



DEVELOPMENT

Aus der Praxis

Wie kommt meine Oracle-Datenbank in die Oracle Cloud?



Im Interview

Carolin Krützmann und Kai Donato, Redaktion Podcast "Devs on Tape"

Datenbanken

Ein kritischer Blick auf PostgreSQL



18. – 21. JUNI



#CLOUDLAND2024





Niels de Bruijn Leitungskraft in der Development Community & Themenverantwortlicher APEX connect



Liebe Mitglieder, liebe Leserinnen und Leser,

Seit der Cloud World 2023 ist klar: Oracle setzt mit 23c auf Entwicklerthemen in der Datenbank. So beinhaltet das "App Simple"-Release beispielsweise das neue Feature "Duality Views". Es kombiniert die Vorteile von JSON-Dokumenten mit denen des relationalen Datenmodells. Die "Converged Database"-Strategie wird so konsequent fortgesetzt und dies kommt nicht zuletzt auch Oracle Application Express (APEX) zugute. Diese Low-Code-Plattform in der Datenbank verbreitet sich kontinuierlich, auch bei Oracle. So hat Larry Ellison angekündigt, dass seine Softwareprodukte zukünftig meist auf Basis von APEX entwickelt werden. Dazu werden komplexe Applikationen beim aufgekauften Unternehmen Cerner sukzessive auf APEX umgestellt. Und allein intern bei Oracle laufen Tausende von APEX-Applikationen bereits produktiv. Wer hätte dies vor 10 Jahren gedacht?

"Tue was Gutes und rede darüber", da kann Oracle mit Sicherheit noch dazu lernen. Zwar ist Oracle mit seiner Datenbank für Entwickler:innen super aufgestellt, das Image von Oracle ist jedoch angekratzt. Kurzfristiges Denken führt häufig zu Situationen, bei denen der Kunde nicht "abgeholt" wird und deswegen Oracle komplett verlässt. Eine ehrliche Beratung ist, was notwendig ist und daran mangelt es aktuell. So muss beispielsweise die Oracle-Datenbank nicht teuer sein: Eine (APEX)-Datenbank inklusive Support in der Oracle Cloud gibt es bereits ab +/- 115 EUR/Monat! Und wissen Kunden bereits über die Vorteile von EU Sovereign Cloud Bescheid?

Vielleicht ist es deswegen auch so wichtig, eine großartige Development Community in Deutschland zu haben. So beleuchten Kai Donato und Carolin Krützmann in ihrem Podcast "Devs on Tape" alle zwei Wochen ein Entwicklerthema. Aktuell gibt es bereits 58 Folgen! In dieser Ausgabe könnt ihr mehr darüber lesen, ebenso über die APEX connect, die in diesem Jahr zum 10. Mal stattfindet. Als Initiator und Programmleiter der Konferenz gebe ich einen Rückblick über 10 Jahre APEX connect. Diese und viele weitere Beiträge findet ihr in dieser Development-Ausgabe.

Viel Spaß beim Lesen wünscht euch

Mich de Brug

Niels de Bruijn



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





Einleitung

- Editorial
- Timeline
- "Wir möchten einen Mehrwert für alle schaffen und dies ist nur möglich, wenn man mit der Zeit geht." Interview mit Carolin Krützmann und Kai Donato

Development

- Requirements Engineering -... und wie das zu APEX passt Tobias Schweiker
- 20 Oracle Cloud für Anwendungsentwickler Marcel Uusvel-Amende
- 26 Konsistenz und Synchronisation in verteilten Systemen Dr. Stefan Koch
- 34 Oracle 23c für Entwickler Rainer Willems

APEX

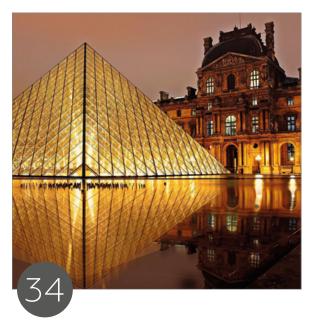
42 10 Jahre APEX connect! Niels de Bruijn



Requirements Engineering -... und wie das zu **APEX** passt



Oracle Cloud für Anwendungsentwickler



Oracle 23c für Entwickler

Cloud

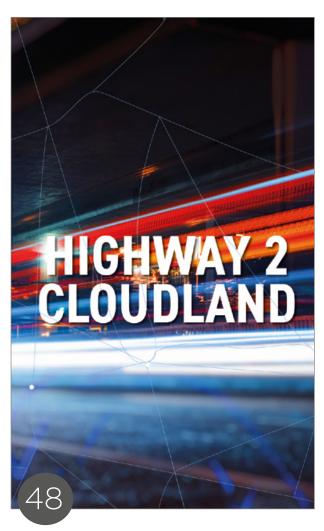
- 48 Highway to CloudLand die DOAG-Roadshow Marcos López
- 52 "I like to move it, move it"- oder: Wie kommt meine Oracle-Datenbank in die Oracle Cloud? Kai-Uwe Fischer

Datenbank

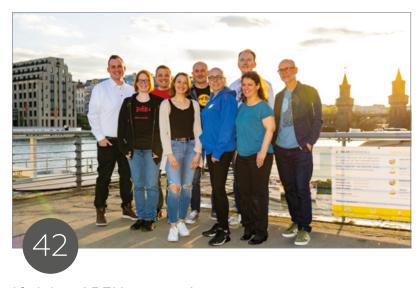
63 Ein kritischer Blick auf PostgreSQL Markus Flechtner

Forms

68 Nachlese zum DOAG Oracle-Forms-Thementag 2023 Frank Hoffmann



Highway to CloudLand - die DOAG-Roadshow



10 Jahre APEX connect!



"I like to move it, move it"- oder: Wie kommt meine Oracle-Datenbank in die Oracle Cloud?



Nachlese zum DOAG Oracle-Forms-Thementag 2023

Intern

- 77 Neue Mitglieder + Termine
- Impressum + Inserenten

News

- 57 Oracle Datenbanken Monthly News
- 67 Berliner Experimentseminare
- Best of DOAG Online

TIMELINE-

28. Februar 2024

Das Regionaltreffen Osnabrück/Bielefeld/Münster findet in Gütersloh statt. Die Vortragsthemen "Automatisierung im Oracle-Umfeld mit Ansible" und "Oracle-Datenbank 23c: Überblick über das kommende Long Term Release" stehen auf der Agenda.

8 März 2024

In der WebSession mit Christian Pfundtner erfahren die Teilnehmenden alles rund um das Thema "CIS Security Benchmark und GLOBAL_NAMES."

13 März 2024

Das Regionaltreffen Berlin Brandenburg findet in Berlin statt. Zwei Themen werden als Hands-on-Sessions angeboten: "SQL Empowerment in Oracle 23c: Leveraging Domains and New Features" und "Easy Text Search over Multiple Tables and Views with DBMS_SEARCH in 23c."

9. bis 11. April 2024

Die Jubiläumsausgabe der JavaLand findet vom 9. bis 11. April 2024 am Nürburgring statt. Zwei Tage Konferenz, ein Schulungstag, zehn Jahre JavaLand! Eine neue Erfahrung erwartet die Besucherinnen und Besucher an der diesmaligen neuen Location Nürburgring. Rund 160 Sessions mit Präsentationen jeden Levels, Bühnenpremieren unserer Newcomer, eine bunte Mischung aus Community-Aktivitäten und einige Überraschungen erwarten die Teilnehmenden im Motorsport-Mekka. Ein Adrenalinkick der besonderen Art erwartet die Teilnehmenden beim Driften, harten Bremsmanövern und anspruchsvollen Kurvenkombinationen beim Fahren der Elektro-Karts auf dem Nürburgring, die exklusiv an allen drei Konferenztagen zur Verfügung stehen. Des Weiteren können die legendären Kurven des Nürburgrings bei einem faszinierenden und informativen Trackwalk erkundet werden. Im Museum kann eine historische Ausstellung bestaunt werden, die Geschichten von mutigen Fahrerinnen und Fahrern sowie epischen Rennen erzählt.

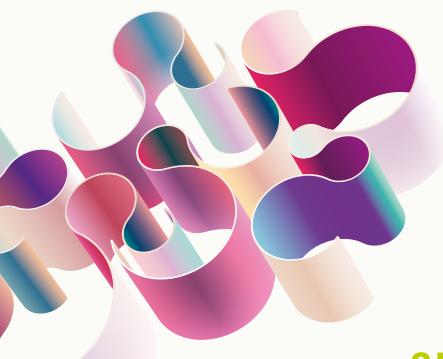
Top-Speaker Ralf Müller meinte zur 10. JavaLand "Die JavaLand ist wie ein großes Klassentreffen. Ich freue mich vor allem darauf, mit der Community ins Gespräch zu kommen und neue Ideen abseits des Mainstreams zu hören."





22. - 24.04.2024

VAN DER VALK AIRPORTHOTEL DÜSSELDORF





apex.doag.org



Carolin Krützmann und Kai Donato (Quelle: Connor McDonald)

"Wir möchten einen Mehrwert für alle schaffen und dies ist nur möglich, wenn man mit der Zeit geht."

Martin Meyer, Redaktionsleiter des Red Stack Magazin, sprach mit Carolin Krützmann und Kai Donato über die Idee, die Herausforderungen, Gäste und besondere Erfahrungen bei ihrem Podcast "Devs on Tape", der sich mit allen Themen rund um die professionelle Software-Entwicklung beschäftigt. Die Serie umfasst mittlerweile schon 58 Ausgaben.

Könnt ihr Euch unseren Lesern kurz vorstellen? Wer seid ihr und mit was beschäftigt ihr euch?

Kai: Mein Name ist Kai Donato, ich bin 33 Jahre alt und lebe mit meiner kleinen Familie in der Nähe von Düsseldorf. Als Fachbereichsleiter APEX & JavaScript betreue ich ein Team von engagierten Mitarbeitern, die sich, wie der Name schon sagt, mit APEX und JavaScript beschäftigen. Seit mittlerweile 10 Jahren bin ich im Umfeld der DOAG unterwegs und seit einigen Jahren auch als Themenverantwortlicher für Java-Script aktiv. Ich sitze im Programm- und Organisationsteam der APEX-connect-Konferenz und bin als Speaker auf diversen Oracle-Konferenzen unterwegs. Seit mittlerweile 50 Folgen betreibe ich zusammen mit Caro den Podcast¹⁾ "Devs on Tape" und bin sehr froh darüber, dieses Format in Angriff genommen zu haben.

Caro: Ich bin Carolin Krützmann – oder auch einfach nur Caro - und lebe in der Nähe von Hamburg. Ich bin als selbstständige IT-Beraterin hauptsächlich in der Datenbank- und Oracle-APEX-Entwicklung unterwegs. Bis letztes Jahr war ich Themenverantwortliche für APEX bei der DOAG, plane APEX-Meetups und Regionaltreffen in Hamburg und versuche mein Wissen oder Themen, die interessant sein könnten, mit anderen zu teilen. Um einen der Wege geht es hier heute: Unseren Podcast "Devs on Tape".:)

Wie seid Ihr darauf gekommen, einen Podcast zu machen?

Kai: Als aktive Mitglieder in der Development Community der DOAG sind wir stehts auf der Suche nach neuen Möglichkeiten, um existierende Mitglieder und solche, die es potenziell werden wollen, mit interessanten Inhalten zu erreichen. Wir möchten einen Mehrwert für alle schaffen und dies ist nur möglich, wenn man mit der Zeit geht. Dazu gehört es auch, sich nicht auf das bestehende und alt-bewährte zu verlassen, sondern auch neue Wege zu gehen. Ein Podcast schien uns hierfür bestens geeignet zu sein. Es ist der erste Podcast der DOAG und somit natürlich mit viel Pionierarbeit verbunden.

Caro: Was Kai noch nicht erzählt hat, ist die Kreativität, mit der wir bereits vor der Gründung des Podcast an Ideen gefeilt haben, um gemeinsam Projekte angehen zu können. Aus einem dieser Impulse heraus wurde in etwa mit dem Dialog "Hey, lass uns einen Podcast machen!" - "Oh, ja, das ist eine super Idee", der Entschluss gefasst, den Podcast anzugehen.

Habt Ihr Vorbilder? Lieblingspodcasts? Habt Ihr eine journalistische Vorerfahrung oder früher schon Erfahrung mit Podcasts oder sozialen Medien sammeln können?

Kai: Ich habe eine Liste von circa 20 Podcasts, die ich mal mehr, mal weniger höre. Teilweise höre ich neue Episoden direkt am Tag der Veröffentlichung.

Caro: Tatsächlich sind Podcasts eigentlich gar nicht mein Medium. Ich bin eher visuell unterwegs und gucke mir lieber You-Tube-Videos an. Wenn ich mal Podcasts höre, dann drehen sich diese meistens eher um Themen, über die ich selten Videos gucke oder Bücher lese (zum Beispiel Meditation) oder ich höre Musik.

Wie findet Ihr die Interview-Partner? Wie wird ein Interview geplant?

Kai: Durch unsere Aktivitäten bei der DOAG und unserem beruflichen Netzwerk haben wir relativ schnell Personen identifizieren können, die wir gerne in unseren Podcast einladen wollten. Dazu kamen natürlich auch noch Empfehlungen und Personen, die wir kennenlernen durften. Eines ist uns aber schnell klar geworden: Fragen kostet nichts und gute Gelegenheiten, lassen wir nicht liegen.

Wie bekommt Ihr die Ideen bezüglich der Fragen?

Caro: Neben den Kategorien, die mögliche Parallelen zwischen den unterschiedlichen Gästen aufzeigen, bereiten wir uns auf jeden Gast individuell vor und stellen in erster Linie die Fragen, die uns persönlich interessieren.

Wir gehen das Sammeln und die Strukturierung recht formlos an. Meistens haben wir für jeden Gast ein Miro-Board, auf dem wir einige Informationen sammeln und uns die Fragen dazu notieren. Da kommen dann selbst während des Interviews auch mal spontan welche dazu.

Trotz der Vorbereitung legen wir Wert auf ein fließendes und persönliches Gespräch. Da passiert es auch nicht selten, dass wir nur wenige von unseren vorab notierten Fragen stellen und eher spontan auf die eine oder andere Frage kommen.

Welche Gäste sind euch am meisten im Gedächtnis geblieben? Gibt es schwierige Interview-Partner?

Kai: Für uns ist jede Episode und jeder Gast etwas Besonderes. Natürlich stechen Interview-Partner wie Andy Mendelsohn und Stefan Latuski etwas hervor, da der Bekanntheitsgrad uns besonders herausfordert. Man merkt dem einen oder anderen Interview-Partner an, dass sie/er bereits Erfahrungen mit einem solchen Format hat und das macht es uns etwas leichter. "Der Gesprächsanteil variiert je nachdem, ob unser Interview-Partner eher schüchtern und introvertiert ist oder, ob er das Herz auf der Zunge trägt."

Wieviel Zeitaufwand steckt hinter einem kompletten Podcast? Was waren die größten technischen Probleme?

Kai: Wir teilen uns den Aufwand für den Podcast bestmöglich auf und so übernimmt Caro neben der Aufnahme selbst auch noch den kompletten Schnitt jeder Folge. Ich kümmere mich in den meisten Fällen um die Technik und schreibe die Shownotes sowie die Postings auf den Social-Media-Accounts.

Die meisten technischen Probleme, die wir im Laufe der Zeit hatten, ließen sich kurzfristig lösen. So hatten wir bereits Mikrofonaussetzer bei Remote-Aufnahmen und nervige Hintergrundgeräusche, die Caro dann im Schnitt beseitigen konnte. Außerdem ist es auch ein technisches Problem, wenn man Teile der Technik schlichtweg zuhause vergisst. :)

Was motiviert Euch beim Podcasten am meisten?

Caro: Ein Nachteil am Podcasten ist, dass man wenig Feedback erhält. Umso mehr freut und motiviert es uns, wenn wir auf Konferenzen angesprochen werden und positive Kritik erhalten. Die tollen und spannenden Gespräche und die witzigen Momente "Off-mic" werden uns lange erhalten bleiben. Wir sind wirklich froh, den Podcast gestartet zu haben.

Welche Podcasts wurden am häufigsten aufgerufen?

Caro: Die bereits genannten Episoden mit Andy Mendelsohn und Stefan Latuski wurden bis dato am meisten gehört und darüber hinaus erreichen unsere englischsprachigen Episoden natürlich ein größeres Publikum.

Wen hättet Ihr gerne mal im Podcast, aber seid bisher nicht herangekommen?

Kai: Basierend auf unserem bisherigen Line-Up wäre natürlich Larry Ellison ein interessanter Interview-Partner. Auch wenn dies sehr unwahrscheinlich ist, haben wir viele interessante Gesprächspartner in der Pipeline. Wir freuen uns selbstverständlich auf jeden Einzelnen.

Caro: Eine ganz besondere Folge für Kai wäre sicherlich noch mit Sascha Lobo ;) (kleiner Insider).

Welche Trends seht ihr beim Thema Development?

Kai: Wir haben es bereits in mehreren Episoden thematisiert - KI-getriebene Entwicklung ist definitiv auf dem Vormarsch und auch die neuen Features, die uns in der Oracle-Datenbank erwarten.

Jede Themenwelt hat ihre eigenen Ausprägungen von Trends (neben den offensichtlichen wie KI natürlich). Daher ist es auch immer sehr spannend zu hören, was die Menschen rund um die Entwicklung sonst so bewegt. Ein Thema, über das wir beispielsweise auch schon viel gesprochen haben, ist Code-Qualität.

Zum Podcast

[1] https://devsontape.podigee.io/



CAROLIN KRÜTZMANN

Carolin Krützmann, IT-Consultant, DOAG Regionale Repräsentantin Hamburg, Oracle ACE Pro ist als selbstständige IT-Beraterin hauptsächlich in der Datenbank- und Oracle-APEX-Entwicklung unterwegs. Sie plant APEX-Meetups und Regionaltreffen in Hamburg und versucht ihr Wissen oder Themen, die interessant sein könnten, mit anderen zu teilen.



KAI DONATO

Kai Donato aus Düsseldorf leitet als Department Manager APEX & JavaScript ein Team von engagierten Mitarbeitern bei der Hyand Group GmbH, die sich mit APEX und Java-Script beschäftigen. Seit mittlerweile 10 Jahren ist er im Umfeld der DOAG unterwegs und seit einigen Jahren auch als Themenverantwortlicher für JavaScript aktiv. Er ist im Programm- und Organisationsteam der APEX-connect-Konferenz und als Speaker auf diversen Oracle-Konferenzen zu erleben.



Werden Sie DOAG-Mitglied!

"Gemeinsame Interessen gemeinsam vertreten"

- + attraktive Rabatte für Mitglieder
- + kostenfreier Bezug der Zeitschriften

Red Stack Magazin und Java aktuell



Requirements Engineering -... und wie das zu APEX passt

Tobias Schweiker, Zweckverband Bodensee-Wasserversorgung

Gut erfasste und dokumentierte Anforderungen sind unerlässlich für Anwendungen, die die Anwender bei ihrer Arbeit gut unterstützen. Deshalb sind Software- und Anwendungsentwickler auf gutes Requirements Engineering angewiesen.

"Aber bei APEX geht es doch "auch ohne." – Wirklich?

Nach einem Überblick über Requirements Engineering, gehe ich auf die Frage ein, wie man die prototypische Vorgehensweise in der APEX-Entwicklung mit Requirements Engineering zusammenbringen kann.

Was ist Requirements Engineering?

Die meisten in der IT, besonders in der Anwendungsentwicklung, kennen den Begriff und haben so ihre Vorstellungen davon. Aber meinen alle das gleiche?

Eine allgemein anerkannte Definition ist die des IREB (International Requirements Engineering Board) [1]:

Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:

- Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
- Die Wünsche und Bedürfnisse der Stakeholder zu verstehen und zu dokumentieren.
- Die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den

Wünschen und Bedürfnissen der Stakeholder entspricht.

Nun bedeutet diese Definition nicht, dass es das eine Requirements-Engineering-Vorgehen gibt. Systematisches Vorgehen bedeutet auch, dass zu Beginn eines Projektes alle Beteiligten festlegen, welche Methoden wie genutzt werden sollen. Diese werden dann diszipliniert angewandt.

Aus der Definition des IREB ergeben sich zwei Zwecke des RE: Erstens die Spezifikation der Anforderungen. Dazu gehören das Ermitteln, Erfassen, Konsolidieren und Festschreiben der Anforderungen. Dies wird auch Anforderungserhebung genannt. Der zweite Zweck ist das Management der Anforderungen, bei dem die Anforderungen dokumentiert und verwaltet werden.

Wie funktioniert RE?

Anhand einiger Schlaglichter stelle ich nun vor, wie man Anforderungen erheben und verwalten kann. Die Aufstellung erhebt keinen Anspruch auf Vollständigkeit.

Anforderungserhebung

Gemäß obiger Definition, ist die Anforderungserhebung eine der beiden wesentlichen Tätigkeiten im Requirements Engineering. Hier werden die Anforderungen von Anwendern und anderen Stakeholdern erfragt und konsolidiert. Andere Stakeholder sind Personen oder Organisationen, die von dem Produkt betroffen sind oder die Anforderungen haben, die nicht direkt die Anwendung des Produkts betreffen. Dies können zum Beispiel Kunden und Firmenleitungen, aber auch Gesetze (zum Beispiel DSGVO) oder Normen sein.

Im Wesentlichen gibt es zwei Möglichkeiten, Anforderungen zu erheben und zu notieren: graphisch und textuell.

Graphische Modellierung von Anforderungen

Ein Bild sagt mehr als tausend Worte. Das gilt auch für das Requirements Engineering, weshalb verschiedene graphische Modellierungsdarstellungen entwickelt wurden. Die bekanntesten sind die UML, die Unified Modelling Languge (siehe Abbildung 1), und die BPMN, die Business Process Modelling Notation. Mit den verschiedenen Diagrammtypen können verschiedene Aspekte der zu entwickelnden Software graphisch dargestellt werden. Die für mich wichtigsten Diagrammtypen sind das Klassendiagramm, das Aktivitätsdiagramm, das Sequenzdiagramm und die Zustandsmaschine. Mit dem Klassendiagramm werden Objekte, ihre Attribute und Beziehungen modelliert. Das Aktivitätsdiagramm kann Abläufe wie beispielsweise Workflows darstellen. Im Sequenzdiagramm wird das zeitliche Zusammenspiel der Aktivitäten verschiedener Objekte, zum Beispiel von Soft-

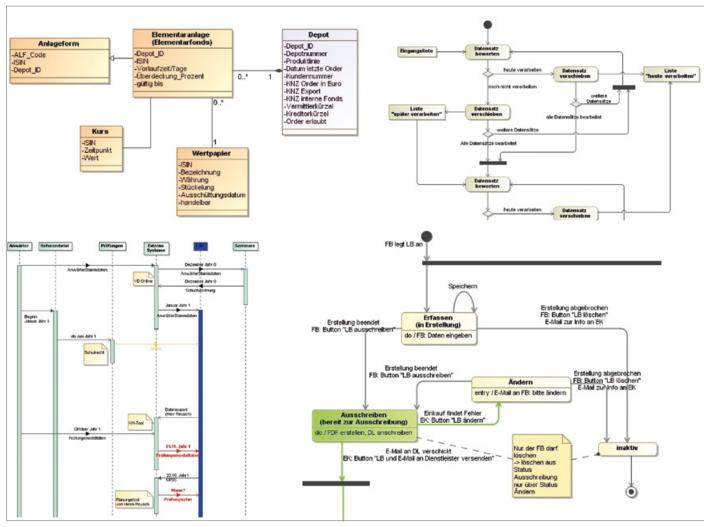


Abbildung 1: Beispiele für UML-Diagramme (Quelle: Tobias Schweiker)

waremodulen, aufgezeichnet. Mit dem Zustandsautomat werden Zustände eines Systems und deren Übergänge mit ihren Bedingungen modelliert.

Anforderungsebenen und Bedarfe

Bevor wir zu den textuellen Formulierungen von Anforderungen kommen, braucht es noch zwei Vorbemerkungen:

- In vielen Projekten hat sich herausgestellt, dass es sinnvoll ist, vor der Anforderungsdefinition Bedarfe zu formulieren. Ein Bedarf beschreibt aus der Sicht eines Anwenders eine Funktion des zu entwickelnden Systems, die ihm einen bestimmten Nutzen bietet. Manchmal wird ein Bedarf auch als User Story bezeichnet.
- Es gibt nicht "die" Anforderungen an ein zu entwickelndes System. Die An-

forderungen können auf verschiedenen Ebenen erhoben werden. Die Anforderungen der nachfolgenden Ebenen werden aus den darüber liegenden hergeleitet.

Die Anwenderanforderungen beschreiben die Funktionen, die vom Anwender benötigt werden, um einen Zweck zur Nutzung der Anwendung zu erfüllen (siehe Abbildung 2).

Systemanforderungen beschreiben technische Funktionen, die für den Betrieb der Software nötig sind. Dies können Anforderungen an die Systemarchitektur, Leistungsfähigkeit von Computern oder an Schnittstellen sein. Nicht für alle Anwenderanforderungen gibt es Systemanforderungen.

Aus den Anwender- und Systemanforderungen wird die Software-Architektur hergeleitet. Hier kann festgelegt werden, wie die Anforderungen auf Softwaremodule aufgeteilt werden.

Die Softwareanforderungen beschreiben, wie die Anwenderanforderungen implementiert werden sollen. Oft gibt es zu einer Anwenderanforderung mehrere Softwareanforderungen. Manchmal erfüllen Softwareanforderungen Teile mehrerer Anwenderanforderungen.

Im Gegensatz zu stringenteren Vorgehensmodellen, wie beispielsweise dem V-Modell, habe ich festgestellt, dass es sinnvoll ist, nicht für alle Anwenderanforderungen alle darunter liegenden Anforderungen herzuleiten. Manche Anwenderanforderungen haben keinen Einfluss auf das System oder die Softwarearchitektur.

Textuelle Beschreibung von Anforderungen

Bedarfe und Anforderungen können grundsätzlich frei formuliert werden. Dann besteht aber die Gefahr, dass sie missverständlich oder unklar sind. Da-

Anforderungsebenen in der Software-Entwicklung Bedarfe Anwenderanforderung Systemanforderung Softwarearchitektur

Abbildung 2: Die Anforderungsebenen (Quelle: © Tagueri AG)

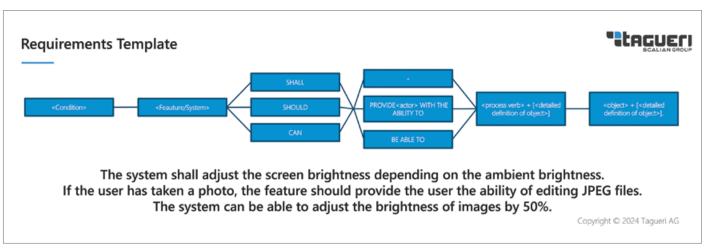


Abbildung 3: Satzvorlage für Anforderungen (Quelle: © Tagueri AG)

Softwareanforderung

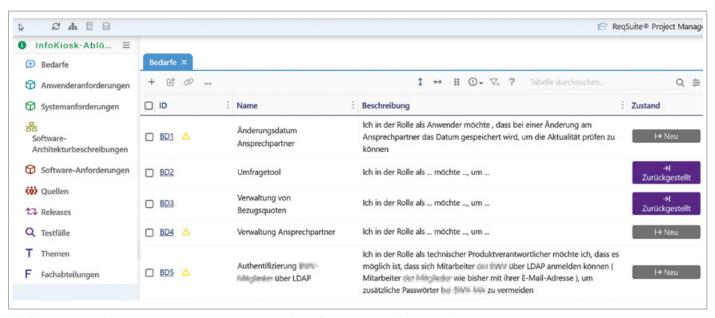


Abbildung 4: Beispiel für ein Requirements Management Tool (Quelle © OSSENO Software GmbH)

mit Anforderungen eindeutig und korrekt werden, gibt es Vorlagen für die Formulierung. Diese werden auch Satzschablonen oder Templates genannt.

Die Bedarfsvorlage sieht so aus:

Ich in der Rolle als ... möchte ... um ...

Beispiel: Als Autofahrer möchte ich die Geschwindigkeit in der Windschutzscheibe sehen, um nicht vom Verkehr abgelenkt zu sein, wenn ich wissen möchte, wie schnell ich fahre.

Die Anforderungsvorlage ist in *Abbildung 3* ausführlicher dargestellt.

Die deutschen Versionen lauten:

- <Das System> muss/soll/kann <Objekt> <Prozesswort/Verb>.
- <Das System> muss/soll/kann fähig sein/<dem Akteur> die Möglichkeit bieten, <Objekt> <Prozesswort/Verb>.
- <Bedingung> muss/soll/kann <Das System> <Objekt> <Prozesswort/ Verb>.
- <Bedingung> muss/soll/kann <Das System> fähig sein/<dem Akteur> die Möglichkeit bieten, <Objekt> <Prozesswort/Verb>.

Glossar

Viele Begriffe haben in unterschiedlichen Kontexten unterschiedliche Bedeutungen. Deshalb ist es sinnvoll, ein Glossar zu erstellen. Dieses kann eine tabellarische Liste sein, in der die Begriffe für den Kontext der Entwicklung eindeutig beschrieben sind. Eventuell kann es sinnvoll sein, zu ergänzen, was der Begriff im Kontext nicht bedeutet.

Eine andere oder die Liste ergänzende Möglichkeit, Begriffe zu definieren, ist das Klassendiagramm der UML. Neben der Beschreibung kann man so verschiedene Beziehungen der Begriffe graphisch abbilden.

Anforderungsmanagement

Die zweite wesentliche Tätigkeit im Requirements Engineering ist, gemäß der oben genannten Definition, die Dokumentation und Verwaltung der Anforderungen.

Wurde dies früher mit Word und Excel gemacht, gibt es mittlerweile verschiedene Softwaretools für die Dokumentation und Verwaltung von Anforderungen auf dem Markt. Mit diesen werden die Anforderungen nicht nur aufgeschrieben, sie unterstützen den gesamten Prozess des Requirements Engineerings. Zu Bedarfen und Anforderungen können die Satzvorlagen hinterlegt werden und weitere Objekte im Zusammenhang mit den Anforderungen können definiert und gespeichert werden. Damit kann man beispielsweise die Herkunft der Anforderungen dokumentieren oder Anforderungen gruppieren (siehe Abbildung 4).

In manchen Tools können verschiedene mögliche Beziehungen zwischen den

Objekten festgelegt werden. Mit diesen Beziehungen können die Anforderungen detailliert dokumentiert werden (siehe Abbildung 5).

... und warum das Ganze?

In meinem Artikel "Requirements Engineering – wieso, weshalb, warum?" im Red Stack Magazin 01/2021 [2] habe ich "die Benefits eines guten Requirements Engineerings auf[gezeigt] und Tipps und Argumente für einen frühzeitigen und erfolgreichen Einsatz von Anforderungserhebung, -analyse und -verwaltung" gegeben. Deshalb möchte ich hier nur kurz das Fazit dieses Artikels zitieren:

"Gutes Requirements Engineering unterstützt alle Projektbeteiligten bei ihren Aufgaben im Projekt und verbessert die Kommunikation zwischen ihnen. Es hilft, die Anforderungen eindeutig zu dokumentieren, das Projekt zu strukturieren und Risiken zu erkennen. Damit können Kosten gespart und der Gewinn erhöht werden [2]."

... und wie passt das zu APEX?

"Wir haben APEX, da brauchen wir kein Anforderungsmanagement. Wir entwickeln agil mit Prototypen", so höre ich es oft von APEX-Entwicklern. Doch sie verkennen, dass sie auch nur

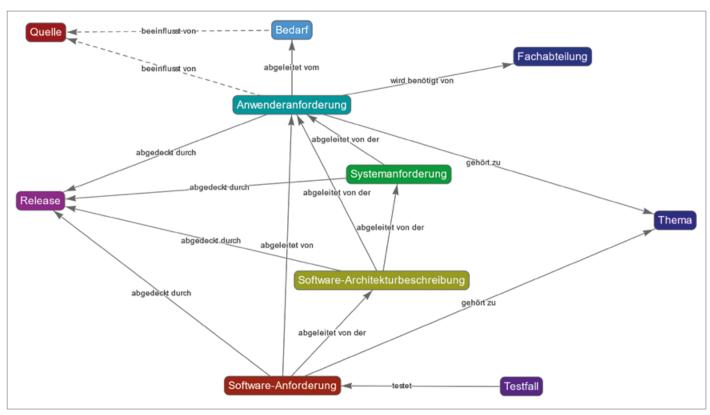


Abbildung 5: Objektbeziehungen im Requirements Management Tool (Quelle © OSSENO Software GmbH)

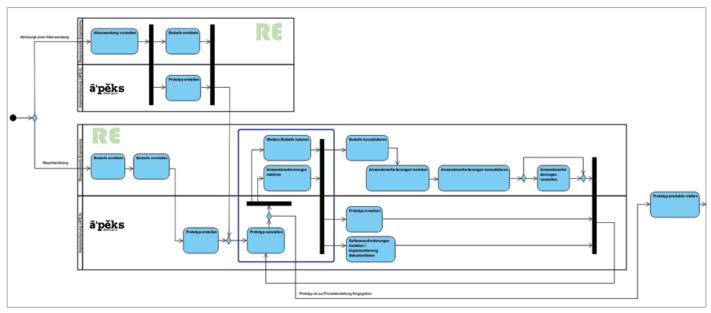


Abbildung 6: Ablauf Requirements Engineering für APEX-Anwendungsentwicklung (Quelle: Tobias Schweiker)

dann gute Anwendungen entwickeln können, wenn sie wissen, was die Anwender wirklich brauchen. Womit wir wieder beim Requriments Engineering wären. Um zu verstehen, was es für APEX-Entwickler schwierig macht, sich auf das Requirements Engineering einzulassen, müssen wir uns die scheinbar unterschiedlichen Ansätze anschauen.

Mit APEX kann man schnell und einfach Prototypen einer Anwendung erstellen. Oft braucht es dazu nur wenige Erklärungen der Wünsche der Anwender. Anhand dieser Prototypen wird dann mit den Anwendern überlegt, ob das, was sich die Anwender gewünscht haben, umgesetzt ist. Daraus ergeben sich neue Wünsche an die Anwendung, die im nächsten Prototyp implemen-

tiert werden. Dieser Entwicklungszyklus von Prototyp und neuen Wünschen wird mehrfach wiederholt. Oft erhält man damit sehr schnell ein Minimum Viable Product (MVP), das bereits von den Anwendern genutzt werden kann. Dieses wird dann entsprechend bis zur fertigen Anwendung weiterentwickelt.

Das klassische Requirements Engineering, insbesondere wenn es beispielsweise im Wasserfall- oder V-Modell eingesetzt wird, ermittelt erst alle Anforderungen an das Produkt und danach wird mit der Umsetzung begonnen.

APEX hat damit den Vorteil, dass schneller und früher entwickelt werden kann. Dabei können sich die Anwender weiterhin Wünsche überlegen, während der nächste Prototyp entwickelt wird. Wenn man aber die Anforderungen nicht richtig dokumentiert, kann es zu Unstimmigkeiten kommen, ob eine Anforderung tatsächlich so umgesetzt wurde, wie die Anwender es sich gewünscht haben. Außerdem besteht grundsätzlich die Gefahr, dass die Dokumentation der Funktionen der Anwendung vernachlässigt wird.

Dagegen gestaltet sich beim klassischen Requirements Engineering der Austausch der Entwickler und der Anwender über die Anforderungen konkreter und zielgerichteter, was zu einem besseren gemeinsamen Verständnis der Anforderungen führt. Weiterhin sind die Anforderungen gleich dokumentiert. Allerdings ist es schwieriger, nach der vollständigen Anforderungsdefinition noch Änderungen oder Ergänzungen an den Anforderungen vorzunehmen.

In einem aktuellen Projekt zur Neuprogrammierung vorhandener Anwendungen sind diese beiden Philosophien aufeinandergetroffen. APEX-Entwickler waren von der Detailierung und Atomisierung der Anforderungen überrascht. Sie hatten mit der Alt-Anwendung eine Vorlage, anhand der sie schnell einen Prototyp erstellen konnten. Die Requirements-Ingenieure wussten nicht, dass man so schnell zu einem brauchbaren Ergebnis kommen kann. Aus meiner langjährigen Projekterfahrung kenne ich sowohl das Requirements Engineering als auch das Programmieren mit APEX. Ich beschäftige mich schon länger damit, einen Weg zu finden, die Vorteile beider Vorgehensweisen zu verbinden. Hier entstand nun die Möglichkeit, zusammen mit den beiden Gruppen ein für beide Seiten sinnvolles Vorgehen zu finden. Angelehnt an agile Methoden haben wir für uns ein passendes Vorgehensmodell entworfen, das ich als UML-Aktivitätsdiagramm notiert habe (siehe Abbildung 6).

Mit diesem Vorgehen können gleichzeitig Anforderungen definiert und Prototypen entwickelt werden. Nach der Vorstellung der Funktionen durch die Anwender der Alt-Anwendung beginnt das Entwicklungsteam mit der Erstellung des ersten Prototyps. Gleichzeitig ermitteln die Requirements-Ingenieure mit den Anwendern die Bedarfe aufgrund der Alt-Anwendung. Bei einer Neuentwicklung müssen erst die Bedarfe der Anwender aufgeschrieben und dem Entwicklungsteam vorgestellt werden, das dann einen Prototyp entwickelt. In einer gemeinsamen Besprechung, in Abbildung 6 mit dem blauen Rahmen gekennzeichnet, wird dann der Prototyp vorgestellt und mit den Bedarfen der Anwender verglichen. Die verbesserten und neu gefundenen Bedarfe werden notiert, ebenso Anwenderanforderungen, die bei der Besprechung des Prototyps festgestellt werden. Anhand dieser Bedarfe und Anforderungen erstellt das Entwicklungsteam den nächsten Prototyp. Dabei sind auch, soweit erforderlich, die Softwareanforderungen aus den Anwenderanforderungen und Bedarfen hergeleitet und die Implementierung dokumentiert. Die Anwender verfeinern und konsolidieren die festgestellten Bedarfe und Anforderungen. Wenn beides abgeschlossen ist, findet das nächste Treffen zur Vorstellung und Besprechung des Prototyps statt. Dabei werden auch neue Anwenderanforderungen vorgestellt, die während der Besprechung des Prototyps festgestellt wurden. Wenn in der Besprechung festgestellt wird, dass die Funktionen des Prototyps nutzbar sind, kann er zur Produktivstellung freigegeben werden.

Mit dieser Arbeitsweise verbinden wir die Vorteile beider Vorgehensweisen. Die Requirements-Ingenieure sehen den Vorteil des Rapid Prototyping und wollen dies mit der konsolidierten Beschreibung der Anforderungen unterstützen. Die APEX-Entwickler verstehen die Notwendigkeit, Anforderungen gut zu dokumentieren und bringen ihre Erfahrungen aus der Prototypenentwicklung in das Anforderungsmanagement ein. Mit dieser neuen Vorgehensweise und dem gemeinsamen Verständnis freuen wir uns auf ein erfolgreiches Projekt.

Danksagung;

Mitarbeit: Margrith Kruip, Tagueri AG

Quellen

- Klaus Pohl, Chris Rupp (3. Auflage, 2011): Basiswissen Requirements Engineering – Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineer-ing Foundation Level. dpunkt.verlag.
- Red Stack Magazin 01/2021: Tobias Schweiker, Requirements Engineering wieso, weshalb, warum?

Über den Autor

Tobias Schweiker ist Datenbank-Anwendungsentwickler und zertifizierter Requirements Engineer (Certified Professional for Requirments Engineering (CPRE) Advanced Level). In vielen Projekten für verschiedene Kunden verschiedener Branchen konnte er vielfältige Erfahrungen mit dem Requirements Engineering und mit agilen Vorgehensweisen, wie beispielsweise APEX, sammeln. Bei der Bodensee-Wasserversorgung nutzt er als Applikationsmanager das Requirements Engineering für Software-Beschaffung und Anwendungsentwicklung. In der Development Community der DOAG engagiert er sich als Themenverantwortlicher für den Bereich "Requirements Engineering" und in der Initiative "Softskills und Methoden".



Tobias Schweiker tobias.schweiker@bodenseewasserversorgung.de



on demand

APEX 2023 VERPASST?

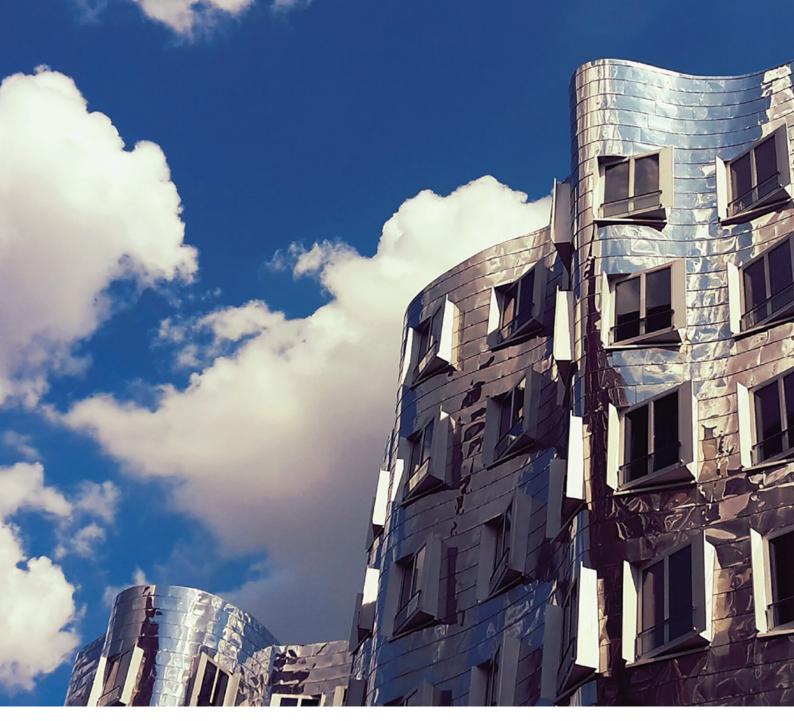
Jetzt On-demand-Ticket buchen und Vortragsaufzeichnungen anschauen!



Alle Angebote im On-demand-Ticket-Shop

apex.doag.org





Oracle Cloud für Anwendungsentwickler

Marcel Uusvel-Amende, Oracle Deutschland

Die Aussicht auf eine flexible, effiziente und skalierbare IT zu reduzierten Kosten führt Unternehmen in die Cloud. Die Oracle-Datenbank mit ihren vielfältigen Varianten und Betriebsmodellen ist dabei ein wirkungsvoller und auch praktikabler Einstieg. Während man beim "Cloud@Customer"-Modell zwar das Betriebsmodell, nicht jedoch die lokale Anbindung ändert, hebt ein Übergang in die Public Cloud zwar größere Potentiale, er verändert aber auch die Gravitation der Systemlandschaft. Die Masse einer Datenbank in der Cloud zieht zwangsläufig die umgebenden Applikationen an, weshalb eine Betrachtung der Möglichkeiten und Alternativen, die eigenen Anwendungen in der Cloud zu betreiben, lohnend erscheint.



Cloud-Native-Entwicklung

Das Cloud Computing fußt auf Paradigmen, die einen Technologieschub ausgelöst und die Art und Weise, wie in der Cloud und für die Cloud entwickelt wird, grundlegend verändert haben:

- Agilität und Flexibilität: Entwickler haben sich von physikalischer Hardware gelöst und arbeiten in virtuellen Umgebungen und mit vorgefertigten Diensten, was die Bereitstellung von Anwendungen und neuen Anwendungsversionen beschleunigt und Markteinführungszeiten verkürzt.
- Skalierbarkeit, Ressourcen- und Kostenoptimierung: Cloud-Plattformen bieten gefühlt unbegrenzte, elastische Ressourcen, die sich - auch au-

tomatisch - an die Anforderungen der Anwendung anpassen können. Dies ermöglicht, mit steigenden Lasten umzugehen, ohne dass es zu Engpässen kommt. Die Abrechnung nach tatsächlich genutzten Ressourcen in einem Pay-as-you-go-Modell ist kosteneffizient und reduziert die Kapitalbindung.

Entkopplung, Modularisierung und Containerisierung: Anwendungen werden in eigenständige, voneinander unabhängige Microservices aufgeteilt, die in Containern ausgeführt werden. Das einzelne Modul ist weniger komplex als ein Monolith, unabhängig test- und ausführbar und kann die jeweils am besten geeigneten Sprachen und Technologien nutzen. Oft geht dies Hand in Hand mit einem iterativen, kollaborativen und agilen Entwicklungsansatz, integrierten Entwicklungsumgebungen (Continuous Integration, CI) und automatisierter Bereitstellung (Continuous Delivery, CD), was in Summe für kürzere Entwicklungszyklen und qualitativ hochwertigere Software sorgt.

Oracle Cloud Infrastruktur (OCI) für Entwickler

Als Platinum-Mitglied der Cloud Native Computing Foundation (CNCF) [1] und einer von vier Anbietern im Leader 2023 Gartner® Magic Quadrant™ für strategische Cloud-Plattformdienste [2], unterstützt Oracle diese Cloud-Computing-Ansätze mit einer vollständigen Palette von verwalteten beziehungsweise. "serverless" bereitgestellten Diensten, deren Herzstück die Container- und Kubernetes-Plattformen darstellen.

Erste Gehversuche in der Oracle Cloud lassen sich mit einem Budget von \$300 in einem "Free Tier" [3] machen, der nach 30 Tagen in einen "Always Free"-Service übergeht. Wenn man die Developer Tools in der OCI-Web-Konsole nutzt, braucht man auf seinem Entwicklungsrechner zunächst nichts zu installieren, um Tutorials durchzuarbeiten oder erste Eigenentwicklungen zu starten. In der Cloud Shell, dem Terminal einer kleinen, kostenfrei nutzbaren, flüchtigen Linux VM mit 5GB persistentem Speicher, ist man sofort gegen den eigenen Cloud Tenant authentifiziert. Eine ganze Reihe nützlicher Werkzeuge sind hier bereits in aktuellen Versionen installiert, neben zum Beispiel Git, Java, Python, Docker, Terraform, Ansible, Kubectl, Helm, Node.js und den OCI SDKs unter anderem auch das OCI Command Line Interface (CLI), mit dem man die komplette Cloud-Infrastruktur skripten [4] kann, um Klicks in der Web-Oberfläche zu sparen (siehe Listing 1).

Darüber hinaus kann man - ebenfalls in der Web-Konsole - den auf Eclipse Theia basierten und mit der Cloud Shell integrierten Code-Editor nutzen, um Code und Skripte komfortabel zu editieren.

Terraform und die Secure Landing Zone

In der Praxis wird man für einen effizienten Betrieb eine Cloud-Infrastruktur deklarativ beschreiben und zum Beispiel per Ansible oder Terraform automatisiert bereitstellen wollen, um eine möglichst hohe Konsistenz zwischen den Umgebungen (Dev, Test, Prod) zu erlangen. Auch die so gewonnene Versionier- und Wiederholbarkeit ist im Betriebsalltag wichtig, um Änderungen transparent nachverfolgen und auch rückgängig machen zu können.

Einen sicheren und schnell anzuwendenden Startpunkt bietet hierfür die von Oracle in Form von Terraform-Skripten bereitgestellte Vorlage für eine CIS-konforme Landing Zone [5]. Diese ist eine Referenzarchitektur für eine Cloud-Umgebung, die dem "CIS (Center for Internet Security) Oracle Cloud Infrastructure Foundations Benchmark" und somit branchenweit anerkannten Sicherheitsstandards entspricht, auf diesem Wege Sicherheitsrisiken minimiert und vor potenziellen Bedrohungen schützt.

Die Architektur der Landing Zone beginnt mit dem Aufbau von Teilbereichen, den sogenannten OCI Compartments, und dem Einrichten von Benutzergruppen unter anderem für die Netzwerk-, Sicherheits-, Applikations- und Datenbankadministration. Durch gezielte Rechtevergabe wird eine sinnvolle Aufgabentrennung gewährleistet und sichergestellt, sodass Benutzer nur Ressourcen innerhalb des ihnen zugewiesenem Compartments verwalten können. Ebenfalls wird das Virtuelle Cloud Netzwerk (VCN) für eine Huband Spoke- Architektur aufgesetzt, um gemeinsam genutzte von anwendungsspezifischen Diensten zu trennen. Eine Palette von Sicherheits-Diensten, die unter anderem Cloud Guard, Key Vault, Vulnarability Scanning, Bastion und Security Zones umfassen, kann ebenfalls ausgebracht werden. Das Einrichten der Logund Benachrichtigungsdienste schließt die sichere Basiskonfiguration ab.

Serverless Container Instances

Eine sichere infrastrukturelle Basis ist mit der Landing Zone gelegt, nun kann man sich an die Ausbringung der eigenen Arbeitslasten begeben. Im Falle einer Anwendung mit übersichtlicher Komplexität, die keine Container-Orchestrierung benötigt und in einen oder wenige, isolierte Docker-Container gepackt werden kann, sind die OCI-Container-Instances sicherlich der einfachste und schnellste Weg. Zum gleichen Preis, wie bei klassischem "Compute" für selbst verwaltete, virtualisierte Ausbringung, erfährt man hier die Vorteile schnellerer Startzeiten und die Wartungsfreiheit eines "serverless"-Container-Dienstes.

Bei der Erstellung einer Container Instance muss man nur wenige Parameter angeben. Man wählt einen flexiblen "Shape" mit AMD x86 oder Ampere ARM Prozessorarchitektur und weist einen bis zu 64 beziehungsweise 76 Prozessorkerne und einen bis zu 1024 beziehungsweise 488 GB Hauptspeicher zu. Danach wählt man ein virtuelles Netzwerk und zu startende Container aus der OCI-internen oder einer externen Container Registry aus, denen prozentuale oder absolute Ressourcenanteile zugewiesen werden können, wenn etwa Sidecar- oder Log-Container mit ausgebracht werden, die nur wenige Ressourcen benötigen.

Container Instances bringen aber auch einige Einschränkungen mit sich. Zum Zeitpunkt der Schriftlegung warten wir noch auf die Möglichkeit, Block- und File-Storage anzuhängen. Für Persistenz im Rahmen dieser flüchtigen Container wird derzeit noch die Verwendung einer Datenbank empfohlen. Zudem gibt es nur die Möglichkeit, vertikal zu skalieren, das heißt, einer Instanz mehr Rechenleistung und Speicher zuzuweisen. Für horizontale und vertikale Skalierbarkeit und für die Möglichkeit der Orchestrierung komplexer Container-Anwendungen sollte man auf die OCI Container Engine für Kubernetes (OKE) wechseln.

Kubernetes in der Cloud

Die Container-Orchestrierung mit Kubernetes bietet dem Cloud-Native-Entwickler enorme Möglichkeiten, auch komplexe Anwendungen schnell, flexibel und skalierbar ausbringen und ablaufen lassen zu können. Der eigenhändige Umgang mit dem Framework ist inhärent komplex und mit nicht unerheblichen Installations- und Betriebsaufwänden verbunden. Die Nutzung eines vollständig verwalteten Kubernetes-Dienstes in der Cloud kann daher die Kosten erheblich reduzieren, die Betriebsteams entlasten und durch die Vermeidung von Konfigu-

```
# Tenancy Namespace anzeigen:
oci os ns get
# Gefilterte Ausgabe der Compartments als Tabelle:
oci iam compartment list --all --query "data[].{\"NAME\":name,\"ID\":id}" --output table
# Liste der verfügbaren Compute Images:
oci compute image list -c <compartment_id>
```

Listing 1: OCI CLI Skript-Beispiele

```
oci ce node-pool update --cluster-id <cluster-id> --node-pool-id <node-pool-
id> --kubernetes-version <new-version>
```

Listing 2: OCI CLI Node Pool Upgrade

```
# Initiieren einer Java-Funktion
fn init --runtime java myjavafn
# Erstellen einer Applikationsgruppe für die Funktion
fn create app myjavaapp
# Kompilieren und lokales Deployment der Funktion
fn --verbose deploy --app myjavaapp -local
# Lokales Ausführen des Funktions-Containers
fn invoke java-app myjavafn
```

Listing 3: OCI Java-Function per CLI erstellen

rationsfehlern die Sicherheit der Plattform erhöhen.

Die OCI Container Engine für Kubernetes (OKE) bietet eine solche, leistungsstarke und vollständig verwaltete Kubernetes-Plattform in der Oracle Cloud, die Entwicklern und Betriebsteams die größte Bandbreite an Möglichkeiten an die Hand gibt, containerisierte Anwendungen effizient bereitstellen, verwalten und skalieren zu können.

Bei der Provisionierung eines Kubernetes-Clusters über die OCI-Konsole kann man zwischen einer schnellen und einer benutzerdefinierten Erstellung wählen. Bei der schnellen Erstellung sind nur wenige konfigurative Vorarbeiten erforderlich, da das benötigte virtuelle Netzwerk mit den Subnetzen und Gateways für die Kubernetes API, die Worker Nodes und die Load Balancer mit dem Kubernetes-Cluster und seinen Node Pools und Worker Nodes automatisch generiert werden. Bei einer benutzerdefinierten Erstellung wählt man im Sinne einer thematischen Aufgabentrennung in der Administration bestehende Netzwerkkomponenten für die Verwendung aus.

Für jedes der drei Subnetzte hat man die Wahl zwischen einem privaten Subnetz, das nicht aus dem Internet erreichbar ist, oder einem Subnetz für öffentlich erreichbare Anwendungen, in dem man öffentliche IP-Adressen zuweisen kann. Aus Sicherheitsgründen sollte man – bis auf wenige Ausnahmen – die Kubernetes API und die Worker Nodes immer in privaten Subnetzen betreiben. Öffentliche Zugriffe können bei Bedarf über Load Balancer in öffentlichen Subnetzen gestaltet und gegebenenfalls mit dem "API Gateway" und der "Web Application Firewall" weiter abgesichert werden.

Natürlich lässt sich die Erstellung eines Kubernetes Clusters beispielsweise mit Terraform automatisieren. Dies ergibt nicht nur für die Professionalisierung des Betriebs Sinn, denn auch Entwickler können dies nutzen, um flexibel und nach eigenem Bedarf Entwicklungs- und Testressourcen bereitzustellen, sogar für kürzeste Zeiträume. Auch wenn es sich zum Zeitpunkt der Schriftlegung um eine Vorabversion handelt, gibt das in GitHub veröffentlichte "Terraform OKE Quickstart Module" [6] einen vollständigen und sofort anwendbaren Einblick in die Automatisierung einer Cluster-Erstellung.

Generell kann man sich entscheiden. ob man selbst verwaltete, verwaltete oder virtuelle Worker Nodes nutzen möchte. Letztere bieten eine "serverless"- Nutzererfahrung, bei der man Betriebsaufwände spart und bei der man sich hinsichtlich Skalierung im Vorhinein keine Gedanken um die Kapazitäten eines Clusters machen muss. Hier wählt man nur einen flexiblen AMD x86 oder Ampere ARM Shape. Eine Skalierung und Isolierung findet nun auf Ebene eines Kubernetes Pods statt, nicht auf Ebene eines Nodes, wodurch man auch exakt für die von einem Pods benötigten Ressourcen abrechnet, und nicht mehr für eventuell ungenutzte, ganze Server.

Für klassische Nodes stehen eine Vielzahl von vordefinierten und flexiblen AMD- und Intel-x86- sowie ARM-basierten VMs und Bare Metal Shapes zur Verfügung, einschließlich optimierter GPU, HPC und Dense-IO Shapes für anspruchsvolle Aufgaben.

Unterschiede zeigen sich auch beim Upgrade der Nodes auf neuere Kubernetes- Versionen, was beim Control Plane hingegen immer ein automatisierter Vorgang ist, der durch einen Klick beziehungsweisse einen API-Aufruf im laufenden Betrieb gestartet werden kann.

Bei selbstverwalteten Nodes wechselt man die Knoten einfach gegen Neue in der neueren Version aus. Hier obliegt einem selbst die Kontrolle und man muss dafür sorgen, dass die einzelnen Komponenten des Kubernetes-Clusters zu jedem Zeitpunkt in kompatiblen Versionen laufen. Bei verwalteten Nodes reicht es, eine neue Version für einen Node Pool zu spezifizieren und ein "Cycle Nodes" auszuführen. Dies kann nicht nur über die Konsole, sondern auch über die OCI API erfolgen (siehe Listing 2).

Bei Verwendung von virtuellen Nodes werden diese im Rahmen eines Control-Plane-Upgrades koordiniert sowie an Ort und Stelle aktualisiert.

Will man die Kommunikation zwischen den Mikroservices in einer Kubernetes-Anwendungsinfrastruktur effizienter und zuverlässiger gestalten, bietet sich die Nutzung des OCI Service Meshan, einem auf ISTIO basierten Framework, das separate Proxy-Container an Seite der Microservices andockt. Entkoppelt vom Applikationscode, bietet es konfigurierbare Funktionalität für

Lastverteilung, Fehlererkennung, (Policy-basierte Zero Trust) Security, Überwachung und Logging. Es kann über die OCI-Konsole, Kubectl, Helm und die OCI APIs sowie SDKs konfiguriert werden.

Serverless Functions

Bei ereignisgesteuerten, zustandslosen und kurzlaufenden Anwendungen, die man in den Anwendungsbereichen Cloud Events, ETL, IoT, ChatBots, mobilen Backends oder API-Bereitstellungen häufig findet, benötigt man oft keine dauerhafte Infrastruktur. Hier bietet OCI Functions eine flexible und effiziente Möglichkeit, Code auszuführen, der auf Ereignisse in der Cloud reagiert und bei der man nur für die Ressourcen zahlt, die man tatsächlich nutzt. Entwickler können sich auf die Implementierung von Geschäftslogik konzentrieren, während die zugrunde liegende "Serverless"-Infrastruktur von der OCI verwaltet und skaliert wird.

Da OCI Functions auf dem Containernativen Open-Source Framework "Fn Project" [7] basiert, können Entwickler serverlose Funktionen erstellen, die problemlos in andere Cloud- und On-Premises-Umgebungen portiert werden können. Entwickler können Funktionen in einer Vielzahl von Programmiersprachen und für verschiedenste Laufzeitumgebungen schreiben, darunter Python, Node.js, Java, Go, C# und Ruby. Die Entwicklung erfolgt mit dem FN-Kommandozeilenwerkzeug (Fn CLI) oder noch komfortabler mit Unterstützung des OCI-Plugins für Visual Studio Code. Die Initialisierung eines Fn-Projektes für eine bestimmte Laufzeitumgebung generiert einen beispielhaften, bereits - auch lokal - lauffähigen Funktionsrumpf. Diesen implementiert man nun aus, bevor man ihn kompiliert, als Container packt und in den OCI Functions Service ausbringt. Functions stellt dafür vorgefertigte Build- und Run-Container zur Verfügung, so dass man für eine Java-Function im Prinzip nur vier Befehle braucht (siehe Listing 3).

Streaming oder Queueing?

Durch Gerätevernetzung, das mobile Internet, das Wachstum der sozialen Medien und der Prozessdigitalisierung steigt

die Menge an generierten Daten exponentiell an. Schätzungen gehen davon aus, dass mehr als 90% der weltweiten Daten in den letzten paar Jahren generiert wurden. Um diese hohen Lasten bewältigen zu können, bietet sich eine asynchrone Kommunikation zwischen Services und Systemen an, nicht nur aus Gründen der Entkopplung, sondern auch für leichtere Skalierbarkeit, Fehlerbehandlung und bessere Resilienz.

Zwei Konzepte stehen hier alternativ zur Verfügung: die verteilte Datenstromverarbeitung und Nachrichtenwarteschlangen. Beide sind in der Oracle Cloud als autonome Dienste mit 99,9-prozentiger Verfügbarkeitsgarantie nutzbar, in Form des Apache-Kafkakompatiblen OCI Streams und als OCI Queue. Eine Entscheidung für eines der beiden Konzepte ist nicht immer einfach, kann aber anhand folgender Überlegungen festgemacht werden:

Beim Streaming steht die Verarbeitung von kontinuierlichen Datenströmen in Echtzeit im Vordergrund, vor allem wenn die Reihenfolge der Daten von Bedeutung ist. Es skaliert unterbrechungsfrei mit der Anzahl an Knoten, kann Daten mehrerer Produzenten an mehrere Konsumenten verteilen und ist fehlertolerant, da die Daten über mehrere Tage in einem verteilten - im Falle von OCI Streams sogar über mehrere Rechenzentren hinweg -, replizierten Commit-Log persistiert werden können. Eine Anwendung kann sogar Daten aus dem Cluster bezüglich einer Fragestellung konsumieren und auswerten, die beim Produzieren der Daten noch nicht bekannt war. Das weitreichende Kafka-Ökosystem ist ein Vorteil, wenn man eine komplexe, händische Installation durch einen vollautomatisch provisionierten Cloud-Dienst vermeiden kann.

Die Message Queue kann hingegen in einer Punkt-zu-Punkt-Kommunikation die genau einmalige, zuverlässige Zustellung einer Nachricht garantieren. OCI Queue skaliert vollautomatisch und unterbrechungsfrei abhängig vom Durchsatz. Abgerechnet wird die Anzahl der Zugriffe, nicht die benötigte Infrastruktur, wobei die ersten eine Million Zugriffe pro Monat kostenfrei sind. Bis zu 1000 Zustellversuche innerhalb einer Aufbewahrungszeit von sieben Tagen sind konfigurierbar, danach wandern nicht zustellbare Nachrichten in eine "Dead Letter Queue".

Beide Dienste verschlüsseln Daten bei Transport und Speicherung, auch mit kundeneigenen Schlüsseln.

Zusammenfassung

Die Oracle-Cloud-Infrastruktur bietet ein umfassendes Portfolio an Diensten für die Entwicklung von Cloud-Native-Anwendungen, die sich an den Richtlinien der Cloud Native Computing Foundation orientieren. Sie stehen als einfach zu nutzende, verwaltete und autonome Plattformdienste in gleichem Umfang und zum gleichen Preis in allen Varianten der Oracle Cloud zur Verfügung: In 44 Regionen und 23 Ländern der Oracle Public Cloud, in der Oracle EU Sovereign Cloud, als Alloy für Partner und Dedicated Cloud im Kunden-eigenen Rechenzentrum.

Quellen

- [1] https://www.cncf.io
- [2] https://www.oracle.com/de/cloud/ gartner-mq-strategic-cloud-platform-services-leader/
- https://www.oracle.com/cloud/free/
- https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/gettingstartedwiththeCLI.
- [5] https://github.com/oracle-quickstart/ oci-cis-landingzone-quickstart
- https://github.com/oracle-quickstart/terraform-oci-oke-quickstart
- [7] https://fnproject.io/

Über den Autor

Marcel Uusvel-Amende, gelernter Dipl.-Ing. der Elektrotechnik, begeht dieses Jahr sein 25. Firmenjubiläum bei Oracle. Als Senior Leitender Cloud Engineer und Spezialist für Integrations- und Entwicklungsthemen, unterstützt er mit seiner Erfahrung die deutschen Finanzdienstleiter bei ihren Lösungsarchitekturen und Cloud-Migrationen.



Marcel Uusvel-Amende marcel.amende@oracle.com



DOAG Datenbank mit Exaday

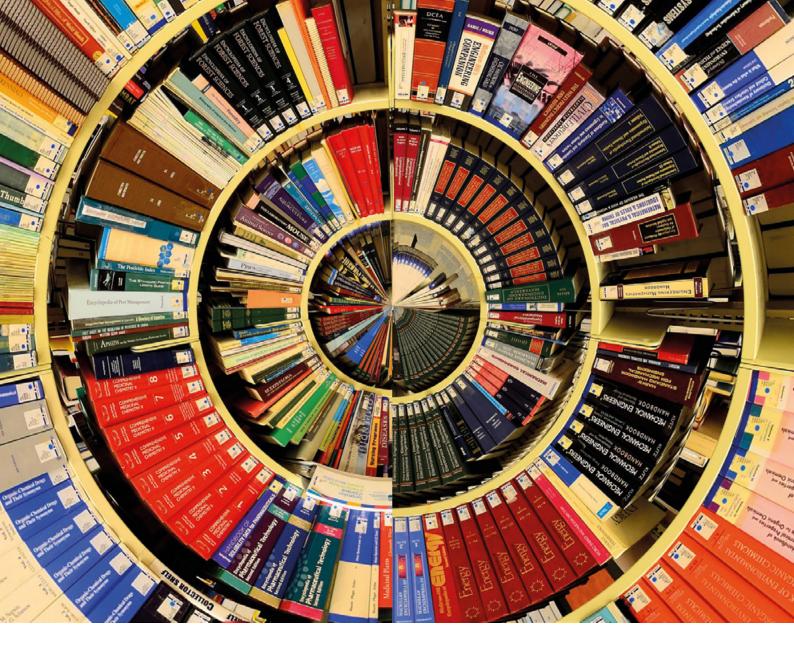
2023

ON DEMAND

DATENBANK 2023 VERPASST?

Jetzt On-demand-Ticket buchen und Vortragsaufzeichnungen anschauen!





Konsistenz und Synchronisation in verteilten Systemen

Dr. Stefan Koch, Ordix

Schon allein aufgrund der regulatorischen Anforderungen der BaFin ist eine Bank verpflichtet, Daten ihrer Kunden zu verwalten. Ein Finanzinstitut kann nur dann für einen Kunden verantwortungsvoll Finanztransaktionen durchführen, wenn es den Kunden kennt: Know-Your-Customer. Die Qualität der Kundendaten wird durch eine zentrale Datenhaltung sichergestellt. Nicht selten werden Kundendaten allerdings auch dezentral benötigt. Beispielsweise ist für die Verwaltung von Krediten eine Standardanwendung im Einsatz, die Teile der Kundendaten in ihrer eigenen Anwendung benötigt. Für solche Anwendungen ist eine Datensynchronisation im Einsatz, um eine Kopie der Kundendaten in der Kreditanwendung aktuell zu halten. Die Anforderungen an die Datensynchronisation sind überschaubar. Die abzulösende Altanwendung, die mehr als 20 Konsumenten mit den Daten aus 170 Tabellen versorgt, hat uns aber gezeigt, dass aus einer einfachen Anforderung eine hochkomplexe Anwendung werden kann. Die Herausforderung besteht also darin, zu einer einfachen Anforderung auch eine einfache Lösung zu finden. Die Entscheidungen, die einen wesentlichen Beitrag zur Einfachheit der Lösung beitragen, sind im Folgenden aufgeführt.

1 Fachliche Anforderungen

Banken müssen verhindern, dass ihre Kunden Geldwäsche praktizieren, Terroristen finanzieren oder mit sanktionierten Partnern Geschäfte machen. Bereits aus diesem Grund müssen Banken. Daten zu ihren Kunden erfassen. Die Erfassung und Verwaltung von Kundendaten erfolgt in der Kundenverwaltung. Verändert sich beispielsweise die Anschrift eines Kunden durch Umzug, so werden die Kundendaten in der Kundenverwaltung angepasst. Die Datenveränderung wird zeitnah durch Synchronisation an sogenannte Umsysteme übermittelt. Das Umsystem hat nach der Synchronisation einen zur Kundenverwaltung konsistenten Datenbestand.

In Abbildung 1 ist der Datenfluss zwischen einem Umsystem (zum Beispiel Anwendung Kreditvergabe) und der zentralen Kundendatenbank dargestellt. Das Umsystem speichert die Daten des Kunden in seiner Datenbank, es hält somit eine Kopie der Kundendaten. Änderungen der Kundendaten erfolgen ausschließlich über die Kundenverwaltung: Nach einer Änderung sendet die Kundenverwaltung ein Data-Event an das Umsystem. Das Event enthält Kundendaten, die von der Änderung betroffen sind. Das

Umsystem verwendet das Data-Event, um seinen Bestand der Kundendaten zu aktualisieren. Durch diese Kooperation von Kundenverwaltung und Umsystem erfolgt die Synchronisation der Kundendaten von der zentralen Kundendatenbank in die dezentrale Datenbank des Umsystems.

Wesentlich für die Synchronisation sind die folgenden nicht-funktionalen Anforderungen:

- Die Performance der Kunden-Verwaltung darf durch die Synchronisation nicht wesentlich beeinflusst werden.
- Die Aktualisierung der Daten in den Umsystemen muss zeitnah – im Maßstab von Sekunden – erfolgen.
- Nach der Synchronisation sind die Kundendaten in den Umsystemen aktuell und konsistent.
- Die Synchronisation ist auf beliebig viele Umsysteme skalierbar.

2 Erfolgsfaktoren

Die folgenden Faktoren sind dafür entscheidend, dass eine implementierte Synchronisation einfach implementiert werden kann, sich im Betrieb robust verhält und eine hohe funktionale Qualität aufweist:

- Schnittstellen-Design:
 - Biete die Synchronisation als Service an.
 - Verwende das Objektmodell des Daten-Owners.
 - Ordne die Objekte Aggregaten zu.
- Lösungs-Design:
 - Führe die Daten-Synchronisation asynchron aus.
 - Separation of Concern: Die Erstellung von Data-Events erfolgt in eigener Anwendung.
 - Biete neuen Konsumenten ein bekanntes Konzept an: Event-Driven-SOA.
 - Biete den Konsumenten einen einfachen Migrationsweg an.
- Implementierung:
 - Entwickle evolutionär. Verwende das Open-Closed-Principle.
 - Setze Domain-Specific-Language für komplexe Entscheidungslogik ein.
 - Verwende Contract-Tests als Unit-Tests.

2.1 Schnittstellendesign

2.1.1 Synchronisation als Service

Die Schnittstellenabstimmung kann eine umfangreiche Aufgabe sein: Die beteiligten Schnittstellenpartner diskutieren dabei über Inhalt, Verhalten, Technologie und Format der Schnittstelle. Mit mehr als 20 Schnittstellenpartnern ist eine solche Einigung faktisch unmöglich – zu unterschiedlich sind die Bedürfnisse der Umsysteme.

Einfacher ist der Weg, einen Service anzubieten. Der Service findet dann Akzeptanz, wenn mit seiner Hilfe Anforderungen der Umsysteme umsetzbar sind. Für das Erwartungsmanagement der Umsysteme muss die Zielsetzung des Services deutlich kommuniziert werden: In diesem Fall bietet der Service aktuelle Kundendaten an, wenn sich Daten in der Kundenverwaltung ändern.

2.1.2 Objektmodell des Daten-Owners

Ein Bestandteil der Schnittstellen-Spezifikation ist das Objektmodell. Da die Daten der Kundenverwaltung Gegenstand der Synchronisation sind, liegt bereits das Objektmodell der Kundenverwaltung vor. Dieses Objektmodell des Data-Owners

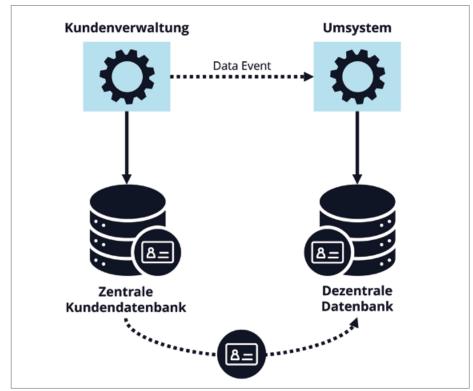


Abbildung 1: Synchronisation zwischen zentraler und dezentraler Datenbank (Quelle: Dr. Stefan Koch)

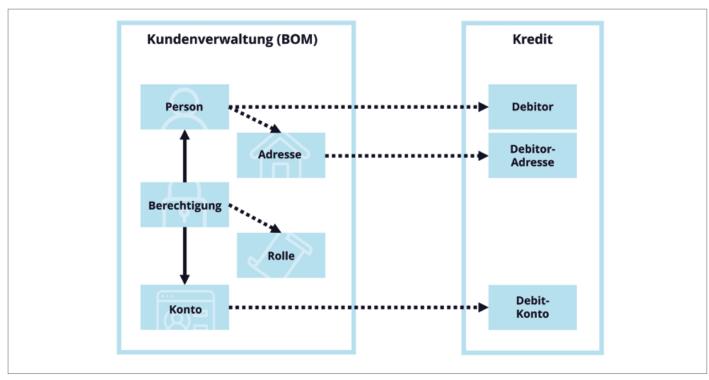


Abbildung 2: Synchronisation zwischen Kundenverwaltung und Kredit (Quelle: Dr. Stefan Koch)

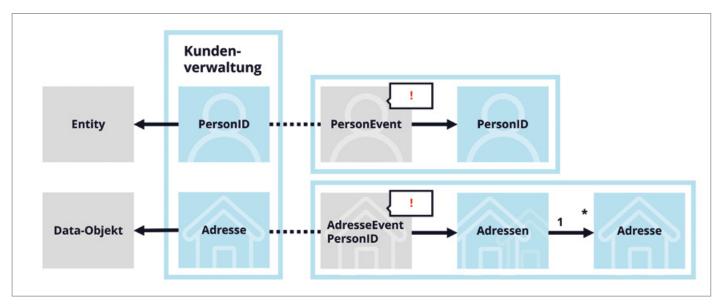


Abbildung 3: Zuordnung des BOM zu Eventobjekten (Quelle: Dr. Stefan Koch)

wird auch für die Schnittstelle verwendet. Das verringert den konzeptionellen Aufwand und bringt zudem folgende Vorteile mit sich:

- Vollständigkeit: Jedes Attribut wird berücksichtigt.
- **Redundanzfreiheit:** Jedes Attribut wird genau einmal berücksichtigt.
- Hohe Stabilität: Veränderungen am Business-Object-Model sind selten.
- Verständlichkeit: Die Datenobjekte der Synchronisation lassen sich ein-

fach über den Verweis auf das Business-Object-Model dokumentieren.

Die Verwendung des Business-Object-Modells (BOM) der Kunden-Verwaltung kann für die Umsysteme eine Herausforderung darstellen (siehe Abbildung 2). Die Objektbeziehungen in der Kundenverwaltung müssen nicht zwangsläufig mit denen im Umsystem übereinstimmen. So werden beispielsweise Personendaten in die Tabelle Debitor übernommen. Die Debitor-Tabelle vergibt für ihren Debitor

einen eigenen Identifier. Wichtig ist, dass der Identifier zur Person aus der Kundenverwaltung auch in der Kreditdatenbank abgelegt wird. Nur so können Adressdaten der Synchronisation als Debitor-Adresse dem richtigen Debitor zugeordnet werden.

2.1.3 Aggregate

Der Datenbedarf der Umsysteme ist unterschiedlich: Eine Anwendung ist an der Meldeadresse interessiert, eine andere möchte nur die Basisdaten eines Kunden

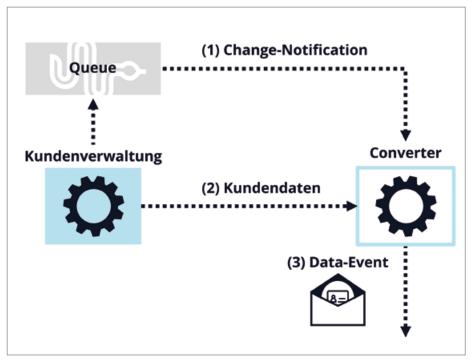


Abbildung 4: Converter empfängt Change-Notification und erzeugt Data-Event (Quelle: Dr. Stefan Koch

synchronisieren. Das Service-Angebot folgt dem Ansatz Need-to-Know: Es werden nur die Daten geliefert, die das Umsystem benötigt. Bei der Aufteilung der Kundendaten ist das Konzept der Aggregate wie von Eric Evans beschrieben [1] hilfreich. Ein Aggregat ist beispielsweise die natürliche Person. Die natürliche Person besteht aus den Daten zur Person wie Name, Vorname oder Geburtsdatum und den angegliederten Objekten wie Meldeadresse, E-Mail-Adresse oder Telefonnummer. Die Zuordnung von Objekten zu Aggregaten legt die Zugehörigkeit fest. Eine Adresse gehört zu einer Person. Die Synchronisation liefert Objekte im Kontext eines Aggregats aus. Die Zugehörigkeit der Meldeadresse zur Person erfolgt über den Identifier der Person. Diese Zuordnung muss das Umsystem

verwenden, um in seiner Datenbank die Meldeadresse zuordnen zu können.

Die Einteilung der Objekte in Entityund Data-Objekte [1] ist für die Semantik der Schnittstelle hilfreich. Ein Entity-Objekt besitzt eine Identität. So kann sich der Name einer Person durch Heirat verändern - die Person bleibt aber dieselbe. Das ist gerade im Bankenumfeld wichtig, damit der Kunde auch nach der Hochzeit auf seine Konten zugreifen kann. Im Gegensatz dazu gibt es Objekte ohne Identität: zum Beispiel die Adresse. Eine Adresse verändert sich nicht - ganz selten wird eine Straße umbenannt. Im Fall eines Umzugs wird eine alte Adresse gegen die neue Adresse ausgetauscht. Für die Datensynchronisation werden Data-Objekte immer als Liste übertragen (siehe Abbildung 3). Eine leere Liste bedeutet: Es gibt kein Data-Objekt dieses Typs im Aggregat. Liegt also keine E-Mail-Adresse zu einem Kunden vor, so wird eine leere Liste von E-Mail-Adressen übertragen. Alternativ wird eine Liste aller E-Mail-Adressen des Kunden übermittelt.

2.2 Lösungsdesign

2.2.1 Asynchrone Datensynchronisation

Die Kunden-Verwaltung soll möglichst wenig von der Synchronisation beeinträchtigt werden. Um das zu erreichen, erfolgt die Synchronisation asynchron. Die Hauptaufgabe der Kundenverwaltung besteht darin, die fachliche Konsistenz der Kundendaten zu gewährleisten. Dazu sind mehr als 200 Geschäftsregeln im Einsatz, die unzulässige Datenveränderungen verhindern. Ein wesentlicher Teil der Datensynchronisation wird durch den sogenannten Converter übernommen (siehe Abbildung 4). Als Trigger für die Synchronisation erstellt die Kundenverwaltung eine Change-Notification. Diese enthält die Information, welches Objekt sich verändert hat. Diese Change-Notification wird asynchron, über eine Message-Queue, dem Converter übermittelt. So kann die Kundenverwaltung, unabhängig vom weiteren Verlauf des Synchronisationsprozesses, ihre Arbeit fortsetzen. Der Converter liest die Change-Notification aus der Queue. Aus der Notification ermittelt der Converter, welche Daten sich verändert haben. Daraufhin liest der Converter das dazu passende Aggregat aus der Kundenverwaltung. Diese Daten verwendet der Converter, um das Data-Event zu erzeugen.

2.2.2 Separation of Concern

Durch die asynchrone Kommunikation wird der Prozess der Kundenverwaltung von dem Prozess der Synchronisierung entkoppelt. Das hat allerdings nicht nur Vorteile. Um den erforderlichen Durchsatz an Change-Notifications zu bewältigen, verarbeitet der Converter Change-Notifications parallel. Infolgedessen entspricht die Reihenfolge der Data-Events nicht der Reihenfolge der Change-Notifications. Am Beispiel der Kundenneuanlage lässt sich das Reihenfolge-Problem erläutern. Der Mitarbeiter trägt bei der Kundenadres-

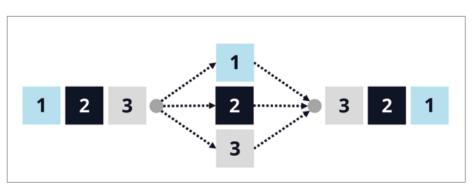


Abbildung 5: Versionen der Daten und Reihenfolge der Data-Events (Quelle: Dr. Stefan Koch)

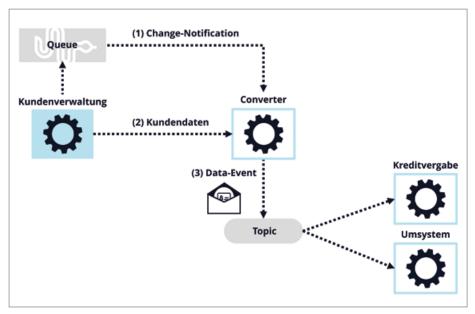


Abbildung 6: Datensynchronisation nach dem Muster Event-Driven-SOA (Quelle: Dr. Stefan Koch)

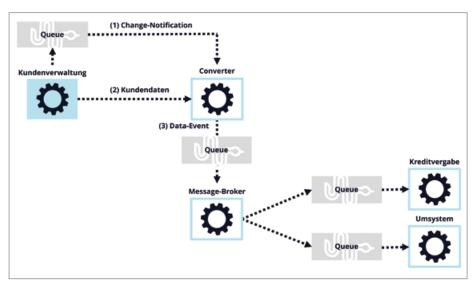


Abbildung 7: Alternative zu Event-Driven-SOA: Message-Broker liefert Events an Umsysteme (Quelle: Dr. Stefan Koch)

se zunächst die 13 als Hausnummer ein. Er bemerkt kurz darauf seinen Fehler und trägt als Hausnummer aber 31 ein. Zwei Change-Notifications werden dadurch erzeugt. Aufgrund der Kundenneuanlage werden viele Data-Events erzeugt. Die Korrektur der Adresse erzeugt nur ein Data-Event. So kann es vorkommen, dass das Data-Event zur Adresse zuerst beim Umsystem ankommt. Das Umsystem kennt aber den neuen Kunden, der zu dieser Adresse gehört, noch nicht.

Das Beispiel zeigt, dass die Reihenfolge der Data-Events nicht der Reihenfolge der Datenänderungen entsprechen muss. Aus der Perspektive der Umsysteme handelt es sich um einen Überholer: Das

Adress-Data-Event hat das initiale Data-Event zur Person überholt. Auch die Weiterleitung der Data-Events kann zur Veränderung der Reihenfolge führen.

In Abbildung 5 ist der Fall dargestellt, dass ein Entity durch Veränderung in der Kundenverwaltung in drei Versionen vorliegt. Die erste Veränderung erzeugt die Version ,1', die zweite Veränderung die Version ,2' und so weiter. Aufgrund der Veränderung erzeugt die Kundenverwaltung zuerst die Notification eins, dann zwei, dann drei. Diese Notifications werden vom Converter aber parallel verarbeitet. In der Abbildung erfolgt die Verarbeitung gleichzeitig. Der Converter erzeugt (zufälligerweise) zuerst ein Data-

Event zur Version ,3', dann zur Version ,2' und schließlich zur Version ,1'.

Das Umsystem muss seinen Teil der Datensynchronisation auf eine so veränderte Reihenfolge einstellen. Mindestens zwei Maßnahmen müssen ergriffen werden. Der Verweis auf ein unbekanntes Aggregat muss möglich sein. Kennt das Umsystem die Person-ID einer Adresse nicht, so kann es die Person mit Dummy-Daten erstellen - in der Erwartung, dass Daten zu dieser Person in Kürze geliefert werden. Darüber hinaus wertet das Umsystem den Zeitstempel des Data-Events aus. Dieser Zeitstempel kann als Version des Objekts verstanden werden. Kommen Data-Events zu früheren Versionen des Objekts darf das Umsystem diesen Datenbestand nicht übernehmen, da es sich um einen veralteten Datenbestand handelt. Diese Art der Verarbeitung ist robust, denn der Verlust von Data-Events führt nur vorübergehend zu einem Datenschiefstand. Das nächste Data-Event zu diesem Objekt heilt den Daten-Schiefstand wieder.

Neben den Data-Events, die Daten zu einem Objekt anliefern, verfügt die Synchronisation auch über ein Delete-Event. Ein Delete-Event kommuniziert die Löschung eines ganzen Aggregats. Das Umsystem muss daraufhin feststellen, welche Objekte zu diesem Aggregat gehören, um diese dann zu löschen.

2.2.3 Event-Driven-SOA

Für den Transport der Data-Events vom Converter zu den Umsystemen ist das Design-Pattern Event-Driven-SOA [2] geeignet: Die Data-Events werden zur Kommunikation in ein Topic eingestellt (siehe Abbildung 6). Die Umsysteme können Typen von Data-Events abonnieren. Auf diese Weise ist der Converter als Message-Producer getrennt von den Umsystemen, die die Rolle Message-Consumer der Data-Events einnehmen. Der Converter muss nicht wissen, welche Umsysteme seine Data-Events konsumieren. Der Service besteht darin, dass Data-Events in einem Topic vorliegen. Wer diese Data-Events erstellt und wer die Data-Events liest, ist für die Kommunikation bedeutungslos. So lässt sich das Data-Event selbst als Service verstehen.

2.2.4 Vereinfachung der Migration

Zusätzlich zur Implementierung des Event-Driven-SOA (siehe Abbildung 6) wird

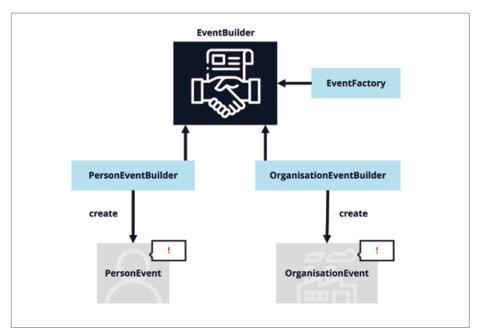


Abbildung 8: Implementierung des Open-Closed-Principles (Quelle: Dr. Stefan Koch)

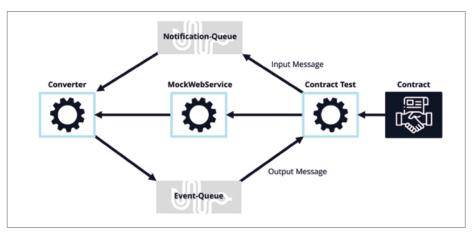


Abbildung 9: Architektur des Contract-Driven-Tests (Quelle: Dr. Stefan Koch)

ein weiterer Kommunikationsweg angeboten. Der Converter leitet Data-Events in eine Message-Queue. Diese Data-Events liest der Message-Broker, um sie in die Queues der Umsysteme auszuliefern (siehe Abbildung 7). Dafür gibt es folgende Gründe:

- Die normative Kraft des Faktischen: Der Message-Broker wurde in der vorherigen Lösung eingesetzt.
- Die Umsysteme sind mit Message-Queues vertraut.
- Need-to-Know: Jedes Umsystem bekommt exakt die Daten geliefert, die es braucht.
- Autorisierung: Der Message-Broker autorisiert den Datenbedarf eines jeden Umsystems auf der Ebene Mandant und Event.

2.3 Implementierung

2.3.1 Open-Closed-Principle

Data-Events lösen ein bestehendes Verfahren zur Synchronisation ab. Mehr als 20 Umsysteme wurden von dem bestehenden Verfahren mit Daten versorgt. Die Einführung der Data-Events ist so Schritt für Schritt möglich gewesen. Zuerst wurden die Umsysteme umgestellt, die einen geringen Datenbedarf haben, bevor weitere Umsysteme angeschlossen wurden. So wurden zuerst wenige Typen von Data-Events im Converter implementiert. Der evolutionäre Aufbau des Converters war hilfreich, damit die zu erwartenden Fehler eine geringe Auswirkung haben und schnell gefixt werden können. Der evolutionäre Gedanke ließ sich besonders effektiv umsetzen, indem das *Open-Closed-Principle* [3] eingesetzt wird.

Das Open-Closed-Principle wird entsprechend Abbildung 8 implementiert. Jedes Data-Event verfügt über einen eigenen Event-Builder. Dieser Builder entscheidet selbständig, ob er eine Notification behandeln muss und woraus sich das Data-Event zusammensetzt. Jeder Event-Builder ist als Spring-Bean [10] registriert. Das Spring-Framework stellt eine Liste aller Event-Builder als Spring-Beans zur Verfügung. Diese Liste ist die Grundlage der Verarbeitung. Die Event-Factory ruft jeden Builder auf und fragt, ob er eine Notification behandelt. Wenn ja, wird der Builder aufgerufen, um Data-Events zu erzeugen. Die Event-Factory sammelt die erstellten Events ein.

2.3.2 Domain-Specific-Language

Eine Herausforderung stellt die Implementierung der Entscheidungslogik dar: Wann erzeugt ein Event-Builder ein Event? Das ist abhängig von der Art der Notification. Es ist aber auch wichtig, dass die Entität in der Kundenverwaltung noch existiert und dass sie nicht den Mandanten gewechselt hat. Diese Regel ist für alle Data-Events die gleiche – und ist daher auch nur an einer Stelle implementiert.

Die Entscheidungslogik wird in Form einer Domain-Specific-Language [4] in Java auf Grundlage von java.util.function.Predicate implementiert (siehe Listing 1). Die Methode is-Responsible-4Root-Update entscheidet anhand von Predicates, ob ein Data-Event für das Root-Objekt eines Aggregats erzeugt werden soll. Das Prädikat is-Notification-present prüft, ob sich die Change-Notification überhaupt auf das Aggregat des Data-Events bezieht. Ist das nicht der Fall, so ist der Event-Builder nicht zuständig.

Bezieht sich die Notification auf das Aggregat des Data-Events, müssen weitere Bedingungen erfüllt werden. Gibt die Notification an, dass das Root-Objekt aktualisiert wurde (Transition-Type ist Update und die Notification gibt das Root-Objekt als Gegenstand der Änderung an), wird für das Root-Objekt ein Data-Event erzeugt, sofern das Aggregat in der Kundenverwaltung noch existiert (Prädikat exists) und noch Kunde der Bank ist (nicht ibs-Tenant-Has-Changed). Dabei ist der Transition-Type Update und die Notification gibt das Root-Objekt als Gegenstand der Änderung an.

```
import java.util.function.Predicate;
@Override
public Predicate<EventResponsibilityDecider> isResponsible4RootUp-
date() {
  return
      isNotificationPresent()
      .and(
         isTransitionType (TRANSITION TYPE UPDATE) .and (isNotification-
4Root())
         .or(isTransitionType(TRANSITION TYPE CREATE))
         .or(isTransitionType(TRANSITION TYPE REACTIVATE))
         .or(isTransitionType(TRANSITION TYPE DEACTIVATE))
      .and(exists())
      .and(ibsTenantHasChanged().negate()));
}
```

Listing 1: Entscheidungslogik wird als Predicate implementiert

```
inputMessage:
 payload: |
    <?xml version="1.0" encoding="UTF-8"?>
    <TransactionNotification>
        <Notifications>
          <PersonenFunktionNotification>
          <NotificationKev>
            <RelationshipId>1234567890</RelationshipId>
            <BpkennGrantor>12345678911/BpkennGrantor>
            <BpkennHolder>47112341234/BpkennHolder>
          </NotificationKev>
          <Transition>Create</Transition>
          </PersonenFunktionNotification>
        </Notifications>
    </TransactionNotification>
 headers:
```

Listing 2: Contract - Definition der Notification

```
Endpoint:
  path: "/personfunction/12345678911/47112341234"
  status: "HTTP/1.1 200"
  responseBody: |
    "data": {
      "personfunctionKey": {
        "grantorBpkenn": "12345678911",
        "holderBpkenn": "47112341234"
      },
      "businessUnit": "Entwicklung",
      "competence": "2",
      "isContact": true,
    }
```

Listing 3: Contract - Definition der Daten aus der Kundenverwaltung

Diese Regel gilt für alle Root-Objekte von Aggregaten. Zu jedem Aggregat müssen nur noch die Prädikate is-Notification-Present und exists angepasst werden.

Durch diese Art der funktionalen Programmierung sind die Regeln lesbar. Sehr hilfreich ist, dass die boolschen Operatoren and und or als Methoden des Interfaces Predicate zur Verfügung stehen. So wird die Überprüfung unmittelbar abgebrochen, falls eine notwendige Bedingung nicht erfüllt ist.

2.3.3 Integrationstests als **Contract-Driven-Tests**

Aufgrund der Vielzahl zu erzeugender Typen von Data-Events kommt dem Test eine große Bedeutung zu. Die zentrale Frage ist: Werden zu einer Change-Notification die passenden Data-Events erzeugt? Getestet wird die Funktionalität des Converters. Der Test-Ablauf ist immer der gleiche. Eine Change-Notfication ist zusammen mit den Daten der Kundenverwaltung gegeben. Das vom Converter erzeugte Data-Event wird überprüft. Die Daten für den Test werden als Contract vorgegeben. Für jeden Testfall gibt es genau eine Contract-Datei, die vom Test eingelesen und verarbeitet wird. Durch das YAML-Format der Contract-Datei ist die Datei auch für Nicht-Entwickler lesbar. So lassen sich Testfälle einfach erzeugen (siehe Listing 2).

Damit der Contract-Test direkt in der Entwicklungsumgebung ausführbar ist, werden die Ressourcen Datenbank, Message-Broker und REST-Services durch eingebettete Äquivalente ersetzt. Der Contract-Test (siehe Abbildung 9) liest den Contract: Der Contract-Test sendet die Notification in die Notification-Queue und stellt einen Web-Service (Mock-Web-Service) mit den Daten der Kundenverwaltung zur Verfügung. Dann wartet der Contract-Test, bis der Converter die Verarbeitung abgeschlossen hat. Anschließend liest der Contract-Test die vom Converter erzeugten Data-Events aus der Event-Queue. Er vergleicht die erstellten Data-Events mit den im Contract definierten Daten. Der Converter wird als JUnit-Test [5] gestartet.

So wird der gesamte Zyklus der Verarbeitung durch den Contract-Test überprüft. Die Formulierung des Contracts wird im Folgenden exemplarisch dargestellt (siehe Listing 2).

Das Element inputMessage (siehe Listing 2) enthält die Notification. Der Con-

```
outputMessage:
  - jmsSelector: "CoreCorrelationId = '1234' AND MainBusinessOpera-
tion = 'PERSONFUNCTION'"
    messageXml:
      class: "personfunction.PersonFunctionEvent"
        <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <core:personFunctionEvent</pre>
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <EventHeader>
             <CreatedAt>${isDate()}</CreatedAt>
             <MessageId>${isNumber()}</MessageId>
             <MandantSL>001</MandantSL>
         <MainBusinessOperation>PERSONFUNCTION</MainBusinessOpera-</pre>
tion>
    <BusinessTransactionTime>
                 2021-05-13T12:02:03.569+02:00
             </BusinessTransactionTime>
             <PersonFunctionId>
    <GrantorPartyId>12345678911</GrantorPartyId>
                 <HolderPartyId>47112341234/HolderPartyId>
             </PersonFunctionId>
         </EventHeader>
         < EventBody>
         </core:EventBody>
        </core:personFunctionUpdateEvent>
```

Listing 4: Contract – Definition der erzeugten Data-Events (vereinfacht)

tract-Test schreibt diese Notification in eine Queue. Für den Test wird ActiveMQ [7] eingesetzt. Der Converter liest aus dieser Queue die Notification. In einem nächsten Schritt liest er die Daten aus der Kundenverwaltung. Diese Kommunikation erfolgt über REST.

Der REST-Request wird im Contract-Test (siehe Listing 3) gegen einen MockWeb-Server [8] gerichtet. Dieser erstellt einen REST-Service, der über localhost erreichbar ist. Der Contract-Test konfiguriert den MockWebServer, indem er den Request-Path und die Response aus dem Attribut Endpoint des Contracts verwendet.

Schließlich enthält der Contract die Events, die in Folge der angegebenen Notification und der Daten im Endpoint durch den Converter erzeugt werden sollen.

Der Contract-Test liest die Nachricht anhand des Selektors (siehe Attribut jms-Selector in Listing 4) aus der Queue und vergleicht die Nachricht mit der Erwartung, die im Attribut text des Contracts angegeben ist. Für den Vergleich verwendet der Contract-Test XMLUnit [9]. Dieses Packet vergleicht zwei XML-Do-

kumente, wobei die Formatierung der XML-Daten keine Rolle spielt. Differenzen werden auf der Ebene von Element und Attribut ausgegeben.

3 Praxiserfahrungen

Der Converter versorgt seit nunmehr zwei Jahren zuverlässig mehr als 20 Umsysteme mit Data-Events. Der verwendete Technologie-Stack, bestehend aus Java mit dem Spring-Framework in OpenShift, hat sich bewährt. Der eingesetzte Message-Broker IBM-MQ hat bei einer sehr hohen Belastung erhebliche Performance-Probleme beim Einsatz von Selektoren verursacht. Daher arbeitet der Converter nicht mehr mit Selektoren, sondern konsumiert mit einem Listener alle Arten von Nachrichten aus der Notification-Queue.

Danksagung

Gemeinsam mit unserem Kunden arbeiten wir agil nach Scrum. Das Entwicklerteam

besteht aus drei Entwicklern. Das hier präsentierte Ergebnis ist in diesem Team erarbeitet worden. Die Entwicklung und das Deployment wurden durch die Infrastruktur der Bank erheblich vereinfacht.

Quellen

- [1] Eric Evans: Domain Driven Design: Tackling Complexity in the Heart of Software, Massachusetts 2003.
- [2] Event-Driven SOA: https://en.wikipedia. org/wiki/Event-driven_SOA
- [3] Open-Closed-Principle: https://en.wikipedia.org/wiki/Open%E2%80%93closed_ principle
- [4] Domain-Specific-Language: https:// en.wikipedia.org/wiki/Domain-specific_ language
- [5] JUnit: https://junit.org/junit5/
- [6] Yaml: https://de.wikipedia.org/wiki/ YAML#:~:text=YAML%20%5B%CB%-88j%C3%A6m%C9%99I%5D%20ist%20 eine%20maschinenlesbare,vorgestellten%20E%2DMail%2DFormat.
- [7] ActiveMQ: https://activemq.apache.org/
- [8] MockWebServer: https://github.com/ square/okhttp/tree/master/mockwebserver
- [9] XMLUnit: https://www.xmlunit.org/
- [10] Spring-Framework: https://spring.io/projects/spring-framework

Über den Autor

Dr. Stefan Koch ist als Principal Consultant bei der ORDIX AG beschäftigt. Als Bereichsleiter für Application Development betreut er eine Vielzahl der Entwicklungsprozesse des Unternehmens. Neben seiner Referententätigkeit im Seminarzentrum der ORDIX AG ist er auch auf Konferenzen als Speaker tätig.



Dr. Stefan Koch info@ordix.de



Oracle 23c für Entwickler

Rainer Willems, Oracle Deutschland

Das Release 23c der Oracle-Datenbank steht unter dem Motto App Simple. Dahinter steckt der Fokus des Releases, welches neben Verbesserungen für den Betrieb, insbesondere dem Entwickler helfen soll, einfacher und schneller seine Anwendungen entwickeln zu können. Dieser Artikel beschränkt sich auf genau dieses Thema – also die Entwickler – und erläutert einige der Neuerungen des Releases.

SQL-Erweiterungen

Im Release 23c gibt es einige Neuerungen im SQL. Angefangen damit, dass man statt FROM DUAL zu schreiben, nun die FROM-Clause auch weglassen kann, bis zu wirklich großen neuen Features. Die Erstellung, Änderung und Löschung von Objekten unterstützt jetzt die Syntaxmodifikatoren IF EXISTS und IF NOT **EXISTS**. Objekte werden eben nur angelegt (beziehungsweise gelöscht), wenn diese nicht (oder eben doch) existieren. Aber aufgepasst, Fehlermeldungen sind nicht generell schlecht. Erstellt man eine Tabelle mit einer Klausel, die es bereits gibt, wirft dies keine Fehlermeldung, auch wenn die Tabellenstruktur im Statement nichts mit der in der Datenbank zu tun haben muss (siehe Listing 1).

Ebenso wird bei einem Typo im Objekt-Namen bei DROP <object_type> IF EXISTS die Löschmeldung erfolgen, auch wenn das richtig geschriebene Objekt noch da ist, Trotzdem dürfte dieses Feature - gezielt eingesetzt hilfreich sein.

Die SQL-Engine der Datenbank unterstützt jetzt eine VALUES-Klausel für SE-LECT-, INSERT- und MERGE-Anweisungen. Diese neue Klausel ermöglicht es, Datenzeilen im laufenden Betrieb zu materialisieren, ohne auf bestehende Tabellen zurückzugreifen (siehe Listing 2). Hierdurch werden Skripte übersichtlicher und mittels der WITH-Klausel lassen sich SQL-Statements auch einmal ohne die Materialisierung von Daten testen (siehe Listing 3).

Wir haben nicht mehr so richtig daran geglaubt, aber jetzt ist er doch da: der BOOLSCHE Datentyp nach ISO-SQL-Standard. Analog zu den anderen Datentypen gibt es auch für diesen eine Konvertierungsfunktion (TO_BOOLEAN), aber die implizite Konvertierung ist auch breit aufgestellt. So akzeptiert der Datentyp neben true/false als native Werte auch t/f, yes/no, y/n, on/off als Text, wobei die Groß-/Kleinschreibung hier keine Rolle spielt. 1/0 werden sowohl als Text als auch als numerische Werte erlaubt.

Es dürfte viele freuen, dass man nun im GROUP BY und in der HAVING-Klausel den Alias-Namen einer Ausgabespalte verwenden kann, denn das macht die Statements doch deutlich besser lesbar und vermeidet viel Copy & Paste (siehe Listing 4). Nicht übersichtlicher, aber noch kürzer wäre es, ähnlich wie im ORDER BY, zumindest für das GROUP BY, die Position der Ausgabespalte zu verwenden. Auch dies geht, dafür muss aber der Parameter group_by_position_enabled auf TRUE gesetzt werden. Aufgrund der Möglichkeit der Verwendung von Alias-Namen erscheint mir dieser Weg aber gar nicht notwendig.

Ohne den Umweg über Subqueries gibt es nun die Möglichkeit, mittels Joins, Zeilen einer Tabelle in Abhängigkeit von anderen Tabellen zu löschen oder zu ändern (siehe Listing 5).

Zusätzlich zu einem DEFAULT ON NULL [FOR INSERT] gibt es nun auch ein **DEFAULT ON NULL FOR INSERT AND UPDATE**. Damit kann verhindert werden, dass ein bestehender Wert, der gegebenenfalls durch den Default beim Einfügen erzeugt wurde, nachträglich auf Null gesetzt wird. Damit benötigt man weder Anwendungscode noch einen Datenbank-Trigger, um dieses Verhalten zu erzielen.

Zeitbezogene Datentypen (DATE, TIMESTAMP & INTERVAL) konnten bisher (jeweils in Abhängikeit eines Parameters für die Stellen) gerundet (ROUND) oder abgeschnitten (TRUNC) werden. Nun können auch die Funktionen FLOOR & CEIL hier angewendet werden.

Mit 23c können INTERVAL-Datentypen an die Aggregatfunktionen SUM und AVG übergeben werden, was die Berechnung von Summen und Durchschnittswerten über INTERVAL-Werte deutlich vereinfacht (siehe Listing 6).

Eine Domain ist ein neues Dictionary-Objekt, dass einen Satz optionaler Eigenschaften und Einschränkungen für Datentypen wie beispielsweise Kreditkartennummern oder E-Mail-Adressen kapselt. Nachdem eine Domäne definiert ist, können Tabellenspalten diese referenzieren. Das erleichtert die konsistente Behandlung von gleichartigen Daten in unterschiedlichen Tabellen. Die Eigenschaften beziehen sich neben dem Datentyp auf Constraints sowie die Anzeige und Sortierung dieser Daten (siehe Listing 7).

Mittels der Funktion DOMAIN_CHECK können Werte auch vorher gegen eine Domaine geprüft werden. Es werden

```
SQL> CREATE TABLE myt(id number, col2 number);
Table created.
SQL> CREATE TABLE IF NOT EXISTS myt(id number, datacol date);
Table created.
/* In der DB gibt es nun eine Tabelle mit col2 als Number-Spalte */
```

Listing 1: IF [NOT] EXISTS

```
SQL> INSERT INTO dept (deptno, dname, loc) VALUES
    (10, 'ACCOUNTING', 'NEW YORK'),
    (20, 'RESEARCH', 'DALLAS'),
    (30, 'SALES', 'CHICAGO'),
    (40, 'OPERATIONS', 'BOSTON');
4 rows created.
```

Listing 2: TABLE-VALUES-Konstruktor

```
SQL> WITH virtual tab (id, value, description) AS
       (VALUES (1,100,'Text a'),
               (2,50,'Text b'),
               (3,80,'Text c'),
               (4,140,'Text d')
       ) SELECT max(value) FROM virtual tab;
MAX (VALUE)
       140
```

Listing 3: WITH-Klausel mit TABLE-VALUES-Konstruktor

```
SQL> SELECT initcap(dname)||'-('||initcap(loc)||')' AS depart,
           count(*) AS num of employees,
           sum(sal+nvl(comm,0))AS sum income
    FROM dept, emp
    WHERE emp.deptno = dept.deptno
    GROUP BY depart
    HAVING num_of_employees > 3
    AND sum_income < 11000;
```

Listing 4: Alias Namen in GROUP BY & HAVING

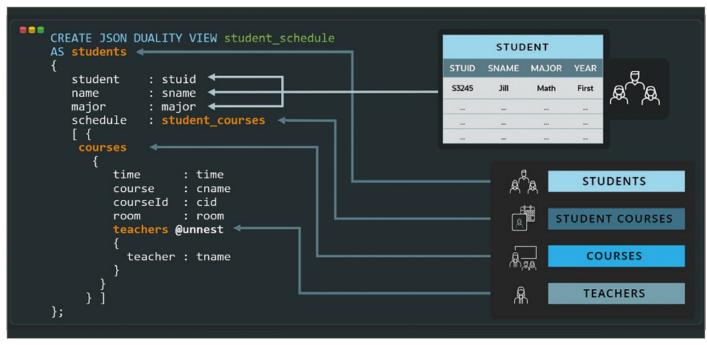


Abbildung 1: JSON Relational Duality View mittels GraphQL (Quelle: Oracle)

eine Reihe vordefinierter DOMAINs mitgeliefert.

Eine weitere beschreibende Möglichkeit von Daten bieten Annotations. Mit Hilfe von Annotation können Metadaten über Datenbankobjekte gespeichert und abgerufen werden. Hierbei handelt es sich (auf Tabellen und Spalten-Ebene) um Name-Werte-Paare oder einfach nur Bezeichnungen in Form von Freitext. Diese können aber von Anwendungen herangezogen werden, um beispielsweise die Darstellung der Daten anzupassen. So könnten Spalten mit der Annotation (Classification ,Sensitive') grundsätzlich maskiert angezeigt werden. Wie bei den Domains helfen Annotations dabei, Daten konsistent in der gleichen Weise zu verwenden (siehe Listing 8),

Limitanhebungen

Insbesondere bei ML- oder IoT-Anwendungen kann man mal schnell an das Limit von 1.000 Spalten pro Tabelle stoßen. Ähnlich wie bei der Erweiterung der Länge des VARCHAR2-Datentypes mittels des Parameters MAX_STRING_SIZE, kann nun die mögliche Spaltenanzahl von Tabellen mittels des Setzens von MAX_COLUMNS auf den Wert EXTENDED auf 4.096 angehoben werden. Ebenso angehoben wurde die maximale Größe von LOBs, damit sie noch inline mit

```
/* aus */
DELETE emp
WHERE emp.deptno= ( SELECT deptno from DEPT WHERE loc = 'DALLAS');
/* wird */
DELETE emp
FROM dept
WHERE emp.deptno = dept.deptno
AND loc = 'DALLAS';
```

Listing 5: Direkte Updates/Deletes über Joins

der Tabelle gespeichert werden können. Statt 4.000 Bytes sind nun 8.000 Bytes inline möglich.

JSON

Wieder gibt es Neuerungen im Umgang mit JSON-Dokumenten. Bereits in 21c wurde ein nativer Datentyp eingeführt. In 23c wird es nun JSON-Schemas geben und die Funktion JSON_TRANSFORM wurde deutlich erweitert. Diese und andere Verbesserungen im Umgang mit JSON-Dokumenten werden aber überstrahlt von einem der wichtigsten neuen Features in 23c: den JSON Relational **Duality Views**. Das Ergebnis der neuen Views hätte man auch mit Vorgängerversionen erreichen können, dazu wäre aber viel manuelles Coding mit PL/SQL notwendig gewesen, während JSON Relational Duality Views dafür nur ein wenig GraphQL-Syntax benötigen (siehe Abbildung 1). Aber was machen diese Views denn eigentlich? JSON Relational Duality Views sind vollständig aktualisierbare JSON-Ansichten über relationalen Daten. Die Daten werden nach wie vor in relationalen Tabellen gespeichert, können aber von Anwendungen in Form von JSON-Dokumenten abgerufen werden. Duality Views bieten eine bahnbrechende Einfachheit, mit Daten sowohl relational als auch mittels der JSON-Dokumente arbeiten zu können. Einmal definiert, können sowohl SQL- als auch JSON-Entwickler mit den gleichen Daten in der jeweils gewohnten Form arbeiten. Weniger Daten speichern, weniger Inkonsitenzen und weniger Datentransfers sind die angenehmen Folgen. So schreibt die IDC dazu: "Oracle's JSON Relational Duality, a truly revolutionary solution, is perhaps one of the most important innovations in information science in 20 years." [1]

```
CREATE TABLE Processes
  ( id
              NUMBER GENERATED ALWAYS AS IDENTITY,
   ptitle VARCHAR2(30),
   pstart TIMESTAMP,
   pend
             TIMESTAMP,
    pduration INTERVAL day to second GENERATED ALWAYS AS (pend - pstart) VIRTUAL);
Table PROCESSES created.
INSERT INTO processes (ptitle, pstart, pend) VALUES
    ('Process A', trunc(sysdate) +540/1440, trunc(sysdate) +1080/1440),
    ('Process B', trunc(sysdate)+640/1440, trunc(sysdate)+1000/1440),
    ('Process C', trunc(sysdate)+800/1440, trunc(sysdate)+910/1440),
    ('Process D', trunc(sysdate) +800/1440, trunc(sysdate) +1200/1440),
    ('Process E', trunc(sysdate) +840/1440, trunc(sysdate) +1080/1440);
5 rows inserted.
SELECT sum(pduration), avg(pduration) FROM processes;
SUM (PDURATION) AVG (PDURATION)
+01 03:30:00.000000 +00 05:30:00.000000
```

Listing 6: Aggregation von INTERVAL-Datentypen

```
CREATE DOMAIN my_email_domain AS VARCHAR2(100)
  CONSTRAINT email address check CHECK (regexp like (my email domain, '^(\S+)\@(\S+)\.(\S+)\*!))
     DISPLAY 'EMAIL : '||lower(my email domain)
        ORDER lower(my email domain);
CREATE TABLE MyColleagues
   (id NUMBER GENERATED ALWAYS AS IDENTITY,
    email DOMAIN my email domain );
SELECT domain_display(email)
 FROM MyColleagues
ORDER BY domain_order(email);
```

Listing 7: SQL Domains

Property-Graphen

Die Datenbank bietet mit 23c eine native Unterstützung für Property-Graphen und Graphenabfragen. Property-Graphen bieten eine intuitive Möglichkeit, direkte oder indirekte Abhängigkeiten in Datenelementen zu finden und Erkenntnisse aus diesen Beziehungen zu gewinnen. Es lassen sich nun solche Graphen auf Basis der relationalen Daten definieren, um darauf graphenspezifische Abfragen laufen zu lassen. Zum einen ist hierfür dann kein Datentransfer in eine Spezial-Graphen- Datenbank notwendig, zum anderen lassen sich beispielsweise auch JSON-Dokumente oder geographische Informationen aus der Oracle-Datenbank mitverarbeiten. Die Abfrage aus Listing 9 würde rein relational umsgesetzt viele Joins und Unions benötigen.

Developer Role & Schema-Privilegien - vereinfachte Rechtevergabe

Mit der DB_DEVOLOPER Rolle wurde eine Rolle eingeführt, die einen Entwickler möglichst umfassend mit Entwicklungsrechten versorgen soll. Nicht nur

die CREATE-System-Privilegien werden hier auf weitere Objekte wie beispielsweise die neuen Domains ausgeweitet, auch EXECUTE-Rechte für JavaScript oder verschiedene Packages und andere Rechte sind hier gebündelt. Ebenfalls vereinfacht wurde die Rechtevergabe von einem Schema zu einem anderen. Musste man bisher Objekte einzeln granten, kann dies nun pauschal für Objekttypen eines Schemas gemacht werden, was dann praktischerweise auch für später hinzukommende Objekte gilt. So gibt es beispielsweise ein GRANT

```
CREATE TABLE EMP2

(empno NUMBER(4) ANNOTATIONS(Key, Full_Name 'Employee ID'),
ename VARCHAR2(10) ANNOTATIONS(Full_Name 'Emp Last Name'),
job VARCHAR2(9),
sal NUMBER(7,2) ANNOTATIONS(Class 'Sensitive Info'),
deptno NUMBER(2,0)
)
ANNOTATIONS (Full_Name 'Employees')
```

Listing 8: Annotations

```
/* Definition eines Graphen mit Punkten (VERTEX TABLES) und
   Verbindungen (EDGE TABLES) dazwischen */
CREATE OR REPLACE PROPERTY GRAPH bank graph
   VERTEX TABLES (
     bank accounts KEY (ID)
      PROPERTIES (ID, Name, Balance)
   )
   EDGE TABLES (
     bank transfers KEY (TXN ID)
      SOURCE KEY (src_ac_id) REFERENCES bank_accounts(ID)
      DESTINATION KEY (dst ac id) REFERENCES bank accounts (ID)
      PROPERTIES (src acct id, dst acct id, amount)
   );
/* Ausgabe von Accounts, die Zahlungskette in einem Kreis
   mit 3 bis 5 Hops aufweisen */
SELECT DISTINCT (account id)
  FROM GRAPH TABLE ( BANK GRAPH
       MATCH (v1)-[IS BANK TRANSFERS]->{3,5}(v1)
         COLUMNS (v1.id AS account id)
   );
```

Listing 9: Definition & Abfrage eines Graphen

Listing 10: Ubiquitous Database Search

[SELECT|INSERT|UPDATE|DELETE] ANY TABLE ON SCHEMA s1 TO s2. Eben ein solches gibt es für das Ausführen von prozeduralen Objekten.

Transaktionsverhalten

Das Transaktionsverhalten der Datenbank bleibt unverändert, das hat sich ja durchaus bewährt. Aber mit zwei neuen Features kann man besseren Einfluss darauf nehmen. Durch das erste Feature können Prioritäten für Transaktionen vergeben werden. Hält nun eine niedrige priorisierte Transaktion ein Lock, auf welches eine höherpriorisierte Transaktion wartet, kann diese automatisiert nach einer bestimmten Zeit abgebrochen werden. Hierzu ist dann keine DBA-Interaktion mehr notwendig. Beim zweiten Feature geht es um Spalten, auf denen eine hohe Änderungslast liegt. Und zwar handelt es sich um numerische Spalten, die einer Bedingung (CHECK Constraint) unterliegen (beispielsweise sollte ein Kino ja nicht mehr Tickets verkaufen, als es Plätze hat). Solche Spalten können nun als RESERVABLE definiert werden und damit wird ein spezielles Transaktionskonzept für diese Spalten verwendet. Ändert man eine solche Spalte und legt beispielsweise zwei Kinotickets in den Warenkorb, dann wird die Zeile nicht gelockt, sondern die Reservierung wird in einem Journal festgehalten. Andere Cineasten können also ungestört ebenfalls Tickets in den Warenkorb legen. Erst beim COMMIT wird das Journal in die Tabelle übertragen. Entscheidend ist aber, dass bei der Prüfung des CHECK-Constraint die vorhandenen Reservierungen bereits mitgeprüft werden. Sind also noch 10 Tickets vorhanden und eine Session reserviert sich 6 davon, wird eine zweite Session, die mehr als 4 Tickets versucht zu reservieren, am CHECK-Constraint scheitern. Ein Verhalten, dass bisher applikationsseitig umgesetzt werden musste.

Textsuche in der Datenbank

Oracle Text ist seit vielen Jahren integraler Bestandteil der Oracle-Datenbank.
Das neue DBMS_SEARCH PL/SQL-Paket

(Ubiquitous Database Search) ermöglicht nun die Indizierung von mehreren Schemaobjekten in einem einzigen Index. Alle Spalten in den angegebenen Quellen werden indiziert und stehen für eine Volltextsuche zur Verfügung (siehe Listing 10). Mit 23c werden zusätzlich zwei neue Operatoren für den Abgleich von Zeichenketten eingeführt, die auf dem approximativen oder unscharfen Abgleich von Zeichenketten basieren. PHONIC_ENCODE konvertiert Wörter oder Phrasen in sprachspezifische Codes auf der Grundlage der Aussprache und FUZZY_MATCH, das sprachneutral ist, misst die Textähnlichkeit zwischen zwei Zeichenfolgen auf Basis verschiedener zur Verfügung stehender Algorithmen.

Weiteres

Danke der Multilingual Engine (MLE) in der Datenbank ist es nun möglich, Java-Script direkt in der Datenbank laufen zu lassen. Dies soll nicht unbedingt PL/ SQL ersetzen, gibt aber insbesondere die Möglichkeit, vorhandene Bibliotheken in die Datenbank zu laden und dort mit einem PL/SQL API zu versehen. Warum etwas neu entwicklen, wenn es das doch schon gibt. Das Konzept basiert auf der GraalVM und wir können gespannt sein, ob und wenn ja, welche Sprache hier folgen werden.

Der Kontext-Switch zwischen der SQL- und der PL/SQL-Engine kann bei massenhaftem Auftreten durchaus nachteilig für die Performance sein. Der **SQL Transpiler,** der mit dem Parameter sql_transpiler aktiviert werden kann, schaut, ob SQL-Statements mit Funktionen-Aufrufen derart umgeschrieben werden können, dass bei der Ausführung der Funktionsaufruf gar nicht durchgeführt wird, sondern der Inhalt der Funktion in das SQL übertragen wird. Ähnliches kann man auch selbst steuern, in dem man Funktionen als skalare SQL Macros schreibt. Faszinierende Möglichkeiten der Modularisierung von komplexeren Aufgaben bieten die tabellarischen SQL Macros, worauf hier aber nicht eingegangen wird, da es einen eigenen Artikel benötigen würde, um diesen gerecht zu werden.

Die in der Oracle-Datenbank enthaltene SQL-Firewall, die nicht umgangen werden kann, prüft alle eingehenden SQL-Anweisungen und stellt sicher, dass nur ausdrücklich autorisiertes SQL ausgeführt wird. Hierzu gehören auch die Verbindungspfade. Nicht autorisiertes SQL kann protokolliert und blockiert werden.

Transactional Event Queues unterstützen jetzt die Klassen KafkaProducer und KafkaConsumer von Apache Kafka, somit kann die Datenbank als Quelle und Ziel für Anwendungen verwendet werden, die die Kafka-APIs nutzen.

Die Oracle Saga-APIs in der Datenbank bieten ein Framework zur Implementierung einer transaktionalen Semantik für Microservices, die mit der Oracle-Datenbank aufgebaut sind.

Es gäbe noch so viel mehr zu schreiben, aber hoffentlich konnte hier ein guter Überblick über die wichtigsten Neuerung gegeben werden.

Quellen

[1] https://www.idc.com/getdoc.jsp?containerld=US49859922

Über den Autor

Rainer Willems arbeitet seit vielen Jahren als Solution Engineer bei Oracle und interessiert sich für datenbanknahe Applikationsentwicklung.



Rainer Willems rainer.willems@oracle.com



DIE DOAG ANWENDERKONFERENZ.DOAG.ORG ANWENDERKONFERENZ.DOAG.ORG

SAVE THE DATE

19. BIS 22. NOVEMBERG IN NÜRNBERG



10 Jahre APEX connect!

Niels de Bruijn, Hyand (ehemals MT), DOAG Themenverantwortlicher APEX connect

Es fing alles am 08.04.2013 an. Oliver Lemm und ich waren in Berlin für ein Expertenseminar, natürlich zum Thema APEX. Mit Stefan Kinnen, damals verantwortlich für die Development Community, hatten wir uns zum Abendessen verabredet. Irgendwann haben wir über eine APEX-Konferenz gesprochen und Stefan fragte, ob wir glauben, dass wir genügend Teilnehmer:innen hierfür bekommen würden. Die Antwort sollte bekannt sein :)

Damals gab es noch die Development-Konferenz mit viel mehr Themen als nur APEX. Auch im Jahr 2014 fand diese Konferenz wieder statt, aber es wurde schnell klar, dass die Vorträge und das Publikum mehrheitlich auf APEX fokussiert waren. Die Zeit für eine APEX-Konferenz war gekommen. Was fehlte, war zuerst ein Name. Nach einer Brainstorming-Session kam "APEX connect" dabei heraus, bis heute aus meiner Sicht keine schlechte Wahl (siehe Abbildung 1).

Nachdem das Programm dann stand, war es am 09.06.2015 so weit: Die APEX connect fand zum ersten Mal statt, und zwar im Melia Hotel in Düsseldorf (wer kann sich noch daran erinnern?). Der Erfolg war da: Wir waren ausgebucht, hatten jede Menge Spaß und dazu gutes Feedback (siehe Abbildung 2 und 3). APEX connect is here to stay!

Mit den Jahren haben wir stets versucht, nicht nur die Veranstaltungsorte zu wechseln, sondern wir haben uns auch zunehmend professionalisiert. Kritik gab es anfangs auch. So gab es Referent:innen, die nicht verstanden hatten, warum deren:dessen Abstract abgelehnt wurde. Es gab offenbar zu wenig Transparenz und dies führte zu Irritationen. Es gab sogar einen anonymen Blogpost, um Dampf abzulassen. Feedback haben wir im Konferenzteam immer angenommen und so wurde sowohl das Konferenzteam als auch die Bewerter-Runde größer und es wurde damals eine anonyme Bewertung

Summit - Curape - World - con do - inside - improve yourself - devolopment - u u - 4all - gat intouch with ASEX - enjoy APEX - nort local - round about - dixer - horum APEX?	- DPER only - discover - discover - discover - discover - lourninge - inside out - hetwork - explore - compect - tagether - lotal - robet - online - mind - open - nevs - time - now - koday	-present -Future - improvement - intelligence - Social - Solutions - now x tomorrow - coding - hands-on - enhance - simply - pure - web - advanced - cammunity - share
---	--	---

Abbildung 1: Ergebnis des Brainstormings (Quelle: Niels de Bruijn)

für Abstracts eingeführt. Wer sich für den Bewertungsprozess von Abstracts der APEX- connect-Konferenz im Detail interessiert, kann es hier nachlesen [1]. Der Community-Gedanke stand und steht nicht nur für mich immer an erster Stelle, daher kann ich nur dazu ermutigen, sich beim Konferenz-Team (siehe Abbildung 4) zu melden, wenn etwas ist.

Was die Teilnehmerzahlen angeht, haben wir uns immer zwischen 250 und 400 bewegt. Die APEX connect bleibt dadurch

familiär und dies ist auch gut so. Dabei gibt es viele APEX-Interessierte, die gerne jedes Jahr wieder zur Konferenz kommen. In diesem Zusammenhang erwähnenswert ist die Online-Ausgabe von 2020: Obwohl es die Konferenz Corona-bedingt online und kostenfrei gab, traf man wieder auf das gleiche Publikum wie in den Jahren zuvor. Erkenntnisse: die APEX-Familie hält zusammen, aber auch: Was die Teilnehmer:innen angeht, bleibt die APEX connect eine Konferenz für deutschspra-

Willkommen zu #apexconn15

APEX connect 2015



Niels de Bruijn



Carsten Czarski



Carmen Al-Youssef



Abbildung 3: Übungen (Quelle: Niels de Bruijn)



Abbildung 4: Das Konferenz-Team (Quelle: Ronny Weiß)



Abbildung 5: Verein Mutige Kinder (Quelle: Niels de Bruijn)



Abbildung 6: Verein Freie Ukraine Braunschweig e. V. (Quelle: Niels de Bruijn)



Abbildung 7: Oxfam-Team (Quelle: Marcos López)



Abbildung 8: UNICEF (Quelle: Niels de Bruijn)



Abbildung 9: Apollo-Theater (Quelle: Niels de Bruijn)

chiges Publikum. Mittlerweile gibt es auch ein gutes Angebot an regionalen Konferenzen in Europa und darüber hinaus, die jeweils einen eigenen Character haben. Ob Kscope (USA), APEX World (NL) oder Alpe Adria (AS) - alle sind für APEX-Entwickler:innen einen Besuch wert.

Bei der APEX connect nutzen wir die Bühne, um auch auf Themen in der Welt aufmerksam zu machen, die es verdienen: Klimakrise, Armut, Kriege und radikale Bewegungen betreffen uns alle. Wir müssen uns dieser Themen stets bewusst sein und vorbildlich handeln. Auf der APEX connect kann man mit Vertreter:innen eines gemeinnützigen Vereins in Kontakt treten und deren vorbildliche Arbeit unterstützen. Dies sind unsere wahre Held:innen (siehe Abbildung 5 bis 8).

Mit der 10. Ausgabe werden wir zum zweiten Mal im Van der Valk Hotel sein, aber auch diesmal gibt es einige Veränderungen: am ersten Tag gibt es ein Bootcamp für APEX-Anfänger und parallel dazu Vorträge über Kundenprojekte. Für das Abendprogramm geht es wieder zur Dinnershow von Roncalli direkt am Rhein. Besonders freue ich mich auf die Keynote am zweiten Tag von Mike Hichwa, dem Erfinder von APEX. Auch auf die Vorträge von Kris Rice dürfen wir uns freuen, er ist unter anderem für ORDS und SQL Developer (Web) zuständig und kann uns viel über die Implementierung der Oracle Cloud erzählen. Beide Oracle-Kollegen

kommen extra für uns aus den USA angereist.

Ich freue mich darauf die APEX-Familie am 22. April 2024 zum 10. Mal begrüßen zu dürfen (siehe Abbildung 9). Oder wie Joel immer sagte: #letswreckthistogether!

Veranstaltungsorte:

2015: Melia Hotel, Düsseldorf 2016: nhow Berlin Hotel, Berlin 2017: nhow Berlin Hotel, Berlin 2018: Van der Valk Hotel, Düsseldorf

2019: Kameha Hotel, Bonn

2020: Online 2021: Online

2022: Phantasialand, Brühl 2023: nhow Berlin Hotel, Berlin 2024: Van der Valk Hotel, Düsseldorf

2025: Europapark, Rust

nect, Kscope und APEX World. Niels gilt als Initiator der Konferenz DOAG APEX connect (apex.doag.org) und ist hier als Programmleiter aktiv. Die enge Zusammenarbeit mit dem APEX-Team, sowie seine Beiträge für die Oracle Community blieben bei Oracle nicht unbemerkt und führten im Jahr 2016 zu seiner Benennung "Oracle ACE Director".

Seitdem es das Produkt Oracle Appli-

cation Express (APEX) gibt, hat er zusam-

men mit seinem Team hunderte APEX-

Projekte hinter sich gebracht. Niels teilt

sein Wissen über die Webseite apex.mt-

itsolutions.com, sowie über diverse Vor-

träge auf Veranstaltungen wie APEX con-

Quellen

[1] https://nielsdebr.blogspot.com/2019/01/ abstract-selection-process-for-apex.html

Über den Autor

Niels de Bruijn arbeitet seit 2003 für die Hyand (ehemals MT), ein mittelständisches IT- Dienstleistungsunternehmen mit Hauptsitz in Ratingen.



Niels de Bruijn niels.debruijn@hyand.com



Highway to CloudLand: Die DOAG-Pizza-Bier-Party-Cloud-Meetup-Roadshow

Marcos López

Im Januar ging die Landingpage des Projektes *Highway to CloudLand* h2c.doag.org offiziell online. Seit April nimmt die vor Jahren im stillen Kämmerlein angedachte Roadshow der DOAG Cloud Native Community (DCNC) im ,real life' Fahrt auf.



Ganz nüchtern betrachtet ist H2C nur eine Bustour durch Deutschland, die an bestimmten Stationen Halt machen wird, damit Menschen sich begegnen können. Genauer betrachtet ist H2C aber eine Bustour der ungewöhnlichen Art, denn der große DOAG-Tourbus steuert User Groups und Meetups in der ganzen Republik an. So können sich dann im und um das Gefährt herum Entwicklerinnen und Entickwickler kennenlernen, über die Themen Cloud, KI und Digitalisierung ins Gespräch kom-



men sowie spannenden Impulsvorträgen lauschen.

Der Clou: Der DOAG-Tourbus ist kein gewöhnlicher Gelenkbus, der mit 18 Metern Länge, feuerroter Folierung und dem DOAG-Maskottchen an sich schon ein Eyecatcher ist. Der DOAG-Tourbus ist ein richtiger Partybus, der mit seiner speziellen Innenausstattung, der Beleuchtung und dem Soundsystem ein richtiges Vortrags-Happening ermöglicht, das von frischgebackener Pizza (im Bus zubereitet) und kühlem Bier (im Bus bereitgestellt) im Anschluss noch getoppt wird.

SO BUCHST DU EIN TICKET FÜR DEINE STATION

- Finde deine Stadt im Tourkalender von h2c.doag.org!
- **☑** Buche dir eines der limitierten Tickets im H2C-Tourshop!
- ☑ Erzähle deinem Netzwerk davon!
- Freue dich auf erstklassige Informationen und neue Begegnungen!
- Freue dich auf knusprige Pizza und kühles Bier!



SO WIRST DU HOST BEIM H2C

- ☑ Du findest das Thema Cloud & Co. superspannend?
- **☑** Du organisierst ein Meetup oder eine User Group?
- ✓ Du leitest eine Firma oder ein Unternehmen?
- **☐** Du hast eine Idee für den Stellplatz des Tourbusses?
- ☑ Du kennst einen Speaker für den Impulsvortrag?
- ☑ Du stehst auf knusprige Pizza und kühles Bier?
- Du findest, 'ne Party im Bus ist 'ne coole Sache?
- □ Dann werde Host der H2C-Tour!
- Schreibe eine E-Mail an h2c@doag.org.
- Wir melden uns bei dir und klären den Rest.

STECKBRIEF DOAG-TOURBUS

Modell: MAN LION's CITY 18

Bauiahr: 2018

Kraftstoff: Diesel / Biodiesel / HVO

PS: 330 Länge: 18 m

Breite: 2,55 m Höhe: 3,06 m

Specials: Soundsystem, Lichtanlage, Pizzaofen, Kühlschrank, Good Vibrations

Natürlich auch im Bordbistro: eine Auswahl an alkoholfreien Getränken.

Warm-up, Impulsvortrag, Come Together Party

Es läuft so: Der Bus, der ausdrücklich nicht für die Personenbeförderung gedacht ist, macht Station beim Host des Events (Firma, Unternehmen, User Group, Meetup, Initiative) und präsentiert nach dem lockeren Warm-up einen Impulsvortrag (im Bus). Anschließend gibt es ein Cometogether in einem außergewöhnlichen Ambiente - Verköstigung, persönliche Begegnungen, Netzwerken und Know-how-Transfer inklusive (drinnen und draußen).

Die Idee dazu entstand in den oftmals einsamen Zeiten der Lockdowns, der Kontakt- und Veranstaltungsverbote. Über das Netzwerk der DCNC und der gelungenen Festivals CloudLand 2022 und 2023 nahm das Projekt dann mehr und mehr Gestalt an, bis sich im Winter 2023/2024 schließlich die einmalige Gelegenheit ergab, einen geeigneten Bus für dieses Vor-



Tourstopp Nürnberg 15.04.2024 17:00 - 21:00

Tourstopp in Nürnberg mit Frank Prechtl zum Thema Cloud-native mit AWS



Tourstopp Würzburg

16.04.2024 18:00 - 21:30

Tourstopp in Würzburg mit Björn Raupach



Tourstopp Ingolstadt

17.04.2024 17:30 - 21:00

Tourstopp in Ingolstadt mit Florian Heubeck



Tourstopp München

18.04.2024 17:00 - 21:00

Tourstopp in München bei Backstage



Tourstopp Rosenheim

19.04.2024 16:30 - 21:00

Tourstopp in Rosenheim (Ferchau)



Tourstopp Augsburg

22.04.2024 17:00 - 21:00

Tourstopp in Augsburg mit der JUG Augsburg



Tourstopp Göttingen

30.04.2024 17:00 - 21:00

Tourstopp in Göttingen bei xplace



Tourstopp Oldenburg

21.05.2024 17:00 - 21:00

Tourstopp in Oldenburg bei BTC



Frank Prechtel, AWS User Group Nürnberg, DCNC und CloudLand-Programmkomitee

"Mit dem Tourbus fördert die DOAG auf eine völlig neue Art und Weise den Austausch innerhalb der IT-Communities quer durchs Land. Die besten Gespräche, Ideen und Kooperationen entstehen ja bekanntlich in einer entspannten Atmosphäre an ungewöhnlichen Orten – der DOAG-Tourbus ist prädestiniert dafür."

Andreas Badelt, DCNC und CloudLand-Programmkomitee

"Ich habe schon einige coole Meeting Locations gesehen, aber ein Bus – erst recht so ein Hammer-Tourbus – war noch nicht dabei. Ich freue mich schon wie ein kleines Kind darauf, im Frühjahr von Gig zu Gig... Sorry, von Meetup zu Meetup zu reisen."





André Sept, DCNC und CloudLand-Programmkomitee

"Ich bin megastolz, was die Community-Mitglieder in den letzten Wochen auf die Beine gestellt haben. Mit Pizza, Getränken und guten Vorträgen bringen wir Meetups vor Ort in die Lage, eine tolle Veranstaltung für die Teilnehmenden zu bieten. Gleichzeitig stärken wir unser Netzwerk der DOAG Cloud Native Community (DCNC) und gewinnen neue Fans für das CloudLand-Festival."

haben zu erwerben und ihn für die beabsichtigte Roadshow zu modifizieren.

Mit der H2C-Tour präsentiert sich die DCNC nun einer breiteren Öffentlichkeit und erschließt neue Gruppen – und wie der Name der Tour schon sagt, weist man so ganz nebenbei auf das sommerliche CloudLand-Festival hin, das als das Finale Furioso dieser einmaligen Roadshow angesehen werden darf. Das viertägige Cloud Native Festival vom 18. bis 21. Juni, dessen Besucherzahlen von 250 zur Premiere auf 500 beim Follow-up gestiegen sind, findet bereits zum dritten Mal in der beliebten Freizeitparkkulisse des Phantasialand (Brühl) statt. Die Zahl der Einreichungen für die 2024er-Ausgabe der Cloud-Native-Konferenz legt schon jetzt

die Vermutung nahe, dass erneut mit einem Besucherrekord zu rechnen ist.

Auftakt am Nürburgring

Erstmals wird der DOAG-Tourbus auf der JavaLand 2024 am Nürburgring zu bewundern sein. Hier präsentiert sich das extravagante Gefährt während der Java-Konferenz vom 8. bis 10. April auf dem sogenannten ring°boulevard. Mitglieder des CloudLand-Programmkomitees und der DCNC stehen hier bereit, um Interessierten und Neugierigen alles über den "Highway to CloudLand" zu erzählen, über die jeweiligen Stationen des Busses zu informieren und auf das CloudLand-

Festival zu verweisen, wo die Tour offiziell endet. Es werden auch alle Fragen zur Roadshow beantwortet, zum Beispiel ob und wie man noch zum Host der H2C-Tour werden kann, wie man als Besucherin und Besucher ein kostenfreies, aber limitiertes Ticket ergattern kann und was es dabei zu beachten gilt.

Der offizielle Auftakt der H2C-Tour findet am 15. April in Nürnberg statt. Hier macht der Bus bei der AWS User Group der Frankenmetropole halt. Los geht's um 17 Uhr.

Im Web

[1] h2c.doag.org



"I like to move it, move it"oder: Wie kommt meine Oracle-Datenbank in die Oracle Cloud?

Kai-Uwe Fischer, SVA System Vertrieb Alexander

Die Public Cloud ist in aller Munde, die Vorteile wie Skalierbarkeit auf Abruf, flexible Abrechnungsmodelle, bessere Performance und keine Investitionskosten liegen auf der Hand. Warum also nicht auch die Oracle-Datenbank in die Oracle Cloud migrieren? Ein Datenbank-System, bestehend aus einer Oracle-Datenbank inklusive Grid Infrastructure, ist als virtuelle Maschine in circa 75 Minuten in der Oracle Cloud erstellt. Doch wie kommen die Daten beziehungsweise Datenbanken in die Cloud? In diesem Artikel stelle ich einige Migrationswege vor und zeige, welche Faktoren die Migration beeinflussen.

Oracle-Datenbank-Lösungen in der Oracle Cloud Infrastructure (OCI)

Oracle-Datenbank-Cloud-Lösungen stehen sowohl in der OCI als auch im Kunden-Rechenzentrum zur Verfügung. Es wird dabei zwischen den Semi-Managedund den Fully-Managed-Datenbanksystemen unterschieden (siehe Abbildung 1: Oracle-Datenbank-Cloud-Lösungen).

Zu den Semi-Managed-DB-Systemen zählen:

- Oracle Database Cloud Service Virtual Machines
- Oracle Database Cloud Service Bare Metal (Dieser Service steht nur noch Bestandskunden zur Verfügung)
- Oracle Database Cloud Service Exadata Cloud Service
- Oracle Database Cloud Service Exadata Cloud at Customer

Zu den Fully-Managed-DB-Systemen zählen:

- Oracle Autonomous Database -Shared Deployments
- Oracle Autonomous Database Dedicated Deployments
- Oracle Autonomous Database Exadata Cloud at Customer

Virtual Machine Database System Options

Seit längerem hat der Kunde bei den Virtual Machines die Wahl zwischen Intelund AMD-Prozessoren, wobei für beide Optionen die Möglichkeit besteht, eine Oracle-Datenbank als Single Instance oder als Real Application Cluster (RAC) zu betreiben. Seit Juni 2023 gibt es eine dritte Plattform (ARM) basierend auf den Single Thread CPUs von Ampere (siehe Abbildung 2: Prozessortypen bei Oracle Database VMs).

Die drei unterschiedlichen Prozessor-VMs haben drei Gemeinsamkeiten:

Mit der Erhöhung der Anzahl der CPU-Kerne, bei Oracle OCPU (Oracle Compute Unit) genannt, erhöht sich automatisch der Hauptspeicher und die Netzwerkbandbreite. Des Weiteren kann eine Storage-Vergrößerung im laufenden Betrieb erfolgen. Eine Verkleinerung ist nicht möglich. Eine Erhöhung beziehungsweise Verringerung der OCPUs ist möglich, bedarf aber einer Downtime.

Für ARM-basierte VMs gibt es im Oracle-Datenbank-Umfeld die folgenden Einschränkungen:

- Die Oracle-Datenbank-Version muss Version 19.19 oder höher sein.
- Es werden nur Single Instances unterstützt, kein RAC.
- Es wird keine Grid Infrastructure, sondern nur der Logical Volumne Manager unterstützt.
- Es wird nur die Oracle Enterprise Edition, nicht aber die Standard Edition unterstützt.

Oracle-Datenbank-Lizensierung in der Oracle Cloud Infrastructure

Grundsätzlich hat der Kunde die Wahl zwischen einem Subscription Model oder Bring Your Own Licence (BYOL). Unabhängig des gewählten Subscription-Models sind alle Oracle-Datenbanken, auch die Standard-Edition, per Default mit Transparent Data Encryption (TDE) verschlüsselt (siehe Abbildung 3: OCI-Datenbank-Lizenz-Bundles).

Anders als On-Premise muss bei der Enterprise Edition nicht jede Option oder jedes Management Pack einzeln lizensiert werden. Oracle bietet diese in den drei folgenden Varianten an:

- Enterprise Edition, hier sind bereits Features wie Data Masking & Subsetting, Diagnostic & Tuning Pack und Real Application Testing inkludiert.
- Enterprise Edition High Performance
- Enterprise Edition Extreme Performance, beinhaltet alle Oracle-Optionen und Management Packs.

Migrationswege in die Cloud

Eine Oracle-On-Premise-Datenbank kann mit verschiedenen Methoden und Tools in die Oracle Cloud Infrastructure migriert werden. Welche Methode sich für ein bestimmtes Migrationsszenario eignet, hängt von folgenden Faktoren ab:

Datenbank-Version

- Datenbank-Edition
- Datenbank-Zeichensatz
- Datenbank Block Size
- Datenbankgröße
- Verfügbarkeit der Datenbank während der Migration
- Datenbank-Endian-Format
- Verwendete Datenbank-Architektur
- Verwendete Data Types
- Netzwerkbandbreite

Bei einer Datenbankmigration wird grundsätzlich zwischen einer Physical Migration und einer Logical Migration unterschieden. Der Unterschied zwischen diesen beiden Migrationsarten lässt sich am einfachsten anhand einer einfachen Textdatei erklären.

Bei einer Physical Migration wird diese von der Quelle auf das neue Zielsystem kopiert, dabei ändert sich weder der Dateiname noch deren Inhalt. Dies kann auf Datenbankebene zum Beispiel durch das Übertragen eines RMAN-Backups auf das Zielsystem erfolgen. Beim anschließenden Restore der Datenbank werden die Original-Datenbankdateinamen wieder hergestellt.

Bei einer Logical Migration wird auf dem Zielsystem eine neue Textdatei angelegt, dabei kann der gleiche Dateiname verwendet werden, muss aber nicht. Des Weiteren kann entweder der gesamte Inhalt dieser Datei oder auch nur Teile davon kopiert und auf das Zielsystem übertragen werden. Aus Datenbank-Sicht können zum Beispiel mit Hilfe von Data Pump entweder die ganze Datenbank oder einzelne Schemata exportiert und auf dem Zielsystem eingespielt werden.

Bei der Migration auf eine Autonomous Database muss grundsätzlich die Logical Migration verwendet werden, da der User nur Zugriff auf eine Pluggable Database (PDB) und keinen Zugriff auf das Betriebssystem über die Kommandozeile erhält.

Folgende Werkzeuge können für diese Art der Migration verwendet werden:

- SQL Loader
- Data Pump
- SQL Developer
- Golden Gate
- Enterprise Manager Migration Workbench
- Zero Downtime Migration (ZDM)
- OCI Database Migration Service (DMS)

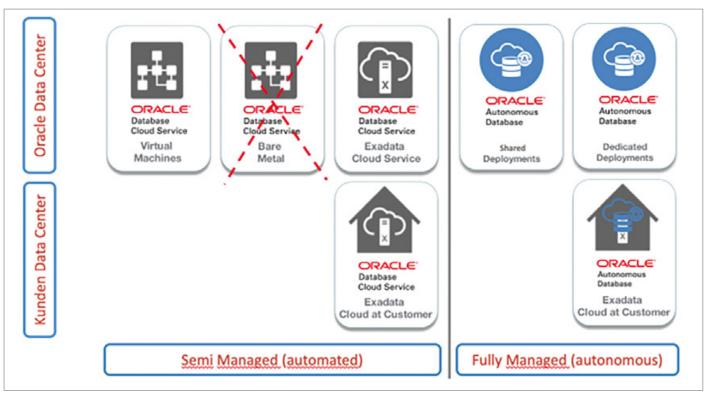


Abbildung 1: Oracle-Datenbank-Cloud-Lösungen (© Oracle)

INTEL Flex Shapes						
Platform	OCPUs	Memory	Storage	Network	Nodes	
X9 VM.Standard3.Flex	1 - 32	16 GB – 512 GB	256 GB - 80 TB	1 -32 Gbps	1 - 2	
AMD Flex Shapes						
Platform	OCPUs	Memory	Storage	Network	Nodes	
VM.Standard.E4.Flex	1 - 64	16 GB - 1 TB	256 GB - 80 TB	1 -40 Gbps	1 - 2	

Abbildung 2: Prozessortypen bei Oracle Database VMs (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

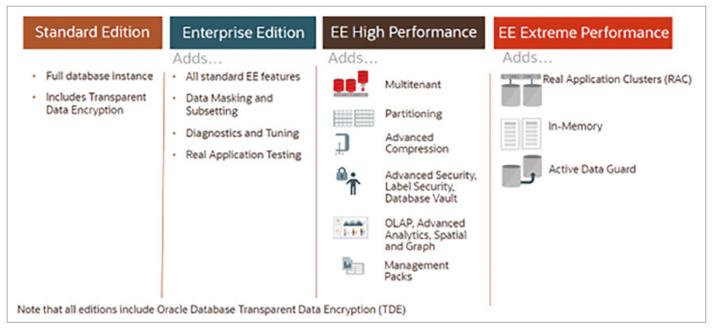


Abbildung 3: OCI-Datenbank-Lizenz-Bundles (© Oracle)

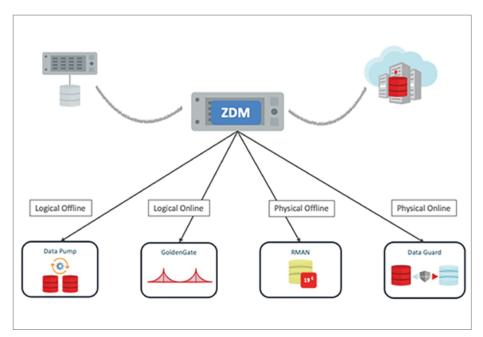


Abbildung 4: ZDM-Migrationsmethoden (© Oracle, Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

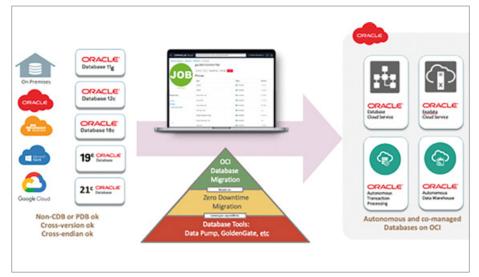


Abbildung 5: OCI Database Migration Service (© Oracle)

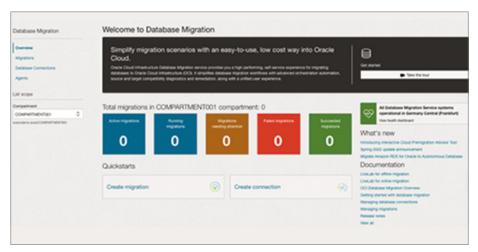


Abbildung 6: Übersicht Database Migration (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

Bei der Migration auf einen der anderen Oracle Database Cloud Services kann sowohl die Physical Migration als auch Logical Migration verwendet werden.

Dabei wird die Liste der oben genannten Werkzeuge noch durch folgende ergänzt:

- RMAN
- Data Guard
- PDB (unplug/plug, cloning, hot cloning)

In diesem Artikel möchte ich auf die folgenden Tools eingehen:

- Data Pump
- PDB Remote Hot Cloning
- Zero Downtime Migration (ZDM)
- OCI Database Migration Service (DMS)

Data Pump

Data Pump ist ein universelles Migrationstool, da es für fast alle Szenarien einsetzbar ist, dazu gehören:

- Quell- und Zieldatenbanken werden in unterschiedlichen Datenbank-Versionen betrieben.
- Quell- und Zieldatenbanken verwenden unterschiedliche Architekturen, etwa Ziel-DB verwendet Tenant-Architektur.
- Quell- und Zieldatenbanken besitzen unterschiedliche Endian-Formate, etwa Quell-DB läuft auf AIX, Ziel-DB auf Linux.
- Quell- und Zieldatenbanken verwenden unterschiedliche Zeichensätze.

Des Weiteren kann Data Pump sowohl von der Standard Edition als auch der Enterprise Edition genutzt werden. Der Nachteil von Data Pump ist, dass die benötigte Downtime mit der Größe der Datenbank zunimmt.

Bei der Migration in die Oracle Cloud mittels Data Pump sollte beachtet werden, dass der lokale Speicherplatz auf den Oracle Database Cloud Service VMs in der Regel für den Dump nicht ausreicht. Daher empfehle ich den Einsatz von File Storage, der mit nfs an die gewünschte Datenbank-VM gemountet werden kann. Bei der Verwendung von Autonomous-Datenbanken muss der Dump auf Object Storage abgelegt werden.

```
--#### Create and Grant Privileges to the CDB user. ####
-- GoldenGate CDB User does not exist, create CDB user is required to extract transactions from the data-
base.
ALTER SESSION SET CONTAINER = CDB$ROOT:
CREATE USER C##GGADMIN IDENTIFIED BY "xxxx" CONTAINER=ALL DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE
TEMP QUOTA UNLIMITED ON USERS;
GRANT CONNECT TO C##GGADMIN CONTAINER=ALL;
GRANT RESOURCE TO C##GGADMIN CONTAINER=ALL;
GRANT CREATE TABLE TO C##GGADMIN CONTAINER=ALL;
GRANT CREATE VIEW TO C##GGADMIN CONTAINER=ALL;
GRANT CREATE SESSION TO C##GGADMIN CONTAINER=ALL;
GRANT SELECT CATALOG ROLE TO C##GGADMIN CONTAINER=ALL;
GRANT DV GOLDENGATE ADMIN TO C##GGADMIN CONTAINER=ALL;
GRANT DV GOLDENGATE REDO ACCESS TO C##GGADMIN CONTAINER=ALL;
GRANT ALTER SYSTEM TO C##GGADMIN CONTAINER=ALL;
GRANT ALTER USER TO C##GGADMIN CONTAINER=ALL;
GRANT SELECT ANY DICTIONARY TO C##GGADMIN CONTAINER=ALL:
GRANT SELECT ANY TRANSACTION TO C##GGADMIN CONTAINER=ALL;
GRANT EXECUTE ON dbms_lock TO C##GGADMIN CONTAINER=ALL;
EXEC DBMS GOLDENGATE AUTH.GRANT ADMIN PRIVILEGE ('C##GGADMIN',CONTAINER=>'ALL');
--#### Create and Grant Privileges to the PDB user. ####
-- GoldenGate PDB User does not exist, create PDB user is required to extract transactions from the data-
base.
ALTER SESSION SET CONTAINER = MIG1;
CREATE TABLESPACE GG_ADMIN DATAFILE '/u01/app/oracle/oradata/CDDBMIG/MIG1/ggadmin_data.dbf' SIZE 100m AU-
TOEXTEND ON NEXT 100m;
CREATE USER GGADMIN IDENTIFIED BY "xxx" CONTAINER=CURRENT DEFAULT TABLESPACE GG ADMIN TEMPORARY TABLESPACE
TEMP QUOTA UNLIMITED ON GG ADMIN;
GRANT CONNECT TO GGADMIN CONTAINER=CURRENT;
GRANT RESOURCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE SESSION TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT CATALOG ROLE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER SYSTEM TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER USER TO GGADMIN CONTAINER=CURRENT;
GRANT DATAPUMP_EXP_FULL_DATABASE TO GGADMIN CONTAINER=CURRENT;
GRANT DATAPUMP IMP FULL DATABASE TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ANY DICTIONARY TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ANY TRANSACTION TO GGADMIN CONTAINER=CURRENT:
GRANT INSERT ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT UPDATE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT DELETE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT LOCK ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY INDEX TO GGADMIN CONTAINER=CURRENT:
GRANT CREATE ANY CLUSTER TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY INDEXTYPE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY OPERATOR TO GGADMIN CONTAINER=CURRENT:
GRANT CREATE ANY PROCEDURE TO GGADMIN CONTAINER=CURRENT;
```

```
GRANT CREATE ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT:
GRANT CREATE ANY TRIGGER TO GGADMIN CONTAINER=CURRENT:
GRANT CREATE ANY TYPE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT:
GRANT CREATE ANY VIEW TO GGADMIN CONTAINER=CURRENT:
GRANT ALTER ANY TABLE TO GGADMIN CONTAINER=CURRENT:
GRANT ALTER ANY INDEX TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY CLUSTER TO GGADMIN CONTAINER=CURRENT:
GRANT ALTER ANY INDEXTYPE TO GGADMIN CONTAINER=CURRENT:
GRANT ALTER ANY OPERATOR TO GGADMIN CONTAINER=CURRENT:
GRANT ALTER ANY PROCEDURE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT:
GRANT ALTER ANY TRIGGER TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY TYPE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE DATABASE LINK TO GGADMIN CONTAINER=CURRENT;
GRANT EXECUTE ON dbms lock TO GGADMIN CONTAINER=CURRENT;
EXEC DBMS GOLDENGATE AUTH.GRANT ADMIN PRIVILEGE ('GGADMIN',CONTAINER=>'CURRENT');
```

Listing 1: Output dms_prep_db.sql

PDB - Remote Hot Cloning

Anhand eines Kundenbeispiels möchte ich die Migration einer Enterprise Edition Datenbank 18c mit klassischer Architektur auf eine 19c-Datenbank in der Oracle Cloud mit Tenant-Architektur mittels Remote Hot Cloning vorstellen. Dabei wird die NON-CDB mit dem Namen CONTROL in eine PDB namens CONTROL migriert.

Auf der Quelldatenbank wurde dazu ein User DBCLONE mit folgenden Rechten angelegt:

SQL> create user DBCLONE identified by xxx;

SQL> grant CONNECT, RESOURCE, CRE-ATE PLUGGABLE DATABASE to DBCLO-

Auf dem Zieldatenbanksystem wurde in der Datei TNSNAMES.ORA die Quelldatenbank CONTROL hinzugefügt und der folgende Datenbanklink angelegt:

Oracle Datenbanken Monthly News

Auf dem deutschsprachigen Oracle-Blog ist die März-Ausgabe der News-Serie erschienen.

DOAG Online



Es ist wieder so weit: die neue Ausgabe ist online! Das sechsköpfige Redaktionsteam von Oracle Deutschland hat wieder Neuigkeiten rund um die Oracle-Datenbank für On-Premises und Cloud-Installation zusammengestellt.

Alles wird wieder in einem Video präsentiert.

In der aktuellen Ausgabe wird wieder ein zusätzliches Quick Link Posting (in Englisch) zur Verfügung gestellt, um einen schnellen Zugriff auf die zugehörigen Links zu gewährleisten.

https://www.doag.org/de/home/news/ oracle-datenbanken-monthly-news-32/

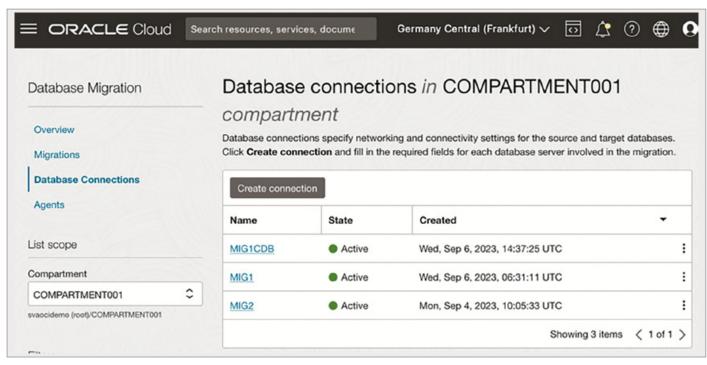


Abbildung 7: Datenbank-Verbindungen (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

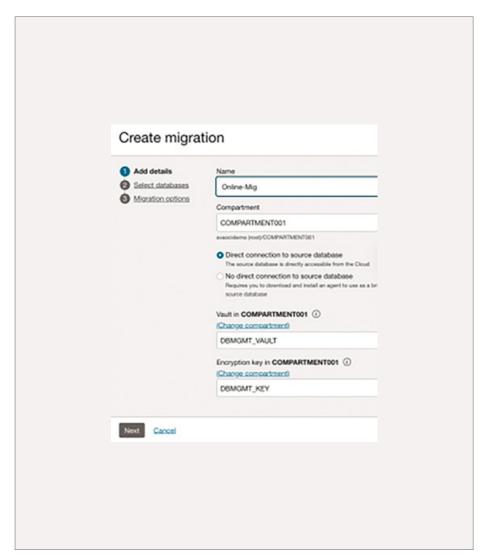


Abbildung 8: Create migration – Add details (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

SQL> CREATE DATABASE LINK sourcedblink CONNECT TO DBCLONE IDENTI-FIED BY xxx using 'CONTROL';

Damit wären die Vorbereitungen auch schon abgeschlossen und der Klon-Vorgang kann mit diesem Kommando gestartet werden, wobei der Parameter file_ name_convert optional ist:

SQL> CREATE pluggable database CONTROL FROM NON\$CDB@sourcedblink file_name_convert=('/u01/app/oracle/oradata/CONTROL/','+DATA/CONTROL/');

Während des Klon-Vorganges steht die Quelldatenbank den Anwendern zur Verfügung, eine Downtime wird erst nach dem Klonen notwendig, da die erstellte PDB sich zu diesem Zeitpunkt noch auf der Version 18c befindet.

Mittels dbupgrade -I /home/oracle/ logs -c "CONTROL"-Kommando wird diese anschließend nach 19c migriert, was 36 Minuten in Anspruch genommen hat. Im letzten Schritt muss noch das folgende Skript ausgeführt werden und die PDB neu gestartet werden:

SQL> ALTER SESSION SET CONTAINER=CONTROL;

SQL> **@?/rdbms/admin/noncdb_to_** pdb.sql

SQL> shutdown immediate;

SQL> startup;

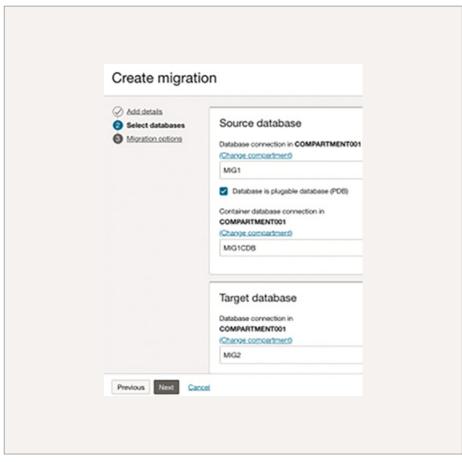


Abbildung 9: Create migration – Select database (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

Zero Downtime Migration (ZDM)

Zero Downtime Migration ist ein kostenfreies Tool von Oracle, welches eine automatisierte Datenbank-Migration in die Oracle Cloud, unter der Verwendung der Oracle MAA Best Practices, ermöglicht. Die aktuelle Software in der Version 21.4 (820MB) kann unter folgendem Link heruntergeladen werden: https://www.oracle. com/database/technologies/rac/zdm-downloads.html.

Oracle empfiehlt die ZDM-Software auf einem dedizierten Oracle Linux 7/8 oder Red Hat Enterprise Linux 8 Server/ VM mit 100 GB freiem Speicher zu installieren. Dieser Server wird als Zero-Downtime-Migration-Service-Host bezeichnet und kann sich sowohl in der Oracle Cloud als auch On-Premise befinden. Dieser kann eine oder mehrere Migrationsaufträge gleichzeitig ausführen, benötigt aber auf jeden Fall eine Verbindung zu den Quell- und Zieldatenbanken.

Features von ZDM in der Übersicht:

- Als Quelldatenbanken von Oracle werden sowohl Standard als auch Enterprise Edition in den folgenden Versionen supportet: 11g, 12c,18c, 19c, 21c.
- Folgende Oracle-Zieldatenbanksysteme werden supportet:
 - Oracle Database Cloud Service Virtual Machine
 - Oracle Database Cloud Service Bare Metal (sofern noch verfügbar)
 - Exadata Cloud Service
 - Exadata Cloud at Customer
- Exadata On-Premises
- Autonomous Database
- Die Quelldatenbank kann auf Linux, AIX oder Solaris-Umgebung betrieben werden.
- Eine Migration von NON-CDB auf CDB ist mittels ZDM möglich.
- Es wird sowohl Physical- als auch Logical Migration unterstützt, wobei mit Logical Migration auch in-flight Upgrades möglich sind.

ZDM unterstützt folgende Migrationsmethoden (siehe Abbildung 4: ZDM-Migrationsmethoden):

- Logical Offline
- Logical Online (verwendet Golden-Gate als Marketplace Image; keine Golden-Gate-Lizenz notwendig)
- Physical Offline
- Physical Online (nur in der Enterprise Edition möglich, da Data Guard verwendet wird)
- Audit Alle Migrationsphasen werden auditiert.
- Migrationsphasen können durch eigene Skripte ergänzt beziehungsweise erweitert werden.
- Migrationsjobs können nach ausgewählten Migrationsphasen angehalten
- Job-Precheck: Job validiert, ob die Voraussetzungen auf Quell- und Zieldatenbank erfüllt sind.
- Job Termination/Rerun-Möglichkeit
- lob Scheduler

Den Ablauf einer Physical Online Migration habe ich auf der DOAG 2021 in meinem Vortrag "Aufbau einer Desaster Recovery DB mittels Zero Downtime Migration Tool in OCI" vorgestellt und gehe daher in diesem Artikel nicht weiter auf den Migrationsablauf ein.

OCI Database Migration Service (DMS)

OCI Database Migration Service ist eine Oberfläche in der OCI Console und basiert auf dem Zero Downtime Migration Tool (siehe Abbildung 5: OCI Database Migration Service).

Im Gegensatz zu ZDM unterstützt DMS im Moment noch nicht die Möglichkeit einer Physical Migration mittels RMAN und DataGuard. Dies befindet sich allerdings auf der Roadmap und soll in einer späteren Version verfügbar sein.

Wer schon immer mal eine Logical Migration mittels GoldenGate durchführen wollte, sich aber aus Kostengründen oder mangelndem GoldenGate-Wissen dagegen entschieden hatte, sollte sich DMS anschauen. Denn hier muss, anders als bei ZDM, noch nicht einmal ein GoldenGate Marketplace Image installiert werden.

Aber der Reihe nach: Zuerst müssen die Voraussetzungen der Quell- und Zieldatenbank überprüft werden. Dies kann auf zwei Arten erfolgen. Erstens man folgt der Dokumentation Preparing the

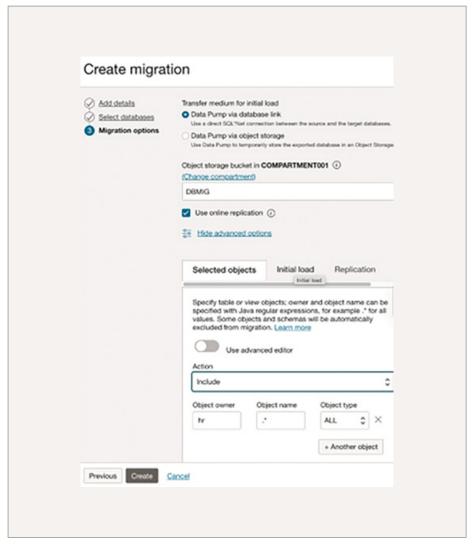


Abbildung 10: Create migration - Migration options (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

Source Database for Migration [2] beziehungsweise Preparing the Target Database for Migration [3] oder man verwendet das Preparation Utility, welches bei My Oracle Support unter OCI Database Migration Service - Download & Use Database Preparation Utility to Prepare Your Databases for Migration (Doc ID 2953866.1) [4] gefunden werden kann.

Das Skript dms-db-prep-v2.sh wird auf der Quell- und Zieldatenbank ausgeführt und generiert ein SQL-Skript namens dms_prep_db.sql, welches anschließend auf der entsprechenden Datenbank ausgeführt werden muss.

Das folgende Beispiel in Listing 1 zeigt den Output des Skripts dms_prep_db.sql für die Quelldatenbank.

Sind die Vorbereitungen auf der Quellund Zieldatenbank abgeschlossen, kann in die OCI Console gewechselt werden. Über das Menü Migration & Disaster Recovery gelangt man auf die Willkommensseite von Database Migration (siehe Abbildung 6: Übersicht Database Migration).

Bevor mit der Migration begonnen werden kann, müssen im Menü Database Connection (siehe Abbildung 7: Datenbank-Verbindungen) zuerst die Verbindungen zur Quell- und Zieldatenbank definiert werden, wobei in diesem Fall MIG1 die Quell-PDB, MIG1CDB die Quell-CDB und MIG2 die Ziel-PDB ist.

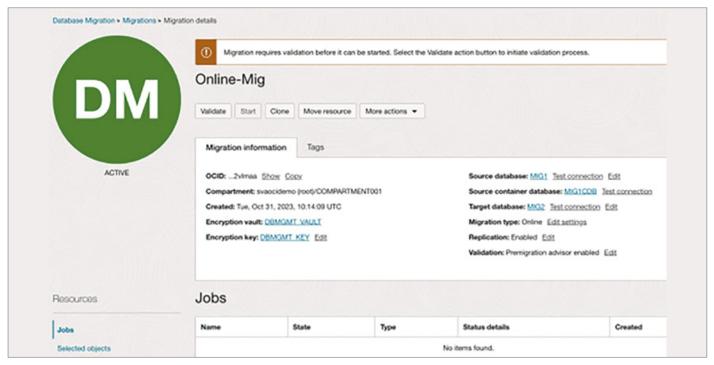


Abbildung 11: Online-Mig Übersicht (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

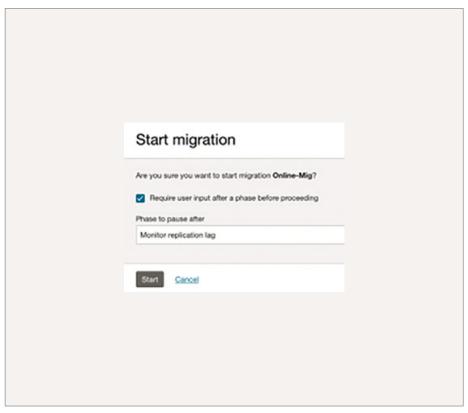


Abbildung 12: Phase to pause after (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

Anschließend kann über den Menüpunkt Migration in drei Schritten die Migrationsumgebung angelegt werden.

Im ersten Schritt (siehe Abbildung 8: Create migration - Add details) werden folgende Details festgelegt:

- Name, in diesem Fall Online-Mig
- Compartment
- Direct connection to source database
- Vault
- Encryption key

Im zweiten Schritt wird die Source- und Target Database ausgewählt (siehe Abbildung 9: Create migration - Select database).

Im dritten und letzten Schritt werden die folgenden Migrations-Optionen festgelegt (siehe Abbildung 10: Create migration - Migration options):

- Data Pump via database link
- Object Storage Bucket
- Use online replication (Wenn dies ausgewählt ist, wird automatisch eine Gol-

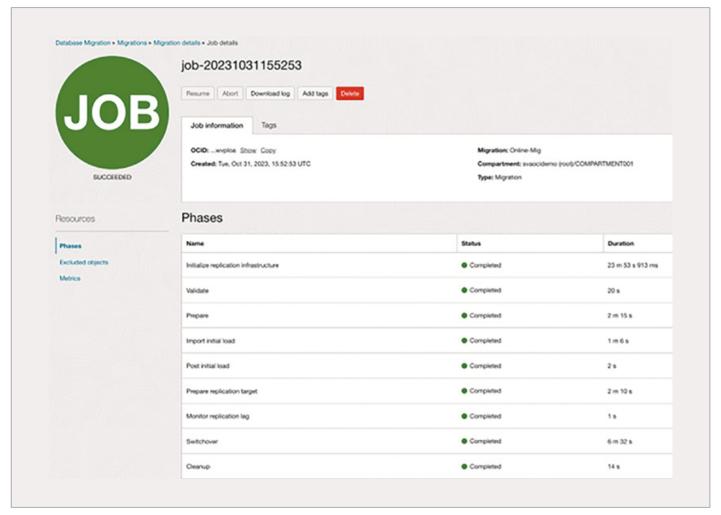


Abbildung 13: Erfolgreich ausgeführter Migrationsjob (© Kai-Uwe Fischer, SVA System Vertrieb Alexander GmbH)

- denGate-Instanz für die Dauer der Migration bereitgestellt.)
- Selected Objects (In diesem Fall wird das Schema HR repliziert.)

Die Migration **Online-Mig** ist nun erstellt und muss im nächsten Schritt validiert werden, bevor mit der Migration begonnen werden kann (Start-Button ist noch ausgegraut) (siehe Abbildung 11: Online-Mig-Übersicht).

Nach erfolgreicher Validierung ist der Start-Button verfügbar und der Anwender kann definieren, bis zu welcher Phase die Migration durchgeführt werden soll. Per Default stoppt die Migration nach der Phase **Monitor replication lag** (siehe Abbildung 12: Phase to pause after), somit besteht an dieser Stelle die Möglichkeit, die übertragenen Daten zu prüfen und die Migration abzuschließen.

Nach Beendigung einer erfolgreichen Online-Migration (siehe Abbildung 13: Erfolgreich ausgeführter Migrationsjob) werden die einzelnen Phasen und deren Dauer angezeigt, wobei in der ersten Phase Initialize replication infrastructure die entsprechende GoldenGate-Umgebung bereitgestellt wird. Dies dauert knapp 24 Minuten.

Zum Schluss stellt sich die Frage: Was kostet die Verwendung des OCI Database Migration Services?

Die Verwendung von DMS für alle Anwendungsfälle ist **kostenfrei**, das inkludiert:

- Den DMS-Cloud-Service
- Die OCI-Infrastruktur, auf der DMS läuft
- Die GoldenGate-Lizenz f
 ür die Logical Online DMS-Migration
- Optionale On-Premise-Agenten für indirekte Szenarien

Bei DMS nicht inkludiert sind:

- Die vom Kunden verwalteten OCI-Ressourcen, die für den DMS-Betrieb verwendet werden, wie Object Storage oder Oracle Stream Service
- Kosten für die Quell- beziehungsweise Zieldatenbanksysteme
- Netzwerkanbindung, wie zum Beispiel Fast Connect

Auf zwei Ausnahmen, für die Kosten anfallen, möchte ich noch hinweisen:

- Migrationen, die mehr als sechs Monate (183 Tage) nach ihrer Erstellung laufen
- 2. Migrationen, die mehr als 60 Tage im Leerlauf laufen (keine Datenübertragung).

Nach Überschreiten der Zeitgrenzen fallen Kosten von \$0,20 pro Stunde pro Migration an.

Fazit

Die optimale Datenbankmigration ist von verschiedenen Faktoren abhängig Laut Oracle gibt es insgesamt 27 verschiedene Möglichkeiten eine Oracle-Datenbank in die Oracle Cloud zu migrieren. Egal, für welchen Weg Sie sich entscheiden, es gilt das Motto: TESTEN, TESTEN, TESTEN. Welche Variante gewählt wird, hängt neben den oben genannten Faktoren vor allem davon ab, ob einige wenige oder eine große Menge an Datenbanken migriert werden sollen. Ist Letzteres der Fall, würde ich auf die Automatisierungsmöglichkeiten zurückgreifen, die mir ZDM und auch DMS bieten.

Für alle, die unschlüssig sind, welcher Migrationsweg der Beste für die eigene Datenbank ist, bietet Oracle jetzt den Cloud Migration Advisor (CMA) an. Dieser sammelt Datenbank-Metadaten einschließlich des Patch-Levels, wobei laut Oracle keine Geschäftsdaten oder Kundeninformationen erfasst werden. CMA nutzt diese Datenbank-Metadaten, um eine detaillierte technische Beratung über mögliche Migrationsziele und -methoden zu geben.

Quellen

- [1] Einleitung zu Zero Downtime Migration https://docs.oracle.com/en/database/ oracle/zero-downtime-migration/21.4/zdmug/introduction-to-zero-downtime-migration.html#GUID-A4EC1775-307C-47A6-89FB-E4C3F1FBC4F5
- [2] Preparing the Source Database for Migration https://docs.oracle.com/en-us/iaas/database-migration/doc/preparing-source-database-offline-migration.
- [3] Preparing the Target Database for Migration https://docs.oracle.com/en-us/iaas/database-migration/doc/preparing-target-database-offline-migration.

- [4] OCI Database Migration Service Download & Use Database preparation Utility to Prepare Your Database for Migration (DOC ID 2953866.1) https://docs.oracle.com/en-us/iaas/database-migration/doc/preparing-source-database-offline-migration.html
- [5] OCI Database Migration https://docs. oracle.com/en/cloud/paas/database-migration/
- [6] Cloud Migration Advisor https://www. oracle.com/database/upgrades/#rc30p2

Über den Autor

Kai-Uwe Fischer ist seit nunmehr 25 Jahren im Oracle-Datenbank-Umfeld als Consultant unterwegs und beschäftigt sich seit Jahren intensiv mit der Oracle Cloud (OCI-C & OCI). In dieser Zeit hat er einige Kunden auf dem Weg in die Oracle Cloud beziehungsweise Multicloud begleitet.

Herr Fischer hat mehrere wichtige Cloud-Zertifizierungen wie zum Beispiel Oracle Cloud Infrastructure Certified Architect Professional, Oracle Base Database Services Professional, Oracle Cloud Infrastructure Certified Multicloud Architect Associate & Specialist Oracle Database Cloud Migration and Integration.

Seit Januar 2023 ist er als Solution Architect Oracle Database & OCI bei der SVA System Vertrieb Alexander GmbH tätig.



Kai-Uwe Fischer kai-uwe.fischer@sva.de



Ein kritischer Blick auf PostgreSQL

Markus Flechtner, Ordix

Es ist ein Hype-Thema in der (Oracle-)Datenbank-Welt: die Migration nach PostgreSQL. Kosten, Lizenzdiskussionen, Unzufriedenheit mit dem Oracle-Support – es gibt viele Gründe, warum Oracle-Anwender mit einem Wechsel zu der OpenSource-Datenbank liebäugeln. Aber ist das Gras auf der anderen Seite des Zaunes wirklich grüner?

In vielen Fällen – dem Mantra "keine Logik in die Datenbank" geschuldet – wird die Datenbank nur als dummer Datenspeicher genutzt. Die Möglichkeiten einer modernen Datenbank, egal ob sie nun Oracle, PostgreSQL, Microsoft SQL Server, MariaDB/MySQL und so weiter heißt, werden damit sicher nur unzureichend genutzt. Das ist aber eine Grundsatz-Architektur-Diskussion, die wir an dieser Stelle nicht führen werden.

Eine derartige Verwendung von Datenbanken, bei der die erforderliche Datenbank-Funktionalität auf ein Minimum, das heißt praktisch, auf eine SQL-Kernfunktionalität reduziert ist, sorgt aber dafür, dass das darunterliegende RDBMS austauschbar wird.

Anders formuliert: Für einen großen Teil der Oracle-Anwender könnte eine PostgreSQL-Datenbank ausreichen und ist auch aus reiner Lizenz- und Supportkosten-Sicht in den allermeisten Fällen günstiger. Das erfreut den kostenbewussten Einkäufer und spricht im Fall von solchen Anwendungen für PostgreSQL. Viel zu oft werden bei solchen Kalkulationen nur die reinen Lizenz- und Supportkosten bedacht. Die anderen Kosten, wie Kosten für Migration und Tests, Schulungen, Erarbeitung von Betriebskonzepten und mehr werden außen vorgelassen, denn es sind EDA-Kosten. Dass sie in der Realität - wenn man alles berücksichtigt - oftmals höher sind als die reinen Software-Kosten, interessiert bei dieser Betrachtung nicht. Letztlich will man von kostenloser Software profitieren, sie "konsumieren" und die OpenSource-Entwickler seien nichts anderes als - wie Simon Brooke es auf Mastodon formuliert (siehe https://mastodon.scot/@simon_brooke/109616718918442031) - ein kostenloses Outsourcing-Team für die Software-Entwicklung ("OpenSource maintainers are effectively unpaid outsourcing teams for giant corporations.")

OpenSource-Software ist anders

Das System mit kommerzieller Software funktioniert - sehr vereinfacht gesagt -, solange die Software-Anbieter genug Kunden haben und die Anbieter ihr Geschäft weiterführen wollen. Der Open-Source-Bereich - genauer "Free/Libre Open Source Software" (FLOSS) - funktioniert, solange sich genug Menschen und Firmen engagieren. Das sind Menschen, die sich in ihrer Freizeit engagieren oder aber auch Menschen, die im Rahmen eines Arbeitsverhältnisses im FLOSS-Bereich arbeiten können und dürfen. Bei PostgreSQL - so ergab eine Auswertung (siehe https://rhaas.blogspot.com/2023/04/ who-contributed-to-postgresql.html) - kommen circa 90% des neuen Codes von gerade einmal 40 Entwicklern. Und die Mehrzahl davon macht das beruflich.

Es gibt andererseits mehr und mehr Unternehmen, die OpenSource-Software nutzen - oder die kommerzielle Software nutzen, die ihrerseits OpenSource-Software nutzt und die somit von OpenSource-Software zu einem gewissen Grad abhängig sind. Unter dem Titel "Log4j - Warum OpenSource kaputt ist" - hat heise.de (siehe https://www. heise.de/blog/Log4j-warum-Open-Source-kaputt-ist-6317678.html) diesen Aspekt im Rahmen der Diskussion über die im Dezember 2021 bekannt gewordene Sicherheitslücke in der OpenSourceKomponente log4j beleuchtet: "... Der gesellschaftliche Grund ist, dass zwar faktisch alle Unternehmen sich bei Open Source bedienen und davon abhängen, im Gegenzug aber kaum ein Unternehmen bereit ist, auch etwas beizusteuern oder zurückzugeben. ... Weil es zumindest kurzfristig weniger kostet: Weniger Zeit und vor allem weniger Geld. Aber das ist eben nur eine kurzfristig erfolgreiche Strategie. Langfristig wird dieser Ansatz schiefgehen, ..."

Diese Entwicklung hat nur am Rande mit PostgreSQL zu tun, aber man muss sie im Hinterkopf haben, denn es besteht durchaus das Risiko, dass dieses aktuelle "OpenSource-Geschäftsmodell" kippt. Bei kommerzieller Software bezahlt man mit Geld, OpenSource lebt von der Community, in die man mit Engagement einzahlt. Nicht jeder kann Code schreiben, aber es gibt vielfältige Möglichkeiten sich zu engagieren: Dokumentation, Veranstaltungen unterstützen oder Wissen weitergeben. Immer nur nehmen, nie geben - das wird auf Dauer möglicherweise nicht funktionieren – und es erfordert ein Umdenken in vielen Unternehmen, die immer nur den kurzfristigen finanziellen Vorteil von OpenSource-Software sehen.

Kommen wir nur zu den technischen Aspekten:

PostgreSQL umfasst gerade einmal 140 MB und ist fast beliebig erweiterbar

PostgreSQL ist eine sehr kompakte Software. Gerade einmal circa 140 MB umfasst die Software, wenn man sie aus dem Repository der PostgreSQL-Development Group (PGDG) installiert. Aber dafür ist die Software auch rudimentär, oder sagen wir mal "minimalistisch". Abhängig von den genauen Anforderungen kann man sich mit Hilfe von Erweiterungen und Tools sein PostgreSQL-Paket zusammenstellen. Aber diese Wahlfreiheit ist Fluch und Segen zugleich, der Anwender hat die sprichwörtliche Qual der Wahl. Welche Erweiterungen brauche ich? Wo finde ich die Erweiterungen für PostgreSQL überhaupt? Und wenn es mehrere ähnliche Erweiterungen gibt, welche nehme ich? Ein Beispiel: Nehme ich für meine Backups eher barman, pgBackRest oder doch was ganz anderes? In seinem Blog-Post Thoughts

on PostgreSQL in 2024 hat Jonathan Katz, Mitglied des PostgreSQL Core-Teams, dieses Thema als einen der Punkte beschrieben, bei denen PostgreSQL in diesem Jahr Fortschritte machen könne: "The side-effect of all this is we're now seeing ,extension sprawl'. How do I know which extension to use? What is the level of support of an extension? How do I know an extension will continue to be actively maintained? [...] Even "where can I download an extension" has become a big question [...]" (siehe https://jkatz05.com/post/postgres/postgresql-2024/).

In den meisten Fällen wird die PostgreSQL-Software über das Repository der PostgreSQL Development Group installiert. Ebenso können viele Tools und Erweiterungen über dieses Repository installiert werden, aber nicht alle. Und da fängt die Arbeit an. Selbst kompilieren, Abhängigkeiten auflösen und vieles mehr. Gleiches gilt in den seltenen Fällen, in denen man zum Beispiel die Standard-Page-Größe ändern will. Keine Raketenwissenschaft, aber auch nicht jedermanns Sache.

Patches auch im 3-Monats-Rhythmus

Ähnlich wie Oracle veröffentlicht auch die PGDG im Drei-Monatsrhythmus Patches für die PostgreSQL-Software. Man installiert die Software vom Repository, startet den Cluster und schon ist die neue Software aktiv. Änderungen am Systemkatalog des Clusters gibt es nicht - ein Vorteil gegenüber Oracle, da man nach der Installation eines Release Updates noch "datapatch" laufen lassen muss.

Auch andere Tools, die über die das PostgreSQL-Repository zur Verfügung gestellt werden, kann man einfach über ein "dnf update" (im RHEL-Fall) aktualisieren.

Aber was ist mit den anderen Tools, die nicht im Repository der PGDG vorhanden sind? Teilweise wird die Verfügbarkeit neuer Versionen über die Mailing-Liste pg_sql_announce@lists.postgresql.org bekanntgegeben. Aber das machen nicht alle Entwickler so. Für die "AWR-Kopie" pg_profile (siehe https://github.com/zubkov-andrei/pg_profile) gab es zum Beispiel keine derartige Informati-

```
[local] postgres@postgres=# \set AUTOCOMMIT off
[local] postgres@postgres=# select *,xmin,xmax from doag demo;
c1 | c2 | xmin | xmax
---+----
 1 | Test 1 | 310116 | 0
 2 | Test 2 | 310116 | 0
[local] postgres@postgres=# update doag demo set c2='Test 1a'
[local] postgres@postgres-# where c1=1;
UPDATE 1
[local] postgres@postgres=# delete from doag_demo where c1=2;
DELETE 1
[local] postgres@postgres=# select *,xmin,xmax from doag demo;
c1 | c2 | xmin | xmax
----+----
 1 | Test 1a | 310117 | 0
(1 row)
```

Listing 1: Session1: Ändern und Löschen

```
[local] postgres@postgres=# select *,xmin,xmax from doag demo;
c1 | c2 | xmin | xmax
----+----
 1 | Test 1 | 310116 | 310117
 2 | Test 2 | 310116 | 310117
(2 rows)
```

Listing 2: Abfrage in Session 2

on, als die mit PostgreSQL 16 kompatible Version 4.3 veröffentlicht wurde. In solchen Fällen heißt es: Selbst recherchieren, regelmäßig Webseiten abklappern und so weiter – am besten mindestens in dem gleichen Turnus, in dem auch Minor Upgrades für die PostgreSQL-Software veröffentlicht werden.

Updates

Neue PostgreSQL-Versionen erscheinen jährlich im September/Oktober. Daran hält sich die PostgreSQL-Development-Group seit Jahren und im Zweifelsfall fliegt ein neues Feature kurzfristig aus dem Release, wenn es noch nicht "fertig" ist.

Und viele Entwickler von Tools und Erweiterungen testen ihre Software schon gegen Beta-Versionen der neuen PostgreSQL-Software, passen ihre Software an und veröffentlichen - teilweise schon vor Freigabe der neuen

PostgreSQL-Versionen - eine neue Version ihrer Software, die mit der neuen PostgreSQL-Version kompatibel ist. Aber nicht alle.

So gab es zum Beispiel - Stand Anfang November 2023 - keine Versionen von pg_repack oder pg_autofailover, die mit PostgreSQL 16 kompatibel sind. Und wenn man diese Tools bislang eingesetzt hat, dann musste man entweder warten, Alternativen suchen oder selbst aktiv werden (und die Open-Source-Software forken, selbst weiterentwickeln).

Es kann gute Gründe geben, warum für eine neue Version eine Erweiterung noch nicht verfügbar ist. Es kann gute Gründe geben, warum ein Entwickler die Weiterentwicklung "seiner" Erweiterung beziehungsweise "seiner" Software eingestellt hat. Aber obwohl viele Tools federführend von kommerziellen Firmen gepflegt werden, so gibt es immer noch den "engagierten Freizeit-Entwickler", der in seiner Freizeit OpenSource-Software entwickelt. Das soll jetzt keine Kritik an diesem ehrenamtlichen Engagement sein - im Gegenteil - sondern soll lediglich aufzeigen, dass das für einen Produktionsbetrieb zu Abhängigkeiten und Problemen führen kann.

Architektur

Die ACID-Prinzipien, die Codd-schen Regeln und der SQL-Standard lassen kaum Spielraum für Variationen und so ähneln sich alle relationalen Datenbank-Systeme. Daher wird man beim Wechsel von Oracle zu PostgreSQL viele Ähnlichkeiten feststellen können. Der Teufel steckt im Detail, in der Implementierung, in der unterschiedlichen Verwendung des gleichen Begriffes oder der Verwendung unterschiedlicher Begriffe für vergleichbare Funktionalität.

Hauptkritikpunkt: MVCC &Vacuum

Hauptkritikpunkt an PostgreSQL ist die Art und Weise, wie Multiversion Concurrency Control (kurz MVCC) realisiert wird. "The part of PostgreSQL we hate the most", heißt es auf https://ottertune. com/blog/the-part-of-postgresql-we-hate-the-most zu diesem Thema und weiter: "Our research [...] [has] shown that its MVCC implementation is the worst among the other widely used relational DBMSs."

Wie funktioniert das nun bei PostgreSQL:

Vereinfacht gesagt, wird beim Löschen eines Datensatzes der Datensatz nicht physisch gelöscht, sondern nur "logisch gelöscht" beziehungsweise als "veraltet" gekennzeichnet. Bei der Änderung eines Datensatzes wird eine komplett neue Kopie des Datensatzes angelegt und die alte Version als "veraltet" gekennzeichnet. In einem zugehörigen Index gibt es dann zwei Einträge. Ein Beispiel dazu findet sich in Listing 1.

Session 2 in Listing 2 sieht man - Stichwort "read committed" - also noch den alten Stand (mit entsprechendem "Gültigkeitsende") und es gibt zwei Versionen des Datensatzes mit der ID 2.

Effekt: "Table Bloat"

Dieses Vorgehen führt zu dem Effekt, dass sich die Tabellen bei DML (UPDATE, DELETE) "aufblähen", englisch "Table Bloat" genannt (siehe Listing 3). Wenn man alle Datensätze ändert, dann wird die Tabelle doppelt so groß.

Aufräumen nicht vergessen

Wenn es keine Session mehr gibt, die die veralteten Datensätze ("Dead Tuples") benötigt, dann kann PostgreSQL aufräumen. Das geschieht mit dem Befehl "vacuum", der entweder manuell ausgeführt werden muss oder (empfohlen) über den Autovacuum-Daemon gestartet wird. Beim normalen Vacuum wird allerdings der Platz auf der Platte nicht freigegeben, sondern nur der Platz in den Datendateien der jeweiligen Tabellen. Der Platz auf der Platte wird nur bei einem "vacuum full" freigegeben, der allerdings einen exklusiven Lock auf die Tabelle benötigt. Außerdem verursacht dieses Aufräumen via "vacuum" auch zusätzliches I/O.

MVCC ist bei PostgreSQL ein einfacher Mechanismus, der auch funktioniert und in den meisten Fällen wird man damit keine Probleme haben. Aber es sind Situationen denkbar (Applikationen mit vielen großen UPDATEs und häufigen DELETEs), in denen dieser Aspekt durchaus relevant sein kann.

Und wie bereits erwähnt, wird die MVCC-Implementierung bei PostgreSQL von vielen als der Hauptkritikpunkt an der Datenbank angesehen.

Fehlender Statement Cache

PostgreSQL hat keinen Statement Cache, keinen "Shared Pool" wie bei Oracle. Jedes Parsing ist also ein "Hard-Parsing" und der Query Planner muss jedes Mal den Ausführungsplan neu ermitteln. Mit "prepared statements", also Befehlen, die meist mit Parametern arbeiten und vorab geparst und analysiert werden, kann man das etwas lindern, aber prepared statements gelten nur immer nur innerhalb einer Session (siehe Listing 4).

Den größten Vorteil erzielt man, wenn man in einer Session viele gleichartige

komplexe Befehle ausführt. Wenn eine Anwendung nur einfache SQL-Befehle ausführt, dann ist das Fehlen eines Shared Pools kein Problem. Aber wenn in der Anwendung viele Sessions gleichartige komplexe SQL-Befehle ausführen, dann kann die Planungszeit und der zugehörige CPU-Aufwand schon relevant werden.

Performance & Instrumentierung

"Out-of-the-box" bietet PostgreSQL nur wenige Möglichkeiten zum Monitoring und zum Tuning. Viele Tools für Monitoring und Instrumentierung sind nur als Erweiterungen verfügbar. Es gibt zum Beispiel einen "AWR-Nachbau" (pg_profile), einen Index-Advisor (pg_qualstats) oder einen "PASH-Viewer", mit dem ASH nachgebaut wird. Man muss sich also auch hier sein Paket an Tools und Erweiterungen selbst zusammensuchen und dann noch pflegen. Features wie SQL Plan Baselines, SQL Profiles oder Real-Time-Monitoring (Real-Time-SQL-Monitoring, Real-Time-ADDM) fehlen gänzlich.

Ein interessantes, aber nur sehr selten genutztes, Performance-Feature von Oracle ist übrigens der In-Memory-Base-Level (16 GB In-Memory-Column-Store, bei der Enterprise Edition kostenfrei dabei). In-Memory-Funktionalität gibt es bei PostgreSQL nur als kostenpflichtige, proprietäre Erweiterung (PostgresPro Enterprise Edition).

Features allgemein

Man kann eine Feature-Diskussion beliebig weiterführen, zum Beispiel: "Datenbank B ist schlecht, weil sie mein Lieblingsfeature von Datenbank A nicht hat". Aber allgemein gilt: Alle RDBMS entwickeln sich weiter und "schauen voneinander ab". Wenn PostgreSQL den "Merge"-Befehl als neues Feature in PostgreSQL 15 (2022) feiert, kommt von Oracle nur der Hinweis, dass das ja seit Oracle 9i (2003) ein alter Hut sei. Und wenn Oracle den Boolean-Datatype als neues Feature in Oracle 23c präsentiert, dann entlockt das der PostgreSQL-Welt nur ein müdes Lächeln.

```
select pg_size_pretty(pg_relation_size('doag_grosse_tabelle'));
pg size pretty
_____
106 MB
(1 row)
update doag grosse tabelle set c2=c2*2;
UPDATE 2111110
select pg size pretty(pg relation size('doag grosse tabelle'));
pg size pretty
211 MB
(1 row)
```

Listing 3: Table Bloat

```
PREPARE usrrptplan (int) AS
SELECT * FROM users u, logs 1
WHERE u.usrid=$1 AND u.usrid=1.usrid AND 1.date = $2;
EXECUTE usrrptplan(1, current date);
```

Listing 4: Beispiel für ein "Prepared Statement"

Hochverfügbarkeit

Hochverfügbarkeit bedeutet bei PostgreSQL: Replikation. Eine Primär-Datenbank repliziert auf 1 bis n Read-Only-Replika-Datenbanken. "Active DataGuard" ist bei PostgreSQL also kostenlos dabei. Das Ganze wird durch eine Menge an Tools und Frameworks unterstützt, das Bekannteste und meinem Eindruck nach am häufigsten genutzte, ist Patroni. Patroni ist eine Vorlage für PostgreSQL-Hochverfügbarkeit, die Primärdatenbank und Replika überwacht sowie automatisch umschaltet. Dazu kommt ein vorgeschalteter Load-Balancer, der beispielsweise die R/W-Sitzungen zur Primärdatenbank leitet und Sitzungen, die nur Lesen wollen, was so zu einer Read-Only-Replika führt. Damit skaliert PostgreSQL bei R/O-Abfragen, aber alle ändernden Transaktionen landen immer auf dem einen Primär-Cluster.

Oracle bietet mit RAC read-write-Skalierbarkeit für Anwendungen (wobei zugegebenermaßen nicht jede Applikation "RAC-fähig" ist) und Active-DataGuard für R/O-Skalierbarkeit (inklusive der Möglichkeit, DML auf ADG-Instanzen zu machen). RAC ist ebenso durch Patente geschützt wie Application Continuity, mit der DML-Transaktionen im Falle eines Failovers oder Switchovers automatisch wiederholt werden. Das sind Funktionalitäten, die vielleicht nur in den wenigsten Fällen benötigt werden, die aber dann im High-End-Bereich den Unterschied machen können.

Zusammengefasst

PostgreSQL ist in sehr, sehr vielen Fällen eine valide Alternative zu Oracle. Die Leistung, die PostgreSQL trotz der aufgeführten Kritikpunkte bietet, wird für die meisten Anwendungen ausreichen. Aber man kann nicht immer erwarten, dass die neue Datenbank ohne Änderungen weiterläuft, wenn man diese einfach nur austauscht. Wenn etwa ein Softwareanbieter seine Anwendung für mehrere Datenbanken anbietet, dann ist die Datenbank dabei meist nur der bereits erwähnte "dumme Datenspeicher". Ist man nun im High-End-Bereich unterwegs, dann nutzt man die Features "seiner Datenbank" auch aus. Wenn nicht, dann erreicht man nicht die maximale Performance. Allerdings wechselt man im High-End-Bereich auch nicht mitten im Rennen das Pferd, sprich: die Datenbank.

Letztlich muss man die Unterschiede in der Implementierung, die Stärken und Schwächen der jeweiligen Datenbanken kennen und dann entscheiden. Es gibt kein "gut" oder "schlecht", kein "schwarz" oder "weiß" - sondern nur einen Gesamtkontext, anhand dessen entschieden werden muss - und dazu gehören nicht nur die Lizenzkosten.

Über den Autor

Markus Flechtner ist Principal Consultant bei der ORDIX AG. Sein Schwerpunkt liegt im Bereich Multitenant, Hochverfügbarkeit sowie Upgrades und Migrationen. Weiterhin gibt er Seminare zu Oracle und PostgreSQL. In der DOAG ist er in der Datenbank-Community Themenverantwortlicher für Open-Source-Datenbanken.



Markus Flechtner mfl@ordix.de

BERLINER EXPERTENSEMINARE





07.05. - 08.05.2024

Oracle Reorg + Wartung

Berliner Expertenseminar mit Marco Patzwahl, Muniqsoft Training GmbH

In diesem Seminar lernen Sie, in Ihrer Datenbank ordentlich aufzuräumen und sie schlank zu halten. Es geht unter anderem um Reorg (inkl. Controldateien, Redologdateien, Archivelogdateien, Datendateien) und Datenbankwartung (AWR, Statspack, Log- & Audit-Bereiche aufräumen), Reorganisation von Tabellen und Tabellen-Partitionen (Datapump, Rename von Objekten) sowie den Segment-Advisor.





Nachlese zum DOAG Oracle-Forms-Thementag 2023

Frank Hoffmann, Cologne Data

Für viele fiel die Auswahl in Nürnberg nicht leicht, unter 19 Themenangeboten das Richtige auszuwählen. Wer hätte gedacht, dass bis zu 70 Teilnehmer den Weg zum Forms-Thementag gefunden haben. Die diesjährigen Besucher kamen überwiegend von Consulting-Firmen und Kunden, die Oracle Forms schon seit 20 bis 30 Jahren einsetzen und mittelfristig noch mindestens 5-10 Jahre weiter betreiben wollen.

Der weltweit bekannte Blogger und ausgewiesene PL/SQL-Experte Connor Mc-Donald (siehe Abbildung 1), der wie viele Oracle-Entwickler mit Oracle Forms seine Karriere begonnen hatte, staunte nicht schlecht.

Zum Forms-Vortrag von Michael Ferrante, dem Produktmanager von Oracle Forms fand er einen vollen Raum vor.

In der Vorbereitung hatten wir mit nicht mehr als 40 Teilnehmern gerechnet und am Konferenztag die Bestuhlung auf 70 Stühle erhöhen müssen (siehe Abbildung 2).

Der klassische Forms-Entwickler ist ohne soziale Medien aufgewachsen und interessiert sich daher auch wenig dafür – für Ihn sind solche Veranstaltungen wie die in Nürnberg wichtig, um sich zu vernetzen, aktuelle Informationen aus erster Hand zu bekommen, und aus anderen Projekten oder Problemlösungen zu

Weiterhin scheint mir die Präsenz auch wichtig, um Oracle zu zeigen, dass allein in Deutschland noch 3.000 Forms-Kunden auf der Mailing-Liste stehen, die man nicht vergessen sollte.

Die Happy Hour, die bereits am Abend vorher stattfand, wurde auch gut besucht und hat Jung und Alt viel Spaß gemacht (siehe Abbildung 3).

Mit dem Release 12.2.1.19 letztes Jahr und dem Release 14.1.2 kommt doch einiges an Bewegung in den Releasewechsel und da will man früh informiert sein, um rechtzeitig planen zu können. Das von vielen aktuell eingesetzte Release 12.2.1.4 hat nur noch bis September 2024 Support und da muss der Umstieg auf 12.2.1.19 schon in die Projektplanung aufgenommen werden.

Daher lag ein Schwerpunkt des Thementages auch auf der aktuellen Version 12.2.1.19 (letztes Release der 12er Version) und auf dem zukünftigen "Long Term Release" 14.1.2 mit Datenbankclient 23, welches auch ganz offiziell für dieses Jahr von Oracle angekündigt worden ist. Für Forms und Reports bedeutet das, wie bei dem letzten Release, etwa weitere 10 Jahre garantierter Support und die Möglichkeiten auf aktuellen Windows- und Unix-Maschinen zu entwickeln sowie aktuelle Datenbankenversionen zu betreiben.

Inzwischen werden auch wieder Forms-Schulungen angeboten, zum Beispiel von Dr. Jürgen Menge (Der IT-Macher, Wolfratshausen) oder von Jan Peter Timmermann. Damit sollte es auch möglich sein, wieder junge Leute zu rekrutieren und gut mit dem Werkzeug auszubilden.

Oracle Forms im Einsatz bei der Uniklinik Bonn

Spannend war sicherlich dieses Jahr auch von Kunden zu erfahren, warum Sie Ihre 25 Jahre alte Oracle-Forms-Software Version 6i weiter betreiben wollen und bereit sind, mehr zu investieren, um Sie auf die aktuelle Version zu bringen. Ich möchte ein Projekt, welches fachlich sehr ansprechend auf dem diesjährigen FormsThementag von einem Forms-Kunden präsentiert wurde, noch etwas tiefer technisch beleuchten.

Die Migration (6i/12c) mit Facelifting (Nutzung der neuen nativen GUI-Features aus Forms 11,12 und dann auch 14 ohne Code-Änderungen) wurde vom Kunden beauftragt, weil sich die Software bewährt hat und die User gerne damit arbeiten. Die Software läuft seit 1999 sehr stabil im 24/7-Betrieb, die Dateneingaben und Abfragen gehen, wie bei Forms üblich, blitzschnell und Anpassungen und Änderungen in den Masken sind sehr schnell und leicht umsetzbar. Ein großer Vorteil ist aber auch, dass das Softwareprodukt ursprünglich maßgeschneidert für den Kunden entwickelt wurde. Es bildet alle Arbeitsprozesse präzise ab. Die Entwicklung der FormsModule geschah 1999 bereits sehr fortschrittlich und voll automatisiert generiert mit dem Oracle Designer. Damit war der Code über alle insgesamt circa 200 Forms- und Reports-Module einheitlich generiert worden und die Masken erhielten eine einheitliche Grundfunktionalität. Nahezu alle Objekte und Objekteigenschaften sind über Objektbibliotheken referenziert worden, was für ein einheitliches "look&feel" sowie mühelose Anpassungen von Schriftgrößen oder Farben und Umrisse sehr hilfreich

Neben einer automatisierten Migration mit dem Forms API Master wurde nun nach 25 Jahren auch das Erscheinungsbild modernisiert: ein Facelifting mit neuer Farbpalette, neuen Icons, Logos und Buttongrafiken aus professioneller



Abbildung 1: PL/SQL-Experte Connor McDonald (Foto: Frank Hoffmann)



Abbildung 2: Volles Haus beim Forms-Vortrag von Michael Ferrante (Foto: Frank Hoffmann)



Abbildung 3: Happy Hour (Foto: Frank Hoffmann)

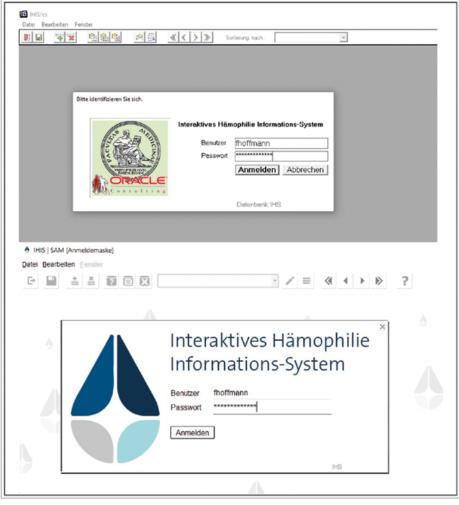


Abbildung 4: Anmeldemaske (Foto: Frank Hoffmann)

Hand und die Anpassung der Masken von kleiner INCH-Größe auf FULL-HD-Pixel-Größe (siehe Abbildung 4 und 5).

Die vorgeschalteten alten Reportparameter-Masken sind in die Module übertragen worden und werden dort dynamisch zur Laufzeit aus SQL-Code, der transparent im Menü abgelegt ist, generiert. Das sorgt auch für einen einheitlichen Workflow ohne Medienbrüche. Reports wird im Hintergrund ausgeführt und liefert ein PDF-File oder legt Dateien ab (siehe Abbildung 6).

Weiterhin wurden alle Schriftarten und -größen zur besseren Lesbarkeit angepasst. Sicherheitstechnisch wurde die alte Client-/Server-Verbindung mit SQL-NET8 durch eine sichere SSL TLS 1.3-Verbindung abgelöst (Ende-zu-Ende-Verschlüsselung). Mit dem möglichen Sprung der Datenbankversion von Oracle 12.1 auf 19.x/23 wurde auch hier der Weg in die Zukunft geebnet. Der Client wird über ein signiertes MSI installiert, damit die Client-Installation auch

über die Softwareverteilung mühelos erfolgen konnte.

Der Einstieg über die Auswahl von 15.000 Patienten wurde nun über eine dynamische, blitzschnelle Poplist statt über eine lästige Zwischenmaske stark vereinfacht (siehe Abbildung 7).

Jetzt braucht man nur noch 2-3 Buchstaben einzutippen, um die Vorauswahl des Patienten für die Applikation vorzubereiten. Nach knapp drei Monaten Entwicklungszeit konnte die Software auf Basis der neuen Version in Produktion gehen. Nach einem Monat Parallelbetrieb war alles schon so stabil, dass man die 6i-Version deinstallieren konnte.

Mit einer einfachen Migration (Rekompilierung) von 12.2.1.4 auf 12.2.1.19 und dann auf das nächste Release 14.1.2 kann die Software nun vom Kunden mit Premium Support auf dem Forms Java und Datenbankclient sowie Weblogic-Formsserver weiter genutzt werden. Der Kunde kann nun mit seiner Investition in die FORMS:FORMS-Migration sicher, schick und reibungslos mindestens weitere 10-12 Jahre seine bewährte und erfolgreiche Software betreiben sowie die vielen neuen nativen Forms 14 Features nutzen.

Abgerundet wurde die Migration noch mit der grafischen Button-Modernisierung der Ärzte-Informationsmaske (siehe Abbildung 8).

Wie sieht die Zukunftsplanung mit Oracle Forms bei den Forms-Kunden aus?

In Nürnberg habe ich mit vielen Kunden gesprochen, die langfristig bei Forms bleiben wollen. Sie kennen die Vor- und Nachteile ihrer Software und Tools und haben eine sehr komplexe Forms-Anwendung, mit der sie zufrieden sind, und wägen sorgfältig ihre Entscheidungen ab. Es gibt in solchen Fällen keine zwingenden Gründe von Forms wegzugehen. Einige ergänzen ihre Software HYBRID mit Web-Technologien oder Microsoft Tools wie POWER BI für Statistiken.

Aus kürzlichen Gesprächen mit Kunden, die schon im Jahr 2020 mit der Ablösung ihrer Forms-Software begonnen haben, und die sich entschieden haben, Forms mit DOT.NET oder ANGULAR/ Java abzulösen, höre ich von einem bis zu 5-fachen Aufwand, um eine komplexe Forms-Maske in der neuen Technologie im Vergleich zu der Erstentwicklung nachzubilden. Die Kunden wollten neu entwickeln und oft auch ganz von Oracle weg, haben aber den Aufwand dafür leider unterschätzt.

Die Stärke von Forms war und ist die komplexe Transaktionslogik, die von dem Tool automatisch generiert wird. Diese lässt sich zum Teil ungünstig oder nur sehr aufwendig nachbilden. Ich war etwas geschockt, von beiden Kunden zu hören, dass nach 4 Jahren und sehr hohem Aufwand nur etwa 10-20% der alten Funktionalität nachgebildet werden konnte. Der Erfolg dieser Projekte steht auf der Kippe.

Die ehemaligen Forms-/Reports-Entwickler werden bis in ein hohes Rentenalter weiter verpflichtet, das alte Produkt zu betreuen oder man bildet wieder Forms-Entwickler aus. Die Produktivität, mit der unter Forms schnell und komplex entwickelt werden konnte, bleibt unerreicht. Da kommt kein Tool ran.

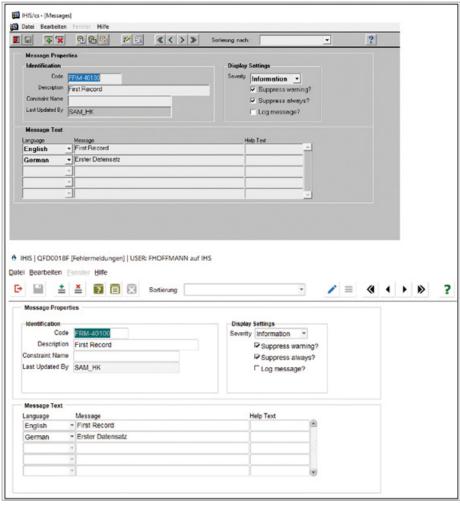


Abbildung 5: Fehlermeldungen (Foto: Frank Hoffmann)

Ein anderer sehr großer Forms-Kunde betreibt ein paar sehr komplexe Forms-Masken, die trotz mehrfachen Versuches bisher nicht brauchbar abgebildet werden konnten. So hat er bis heute in ganz Europa noch eine Forms-6i-Anwendung in Betrieb und will diese auch auf 12c migrieren.

Warum hinkt Forms der Zeit hinterher?

Oracle hat es leider nicht geschafft den PACE von Forms 6i zu halten und die IDE sowie die Runtime-Umgebung zeitgemäß in den letzten 20 Jahren zu modernisieren. Es fehlt aber auch zum Beispiel ein moderner JAVASCRIPT-GUI, der sogar schon fertig entwickelt wurde, um wieder mit modernen Webanwendungen voll konkurrenzfähig zu werden. Die Oracle- Ablösungsvorschläge 2001 mit Java, 2010 mit ADF und seit 2020 mit APEX oder Visual Builder wurden

oft von den erfahrenen Kunden nicht akzeptiert, weil keine der angebotenen Technologien einen vernünftigen Migrationspfad bieten kann.

Forms ist zwar weiter sehr mächtig aber leider, leider nicht mehr so "cool und sexy" wie vor 25 Jahren.

Was ist von Oracle 2024 zu erwarten?

So hat man mit Forms zwar transaktionstechnisch einen Porsche unter der Haube, ist aber im Erscheinungsbild um Jahre stecken geblieben (wenn man die Modernisierungsmöglichkeiten von Forms 12 nicht nutzt, vielleicht sogar im Jahr 1999). Mit Forms 14.1.2 (siehe auch das YouTube DOAG-Interview auf der K&A im November 2023 zu 14.1.2 mit Michael Ferrante) werden viele neu GUI-Features dazukommen, die sich sehr einfach nativ in die Formsmaske integrieren lassen: zum einen durch Properties in der

Maske und zum anderen durch Middleware-Konfigurationseinstellungen ohne Codeänderungen. Damit können die Masken, nach der Modernisierung mit den Features aus 12c, noch einmal deutlich moderner werden und mit pfiffigen UX-Features verbessert werden wie "Responsive Multirecord Display", "Block Grid Sort", "Sortierfunktionen im Clientcache ohne Datenbankaufrufe", "Restservices auf FORMS-Blöcke" (ohne Datenbank möglich) und grafische (statistische) Darstellungen in Forms zu Item-Werten.

Forms2APEX? Ist dieser Weg unsere Zukunft?

Aus meiner Sicht ist Forms2APEX vielleicht für Anwendungen kleinerer Komplexität sinnvoll.

Im Falle einer Neuentwicklung mit APEX könnte man zwar Datenbank-PL/ SQL-Logik mitnehmen, nur das Frontend muss dabei komplett neu entwickelt werden. So rechnet man in etwa mit dem 1.5-fachen Aufwand, der für die Erstentwicklung investiert wurde. Hinzu kommt, dass aus 300 Formsmasken schnell mal 1000+ APEX-Seiten werden, die ja auch mal gewartet werden müssen. Hier kann man sicherlich nicht von einer Migration, geschweige denn einer Modernisierung sprechen - Neuentwicklung wäre der passende Begriff.

Forms2JAVA? Forms2DOTNet?

Alle Forms-Ablösungen sind im Kern Frontend-Neuentwicklungen mit hohen Risiken. JavaScript-Oberflächen wie AN-GULAR sind modern und populär und bieten viele schicke UI-Features. Wir finden auch sicherlich 1.000-mal mehr Entwickler auf dem Markt dafür als für APEX oder Forms. Aber hier wäre eine Nachbildung einer komplexen Logik wie in unserem Forms2Apex-Beispiel ebenfalls ein Vortasten mit Prototyping und vielen Ansätzen. Wer will hier ins Blaue ein Feinkonzept schreiben, welches von einem Entwickler entwickelt werden soll, der Oracle-Datenbanken oder PL/ SQL kaum kennt? Forms hat immer einen aktuellen, voll supporteten Daten-

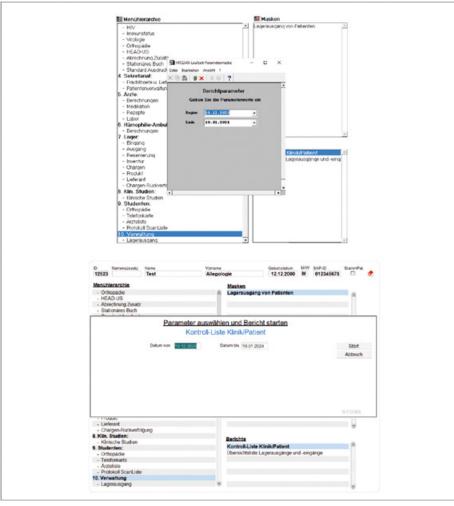


Abbildung 6: Berichtparameter (Foto: Frank Hoffmann)

bankclient an Board, der eine sichere, schnelle, also optimale (dauerhafte/ stateful session)-Verbindung zur Datenbank gewährleistet. Auch darüber müsste ich mir bei JDBC oder ODBC mehr Gedanken machen und bei Weblösungen kommt die aufwendige Anpassung für verschiedene Browser und JavaScript-Versionswechsel dazu.

Und was machen wir nun in Zukunft mit unserer Oracle-Forms-Anwendung?

Wir haben zum Glück noch gut 10 Jahre Zeit über diese Frage nachzudenken. Vielleicht wird es eines Tages eine KI geben, die uns den Code aus Forms herauslesen kann und daraus eine neue Anwendung erstellen kann.

Wir werden daher in der Forms-Gruppe sorgfältig alle neuen Web- oder Desktop-Technologien prüfen, diskutieren, vorstellen und aus solchen Projekten berichten. Selbst Oracle empfiehlt ab einer gewissen Komplexität bei Forms zu bleiben und alternative Technologien dann mit Forms zusammen Hybrid zu betreiben. Das kann sicher nicht schaden.

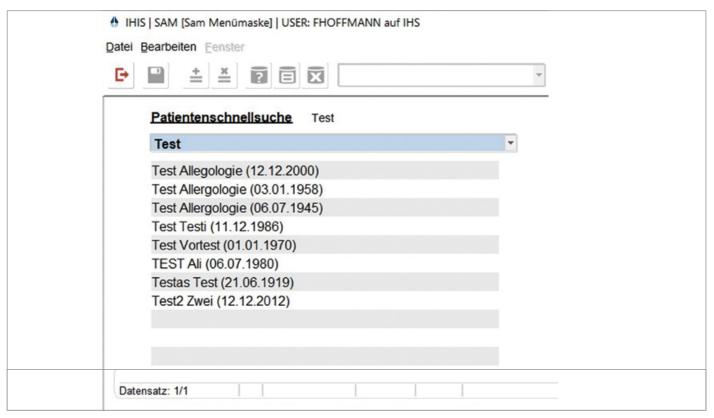


Abbildung 7: Poplist (Foto: Frank Hoffmann)

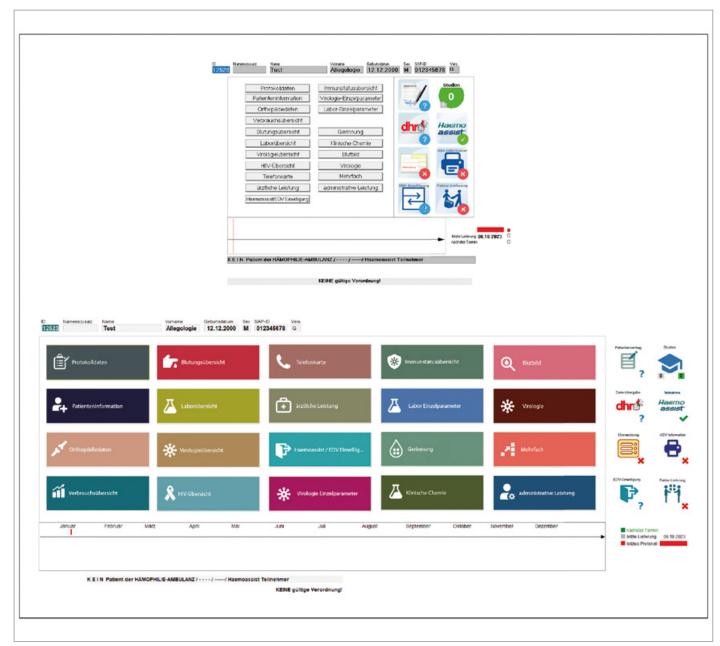


Abbildung 8: Dashboard (Foto: Frank Hoffmann)

Ziel von solchen Vorträgen und Diskussionen ist dabei immer, den Kunden sachlich, fundiert und neutral zu informieren, damit er selbst eine gute Entscheidung treffen kann. So läuft jede Planung einer Zukunft für eine Forms-Anwendung auf eine Einzelfallentscheidung hinaus und sollte immer sehr sorgfältig durchdacht werden.

Wir gehen im Moment davon aus, dass wir uns noch mindestens 10 Jahre mit Forms beschäftigen werden und freuen uns über den Platz in der DOAG. Arne Hattendorf, Leiter der Gruppe DEVELOP-MENT, sagte unlängst in einem Interview, dass "Oracle Forms" ein wichtiges Thema in der DOAG sei. Das Interesse auf dem

diesjährigen Thementag hat diese Annahme unterstrichen.

Weitere Veranstaltungen

Der nächste Forms Day, zu dem Oracle Berlin am 11/12.6.2024 einlädt, ist sicherlich ein guter Ort zur Vernetzung der Community, zu der inzwischen auch ein paar jüngere Forms-Entwickler gestoßen sind [2]. Motiviert durch die hohe Beteiligung des DOAG Forms Thementages im Jahr 2023 und des Oracle Forms Day 2023, sollen diese Veranstaltungen weiterlaufen und den Forms-Kunden helfen, ihre Forms-Anwendungen sicher und optimal weiter nutzen zu können oder ihnen mit Rat und Tat auch bei einer geplanten Ablösung zur Seite zu stehen.

Quellen

- [1] Ankündigung Oracle Forms 14.1.2 im Jahr 2024: https://www.oracle.com/a/ ocom/docs/middleware/fusion-middleware-statement-of-direction.pdf
- [2] Forms Day 2024 bei Oracle Berlin vom 11/12.6.2024 Noon2Noon, Anmeldung zum Oracle Forms Day 2024: https:// eventreg.oracle.com/profile/web/index. cfm?PKwebID=0x860777abcd
- [3] Presseportal: Forms-Schulungen https://www.presseportal.de/ pm/121145/5630081

[4] Aktualisierte Oracle-Forms-Informationen: https://de.wikipedia.org/wiki/Oracle_Forms

Über den Autor

Frank Hoffmann ist DOAG Forms Themenverantwortlicher in der Development-Gruppe und beschäftigt sich seit 1994 mit Oracle Forms sowie anderen Oracle-Technologien. Anfangs war er als Consultant für Opitz Consulting und von 1997-1999 als Seniorconsultant für Oracle Bonn tätig. 1999 gründete Frank die Cologne Data GmbH, die sich als Oracle Gold Partner, heute Cloud Builder, ausschließlich mit Oracle-Technologien beschäftigt und ihren Kundenstamm in der öffentlichen Verwaltung hat (zum Beispiel Landtag NRW, IM NRW, JM BW, JM Hessen, Bund.de, UNI Köln, UKB Bonn). Er betreibt mit Genehmigung von Oracle (Demo Addendum) auch einen Forms-Server live im Internet. Seine Firma entwickelt, modernisiert und unterstützt Kunden beim Betrieb komplexer Oracle-Forms-Anwendungen und Hochgeschwindigkeits-Webportalen mit Datenbankplugin. Frank ist Autor zahlreicher Red-Stack-Artikel und spricht regelmäßig auf DO-AG-Veranstaltungen und dem Oracle Forms Day. Seine Firma arbeitet eng mit den Produktmanagern von Oracle Forms zusammen, um für die Kunden und die Community aktuelle Informationen zu bekommen und um die Technologien erfolgreich umsetzen zu können.



Frank Hoffmann fh@cologne-data.de



BEST OF DOAG ONLINE

Eine Auswahl der besten DOAG News Februar/März 2024





"Die größte Herausforderung ist es, Menschen zusammenzubringen, die sich ehrenamtlich engagieren möchten" – Interview mit Katharina Schraft

Im April 2023 hat Katharina Schraft die Nachfolge von Thomas Karle als DOAG Vorstand Business Solutions angetreten. In unserem Interview berichtet sie von den ersten Monaten und blickt auf die kommenden Aufgaben.





DOAG.tv mit Moritz Klein und Niels de Bruijn über Flows for APEX

Flows for APEX ist ein Renner. 8.000 Downloads nach drei Jahren sind kein Pappenstiel. Warum die Open Source Process Engine auf Basis von BPMN so erfolg-reich ist, erläutern die Experten im "DOAG@Talk".





CloudLand 2024: Vielfältiges Line-up veröffentlicht

Die dritte Ausgabe des Cloud Native Festivals fährt erneut mit einem großartigen Line-up auf





Eine kleine Geschichte der Cloud

Während wir bereits voller Vorfreude auf die kommende CloudLand sind, werfen wir heute einen kleinen Blick auf die Historie der Cloud.





DOAG.tv mit Michael Ferrante und Frank Hoffmann über die Zukunft von Oracle Forms – Teil 1

In einem zweiteiligen Interview stand der Senior Principal Product Manager von Oracle Rede und Antwort über das anstehende 'Facelifting' und die kommenden Neuerungen.





DOAG Datenbank Kolumne: Schema-Only-Accounts – User ohne Password

In anderen Datenbanken sind "Benutzer" und "Schemata" getrennte Namensräume, bei Oracle gilt: "Ein Schema ist





April 04

09. - 11.04.2024

JavaLand 2024 Zwei ereignisreiche Konferenztage mit anschließendem Schulungstag rund um das Java-Ökosystem

Nürburgring, Nürburg

22. - 24.04.2024

APEX connect 2024 Konferenz mit zahlreichen Vorträgen und Workshops zu den Themen APEX, JavaScript und PL/SQL

Düsseldorf

15.4. - 14.06.2024

DOAG Tour: Highway to CloudLand. Roadshow der DOAG Cloud **Native Community**

Der DOAG Bus kommt zu Meetups und User Groups im ganzen Land.

25. - 27.04.2024

DOAG Leitungskräfteforum und Delegiertenversammlung

Mai

07. - 08.05.2024

Berliner Expertenseminar mit Marco Patzwahl Oracle Reorg + Wartung

15. - 16.05.2024

DOAG 2024 Datenbank mit Exaday Konferenz rund um die Oracle **Database und Engineered Systems**

Juni

06

18 - 21.06.2024

05

CloudLand 2024 - Das Cloud Native Festival Community-Veranstaltung rund um

die Themen Cloud- und Container-Technologien, Continuous Delivery, Microservices und DevOps

Phantasialand, Brüh

Impressum

Red Stack Magazin wird gemeinsam herausgegeben von DOAG e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, www.doag.org), AOUG Austrian Oracle User Group (Österreich, Lassallestraße 7a, 1020 Wien, www.aoug.at) und SOUG Swiss Oracle User Group (Schweiz, Dornacherstraße 192, 4053 Basel, www.soug.ch).

Red Stack Magazin ist die Community-Publikation für angewandte Informations- und Kommunikationstechnologie (ITK) im Raum Deutschland, Österreich und Schweiz. Es setzt bewusst auf Technologieoffenheit mit Blick auf Anwendung und IT-Innovationen.

Es bildet die Interessensschwerpunkte der Anwenderinnen und Anwender ab – von Cybersicherheit bis Datenschutz, von Datenbank und Development über Data Analytics bis Digitalisierung, von Cloud und Infrastruktur über Künstliche Intelligenz bis Open Source und Soft Skills – vermittelt praktisches Wissen und fördert den Know-how-Transfer und die Netzwerkbildung zwischen den Leserinnen und Lesern.

Die Inhalte des Red Stack Magazin werden von ausschließlich ehrenamtlichen Autorinnen und Autoren eingereicht und von der Redaktion aufbereitet.

Red Stack Magazin 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. DOAG e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. DOAG e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Björn Bröhl

Redaktion:

Sitz: DOAG Dienstleistungen GmbH (Anschrift s.o.) ViSdP: Fried Saacke Redaktionsleitung Red Stack Magazin: Martin Meyer Kontakt: redaktion@doag.org

Autorinnen und Autoren dieser Ausgabe (in alphabetischer Reihenfolge): Niels de Bruijn, Kai Donato, Kai-Uwe Fischer, Markus Flechtner, Frank Hoffmann, Dr. Stefan Koch, Carolin Krützmann, Marcos López, Martin Meyer, Tobias Schweiker, Marcel Uusvel-Amende, Rainer Willems

Titel, Gestaltung und Satz:

Diana Tkach DOAG Dienstleistungen GmbH (Anschrift s.o.)

Fotonachweis:

S. 6. © Nürburgring | nuerburgring.de S. 12: © Ludo-Photos | www.pixabay.com S. 20: © cocoparisienne | www.pixabay.com S. 26: © geralt | www.pixabay.com S. 34 © EdiNugraha | www.freepik.com S. 42: © Ronny Weiß S. 48: @ fanjianhua | www.freepik.com

Titel: © freepik | www.freepik.com

S. 52: © pcdazero | www.pixabay.com S. 63: © MamaClown | www.pixabay.com S. 68: © jpeter2 | www.pixabay.com S. 77: © 3974931 | www.pixabay.com

Anzeigen:

sponsoring@doag.org

Mediadaten und Preise:

www.doag.org/go/mediadaten

Druck:

WIRmachenDRUCK GmbH. www.wir-machen-druck.de

Inserentenverzeichnis

DOAG e.V. www.doag.org U 2, U 3, U 4

DOAG e.V www.doag.org S. 3, S. 7, S. 11, S. 19, S. 25, S. 40-41

Promatis Gruppe www.promatis.de

S. 75





at NCC Ost Nuremberg

netsuite.doag.org #NetSuiteUserDays

DOAG

DOA Datenbank mit Exaday

15. und 16. Mai in Düsseldorf

datenbank.doag.org

