformationen zu verbessern und zu erweitern. Sie gibt Empfehlungen darüber, welche Informationen Unternehmen offenlegen sollten. Die Berichtsstruktur der TCFD dient vielen anderen Standards, wie zum Beispiel den ESRS, als Orientierung.

Anmerkung der Redaktion: Dieser Beitrag von Chris Neubauer erschien zuerst als Blogpost am 8. März 2023 auf dem Portal [barc.com](https://barc.com). URL: <https://barc.com/de/was-ist-esg-reporting/>

### Quellen

[1] <https://barc.com/de/glossar/data-analytics/reporting/>

[2] <https://barc.com/de/glossar/data-analytics/data-governance-dg/>

[3] <https://barc.com/de/glossar/data-analytics/governance-risk-and-compliance-grc/>

[4] <https://barc.com/de/events/df-conline-group-accounting/dfconline-group-accounting-registrierung/>

## VERANSTALTUNGSHINWEIS

### #DFConline – Group Accounting: Konzernkonsolidierung & ESG Reporting

Am 8. und 9. November 2023 dreht sich beim BARC Online-Event alles um Group Accounting: Lernen Sie marktführende Tools kennen und erfahren Sie, wie die Konzernkonsolidierung gelingt und wie Sie die neuen Anforderungen des ESG Reporting bewältigen können [4].



**Chris Neubauer**  
[cneubauer@barc.com](mailto:cneubauer@barc.com)

Chris Neubauer ist Analyst für Data, Analytics und ESG beim Analystenhaus BARC. Chris' Fachgebiete sind Softwarelösungen im Bereich Corporate Performance Management (CPM), Planung, Analytics und ESG (Environmental, Social and Governance). Er berät Unternehmen bei der Identifikation von Einsatzszenarien sowie bei der Softwareauswahl und unterstützt bei strategischen Fragestellungen. Darüber hinaus ist er Autor von BARC-Marktstudien und Fachartikeln und begleitet Projektseminare im Bereich ESG an Hochschulen.



# *Datentransfer mit Oracle GoldenGate Microservices leichtgemacht*

Gerald Hacke, ORDIX AG

Viele kennen das Problem. Man möchte einfach die Daten zwischen zwei Systemen „kopieren“, beziehungsweise replizieren und stößt dabei auf viele Probleme. Manche sind unüberwindbar, manche nicht.

Eine einfache Lösung dafür ist: **Oracle GoldenGate Microservices**. Klingt vielversprechend.

Eins vorweg:

Mit Oracle GoldenGate Microservices wurde die Replikation nicht neu erfunden! Das Herzstück bleibt die klassische Replikation mit GoldenGate.

Aber für den Anwender wurde es vereinfacht und in eine benutzerfreundliche Umgebung verpackt.

## Ein kurzer Überblick

Mit der Abkündigung von Oracle Streams in der Datenbank Version 12c wurde Oracle GoldenGate das neue Replikationswerkzeug. In der klassischen Version erfolgte die Verwaltung von Oracle GoldenGate mit dem Software Command Interface: ggsci. Eine einfache Handhabung der Umgebung war möglich, allerdings war ein direkter Zugriff auf dem Server notwendig.

Mit Oracle GoldenGate, Version 12.3, wurde eine neue Architektur – Oracle GoldenGate Microservices – eingeführt. Auch diese Version unterstützt die Replikation von Daten für verschiedene heterogene Plattformen (Quelle: ORDIX AG).

In der ersten Version (12.3.0.1) wurden nur Oracle-Datenbanken unterstützt. Ab der Version 19c wurden immer mehr Datenbank-Systeme hinzugefügt (siehe Abbildung 1).

Oracle GoldenGate Microservices ist eine Microservice-Architektur, welche REST-fähige Dienste als Teil der neuen Umgebung bereitstellt. Diese REST-fähigen Dienste bieten die Möglichkeit an, Remote-Konfiguration, Remote-Verwaltung und Remote-Überwachung über HTML5-Webseiten und über Kommandozeilenbefehle sowie API-Calls durchzuführen.

Die Verwaltung von Oracle GoldenGate Microservices erfolgt über den Service Manager oder alternativ über den adminclient. Die einzelnen Prozesse werden von den jeweiligen Microservice-Prozessen gesteuert.

Der Service Manager dient dabei der Überwachung der Oracle GoldenGate Microservices Deployments. Dabei bildet ein Deployment jeweils eine Replikations-

DATENBANKEN
Oracle für Linux, Unix, Windows
Db2 für z/OS
PostgreSQL
MySQL
Microsoft SQL Server für Linux, Windows

Abbildung 1: Unterstützte Datenbanken

umgebung einer Pluggable Datenbank ab. Existieren mehrere Replikationsumgebungen, so werden diese in unterschiedlichen Deployments abgebildet. Jedes Deployment beinhaltet dann die Microservice-Prozesse: Admin Server, Distribution Server, Receiver Server und Metric Server (siehe Abbildung 2).

Der Admin Server bildet die Hauptkomponente des Oracle GoldenGate Microservices Deployments und dient der Verwaltung der Replikationsumgebung.

Hier können die Replikationsprozesse erstellt werden. Weiterhin ist die Verwaltung des Credential Stores möglich.

Sollen Benutzer mit unterschiedlichen Berechtigungen zum Einsatz kommen (zum Beispiel Administrator, Sicherheitsadministrator, Informationsbenutzer oder Operator), so können diese Benutzer hier angelegt werden (siehe Abbildung 3).

Der Distribution-Server übernimmt die Funktionalität des Transportes der Trails zwischen dem Quell- und dem Zielsystem. Für einen verschlüsselten Transport erfolgt die Verwaltung der Zertifikate im Admin Server.

Die vom Distribution Server bereitgestellten Trails werden auf dem Zielsystem vom Receiver-Server empfangen und lokal abgespeichert.

Der Distribution- und der Receiver-Server ersetzen den klassischen Data-Pump-Prozess. Zusätzlich bietet das Oracle GoldenGate Microservices Deployment eine eigene Monitoring-Funktion der Prozesse an: den Metric Server (siehe Abbildung 4).

## Unterstützte Datenbanken

Mit Oracle GoldenGate Microservices (Version 21c) werden die nachfolgenden Datenbanken unterstützt (siehe Tabelle 1).

In den kommenden Versionen werden die unterstützten Datenbank-Systeme erweitert.

## Verschiedene Methoden vom Monitoring

Mit der Installation von Oracle GoldenGate Microservices wird eine eigene Option zum Monitoring angeboten. Der Performance Metric Server bietet die Möglichkeit, die einzelnen Prozesse der Replikationsumgebung grafisch darzustellen. Der Zugriff auf die Überwachungsdaten kann über die grafische Oberfläche erfolgen oder mit Hilfe von Metric-API aufgerufen werden.

Als weiteres Hilfsmittel für das Monitoring der Replikationsumgebung kann Oracle Cloud Control 13c verwendet werden. Die Anbindung an die Oracle-GoldenGate-Microservices-Umgebung erfolgt über das GoldenGate Plug-In. Eine zusätzliche Installation des GoldenGate Monitor Agent ist **NICHT MEHR** notwendig. Die Anbindung erfolgt direkt an den Oracle GoldenGate Microservices Service Manager. Neben dem Monitoring der einzelnen Prozesse kann auch die Konfiguration überprüft, beziehungsweise angepasst werden. Weiterhin können die Log-Dateien der Prozesse angeschaut werden.

## Die neue Verzeichnis-Struktur

Mit Oracle GoldenGate Microservice werden die Daten der Replikation nun nicht mehr im Basisverzeichnis der GoldenGate-Installation abgelegt, sondern jeweils im Deployment-Verzeichnis. Sind mehrere Replikationen auf einem Server instal-

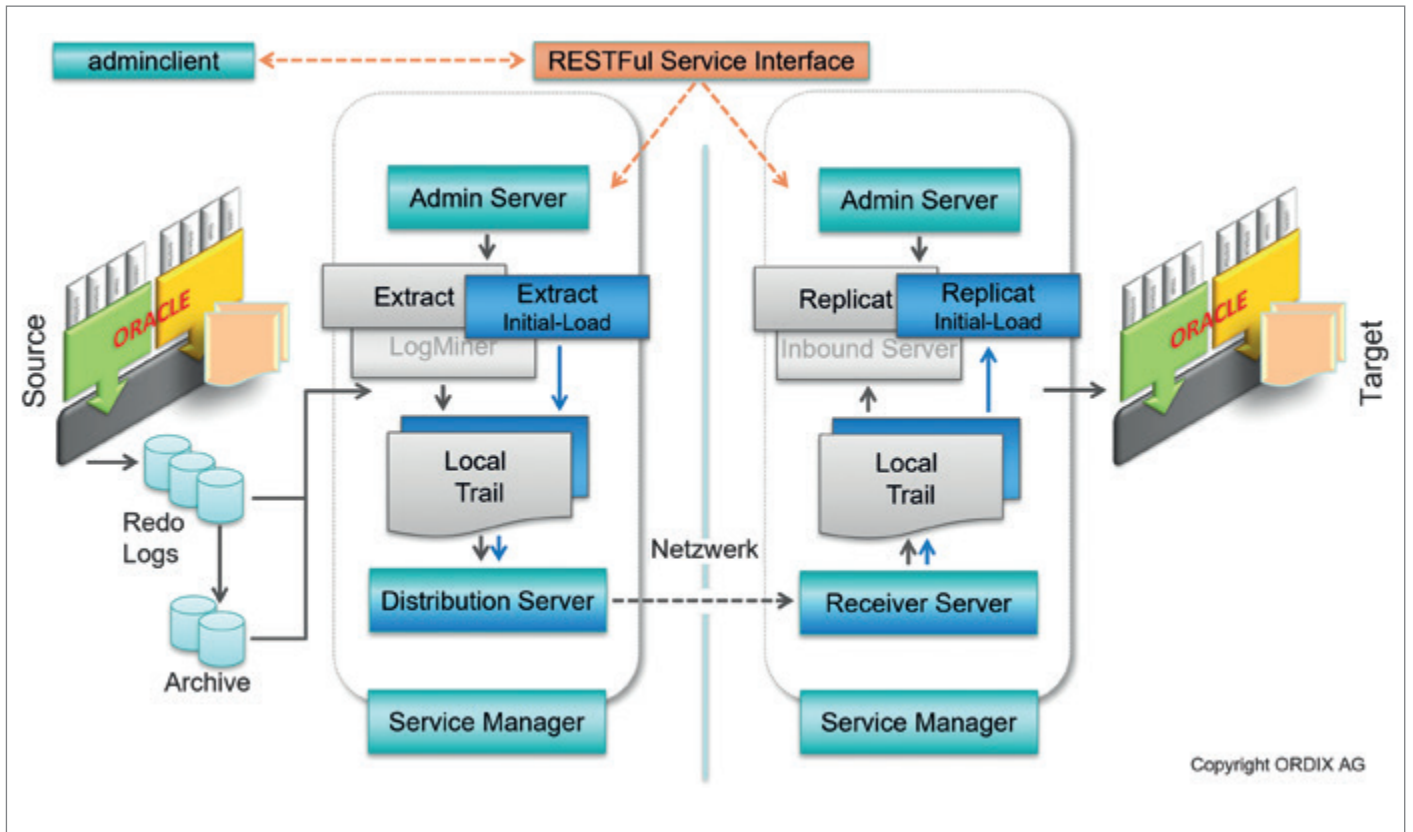


Abbildung 1: Oracle GoldenGate Microservices Architektur (Quelle: ORDIX AG)

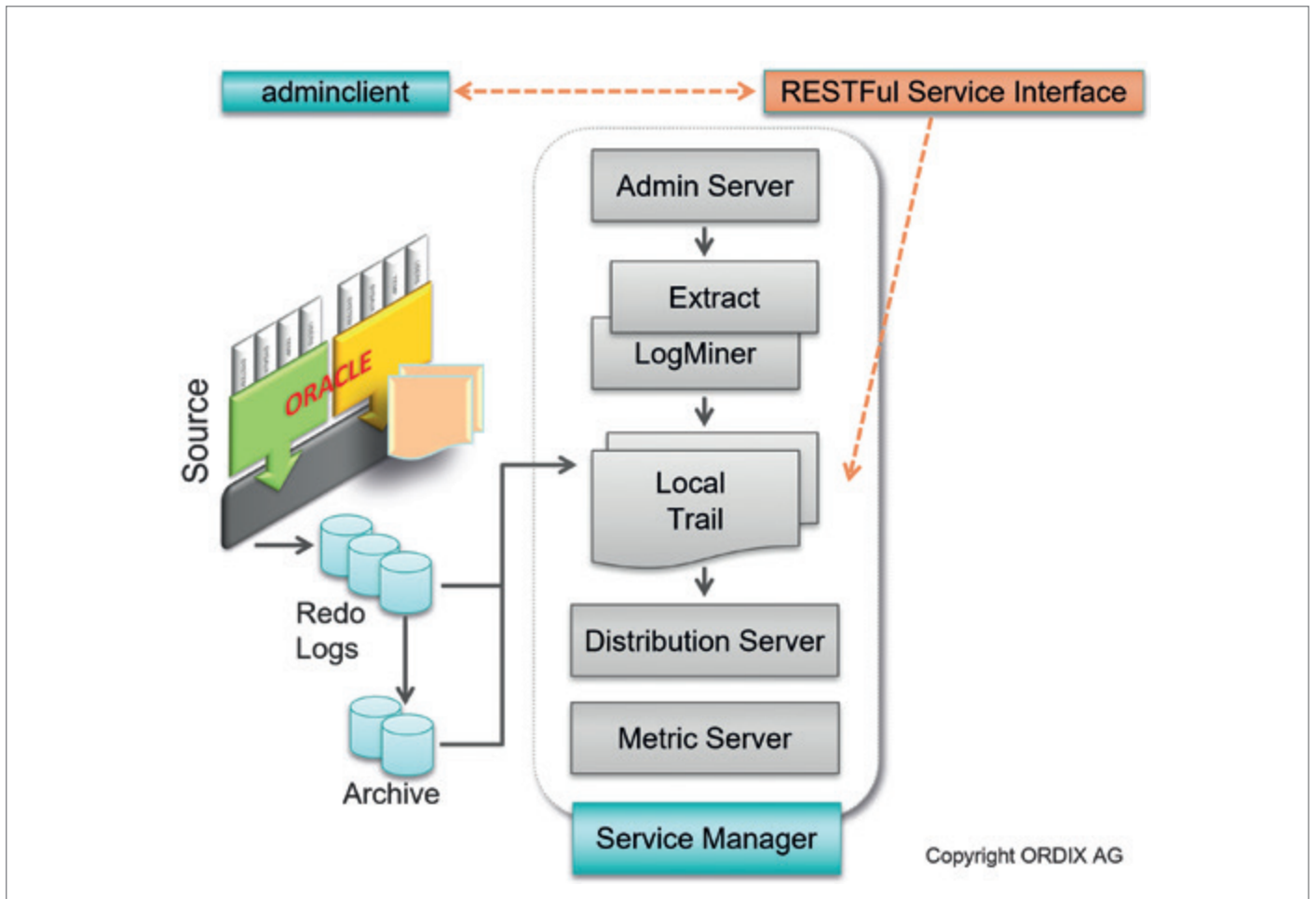


Abbildung 2: Die Verwaltungstools (Quelle: ORDIX AG)

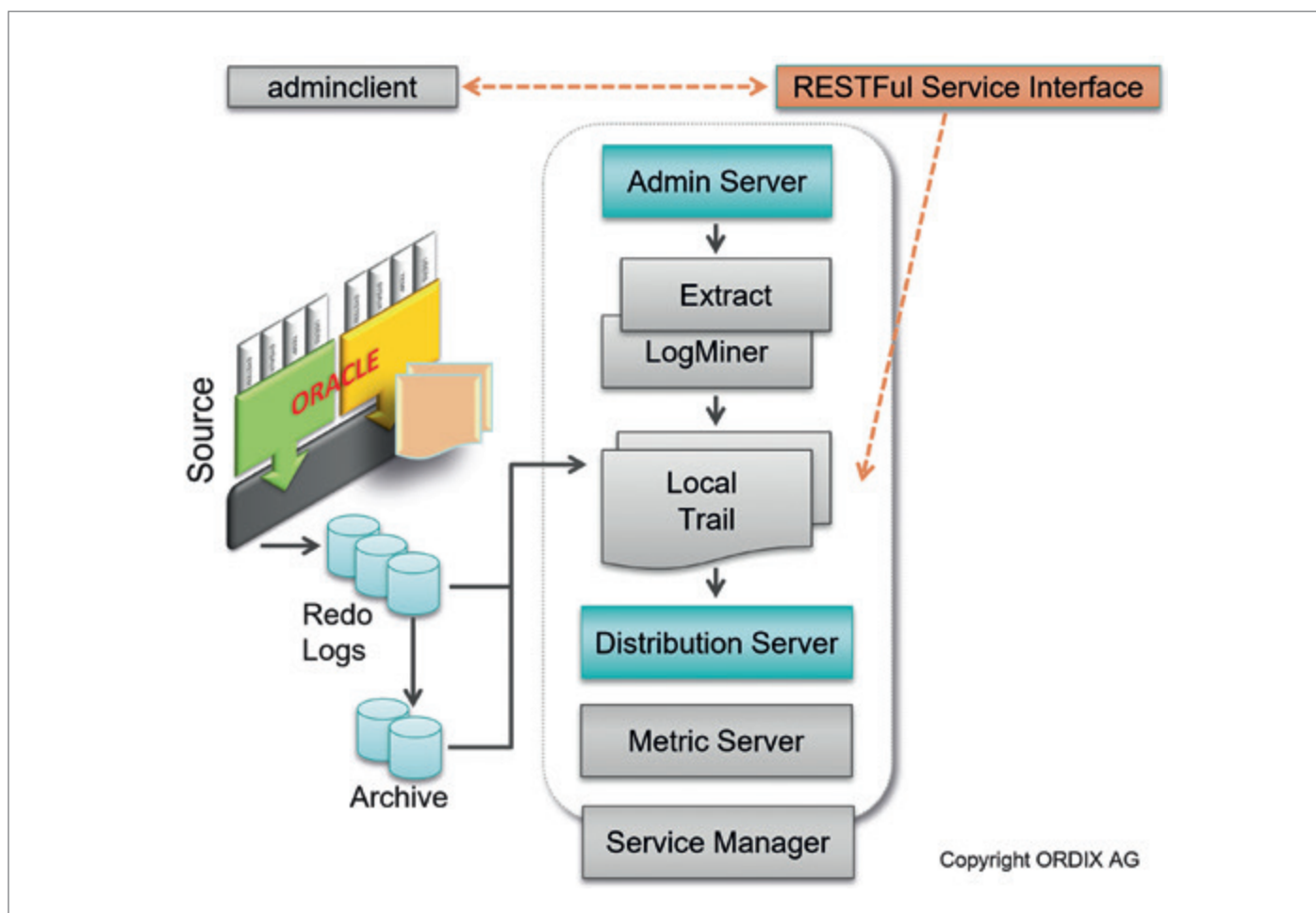


Abbildung 3: Hauptkomponenten für den Extract (Quelle: ORDIX AG)

liert, so werden die Daten im jeweiligen DEPLOYMENT\_HOME-Verzeichnis abgelegt.

- Oracle GoldenGate Microservices
  - Basisverzeichnis /oracle/gg19
- Oracle GoldenGate Microservices Service Manager (/oradata/ggSMgr)
  - Service Manager Konfiguration /oradata/ggSMgr/etc/conf
  - Service Manager SSL Wallet Location /oradata/ggSMgr/etc/ssl
  - Service Manager Daten-Verzeichnis /oradata/ggSMgr/var
- Oracle GoldenGate Microservices Deployment (/oradata/DEP1)
  - Deployment Konfiguration /oradata/DEP1/var/lib/conf
  - Deployment SSL Wallet Location /oradata/DEP1/etc/ssl
  - Deployment Parameter Dateien /oradata/DEP1/etc/conf/ogg
  - Deployment Log Dateien /oradata/DEP1/var/log/
  - Deployment Trail Dateien /oradata/DEP1/var/lib/data

- Deployment Report Dateien /oradata/DEP1/var/lib/report
- Deployment Credential Store /oradata/DEP1/var/lib/credential
- Deployment Checkpoint Dateien /oradata/DEP1/var/lib/checkpt
- Deployment Metric Datenbank /oradata/DEP1/var/lib/db

### Der einfache Weg der Daten-Replikation

Um die Daten-Replikation aufzusetzen, erfolgt die Anmeldung am GoldenGate Administrator-Server (<https://<SERVERNAME>:7010>). Über die grafische Oberfläche werden die Replikationsprozesse in einfachen Schritten angelegt (siehe Abbildung 5).

Dabei handelt es sich bei den EXTRACTS um den Extract- und den Initial-Load- Extract-Prozess. Für jeden Prozess dieser Art besteht die Möglichkeit des Startens, des Stoppens, der Modifikation und der Abfrage von Informationen (De-

tails). Bei Bedarf kann jeder Prozess auch gelöscht werden.

Soll die Parameter-Datei eines Prozesses angepasst werden, so erfolgt das über die Details eines jeden Prozesses. Anpassungen der Parameter-Datei erfordern immer einen Neustart (siehe Abbildung 5).

Sind die Extract-Prozesse erstellt, werden im nächsten Schritt die Prozesse zum Transport der Trails angelegt. Dies geschieht über den Distribution-Server (<https://<SERVERNAME>:7011>). Ein klassischer Data-Pump-Prozess ist im Oracle-GoldenGate-Microservices-Umfeld ein Distribution Path (siehe Abbildung 6).

Für jeden Extract-Prozess ist ein Distribution Path anzulegen. Im Standard erfolgt der Transport mit Verschlüsselung.

Wird ein Distribution Path nach dem Anlegen gestartet, so wird auch auf dem Zielsystem (Receiver-Server) dieser Path angelegt und gestartet.

Für den Distribution Path besteht ebenso die Möglichkeit des Startens, des Stoppens, der Modifikation und der Ab-

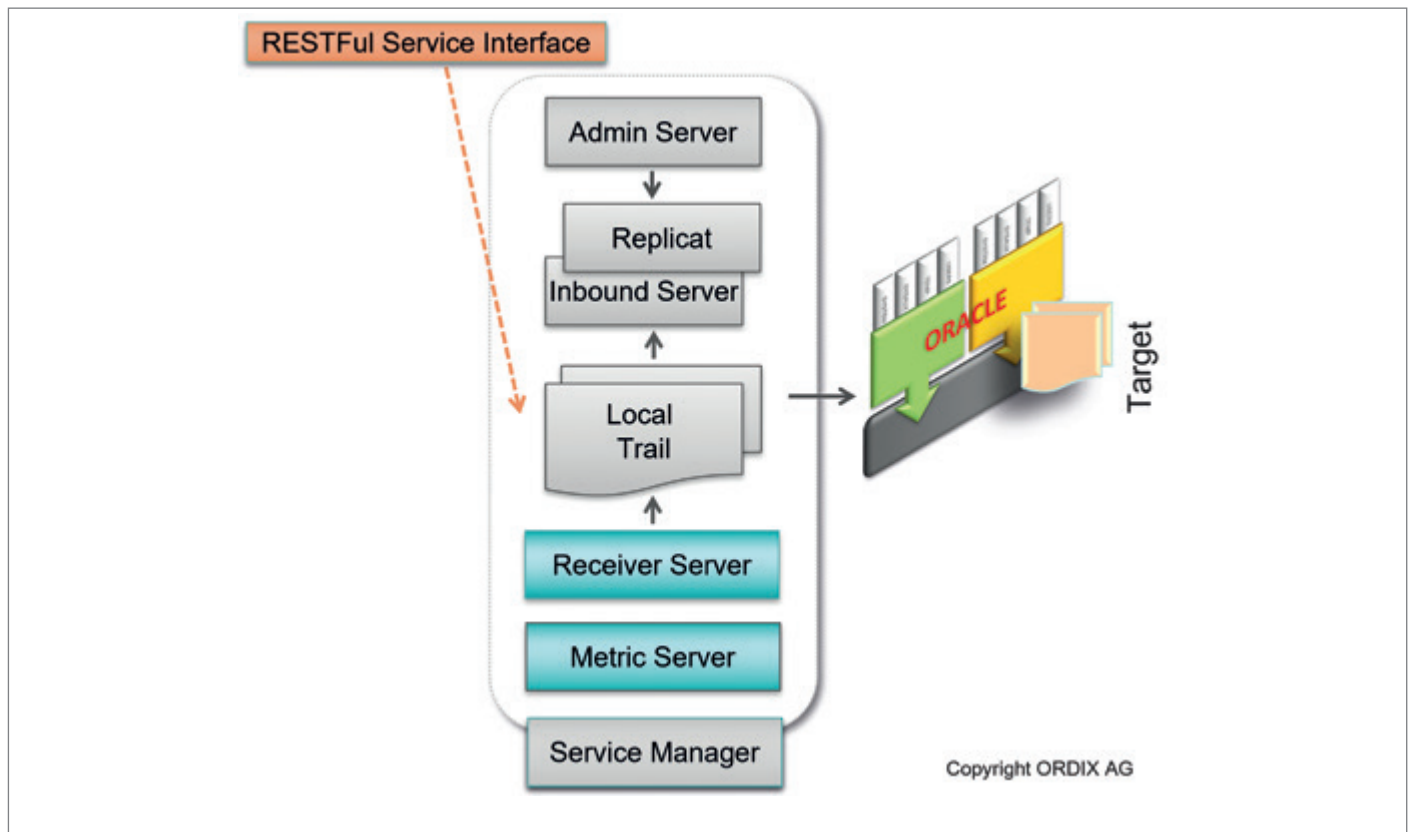


Abbildung 4: Das Zielsystem und das Monitoring (Quelle: ORDIX AG)

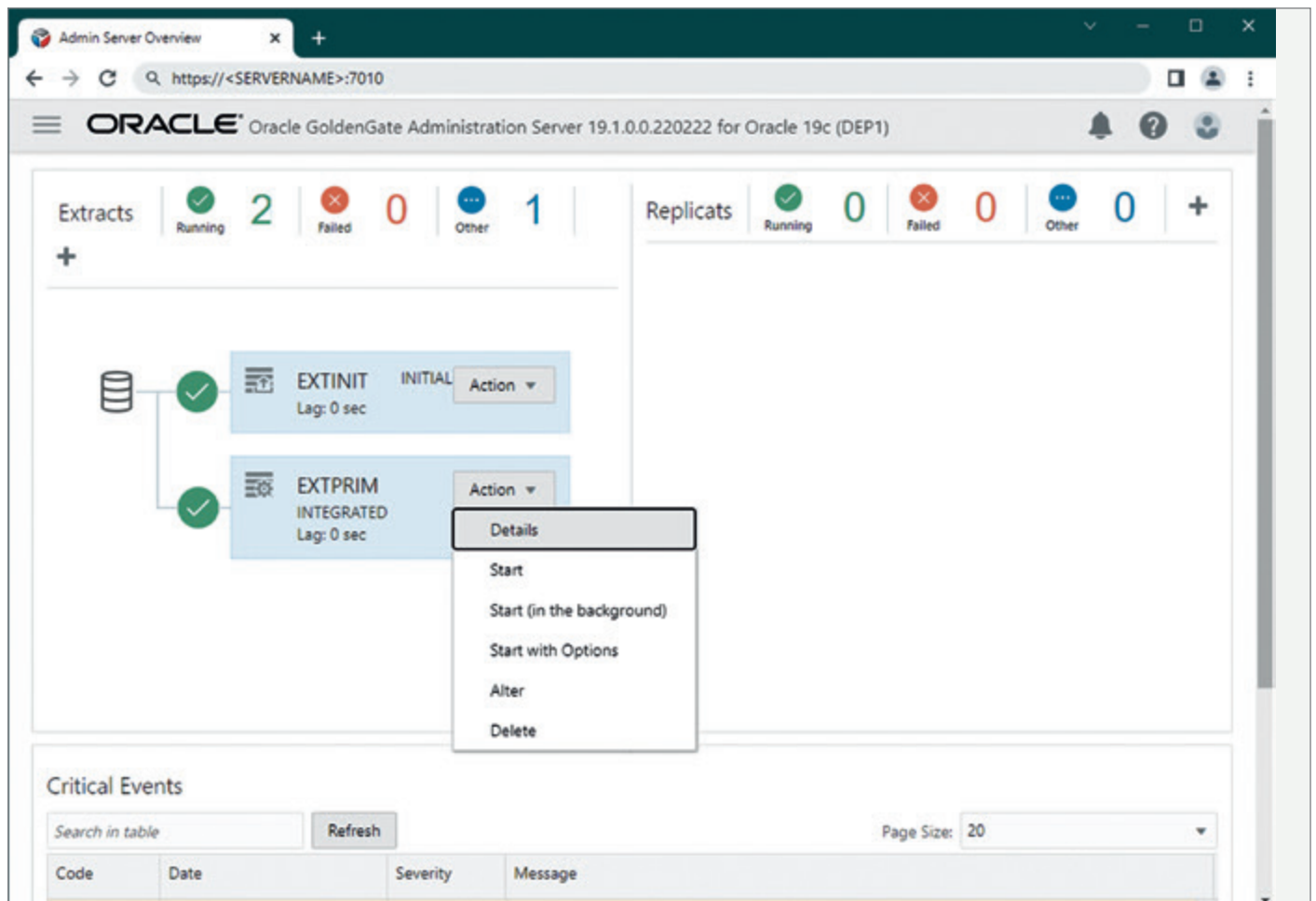


Abbildung 5: Extract-Prozesse im Oracle GoldenGate Microservices-Umfeld (Quelle: ORDIX AG)

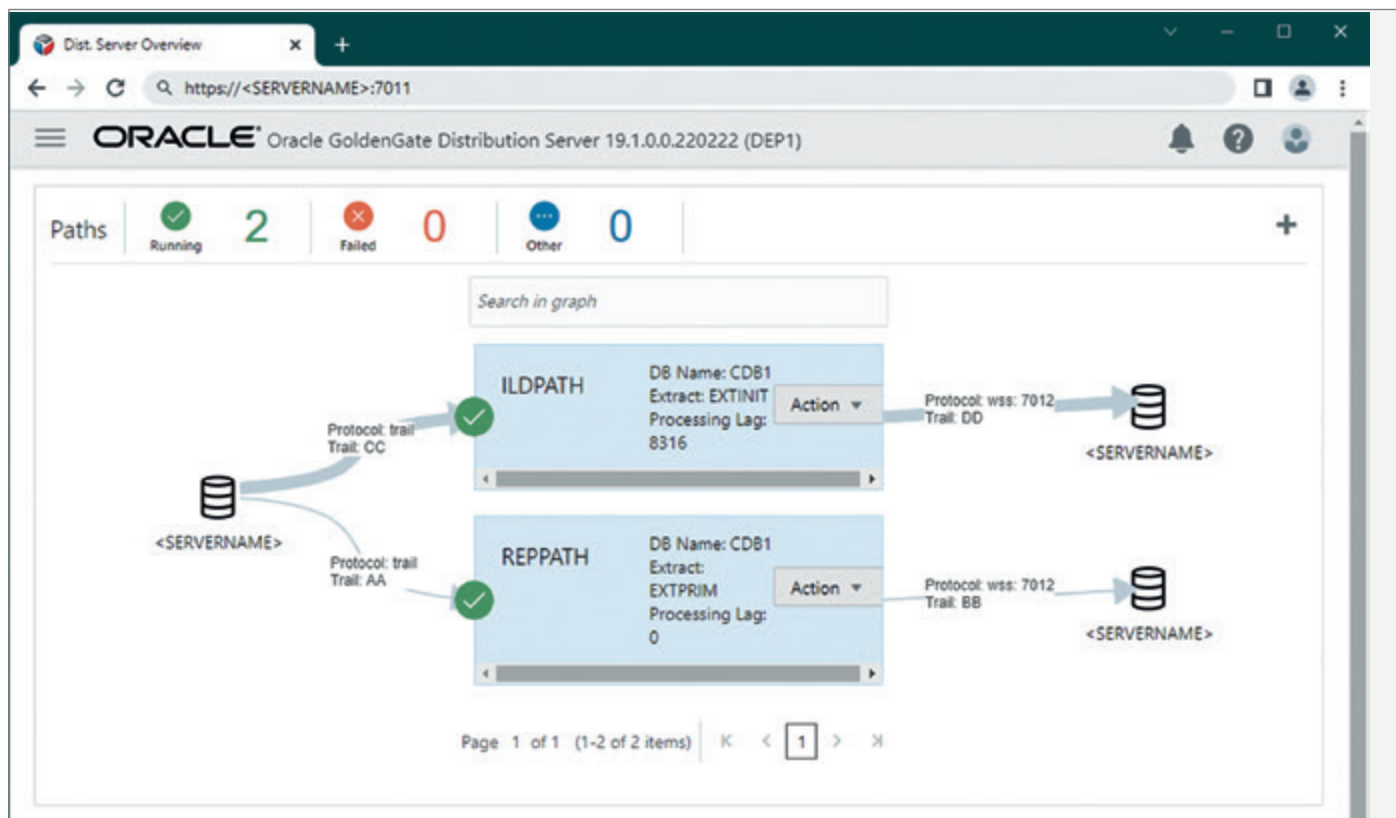


Abbildung 6: Der Distribution Path als Transporter der Trails (Quelle: ORDIX AG)

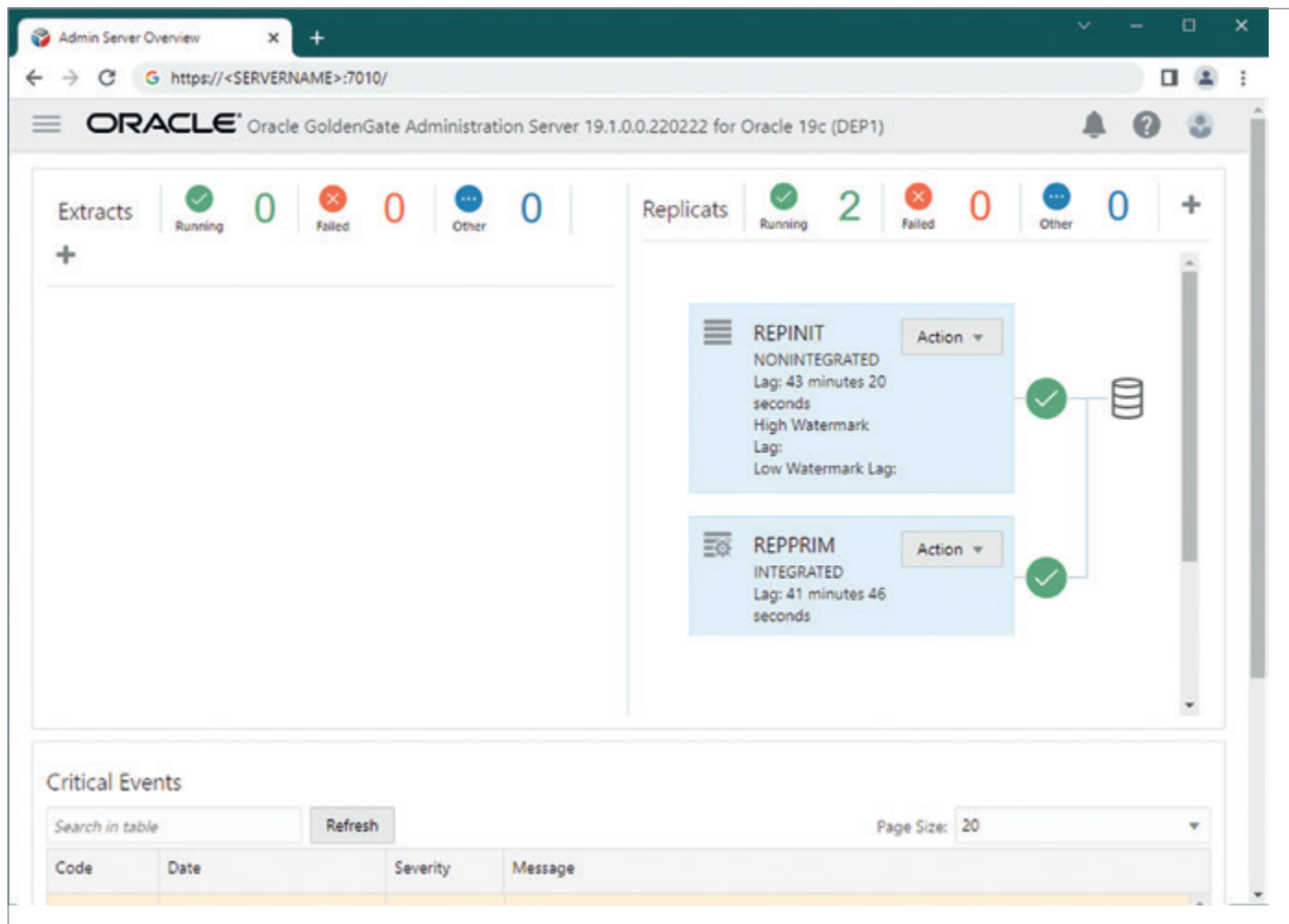


Abbildung 7: Replicat-Prozesse auf den Zielsystem (Quelle: ORDIX AG)

```

curl -s -k \
-H "Content-Type: application/json" \
-H "Authorization: Basic <Base64EncodedCredentials>" \
-H "Accept: application/json" \
-X PATCH \
https://<SERVERNAME>:7000/services/v2/deployments/ServiceManager \
-d '{
  "oggHome":"/oracle/gg19p1",
  "status":"restart"
}' | python -mjson.tool

curl -s -k \
-H "Content-Type: application/json" \
-H "Authorization: Basic <Base64EncodedCredentials>" \
-H "Accept: application/json" \
-X PATCH https://<SERVERNAME>:7000/services/v2/deployments/DEP1 \
-d '{
  "oggHome":"/oracle/gg19p1",
  "status":"restart"
}' | python -mjson.tool

```

Listing 1: Patchen durch zwei cURL-Aufrufe

frage von Informationen (Details). Zusätzlich können Filterungen in mehreren Stufen mit dem Distribution Path abgebildet werden.

Zum Abschluss werden die Replicat-Prozesse für den Datentransfer aufgesetzt. Dies erfolgt über den GoldenGate-Administrator-Server des Zielsystems (<https://<SERVERNAME>:7010>) (siehe Abbildung 7).

Bei den REPLICATS handelt es sich um den Replicat- und den Initial-Load-Replicat-Prozess. Auch für diese Prozesse besteht die Möglichkeit des Startens, des Stoppens, der Modifikation und der Abfrage von Informationen (Details).

## Die Benutzerverwaltung

War es im klassischen GoldenGate-Umfeld notwendig, dass man sich mit dem Eigentümer der Oracle-Software auf dem Server anmeldet, so bietet Oracle GoldenGate Microservices die **neue Möglichkeit**, verschiedene Benutzer mit unterschiedlichen Rollen anzulegen. Somit kann zum Beispiel im DataCenter der Zugriff auf das GoldenGate-System gesteuert werden. Eine Verwaltung beispielsweise durch die Oracle-DBA-Abteilung ist nicht mehr zwingend notwendig.

Während der Installation wird ein admin-Benutzer angelegt.

Weitere Benutzer können über den GoldenGate-Administrator-Server ange-

legt werden. Folgende Rollen können zugewiesen werden:

**Administrator** Gewährt dem Benutzer vollen Zugriff, einschließlich der Möglichkeit, allgemeine, nicht sicherheitsrelevante Betriebsparameter und Profile des Dienstes zu ändern.

**Security** Gewährt dem Benutzer die Verwaltung von sicherheitsbezogenen Objekten und ruft sicherheitsbezogene Dienstanforderungen auf. Diese Rolle hat volle Berechtigungen.

**User** Erlaubt reine Informationsabfrage.

**Operator** Ermöglicht Benutzern nur operative Aktionen wie das Erstellen, Starten und Stoppen von Ressourcen.

## Das einfache Patchen der Oracle-GoldenGate-Microservices-Umgebung

Aufwändige Verfahren wie bei der klassischen GoldenGate-Umgebung, um auf eine neue Version zu patchen, entfallen!

Durch die Trennung der Oracle-GoldenGate-Microservices-Basisinstallation, welche alle Programmdateien beinhaltet, und der einzelnen Oracle GoldenGate Microservices Deployments ist es möglich, mit wenigen Handgriffen ein Update durchzuführen.

Nach der Installation einer neuen GoldenGate-Microservices-Basisinstallation kann über den GoldenGate Service Manager (grafische Oberfläche) das GoldenGa-

te-Home vom *Service Manager* und von jedem *Deployment* angepasst werden. Nach dem Neustart der jeweiligen Services ist das Patchen abgeschlossen.

Auch ohne grafische Oberfläche kann man das Patchen durchführen – durch zwei cURL-Aufrufe (siehe Listing 1).

That's it!

## Wer die grafische Oberfläche nicht nutzen will, ...

... kann gerne das adminclient-Tool benutzen, wie er es mit dem ggsci-Tool gewohnt war.

Dabei ist zu beachten, dass neben der Anmeldung an der Datenbank auch eine Anmeldung am Service Manager notwendig ist.

```

GG DEP1 1> CONNECT <SERVICEMANAGER> DEPLOYMENT DEP1 AS admin
GG DEP1 2> DBLOGIN USERIDALIAS gg-admin DOMAIN OracleGoldenGate

```

Ein weiterer Vorteil ist, dass man sich von jedem adminclient-Tool auf jedes beliebige Oracle GoldenGate Microservices Deployment anmelden kann.

Wer es noch einfacher haben möchte, kann cURL-Aufrufe verwenden.

Oracle GoldenGate Microservices nutzt das RESTful Interface für die interne



```

curl -s \
-H "Content-Type: application/json" \
-H "Authorization: Basic <Base64EncodedCredentials>" \
-H "Accept: application/json" \
-X <METHOD> https://<SERVERNAME>:<IP PORT>/services/v2/<RESOURCE> \
...

```

Listing 2: cURL-Aufruf

```

export CURL_CA_BUNDLE=/oradata/DEP1/etc/ssl/server/<SERVER>_Cert.pem
curl -s \
-H "Content-Type: application/json" \
-H "Authorization: Basic <Base64EncodedCredentials>" \
-H "Accept: application/json" \
-X POST https://<SERVER>:7010/services/v2/connections/
      OracleGoldenGate.ggadmin/trandata/table \
-d '{
  "operation": "add",
  "tableName": "pdb1.gguser.abteilung"
}'
| python -mjson.tool

```

Listing 3: cURL-Befehl zur Durchführung von ADD TRANDATA (Änderungsprotokollierung) für eine Tabelle

Kommunikation zwischen den einzelnen Komponenten. Auch das adminclient-Tool nutzt diese Schnittstelle. Mit Hilfe des RESTful Interfaces kann auf einfache Weise eine Replikationsumgebung verwaltet und konfiguriert werden. Es werden dazu cURL-Aufrufe verwendet. Die verwendete cURL-Version muss SSL-Zertifikate unterstützen.

Der cURL-Aufruf ist bei allen Aufrufen wie folgt aufgebaut (siehe Listing 2).

Folgende **<METHOD>** stehen zur Verfügung: GET, HEAD, POST, PUT, PATCH, OPTIONS und DELETE.

**<SERVERNAME>** und **<IP PORT>** entsprechen der Oracle-GoldenGate-Microservices-Deployment-Umgebung.

**<RESOURCE>** ist entsprechend der gewünschten Operation anzupassen (zum Beispiel CREATE EXTRACT).

Die Anmeldeinformationen werden im Base64-Format verschlüsselt und dem cURL-Aufruf übergeben (**<Base64EncodedCredentials>**).

Um etwa ein *ADD TRANDATA* (Änderungsprotokollierung) für eine Tabelle durchzuführen, kann dies mit einem cURL-Befehl erfolgen (siehe Listing 3).

Weitere Informationen über die Verwendung der einzelnen RESTful-Interface-Nutzungen können über die Dokumentation abgerufen werden:

<https://docs.oracle.com/en/middleware/goldengate/core/21.3/oggra/>

## Fazit

Mit Oracle GoldenGate Microservices wird dem Datenbankadministrator ein einfach zu bedienendes Werkzeug für den Datentransfer zwischen Systemen bereitgestellt.

Trotz der hohen Lizenzkosten ist der Einsatz lohnend. Auch der Datentransfer in Cloud-Systeme wird unterstützt.

Durch die Microservice-Architektur und die Nutzung des RESTful Interfaces kann man auf sehr einfache Weise die Replikation aufbauen und verwalten. Mit der Benutzerverwaltung ist ein abteilungsübergreifender Zugriff, zum Beispiel im DataCenter, auf die Replikationsumgebung möglich.

Und wer vom klassischen GoldenGate auf die neue Architektur umsteigen möchte, kann seine Konfiguration mit Hilfe des Oracle-GoldenGate-Microservices-Migrationstools mitnehmen. Weitere Informationen zu diesem Thema gibt es in meinem Vortrag auf der DOAG 2023 Konferenz und Ausstellung.

Wer sich weiter mit Oracle GoldenGate Microservices beschäftigen möchte, kann

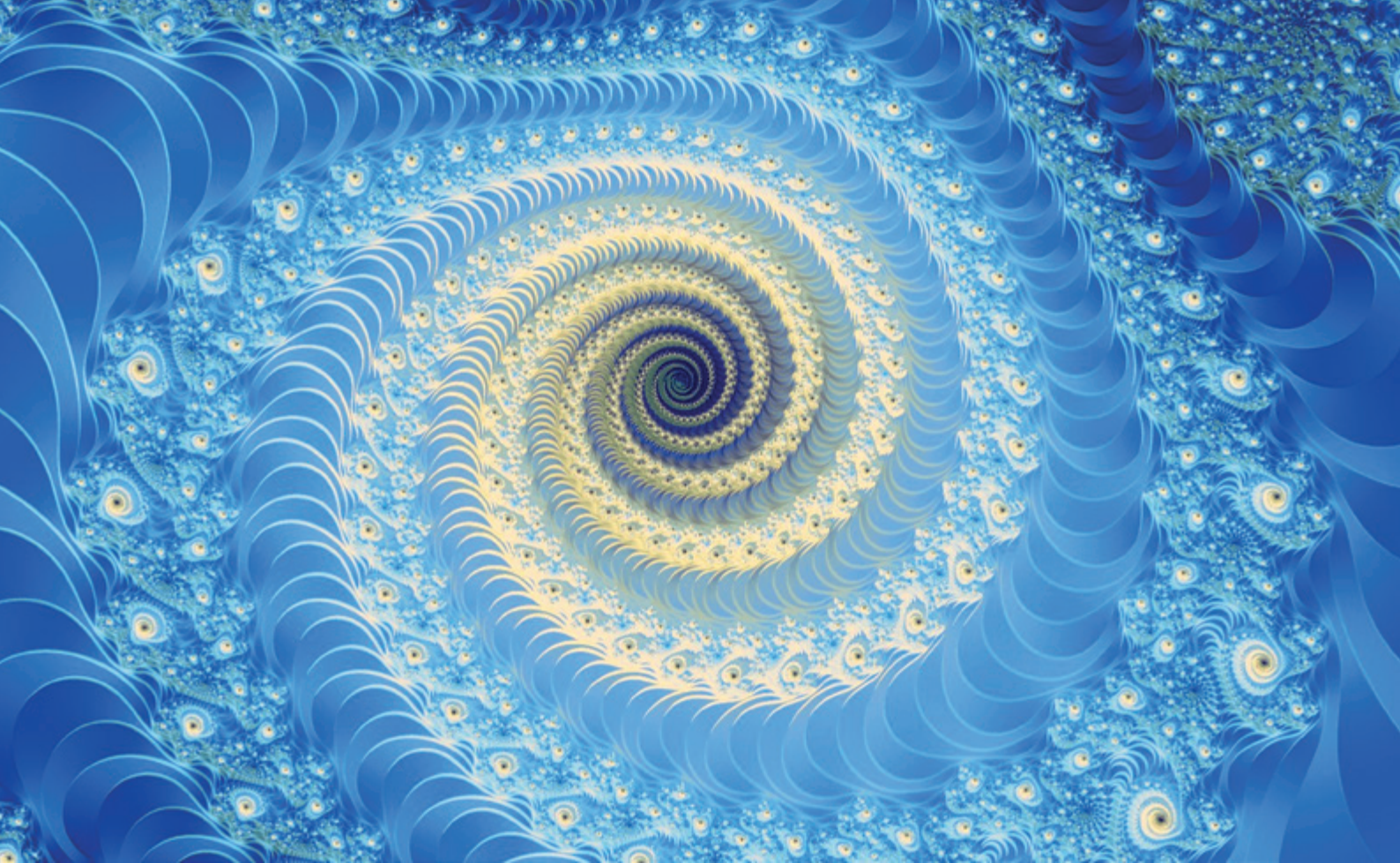
gerne unsere ORDIX-Seminar-Webseite (<https://seminare.ordix.de/>) besuchen.

## Über den Autor

Gerald Hacke ist Principal Consultant bei der ORDIX AG. Sein Tätigkeitsschwerpunkt liegt im Bereich Datenbanken und Datenreplikation. 1996 hat er begonnen, das breite Spektrum der Oracle-Datenbanken und der Datenreplikation zu erkunden. Neben seiner Beratertätigkeit gibt er sein Wissen durch Schulungen weiter.



Gerald Hacke  
gha@ordix.de



# Eine Einführung in Container Management, Kubernetes und Stateful Applications

Martin Bach, Oracle ST & Database Development

Seit vielen Jahren schon vollzieht sich ein Paradigmenwechsel im Erstellen von Applikationen. Unternehmen- und zwar meistens solche, die Web-basierte Dienste anbieten – sehen sich einem hohen Konkurrenzdruck ausgesetzt. Um am Markt erfolgreich agieren zu können, ist es daher notwendig, schnell neue Features zu integrieren und live zu stellen.

Eine Möglichkeit eine höhere Release-Kadenz zu erreichen, ist das Entkoppeln von Anwendungen in Dienste. In einer so genannten Microservices-Architektur wird eine ehemals monolithische Anwendung, meistens in genau einer Programmiersprache geschrieben, in verschiedene kleine Häppchen zerteilt. Diese so ge-

nannten Microservices interagieren oftmals über Events miteinander (Stichwort Event-Based Micro-Architecture).

Die *Abbildung 1* veranschaulicht dieses Konzept. In der rechten Hälfte des Schaubildes sieht man die monolithische Architektur: Alle zum Betrieb der Anwendung notwendigen Komponenten finden sich

unter einem Dach. Je nach Größe der Anwendung kann dies zu den folgenden potenziellen Problemen führen:

- Schwierige Release-Koordination erfordert, die Anzahl an Releases auf wenige, große Software-Releases zu beschränken

- Höherer Abstimmungsaufwand zwischen den verschiedenen Schichten (User Interface, Geschäftslogik, Datenspeicherung)
- Oftmals eine Beschränkung auf eine gemeinsam zu verwendende Programmiersprache
- Häufig kann lediglich der kleinste gemeinsame Nenner an externen Bibliotheken verwendet werden

In den oben genannten Fällen kann es sinnvoll sein, die Anwendung in kleinere Microservices zu unterteilen. An dieser Stelle sei darauf hingewiesen, dass die höhere Flexibilität zu Lasten der Komplexität geht. Verteilte Anwendungen zu erstellen und zu betreiben, erfordert ein hohes Maß an Know-how. Außerdem ist nicht jede monolithische Anwendung automatisch ein Kandidat, um in Microservices zerlegt zu werden. Dies kommt immer auf den Einzelfall an: In der jüngsten Vergangenheit hat sich herausgestellt, dass Microservices nicht die Lösung für alle Probleme darstellen!

## Microservices-Architektur

Die linke Hälfte in *Abbildung 1* zeigt exemplarisch, wie eine verteilte Anwendung aussehen kann. In der reinen Lehre be-

steht die Anwendung aus verschiedenen, kleinen Microservices, die unabhängig voneinander entwickelt werden. Jeder der abgebildeten Microservices ist ein Bestandteil des User Interface. Typischerweise besitzen Microservices ihre eigene Datenbank, das muss aber nicht zwangsläufig so sein. Die Oracle-Datenbank stellt über die Multi-Tenant-Option eine Möglichkeit bereits die Datenhaltung dennoch zentral zu verwalten: Microservices können ihre eigene Pluggable Database (PDB) nutzen, was dem Modell entgegenkommt und gleichzeitig vom Converged-Database-Ansatz profitieren.

Microservices können über Nachrichten (Events) miteinander kommunizieren. So könnte zum Beispiel ein Checkout-Micro-Service in einem Webshop einen weiteren Microservice zur Fraud Detection oder Bonitätsprüfung aufrufen. Nach erfolgreicher Prüfung des Kunden kann der Checkout-Service die Bestellung freigeben.

Laut einer Definition der Wikipedia besitzen Microservices die folgenden Attribute

- Sie sind dem Umfang nach klein
- Sie kommunizieren untereinander mittels Nachrichten
- Ihre Entwicklung erfolgt in unterschiedlichen Teams
- Sie sind Teil einer dezentralen Architektur

- Entwicklung und Deployment erfolgen hochgradig automatisiert
- In der Summe sind Microservices polyglot, das heißt sie werden in unterschiedlichen Programmiersprachen entwickelt

Microservices sind daher komplexer als monolithische Systeme, der klassische Ansatz wonach eine Anwendung in einer virtuellen Maschine (VM) betrieben wird, stößt hier an seine Grenzen. Es ist so gut wie unmöglich, mehrere Microservices unfallfrei auf einer VM zu betreiben. Der Ansatz eine VM pro Microservice zu verwenden, ist auf Dauer wahrscheinlich nicht kosteneffizient. Eine andere Art des Deployments musste gefunden werden.

## Einführung in Container-Technologie

Container bieten derzeit die bevorzugte Art einen Microservice zu betreiben.

Sehr vereinfacht gesprochen, offerieren Container eine Abstraktionsschicht zwischen Betriebssystem und der Anwendung. Es gibt keinen Hypervisor wie zum Beispiel bei KVM und es ist nicht notwendig, ein weiteres komplettes Betriebssystem zu installieren und vor allem zu warten. Ein Container Image – in diesem Fall

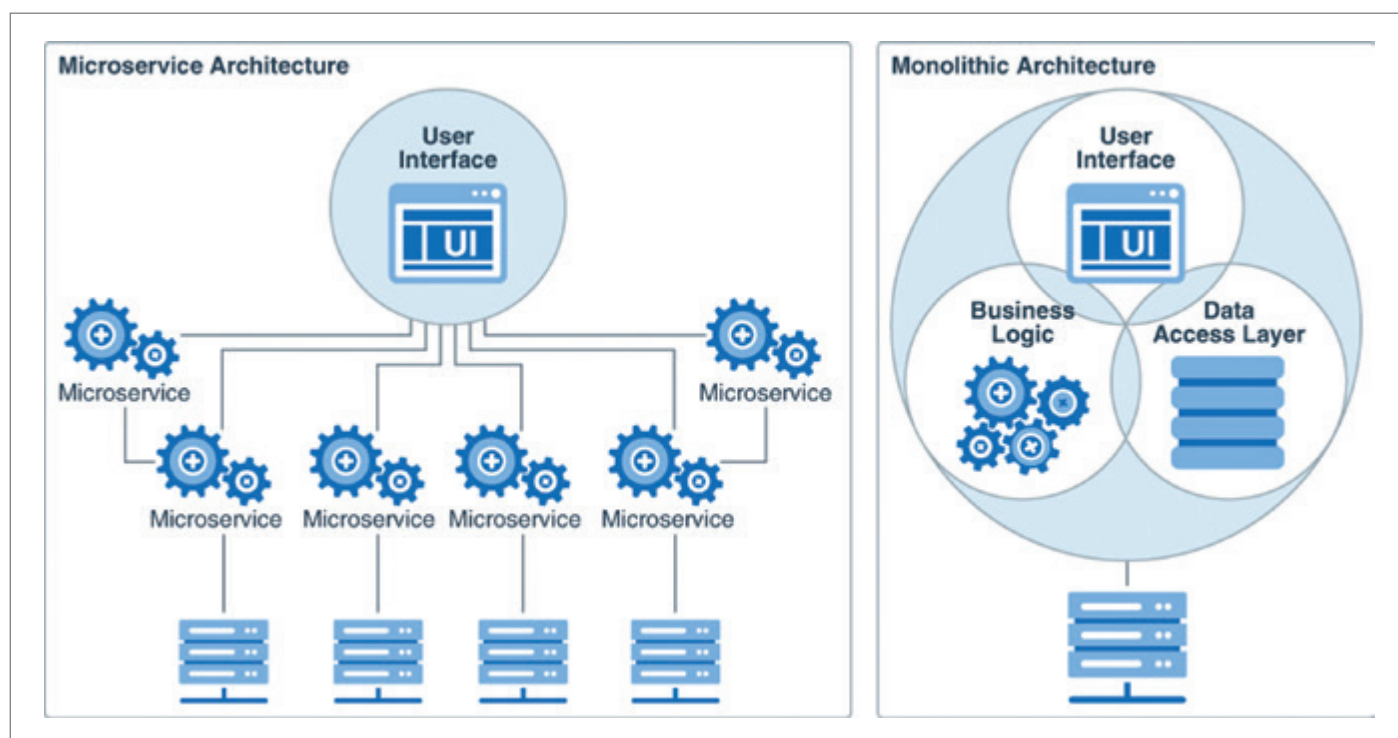


Abbildung 1: Konzept der verteilten Anwendung von Microservices und von interagierenden Microservices (Quelle: Oracle)

der Microservice – wird durch die Container Engine gestartet und ist danach erreichbar. Container Images können entweder so wie sie sind von öffentlichen Container Registries wie Oracles Container Registry bezogen und verwendet werden, oder sie dienen als Grundlage für eigene Applikationen (siehe Abbildung 2).

Docker hat das Container-Format standardisiert und mittlerweile gibt es neben der Docker Engine auch weitere Container Engines wie etwa Podman. Im Weiteren wird davon ausgegangen, dass die Container Engine auf Linux betrieben wird.

Oftmals werden Container Images als Output einer Continuous Integration/Continuous Delivery (CI/CD)-Pipeline erstellt und in einer Registry hinterlegt. Die Registry ist ein zentraler Speicherort für Container Images. Je nach Bestimmung kann sie öffentlich oder privat sein.

Container Images werden typischerweise mithilfe einer Datei namens Dockerfile erstellt, die bestimmt,

- auf welcher Linux-Distribution das Image basiert (Oracle, Arch, Debian, etc.).
- welche Software/Bibliotheken zusätzlich installiert werden.

- wie die Software kompiliert wird (Maven, Gradle, Go, Make, etc.).
- auf welchen Netzwerk-Ports die Anwendung erreichbar ist.
- was geschieht, wenn der Container gestartet wird.

Das folgende Beispiel in Listing 1 zeigt ein Dockerfile für eine kleine in Python geschriebene Anwendung.

Mit Hilfe des Dockerfile kann das Container Image erstellt werden. In diesem einfachen Fall kommt Podman zum Einsatz, es gibt mit Buildah und Docker weitere Möglichkeiten.

```
$ podman build -t flaskapp:1.0
```

Nach Eingabe des obenstehenden Kommandos wird das Container Image erstellt und lokal unter dem Tag „flaskapp:1.0“ gespeichert. Nachdem der lokale Test des Images erfolgreich war, kann es mittels einer CI/CD-Pipeline erstellt und in die Registry geschrieben werden.

Container sind per Definition unveränderlich (immutable). Notwendige Änderungen an der Applikation werden durch neue Versionen des Container Image durchgeführt. Existierende Container werden gestoppt und durch die neue

Version ersetzt. Damit unterscheiden sich Container fundamental von VMs oder Bare-Metal-Systemen, die von ihren Administratoren manchmal jahrelang liebevoll gepflegt werden („Cattle and Pet“-Ansatz).

## Orchestrierung mehrerer Container

Sobald mehrere Container über viele Server hinweg betrieben werden müssen, wird die Situation kompliziert(er). Betriebssicherheit ist durch manuellen Betrieb nicht störungsfrei zu gewährleisten. Eine Container Orchestration Engine, von denen Kubernetes sicherlich die am weitesten verbreitete ist, erfüllt diese Aufgabe.

Die Hauptaufgaben der Orchestration Engine sind

- Sicherstellen der Verfügbarkeit der gestarteten Container,
- Starten neuer Container, Neustart abgestürzter Container,
- Bereitstellung eines Systems zur Namensauflösung,
- Container Ports zu öffnen und Netzwerk Traffic zwischen Containern zu ermöglichen,

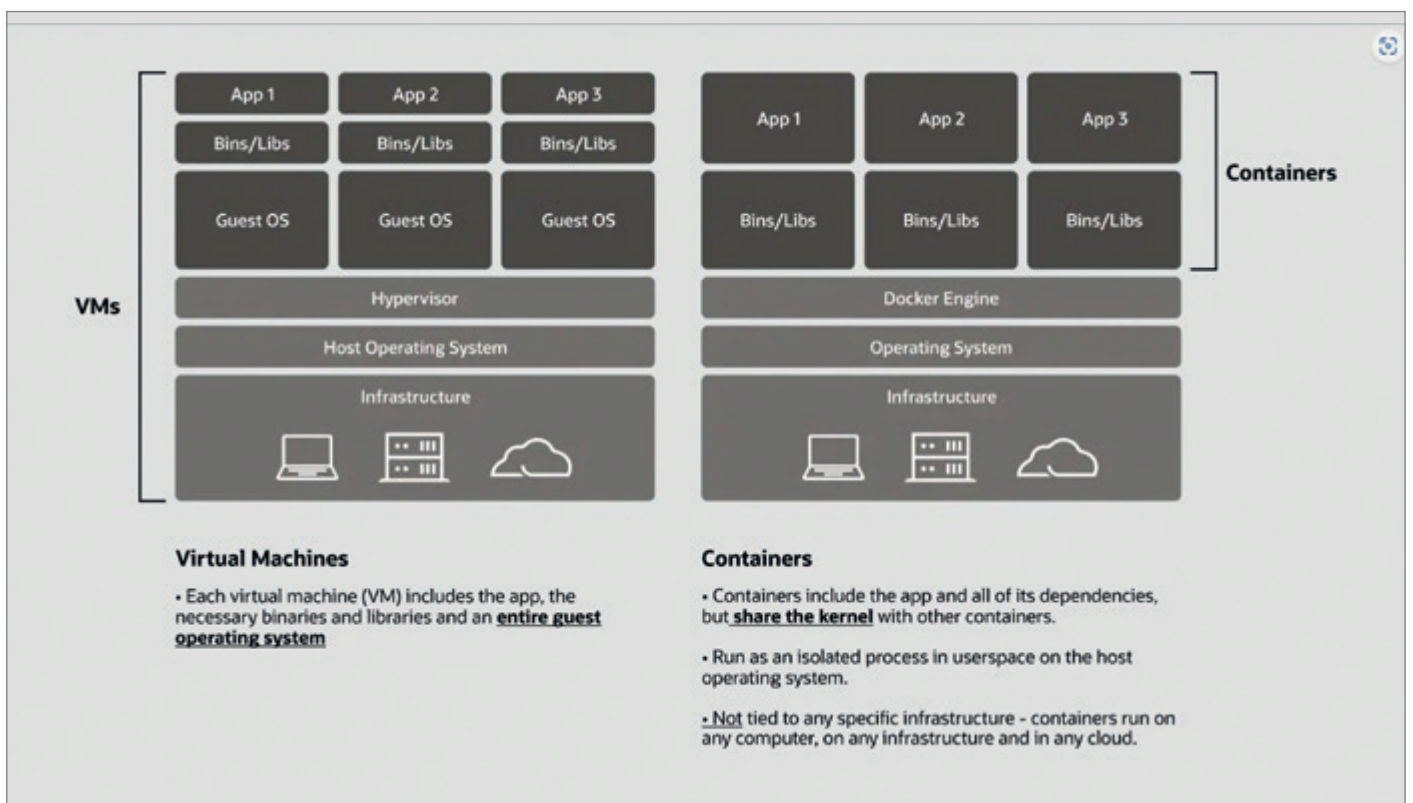


Abbildung 2: Vergleich der Architekturen von Virtuellen Maschinen und Containern (Quelle: Oracle)

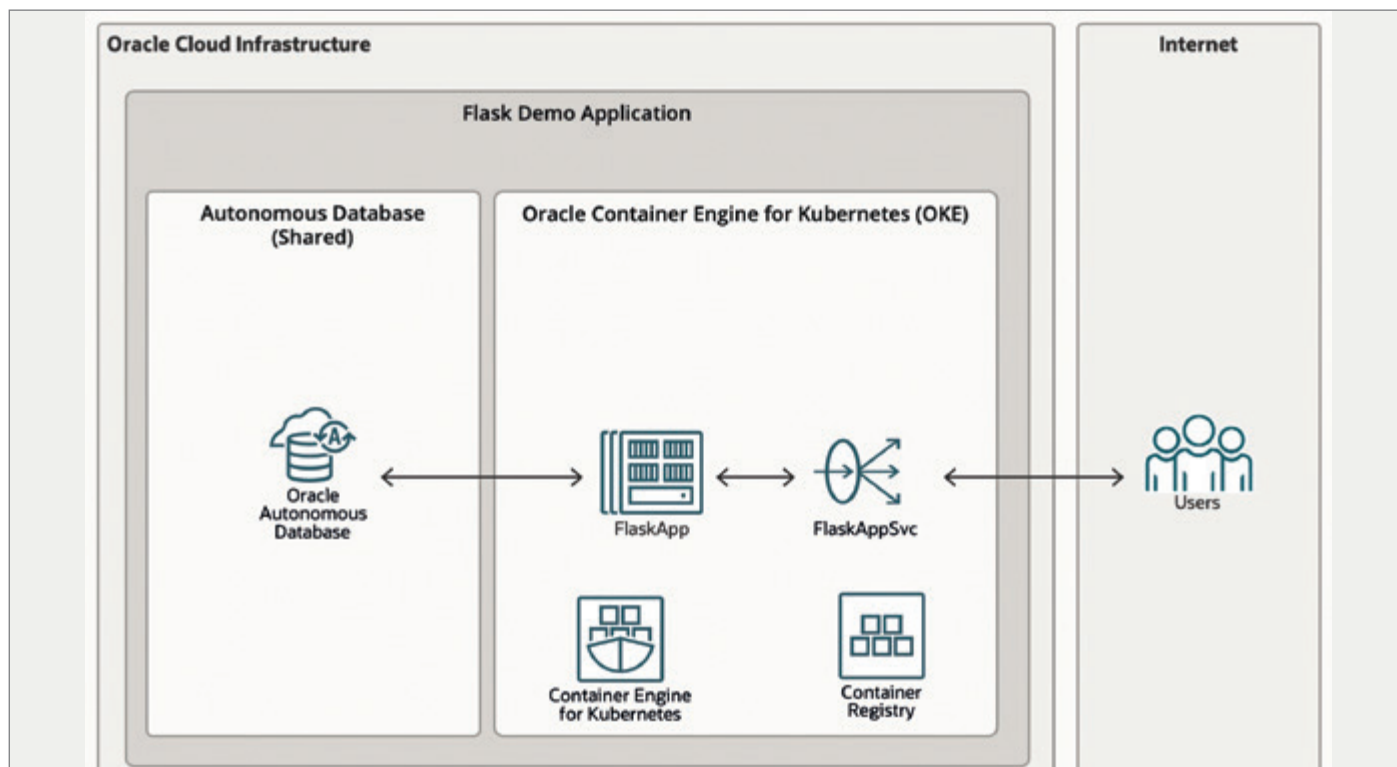


Abbildung 3: Betrieb der ehemals lokalen Anwendung in Kubernetes aus der Vogelperspektive betrachtet (Quelle: Martin Bach)

```
# use Oracle Linux 8 as the basis of this application
FROM container-registry.oracle.com/os/oraclelinux:8
# switch working directory to /code
WORKDIR /code
# remove Python 3.6 and replace it with 3.9
RUN dnf -y module disable python36 && \
    dnf -y module enable python39 && \
    dnf -y install python39 python39-pip python39-setuptools python39-wheel && \
    rm -rf /var/cache/dnf
# now build the Python application
COPY requirements.txt requirements.txt
COPY . .
RUN pip3 install -r requirements.txt --no-cache-dir
# expose the application on port 8080 using Waitress (a Python WSGI server)
EXPOSE 8080
CMD ["waitress-serve", "hitCounter:app"]
```

Listing 1: Dockerfile für eine kleine in Python geschriebene Anwendung

- Rollout/Rollback von neuen Container Images,
- Storage Management,
- und vieles andere mehr.

Die Kubernetes-Architektur unterscheidet grob zwischen einer Control-Plane und den Worker Nodes. Die Control-Plane bietet den Administratoren die API-Endpunkte, mit denen das Verhalten der Container und des Kubernetes-Clusters selbst gesteuert wird. Container werden

dann auf dem passenden Worker Node gestartet und betrieben.

Im Kontext von Kubernetes spricht man allerdings nicht vom Betrieb einzelner Container, sondern von Pods. Ein Pod kann einen oder mehrere Container beheimaten, meistens findet man allerdings pro Pod lediglich einen Container.

Pods werden selten direkt gestartet, sie finden sich meistens in einer Abstraktionsschicht namens Deployment. Ein Deployment beschreibt deklarativ, wie viele

Instanzen eines Containers gestartet werden, diese werden dann von Kubernetes aus der (Container) Registry geladen und anschließend auf die passenden Worker Nodes verteilt. Sollte beim Betrieb eines Containers ein Fehler auftreten, kann Kubernetes diesen stoppen und durch eine neue Instanz ersetzen.

Wie fast alle Konfigurationsdateien heutzutage wird ein Deployment in YAML verfasst. Das eben erstellte Container Image „flaskapp:1.0“ kann folgendermaßen

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: pydemo-deployment
  labels:
    app: pydemo-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: pydemo-deployment
  template:
    metadata:
      labels:
        app: pydemo-deployment
    spec:
      containers:
      - name: flask
        image: registry/namespace/flaskapp:1.0
        env:
          - name: PYTHON_USERNAME
            value: flask_user
          - name: PYTHON_CONNECTSTRING
            value: service-name:port/FREEPDB1
          - name: PYTHON_PASSWORD
            valueFrom:
              secretKeyRef:
                name: flask-user-password
                key: password

```

Listing 2: Start des Container Image „flaskapp:1.0“ in einem Kubernetes-Cluster

```

apiVersion: database.oracle.com/v1alpha1
kind: AutonomousDatabase
metadata:
  name: doag-demo
spec:
  details:
    compartmentOCID: ocid1.compartment.oc1.....
    dbName: DOAG
    displayName: "ADB-S database"
    dbWorkload: OLTP
    dbVersion: "19c"
    cpuCoreCount: 1
    adminPassword:
      ociSecret:
        ocid: ocid...
    dataStorageSizeInTBs: 1
    networkAccess:
      accessType: PRIVATE
      privateEndpoint:
        subnetOCID: ocid1.subnet.oc1.....
        nsgOCIDs:
          - ocid1.networksecuritygroup.oc1.....
      isMTLSConnectionRequired: false

```

Listing 3: Bereitstellung einer Autonomous Database mittels des OraOperators

ßen in einem Kubernetes-Cluster gestartet werden (siehe Listing 2).

Pods kommunizieren untereinander mit Hilfe von Services. Ein Service ist ein weiteres API-Objekt in Kubernetes. Sie erlauben es, die Anwendung für Endnutzer bereitzustellen. In Oracles Container Engine for Kubernetes kann letzterer automatisch einen Load Balancer erstellen, mit dem die Anwendung erreichbar ist.

Möchte man etwa eine ehemals lokale Anwendung in Kubernetes betreiben, erhöht sich die Komplexität wie in *Abbildung 3* ersichtlich. Diese vereinfacht absichtlich viele Aspekte, um den Blick auf das Wesentliche zu fokussieren. Da wichtige Sicherheitsaspekte in diesem Schaubild nicht berücksichtigt sind, eignet es sich nicht als Vorlage für eigene Anwendungen!

In diesem stark vereinfachten Beispiel wird das Frontend (basierend auf Python Flask) über einen Load Balancer im Internet bereitgestellt. IAM Policies und Network Security Lists stellen sicher, dass lediglich autorisierte Benutzer Zugriff auf den Service haben. Der Load Balancer wird automatisch von dem Kubernetes „FlaskAppSvc“ erstellt, der als Typ „Load-Balancer“ definiert ist. In produktiven Anwendungen wird hingegen oft ein Ingress Controller verwendet, der flexibleres Request Routing und ein mehr an Sicherheit bieten kann.

Die Flask-Anwendung persistiert ihre Daten in einer Autonomous Database, was direkt zum nächsten Punkt führt: stateful applications.

## Persistenter Speicher in Kubernetes

Container und Pods sind per Definition stateless. Sobald Informationen dauerhaft gespeichert werden sollen, benötigt man persistenten Storage. In Kubernetes ist es natürlich auch möglich, Storage an Pods durchzureichen, allerdings benötigt man dazu mehrere Schritte.

1. Zuerst muss passender Speicher bereitgestellt werden. In der Cloud ist es oftmals Block Storage oder Network Storage. In Kubernetes spricht man von einer Persistent Volume (PV).
2. Will ein Pod diese PV nutzen, muss es dies über einen Persistent Volume

Claim (PVC) tun. Dieses ist ein weiteres API-Objekt in Kubernetes.

3. Sind PV und PVC erstellt, kann das PVC in der Pod-Spezifikation des Deployments verwendet werden und steht dann zur Verfügung. Die Datenbank kann anschließend dort gespeichert werden.

Da sowohl Bereitstellung als auch Betrieb von Datenbanken in Kubernetes nicht ganz trivial sind, hat die Kubernetes-Community seit einiger Zeit das Operator Pattern erdacht. Oracle stellt mit dem Oracle Operator for Kubernetes, kurz „OraOperator“ einen solchen für die Oracle-Datenbank bereit. Der Operator inklusive Dokumentation findet sich auf GitHub (siehe <https://github.com/oracle/oracle-database-operator>).

Der OraOperator erlaubt es Entwicklern, DBAs, DevOps und GitOps-Teams die Zeit und Komplexität des Bereitstellens von Oracle-Datenbanken im Kubernetes-Umfeld zu reduzieren. Indem der OraOperator die bereits bekannte YAML-Syntax zur Konfiguration verwendet, ist es möglich, die gleichen Werkzeuge für Frontend- und Backend-Deployments zu verwenden. Viele Datenbankoperationen im Kontext von Microservices können dank des OraOperator stark vereinfacht werden.

Der Betrieb einer Datenbank in einem Container ist jedoch nicht trivial und erfordert Einarbeitungszeit. Die Verwendung einer Autonomous Datenbank (ADB), außerhalb des Clusters, kann eine einfachere Lösung darstellen und schneller zum Ziel führen.

Das folgende Beispiel, basierend auf dem offiziellen Template (siehe [\[tonomousdatabase\\\_create.yaml\]\(#\)\), verdeutlicht wie eine Autonomous Database mittels des OraOperators bereitgestellt werden kann. Dazu wird die Datenbank mithilfe einer YAML-Datei definiert \(siehe Listing 3\).](https://github.com/oracle/oracle-database-operator/blob/main/config/samples/adb/au-</a></p>
</div>
<div data-bbox=)

Nutzt man nun das Kubernetes-Kommandozeilenwerkzeug kubectl kann die ADB-Instanz erstellt werden. Zu beachten ist, dass die von dem oben abgebildeten YAML-Format erzeugte Datenbank keine Always-Free-Datenbank ist und daher Kosten verursachen kann.

Mehr Informationen zum OraOperator und den unterstützten Datenbanken finden sich in der Dokumentation auf GitHub (siehe <https://github.com/oracle/oracle-database-operator>).

## Zusammenfassung

In den letzten 5 Jahren wurden viele monolithische Anwendungen (zum Teil) zu Microservices konvertiert. Diese Maßnahme wurde notwendig, um die Geschwindigkeit, in der neue Features veröffentlicht werden, zu erhöhen. Eine Microservice-Architektur bietet aber nicht nur Vorteile: Der erhöhte Flexibilität steht eine größere Komplexität gegenüber. Daher ist immer im Einzelfall abzuwägen, ob eine Konvertierung einer bestehenden Anwendung sinnvoll ist.

Container-Technologie ist die Grundlage für die meisten Microservice-Architekturen. Ein Container Image erlaubt es, eine Anwendung effizient zu packen und dank des standardisierten Formates sowohl auf dem Entwickler-Laptop als auch auf Produktion in genau der gleichen Art und Weise zu betreiben. Automatisierung ist hierbei der Schlüssel zum Erfolg, Agile, DevOps, CI/CD sind die Stichworte.

Sobald mehrere Container über viele Container Hosts zu betreiben sind, wird der manuelle Aufwand diese zu betreiben zu hoch. Eine Orchestrierungs-Engine wie Kubernetes hilft den Überblick zu behalten und die verteilte Anwendung effizient zu betreiben. Per Definition sind Container stateless, was dem Betrieb von Datenbanken nicht förderlich ist. Obwohl es möglich ist, Storage Kubernetes Pods zuzuweisen, ist dieser Ansatz nicht sehr komfortabel. Das Operator Pattern, wie im OraOperator implementiert, trägt maßgeblich dazu bei, den Betrieb der Oracle-Datenbank in der Cloud zu vereinfachen. Aber genau wie im Falle der Konvertierung einer monolithischen Anwendung in eine Microservice-Architektur ist auch hier abzuwägen, ob der Betrieb einer Datenbank im Container nicht zu aufwendig ist. Der OraOperator kann ebenso eine Autonomous-Datenbank verwalten, was im Zweifelsfall leichter fallen sollte.

## Über den Autor

Martin Bach arbeitet als Produktmanager bei Oracle und hilft Kunden in Europa und überall auf der Welt ihre IT-Probleme zu lösen. Er interessiert sich insbesondere für Cloud-Technologien, DevOps, und wie diese sich am besten mit der Oracle-Datenbank kombinieren lassen.



Martin Bach  
martin.b.bach@oracle.com



# Die Qual der Wahl: Exadata Database Machine On-Premises oder Cloud Service?

Oliver Röniger, Oracle Deutschland

Viele Oracle-Datenbankkunden nutzen Oracle Exadata seit Jahren für ihre geschäftskritischsten operativen Systeme, als analytisches Data Lakehouse oder auch als hochperformante Konsolidierungsplattform multipler Oracle Datenbankworkloads. Am Ende der Laufzeit der bestehenden Exadata muss jeder Kunde entscheiden, wie es weitergehen soll. Es gibt neben der etablierten Exadata Database Machine On-Premises auch zwei weitere Exadata-Varianten: die Exadata Database Cloud@Customer und Exadata Database Dedicated als Cloud Service in der Oracle Public Cloud (OCI). Bei der Bewertung sollte es keinesfalls nur um die Kosten gehen. Auch ein schlichtes Lift & Shift in die Cloud ist eigentlich zu kurz gesprungen, es können weitere Potentiale erschlossen werden. In diesem Beitrag wird aufgezeigt, welche Aspekte berücksichtigt werden sollten, um eine fundierte Managemententscheidung zu treffen.



## Baseline

Im Folgenden gehen wir davon aus, dass zwischen den drei Varianten der Exadata eine Auswahl getroffen werden soll (siehe *Abbildung 1*).

OCI Dedicated Region und EU Sovereign Cloud werden im Folgenden nicht explizit betrachtet, auch wenn hier jeweils Exadata technisch mit verbaut ist. Das sind beides Spezialfälle, die ein bestimmtes Geschäftsmodell des Kunden oder Partners erfordern.

In der Exadata-Entscheidungsvorlage für das Management stehen als maßgebliche entscheidungsrelevante Kategorien die Cloud-Machbarkeit, die Flexibilität und Zukunftsfähigkeit sowie die Kosten im Mittelpunkt der Betrachtung. Die Gewichtung dieser Kategorien kann je nach Kundenanforderung sehr unterschiedlich ausfallen, bis hin zur K.-o.-Entscheidung aufgrund eines einzelnen Kriteriums.

In den Projekten haben sich folgende Kriterien, die nicht alle überschneidungsfrei sind, als entscheidend herauskristallisiert (siehe *Abbildung 2*).

Wir beginnen mit der wichtigen Frage der Cloud-Machbarkeit: Ist das aktuelle System überhaupt cloudfähig oder sprechen gewichtige Gründe dagegen?

## 1 Eigenes Rechenzentrum versus Cloud First

Viele Unternehmen haben eine klare Cloud-First-Strategie, sie reduzieren ihre bestehenden Rechenzentren deutlich oder lösen sie komplett auf. Das ist entweder kostengetrieben oder das Resultat einer Sustainability-Strategie: Cloud ist grüner, denn Cloud optimiert die Ressourcenausnutzung („do more with less“), das Teilen von Kapazitäten mit anderen ist ökologischer als das eigene Vorhalten von Leerlauf.

Damit ist Exadata Database Machine (ExaOP) für Cloud-First-Kunden nahezu ausgeschlossen. Demgegenüber ist Exadata Database Dedicated (ExaCS) als reine Cloudlösung prädestiniert. Exadata Database Cloud@Customer (ExaCC) kommt auch in Betracht, da der Betrieb cloudbasiert erfolgt. Allerdings ist weiterhin eine eigene Stellfläche notwendig (oder bei einem Anbieter wie Equinix oder anderen Hostern abbildbar).

Im Gegensatz zu Cloud First gibt es auch Kunden, die aus unterschiedlichen Gründen weiterhin voll auf eigene Rechenzentren setzen, zumindest für ihre unternehmenskritischen Kernanwendungen. Diese präferieren klassisch eher

ExaOP oder manchmal auch ExCC. ExaCS scheidet hier aus (K.-o.-Kriterium).

## 2 Datensouveränität

Gesetzliche Regularien und die Sorge vor unberechtigtem Datenzugriff vor allem auf personenbezogene, sensitive Daten sind maßgebliche Gründe, warum ganze Branchen die Public Cloud auch heute noch für zahlreiche Datenbereiche weiterhin kategorisch ausschließen. Typische derartige Branchen sind Banken, Versicherungen, Behörden und der Health-Care-Bereich. Damit ist ExaCS in der OCI aus dem Rennen (mögliche Ausnahme: Deployment in der Oracle EU Sovereign Cloud). ExaOP wird von diesen Kunden klassisch präferiert. Da aber ExaCC hinter der Firewall des Kunden steht und keine Daten das eigene Rechenzentrum verlassen, ist auch diese Variante denkbar und immer häufiger in der Praxis anzutreffen [1].

## 3 Latenz

Exadata ist letztlich nur die Datenmanagementschicht, was aber ist mit den Applikationen? Wenn diese extrem schnell-

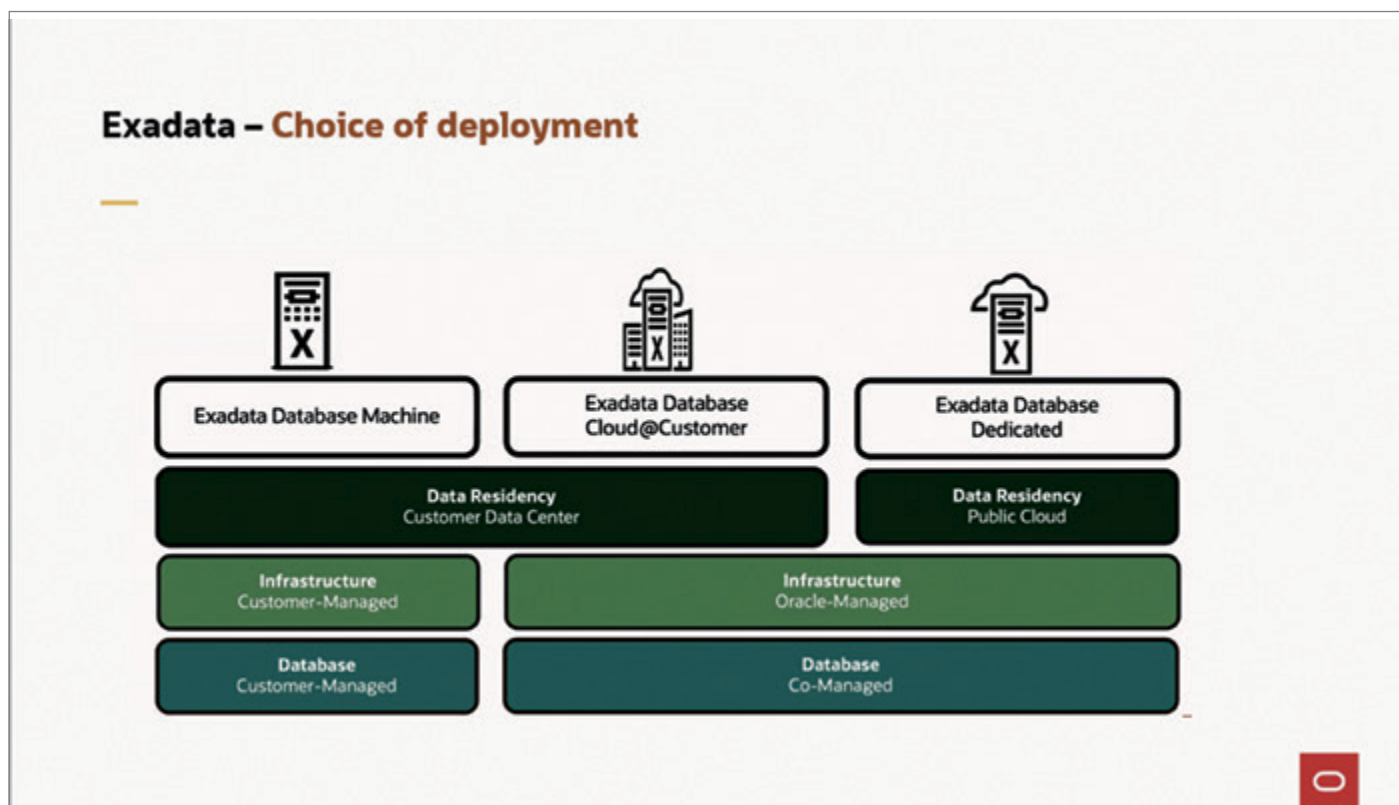


Abbildung 1: Exadata-Varianten (Quelle: Oracle)



Abbildung 2: Entscheidungskriterien (Quelle: Oliver Röniger)

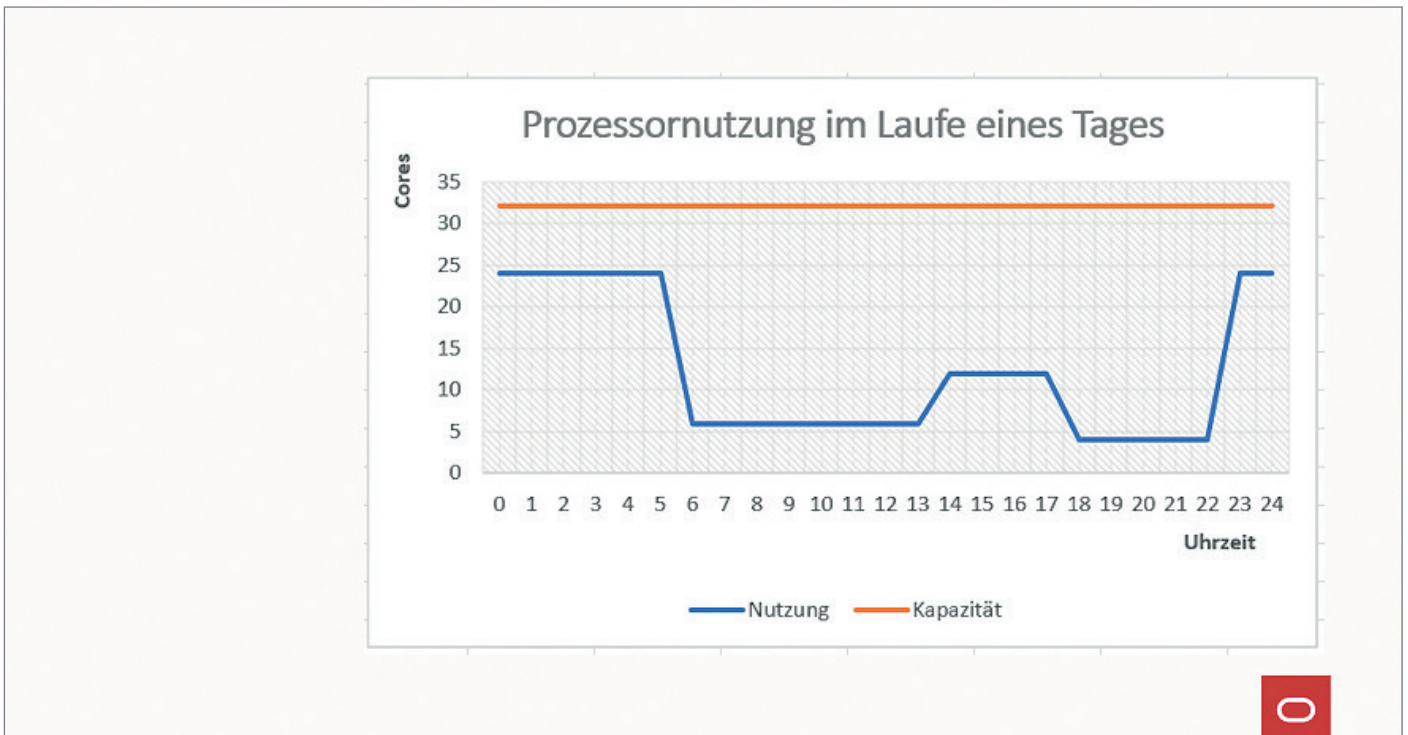


Abbildung 3: Volatilität der Prozessornutzung (Quelle: Oliver Röniger)

le Zugriffszeiten fordern oder „sehr geschwätzig“ sind, hat die ExaCS in der Oracle Public Cloud zunächst schlechte Chancen. Kunden stellen heute die ExaOP nah an den Applikationsserver in ihrem Rechenzentrum und halten dieses Szenario für unschlagbar. Die ExaCC kann

genau an die gleiche Stelle wie die ExaOP aufgestellt werden – also sind diese beiden Varianten pari.

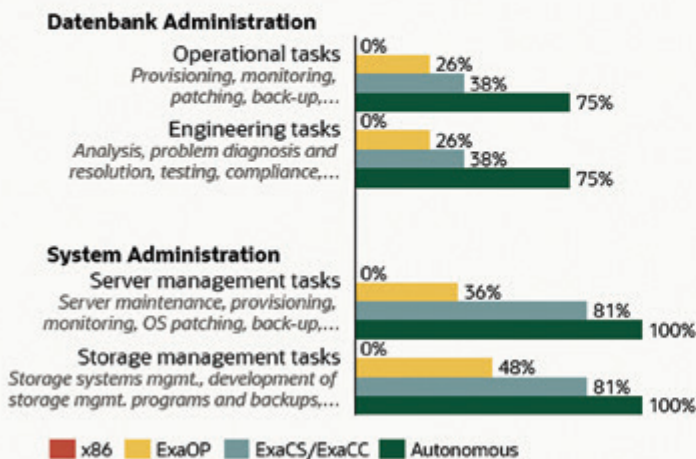
ExaCS kann ebenfalls sehr latenzarm arbeiten, wenn auch die Applikationsserver (über virtuelle Server) mit in die OCI wandern. Nur wenn Datenbank- und Ap-

plikationsschicht beide in der Cloud sind, ist das Szenario aus Latenzsicht optimal, hybrid funktioniert es nicht. Die Cloud muss aber nicht unbedingt nur die OCI sein. Oracle setzt als erster Player im Markt verstärkt auf Multicloud, wobei zum Beispiel die Applikationsserver in Mi-

## Der Wert der Automatisierung

### Operationale Einsparungen im Datenbankbetrieb

#### Beobachtete Effizienzeinsparung (in % der Zeit)



#### Administrationsrollen

Typische administrative Aufgaben, bei denen Effizienzverbesserungen erwartet werden

#### Datenbank Administrator Rolle

- Operational, includes all basic & routine database administrator tasks
- Engineering, includes more complex DBA tasks (e.g., related to specific projects or requests)

#### System Administrator Rolle

- Server management, includes all system administrator tasks related to the operations of servers
- Storage management, includes all system administration tasks related to the deployment and maintenance of the storage-based systems

Hinweis: Diese Einsparungen sind Durchschnittswerte verschiedener Kundenprojekte. Es handelt sich nicht um eine repräsentative Studie oder ein standardisiertes Benchmarking.



Abbildung 4: Operationale Einsparungen im Datenbankbetrieb (Quelle: Oracle)

## Oracle Database Cloud Services – Packages & Options

Nur verfügbar als Exadata Cloud Service

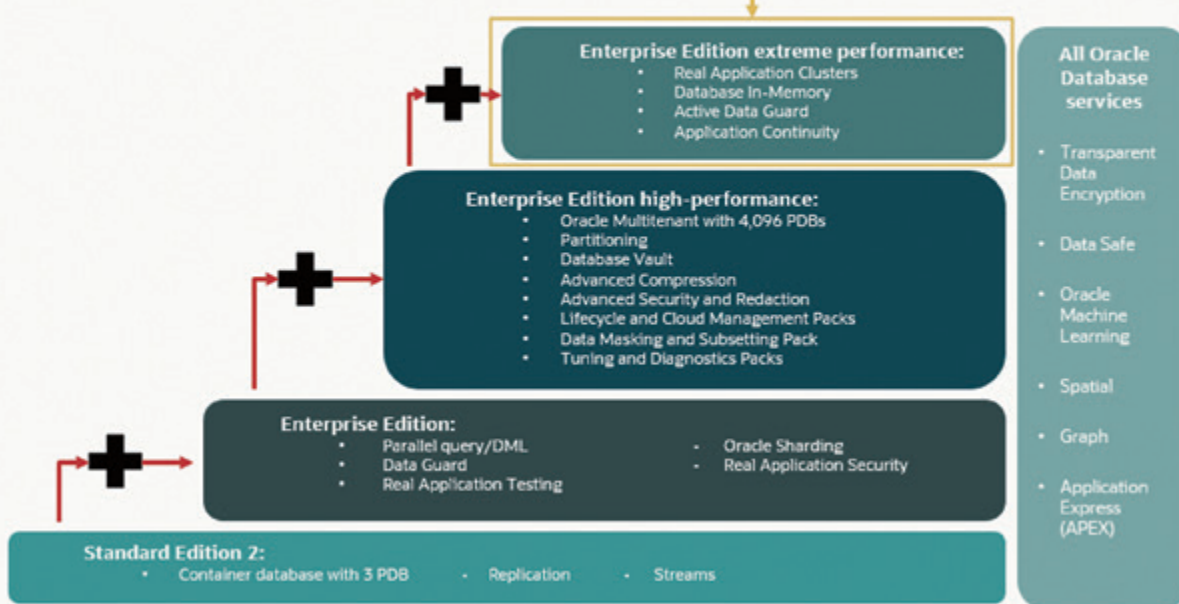


Abbildung 5: Überblick der Oracle Datenbank Cloudservices (Quelle: Oracle)

Microsoft Azure und die Datenbankserver in OCI stehen [2]. Auch für Google und AWS gibt es ähnliche Ansätze. Die Cloudwelt rückt zunehmend enger zusammen.

Die bisher diskutierten drei Kriterien sind entscheidend, denn sie sind es, die manchmal zum K. o. der Cloud Services im Auswahlprozess führen. Die weiteren Kriterien verlieren demgegenüber an Relevanz, auch eine tiefere Kostenevaluierung kann man sich dann sparen.

## 4 Skalierbarkeit

Ein Exadata Cloud Infrastructure Quarter Rack X9M hat im Standard mit zwei Database Servern 128 Cores, bis zu 1.216 Cores kann ein Rack haben. Viele Kunden nutzen die Prozessorkapazitäten ihrer ExaOP heute nicht voll aus, weil sie deutlich weniger Oracle-Datenbanklizenzen zur Verfügung haben. Man kann die Exadata derart regulieren, dass nur Teilmengen der verfügbaren Maschinenkapazität genutzt werden.

In der Cloud mit ExaCS oder ExaCC besteht die Möglichkeit, bei Bedarf alle Cores zu nutzen und im Normalbetrieb oder zu Leerzeiten die Maschine skriptbasiert deutlich zu drosseln (herunterfahren bis auf vier OCPU im Minimum bei RAC möglich). Bei Autonomous Database ist auch Autoscaling nutzbar, so kann man sich die eigenen Skripte sparen [3].

Das dynamische Scaling ist ein großer Mehrwert der Cloudservices. Gerade, wenn die Volatilität der Datenbankworkloads besonders stark ist, entfaltet das Cloudszenario im Vergleich zu einer On-Premises-Umgebung sein wahres Potential. Zumal es viele Business Cases gibt, die nur an wenigen ausgewählten Tagen oder zu sehr spezifischen Zeiten viel Kapazität benötigen und an anderen sehr wenig bis nichts. Als extreme Beispiele seien der Black Friday in einem Webshop oder auch das Monatsende im Controlling genannt. Hier ermöglicht die Cloud neue Business Cases, die On-Premises unmöglich oder viel zu teuer sind.

Die *Abbildung 3* zeigt als Beispiel ein typisches Data-Warehouse- Auslastungsmuster: On-Premises stehen 32 Cores bereit (orange Kapazitätslinie), ca. 25% Puffer im Vergleich zur Spitzenlast wird vorgehalten. Außer für die ETL-Nachtverarbeitung ist die Maschi-

ne stark überdimensioniert, die vorhandenen Ressourcen werden zu mehr als 50% tagsüber nicht ausgenutzt. Eine Cloud kann sich dem realen Verbrauch anpassen, man zahlt nur das, was tatsächlich genutzt wird (Fläche unter der blauen Nutzungskurve).

## 5 Innovation

Autonomous Database ist der innovativste Oracle-Datenbankservice [4]. Dieser ist auf ExaCS oder ExaCC verfügbar, nicht aber auf ExaOP. Eine Entscheidung gegen eine Oracle-Cloud-Lösung schneidet den Kunden also von diesem wichtigen Innovationspfad ab.

Es gibt aber noch ein weiteres gewichtiges Argument, aus Innovationssicht ExaCS deutlich gegenüber den anderen beiden Exadata-Varianten zu präferieren. Ein Vertrag über ExaOP oder ExaCC wird im Standard über vier Jahre geschlossen. Für diese Zeit bindet sich der Kunde fest an die Maschine. Wenn heute eine ExaOP X9M bestellt wird, steht diese auch noch in vier Jahren im Rechenzentrum des Kunden. Selbst wenn es von Oracle bis dahin die X10M oder weitere Nachfolger gibt, hat das keine Konsequenzen für die eigene Installation. Das ist bei ExaCS anders. Die Mindestlaufzeit einer ExaCS beträgt 48 Stunden. Auch wenn man aus Vergleichsgründen (und Konditionssicherheit) einen vier-Jahres-Vertrag mit Oracle abschließt, besteht bei ExaCS die Möglichkeit, zum Beispiel nach zwei Jahren auf das nächste ExaCS Release zu wechseln (z. B. von der X9M auf die X10M). Der Cloudservice-Vertrag ist nicht an eine spezielle Hardware gebunden. Der Kunde ist grundsätzlich frei, wofür er seine Universal Cloud Credits einsetzt.

## 6 Know how

Bei vielen Kunden wird die Exadata nicht vom Kunden selbst betrieben, weil man das notwendige Spezialwissen nicht aufbauen möchte, sondern es erfolgt externe Unterstützung durch Partner (oder Oracle ACS), die sich auf Exadata spezialisiert haben.

Beim Sprung in die Cloud verändern sich Betriebsverantwortlichkeiten und die mit der Exadata verbundenen Arbeitsauf-

wände stark, mit Autonomous Database sogar noch mehr. In *Abbildung 4* wird aufgezeigt, welche Effizienzgewinne potenziell möglich sind.

Automatisierung und Standardisierung sind die beiden wichtigsten Argumente für eine Cloudifizierung. Prüfstein für einen guten Partner sollte immer sein, ob er die Schöpfung der Effizienzgewinne und Innovation unterstützt oder auf den Status quo beharrt.

## 7 Capex versus Opex

Leicht vereinfacht gesagt werden in On-Premises-Projekten Hardware und Lizenzen gekauft und diese Investments stellen Anlagevermögen dar, welches über mehrere Jahre abgeschrieben wird (Capex). In der Cloud hingegen wird mit Subscriptions gearbeitet, dabei handelt es sich um operative Kosten (Opex). Der Kunde kann jederzeit die Cloudnutzung einstellen und es entstehen sofort keine weiteren Kosten mehr („pay as you use“). Vor dem Hintergrund des straffen Fixkostenmanagements in volatilen Zeiten präferieren CFOs aus Gründen der Bilanzpolitik und monetären Flexibilität deutlich die Cloud als Abrechnungsmodell [5].

In dieser CFO-Logik ist ExaOP klassisch Capex und damit vergleichsweise unattraktiv, ExaCS ist Opex und somit sehr interessant. ExaCC ist ein Mix aus beiden, aber aufgrund der zeitlichen Bindung und dem relativ hohen Plattformkostenanteil ist es eher Capex als Opex.

## 8 Oracle-Lizenzsituation

Jeder bestehende Exadata-Kunde hat bereits Oracle-Lizenzen, die bei einer Entscheidung für die nächste Exadata-Generation wieder eingesetzt werden können. Hier ist wichtig zu differenzieren, ob es sich um Lizenzen handelt, die aus einer ULA stammen (aktiv oder auch zertifiziert) oder ob es Einzelverträge für den laufenden Oracle-Softwaresupport gibt, die dem Exadata-Projekt direkt zugeordnet werden können. Wenn ersteres der Fall ist, wird es in der Regel wirtschaftlich sein, BYOL („bring your own licences“) für die Cloudnutzung bei der ExaCS oder ExaCC vorzusehen. Dabei gilt die Regel, dass alle Datenbankoptionen

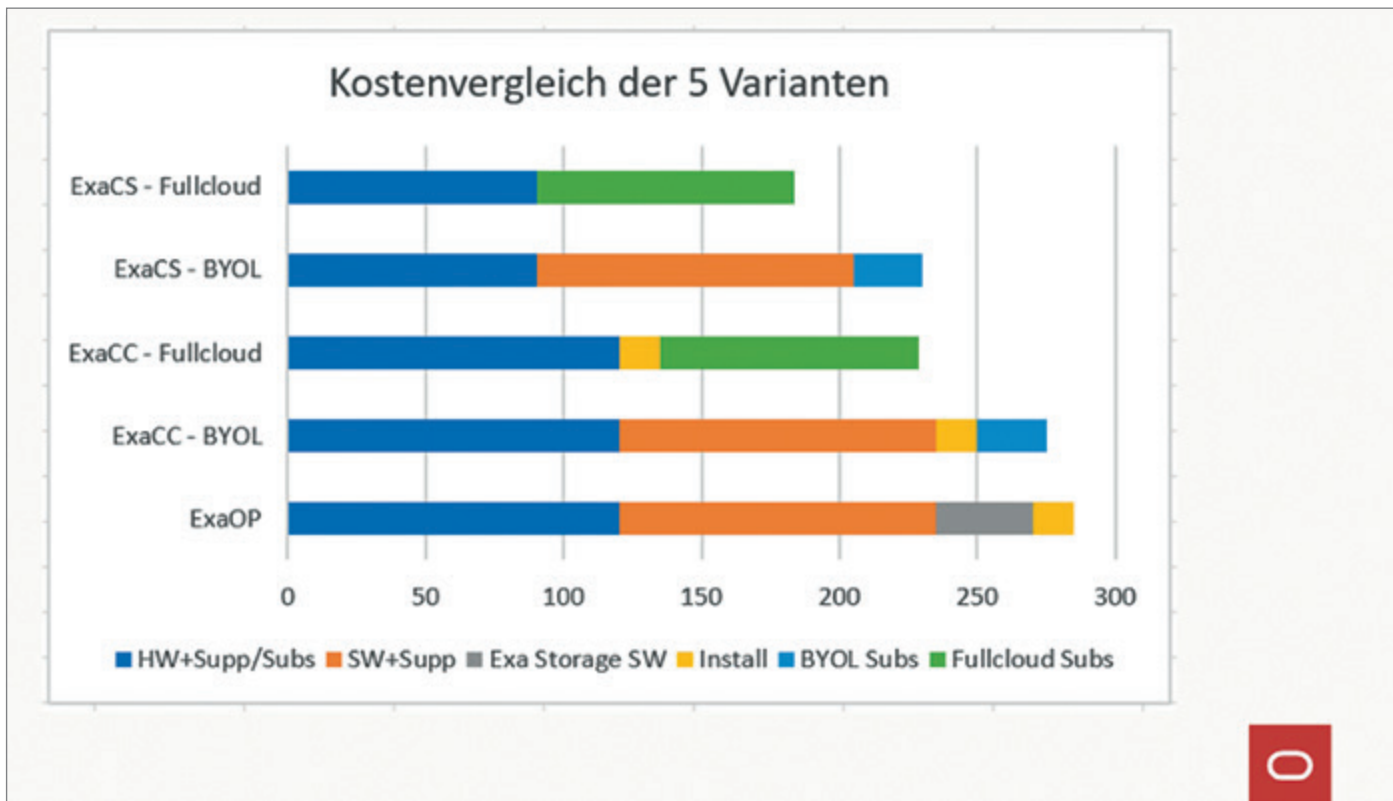


Abbildung 6: Kostenblöcke ExaOP, ExaCC und ExaCS (Quelle: Oliver Röniger)

BYOL						
Unit	Metrik	Menge	Nettopreis pro Stunde	Stunden pro Monat	monatliche Consumption in EUR	jährliche Consumption in EUR
<b>PaaS</b>						
Exadata X9M CS	Quarter Rack dedicated	1	13,50 €	744	10.044,00 €	120.528,00 €
Exadata Database OCPU BYOL	OCPU per hour	24	0,30 €	350	2.520,00 €	30.240,00 €
<b>IaaS</b>						
FastConnect 10Gbit	Port per hour	1	1,19 €	744	882,20 €	10.586,38 €
					<b>PaaS/IaaS Summe</b>	<b>161.354,38 €</b>
					OSR 25%	40.338,59 €
					laufende Supportkosten	115.000,00 €
					<b>Jahr 1</b>	
					<b>TCO</b>	<b>236.015,78 €</b>

Abbildung 7: ExaCS-Kalkulation für BYOL (Quelle: Oliver Röniger)

und -packs, die aktuell unter Support sind, auch in der Cloud genutzt werden dürfen, und zwar in gleicher Anzahl. Es können aber sowohl keine anderen Optionen/Packs als auch keine anderen Mengen genutzt werden. Wenn man zum

Beispiel 40 Prozessorlizenzen der Oracle Datenbank Enterprise Edition hat, kann man damit 80 OCPU Oracle Datenbankservices in der OCI einsetzen. Wenn man dann aber einmalig für zwei Tage 120 OCPU nutzen möchte, geht das mit BYOL

nicht. Damit wird das „Atmen“ verhindert, ein wichtiges Cloudmerkmal wird verletzt. Eine mögliche Lösung besteht darin, temporär auf Fullcloud-Subscriptions umzustellen und danach wieder auf BYOL zu wechseln.

<b>Fullcloud</b>						
Unit	Beschreibung / Metrik	Menge	Nettopreis	Stunden pro Monat	monatliche Consumption in EUR	jährliche Consumption in EUR
<b>PaaS</b>						
Exadata X9M CS	Quarter Rack dedicated	1	13,50 €	744	10.044,00 €	120.528,00 €
Exadata Database OCPU Fullcloud	OCPU per hour	24	1,25 €	350	10.500,00 €	126.000,00 €
<b>IaaS</b>						
FastConnect 10Gbit	Port per hour	1	1,19 €	744	882,20 €	10.586,38 €
					<b>PaaS/IaaS Summe</b>	<b>257.114,38 €</b>
					OSR 25%	64.278,59 €
					laufende Supportkosten	0,00 €
					Jahr 1	
					<b>TCO</b>	<b>192.835,78 €</b>

Abbildung 8: ExaCS Fullcloud (Quelle: Oliver Röniger)

Zeitscheibe	OCPU	Hrs / Day	Days / Year	Hrs / Year	OCPU / Year	%
7 hrs Mo-So // Nacht	24	7	365	2.555	61.320	61%
8 hrs Mo-Fr // Vormittag	6	8	260	2.080	12.480	12%
4 hrs Mo-Fr // Nachmittag	12	4	260	1.040	12.480	12%
5 hrs Mo-Fr // Abend	4	5	260	1.300	5.200	5%
17 hrs Sa-So // Alles ausser nachts	4	17	105	1.785	7.140	7%
starke Peaks: fünfmal im Jahr	48	6	5	30	1.440	1%
<b>Total OCPU / Year</b>	<b>72</b>				<b>100.060</b>	<b>100%</b>

Abbildung 9: Abschätzung benötigter Datenbankstunden pro Jahr (Quelle: Oliver Röniger)

Im Nicht-ULA-Fall ist es oft attraktiver, direkt mit Fullcloud zu kalkulieren und die bestehenden On-Premises-Supportverträge zum nächstmöglichen Zeitpunkt auslaufen zu lassen. Denn im Fullcloudpreis sind der Oracle Support und die notwendigen Upgraderechte entsprechend enthalten. Eine Prüfung, bei der alles genau berechnet wird, lohnt sich.

In der Praxis zeigt sich immer wieder, dass die Kunden für die neue Exadata-Generation nicht ausreichend Lizenzen und Optionen/Packs beistellen können und nachkaufen müssten. Klassische Fälle sind:

- Die relevanten Lizenzen (Oracle Datenbank Enterprise Edition, Partitioning, Diagnostic Pack, Tuning Pack) liegen vor, aber die Workloads sollen größer werden, es besteht eine Mengendifferenz.
- Es fehlt oft RAC als Datenbankoption. Für den optimalen Betrieb der Exadata wurde RAC schon immer empfohlen, bei ExaOP aber oft umschifft, weil viele Kunden RAC als zu komplex empfinden und lieber mit Data Guard arbeiten. In der Cloud wird RAC dringend empfohlen, weil es deutliche Operationsvorteile bringt. Anders

als bei ExaOP ist aber bei ExaCS/ExaCC kein spezifisches RAC Know how erforderlich.

- Es fehlt oft Multitenant als Datenbankoption. Viele Kunden haben noch nicht vollständig auf 19c mit PDB/CDB umgestellt und kommen aktuell mit der Begrenzung auf drei PDBs pro CDB aus.

Mit dem Nachkauf sind unter Umständen größere zusätzliche Lizenz- und Supportkosten verbunden. In einem solchen Fall sollte alternativ Fullcloud kalkuliert werden. Es kann Fälle geben, wo sich der

Lizenznachkauf rechnet, aber in einer Mehrjahresrechnung wird Cloud sehr oft günstiger sein.

Da bei der ExaCS und ExaCC immer die Extreme Performance Edition als Datenbankservice verwendet wird und in dieser alle relevanten Datenbankoptionen und -packs enthalten sind, stellt sich bei Fullcloud die Frage nicht, ob man zusätzliche Funktionen nutzen kann, sie sind bereits vorhanden (siehe *Abbildung 5*).

Eine spezielle Komponente ist die Exadata Storage Software. Diese ist bei der ExaOP klassisch separat zu lizenzieren, in der ExaCS und ExaCC ist diese Komponente kostenfrei enthalten. Das heißt, bei einem Umstieg auf ExaCS oder ExaCC können bestehende Supportverträge der Exadata Storage Software zum nächstmöglichen Zeitpunkt auslaufen. Über vier Jahre gerechnet, ergibt sich oft ein sechsstelliger Einsparbetrag im direkten Kostenvergleich der Alternativen ExaOP versus ExaCS/ExaCC.

## 9 Total Cost of Ownership (TCO)

Am Beispiel eines Exadata Quatterrack X9M wird nun dargestellt, wie sich die Varianten hinsichtlich der relevanten Oracle-Kostenblöcke Hardware, Software beziehungsweise Subscriptions und Installation unterscheiden. Alle zugrunde gelegten Konditionen und Preislogiken entsprechen dem Stand Anfang Juni 2023.

Interne oder externe Personalaufwände und auch die Kosten für Strom/Kühlung/Betriebsfläche werden im Folgenden nicht betrachtet, sie gehören bei einer Vollkostenrechnung aber dazu. Sie werden die Gewichtung ziemlich sicher zusätzlich in Richtung Cloud verschieben.

*Abbildung 6* zeigt die vier Kostenblöcke. Dabei ist bei der Kalkulation der ExaCC und ExaCS jeweils zwischen BYOL und Fullcloud zu differenzieren, deshalb sind fünf Varianten aufgeführt.

Bezogen auf *Abbildung 6* sind einige Erläuterungen notwendig:

- **Hardware (dunkelblau):** Die Hardware inkl. Support ist bei allen drei Varianten preislich gleich kalkuliert. Letztlich ist ExaCS dennoch günstiger, weil hierfür Oracle Support Rewards (OSR)

in Höhe von 25% nachträglich abgezogen werden können. Dabei wurde unterstellt, dass weitere laufende Oracle-Supportverträge beim Kunden existieren. OSR ist bei ExaCC für den Hardwareteil nicht zulässig, bei ExaOP gibt es ohnehin kein OSR.

- **Installation (gelb):** Fällt bei ExaCS nicht an, bei ExaOP und ExaCC sind es verpflichtende Zusatzkosten, einmalig am Projektbeginn.
- **Fullcloud Subscription (grün)** ist oft günstiger als Software plus Support (orange), dass muss je Einzelfall genau betrachtet werden.
- **Exadata Storage Software (grau):** Diese Kosten fallen nur bei ExaOP separat an, bei ExaCS und ExaCC nicht. Hier schlummert je nach Größe des Systems einiges an Einsparpotential.

Im Folgenden wird die Kalkulation der ExaCS genauer betrachtet – sowohl BYOL als auch Fullcloud. Die Kalkulation der ExaCC ist sehr ähnlich (Besonderheit: vier Jahre Laufzeit vorgegeben), die Kalkulation einer ExaOP wird als bekannt vorausgesetzt.

*Abbildung 7* zeigt eine ExaCS-Kalkulation für BYOL.

Die gelb markierten Felder sind die wichtigen Stellschrauben:

- Der Kunde hat Bestandslizenzen für 16 Prozessoren, diese entsprechen 32 OCPU in OCI. Es werden künftig im Normalbetrieb nur noch 24 OCPU benötigt, da die neuere Exadata-Maschine 25-30% effizienter arbeitet als der vier Jahre alte Vorgänger.
- 24 OCPU der Exadata Database Services werden an 350 Stunden/Monat benötigt, also werden insgesamt 8.400 Datenbankstunden pro Monat angesetzt. Unten wird erläutert, wie es zu dieser Schätzung kommt.
- Der BYOL-Preis je Datenbankstunde beträgt 30 Cent.
- Die laufenden Supportkosten im Jahr beziehen sich auf 16 Prozessoren Oracle Datenbank Enterprise Edition plus einige DB-Optionen und -Packs. Diese Verträge laufen bei BYOL weiter.

In *Abbildung 8* wird die gleiche Kalkulation für Fullcloud dargestellt.

Die gelb markierten Felder zeigen die Unterschiede:

- Der Fullcloudpreis beträgt 1,25 Euro pro Datenbankstunde. Damit explodieren die Kosten für die Datenbankprozessoren (vorher 30.240 Euro, nun 126.000 Euro).
- Da Fullcloud als Subscription gewählt wird, kann aber der laufende Supportvertrag in Höhe von 115.000 Euro zum Vertragsende entfallen (sofern das vertraglich möglich ist, siehe ULA-Diskussion oben). Darin liegt ein großer Hebel, insbesondere wenn man das Szenario mehrjährig betrachtet und On-Premises-Supporterhöhungen von 8% pro Jahr einpreist. Die Fullcloudkosten können für vier Jahre mit konstanten Preisen verhandelt werden

Die Fullcloud kostet im gezeigten Beispiel ca. 80% im Vergleich zur BYOL-Kalkulation. Ähnliche Größenordnungen zeigen auch andere Fälle. Das allerdings nur, wenn die Oracle Support Rewards (OSR) angesetzt werden können, ansonsten ist das Fullcloud-Szenario hier ca. 10% teurer als BYOL.

Eigentlich sind die relativ geringen Preisdifferenzen zwischen BYOL und Fullcloud nicht der entscheidende Punkt. Es geht eher um die Frage, Cloud Ja oder Nein. Der Entscheider sollte sich hinsichtlich Fullcloud fragen, ob nicht eher diese beiden Argumente maßgeblich sind:

- Mit Fullcloud kann man alle Datenbankoptionen und -packs nutzen (siehe *Abbildung 5*). Mit BYOL ist man auf die Produkte beschränkt, für die ein On-Premises-Supportvertrag besteht. Die technischen Möglichkeiten mit Fullcloud sind damit ganz andere.
- Mit Fullcloud kann man bei Bedarf stellenweise deutlich mehr Prozessoren nutzen, wenn das Geschäft es erfordert. Mit BYOL ist das über den laufenden On-Premises-Supportvertrag limitiert.

Ein genauer Blick in die Kalkulationen oben zeigt, wie entscheidend die valide Schätzung der tatsächlich benötigten Datenbankstunden ist. Cloud rechnet sich nie im Vergleich zu On-Premises, wenn permanent eine Vollausslastung der Maschine gegeben ist. Ein Monat hat 744 Stunden. Nur sehr wenige Kunden haben ein einzelnes Exadata Quatterrack im Einsatz, welches unter Maximallast 24x7

läuft. Oft werden deutlich geringere Auslastungen von durchschnittlich 30-40% bezogen auf die vorhandene Lizenzmenge gefahren.

Für unser Data-Warehouse-Beispiel wurden Auslastungen gemäß *Abbildung 9* ermittelt (unter Zuhilfenahme von AWR Reports und im Rahmen eines sechswöchigen Proof of Concepts erprobt).

Diese 100.000 Datenbankstunden pro Jahr entsprechen 24 OCPU, die 350 Stunden pro Monat laufen ( $24 \times 350 \times 12 = 100.800$ ). Diese Parameter wurden in den Kalkulationen oben angesetzt. Die *Abbildung 9* zeigt deutlich, dass real ein sehr sprunghaftes Nutzungsverhalten zugrunde liegt. Ein hoher geschäftlicher Mehrwert entsteht durch die starken Peaks, wobei fünfmal im Jahr dreimal so viele Prozessoren (72) genutzt werden als im normalen Alltag (24). Diese Peaks machen aber nur 1% der Gesamtdatenbankstunden aus und fließen gleichberechtigt in die Kalkulation ein. In einer On-Premises-Welt ist so etwas nicht abbildbar. Umso öfter diese sprunghaften Muster in Projekten vorkommen, desto eher ist die Cloud relevant.

## Fazit

Als Management Summary ist festzuhalten, dass die Exadata in der Cloud meist günstiger ist als die Exadata On-Premises-Lösung. Dabei kann sie leicht mit größeren Peaks der Datenbanknutzung umgehen, womit neue Geschäftspotentiale erschlossen werden können. Außerdem bringt ExaCS bezüglich der vertraglichen Bindung eine größere Flexibilität mit (Fixkostenmanagement, Capex).

Solange die ExaCS beziehungsweise ExaCC noch nicht voll ausgelastet ist (oben im Beispiel nur 24 OCPU genutzt, außer bei den Peaks), sollte der Kunde immer überlegen, ob nicht weitere Datenbankworkloads auf der Maschine konsolidiert werden können. Die fixen Plattformkosten sind durch das originäre Exadata-Projekt schon bezahlt, jeder weitere Datenbankworkload kostet als variable Kosten nur 1,25 Euro/Stunde. Viele Unternehmen haben neben der Exadata weitere Postgres- oder MongoDB- oder MySQL-Datenbanken für nicht so kritische Anwendungen auf anderer Hard-

ware laufen. Hierfür waren die Oracle-Lizenzen und die Exadata bisher immer zu teuer. Deren aktuelle Hardwarekosten und die Open-Source-Supportkosten können mit dem Preis von 1,25 Euro/Datenbankstunde leicht verglichen werden. Eine Migration auf Exadata rechnet sich und ist leicht machbar.

Etwas zu kurz gekommen sind in diesem Aufsatz die starken Vorteile der Autonomous Database Services, die nur in der Cloud nutzbar sind (ExaCS oder ExaCC). Kostenseitig gelten ähnliche Aussagen wie bei der ExaCS, aber durch die extreme Automatisierung werden mit Autonomous Database erhebliche betriebliche Kostenpotentiale durch die Reduktion personeller Aufwände erschlossen. Das Endziel aller Exadata Workloads sollte es daher sein, diese konsequent in Richtung Autonomous-Datenbank-Nutzbarkeit umzubauen. Oracle stellt ein Tool bereit, mit dem die Autonomous-Migrationsfähigkeit einer Oracle-Datenbank selbst geprüft werden kann [6].

Am letzten Ende der Modifikation der Workloads steht dann eventuell nicht immer Autonomous Database Dedicated im Mittelpunkt, sondern sogar Autonomous Database Shared. Der maßgebliche Vorteil von Autonomous Database Shared ist, dass der Kunde mit der zugrundeliegenden Infrastruktur gar nichts mehr zu tun hat. Es fallen auch keine Infrastrukturkosten für die ExaCS an (oben in der Kalkulation ca. 50% der Kosten), diese werden zentral von OCI gestellt. Aber das ist ein Thema für einen weiteren Aufsatz.

## Quellen

- [1] Siehe als aktuelles deutsches ExaCC-Referenzbeispiel die Volkswagen Financial Services <https://www.oracle.com/emea/news/announcement/volkswagen-financial-services-partners-with-oracle-to-accelerate-technology-2023-04-19/>, zuletzt aufgerufen am 2. Juni 2023.
- [2] Siehe Fischer, Kai-Uwe: Multicloud 2.0: Oracle Database Service for Azure (ODSA), in: Red Stack Magazin, Heft 3/2023, S. 66-75.
- [3] Siehe Thome, Bob: Automatically Scaling Exadata Cloud Service Based on Workload (Part 1), <https://blogs.oracle.com/database/post/automatically-scaling-exadata-cloud-service-based-on-workload-part-1>, Blog vom 2. März 2021, zuletzt aufgerufen am 2. Juni 2023.
- [4] Siehe zur Oracle Autonomous Database einführend auch Röniger, Oliver: Oracle

Autonomous Database: Läuft!, in: DOAG Business News, Heft 3/2019, S. 16-20.

- [5] Eine intensivere Betrachtung der Argumente: CapEx Vs. OpEx In Cloud Computing: What's The Difference?, <https://www.cloudzero.com/blog/capex-vs-opex>, zuletzt aufgerufen am 2. Juni 2023.
- [6] Cloud Premigration Advisor Tool (CPAT) Analyzes Databases for Suitability of Cloud Migration, [https://support.oracle.com/knowledge/Oracle%20Cloud/2758371\\_1.html](https://support.oracle.com/knowledge/Oracle%20Cloud/2758371_1.html), Letztes Update 17. April 2023, zuletzt aufgerufen am 2. Juni 2023.

## Über den Autor

Oliver Röniger ist seit Juni 2019 Bereichsleiter für Oracle Cloud Services (OCI) bei Oracle mit Sitz im Büro Düsseldorf. Zuvor war er jahrelang für die Themen Business Analytics und Big Data bei Oracle für den deutschen Markt vertrieblich verantwortlich. Sein Arbeitsschwerpunkt ist der Aufbau von Enterprise-Data-Management-Plattformen.



Oliver Röniger  
oliver.roeniger@oracle.com





DOAG

# Werden Sie DOAG-Mitglied!

„Gemeinsame Interessen gemeinsam vertreten“

**+ 30 % Rabatt auf Veranstaltungen**  
**+ Kostenfreier Bezug unserer Zeitschriften**

Red Stack Magazin inkl. Business News und Java aktuell

Ab 120 EUR/Jahr (zzgl. MwSt.)

[www.doag.org](http://www.doag.org)



# Automatisiertes Patching der Oracle Exadata

Pavel Tolstorukov, Accenture

Es ist wichtig, Oracle-Produkte regelmäßig zu patchen, um die Sicherheit, Stabilität und Leistung der Systeme zu gewährleisten. Obwohl das Patch-Verfahren gut dokumentiert ist, hat sich in der Praxis gezeigt, dass es eine aufwändige und zeitraubende Aufgabe ist. Je komplexer das Oracle-Produkt ist, desto aufwändiger ist es, das System zu patchen. Eines der komplexeren Produkte von Oracle ist Exadata, ein für Oracle-Datenbanken optimiertes Engineered System. Es enthält mehrere Komponenten, zum Beispiel Hardware wie etwa Cell oder Database Server und Software wie Grid-Infrastruktur oder Relational Database Management System (RDBMS). Mit Hilfe von Ansible Playbooks und dem Jenkins-Automatisierungsserver ist es möglich, den Patch-Aufwand und die Fehlerquote signifikant zu reduzieren. In diesem Artikel wird anhand eines Exadata-Database-Server-Beispiels die Entwicklung eines automatisierten Patching-Prozesses beschrieben und es werden die notwendigen DevOps-Komponenten eingeführt.

## Beispiel-Konfiguration der Oracle Exadata

Eine Engineered System Oracle Exadata kann man sich wie einen Schrank vorstellen, in welchem verschiedene Hardware mit vorinstallierter Software miteinander verbunden ist, um eine performante, zuverlässige und skalierbare Datenbankplattform zur Verfügung stellen.

In *Abbildung 1* ist eine schematische Repräsentation des Exadata Systems dargestellt. Oracle empfiehlt, jede der einzelnen Komponenten (Database Server, Storage Server, Network Switches usw.) regelmäßig zu patchen.

Die zeitaufwändigsten Komponenten für das Patchen sind die Database Server. Das Patching der anderen Komponenten, wie zum Beispiel Network Switch und Storage Server, lässt sich in deutlich weniger Schritten erledigen. Im Folgenden betrachten wir daher das Patching eines Database Servers.

Eine der vielen möglichen Database Server Konfigurationen ist ein virtualisiertes System. In *Abbildung 2* ist ein Schema einer Database-Server-Beispielkonfiguration zu sehen. Die einzelnen Komponenten haben folgende Funktionen:

- Der Hypervisor verwaltet die Ressourcen des Systems und führt mehrere virtuelle Maschinen aus.
- In den Domänen werden die einzelnen virtuellen Maschinen ausgeführt.
- Die primäre Management-Domäne, dom0, startet und verwaltet die weiteren unprivilegierten Domänen (domU).
- Die dom0 enthält das Image-Repository mit den Images und virtuellen Disks von domU sowie die domU-Konfigurationsdateien.
- Auf den domU werden Grid Infrastructure und RDBMS installiert und eine oder mehrere Multitenant-Datenbanken (CDB) erstellt.

Alle Komponenten, die dom0, die domU, die Grid Infrastructure, das RDBMS und die Datenbanken müssen gepatcht werden.

## Oracle-Exadata-Patching-Prozess

Oracle stellt eine gute, detaillierte Dokumentation zum Patch-Verfahren zur Ver-

fügung. Die Liste der Aufgaben ist aber sehr lang.

Um beispielsweise die primäre Management-Domäne dom0 zu patchen, sind unter anderem die folgenden Schritte auszuführen:

- Eine dom0 für die Patch-Manager-Software-Installation definieren.
- Patch-Manager-Software herunterladen und auf dom0 deployen.
- Auf den domU die GI stoppen und deaktivieren.
- Patch Manager mit passenden Parametern starten.
- ...

Die Liste in *Abbildung 3* zeigt beispielhaft die notwendigen Schritte für das Patching der einzelnen Komponenten des Exadata Database Servers.

Das Patchen der anderen Komponenten des Database Servers ist ebenso anspruchsvoll und aufwändig. Die einzelnen Aufgaben sowie die ungefähre Zeiteinschätzung der **reinen** Arbeitszeit (einige

Tasks wie root.sh oder GI Home Switch können lange ohne menschlichen Eingriff laufen) sind ebenfalls in der Abbildung aufgelistet.

Für das vollständige Patchen eines Database Servers ist – je nach Konfiguration – mit einer Arbeitszeit von ca. 7 Stunden zu rechnen. Hat ein Kunde zum Beispiel eine Cloud Plattform mit 8 Full Rack Exadatas mit je 8 Database Servern kommen wir auf ca. 448 Arbeitsstunden.

## Gründe für das Automatisieren des Patchingprozesses

Die Reduzierung der notwendigen Arbeitszeit ist bereits ein großer Vorteil der Automatisierung. Es gibt jedoch noch weitere Gründe, warum es sinnvoll ist, das Exadata-Patching zu automatisieren:

- Wesentliche Verminderung von menschlichen Fehlern

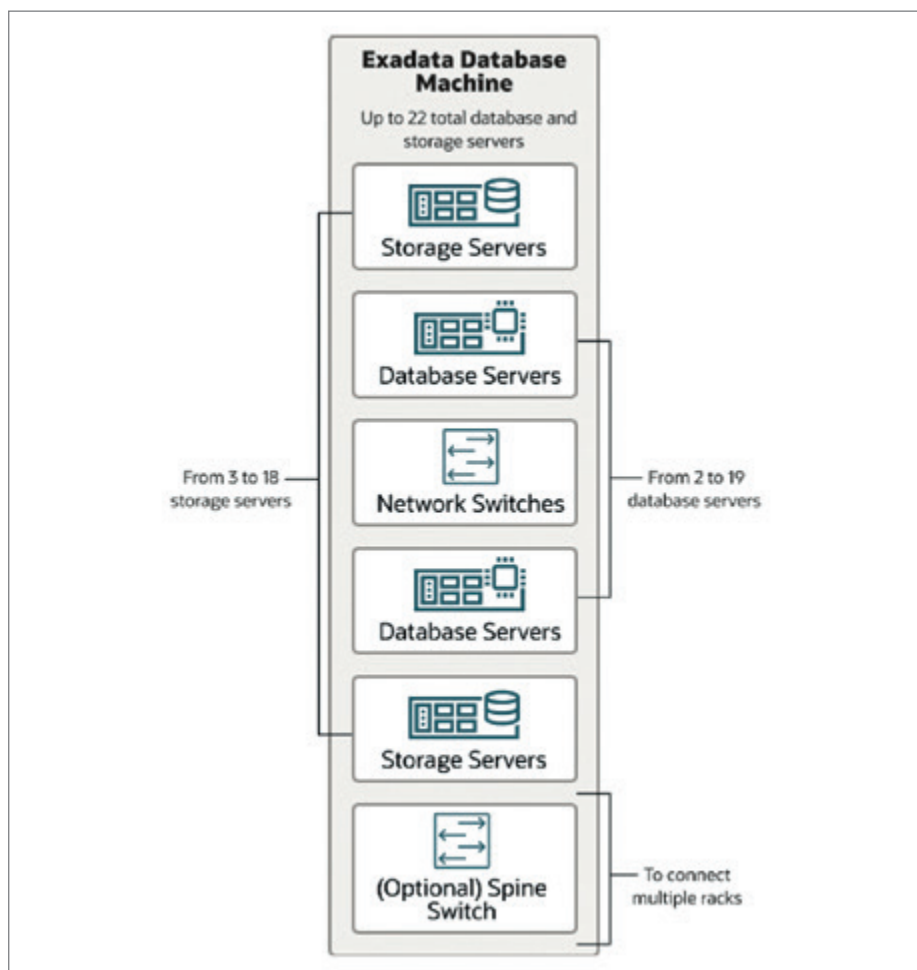


Abbildung 1: Exadata-Komponenten (@Oracle Exadata Database Machine Technical Architecture)

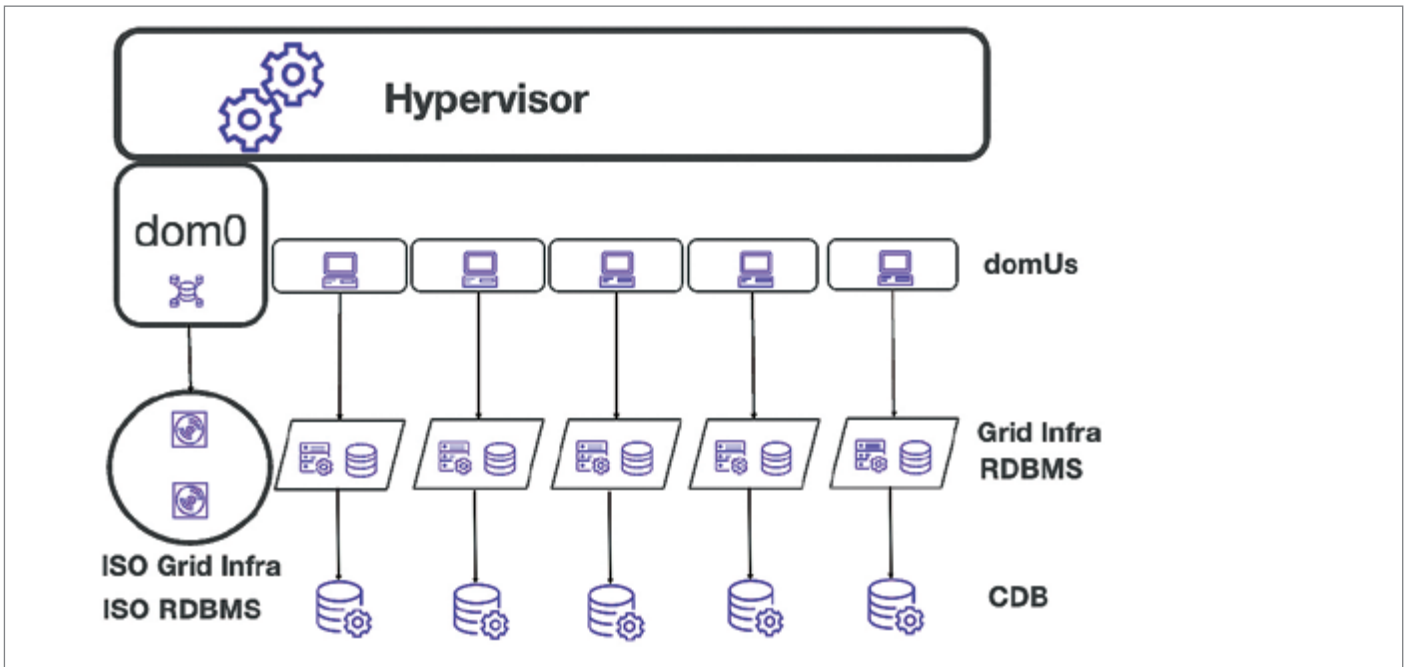


Abbildung 2: Database-Server-Beispielkonfiguration (Quelle: Pavel Tolstorukov)

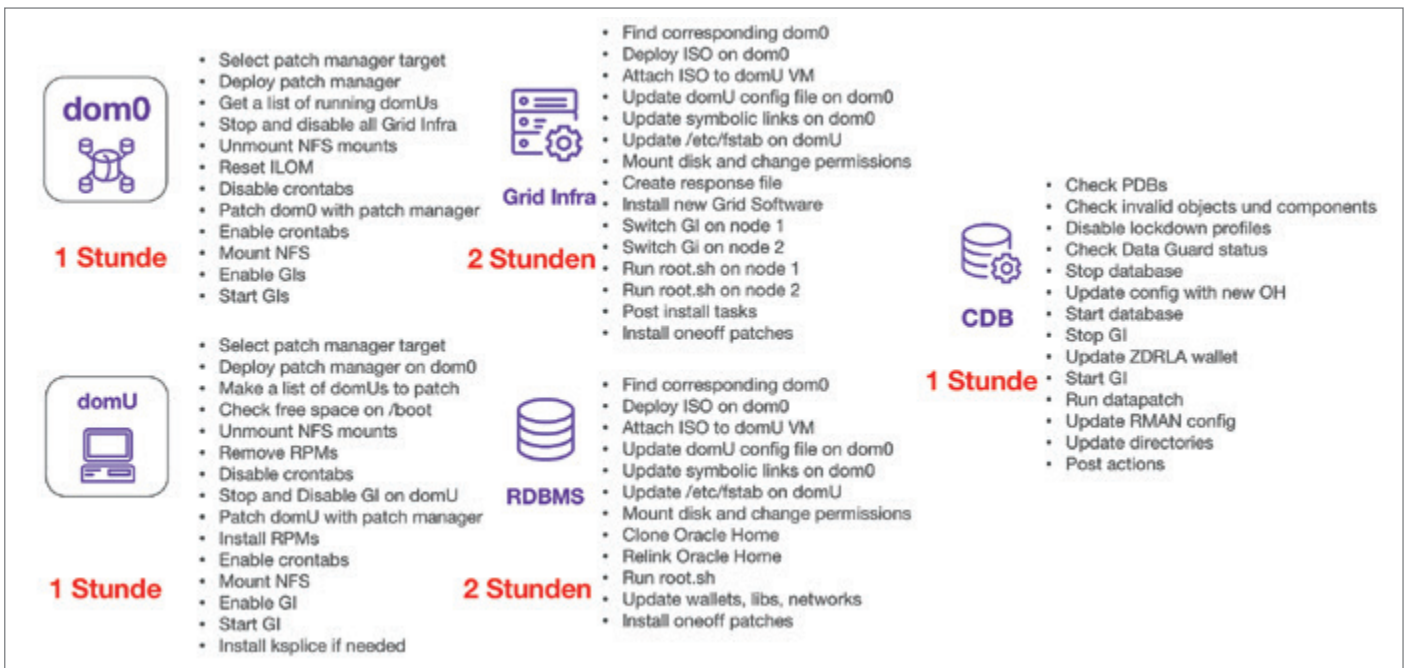


Abbildung 3: Patchen der Exadata-Database-Server-Komponenten (Quelle: Pavel Tolstorukov)

- Standardisierung des Patchingprozesses
- Standardisierung der ganzen Infrastruktur
- Standardisierung des Lifecycle Dev/Test/Prod Workflows
- Reproduzierbarkeit und Vorhersehbarkeit des Patchingprozess
- Nachvollziehbarkeit des Patchingprozess

Für die Modellierung und Ausführung von Patching-Automatisierungen wer-

den unter dem Begriff DevOps zahlreiche Techniken und Methoden bereitgestellt.

### Blick auf DevOps

DevOps beinhaltet unter anderem verschiedene Methoden und Techniken: die Automatisierung der Prozesse, deren Orchestrierung, Versionskontrolle des Entwicklungscodes, Development Pipelines, Verwaltung der Infrastruktur als Code

und Zusammenarbeit zwischen Entwickler und Administratoren.

Im Folgenden gehen wir auf die wesentlichen Methoden und Techniken ein.

### Automatisierung der Prozesse

Eines der wichtigsten DevOps-Werkzeuge, dass für die Automatisierung des Exadata-Patchings angewendet werden kann, ist Ansible.

Ansible ist die Software zur allgemeinen Konfiguration und Administration

von Computern [1]. Man kann mit Ansible-Software installieren, tägliche operative Aufgaben automatisieren sowie die Infrastruktur bereitstellen. Ansible ist agentenlos und kann daher Knoten verwalten, ohne dass darauf eine spezielle Software installiert werden muss.

Die Ansible-Konfiguration sieht vereinfacht folgendermaßen aus:

Auf dem Jump host wird die Ansible-Software installiert. In einer Inventory-Datei sind die Target Hosts definiert, welche von Ansible verwaltet werden sollen, zum Beispiel Targets wie dom0, domU oder Storage Cells. Um Ansible zu verwenden, muss eine password-less ssh-Verbindung zwischen dem Ansible-Server und dem Target bestehen.

Die Ingenieure schreiben die Playbooks und Ansible führt diese auf den

```
- name: "configure oracle user and groups"
  hosts: host1,host2
  gather_facts: true
  tasks:
    - name: "add oracle groups"
      group:
        name: "{{ item }}"
        state: true
      become: true
      loop:
        - oinstall
        - dba
    - name: "add oracle user"
      user:
        name: oracle
        group: oinstall
        groups: dba
        append: true
      become: true
```

Listing 1: Hinzufügen des Benutzers "oracle" mittels Ansible Playbook

## Pipeline upgrade\_GI\_iso\_prd

This build requires parameters:

**TARGET\_HOSTS**  
define the target hosts where to execute the pipeline (e.g. vpdm4102db\* or multiple VMs comma separated)

vpdm4101db01,vpdm4101db02,vpdm4103db0\*

**STAGE\_TO\_BEGIN**  
choose from wich stage begin the pipeline

00\_add\_iso

**GI\_VERSION**  
version of GI to install

19.18.0.0

**ONEOFF\_PATCHES**  
oneoff patches to apply

34932268

**DISK\_NAME**

grid-clone-Linux-x86-64-19000230117

**Build**

Abbildung 4: Jenkins GUI für die Grid-Infrastruktur-Patching-Pipeline (Quelle: Pavel Tolstorukov)

```

if (TARGET_ENVIRONMENT == 'prd')
{ INV = 'inventory/prod/' }
else { INV = 'inventory/test/' }
ENV = "$TARGET_ENVIRONMENT"
pipeline {
agent { label "$ENV" }
stages {
...
stage('stop GIs and patch dom0')
ansiblePlaybook extras: '-e "G_GROUP_NAME=$TARGET_HOSTS e_restart_dom0=$RESTART_DOM0 e_exa_image_
date=$EXA_IMAGE_DATE e_exa_image_version=$EXA_IMAGE_VERSION e_exa_patch_manager_name=$DBSERVER_PATCHMGR_
NAME e_dom0_patchmgr=$PATCHMGR_TARGET e_nodes=$NODES"',
forks: 150, inventory: "$INV" , limit: '$LIMIT',
playbook: 'p_patch_domX.yml', tags: 'patch_dom0'
when { anyOf
{environment name: 'STAGE_TO_BEGIN', value: 'patch_dom0'}}}

```

Listing 2: Jenkins Pipeline DSL

```

...
- name: "patch_domU | check if root crontab exists"
stat:
  path: /var/spool/cron/root
  get_checksum: no
  get_mime: no
register: r_root_crontab

- name: "patch_domU | pre-patching backup root crontab"
copy:
  src: /var/spool/cron/root
  dest: "/u01/{{ l_exa_image_version }}/root_crontab_bkp"
  mode: preserve
failed_when: false
when: r_oracle_crontab.stat.exists

- name: "patch_domU | remove known custom RPMs packages"
yum:
  name: "{{ item }}"
  state: absent
failed_when: false
loop:
  - telegraf-*
  - python-devel-*

- name: "patch_domU | if GI is running - stop GI"
include_tasks: t_stop_disable_GI.yml
vars:
  l_target_to_patch: 'domU'
when: ansible_local['dyndb_grid']['cluster']['oracle_home'] | default('') | length > 0

- name: "block run_once to execute on dom0_delegate"
block:
  - name: "patch_domU | deploy DBSERVER patchmgr utility "
    include_tasks: t_deploy_DBSERVER_patchmgr.yml

  - name: "patch_domU | check if log file patchmgr exists"
    stat:
      path: "{{ l_patchmgr_fqp }}/patchmgr.stdout"
      get_attributes: no
      get_checksum: no
      get_mime: no
    register: r_patchmgr_log

  - name: "patch_domU | backup previous patchmgr.stdout"
    command: "mv {{ l_patchmgr_fqp }}/patchmgr.stdout {{ l_patchmgr_fqp }}/patchmgr.stdout_{{ ansible_
date_time.iso8601_basic_short }}"

```

```

when: r_patchmgr_log.stat.exists

- name: "patch_domU | patch command to execute"
  debug:
    msg: "{{ l_patchmgr_fqp }}/patchmgr -dbnodes /tmp/domU_to_patch -upgrade -yum_repo {{ l_domU_yum_path }} --allow_non_signed_repo --modify_at_prereq --target_version {{ l_exa_image_version }}.{{ l_exa_image_date }}"

- name: "patch_domU | patchmgr execution"
  shell: |
    {{ l_patchmgr_fqp }}/patchmgr -dbnodes /tmp/domU_to_patch -upgrade -yum_repo {{ l_domU_yum_path }} --allow_non_signed_repo --modify_at_prereq --target_version {{ l_exa_image_version }}.{{ l_exa_image_date }}

  environment:
    PATH: /bin:/usr/bin:/usr/sbin:/usr/local/bin
  register: r_domU_patching
  become: true
  failed_when:
    - r_domU_patching.rc != 0
    - ('Exit status:0' not in r_domU_patching.stdout)

- name: "patch_domU | get content of patchmgr.stdout"
  shell: "cat {{ l_patchmgr_fqp }}/patchmgr.stdout"
  environment:
    PATH: /bin:/usr/bin:/usr/sbin:/usr/local/bin
  register: r_domU_patching_out
  become: yes

- name: "patch_domU | output of patchmgr.stdout"
  debug:
    var: r_domU_patching_out.stdout_lines
...

```

Listing 3: Playbook-Auszug für das domU-Patchen in Ansible

```

stages {
  stage('checkout ansible sources') {
    steps {
      git credentialsId: 'modshe_global_access_token', url: 'https://gitlab.com/mods-he/platform.git', branch: '$BRANCH'
    }
  }
}

```

Listing 4: GitLab-Konfiguration im Jenkins Pipeline DSL

Targets aus. Ein Playbook ist eine Datei im YAML-Format, die ein oder mehrere so genannte Plays definiert. Plays bestehen aus einer bestimmten Abfolge von Tasks. Jeder Task ruft ein vordefiniertes Programm oder Modul auf, das eine bestimmte Aufgabe löst.

In Listing 1 ist ein Playbook dargestellt, welches den Benutzer „oracle“ zu den Target-Systemen host1 und host2 hinzufügt.

Die einzelnen Tasks, die entsprechend Ihrer Aufgabe benannt werden sollten, rufen die Module auf. In dem oben gezeigten Listing 1 wird im ersten Task das **group**-Modul mit den zwei Parametern „name“ und „state“ aufgerufen, welches

zwei Gruppen auf dem Target erstellt. Im zweiten Task wird mit einem **user**-Modul ein Benutzer „oracle“ mit der Primary Group „oinstall“ und Secondary Group „dba“ hinzugefügt.

Die Plays beschreiben den gewünschten Zustand des Target-Systems. Wenn Ansible feststellt, dass der aktuelle Zustand des Systems nicht mit der Playbook-Beschreibung übereinstimmt, nimmt Ansible alle in den Plays definierten Änderungen vor, damit das Target-System dem Playbook entspricht.

Wenn Ansible wie in unserem oberen Beispiel feststellt, dass der Benutzer „oracle“ auf dem Target-System bereits

vorhanden ist, und in der richtigen Gruppe ist, werden keine Änderungen auf dem Target-System durchgeführt.

Ansible hat tausende vordefinierte Module. Zusätzlich können mit Hilfe eines generischen "shell"-Moduls beliebige Shell-Befehle auf dem Zielsystem ausgeführt werden. Es ist auch möglich, eigene Module in beliebigen Sprachen wie Python, Perl, PL/SQL oder sogar bash zu schreiben.

In der Praxis können Ansible Playbooks deutlich komplexer gebaut werden. Dafür stehen weitreichende Konstrukte wie zum Beispiel Rollen, Fakten oder magische Variablen zur Verfügung.

```

...
pipeline {
...
stages {
...
stage('patch domU') { steps {
ansiblePlaybook extras: '-e "G_GROUP_NAME=$TARGET_HOSTS e_dom0_patchmgr=$PATCHMGR_TARGET e_exa_image_ver-
sion=$EXA_IMAGE_VERSION e_exa_image_date=$EXA_IMAGE_DATE e_exa_patch_manager_name=$DBSERVER_PATCHMGR_NAME
e_role=$DB_ROLE"',
forks: 150, inventory: "$INV", limit: '$LIMIT', playbook: 'p_patch_domX.yml', tags: 'patch_domU' }

stage('apply ksplices') { steps {
ansiblePlaybook extras: '-e "G_GROUP_NAME=$TARGET_HOSTS e_ksplice_list=$KSPLICE_LIST"',
forks: 150, inventory: "$INV", limit: '$LIMIT', playbook: 'p_patch_domX.yml', tags: 'ksplice_domU' }

stage('start upgrade GI pipeline') { steps {
build job: "upgrade_GI_iso_${ENV}", parameters: [[${class: 'StringParameterValue', name: 'TARGET_HOSTS',
value: TARGET_HOSTS}, [${class: 'StringParameterValue', name: 'ROLLING', value: 'NO'}, [${class: 'StringPa-
rameterValue', name: 'GI_VERSION', value: GI_VERSION}, [${class: 'StringParameterValue', name: 'DISK_NAME',
value: DISK_NAME}, [${class: 'StringParameterValue', name: 'DB_ROLE', value: DB_ROLE}, [${class: 'StringPa-
rameterValue', name: 'ONEOFF_PATCHES', value: ONEOFF_PATCHES}, [${class: 'StringParameterValue', name: 'SET-
PERM', value: 'YES'}, [${class: 'StringParameterValue', name: 'BRANCH', value: BRANCH}, [${class: 'StringPa-
rameterValue', name: 'LIMIT', value: LIMIT]] ] }

stage('start upgrade CDB home pipeline') {
build job: "upgrade_CDB_home_${ENV}", parameters: [[${class: 'StringParameterValue', name: 'TARGET_HOSTS',
value: TARGET_HOSTS}, [${class: 'StringParameterValue', name: 'SKIP_DG_CHECK', value: 'NO'}, [${class:
'StringParameterValue', name: 'NEW_ORACLE_HOME', value: NEW_ORACLE_HOME}, [${class: 'StringParameterValue'},
[${class: 'StringParameterValue', name: 'DB_ROLE', value: DB_ROLE}, [${class: 'StringParameterValue', name:
'BRANCH', value: BRANCH}, [${class: 'StringParameterValue', name: 'LIMIT', value: LIMIT]] ] }
...}}

```

Listing 5: Das Patchen von domU, GI und CDB in einer Pipeline

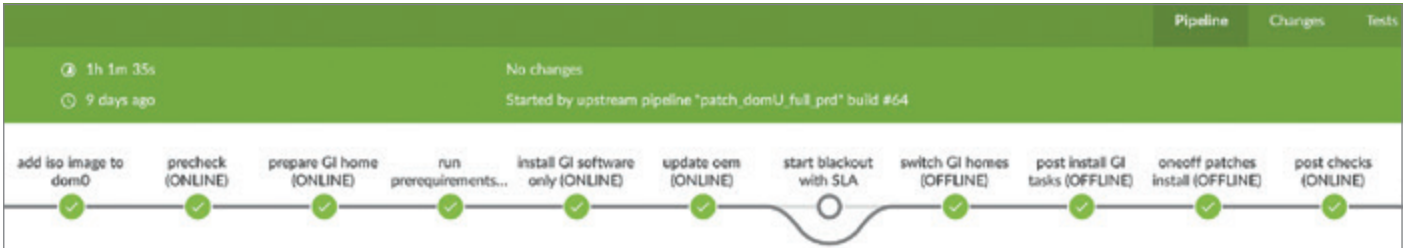


Abbildung 5: Jenkins-Fortschritt für die Grid-Infrastruktur-Patching-Pipeline (Quelle: Pavel Tolstorukov)

```

post {
failure {
  emailx body: "$TARGET_HOSTS: pipeline ${env.JOB_NAME} failed", recipientProviders: [buildUser()],
  subject: '[External] JENKINS ALERT $TARGET_HOSTS: FAIL of patching domB!', to: 'M...@accenture.com devops.dyndb3@...com m'
}
success {
  emailx body: "$TARGET_HOSTS: pipeline ${env.JOB_NAME} finished successfully", recipientProviders: [buildUser()],
  subject: '[External] JENKINS ALERT $TARGET_HOSTS: finish of patching domB!', to: 'M...@accenture.com devops.dyndb3@...com m'
}
}

```

Abbildung 6: Alerting-Konfiguration (Quelle: Pavel Tolstorukov)

**Versionskontrolle und Kollaboration**

Die Ansible Playbooks für eine Exadata-Automatisierung werden als Code geschrieben, der auch von mehreren Spe-

zialisten gemeinschaftlich erstellt werden kann. Um das Management des Playbooks und die Zusammenarbeit zu ermöglichen, benötigt man ein Werkzeug für die Versionskontrolle wie zum Beispiel GitLab.

GitLab ist eine Anwendung zur Versionskontrolle für Softwareprojekte auf Git-Basis. Es bietet ausgereifte Funktionen für eine kollaborative Codeentwicklung und speichert alle Änderungen an Datei-



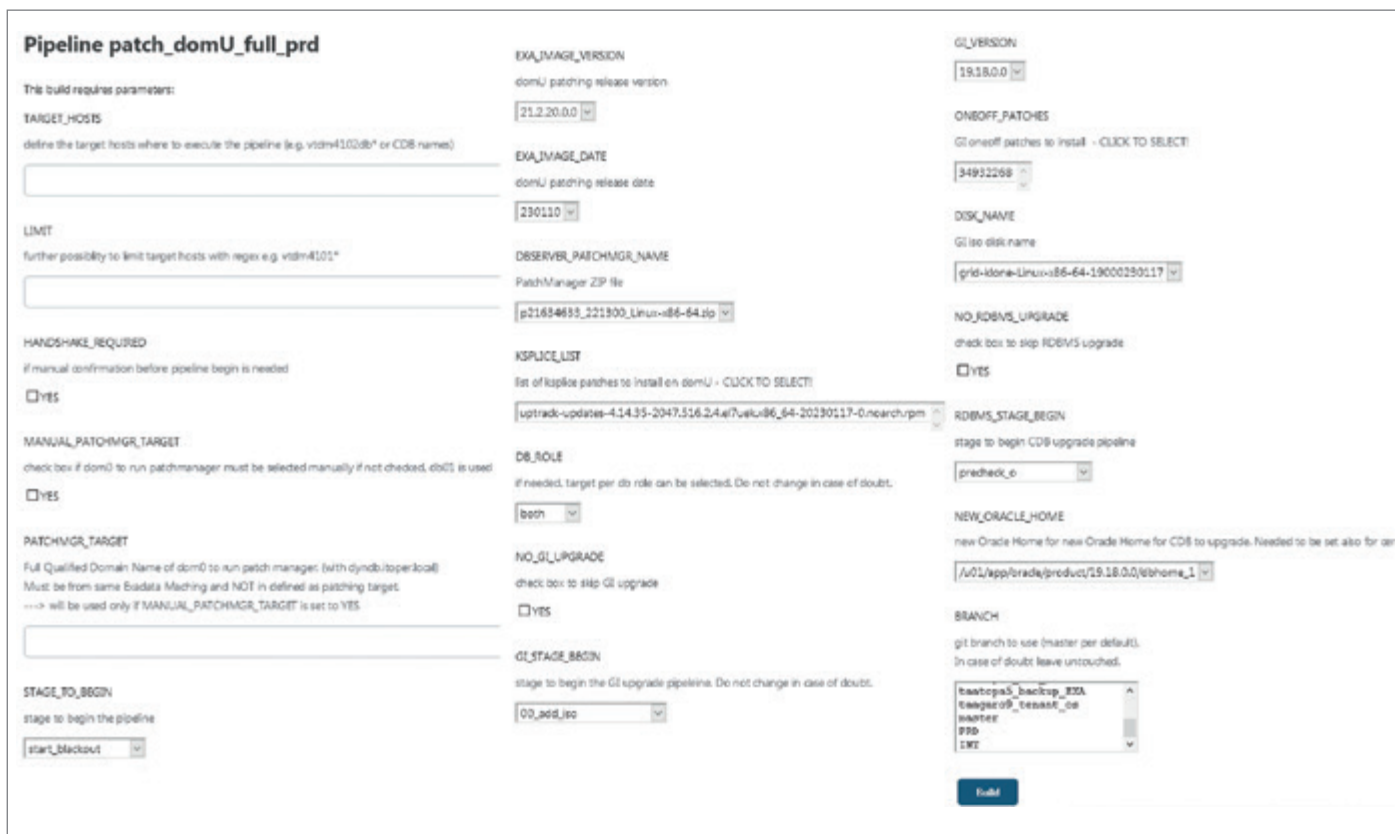


Abbildung 7: Jenkins GUI für domU-Patching einschließlich GI und CDB (Quelle: Pavel Tolstorukov)



Abbildung 8: Jenkins Scheduler (Quelle: Pavel Tolstorukov)

en und ihrem Quellcode, sodass diese jederzeit nachvollzogen werden können.

### Orchestrierung

Es stehen einige Orchestrierungswerkzeuge auf dem Markt zu Verfügung, beispielsweise Bamboo oder Ansible Tower. Ein weiteres ist das auf Java geschriebene, webbasierte System Jenkins.

Das Jenkins-Orchestrierungswerkzeug eignet sich aus mehreren Gründen gut für die Automatisierung des Exadata-Patchings:

- Es ist open-source und mit einer Vielzahl von Plugins erweiterbar. Diese machen Jenkins zu einem universellen Werkzeug.
- Dank spezifischen Plugins kann Jenkins Ansible Playbooks starten und eine Pipeline daraus bauen.
- Jenkins ist mit Git integrierbar und ermöglicht damit die Versionskontrolle mit GitLab.
- Die Skriptsprache DSL (Domain-Specific Language), in der die Pipelines be-

schrieben sind, lässt sich um zusätzliche Groovy-Logik erweitern, z. B. Schleifen, Fallunterscheidungen oder Variablen.

- Auch Pipeline-Variablen können unter Verwendung der Groovy- Programmiersprache definiert werden.
- Jenkins hat außerdem ein fortgeschrittenes Monitoring und Alerting System.

Listing 2 zeigt einen Ausschnitt der Jenkins DSL Pipeline mit if-then-else-Groovy- Lo-

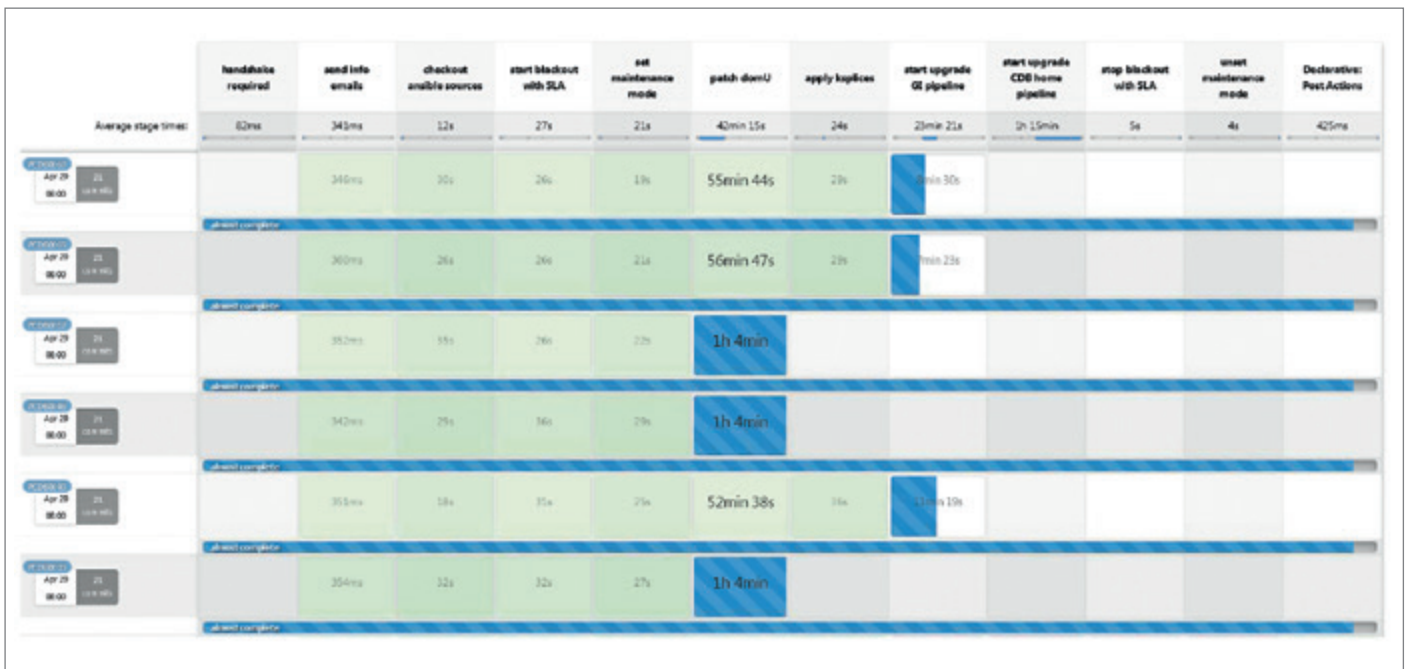


Abbildung 9: domU-voller Stapel-Patching-Rennen (Quelle: Pavel Tolstorukov)

gik, Ansible-Playbook-Aufruf und Jenkins-Umgebung sowie Input-Variables- Verwendung.

### Oracle-Exadata-Patching-Automatisierung

In diesem Abschnitt wird das Gesamtbild des Oracle-Exadata-Patchings mit der Hilfe von DevOps-Werkzeugen beschrieben.

Zunächst müssen alle Schritte mit Ansible Playbooks beschrieben werden, um jede Komponente des Exadata-Datenbank-Servers zu patchen. GitLab dient hierbei als Versionskontrolle und Kollaborations-System.

Listing 3 zeigt den Ausschnitt eines Playbooks, dass domU für das Patchen vorbereitet und den Patch Manager ausführt.

Als nächstes wird GitLab als Git-Repository-Quelle in Pipeline beschrieben (siehe Listing 4). Hierbei wird bei jeder Ausführung die neueste Playbook-Code-Version von GitLab bezogen.

Abschließend baut Jenkins eine Pipeline bestehend aus einzelnen Playbook-Aufrufen zusammen und führt diese auf den Target-Systemen aus.

Somit ist die Automatisierung des Patching-Prozesses erreicht. Das Patchen jeder Database-Server-Komponente (dom0, domU, GI, RDBMS und CDB) wird mit Jenkins gestartet. Der Ingenieur

gibt die benötigten Parameter (Target-System, Softwareversion oder Oracle Home Name usw.) ein und drückt die Build-Taste (siehe Abbildung 4). Der Rest wird automatisch ausgeführt.

In Abbildung 5 ist der Jenkins-Fortschritt für die Ausführung der Pipeline des Patchens der Grid-Infrastruktur abgebildet. Jeder Schritt ist die Ausführung eines Ansible Playbooks, welches die vordefinierten Aufgaben des Patchings erledigt.

Bei einem Fehler in der Ausführung eines Playbooks wird die laufende Pipeline unterbrochen und Jenkins führt einen entsprechenden Alerting Task aus, der in der Pipeline-Beschreibung definiert ist.

In diesem Fall muss ein Spezialist eingreifen und das Problem beheben. Dann kann die Pipeline ab dem unterbrochenen Schritt fortgesetzt werden.

Abbildung 6 zeigt ein Beispiel einer Alerting-Konfiguration in der Pipeline. Mittels eines Jenkins-E-Mail-Plugins wird der Status der Pipeline-Ausführung (Success oder Failure) an die angegebenen E-Mail-Adressen versendet.

### Erweiterung der Patching-Automatisierung

Damit ist das Patching der einzelnen Exadata-Datenbank-Server-Komponenten automatisiert. Für jede Komponente startet

der Ingenieur – in der vorgeschriebenen Reihenfolge – die jeweiligen Jenkins-Pipelines.

Dieser Prozess kann noch weiter optimiert werden.

### Pipelines-Verkettung

Da die Pipelines bei einem Fehler in der Ansible-Playbook-Ausführung stoppen, besteht grundsätzlich keine Gefahr darin, diese zu verketteten. Mit der Scripting Sprache DSL, in der die Pipelines beschrieben sind, können zudem auch weitere Pipelines aufgerufen werden. Diese Funktionalität ermöglicht es, mehrere Pipelines sequenziell oder parallel auszuführen, und eine komplexere größere Pipeline im Baukasten-Prinzip zusammenstellen. Damit ist es möglich, den Exadata-Datenbank-Server vollständig (von dom0 bis CDB) oder teilweise (z. B. GI und CDB) in einem Durchgang zu patchen. Die notwendigen Input-Parameter werden dabei einmalig zu Anfang spezifiziert.

In Abbildung 7 ist eine Jenkins GUI dargestellt, in der das vollständige domU-Patching, einschließlich GI-Upgrade und CDB-Patching, abgebildet ist.

Das Listing 5 zeigt den DSL-Code-Abschnitt für den Pipeline-Aufruf.

### Pipeline Schedule

Mit der Verkettung der Pipelines ist der größte Teil der vollständigen Automatisierung erledigt. Der Ingenieur muss nur

noch eine Pipeline starten, um das komplette Patching durchzuführen. Auch dieser Schritt lässt sich automatisieren.

Jenkins stellt hierfür ein Scheduling Framework zur Verfügung. Die Scheduling-Konfiguration ist ähnlich zu der Linux-crontab-Konfiguration. Man stellt Minute, Stunde, Tag, Monat, Jahr sowie den Pipeline-Namen mit den gewünschten Parametern in der Jenkins Option „Build Triggers“ ein. In *Abbildung 8* sieht man ein Scheduler-Beispiel für das vollständige domU-Patching.

Mit dem Jenkins Scheduling ist der letzte Schritt der vollständigen Automatisierung des Exadata-Database-Patchings erreicht. Der Zeitplan für die Ausführung der Patching-Pipelines kann frühzeitig und sorgfältig eingestellt werden. Während des Patching-Wartungsfensters startet der Patchingprozess ohne menschlichen Eingriff und läuft von allein ab. Sobald das Patching zu Ende ist oder ein Fehler aufgetreten ist, wird die verantwortliche Person über die in Jenkins konfigurierte Alerting-Funktion informiert.

Bei Bedarf kann man, wie in *Abbildung 9* dargestellt, das Pipeline-„Rennen“ beobachten. Hierfür gibt es ein Jenkins Dashboard, welches die einzelnen Automatisierungsschritte und die Targets zeigt. Das Beispiel zeigt ein vollständiges domU-Patching einschließlich GI und RDBMS für sechs Targets. Die Anzahl an parallel gepatchten Targets ist dabei nur durch die Jenkins-Konfiguration, beziehungsweise durch die Anzahl der verfügbaren Jenkins Workers, begrenzt.

## Fazit

Mit der vollen Automatisierung des Exadata-Patchings lässt sich viel Arbeitszeit sparen. Aber es gibt noch weitere Vorteile.

- Menschliche Fehler während des Patching-Prozesses sind fast ausgeschlossen, da der Mensch nicht an der Ausführung des Prozesses beteiligt ist (nur im Falle eines Issues während des Patchings).
- Die Ansible Playbooks und Jenkins Pipelines definieren standardisierte Bausteine, die nach dem Baukasten-Prinzip für den Patchingprozess eingesetzt werden können.

- Gleiche Ansible Playbooks und Jenkins Pipelines können für Dev, Test und Prod verwendet werden, womit der Lifecycle der Dev/Test/Prod Workflows ebenfalls standardisiert werden kann.
- Jede Patching-Pipeline-Ausführung ist reproduzierbar und die Ergebnisse nach entsprechenden Tests vorhersehbar.
- Jenkins speichert alle Logs der Pipeline-Ausführung sowie die dazu gehörigen Dateien, sodass der Patching-Prozess im Nachhinein nachvollziehbar ist.

Das sind einige der Vorteile, die die Patching-Automatisierung mit sich bringt.

Diese ist aber nicht nur auf Exadata begrenzt – jede Oracle-Umgebung kann automatisiert gepatcht werden.

Zudem ist die Patching-Automatisierung nur der Anfang. Automatisierung lässt sich auf zahlreiche weitere Systeme, Umgebungen und IT-Prozesse anwenden. Schon heute ist es ein unerlässliches Werkzeug für den Betrieb von modernen Datenplattformen geworden. In Zukunft wird es noch weiter an Relevanz gewinnen, um die Sicherheit, Stabilität und Leistung von IT-Systemen zu gewährleisten.

## Quelle

[1] <https://de.wikipedia.org/wiki/Ansible>

## Über den Autor

Pavel Tolstorukov arbeitet als Tech Architecture Associate Manager für Accenture Schweiz. Sein Einstieg in die IT-Welt begann mit einem Mathematik- und Informatik-Studium. Seit 2002 steht Oracle im Mittelpunkt seiner Tätigkeit. In den letzten Jahren war Pavel Tolstorukov hauptsächlich als Architekt für die Automatisierung von Oracle-Produkten bei verschiedenen Kunden im Einsatz. Er ist zertifizierter Experte in Oracle, RedHat und Jenkins.



Pavel Tolstorukov  
pavel.tolstorukov@accenture.com

# BEST OF DOAG ONLINE

Eine Auswahl der besten DOAG News News August/September 2023



## "Wir müssen vor der digitalen Transformation keine Angst haben. Im Gegenteil."

Der Kabarettist und Diplom-Physiker Vince Ebert ist im November Keynote Speaker der DOAG 2023 Konferenz + Ausstellung und der KI NAVIGATOR. Wir haben vorab mit ihm gesprochen.



## KI NAVIGATOR 2023: Highlights "KI & Technologie"

Wir werfen einen Blick auf den Schwerpunkt "KI & Technologie" der KI NAVIGATOR 2023 und stellen die ersten Highlights vor.



## DOAG-Podcast "Devs on Tape" feiert 40. Episode

Die Initiative der Development Community hat sich zu einer regelrechten Erfolgsgeschichte entwickelt und in der Welt der deutschsprachigen Tech-Podcasts etabliert.



## Thematische Öffnung der DOAG durch Satzungsänderung untermauert

Seit einigen Jahren verfolgen DOAG-Vorstand und -Delegierte eine Strategie zur thematischen Öffnung des Vereins.



## DOAG Datenbank Kolumne: 23c – Much better error messages?

Gerald Venzl zeigte in einer seiner Präsentationen über Neuerungen in Oracle 23c eine Folie mit dem Hinweis "Much better error messages".



## Oracle reagiert auf Red Hats Ankündigung, keine öffentlichen Quellen von RHEL mehr anzubieten

Nach vielen Unsicherheiten und offenen Fragen reagierten nun CIQ, Oracle und SUSE mit der Gründung einer neuen Open Enterprise Linux Association.



## Wir begrüßen unsere neuen Mitglieder

### Natürliche Mitglieder:

- André Buck
- Rudolf Konrad
- Serkan Arslan
- Hans Hahn
- Mark Oligslagers
- Jingquan Xie
- Jörg Danzenbächer
- Dr. Benjamin Linnik
- Armenoui Gurgenovna Pogosian
- Jan Kim Jørgensen

### Korporative Mitglieder:

- record-consult GmbH, Repräsentant: Marcel Pernak
- Transfact GmbH, Repräsentant: Ulrik Hoffmann
- Cleverbridge GmbH, Repräsentant: Edwin Top



## Termine

Oktober

10

13.10.2023

Oracle 23c DB Nest im praktischen Einsatz  
DB WebSession mit Stefan Oehrli  
Online

17.10.2023

Regionaltreffen Freiburg  
APEX aus Sicht eines langjährigen Forms-Entwicklers  
Freiburg

19.10.2023

Back-end Testing bei DB-Entwicklungsprojekten – Teil 2  
DevTalk mit Oliver Lemm und Samuel Nitsche  
Online

26.10.2023

Regionaltreffen NRW  
Thema wird zeitnah benannt  
Wuppertal

26.10.2023

Regionaltreffen Dresden  
"Database Experience Report Migration 19c and Outlook"  
Dresden

November

11

10.11.2023

Power Shell für den Oracle DBA  
DB WebSession mit Andreas Jordan  
Online

20. - 21.11.2023

European NetSuite User Days 2023  
im Rahmen der DOAG 2023 Konferenz  
+ Ausstellung im NürnbergConvention Center (NCC Ost)  
Nürnberg

21. - 24.11.2023

DOAG 2023 Konferenz + Ausstellung  
Die führende Anwender-Konferenz mit Fokus auf Oracle-Technologien im deutschsprachigen Raum.  
Nürnberg

22.11.2023

Regionaltreffen Rhein-Main  
Oracle 23c: Was gibt es neues aus den Bereichen Administration & Entwicklung  
Wiesbaden

22. - 23.11.2023

KI NAVIGATOR 2023  
Konferenz zur Praxis der KI in IT, Wirtschaft und Gesellschaft  
Nürnberg

April

04

09. - 11.04.2024

JavaLand 2024  
Zwei ereignisreiche Konferenztage mit anschließendem Schultag rund um das Java-Ökosystem  
Am Nürburgring, Nürburg

22. - 24.04.2024

APEX connect 2024  
Konferenz mit zahlreichen Vorträgen und Workshops zu den Themen APEX, JavaScript und PL/SQL  
Düsseldorf

## Impressum

Red Stack Magazin inkl. Business News wird gemeinsam herausgegeben von den Oracle-Anwendergruppen DOAG Deutsche ORACLE-Anwendergruppe e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, [www.doag.org](http://www.doag.org)), AOUG Austrian Oracle User Group (Österreich, Lassallestraße 7a, 1020 Wien, [www.aoug.at](http://www.aoug.at)) und SOUG Swiss Oracle User Group (Schweiz, Dornacherstraße 192, 4053 Basel, [www.soug.ch](http://www.soug.ch)).

Red Stack Magazin inkl. Business News ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin inkl. Business News wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Björn Bröhl. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führt einen konstruktiv-kritischen Dialog mit Oracle.

### Redaktion:

Sitz: DOAG Dienstleistungen GmbH  
(Anschrift s.o.)  
ViSdP: Fried Saacke  
Redaktionsleitung Red Stack Magazin:  
Martin Meyer  
Redaktionsleitung Business News:  
Marcos López  
Kontakt: [redaktion@doag.org](mailto:redaktion@doag.org)

Weitere Redakteure (in alphabetischer Reihenfolge): Martin Bach, Wolf Beckmann, Jörg Doppelreiter, Andreas Fritsch, Markus Geis, Gerald Hacke, Moritz Klein, Carolin Krützmann, Florian Lösch, Martin Meyer, Dr. Thomas Mützel von Schwartz, Chris Neubauer, Pavel Tolstorukov, Oliver Röniger, Michael Schmid, Marc Sewtz, Günther Stürner, Marc Wagner, Mathias Weber.

### Titel, Gestaltung und Satz:

Diana Tkach  
DOAG Dienstleistungen GmbH  
(Anschrift s.o.)

### Fotonachweis:

Titel: © freepik | [www.freepik.com](http://www.freepik.com)  
S. 6: © Mimi Thian | [www.unsplash.com](http://www.unsplash.com)  
S. 8: © Rescue | [www.pixabay.com](http://www.pixabay.com)  
S. 14: © Markusspiske | [www.pixabay.com](http://www.pixabay.com)  
S. 18: © Nextvoyage | [www.pixabay.com](http://www.pixabay.com)  
S. 26: © Tama66 | [www.pixabay.com](http://www.pixabay.com)  
S. 34: © Jplenio | [www.pixabay.com](http://www.pixabay.com)  
S. 44: © Broesis | [www.pixabay.com](http://www.pixabay.com)  
S. 46: © Pexels | [www.pixabay.com](http://www.pixabay.com)  
S. 54: © zaie | [www.freepik.com](http://www.freepik.com)  
S. 55: © kjpargeter | [www.freepik.com](http://www.freepik.com)  
Titel S. 56: © freepik | [www.freepik.com](http://www.freepik.com)  
S. 62: © 4045 | [www.freepik.com](http://www.freepik.com)  
S. 70: © redgreystock | [www.freepik.com](http://www.freepik.com)  
S. 74: @ Pexels | [www.pixabay.com](http://www.pixabay.com)  
S. 82: © 4047259 | [www.pixabay.com](http://www.pixabay.com)

S. 88: © analogicus | [www.pixabay.com](http://www.pixabay.com)  
S. 98: © Tama66 | [www.pixabay.com](http://www.pixabay.com)  
S. 109: © freepik | [www.freepik.com](http://www.freepik.com)

### Anzeigen:

[sponsoring@doag.org](mailto:sponsoring@doag.org)

### Mediadaten und Preise:

[www.doag.org/go/mediadaten](http://www.doag.org/go/mediadaten)

### Druck:

WIRmachenDRUCK GmbH,  
[www.wir-machen-druck.de](http://www.wir-machen-druck.de)

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags.

Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

## Inserentenverzeichnis

DOAG e.V. <a href="http://www.doag.org">www.doag.org</a>	<b>U 2, U 3, U 4</b>	DOAG e.V. <a href="http://www.doag.org">www.doag.org</a>	<b>S. 3, S. 7, S. 13, S. 43, S. 97,</b>
Promatis Gruppe <a href="http://www.promatis.de">www.promatis.de</a>	<b>S. 71</b>	Deutschen Bahn AG <a href="http://db.jobs/it-apex">db.jobs/it-apex</a>	<b>S. 61</b>
		SIGS DATACOM GmbH <a href="http://www.sigs-datacom.de">www.sigs-datacom.de</a>	<b>S. 17</b>

DOAG

DOAG<sup>2024</sup> Datenbank  
mit Exaday

---

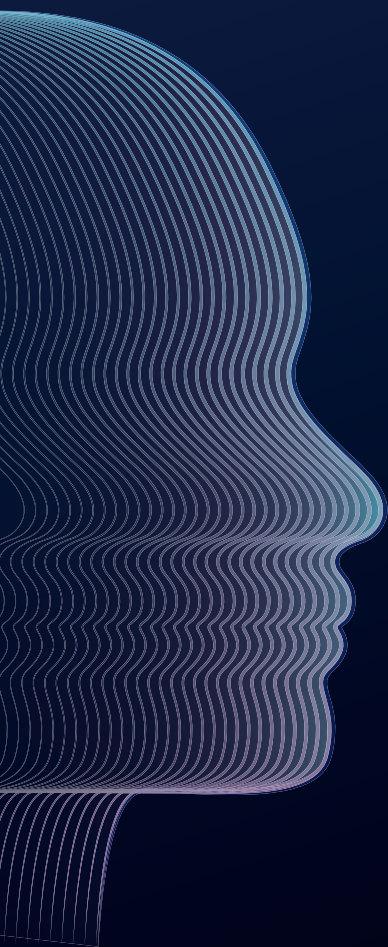
15. und 16. Mai in Düsseldorf

[datenbank.doag.org](https://datenbank.doag.org)



# KI Navigator 2023

Konferenz zur Praxis der KI  
in IT, Wirtschaft und Gesellschaft  
22. + 23. November in Nürnberg



## KI verstehen und einsetzen!

“Im Jahr 2023 erleben wir eine historische Zäsur. KI-Modelle mit beispiellosem Potenzial zur Problemlösung und Entscheidungsfindung markieren einen Wendepunkt. Auf der KI Navigator kann ich von führenden Experten lernen und mich von realen Anwendungsfällen inspirieren lassen, um mich optimal auf diese neue Ära vorzubereiten.“



**Oliver Szymanski**  
Chief Technical Architect,  
IT-Systemhaus der  
Bundesagentur für Arbeit

[ki-navigator.doag.org](https://ki-navigator.doag.org)



Veranstalter:

**DOAG**

**Heise Medien**

**de|ge|pol**