

Red Stack

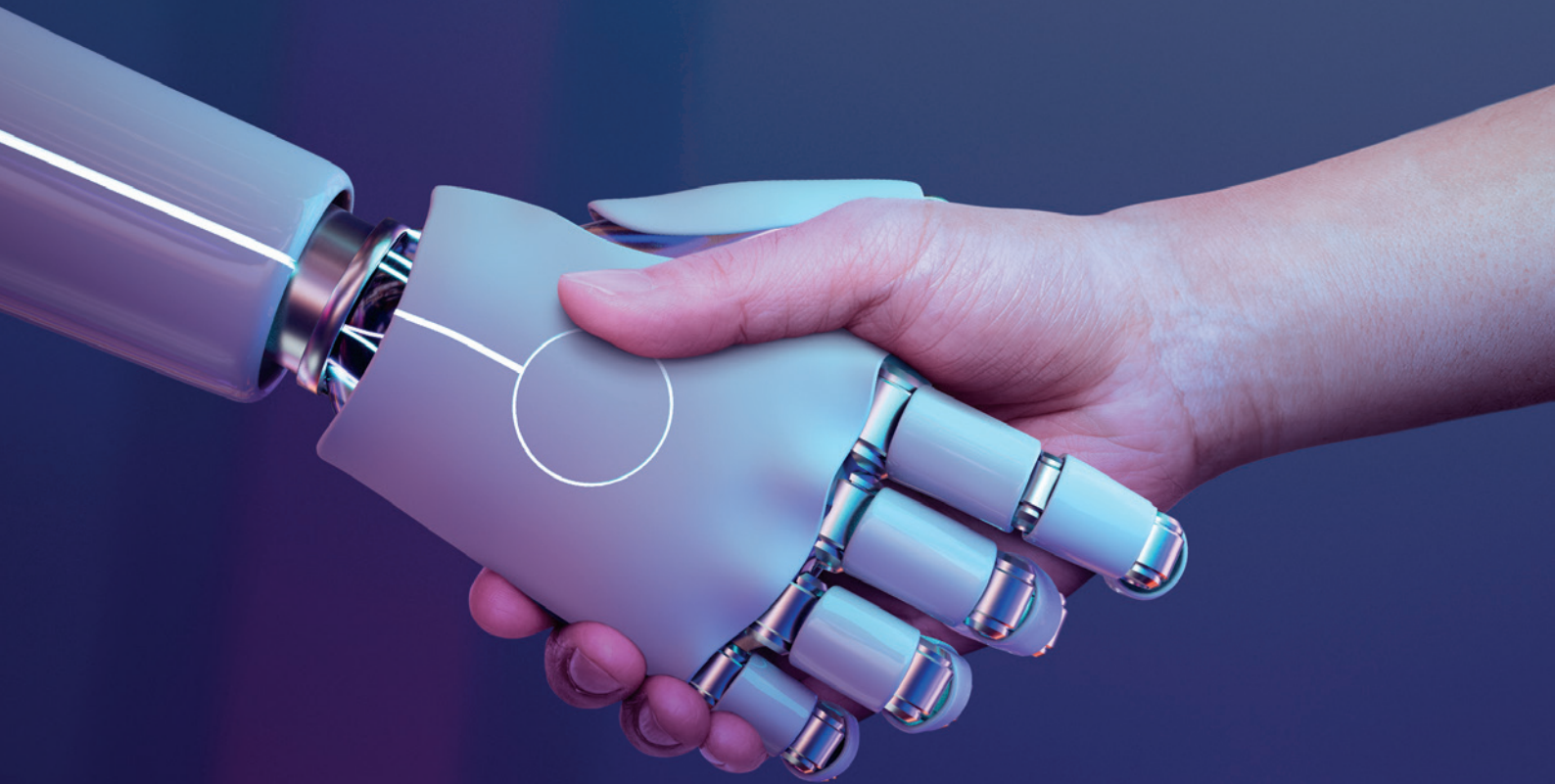
Magazin

DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP

inklusive BUSINESS NEWS



CI/CD & AUTOMATISIERUNG

Aus der Praxis

In zwei Tagen zur
neuen Exadata-Version



Im Interview

Martin Bach, Senior Principal
Product Manager, Oracle

Business News

Geodaten-IT

JavaLand

21. – 23. MÄRZ 2023

im Phantasialand bei Köln

Die Konferenz der Java-Community

www.javaland.eu

Mehr Infos zum
Event findet ihr hier:



The Zen of Programming

In seiner Keynote schildert Sander Hoogendoorn seine persönliche Reise durch Plattformen, Sprachen, Prinzipien, Zweifel und Kämpfe, die vielen Entwickler:innen im Laufe ihrer Karriere begegnen.



Community-Keynote

Sexismus, Mobbing und Bossing, unkontrollierter Narzissmus und fehlende Eskalationswege im Job haben in den letzten Jahren auch vor der Java-Community keinen Halt gemacht. Wir lüften den dunklen "Tabu-Vorhang".

Teilt eure Story mit uns: kummerkasten@ijug.eu



2.000+ JAVA-FANS



160+ SESSIONS



50+ AUSSTELLER



100 % AUSTAUSCH



54 JAVA USERGROUPS



100 % SPASS

Präsentiert von:



Heise Medien

DOAG

Veranstalter:





Thomas Lehmann
Themenverantwortlicher
Betriebs-Lösungen
Robotron Datenbank-
Software GmbH

Liebe Mitglieder, liebe Leserinnen und Leser,

der Frühling steht vor der Tür. Nicht nur in den eigenen vier Wänden, sondern durchaus auch in der IT-Landschaft ist es gang und gäbe, einen digitalen Frühjahrsputz durchzuführen. Jetzt ist ein guter Zeitpunkt, angestaubte Methoden auf den Prüfstand zu stellen und vielleicht hier und da einzelne Stell-schrauben zu drehen, um letztendlich die Effizienz des täglichen Handelns zu erhöhen.

In der aktuellen Ausgabe des Red Stack Magazin wollen wir uns den Themenkomplexen CI/CD sowie der Automatisierung widmen. Freuen Sie sich auf spannende Inhalte, bei denen für nahezu jede Rolle innerhalb unserer DOAG-Community etwas dabei ist!

Für Oracle-APEX-Entwickler haben wir die Themen Lifecycle Management, CI/CD mittels GitLab im Gepäck, ferner die weitere Vertiefung in die Funktionalitäten der APEX Workflow Engine sowie die Vorstellung einer Möglichkeit zum automatischen Generieren von Unit-Tests.

Für den Betrieb von IT-Infrastrukturen innerhalb der OCI werfen wir einen Blick auf Best Practices und bekommen erste Eindrücke, wie Administrationstätigkeiten automatisiert und damit auch ein Stück weit revisionsicher gestaltet werden können.

Im Bereich der klassischen Datenbankadministration beleuchtet das aktuelle Magazin Pluggable Databases sowie den Umgang von großen Datenbanksystemen mit der Oracle Standard Edition.

Die neue Ausgabe der DOAG Business News widmet sich voll und ganz dem Thema Geodaten. Mit Oracle Spatial steht eine mächtige Datenbankoption zur Verfügung, um Geodaten abzulegen und performant abzufragen. Wir werfen einen Blick auf einige ausgewählte Features und zeigen darüber hinaus auf, wie auf Basis von Geodaten ein einfaches Track & Trace stattfinden kann.

Ich wünsche Ihnen viel Spaß beim Lesen und ein paar Anregungen für den diesjährigen digitalen Frühjahrsputz.

Thomas Lehmann



Ausgabe Nr. 2/2023
auf Abruf!

DOAG WEBSESSION

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

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



www.doag.org/go/websessions



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



Interview mit Martin Bach



26

Oracle-Tool-Unterstützung für APEX Application Lifecycle Management



42

dbFlow – mit Git zu CI/CD in der Datenbank-Entwicklung

Einleitung

- 3 Editorial
- 6 Timeline
- 8 „Das Ziel von (CI/)CD ist es, immer lauffähigen, fehlerfreien Code zu haben, der jederzeit produktiv gehen könnte.“
Interview mit Martin Bach
- 14 Aus der Ferne betrachtet: Informatik-ferne Gesellschaft
Günther Stürner

Automatisierung

- 16 Automationen und Best Practices im Betrieb eines OCI-Tenant
Marcus Schröder
- 26 Oracle-Tool-Unterstützung für APEX Application Lifecycle Management
Arne Brüning
- 32 CI/CD für Oracle-APEX-Anwendungen mit GitLab
Johannes Michler und Simon Grossmann
- 38 Rethinking Unit Testing: Automatisieren der Generierung von Java-Unit-Tests
Markus Zimmermann und Evelyn Haslinger

CI/CD

- 42 dbFlow – mit Git zu CI/CD in der Datenbank-Entwicklung
Maik Michel

Exadata

- 47 In zwei Tagen zur neuen Exadata-Version – aus dem Protokoll eines DBA
Andrzej Rydzanicz

Datenbank

- 56 Schön, dass der Nachbar nicht stört! – Teil 2
Norman Sibbing
- 65 Große Datenbanken in der Standard Edition – Backup-Ideen
Thomas Beetz

71 Ist Ihre Datenbank 23c-ready?
Dierk Lenz

APEX

74 Workflow-Funktionalität in Oracle
APEX 22.2 (2/2)
Ralf Müller

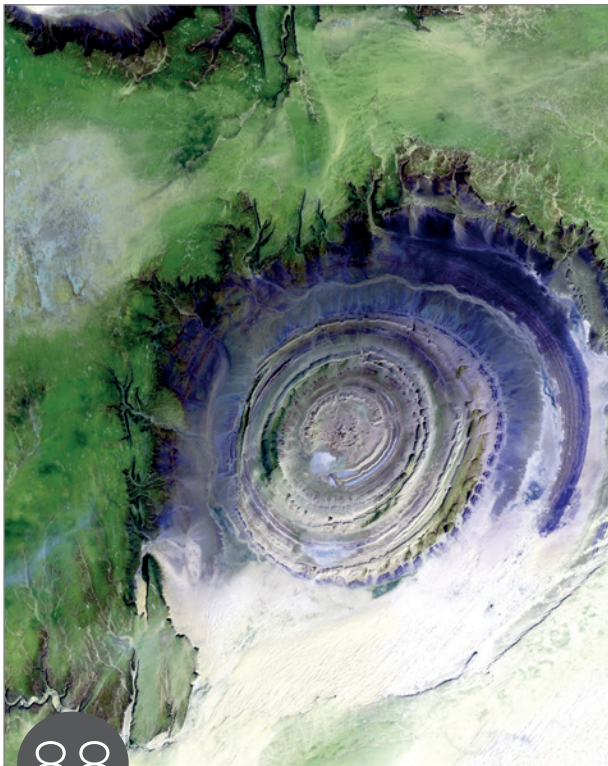
BUSINESS NEWS

Geodaten-IT

88 Einfaches Tracking und Tracing auf
Basis von Geodaten
Karin Patenge

95 Die fünf spannendsten Oracle
Spatial Features
Sven Bosinger

100 Digitale Modernisierung
in 2 Phasen – Einführung von
NetSuite und Oracle Integration
Cloud als Herzstücke digitaler
Prozessinfrastruktur
Michael Schäfer und Matthias Sauer



88

Einfaches Tracking und Tracing auf
Basis von Geodaten



65

Große Datenbanken in der Standard Edition –
Backup-Ideen



95

Die fünf spannendsten
Oracle Spatial Features



100

Digitale Modernisierung
in 2 Phasen – Einführung
von NetSuite und Oracle
Integration Cloud als
Herzstücke digitaler
Prozessinfrastruktur

Intern

105 Neue Mitglieder + Termine

106 Impressum + Inserenten

News

41 Oracle Datenbanken Monthly News

64 Ein Trio zum Early-Bird-Preis

83 Best of DOAG Online (Dezember/Januar)

84 Platinum Events wird Teil der DOAG

TIMELINE

12. Januar 2023

Im DevTalk mit Ronny Weiß, Christian Rokitta und Oliver Lemm wird das Thema „APEX & CSS“ behandelt.

13. Januar 2023

Auf der Agenda der DB WebSession mit Clemens Bleile steht das Thema „Nächste Generation Oracle DB Plattform: «Best of breed» versus Exadata. Vergleich von Performance und Snapshot-Funktionalität in einem PoC“.

26. Januar 2023

Das erste Treffen der Regio NRW im neuen Jahr findet statt.

26. Januar 2023

„Jenkins“ lautet das Thema des DevTalk mit Oliver Lemm und Torsten Kleiber.

09. Februar 2023

Im DevTalk mit Carsten Czarski & Christian Neumüller wird das Thema „Security für PL/SQL Entwickler:innen“ behandelt.

10. Februar 2023

In der DB WebSession mit Ernst Leber heißt das Thema "Migration mit ora2pg – ein Projektbericht".

23. Februar 2023

Im von Carsten Czarski moderierten DevTalk mit Jürgen Menge werden Möglichkeiten des Reportings in Oracle-Anwendungen (APEX, Forms, ...) behandelt.





DOAG

Werden Sie DOAG-Mitglied!

„Gemeinsame Interessen gemeinsam vertreten“

+ 30 % Rabatt auf Veranstaltungen
+ Kostenfreier Bezug unserer Zeitschriften

Red Stack Magazin inkl. Business News und Java aktuell

Ab 120 EUR/Jahr (zzgl. MwSt.)

www.doag.org


```
<a href="index.html">Home</a>  
<a href="home-events.html">Home Events</a>  
<a href="multi-col-menu.html" class="has-children">  
<ul>  
  <li><a href="tall-button.html">Tall Button</a>  
  <li><a href="image-logging.html">Image Logging</a>  
  <li class="active"><a href="current.html">Current Page</a>  
</ul>  
</a>  
<a href="multi-col-menu.html" class="has-children">  
<ul>  
  <li><a href="variable-length.html">Variable Length</a>  
  <li><a href="testimonials.html">Testimonials</a>  
  <li><a href="features.html">Features</a>  
  <li><a href="equal-columns.html">Equal Columns</a>  
  <li><a href="video-tutorial.html">Video Tutorial</a>  
  <li><a href="mini-button.html">Mini Button</a>  
</ul>  
</a>
```


„Das Ziel von (CI/)CD ist es, immer lauffähigen, fehlerfreien Code zu haben, der jederzeit produktiv gehen könnte.“

Martin Meyer, Redaktionsleiter des Red Stack Magazin, sprach mit Martin Bach, Senior Principal Product Manager bei Oracle ST & Database Development in Frankfurt, über Continuous Integration (CI), Continuous Delivery (CD) und DevOps im Kontext von Automatisierung in der Software-Entwicklung.

Bitte stellen Sie sich kurz unseren Lesern vor. Wer sind Sie und mit was beschäftigen Sie sich beruflich bei Oracle?

Sehr gerne! Ich bin seit einem guten Jahr bei Oracle und beschäftige mich mit dem Thema Software Engineering, und zwar sowohl was die eigentliche Interaktion mit der Datenbank angeht als auch in Bezug auf Automatisierung und DevOps. Bevor ich bei Oracle als Product Manager angefangen habe, war ich viele Jahre bei Enkitech, wo ich mich für Engineered Systems, Cloud-Architektur und vor allem Performance interessiert habe. Die Erfahrungen, die ich dabei machen konnte, habe ich auf meinem Blog und in drei Büchern mit der Community geteilt. Ich freue mich auch jedes Mal, wenn ich mich nach Vorträgen auf Konferenzen mit der Community austauschen kann.

Was versteht man eigentlich unter Continuous Integration (CI)?

Continuous Integration ist schon recht lange ein Begriff, so hat zum Beispiel Grady Booch bereits vor knapp 30 Jahren das Konzept aufgegriffen und für objektorientierte Programmierung (OOP) ins Spiel gebracht. Über die Jahre wurde CI von Autoren wie Kent Beck im Kontext des Extreme Programming, Martin Fowler sowie vielen anderen aufgegriffen. Im Bewusstsein der Entwickler ist das Thema spätestens dann angekommen, als klar wurde, dass längerfristige Kundenbindung nur dann möglich ist, wenn man schnell auf die Bedürfnisse des Marktes reagiert. Strikte Release-Zyklen einmal im Quartal reichen für viele Anwendungsfälle nicht mehr aus.

Will man seine Software schneller bereitstellen, geht das nicht ohne eine Änderung der Kultur der Softwareentwicklung innerhalb der Teams. Kommunikation innerhalb und außerhalb des Teams ist von kritischer Bedeutung, insbesondere wenn es um Anwendungen geht, die eine Persistenzschicht (sprich: Daten-

bank) benötigen. Eine Grundvoraussetzung für CI in diesem Fall ist die Kooperation von Entwicklern und DBAs.

Mit CI versucht man das Hauptproblem komplexer Softwareentwicklung – die Integrationsphase – möglichst kurz und im wahrsten Sinne des Wortes schmerzlos zu gestalten.

Für eine erfolgreiche Implementierung von CI sind laut der Literatur eine Reihe von Voraussetzungen zu beachten:

- Jedweder Code, der Bestandteil der Anwendung ist, muss versioniert werden.
- Trunk-based Development ist dem Gitflow-Modell vorzuziehen.
- Tests werden automatisch beim Commit und regelmäßig auf dem Hauptzweig durchgeführt. Jedwede Probleme werden sofort beseitigt, damit der Code jederzeit ohne Fehler ausgeführt werden kann.
- Kommunikation zwischen den Teams muss auf Arbeitsebene funktionieren.
- Die Teams müssen sich einig sein, CI zu implementieren, und keine Shortcuts wie zum Beispiel manuelle Arbeiten an der Datenbank zu dulden.

Die erste oben genannte Voraussetzung mag auf den ersten Blick seltsam anmuten, ist aber in der Praxis leider noch nicht selbstverständlich. Oftmals findet sich der Code des Frontend in Git oder einem vergleichbaren Versionskontrollsystem, der Code für das dazugehörige Datenbankschema jedoch nicht (*siehe Frage unten*). In der Literatur wird oftmals postuliert, sämtlichen zur Erzeugung einer Umgebung – egal ob Development oder Produktion – notwendigen Code (Frontend/Backend/Infrastruktur) unter Versionskontrolle zu stellen.

Trunk-based Development ist eine weitere Voraussetzung für eine erfolgreiche Umsetzung von CI. Es steht im Kontrast zum Gitflow-Modell, bei dem anstatt vieler, lange dauernder Feature

Branches lediglich der Hauptzweig (trunk/main etc.) gepflegt wird. Entwickler checken ihren Code idealerweise täglich ein, das heißt in kleinen Iterationen. Automatisierte Tests vor dem Commit und auf dem „Trunk“ stellen sicher, dass Code jederzeit ohne Fehler ausführbar ist. Wenn überhaupt, dann gibt es Feature Branches nur für äußerst kurze Zeit und für in das CI-Werkzeug eingebaute Code Reviews. Nach dem Merge in „Trunk“ werden lediglich die Merge- (oder Pull-) Requests mit ihrer Historie und den Kommentaren behalten, die Branches selbst werden gelöscht.

Idealerweise ist mit diesem Ansatz die Feedback-Schleife sehr kurz, Pair- oder Mob-Programming können dabei helfen, sie weiter zu verkürzen. Sollte es trotzdem einmal Probleme mit den (automatisierten!) Tests geben, wird der Trunk eingefroren, bis das Problem gelöst ist. Dies muss natürlich sehr schnell geschehen, damit die Kadenz der Commits in den Trunk nicht gestört wird. Test-Driven Development (TDD) ist eine mögliche Strategie, mit der man Fehlern schon entgegenwirken kann, bevor sie ein Problem bereiten.

Wissenschaftliche Untersuchungen wie zum Beispiel der State-of-DevOps-Report haben gezeigt, dass Teams, die häufig Code einchecken, wesentlich weniger Fehler machen und Fehler schneller beheben als solche, die lange Feature Branches pflegen. In letzterem Fall sind Merge-Konflikte geradezu vorprogrammiert, die zu beheben in der Regel lange dauert und damit kostspielig sind.

Natürlich ist das CI-Konzept nicht frei von Kritik. Die Anforderungen an Trunk Based Development sind sehr hoch, vor allem wenn Software sehr oft produktiv geht. Doch nicht nur die technischen Hürden sind hoch: Es gilt, alle Entwickler im Team für die Änderung der Arbeitsweise zu begeistern oder zumindest zur Kooperation zu bewegen, ansonsten wird es schwierig, CI in der oben genannten Art einzuführen. TDD mag ebenfalls eine neue Technik sein, mit der man erst vertraut werden muss. Ein weiterer, ganz wichtiger Punkt ist, dass CI/CD nicht für jede Anwendung die richtige Wahl ist!

Wie passt Continuous Delivery (CD) zu dem Thema?

Continuous Delivery oder kurz CD genannt ist der nächste, notwendige Schritt, um Software schneller auf den Markt zu bringen. Eine scharfe Abgrenzung zu CI ist schwierig, die beiden Konzepte gehen Hand in Hand. Das Ziel von (CI/ CD) ist es, immer lauffähigen, fehlerfreien Code zu haben, der jederzeit produktiv gehen könnte. In diesem Kontext wird Continuous Deployment auch oft als CD bezeichnet, die Wissenschaft misst diesem Konzept eine andere Bedeutung zu (*siehe Frage 4*). Wenn im Folgenden in dieser Antwort von CD die Rede ist, bezieht es sich auf Continuous Delivery.

Ohne CI, wie oben beschrieben, kann CD nicht erfolgreich eingesetzt werden. Bei der Einführung von CD ist es wichtig, die richtigen Werkzeuge einzusetzen, um nicht unnötig Zeit mit der Wartung der Infrastruktur zu vergeuden. Oftmals ist es zielführender, eine existierende Lösung einzusetzen als ein selbst entwickeltes System.

Ein wichtiger Baustein ist ein CI/CD-Server. Ein solcher CI/CD-Server unterstützt die Entwickler, indem er notwendige Prozesse automatisiert. Dazu gehören im Besonderen die

- Erstellung eventuell notwendiger Artefakte für die Anwendung (Executables, Webanwendungen, etc.)
- Bereitstellung einer Umgebung für die folgenden Tests, insbesondere im Cloud-Umfeld sinnvoll (*siehe Frage 10*)
- Einhaltung von Code-Richtlinien
- Automatisierung von Unit-Tests
- Ausführung von Acceptance-Tests
- andere, wie etwa das Erstellen von API-Dokumentation etc.

Die meisten CI/CD-Server erlauben die Definition dieser Schritte mit dem Konzept der Pipeline. Diese werden in der Regel in YAML-Dateien oder ähnlichen (Software-) Formaten definiert und müssen zwingend Teil des unter Versionskontrolle stehenden Codes sein.

Die Ausführung von Pipelines muss erfolgreich sein, ansonsten tritt, wie unter Frage 1 beschrieben, ein Code Freeze auf „Trunk“ ein, bis das Problem gelöst ist. Es ist sehr wichtig, bei der Ausführung von Pipelines aufs Tempo zu achten. Wenn die Ausführung einer Pipeline zu lange dauert, sinkt die Akzeptanz des CD-Ansatzes womöglich.

Ziel der Pipeline-Tests können entweder statische oder dynamisch erzeugte Umgebungen sein. Vor allem im Cloud-Umfeld werden Pipelines verwendet, um eine Testumgebung automatisch zu erstellen. Egal welcher Weg eingeschlagen wird, muss großer Wert darauf gelegt werden, dass die neue Zielumgebung mit den produktiven Systemen identisch ist. Ansonsten besteht die Gefahr, dass Tests keine aussagekräftigen Resultate bringen und jedes Deployment wieder ein unbekanntes Risiko birgt.

Steht „CD“ nicht für Continuous Deployment? Wo ist denn der Unterschied zu der vorigen Frage?

Die Abkürzung „CD“ ist in der Tat nicht eindeutig definiert, in der Literatur wird jedoch zwischen Delivery und Deployment unterschieden. Die meisten CI/CD-Server machen das auch so. Grob gesagt erlaubt es Continuous Delivery, eine Anwendung jederzeit so zu erzeugen, dass sie sofort und mit minimalen Risiko live gehen kann. Dieser letzte Schritt des Deployment erfolgt jedoch nicht automatisch.

Das Deployment kann natürlich auch automatisiert werden. Dabei ist nicht nur die Produktivumgebung gemeint: User Acceptance Test, Pre-Production, Production etc. können theoretisch alle als Ziel ausgewählt werden. Je nach Unternehmen kann dies entweder als manueller Eingriff nach erfolgreichem Abschluss der (Build) Pipeline geschehen oder sogar komplett automatisch. Wobei Letzteres, insbesondere für produktive Umgebungen, ein gehöriges Maß an Vertrauen an die eigenen Prozesse verlangt.

In der Deployment-Phase ist es ungemein wichtig,

- absolut identische Prozesse für alle Zielumgebungen zu nutzen,
- die gleichen Artefakte zu verwenden und nicht ein neues Artefakt pro Umgebung zu erstellen,
- identische Umgebungen zu verwenden.

Im Englischen gibt es dazu die schöne Bezeichnung „like for like“. Der Wert der automatisierten Bereitstellung einer Anwendung hängt sehr von der Umgebung ab. Ein häufiges Problem ist es

zum Beispiel, Performance-Tests mit einer auf 10 % verkleinerten Datenbank zu fahren. Was in diesem Fall gut funktioniert, wird mit hoher Wahrscheinlichkeit auf der produktiven Umgebung zu Problemen führen („it worked on my laptop“). Die Versionen aller eingesetzter Software müssen zudem überall auf dem gleichen Stand sein. Es macht wenig Sinn, eine Anwendung auf 19c zu testen, wenn die Produktion noch auf 12.2 betrieben wird und das Upgrade zwar geplant, aber noch nicht durchgeführt wurde.

Welche Werkzeuge und Prozesse stehen zur Verfügung, um dies umzusetzen?

CI/CD benötigt fast schon zwingend die Unterstützung von spezialisierten Werkzeugen, damit sich das Team mit der Programmierung der Anwendung beschäftigen kann und nicht auch noch selbstentwickelte Software an den (sich i. d. R. ständig ändernden) Prozess anpassen muss.

Es gibt eine Vielzahl von Werkzeugen auf dem Markt, die bei der Umsetzung von CI/CD helfen. Ob eine Cloud-basierte Lösung gewählt oder eine Lösung doch lieber im eigenen Rechenzentrum betrieben wird, kommt sehr auf den Einzelfall an. In IX 12/2021 haben eine Reihe von Autoren ihre Erfahrungen mit weit verbreiteten CI/CD-Systemen beschrieben.

Wichtig ist, bei der Auswahl des Systems darauf zu achten, dass es die CI-Prinzipien unterstützt, Pipelines einfach darstellen und versionieren zu können sowie bei Fehlern die zuständigen Entwickler schnell informieren zu können. Und zwar idealerweise nicht als E-Mail, da diese gerne per Filter aussortiert werden und dann keine Beachtung mehr finden. Chat-Systeme können bessere Dienste leisten.

Unit Test Frameworks, Linting und Code Coverage Tools sind ein integraler Bestandteil von CI/CD. Moderne Programmiersprachen bringen diese Werkzeuge entweder selbst mit oder sie werden von kommerziellen Herstellern/der Community bereitgestellt.

Das klingt logisch, aber wie mache ich das mit einer Datenbank?

Datenbanken sind, was das Thema CI/CD angeht, oftmals in der Praxis noch nicht so gut integriert. Dies hat häufig historische Gründe, was allerdings kein Hindernis sein sollte, Datenbanken in den CI/CD-Prozess zu integrieren.

Der erste Schritt dorthin ist die Einführung eines Versionskontrollwerkzeugs. In den allermeisten Fällen dürfte die Wahl auf Git fallen. Wie man jetzt den Datenbank-Code versioniert – als Git-Subprojekt, eigenständiges Git-Projekt, als Teil des Frontend –, hängt zu sehr von den lokalen Gegebenheiten ab, als dass eine allgemeine Aussage getroffen werden kann. Am wichtigsten ist die Einführung von Git! Wie so oft in der IT-Industrie ist die Einführung von Git für die Datenbank nicht nur ein technisches, sondern auch ein kulturelles Problem: Es gilt auch hier wieder das Team von den Vorteilen von Git zu überzeugen.

Ist eine Entscheidung zur Nutzung von Git gefallen, gilt es ein geeignetes Werkzeug für das Deployment zu finden. Oracle bietet Unterstützung für Liquibase in sql, was eine Möglichkeit ist, Datenbankänderungen durchzuführen. Es existieren noch weite-

re kommerzielle und Open-Source-Werkzeuge, die den gleichen Zweck erfüllen. Zu den vielen Vorteilen dieser Tools zählen:

- die Verwendung eines einheitlichen Formats für alle Änderungen
- Einführung eines Standards zur Bereitstellung des Codes in der Datenbank
- Pflege von Metadaten, die dabei helfen, die Historie an Änderungen nachzuverfolgen
- Erreichung von Idempotenz, das heißt, mehrfache (versehentliche) Ausführungen des gleichen Skripts können unterbunden werden

Die für die Änderungen des Schemas verwendeten Skripte-Changelogs in Liquibase werden erstellt und dann in Git eingepflegt, genauso wie Code im Frontend. Sie müssen sich genau wie jeder andere Code den automatischen Tests unterwerfen, Linting überstehen und weitere automatische (Unit-) Tests erfolgreich absolvieren. Gleiche Regeln für alle!

Um möglichst gut für unerwartete Zeitpunkte für das Release gewappnet zu sein, bietet es sich an, die Migrationskripte so zu schreiben, dass sie online durchführbar sind. Bei Oracle gibt es etwa Edition Based Redefinition (EBR) und eine Vielzahl anderer Möglichkeiten, die Nutzung der Anwendung so wenig wie möglich einzuschränken.

Ein Begriff im Kontext von CI/CD ist DevOps. Ist CI/CD das Gleiche wie DevOps? Oder wie hängen die beiden zusammen?

DevOps ist leider in der Wissenschaft kein zu 100 % fest definierter Begriff, je nach Autor unterscheiden sich die Definitionen. Allen gemein ist das nähere Zusammenrücken von Development (Dev) und Operations (Ops). Bevor DevOps ein Thema wurde und seinen Weg in die Unternehmen fand, arbeiteten Teams (Frontend-/Backend-Entwickler, Operations) zu sehr unabhängig voneinander. Oftmals bedeutete das lange Feedback-Schleifen, mangelnde Koordination und Probleme während des (sehr manuellen) Software-Release. Diese stehen dem Ziel, gute Software zu produzieren und kurzfristig auf Änderungen am Markt zu reagieren, diametral entgegen.

Es haben sich folgende Merkmale für DevOps etabliert:

- Projekte laufen unter gemeinsamer Verantwortung
- effiziente Kommunikation
- Automatisierung
- kurze Feedback-Schleifen

Unter diesen Gesichtspunkten ist CI/CD im Punkt Automatisierung zu finden und ein notwendiger Baustein.

Nicht nur im Zusammenhang mit CI/CD wird Automatisierung immer wieder hervorgehoben. Warum ist es heute wichtig, in Automatisierung zu investieren?

Automatisierung ist in der heutigen IT-Welt nicht mehr wegzudenken. Systeme werden in der Regel eher komplexer als einfacher, Stichwort Micro-Service-Architektur. In Umgebungen, in de-

nen viele Komponenten produktive Dienstleistungen für Kunden erbringen, ist es notwendig, die Infrastruktur im Griff zu haben.

Manuelles Einspielen von Patches in Systeme – sei es auf physischer Hardware, VMs oder schlimmer noch: Containern – ist aufgrund des damit verbundenen Aufwandes selten zeitgemäß. Puppet, Chef, Salt und Ansible sind Vertreter von Werkzeugen für das Konfigurationsmanagement, die in diesem Zusammenhang im Einsatz sind. Unter dem Schlagwort „Immutable Infrastructure“ wird ein anderer Ansatz verfolgt, bei dem man eine VM oder einen Container nicht mit Updates versorgt, sondern komplett durch eine neu gebaute Version ersetzt. Dies bietet sich eher für eine Stateless Application an als für eine Datenbank, es sind jedoch durchaus hybride Ansätze denkbar, vor allem wenn das Frontend in Containern implementiert wurde und in einem Kubernetes Cluster betrieben wird. Das Upgrade kann dann automatisch erfolgen, was Zeitaufwand und Risiko minimiert.

Automatisierung bietet sich immer dort an, wo repetitive Aufgaben zu erledigen sind, die wenig Mehrwert bringen, aber für den Betrieb unersetzlich sind. Eine Investition in Automatisierung erlaubt es anschließend, mehr Zeit für interessante Dinge zu haben.

Wo kommt Automatisierung heutzutage zum Einsatz?

Neben CI/CD, die oben bereits ausreichend beschrieben wurden, gibt es weitere Einsatzfelder für Automatisierung.

Cloud-Umgebungen werden häufig als „Infrastructure as Code“ in Terraform oder Pulumi bereitgestellt, was Zeit spart und konsistent richtige Ergebnisse liefert. Deklarative Sprachen erlauben dabei die wiederholte Ausführung des Codes, ohne dass es zu unerwünschten Ergebnissen kommen sollte.

Sobald eine Cloud-Umgebung erstellt worden ist, können Konfigurationsmanagement- Werkzeuge wie Ansible, Puppet, Salt und Chef diese konfigurieren.

Plattformen wie Kubernetes automatisieren die Bereitstellung von Container-gestützten Anwendungen und deren Lebenszyklus.

Welche Trends sehen und welche Entwicklungen erwarten Sie für die Zukunft hinsichtlich CI/CD und Automatisierung?

Ich möchte an dieser Stelle gerne noch einmal den State-of-DevOps-Report von 2021 zitieren: Die nach dem Bericht als Elite Performer eingeordneten Teams erreichen mittels DevOps und verwandter Techniken Dinge, die vor 10 Jahren noch unerreichbar schienen. Der Report zeigt auch, dass es immer mehr Teams gibt, die in die höchste Kategorie aufsteigen. Sie erreichen das fast paradox anmutende Ziel, durch häufige Releases zu sehr viel weniger Fehlern zu gelangen. Damit sind sie viel schneller mit ihrem Release am Markt. Eine Prognose für die Zukunft könnte so aussehen, dass sich der 2021 gezeigte Trend weiter fortsetzen wird. Mir ist aber noch wichtig hervorzuheben, dass CI/CD nicht die einzige Art ist, Anwendungen zu entwickeln, es kommt immer auf den Einzelfall an.



MARTIN BACH

Martin arbeitet als Produktmanager bei Oracle und hilft Kunden in Europa und überall auf der Welt ihre IT-Probleme zu lösen. Er interessiert sich insbesondere für Cloud-Technologien, DevOps, und wie diese sich am besten mit der Oracle-Datenbank kombinieren lassen.

Die Oracle- Anwenderkonferenz

2023
DOAG
Konferenz + Ausstellung

21. - 24.
Nov. 2023
Nürnberg



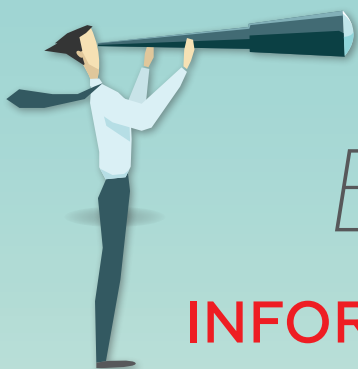
Eventpartner:

AOUG
AUSTRIAN ORACLE USER GROUP

SOUG

swiss oracle
user group

anwenderkonferenz.doag.org



AUS DER FERNE BETRACHTET:

INFORMATIK-FERNE GESELLSCHAFT

„Wissen hat im Prinzip nichts mit Informationstechnologie zu tun, sondern mit Gehirnen und noch mehr mit Verstand und Vernunft.“ (Fredmund Malik, österreichischer Wirtschaftswissenschaftler)

Hätten die für Schulfragen verantwortlichen Politiker in den Bundesländern und sonstige Meinungsmacher diesen ach so wahren Spruch doch besser beherzigt, bevor sie ohne Plan und ohne Strategie Tausende von iPads, Surface-Devices etc. gekauft und an Schüler und Schülerinnen verteilt haben. Geräte (Hardware) kaufen und verteilen ist das dünnste Brett, das für die Digitalisierung der Schule gebohrt werden muss, und ohne die dicken Bretter macht das Ganze nur wenig Sinn.

Aber diese Erkenntnis ist bei den wenigsten Entscheidungsträgern in Regierungen, Regierungspräsidien und Schulleitungen angekommen. Wie soll sie auch, wenn die Protagonisten zum großen Teil völlig frei von Informatik- oder IT-Wissen sind. Blank wie ein frisch gebohrter Dielenboden.

Nein, nicht jeder muss auf allen Gebieten Fachmann oder Fachfrau sein. Nicht jeder oder jede muss programmieren können und nicht jeder oder jede muss das kleine oder große Einmaleins der IT auswendig aufzusagen in der Lage sein. Aber man sollte erwarten können, dass wichtige Entscheidungen und Weichenstellungen für die Zukunft unserer Schulen flankiert werden durch sachkundige Analysen und Schlussfolgerungen. Und die gibt es! Die Gesellschaft für Informatik (GI) hat eine „Offensive Digitale Schultransformation“ ins Leben gerufen und die Data-Literacy-Charta hat das „Verständnis von Datenkompetenzen und

deren Bedeutung für Bildungsprozesse“ zum Ziel. Insbesondere die GI hat mit ihrem Sieben-Punkte-Plan den Weg hin zu einer modernen Schule detailliert aufgezeigt.

Es ist ja nicht so, dass wir hierzulande jetzt erst lernen, das Wort Informatik zu buchstabieren. Es gibt eine hervorragende Informatikausbildung an den Hochschulen, es gibt exzellente Fachleute in vielen Software-Unternehmen, in der Industrie und in der Verwaltung. Informatiker „Made in Germany“ sind gesuchte und hoch geachtete Spitzenkräfte, hierzulande aber auch international. Und doch muss man feststellen, dass die meisten Bundesbürger und viele Entscheidungsträger in der Politik, aber auch in der Industrie zu einer eher „Informatik-fernen“ Gesellschaftsschicht (engl. Digital Outsiders) gehören. Es geht hierbei jedoch keineswegs nur um die ältere Generation. Das Problem ist generations- und bildungsübergreifend. Wer kennt nicht das Kokettieren von „Schlaumeiern“ mit dem völligen Unverständnis und Unkenntnis der heutigen Technik-Welt. Oder die anderen, die sich bereits als IT-Spezialisten gerieren, weil sie ein iPhone in der Tasche tragen oder weil sie Alexa oder Siri überlisten können. Und die sogenannten „Digital Natives“ sind vielleicht flott bei der Bedienung ihrer Apps, aber bei den allermeisten hört das Verständnis damit auch bereits auf. Fakt ist, Informatik und Informations-Technologie ist für die meisten hierzulande ein Buch mit sieben Siegeln. Digitale Mündigkeit – Fehlanzeige!

Aber, man kann es drehen und wenden, wie man will, die akademische Disziplin Informatik ist eine der wichtigen Säulen für unsere Zukunft. Sie beeinflusst heute alle Wirtschafts- und Le-

bensbereiche und entwickelt sich in einem rasanten Tempo weiter. Es ist dringend geboten, dass zumindest Grundkenntnisse dieser wichtigen Wissenschaft allen gesellschaftlichen Schichten zur Verfügung gestellt werden.

Das Bundesministerium für Bildung und Forschung (BfBF) hat 2016 eine „Nationale Dekade der Alphabetisierung und Grundbildung“ ausgerufen. Ähnliches wäre für die Informatik und die Grundausbildung der Gesellschaft zu diesem Thema dringend erforderlich. Aber hier gilt, wie überall, dass Angebote auch angenommen werden müssen. Wie heißt es so schön: „*Das Genie des Wissenden ist ohne das Genie des Verstehenden wertlos!*“

Dass in den Schulen das Fach Informatik immer noch keinen flächendeckenden Einzug gehalten hat und dass es heute kaum (wenn überhaupt) ausgebildete Informatiklehrer gibt, ist eine ernüchternde Erkenntnis. Und dort, wo es die Informatik in die Unterrichtsräume geschafft hat, stehen den Schülern meist völlig fachfremde und auch sonst von IT eher unbelegte Lehrer gegenüber. Dass dabei Begeisterung und Interesse aufkommt – bei Schülern wie bei Lehrern – ist eher unwahrscheinlich.

Das Schulfach Informatik muss vom Abstellgleis ins Zentrum gerückt werden und Schulen müssen dem heutigen Stand der Technik so angepasst werden, dass sie der rasanten Entwicklung, der wir uns alle gegenübersehen, gewachsen sind.

Die Vorschläge der Gesellschaft für Informatik (GI) liegen auf dem Tisch. Man muss sie nur lesen, verstehen und umsetzen, denn auch hier gilt der alte Sponti-Spruch: „*Machen ist wie wollen, nur krasser!*“.

Links zu diesem Thema:

<https://gi.de/meldung/handlungsempfehlungen-fuer-die-bildungspolitik-gi-startet-offensive-digitale-schultransformation>

<https://offensive-digitale-schultransformation.de/>

<https://www.stifterverband.org/charta-data-literacy>

<https://www.wissenschaftsrat.de/download/2020/8675-20.html>



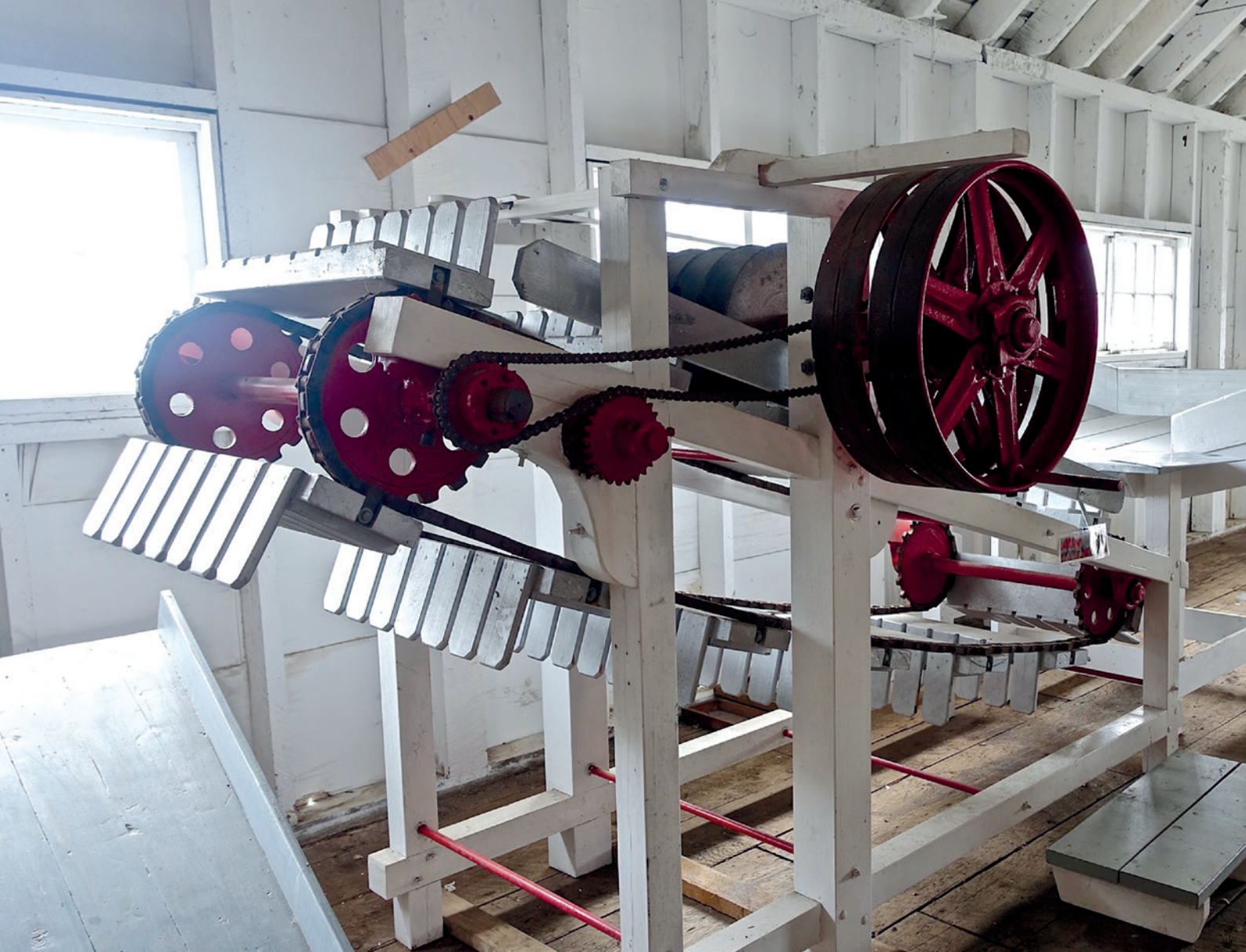
European NetSuite User Days



IN NÜRNBERG

netsuite.doag.org

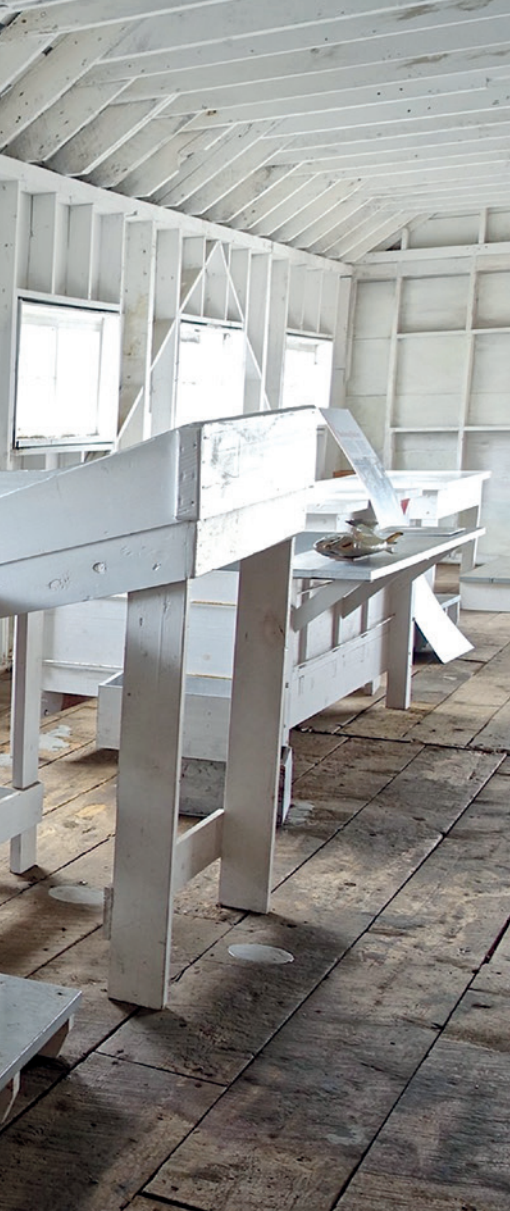
DOAG



Automatationen und Best-Practices im Betrieb eines OCI-Tenant

Marcus Schröder, Oracle Deutschland

Selbstverständlich stehen bei dem Anbietermodell einer Cloud die einfache Nutzung von Services, Agilität und Kosteneffizienz im Mittelpunkt. Aber auch in einer Cloud-Umgebung gibt es eine Reihe von administrativen Aufgaben, die es zu automatisieren gilt. Dieser Artikel handelt von den Möglichkeiten der Automatisierung von administrativen Aufgaben in einem Oracle Cloud Infrastructure Tenant. Die hier vorgestellten Services und Skripte stellen eine Momentaufnahme dar, das heißt, in Zukunft können weitere OCI-Services oder Skripte hinzukommen, die die momentanen Lösungen ersetzen beziehungsweise erweitern.



- Benutzerverwaltung
- Kostenüberwachung
- Kapazitätsmanagement
- Support
- Durchsetzen von Sicherheitsrichtlinien

Diese Tätigkeiten haben Abhängigkeiten und Prioritäten, die bei der Toolauswahl berücksichtigt wurden. Links auf Dokumentationen der Servicefunktionalität und der verwendeten Skripte finden Sie am Ende dieses Artikels.

Nutzung der OCI-Servicefunktionalität

Die OCI stellt eine Reihe von Services bereit, um alltägliche administrative Aufgaben zu automatisieren. Vorteil der Nutzung von OCI-Services sind der offizielle Support, die Integration in die OCI-Infrastruktur, die Sicherheit bezüglich Authentifizierung und Autorisierung sowie die ständige Weiterentwicklung und Optimierung.

Group Mapping – Gruppensynchronisierung zwischen IDCS und IAM

Für die Einbindung in ein allgemeines zentrales Benutzermanagement ist das interne IAM-System (Identity & Access Management) der OCI durch eine Federation mit einem übergeordneten Identity Management Service verbunden. Dies ist bei älteren Tenants der OCI „per Default“ der Oracle Identity Cloud Service (IDCS). Es kann aber auch ein Identity Management Service eines anderen Anbieters sein (z. B. MS Active Directory Azure etc.).

Dabei werden Benutzer und Gruppenmanagement innerhalb des IDCS gesteuert und das Rechteverwaltung im IAM der OCI. Die Synchronisation der Benutzer und Gruppen zwischen IDCS und IAM stellt die Grundlage der gesamten Verwaltung dar. Da die Gruppen im IDCS und IAM unterschiedlich bezeichnet werden können, erfolgt die Gruppenzuordnung für die Synchronisation manuell. Ist diese Synchronisation eingerichtet, werden die Gruppenmitglieder der IDCS-Gruppen automatisch in die IAM-Gruppen synchronisiert und erhalten alle Rechte, die

für die IAM-Gruppen auf der OCI-Seite eingerichtet wurden.

Die beiden Gruppen müssen vor der Zuordnung im IDCS und in der OCI angelegt werden. Die Zuordnung der Gruppen wird in der OCI-Konsole unter dem Punkt „Identity & Security“ > „Federation“ > „Group Mappings“ eingerichtet.

Die Trennung von der Benutzergruppenzugehörigkeit eines zentralen Identity-Systems (in diesem Fall IDCS) und der spezifischen Zuordnung zu der jeweiligen OCI-Gruppe deckt zum Beispiel folgende Anwendungsfälle ab:

- Bulk-Load von neuen Benutzern und Gruppen: Dies ist in älteren OCI-IAMs nicht möglich und erfolgt über das IDCS oder gegebenenfalls eine andere Federation, wie zum Beispiel MS Active Directory.
- Zentrales Zurücksetzen des Passwortes mittels Änderungsmail.
- Zentrales Inaktivieren oder Löschen eines Benutzers.
- Auslesen des Security-Logs.
- Und vieles mehr.

Default Tagging – Tag-Zuordnung von Ressourcen

Eines der wichtigsten Hilfsmittel bei der Administration von modernen agilen Umgebungen stellt das Tagging dar. Einem Objekt werden definierte oder freie Tags angehängt, die als Schlüssel-/Werte-Paare Metadaten enthalten.

So können beispielsweise bei der Erstellung eines neuen Datenbankservice der Erstellungszeitpunkt und der eingeloggte Benutzer via Tag der DB-Instanz zugeordnet werden. Da das in der Praxis nicht von jedem Benutzer manuell eingetragen wird, kann dieser Vorgang in der OCI automatisiert werden.

Es können sogenannte Default-Tags erzeugt werden, die automatisch während des Bereitstellungsprozesses mit der Ressource verknüpft werden.

Um die Default-Tags zu erstellen, muss im ersten Schritt ein Tag-Name-space erzeugt werden. Dieser sollte sich im Root-Compartment befinden, da er für den gesamten Tenant gültig ist.

In diesem Namespace werden für die Default-Tags etwa folgende zwei Key-Definitionen angelegt. Eine für den angemel-

Die Erfahrungen, die zur Auswahl der aufgeführten Services und Tools geführt haben, beruhen auf vier Jahren Betrieb von verschiedenen Oracle-internen OCI-Tenants mit 100 bis 200 Benutzern. Die gesamte Administration erfolgte als Nebenaufgabe, daher ist es das Ziel, einen möglichst hohen Automationsgrad zu erreichen. Es werden zwei Lösungsansätze verfolgt:

1. Nutzung von OCI-Service-Funktionalitäten
2. Einsatz von Skripten

Die administrativen Aufgaben werden nicht alle von Servicefunktionalitäten der OCI abgedeckt, daher werden für bestimmte Aufgaben Skripte verwendet, die von Oracle-Mitarbeitern und externen OCI-Experten entwickelt wurden.

Die Administration eines OCI-Tenant lässt sich in diesem spezifischen Fall in folgende Gruppen einteilen:

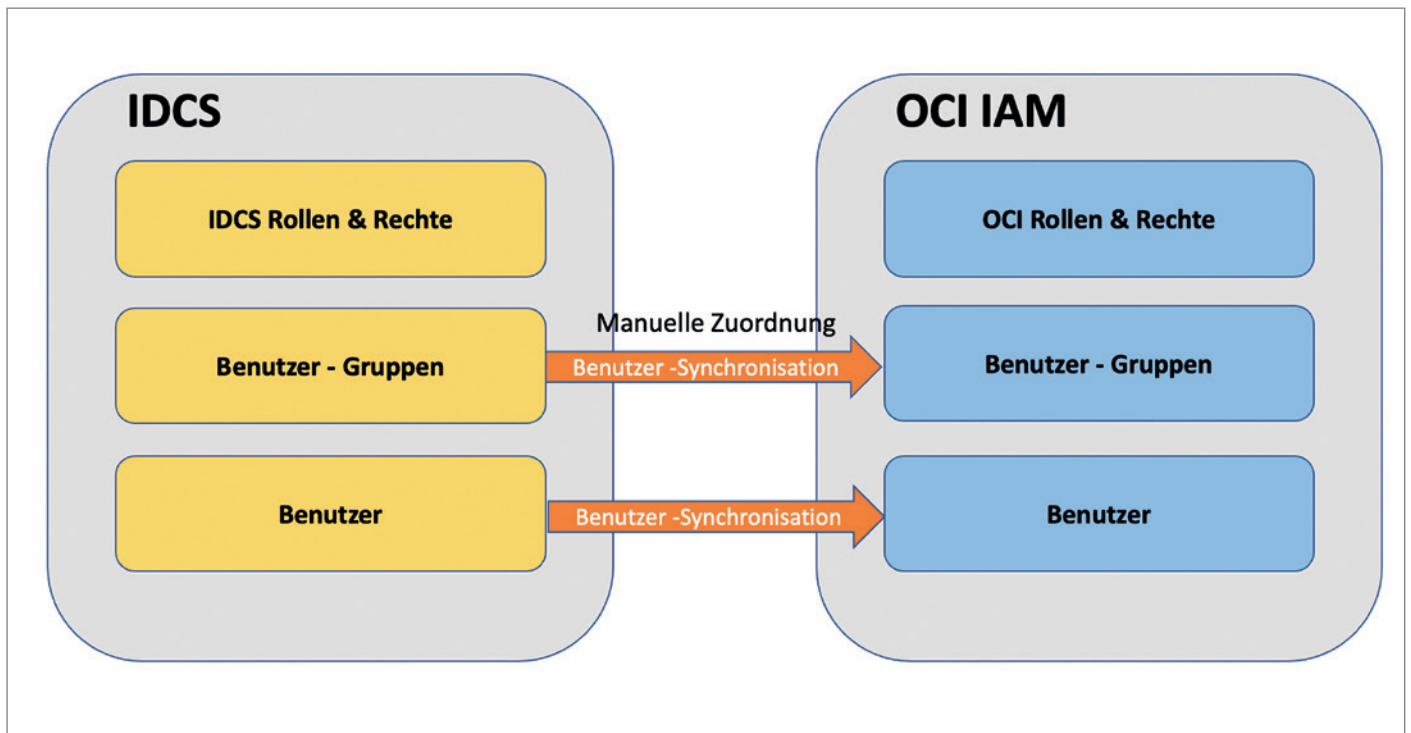


Abbildung 1: Schemadarstellung Benutzer, Gruppen und Rollen im IDCS und IAM (Quelle: Marcus Schröder)

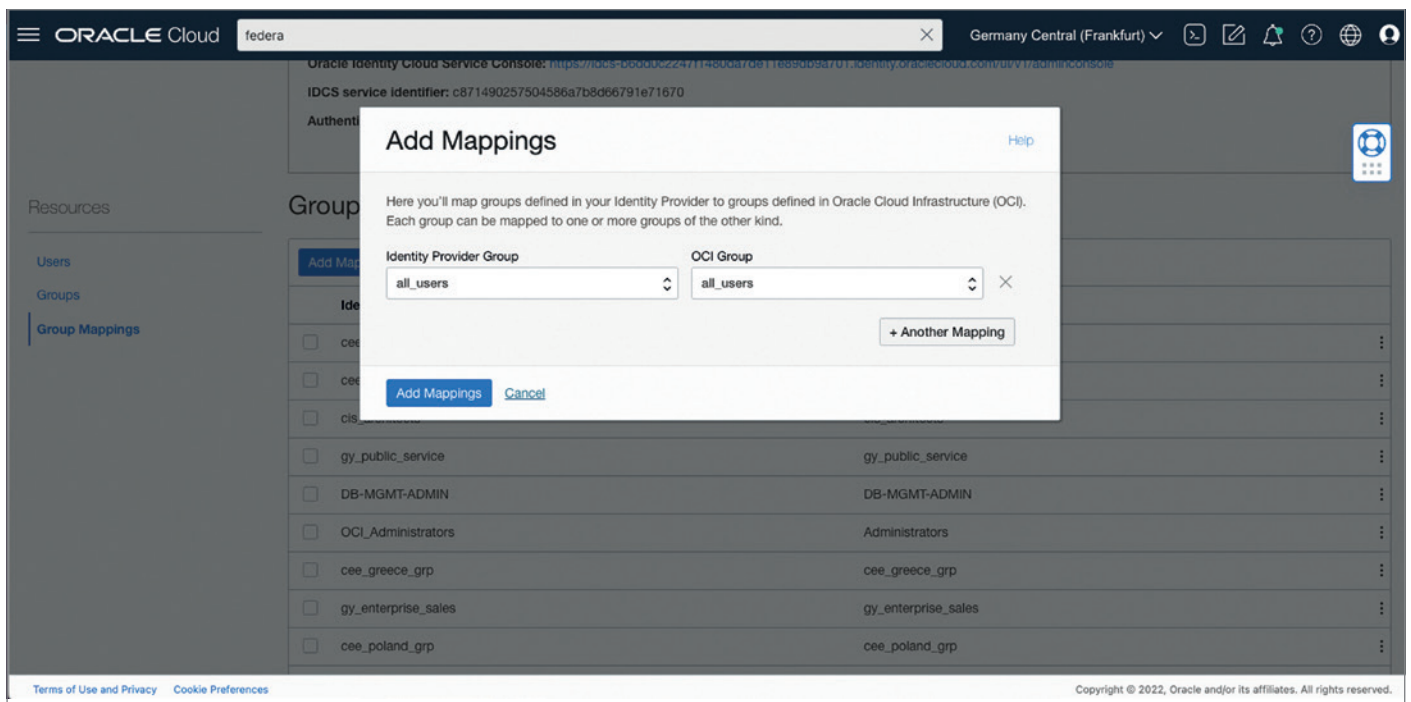


Abbildung 2: Screenshot – Group Mapping (Quelle: Marcus Schröder)

den Benutzer (CreatedBy) und eine für die Erstellungszeit (CreationDate).

Die Default-Tags werden über das Compartment-Menü erstellt. Dazu navigiert man über „Identity & Security“ > „Compartments“ und wählt das Root-Compartment aus. Unter dem linken Menüpunkt „Tag Defaults“ werden die zwei Key Definitions ausgewählt und mit

den Variablen $\${iam.principal.name}$ und $\${oci.datetime}$ belegt. Anschließend wird jede Ressource bei der Erstellung automatisch mit aktuellem Benutzernamen und Datum belegt.

An dieser Stelle ein kleiner Exkurs zum Thema „Navigation innerhalb der OCI“: Durch den Einsatz von Default-Tags kann die allgemeine Suche der OCI sehr

effektiv verwendet werden. Sucht man zum Beispiel nach allen Ressourcen des Benutzers captain.jack.sparrow@black-pearl.de, gibt man diesen Suchbegriff oder Teile davon in die obere Suchleiste ein und erhält alle Ressourcen mit diesem Tag-Schlüssel. Das Ergebnis kann man anschließend nach Servicearten filtern. Dies funktioniert nicht nur mit Tags, son-

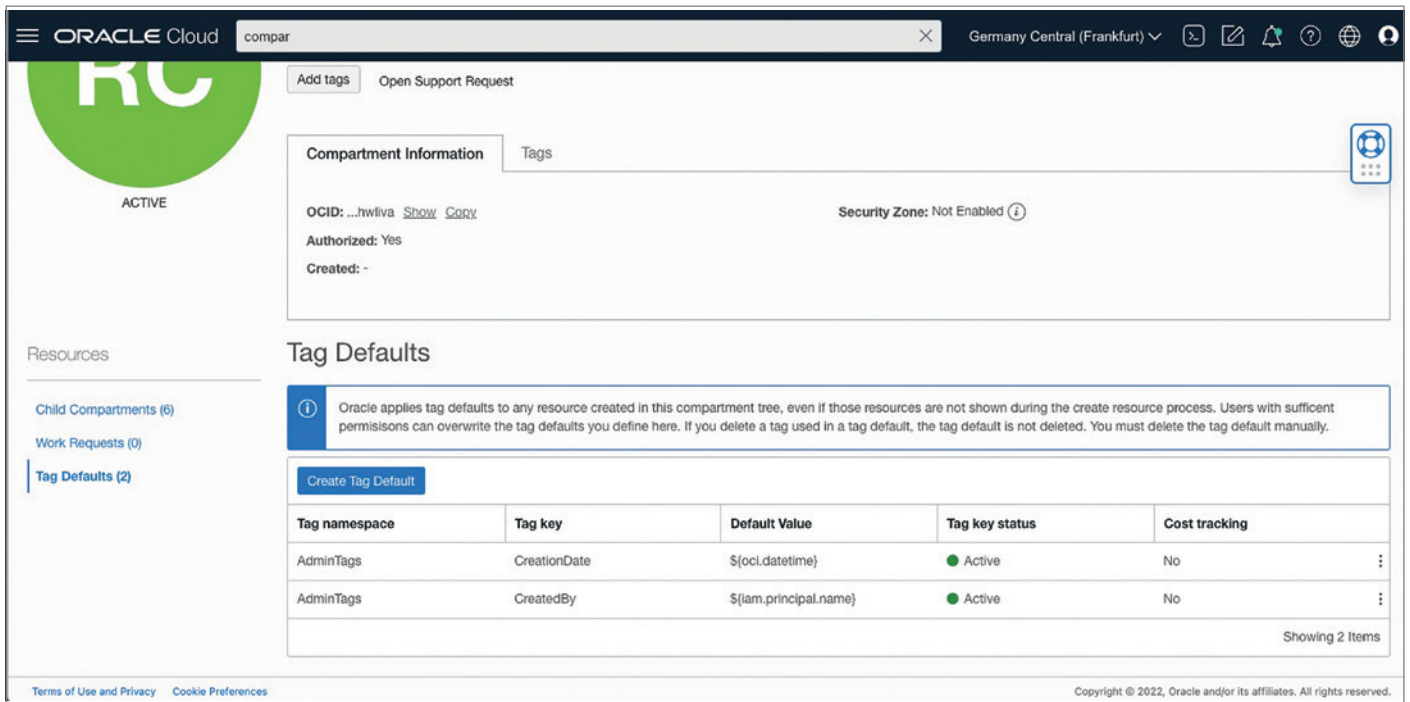


Abbildung 3: Screenshot – Tag Defaults (Quelle: Marcus Schröder)

den auch mit Servicebezeichnung, Service-namen etc.

Persönlich verwende ich bei der täglichen Arbeit mit der OCI ausschließlich die Suchfunktion. Wenn man weiß, was man sucht, kann man sich sehr viel Zeit sparen.

Die Verwendung von Tags und der Schnellsuche decken zum Beispiel folgende Anwendungsfälle ab:

- Schnelle Navigation auf Ressourcen, Services, Dokumentation und Marketplace
- Zuordnung von Ressourcen zu deren Erstellern bei der Fehlersuche, Kostenzuordnung und beim gezielten Löschen
- Und vieles mehr

Budgets – Einrichten von Kostenschwellwerten

Budgets ermöglichen die automatische Gesamtkostenüberwachung in einem OCI-Tenant. Im monatlichen Zyklus werden die Gesamtkosten mit einem definierten Schwellwert verglichen. Bei einer sich anbahnenden oder aktuellen Überschreitung erfolgt eine automatische Benachrichtigung per E-Mail. Der Beginn des monatlichen Zyklus ist frei wählbar, da nicht jedes Unternehmen am Ersten eines Monats abrechnet. Der Kostenschwellwert kann absolut oder prozentual angegeben werden und richtet sich nach der aktu-

ellen Währung des Tenant. Eine weitere Funktion ist die Berechnung des Kostenschwellwertes nach der Forecast-Methode. Diese kalkuliert die Kosten beruhend auf den aktuellen Verbrauchswerten bis zum Ende der Periode und meldet eine sich anbahnende Übertretung. Budgets werden immer auf Compartment-Ebene bezogen, zur Betrachtung der Gesamtkosten eines Tenant wird ein Budget im Root-Compartment konfiguriert.

Die Benachrichtigung erfolgt ausschließlich per E-Mail, deren Inhalt individuell gestaltet werden kann. Um zur Konfiguration des Budgets zu gelangen, reicht es, in die Suchleiste das Wort „Budget“ einzugeben und anschließend den Service auszuwählen.

Die Verwendung von Budgets deckt zum Beispiel folgende Anwendungsfälle ab:

- Benachrichtigung bei Überschreitung einer monatlichen Kostengrenze
- Benachrichtigung einer sich anbahnenden Überschreitung
- Automatische Kostenkontrolle
- Und vieles mehr

Quota Policies – Einrichten von Ressourcenbegrenzung

Mit den Quota Policies kann der Administrator die bestehenden Hard-Limits

des Tenant zusätzlich einschränken. Die Hard-Limits werden bei der initialen Erstellung eines Oracle-OCI-Tenant gesetzt und limitieren die maximal mögliche Anzahl von Service-Instanzen. Hintergründe sind die Kostenkontrolle auf der Kundenseite sowie das Kapazitätsmanagement für den Oracle-Cloud-Betrieb. Reichen die initial gesetzten Limits nicht aus, können sie per Service Request erweitert werden.

Die Hard-Limits erfüllen jedoch nicht alle Anforderungen, so kann es etwa sein, dass bestimmte Services stärker limitiert werden sollen oder ein ausreichender Puffer vorhanden sein muss, da die Erweiterungsanforderung eines Hard-Limits per Service Request eine gewisse Laufzeit hat.

Diesen Zweck erfüllen Soft-Limits oder Quota Policies. Diese werden durch Regeln definiert, die als Quota Policy abgespeichert werden. Die Regeln werden in einer spezifischen Syntax definiert, es gibt jedoch einen Trick, wie man das korrekte Statement erhält und nur noch die Limits eintragen muss. Dazu geben wir in die Suchleiste den Begriff „Limits“ ein, wählen den Service „Limits, Quotas and Usage“ aus, suchen den gewünschten Service auf regionaler Ebene und/oder in der Availability Domain und klicken auf die drei Punkte am Ende der Zeile. Dann wählen wir „Create Quota Policy Stub“ aus, kopieren das Statement und fügen es unter „Quota Policies“ ein. Anschlie-

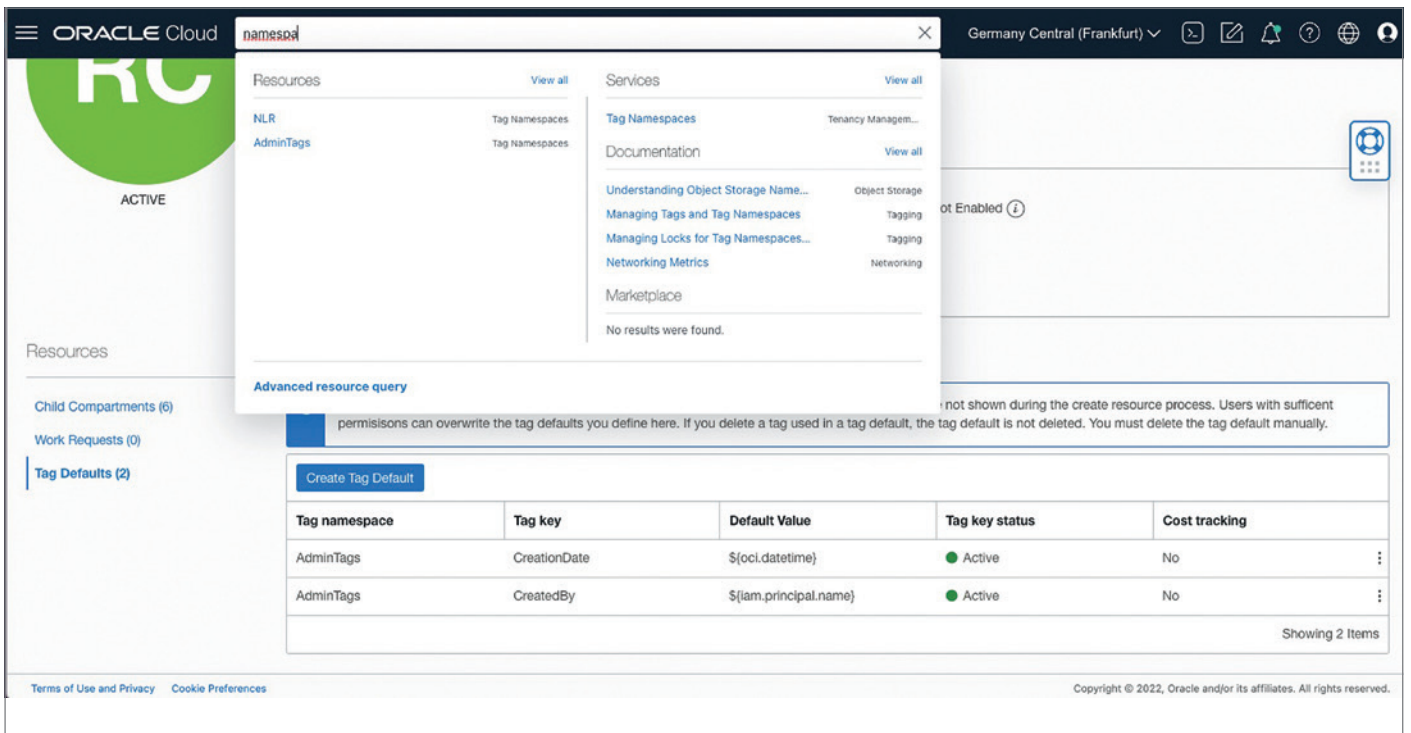


Abbildung 4: Screenshot – Suche nach Namespace (Quelle: Marcus Schröder)

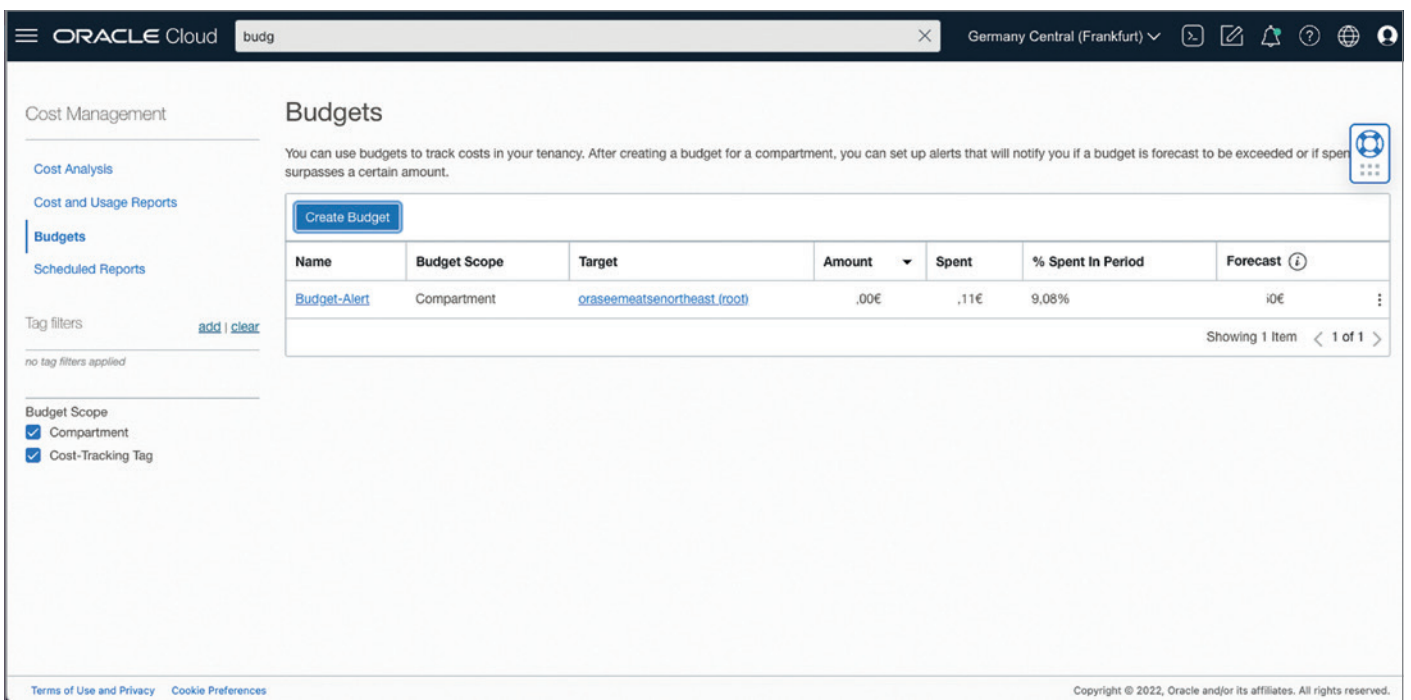


Abbildung 5: Screenshot – Budgets (Quelle: Marcus Schröder)

ßend können wir die Limits in dem State-ment anpassen.

Trotzdem lohnt es sich, in die Doku-mentation zu schauen, denn Quota Poli-cies können auch auf ganze Servicefa-milien angewendet werden. Wenn etwa ausschließlich bestimmte Prozessorty-pen verwendet werden sollen, kann man durch die Syntax erst alle Prozessorty-

pen auf null setzen, um anschließend das Limit nur für den gewünschten Pro-zeessortyp auf den gewünschten Grenzwert zu setzen. Im Folgenden werden für Compute-Instanzen alle Prozessortypen in der Tenancy auf null gesetzt, um anschließend 240 E4-Prozessor-Kerne zu-zulassen.

zero compute-core quotas in tenancy

Set compute-core quota standard-e4-core-count to 240 in tenancy

In einem anderen Beispiel wird durch ein Statement verhindert, dass in einem bestimmten Compartment E-Mails oder Benachrichtigungen gesendet werden können.

zero email-delivery quotas in compart-ment SecureCompartment

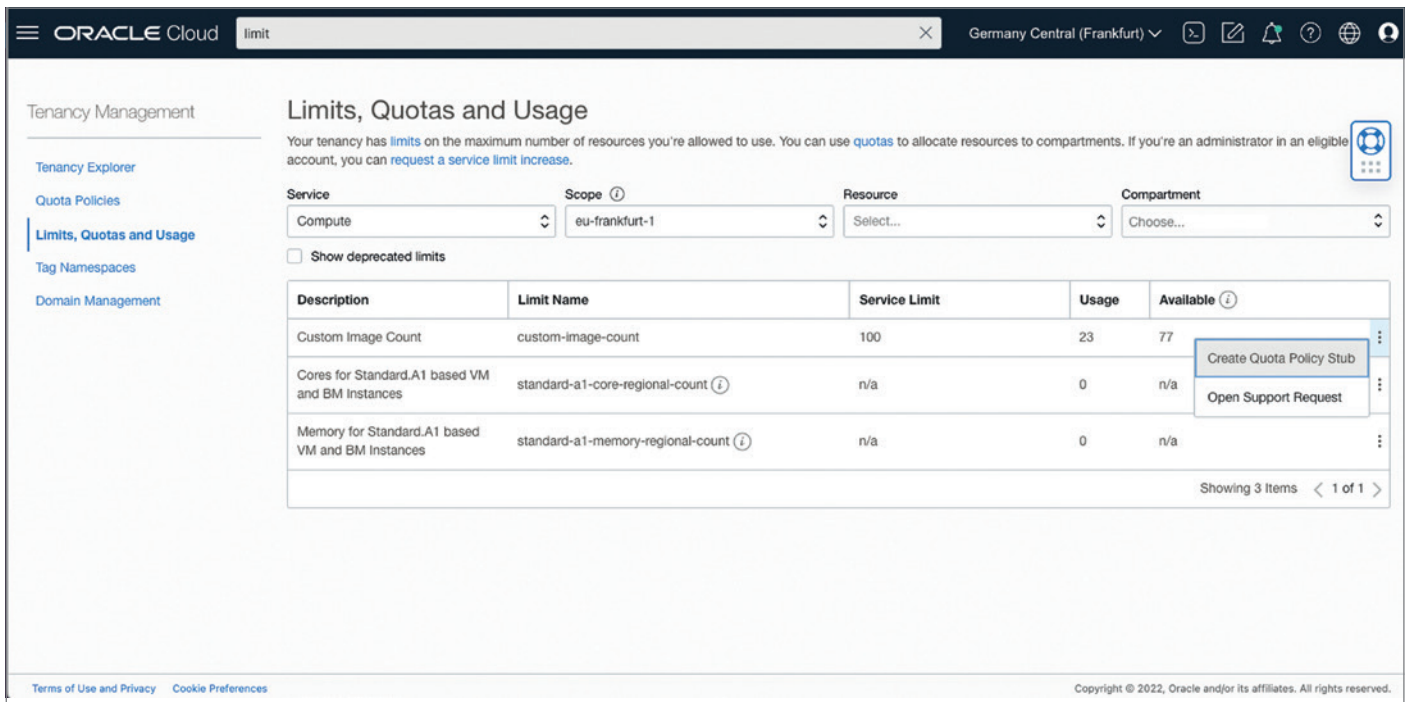


Abbildung 6: Screenshot – Erstellen eines Quota Policy Stub (Quelle: Marcus Schröder)

zero notifications quotas in compartment SecureCompartment

Diese Beispiele zeigen, was für ein flexibles Tool die Quota Policies darstellen und wie der OCI-Tenant dadurch sicherer und kontrollierbarer konfiguriert werden kann.

Die Verwendung von Quota Policies deckt beispielsweise folgende Anwendungsfälle ab:

- Einschränkungen bestimmter Services für Compartments
- Freie Limitierung von Serviceinstanzen
- Erhöhen der Sicherheit
- Kontrolle von Kosten
- Und vieles mehr

Nutzung von Skripten

Die OCI stellt weitere Services bereit, um administrative Aufgaben zu automatisieren, die allerdings für meine persönlichen und spezifischen Anforderungen nicht relevant sind.

Für weitere administrative Aufgaben werden Skripte verwendet, die aktuell nicht durch bestehende OCI-Services abgedeckt werden. Es ist wahrscheinlich, dass die fehlenden Funktionalitäten in naher Zukunft von neuen OCI-Services abgedeckt werden und die Skripte dadurch ersetzt werden.

Die administrativen Aufgaben werden von frei verfügbaren Skripten abgedeckt. Wichtig hierbei: Die Skripte sind Open Source, das heißt, es gibt dafür keinen offiziellen Support von Oracle. Da die verwendeten Skripte in Python geschrieben wurden und das offizielle OCI-API nutzen, gelten für die Nutzung die allgemeinen Sicherheitsrichtlinien der OCI. Die Skripte können lokal auf einem Rechner oder auf einer Compute-Instanz in der OCI konfiguriert und verwendet werden. Tipp: Die Nutzung der Always-Free-Compute-Instanzen der OCI lässt keine zusätzlichen Kosten entstehen.

Autostopping – geplantes Herunterfahren von Services

Einer der größten Kostenfaktoren in der Cloud ist die Vergesslichkeit ihrer Nutzer. Viele Benutzer legen Services an, verwenden diese einige Male und vergessen anschließend, diese zu löschen beziehungsweise herunterzufahren. Deshalb birgt das Stoppen von Services in regelmäßigen Abständen ein großes Kosteneinsparpotenzial. Es existieren verschiedene Skripte, die ein geplantes Herunterfahren ermöglichen. Die Funktionen reichen vom spezifischen Stoppen einzelner Instanzen bis zum Stoppen ganzer Service-Familien.

In der von mir betriebenen Umgebung wird das Autostopping-Skript verwendet. Das Skript stoppt alle Datenbank- und Compute-Instanzen jede Woche Freitagnacht. Die Benutzerinnen und Benutzer können bei Bedarf die Umgebung jederzeit manuell wieder hochfahren. Für Ausnahmen bei kritischen Umgebungen wird ein benutzerdefiniertes Tag AUTOSTOPPING auf NO gesetzt. Das Stoppen wird durch einen Linux-Cron-Job initiiert, der auf Freitagnacht konfiguriert ist. Das Skript läuft auf einer Always-Free-Compute-Instanz, die das benutzerdefinierte Tag AUTOSTOPPING NO besitzt, um zu verhindern, dass die Arbeitsinstanz ebenfalls herunterfährt.

Die Verwendung von Autostopping deckt zum Beispiel folgende Anwendungsfälle ab:

- Einsparung von Kosten
- Einsparung von Kapazitäten
- Und vieles mehr

Bulk Delete – OCI-SuperDelete

Eine große und zeitintensive Herausforderung stellt das Terminieren von Compartments dar. Es gibt aus Sicherheitsgründen keine Funktion in der OCI, die ein Compartment samt darin enthaltenen

Services terminiert, obwohl dies manchmal nötig ist.

Wenn zum Beispiel jemand das Unternehmen verlässt oder ein Tenant stillgelegt wird, muss sichergestellt werden, dass keine weiteren Kosten entstehen. Ein erster Schritt ist, die Services des Benutzers beziehungsweise alle Services zu stoppen. Trotzdem bleiben die Storage-Kosten und einige Services nur dann kostenfrei, wenn sie terminiert sind.

Das OCI-SuperDelete-Skript ermöglicht das Löschen von Compartments inklusive der Sub-Compartments und deren darin enthaltenen Services.

Das Skript ist einfach zu bedienen. Man gibt das Top-Level-Compartment, die OCI-Regionen und einige zusätzliche optionale Flags an. Das Skript läuft ohne Interaktion und löscht alle in dem Compartment vorhandenen Services. Dies kann eine gewisse Zeit dauern, daher sollte man das Skript von einer laufenden Always-Free-Instanz im Hintergrund starten.

Die Verwendung von OCI-SuperDelete deckt zum Beispiel folgende Anwendungsfälle ab:

- Einsparung von Kosten
- Einsparung von Kapazitäten
- Durchsetzung von Sicherheitsrichtlinien
- Und vieles mehr

Limitkontrolle – SnapshotComparisons

Dieses Skript habe ich lange gesucht und war kurz davor, es selbst zu schreiben. Eine Übersicht über alle Hard-Limits bezogen auf Regionen und Availability Domains zu erhalten, kann in der OCI-GUI eine zeitraubende Arbeit sein. Erschwerend kommt hinzu, dass es kritisch ist, in einem Cloud-Projekt an ein Hard-Limit zu stoßen. Die Hard-Limits müssen per Support Service Request erhöht werden, was eine gewisse Zeit benötigt, die man in kritischen Projekten oft nicht hat.

Das SnapshotComparisons-Skript liest alle Limits des Tenant aus und speichert sie in einem MS-Excelsheet (XLSX-Format). Das Skript kommt ohne Parameter aus; das einzige, was geändert werden muss, ist der Ablagepfad des Excelsheets. Dafür muss der Parameter usage_location in der Datei GetSnapshots.py editiert werden. Optional können die Regionen reduziert werden, für die die Limits ausgelesen werden, dies ist der Parameter scope_list in der gleichen Datei. Das Programm arbeitet menübasiert, nach dem Aufruf von „python3 SnapshotComparisons“ hat man folgende Auswahlmöglichkeiten:

- Get info in these regions:
 - all
- Regions subscribed by oraclenetsuite tenancy:
 - eu-amsterdam-1
 - eu-frankfurt-1
- Please choose an option:
 - 1. Tenancy Limit
 - 2. Quota Policies
 - 3. Compartment Usage
 - 4. Quit

Um das Excelsheet zu erzeugen, muss Punkt 1 ausgewählt werden. Wichtig: Die Limits werden an das Terminalfenster ausgegeben und erst in das Excelsheet geschrieben, wenn die Ausgabe vollständig erfolgt ist und man die 4 für Quit gedrückt hat. Also: Weder die Ausgabe abbrechen noch das Programm verlassen, ohne Quit (4) auszuwählen, ansonsten werden die Limits nicht in das Excelsheet geschrieben.

Ein kleiner Hinweis zum Excelsheet: Leider hat der Programmierer des Programms vergessen, eine prozentuale Ansicht der noch verfügbaren Service-Instanzen als Spalte hinzuzufügen. Möchte man diese doch sehr wichtige Spalte hinzufügen, kann man das mit der Formel =G2/F2*100 in Excel tun. Nachdem man absteigend sortiert hat, ist in den Prozent-



Abbildung 7: Screenshot – Log-Datei-Ausschnitt des Stoppens der Instanzen (Quelle: Marcus Schröder)


```

OCI-SuperDelete -- opc@shutdown:~/ocicli/autostopping/log -- -zsh -- 187x42
how to run:
...
usage: delete.py [-h] [-cf CONFIG_FILE] [-cp CONFIG_PROFILE] [-force] [-debug] [-rg REGIONS] [-c COMPARTMENT]

optional arguments:
  -h, --help            show this help message and exit
  -cp CONFIG_PROFILE    Config Profile inside the config file
  -ip                   Use Instance Principals for Authentication
  -dt                   Use Instance Principals with delegation token for cloud shell
  -force                force delete without confirmation
  -debug               Enable debug
  -rg REGIONS           Regions to delete comma separated
  -c COMPARTMENT        top level compartment id to delete
  -skip_delete_compartment Skip Deleting the compartments at end of the process

python3 delete.py -c <CompartmentID>
...

## WORK_IN_PROGRESS
This script is continuously being maintained based on newly released services. If you are noticing this script is missing a particular service, please let us know, so we can add this.

This script has been massive overhauled in January/February 2022. Instead of having a complete python module for each service, most services are now being deleted via the AnyDelete method included in this script.

The KMS Vaults and Keys can not instantly be deleted, but require a minimal 7 day grace period. The script will move all keys and vaults to the root compartment and will schedule the deletion with 7 days grace period. This will allow all sub compartments to be instantly deleted, while the top compartment will only be able to be deleted after the grace period.

## Purpose
The purpose of this script is to remove all resources from a compartment, including subcompartments. In OCI you can only remove a compartment when it contains no more resources, but it can be a challenge to find all the resources tied to a compartment.

This script will hunt for all resources in a compartment and delete/terminate/retire them.

## TODO
Currently this script only support authentication via config file. Plan is to add instance principle support and cloud shell authentication support.

## Disclaimer
This is a personal repository. Any code, views or opinions represented here are personal and belong solely to me and do not represent those of people, institutions or organizations that I may or may not be associated with in professional or personal capacity, unless explicitly stated.

MARSCHRO@MARSCHRO-mac OCI-SuperDelete %

```

Abbildung 8: Screenshot – Super-Delete-Skript-Optionen (Quelle: Marcus Schröder)

	A	B	C	D	E	F	G	H	I	J	K
1	name	description	limit_name	availability_domain	scope_type	limit	used	available	Limit Erreichung in %		
2	database	Database	apex-total-storage-tb		REGION	8	6	2	75		
3	database	Database	adw-ocpu-count		REGION	20	13	7	65		
4	database	Database	vm-standard2-ocpu-count	AMSW-EU-FRANKFURT-1-AD-2	AD	30	19	11	63		
5	analytics	Analytics	ee-ocpu-count		REGION	10	6	4	60		
6	database	Database	vm-standard2-ocpu-count	AMSW-EU-FRANKFURT-1-AD-1	AD	30	17	13	57		
7	vcn	Virtual Cloud Network	vcn-count		REGION	150	83	67	55		
8	vcn	Virtual Cloud Network	reserved-public-ip-count		REGION	75	38	37	51		
9	analytics	Analytics	se-ocpu-count		REGION	4	2	2	50		
10	filesystem	File Storage	mount-target-count	AMSW-EU-FRANKFURT-1-AD-1	AD	4	2	2	50		
11	database	Database	vm-standard2-ocpu-count	AMSW-EU-FRANKFURT-1-AD-3	AD	30	14	16	47		
12	database	Database	atp-ocpu-count		REGION	30	12	18	40		
13	kms	Key Management	virtual-vault-count		REGION	10	4	6	40		
14	database	Database	ajd-ocpu-count		REGION	8	3	5	38		
15	database	Database	ajd-total-storage-tb		REGION	8	3	5	38		
16	load-balancer	LbaaS	lb-100mbps-count		REGION	100	33	67	33		
17	database	Database	apex-ocpu-count		REGION	20	6	14	30		
18	database	Database	atp-total-storage-tb		REGION	40	12	28	30		
19	database	Database	adw-total-storage-tb		REGION	40	11	29	28		
20	data-science	Data Science	ds-block-volume-count		REGION	30	8	22	27		
21	filesystem	File Storage	mount-target-count	AMSW-EU-FRANKFURT-1-AD-2	AD	4	1	3	25		
22	database	Database	vm-standard-e4-ocpu-count	AMSW-EU-FRANKFURT-1-AD-1	AD	75	18	57	24		
23	compute	Compute	custom-image-count		REGION	100	23	77	23		
24	block-storage	Block Volume	total-storage-gb	AMSW-EU-FRANKFURT-1-AD-1	AD	75000	16070	58930	21		
25	network-load-balancer-api	Network Load Balancer	max-nlb-flexible-count		REGION	10	2	8	20		
26	database	Database	vm-standard-e4-ocpu-count	AMSW-EU-FRANKFURT-1-AD-2	AD	75	14	61	19		
27	filesystem	File Storage	file-system-count	AMSW-EU-FRANKFURT-1-AD-1	AD	100	12	88	12		
28	block-storage	Block Volume	total-storage-gb	AMSW-EU-FRANKFURT-1-AD-2	AD	75000	8564	66436	11		
29	cloud-shell	Cloud Shell	user-volumes		GLOBAL	400	38	362	10		
30	block-storage	Block Volume	total-storage-gb	AMSW-EU-FRANKFURT-1-AD-3	AD	75000	5478	69522	7		
31	dns	DNS	view-count		REGION	100	5	95	5		
32	logging	Logging	log-group-count		REGION	100	5	95	5		
33	visualbuilder	Visual Builder	instance-count		REGION	20	1	19	5		
34	compartments	Compartments	compartment-count		REGION	2000	79	1921	4		
35	resource-manager	Resource Manager	stack-count		REGION	1000	38	962	4		
36	database	Database	vm-standard-e4-ocpu-count	AMSW-EU-FRANKFURT-1-AD-3	AD	75	2	73	3		
37	logging	Logging	log-count		REGION	500	11	489	2		
38	blockchain	Blockchain	platform-count		REGION	100	2	98	2		
39	events	Events	rule-count		REGION	50	1	49	2		
40	waf	Web Application Firewall	policy-count		REGION	50	1	49	2		
41	database	Database	vm-block-storage-gb	AMSW-EU-FRANKFURT-1-AD-1	AD	200000	3072	196928	2		

Abbildung 9: Screenshot – Excel-Ausschnitt über die aktuellen Limits der Tenancy (Quelle: Marcus Schröder)

spalten ersichtlich, welche Hard-Limits man gegebenenfalls vergrößern sollte.

Die Verwendung von SnapshotComparisons deckt zum Beispiel folgende Anwendungsfälle ab:

- Einsparung von Kosten
- Einsparung von Kapazitäten

- Durchsetzung von Sicherheitsrichtlinien
- Und vieles mehr

Kostenanalyse – Usage2ADW

Das Thema: „Kostenanalyse/Kostenkontrolle“ ist in jeder Cloud-Umgebung von

immenser Wichtigkeit. Um die Kosten für die Cloud-Benutzung zu optimieren, müssen die angefallenen Kosten nach verschiedenen Aspekten analysiert werden. Beispielhafte Fragen sind

- Wie sind die laufenden Gesamtkosten?
- Welche Servicegruppen sind am teu-

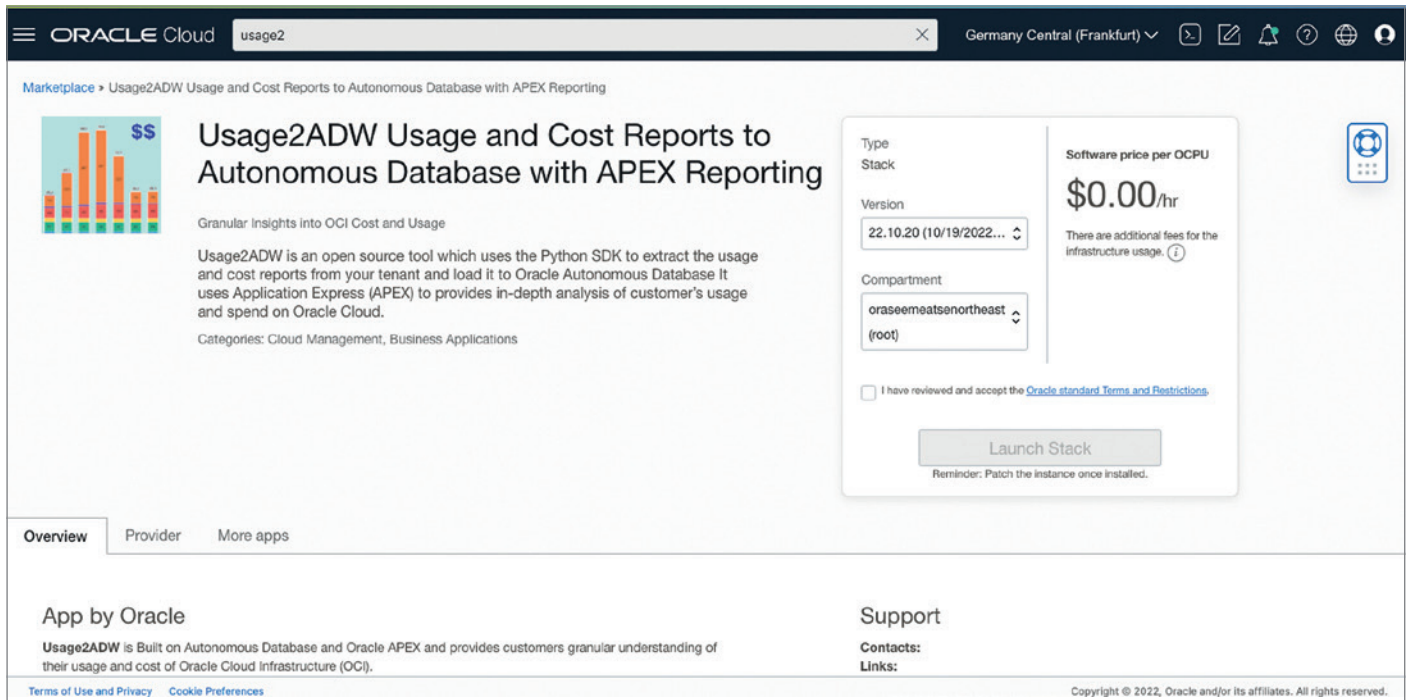


Abbildung 10: Screenshot – Usage2ADW-Anwendung im Marketplace (Quelle: Marcus Schröder)

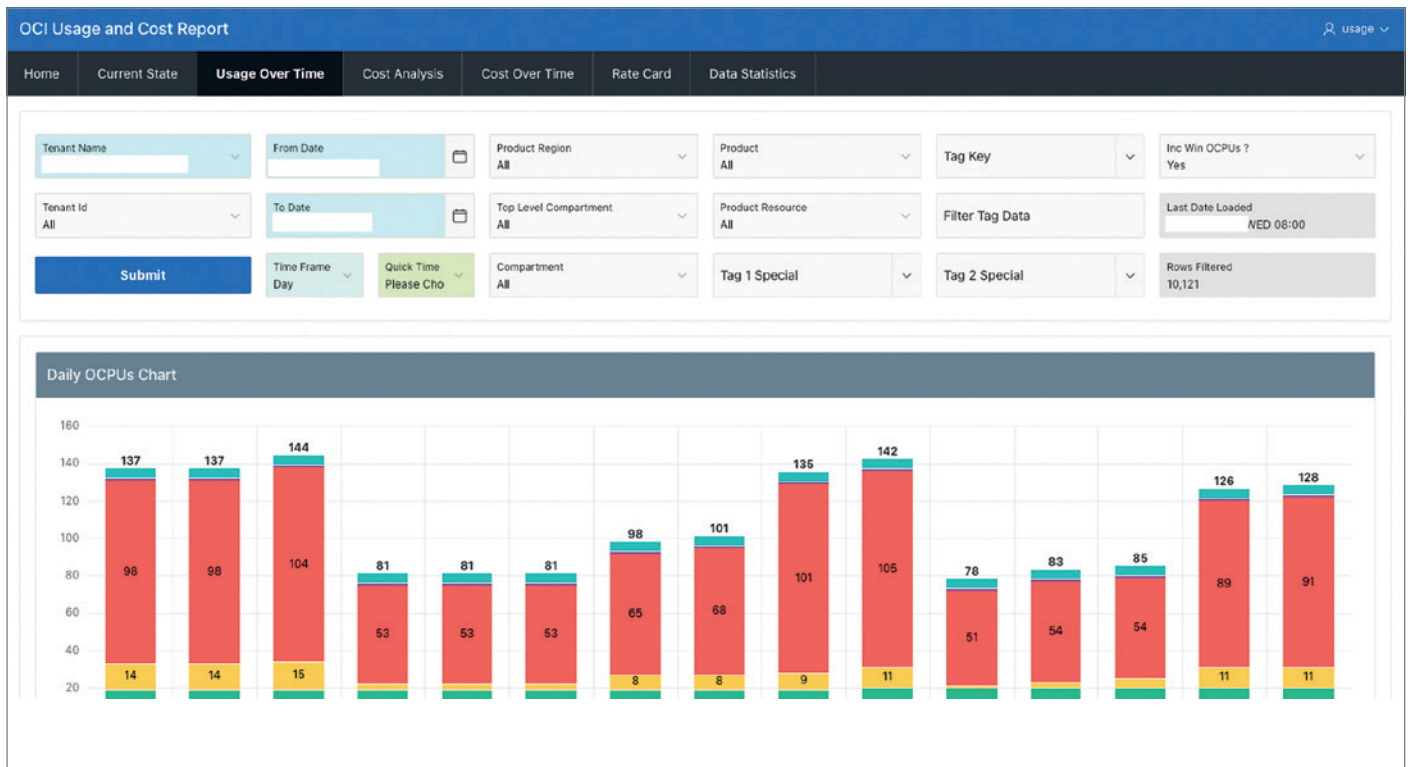


Abbildung 11: Screenshot – Usage2ADW-Analyse (Quelle: Marcus Schröder)

- ersten, kann man beispielsweise Umgebungen konsolidieren?
- Gibt es einen Benutzer, der ungewöhnlich hohe Kosten verursacht? Wie können diese reduziert werden?
- Wie ist die Entwicklung der Kosten?
- Welche Projekte kosten wie viel?
- Und vieles mehr

Alle diese Fragen können mit der integrierten OCI-Cost-Analyse beantwortet werden. Voraussetzung ist, dass der Analyst einen Benutzeraccount mit bestimmten Rechten haben muss, um auf die OCI-Konsole zugreifen zu können. Falls dies nicht gewünscht ist, gibt es eine Möglichkeit, die Kosten mithilfe einer

APEX-Anwendung zu analysieren. Diese Applikation sammelt die Rohdaten der Tenant-Kosten und des Verbrauchs und stellt diese Daten sehr übersichtlich aufbereitet mittels APEX-Anwendung für den Analysten bereit. Die APEX-Anwendung hat ein eigenes Benutzer-Management und ist vom OCI-IAM unabhängig.

Schon vor einigen Jahren, als die interne OCI-Kosten-Analyse noch nicht den heutigen Reifegrad erreicht hatte, entwickelte ein Kollege von Oracle eine APEX-Anwendung mit dem Namen Usage2ADW, die alle Kostenanalysen durchführen konnte. Mittlerweile hat die APEX-Anwendung einen derartigen Reifegrad erreicht, dass sie als Stack im Oracle Marketplace zur Verfügung gestellt wurde. Sie ist kostenfrei; wenn man sie auf einer „Always-Free-Instanz“ bereitstellt, fallen auch keine Compute- und Autonomous-Datenbank-Kosten an.

Die Anwendung findet man, indem man auf den OCI-Marketplace navigiert (einfach Marketplace in die Suchleiste eingeben) und in die Suche des Marketplace „Usage2ADW“ eingibt.

Die Anwendung ist als Terraform-Stack implementiert, das bedeutet, man gibt einmalig bestimmte Parameter an und kann anschließend die Umgebung beliebig erzeugen und wieder löschen. Wenn man zum Beispiel nur an bestimmten Tagen eine Analyse anbieten muss, kann man den Stack konfigurieren und die Umgebung nur an diesen Tagen bereitstellen, anschließend kann die Umgebung per Stack-Job wieder gelöscht werden.

Die Stack-Konfigurationsparameter sind selbsterklärend, es erfolgt an dieser Stelle nur der Hinweis, dass man bei der Erstellung des Applikationspasswortes genau die Vorgaben lesen sollte.

Anschließend den Apply-Job starten und die Umgebung installiert sich automatisch. Benötigt man die Umgebung nicht mehr, einfach zu den Stacks navigieren (Stack in die Suchleiste eingeben). Den Stack „Usage2ADW ...“ auswählen und auf Destroy klicken.

Die Terraform-Skripte erzeugen eine Compute- und eine Autonomous-Warehouse-Instanz, kurz ADW. Auf der Compute-Instanz werden die Kosten- und Verwendungs-CSV-Dateien geladen und anschließend in die ADW geladen. In der ADW läuft eine APEX-Instanz, in die die Daten aus den CSV-Dateien importiert werden. Jedes Mal, wenn die Usage2ADW-Umgebung erstellt wird, werden alle CSV-Dateien importiert. Deshalb dauert es einen Augenblick, bevor in den APEX-Anwen-

dungen Daten sichtbar werden. Die Kosten-/Nutzungsdaten sind, wie auch in der OCI, mit einem Tag Verzögerung sichtbar. Grund dafür ist, dass der Kosten-/Nutzungs-Batchlauf in der OCI einmal täglich läuft. Wenn alle Daten in die APEX-Applikation geladen worden sind, können folgende Analysen durchgeführt werden:

- Nutzungsanalyse
- Kostenanalyse auf benutzerdefiniertem Zeitraum
- Kostenanalyse auf einen vorgegebenen Zeitraum
- Rate-Card, Darstellung der aktuellen Discounts
- Statistiken über die importierten CSV-Dateien
- Tages-Nutzungs-Analyse

Die meisten Analysen können mit einer Reihe von Rahmenparametern spezifiziert werden, so kann der Zeitraum angegeben werden, die Compartments, spezielle Tags, Regionen, Service-Gruppen etc.

Die Verwendung von Usage2ADW deckt etwa folgende Anwendungsfälle ab:

- Kostenanalyse
- Durchsetzung von Sicherheitsrichtlinien
- Und vieles mehr

Fazit

Es existieren eine Reihe von Services und Tools, die die Administration eines kleinen bis mittleren OCI-Tenant so weit vereinfachen, dass man diese Aufgaben „nebenbei“ erledigen kann. Dieser Artikel zeigt nur einen kleinen Ausschnitt aus den aktuellen Möglichkeiten. Es gibt zusätzliche Services/Tools, die bei der Administration helfen, hier aber nicht zum Einsatz kommen. Ein kurzes Beispiel ist der Cloud Advisor. Durch diesen werden ungenutzte Ressourcen und die durch Terminierung möglichen Einsparungen aufgezeigt. In der weiteren Zukunft der OCI werden noch weitere Services hinzukommen, die die täglichen Arbeiten eines Administrators erleichtern, ich persönlich bin gespannt!

Weiterführende Literatur und Links auf die verwendeten Skripte

1. Group Mapping – Gruppensynchronisierung zwischen IDCS und IAM
<http://bit.ly/OCIGroupsync>
2. Default Tagging – Tag-Zuordnung von Ressourcen
3. Budgets – Einrichten von Kostenschwellwerten
http://bit.ly/OCIdefaultTag_budgets
4. Quota Policies – Einrichten von Ressourcenbegrenzung
<http://bit.ly/OCIquotas>
5. Autostopping – geplantes Herunterfahren von Services
<https://bit.ly/OCIautostopping>
6. Bulk Delete – OCI-SuperDelete
<http://bit.ly/OCISuperDelete>
7. Limitkontrolle – SnapshotComparisons
<http://bit.ly/OCILimit>
8. Kostenanalyse – Usage2ADW
<http://bit.ly/usage2adw>

Über den Autor

Marcus Schröder, Master Principal Sales Technology Solution Engineer bei der Oracle Deutschland B.V. & Co. KG, kümmert sich um verschiedene Themen rund um Oracle Technology in und außerhalb der Cloud und verfügt über jahrelange Erfahrung speziell in den Bereichen Cloud, Betrieb, Middleware und Datenbank. Er studierte Technische Informatik und arbeitet seit über 20 Jahren bei Oracle.



Marcus Schröder
marcus.schroeder@oracle.com





Oracle-Tool-Unterstützung für APEX Application Lifecycle Management

Arne Brüning, Oracle Deutschland

Ob es nun um eine komplette CI/CD-Pipeline geht oder lediglich zwei Entwickler ihre Arbeit über ein Git-Repository abgleichen möchten: Die Anforderungen an das APEX-Tooling sind grundsätzlich dieselben. Dieser Artikel beschreibt den aktuellen Stand der Werkzeugunterstützung von Oracle in Application Lifecycle Management Workflows für APEX-Entwickler.

Oracle Application Express war schon immer ein Werkzeug, mit dem sich mit verhältnismäßig geringem Aufwand auch umfangreiche Anwendungen erstellen lassen. Da APEX-Anwendungen aber zu großen Teilen aus Metadaten bestehen, haben sie keinen Quellcode im eigentlichen Sinne. Dies ist ein großer Vorteil gegenüber Systemen, die stattdessen Code generieren, der dann über Jahre gepflegt werden muss. Um jedoch Änderungen in APEX-Anwendungen über typische, Quellcode-basierte Versionskontrollsysteme versionierbar zu machen, muss aus den Metadaten zunächst Quellcode erzeugt werden, der diese Metadaten erstellen kann. Die Unterstützung hierfür ist allerdings erst über die Jahre gewachsen. Dies war eine Herausforderung, weil moderne Entwicklungs-Prozesse bis hin zu komplexen CI/CD-Pipelines typischerweise Quellcode-basiert arbeiten und beim Commit in die Versionsverwaltung beginnen. Oracle hat im Laufe der Zeit und insbesondere über die letzten Jahre bei der Werkzeugunterstützung deutlich nachgelegt.

Historie

Am Anfang war der Export in eine einzelne Datei. Praktisch die komplette APEX-Anwendung wurde in eine lange SQL-Datei geschrieben und konnte so am Stück in einer anderen Umgebung wieder importiert werden. Was aber eine einfache Installation einer APEX-Anwendung in einer anderen Umgebung ermöglichte, war für ein Szenario, in dem mehrere Entwickler an einer Anwendung arbeiten, die denkbar ungünstigste Lösung. Es ließen sich auch einzelne Seiten oder Komponenten ex- und wieder importieren, was die Zusammenarbeit zwischen Entwicklern grundsätzlich ermöglichte. Doch es war immer ein gewisser Abstimmungsaufwand nötig und die manuellen Prozesse bargen die Gefahr von Fehlern, beispielsweise dass nicht alle geänderten Komponenten auch wirklich manuell exportiert wurden. Auch konnte eine große Anwendung in mehrere kleinere Anwendungen zerlegt werden, die unabhängig voneinander gepflegt werden. Dies erleichtert die gemeinsame Arbeit an großen Anwendungen, ohne dass der

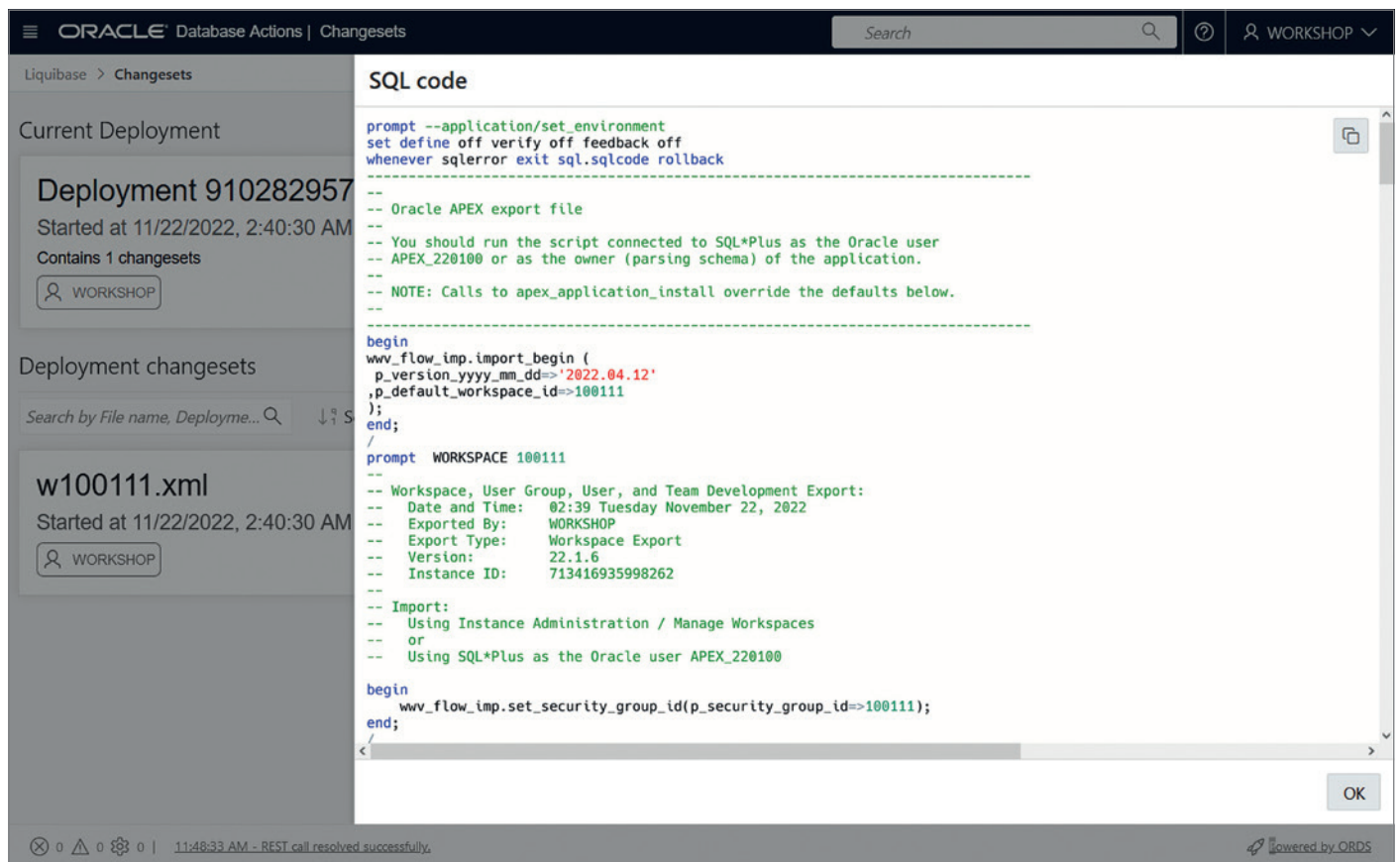


Abbildung 1: Liquibase-Support in den ORDS Database Actions (Quelle: Arne Brüning)

Endanwender diese als getrennte Anwendungen wahrnimmt.

Die Situation besserte sich für Entwickler merklich, als Oracle die „Split into Multiple Files“-Option für den Export einführte. Beim Export aus der UI konnte man nun wahlweise ein ZIP-Archiv schreiben, das eine separate Datei pro Komponente beinhaltet. Wichtig ist hierbei die Option „Export with Original IDs“, damit die Komponenten keine neue IDs zugewiesen bekommen und sich damit de facto doch wieder alle Dateien ändern.

Dazu gesellte sich das Tool APEXExport, das sich unter dem Utilities-Verzeichnis im APEX-Installationsverzeichnis befindet. Hiermit können komplette APEX-Anwendungen auf der Kommandozeile exportiert werden. Anfangs noch über die separate Klasse APEXExportSplitter.class, später über die Option `-split` werden einzelne Dateien pro Komponente erzeugt. Die Option `-expOriginalIds` sorgt dafür, dass die IDs nicht neu vergeben werden, und `-skipExportDate` bewirkt, dass bei nicht geänderten Komponenten beim Export nicht ein neues Export-Datum in die Kommentare geschrieben wird. Zusätz-

lich lassen sich über `-expComponents TYPE:ID` auch einzelne Komponenten exportieren. Damit lässt sich dann auch grundsätzlich mit mehreren Entwicklern über ein Versionkontrollsystem zusammenarbeiten, was so auch schon Anwendung findet.

SQLcl

Wer sich mit dem aktuellen Stand der Werkzeugunterstützung durch Oracle beschäftigt, kommt an SQLcl nicht vorbei. Dabei handelt es sich um ein Kommandozeilen-Programm, um wie mit SQL*Plus interaktiv mit der Datenbank zu arbeiten. Im Gegensatz zu seinem älteren Vorgänger hat SQLcl jedoch zum Beispiel viele Komfort-Funktionen wie etwa Syntax-Highlighting oder Tab-Completion. Auch die Historie und das Editieren von Eingaben wurden gegenüber SQL*Plus stark vereinfacht. Da SQLcl kurze Update-Zyklen hat, wird es gar nicht erst mit der Datenbank ausgeliefert, sondern steht als separater Download zur Verfügung. Für Prozesse, in denen automatisch die neu-

esten Werkzeug-Versionen für Build oder Test heruntergeladen werden, gibt es einen festen Link auf die jeweils aktuelle Version, für den kein Login erforderlich ist. SQLcl ist ohne weitere Software von Oracle lauffähig, es muss kein (Thin-)Client installiert sein.

Da SQLcl in Java entwickelt wurde, ist eine unterstützte Java-Runtime erforderlich, aktuell mindestens Version 11. Durch den Wechsel auf Java besteht für SQLcl die Möglichkeit, vorhandene in Java implementierte Funktionalität einzubinden. So gibt es neben den im Folgenden besprochenen Möglichkeiten beispielsweise auch Unterstützung für Data Pump oder PGQL. Wer an der Weiterentwicklung von SQLcl interessiert ist, dem sei hierzu der Blog von Oracle Product Manager Jeff Smith empfohlen.

SQLcl und APEX

Eine dieser Erweiterungen von SQLcl ist das Einbinden der Funktionalität von APEXExport als APEX-Kommando. Über das Kommando `apex help export` lassen

sich alle verfügbaren Optionen anzeigen, inclusive der oben bereits empfohlenen `-split`, `-exporiginalids` und `-skipexportdate`. Einzelne Komponenten lassen sich wie bei APEXExport exportieren via `-expcomponents` im Format `TYPE:ID`.

Ebenfalls von APEXExport stammt seit ein paar Versionen die Möglichkeit, anstelle von Code per `-expType` `READABLE_YAML` oder `READABLE_JSON` auch in ein besser lesbares Format zu exportieren. Diese Formate können aktuell nicht wieder importiert werden. Stattdessen ist die Idee dahinter, dass im Falle eines manuellen Code-Merge in diesen Formaten leichter erkannt werden kann, was sich zwischen zwei Versionen geändert hat.

Die Integration von APEXExport in SQLcl vereinfacht die Nutzung auch, weil APEXExport.class standardmäßig nur im APEX-Home-Verzeichnis installiert ist und einen entsprechenden CLASSPATH benötigt. Mit SQLcl bekommt man von überall Zugriff auf diese Funktionalität und es lässt sich interaktiv damit arbeiten. Genau wie bei der Arbeit mit APEXExport lassen sich also die geänderten Komponenten jeweils als einzelne SQL-Dateien exportieren (siehe Listing 1).

Diese können dann über die Versionsverwaltung auf ein führendes System übertragen und dort angewendet werden. Wenn immer die kompletten Quellen einer Anwendung exportiert werden, kann man es dem Versionskontrollsystem überlassen, die geänderten Dateien zu finden.

SQLcl und Liquibase

Neben den APEX-Anwendungen müssen dazu auch die jeweiligen Datenbanken versioniert werden. Änderungen in statusbehafteten Systemen wie Datenbanken sind grundsätzlich aufwendiger. Anstelle des kompletten Codes dürfen nur die Änderungen zum bestehenden Zustand angewendet werden. Und diese auch nur genau ein Mal, ein korrektes `ALTER TABLE` führt beim wiederholten Aufruf zu einer Fehlermeldung. Außerdem muss der Quellcode immer doppelt gepflegt werden. Als vollständiger Quellcode, der die aktuelle Version beschreibt, sowie die jeweiligen Delta-Dateien. Ohne Automation ist dies ein potenziell aufwendiger und fehleranfälliger Prozess.

```
sql username/password@host:port/service
SQL> apex export -applicationid 12345 -split -skipExportDate -expOriginalIds -expSupportingObjects Y -expType APPLICATION_SOURCE,
READABLE_YAML
```

Listing 1: Export einer APEX-Anwendung mit SQLcl

```
conn workshop/workshop@truncpdb
lb generate-schema -sql -grants

conn workshop/workshop@devpdb1
lb update -changelog-file controller.xml

select table_name from user_tables;
TABLE_NAME
-----
DATABASECHANGELOGLOCK
DATABASECHANGELOG_ACTIONS
DATABASECHANGELOG
DEPARTMENTS
EMPLOYEES
```

Listing 2: Generieren und Anwenden der Schema-Dateien per SQLcl Liquibase

Liquibase ist eines der Werkzeuge, die dies automatisieren. Änderungen zwischen Datenbank-Schemata können damit erfasst und übertragen werden, auch ist beispielsweise ein Zurückrollen von Änderungen möglich.

Liquibase ist wie SQLcl in Java entwickelt, somit konnte es von Oracle einfach in SQLcl integriert werden. Über das Kommando `liquibase` oder `lb` steht die Funktionalität von Liquibase somit direkt in SQLcl zur Verfügung. Der Funktionsumfang ist dabei allerdings nicht immer identisch. Manche Änderungen von Liquibase stehen erst mit Verzögerung in SQLcl zur Verfügung. Auf der anderen Seite ist Liquibase ein Tool, das verschiedene Datenbanken unterstützt, damit aber nicht alle Eigenschaften von Oracle nutzt. Oracle hat daher die Funktionalität von Liquibase in SQLcl erweitert und unterstützt beispielsweise das automatische Generieren von SQL-Kommandos zum Zurückrollen von Änderungen.

Mit dem in Listing 2 gezeigten Kommando `lb generate-schema` wird für jedes Schema-Objekt eine Datei im Liquibase-XML-Format erzeugt, in diesem Beispiel eines, das nur die Tabellen `Employees` und `Departments` aus den Oracle-Beispiel-Schemata enthält. Durch die Option `-sql` werden zusätzlich die entsprechenden SQL-Dateien angelegt. In einem an-

deren Schema können diese Objekte per `lb update` übernommen werden. Beim ersten Anwenden im Ziel-Schema werden dazu auch die Verwaltungstabellen von Liquibase angelegt. `DATABASECHANGELOG` und `DATABASECHANGELOGLOCK` gehören zum Liquibase Standard. Die erste protokolliert, welche Changelogs bereits auf das Schema angewandt wurden, die andere dient dem Liquibase-eigenen Logging-Mechanismus.

Die `DATABASECHANGELOG_ACTIONS` ist eine Erweiterung seitens Oracle, die das durchgeführte DDL protokolliert. Hieraus kann die Liquibase-Variante von Oracle dann automatisch die für einen Rollback benötigten Kommandos generieren.

Die neueren Versionen der Oracle REST Data Services (ORDS) unterstützen Liquibase und die Oracle-Erweiterungen. In den Database Actions des ORDS lassen sich die durchgeführten Deployments inclusive der in der `DATABASECHANGELOG_ACTIONS` protokollierten SQL-Kommandos anzeigen.

SQLcl, Liquibase und APEX

Die oben vorgestellten Tools reichen für den kompletten APEX Application Lifecycle Workflow wie in [1] beschrieben schon aus, das Folgende ist also optio-

```

conn workshop/workshop@truncpdb
-- Workspace-ID und Application ID finden
apex list

-- Generieren der Liquibase Changesets
cd /home/oracle/Workshop
lb generate-apex-object -expworkspace -workspaceid 100111 -dir application
lb generate-apex-object -applicationid 100 -dir application

-- Anwenden der Liquibase Changesets
conn workshop/workshop@devpdb1
lb update -changelog-file application/w100111.xml
lb update -changelog-file application/f100.xml

```

Listing 3: Ex- und Importieren einer APEX-Anwendung per SQLcl Liquibase

nal. In der aktuellen Version SQLcl 22.3.x bietet Oracle zusätzlich Liquibase-Unterstützung für APEX. Grundsätzlich ist dies nicht notwendig, da APEX-Anwendungen im Gegensatz zur Datenbank zustandslos sind und einfach neu deployed werden können. Der Code zum Erstellen der Metadaten, die eine APEX-Anwendung ausmachen, wird einfach erneut ausgeführt. Andererseits wird Entwicklern, die datenbankseitig mit Liquibase arbeiten, ein äquivalenter Workflow auch für APEX geboten.

Das Beispiel in *Listing 3* zeigt, wie ein APEX Workspace und eine Anwendung per SQLcl Liquibase auf eine andere Datenbank übertragen werden können.

Fazit

Aus der Funktionalität der oben vorgestellten Tools ergibt sich noch nicht automatisch ein kompletter Application-Lifecycle-Management-Prozess. Dieser würde auch den Umfang dieses Artikels sprengen. Oracle veröffentlicht aber in unregelmäßigen Abständen hierzu Whitepaper, die eine umfassende Sammlung von Best-Practices enthalten. Während die letzte Version aus dem Jahr 2016 viele Möglichkeiten, wie die Einbindung in SQLcl, noch nicht kennt, geht die aktuelle Version von 2022 [1] explizit darauf ein.

Modernes Application Lifecycle Management und die Entwicklung mit APEX sind kein Widerspruch. Oracle hat gerade in den letzten Jahren die Werkzeugunterstützung hierfür stark verbessert, mit dem aktuellen Werkzeugkasten ist APEX deutlich besser aufgestellt für Applica-

tion-Lifecycle-Management-Prozesse, als dies noch vor einigen Jahren der Fall war.

Über den Autor

Arne Brüning programmiert seit den 80ern und begann in den 90ern, Datenbankanwendungen zu entwickeln. Heute arbeitet er als Solution Engineer mit Schwerpunkt auf Datenbank-Entwicklung bei Oracle Deutschland.

Quellen

- [1] Oracle (2022): Understanding the Oracle APEX Application Development Lifecycle. apex.oracle.com/go/lifecycle-technical-paper



Arne Brüning
arne.bruening@oracle.com

DOAG

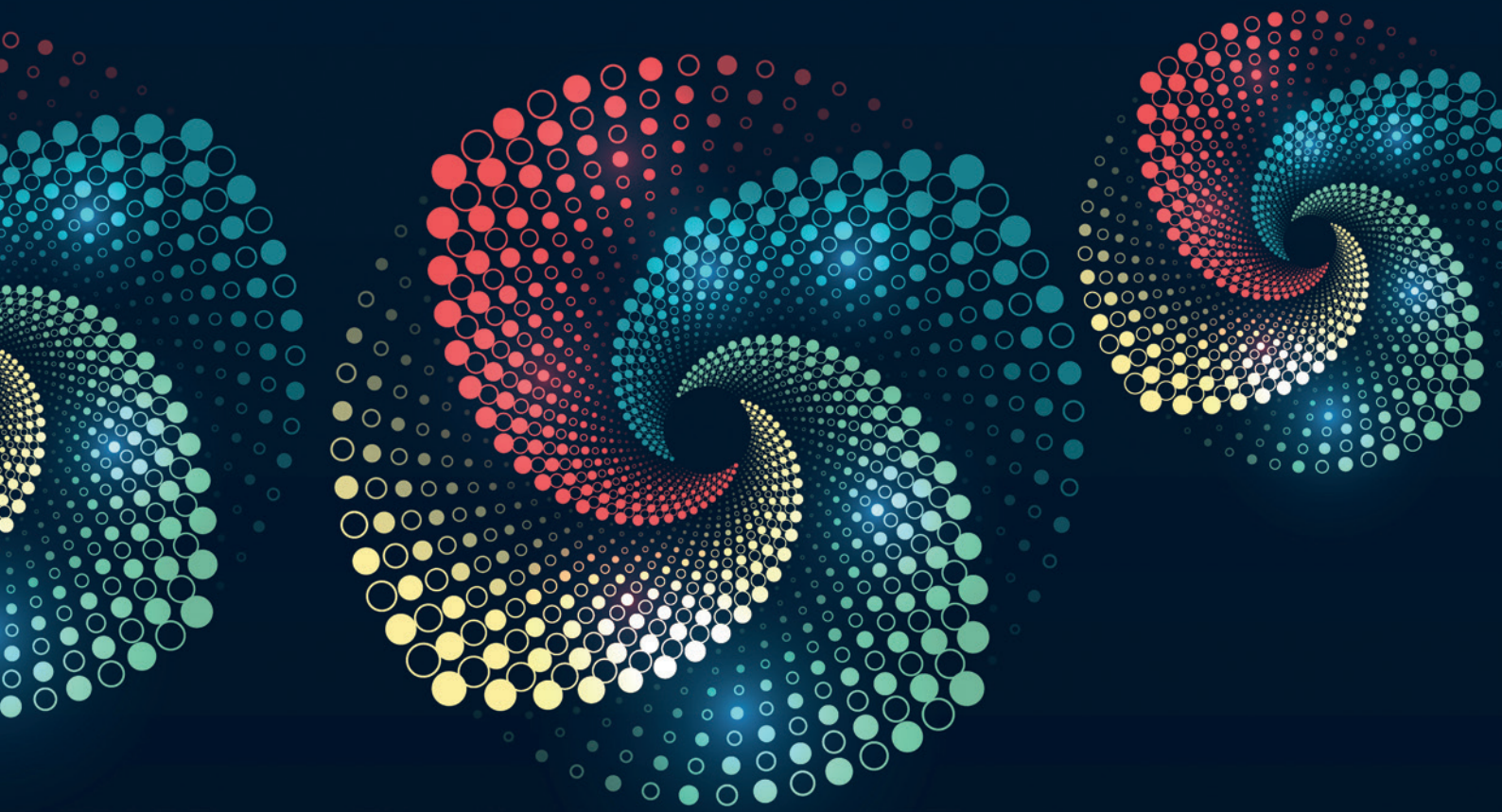
DOAG
Datenbank
mit Exaday

2023

24. und 25. Mai in Düsseldorf



datenbank.doag.org





CI/CD für Oracle-APEX-Anwendungen mit GitLab

Johannes Michler und Simon Grossmann, Promatis Gruppe, Ettlingen (TechnologieRegion Karlsruhe)

Oracle Application Express (APEX) ist eine Low-Code-Plattform zur Erstellung von datenbasierten Webanwendungen, die direkt in einer Oracle-Datenbank installiert wird. Die Entwicklung einer solchen Webanwendung findet in der APEX-eigenen Entwicklungsumgebung statt. Dieser Artikel zeigt eine Möglichkeit auf, wie die Entstehung einer APEX-Applikation anhand CI/CD automatisiert auf verschiedene Umgebungen (TEST und PROD) ausgerollt werden kann.

Einleitung

Die Low-Code-Plattform Oracle APEX wird von Oracle seit vielen Jahren als Teil aller Datenbankeditionen entwickelt. Sie beinhaltet eine grafische Benutzeroberfläche zur Entwicklung von datenbasierten Webanwendungen und besteht neben der Benutzeroberfläche in der Regel aus zusätzlichen Oracle-Datenbank-Objekten (wie Tabellen, Packages usw.).

Ein häufiges Problem bei der Nutzung von Oracle APEX in größeren Projekten ist der aufwendige, manuelle Prozess, um eine Applikation und alle dazugehörigen Komponenten von einer Umgebung, beispielsweise der Entwicklungsumgebung, auf eine andere Umgebung, wie die Testumgebung, zu übertragen. Hierzu müssen alle Datenbankobjekte, statische Daten sowie die Applikation selbst manuell von der Entwicklungsumgebung auf die Testumgebung und später auf die Produktivumgebung übertragen werden.

Oracle APEX besitzt hierfür seit der Version 20.1 ein eingebautes Feature namens Remote Deployment, das den Deployment-Prozess von einer Umgebung auf eine andere vereinfachen soll. Dennoch besteht der Remote-Deployment-Prozess aus vielen einzelnen Schritten, auf die bereits während der Entwicklung geachtet werden muss. So müssen beispielsweise Skripte zur Aktualisierung des Datenbankschemas manuell geschrieben und in der APEX-Applikation

als Update-Skripte hinterlegt werden. Der Prozess des eigentlichen Deployments ist hier ebenfalls ein händischer Prozess. In der APEX-Entwicklungsfläche muss eine APEX-Applikation mit manuellen Klicks durch mehrere Masken auf der Zielumgebung installiert werden (vgl. [1]). Hierbei können sich durch die manuelle Durchführung und Komplexität der Weboberfläche immer wieder Fehler einschleichen.

Diese und weitere Probleme werden in der modernen Softwareentwicklung typischerweise anhand einer Automatisierung durch CI/CD gelöst. CI/CD beschreibt die Automatisierung des Tests, der Auslieferung und der Überwachung von Software. Der Prozessablauf hilft dabei, die Änderungen der Entwickler in kürzeren Intervallen auf den Haupt-Entwicklungsbranch zu bringen. Darüber hinaus sorgt CI/CD mit automatischen Tests und Deployments für eine bessere Qualität sowie häufigere Releases einer Software (siehe Abbildung 1).

Im weiteren Verlauf dieses Artikels zeigen wir auf, wie wir es geschafft haben, eine CI/CD-Pipeline für APEX-Applikationen aufzubauen, um diese erfolgreich bei Kunden einzusetzen.

Tools

Zur Umsetzung des automatisierten Deployments sind einige Tools erforderlich,

die zunächst kurz vorgestellt werden, bevor der Ansatz erläutert wird.

Git

Git ist ein verteiltes Versionskontrollsystem, das von der Open-Source-Community vorangetrieben wird. Heutzutage ist Git das am meisten eingesetzte Tool für Versionskontrolle in der Softwareentwicklung und verfügt dabei über ein Command-Line-Interface – ist aber auch mit einer grafischen Benutzeroberfläche in vielen gängigen Entwicklungsumgebungen integriert. Für jede Entwicklungsaufgabe oder jedes Feature wird in Git ein neuer Entwicklungsbranch (Branch) erstellt und nach Abschluss der Entwicklung einer Aufgabe (z.B. Bug oder Feature) wird dieser Branch zurück in den Master-Branch gemerged (siehe Abbildung 2).

Eine Best-Practice-Lösung im Umgang mit Git und APEX ist, einen Branch pro Umgebung zu haben. Häufig wird dabei mit drei Umgebungen gearbeitet: einer Entwicklungsumgebung (Dev), einer Testumgebung und natürlich einem Produktiv-System. Die Entwicklungen werden dabei zunächst im Dev-Branch abgelegt. Sobald die Testumgebung aktualisiert werden soll, wird der Dev-Branch in den Test-Branch gemergt. Gleiches gilt für den Master-Branch, der den aktuellen Stand der Produktivumgebung widerspiegelt.

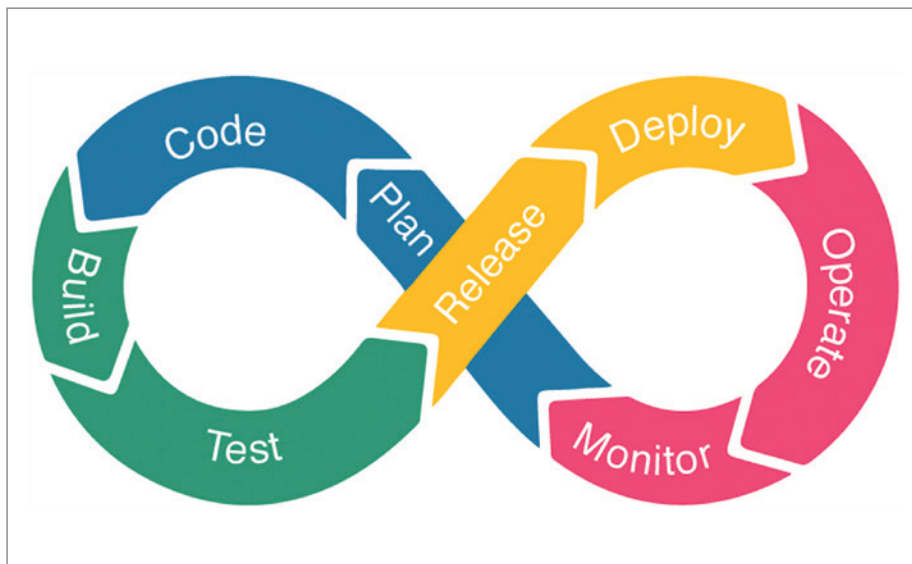


Abbildung 1: Veranschaulichung des Dev-Ops Lifecycle (Quelle: Yıldırım, Abdurrahim, 2019: DevOps Lifecycle: Continuous Integration and Development [2])

Zur Verwaltung von Projekten und Git-Repositories gibt es DevOps-Plattformen wie GitLab. Unser hier vorgestellter Ansatz würde auch auf anderen Plattformen wie GitHub oder Oracle Visual Builder Studio funktionieren. In unserem Fall konzentrieren wir uns allerdings auf GitLab. In GitLab lassen sich verschiedene Projekte anlegen, die als Git-Repository fungieren. Zusätzlich gibt es in GitLab die Möglichkeit, CI/CD Pipelines als Teil eines Projekts zu erstellen. Die Pipeline ist ebenfalls Teil des Quellcodes und kann somit von den Vorteilen der Versionskontrolle profitieren. Eine Pipeline kann dann beispielsweise bei jedem Push, also dem Laden der Änderungen, auf einen Branch ausgeführt werden.

Liquibase

Liquibase ist eine Software, die sich um Versionsmanagement von Datenbankschemata kümmert. Änderungen an Datenbankschemata werden in sogenannten Change-Sets auf XML-Basis dargestellt und können durch Liquibase auf Datenbanken durchgeführt werden. Liquibase erleichtert durch diese Technologie die Migration von Änderungen an Datenbankschemata zwischen verschiedenen Datenbanken. Das Übertragen von Änderungen von einer Datenbankumgebung auf eine andere Umgebung wird somit deutlich erleichtert und ist durch die Automatisierung weniger fehleranfällig.

SQLcl

Um den Code und die APEX-Applikation aus der Datenbank zu extrahieren, wird SQLcl verwendet – der Nachfolger von SQLPlus ist ein Command-Line-Interface für PL/SQL Code. Der Vorteil von SQLcl ist, dass Liquibase nativ mitgeliefert wird. Mithilfe von Liquibase können Datenbankobjekte jeglicher Art sowie APEX-Applikationen aus einer Datenbank extrahiert und in einer anderen Datenbank eingespielt werden. Dabei speichert Liquibase das Datenbankschema als XML-Dateien, die in unser Git-Repository eingekickt werden können. Eine XML-Datei besteht dabei aus Change-Sets, wobei ein Change-Set eine Änderung an einem Datenbankschema beschreibt.

Als Besonderheit der in SQLcl integrierten Liquibase-Version sind zusätzliche Features im Umgang mit den Spezifika einer Oracle-Datenbank hervorzuheben (siehe [3]). Die von Liquibase generierten XML-Dateien sind Oracle-spezifisch und können dadurch einfacher mit Paketen, Jobs oder anderen Oracle-Datenbankobjekten umgehen.

Liquibase speichert in der Zieldatenbank Change-Sets, die bereits angewendet wurden, und kann so automatisiert Änderungen eines Schemas von einer Datenbank auf eine andere übertragen. Mit SQLcl kann darüber hinaus auch die APEX-Applikation extrahiert werden. Die daraus resultierenden SQL-Dateien können ebenfalls im Git-Repository eingek-

checkt werden. SQLcl kann mittels dieser SQL-Dateien die APEX-Applikation wieder auf einer Datenbank installieren.

SQLcl wird ständig weiterentwickelt und kürzlich wurde die Version 22.3 veröffentlicht. Unsere Code-Beispiele beziehen sich allerdings auf die Version 22.2, da Version 22.3 eine überarbeitete Syntax besitzt, die aber aufgrund einer veränderten Herangehensweise bei der Verwaltung der generierten Dateien für uns nicht infrage gekommen ist. Dadurch sind wir – trotz diverser Show Stopping Bugs – bei der Version 22.2 geblieben.

Methode

Wie oben im Git-Überblick bereits gezeigt, arbeitet unsere Methode mit drei Branches:

- Dev: Beinhaltet den aktuellen Stand der Entwicklungsumgebung. Hier wird nach einem Export der aktuelle Stand eingekickt.
- Test: Beinhaltet den Stand der Testumgebung. Wann immer neue Entwicklungen auf die Testumgebung installiert werden, muss der Dev-Branch in den Test-Branch gemergt werden.
- Master: Beinhaltet den Stand der Produktion. Der Test-Branch wird bei einem neuen Release in den Prod-Branch gemergt.

Abbildung 3 stellt den Prozess dar, der durchgeführt werden muss, um eine APEX-Applikation von der Entwicklungsumgebung auf die Testumgebung zu übertragen. Hierbei müssen im ersten Schritt das Datenbankschema und die APEX-Applikation exportiert werden. Um den Export zu erleichtern, haben wir ein Skript entwickelt, das alle notwendigen Datenbankobjekte, statische Daten sowie die APEX-Applikation selbst, automatisch exportiert.

Bevor wir mit dem Export beginnen, muss zunächst die Verbindung zur Datenbank aufgebaut werden. Dafür wird erst das Datenbankschema exportiert, das – im Voraus mithilfe von durch Namenskonvention festgelegten Präfixen – gefiltert wird, sodass nur die relevanten Objekte exportiert werden (und dadurch z. B. verschiedene Teil-Applikationen getrennt abgelegt werden können). Dies

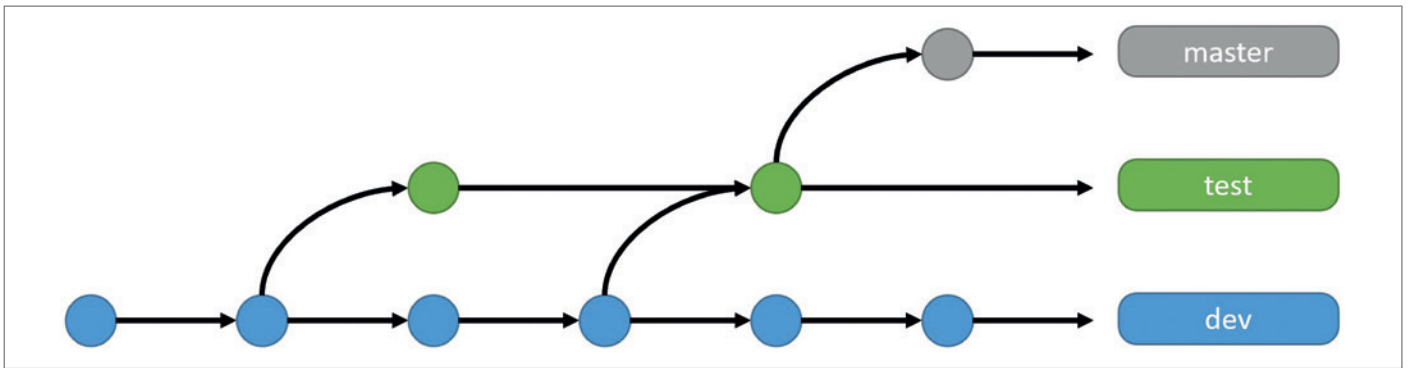


Abbildung 2: Überblick der Git-Branches für unsere APEX-Applikationen (Quelle: PROMATIS)

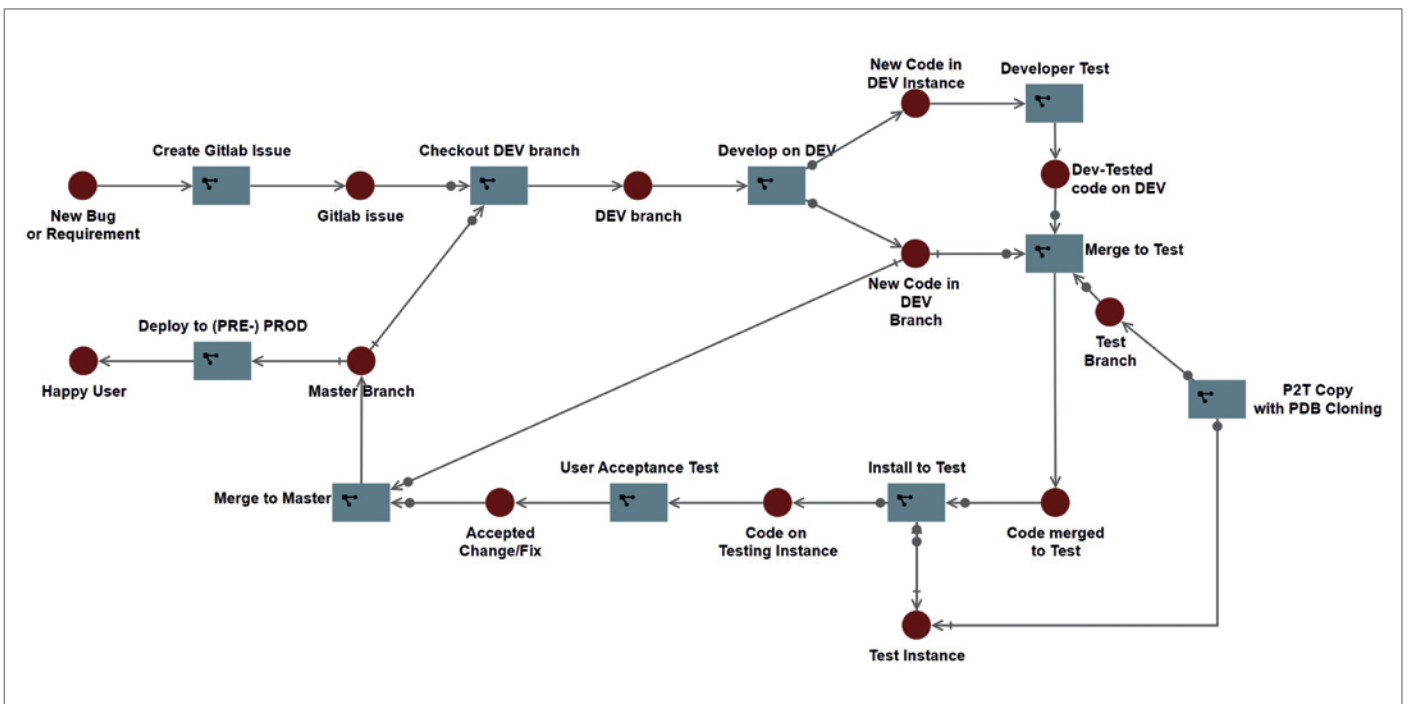


Abbildung 3: Überblick der Methode für automatisiertes Deployment für APEX-Applikationen (Quelle: PROMATIS)

wird mit dem Ausführen eines einzigen Befehls erreicht. Anschließend werden die statischen APEX-Daten mittels Spooling in CSV-Dateien exportiert. Dabei wird für jede Tabelle eine eigene CSV-Datei angelegt. Zuletzt wird nun die APEX-Applikation anhand des Standard-APEX-Export-Befehls exportiert (siehe Listing 1). Wir setzen jedoch bewusst nicht auf die SQLcl-Liquibase-APEX-Integration, da diese für unseren Anwendungsfall zu viele Probleme verursacht hat. Um Komplikationen mit dem Überschreiben von Datenbank-Sequenzen in den Zieldatenbanken vorzubeugen, werden alle Änderungen an Sequenzen nach dem Export rückgängig gemacht.

Anschließend müssen die Änderungen in Git eingeecheckt werden. Diesbezüglich ist zu beachten, dass vor dem Export der

richtige Branch ausgecheckt wurde – in unserem Beispiel ist dies der DEV-Branch. In der vorliegenden Konfiguration ist es für jeden Nutzer möglich, direkt auf den DEV-Branch zu pushen; TEST- und PROD-Branch können nur via Merge Request aktualisiert werden. Dies ist so konfiguriert, um zusätzliche Sicherheit bei der Aktualisierung der Test- und Prod-Umgebung zu erreichen (siehe Listing 2).

Mit den auf den Dev-Branch in Git eingeecheckten und gepushten Änderungen kann in GitLab ein Merge Request vom Dev-Branch auf den Test-Branch erstellt und gemergt werden. Wir haben in unserem GitLab-Projekt die CI/CD-Pipeline so konfiguriert, dass bei einer Änderung auf dem Test-Branch automatisch diese auf die Test-Instanz ausrollt (siehe Abbildung 4). Das Ausrollen der Änderungen

auf die Prod-Instanz für den Prod-Branch wird hingegen (aus Sicherheitsgründen noch) nicht vollautomatisch ausgeführt, sondern muss „per Knopfdruck“ über die GitLab-Weboberfläche gestartet werden.

Nachdem der Merge durchgeführt worden ist, führt unsere CI/CD-Pipeline die Installation durch. Nun werden automatisch die folgenden Schritte aus Listing 3 in einer Kommandozeile durchgeführt.

Nachdem die CI/CD-Pipeline erfolgreich durchlaufen ist, sind alle Datenbankobjekte, Setup-Daten und die APEX-Applikation auf der Testumgebung installiert (siehe Abbildung 5). Die Testumgebung ist damit auf dem gleichen Stand wie die Entwicklungsumgebung. Einen kleinen Unterschied gibt es dennoch: Liquibase erstellt auf der Zielumgebung bei der ersten Ausführung drei weitere Tabellen.

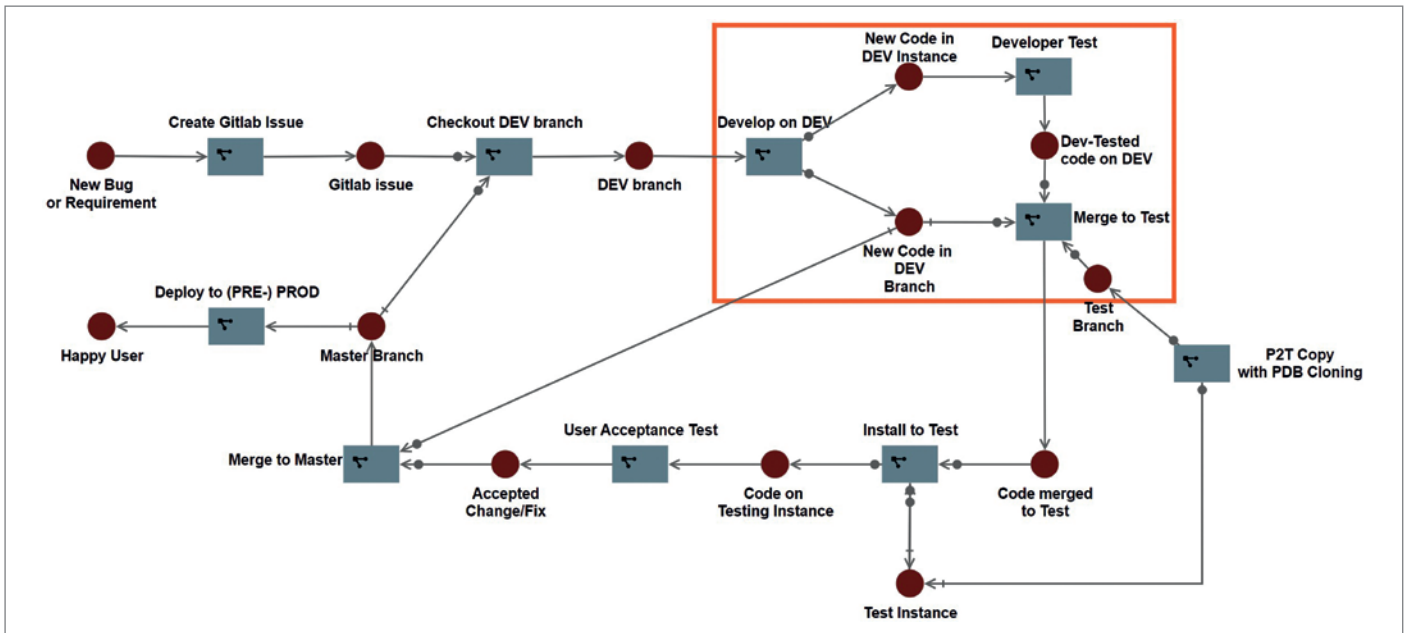


Abbildung 4: Überblick der Methode für automatisiertes Deployment für APEX-Applikationen mit der Markierung des aktuellen Prozessschritts. (Quelle: PROMATIS)

```

$ cd <pfad_zu_git>
$ sql user/password@//host:port/service
sql> cd database/mul;
sql> lb genschema -split -filter "LIKE 'MUL_%'";
sql> cd ../..;
sql> set sqlformat csv;
sql> set feedback off;
sql> alter session set nls_date_format = 'YYYY-MM-DD';
sql> spool 'data/mul/<tabelle>.csv';
sql> select * from <tabelle>;
sql> spool off;
sql> alter session set nls_date_format = 'DD.MM.RR';
sql> set sqlformat ansiconsole;
sql> set feedback on;
sql> cd apex;
sql> apex export -workspaceId <workspace_id> -applicationId <app_id>
-skipexportdate -exportoriginalids -split;
sql> cd ..;
sql> exit;
$ git checkout HEAD -- ./database/mul/sequence/
    
```

Listing 1: Befehle zum Export aus der Entwicklungsumgebung

```

$ cd <pfad_zu_git>
$ git add .
$ git commit -m "<commit_nachricht>"
$ git push
    
```

Listing 2: Befehle zum Einchecken der Änderungen in das Git-Repository

```

$ cd <pfad_zu_git>
$ sql user/password@//host:port/service
sql> cd database
sql> lb update -changelog controller.xml
sql> cd ../data/<tabelle>
sql> lb update -changelog data.xml
sql> cd ../../apex/<applikation>
sql> @install.sql
    
```

Listing 3: Befehle zur Installation auf der Testumgebung

Anhand dieser Tabellen merkt sich Liquibase, welche Änderungen (Change-Sets) bereits angewandt worden sind, sodass diese nicht erneut angewendet werden.

Fazit

Deployments von APEX-Applikationen lassen sich in nur wenigen Schritten fast vollständig automatisieren und sorgen so dafür, dass die Deployments einfacher und in kürzeren Zyklen stattfinden können. Durch die Automatisierung des gesamten Deployments lassen sich Fehler minimieren und somit positive Auswirkungen auf die Qualität der Software ermöglichen. Oracle hat für die Zukunft weitere Verbesserungen hinsichtlich Built-in CI/CD angekündigt (siehe [4]).

Darüber hinaus kann die CI/CD-Pipeline um automatisiertes Testen erweitert werden. Denkbar wäre hier beispielsweise das Ende-zu-Ende-Testen der APEX-Applikation mittels Selenium. So könnte die Qualität der Software weiter verbessert werden. Des Weiteren wird durch das automatisierte Testen sichergestellt, dass durchgeführte Änderungen keine Fehler in der Software verursachen.

Quellen

[1] Chaitanya Koratamaddi, 2020: Remote Deployment of your APEX App is just

passed Job #31269 triggered 3 days ago by Grossmann, Simon

This job is deployed to Lieferantenportal-QAT.

```

1 Running with gitlab-runner 15.4.0 (43b2dc3d)
2   on apex-portal-qatdb-intern-dns UNB2LBHK
3   ✓ Preparing the "shell" executor 00:00
4   Using Shell executor...
5   ✓ Preparing environment 00:00
6   Running on apex-portal-qatdb-intern-dns...
7   ✓ Getting source from Git repository 00:01
8   Fetching changes with git depth set to 50...
9   Reinitialized existing Git repository in /home/gitlab-runner/builds/UNB2LBHK/0/software-development/externe-apex-portale/.git/
10  Checking out 77fe3b37 as qat...
11  Skipping Git submodules setup
12  ✓ Executing "step_script" stage of the job script 00:27
13  $ echo "CI_COMMIT_MESSAGE:${CI_COMMIT_MESSAGE}"
14  CI_COMMIT_MESSAGE:Merge branch 'develop' into 'qat' Fixes in Bilanz-Templates und User Management. See merge request software-development/externe-apex-portale!42
15  $ echo "CI_COMMIT_BRANCH:${CI_COMMIT_BRANCH}"
16  CI_COMMIT_BRANCH:qat
17  $ sh install_auto.sh ${CI_COMMIT_SHORT_SHA}
18  Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8
19  SQLcl: Release 22.2 Production on Mon Nov 07 11:05:40 2022
20  Copyright (c) 1982, 2022, Oracle. All rights reserved.
21  Connected to:
22  Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
23  Version 19.16.0.0.0
24  --Starting Liquibase at 11:05:41 (version 4.9.1 #0 built at 2022-05-03 17:23+2221)
25  -- Loaded 25 changeSets
26  No Errors Encountered

```

Abbildung 5: GitLab Job Log eines Deployments auf die Testumgebung. (Quelle: PROMATIS)

- One Click Away! <https://blogs.oracle.com/apex/post/remote-deployment-of-your-apex-app-is-just-one-click-away>
- [2] Abdurrahim Yildirim, 2019: Dev-Ops Lifecycle: Continuous Integration and Development <https://medium.com/t%C3%BCrk-telekom-bulut-teknolojileri/devops-lifecycle-continuous-integration-and-development-e7851a9c059d>
- [3] Jeff Smith, 2022: Oracle Change Management with SQLcl and Liquibase <https://speakerdeck.com/thatjeffsmith/oracle-change-management-with-sql-cl-and-liquibase>
- [4] Oracle: Oracle APEX Roadmap <https://apex.oracle.com/en/learn/resources/roadmap/>

Über die Autoren

Johannes Michler ist Senior Principal Consultant, Systemarchitekt und Projektleiter für die Promatis-Gruppe mit Fokus auf serviceorientierte Architekturen (SOA), Web-Portale (insbesondere mit ADF und APEX) sowie Prozessautomatisierung. Als Mitglied im Management Board bekleidet er die Funktion „Executive Vice President – Head of Platforms & Development“ und ist seit 2010 für die DOAG

als Referent und Autor mit wissenschaftlichen und anwendungsnahen Beiträgen aktiv. Er nimmt als Referent zahlreiche Veranstaltungen der Oracle Community (IOUG & OATUG) wahr und ist als „ACE Director“ Teil der Oracle-ACE-Community.

Simon Grossmann ist seit 2015 bei der Promatis-Gruppe und besitzt als technischer Consultant mehrjährige Erfahrung in der Java-Anwendungsentwicklung und der Realisierung prozessorientierter Informationssysteme mit Oracle-Komponenten sowie fundierte Kenntnisse im Einsatz moderner Web-Technologien (APEX, JEE) in Verbindung mit leistungsfähigen Software-Entwicklungsumgebungen, Application-Servern und Service-orientierten Architekturen (SOA).



Johannes Michler
johannes.michler@promatis.de



Simon Grossmann
simon.grossmann@promatis.de



Rethinking Unit Testing: Automatisieren der Generierung von Java-Unit-Tests

Markus Zimmermann und Evelyn Haslinger, Symflower

In einer idealen Welt müssten sich Softwareentwickler nicht mit fehlerhafter Software beschäftigen. Auf die Frage, welche Superkraft ein Softwareentwickler gerne hätte, würden wir mit an Sicherheit grenzender Wahrscheinlichkeit einige Male hören: „... dazu in der Lage sein, zu 100 % fehlerfreien Code zu schreiben“. Unit-Tests sind ein Werkzeug, um diesem Ideal etwas näher zu kommen – doch die Art und Weise, wie Unit-Tests derzeit erstellt werden, hat ihre eigenen Risiken und Nachteile: Sie sind sehr zeitaufwendig zu erstellen und dennoch rutschen Fehler durch. Aber was wäre, wenn wir die Erstellung von Unit-Tests automatisieren und den menschlichen Faktor und die damit verbundenen Fehler beseitigen könnten?



Softwarefehler sind frustrierend, produktivitätsmindernd, können zum Wegfall von Bestandskunden führen sowie die Akquise neuer Kunden erschweren. Bis heute ist es uns nicht möglich, auf Anhieb fehlerfreien Code zu schreiben. Das ist im Wesentlichen der Grund, warum wir Code testen. Das Ziel des Testens ist sicherzustellen, dass wir Fehler und falsches Softwareverhalten finden, idealerweise so früh wie möglich. Je näher wir dieses Testen zeitlich an den eigentlichen Prozess des Codeschreibens heranrücken können, desto näher kommen wir der ersehnten Superkraft der Entwickler, nur 100 % korrekten Code zu liefern.

Testen, um die Korrektheit von Source Code sicherzustellen

Automatisierte Testfälle helfen uns, dem Ideal von fehlerfreier Software näherzukommen und Implementierungen stabil sowie regressionsfrei zu halten. Das Problem ist, dass das Testen von Software schwierig ist. Nicht, weil wir nicht wissen, wie man Tests schreibt, oder uns die passende Ausbildung dazu fehlt. Sehr oft müssen wir auch mit Codebasen arbeiten, die einfach nicht für automatisiertes Testen konzipiert wurden. Dies erschwert es, Tests hinzuzufügen, und macht das

Refactoring der Code-Basis, um diese testbar zu machen, notwendig.

Genau aus diesen Gründen hat sich die Entwicklungsmethodik von Test Driven Development (TDD) [2] etabliert. Mit diesem Ansatz werden Tests geschrieben, bevor der eigentliche Produktionscode erstellt wird. Neue Funktionalität wird nur dann hinzugefügt, wenn es dazu bereits einen passenden automatisierten Testfall gibt. Dieser Testfall stellt sicher, dass das System die beabsichtigte Funktionalität jetzt und in Zukunft zur Verfügung stellt.

Das Problem ist jedoch, dass all diese Testfälle immer noch manuell geschrieben werden müssen. Das ist nicht nur ein mühsamer, zeitraubender Prozess, sondern auch fehleranfällig. Während TDD zu besser wartbarem und testbarem Source Code mit einer grundlegenden Testabdeckung führt, fehlen zumeist dennoch die Ressourcen, um alle Eventualitäten, die der Source Code bietet, zu überprüfen. Das heißt, Sicherheitslücken und nicht beabsichtigtes Verhalten können trotz TDD durchrutschen.

Warum sollten Softwarefehler so bald wie möglich aufgezeigt werden?

Fehler so bald wie möglich im Entwicklungsprozess aufzuzeigen, spart Zeit,

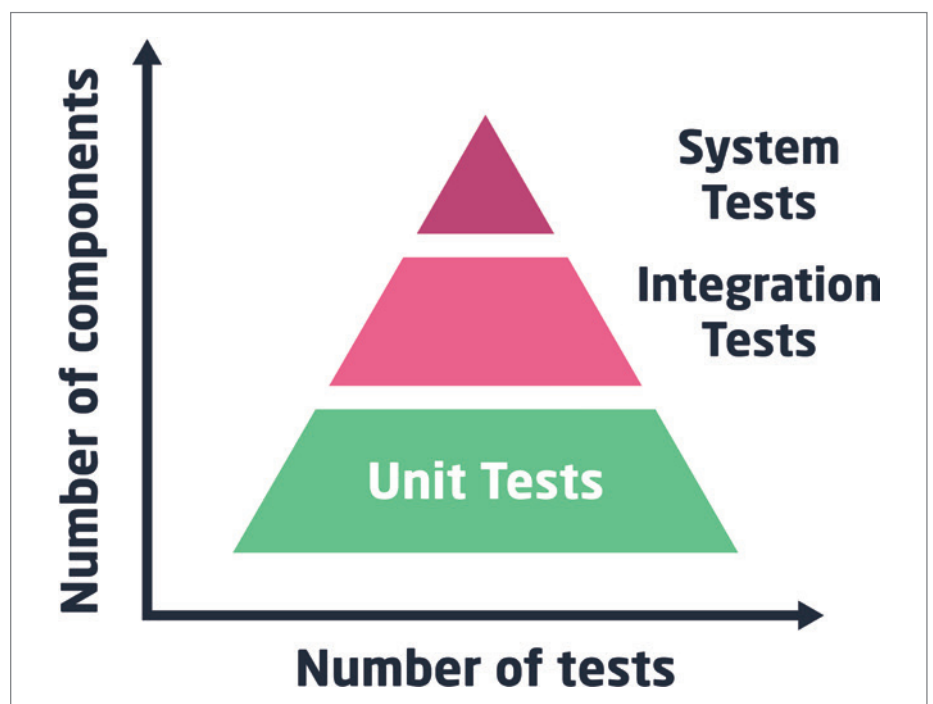


Abbildung 1: Testpyramide (Quelle: Symflower)

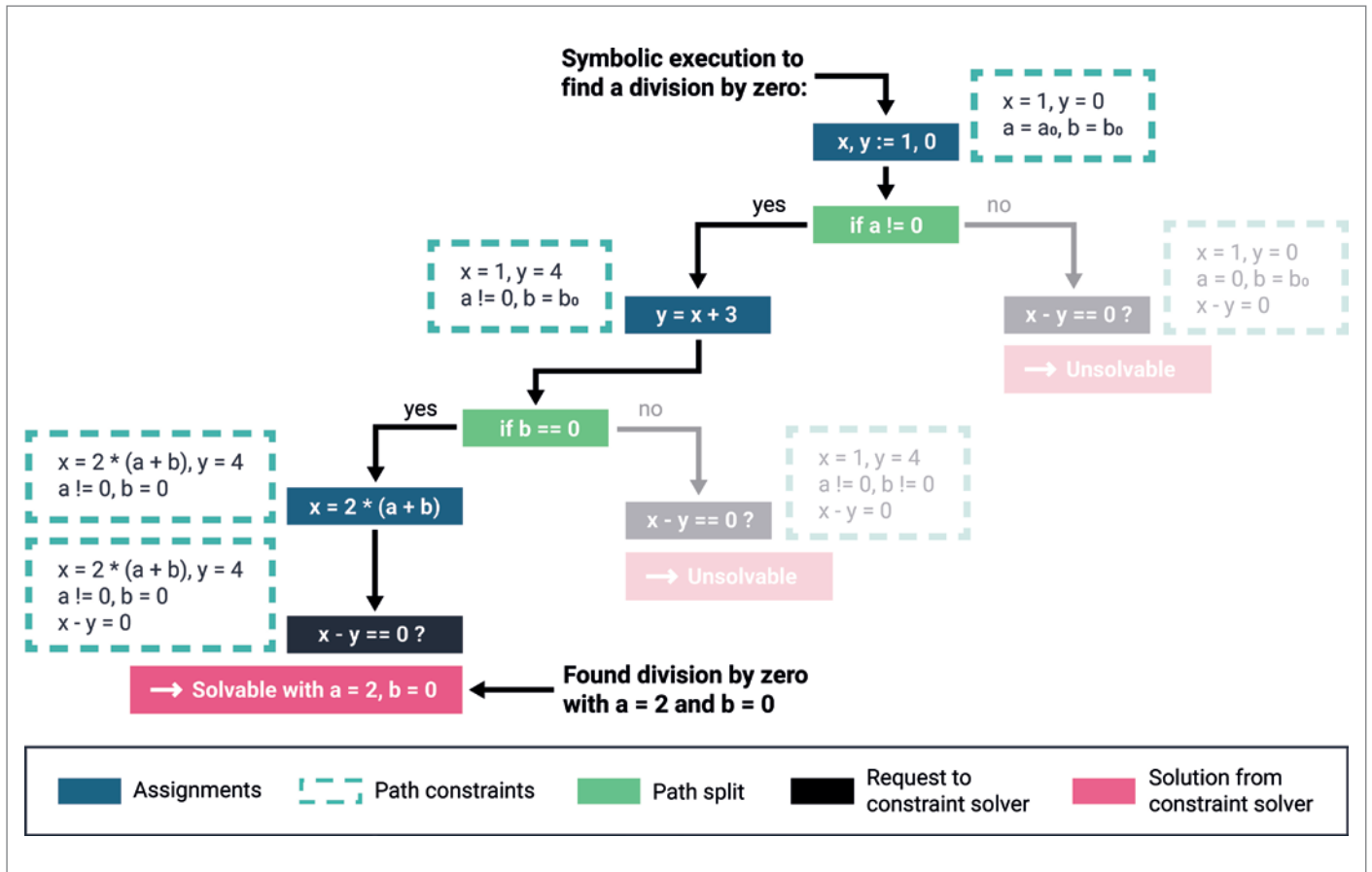


Abbildung 2: Schematische Ansicht einer symbolischen Ausführung der einzelnen Statements einer Funktion (Quelle: Symflower)

Energie und unsere geistige Gesundheit. Die viel zitierte Quelle [1], dass ein Fehler in der Produktion 50- bis 200-mal mehr kostet als in der Entwicklung, ist eine allgemeine Faustregel, die je nach Projekt und Fehler stark variiert. Sicher ist, dass ein Fehler, je länger er unentdeckt bleibt, mehr Probleme verursachen kann. Aus diesem Grund wird Shift-Left-Testing, also das Testen so bald wie möglich im Entwicklungsprozess, in immer mehr Projekten angewandt.

Für jedes Problem, das früh im Entwicklungslebenszyklus gefunden wird, gibt es folgende drei Vorteile:

1. Das Problem wird normalerweise nicht vom Endbenutzer gemeldet, was bedeutet, dass es einfacher ist, den Fehler zu reproduzieren. (Ganz zu schweigen von dem immateriellen Vorteil, dass wir Benutzer keinen Fehlern in unserer Software aussetzen.)
2. Es gibt keine Release-Loops, sodass uns Reviews, Abnahmephasen, Release-Management oder gar User-Migrationen erspart bleiben. Die Änderung wird einfach nicht freige-

geben, also sparen wir all diese Zeit und Mühe.

3. Es gibt viel weniger Nacharbeit. Wir konzentrieren uns zu jedem Zeitpunkt auf eine bestimmte Anforderung anstatt auf mehrere Anforderungen gleichzeitig, was noch mehr Zeit spart.

Testfälle von der Implementierung ableiten

Unit-Tests können eine grundlegende Testabdeckung für jede Komponente und Funktionalität darstellen und uns bereits auf die meisten Fehlerarten aufmerksam machen. Im Idealfall liegt ein Unit-Test für jeden Ausführungspfad einer Funktion vor. Eine Technik namens Symbolic Execution kann uns helfen, einen Paradigmenwechsel vorzunehmen: Testfälle werden nicht länger manuell erstellt, sondern direkt aus der Implementierung abgeleitet, ohne dass der Entwickler eingreifen muss.

Da Unit-Tests das Fundament der Testpyramide (siehe Abbildung 1) bilden, wirken sich hier realisierte Einsparungen

drastisch auf die Gesamteffizienz aus. Mit anderen Worten, die Möglichkeit, die Erstellung von Unit-Tests zu automatisieren, ist ein großer Schritt nach vorn. Allerdings müssen wir bei der automatischen Erstellung von Testkandidaten vorsichtig sein, da beispielsweise randomisiert erstellte Werte mehr Wartungsarbeiten als Nutzen verursachen können.

Damit Unit-Tests einen starken Mehrwert im Entwicklungsprozess bieten, müssen sie für Menschen lesbare Werte beinhalten und sollten im Idealfall minimale Testsuites darstellen, also keine redundanten Testfälle beinhalten. Die Millionen-Dollar-Herausforderung besteht natürlich darin, basierend auf dem vorhandenen Code automatisch wertvolle und aussagekräftige Testkandidaten zu berechnen. Eine Lösung für diese Herausforderung ist es, eine symbolische Ausführung heranzuziehen. Gängige Rechnermodelle arbeiten immer auf konkreten Werten, also Bits und Bytes. Bei einer symbolischen Ausführung wird anstatt mit konkreten Werten mit Wertebereichen und Gleichungssystemen gearbeitet. Für jeden Pfad durch ein Sys-

tem wird hierzu ein Gleichungssystem aufgestellt, das in einem weiteren Schritt mithilfe eines Constraint Solver gelöst wird, um einen konkreten Testkandidaten zu ermitteln.

Abbildung 2 zeigt schematisch, wie eine symbolische Ausführung die einzelnen Statements einer Funktion bearbeitet, um beispielsweise eine Division durch null zu detektieren.

Derzeit gibt es folgende Tools, die symbolische Ausführung verwenden, um Testfälle zu erstellen:

- KLEE (C/C++) ist das Resultat eines langjährigen Forschungsprojektes. KLEE ist ein frei verfügbares Open-Source-Projekt und kann zur Analyse vollständiger Binaries verwendet werden [3].
- IntelliTest (C#) wird als Teil der Enterprise-Version von Microsoft Visual Studio angeboten und kann zum automatischen Erstellen von Unit-Tests für C#-Code verwendet werden [4].
- Symflower (Java, Go) unterstützt Entwickler im gesamten Unit-Testing-Prozess, indem einerseits Test-Templates für TDD-Prozesse zur Verfügung gestellt werden und komplementär dazu Symbolic Execution eingesetzt wird, um Test-Kandidaten direkt aus der Implementierung abzuleiten [5].

Über die Autoren

Markus Zimmermann ist Gründer und CTO des Start-ups Symflower. Das Unternehmen setzt mathematische Modelle und künstliche Intelligenz ein, um die Produktivität von Softwareentwicklern zu steigern. Vor der Gründung von Symflower hat Markus für mehr als 15 Jahre als Software-Entwickler und Software Test Consultant gearbeitet.

Evelyn Haslinger ist Gründerin und COO des Start-ups Symflower. Vor der Gründung von Symflower arbeitete sie als Senior Developer und Scrum Master bei Sophos im Bereich Verschlüsselung.

Quellen:

- [1] „Understanding and Controlling Software Costs - ACM Digital Library.“ Accessed March 8, 2017, <http://dl.acm.org/citation.cfm?id=53072>
- [2] https://en.wikipedia.org/wiki/Test-driven_development
- [3] <https://klee.github.io/>
- [4] <https://docs.microsoft.com/en-us/visual-studio/test/intellitest-manual>
- [5] <https://symflower.com>



Markus Zimmermann
markus.zimmermann@symflower.com



Evelyn Haslinger
evelyn.haslinger@symflower.com

Oracle Datenbanken Monthly News

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

DOAG Online

Es ist wieder soweit: 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) mit den Links zur Verfügung

gestellt, um einen schnellen Zugriff auf die zugehörigen Links zu gewährleisten.

<https://www.doag.org/de/home/news/oracle-datenbanken-monthly-news-18/>





dbFlow – mit Git zu CI/CD in der Datenbank-Entwicklung

Maik Michel, Opitz Consulting Deutschland

CI/CD ist ein heiß begehrter Begriff. Doch im Datenbankumfeld ist die Umsetzung einer kontinuierlichen Integration gar nicht so einfach umzusetzen. Wie bekommen wir alle Änderungen für eine Applikation in die nächste Umgebung? Wie stelle ich eine fehlerfreie Auslieferung bereit und stelle sicher, dass alle Objekte und Abhängigkeiten in der richtigen Reihenfolge eingespielt werden?

Als Softwarearchitekten und Entwickler müssen wir in der Lage sein, die Perspektive zu wechseln. Sowohl *Bottom Up* als auch *Top Down* sind dabei wichtig. Das bezieht sich natürlich nicht nur auf die Applikation selbst, sondern auch auf die Art und Weise, wie man eine Applikation betrachtet. Betrachten wir unsere Applikation als Produkt. Welche Stufen durchläuft unser Produkt, bis es schließlich vom Benutzer bedient wird? Unser Produkt ist ständigen Neuerungen ausgesetzt und wird im Prinzip nie fertig. Es gibt immer wieder neue Ideen, die in Form von Features implementiert werden sollen. Zusätzlich ändern sich immer wieder Anforderungen. Bestehende Systeme unterliegen einem permanenten Wandel.

Anders als bei anderen Softwareprodukten haben wir in der Datenbankwelt das Problem vorhandener Daten und deren Struktur. Während also in anderen Bereichen „nur“ Quellcode kompiliert und ausgeliefert wird, müssen wir auch noch auf die bereits vorhandenen Daten und Strukturen eingehen. Das ist, als würden wir Komponenten und Features an einem Auto erneuern, das voll beladen auf der Autobahn fährt.

dbFlow

Im Datenbankumfeld sind mit Liquibase und Flyway zwei Lösungen im Umlauf, die sich dieser Problematik widmen. Beide Lösungen kümmern sich in Form von Changesets um die Migration verschiedener Versionsstände von Datenbankschemas. Das Migrieren von Tabellen beziehungsweise das Einspielen der Änderungen ist nur ein kleiner Teil einer Softwareeinspielung. Im APEX-Bereich gibt es viel mehr zu berücksichtigen, damit das Deployment gelingt. Applikationen sollten während der Einspielung offline gesetzt, REST-Services sollten deaktiviert werden. Während der Einspielung sollten automatisierte Tests (nicht auf Produktion) durchgeführt werden. Bestimmte Automatismen, wie das Erstellen von beispielsweise Table-APIs, sollten ausgeführt werden. Changelogs über die Änderungen sollten gesammelt und bestimmten Benutzerkreisen zur Verfügung gestellt werden.

Mit dbFlow stelle ich Ihnen eine Lösung vor, die sich neben der Veränderung

von Schemas beziehungsweise Versionsständen um all diese Themen kümmert und somit einen ganzheitlichen Lösungsansatz in der Auslieferung und Einspielung von Datenbank-basierten Anwendungen bietet.

dbFlow ist ein Open-Source-Tool unter der MIT-Lizenz. dbFlow läuft unter Bash und benötigt zum Bauen der Deployments zusätzlich nur noch Git. Zum Einspielen auf den jeweiligen Datenbanken kann wahlweise SQL*Plus oder SQLcl konfiguriert werden.

dbFlow baut im Kern auf 4 Konzepten auf:

1. Versionsverwaltung über Git und dem damit verbundenem Branch-Modell **Gitflow**
2. **2-Phasen-Deployment** für das Deployment als neues oder geändertes Produkt
3. **smartFS** als standardisierte Grundlage der Verzeichnisstruktur
4. Eigenes **Depot** als Artifactory

Gitflow

dbFlow setzt auf das Branch-Modell Gitflow auf. Gitflow beschreibt dabei einen Weg, wie Features und Änderungen in den Haupt-Entwicklungs-Branch (develop) gemergt werden. Damit wir diese Strategie in der Datenbankwelt benutzen können, legen wir Folgendes fest.

Für jede Zielstage unseres Produktes gibt es einen Branch, zum Beispiel test, acceptance, master (Production).

Somit werden Artefakte in der Testdatenbank eingespielt, die durch das Mergen in den Test-Branch mit dbFlow erstellt wurden. Das Gleiche gilt für eine mögliche Acceptance- oder Produktionsumgebung, in der Code aus dem Master-Branch eingespielt wird. Die Entwicklung findet im Haupt-Entwicklungs-Branch statt und kann gegebenenfalls noch auf weitere Feature-Banches verteilt werden. Der Develop-Branch sowie die Feature-Banches werden in der Entwicklungsdatenbank entwickelt. Ein zusätzlicher Release-Branch kann als Release-Schranke beziehungsweise Sperre fungieren, in der Anpassungen für das aktuelle Release nur noch im Release-Branch implementiert werden.

So kann die Entwicklung im Develop-Branch unabhängig vom aktuellen Release fortgesetzt werden.

dbFlow kann nun durch das Mergen in einen Branch das Deployment bauen. Im Grunde wird durch das Delta der veränderten Dateien ein Patch gebaut, der diese dann in die jeweilige Zieldatenbank einspielt. Durch die Nutzung von Git kann man sich dabei auf den exakten Zustand vor dem Mergen oder irgendeinen anderen in der Vergangenheit beziehen. Das Artefakt, das dbFlow nun erstellt (Tarball), kann jetzt in der Zieldatenbank eingespielt werden (*siehe Abbildung 1*).

2-Phasen-Deployment

dbFlow bedient sich eines sogenannten 2-Phasen-Deployments. Dabei kann ein Artefakt gegen eine „leere“ Datenbank oder als Patch auf eine bestehende Version eingespielt werden. Das funktioniert wie die Installation eines Office-Paketes. Entweder habe ich noch kein Office installiert, dann muss ich install.exe starten. Oder ich habe bereits eine Version installiert und muss nun update.exe ausführen.

Diese Art des Deployments bedeutet, dass man jede Version als Initial-Version und zusätzlich als passendes Update (Patch) für den jeweiligen Vorgänger erstellen kann. So kann man zum Beispiel Version 3 seiner Applikation initial deployen oder als Patch für die Version 2 (Delta) (*siehe Abbildung 2*).

Mit diesem Konzept ist es ein Leichtes, die Release-Fähigkeit einer Applikation permanent feststellen zu lassen (Continuous Integration). Wir haben hierfür eine entsprechende Pipeline in Jenkins hinterlegt. Erst, wenn sich alles einspielen lässt und alle Tests erfolgreich waren, ist ein Versionsstand releasefähig.

smartFS

Das Konzept von dbFlow setzt auf Git als Versionsverwaltung auf. Das heißt, es werden nur Dateien versioniert. Diese **MÜSSEN** im Dateisystem vorhanden sein!

Es gilt: Das Dateisystem gewinnt. Immer. *Immer!*

Dieser Ansatz führt zu einem Paradigmenwechsel. Und genau diesen kann

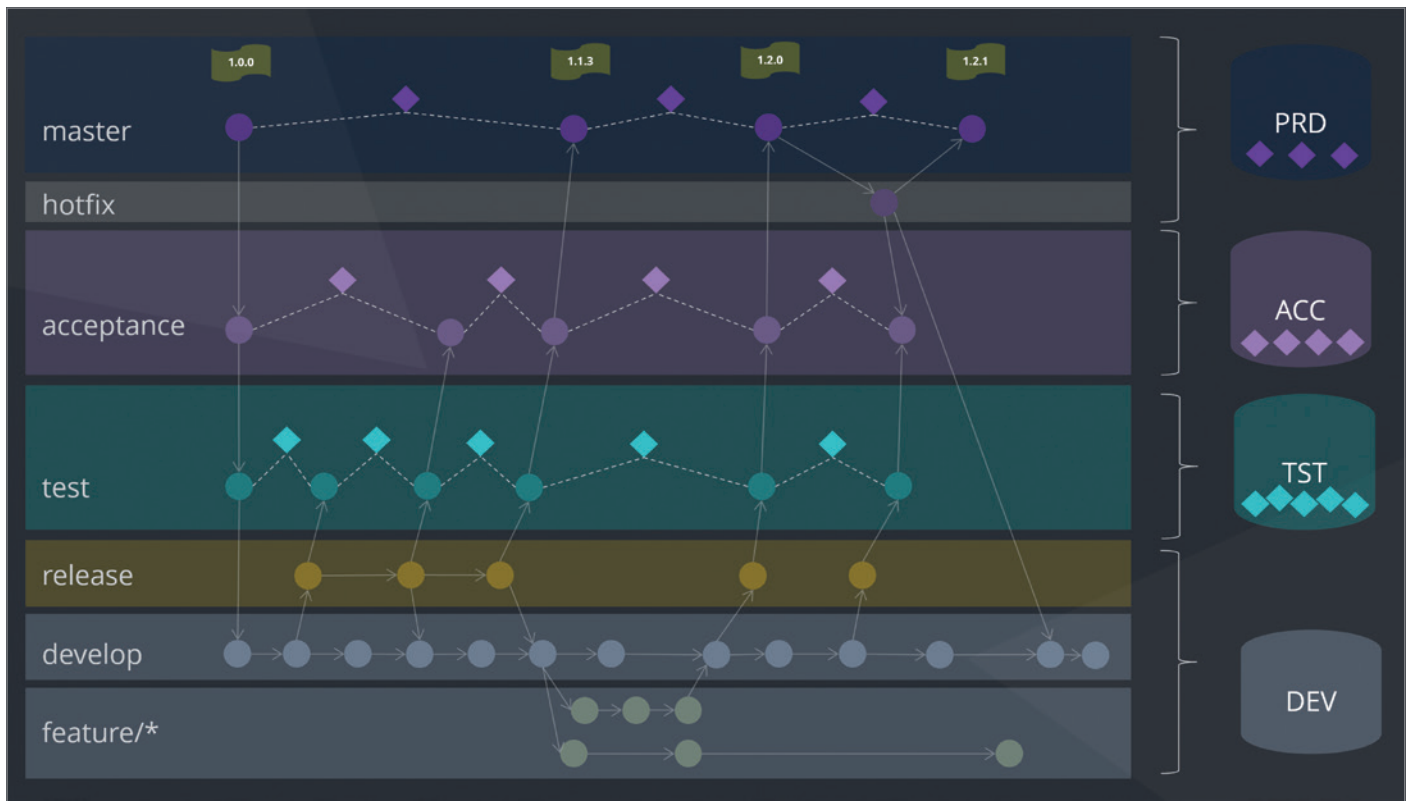


Abbildung 1: Gitflow-Branch-Modell mit Zuordnung zu Stage-Datenbanken (Quelle: Maik Michel)

ich Ihnen, liebe Leserin / lieber Leser, nur wärmstens ans Herz legen. Nutzen Sie zur Entwicklung einen Quellcode-Editor, wie zum Beispiel VSCode in Verbindung mit dbFlux (siehe Red Stack 01/2022, S.64). Wie schnell hat man seine Änderungen im SQL-Developer oder TOAD umgesetzt und vergessen, in die Versionskontrolle einzuchecken. Arbeiten Sie lieber von außen nach innen (Dateien schreiben und in die DB einspielen) als von innen nach außen (Packages, Tabellen anlegen und in das Dateisystem exportieren/speichern).

smartFS stellt hier eine standardisierte Verzeichnisstruktur dar. Auf der obersten Ebene stehen Verzeichnisse für verschiedene APEX-Applikationen, REST-Module sowie Datenbankschemas bereit. Ferner werden hier Konfigurationsdateien abgelegt, die das Projekt sowie die jeweilige Zieldatenbank definieren (build.env, apply.env). smartFS unterstützt drei verschiedenen Projektarten, die sich von der Anzahl der benutzten Schemas herleiten.

Das SingleSchema kennt nur ein Schema, während das MutliSchema drei Schemas kennt und sich hier des klassischen 3-Schichten-Modells bedient (DATA für Tabellen, LOGIC für die Geschäftslogik und APP für die Applikationslogik).

Dann gibt es noch das FlexSchema, das n Schemas kennt. Zuordnungen zum Workspace und Schema werden im SingleSchema- und MultiSchema-Mode in der Konfiguration hinterlegt. Beim FlexSchema-Mode werden diese Informationen als zusätzliches Level in der Verzeichnishierarchie dargestellt.

Neben den jeweiligen Schemas wird im db-Ordner auch ein _setup-Ordner bereitgestellt. Hier werden die Skripte abgelegt, die dann eingesetzt werden zur Erstellung der Schemas, der jeweiligen DB-Benutzer sowie der abhängigen Features, wie beispielsweise Logger oder utPLSQL. In der Projektdefinition wird hinterlegt, mit welchem Admin-Account diese Skripte ausgeführt werden sollen. Jedes Datenbankschemaverzeichnis hält für jeden in der DB befindlichen Objekttyp ein eigenes Unterverzeichnis bereit. Manche Verzeichnisse sind noch weiter unterteilt. Diese Unterteilung bestimmt später die Reihenfolge der jeweiligen Dateien während der Einspielung.

In jedem Level des Verzeichnisbaums gibt es sogenannte Hook-Ordner (.hooks), die sich abhängig von der jeweiligen Installationsart (init oder patch) und dem jeweiligen Einspielungszeitpunkt während eines Deployments untertei-

len. Diese Hook-Ordner können Skripte enthalten, die während der Installation aufgerufen werden. Somit können zum Beispiel TableAPIs erstellt oder UnitTests ausgeführt werden.

Da dbFlow das sogenannte 2-Phasen-Deployment unterstützt, werden Tabellenänderungen doppelt gepflegt. Das bedeutet, es gibt ein Tabellenskript im Verzeichnis tables, das das komplette Create-Table-Statement enthält. Ein weiteres Skript enthält die Änderungen an der jeweiligen Tabelle und liegt im Unterverzeichnis tables/tables_ddl.

Beispiel: Fügen wir eine neue Spalte an die Tabelle employees ein, dann wird das create statement in der Datei employees.sql angepasst und es wird eine neue Datei employees.1.sql (1 steht für die erste Änderung) im Verzeichnis tables_ddl erstellt, die das Alter-Table-Statement enthält. Später, während des Deployments, wird dbFlow feststellen, dass es Änderungen an Dateien mit dem gleichen Dateistamm im tables-Verzeichnis sowie im tables_ddl-Verzeichnis gibt, und daher nur das Skript aus dem tables_ddl-Verzeichnis ausführen, sofern es sich beim Deployment um einen Patch handelt. Im Initfall wird nur das Create-Table-Skript ausgeführt.

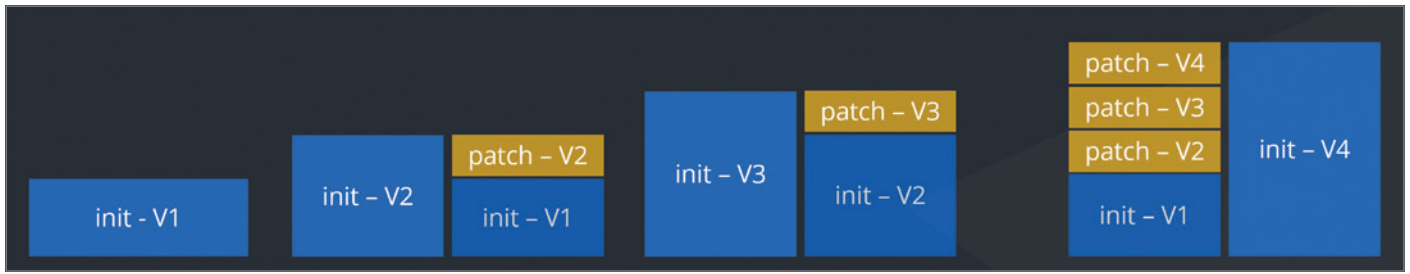


Abbildung 2: Version 3 seiner Applikation initial deployen oder als Patch für die Version 2 (Delta) (Quelle: Maik Michel)

Für das Ändern von Daten oder Spezialfälle wie Aufräumarbeiten stehen die Ordner *ddl* beziehungsweise *dml* bereit. Hier werden ähnlich wie bei den Hooks Skripte hinterlegt, die abhängig von der Installationsart oder dem Einspielungszeitpunkt aufgerufen werden.

depot

Das Depot stellt eine Art Artifactory dar und wird als Zielverzeichnis konfiguriert. Abhängig vom Git-Branch, von dem aus ein Deployment erstellt wird, wird das einzuspielende Artefakt, das dbFlow erstellt (Tarball), in ein Unterverzeichnis des Depots abgelegt. Zum Beispiel: *depot/test/patch_4.7.12.tar*

Im Instanzverzeichnis, von dem aus jede Installation in der entsprechenden Stage ausgeführt wird, kann nun mit dbFlow dieses Artefakt installiert werden. dbFlow wird dabei im *depot* im Unterverzeichnis *test* den Patch mit der entsprechenden Version holen und einspielen, weil die Stage dem Branch **test** zugeordnet wurde. Sofern es kei-

ne Fehler beim Deployment gab, werden der Patch sowie die jeweiligen Logs in einem Unterverzeichnis mit dem Namen **success** passend zum Artefakt abgelegt. Einspielungen, die schiefgelaufen sind, werden dabei im Ordner **failure** abgelegt.

dbFlow hat die Möglichkeit, eine Installation an dem Punkt fortzusetzen, an dem etwas fehlgeschlagen ist. Dabei wird das jeweilige Logfile als Parameter übergeben. dbFlow wird nun mit der Datei fortfahren, die nicht zu einem positiven Feedback in der alten Logdatei geführt hat.

Ich empfehle, das Depot-Verzeichnis außerhalb des Entwicklungsverzeichnisses zu platzieren und ebenfalls als Git-Repository anzulegen. Somit stehen Automatismen zur Verfügung, um ein Artefakt automatisiert in eine Zielumgebung einzuspielen.

Prozess im Ganzen

In der Gesamtbetrachtung stellt sich der Workflow so dar: Sie entwickeln in einem Verzeichnisbaum, der dem

smartFS-Aufbau entspricht, in einem Haupt-Entwicklungs-Branch oder Feature-Branch, auf einer Entwicklungsdatenbank und checken ihre jeweiligen Änderungen entsprechend mit Git in die Versionsverwaltung ein. Hier kann ein CI/CD-Tool, zum Beispiel Jenkins, den Entwicklungs-Branch komplett oder als Delta zum Test-Branch in eine Builddatenbank einspielen (Nightlybuilds). Der Releaseverantwortliche kann, sofern natürlich alle Tests erfolgreich abgeschlossen wurden, durch das Mergen über den Release-Branch (Release-Schranke) in den Test-Branch das Artefakt mit dbFlow erstellen und im depot ablegen. Automatisiert oder manuell kann nun, von einem Instanzverzeichnis aus, das Artefakt mit dbFlow eingespielt werden. Anfallende Logs werden von dbFlow in das Depot-Verzeichnis unter *success* wieder abgelegt (siehe Abbildung 3).

Wie kann ich starten?

dbFlow wird als Submodul in einem bereits bestehenden Git Repository installiert. (siehe Listing 1 für die Anlage eines dbFlow-Projekts).

dbFlow hat nun ein Verzeichnis inklusive Schemas, User, Features und APEX Workspace angelegt. Jetzt kann das Entwickeln auf dem Develop-Branch losgehen.

Sobald alle auszuliefernden Änderungen commitet wurden, kann in den Ziel-Branch, zum Beispiel *test*, gemergt werden. Um nun einen Patch mit der Version 1.0.0 aus dem Delta zu erzeugen, führt man folgenden Befehl aus: `.dbFlow/build.sh --patch --version 1.0.0`

dbFlow baut nun den Patch und legt diesen im Depot ab. Wechselt man nun in das Instanzverzeichnis oder etwa auf

```
# create a folder for your project and change directory into
$ mkdir demo && cd demo

# init your project with git
$ git init

# clone dbFlow as submodule
$ git submodule add https://github.com/MaikMichel/dbFlow.git .dbFlow

# generate and switch to your development branch
$ git checkout -b develop

# generate project structure
$ .dbFlow/setup.sh generate <project_name>

# after processing the wizard steps, just install
$ .dbFlow/setup.sh install
```

Listing 1: bash-Befehle für die Anlage eines dbFlow-Projekts

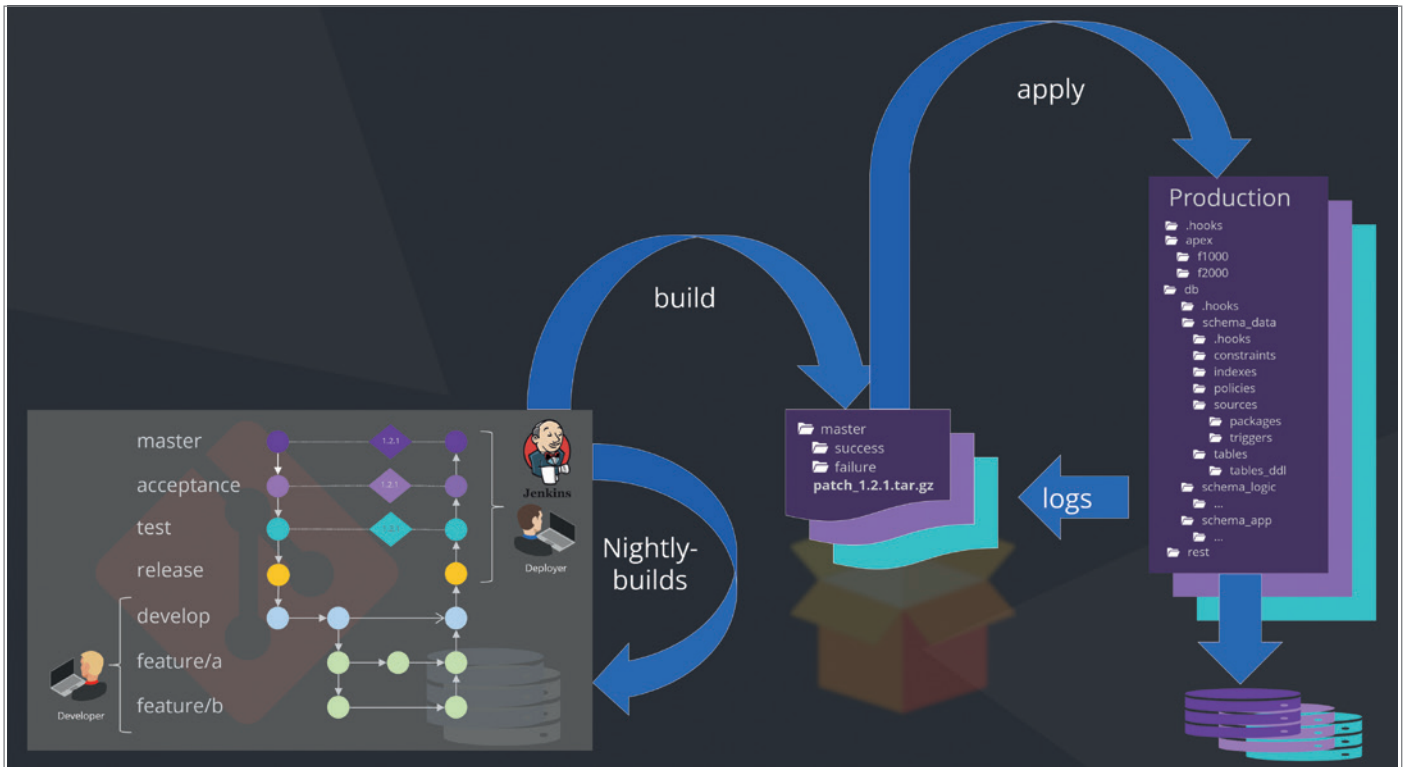


Abbildung 3: Zusammenfassung des Prozesses im Ganzen (Quelle: Maik Michel)

eine andere VM, kann man mit dem folgenden Befehl den Patch einspielen: `.dbFlow/apply.sh --patch --version 1.0.0`

Dieser Artikel kann nicht alle Funktionalitäten von dbFlow beschreiben. Ich denke, dass Sie einen ersten Eindruck von dbFlow gewonnen haben, und lade Sie herzlich ein, dbFlow einfach mal auszuprobieren.

dbFlow inklusive Dokumentation finden Sie auf GitHub unter folgendem Link: <https://github.com/MaikMichel/dbFlow>

Quellen

- 1. Titelbild: Unsplash - <https://unsplash.com/photos/KPAQpJYzHOY>

Über den Autor

Maik Michel ist LEAD Developer und Consultant bei der Opitz Consulting Deutschland GmbH. Hier verantwortet er verschiedene Projekte und Teams rund um das Thema Digitale Transformation mit der Low-Code-Plattform APEX. Daneben ist er Autor des Blogs micodify.de und als Sprecher auf verschiedenen Konferenzen zu finden.



Maik Michel
maik.michel@opitz-consulting.com



In zwei Tagen zur neuen Exadata-Version – aus dem Protokoll eines DBA

Andrzej Rydzanicz, Opitz Consulting Deutschland

Der Artikel beschäftigt sich mit der Migration von der Oracle Exadata X8 in der Public Cloud auf die Version X9M, ebenfalls in der Cloud. Wir tauchen ein in ein Kundenprojekt, bei dem es darum ging, die fast 300 Terabyte große Datenbank eines Online-Händlers auf die neue Exadata-Version zu migrieren und diese in kürzester Zeit auf die neue Hardware umzuziehen. Der Artikel ist für alle interessant, die eine Exadata in der Public Cloud betreiben und sich mehr Flexibilität bei Speicher- und Rechenleistung wünschen – und natürlich für alle, die ein Exadata-System in kürzester Zeit auf die neueste Hardware umziehen wollen.

Die wichtigsten Arbeitsschritte, die wir beim Umzug der Exadata X8 zur X9M in der Cloud gegangen sind, Tools, die wir empfehlen, und Stolperfallen, die Sie vermeiden können – dies alles gehen wir in diesem Artikel mit Ihnen gemeinsam durch. Auch andere Dinge, die für die Migration wichtig sind, kommen dabei zur Sprache:

- Wie baue ich die Umgebung für die Exadata auf?
- Wie stelle ich den VM-Cluster bereit?
- Wie greife ich auf den bestehenden Object Storage zu?
- Wie skaliere ich die Exadata in den Bereichen Storage und CPU?

Bei der Konfiguration des Object Storage, wo Backups der alten Prod-Datenbank gespeichert werden, haben wir beispielsweise Dinge erlebt, die uns vorher nicht bewusst waren.

Knöpfchen drücken, und fertig?

In der letzten Zeit habe ich mit meinem Service-Team viele interessante Projekte durchgeführt. In einem von ihnen

ging es um die Migration eines 300-Terabyte-Data-Warehouse von der Exadata X8 zur Exadata X9M – beide Systeme in der Public Cloud von Oracle. Wir hatten nicht viel Zeit. Zu gerne hätten wir nur ein Knöpfchen gedrückt, und fertig. Nicht nur wegen der Größe der Datenbank, sondern auch wegen der vielen Tests, die wir in Vorfeld durchführen mussten, um beim Go-live auf der sicheren Seite zu sein, brauchten wir am Ende 2 Tage und 30 Minuten.

Doch zurück zum Anfang: Das Ganze fing damit an, dass einer unserer Kunden, ein Online-Händler, mit Elan in eine beliebte Rabattwoche starten wollte. Alles war vorbereitet für die „Cyber Week“ im November, als Zweifel aufkamen. Die IT befürchtete, dass die Datenbanksysteme den vermehrten Transaktionen in dieser Woche nicht gewachsen sein könnten. Genau genommen ging es um die zu geringe Plattenkapazität der Exadata. Unglücklicherweise ließ sich der Speicher der bestehenden Version nicht durch zusätzliche Storage Cells erweitern. Bei der Version X9M hingegen sind Erweiterungen überhaupt kein Problem. Hier lassen sich Storage Cells dynamisch hinzufügen. Gesagt, getan: Eine Migration sollte das Problem lösen!

Da nicht viel Zeit blieb, mussten wir einen Weg finden, um das 300-Terabyte-Data-Warehouse vor allem schnell, aber natürlich auch fehlerfrei zu migrieren. Für die Datenmigration nutzten wir Data Guard. Denn Data Guard bietet die Möglichkeit, die Funktionalität der Datenbank und die Integrität der migrierten Daten in einer Snapshot-Standby-DB zu testen. Wie sich zeigen sollte, nicht ganz ohne Hindernisse, aber dazu später mehr ... Fangen wir vorne an:

Ein Umfeld für die neue Exadata

Der Aufbau der Umgebung für die Exadata X9M lief wie üblich in zwei Schritten ab:

1. Neue Hardware und Exadata-Infrastruktur verfügbar machen
2. VM-Cluster einrichten

Da das Unternehmen bereits eine Exadata X8-2 in der Public Cloud betrieb und fast alle Komponenten der Infrastruktur wie Compartment, Availability Domain oder Object- und Archive-Storage-Buckets vorhanden waren, gestaltete sich Schritt 1, also der Aufbau von Hardware und Infrastruktur, nicht

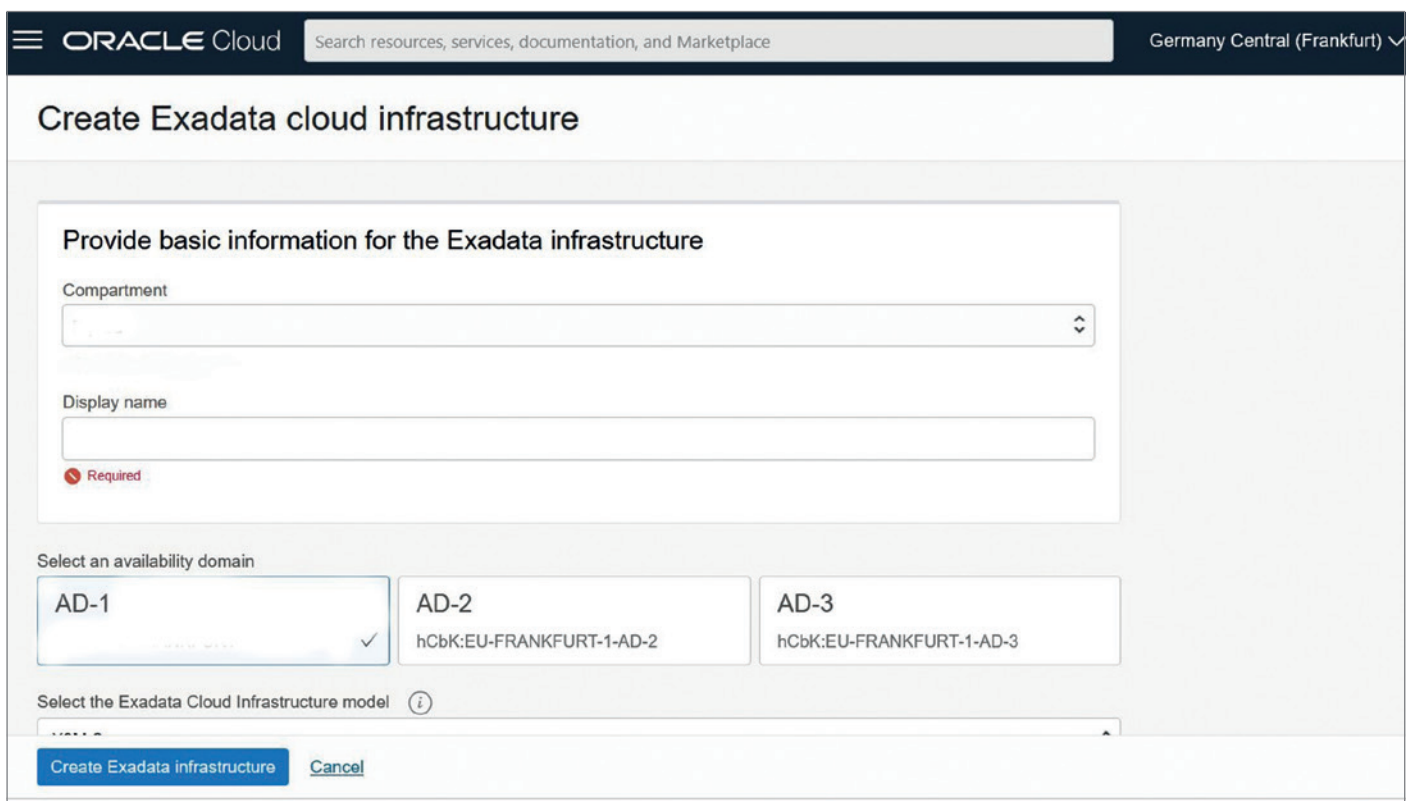


Abbildung 1: Basisinformationen für die Exadata-Infrastruktur (Quelle: Oracle)

ORACLE Cloud Search resources, services, documentation, and Marketplace Germany Central (Frankfurt)

Create Exadata cloud infrastructure

Select an availability domain

AD-1 AD-2 AD-3

Select the Exadata Cloud Infrastructure model ⓘ

X9M-2

Compute and storage configuration

You can add additional database and storage servers after provisioning. [Learn more.](#)

Database servers

4

Storage servers

6

Resource Totals

OCPUs: 504

Storage: 384 TB

Create Exadata infrastructure Cancel

Abbildung 2: Konfiguration der Compute- und Speicher-Einheiten (Quelle: Oracle)

Create Exadata VM cluster

Compartment

Display name

VMCluster-202211081112

Cluster name *Optional*

Select Exadata infrastructure in [\(Change Compartment\)](#)

Select an infrastructure

Configure the VM cluster

Specify OCPU count per virtual machine ⓘ

2

Requested OCPU count for the Exadata VM cluster *Read-only*

4

Specify a number of OCPUs between 2 and 50, or specify 0 to provision with no OCPUs enabled.

Abbildung 3: VM-Cluster konfigurieren (Quelle: Oracle)

so kompliziert. Neu hinzu kam ein Subnet im Virtual Cloud Network. Denn die IP-Adressen im bestehenden Subnet reichten für die neue Exadata-Infrastruktur nicht aus.

Aber auch Details waren zu beachten: Ein Cloud-Infrastrukturmodell musste aufgesetzt, die Rechen- und Speicher-

Konfiguration musste eingerichtet und benannt werden, ebenso die Availability Domain und das in unserem Fall bereits vorhandene Netzwerk-Compartment.

Die Screenshots (siehe Abbildungen 1 bis 7) zeigen die wichtigsten Eingabemaschinen im Provisioning-Prozess:

Weiter ging es mit Schritt 2: der Erstellung eines VM-Clusters. Hier galt es festzulegen, welche Grid-Infrastruktur-Version installiert werden soll. Wer mit dem Deployment von alten Bare-Metal-Exadata-Systemen vertraut ist, findet sich hier schnell zurecht. Hier werden Dinge wie

Create Exadata VM cluster

i The following storage configuration options cannot be changed after VM cluster creation.

- Allocate storage for Exadata sparse snapshots **i**
- Allocate storage for local backups **i**

Usable storage allocation: Data: 80%. Reco: 20%. Sparse: 0%.

Oracle Grid Infrastructure version **i**

Select an Exadata Cloud Service infrastructure resource first

Add SSH key **i**

You can access the services of your system's databases by using SSH tunneling. When you connect to the system, you will provide the associated private key.

- Generate SSH key pair
- Upload SSH key files
- Paste SSH keys

i Download the private key so that you can connect to the database system using SSH. It will not be shown again.

[↓ Save private key](#) [↓ Save public key](#)

Abbildung 4: SSH-Key hinzufügen (Quelle: Oracle)

Configure the network settings **i**

Virtual cloud network in [\(Change Compartment\)](#)

Select a virtual cloud network

Client subnet in [\(Change Compartment\)](#)

First select Virtual Cloud Network

Use network security groups to control traffic

Backup subnet in [\(Change Compartment\)](#)

First select Virtual Cloud Network

Use network security groups to control traffic

Hostname prefix **i**

Host domain name *Read-only*

This value is determined by the client subnet.

Host and domain URL *Read-only*

This value is determined by the hostname prefix and the host domain name.

Choose a license type

License included Subscribe to new Oracle Database software licenses and the Database service.	Bring Your Own License (BYOL) Bring my organization's Oracle Database software licenses to the Database service. Learn more.
---	--

Abbildung 5: Netzwerkeinstellungen vornehmen (Quelle: Oracle)

ORACLE Cloud Search resources, services, documentation, and Marketplace

Create database software image

Database software images can be created for virtual machine, bare metal, and Exadata DB systems. [Learn more](#)

Provide basic information for the database software image

Display name
RDBMS_19_8_OneOffs

Select a compartment

Shape family
Virtual Machine and Bare Metal Shapes Exadata Shapes ✓

Configure the database software image

Database version
19c

Choose a patch set update, proactive bundle patch, or release update ⓘ
19.8.0.0

Enter one-off patch numbers *Optional*

Create database software image Cancel

Abbildung 6: Database-Software-Image einrichten (Quelle: Oracle)

Host Prefix, Host Domain, Client Subnet oder Backup Subnet definiert.

In zwei Tagen hatten wir die Umgebung für die neue Exadata-Version installiert, ein gutes Zwischenergebnis!

RDBMS Deployment und Patches

Wenn die Exadata läuft, gibt es noch einiges zu tun. Es müssen Patches und weitere Dinge deployt und installiert werden, wie zum Beispiel das RDBMS Home. Da wir die produktive Datenbank auf RDMS 19.8 Home betrieben, mussten wir diese Version nun auf den neuen Cluster heben. Hier kam für uns eine sehr interessante Möglichkeit zum Vorschein: Wir erstellten via OCI Console ein „Gold-Image“ mit allen OneOffs, RUs und CVEs, die wir brauchten. Aber Achtung: Das geht nur, wenn die Patches nicht passwortgeschützt sind!

Was die Grid-Infrastruktur-Version 19.16 betrifft: Alle Patches, die hierfür zur Ver-

fügung stehen, hatten wir schon im Zuge der VM-Cluster-Erstellung installiert. Ebenso die Patches für das OS-Image. Dies war wichtig, um später eine Downtime nach dem Switchover zu verhindern. Vor dem Switchover war das System also bereits auf dem neuesten Stand.

Warum Oracle Data Guard?

Auf welche Art migriert man eine fast 300 Terabyte schwere Datenbank voller Kundendaten? Die Entscheidung für das Tooling ist uns nicht leichtgefallen. Wir haben uns diverse Möglichkeiten für die Datenbankreplikation angeschaut. Wie schon erwähnt auch GoldenGate. Letztendlich überzeugte uns aber Data Guard. Warum? Diese Vorteile haben uns überzeugt:

- Data Guard ist seit vielen Jahren als Technologie ausgereift.
- Es bietet viele Möglichkeiten, um Probleme im Vorfeld zu analysieren.

- Es gibt diverse Tests, die vor der tatsächlichen Migration durchgeführt werden können. Vor allem die Snapshot-Standby-Datenbank bietet hierfür viele Möglichkeiten. Dorthin kann man innerhalb von Data Guard konvertieren.
- Die Endian-Plattform, die für die Migration zur Verfügung steht, hilft Zeit zu sparen.
- Data Guard ist weniger komplex als beispielsweise GoldenGate und damit leichter zu handhaben.

Die Standby-Datenbank war mit RMAN Restore/Recovery aufgebaut worden. Unser Ziel war eine RAC-Standby-Datenbank. Die Herausforderung lag darin, den schnellsten Weg zu finden, um die enorme Datenmenge der produktiven Datenbank zu übertragen, aber auch kein Risiko einzugehen. Eine Datenbank dieses Kalibers mittels RMAN Duplicate oder Restore aufzubauen, kann bedeuten, dass während eines Speichervorgangs neue

```

run {
  ALLOCATE CHANNEL ch01 DEVICE TYPE 'SBT_TAPE' CONNECT 'sys/@<DB_UNIQUE_NAME[1-4]>' PARMS 'SBT_LIBRARY=/
var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/
opc<DB_NAME>.ora)';
.
.
.
  SET NEWNAME FOR DATABASE TO '+DATA1';
  restore database;
}
switch database to copy;

```

Listing 1: Restore-Skript

```

run {
  ALLOCATE CHANNEL ch01 DEVICE TYPE 'SBT_TAPE' CONNECT 'sys/@<DB_UNIQUE_NAME>' PARMS 'SBT_LIBRARY=/var/
opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/op-
c<DB_NAME>.ora)';
.
.
recover database from service '<DB_UNIQUE_NAME_PROD>' section size 512G;
}
switch database to copy;

```

Listing 2: Recovery-Skript

```

dbaascli database backup --dbName <DB_NAME> --configure --configFile /home/oracle/backup_config/bkup-<DB_
NAME>.cfg
DBAAS CLI version 22.3.1.1.0
Executing command database backup --dbName <DB_NAME> --configure --configFile /home/oracle/backup_config/
bkup-<DB_NAME>.cfg
DBaaS Backup API V1.5 @2022 Multi-Oracle home
-> Action : set_config
-> logfile: /var/opt/oracle/log/<DB_NAME>/bkup_api_log/bkup_api_3e777dfe_20225123543.717530.log
cfgfile : /home/oracle/backup_config/bkup-<DB_NAME>.cfg
Using configuration file: /home/oracle/backup_config/bkup-<DB_NAME>.cfg
API::Parameters validated.
UUID 77b3e5ae449910200990708cd for this set_config(configure-backup)
** process started with PID: 158063
** see log file for monitor progress
-----
dbaascli execution completed

```

Listing 3: Kommando

```

dbaascli database backup --status --uuid 77b3e5ae449910200990708cd --dbname <DB_NAME>

DBaaS Backup API V1.5 @2022 Multi-Oracle home
@ STARTING CHECK STATUS 77b3e5ae449910200990708cd
[ REQUEST TICKET ]
[UUID    -> 77b3e5ae449910200990708cd
[DBNAME  -> <DB_NAME>
[STATE   -> success
[ACTION  -> configure-backup
[STARTED -> 2022-10-05 10:35:45 UTC
[ENDED   -> 2022-10-05 10:38:31 UTC
[PID     -> 158563
[TAG     ->
[PCT     ->
[ END TICKET ]
dbaascli execution completed

```

Listing 4: Abruf des aktuellen Stands der Konfiguration

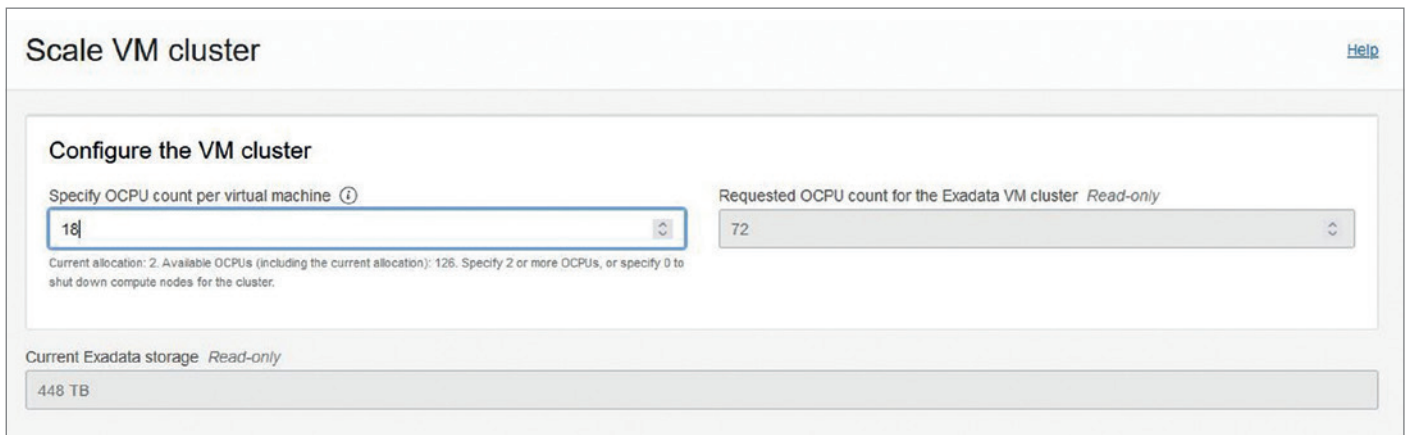


Abbildung 7: Anzahl der OCPUs pro Cluster erhöhen (Quelle: Oracle)

```
ALTER DATABASE FLASHBACK ON;
```

Listing 5: Anpassung einer Einstellung auf der Standby-Seite

```
ALTER SYSTEM SET db_flashback_retention_target=<Anzahl-Minuten>;
```

Listing 6: Anpassung einer zweiten Einstellung auf der Standby-Seite

Daten auf der Primary-Datenbank hinzugefügt werden. Was leicht zu Problemen bei Recovery-Vorgängen führen kann.

Um dem vorzubeugen, stellten wir den Restore auf 50 Channels per Instanz ein. Damit kamen wir auf insgesamt 200 Channels. Mit dieser Konfiguration dauerte der Restore-Vorgang ungefähr 14 Stunden. Wobei ich erwähnen muss, dass wir, um das Ganze performant zu halten, die Anzahl der CPUs auf dem VM-Cluster noch auf 18 OCPUs pro virtuelle Maschine erhöht haben, auf 72 OCPUs pro Cluster (siehe Abbildung 7).

Der Vorteil liegt auf der Hand: Beim Restore-Vorgang verteilt man die Arbeit auf mehrere Instanzen, um den Prozess zu beschleunigen, beim Recovery-Vorgang profitiert man allerdings nur von Channels, die innerhalb eines Datenbankknotens allokiert wurden (siehe Listing 1 und 2). Hier kann jeweils nur eine Apply-Instanz aktiv sein. Der Recovery-Vorgang kostete uns etwa drei Stunden. Kein Wunder bei Redo-Logs in einer Größe von 30 Gigabyte!

Der Restore- und Recovery-Prozess hätte ohne die Konfiguration des Object-Store-Zugriffs so nicht funktioniert. Die Konfiguration des OCI-Backup-Moduls erwies sich als problematisch. Ein Workaround musste helfen:

Der Workaround bestand in der Erstellung einer Template-Datenbank (OCI „create database“). Damit konnten wir sichergehen, dass alle Verzeichnisstrukturen zum Beispiel für Wallets und Logfiles für die zukünftige Standby-Datenbank vorliegen. Außerdem wurde die Datenbank automatisch im CRS registriert. Das ersparte uns manuelle Schritte wie das Hinzufügen einer Instanz oder Ähnliches. Der Zugriff auf den Object Store wurde bereits auf der Primary-Seite konfiguriert, sonst wäre es nicht möglich gewesen, die Backups auf der Primary-Datenbank zu speichern.

Die gleiche Konfiguration führten wir auf der Standby-Datenbank durch. Dafür generierten wir eine Konfig-CFG-Datei auf der Primary-Datenbank und übertrugen sie folgendermaßen auf die Standby-Seite:

```
dbaascli database backup --get-Config --dbname <DB_NAME>
```

Es ist nicht nötig, jeden einzelnen Punkt aus dieser Datei zu besprechen, die Kommentierung ist ausreichend und selbsterklärend. Was ich aber erwähnen möchte, ist der Fakt, dass man das OSS-Passwort (oss_password) in der Datei im Plain-Text-Format angeben muss, bevor

man das in Listing 3 abgebildete Kommando ausführt.

Der aktuelle Stand der Konfiguration lässt sich mit der in Listing 4 abgebildeten Abfrage aufrufen.

Natürlich müssen wie üblich Standby-Redo-Logs auf der Primary- und Standby-Datenbank erstellt werden. Mittels „Restore/Recover from Service“ werden die Standby-Redo-Logs automatisch auf der Standby-Datenbank angelegt.

„Snapshots“ für mehr Sicherheit

Mit Data Guard waren wir in der Lage, die physikalische Standby-Datenbank in eine Snapshot-Standby-Datenbank zu konvertieren.

Was ist eigentlich eine Snapshot-Standby-Datenbank?

Eine Snapshot-Standby-Datenbank ist eine geöffnete Standby-Datenbank, mit der alle Aktionen durchgeführt werden können, wie mit jeder anderen Datenbank auch. Es sind also sowohl Lese- als auch Schreiboperationen erlaubt. Das Besondere: Während die Snapshot-Standby-Datenbank geöffnet ist, unterbricht der Recovery-Vorgang. Die Snapshot-Standby-Datenbank wird also zunächst auf den Zeitpunkt zurückgesetzt, an dem sie geöffnet wurde.

Und so funktioniert es:

- Beim Öffnen der Datenbank wird ein Restore Point erstellt, auf den die Datenbank wieder zurückgesetzt werden soll.
- Alle Recovery-Aktionen, die durch das Öffnen der Snapshot-Standby-Datenbank ausgesetzt waren, werden nachgeholt.

```
DGMGRL> connect sys/password
Connected to "<standby_db_unique_name>"
Connected as SYSDBA.
DGMGRL> convert database "<standby_db_unique_name>" to snapshot standby ;
Converting database "<standby_db_unique_name>" to a Snapshot Standby database, please wait...
Database "<standby_db_unique_name>" converted successfully
```

Listing 7: Konvertieren zur Snapshot-Datenbank

```
DGMGRL
DGMGRL> connect sys/password
Connected to "<standby_db_unique_name>"
Connected as SYSDBA.
DGMGRL> convert database <standby_db_unique_name> to physical standby;
Converting database "<standby_db_unique_name>" to a Physical Standby database, please wait...
Operation requires a connection to database "<primary_db_unique_name>"
Connecting ...
Connected to "<primary_db_unique_name>"
Connected as SYSDBA.
Oracle Clusterware is restarting database "<standby_db_unique_name>" ...
Connected to "<standby_db_unique_name>"
Connected to "<standby_db_unique_name>"
Continuing to convert database"<standby_db_unique_name>" ...
Database "<standby_db_unique_name>" converted successfully
```

Listing 8: Restore Flashback

```
Select job_name, database_role, enabled from dba_scheduler_job_roles;

JOB_NAME          DATABASE_ROLE     ENABL
-----
JOB_NAME_XYZ      PRIMARY           FALSE
```

Listing 9: Ansicht der Job-Konfiguration

```
SQL> select database_role from v$database;

DATABASE_ROLE
-----
SNAPSHOT STANDBY
```

Listing 10: Abfrage der Snapshot-Datenbank

```
Dbms_scheduler.set_attribute (name => '<job_name> ', attribute => 'database_role', value => 'snapshot
standby or ALL ');
End;
```

Listing 11: Änderung der Job-Definition auf der Primary-Datenbank

```
select 'exec Dbms_scheduler.set_attribute (name => ''' || owner || '.' || job_name || ''' || ',' || 'at-
tribute => ' || ''' || 'database_role' || ''' || ',' value => '|| '''|| 'SNAPSHOT STANDBY' || ''' || '''
|| ');'|| ' '
from dba_scheduler_jobs;
```

Listing 12: SQL-Skript

- Die Länge des Zeitraums, in dem die Snapshot-Standby-Datenbank geöffnet war, bestimmt die Dauer der Synchronisation. Die Produktivdatenbank ist in keiner Weise betroffen, es gibt also keinen Overhead.

Für die Snapshot-Standby-Datenbank müssen zwei Einstellungen auf der Standby-Seite angepasst werden (siehe Listing 5 und 6).

Das Konvertieren zur Snapshot-Datenbank ging in rund 30 Minuten über die Bühne (siehe Listing 7).

Wenn man im Hinterkopf behält, dass wir die Standby-Datenbank im Snapshot-Modus zwei Tage lang intensiv getestet und viele Änderungen bei den Daten durchgeführt haben, dann sind 30 Minuten für das Zurückwandern zur physikalischen Standby-Datenbank relativ wenig. In diesen 30 Minuten fand auch der Restore Flashback statt, der die Datenbank auf die Zeit des Restore Point zurücksetzte (siehe Listing 8).

Als 8.000 Scheduling-Jobs plötzlich weg waren ...

Da unser Kunde über 8.000 Scheduling-Jobs für interne Zwecke benutzt, war es wichtig, diese Funktionalität auf der Standby-Seite zu testen. Hier erwarteten wir keine Probleme. Schließlich machen wir die Datenbank dafür im Read/Write-Modus auf. Hier sollte alles 1:1 identisch sein mit der Primary-Datenbank, oder?

Leider nicht so ganz: Als wir die Jobs starten wollten, waren die überhaupt nicht da. Ein `select count(*) from v$scheduler_jobs` zeigte nur interne Jobs, aber nicht die, die für unseren Kunden wirklich wichtig waren. Als wir uns die Job-Konfiguration anschauten, bekamen wir dies zu sehen (siehe Listing 9).

Die Abfrage der Snapshot-Datenbank ergab das in Listing 10 abgebildete Ergebnis.

Kein Wunder also, dass die Jobs auf der Standby-Datenbank nicht mehr sichtbar waren. Dafür sahen wir zwei Lösungswege:

- Wir ändern die Job-Definition auf der Primary-Datenbank (siehe Listing 11).
- Wir passen die Job-Definitionen auf der Snapshot-Datenbank an.

Option 2 war natürlich die bessere Variante. Naja, jedenfalls, wenn es um 20 Definitionen gegangen wäre. Doch bei mehr als 8.000? Glücklicherweise kam uns ein simples SQL-Skript zu Hilfe. Mit ihm konnten wir 8.000 Definitionen so behandeln, als wären es 20 (siehe Listing 12).

Ende gut, alles gut: Die Jobs waren wieder sichtbar und ausführbar.

Zwei kleine Tipps zum Schluss

Das Umschichten der Standby- zur Primary-Datenbank haben wir übrigens über das Command-Line Interface von Data Guard ausgeführt. Dazu zwei kleine Tipps:

- ✓ Die Verbindung immer via TNS herstellen, also über User und Passwort, und nicht lokal mit OS-Authentifizierung.
- ✓ Bitte immer „Screen“ benutzen.

Fazit

Rückblickend kann ich sagen, dass ich keine bessere Lösung für die Datenmigration innerhalb der Public Cloud empfehlen kann als Data Guard. Die Technologie bietet eine Vielfalt von Möglichkeiten, vor allem wenn es um Tests im Vorfeld der Migration geht. Die Snapshot-Standby-Datenbank, die man als normale Read/Write-Datenbank benutzen kann, ist unschlagbar. Dennoch gibt es Stolperfallen, auf die es aufzupassen gilt. Das mussten wir zum Beispiel bei den Scheduling-Jobs feststellen, die in ihrer Definition die Datenbankrolle beinhalteten.

Den Übergang zwischen physikalischer Standby-Funktion und Snapshot-Standby-Funktion haben wir bei unserer 300 Terabyte mächtigen Datenbank schnell hinbekommen: 2 Tage im Snapshot-Modus, 30 Minuten für den Switch zur physikalischen Datenbank.

Aber was sagte unser Kunde dazu?

Für den Online-Händler hatte sich der Umstieg auf die Oracle Exadata X9M gelohnt. Die Cyber Week wurde ein voller Erfolg. Nicht zuletzt dank der sehr ausgeprägten Technologie und der Flexibilität der X9M, wenn es um die Skalierbarkeit von Ressourcen geht.

Quellen

1. <https://docs.oracle.com/en-us/iaas/exadatacloud/exacs/using-data-guard-with-exacc.html#GUID-6EBC4D6A-C58B-4721-B756-F22FC6819A45> - Data Guard
2. <https://docs.oracle.com/en-us/iaas/exadatacloud/exacs/ecs-using-dbaascli.html#GUID-232E5D26-420A-4B7A-BE30-79929C95314F> - dbascli
3. <https://docs.oracle.com/en-us/iaas/exadatacloud/exacs/ecs-create-instance.html#GUID-5E520C3B-F141-48C4-9AC2-0FE0AE8055FD> - Exadata Infrastructure & VM Cluster
4. OCI: How To Configure & Manage Database Backups On OCI EXACS DB System (Doc ID 2708469.1) - OCI Object Storage
5. Creating a Physical Standby database using RMAN restore database from service (Doc ID 2283978.1) - Restore/Recovery from service
6. Exadata Cloud Service Software Versions (Doc ID 2333222.1) - Exadata-Versionen im OCI



Andrzej Rydzanicz
andrzej.rydzanicz@opitz-consulting.com



Schön, dass der Nachbar nicht stört! – Teil 2

Norman Sibbing, Oracle Deutschland

In der letzten Ausgabe des Red Stack Magazin (01/2023) haben wir mit der Isolation von Pluggable-Databases begonnen. Die dort beschriebenen Funktionalitäten, Lockdown-Profile und PATH_PREFIX, waren aber nur die ersten Schritte in Richtung einer umfangreichen Isolation von Pluggable-Databases. Darauf aufbauend, werden wir in diesem Artikel weitere Isolationsmöglichkeiten kennenlernen, die die Bereiche Betriebssystem-Zugriffe, Keystores und Netzwerk adressieren. Gehen wir nun die nächsten Schritte und starten mit dem Thema „Zugriff auf das Betriebssystem“.

Betriebssystem-Isolierung

Das Feature PATH_PREFIX dient dazu, den Zugriff auf das Filesystem unter kontrollierten Bedingungen zuzulassen. Bei der Verwendung von External Procedure Calls, External Table Pre-Proces-

sing, External Jobs oder der JVM in der Datenbank wird neben dem Zugriff auf das Filesystem ein Betriebssystem-Benutzer benötigt, der die notwendigen Betriebssystem-Ressourcen zur Verfügung stellt. Im Standard wird dazu der Oracle-Software-Owner verwendet. Auf-

grund seiner hohen Privilegien muss die Verwendung dieser Datenbank-Funktionalitäten als hohes Risiko eingestuft und gegebenenfalls untersagt werden. Oracle hat jetzt mit dem Pluggable-Database-Feature PDB_OS_CREDENTIAL die Möglichkeit geschaffen, diese Funktionalitäten

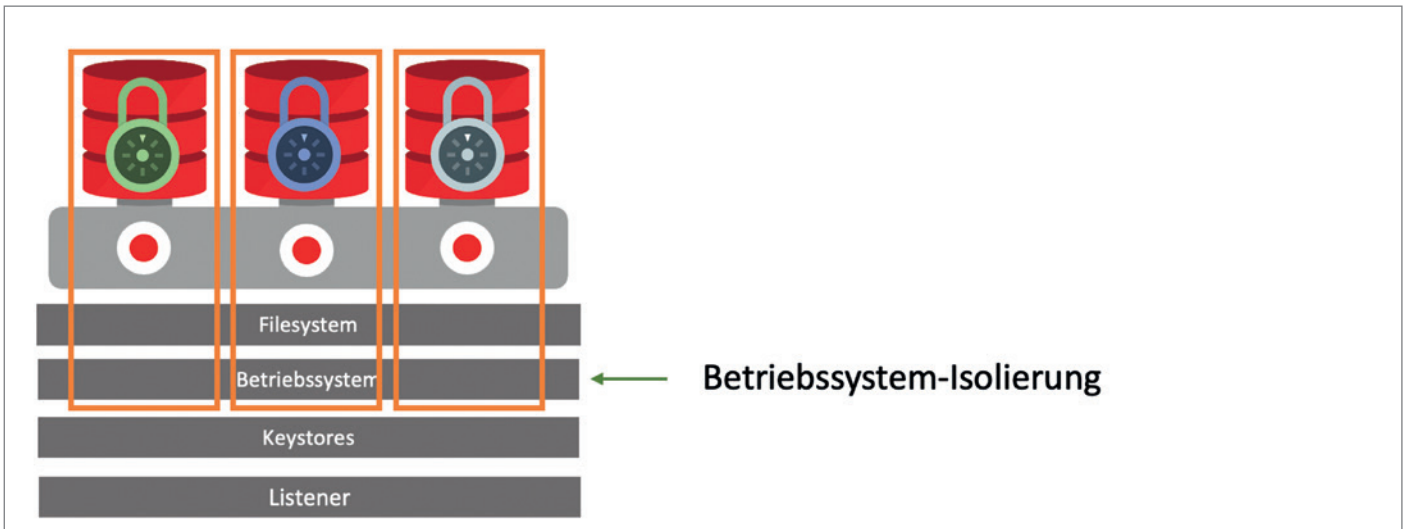


Abbildung 1: Betriebssystem-Isolierung (Quelle: Norman Sibbing)

```
[oracle@dbsec-lab ~]$ id pdbosuser
uid=54322(pdbosuser) gid=54331(pdbosuser) groups=54331(pdbosuser)
[oracle@dbsec-lab ~]$ █
```

Listing 1: Stellvertretender Betriebssystem-Benutzer für PDBs

```
SQL> begin
  2 DBMS_CREDENTIAL.CREATE_CREDENTIAL(credential_name => 'PDB_OS_USER',
  3                                     username => 'pdbosuser',
  4                                     password => 'VVO4epgFU#3L0!eFhXvAic');
  5 end;
  6 /
```

PL/SQL procedure successfully completed.

Listing 2: Erstellung einer Credential

```
SQL> ALTER SYSTEM SET PDB_OS_CREDENTIAL = PDB_OS_USER SCOPE=SPFILE;
ALTER SYSTEM SET PDB_OS_CREDENTIAL = PDB_OS_USER SCOPE=SPFILE
*
ERROR at line 1:
ORA-32017: failure in updating SPFILE
ORA-65046: operation not allowed from outside a pluggable database
```

Listing 3: Globales Setzen des Parameters PDB_OS_CREDENTIAL

litäten weiterhin zu verwenden, aber mit einem wesentlich geringeren Risiko (siehe Abbildung 1).

Hierzu werden den Pluggable-Databases stellvertretende Betriebssystem-Benutzer zur Verfügung gestellt. Diese Benutzer, beziehungsweise nur deren Anmeldeinformationen (Credentials), werden durch das Datenbank-Package DBMS_CREDENTIAL den Datenbank-Funktionalitäten bereitgestellt. Das

Privileg CREATE CREDENTIAL ist zum Erstellen der Credentials notwendig. Die Betriebssystem-Benutzer selbst müssen von einem Betriebssystem-Administrator erstellt werden und sollten nur die Privilegien besitzen, die zur Ausführung der Aufgabe notwendig sind (siehe Listing 1).

Als Betreiber einer Multitenant-Umgebung sollte man den Pluggable-Databases zumindest einen stellvertretenden Betriebssystem-Benutzer zur Verfügung

stellen. Dessen Credentials können dann global für alle Pluggable-Databases in der CDB vorgeben werden (siehe Listing 2).

Ähnlich wie bei den Lockdown-Profilen kann PDB_OS_CREDENTIAL sowohl als globale Einstellung in der Container-Database als auch je Pluggable-Database angegeben werden. In der Container-Database kann der Parameter nicht durch einen ALTER SYSTEM-Befehl gesetzt werden (siehe Listing 3).

Hier muss leider der etwas umständliche Weg über ein PFILE gegangen werden. Das wiederum setzt voraus, dass die Datenbank durchgestartet werden muss (siehe Listing 4 und 5).

Das Setzen von PDB_OS_CREDENTIAL in der Pluggable-Datenbank überschreibt die globale Einstellung der Container-Datenbank. Dies kann nur durch einen COMMON User mit dem ALTER SYSTEM-Privileg durchgeführt werden. Auf diese Weise ist es möglich, dass bei Bedarf jede Pluggable-Datenbank einen eigenen stellvertretenden Betriebssystem-Benutzer hat. Die so angegebenen Credentials werden immer dann verwendet, wenn entsprechende Datenbank-Funktionalitäten einen Betriebssystem-Benutzer benötigen. Manche der Datenbank-Funktionalitäten erlauben die Nutzung verschiedener OS Credentials, unabhängig von der Standardeinstellung. So erlaubt zum Beispiel das Kommando CREATE LIBRARY das Setzen verschiedener Credentials pro Library-Object. Damit ist es möglich, External Procedure Calls innerhalb einer PDB mit unterschiedlichen Betriebssystem-Benutzern auszuführen. Bei der Verwendung von Libraries ist zusätzlich darauf zu achten, dass Verzeichnis-Objekte – DIRECTORY OBJECTS – verwendet werden müssen, wenn es sich um eine Umgebung handelt, bei der das Feature PATH_PREFIX aktiv ist. Beispiel: External Procedure Calls. Um einen External Procedure Call auszuführen, wird eine entsprechende Library im Betriebssystem benötigt. Die in diesem Beispiel verwendete C-Library (shell.so) macht nichts anderes, als Betriebssystem-Befehle 1:1 auszuführen. Sie dient somit also als Schnittstelle zwischen PL/SQL und dem Betriebssystem.

Damit Libraries generell genutzt werden können, müssen sie in der Konfigurationsdatei „extproc.ora“ angegeben werden. Diese Datei befindet sich im Verzeichnis „\$ORACLE_HOME/hs/admin“. Dort können einzelne Libraries oder komplette Verzeichnisse, in denen sich mehrere Libraries befinden, freigegeben werden.

In meiner „extproc.ora“ gebe ich nur die Library „shell.so“ frei. Da ich eine Multitenant-Umgebung mit gesetztem PATH_PREFIX verwende, muss ich darauf achten, dass die entsprechende Pluggable-Datenbank Zugriff auf das Verzeichnis

hat, in dem die Library liegt. Im aktuellen Beispiel also mindestens unter:

```
/u02/pdbs/DC12E5633C487130E-053CE6EC80AF84D/
```

Hier lege ich jetzt ein Unterverzeichnis mit dem Namen „extproc_lib“ an und kopiere danach die Library „shell.so“ dorthin (siehe Listing 6).

Als Nächstes benötigen wir ein Verzeichnis-Objekt in der Pluggable-Datenbank, das auf das Verzeichnis der Library verweist (siehe Listing 7).

Da PDB_OS_CREDENTIAL global gesetzt wurde, kann das Verzeichnis-Objekt in der Pluggable-Datenbank ohne explizite Angabe einer CREDENTIAL erstellt werden. Benötigt werden lediglich der Dateiname der Library und das entsprechende Verzeichnis-Objekt (siehe Listing 8).

Zuletzt noch die PL/SQL-Prozedur, die die externe Library verwendet (siehe Listing 9).

Zur Ausführung der Prozedur wird jetzt der Betriebssystem-Benutzer verwendet, dessen Credential über den Parameter PDB_OS_CREDENTIAL angegeben wurde (siehe Listing 10).

Fazit

PDB_OS_CREDENTIAL erleichtern das Umsetzen von Least-Privilege bei Interaktionen zwischen unterschiedlichen Datenbank-Funktionalitäten und dem Betriebssystem. Sie minimieren dadurch das Missbrauchsrisiko entsprechender Datenbank-Funktionalitäten enorm.

Keystores per Pluggable

Keystores werden zur sicheren Speicherung von Krypto-Schlüsseln und Geheimnissen jeglicher Art verwendet. Die Oracle-Datenbank verwendet Keystores zur Speicherung des externen TDE-Schlüssels, zur Speicherung von Passwörtern, Zertifikaten und anderen „Geheimnissen“. Als Keystore wird das Oracle Wallet verwendet, eine Datei im PKCS#12-Format. Bisher war es so, dass es nur zwei Keystores, einen Keystore für TDE und einen für Zertifikate und Passwörter, pro Datenbank gab. Diese zwei Keystores wurden mit den SQLNET.ORA-

Parametern WALLET_LOCATION und ENCRYPTION_WALLET_LOCATION auf dem Datenbank-Server angegeben. Problematisch wird es, wenn sich mehrere Datenbanken ein Oracle Home teilen. Hier muss sehr sorgfältig mit Umgebungsparametern gearbeitet werden. In einer Multitenant-Umgebung ist diese Vorgehensweise nicht denkbar. Das widerspricht der strikten Trennung zwischen den PDBs untereinander, aber auch zwischen der CDB und den PDBs. Mit der Oracle-Datenbank 18c wird diese Problematik gelöst (siehe Abbildung 2).

Jede PDB kann jetzt eigene Keystores für Transparent Data Encryption (TDE), Enterprise User Security (EUS), Centralized User Management (CMU) und Transport Layer Security (TLS) verwenden. Dies wird durch das Setzen des Datenbank-Parameters WALLET_ROOT beziehungsweise CMU_WALLET ermöglicht. Der Parameter WALLET_ROOT ersetzt den SQLNET.ORA-Parameter SQLNET.ENCRYPTION_WALLET_LOCATION, der ab 18c ebenfalls abgekündigt wurde, aber weiterhin noch eine gewisse Zeit lang funktioniert. Ist der Parameter nicht gesetzt, bezieht die Datenbank das TDE-Wallet-Verzeichnis weiterhin über den SQLNET.ENCRYPTION_WALLET_LOCATION-Parameter aus der SQLNET.ORA. Bei der Verwendung von EUS und TLS ersetzt dieser Parameter ebenfalls den SQLNET.ORA-Parameter WALLET_LOCATION. WALLET_ROOT muss in der CDB gesetzt werden. Der Parameter beschreibt das Keystore-Basis-Verzeichnis aller sich in der CDB befindlichen PDBs. Um eine Zuordnung der Keystores zu den PDBs zu erhalten, werden unterhalb des WALLET_ROOT-Verzeichnisses weitere Verzeichnisse pro PDB angelegt. Der Verzeichnisname entspricht der eindeutigen GUID der PDB. Letztendlich können in diesem Verzeichnis weitere Unterverzeichnisse für die eigentlichen Keystores von TDE, EUS, OKV und TLS angelegt werden.

Verzeichnisstruktur für Wallets (CDB und PDBs)

```
CDB: <WALLET_ROOT>/{tde, okv, tde_seps, eus, tls}
PDBs: <WALLET_ROOT>/<PDB_GUID>/{tde, okv, tde_seps, eus, tls}
```

Die Ausnahme ist Centralized User Management (CMU). CMU ermöglicht die


```

SQL> show parameter pdb_os_credential

NAME                                TYPE                                VALUE
-----                                -
pdb_os_credential                    string
SQL>
SQL> CREATE PFILE ='/tmp/pfile.txt' from SPFILE;

File created.

SQL> !echo "*.pdb_os_credential=PDB_OS_USER" >> /tmp/pfile.txt

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
SQL> CREATE SPFILE FROM PFILE='/tmp/pfile.txt';

File created.

SQL> █

```

Listing 4: Setzen des Parameters PDB_OS_CREDENTIAL via pfile

```

SQL> startup
ORACLE instance started.

Total System Global Area 1795159792 bytes
Fixed Size                  9135856 bytes
Variable Size              452984832 bytes
Database Buffers          1325400064 bytes
Redo Buffers               7639040 bytes
Database mounted.
Database opened.
SQL>
SQL> show parameter pdb_os_credential

NAME                                TYPE                                VALUE
-----                                -
pdb_os_credential                    string                                PDB_OS_USER
SQL> █

```

Listing 5: Neustart der Datenbank

```

[oracle@dbsec-lab ~]$ ls -la /u02/dbfs/DC12E5633C487130E053CE6EC80AF84D/extproc_lib
total 8
drwxr-xr-x. 2 oracle oinstall  22 Jan 13 14:39 .
drwxr-x---. 3 oracle oinstall  25 Jan 13 14:39 ..
-rw-r-----. 1 oracle oinstall 5768 Jan 13 14:39 shell.so
[oracle@dbsec-lab ~]$ █

```

Listing 6: Speicherort der C-Library

```

SQL> CREATE OR REPLACE DIRECTORY LIB_DIR as 'extproc_lib';

Directory created.

SQL> █

```

Listing 7: Erstellung des Directory Object für die Library

```

SQL> CREATE OR REPLACE LIBRARY shell_lib IS 'shell.so' IN LIB_DIR;
  2 /

Library created.

```

Listing 8: Erstellung des Library-Objects

```
SQL> CREATE OR REPLACE PROCEDURE shell (cmd IN CHAR)
  2   AS EXTERNAL NAME "sh"
  3   LIBRARY shell_lib
  4   LANGUAGE C PARAMETERS (cmd STRING);
  5 /
```

Procedure created.

Listing 9: Erstellung der PL/SQL-Prozedur für den External Procedure Call

```
SQL> exec shell('/bin/touch /tmp/extproc-connect-$(/bin/date +%Y-%m-%d-%H-%M-%S).txt');
PL/SQL procedure successfully completed.
SQL> !ls -rtla /tmp/extproc-connect-*
-rw-rw-rw-. 1 pdbosuser pdbosuser 0 Jan 13 14:52 /tmp/extproc-connect-2023-01-13-14-52-07.txt
```

Listing 10: Ausführung des External Procedure Calls

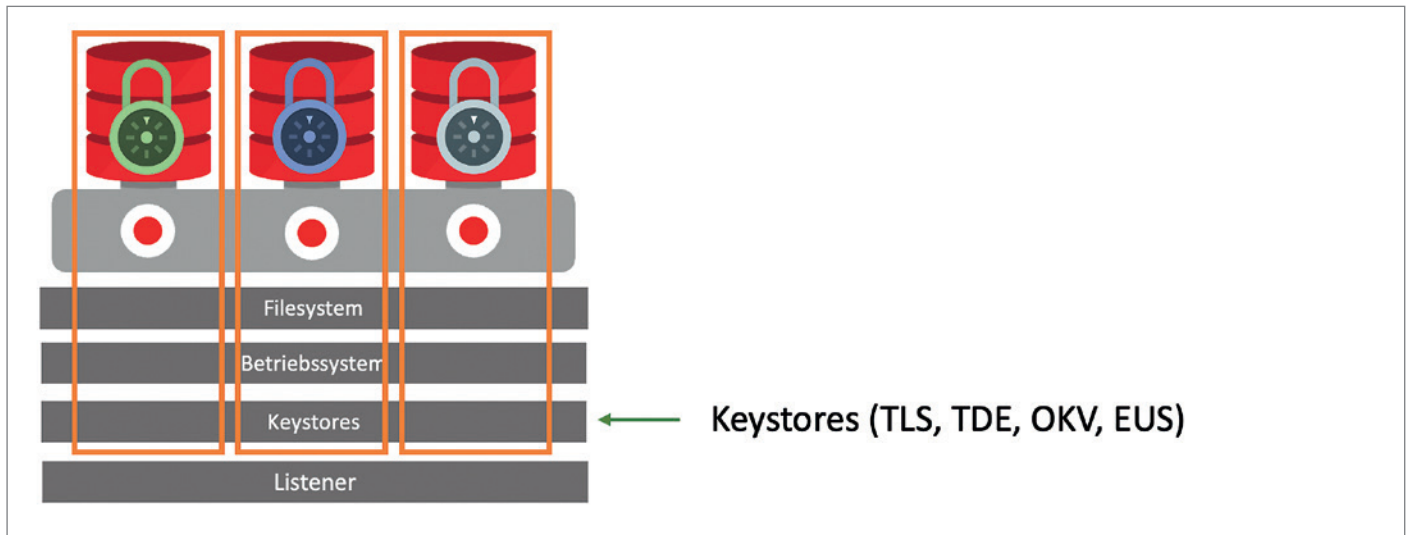


Abbildung 2: Keystores per Pluggable-Database (Quelle: Norman Sibbing)

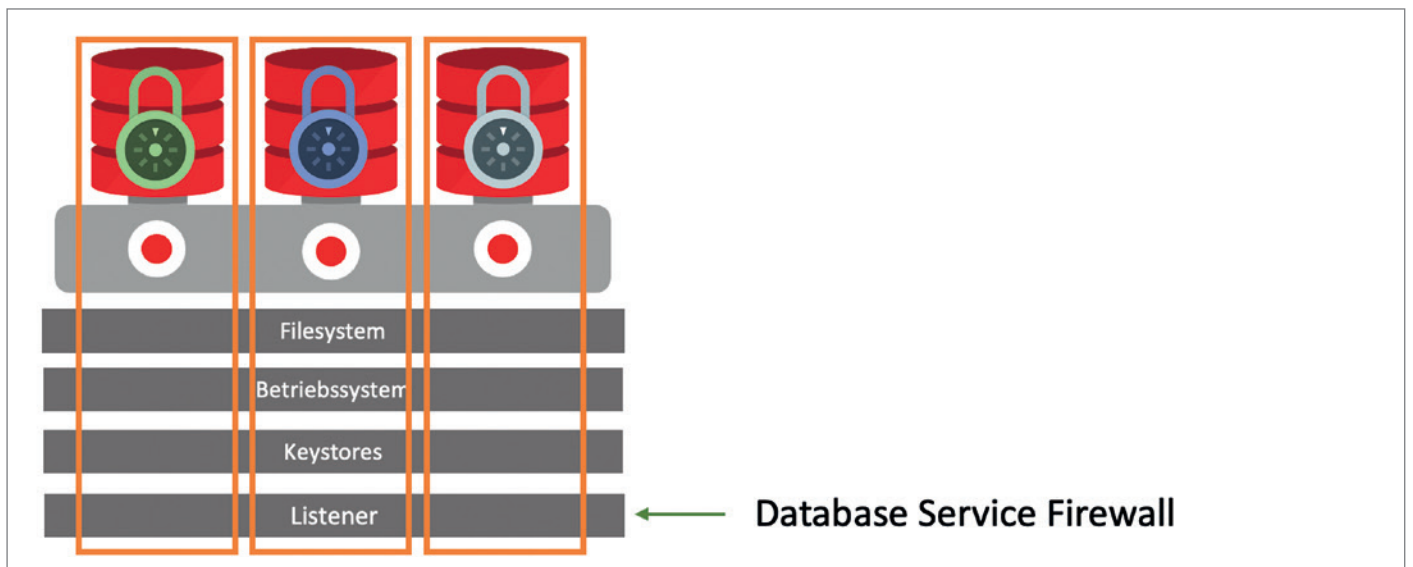


Abbildung 3: Database Service Firewall (Quelle: Norman Sibbing)

Verwaltung von Datenbank-Benutzern direkt in einem Microsoft Active Directory. Sollte CMU verwendet werden, muss der Pluggable-Database Parameter `CMU_WALLET` verwendet werden. Das für die Verwendung von CMU notwendige Wallet muss nicht unterhalb des `WALLET_ROOT` gespeichert werden.

Der Parameter `WALLET_ROOT` steht auch in der klassischen Oracle-Datenbank-Architektur – `NON-CDB` – zur Verfügung. Im Standard ist der Parameter nicht gesetzt und es handelt sich um einen statischen Parameter. Demzufolge muss die Datenbank nach dem Setzen neu gestartet werden.

Fazit

Der Parameter ermöglicht die Verwendung individueller Keystores pro Pluggable-Database. Zusätzlich hilft er bei der Zuordnung zwischen den Keystores und den Pluggable-Databases. Das ist sehr hilfreich bei Aktionen wie `PDB Relocate`, `Klonen` oder `Plug-In/Plug-Out`. Bei diesen Aktivitäten müssen die Keystores mit den Pluggable-Databases „wandern“. Mit den Funktionalitäten `WALLET_ROOT` und `PATH_PREFIX` hat man immer den Überblick darüber, welche Keystores, Files und Libraries zu einer Pluggable-Database gehören. Damit hat man sozusagen immer alles beisammen.

Database Service Firewall (Service-Level-ACL)

Bitte nicht verwechseln mit dem Produkt „Oracle Database Firewall“ von „Audit Vault and Database Firewall“. Das Feature Database Service Firewall ist Bestandteil der Multitenant-Architektur und steht in jeder Oracle Database Edition zur Verfügung (siehe *Abbildung 3*).

Das Feature Database Service Firewall für Pluggable-Databases ermöglicht es, für jede Pluggable-Database eine eigene Zugriffskontrollliste (Access Control List) auf Basis von IP-Adressen zu konfigurieren. Diese Zugriffskontrolllisten werden in der Container-Database konfiguriert und vom Oracle Listener umgesetzt. Zur Konfiguration wird das Package `DBMS_SFW_ACL_ADMIN` zur Verfügung gestellt. Das Package bietet Prozeduren zur Er-

stellung und Verwaltung der Database-Service-Firewall-Zugriffskontrollliste. Die Zugriffskontrollliste kann aus einem Hostnamen, einer IPv4-Adresse oder einer IPv6-Adresse bestehen. Platzhalter „*“ für IPv4 und das CIDR-Format sind ebenfalls zulässige Werte. Diese Werte entscheiden, welche Clients beziehungsweise Hosts Zugriff auf bestimmte Datenbank-Dienste haben. Der lokale Oracle Listener ist dabei der Enforcement-Point, validiert alle eingehenden Verbindungen auf Basis der Zugriffskontrollliste und handelt entsprechend.

In einer Multitenant-Umgebung bietet der Listener grundsätzlich alle in der Container-Database gestarteten Pluggable-Databases als Datenbank-Dienst an. Jeder Client hat somit die Möglichkeit, jede dieser Pluggable-Databases und die Container-Database direkt anzusprechen. Das wiederum stellt ein hohes Risiko für Denial-of-Service und Brute-Force-Attacken dar. Ziel ist es jetzt, den Zugriff auf einzelne Pluggable-Databases oder auf die Container-Datenbank nur von bestimmten IP-Adressen aus zuzulassen. Dazu ist eine klassische Firewall nicht in der Lage, da sie schlichtweg Database-Services nicht kennt. Genau diese Lücke schließt die Database Service Firewall. Die optimale Voraussetzung dafür ist die Verwendung von zwei unterschiedlichen Netzwerken auf dem Datenbank-Server: einem Admin-Netzwerk und einem Client-Netzwerk. Ist das nicht möglich, sollten mindestens zwei Listener-Endpunkte auf unterschiedliche Ports konfiguriert werden. Ein Port für den Admin-Zugriff und ein anderer Port für den Client-Zugriff. Mit dieser Konfiguration kann dann eine klassische Web Application Firewall zumindest eine Trennung zwischen Admin- und Client-Netzwerk umsetzen (siehe *Listing 11*).

Die Database Service Firewall kann jetzt für das Client-Netzwerk aktiviert werden. Hierzu wurden für den Oracle Listener zwei neue Parameter eingeführt. Der Parameter `FIREWALL` aktiviert die Database Service Firewall für einen bestimmten Listener-Endpunkt und der Parameter `LOCAL_REGISTRATION_ADDRESS_<listener_name>` ist notwendig, damit der Listener das dynamische Registrieren der Zugriffskontrolllisten zulässt (siehe *Listing 12*).

Nach dem Neustart des Listener ist die Database Service Firewall für den End-

punkt mit dem Port 1521 (Client-Netzwerk) aktiviert (siehe *Listing 13*).

Ab jetzt werden für Verbindungen aus dem Client-Netzwerk entsprechende Database-Service-Firewall-Regeln benötigt. Versucht ein Client, eine Verbindung zu einer Pluggable-Database aus dem Client-Netzwerk heraus ohne eine solche passende Regel aufzubauen, erhält er eine Fehlermeldung (siehe *Listing 14*).

Die Verbindung des Clients wurde vom lokalen Listener abgewiesen, weil keine entsprechende Zugriffskontrollliste existiert. Die Zugriffskontrollliste wird in der Container-Database erstellt und verwaltet. Dazu existiert in der Container-Database ein Datenbank-Benutzer namens `DBSFWUSER`. Der Name steht für **DatabaseServiceFireWallUser**. Er besitzt alle notwendigen Berechtigungen, um die Zugriffskontrolllisten für die unterschiedlichen Pluggable-Databases zu erstellen und zu verwalten. Dieser Benutzer ist standardmäßig gesperrt und muss entsperrt werden (siehe *Listing 15*).

Über diesen Benutzer verwenden wir jetzt das `DBMS_SFW_ACL_ADMIN`-Package, um die Zugriffskontrollliste zu erstellen. `DBMS_SFW_ACL_ADMIN` beinhaltet aktuell sieben Prozeduren. Im Wesentlichen sind es Prozeduren zum Hinzufügen, zum Entfernen und zum „Committed“ von Regeln. Damit jetzt der Client auf seine Pluggable-Database (z.B. `PDB1`) zugreifen kann, müssen wir wissen, aus welchem Netzwerk er kommt. In diesem Beispiel ist es ein privates Netzwerk (192.168.20.0/24).

Für dieses Netzwerk muss nun eine Zugriffskontrollliste erstellt werden, damit Clients aus diesem Netzwerk auf die Pluggable-Database `PDB1` zugreifen können (siehe *Listing 16*).

Damit wurde die Zugriffskontrollliste erstellt. Aktiv wird sie aber erst, nachdem sie „Committed“ wurde (siehe *Listing 17*).

Um zu überprüfen, ob die Zugriffskontrollliste aktiviert ist, bietet der Oracle Listener ein neues Kommando an. Mit dem Kommando `!snrctl servacl` (`servacl` = `service-access-control-list`) lässt sich abfragen, aus welchem Netzwerk die entsprechende Pluggable-Database erreichbar ist (siehe *Listing 18*).

Nachdem die Zugriffskontrollliste aktiviert wurde, können entsprechende Clients aus dem freigegebenen Netzwerk

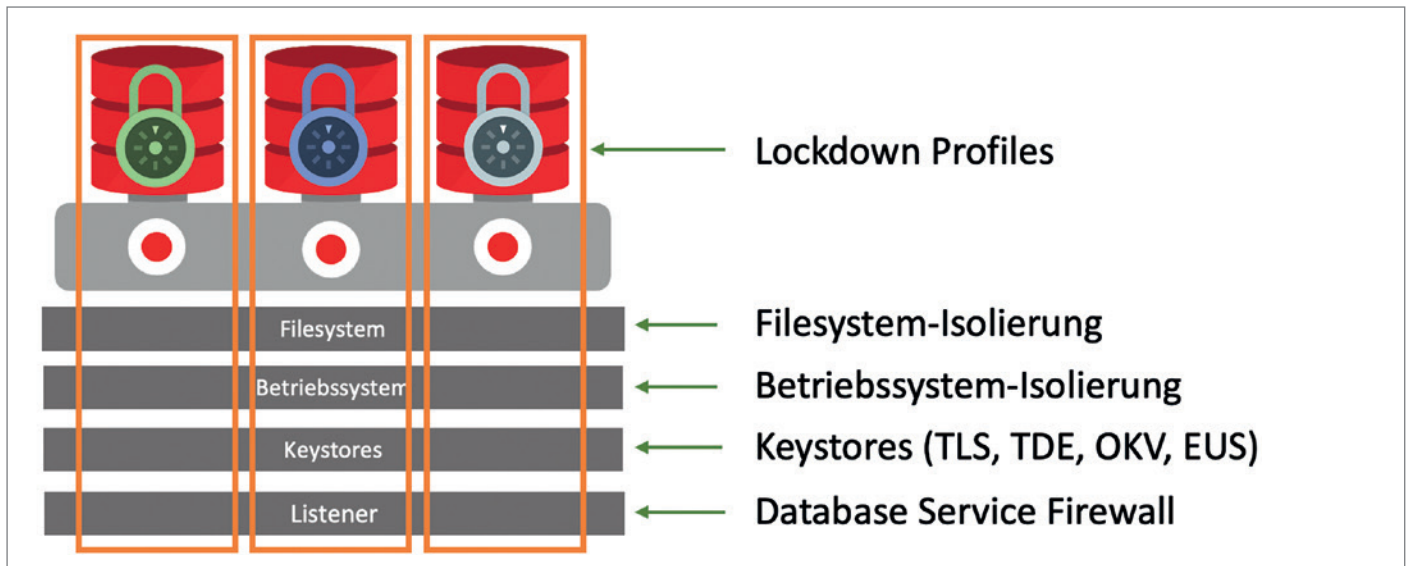


Abbildung 4: Maßnahmen zum Schutz von Pluggable-Databases (Quelle: Norman Sibbing)

```

Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=dbsec-lab.ns01sub02.ns01.oraclevcn.com)(PORT=1621))) ADMIN-Netz
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=dbsec-lab.ns01sub02.ns01.oraclevcn.com)(PORT=1521))) CLIENT-Netz
    
```

Listing 11: Listener mit zwei Endpunkten

```

LOCAL_REGISTRATION_ADDRESS_LISTENER=ON

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = dbsec-lab)(PORT = 1621))
      (ADDRESS = (PROTOCOL = TCP)(HOST = dbsec-lab)(PORT = 1521)(FIREWALL = ON))
    )
  )
    
```

Listing 12: listener.ora

```

Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=dbsec-lab.ns01sub02.ns01.oraclevcn.com)(PORT=1621)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=dbsec-lab.ns01sub02.ns01.oraclevcn.com)(PORT=1521)(FIREWALL=ON)))
    
```

Listing 13: Status Listener Endpoints

```

NSIBBING>sqlplus pdbadmin/oracle123@10.0.2.157:1521/pdb1

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jan 13 16:06:29 2023
Version 19.14.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

ERROR:
ORA-12506: TNS:listener rejected connection based on service ACL filtering
    
```

Listing 14: Verbindung aus dem Client-Netz ohne Firewall-Regeln

```
SQL> alter user dbsfwuser identified by "Oracle123" account unlock;

User altered.
```

Listing 15: DBSFWUSER entsperren

```
SQL> show user
USER is "DBSFWUSER"
SQL>
SQL> begin
  2  DBMS_SFW_ACL_ADMIN.IP_ADD_PDB_ACE (p_pdb_name => 'PDB1',
  3                                     P_HOST=> '192.168.20.0/24');
  4  end;
  5  /

PL/SQL procedure successfully completed.
```

Listing 16: Erstellung der Zugriffskontrollliste

```
SQL> exec dbms_sfw_acl_admin.commit_acl;

PL/SQL procedure successfully completed.
```

Listing 17: Aktivierung der Zugriffskontrollliste

```
Service "pdb1" has 1 instance(s).
  Instance "cdb1": 1 failure(s), status READY...
  ACL(s):
    IP_LIST has 1 item(s)
    192.168.20.0/24
```

Listing 18: Isnrctl servacl

```
NSIBBING>sqlplus pdbadmin/Oracle123@10.0.2.157:1521/pdb1
SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jan 13 16:19:58 2023
Version 19.14.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Last Successful login time: Fri Jan 13 2023 15:47:16 +01:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.16.0.0.0

SQL>
```

Listing 19: Verbindung aus dem Client-Netz mit Firewall-Regeln

```
[oracle@dbsec-lab ~]$ sqlplus pdbadmin/Oracle123@10.0.2.157:1521/pdb1
SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jan 13 15:23:51 2023
Version 19.16.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

ERROR:
ORA-12506: TNS:listener rejected connection based on service ACL filtering
```

Listing 20: Verbindung aus anderen Netzwerken

auf die Pluggable-Database PDB1 zugreifen (siehe Listing 19).

Verbindungen aus anderen Netzwerken werden weiterhin abgewiesen (siehe Listing 20).

Fazit

Mit der Database Service Firewall ist es möglich, den Zugriff auf eine Pluggable-Database in Abhängigkeit vom Netzwerk zu steuern. Dadurch kann das Risiko einer Denial-of-Service- oder einer Brute-Force-Attacke auf die Pluggable-Databases erheblich reduziert werden. Alternative Technologien, mit denen dies auch möglich wäre, sind der Connection Manager (Enterprise Edition Feature) und die Database Firewall (Audit Vault and Database Firewall).

Zusammengefasst

Insgesamt stellen die hier beschriebenen Sicherheitsmaßnahmen gute Möglichkeiten zur Isolierung von Pluggable-Databases zur Verfügung (siehe Abbildung 4).

Allein mit den Datenbank-Security-Features, die in der Standard-Edition der Oracle-Datenbank zur Verfügung stehen, kann schon sehr viel Sicherheit umgesetzt und das Risiko eines Missbrauchs reduziert werden. Darunter fallen Funktionalitäten wie die Kommunikations-Verschlüsselung, Unified Auditing, starke Authentifizierung, Gradual Database Password Rollover und die hier beschriebenen Sicherheitsfunktionalitäten der Multitenant-Architektur. Sollte das nicht ausreichen, bietet Oracle weitere Sicherheits-Funktionalitäten mit der Oracle Enterprise Edition an.



Norman Sibbing
Norman.Sibbing@oracle.com



Ein Trio zum Early-Bird-Preis: Kräftig sparen bei den nächsten DOAG-Konferenzen

Marcos López

In der ersten Jahreshälfte 2023 finden noch drei DOAG-Konferenzen statt, für die ein attraktiver Frühbucherrabatt angeboten wird.

APEX connect 2023

Der bis zum 30. März 2023 laufende Frühbucherpreis bietet eine satte Ersparnis von über 250 EUR auf das reguläre Zwei-Tages-Ticket der APEX-Konferenz am 3. und 4. Mai in Berlin, inklusive Community-Abend und Verpflegung an allen Konferenztagen. <https://shop.doag.org/shop/prd.509.apex-connect--2-tage/>

DOAG 2023 Datenbank mit Exaday

Am 24. und 25. Mai dreht sich in Düsseldorf alles um die Datenbankthemen Oracle Database, PostgreSQL und MySQL sowie die Engineered-Systems-Schwerpunkte Exadata und Oracle Database Appliance (ODA). Das reguläre Ticket ist noch bis zum 19. April 2023 anstatt für 950 EUR für 800 EUR zu haben. <https://shop.doag.org/shop/prd.593.doag-2023-datenbank-konferenzpass/>

CloudLand 2023

Die Early-Bird-Aktion des Cloud Native Festivals, das vom 20. bis 23. Juni 2023 wieder im Phantasialand stattfindet, ermöglicht einen rabattierten Ticketkauf für drei verschiedene Ticketvarianten. Der Frühbucherpreis für die CloudLand ist noch bis zum 9. Mai 2023 verfügbar. <https://shop.doag.org/events/cloudland/shop/>

PremiumCard

An dieser Stelle möchten wir noch auf ein besonderes Angebot hinweisen: die DOAG PremiumCard. Sie ermöglicht eine Event-Flatrate für die meisten DOAG-Veranstaltungen. Alle weiteren Informationen dazu finden Sie hier: <https://www.doag.org/de/verein-mitgliedschaft/mitgliedschaft/#c7321>



Große Datenbanken in der Standard Edition – Backup-Ideen

Thomas Beetz, merlin.zwo Info Design

Es werden die Punkte Kompression und Parallelverarbeitung angesprochen. Komprimierung einmal durch Zusatzprogramme und auf Dateisystemebene. Für rman ist die Frage, ob man eine Parallelverarbeitung erreichen kann. Stichworte sind pigz, vdo, ntfs und rman pipe.

Messergebnisse

Wir betrachten eine Oracle-12c-SE-RAC-Datenbank mit einer Größe von 380GB. Die Imagekopie liegt auf einer zweiten ASM-Disk und die Backups auf einer CIFS-Freigabe. Bei größeren Datenbanken benötigt ein Backup über einen Tag.

Auf dem RAC laufen fünf Datenbanken, zwei große und drei kleinere. Die Messungen wurden während des regulären Betriebs ausgeführt. Jeder RAC-Knoten verwendet eine CPU mit zehn Cores und aktivierten HT.

Anmerkungen:

- Bei Imagekopie Level 1 steht in SE kein Block Change Tracking zur Verfügung
- Imagekopie Level 1: die inkrementelle Sicherung liegt im ASM und nicht auf dem Backuplaufwerk
- Bei NTFS komprimiert ist die Anzeige „freier Platz“ irreführend
- Compressed Backupset parallel wurde manuell in mehreren Terminals gestartet
- Compressed Backupset skaliert mit Anzahl der eingesetzten rman-Prozesse, genügend freie Cores vorausgesetzt
- Bigfile Tablespace können in der EE dank dem „SECTION SIZE“-Parameter von mehreren Kanälen profitieren. Das kann in der SE nicht nachgebildet werden

pigz (Parallel Implementation of gzip)

pigz ist als Paket im Oracle Repository enthalten und kann einfach mit dnf/yum installiert werden. Die Parameter für gzip sind leider nicht eins zu eins übertragbar, daher ist ein Austausch dann etwas aufwendiger.

Der Parameter `-p` pigz begrenzt die Anzahl der Threads, ansonsten bildet die Anzahl der Cores das Limit.

Für datapump ist es eine Option, um Platz zu sparen, für rman ist es nicht sinnvoll. Eine Komprimierung, die direkt auf dem Dateisystem aufsetzt ist, ist vorteilhaft. Windows bietet die Möglichkeit, in NTFS-Volumen zu komprimieren, seit über zwanzig Jahren. Dieses Feature ist in ReFS nicht mehr verfügbar. Für Linux-Dateisysteme möchte ich auf [1] verweisen. Seit

Oracle Linux 7 bietet der Red-Hat-Kernel nun auch VDO an, im UEK wird es nicht unterstützt.

VDO (Virtual Data Optimizer)

VDO bietet neben der Möglichkeit zu komprimieren auch die Option Datenduplizierung an. Wobei Datenduplizierung im Gegensatz zur Komprimierung noch mehr Ressourcen benötigt. VDO unterstützt auch lvm und damit verschiedene Dateisysteme. Damit ist eine große Bandbreite von Einsatzmöglichkeiten verbunden von ISCSI über NFS, SAMBA bis hin zum bevorzugten Dateisystem (wie xfs, ext4) (siehe Abbildung 1). [2]

Die Möglichkeiten von lvm werden nicht eingeschränkt wie Vergrößerungen oder Snapshots. Zu Testzwecken kann eine VM mit Oracle Linux 8 installiert werden mit einem RHCK. Die UEK-Kernel unterstützen dies aktuell noch nicht (siehe Listing 1).

Es besteht auch die Möglichkeit, im lvm mit `lvcreate -type vdo` ein entsprechendes Volume zu erstellen. Eine weitere Möglichkeit ab Oracle Linux 8 ist die Cockpit Web Console [3].

Fazit

Die Komprimierung muss nicht auf dem Datenbankserver stattfinden, was vor allem bei einer Enterprise Edition nicht ganz uninteressant ist, da die teuren CPU-Lizenzen nicht für die Komprimierung des Backups verwendet werden. Das gilt zwar nicht für die Standard Edition, da werden die Datenbanken bei 16 aktiven Prozessen abgeregelt und oft sind weitere Cores verfügbar. Aber Server mit MCM (Multi Chip Module) hätten hier wieder ein Einsatzgebiet im Oracle-Umfeld als Storage. Die Komprimierung kann aber nicht einfach auf einem weiteren Backupmedium wie einem Bandlaufwerk oder einem Storage in der Cloud übernommen werden.

LSYNC (Live Syncing Daemon)

In der Regel findet man Skripte mit `rsync`, um Dateien und Verzeichnisse zu replizieren. Wie bei `rsync` müssen die

ssh-Keys ausgetauscht werden, wenn man mit verschiedenen Servern arbeitet. Die Vorteile sind eine zentrale Konfigurationsdatei und Einbindung in `linux systemd`. Sollte der Zielservers zeitweilig ausfallen, wird ein `rsync`-Skript ohne Absicherung gegen mehrfache Ausführungen schnell zu einem Problem (siehe Listing 2).

Es ist aber zu empfehlen, den `LSYNC`-CD-Prozess zu überwachen, ein regelmäßiger Neustart ist nicht sinnvoll.

PL/SQL-Pakete

Es sollen drei Pakete angesprochen werden, `DBMS_FILE_TRANSFER`, `DBMS_PIPE` und `DBMS_BACKUP_RESTORE`. Die ersten beiden Pakete gehören zu denen, die man schon mal gesehen hat. Das dritte findet man, wenn man einen Blick in das `rman`-Programm geworfen hat. Der Package Body ist gewrappt und die Dokumentation ist übersichtlich. Auch bei Oracle Support sind die Informationen begrenzt. Zu Data Pump bietet Oracle ein API `DBMS_DATAPUMP`, für `rman` würde ich mir auch eins wünschen.

DBMS_BACKUP_RESTORE

Das Package arbeitet nicht mit der `rman`-Konfiguration zusammen. Die Pfade und Dateinamen für Backups müssen manuell gesetzt werden (siehe Listing 3).

Mehr Informationen findet man in [4] und dem zugehörigen Git-Repository.

DBMS_FILE_TRANSFER

Ein kleines Paket mit nur drei Prozeduren:

- `COPY_FILE`
- `GET_FILE`
- `PUT_FILE`

Die Namen sprechen für sich, ein Überschreiben einer Datei ist nicht möglich. Wichtig: Der Zugriff auf ASM ist möglich. `GET_FILE` und `PUT_FILE` arbeiten auf einer Datenbank, die Prozedur `COPY_FILE` mit zwei Datenbanken via Database Link (siehe Listing 4).

Backup-Verfahren	Größe Backup (GB)	Dauer (Stunden)
COMPRESSED BACKUPSET	180	8:10
BACKUPSET	365	0:50
IMAGEKOPIE (LEVEL 0 NEU)	400	0:21
IMAGEKOPIE (LEVEL 1)	400 (0,25)	0:10+0:01
BACKUPSET (NTFS komprimiert)	180	3:40
COMPRESSED BACKUPSET (4 rman-Prozesse parallel)	180	1:40

Tabelle 1: Backup-Messungen

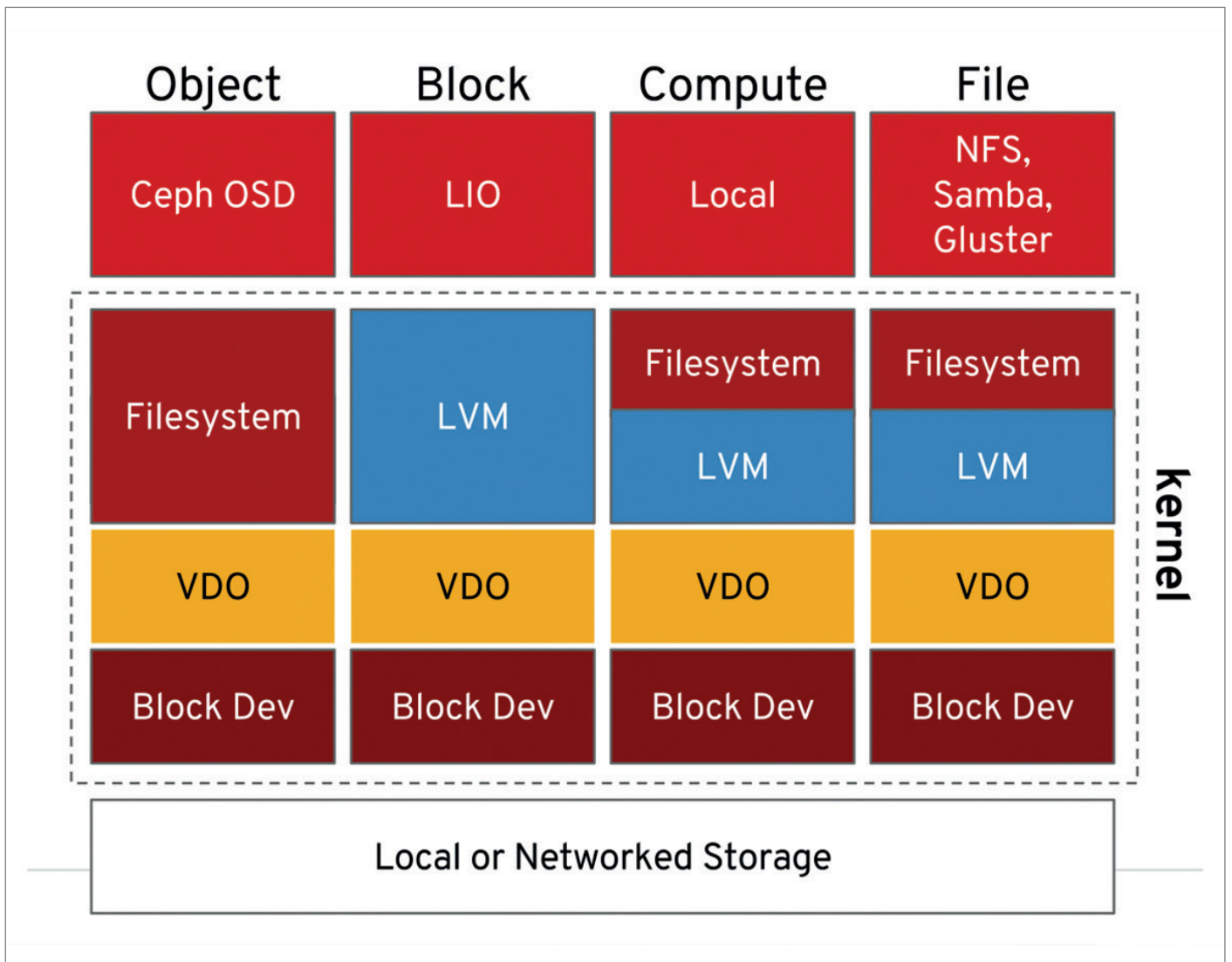


Abbildung 1: VDO Layer (© 2022 Red Hat, Inc)

Kommando	Zeit (s)	Größe (MB)
time gzip test1	426	2873
time pigz -p2 test2	228	2865
time pigz -p4 test4	150	2865
time pigz -p16 test16	81	2865
time pigz test32	29	2865
time cp test1 test16	147	32767

Tabelle 2: gzip versus pigz 32 GB

```
dnf install kmod-vdo vdo
systemctl start vdo
systemctl enable vdo
systemctl status vdo
(bei einem Problem ist ein reboot hilfreich)
vdo create --name=vdo1 --dev=/dev/xvdb --vdoLogicalSize=300G
vdostats -human-readable
mkfs.xfs -K /dev/mapper/vdo1
mount /dev/mapper/vdo1 /compressed
```

Listing 1: VDO-Installation und Konfiguration

```
[oracle@srv-oracle01 oracle]$ yum install lsyncd
[oracle@srv-oracle01 oracle]$ more /etc/lsyncd.conf

----
-- User configuration file for lsyncd.
--
settings {
    logfile = "/var/log/lsyncd/lsyncd.log",
    statusFile = "/var/log/lsyncd/lsyncd.status"
}
sync {
    default.rsyncssh,
    source = "/opt/oracle/standby/srv_oracle01",
    host = "srv-oracle02",
    targetdir = "/opt/oracle/standby/srv_oracle01/",
    rsync = {
        archive = true,
        owner = true,
        perms = true,
        group = true,
        compress = false,
        whole_file = true,
    }
}

[oracle@srv-oracle01 oracle]$ systemctl enable lsyncd
[oracle@srv-oracle01 oracle]$ systemctl start lsyncd
```

Listing 2: Lsyncd-Installation und Konfiguration

DBMS_PIPE und rman pipe

Besser ist rman pipe und DBMS_PIPE. Es geht einfach darum, rman aus einer Session fernzusteuern, ein oder mehrere Kommandos abzusetzen und die Rückgabe in der Session auszuwerten. Meine Motivation ist, lieber eine Prozedur oder ein Paket in PL/SQL zu haben als ein Windows-Skript und ein Linux-Skript, das eventuell für ein anderes Linux nochmals angepasst werden muss. Man startet einen rman-Prozess mit dem PIPE-Parameter und einem Namen, es können also mehrere rman-Prozesse gestartet werden (siehe Listing 5).

In der Datenbank sind zwei Pipe-Prozesse verfügbar:

- 'ORA\$RMAN_' || 'PIPELINE_1' || '_IN'
- 'ORA\$RMAN_' || 'PIPELINE_1' || '_OUT'

Mit PL/SQL können dann Kommandos an den rman gesendet werden. Für den Einstieg dürften die Beispiele von Herrn Debes [5] nützlich sein. Mit der Benutzung von rman wird auf die Konfiguration des rman zugegriffen, was Verzeichnisse, Dateinamen, Dateigrößen und die Retention Policy betrifft. Damit kann ein eigenes paralleles Backup für die Datenbank erstellt werden. Wenn man einen Blick in den dazugehörigen Statspack Report wirft, erhält man folgende Werte (siehe Listing 6).

Die CPU-Zeit für die Komprimierung wird der Datenbankinstanz zugeschlagen und nicht auf die Zeit für aktive Prozesse angerechnet.

```

create or replace NONEDITIONABLE PROCEDURE pr_rman4m2_datafile (
    p_fileid NUMBER,
    p_path   VARCHAR2
) AS

    done          BOOLEAN;
    concur        BOOLEAN;
    pieceno      BINARY_INTEGER;
    handle       VARCHAR2(255);
    comment      VARCHAR2(255);
    media        VARCHAR2(255);
    params       VARCHAR2(255);
    archlog_failover BOOLEAN;
    recid        NUMBER;
    stamp        NUMBER;
    tag          VARCHAR2(32) := 'TEST';
    filename     VARCHAR2(512);
    devicename   VARCHAR2(512);
    set_stamp    NUMBER;
    set_count    NUMBER;
BEGIN
    dbms_output.put_line(' ');
    dbms_output.put_line(' BACKUP: Allocating device... ');
    devicename := dbms_backup_restore.deviceallocate;
    dbms_output.put_line('devicename: ' || devicename);
    dbms_output.put_line(' BACKUP: Specifying datafiles... ');
    dbms_backup_restore.backupsetdatafile(set_stamp, set_count);
    dbms_output.put_line('set_stamp: ' || set_stamp);
    dbms_output.put_line('set_count: ' || set_count);
    filename := p_path
                || '/datafile_'
                || to_char(p_fileid)
                || '_'
                || to_char(sysdate, 'YYYYMMDD_HH24MISS')
                || '.dbf';

    dbms_backup_restore.backupdatafile(p_fileid);
    dbms_output.put_line(' BACKUP: Create piece ');
    dbms_backup_restore.backuppiececreate(filename, pieceno, done, handle, comment,
                                           media, concur, params, reuse => true, archlog_failover =>
archlog_failover,
                                           deffmt => 0, recid => recid, stamp => stamp, tag => tag,
docompress => true);

    dbms_backup_restore.devicedeallocate;
    IF done THEN
        dbms_output.put_line(' BACKUP: datafile '
                               || p_fileid
                               || ' ok');
    ELSE
        dbms_output.put_line(' BACKUP: datafile '
                               || p_fileid
                               || ' error');
    END IF;

END pr_rman4m2_datafile;

```

Listing 3: DBMS_BACKUP_RESTORE

```

begin
  utl_file.fremove ('DBDUMP', 'test.dmp');
  dbms_file_transfer.get_file(
    'DBDUMP',
    'test.dmp',
    'DBLINK_PPSDB',
    'DBDUMP',
    'test.dmp');
exception
  when others then
    dbms_output.put_line(substr(sqlerrm,1,200));
end;
    
```

Listing 4: DBMS_FILE_TRANSFER

```

rman PIPE PIPELINE_1 target / log /tmp/rman_1.log
    
```

Listing 5: rman pipe

```

Elapsed:      66.27 (mins) Av Act Sess:      0.4
DB time:     29.17 (mins) DB CPU:           21.40 (mins)
    
```

Event	Waits	Avg Wait	%Total Call	Time (ms)	Time
-----CPU tim					
e	7,844		76.0		
RMAN backup & recovery IO	151,105	411	3	4.0	
control file sequential r	182,521	192	1	1.9	


```

Instance CPU
~~~~~
% Time (seconds)
-----
Host: Total time (s):      162,042.8
Host: Busy CPU time (s):   68,983.5
% of time Host is Busy:   42.6
Instance: Total CPU time (s): 38,376.1
%DB time waiting for CPU (Resource Mgr): 0.0
    
```

Listing 6: Statspack

Das ist also eine Idee, wenn freie CPU-Ressourcen auf dem Datenbankserver verfügbar sind. Es ist kein Fullbackup oder Level-0-Backup vorhanden, dafür Backupsets mit ein oder mehreren Datenfiles. Das wird im Recovery-Fall kein Problem sein.

Manual Standby

Einige Datenbanken besitzen eine Manual-Standby-Datenbank. Diese Standby-Datenbanken sind für den Recover-Prozess im Mount-Modus. Damit ist es genau wie in der Enterprise Edition

auch möglich, die Standby-Datenbank für die Backups der Datenfiles zu verwenden. Ein Punkt ist das Fehlen des STANDBY_FILE_MANAGEMENT-Parameters. Was bei neuen REDOLOG Files eher ein kleineres Problem ist, wird bei neuen Tablespace-Dateien unangenehm. Eine neue Tablespace-Datei muss einmal auf der Primär-Datenbank ein Backup erhalten und dann auf der Standby-Seite „restored“ werden. Ein zweiter Punkt, den man nicht vergessen sollte: Nach dem Backup auf der Standby-Datenbank sollten auf der Primär-Datenbank die Backups katalogisiert werden. Die Backups der Archive-

log-Dateien würden weiterhin von der Primärseite durchgeführt.

Refreshable PDBs

Oracle hat mit der Version 19c Refreshable PDBs eingeführt. Damit ist das Problem neuer Tablespace-Dateien Geschichte. Für Refreshable PDBs gilt aber

1. Es ist kein Backup möglich.
2. Es kann kein Klon erzeugt werden.

Hebt man das „Refreshable“ einer PDB einmal auf, muss man eine neue Refreshable PDB erstellen.

Quellen

- [1] <https://www.aspicon.de/news/make-backup-compression-great-again>
- [2] <https://www.redhat.com/en/blog/look-vdo-new-linux-compression-layer>
- [3] How to Manage Virtual Data Optimizer(VDO) Volumes using Cockpit Web Console (Doc ID 2621080.1)
- [4] Norbert Debes: Secrets of the Oracle Database
- [5] https://github.com/Apress/secrets-of-oracle-db/blob/master/source_code_depot/rman_pipe_if.sql

Über den Autor

Thomas Beetz ist seit zehn Jahren DBA bei merlin.zwo. Erfahrungen mit Oracle hat er seit über fünfundzwanzig Jahren, ab Version 7.2 bis 19.17. Umgang mit OVM, OLVM und ODA sowie Data Guard und RAC gehören ebenfalls zum Aufgabengebiet.



Thomas Beetz
thomas.beetz@merlin-zwo.de



Ist Ihre Datenbank 23c-ready?

Dierk Lenz, Herrmann & Lenz Services

In diesem Jahr wird es eine neue Oracle-Datenbank-Version geben – 23c! Aktuell kennen wir noch keine Liste von neuen und abgekündigten Features, jedoch ist die Dokumentation des Innovation Release 21c bereits recht aussagekräftig.

Warum soll man sich jetzt schon mit 23c beschäftigen? Die Version 19c läuft mittlerweile bei den allermeisten Kunden und macht sich prinzipiell sehr gut. Der Wertstropfen liegt beim Ende des Premier Support am 30.4.2024 beziehungsweise beim Ende des geschenkten Extended Support am 30.4.2025. Das bedeutet, dass wir – für den Fall, dass der Datenbankbetrieb grundsätzlich mit einer vom Support unterstützten Version laufen

soll – bis April 2025 auf 23c migriert haben sollten, wobei die neue Version noch nicht verfügbar ist. Für Kunden mit Extended Support ist das Enddatum für 19c der 30.4.2027.

Daher macht es Sinn, sich insbesondere die bereits mit 21c abgekündigten Features anzuschauen. Hier steckt im Zweifel eine ganze Menge Vorbereitungsaufwand. Der Artikel beschäftigt sich mit einigen dieser Themen.

Pluggable Datenbanken

Bereits mit 21c ist es so weit: Die alte Datenbankarchitektur ohne Container-Datenbanken wird nicht mehr unterstützt. Daher empfiehlt es sich, bereits jetzt – mit 19c – auf die Multitenant-Architektur zu wechseln.

Die Multitenant-Architektur wurde mit 12c im Jahr 2013 eingeführt. Anfangs hatte das Ganze noch einige Ecken und

Kanten; mit 12c R2 im Jahr 2017 wurde das viel besser. Mit 19c im Jahr 2019 wurde schließlich die Einschränkung von einer PDB pro CDB für die Standard Edition sowie die Enterprise Edition ohne Multitenant-Option auf drei PDBs pro CDB angehoben. Das macht vieles einfacher, zum Beispiel das gelegentliche Test-Clonen von Datenbanken oder den Betrieb von zwei beziehungsweise drei PDBs in einer CDB, wenn die Ressourcen ausreichen.

Die Frage, wie viele PDBs in einer CDB betrieben werden, ist insbesondere bei der Standard Edition sorgfältig abzuwägen, da die 16-Thread-Limitierung auf CDB-Ebene gilt und nicht etwa auf PDB-Ebene.

Es ist generell empfehlenswert, mit der Migration auf 19c auch die CDB-Architektur einzuführen. Somit kann man bereits jetzt entsprechende Erfahrungen sammeln und die Vorteile erkunden.

Case-Insensitive- Passwörter

Mit der Oracle Database 12c wurden der Parameter `sec_case_sensitive_logon` als veraltet (*deprecated*) eingestuft. Dieser Parameter ermöglicht es, die Verwendung der neueren Passwortkodierungen ab 11g auszuschalten. Die neueren Verfahren kodieren die Passwörter *case sensitive*, also unter Beachtung von Groß-/ Kleinschreibung. Dieser Parameter ist ab 21c nicht mehr supported.

Dies bedeutet aktuell (noch) nicht, dass die alte 10g-Kodierung nicht mehr genutzt werden kann. Es muss jedoch in der `sqlnet.ora`-Datei des Datenbank-Servers der Parameter `SQLNET.ALLOWED_LOGON_VERSION_SERVER` auf einen der Werte 11, 10, 9 oder 8 gesetzt sein. Diese Werte unterstützen die Authentifizierungsprotokolle der Versionen 10g, 11g und 12c. Seit 12c R2 ist der Standardwert des Parameters 12, sodass die 10g-Kodierung hiermit ausgeschlossen ist.

Unabhängig von der Migrationsmethode gehen bei der Migration einer Datenbank 10g-Kodierungen verloren, wenn nicht vorher der Parameter `SQLNET.ALLOWED_LOGON_VERSION_SERVER` auf einen der Werte unter 12 gesetzt wird.

Der Standardwert 12 sowie die Abschaffung des Parameters `sec_case_sensitive_logon` zeigen, dass Oracle

insbesondere die 10g-Kodierung so bald wie möglich loswerden will. Oracle möchte hier verständlicherweise einen Schlusstrich ziehen, da die vorhandenen 10g-Kodierungen einfacher zu attackieren sind und somit eine potenzielle Gefahr darstellen.

Falls es Benutzer in der Datenbank gibt, die ausschließlich eine 10g-Kodierung für das Passwort besitzen (erkennbar am Wert '10G' in der Spalte `password_versions` der View `dba_users`), dann sollte für diese Benutzer das Passwort unter Verwendung einer neuen Kodierung geändert werden.

Diverse Features auf der Windows-Plattform

Die Oracle-Integration in das Microsoft Cluster, Fail Safe, wird ab 21c nicht mehr supported. Oracle empfiehlt als Ersatz Produkte wie RAC One Node oder SEHA, beide basierend auf der Grid Infrastructure.

Weiterhin wird das ASM-Cluster-File-System ACFS unter Windows ebenfalls nicht mehr unterstützt. Bei entsprechendem Bedarf kann ASM ohne File System weiter eingesetzt werden.

Enterprise Manager Express

Die Flash-basierte Version wird ab 21c nicht mehr unterstützt, was zu erwarten war. Generell wird der komplette EM Express gleichzeitig als veraltet deklariert. Daher bleibt vom EM Express nicht mehr viel übrig. Oracle empfiehlt, zum Beispiel den SQL Developer als Ersatz zu nutzen. Die Firma Herrmann & Lenz rät, für Monitoring-Zwecke das Werkzeug HL Monitoring Module zu verwenden (Infos unter <https://monitoring-module.com>).



Dierk Lenz
dierk.lenz@hl-services.de



iJUG

Verbund

www.ijug.eu

FÜR 29,00 €
BESTELLEN

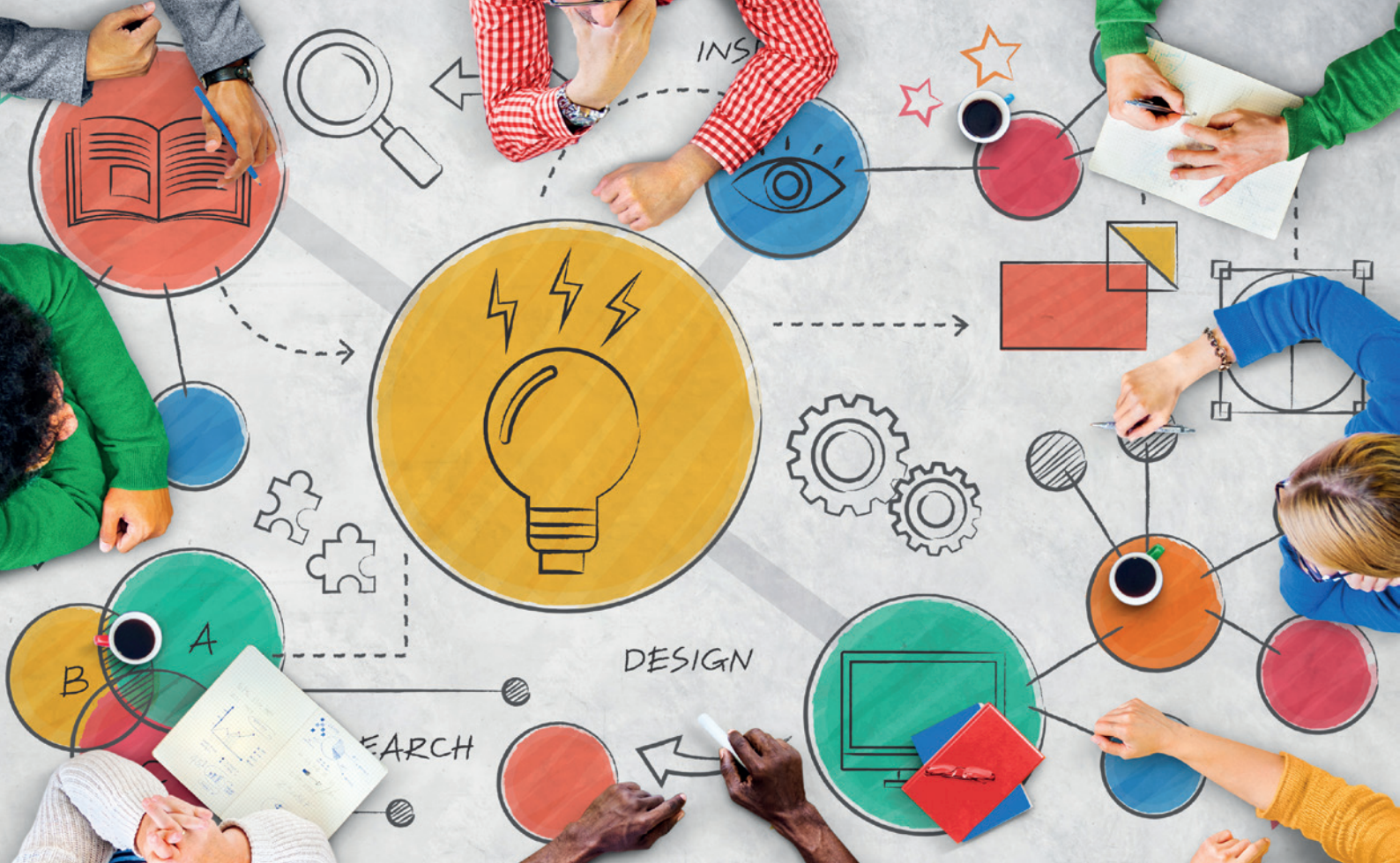
Java aktuell

JAHRESABO

Mehr Informationen zum Magazin und Abo unter:

www.ijug.eu/de/java-aktuell





Workflow-Funktionalität in Oracle APEX 22.2 (2/2)

Ralf Müller, Oracle

Im ersten Teil des Artikels [1] haben wir uns ausführlich mit der Server-Seite der Approvals-Komponente in Oracle APEX 22.2 beschäftigt. In diesem zweiten Teil wollen wir uns mehr auf die Benutzerschnittstelle fokussieren und darauf, welche Hilfsmittel wir dem Entwickler in diesem Bereich zur Verfügung stellen. Darüber hinaus betrachten wir die öffentlichen Schnittstellen der Approvals-Komponente und geben einen Ausblick, wie es mit APEX Workflow weitergeht.

Unified Task List

In der „Unified Task List“ kommen **alle** Tasks von **allen** APEX-Applikationen in **einem** APEX Workspace für den angemeldeten Benutzer und seine zugeordnete Aufgabe zusammen, daher auch der Name „Unified“. Je nach Ausprägung der Unified Task List werden unterschiedliche Tasks angezeigt und Opera-

tionen auf diesen Tasks ermöglicht. *Tabelle 1* gibt einen Überblick.

Der Entwickler kann eine „Unified Task List“ wie jede andere Seite einer APEX-Applikation aus dem APEX App Builder recht einfach erzeugen (*siehe Abbildung 1*).

Im „Create Page“ Wizard steht dafür ein eigenes Element „Unified Task List“ zur Verfügung. Wählt man dieses aus, gelangt man zu einem Dialog wie in *Abbildung 2*.

Wie bei den meisten Seiten im App Builder muss man hier eine Seitennummer sowie einen Namen für die Seite angeben. Darüber hinaus verlangt die Unified-Task-Liste noch die Angabe des sogenannten „Report Context“ (My Tasks, Admin Tasks, Initiated by Me, *siehe Abbildung 2*) und das war's dann auch schon. Optional sind noch die gängigen Einstellungen für die Seitennavigation

konfigurierbar. Das Ergebnis ist eine voll funktionsfähige Task-Liste für den ausgewählten Kontext. Die durch diesen Wizard erzeugte Seite ist eine ganz normale APEX-Seite, die vom APEX-Entwickler an die jeweiligen Bedürfnissen angepasst werden kann.

Anatomie der Unified Task List

Die Unified Task List ist eine generierte Seite mit den folgenden Komponenten (siehe Abbildung 3).

- **Breadcrumb Bar**

Die Breadcrumb Bar besteht im Wesentlichen aus einem Smart Filter zur Suche von Tasks in der Task-Liste nach bestimmten Kriterien. Als Kriterien stehen alle wichtigen Merkmale einer Task zur Auswahl, zum Beispiel die Applikation, die die Task erzeugt hat, die Priorität, das Ablaufdatum etc.

- **Body**

Der Body besteht aus einer Cards Region. Diese dient zur Anzeige der Tasks in der Task-Liste. Zur Anzeige kommt der Titel der Task mit einigen anderen Eigenschaften wie dem Namen der Task Definition, der Angabe, wer die Task erzeugt hat und an wen die Task momentan zugewie-

sen ist, dem Ablaufdatum sowie dem momentanen Zustand der Task. Der Titel der Task ist ein Link, der zur sogenannten „Task-Details“-Seite führt, doch dazu später mehr.

Falls es sich um eine Task-Liste mit dem Report Context „My Tasks“ handelt, sind zwei Knöpfe für das Genehmigen und Ablehnen direkt aus der Task-Liste selbst verfügbar. Ansonsten sind abhängig vom Report Context die verschiedenen Operationen in der Task-Details-Seite verfügbar. Diese Funktionalität ist mithilfe von Dynamic Actions implementiert und kann leicht angepasst werden.

An dieser Stelle ergibt es vielleicht Sinn zu erwähnen, dass die Task-Liste über eine sogenannte „Pipelined Table Function“ gespeist wird, diese Funktion ist im APEX_APPROVAL API verfügbar und kann somit auch für eigene Anwendungen genutzt werden. Ein Blick in die Implementierung des Reports „My Tasks“ der Unified Task

List zeigt folgende SQL Query für die Befüllung der Cards Region (siehe Listing 1).

Hier kommt das API APEX_APPROVAL.GET_TASKS zum Einsatz, als Parameter verlangt es den Report Kontext (MY_TASKS, ADMIN_TASKS, INITATED_BY_ME).

Task Details

Als Nächstes betrachten wir die „Task Details“-Seite und erklären, wie diese mit der Unified-Task-Liste zusammenhängt. Im ersten Teil des Artikels sind wir nur sehr wagen auf dieses Detail in der Task Definition eingegangen – Zeit, etwas mehr Licht ins Dunkle zu bringen. Wir erinnern uns, in den Settings einer Task Definition (unter Shared Components einer APEX-Applikation) gibt es eine Einstellung für „Task Details Page URL“ (siehe Abbildung 4).

Wie der Name schon vermuten lässt, handelt es sich dabei um die URL einer Seite der APEX-Applikation, die für die

```
select *
  from table (
    apex_approval.get_tasks (
      p_context           => 'MY_TASKS',
      p_show_expired_tasks => :P6_SHOW_EXPIRED ) )
```

Listing 1: SQL Query für die Befüllung der Cards Region

Report Context	Was wird angezeigt?	Welche Operationen sind möglich?
My Tasks	Alle Tasks, für die der angemeldete Benutzer ein „Potential Owner“ ist und die nicht von ihr/ihm selbst angelegt wurden.	<ul style="list-style-type: none"> • Genehmigen • Ablehnen • Kommentar hinzufügen • Information anfordern • Delegieren • Freigeben • Anfordern und Reservieren
Admin Tasks	Alle Tasks, für die der angemeldete Benutzer ein „Business Administrator“ ist.	<ul style="list-style-type: none"> • Kommentar hinzufügen • Delegieren • Priorität ändern • Verwerfen • Fälligkeitsdatum ändern • Teilnehmer einladen • Erneuern
Initiated by Me	Alle Tasks, die von dem angemeldeten Benutzer selbst erzeugt wurden.	<ul style="list-style-type: none"> • Kommentar hinzufügen • Priorität ändern • Verwerfen • Information zustellen

Tabelle 1: Report Context der Unified Task List

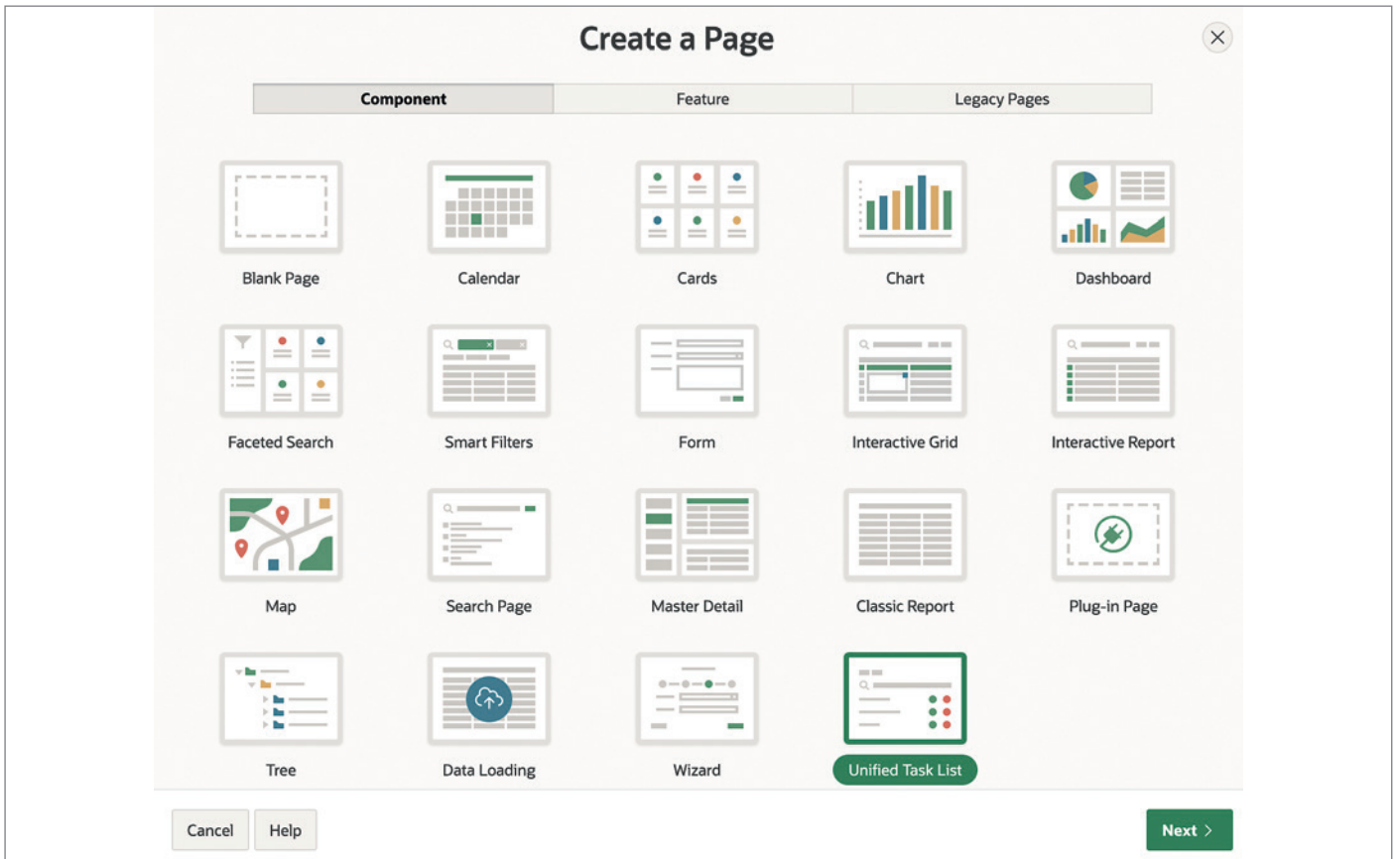


Abbildung 1: Unified Task List im Create Page Wizard (Quelle: Ralf Müller)

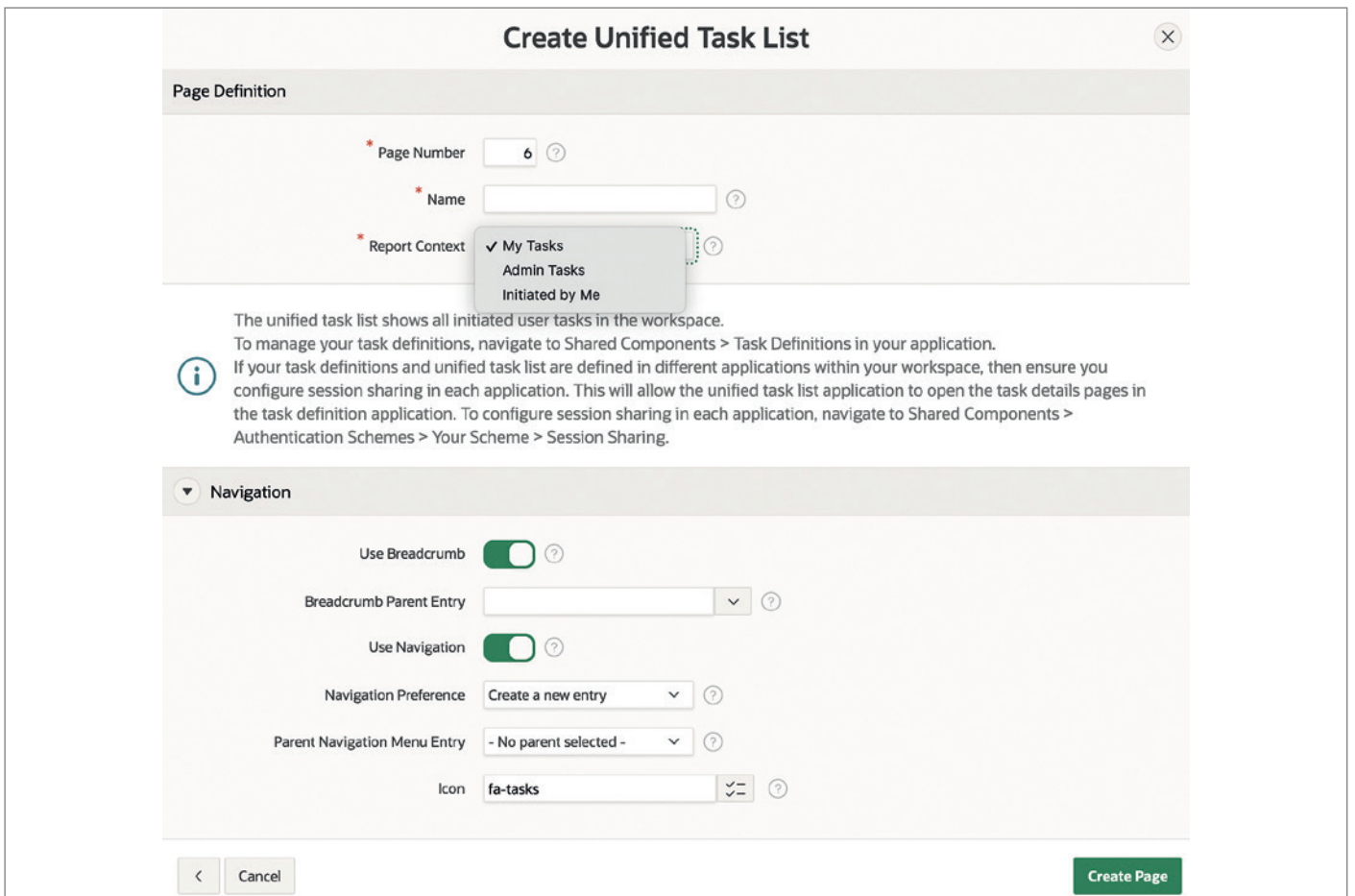


Abbildung 2: Create Unified Task List Wizard (Quelle: Ralf Müller)

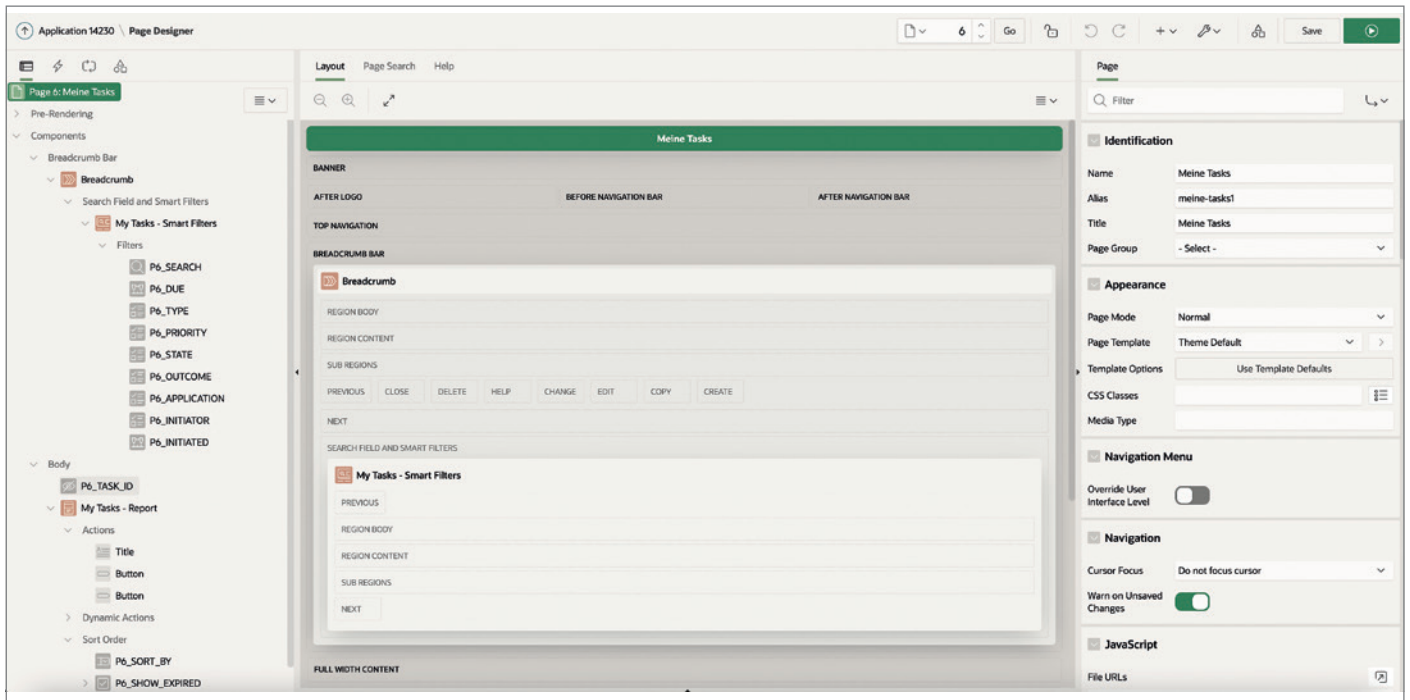


Abbildung 3: Generierte Unified-Task-List-Seite (Quelle: Ralf Müller)

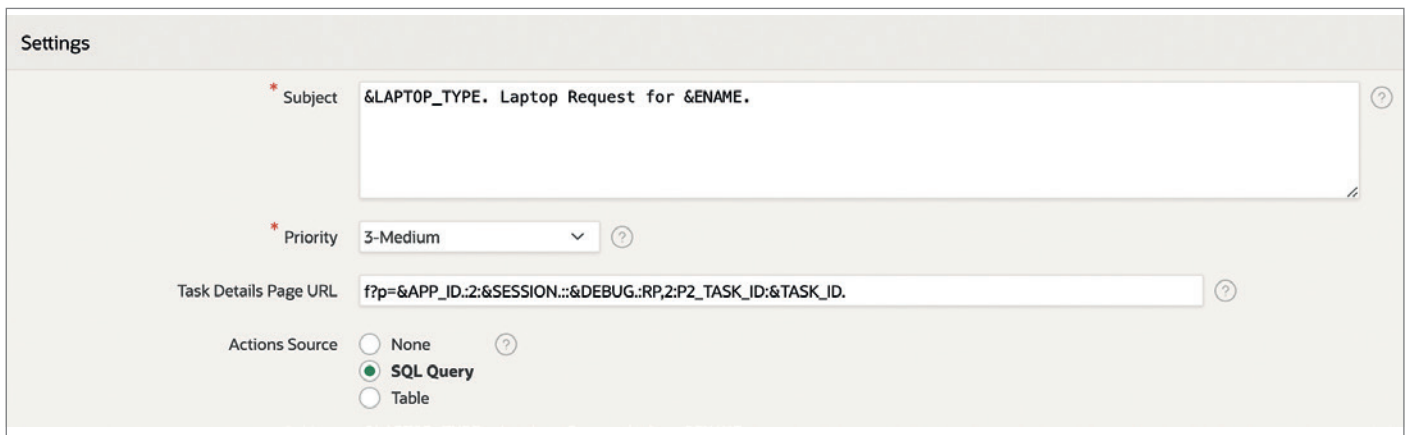


Abbildung 4: Task Definition Settings (Quelle: Ralf Müller)

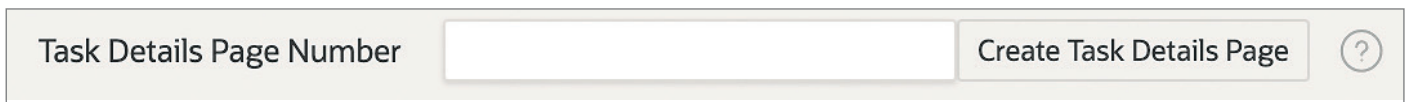


Abbildung 5: Leere Task Details Page Number beim Anlegen einer Task Definition (Quelle: Ralf Müller)



Abbildung 6: Vom Benutzer vorgegebene Seitennummer für die Task-Details-Seite (Quelle: Ralf Müller)



Abbildung 7: Warnung mit der Option, die nächste freie Seitenzahl für die Task-Details-Seite zu verwenden (Quelle: Ralf Müller)

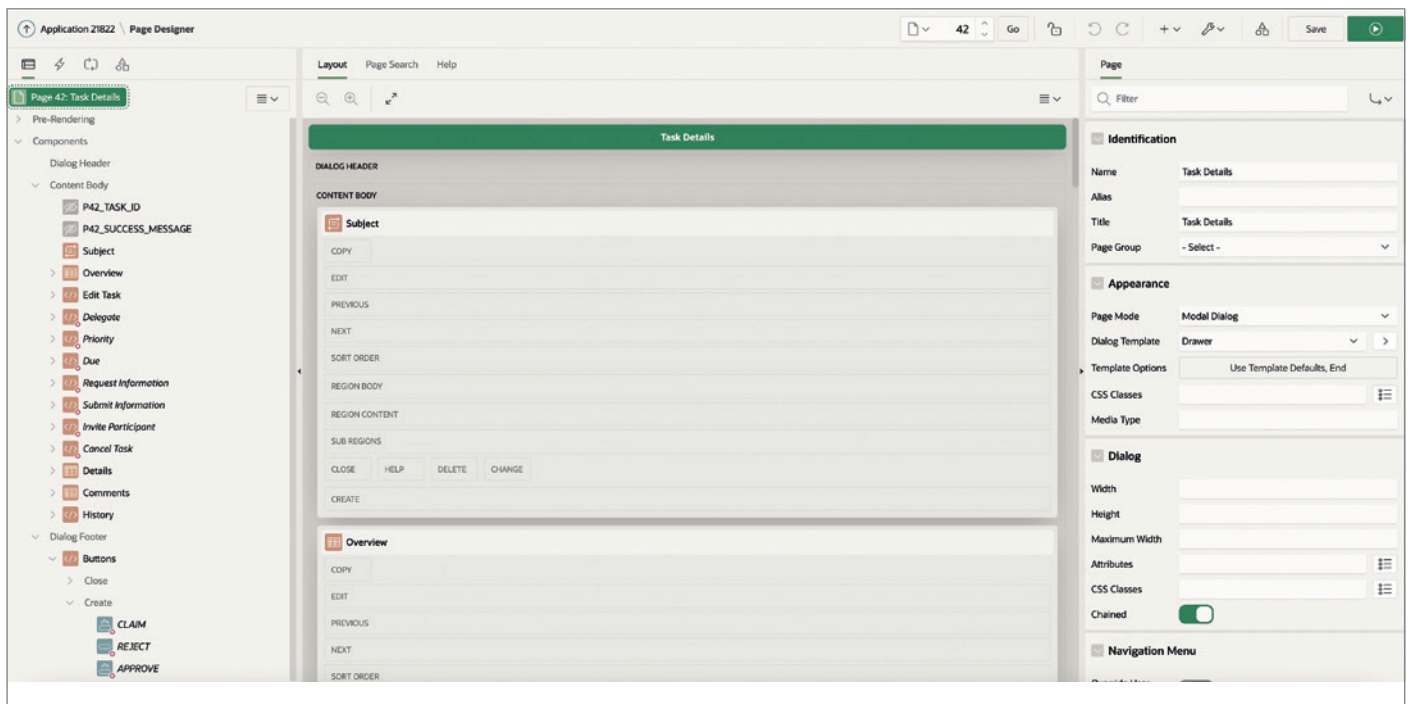


Abbildung 8: Generierte Task-Details-Seite (Quelle: Ralf Müller)

Operation	Beschreibung
Approve	Genehmigen einer Task
Reject	Ablehnen einer Task
Comment	Hinzufügen eines neuen Kommentars
Claim	Exklusive Anforderung einer Task
Delegate	Delegieren der Task an einen anderen Potential Owner
Release	Freigeben der Task, nachdem sie zuvor durch eine Anforderung reserviert war
Cancel	Verwerfen der Task, wenn diese nicht mehr gebraucht wird
Invite Participant	Einladen eines Benutzers zur Bearbeitung der Task
Set Priority	Verändern der Priorität einer Task
Request Information	Anfordern (zusätzlicher) Information
Submit Information	Zustellen von Information
Set Due Date	Ändern des Ablaufdatums der Task
Renew	Erneuern einer abgelaufenen Task

Tabelle 2: Human-Task-Manage-Operationen

Darstellung der Task-spezifischen Informationen zuständig ist. Wie schon bei der Unified Task List lässt sich per Mausklick eine Standardseite für die Darstellung erzeugen. Legt der Entwickler eine neue Task Definition an, so ist das Feld „Task Details Page Number“ zunächst leer (siehe Abbildung 5).

Es besteht die Möglichkeit, hier eine Seitenzahl zu spezifizieren, bevor man die Seite generieren lässt (siehe Abbildung 6).

Wird keine Seitenzahl angegeben, dann erscheint dieser Dialog beim Drücken des „Create Task Details Page“-Knopfes und der Entwickler kann sich dafür entscheiden, dass APEX selbst eine Seitennummer auswählt (siehe Abbildung 7).

Das Ergebnis dieses Vorgangs ist eine neu generierte Seite in der APEX- Applikation zur Darstellung der Task-spezifischen Informationen. Wie schon bei der Unified Task List kann diese Seite vom Entwickler an eigene Bedürfnisse angepasst werden, sie dient lediglich als guter Ausgangspunkt für Anwendungsspezifische Änderungen.

Anatomie der Task-Details-Seite

Die generierte Task-Details-Seite in APEX 22.2 sieht in etwa so aus wie in Abbildung 8 dargestellt.

Die Seite ist ein Modaler Dialog und besteht aus den folgenden Komponenten

- **Subject (Dynamic Content)**

Hier wird der (aufgelöste) Titel der Task angezeigt. Der Titel findet sich auch in der Unified Task List wieder. Dies dient zur einfachen Identifizierung der Task.

- **Overview (Classic Report)**

Allgemeine Informationen über die Task wie der Erzeuger, die Priorität, das Ablaufdatum, der momentane Besitzer etc.

- **Details (Classic Report)**

Dieser Report zeigt per Default die Task-Parameter mit ihren Werten an. Diese Region ist wohl am ehesten für anwendungsspezifische Anpassungen gedacht. Hier sind dem Entwickler keine Grenzen gesetzt. Basierend auf den Task-Parametern oder dem Kontext der Task kann man diese Region dafür benutzen, alle relevanten Informationen aus der Datenbank zusammenzubringen und für den Benutzer in einer geeigneten Form darzustellen.

- **Comments (Classic Report)**

Diese Region enthält die Kommentare für eine Task.

- History (Static Content mit Classic Report Region)

Diese Region gibt einen Überblick über die Historie der Task. Hier wird angezeigt, welcher Benutzer welche Operationen mit der Task ausgeführt hat und wann das geschehen ist.

- **Dialog Footer**

- Claim (Button)
Knopf zum Anfordern der Task
- Reject (Button)
Knopf zum Ablehnen der Task
- Approve (Button)
Knopf zum Genehmigen der Task

- **Static Content**

Die verschiedenen Operationen der Task sind als Static Content mit Buttons generiert. Je nachdem, in welchem Kontext die Task-Details-Seite geöffnet wird, werden verschiedene Knöpfe sichtbar und für den Anwender somit benutzbar.

Die einzelnen Knöpfe in der Task-Details-Seite sind mit Prozessen hinterlegt, die mithilfe des Human-Task-Management-Plug-ins implementiert werden (siehe Abbildung 9).

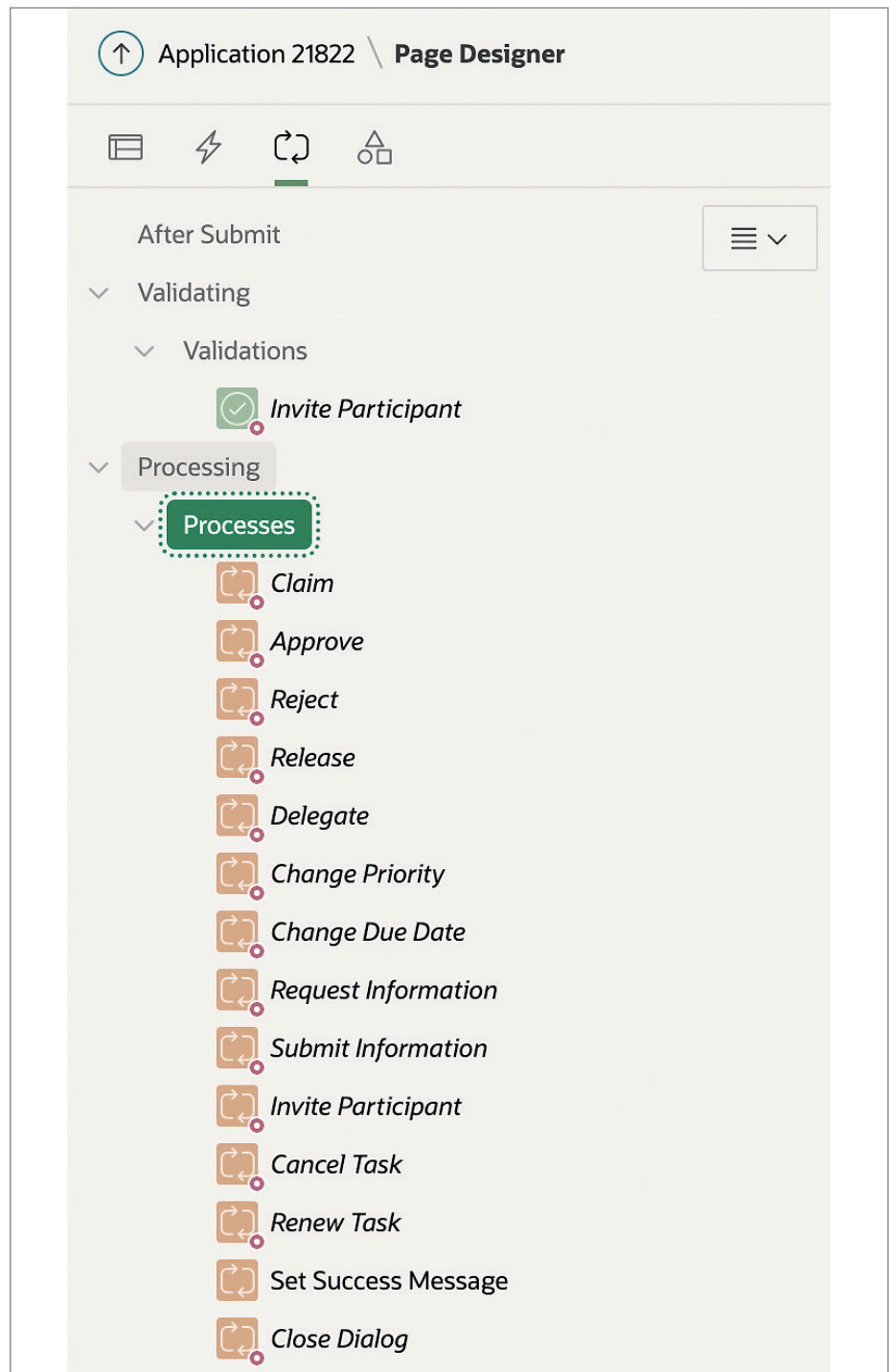


Abbildung 9: Prozesse der Task-Details-Seite (Quelle: Ralf Müller)

Unified Task List und Task Details, wie gehört das zusammen?

Wir haben nun im Detail die Unified Task List sowie die für eine Task Definition einer Applikation konfigurierte Task-Details-Seite betrachtet. Aber wie gehören die beiden unterschiedlichen Seiten zusammen? Zur Erinnerung, die Unified Task List zeigt alle Tasks von allen Anwendungen eines APEX Workspace für den

angemeldeten Benutzer, während die Task-Details-Seite jeweils nur die Informationen einer einzigen Task einer Applikation anzeigt. Bildlich dargestellt sieht das Ganze so aus wie in *Abbildung 10*.

Für den Entwickler bedeutet das

- Es ist egal, in welcher Applikation die Unified-Task-Liste zur Anwendung kommt, sie kann sogar in mehrere Anwendungen eingebettet werden und mit verschiedenen oder demselben

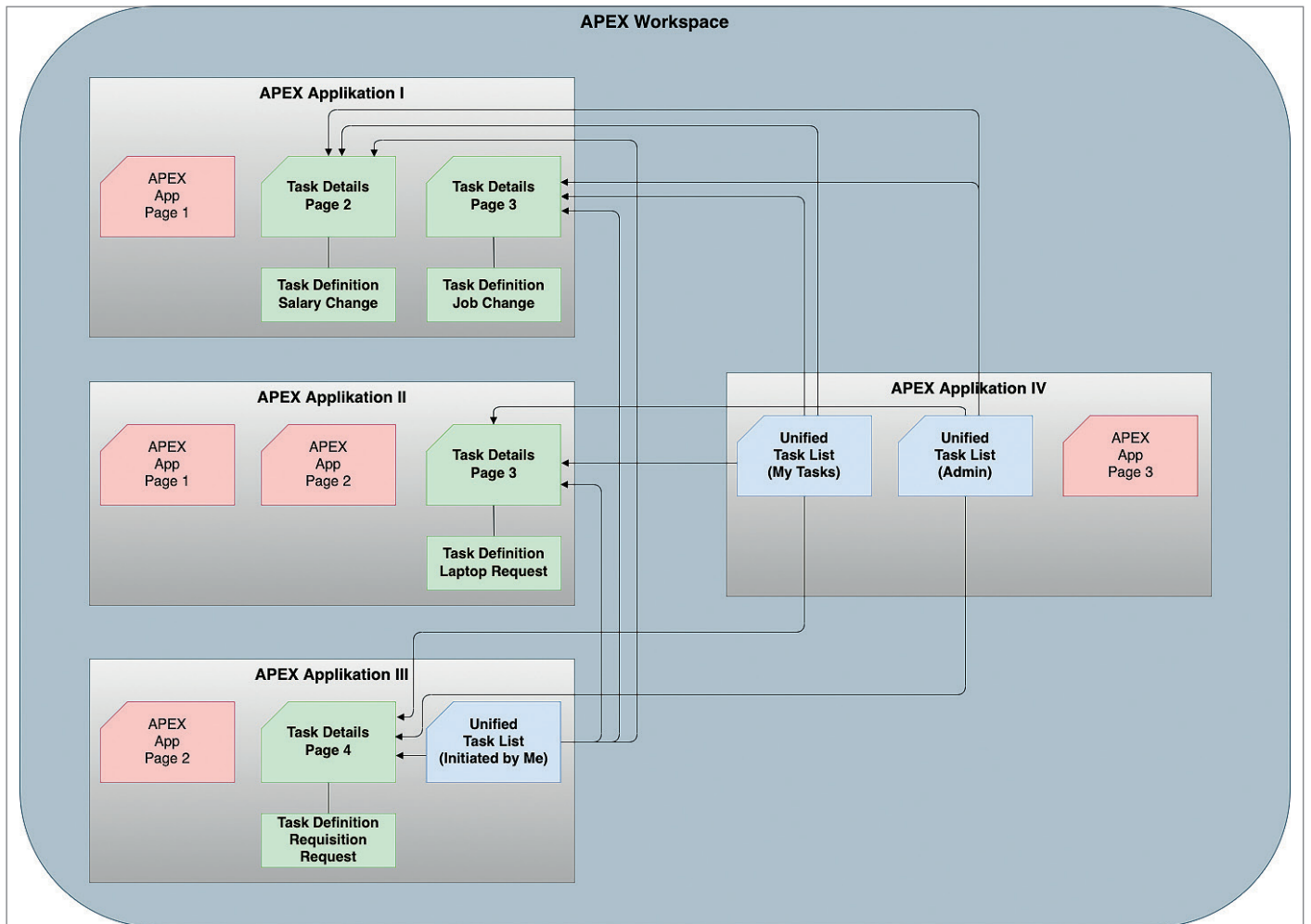


Abbildung 10: Beispiel für das Zusammenspiel von Unified Task List und Task-Details-Seiten (Quelle: Ralf Müller)

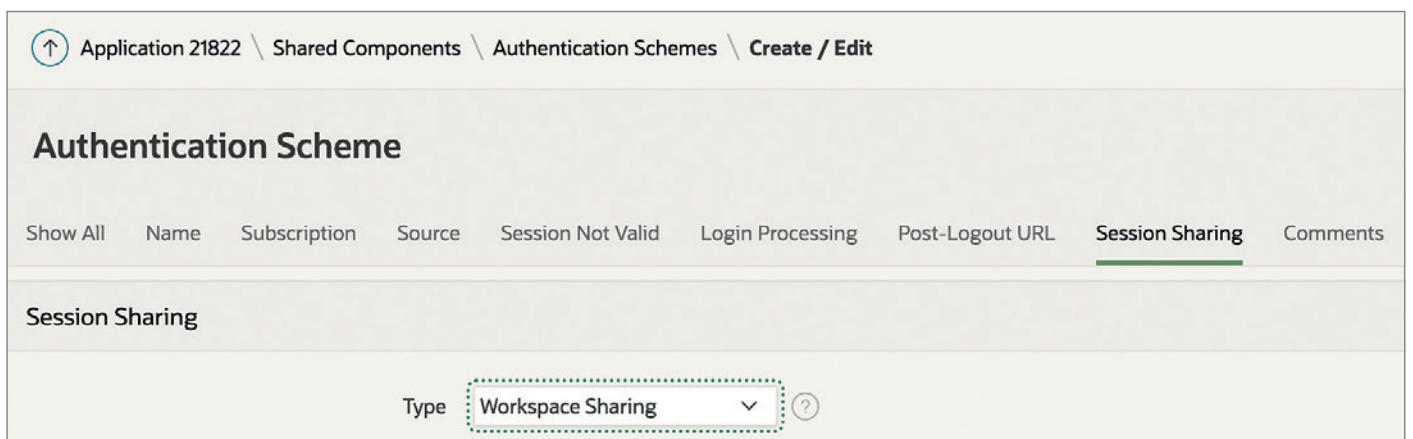


Abbildung 11: Session-Sharing-Einstellung für verteilte Unified Task List und Task-Details-Seiten (Quelle: Ralf Müller)

Kontext. Angezeigt wird immer das Gleiche, unabhängig davon, in welcher Applikation die Unified Task List verwendet wird.

- Die Unified Task List greift auf die Task-Details-Seite der Applikation zu, in der die Task erzeugt wurde. Daher ist die Task-Details-Seite auch als modaler Dialog generiert.

Sind die Unified Task List und die Task-Details-Seiten über Applikationsgrenzen verteilt, muss der Entwickler in den „Shared Components“ der Applikationen das Session Sharing auf Workspace-Ebene konfigurieren (siehe Abbildung 11).

Ansonsten erlaubt APEX das Öffnen der Task-Details-Seite aus der Unified Task List nicht.

Alles schön und gut, aber ich möchte die volle Kontrolle

Wer lieber eine Task List und die einzelnen Task-Details-Seiten von Grund auf selbst bauen möchte, ohne die von APEX zur Verfügung gestellten Hilfsmittel zur Generierung der Unified Task List und der einzelnen Task-De-

Clear Session State
Close Dialog
Data Loading
Execute Code
Form - Automatic Row Processing (DML)
Form - Initialization
Human Task - Create
✓ Human Task - Manage
Interactive Grid - Automatic Row Processing (DML)
Invoke API
Reset Pagination
Send E-Mail
User Preferences
Web Service
Show Legacy...

Abbildung 12: Human-Task-Manage-Plug-in für Prozesse (Quelle: Ralf Müller)

View Name	Beschreibung
APEX_TASKS	View über alle Task-Instanzen eines APEX Workspace
APEX_TASK_PARTICIPANTS	Teilnehmer von Tasks
APEX_TASK_PARAMETERS	Parameter und deren Werte von Tasks
APEX_TASK_COMMENTS	Kommentare von Tasks
APEX_TASK_HISTORY	Historie von Tasks

Tabelle 3: SQL Views für Task-Instanzen

View Name	Beschreibung
APEX_APPL_TASKDEFS	View über alle Task-Definitionen eines APEX Workspace
APEX_APPL_TASKDEF_ACTIONS	Task-Aktionen von Task-Definitionen
APEX_APPL_TASKDEF_PARAMS	Parameter von Task-Definitionen
APEX_APPL_TASKDEF_PARTICIPANTS	Teilnehmer von Task-Definitionen
APEX_APPL_TASKDEF_COMP_PARAMS	Task-Definitionen Parameter-Werte

Tabelle 4: Task-Definitionen Metadaten Views

tails-Seiten, kann das natürlich auch machen. In der Tat benutzen sowohl die Unified Task List als auch die Task-Details-Seiten nur öffentlich zugängliche Schnittstellen und Plug-ins, damit ist gewährleistet, dass sämtliche Funktionalität für die eigene Kreativität zur Verwaltung von Tasks zur Verfügung steht.

Das Human-Task-Manage-Page-Plug-in wurde speziell an die Bedürfnisse einer Task-Details-Seite entwickelt und enthält alle Funktionen zum Umgang mit einer Task. Im APEX App Builder steht es für Prozesse bereit (siehe Abbildung 12).

Das Human-Task-Manage-Plug-in umfasst eine Reihe an Operationen (siehe Tabelle 2).

Diese Operationen sind auch als API im Package **APEX_APPROVAL** vorhanden.

Zum Erstellen einer Task List stellen wir die schon erwähnte Pipelined-Table-Funktion `APEX_APPROVAL.GET_TASKS` bereit.

SQL Views

Die Approval-Komponente veröffentlicht eine Reihe von SQL Views, die vom Entwickler für eigene Reports benutzt werden können. Für Task-Instanzen ist die übergeordnete View `APEX_TASKS`, die Informationen zu allen Tasks eines APEX Workspace enthält, während die anderen Views jeweils Informationen über Ausprägungen (Teilnehmer, Parameter etc.) von Tasks enthalten (siehe Tabelle 3).

Darüber hinaus existieren noch Views zu den Metadaten von Task-Definitionen (siehe Tabelle 4).

Verschiedenes

Für den Entwickler dürften noch die folgenden Funktionen interessant sein

- In APEX 22.2 gibt es nun die Möglichkeit, eine Task Definition zu kopieren. Dabei kann die ursprüngliche Task Definition aus einer anderen Applikation kommen. Damit erleichtern wir den Umgang mit Task-Definitionen; mit der Kopierfunktion müssen diese nicht immer von Grund auf neu angelegt werden. Wenn ähnliche Task-Definitionen

vorhanden sind, kann man diese kopieren und anschließend verändern.

- Task-Instanzen einer Applikation bleiben bestehen, wenn man die Applikation unter der gleichen Applikations-ID neu importiert. Das haben wir im ersten Patchset von APEX 22.2 implementiert. Vorher wurden alle Task-Instanzen beim Neu-Import einer Applikation gelöscht.
- Abgeschlossene Tasks werden nach einem bestimmten Zeitraum als Datei exportiert und anschließend gelöscht. Der Default sind 7 Tage, maximal können 30 Tage konfiguriert werden.
- Ab APEX 22.2 lässt sich für Tasks eine Ablaufmethode konfigurieren. Wenn das Ablaufdatum einer Task überschritten ist, wird diese Methode angewendet. In APEX 22.2 sind dies „Expire“ und „Renew“. Das Ablaufdatum einer Task wird jede Stunde mit einem Job geprüft, für abgelaufene Tasks wird dann jeweils die Ablaufmethode angewendet. Für den Entwickler kann es für den Test einer Applikation mühselig sein, immer eine Stunde zu warten, um die Ablaufmethode zu testen. Daher gibt es in APEX 22.2 ein API `APEX_APPROVAL_HANDLE_TASK_DEADLINES`, das den eben beschriebenen Job sofort anstößt.

Fazit und Ausblick

In dieser zweiteiligen Serie über Workflow-Funktionalität in Oracle APEX haben wir einen Einblick in die Entwicklung gegeben und die Approval-Komponente im Detail behandelt. Was noch fehlt, ist der Ausblick in die Zukunft. Wir haben schon angedeutet, dass wir in APEX eine vollumfängliche „Digital Process Automation“-Lösung anbieten möchten, dies dauert und geschieht nicht von heute auf morgen. Wie geht es also weiter mit APEX Workflow?

Bevor wir darauf eingehen, möchten wir klarstellen, dass es mit Flows for APEX [7] schon eine sehr erfolgreiche Open-Source-Workflow-Lösung gibt, an der Oracle aktiv beteiligt ist und die sich ständig weiterentwickelt. So benutzt zum Beispiel Flows for APEX in der Version 22.2 die Approvals-Komponente für User Tasks. APEX Workflow wird jedoch nicht BPMN-2.0-basiert sein; BPMN 2.0 ist sehr umfangreich und erfordert einiges an Training, während wir mit APEX Workflow einen intuitiv zu benutzenden Workflow-Modellierer mit Runtime

anbieten möchten. Außerdem hat es wenig Sinn, zwei BPMN-2.0-Implementierungen in einem Produkt zu positionieren. Die Eckpunkte von APEX Workflow sind:

• Einfachheit

APEX Workflow wird zu Beginn mit sehr wenigen Modellierungselementen ausgeliefert. Wir werden APEX Workflow inkrementell ausbauen, je nach Anforderungen der Kunden.

• Erweiterbarkeit

APEX Workflow ist erweiterbar, wir verwenden das Konzept der Process Type Plug-ins im Workflow und erlauben damit dem Entwickler, eigene Aktivitäten zu implementieren, die dann im Workflow benutzt werden können.

• Integriert

APEX Workflow ist voll in die APEX-Umgebung integriert und hat eine Benutzerführung, die sehr ähnlich zum Page Designer ist.

• Dynamisch

Die Ausführungs-Semantik von APEX Workflow ist nicht auf Sequenzfluss limitiert (so wie in BPMN 2.0). Die Workflow Engine wird in der Lage sein, Workflows, die mehr Daten- beziehungsweise Event-getrieben sind, zu modellieren und auszuführen. Fast alle Workflows im Health-Care-Bereich sind von dieser Art, aber auch solche für Versicherungsfälle oder komplizierte Finanzprodukte.

• Intelligent

Ein Schwerpunkt wird der Einsatz von Machine-Learning-Methoden in den verschiedenen Bereichen des Workflows sein, beispielsweise um die Dauer eines Workflows zu prognostizieren oder Hilfestellung dabei zu geben, ob eine Task abgelehnt oder genehmigt werden soll.

Ich danke meinem Manager Marc Sewtz für das Korrekturlesen dieses Artikels. Zum Schluss gilt mein besonderer Dank meinem Kollegen Ottmar Gobrecht, der nicht nur diesen Artikel quergelesen und Tipps zur Verbesserung gegeben hat, sondern auch alles rund um die Unified Task List sowie die Task-Details-Seite implementiert hat. Ottmar, vielen lieben Dank!

Quellen

- [1] Ralf Müller (2022): Workflow-Funktionalität in Oracle APEX 22.2 (1/2), Red Stack

Magazin 1/23, <https://www.doag.org/de/mitmachen/red-stack-business-news-archiv/zeitschrift/red-stack-magazin-inkl-business-news-nr-01-2023-ist-online/>

- [2] Oracle APEX 22.2 App Builder User's Guide, Managing Approvals, <https://docs.oracle.com/en/database/oracle/apex/22.2/htmldb/managing-approvals.html#GUID-9CA34E4A-72EB-483D-AC02-F335020ADA43>
- [3] Ananya Chatterjee (2022): Introducing Approvals Component in Oracle APEX, <https://blogs.oracle.com/apex/post/introducing-approvals-component-in-oracle-apex>
- [4] Ananya Chatterjee (2022): APEX Approvals Component Continued – Do More with Less, <https://blogs.oracle.com/apex/post/apex-approvals-component-continued---do-more-with-less>
- [5] Ananya Chatterjee (2022): Implementing Multi-Approvals in Oracle APEX 22.1, <https://blogs.oracle.com/apex/post/implementing-multi-approvals-in-oracle-apex-221>
- [6] OMG Business Process Model and Notation 2.0, <https://www.omg.org/spec/BPMN/2.0/>
- [7] Flows for APEX, <https://flowsforapex.org/>

Über den Autor

Ralf Müller hat Mathematik an der Universität Kaiserslautern studiert und arbeitet seit über 26 Jahren bei Oracle in der Software-Entwicklung. In den letzten 20 Jahren hat er sich mit Integrationsthemen und mit allem rund um Business Process Management, Process Automation und Machine Learning beschäftigt. Er vertritt Oracle bei der Standardisierung von BPMN 2.0 und CMMN 1.0 innerhalb der OMG. Davor war er an den OASIS Standards für BPEL4People und WS-HumanTask beteiligt. Seit Ende 2021 arbeitet Ralf Müller im APEX Development Team an „Digital Process Automation“ für Oracle APEX, ein Thema, das uns im APEX Development noch etwas länger beschäftigen wird.



Ralf Müller
ralf.mueller@oracle.com

BEST OF DOAG ONLINE

Eine Auswahl der besten DOAG News
Dezember 2022/Januar 2023

DOAG.tv mit Carsten Czarski – über die Anfänge von APEX und das Oracle APEX Development Team

<https://www.doag.org/de/home/news/doagtv-mit-carsten-czarski-ueber-die-anfaenge-von-apex-und-das-oracle-apex-development-team/>



DOAG.tv mit Florian Barth – über die Stärken und Schwächen der Oracle Database Appliance (ODA)

<https://www.doag.org/de/home/news/doagtv-mit-florian-barth-ueber-die-staerken-und-schwaechen-der-oracle-database-appliance-oda/>



DOAG.tv mit Alexander Kornbrust – Security Next Generation und die Bedeutung von Workshops

<https://www.doag.org/de/home/news/doagtv-mit-alexander-kornbrust-security-next-generation/>



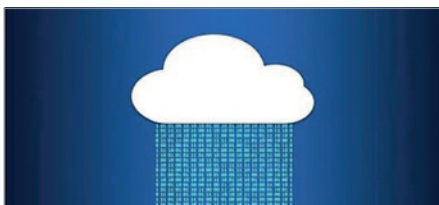
DOAG Datenbank Kolumne: Die Gefahr der Routine

<https://www.doag.org/de/home/news/doag-datenbank-kolumne-die-gefahr-der-routine/>



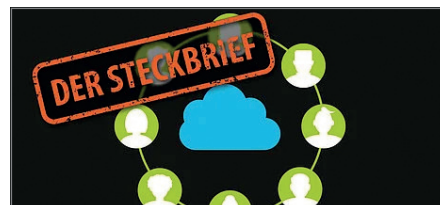
Journey into Cloud ERP am Beispiel Oracle Cloud ERP, Teil 1/5

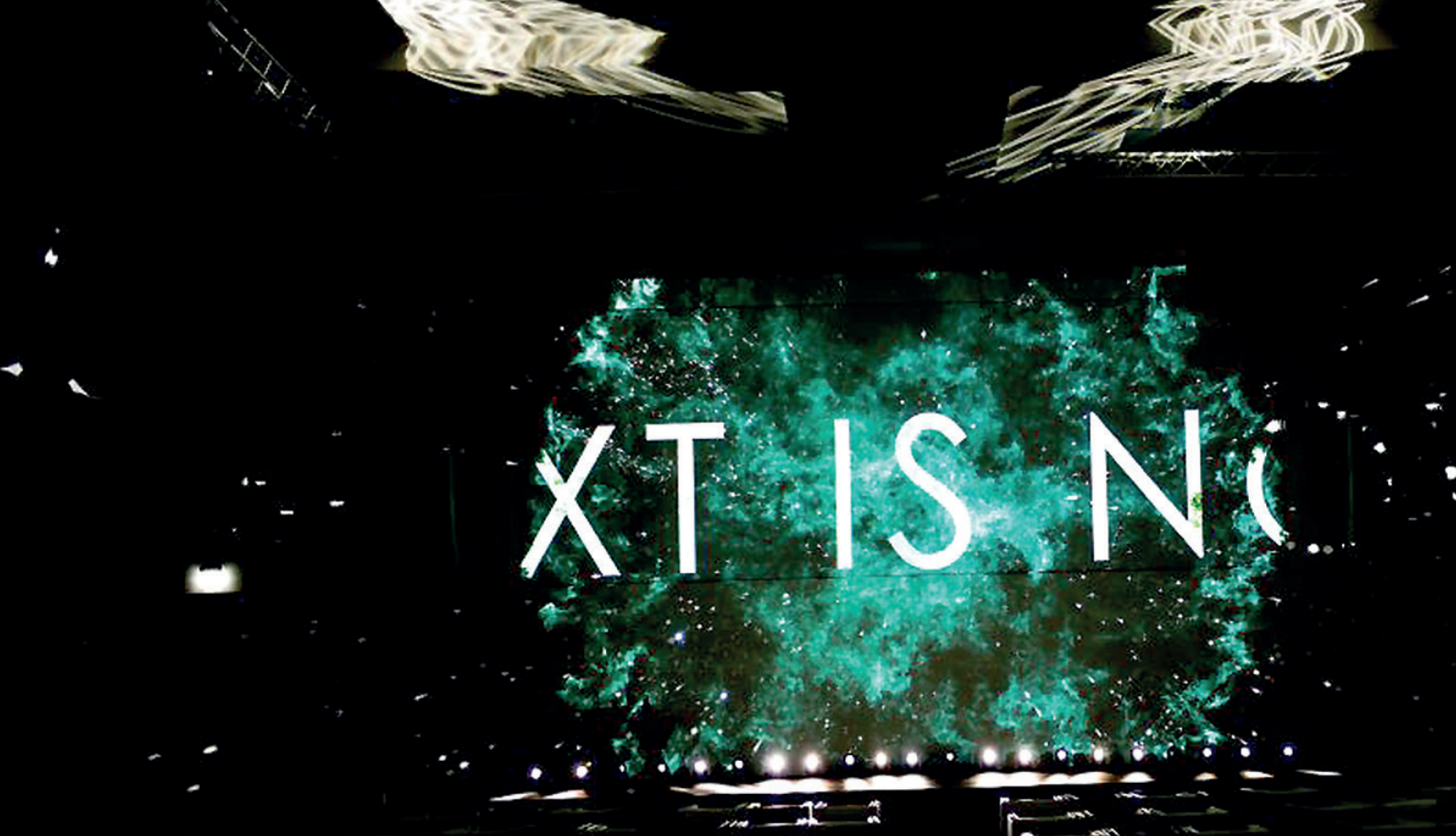
<https://www.doag.org/de/home/news/journey-into-cloud-erp-am-beispiel-oracle-cloud-erp-teil-1-5/>



Der DCNC-Steckbrief – Teil 3: JUG Ingolstadt

<https://www.doag.org/de/home/news/der-dcnc-steckbrief-teil-3-jug-ingolstadt/>





Platinum Events wird Teil der DOAG Dienstleistungen GmbH

Marcos López

Mit der Übernahme des Veranstaltungstechnik-Experten aus NRW vergrößert die DOAG ihr Team und bekommt einen weiteren Standort.



Am 1. Januar 2023 war es so weit. Was sich über Monate anbahnte und auf die äußerst positiven Erfahrungen der vergangenen Jahre, auch mit Blick auf die Krise der Veranstaltungsbranche in den „Corona-Jahren“ 2020/2021, wurde nun Wirklichkeit. Die DOAG Dienstleistungen GmbH, die sowohl für die Realisierung von Events und Konferenzen des DOAG e.V. und des iJUG e.V. als auch für den Betrieb der DOAG-Medien (Zeitschriften, Websites, DOAG.tv, Newsletter, soziale Medien usw.) verantwortlich zeichnet, übernimmt Platinum Events. Die Verträge wurden unterschrieben. Dieser Vorgang ist ein absolutes Novum in der fünfunddreißigjährigen Geschichte der DOAG.



Know-how, Expertise und ein Technik-Park

Das Team um Mark Guy und Kim Nagies, das in ihrer nordrhein-westfälischen Homepage seit über 20 Jahren ein großes Produktportfolio aus dem Veranstaltungsbereich (Planung, Durchführung, Design) anbietet, bringt damit auf einen Schlag nicht nur eine erfolgreiche Expertise ein, die über eine lange Zeit an vorderster Front gewachsen ist – vom Club-Konzert über die große Bühne, von der Betriebsfeier bis zu großen Konferenzen wie JavaLand, DOAG Konferenz + Ausstellung und CloudLand –, sondern auch einen großen Technik-Park. Dieser muss nun zur Durchführung nicht mehr von Drittanbietern abgefragt und eingeholt werden, sondern deckt einen Großteil der benötigten Event-Technik ab, die zentral

– und für alle DOAG-Veranstaltungen! – verfügbar sein wird.

Auch in Zeiten von Lockdown und Remote-Konferenzen fand die DOAG in Platinum Events einen zuverlässigen und hochprofessionellen Partner, der die Herausforderungen von ausschließlich digital durchgeführten Events mit kleinen Projekt- und Moderatoren-Teams für ein großes Online-Publikum aus dem Stand heraus erfolgreich meisterte: Tontechnik, Videoproduktion, Vortragsaufzeichnung und Livestreaming – all das funktionierte reibungslos. Im Lichte der allseits beschworenen Digitalisierung werden diese Bereiche in den kommenden Jahren weiter an Bedeutung gewinnen. Durch den Kauf von Platinum Events sichert sich die DOAG langfristig dieses Know-how und kann nun auch die vorhandene Hardware auf den eigenen Veranstaltungen in vollen Zügen nutzen.



DOAG

PLATINUM

EVENTTECHNIK / CONSULTING / DESIGN





Zwei Standorte, ein Team

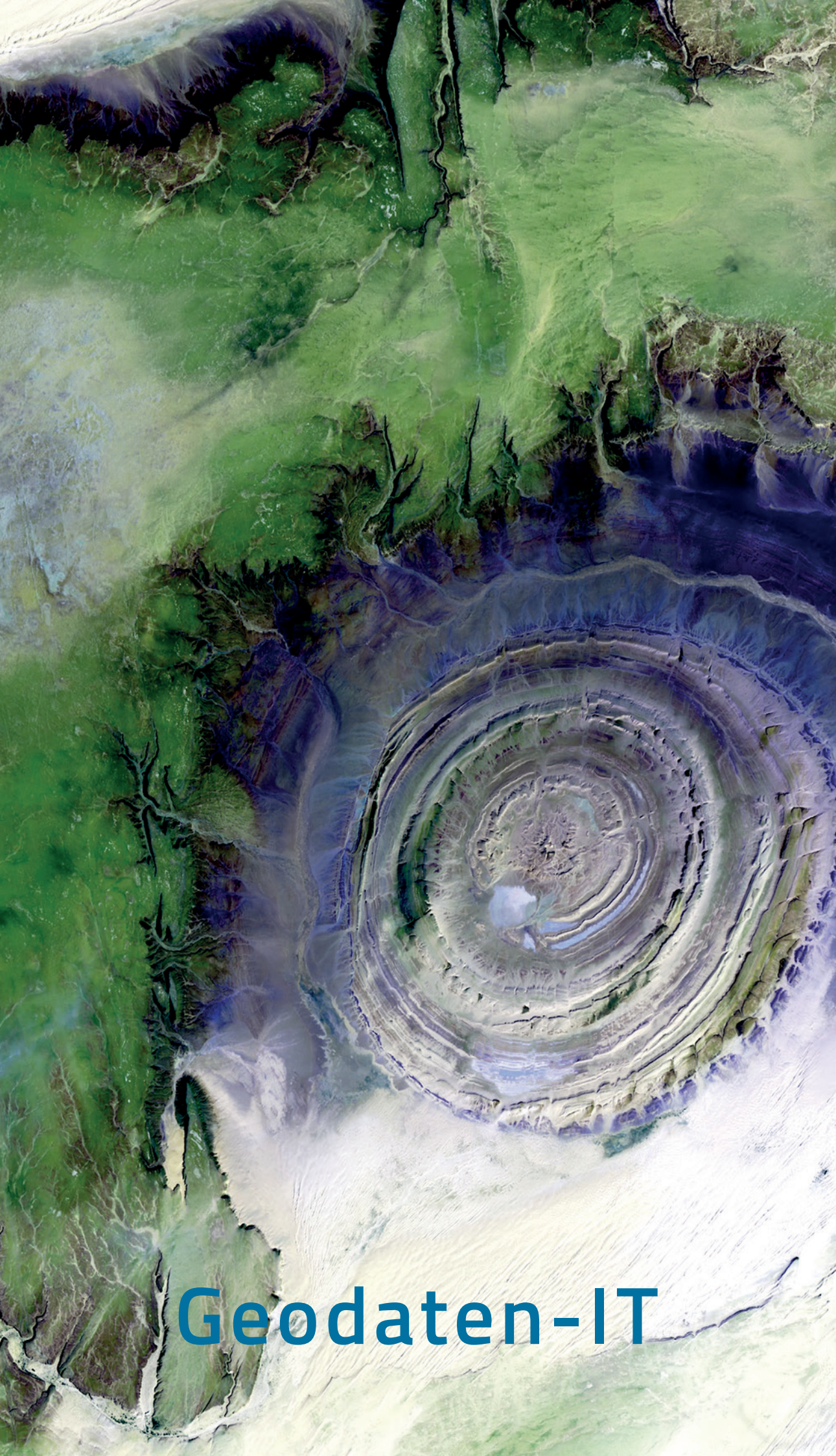
Hinzu kommt auch ein neuer DOAG-Standort in Nordrhein-Westfalen: Hürth. Hier leben und arbeiten die Macher von Platinum Events, die zur Entlastung der Geschäftsstelle in Berlin zukünftig auch den gesamten Lagerbestand der DOAG dort führen werden. Am Standort in der Nähe von Köln werden die kommenden Veranstaltungen entsprechend vorbereitet und ausgeliefert. In regelmäßigen Video-Konferenzen via Teams halten die Standorte Berlin und Hürth Kontakt und justieren so die Bedarfe.

Mark Guy (siehe Foto unten links, gemeinsam mit Fried Saacke, Geschäftsführer DOAG Dienstleistungen GmbH, links im Bild), der bisherige Inhaber von Platinum Events, wird für den Standort Hürth und den Geschäftsbereich Veranstaltungstechnik die Verantwortung übernehmen. Kim Nagies, der als Projektleiter und erfahrener Techniker bisher viele Events von Platinum verantwortet hat, ist dort der zentrale Mitarbeiter. Darüber hinaus hat Platinum Events ein über Jahre vertrauensvoll gewachsenes Netzwerk an freiberuflichen Technikern, die sowohl für die DOAG-Veranstaltungen zur Verfügung stehen werden als auch für zukünftige Kundenprojekte. Dabei bringt Platinum Events eine Reihe von Kunden ein, die ihrerseits interessante Synergien ermöglichen, so dass die DOAG zukünftig im Bereich Veranstaltungstechnik breit am Markt auftreten wird, um den Einsatz der Technik wirtschaftlich zu betreiben. Kim Nagies: „Tolle Sache, dass wir die DOAG nicht mehr ‚nur‘ als Kunden haben, sondern zu einem echten Partner geworden sind. Mit der guten Erfahrung der Vergangenheit können wir nun gemeinsam etwas Neues aufbauen und weitere Bereiche entwickeln.“



BUSINESS — NEWS

NEWS
02/2023



Geodaten-IT

Einfaches Tracking und Tracing auf Basis von Geodaten

Karin Patenge, Oracle

*Wer Nutzungsrechte für die Oracle-Datenbank hat, egal in welcher Edition, hat direkten Zugriff auf umfangreiche Funktionalität innerhalb der Datenbank rund um das Management, die Prozessierung und Analyse sowie zusätzliche Werkzeuge unter anderem für die Visualisierung von Geodaten. Dieser Artikel stellt mit den Themen **Contract Tracing** und **Location Tracking** zwei konkrete fachliche Kontexte für Geodaten vor, die weitreichende Anwendungsmöglichkeiten erschließen.*

Contact Tracing – mittels PL/SQL einfach Kontakte nachverfolgen

Die COVID-19-Pandemie hat alle vor die Herausforderung gestellt, Infektionsketten bestmöglich und zeitnah nachzuverfolgen, das Infektionsrisiko punktgenau einzuschätzen und entsprechende, sinnvolle Maßnahmen daraus abzuleiten. Ein erster Schritt dafür ist, potenzielle Risikokontakte zu ermitteln. Wie das geht mit den Mitteln der Oracle-Datenbank, möchte ich in diesem Kapitel beschreiben.

Seit ungefähr zwei Jahren gibt es das PL/SQL-Paket SDO_OBJ_TRACING, das als Reaktion auf die Notwendigkeit der Kontaktnachverfolgung von mit COVID-19 infizierten Personen entstand. Das Paket ist so implementiert, dass es Bewegungsdaten inklusive ihrer räumlichen und zeitlichen Details – sehr häufig GPS -Daten¹ – hernehmen und analysieren kann. Die Nutzung des Paketes ist sowohl in der MOS Note „Contact Tracing SQL API Added to Oracle Spatial and Graph (Doc ID 2691413.1)“ als auch im Blog Post „Contact Tracing APIs in Oracle Database“ [1] beschrieben. In der Version 21c ist das Paket standardmäßig verfügbar. Für frühere Versionen gibt es das Paket zum Download über die MOS Note weiter oben im Text.

Meine weitere Ausführungen beziehen sich auf Tests mit einer „Always Free Tier“² Autonomous Transactional Processing“-Instanz in der Version 21c.

Das Contact-Tracing-Paket kommt gleich mit zwei APIs daher:

1. Das erste API (Swipe IN/OUT contact tracing) funktioniert ohne GPS-Daten auf der Basis von relational beschriebenen

Informationen zu Orten (z. B. Gebäude, Raum, Flur) und Zuständen. Die Daten dazu können zum Beispiel von Sensoren erhoben und geliefert werden, die an den betreffenden Orten installiert sind.

2. Das zweite API (GPS tracking contact tracing) verwendet Ortsinformationen aus GPS-Tracking-Daten.

Beide APIs benötigen zusätzlich sowohl einen Zeitstempel als auch ein die jeweilige Person oder das Objekt identifizierendes Merkmal (z. B. USER_ID).

Listing 1 beschreibt die sehr einfach gehaltenen Datenmodelle für beide APIs mithilfe von CREATE TABLE Statements.

Das Paket enthält neben anderen Hilfsfunktionen die zwei folgenden Funktionen:

- GET_ALL_SWIPE_IO_DURATIONS für das Swipe IN/OUT contact tracing
- GET_ALL_DURATIONS für das GPS tracking contact tracing

Wie werden diese Funktionen genutzt, welche Analysen werden unterstützt?

Zunächst einmal kann die grundlegende Frage beantwortet werden, welche Personen oder Objekte zur selben Zeit für welche Dauer am gleichen Ort waren. Das Attribut SEGMENT_OR_ALL mit den möglichen Werten SEGMENT oder ALL dient dazu, das Ergebnis auf jede einzelne Begegnung herunterzubrechen (SEGMENT – siehe Listing 2 und Abbildung 2) beziehungsweise in einer weiteren Anfrage alle Kontaktzeiten von zwei Personen (ALL – siehe Listing 3 und Abbildung 3) zu aggregieren.

Durch die flächendeckende Verbreitung und Nutzung von mobilen Geräten mit GPS-Funktion bietet sich ein immenses Potenzial, Bewegungsdaten zu nutzen, die sowohl die räumlichen Informationen in Form von GPS-Koordinaten als auch die Zeitstempel direkt enthalten. Auf dieser Basis können ebenfalls sehr einfach relevante Kontakte ermittelt werden. In Abbildung 4 sind beispielhaft Bewegungsdaten auf einer Karte zu sehen. Der dargestellte Zeitausschnitt (engl.: Timeslider) wurde dabei so gewählt, dass er mit dem Ergebnis der SQL-Anfrage auf ein fiktives Bewegungs-

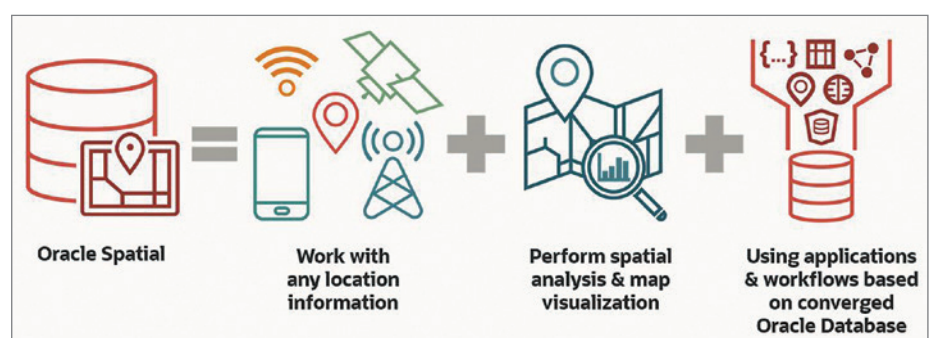


Abbildung 1: Was ist Oracle Spatial? (Quelle: Oracle)

IN_USER_ID	OUT_USER_ID	BUILDING_ID	ROOM_ID	START_TIME	END_TIME	DURATION_IN_MIN
1	2	BUILDING_1	ROOM_1	28-10-2022 13:00:00	28-10-2022 13:30:00	30
1	2	BUILDING_1	ROOM_2	28-10-2022 16:20:00	28-10-2022 16:40:00	20
1	3	BUILDING_1	ROOM_1	28-10-2022 13:00:00	28-10-2022 14:00:00	60

Abbildung 2: Ergebnis der Abfrage aus Listing 2

IN_USER_ID	OUT_USER_ID	START_TIME	END_TIME	DURATION_IN_MIN	NUM_CONTACT_TIMES
1	2	28-10-2022 13:00:00	28-10-2022 16:40:00	50	2
1	3	28-10-2022 13:00:00	28-10-2022 14:00:00	60	1

Abbildung 3: Ergebnis der Abfrage aus Listing 3



Abbildung 4: Räumliche und zeitliche Darstellung von GPS-basierten Bewegungsdaten in Oracle Spatial Studio (Quelle: Oracle)

IN_USER_ID	OUT_USER_ID	DURATION_IN_MINUTES	START_TIME	END_TIME	NUM_CONTACT_TIMES
7	5	0.2	05/31/20 18:53:24	05/31/20 18:53:36	1
7	6	86.08	05/31/20 15:26:34	05/31/20 21:42:57	34
7	8	0.07	05/31/20 14:37:02	05/31/20 14:37:06	1

Abbildung 5: Ergebnis der Abfrage aus Listing 4

GET_ALL_SWIPE_IO_DURATIONS	GET_ALL_DURATIONS
USER_ID (NUMBER)	USER_ID (NUMBER)
START_TIME (DATE)	START_TIME (DATE)
END_TIME (DATE)	END_TIME (DATE)
TRACK_TABLE_NAME (VARCHAR2)	DISTANCE (NUMBER)
USER_ID_COLUMN_NAME (VARCHAR2)	TIME_TOLERANCE_IN_SEC (NUMBER)
BUILDING_ID_COLUMN_NAME (VARCHAR2)	CHAINING_TOLERANCE_IN_SEC (NUMBER)
FLOOR_ID_COLUMN_NAME (VARCHAR2)	TRACK_TABLE_NAME (VARCHAR2)
ROOM_ID_COLUMN_NAME (VARCHAR2)	GEOM_COLUMN_NAME (VARCHAR2)
SWIPE_IO_COLUMN_NAME (VARCHAR2)	USER_ID_COLUMN_NAME (VARCHAR2)
TIME_COLUMN_NAME (VARCHAR2)	TIME_COLUMN_NAME (VARCHAR2)
MUST_MATCH_COLUMNS (SDO_STRING_ARRAY)	DATE_AS_NUMBER_COLUMN_NAME (VARCHAR2)

Tabelle 1: Funktionen im PL/SQL-Paket SDO_OBJ_TRACING für die beiden Contact Tracing APIs

datenset für die Stadt Boston/USA in Listing 4 zusammenfasst. In der Abfrage wird die Funktion GET_ALL_DURATIONS des Contact Tracing API verwendet. Diese Funktion unterscheidet sich in einigen Parametern von der zuvor verwendeten Funktion GET_ALL_SWIPE_IO_DURATIONS. Eine Gegenüberstellung findet ihr deshalb in Tabelle 1.

Der Parameter TIME_TOLERANCE_IN_SEC wird in Abhängigkeit vom Zeitintervall für die Erfassung der Datenpunkte gewählt. Werden die Datenpunkte sekundlich erfasst, dann ist der Standardwert für diesen Parameter 2 (das Doppelte). Der Parameter CHAINING_TOLERANCE_IN_SEC sollte mindestens gleich oder größer dem Wert von TIME_TOLERANCE_IN_SEC sein.

Die Contact Tracing APIs im Paket SDO_OBJ_TRACING sind generisch gehalten. Sie können daher auch für das Nachverfolgen von Begegnungen beispielsweise sich bewegender Objekte genutzt werden, um etwa Fragen zu beantworten wie: Welche Objekte wurden zur gleichen Zeit und auf den gleichen Wegabschnitten an einen anderen Ort verbracht?

Location Tracking - Wo fahren Sie denn?

Eine etwas andere, aber ähnliche Form der Betrachtung räumlicher und zeitlicher Verläufe von Daten ist mit dem Thema Location Tracking verbunden. Hierbei liegt der Fokus darauf, zu jedem Zeitpunkt in der Lage zu sein, den Ort für ein Objekt (oder auch eine Person) zu kennen und in Bezug zu ortsabhängigen Aktionen zu setzen. Das ist zum Beispiel für Speditionsunternehmen wichtig, um zu bestimmen, ob sich deren Fahrzeuge entlang der definierten Routen bewegen oder diese verlassen. Die Verarbeitung der dafür notwendigen Positionsdaten basiert in der Regel auf einem Producer-Consumer-Prinzip, das auch Advanced Queuing benutzt. Verlässt also ein Fahrzeug die vorgeschriebene Route oder passiert das Fahrzeug eine speziell definierte Region (z. B. Grenzüberfahrt), kann eine Nachricht ausgelöst und an die Zentrale verschickt werden.

Das Hineinbewegen in eine solche Region oder auch das Herausbewegen ist eng mit dem englischen Begriff Geofencing verknüpft. Geo und fence zusammen ergeben einen Zaun für ein geografisch definiertes Gebiet.

Das Location Tracking ist ebenfalls direkt in der Datenbank möglich und über das

```

create table swipe_table (
  user_id      number,
  building_id  varchar2(100),
  floor_id     varchar2(100),
  room_id      varchar2(100),
  swipe_in_out varchar2(5),    -- Zustand (IN oder OUT)
  time DATE);

create table osm_tracks_new (
  user_id          number,
  geom             sdo_geometry,
  time_adjusted   date,
  time_adjusted_as_number number);

```

Listing 1: CREATE Table Statements für die Datenmodelle der Contact Tracing APIs

```

select a.in_user_id,
  a.out_user_id,
  a.building_id,
  a.room_id,
  to_char(a.start_time,'DD-MM-YYYY HH24:MI:SS') as start_time,
  to_char(a.end_time,'DD-MM-YYYY HH24:MI:SS') as end_time,
  round(a.duration/60, 2) as duration_in_min
from table (
  sdo_obj_tracing.get_all_swipe_io_durations (
    user_id          => 1,
    start_time       => to_date('28-10-2022 08:00:00','DD-
MM-YYYY HH24:MI:SS'),
    end_time         => to_date('28-10-2022 20:00:00','DD-
MM-YYYY HH24:MI:SS'),
    track_table_name => 'swipe_table',
    user_id_column_name => 'user_id',
    building_id_column_name => 'building_id',
    floor_id_column_name  => 'floor_id',    -- optional
    room_id_column_name   => 'room_id',    -- optional
    swipe_io_column_name  => 'swipe_in_out',
    time_column_name      => 'time')) a    -- optional
where segment_or_all = 'SEGMENT'
order by in_user_id, out_user_id, start_time;

```

Listing 2: Jeder Kontakt von Person mit USER_ID = 1 mit anderen Personen

```

select a.in_user_id,
  a.out_user_id,
  to_char(a.start_time,'DD-MM-YYYY HH24:MI:SS') as start_time,
  to_char(a.end_time,'DD-MM-YYYY HH24:MI:SS') as end_time,
  round(a.duration/60, 2) as duration_in_min,
  a.num_contact_times
from table (
  sdo_obj_tracing.get_all_swipe_io_durations (
    user_id          => 1,
    start_time       => to_date('28-10-2022 08:00:00','DD-
MM-YYYY HH24:MI:SS'),
    end_time         => to_date('28-10-2022 20:00:00','DD-
MM-YYYY HH24:MI:SS'),
    track_table_name => 'swipe_table',
    user_id_column_name => 'user_id',
    building_id_column_name => 'building_id',
    floor_id_column_name  => 'floor_id',    -- optional
    room_id_column_name   => 'room_id',    -- optional
    swipe_io_column_name  => 'swipe_in_out',
    time_column_name      => 'time')) a    -- optional
where segment_or_all = 'ALL'
order by in_user_id, out_user_id, start_time;

```

Listing 3: Summierte Kontaktdauern für Person mit USER_ID = 1 mit anderen Personen

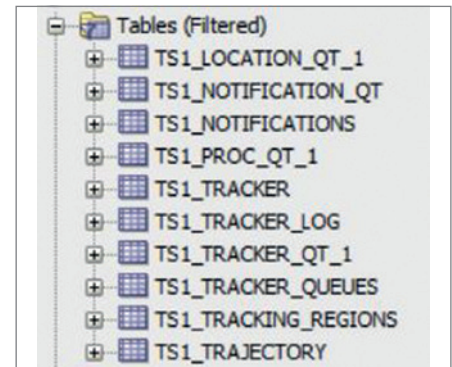


Abbildung 6: Tabellen des Location-Tracker-Servers für ein Tracking-Set

PL/SQL-Paket SDO_TRKR implementiert. Es kann mit jeder Edition der Oracle DB ab Version 12.2 verwendet werden. Das gilt auch für die Autonomous Datenbanken ATP und ADW in den Versionen 19c und 21c. Für die Nutzung des Paketes werden einige zusätzliche Rechte für den DB User benötigt, die mit Advanced Queuing in Zusammenhang stehen. Diese sind in *Listing 5* aufgeführt.

Wie funktioniert das Location Tracking in der Oracle-Datenbank?

Die einzelnen Schritte beschreibe ich nachfolgend ausführlicher mithilfe der jeweiligen Skriptausschnitte. Das gesamte Skript mit allen Schritten und weiteren zusätzlichen Abfragen kann von GitHub [2] heruntergeladen werden.

Ausgangsbasis für das Location Tracking ist ein sogenanntes Tracking-Set. Dieses erzeugt die notwendigen DB-Objekte im Anwenderschema, unter anderem Tabellen für die Verwaltung von Nachrichten oder eine Tabelle zum Speichern der Geofences als SDO_GEOMETRY-Objekte. *Listing 6* zeigt das Anlegen eines solchen Tracking-Sets. Alle für ein Tracking-Set (in meinem Beispiel TS1) angelegten Tabellen sind in *Abbildung 6* zu sehen.

Ist das Tracking-Set definiert, muss es noch aktiviert werden (*siehe Listing 7*).

Danach werden die Regionen als SDO_GEOMETRY-Objekte erzeugt. Im *Listing 8* ist eine solche Region innerhalb von Deutschland als einfaches Rechteck mit fünf Stützpunkten definiert.

Hinweis:

Das komplette Skript auf GitHub nimmt eine kleine Änderung zu den Skriptauszügen hier in diesem Artikel vor. Anstelle der Oracle-eigenen Koordinatensystem-ID 8307 für das von GPS verwendete Koordinatensystem WGS 84 soll die von der EPSG³

vergebene Koordinatensystem-ID 4326 [3] verwendet werden. Das hat zur Folge, dass sowohl die Oracle-Spatial-spezifischen Metadaten (USER_SDO_GEOM_METADATA) anzupassen als auch der räumliche Index neu zu erzeugen sind.

Im nächsten Schritt wird festgelegt, wann für welche Objekte eine Nachricht ausgelöst und in eine Queue geschrieben wird. Im Beispiel in *Listing 9* sollen Nachrichten immer dann generiert und in Queue 1 geschrieben werden, wenn:

- Objekt 1 sich in Region 1 hineinbewegt („I“)
- Objekt 2 sich aus Region 1 herausbewegt („O“)

Nunmehr sind alle Voraussetzungen erfüllt, um Positionsdaten zu empfangen und auf diese zu reagieren, wenn die Bedingungen hinreichend erfüllt sind. *Listing 10* zeigt acht eingehende Nachrichten zu Positionen der zwei Objekte mit den IDs 1 und 2. Die Parameter drei und vier von MDSYS.LOCATION_MSG übergeben jeweils die Position als X- (Längengrad) und Y- (Breitengrad) Koordinate.

Im nächsten Schritt werden die Positionsdaten ausgewertet und Nachrichten erzeugt, wenn Objekt 1 sich in die Region hinein oder Objekt 2 sich aus der Region heraus bewegt hat. Dieser Vorgang wird auch Dequeueing genannt (*siehe Listing 11*).

Die Nachrichten werden in die Tabelle TS1_NOTIFICATIONS geschrieben (*siehe Listing 12*).

Visuell kann das Ergebnis nun leicht anhand der *Abbildung 7* überprüft werden. Die Kartendarstellung wurde ebenfalls mithilfe von Spatial Studio erzeugt. Sie zeigt den Geofence (grün) und genau die Positionen der beiden Objekte, die Benachrichtigungen auslösen. Für Objekt 1 (rot) sind es die Positionen innerhalb, für Objekt 2 (blau) die Positionen außerhalb vom Geofence.

Nach Bedarf können jederzeit weitere Geofences (Regionen) registriert, die Regeln für das Auslösen von Nachrichten geändert oder weitere Queues hinzugefügt werden. Sollen zum Beispiel für das Objekt 1 in Bezug auf die Region 1 keine Benachrichtigungen mehr ausgelöst werden, dann kann diese Zuordnung sehr einfach disabled werden (*siehe Listing 13*).

Im Sinne der Vollständigkeit möchte ich noch auf die beiden Prozeduren SDO_TRKR.STOP_TRACKING_SET sowie SDO_TRKR.

```
select a.in_user_id
      ,a.out_user_id
      ,round(a.duration/60,2) duration_in_minutes
      ,to_char(a.start_time,'MM/DD/YY HH24:MI:SS') start_time
      ,to_char(a.end_time, 'MM/DD/YY HH24:MI:SS') end_time
      ,a.num_contact_times
from table(sdo_obj_tracing.get_all_durations(
  user_id           => 7,
  start_time        => to_date('31-MAY-2020 00.00.03','DD-
MON-YYYY HH24.MI.SS'),
  end_time          => to_date('31-MAY-2020 23.59.54','DD-
MON-YYYY HH24.MI.SS'),
  distance          => 15,
  time_tolerance_in_sec => 5,
  chaining_tolerance_in_sec => 20,
  track_table_name  => 'osm_tracks_new',
  geom_column_name  => 'geom',
  user_id_column_name => 'user_id',
  time_column_name  => 'time_adjusted',
  date_as_number_column_name => 'time_adjusted_as_number')) a
where a.segment_or_all = 'ALL'
order by a.in_user_id, a.out_user_id, a.start_time;
```

Listing 4: Summierte Kontaktdauern für Person mit USER_ID = 7 mit anderen Personen

```
-- Local DB or DBCS: Ausführen als SYS oder SYSTEM
-- Autonomous DB: Ausführen als ADMIN
grant aq_administrator_role, create job, manage scheduler to testuser;
grant execute on dbms_aq to testuser;
grant execute on dbms_aqadm to testuser;
grant execute on dbms_lock to testuser;
grant execute on dbms_aqin to testuser;
grant execute on dbms_aqjms to testuser;
grant select_catalog_role to testuser;
```

Listing 5: Notwendige Rechte für den DB User

```
begin
  sdo_trkr.create_tracking_set(
    'ts1'
    , 1      -- Anzahl der Queues für die Tracking messages
    , 1);   -- Anzahl der Queues für die Location messages
end;
/
```

Listing 6: Anlegen eines Tracking-Sets

```
exec sdo_trkr.start_tracking_set('ts1');

-- Ergebnis überprüfen
-- Queues, die vom Tracking-Set verwendet werden
select name
from user_queues
where name like 'TS1%'
order by name;
-- Scheduler Jobs, die vom Tracking-Set verwendet werden
select job_name, state
from user_scheduler_jobs
where job_name like 'TS%'
order by job_name;
```

Listing 7: Starten des Tracking-Sets und Überprüfen der Queues

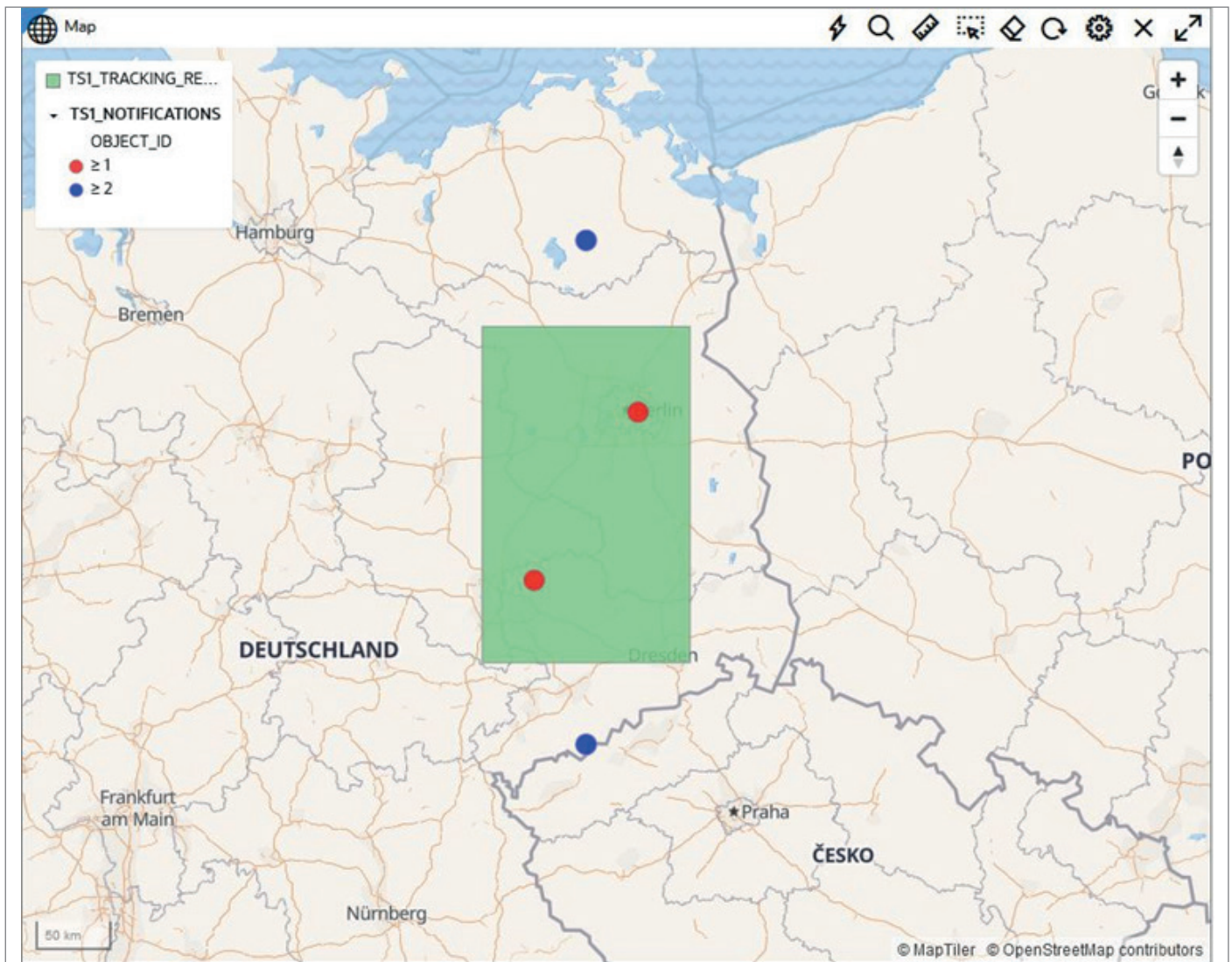


Abbildung 7: Kartendarstellung in Spatial Studio für die Notifications mit Bezug zum Geofence (Quelle: MapTiler / OpenStreetMap contributors)

DROP_TRACKING_SET hinweisen. Anhand der Namensgebung lässt sich leicht erkennen, wofür diese verwendet werden, nämlich entweder zum Deaktivieren oder zum Löschen eines Tracking-Sets.

Fazit

Im Artikel habe ich einen Teilbereich der Arbeit mit Geodaten genauer beleuchtet, der die Entwicklung von Applikationen für spannende und zugleich gängige Anwendungsfälle unterstützt. Die vollständige Integration der beschriebenen Funktionalität in die Oracle-Datenbank vereinfacht sowohl den Betrieb und die Administration als auch die Absicherung der Anwendungen mit den gängigen Werkzeugen beziehungsweise Sicherungsmechanismen der Datenbank. Die Nutzung der beschriebenen Funktionen erfordert weder zusätzliche Lizenzgebühren noch Cloud Credits, sondern bietet die Chance, fachliche Anforder-

```
insert into tsl_tracking_regions values (
  1,
  mdsys.sdo_geometry(
    2003,
    8307,
    null,
    sdo_elem_info_array(1, 1003, 1),
    sdo_ordinate_array(
      12,51,      -- 1. Stützpunkt
      14,51,
      14,53,
      12,53,
      12,51)); -- Letzter Stützpunkt identisch mit 1.
commit;
```

Listing 8: Registrieren einer Region

```
exec sdo_trkr.send_tracking_msg('TSL1', mdsys.tracker_msg(1, 1, 'I'));
exec sdo_trkr.send_tracking_msg('TSL1', mdsys.tracker_msg(2, 1, 'O'));

-- Überprüfen
select object_id, region_id, alert_when
from tsl_tracker;
```

Listing 9: Definition, wann Nachrichten verschickt werden

```

begin
  sdo_trkr.send_location_msgs (
    'TS1',
    mdsys.location_msg_arr(
      mdsys.location_msg(1,CURRENT_TIMESTAMP(),11.5,50.5),
      mdsys.location_msg(1,CURRENT_TIMESTAMP()+1,12.5,51.5),
      mdsys.location_msg(1,CURRENT_TIMESTAMP()+2,13.5,52.5),
      mdsys.location_msg(1,CURRENT_TIMESTAMP()+3,14.5,53.5),
      mdsys.location_msg(2,CURRENT_TIMESTAMP(),13,50.5),
      mdsys.location_msg(2,CURRENT_TIMESTAMP()+1,13,51.5),
      mdsys.location_msg(2,CURRENT_TIMESTAMP()+2,13,52.5),
      mdsys.location_msg(2,CURRENT_TIMESTAMP()+3,13,53.5));
end;
/

```

Listing 10: Verschicken der Positionsdaten

```

declare
  message mdsys.notification_msg;
begin
  loop
    sdo_trkr.get_notification_msg(
      tracking_set_name => 'TS1',
      message => message,
      deq_wait =>2); -- Min. 2 Sekunden warten
    if (message is null) then
      exit;
    end if;
    insert into tsl_notifications (
      object_id, region_id, time, x, y, state)
      (select
        message.object_id,
        message.region_id,
        message.time,
        message.x,
        message.y,
        message.state
        from sys.dual);
  end loop;
end;

```

Listing 11: Dequeuing der Positionsdaten

```

select object_id, region_id, x, y, state
from tsl_notifications
order by object_id, time;

```

-- Ergebnis:

OBJECT_ID	REGION_ID	X	Y	STATE
1	1	12.5	51.5	INSIDE
1	1	13.5	52.5	INSIDE
2	1	13	50.5	OUTSIDE
2	1	13	53.5	OUTSIDE

Listing 12: Notifications-Tabelle mit den erzeugten Nachrichten

```

exec sdo_trkr.send_tracking_msg(
  'TS1',
  mdsys.tracker_msg(1, 1, 'D'));

```

Listing 13: Notifications Tabelle mit den erzeugten Nachrichten

derungen einfach umzusetzen, ohne „das Rad noch mal neu zu erfinden“.

Quellen

- [1] Contact Tracing APIs in Oracle Database, <https://blogs.oracle.com/oraclespatial/post/contact-tracing-apis-in-oracle-database>
- [2] GitHub Repository für Location Tracking, https://github.com/karinpatenge/location_tracker
- [3] WGS 84, <https://epsg.io/4326>

Dokumentation

- Oracle Database 19c Dokumentation – Spatial, <https://docs.oracle.com/en/data-base/oracle/oracle-database/19/spatial.html>
- Oracle Database 21c Dokumentation – Spatial, <https://docs.oracle.com/en/data-base/oracle/oracle-database/21/spatial.html>

Über die Autorin

Karin Patenge ist als Produktmanagerin verantwortlich für die Themenbereiche Graph- und Spatial-Technologien im Bereich der Oracle-Datenbank-Produktentwicklung. Sie arbeitet seit 2007 für Oracle in Deutschland.

Karin hat sowohl Informatik als auch Geoinformatik an Universitäten in Rumänien, Deutschland und Österreich studiert. Sie hat einen Abschluss als Dipl.-Informatikerin und ist seit über 30 Jahren in unterschiedlichen IT-Rollen für verschiedene Unternehmen in Deutschland tätig.



Karin Patenge

karin.patenge@oracle.com

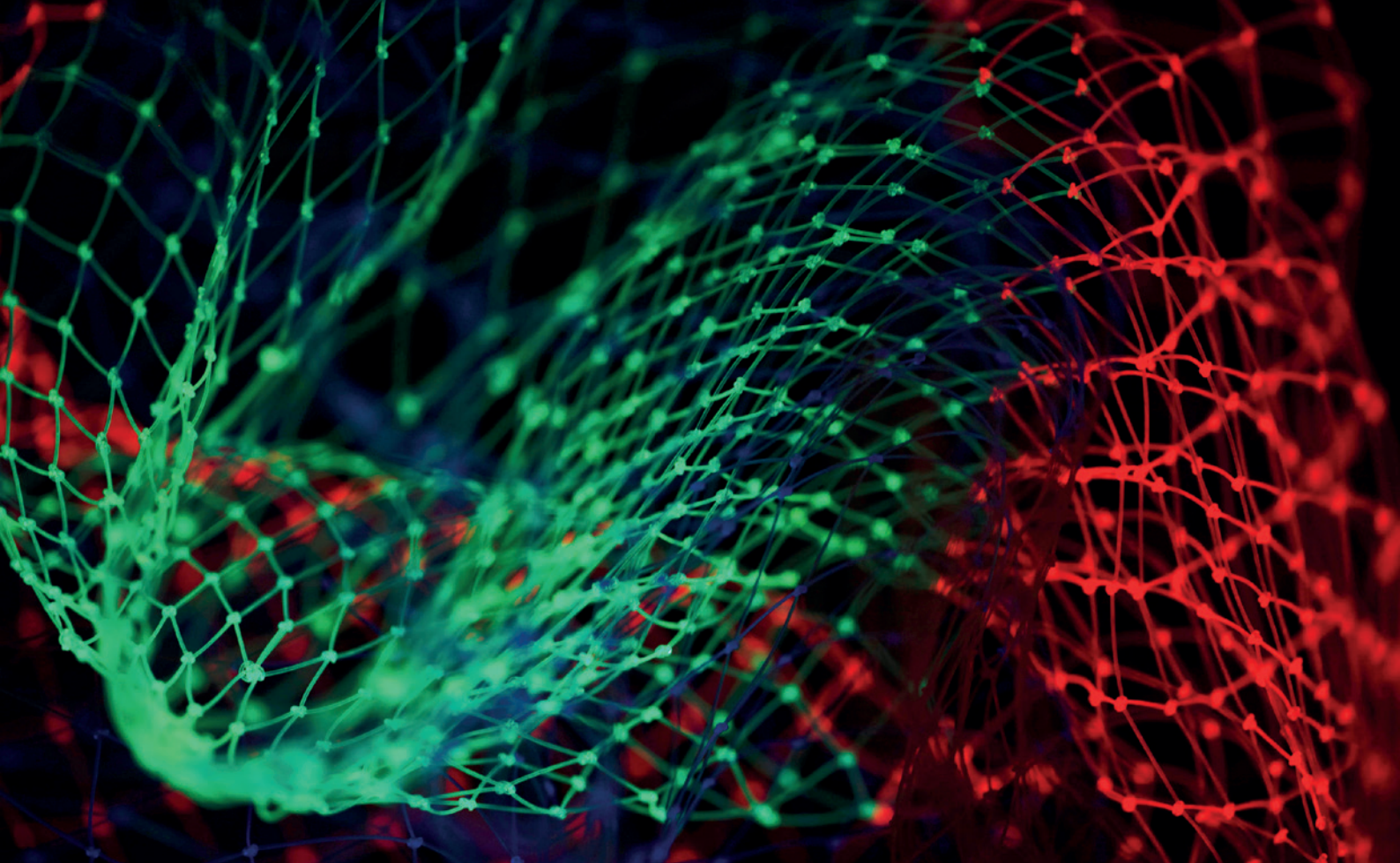
Karin Patenge ist als Produktmanagerin verantwortlich für die Themenbereiche Graph- und Spatial-Technologien im Bereich der Oracle-Datenbank-Produktentwicklung. Sie arbeitet seit 2007 für Oracle in Deutschland.

Karin hat sowohl Informatik als auch Geoinformatik an Universitäten in Rumänien, Deutschland und Österreich studiert. Sie hat einen Abschluss als Dipl.-Informatikerin und ist seit über 30 Jahren in unterschiedlichen IT-Rollen für verschiedene Unternehmen in Deutschland tätig.

¹GPS = Global Positioning System

²OCI Cloud Free Tier (<https://www.oracle.com/cloud/free/>)

³ European Petroleum Survey Group



Die fünf spannendsten Oracle Spatial Features

Sven Bosinger, DOAG Themenverantwortlicher Geodaten und Locational Intelligence

Oracle Spatial ist eine Option der Oracle-Datenbank. Seit der Version 12.2 ist Oracle Spatial kostenfrei nutzbar. Sie steht damit allen Cloud Database Services und On-Premises-Angeboten von Oracle zur Verfügung. Mit Oracle Spatial können räumliche Zusammenhänge von Daten analysiert und als Kartenmaterial präsentiert werden. Man spricht dabei von sogenannten Geo-Daten. Im folgenden Artikel werden die meines Erachtens fünf spannendsten Features beleuchtet und an Beispielen ihre Einsatzmöglichkeit dargelegt. Dabei ist es nicht Ziel, Oracle Spatial im Ganzen zu erklären, dies entnimmt der Leser bitte der mannigfaltigen verfügbaren Dokumentation. Lediglich einige technische Besonderheiten werden zum besseren Verständnis kurz dargelegt.

Mit Oracle Spatial hat Oracle einen eigenen, für die Repräsentation von Geometrien geeigneten Datentyp, SDO_GEOMETRY, eingeführt.

In diesem Datentyp werden geometrische Objekte als Punktwolken gespeichert, das heißt als Paare von X/Y/(Z)-Koordinaten. Dabei gibt es drei grundsätzliche Geometrie-Typen: Punkte, Linien und Polygone.

Punkte werden in der Komponente SDO_POINT als X/Y-Werte abgelegt. Dabei wird in SDO_GTYPE durch die Angabe 2001 festge-

legt, dass es sich um einen 2-dimensionalen Punkt handelt, die erste Stelle der Zahl ist hier ausschlaggebend. Die 1 an letzter Stelle sagt aus, dass es sich um einen Punkt handelt. 3001 gibt folglich einen 3-dimensionalen Punkt an, dieser wird also durch eine X/Y/Z-Koordinate beschrieben. Die Z-Koordinate könnte bei einer topologischen Sicht zum Beispiel die Höhe des Punktes über dem Meeresspiegel beinhalten.

Handelt es sich um die Geometrie-Typen Linie oder Polygon (SDO_GTYPE = 2002

oder 2003), so werden die Punkte, die die Geometrie-Objekte repräsentieren, in dem Array SDO_ORDINATES als Abfolge von X/Y-Koordinaten gespeichert. Das Array SDO_ELEM_INFO beinhaltet Metadaten zu dem Array SDO_ORDINATES und dient der näheren Beschreibung der dort abgelegten Koordinaten. Auch hier sind 2- oder 3-dimensionale Geometrien möglich.

Die Spalte SDO_SRID gibt das Koordinatensystem an, in dem die einzelnen Punkte abgelegt werden. Es gibt eine Vielzahl von

```
CREATE TYPE SDO_GEOMETRY AS OBJECT
  (SDO_GTYPE      NUMBER
  ,SDO_SRID       NUMBER
  ,SDO_POINT      SDO_POINT_TYPE
  ,SDO_ELEM_INFO  SDO_ELEM_INFO_ARRAY
  ,SDO_ORDINATES  SDO_ORDINATE_ARRAY
  );
```

Listing 1: SDO_GEOMETRY – Definition des Datentyps

```
CREATE INDEX <Index-Name>
ON <Tabellen-Name>(<Spalten-Name>)
INDEXTYPE IS MDSYS.SPATIAL_INDEX_V2;
```

Listing 2: Create Spatial-Index – Anlegen eines Spatial-Index auf einer Tabellen-Spalte

```
SELECT <Geo-Spalten-Name>
FROM <Tabellen-Name>
WHERE SDO_RELATE(<Geo-Spalten-Name>
  ,SDO_GEOMETRY(2003
    ,4258
    ,NULL
    ,SDO_ELEM_INFO_ARRAY(1, 2, 1)
    ,SDO_ORDINATE_ARRAY
      (<X_oben_links>
      ,<Y_oben_links>
      ,<X_unten_rechts>
      ,<Y_unten_rechts>
    )
  )
  , 'mask=anyinteract'
) = 'TRUE';
```

Listing 3: Zooming – Selektion aller Objekte innerhalb eines Rechtecks bei Variation desselben

```
SELECT SDO_LRS.LOCATE_PT(geom_segment => <Geo-Spalten-Name>
  ,measure => 104
  )
FROM <Tabellen-Name>;
```

Listing 4: SDO_LRS.LOCATE_PT – X/Y-Koordinate auf einer Strecke an Kilometer 104

```
new_geometrie := SDO_GEOM.SDO_BUFFER
  (geom => <Ursprungsgeometrie>
  ,dist => 50
  ,tol => 0.05
  ,params => 'unit=m arc_tolerance=0.05'
  );
```

Listing 5: SDO_GEOM.SDO_BUFFER – Puffern eines Polygons um 50 Meter

```
schnitt_geometrie := SDO_LRS.LRS_INTERSECTION
  (geom_1 => <Geometrie-Umring>
  ,geom_2 => <Streckenlinie_A7>
  ,tolerance => 0.05
  );
```

Listing 6: SDO_LRS.LRS_INTERSECTION – Schnittgebilde zwischen Umring und A7

unterschiedlichen Koordinatensystemen für unterschiedliche Zwecke. Die folgenden Beispiele sind alle im Koordinatensystem ETRS89 (SDO_SRID = 4258) gespeichert.

Die Speicherung von Geo-Daten wird durch den Datentyp SDO_GEOMETRIE unterstützt. Um mit den Daten zu arbeiten, das heißt sie zur Anzeige zu bringen, oder mit analytischen Funktionen auf diese zuzugreifen, stellt Oracle eine große Zahl von Prozeduren, Funktionen und Paketen bereit. Aus dieser Vielzahl werden im Folgenden fünf aufgegriffen, die häufig wiederkehrende Aufgabestellungen unterstützen.

Zoomen in einer Karte

Soll der Inhalt ein oder mehrerer Tabellen mit Geometrie-Daten zur Anzeige gebracht werden, so können hierfür verschiedene Werkzeuge, auch von Drittanbietern, eingesetzt werden. Ein eingängiges Beispiel ist die kartenbasierte Navigationsmöglichkeit in Fahrzeugen.

Allen gemeinsam ist, dass sie eine sogenannte Zoom-Funktion haben, das heißt, der Maßstab der angezeigten Karte kann verändert werden. Das Zoomen wird dabei in der Regel so realisiert, dass der Kartenausschnitt, also das Rechteck auf der Karte, das zur Anzeige gebracht wird, verändert wird. Dies hat zur Folge, dass die Menge der angezeigten Geo-Objekte variiert und vom gewählten Kartenausschnitt abhängig ist. Schlussendlich lässt sich das Problem des Zoomens in Oracle Spatial dadurch lösen, dass ein Kartenausschnitt definiert wird und alle Objekte innerhalb dieses Ausschnitts als Ergebnismenge eines SELECT-Statements dem Oberflächenwerkzeug zur Verfügung gestellt werden.

Zuallererst ist es notwendig auf der Tabellen-Spalte des Datentyps SDO_GEOMETRY einen sogenannten Spatial-Index anzulegen; ohne diesen Index ist die Funktion SDO_RELATE, die für diese Operation genutzt werden soll, nicht einsetzbar. Wie der Name schon sagt, handelt es sich hier um eine Möglichkeit, den Inhalt der Geometrie-Spalte zu indizieren. Wie dieser Index funktioniert, kann der Oracle-Dokumentation entnommen werden. Es soll nur erwähnt werden, dass er eine völlig andere Struktur hat als Indizes, die typischerweise auf skalaren Datentypen (NUMBER, VARCHAR2, DATE ...) angelegt werden. Der Index wird mit folgendem Befehl erzeugt.

Ein Rechteck kann in Oracle-Spatial dadurch erzeugt werden, dass seine obo-

re rechte und untere linke X/Y-Koordinate angegeben wird. Durch Veränderung dieser Koordinaten kann das Rechteck entweder verschoben und/oder in seiner Größe (Zoom-Faktor bzw. Maßstab) verändert werden.

Wendet man nun die Oracle-Funktion SDO_RELATE auf eine Geometrie-Spalte unter Zuhilfenahme des oben definierten Rechtecks an, so kann man über ein SELECT-Statement alle Geo-Objekte selektieren, die sich innerhalb dieses Rechtecks befinden.

Über den Parameter 'mask=...' kann dabei gesteuert werden, ob sich zum Beispiel das Objekt komplett im Rechteck befinden muss ('mask=contains') oder wie in diesem Fall irgendeine Überlappung oder Berührung vorhanden sein muss. Auch hier gibt die Oracle-Dokumentation Aufschluss über weiteren Möglichkeiten.

Über die Variation der beiden X/Y-Koordinaten kann nun die Ergebnismenge des SELECT-Statements leicht verändert beziehungsweise die Zoom-Funktion realisiert werden

Streckenpositionen

Ein weiteres sehr entscheidendes Konzept bei Geo-Daten, das in Oracle-Spatial implementiert wurde, ist das Line Referencing System (LRS).

Dazu muss man erst einmal betrachten, wie eine Linie in Oracle-Spatial abgebildet wird. Gespeichert wird eine Linie als eine Folge aufsteigender oder absteigender Punkte im Datentyp SDO_GEOMETRY unter dem SDO_GTYPE = 2002.

Die einfachste Form ist dabei eine Gerade, die durch ihren Anfangs- und Endpunkt beschrieben wird.

Wird die Linie nun komplexer, erhöht sich die Anzahl der Punkte, die diese Linie beschreibt. Diese Punkte werden auch Stützstellen genannt.

Wählt man die Abstände zwischen den Stützstellen beliebig klein, so kann man über dieses Verfahren Kreise und Bögen und damit sehr komplexe Linienverläufe beschreiben.

Der Unterschied zwischen einer Linie und einer Strecke ist, dass eine Strecke gerichtet ist und dass die einzelnen Stützstellen mit einem sogenannten Kilometerwert zum Beispiel in Metern versehen sind. Man spricht dann von einer LRS-Geometrie.

Bei einer LRS-Geometrie wird die Kilometerierungsinformation in der Z-Koordi-

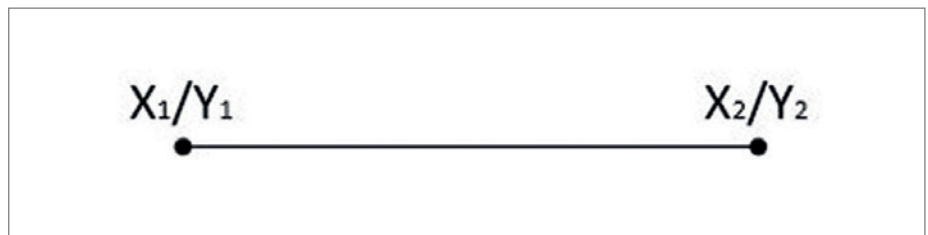


Abbildung 1: Gerade mit Anfangs- und Endpunkt (Quelle: Sven Bosinger)

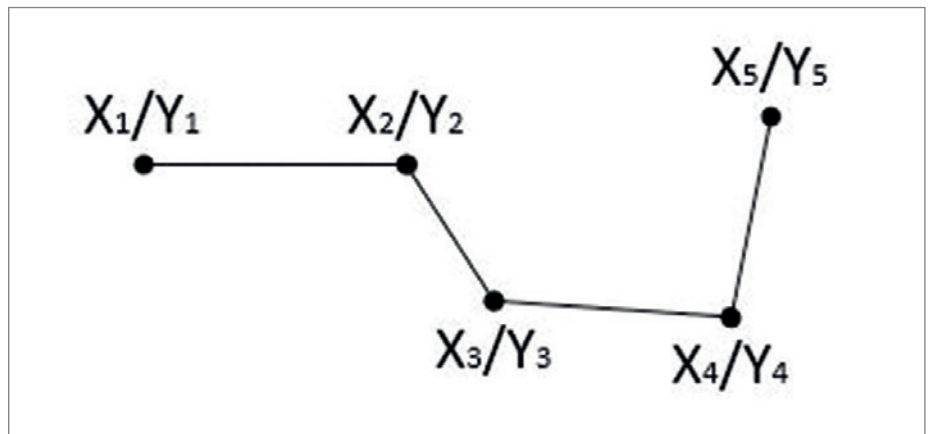


Abbildung 2: Linie mit Stützstellen (Quelle: Sven Bosinger)

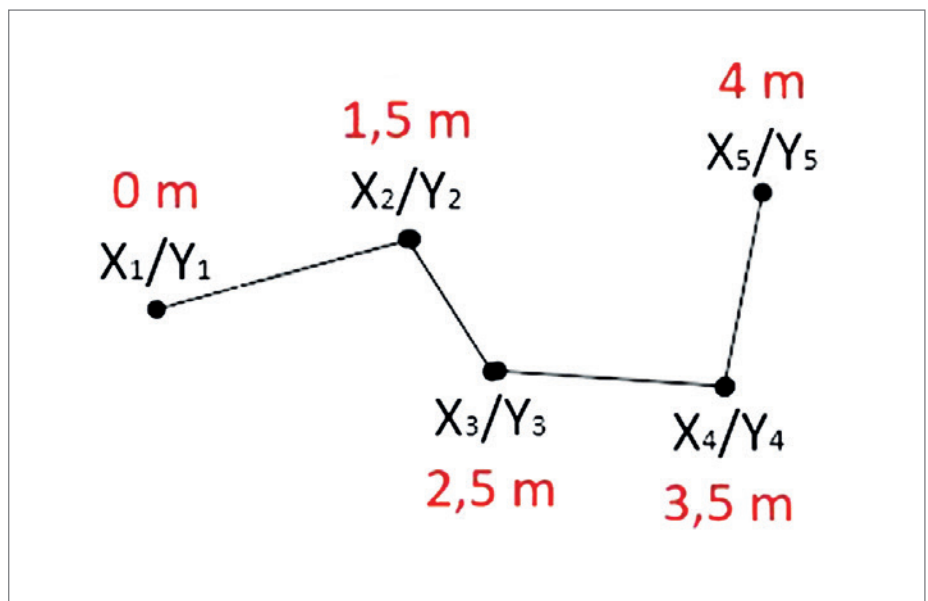


Abbildung 3: Strecke als LRS-Geometrie (Quelle: Sven Bosinger)

```

km_A := SDO_LRS.GET_MEASURE
      (point => SDO_LRS.GEOM_SEGMENT_START_PT
       (geom_segment => <schnitt_geometrie>
        )
       );
km_B := SDO_LRS.GET_MEASURE
      (point => SDO_LRS.GEOM_SEGMENT_END_PT
       (geom_segment => <schnitt_geometrie>
        )
       );
    
```

Listing 7: SDO_LRS.GET_MEASURE – Kilometerwerte von Punkt A und B

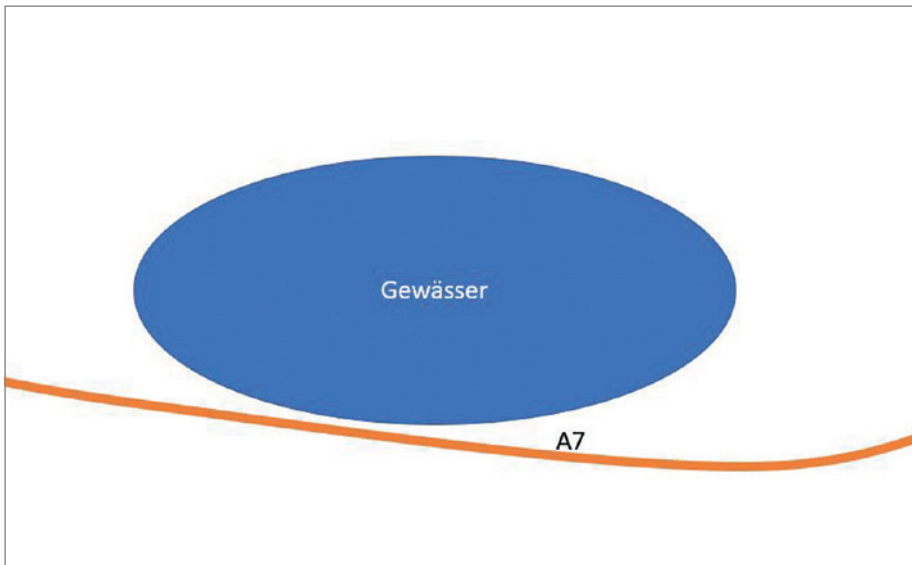


Abbildung 4: Autobahn entlang eines Gewässers (Quelle: Sven Bosinger)

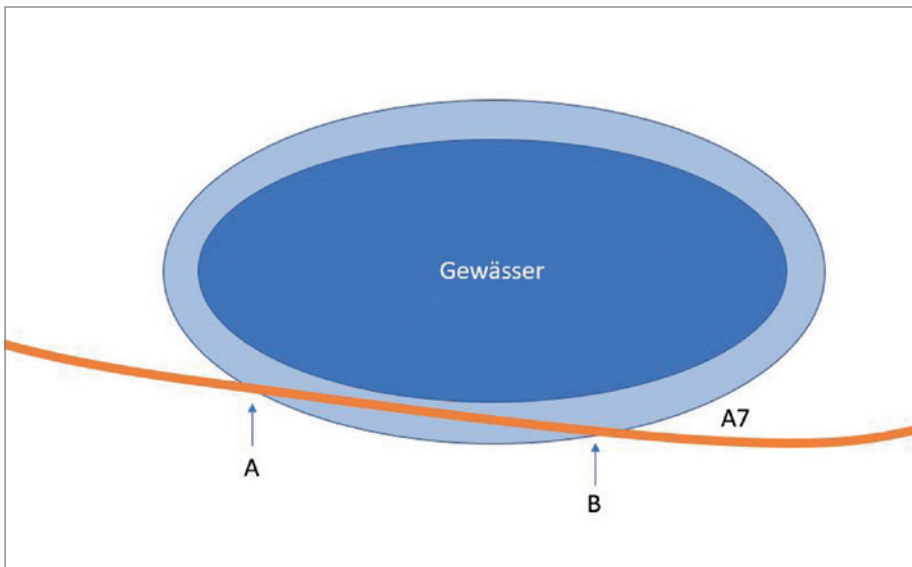


Abbildung 5: Autobahn durchquert den Umring des Gewässers (Quelle: Sven Bosinger)

```
geo_korrigiert := SDO_UTIL.RECTIFY_GEOMETRY
                (geometry => <Original_Geometry>
                , tolerance => 0.05
                );
```

Listing 8: SDO_UTIL.RECTIFY_GEOMETRY – Korrektur einer fehlerhaften Geometrie

```
SELECT    geo.id
          , geo.x
          , geo.y
FROM      <Tabellen-Name> tab
          , TABLE(SDO_UTIL.GETVERTICES
                  (tab.<Geometrie-Spalte>
                  )
          ) geo
WHERE     tab.<PK-Spalte> = <PK-Wert>
ORDER BY geo.id;
```

Listing 9: SDO_UTIL.GETVERTICES – Ausgabe aller Stützstellen einer Geometrie

nate der jeweiligen Stützstellen gespeichert. Ein Beispiel für eine LRS-Geometrie in der realen Welt ist das deutsche Autobahnnetz, bei dem neben der Autobahnnummer (z. B. A7) über den Kilometrierungswert (z. B. Autobahnkilometer 104) jeder Punkt einer Autobahn eindeutig identifiziert werden kann. Man findet die Schilder mit den Kilometerangaben entlang jeder Autobahn.

Mit der Funktion SDO_LRS.LOCATE_PT kann man sich zu einer vorhandenen LRS-Geometrie unter Angabe des Kilometerwertes die X/Y-Koordinate eines Punktes ausgeben lassen.

Dabei muss der angegebene Kilometerwert nicht dem genauen Wert einer Stützstelle entsprechen. Liegt er zwischen zwei Stützstellen, so wird seine X/Y-Koordinate interpoliert. Als Ergebnistyp wird SDO_GEOMETRY zurückgegeben.

Mit dieser Funktion bin ich in der Lage, innerhalb einer Karte eine Position auf eine Strecke eindeutig zu finden. Es handelt sich im weitesten Sinne um eine Suchfunktion.

Geometrien puffern

Möchte man herausfinden, ob eine Strecke (z. B. eine Autobahn) in einem bestimmten Abstand zu einem anderen Geo-Objekt (z. B. einem Gewässer) an diesem vorbeiführt und wenn ja auf welcher Länge, um etwa zu entscheiden, ob man auf diesem Abschnitt Streusalz einsetzen darf, dann gibt es über die Funktion SDO_GEOM.SDO_BUFFER hier eine elegante Lösung.

Man kann das Gewässer mit einem sogenannten Umring (Puffer) versehen. Das heißt, das Gewässer wird um eine definierte Größe, zum Beispiel 50 Meter, in jede Richtung vergrößert.

Die Situation sieht dann etwa wie folgt aus.

Man sieht die Stelle, an der die A7 in den Umring (Puffer) des Gewässers bei Punkt A eintritt und ihn bei Punkt B wieder verlässt.

Man kann nun leicht den Teil der A7 durch die Funktion SDO_LRS.LRS_INTERSECTION ermitteln, der innerhalb des Umrings (Puffer) des Gewässers liegt, also die Teilstrecke von Punkt A nach Punkt B.

Hat man diese Teilstrecke ermittelt, so kann man mit den Funktionen SDO_LRS.GEOM_SEGMENT_START_PT und GEOM_SEGMENT_END_PT die Geometrien der Punkte A und B ermitteln. Setzt man auf dem jeweiligen Ergebnis noch die Funktion SDO_LRS.GET_MEASURE ein, so be-

kommt man die Kilometerwerte der Punkte A und B aus unserem Beispiel.

So könnte dann die fachliche Aussage, die auf Basis einer Analyse der gespeicherten Geo-Daten erfolgt ist, lauten: „Auf der A7 darf von Kilometer 104 bis Kilometer 107 aufgrund eines Abstandes von weniger als 50 Metern zu einem Gewässer kein Streusalz ausgebracht werden!“

Geometrie „reparieren“

Oracle Spatial ist relative großzügig, was alles technisch gesehen in dem Datentyp SDO_GEOMETRY gespeichert werden darf. Die Oberflächenwerkzeuge, die auf diesen Datentyp zugreifen, zum Teil aber nicht. Um den Austausch von Geodaten zwischen verschiedenen Systemen zu gewährleisten, gibt es das Open Geospatial Consortium (OGC). Dieses hat das Ziel, Standards zum Zweck der Interoperabilität festzulegen. Oracle stellt im Rahmen des Oracle Spatial eine Funktion zur Verfügung, die überprüft, ob eine angegebene Geometrie valide ist, das heißt im weitesten Sinne den Vorgaben des OGC genügt oder nicht. Diese Funktion lautet SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT.

Dabei wird etwa überprüft, ob zwei oder mehrere Punkte, die eine Geometrie (Linie oder Polygon) beschreiben, einen Mindestabstand besitzen, der über einen Toleranzwert vorgegeben wird. Ist dies nicht der Fall, so spricht man von Duplicate Vertices, die es laut OGC nicht geben darf. Auch gibt es bei einem Polygon mit einem Loch, zum Beispiel einem Donut, eine Vorgabe, dass die äußere Grenze durch Punkte, die entgegen dem Uhrzeigersinn aufgetragen werden, beschrieben wird und das Loch durch Punkte im Uhrzeigersinn aufgetragen wird. Daneben wird noch eine ganze Reihe weiterer Prüfungen durchgeführt, die man der Oracle-Dokumentation entnehmen kann.

Ergibt die Funktion VALIDATE_GEOMETRY_WITH_CONTEXT, dass es sich um eine fehlerhafte Geometrie handelt, so stellt Oracle Spatial eine weitere Funktion zur Verfügung, die diese Geometrie zu „reparieren“ versucht.

Mit SDO_UTIL.RECTIFY_GEOMETRY ist es möglich, beispielsweise Duplicate Vertices zu erkennen und automatisch zu beseitigen.

Die Funktion ist dabei in der Lage, folgenden Fehler zu erkennen und zu korrigieren:

1. Doppelte Stützstellen (Duplicate Vertices), also Stützstellen, die enger zusam-

menliegen als über den Toleranzwert angegeben.

2. Schnitte eines Polygons mit sich selbst, Beispiel hierfür ist ein Polygon in 8-Form. Dieses wird in zwei kreisförmige einzelne Polygone aufgeteilt.
3. Falschherum aufgetragenen innere und äußere Grenzen bei einem Polygon mit Loch (Donut).

Die Punktwolke

Im Rahmen der vorherigen Funktionen ist immer wieder erwähnt worden, dass in dem Datentyp SDO_GEOMETRY jegliche Repräsentation von Geometrien durch Punkte, dargestellt durch X/Y-Koordinaten, erfolgt. Jeder der darstellbaren Geometrie-Objekte (Punkt, Linie, Polygon) wird durch einen oder mehrere Punkte beschrieben. Die Anzahl der benötigten Punkte nimmt mit der Komplexität der gespeicherten Linie oder des Polygons zu. So gibt es Polygone, die durch mehr als eine Million einzelner Stützstellen (Punkte) dargestellt werden. Ein Beispiel hierfür sind Grenzverläufe von Ländern oder Bundesländern.

Manchmal ist es notwendig auf eine dieser Stützstellen direkt zuzugreifen oder sich alle Punkte eines Polygons in der korrekten Reihenfolge ausgeben zu lassen. Hierfür hat Oracle die Table-Function SDO_UTIL.GETVERTICES entwickelt. Mit ihr kann man die Stützstellen einer Geometrie in Tabellenform mithilfe eines SELECT-Statements ausgeben.

Das obige SELECT-Statement gibt für die angegebene Geometrie alle X/Y-Koordinaten in der Reihenfolge aus, wie sie im Array SDO_ORDINATES eingetragen sind.

Zusammenfassung

Oracle Spatial stellt eine hervorragende Lösung dar, um Geo-Daten in einer Oracle-Datenbank zu speichern. Über verschiedenste Funktionen und Pakete finden sich „Helferlein“ für fast alle denkbaren Fragestellungen und Anwendungen. Dadurch, dass die gesamte Spatial-Funktionalität in SQL abgebildet ist, kann man Oracle Spatial ohne Medienbruch mit allen anderen SQL-Funktionen der Oracle-Datenbank kombinieren. Dabei kann man die volle Kraft und Skalierungsfähigkeit, die die Oracle-Datenbank besitzt, nutzen. Ab Oracle 21c ist Oracle Spatial sogar mit der InMemory-Option der Datenbank nutzbar. Darüber hinaus wird ab Oracle12.2 die komplette Funkti-

onalität ohne weitere Lizenzkosten zur Verfügung gestellt.

Quellen

- [1] Oracle-Dokumentation „Datenbank Oracle 12.2“ und nachfolgende



Sven Bosinger

sven.bosinger@gmx.de

Der Autor ist als aktives Mitglied der Data Analytics Community (DAC) der DOAG Themenverantwortlicher für Oracle Spatial und Geodaten.



Digitale Modernisierung in 2 Phasen – Einführung von NetSuite und Oracle Integration Cloud als Herzstücke digitaler Prozessinfrastruktur

Michael Schäfer, Integrata Cegos GmbH und Matthias Sauer, PROMATIS software GmbH

Der Austausch eines ERP-Systems ist immer eine Herausforderung. Die vollständige Erneuerung der IT-Landschaft eines Unternehmens inklusive ERP-System ist dahingegen schon ein episches Unterfangen. Integrata Cegos ist dies, in einem Zeitraum von über zwei Jahren, im Projekt Smart Data gelungen. Als zentraler Taktgeber treibt NetSuite nun nicht nur Finance-, sondern auch Einkaufs-, Auftragsabwicklungs-, Projektmanagement- und CRM-Prozesse an und ist über die Oracle Integration Cloud nahtlos mit der ebenfalls im Rahmen des Projekts runderneuernten Learning Management Software zur Durchführung des kompletten operativen Geschäfts integriert.

Seit fast 60 Jahren unterstützt Integrata Cegos Unternehmen mit praxisbezogenen Weiterbildungen für Fach- und Führungskräfte bei der Mitarbeiterentwicklung. Integrata Cegos begleitet dabei von der Konzeption bis zur Umsetzung eines Qualifizierungskonzepts und bildet Fach- und Führungskräfte mit modernen Methoden und innovativen Lernformaten weiter, die als Live Online Training, Hybridformat oder als Präsenzveranstaltung durchgeführt werden. Mit ihren passgenauen Lernangeboten schafft Integrata Cegos ein einzigartiges Lernerlebnis.

Diese Einzigartigkeit findet ihren Niederschlag in einer Vielfalt von Geschäftsprozessen, die mit der Modernisierung der kompletten IT-Landschaft digital abgebildet werden.

Geschäftsprozesse als Treiber für die neue Systemarchitektur

Für Integrata Cegos stehen Customer Journey und Learner Experience immer im Zentrum aller geschäftlichen Aktivitäten – aus denen sich die zentralen Anforderungen für die Umsetzung der Prozesse (insbesondere in puncto Digitalisierung der Service Delivery) und entscheidender Impact auf die Ausgestaltung der beteiligten Systeme ergeben haben. Für die Kunden der Integrata Cegos stellen Nutzen und Vorteile zum Beispiel elaborierter Online-Lernformate einen integralen Bestandteil ihrer eigenen Weiterbildungsplanungen dar.

Das Geschäftsmodell der Integrata Cegos ist vor diesem Hintergrund mit einer Prozessarchitektur unterlegt, die im Wesentlichen auf der auftragsartbezogenen Ausdifferenzierung von Datentransaktionen in folgenden Tätigkeitsfeldern fußt:

- Offene Seminare / Open Courses (OC): Virtual Classroom Trainings oder im Präsenzformat
- Inhouse-Seminare (IS): Standardseminare, deren Durchführungstermine und Veranstaltungsorte und -formate mit den Auftraggeber individuell vereinbart werden
- Qualifizierungsprojekte (QP): Kundenspezifisch entwickelte Trainings sowie Re-/Up-Skilling-Projekte, Consulting zu allen Themen aus dem Portfolio der Integrata Cegos
- Managed Training Services (MTS): Dienstleistungen zur Abwicklung von Kundenprozessen der Wertschöpfungs-

kette Training (Business Process Outsourcing Training)

Abbildung 1 zeigt die grobe Übersicht über die Prozessarchitektur, die als Basis für die Umsetzung des hier beschriebenen Projekts Smart Data bei Integrata Cegos diente. Die farbig markierten Prozesse stellen die Kernprozesse dar. Die weiteren Prozesse sind als Management- oder Supportprozesse in dem jeweils entsprechenden grauen Bereich klassifiziert. Die vorliegende Prozessarchitektur wurde auf Basis der drei zentralen Standardkomponenten Oracle NetSuite (ERP- und CRM-System), cimoio (Training-Management-System) und der Oracle Integration Cloud (Komponente zur Integration aller beteiligten Systeme) umgesetzt. Ein erster Artikel, der die organisatorischen Herausforderungen in Corona-Zeiten bei diesem umfassenden Projekt und die dazugehörigen Lösungen beschreibt, wurde bereits in den Business News 01/2022 veröffentlicht [1]. Der hier vorliegende Artikel beschreibt nun die gewählten Wege zur Umsetzung des Gesamtsystems und fokussiert auf Architekturfragen, prozessuale Aspekte und eingesetzte Technologien.

Die Integration von Oracle NetSuite als ERP/CRM-Kern mit dem Training-Management-System cimoio über die Oracle Integration Cloud (OIC) ermöglicht die komplette geschäftsfeldunabhängige Administration der zentralen operativen Prozesse und stellt den gesamten benötigten Funktionsraum sowohl für die Kern- als auch für die Supportprozesse zur Verfügung:

- Workflowgesteuerte Termin- und Teilnehmerverwaltung inklusive Ressourcenmanagement im Training-Management-System cimoio
- Austausch auftragsrelevanter Transaktionsdaten zwischen NetSuite und cimoio via Web Services über die Oracle Integration Cloud
- Vendor Management über das cimoio-Trainerportal (Web Frontend), u. a. als zentrale Informations- und Kommunikationsplattform
- Anbindung weiterer externer (Kunden-) Systeme für die Training und Service Delivery
- Digitale Bereitstellung und Verarbeitung von Teilnehmerfeedbacks als Inputgeber für das Qualitätsmanagementsystem in cimoio

- Vertragsmanagement, Beschaffung, Finanz- und Rechnungswesen inklusive Debitoren- und Kreditorenmanagement in NetSuite
- Geschäftsfeldübergreifende KPI-Analysen für Finance und Controlling in NetSuite und der Oracle Planning & Budgeting Cloud

Das zugrunde liegende Prozessdesign wurde flexibel skalierbar entwickelt, um einer dynamischen Kundenorientierung und einem weiteren Scope-Ausbau zu genügen. Operationalisierung und Controlling der gesamten Value Chain Trainings wurden mithilfe digitaler End2End-Verarbeitungen in beiden Systemen miteinander verbunden. Die existierenden Prozesse wurden im Rahmen der Transformation im Kern nicht notwendigerweise neu designed, vielmehr wurden die systemisch zu implementierenden Abläufe sowie die Prozessierung der vorgangsrelevanten Daten gemäß dem Prinzip „Zero Touch“ in die Zieltopologie der IT-Landschaft digitalisiert eingebettet. Die Optimierung der Datenstrukturen und ein digitaler Prozessausbau dienten im Projekt als Garantien für Standardisierung und Normierung sowie Automatisierung der Sollprozesse. Weitere Determinanten für die prozedurale Datenintegrität resultierten aus Projekten, in denen Kundensysteme die Funktion sowohl als Datenquelle als auch als Datenkonsument einnehmen.

Phase 1: Einführung von NetSuite ERP und CRM

Microsoft Dynamics NAV war zuvor seit mehr als zehn Jahren bei Integrata Cegos im Einsatz. Das System war hochgradig an die Integrata-Cegos-Geschäftsprozesse angepasst. Insbesondere Prozesse aus den Bereichen CRM, Delivery und Finance waren integraler Bestandteil des Systems. Der zweite große Geschäftsbereich Open Courses (OC) wurde in diesem Zeitraum hauptsächlich in den Learning-Management-Plattformen Ecadia und dem proprietären System SVS abgewickelt; lediglich die Rechnungsstellung erfolgte über Dynamics NAV. Phase 1 des Projekts Smart Data umfasste somit die Ablösung von Dynamics NAV mit seinen CRM- und Finance-Prozessen durch NetSuite und die temporäre Integration von NetSuite mit den bestehenden On-Premises-Legacy-Systemen über die Oracle Integration Cloud.

Prozessarchitektur Smart Data

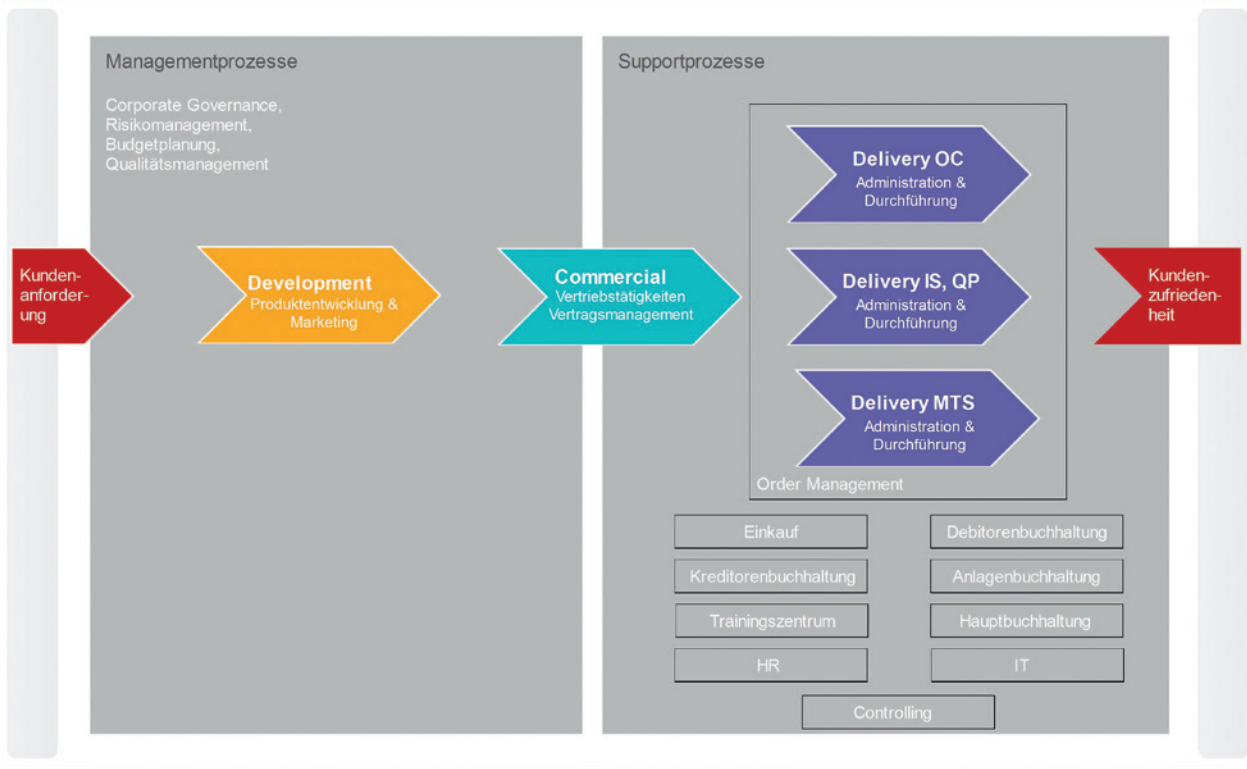


Abbildung 1: Prozessarchitektur Projekt Smart Data (© Integrata Cegos GmbH)

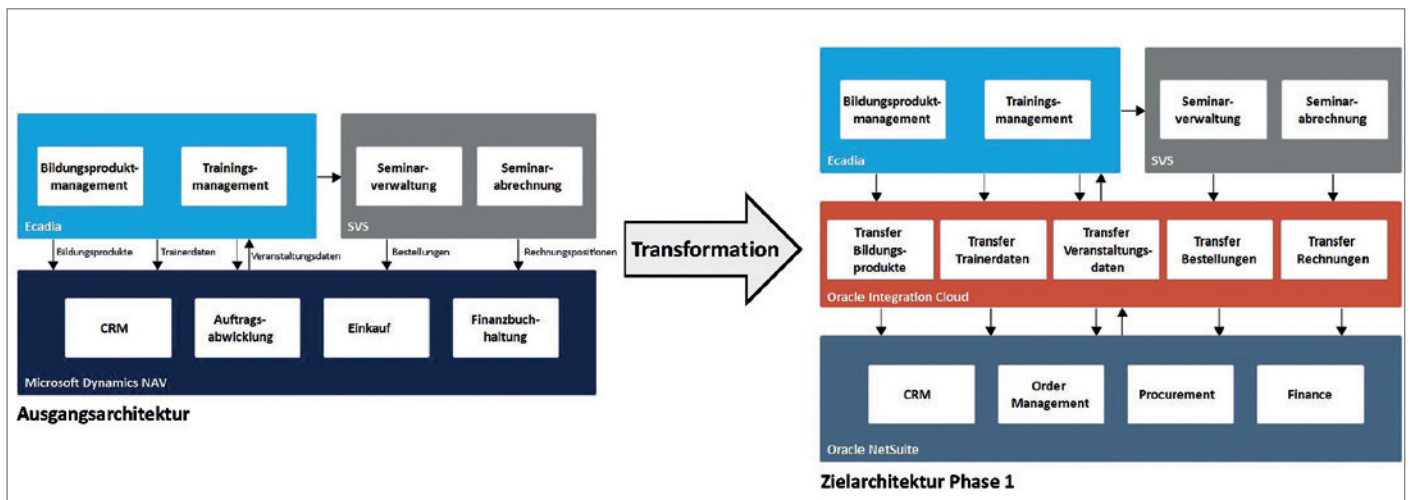


Abbildung 2: Architekturtransformation Phase 1 (© Integrata Cegos GmbH)

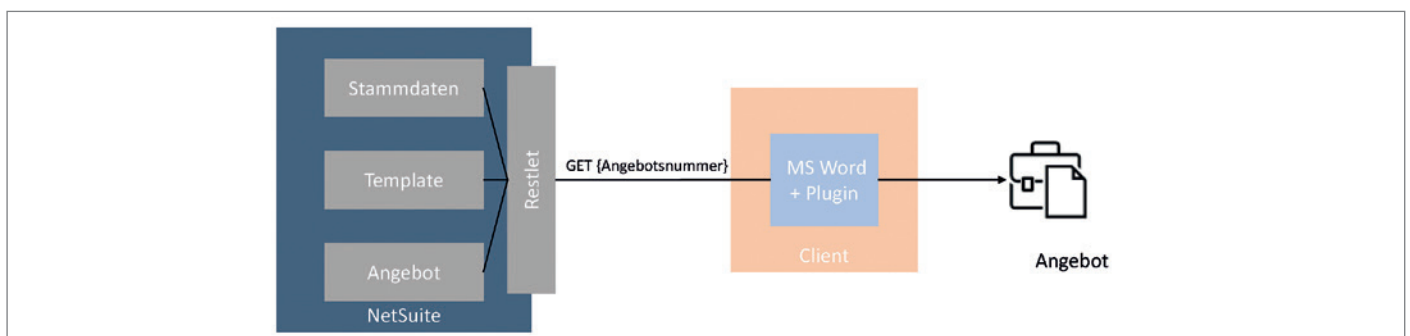


Abbildung 3: Word Plug-in für die Einbindung von NetSuite-Daten in Word-Angebote (© PROMATIS GmbH)

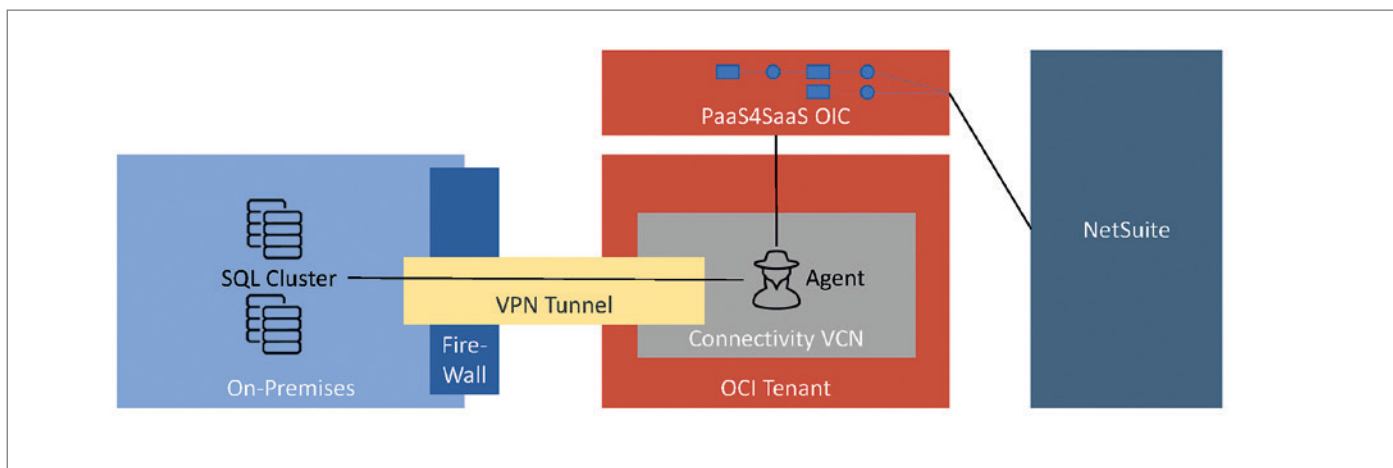


Abbildung 4: OCI-gehosteter Connectivity Agent zur Einbindung der On-Premises-Systeme (© PROMATIS GmbH)

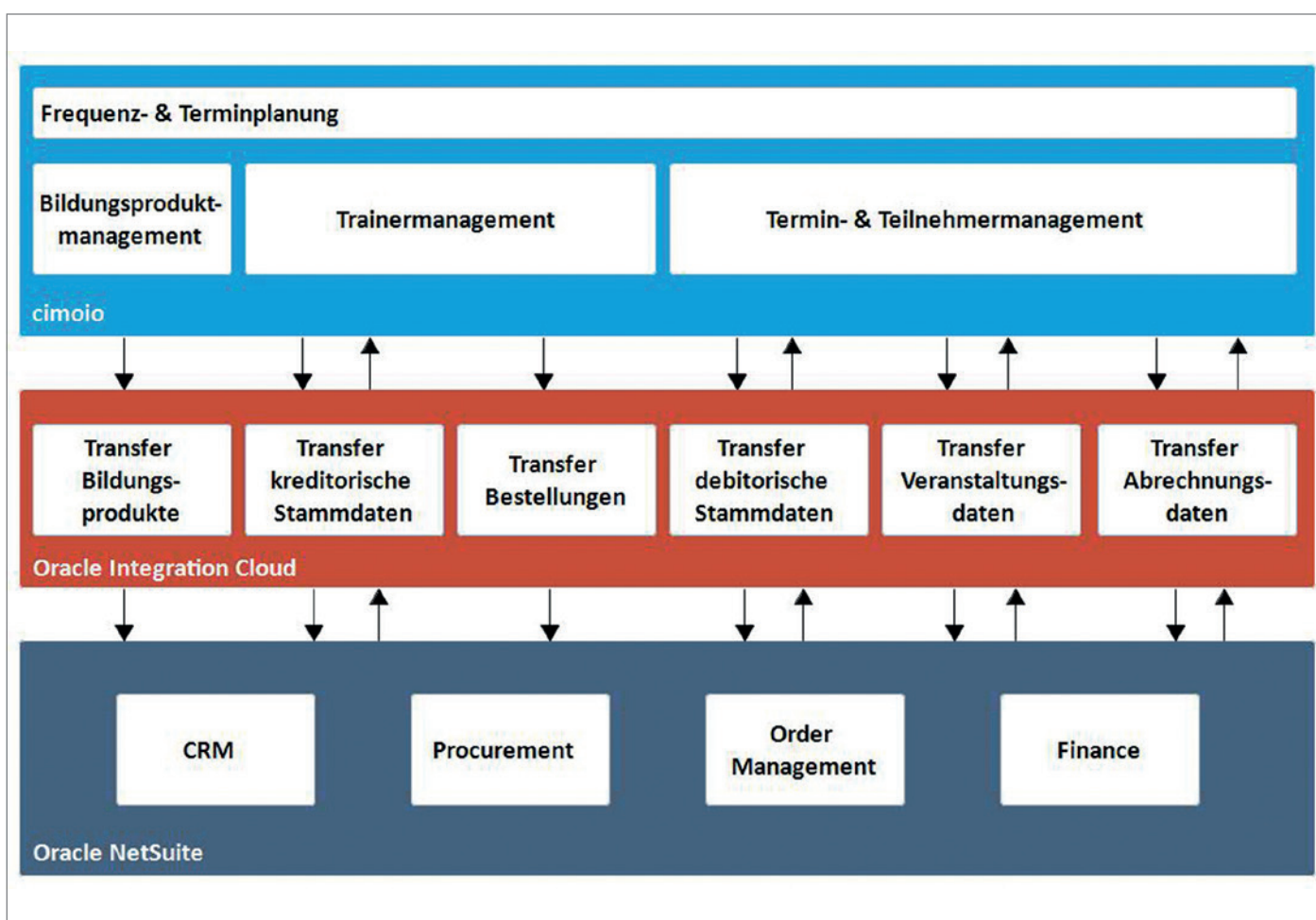


Abbildung 5: Zielarchitektur Phase 2 (© Integrata Cegos GmbH)

Mit der Ablösung von Microsoft Dynamics NAV übernahm NetSuite den vollständigen Lead-to-Cash-Prozess für den Bereich des nicht-öffentlichen Trainingsgeschäfts. In NetSuite CRM wurden ab diesem Zeitpunkt Leads erfasst und Angebote erstellt. Da bei Integrata Cegos Angebote im Allgemeinen kundenspezifisch erstellt werden müssen, wurde ein eigens dafür entwi-

ckeltes Word Plug-in integriert, um auf dem Client PC des Benutzers eine Word-Datei erzeugen zu können, die alle relevanten Kunden- und Artikelstammdaten enthält und die darüber hinaus vom Benutzer beliebig angepasst werden kann [2].

Gewonnene Verkaufschancen werden ab der Phase 1 in NetSuite mit der Standardfunktionalität als Aufträge erzeugt und an

das Order Management Team übergeben. Das Order Management Team verarbeitet hier, über lange Zeiträume hinweg, Aufträge und Bestellungen mit weit über hundert Auftragszeilen. Jeder als Auftragszeile angelegte Seminartermin wird synchron mittels der OIC über eine REST-Schnittstelle an das Training-Management-System übermittelt. Die Rechnungsstellung und die de-

bitorische Weiterverarbeitung werden in NetSuite durchgeführt.

Der OC-Prozess beginnt meist mit einer Buchung durch den Kunden selbst auf der Website und wurde in Phase 1 dann noch über das eigenentwickelte Seminarverwaltungssystem SVS abgewickelt. Damit SVS nach der Umstellung auf NetSuite weiter korrekt funktioniert, mussten Bestellungen und Abrechnungspositionen an NetSuite übermittelt sowie Seminarstammdaten aus Ecadia abgerufen werden. Da SVS in der nächsten Projektphase abgelöst wurde, wurden in Phase 1 keine REST-Schnittstellen entwickelt, sondern ein direkter OIC-Zugriff auf den MS SQL Servercluster umgesetzt. Die On-Premises-Verbindung wurde dabei über den Connectivity Agent der Oracle Integration Cloud hergestellt. Allerdings bestand dabei die Herausforderung, dass keine weiteren Ressourcen im eigenen Rechenzentrum bereitgestellt werden konnten. Die Lösung dafür war die Integration eines OCI Virtual Cloud Network in das Integrata Cegos VPN. Innerhalb dieses VCN wird der Connectivity Agent auf einer virtuellen Maschine bereitgestellt.

Um die volle bisherige Funktionalität aus Microsoft Dynamics NAV abbilden zu können, musste nun noch die Auftragsverarbeitung für Managed Training Services (MTS) in NetSuite integriert werden. Die Buchungsabwicklung der MTS findet zum Großteil auf kundeneigenen Systeminstanzen des Training-Management-Systems statt. Um diese einbinden zu können, wurde mit der OIC eine generische Auftrags- und Bestellschnittstelle für diese Kundensysteminstanzen umgesetzt. Eine weitere Besonderheit des MTS-Prozesses sind die Providerkarten, die abrechnungsrelevante Informationen enthalten und jedem MTS-Kunden zugeordnet werden. Die Providerkarten wurden als Custom Records in NetSuite aufgesetzt und an den Kunden hinterlegt. Ein MTS-Auftrag führt somit automatisiert zu einer Trainerbestellung, erzeugt die korrekte Preisfindung und kann danach als Einzel- oder Sammelrechnung in NetSuite abgerechnet werden.

Phase 2: Modernisierung der operativen Learning Management Software

In Phase 2 wurde das Training-Management-System Ecadia auf die aktuelle, unter neuem Namen und mit deutlich erweitertem Funktionsumfang verfügbare Software

cimoio von STL umgestellt. Die Funktionalitäten der noch in Phase 1 verwendeten, eigenentwickelten Seminarverwaltungs- und Abrechnungssoftware SVS wurden nun komplett mit cimoio- und NetSuite-Funktionalitäten sowie der dazugehörigen Integration über die Oracle Integration Cloud umgesetzt. Darüber hinaus wurden viele neue Funktionen der aktualisierten Learning Management Software wie beispielsweise eine Frequenz- und Terminplanung für die Veranstaltungen, ein Trainerportal und zahlreiche Automatisierungen bei der Abwicklung der Aufträge der dargestellten Geschäftsfelder von Integrata Cegos durch das entsprechend zugeschnittene Zusammenspiel der verwendeten Komponenten realisiert.

Weiterer Ausbau in einem kontinuierlichen Verbesserungsprozess (KVP)

Mit der im Projekt aufgebauten komponentenbasierten Prozess- und Systemarchitektur wurde neben der Umsetzung der fachlichen Anforderungen eine Gesamtlösung geschaffen, die sich in einfacher Weise strukturiert weiter ausbauen lässt. Dadurch können für die naturgemäß weiter entstehenden dienstleistungsbezogenen Verfahrensadaptation und kundenindividuellen Service-Szenarien „Datennebenflussarme“

entstehen, die im Zuge von Ausnahmebehandlungen teilautomatisiert gelenkt werden müssen.

Hierzu wurde ein kontinuierlicher Verbesserungsprozess (KVP) etabliert, um die beiden Standard-Systeme funktional und prozessual dynamisch auf die Belange der Kunden und die Entwicklung des Geschäftsmodells zu adaptieren. Und damit auch die Prozess- und Systemstruktur weiterzuentwickeln, die das „digitale Unternehmen“ – verstanden als erweitertes Unternehmensnetzwerk mit den Kunden und Partnern der Integrata Cegos – jederzeit bedarfsorientiert steuert.

Integrata Cegos ist also in jeder Hinsicht bestens gerüstet, um das Fundament des Unternehmens in der digitalen Transformationsgesellschaft weiter fest zu verankern.

Quellen

- [1] Jöhnk, H., Karle, T. (2022): Umsetzung smarterer ERP-Prozesse in Zeiten von Corona – Herausforderungen und Lösungen, in: DOAG Business News 01/2022 – Business Solutions 2022: Gut aufgestellt für das neue Jahr
- [2] Visual Studio 2022 - Walkthrough: Create your first VSTO Add-in for Word: <https://learn.microsoft.com/en-us/visualstudio/vsto/walkthrough-creating-your-first-vsto-add-in-for-word?view=vs-2022&tabs=csharp>



Matthias Sauer
matthias.sauer@promatis.de

Matthias Sauer ist Vice President Application Development bei PROMATIS und entwickelt seit über zehn Jahren kundenspezifische Anpassungen für ERP-Systeme. Im Smart-Data-Projekt ist er als technischer Projektleiter tätig.



Michael Schäfer
michael.schaefer@integrata-cegos.de

Michael Schäfer ist Prozess- und Qualitätsmanager bei der Integrata Cegos GmbH und verantwortet in dieser Funktion den unternehmensweiten kontinuierlichen Verbesserungsprozess (KVP).



Wir begrüßen unsere neuen Mitglieder

Natürliche Mitglieder:

- Nikola Di Giusto
- Katharina Aholt
- Marco Volkert
- Michael Conliffe
- Claudia Lüdke
- Jacob Grieger
- Stefan Kalscheuer
- Andre Zielinski
- Radovan Vicha
- Dominik Grüning

Korporative Mitglieder:

- JQuad OHG, Repräsentant: Lyubomir Yordanov
- SCOOP Software GmbH, Repräsentantin: Christel Bernschneider
- AGILA Haustierversicherung AG, Repräsentant: Stefan Kramer

Termine

März

03

01. und 02.03.2023

EXPERTENSEMINAR: Praxisworkshop Oracle Database Appliance
Berliner Expertenseminar mit Florian Barth
Berlin

01. und 02.03.2023

Noon2Noon Datenbank: Security Workshop der DOAG DB Community
Hannover

09.03.2023

Low-Code Plattformen
DevTalk mit Niels de Bruijn & Maarten Docter
Moderation: Kai Donato
Online

10.03.2023

DB WebSession mit Johannes Ahrends und Markus Flechtner
Multitenant Q&A
Online

21. bis 23.03.2023

JavaLand 2023. Zwei ereignisreiche Konferenztage und anschließender Schulungstag rund um das Java-Ökosystem
Phantasialand, Brühl

April

04

06.04.2023

Back-end Testing bei DB-Entwicklungsprojekten
DevTalk mit Oliver Lemm und Samuel Nitsche
Online

13. bis 15.04.2023

DOAG Leitungskräfteforum und Delegiertenversammlung
Berlin

14.04.2023

DB WebSession mit Marco Pachaly-Mischke
Praxiserfahrungen mit Multitenant
Online

20.04.2023

JavaScript in der Datenbank (MLE)
DevTalk mit Moritz Klein und Martin Bach
Online

Mai

05

03. und 04.05.2023

APEX connect 2023
Zwei Konferenztage mit zahlreichen Vorträgen und Workshops zu den Themen APEX, JavaScript und PL/SQL
Berlin

11.05.2023

DB-Programmierung: Performance-Aspekte in PL/SQL Programmierung
DevTalk mit Ulrike Schwinn, Carolin Hageman und Jürgen Sieben
Moderation: Christian Neumüller
Online

12.05.2023

Oracle Lizenzen in der Cloud
DB WebSession mit Michael Skowasch
Online

24. und 25.05.2023

DOAG 2023 Datenbank mit Exaday
Die Konferenz zu den Themen Datenbanken (Oracle Database, PostgreSQL, MySQL) und Engineered Systems (Exadata & ODA)
Düsseldorf

Impressum

Red Stack Magazin inkl. Business News wird gemeinsam herausgegeben von den Oracle-Anwendergruppen DOAG Deutsche ORACLE-Anwendergruppe e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, www.doag.org), 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 inkl. Business News ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin inkl. Business News wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Björn Bröhl. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führt einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
ViSdP: Fried Saacke
Redaktionsleitung Red Stack Magazin:
Martin Meyer
Redaktionsleitung Business News:
Marcos López
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Martin Bach, Thomas Beetz, Sven Bosinger, Arne Brüning, Simon Grossmann, Evelyn Haslinger, Thomas Lehmann, Dierk Lenz, Marcos López, Martin Meyer, Maik Michel, Johannes Michler, Ralf Müller, Karin Patenge, Andrzej Rydzanicz, Matthias Sauer, Michael Schäfer, Marcus Schröder, Norman Sibbing, Günther Stürner, Markus Zimmermann.

Titel, Gestaltung und Satz:

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

Fotonachweis:

Titel: © rawpixel.com | www.freepik.com
S. 6: © Brooke Cagle | www.unsplash.com
S. 8: © Jamesmarkosborne
| www.pixabay.com
S. 16: © MemoryCatcher
| www.pixabay.com
S. 26: © Desha | www.pixabay.com
S. 32: © PublicDomainPictures
| www.pixabay.com
S. 38: © DariuszSankowski
| www.pixabay.com
S. 42: © Praveen Thirumurugan
| www.unsplash.com
S. 47: © Veronika Andrews
| www.pixabay.com

S. 56: © jawajeziorski | www.pixabay.com
S. 65: © Neverlan | www.pixabay.com
S. 71: © Wokandapix | www.pixabay.com
S. 74: © rawpixel.com | www.pixabay.com
S. 84-87: © Platinum Events
| www.platinum-events.de
Titel S. 80: © usgs | www.unsplash.com
S. 95: @ Pietro Jeng | www.unsplash.com
S. 100: @ Carlos Alfonso
| www.unsplash.com
S. 105: © Pch Vector | www.freepik.com

Anzeigen:

sponsoring@doag.org

Metadaten und Preise:

www.doag.org/go/metadaten

Druck:

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

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags.

Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

DOAG e.V.
www.doag.org

U 2, U 3, U 4

DOAG e.V.
www.doag.org

S. 3, S. 7, S. 13, S. 15, S. 31, S. 73



CloudLand

2023

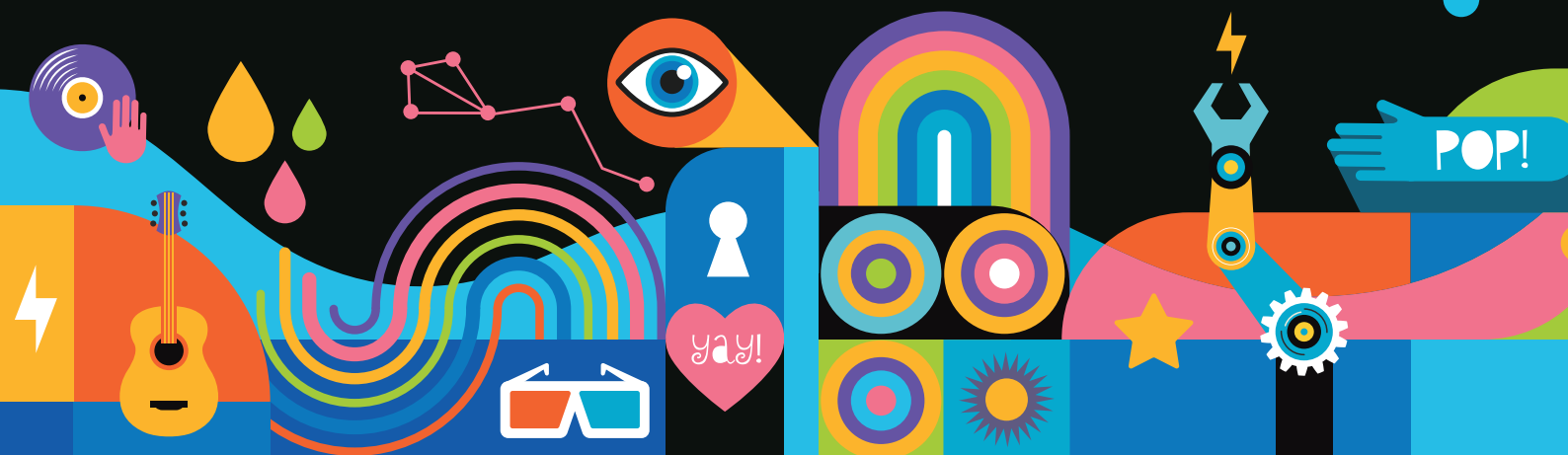
DAS EVENT DER
DEUTSCHSPRACHIGEN
CLOUD NATIVE COMMUNITY

20. - 23. JUNI
im Phantasialand in Brühl

www.cloudland.org



Weitere Informationen



#CloudLand2023

Eventpartner:  Heise Medien

APEX *connect*

by DOAG

3. - 4. Mai 2023

IN BERLIN



apex.doag.org

DOAG