Nr. 5 | August 2023 | ISSN 2366-7915 | www.doag.org

Red Stack Magazin

DOAG SOUG swiss oracle user group AOUG AUSTRIAN ORACLE USER GROUP

inklusive BUSINESS NEWS

DEVOPS & METHODIK

Aus der Praxis

Vulnerability Management für Datenbank-Plattformen

Im Interview

Torsten Kleiber, IKB Deutsche Industriebank AG

Business News

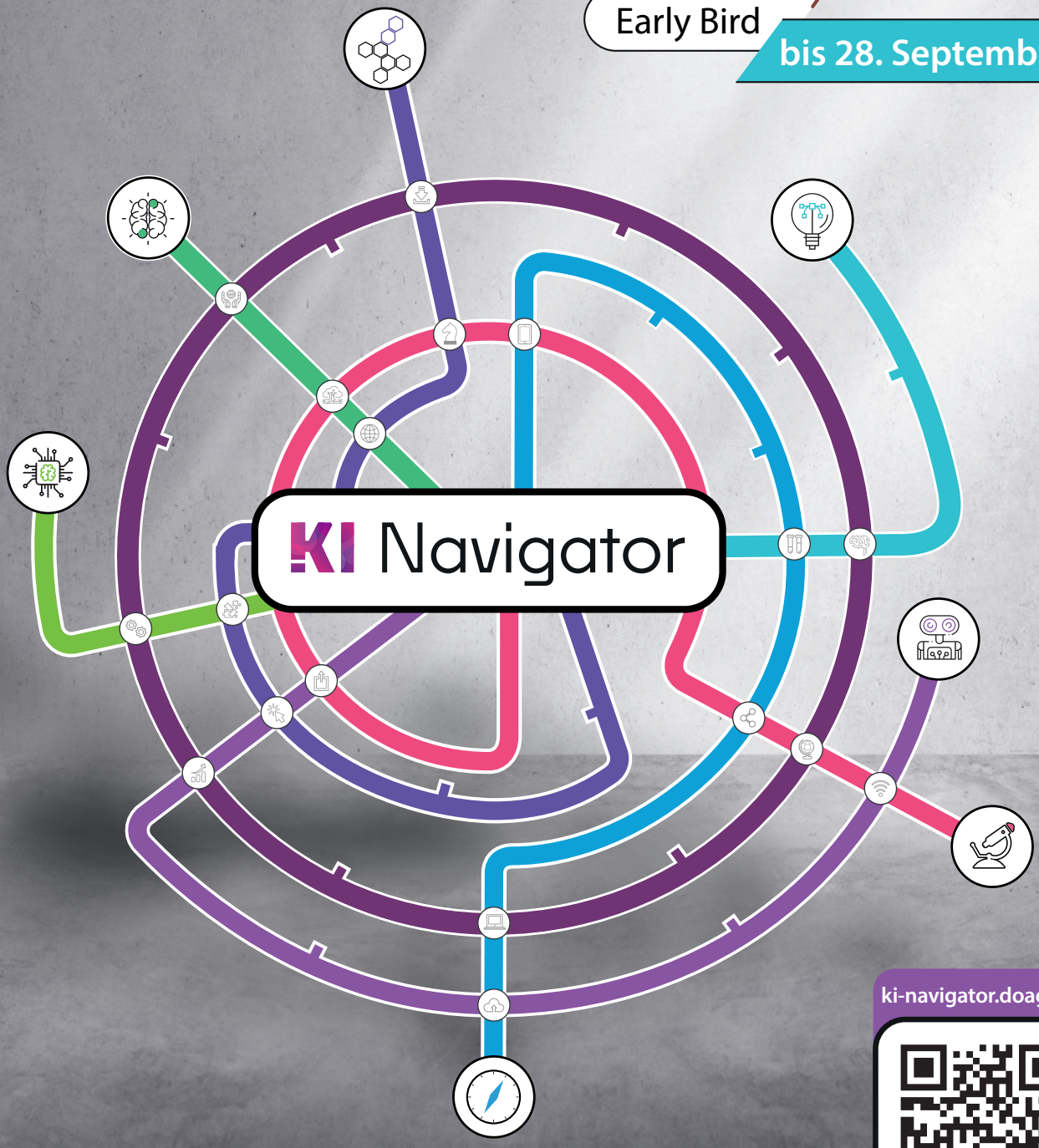
Requirements Engineering

KI Navigator 2023

Konferenz zur Praxis der KI in IT, Wirtschaft und Gesellschaft

22. + 23. November in Nürnberg

Early Bird bis 28. September



ki-navigator.doag.org

Veranstalter: DOAG Heise Medien de|ge|pol

Oliver Lemm
Themenverantwortlicher DevOps

Liebe Mitglieder, liebe Leserinnen und Leser,

das Thema DevOps, im speziellen CI/CD, hat sich mittlerweile in allen Teilen der Softwareentwicklung etabliert. Während es im Umgang mit Java und containergetriebener Entwicklung schon jahrelang gang und gebe ist, hat sich das Thema in der Datenbankentwicklung nicht so schnell durchgesetzt. In vielen Unternehmen gab es noch lange Zeit von Hand geschriebene Deployment-Skripte, die ein spezielles Know-how voraussetzten, um ein Deployment auf der Produktion durchführen zu können.

Aber auch dort profitiert vom Entwickler bis hin zum DBA jeder im Team, von der Code-Versionierung, einem einfachen Deployment, bis hin zu automatisierten Tests. In der Nutzung heutiger CI/CD Tools wie Jenkins, GitLab CI/CD oder Azure DevOps ist entweder eine Versionierung über Git schon integriert, oder es lassen sich vorhandene Code Repositories leicht einbinden.

Existierten bereits SQL-Skripts, die in der Vergangenheit von nur einzelnen Kollegen richtig ausgeführt werden konnten, so können solche Skripte mit in Pipelines aufgenommen werden, um das Deployment für alle transparenter und einfacher nutzbar zu machen. Natürlich mussten eventuell vorhandene Passwörter, welche in solchen Skripts benötigt wurden, über ein Credential Store abgesichert werden, aber wenn dies einmal eingebaut worden war, wurde so die Sicherheit erhöht, da keine Passwörter im Klartext sichtbar waren und nicht jeder der auf der Produktion deployt, benötigte das Passwort im Klartext.

Auch und gerade die Besonderheiten im Datenbankumfeld mit vorhandenen Tabellen und Daten machten das Deployment bis heute schwerer, beziehungsweise anders als mit vielen anderen Programmiersprachen. Mittlerweile haben sich Tools wie Liquibase und Flyway so weiterentwickelt, dass man entweder diese gut einsetzen kann oder aber Ansätze, in welchem SQL-Skripte mittels Generatoren re-run-fähig erzeugt werden, so dass auch bei einem mehrfachen Deployment auf Testumgebungen diese nicht immer wieder vollständig zurückgesetzt werden müssen. Auch die Weiterentwicklung im Bereich SQLcl, als Ersatz für das veraltete SQLPlus, eröffnet neue Wege. Eine Vielzahl von neuen Funktionen, wie der Umgang mit Data Pump oder der Export/Import von APEX-Anwendungen, wurden dort aufgenommen, so dass SQLPlus im Bereich CI/CD nicht mehr benötigt wird und eine komplexe Oracle-Client-Installation entfällt.

Wie immer bei neuer oder weiterentwickelter Technologie ist es wichtig, offen zu sein und sich die Themen anzuschauen. Wer den Schritt dann erfolgreich meistert, kann es damit schaffen, die verschiedenen Zweige der Softwareentwicklung unter einen Hut zu bekommen und somit aus einem Guss zu deployen.
Die Business News widmet sich diesmal dem Thema „Requirements Engineering“ und enthält dazu auch ein Interview mit Katharina Schraft, dem neuen Vorstand Business Solutions.
Ich wünsche viel Spaß beim Durchlesen dieser Ausgabe.

Oliver Lemm

Ausgabe Nr. 5/2023
auf Abruf!

GitOps – Eine grundlegende Einführung inkl. CI/CD, DevOps, IaC und Containern

Holen Sie als APEX-Entwickler das Beste aus Oracle Autonomous Databases heraus

Einleitung

DevOps

APEX

Cloud

Datenbank

PL/SQL

BUSINESS NEWS

Requirements Engineering

44

Vulnerability Management für Datenbank-Plattformen

56

Leitartikel | Requirements Engineering – das Fundament der Softwareentwicklung

60

Jobs to be Done – voller Fokus auf den Job

66

Interview mit Katharina Schraft

Intern

News

TIMELINE

6. Juni 2023

Der DevTalk „CI/CD Pipelines im Datenbankumfeld“ mit Jan Winkels und Oliver Lemm findet statt.

9. Juni 2023

Die DB WebSession mit Thorsten Bruhns widmet sich dem Thema „ansible-oracle: Deployment von Oracle Datenbankserver mittels Ansible“.

13. Juni 2023

DOAG IMC WebSession mit Murtaza Husain und John Beresniewicz zum Thema „Oracle Exadata Insights: AIOps for Optimizing Resources” steht auf der Agenda.

20. bis 23. Juni 2023

Die CloudLand 2023 – das viertägige Cloud Native Festival öffnet seine Tore im Phantasialand, Brühl. Die zweite Ausgabe der Veranstaltung im Festivalformat der Deutschsprachigen Cloud Native Community rund um die Themen Cloud- und Container-Technologien, Continuous Delivery, Microservices und DevOps setzt auf neue Formate, Top-Highlights und noch mehr Entertainment.

Rund 500 Besucherinnen und Besucher lassen sich von über 140 Beiträgen, inspirierenden Sessions, interaktiven Panels und praxisorientierten Workshops sowie kurzweiligen Themennächten begeistern.

Die Konferenz bietet im attraktiven Ambiente des Freizeitparkes eine breite Palette von Themen rund um Cloud Computing: von Sicherheit und Datenschutz bis hin zu künstlicher Intelligenz und maschinellem Lernen. Branchenführende und Fachleute teilen ihr Fachwissen, sodass die Teilnehmenden von den neuesten Trends und bewährten Praktiken profitieren und ihr Verständnis für die Möglichkeiten der Cloud-Technologie vertiefen können. Ausgiebige Gelegenheit zu entspanntem Networking sowie Spiel und Spaß sorgen im Sommermonat Juni für ein Festival mit hervorragender Stimmung

21. Juni 2023

Das Regionaltreffen Rhein-Main findet in Wiesbaden statt. Die zwei gebotenen Vorträge beschäftigen sich mit den Themen „Data Science & KI at a first glance“ und „KI in der Praxis: Was bedeutet eigentlich ‚semantic search‘?”

29. Juni 2023

Der DevTalk widmet sich „DB-Programmierung – Bessere Performance durch Organisation von Tabellendaten“. Mit dabei sind Randolf Eberle-Geist, Carolin Krützmann und Christian Schwitalla.

13. Juli 2023

Im DevTalk Summer Special stellen Beda Hammerschmidt und Carsten Czarski die „Datenbank 23 – Die neuen Wege der JSON-Datenmodellierung“ vor.

18. Juli 2023

Das Regionaltreffen München/Südbayern mit zwei Vorträgen zu den Themen „Zwei-Faktor-Authentifizierung mit APEX-Accounts” und „Oracle Support Best Practices” steht in München an.

27. Juli 2023

Gerald Venzl und Christian Schwitalla sprechen in einem weiteren DevTalk Summer Special über „23c – Domains, Annotations und mehr“.

10. August 2023

„Datenbank 23c – SQL & PL/SQL“ heißt das Thema des DevTalk Summer Special mit Ulrike Schwinn und Sabine Heimsath.

Die Oracle-Anwenderkonferenz

Eventpartner: AOUG AUSTRIAN ORACLE USER GROUP SOUG swiss oracle user group

anwenderkonferenz.doag.org

„DevOps-Tools sollten immer zur Entwicklungs- und Laufzeitarchitektur passen."

Oliver Lemm, DOAG Themenverantwortlicher DevOps, sprach mit Torsten Kleiber, DevOps Engineer bei der IKB Deutsche Industriebank AG, über DevOps/CI/CD und deren Einführung, Trunk Based Development und Mono Repositories, Quality Gates, DevOps- und Konfigurations-Tools sowie Build Pipelines.

Bitte stellen Sie sich kurz unseren Lesern vor.

Ich bin seit über 25 Jahren im Umfeld „Entwicklung mit Oracle-Datenbanken" unterwegs, davon die meiste Zeit im Bereich unserer eigenentwickelten Plattform rund um das Kredit- und Darlehensgeschäft für den Mittelstand. Ich war eine lange Zeit Entwickler und bin dann aber immer mehr mit der Infrastruktur und mit dem Development Lifecycle in Berührung gekommen. Mittlerweile beschäftige ich mich fast ausschließlich damit.

Wie sind Sie zum Thema DevOps beziehungsweise CI/CD gekommen?

Früher war es so, dass wir unsere Anforderungen für unsere Laufzeit-Umgebungen gestellt haben und diese wurden dann durch unsere internen Operation-Teams umgesetzt. Beim Deployment waren die Anforderungen allerdings schon immer sehr detailliert. Dann kam ich zu einem schon etwas länger laufenden ADF-Projekt hinzu, als man dort merkte, dass sich noch jemand um die Laufzeit-Umgebung und das Deployment kümmern müsse.

Unsere Operations-Teams waren zu diesem Zeitpunkt in eine Tochterfirma ausgegründet, sie akquirierten Drittmarkt-Geschäft und hatten für unsere neuen Anforderungen weder Zeit noch Know-how.

Deshalb führte ich Hudson als Automation Server für das Deployment ein, der später dann zu Jenkins wurde. Außerdem automatisierte ich die Installation der zugehörigen Fusion Middleware Server.

Was hat den Anstoß gegeben, DevOps einzuführen?

Laut Definition geht es bei DevOps um Methoden, um die Barrieren zwischen Development- und Operation-Teams zu reduzieren. Da unsere Operations-Tochterfirma mittlerweile an einen Dienstleister verkauft ist, bestehen diese Barrieren bei uns sogar zwischen Firmen. Deshalb nutzten wir im Bereich Application Server von unserem Dienstleister nur Basisdienste wie Managed Server und Monitoring. Installation, Konfiguration und Deployment der Application Server betreiben wir hier mit einem eigenem Sub-Team im Plattform-Development-Team. Im Bereich Datenbanken dagegen ist weiterhin das Deployment der Übergabepunkt. Installation und Konfiguration der Datenbank passieren auf unsere Anforderungen hin beim Dienstleister.

Was ist die Grundlage für eine erfolgreiche Einführung von DevOps?

Zunächst müssen erstmal alle gewillt sein, widerstrebende Interessen zwischen den beiden Seiten in einen fruchtbaren Kompromiss zu führen. Das Development führt ja immer zu einem ständigen Change. Die primäre Aufgabe von Operation ist aber die Sicherung des stabilen Betriebs. Gemeinsam müssen also Change-Prozesse entwickelt werden, die wiederholbar und testbar und gleichzeitig flexibel und schnell sind.

Welchen Einfluss haben spezielle Tools auf den Umgang mit DevOps?

Ich bin der Meinung, dass ich zuerst immer die Prozesse verstehen und gestalten muss, Tools sind dann nur die Unterstützung dieser Prozesse. Ich glaube auch, dass es eher um Typen von Tools geht als spezielle Produkte.

Zunächst sollten die Anforderungen verfolgt und die Umsetzung dokumentiert werden. In unserem Umfeld setzen wir Atlassian Jira ein.

Ein absolutes Muss ist die Versionierung von Code und meines Erachtens auch der Infrastruktur. Dabei kommt man heute nicht an Git vorbei. Wir nutzen hier Atlassian Bitbucket zum Hosten und Unterstützen der Git Repositories.

Als nächstes braucht man einen Automation Server, um seine Prozesse zu steuern. Strategisch nutzen wir hier Jenkins.

Build-Tools sind immer stark an die verwendeten Programmiersprachen gebunden und wir nutzen hier Ant und Shell Scripting, um die Oracle Compiler- und Deployment-Tools anzusteuern, Maven ist gerade im Test.

DevOps-Tools sollten immer zur Entwicklungs- und Laufzeitarchitektur passen. So macht es kaum Sinn, mit On-Premise-Tools seine Cloud Development- oder - Laufzeit-Umgebung zu steuern oder umgekehrt. Der Overhead ist nicht zu unterschätzen, zusätzliche Schnittstellen, Tunnels, Firewalls, Proxys und so weiter aufzumachen. Teilweise sind die Tools auch stark auf bestimmte Development Stacks ausgerichtet und spielen dort ihre Stärken aus. Anforderungsverwaltung, Versionierung und Build Pipelines können dagegen mittlerweile alle.

Letztlich spielt bei der Auswahl der konkreten Produkte immer auch das Unternehmensumfeld und Synergieeffekte eine Rolle. Wenn ich in anderen Plattformen schon DevOps Tools habe, kann es sinnvoll sein, diese zu nutzen. Dabei sind aber genau alle Vorteile und Nachteile abzuwägen.

Die Entwicklungsumgebung sollte zur Laufzeit passen, sowohl in der Version als auch in der Konfiguration. Mögliche Probleme können so frühzeitig erkannt werden.

Was sind die Kernpunkte im Bereich der Versionierung bezogen auf DevOps?

Also ich persönlich bin ein Fan von Trunk Based Development und Mono Repositories.

Mono Repositories speichern alle Technologien einer Plattform, eines Bereichs oder sogar eines Unternehmens. Damit ist der automatisierte koordinierte Einsatz der verwobenen Bestandteile einer Anforderung möglich. Wir haben bei uns Datenmodell, Geschäftslogik, verschiedene GUI- und Reporting-Lösungen sowie Middleware im Einsatz. Im besten Fall ist auch Code für die Installation und Konfiguration der Infrastruktur für zum Beispiel Datenbank und Application Server dabei.

Trunk Based Development fördert aus meiner Sicht die wirkliche kontinuierliche Integration. Kontinuierliche Integration bedeutet per Definition, jeden Tag seine Änderungen mindestens einmal integrativ in den Main Branch zu bringen und vice-versa auch wieder zurück in den Feature Branch. Bei Trunk Based Development ist das Ziel, wenn überhaupt Feature Branches genutzt werden, diese innerhalb kürzester Zeit in den Main Branch zu bringen.

Letztendlich sind aber auch diese Modelle vom speziellen Entwicklungsumfeld abhängig und können bei einem Unternehmen, was verschiedene Major-Versionen parallel unterstützen muss oder selbst bei uns in anderen Plattformen ganz anders aussehen. Wenn ich jetzt mehrere Releases die ganze Zeit parallel führen will, dann könnte ich mir vorstellen, dass so etwas wie Git-Flow ein valides Branching-Modell wäre.

Was halten Sie von Quality Gates im Bereich von CI/CD Pipelines?

Quality Gates sind von der Idee etwas Gutes, sind aber stark abhängig von der Unternehmenskultur. Aus meiner Sicht muss erst bei den Entwicklern die Erkenntnis reifen, dass sie einen Nutzen von automatischen Qualitätsprüfungen haben, zum Beispiel, dass sie später weniger Bugs beheben müssen. Aber im ersten Moment scheint es erstmal mehr Arbeit für sie zu sein. Wenn dieser Nutzen erkannt ist, ist dafür sorgen, dass die Ursache für das Brechen eines solchen Gates in kürzester Zeit gefixt ist. Wenn ich solche Gates einführe, dann unterbreche ich ja den CI/CD-Zyklus. Jeder manuelle Prozess, den ich einführe, und so eine Unterbrechung erfordert ja letztendlich eine manuelle Bereinigung, benötigt ja Zeit.

Oder wenn ein aufgetretener Bruch des Gates nicht in kurzer Zeit behoben ist, setzte ich automatisch die letzten verursachenden Änderungen im Branch zurück. Die zu wählende Zeit ist davon abhängig, wie lange es dauert, bis mein Code am Quality Gate ankommt. Das funktioniert meistens nur dann, wenn ich in dieser Zeit den Code ändern, einchecken und bauen kann.

Generell sollten meines Erachtens die Gates zunächst im Build von Pull Requests geprüft werden. Bei uns ist es so, dass auch die Pull Requests gebaut werden, ganz einfach um die Chance zu minimieren, den Main Branch zu brechen. Im Pull Request wird erstmal der Merge des Feature Branch mit dem aktuellen Main simuliert. Wenn mehrere Pull Requests offen sind, werden bei einem Merge in den Main Branch die weiter offenen Pull Requests automatisch neu gebaut, es könnte sich durch den aktualisierten Main Branch ja das Ergebnis geändert haben.

Wir nutzen das einerseits, um Merge-Konflikte vor dem Merge in den Main zu finden. Andererseits prüfen Tests XML-Strukturen auf Deployment-relevante Fehler. Zum Beispiel wird geprüft, ob jemand JDBC Connections statt JNDI Connections benutzt. JDBC Connections würden fest auf die beim Programmieren genutzte Entwicklungsdatenbank verbinden und die Produktion würde immer noch darauf zugreifen. JNDI Connections sind unabhängig vom Applikations-Code auf der jeweiligen Umgebung im Application Server konfiguriert und zeigen damit auf die passende Datenbank. Bei solchen Fehlern brechen unsere Pipelines ab.

Weiterhin hatten wir auch Gates für scheiternde Oberflächen-Tests, das wurde bei uns wieder „entschärft“. Hier hatten wir oft false-negative Tests, weil die Test-Knoten außerhalb unserer Hoheit automatisiert mit Browser Updates versorgt werden. Dann brauche ich oft auch neue Versionen der Browser-Treiber für das Test-Framework. Diese muss ich dann erstmal wieder konfigurieren und einstellen. Andererseits gibt es auch mal Flaky Tests, die nicht unbedingt sofort zum Abbruch führen sollen. Deshalb werden bei uns jetzt bei fehlerhaften Tests die Pipelines nur noch instabil, also gelb statt rot oder grün. Anhand der Tests kann der Entwickler dann entscheiden, was zu tun ist.

Frameworks für Codeanalyse dagegen bringen in ihren Versionen immer weitere neue Verletzungstypen mit sich. Das heißt, durch Weiterentwicklung der Frameworks gibt es neue Verletzungen an unverändertem Code. Entweder konfiguriere ich mich

dann tot, um diese Regeln erstmal auszuschließen oder der Entwickler hat viel Zeit und fixt alle Verletzungen. Ich würde allerdings das Quality Gate so konfigurieren, dass es nur den geänderten Code so prüft, so dass er zumindest nicht schlechter wird.

Würden Sie rückblickend im Rahmen der DevOps-Einführung etwas anders machen?

Ich würde organisatorisch statt DevOps im Development-Team nur noch ein DevOps-Team betreiben. Jeder Entwickler sollte auch wissen, wie die Anwendung, die er programmiert hat, zu betreiben ist. Das ist in einem stark regulierten Umfeld wie bei uns als Bank wegen des damit verbundenen Ziels Funktionstrennung zwischen Administration und Entwicklung kaum umsetzbar, das führt aber zu neuen internen Barrieren.

Ich würde von vornherein versuchen, möglichst oft Container einzusetzen, um unabhängiger von Hardware und Betriebssystemen zu sein.

Ich würde versuchen, mehr auf mit Terraform erstellte Cloud-Infrastruktur zu setzen, um unabhängig von Provider-Prozessen zu werden und, um die Geschwindigkeit der Bereitstellung der Infrastruktur zu beschleunigen.

Ich würde mehr auf Konfigurations-Tools für die Infrastruktur setzen, wahrscheinlich auf Puppet. Wichtig ist hier die Möglichkeit der idempotenten Beschreibung meiner Infrastruktur, das heißt, ich beschreibe den Zielzustand. Wo ich herkomme und der Weg des Konfiguration-Tools dahin, ist für mich eine Blackbox.

Wie sieht der Ausblick aus, was fehlt Ihnen noch?

Fehlen wird immer etwas, da ständig neue Anforderungen entstehen. Aktuell ist einer der wichtigen Punkte der Deployment-Prozess für die Datenbank. Wir haben einen funktionierenden, der benutzt aber immer noch nicht Git für die Versionierung und einen Automation Server von unserem Dienstleister. Hier haben wir Ideen, wir haben aber noch nicht die Zeit gehabt, es wirklich zu Ende zu testen und zur Einsatzreife zu bringen.

Ein zweites Thema ist der aktuell beschlossene Wechsel von Oracle Reports zu Jasper Reports. Wir haben jetzt Proof of Concepts für die Programmierung gemacht und müssen jetzt den Deployment-Prozess aufbauen. Der basiert zum Beispiel auf Maven, da muss auch der Entwicklungsprozess so gestaltet werden, dass der Entwickler einfach damit arbeiten kann. Außerdem stehen weitere Proof of Concepts für eine neue GUI-Umgebung an.

Aktuell kommt stark der Security-Aspekt hinzu. Mittlerweile ist es nicht mehr so, dass wir regelmäßig nur Server patchen müssen, sondern auch die lokalen Umgebungen der Entwickler. Das setzt entweder Scripting oder Containerisierung voraus. Anders bekomme ich das nicht hin, wenn ich mir überlege, dass man von Oracle allein 4 CPUs im Jahr für jeden unserer Application-Server und jedes unserer Entwicklungstools bekommt. Und die Oracle Tools sind ja nicht die einzigen.

Seit Anfang 2023 ist DevOps kein Einzelkämpferjob mehr für mich auf der Plattform. Somit ist Wissenstransfer ein Thema und natürlich versuchen wir uns hier auch an Documentation as Code. Das heißt, wir schreiben unsere Dokumentation möglichst im Asciidoc-Format und legen diese relativ nah am Programm- und Infrastruktur-Code ab, damit wir den Code in der Dokumentation auch referenzieren können. Über die richtige Referenzierung per Kommentar-Tags sehe ich auch im Code, dass dieser in der Dokumentation benutzt wird und, dass diese geprüft und angepasst werden muss. Auch diese Dokumentation verteilen wir dann per Build Pipelines, in unserem Fall dann per Jenkins nach Atlassian Confluence.

Und ansonsten haben wir aus unserem Backlog immer was zu automatisieren und zu optimieren oder auch nur irgendwas zu dokumentieren, was bisher anderen Prioritäten zum Opfer gefallen ist.

TORSTEN KLEIBER

Torsten Kleiber ist DevOps Engineer für die auf Oracle-Technologien basierende Kredit- und Darlehens-Plattform eines mittelständischen Finanzinstituts. Er sorgt dafür, dass Entwickler entwickeln können, der Betrieb stabil läuft und technische Einführungen und Migrationen gelingen. Sein Wissen gibt er an die Community auf Konferenzen und seinem Blog *https://www.amapac.io* weiter.


CloudLand


LET'S TALK

CloudLand 2023

CloudLand 2023 – eine Rückschau in Bildern

Text und Bilder von Marcos López

Die zweite Ausgabe des Cloud Native Festivals konnte mit 500 Teilnehmenden einen größeren Zuwachs verzeichnen und wusste erneut zu überzeugen.

Das gelang mit über 140 Beiträgen, zahlreichen Workshops und interaktiven Formaten sowie kurzweiligen Themennächten und einem innovativen Sponsorenkonzept. In kleinen und großen Runden präsentierte das Festival vom 20. bis 23. Juni im Phantasialand (Brühl) zahlreiche Innovationen, Mittel und Wege bezüglich Cloud Computing, KI und Zukunftstechnologien.

Eröffnet wurde das Festival mit einem eintägigen CloudCamp, dem die Summer Night mit Konzert und DJ sowie die Gaming Night mit dem Schwerpunkt Gamification als Highlights des Rahmenprogramms folgten.

Die #CloudLand2023 war eine gelungene Veranstaltung, die das Bewusstsein für die Bedeutung von Cloud-Technologien schärfte und die Zusammenarbeit in der Cloud Native Community der DOAG (DCNC) vorantrieb.

Wir dürfen zurecht gespannt sein – auf die CloudLand 2024!

GitOps – Eine grundlegende Einführung inkl. CI/CD, DevOps, IaC und Containern

Moritz Reinwald, MT

DevOps ist einer der meistgenannten Begriffe der letzten Jahre, wenn es darum geht, Entwicklungsprozesse zu automatisieren. Weniger bekannt ist dagegen der Ableger GitOps, dreht sich hier doch alles um die Automatisierung der Infrastruktur im Hintergrund. Dieser Artikel gibt eine grundlegende Einführung in das Thema GitOps und greift dabei auch die Verbindungen sowie Unterschiede zu DevOps auf.

Basics – CI/CD & DevOps

Wer sich mit GitOps beschäftigen möchte, sollte sich vorher mit den Grundlagen im Bereich CI/CD und DevOps vertraut machen. Dazu gehört es, diese beiden Keywords zu verstehen und in Verbindung zu bringen.

In *Abbildung 1* werden die verschiedenen Schritte der Automatisierung visualisiert. Hierbei handelt es sich um eine weit verbreitete Art und Weise, die Konzepte CI/CD und DevOps darzustellen. Zu sehen sind die verschiedenen Prozessschritte einer Automatisierung. Angefangen bei der Planungsphase über die Umsetzungsphase (coding) und den Bau beispielsweise einer Anwendung bis hin zu den Tests. Werden diese vier Schritte in einer sogenannten Pipeline automatisiert ausgeführt, spricht man von Continuous Integration (CI). Wird zudem der nachfolgende „release"-Schritt automatisiert, in welchem das fertige (Teil-)Produkt automatisiert erstellt und anschließend manuell veröffentlicht wird, spricht man von Continuous Delivery (CD). Nun wird es leider etwas „tricky", denn nicht nur Continuous Delivery wird gerne mit CD abgekürzt. Wird noch einen Schritt weiter gegangen und das Release auch direkt automatisiert in die Produktivumgebung eingespielt, wird von Continuous Deployment (CD) gesprochen. CI/CD beschreibt somit die Automatisierung der verschiedenen Schritte während der Entwicklung eines Produkts mit Hilfe von Pipelines. Dabei kann unterschiedlich weit gegangen werden, wobei vor allem zwischen Continuous Delivery und Continuous Deployment unterschieden werden sollte.

Wie passt nun DevOps, eines der großen Schlagworte der letzten Jahre, in diesen Zusammenhang? DevOps geht noch einige Schritte weiter, entfernt sich aber auch von der technischen Umsetzbarkeit. So werden die den Betrieb betreffenden Schritte „operate" und „monitor" aufgenommen. Und genau darum geht es bei DevOps: Den Betrieb mit in die Entwicklung einzubinden. Dies geschieht dann nicht mehr nur technisch in Form von automatisierten Pipelines, sondern muss als Unternehmenskultur gelebt werden.

Wird versucht DevOps genau zu definieren, könnte es wie folgt lauten:

„DevOps bezeichnet eine Reihe von Praktiken zur Automatisierung der Prozesse zwischen Softwareentwicklern und IT-Teams, durch die Software schneller und zuverlässiger entwickelt, getestet, freigegeben [2], betrieben und gewartet werden kann".

Somit vereinigt DevOps sowohl die Schritte des Entwicklungszyklus (Planen, Entwickeln, Bauen, Testen), also **Dev**elop, als auch die Schritte des Betriebszyklus (Release erstellen, Veröffentlichen/Einspielen, Betreiben, Überwachen), also **Op**eration**s**, in einem einzigen Zyklus, welcher sich während der Lebenszeit des Produkts immer wieder wiederholt.

Basics – Infrastructure as Code (IaC)

Weiteres Grundlagenwissen ist im Bereich der Infrastrukturautomatisierung nötig. Infrastructure as Code ist ein Teil des Infrastruktur-Managements. Hierbei wird Infrastruktur deklarativ, dementsprechend beschreibend, definiert. Dadurch ist die Konfiguration von Infrastruktur jederzeit reproduzierbar und zudem versionierbar. Es entsteht eine versionierte Definition der Infrastruktur, das Configuration Model. Bei Ausführung generiert, beziehungsweise erstellt, es die Infrastruktur immer mit der identischen Konfiguration. IaC ist somit Teil des Konfigurationsmanagements (CM) und basiert auf Configuration as Code (CaC).

Sollen nun Änderungen an der Konfiguration der Infrastruktur vorgenommen werden, gilt es, dies nach dem Motto „Editing the source, not the target" zu erledigen. Dementsprechend werden Änderungen nicht direkt in oder an der Infrastruktur vorgenommen, sondern im

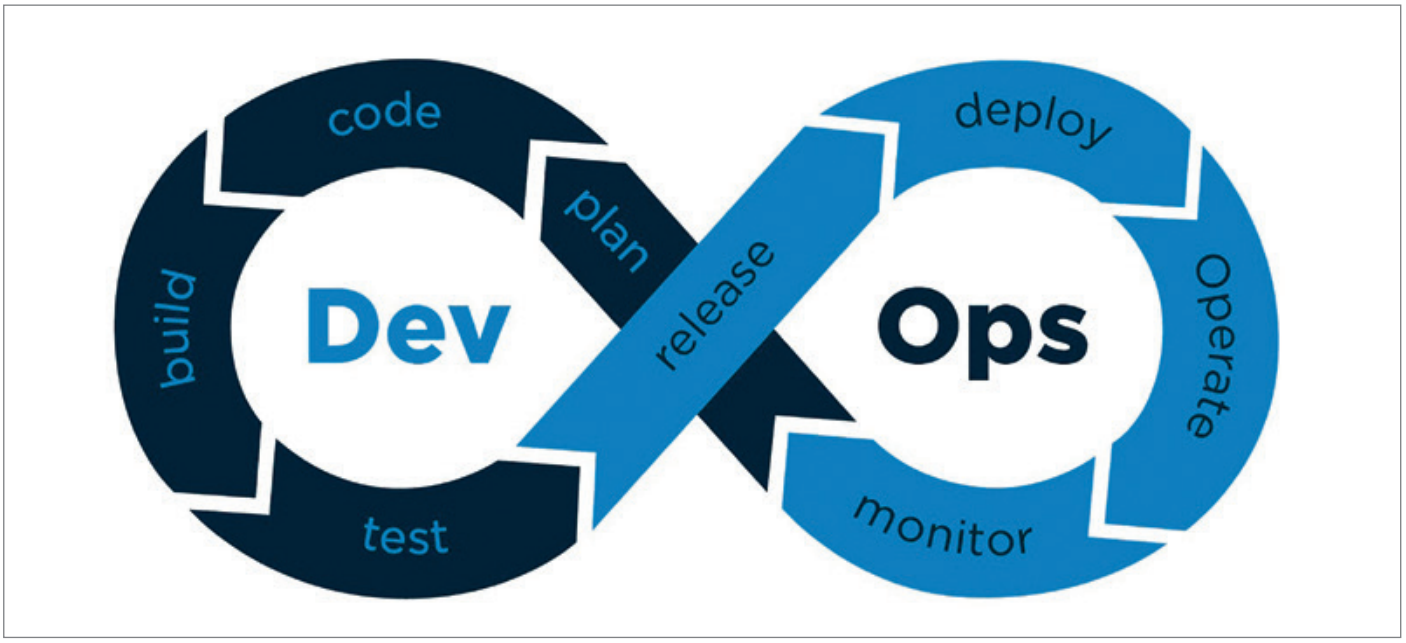


Abbildung 1: Der DevOps Zyklus [1] (Quelle: Moritz Reinwald)

Code definiert und die Infrastruktur mithilfe dieses Codes neu erzeugt. Wird sich an dieses Prinzip gehalten, existieren keine unterschiedlich konfigurierten Umgebungen und somit auch keine Inkonsistenzen. Daher löst Infrastructure as Code das Problem des Configuration Drift. Dieser beschreibt eine Abweichung des aktuellen Zustands (Actual State) vom Desired State, also dem gewünschten und definierten Zustand. Dies kann auf verschiedene Arten geschehen. Zum einen kann der aktuelle Zustand durch verschiedenste Umstände vom definierten Zustand abweichen, zum anderen kann sich der definierte Zustand zum Beispiel durch gewollte Anpassungen verändern. Hier greift dann GitOps ein und versucht diese Differenz beizulegen (reconciliation).

Infrastructure as Code unterstützt DevOps-Teams so mit einer schnellen und skalierbaren Infrastruktur. Bekannte Werkzeuge für IaC sind u.a. Chef *[3]*, Puppet *[4]* und Ansible *[5]*.

Basics – Docker & Kubernetes

Zu guter Letzt sollten wir uns noch einmal mit den grundlegenden Begriffen und Eigenschaften von Docker und Kubernetes beschäftigen, um den Einsatzzweck sowie die Möglichkeiten von GitOps nachvollziehen zu können.

Docker *[6]* ist eine Virtualisierungslösung, welche mit sogenannten Containern arbeitet. Diese Technologie wird dann bei Cloud-Native-Applikationen wie Kubernetes genutzt. Eine Containerisierung hat hierbei insbesondere den Vorteil, dass leichtgewichtige Ausführungseinheiten erstellt werden können, da diese nur die eigentliche Applikation sowie nötige Abhängigkeiten zur Ausführung dieser beinhalten. Ein einmal erstellter Container ist zudem imutable , also nicht veränderbar. Werden Änderungen an der Software vorgenommen, muss anschließend ein neuer Container erstellt werden. Zum Ausführen der Container wird dann eine Container Runtime benötigt. Oftmals wird hier Docker als Synonym für diesen Kontext verwendet. Allerdings gibt es hier viele verschiedene Lösungen von unterschiedlichen Anbietern. Docker wird heutzutage eher als Begriff genutzt, der verdeutlichen soll, dass im Tech Stack mit Containerisierung gearbeitet wird (runtime, containers, images, ...).

Wie in *Abbildung 2* zu erkennen ist, unterscheiden sich die verschiedenen Deployment-Möglichkeiten stark voneinander. Bei der traditionellen Methode werden die Anwendungen direkt auf dem Server veröffentlicht und teilen sich unter Umständen das Betriebssystem und die Serverkapazitäten. Werden beim Deployment virtuelle Maschinen genutzt, wird sehr viel „Overhead" erzeugt, indem in den einzelnen VMs wiederum eigene Betriebssysteme aktiv sind, auf welchen die Anwendungen dann laufen. Hier erreicht man im Gegensatz zum traditionellen Verfahren schon eine Trennung der Belange, hat dafür allerdings auch deutlich mehr Ressourcenverbrauch. Einen Mittelweg bietet das Deployment mittels Container. Hierbei teilen sich die Container ein Betriebssystem, allerdings sind die Anwendungen dennoch in einzelnen Containern samt benötigter Librarys untergebracht, wodurch bei weniger Ressourcenverbrauch ebenfalls eine Trennung der Belange erreicht wird. Für die Nutzung einer solchen Deploymentmethode kann dann beispielsweise Kubernetes genutzt werden.

Kubernetes *[7]* ist eine Open-Source-Container-Orchestration-Platform, welche von Google 2014 ins Leben gerufen wurde. Mit Kubernetes können also containerbasierte Workloads und Services verwaltet werden. Hierbei kümmert sich Kubernetes praktisch um alles, egal ob einzelne Container gerade „sterben" und neu gestartet werden müssen, die Maschine, auf welcher die Container laufen, fehlerhaft ist oder auch, ob und wie Container untereinander kommunizieren können.

Kubernetes sorgt dementsprechend für die Erreichbarkeit von Services inklusive Load Balancing, Speicherverwaltung, automatisierte Rollouts und Rollbacks von Software in Form von Containern und einer Selbstheilung für ebendiese in Form von automati-

sierten Neustarts und dem Aufräumen „toter" Container. Ist Kubernetes deployt, spricht man von einem Kubernetes Cluster. Dieses besteht aus sogenannten Nodes, auf welchen die containerisierten Applikationen dann ausgeführt werden. Ein Cluster kann hierbei On Premise oder – wie mittlerweile weit verbreitet – in der Cloud gehostet werden. Sämtliche großen Cloud-Provider bieten hierfür speziell vorgefertigte Software. Ein weiterer Begriff aus der Kubernetes-Welt sind die sogenannten Pods, die kleinste ausführbare Einheit. Ein Pod ist hierbei eine Gruppe von einem oder mehreren Containern, die sich sowohl die Speicher- und Netzwerkressourcen als auch die Spezifikation teilen.

Möchten wir für das obige Beispieldeployment Kubernetes nutzen, empfiehlt es sich, die Funktionsweise von Kubernetes zu kennen. Wie in *Abbildung 3* sichtbar, definiert Kubernetes ein Container Runtime Interface, welches dann durch unterschiedliche Container Runtimes umgesetzt werden kann. Dieses Interface regelt, wie Kubernetes mit der Runtime interagiert. Beispiele für Container Runtimes sind containerd *[8]*, von Docker entwickelt und CRI-O *[9]*, eine Open Source Runtime. Diese Runtimes ermöglichen dann die Ausführung von Containern, die der „Open Container Initiative (OCI) Specification" folgen. Ein Beispielwerkzeug hierfür ist runc *[10]*, welches entsprechende Container erstellen kann.

Was ist GitOps?

Zurück zum Kernthema: Was ist GitOps? GitLab hat den Begriff definiert *[11]*, frei übersetzt lautet diese Definition:

„GitOps ist eine Vorgehensweise, bei der bewährte DevOps-Methoden für die Anwendungsentwicklung wie Versionskontrolle, Zusammenarbeit, Regelkonformität sowie CI/CD übernommen und auf Infrastrukturautomatisierung anwendet werden."

Der Name GitOps setzt sich hierbei aus Git, dem Versionskontrollsystem, und „Ops" als Abkürzung für Operations, also dem Betrieb zusammen. Definiert wurde GitOps ursprünglich 2017 bei der Firma Weaveworks, im speziellen von Alexis Richardson.

GitOps ist hierbei per se nicht an eine bestimmte Technologie gebunden, sondern ist eher eine Methodik, nach welcher gearbeitet werden soll. Dennoch wird GitOps oft in einem Atemzug sowohl mit Git – als Synonym für ein Versionskontrollsystem – als auch mit Kubernetes-Systemen genannt und dementsprechend oft in diesem Bereich eingesetzt. GitOps soll hierbei ermöglichen, dass Entwickler zu einem gewissen Grad den Betrieb übernehmen, indem dieser sich von der Vorgehensweise der Entwicklung annähert und unter anderem dieselben Tools genutzt werden können. Deswegen wird auch oft von „Developer Friendly Operations", also dem entwicklerfreundlichen Betrieb gesprochen.

Von DevOps werden neben dem Namen auch weitere Methoden abgeleitet

So wird bei DevOps als auch bei Infrastructure as Code das Prinzip von Configuration as Code angewandt. Es wird also der Ansatz verfolgt, sämtliche Konfigurationen sichtbar als Code zu deklarieren. Und Code muss wiederum versioniert werden. Dementsprechend müssen Konfigurationen als Code versioniert werden. GitOps nimmt nun diese Praktiken und erweitert diese. Jede deklarativ als Code in der Versionskontrolle hinterlegte Konfiguration kann automatisiert werden. Dementsprechend sollte so viel wie möglich als Code beschrieben werden: Anwendungslogik, Konfigurationen, Überwachung, Regeln, und so weiter.

Sämtlicher Code sollte versioniert werden. Damit kann dann auch die Automatisierung ausgebaut werden.
Daraus lassen sich einige prägnante Prinzipien für die Anwendung von GitOps ableiten:

- Git bzw. Versionskontrolle = Single Source of Truth → Was nicht im Git ist, existiert nicht

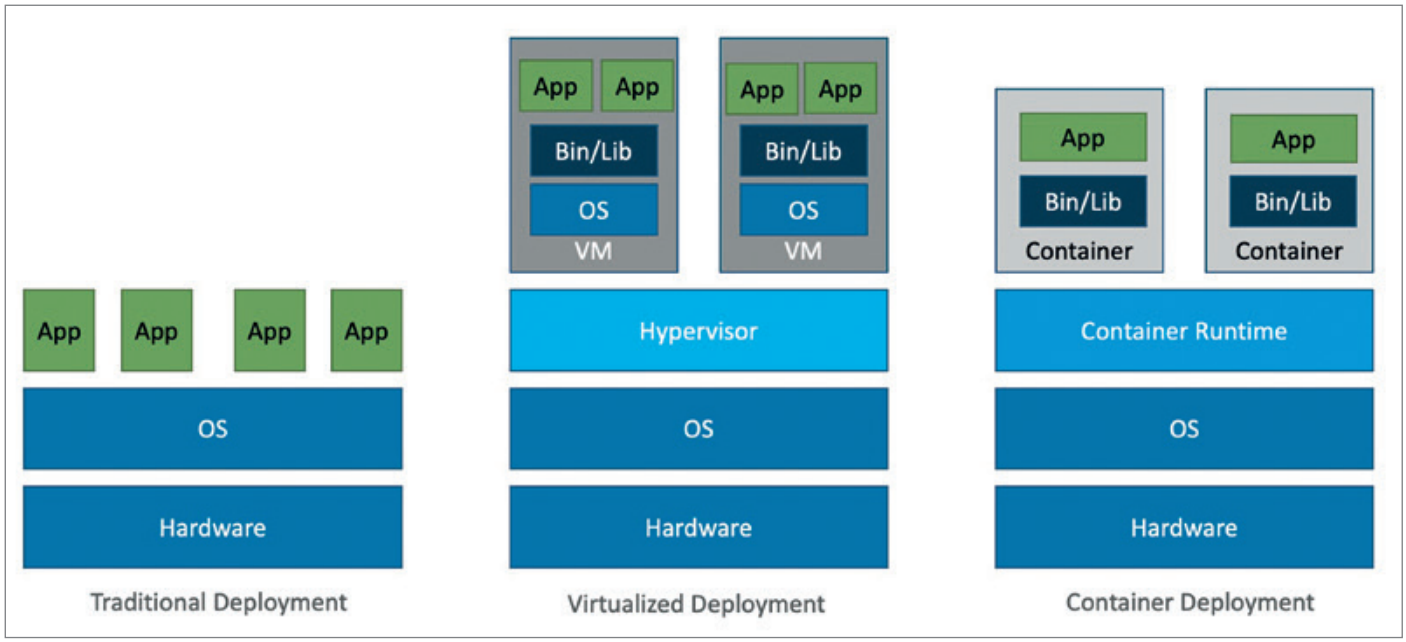


Abbildung 2: Traditionelles Deployment vs. Virtualisierung mittels virtueller Maschinen vs. Virtualisierung mittels Container (Quelle: Moritz Reinwald)

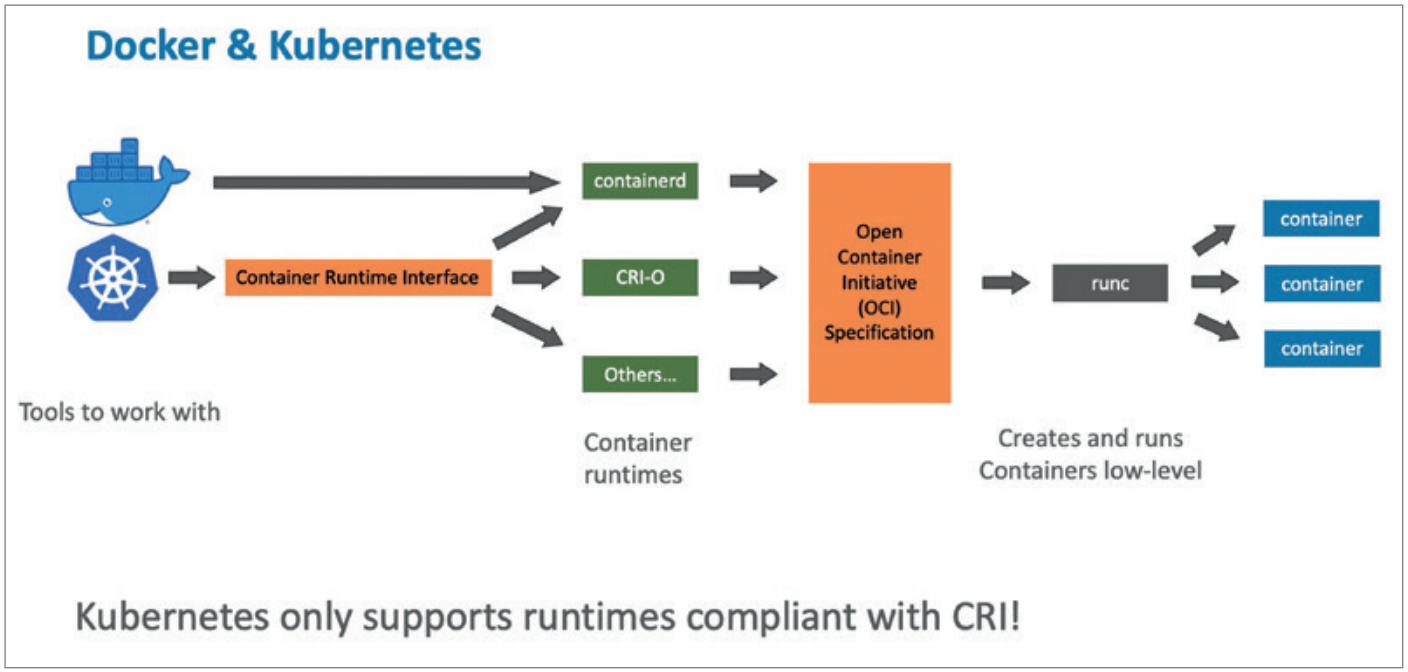


Abbildung 3: Zusammenspiel von Kubernetes mit Container Runtimes und Containern (Quelle: Moritz Reinwald)

- Alles ist deklarativ → Als Code hinterlegt
- Soll-Zustand (Desired State) ist im Code beschrieben
- Konstante Überprüfung des Ist-Zustandes im Vergleich zum Soll-Zustand
- Bei Unterschieden (Configuration Drift) → Differenzen beilegen (reconciliation)
- Infrastruktur ist unveränderbar zur Laufzeit (Immutable)
- Infrastruktur ist versioniert → Nachvollziehbarkeit
- Anpassungen durch deklarative Änderungen am Code, nicht an der Infrastruktur

Die 3 Säulen von GitOps

Neben den oben genannten Prinzipien nutzt GitOps drei unterschiedliche Themen als Grundlagen *(siehe Abbildung 4)*. Die erste Säule auf die GitOps baut, sind automatisierte Pipelines. Hiermit sind sowohl Continuous Integration als auch Continuous Deployment Pipelines gemeint. Zudem wird eine maximale Automatisierung angestrebt, daher ist auch ein automatisiertes Release Management von Vorteil.

Neben Pipelines setzt GitOps als zweite Säule auf die Beobachtbarkeit beziehungsweise Überwachung. Ein konstantes Monitoring und Logging hilft dabei Änderungen wie einen Configuration Drift so schnell wie möglich festzustellen. Hierbei sind eine entsprechend gute Visualisierung sowie Benachrichtigungen bei Änderungen unabdingbar. Ziel ist es, einen ganzheitlichen Blick über das gesamte System genau zum aktuellen Zeitpunkt zu bekommen. Als letzte grundlegende Säule von GitOps ist Kontrolle anzuführen. Wenn durch die Überwachung eine Änderung zum Soll-Zustand festgestellt wird, gilt es, das System schnellstmöglich wieder in den gewünschten Zustand zu überführen. Hierbei geschehen sämtliche Änderungen, welche vorgenommen werden, natürlich über Git beziehungsweise im Allgemeinen über das Versionskontrollsystem. Dies betrifft unter anderem Updates, Regeln wie auch Sicherheitsmaßnahmen. Dabei kann nach dem „Diff and Sync"-Prinzip vorgegangen werden, dementsprechend der aktuelle Ist-Zustand mit dem Soll-Zustand verglichen und, wenn nötig, synchronisiert werden. Ziel ist es, durch die Automatisierung eine dauerhafte Konvergenz von Ist- und Soll-Zustand zu erreichen.

Wie funktioniert GitOps?

Als Teil der Überwachung müssen Änderungen an der Infrastruktur sofort sichtbar werden. Hierfür können sogenannte Diff Alerts genutzt werden. Diese senden eine Benachrichtigung, sobald der aktuelle Zustand Differenzen zum definierten Zustand aufweist. Um Änderungen innerhalb eines Clusters zu überwachen, kann beispielsweise Kubediff *[12]* genutzt werden. Wird die Infrastruktur mit Terraform *[13]* verwaltet, einem sehr bekannten Werkzeug zur Infrastrukturautomatisierung, so empfiehlt sich Terradiff *[14]*. Ansible stellt mit dem Diff Mode *[15]* ebenfalls Möglichkeiten zur Verfügung. Oftmals werden auch alle genannten Werkzeuge genutzt, um beispielsweise die Änderungen in den verschiedenen Ebenen mitzubekommen. Hierbei ist Kubediff für die Anwendungsschicht zuständig, also alles innerhalb des Clusters, Ansiblediff behandelt die mit Ansible aufgebaute Kubernetesschicht und Terradiff meldet Änderungen auf der Infrastrukturschicht, also zwischen Versionierung und der mit Terraform verwalteten Infrastruktur. Der gesamte Aufbau sowie die Nutzung der entsprechenden Werkzeuge wird in einem Blogartikel von Weaveworks *[16]* ausführlich beschrieben.

Sollen Änderungen an der Infrastruktur vorgenommen werden, geschieht dies wie bereits erwähnt nur über den Code. Dieser Prozess folgt bei GitOps einigen Regeln.

Änderungen werden ausschließlich über sogenannte Merge Requests (GitLab) beziehungsweise Pull Requests (GitHub) durchgeführt. Diese sind ein Konstrukt der Versionierung, bei dem eine formale Anfrage für ein

Zusammenführen von Original und geändertem Quellcode gestellt und – im besten Fall von einer anderen Person – genehmigt werden muss. Diese Merge/Pull Requests müssen ebenfalls genutzt werden, um zum Beispiel einen Staging oder Feature Branch in den Main (oder Develop) Branch zu mergen. Hier greift die nächste GitOps-Regel: Es wird immer in solchen Staging/Feature Branches entwickelt und niemals direkt auf dem Main Branch. Entsprechende Sicherheitsmaßnahmen wie etwa „Protected Branches" sollten daher genutzt werden, um eine direkte Entwicklung an der aktiv genutzten Infrastrukturdefinition – und dementsprechend live an der Infrastruktur – zu verhindern und Code Reviews zu erzwingen. Änderungen an der aktiven Definition werden dann über die Merge/Pull Requests abgewickelt. Durch die Nutzung dieser Abläufe werden mehrere Vorteile erreicht. Zum einen werden immer wieder Code Reviews durchgeführt, wodurch der Code mehrfach kontrolliert und unter Umständen verbessert wird. Zudem wird die Zusammenarbeit im Team gestärkt. Darüber hinaus werden durch die starke Nutzung von Versionierung und Merge/Pull Requests eine generell gute Nachverfolgbarkeit von Änderungen sowie formale Genehmigungen integriert.

Der Deployment-Prozess unterscheidet sich bei GitOps deutlich zu dem bei herkömmlichen CI/CD. Wie in *Abbildung 5* zu sehen, besteht der erste Schritt des Prozesses aus einem Entwickler, welcher Code in ein Repository pusht. Dieser Schritt ist bei CI/CD und GitOps identisch. In klassischen CI/CD wird daraufhin der Code von einem Automatisierungsserver wie zum Beispiel Jenkins aus diesem Repository gepullt und mit einigen – hier nicht relevanten – Zwischenschritten anschließend deployt.

Dieser Prozess wird entweder manuell oder bei Änderungen am Repository automatisch angestoßen. Bei GitOps unterscheidet sich dieser Prozess, indem ein sogenannter GitOps Operator wie etwa Flux *[17]* oder ArgoCD *[18]* genutzt wird. Dieser erkennt die Differenz zwischen dem Repository – dem neuen Soll-Zustand – und der aktuellen Infrastruktur – dem Ist-Zustand – und stößt dann eine entsprechende Aktualisierung an. Dadurch werden dann die Änderungen gepullt und deployt. Die Besonderheit ist hierbei, dass der Operator selbst Teil dieser Infrastruktur ist, das heißt zum Beispiel ebenfalls im Cluster ausgeführt wird.

GitOps Operator

GitOps-Operatoren prüfen dauerhaft den aktuellen Zustand und vergleichen diesen mit dem im Code definierten Soll-Zustand. Dafür nutzen sie – zumindest im Kubernetes-Umfeld – Pull-basierte Deployments, das heißt, der Operator fragt das Repository ab.

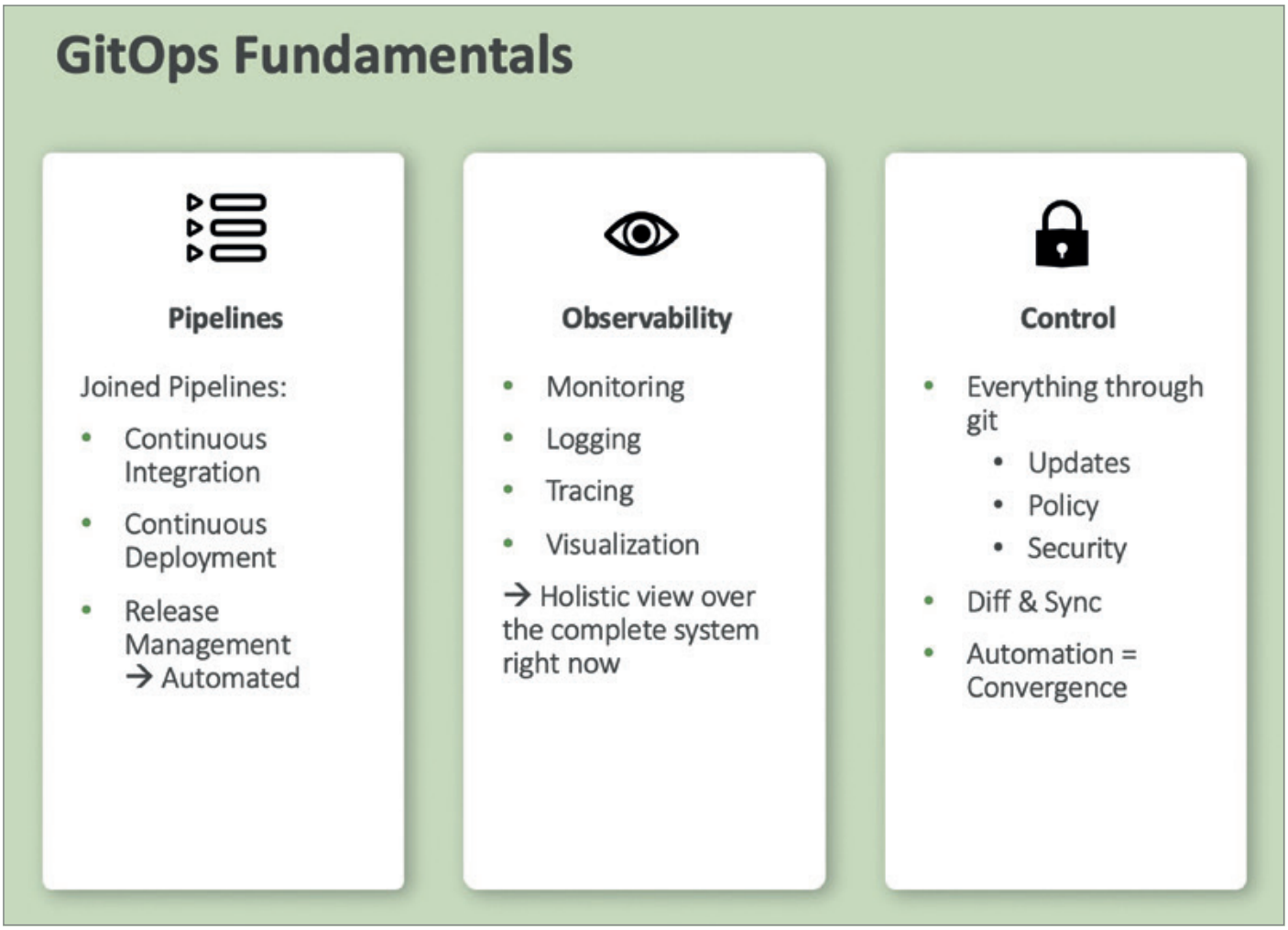


Abbildung 4: Die 3 Säulen, die GitOps als Fundamente nutzt (Quelle: Moritz Reinwald)

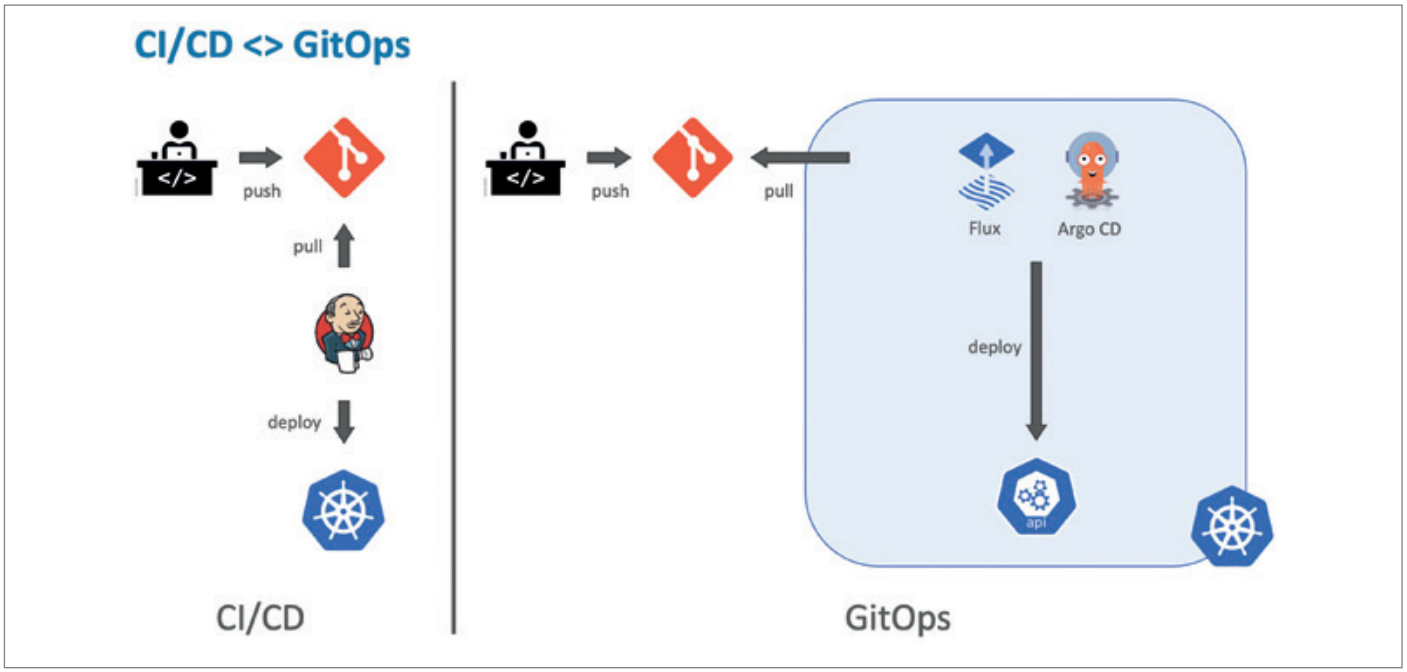


Abbildung 5: Der übliche Deployment-Prozess unterscheidet sich bei herkömmlichem CI/CD und GitOps (Quelle: Moritz Reinwald)

Werden Änderungen festgestellt, so werden diese durch den Operator entsprechend deployt und somit die Infrastruktur wieder in den definierten Zustand überführt. Trotz, oder vor allem auf Grund, dieses hohen Grades an Automatisierung sollte Monitoring hierbei eine wichtige Rolle einnehmen. Dies kann beispielsweise durch automatische Überwachung der Systeme und entsprechende Statusmeldungen, beziehungsweise Warnungen an Slack, MS Teams oder per E-Mail geschehen. Auch der Operator selbst sollte durch diese Maßnahmen überwacht werden, denn er ist das Herzstück dieses Prozesses.

Der Prozess in *Abbildung 6* beginnt mit einer Änderung an einer Anwendung, welche durch einen Commit und Push in das Application Repository durchgeführt wird. Der Push in das Repository triggert dann eine entsprechende Build Pipeline, welche die Anwendung baut, testet und das fertige Image anschließend in das Image Repository ablegt. Zeitgleich wird im Environment Repository die Infrastrukturdefinition geupdatet, indem nun die neue Version der App genutzt werden soll.

Der GitOps Operator überwacht kontinuierlich dieses Repository und stellt nun fest, dass sich der Ist- und Soll-Zustand der Infrastruktur unterscheidet. Aufgrund dessen wird nun die neue Infrastrukturdefinition auf die entsprechende Umgebung gepullt und umgesetzt beziehungsweise deployt.

Die *Abbildung 7* zeigt den Lebenszyklus eines Service bei Nutzung von GitOps. Start- und Endpunkt – und damit zentraler Bestandteil – ist jeweils das Git Repository. Zuerst wird das Release aus dem Git Repository auf der Infrastruktur installiert, hier dargestellt durch ein Kubernetes Cluster. Anschließend wird mit Monitoring- und Logging-Werkzeugen wie etwa Prometheus *[19]* beobachtet, ob der Service so funktioniert wie gewünscht, beziehungsweise wie er im Repository definiert wurde. Ist dies nicht der Fall – das heißt, es ist ein Configuration Drift entstanden – informieren die Werkzeuge entsprechend. Dann gilt es, den oder die Fehler sowie die Ursache zu analysieren, hier können die eben genannten Werkzeuge erneut unterstützen. Anschließend muss entschieden werden, wie das Problem nun angegangen wird, gefolgt vom Handeln und der Umsetzung inklusive Push in das Git Repository, wo der Zyklus dann erneut beginnt.

GitOps, DevOps, CI/CD – wie, wo, was?

Auch wenn GitOps, DevOps und CI/CD oft in einem Atemzug genannt werden und wie Zahnräder ineinandergreifen, sind es unterschiedliche Dinge. *Abbildung 8* verdeutlicht diese Unterschiede. Während DevOps eher eine Arbeitsweise innerhalb eines Unternehmens darstellt und daher Kommunikation und Zusammenarbeit die Hauptthemen sind, ist GitOps technisch deutlich konkreter. Hier dreht sich alles um die Automatisierung von Infrastruktur. Im Vergleich zu CI/CD wird bei GitOps der Zielzustand deklarativ definiert, während CI/CD eher einem imperativen Ansatz folgt, das heißt, der genaue Ablauf und wie der Zielzustand erreicht werden kann, wird vorgegeben. Zudem wird bei GitOps oftmals ein Pull-Based-Ansatz genutzt im Gegensatz zum Push-Based-Ansatz von CI/CD Build Pipelines.

Warum GitOps?

Oftmals wird bei Automatisierung mit dem Software Development Lifecycle begonnen, wodurch in diesem Bereich meistens bereits eine gewisse Automatisierung vorhanden ist, während der Infrastrukturbereich in vielen Fällen in Rückstand gerät. Demnach werden viele Prozesse im Infrastruktur-Deployment nach wie vor manuell ausgeführt und Umgebungen daher auch oftmals unterschiedlich konfiguriert. Zudem werden für diese Aufgaben spezialisierte Teams benötigt.

Eine weitere Herausforderung sind die heutigen Anforderungen an Infrastruktur.

So soll diese mittlerweile „elastisch“ sein, also schnell, skalierbar und auch Cloud-Ressourcen miteinbeziehen.

GitOps bietet hier einige Vorteile. Deployments können schneller und öfter durchgeführt werden, dementsprechend wird der Ansatz des kontinuierlichen Deployments verfolgt. Bei Fehlern und Abstürzen kann Infrastruktur schnell wieder aufgebaut werden, da sämtliche Definitionen und Änderungen versioniert sind. Zudem werden hauptsächlich Werkzeuge genutzt, die aus der Entwicklung bereits bekannt sind, zum Beispiel Git als Kernkomponente. Um neue Versionen zu deployen wird außerdem kein Zugriff auf die direkte Infrastruktur benötigt. Lediglich auf das beziehungsweise die entsprechenden Repositories. Durch die Arbeitsweise über das Repository inkl. Merge/Pull Requests werden sämtliche Änderungen automatisch dokumentiert und durchlaufen entsprechende Freigabeprozesse. Durch Code Reviews und sinnvolle Commit-Nachrichten sind Änderungen so für das gesamte Team nachvollzieh- und reproduzierbar.

Werkzeuge rund um GitOps

Soll GitOps nun umgesetzt werden, ist die Nutzung einiger darauf spezialisierter Werkzeuge sinnvoll und nötig. Eines dieser Werkzeuge ist wie bereits erwähnt Flux, ein GitOps Operator entwickelt von Weaveworks, das auch von GitOps ursprünglich definiert wurde . Flux ist zudem ein Projekt der Cloud Native Computing Fundation (CNCF). Für Version 2 wurde Flux von Grund auf umgeschrieben und nutzt nun das Kubernetes-API-Extension-System, kann also direkt mit Kubernetes interagieren. Da die ursprüngliche Version von Flux nur noch Sicherheitsupdates erhält, empfiehlt es sich, direkt Version 2 zu nutzen. Flux ist in der GitOps-Szene weit verbreitet und wird von einer großen und aktiven Community unterstützt.

Als Alternative sei noch ArgoCD erwähnt. Dieser GitOps Operator verfolgt einen grafischen Ansatz, bei dem sämtliche Aktivitäten über eine grafische Oberfläche getätigt und kontrolliert werden. ArgoCD bezeichnet sich selbst als deklaratives CD-Werkzeug. Eine Übersicht über die Funktionalitäten sowie die Oberfläche gibt die offizielle ArgoCD-Webseite .

Zu guter Letzt bedeutet die Nutzung von GitOps nicht zwangsweise, dass Kubernetes genutzt werden muss, auch wenn der Großteil dies so handhabt und die Idee GitOps aus dieser Richtung stammt. Grundsätzlich lässt sich GitOps allerdings für jegliche Infrastruktur anwenden, die überwacht und deklarativ beschrieben werden kann. Zusätzlich sollte die Infrastruktur über entsprechende Infrastructure-as-Code-Werkzeuge zu verwalten sein. Solche Werkzeuge sind beispielsweise Ansible, Chef, Puppet oder Terraform.

Fazit

GitOps bringt DevOps-Praktiken in die Infrastrukturverwaltung. Die Definitionen und Konfigurationen von Ressourcen werden als Soll-Zustand in Form von Code erstellt und in der Versionskontrolle verwaltet. Der Aufbau und die Anwendung von Änderungen werden automatisiert über entsprechende Werkzeuge durchgeführt. Dank Versionskontrolle werden Änderungen für das ganze Team nachvollziehbar, zusätzlich sorgen entsprechende Freigabe-Prozesse – Stichwort Merge/Pull Request – für eine gute Code-Qualität und schützen vor versehentlichen oder unbedachten Änderungen an der Infrastruktur. Änderungen an Ressourcen erfolgen ausschließlich im Code und über die Versionskontrolle, die Infrastruktur ist, einmal aufgebaut, unveränderbar. Durch geeignete Werkzeuge wird der Ist-Zustand der Ressourcen dauerhaft überwacht, bei Abweichun-

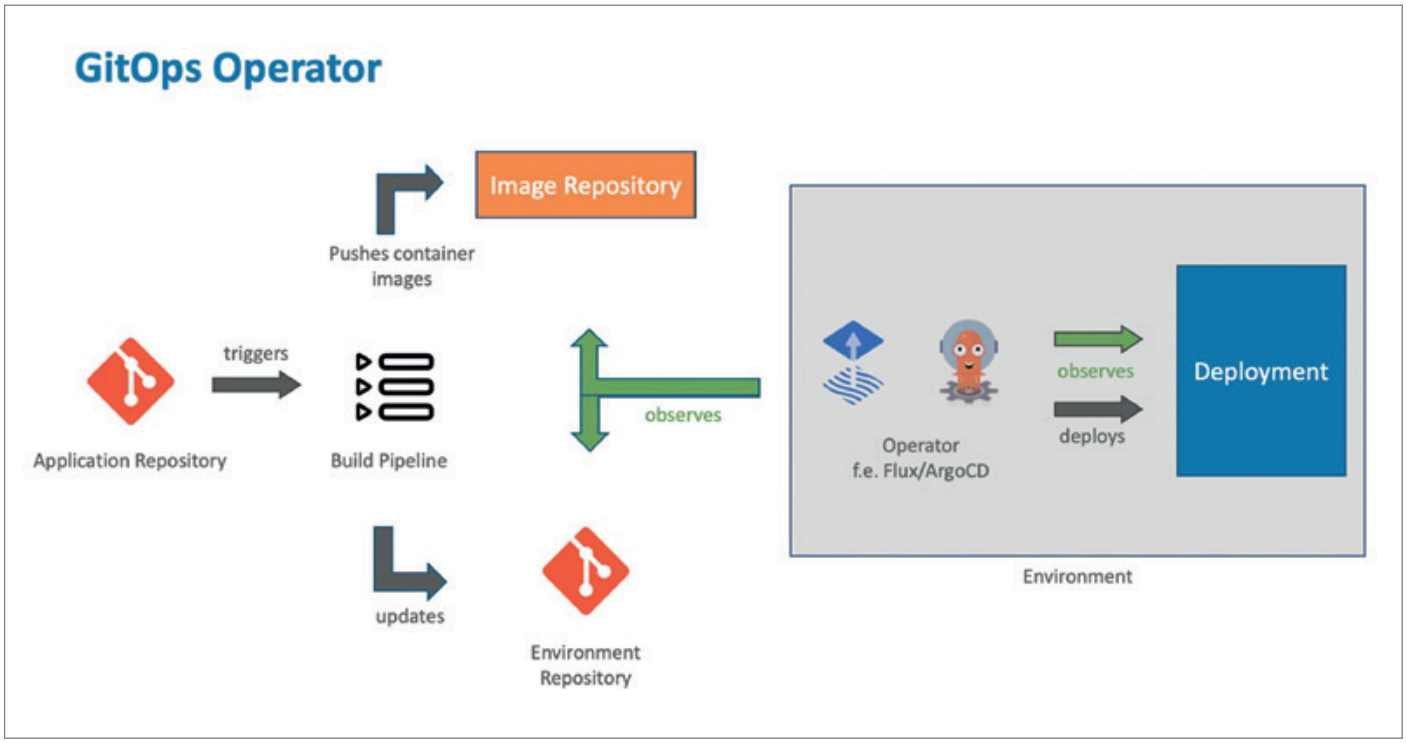


Abbildung 6: Zeigt einen beispielhaften GitOps-Deployment-Prozess mit verschiedenen Repositories, Operator und Build Pipeline (Quelle: Moritz Reinwald)

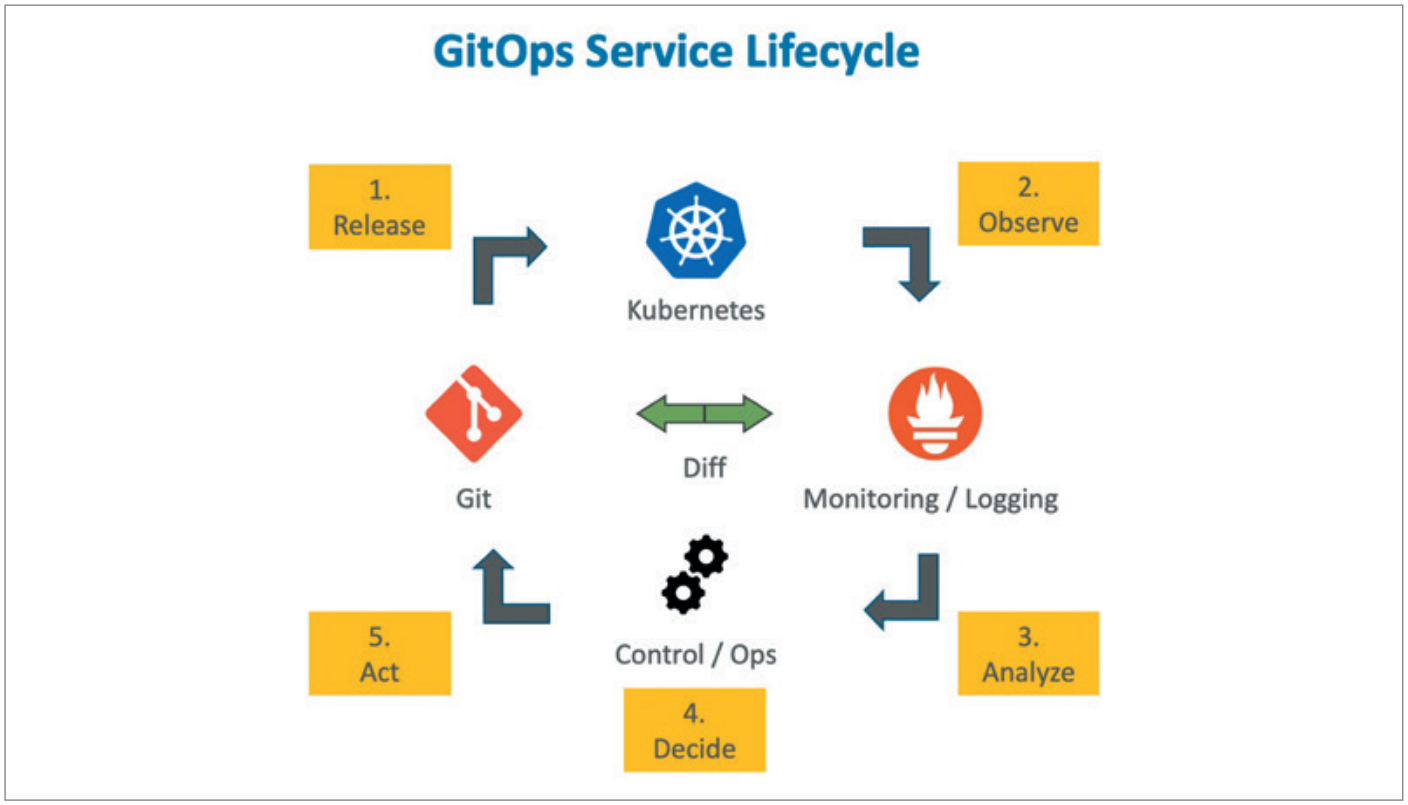


Abbildung 7: Der GitOps Service Lifecycle (Quelle: Moritz Reinwald)

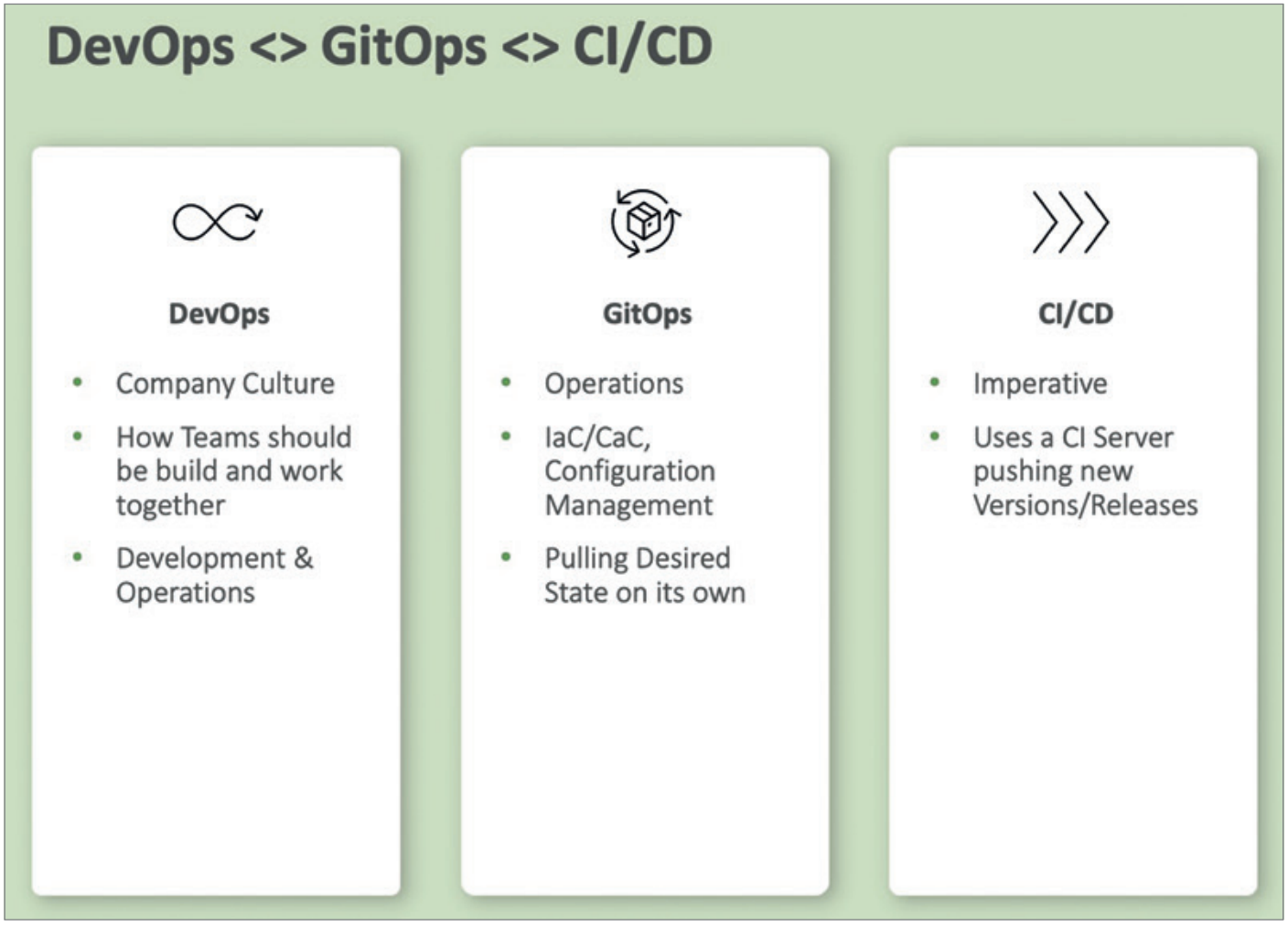


Abbildung 8: DevOps vs. GitOps vs. CI/CD (Quelle: Moritz Reinwald)

gen vom definierten Soll-Zustand wird die Infrastruktur automatisiert wieder in den gewünschten Zustand überführt. Somit können die Anforderungen von Entwicklungsteams erfüllt und diese mit schneller und skalierbarer Infrastruktur unterstützt werden.

Abschließend ein Zitat von Kelsey Hightower, einem der bekanntesten Entwickler der Kubernetes-, Open-Source- und Cloud-Computing-Szene *[20]*:

„GitOps is the best thing since configuration as code. Git changed how we collaborate, but declarative configuration is the key to dealing with infrastructure at scale, and sets the stage for the next generation of management tools".

Quellen und weitere Informationen

[1] DevOps-Zyklus in Anlehnung an *https://medium.com/@neonrocket/devops-is-a-culture-not-a-role-be1bed149b0*
[2] Atlassian Definition DevOps: *https://de.atlassian.com/devops*
[3] Chef *https://www.chef.io/*
[4] Puppet *https://www.puppet.com/*
[5] Ansible *https://www.ansible.com/*
[6] Docker *https://www.docker.com/*
[7] Kubernetes *https://kubernetes.io/de/*
[8] containerd *https://containerd.io/*
[9] CRI-O *https://cri-o.io/*[10] runc *https://github.com/opencontainers/runc*
[11] GitLab Definition GitOps: *https://about.gitlab.com/topics/gitops/*
[12] Kubediff *https://github.com/weaveworks/kubediff*
[13] Terraform *https://www.terraform.io/*
[14] Terradiff *https://github.com/jml/terradiff*
[15] Ansible Diff Mode *https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_checkmode.html*
[16] Weaveworks über Diffs *https://www.weave.works/technologies/weaveworks-aws-how-we-manage-kubernetes-clusters/*
[17] Flux *https://fluxcd.io/*
[18] ArgoCD *https://argoproj.github.io/cd/*
[19] Prometheus *https://prometheus.io/*
[20] Kelsey Hightower zu GitOps *https://twitter.com/kelseyhightower/status/1164192321891528704*

Über den Autor

Moritz Reinwald arbeitet als Berater mit dem Schwerpunkt APEX Development und DevOps bei der MT GmbH in Ratingen. Neben der Erstellung von Webanwendungen mithilfe von APEX, liegen seine Schwerpunkte im Bereich CI/CD mit Jenkins und Testautomatisierung. Zusätzlich beschäftigt er sich mit Technologien rund um das Thema DevOps, insbesondere in Verbindung mit Datenbanken.

Moritz Reinwald
moritz.reinwald@mt-itsolutions.com

APEX 23.1 – New Features

Florian Graßhoff, Oracle Global Services Germany

Mit dem APEX 23.1 Release gibt es wieder einige faszinierende und hochrelevante Features zu entdecken. Es gibt nun eine mächtige Template Component, deklarative Push Notifications, Graph-Unterstützung und vieles mehr. Diese bahnbrechenden Funktionen verleihen der Anwendung einen regelrechten Schub und lassen sie wie eine hochmoderne Rakete im weiten Raum des APEX-Entwicklungsuniversums erscheinen. Das Release befähigt zusätzlich, Prozesse im Hintergrund ausführen zu können und eine Vielzahl von Verbesserungen betreffen die Developer Experience. Die Voraussetzung für APEX 23.1 ist allerdings die Verwendung einer Oracle-Datenbank mit der Version 19c oder höher.

Eine kleine Reise durch das Feature-Universum

Voller Euphorie wollen wir gleich alle neuen Features der APEX-Version 23.1 entdecken, doch wo fängt man am besten an? Wegwerfen war schon bei meiner Oma nicht gerne gesehen, das war eher auf materielle- und Konsumgüter bezogen, aber ich übertrage das auch gerne auf meine Zeit, mit der ich ebenfalls sorgsam und sparsam umgehen möchte. Wir produzieren alle nicht gerne für den Papierkorb, daher entwerfen wir doch einfach direkt eine neue Applikation und überlegen uns dazu einen brauchbaren Use-Case.

Am 01. Februar 2023 wurde das Oracle Forums durch eine APEX-Anwendung abgelöst und die bestehenden Inhalte migriert. Ein wesentlicher Bestandteil eines Forums ist die Kommentarfunktion. Wir schauen nun gemeinsam, wie eine APEX-App umgesetzt werden könnte, bei der die Benutzer Kommentare schreiben können und dabei verschiedene Benachrichtigungen erhalten.

Überarbeiteter Object Browser

Was kommt zuerst? Klar, das Datenmodell. Wir brauchen eine Datenablage für unsere Benutzer und deren Kommentare, was uns die Gelegenheit bietet, den überarbeiteten Object Browser zu erkunden *(siehe Abbildung 1)*.

Erfahrene APEX Developer werden direkt optisch positiv überrascht sein, denn der Object Browser wurde umfänglich modernisiert, was mit einem neuen Look & Feel einhergeht. Auf der rechten Seite *(siehe Abbildung 1)* haben wir eine Übersicht über Typen von Datenbankobjekten, die uns nicht nur mit Erläuterungen versorgt, sondern auch mit nur einem Klick ermöglicht, das jeweilige Fenster zu öffnen, um ein neues Objekt zu erstellen. Davon machen wir auch Gebrauch, um eine neue Tabelle für unsere Anwendung zu erzeugen. Anstelle eines Wizards, erscheint nun ein einziges Modal Window, welches uns übersichtlich, alle notwendigen Eingabefelder bietet. Generieren wir also die Tabelle „Comments“ mit einer ID, dem Erstellungsdatum, dem Kommentartext und zur Vereinfachung den Benutzernamen (anstelle einer Identifikationsnummer für den Benutzer).

Zu unserer linken Seite *(siehe Abbildung 1)* werden für uns alle Datenbankobjekte unseres Schemas gelistet. Mit einem Rechtsklick auf ein beliebiges Objekt kann alles eingeklappt oder ausgeklappt werden, was eine schnelle Navigation ermöglicht. Zusätzlich gibt es ein Suchfeld, welches durch eine „Like“-Suche, die gelisteten Objekte einschränkt und ebenfalls zur erleichterten Bedienung und Navigation beiträgt. Für unser Beispiel suchen wir nach „Comments“ und bekommen die Tabelle, den automatisch generierten Primary Key und das „COMMENTS_PKG“ angezeigt, welches ich mit ein paar Klicks in APEX generieren lassen habe. Dies ist nichts Neues, allerdings sehr nützlich, vor allem in der Verbindung mit dem Invoke API Feature, auf das wir noch eingehen werden. Wer es noch nicht kennt, Packages mit grundlegenden Funktionen (Insert, Update, Delete) können auf der Basis von Tabellen hier generiert werden: SQL-Workshop → Utilitys → Methods on Tables.

Zusammengefasst ist der Object Browser nun attraktiver für das Auge, schneller, praktischer und er wurde in puncto Accessibility vollständig überarbeitet.

Template Components

Wir ziehen weiter und überlegen uns, wie eine Seite aufgebaut sein müsste, um unseren Anwendungsfall zu realisieren. Die Basiskomponente ist die Kommentarfunktion und wie es der Zufall so will, haben wir praktischerweise in 23.1 neue Vorlagen für die Schnellauswahl, dem „Seite erstellen“-Assistenten hinzugefügt, darunter auch die Vorlage für „Comments“.

Das Besondere ist hier, dass diese Komponenten (Avatar, Badge, Comments, Content Row, Media List, Timeline) mit dem neuen Plugin-Typ „Template Component“ umgesetzt worden sind, welches keine PL/SQL-Programmierung erfordert. Es ist jetzt für alle APEX Developer möglich, wiederverwendbare UI-Komponenten zu erstellen und diese in Reports oder als eigenständige Regionen zu rendern. Für die Wiederverwendbarkeit werden Aktionen und Menüs mit benutzerdefinierten Attributen unterstützt, diese können im Page Designer definiert und zugewiesen werden. Dazu lassen sich die Template Components abonnieren. Interessant ist auch die Möglichkeit des Exports, zum Beispiel kann somit die APEX- Community ihre großartigen Ideen und Umsetzungen einfach miteinander teilen. Die bereits erwähnten vom APEX-Team zur Verfügung gestellten Komponenten lassen sich übrigens kopieren, sodass einfaches Customizing möglich ist.

Glücklich und zufrieden, dass uns einiges an Arbeit und Zeit erspart wurde, setzen wir unsere Reise mit der „Comment“ Region fort, weisen im Page Designer unsere Tabelle als Datenquelle zu und mappen unsere Tabellenspalten mit den jeweiligen Attributen *(siehe Abbildung 2)*.

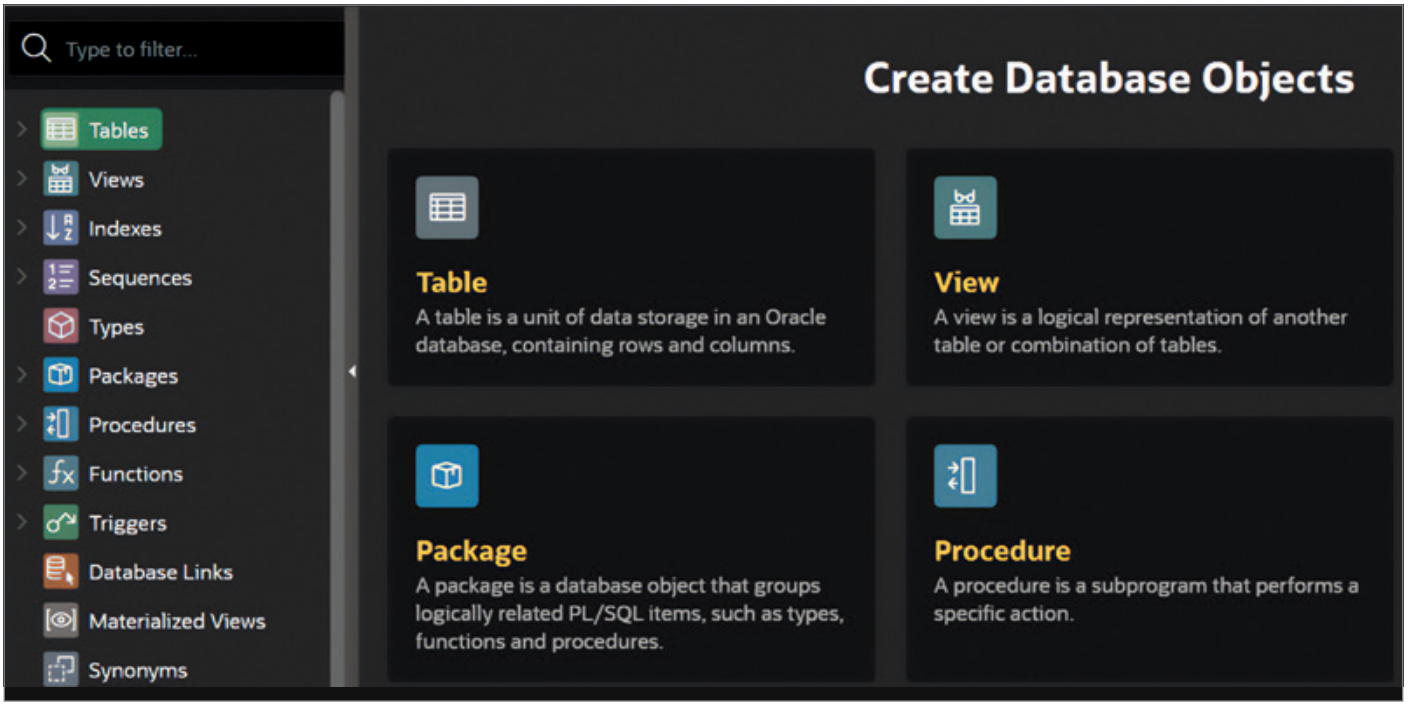


Abbildung 1: Object-Browser-Übersicht – Ausschnitt (Quelle: Florian Graßhoff)

Zum Funktionsumfang sollte aber auch gehören, dass Benutzer ihre Kommentare bearbeiten und Administratoren zusätzlich löschen können. Gesagt, getan. Wir fügen unserer „Comment"-Region eine Aktion hinzu, nennen diese „Bearbeiten", verweisen auf eine neue Seite und übergeben dabei die Kommentar-ID als Parameter. Die neue Bearbeitungsseite können wir mühelos mit unserem Formular-Wizard erstellen, dann noch entsprechende Autorisierungseinstellungen vornehmen und damit haben wir mit ein paar wenigen Klicks die Anforderung umgesetzt. Klasse, damit können wir unsere Kommentare (künstliche Testdaten) in einer typischen Chat-UI darstellen lassen *(siehe Abbildung 3)*.

Invoke API

Existierende Kommentare können wir nun in unserer App anzeigen, aber neue Kommentare sollten von den Benutzern auch verfasst werden können.

Das zu implementieren, braucht nicht viel. Wir erstellen als Eingabefeld ein Rich Text Editor Page Item, dazu noch einen Button, welcher beim Klick einen Page Submit ausführt und einen Prozess, um den neuen Kommentar zu speichern.

Hier bringen wir dann auch den Prozess-Typ „Invoke API" ins Spiel. Dies ermöglicht uns, unsere automatisch generierte Prozedur „INS_COMMENTS" aus dem Package „COMMENTS_PKG" auszuwählen, die Parameter deklarativ zu mappen und schon sind wir fertig. Die Krux liegt, wie wir wissen in den Details, aber auch hier kann sich das Feature stolz und selbstbewusst zeigen. Out-Parameter lassen sich deklarativ verwalten, benutzerdefinierte Datentypen werden unterstützt und anhand der Namensgebungen der Page Items und Parameter werden automatische Mappings durchgeführt.

Dieses Feature kam mit der Version 22.2 und mit der Version 23.1 werden nun auch REST Data Sources unterstützt. Es ist nicht ungewöhnlich, dass kein direkter Zugriff auf die Datenquelle existiert und stattdessen ein Webservice, sowohl für die Abfrage von Daten als auch für das Anlegen und Manipulieren, genutzt wird. Um dies auf unser Beispiel zu übertragen, legen wir beispielhaft einen REST-Service an, erzeugen die Rest Data Source und verwenden diese mit dem Invoke-API-Prozess *(siehe Abbildung 4)*, um einen neuen Kommentar hinzuzufügen.

Deklarative Push-Benachrichtigungen

Was bisher eine technische Herausforderung war, und einer Eigenimplementierung bedarf, ist nun sensationell einfach gelöst, Push-Benachrichtigungen im Browser oder in der installierten Progressive Web-App. Das wollen wir auch und legen als App-Benutzer gleich den richtigen Schalter um, welchen wir im „Settings"-Menü *(siehe Abbildung 5)* finden, damit wir Push Notifications erlauben. Von vielen anderen Applikationen ist man es gewohnt, dass initial erst einmal die Berechtigungen angefragt werden, was, ähnlich zu den Cookie-Fenstern, als störend empfunden wird. Wir wollen hier einen anderen Weg gehen. Aktuell beinhaltet das „Settings"-Menü nur diese eine Einstellung, es ist absehbar, dass weitere Optionen folgen.

Vorstellbar wäre, dass ein Benutzer darüber benachrichtigt werden möchte, wenn ein Administrator darüber entscheidet, den Beitrag zu entfernen. Das zu realisieren, ist nun so leicht wie eine Wolke. Wir fügen im Page Designer einen Prozess vom Typ „Send Push Notification" hinzu und weisen nur noch einen Empfänger, den Titel und den gewünschten Text zu *(siehe Abbildung 6)*. Nach dem Löschen landet die Benachrichtigung in der Queue.

Mit der Version 23.1 sind auch zwei neue Views eingeführt worden. APEX_APPL_PUSH_SUBSCRIPTIONS liefert Informationen über die Subscriptions pro Applikation. APEX_PUSH_NOTIFICA-

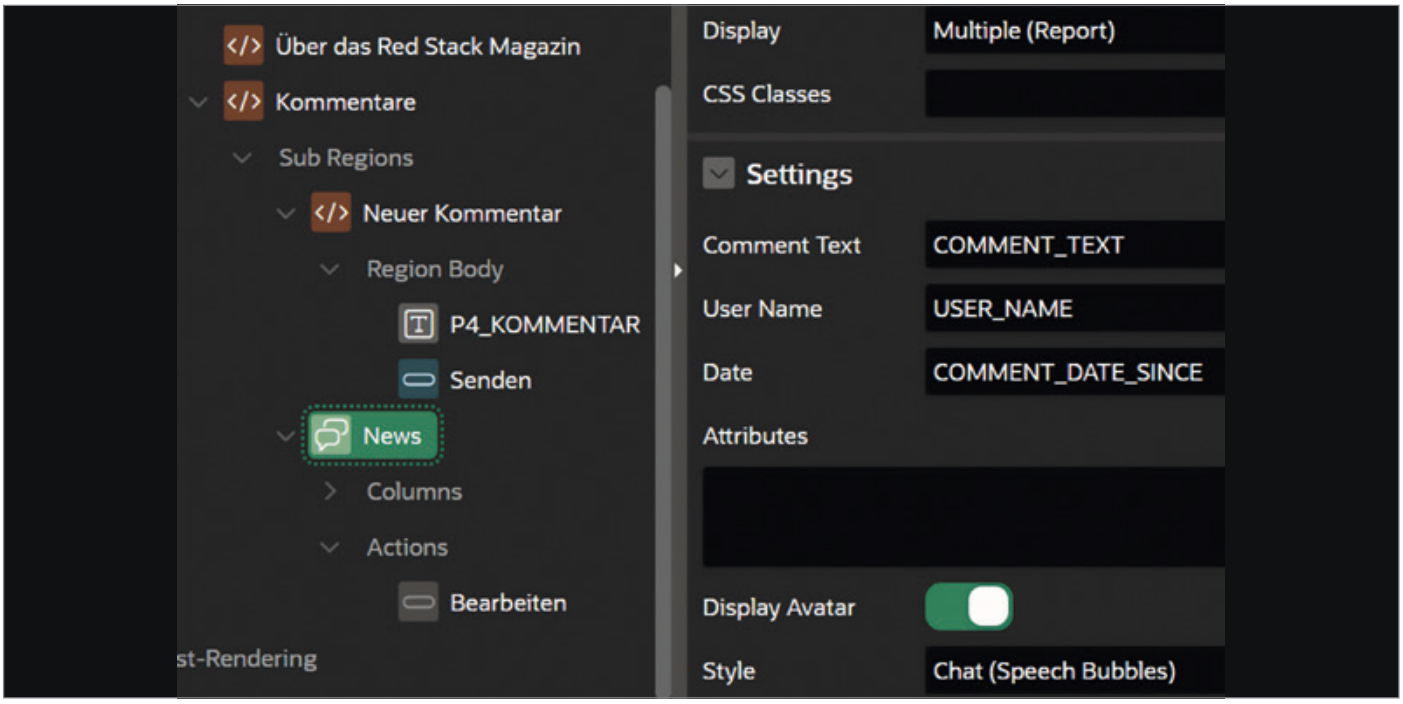


Abbildung 2: Page Designer – Comment-Region-Attribute (Quelle: Florian Graßhoff)

TIONS_QUEUE zeigt Push-Benachrichtigungen die noch nicht gesendet oder, ob dabei Fehler aufgetreten sind. Ein Job in der Datenbank sendet zyklisch die Benachrichtigungen aus der Queue.

Weitere Freiheiten und Kontrolle über das Verhalten sowie den Inhalt von Push- Benachrichtigungen gibt die API „APEX_PWA".

In unserem kleinen Szenario wäre die Anforderung denkbar, dass alle Subscriber benachrichtigt werden sollen, wenn ein neuer Kommentar verfasst worden ist, das könnte beispielsweise, wie in *Listing 1* dargestellt, implementiert werden.

Sehr empfehlenswert ist auch die APEX PWA Reference, hier gibt es Installationshinweise, ein FAQ, eine Roadmap und alle weiteren Informationen rund um das Thema PWA mit APEX.

Gruppierung und Hintergrund-Ausführung von Page-Prozessen

In unserer kleinen App haben wir bereits eine Abfolge von Ereignissen, die nach einem Button-Klick ausgeführt werden. Spinnt man das ganze Szenario weiter, so könnten beispielsweise folgende Prozesse nach dem Absenden eines neuen Kommentares anstehen:

1. Textinhalt prüfen (Hacking, verbotene Inhalte)
2. Dateianhänge prüfen (Upload-Filter)
3. Datensätze schreiben (inklusive allem, was nach einem INSERT passiert)
4. Benachrichtigungen

Ganz schnell wird es für den Benutzer langsam, Ladebalken (Spinner) werden zur Regel, die User Experience leidet, die Benutzerakzeptanz schwindet. Es gibt viele Prozesse die Zeit kosten, teilweise sind die Möglichkeiten des Developers auch sehr begrenzt, zum Beispiel bei einem Upload-Vorgang, der je nach Größe der Datei und den Übertragungsraten nun einmal eine gewisse Zeit in Anspruch nimmt.

Die Problematik wurde teilweise gelöst, indem die Developer Eigenimplementierungen umgesetzt haben, die es ermöglichen, den langlaufenden Prozess mit Scheduler-Jobs im Hintergrund ausführen zu lassen. Das ist aber alles andere als trivial und erfordert gute Kenntnisse über die Scheduler-Jobs in der Oracle-Datenbank.

In der APEX-Version 23.1 haben wir uns dieser Thematik angenommen und eine deklarative Lösung implementiert, die es möglich macht, Prozesse zu gruppieren und im Hintergrund ausführen zu lassen. Dafür gibt es den neuen Prozess-Typ „Execution Chain" mit der Einstellung „Execute in Background" *(siehe Abbildung 7)*.

Einer „Execution Chain" können jegliche Prozesstypen zugeordnet sein, auch weitere Prozesse vom Typ „Execution Chain". Ein Aktivieren der Einstellung „Execute in Background" bringt weitere Einstellungsmöglichkeiten hervor. Eine Session-übergreifende Serialisierung der „Execution Chain" ist einstellbar, Ausführungsgrenzen können definiert werden und ebenfalls das Handling von temporären Dateien. Weiterhin kann die Ausführungs-ID in ein Page Item geschrieben werden, was zum Beispiel die Überwachung des Hintergrundprozesses erleichtert, denn dafür wird uns vieles geboten. Es wurde eine neue View mit dem Namen „APEX_APPL_PAGE_BG_PROC_STATUS" eingeführt, die Informationen über aktuelle Hintergrundausführungen liefert. Dazu bringt die API „APEX_BACKGROUND_PROCESS" mit ihren Funktionalitäten Tools zum Abbrechen eines Prozesses und zum Abfragen/Setzen von Statusinformationen.

Das ist noch nicht alles, denn an diversen Stellen in APEX können aktuelle Hintergrundprozesse von den Developern und Administratoren überwacht und bei Bedarf beendet werden. Außerdem ist die Anzahl der parallelen Ausführungen in verschiedenen Ebenen limitierbar.

Abbildung 3: PWA App Screenshot 1 – Kommentare – Testdaten (Quelle: Florian Graßhoff)

Developer Experience

Die Version 23.1 ist vollgepackt mit schönen Sachen, die zum einen, das Developer-Leben schöner machen und zum anderen, einige neue großartige Features mitbringt. Eine Übersicht erhält man in den Release Notes. Ich möchte auch auf die APEX-Office-Hours-Sessions verweisen, die ebenfalls einen umfangreichen Einblick in die bedeutendsten Features der 23.1 geben. Die Aufzeichnungen findet man bei YouTube im „Oracle Developers"-Channel, hier einfach nach den Videos mit dem Titel „What's new in Oracle APEX 23.1" suchen.

Neben den „Marquee"-Features möchte ich gerne noch ein paar Sätze zur Developer Experience und den nicht so prominenten Features verlieren.

Das **APEX Administrator Digest** bietet dem Administrator der Instanz eine schöne Übersicht (Applikationen, Scheduler Jobs, Database Links, Speicherplatz uvm.), die in der Instanzverwaltung unter → Monitor Activity → Administrator Digest zu finden ist.

Der **SQL Developer** Web ist nun in APEX im SQL-Workshop-Menü integriert und somit mit zwei Klicks aufrufbar, die Aktivierung erfolgt in der Feature Configuration der Instanzverwaltung.

Die **REST-Source**-Discovery-Funktionalität funktioniert nun auch mit **Swagger/OpenAPI**, das Datenprofil kann jetzt also automatisch, zusammen mit anderen Metadaten aus der Response gewonnen werden.

Seit der Version 23c unterstützt die Oracle-Datenbank **Graph Queries in SQL** mit der Property Graph Query Language (PGQL). APEX unterstützt diese Neuerung vollumfänglich, denn für APEX-Komponenten kann nun der „Property Graph"-Typ als Datenquelle ausgewählt werden und SQL-Query-Datenquellen unterstützen die PGQL-Syntax. Dies gilt beispielsweise auch für LOVs und Suchkonfigurationen.

Das **Auto Provisioning** für Workspaces bietet nun die Möglichkeit zur Definition einer E-Mail Blocklist, basierend auf regulären Ausdrücken.

XLIFF Translation Files konnten bisher nur über die GUI hoch- und runtergeladen werden, jetzt ist dies auch über die APIs **„APEX_LANG.GET_XLIFF_DOCU-**

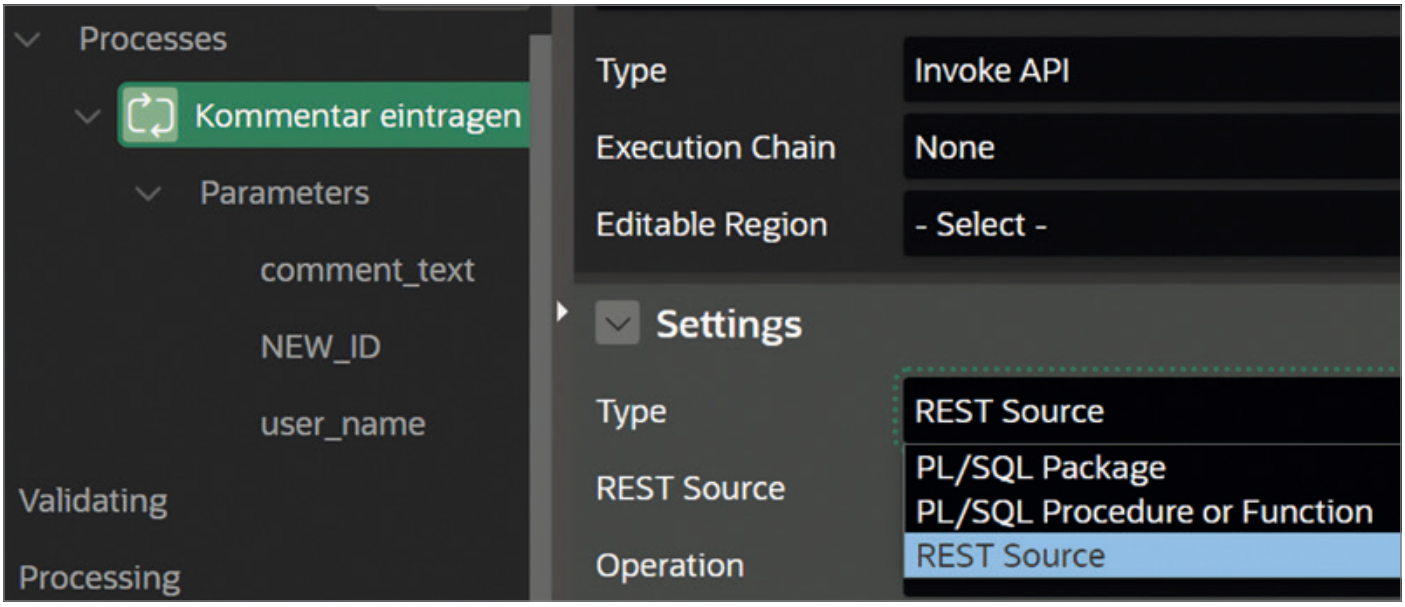


Abbildung 4: Page Designer – Invoke-API-REST-Region-Parameter (Quelle: Florian Graßhoff)

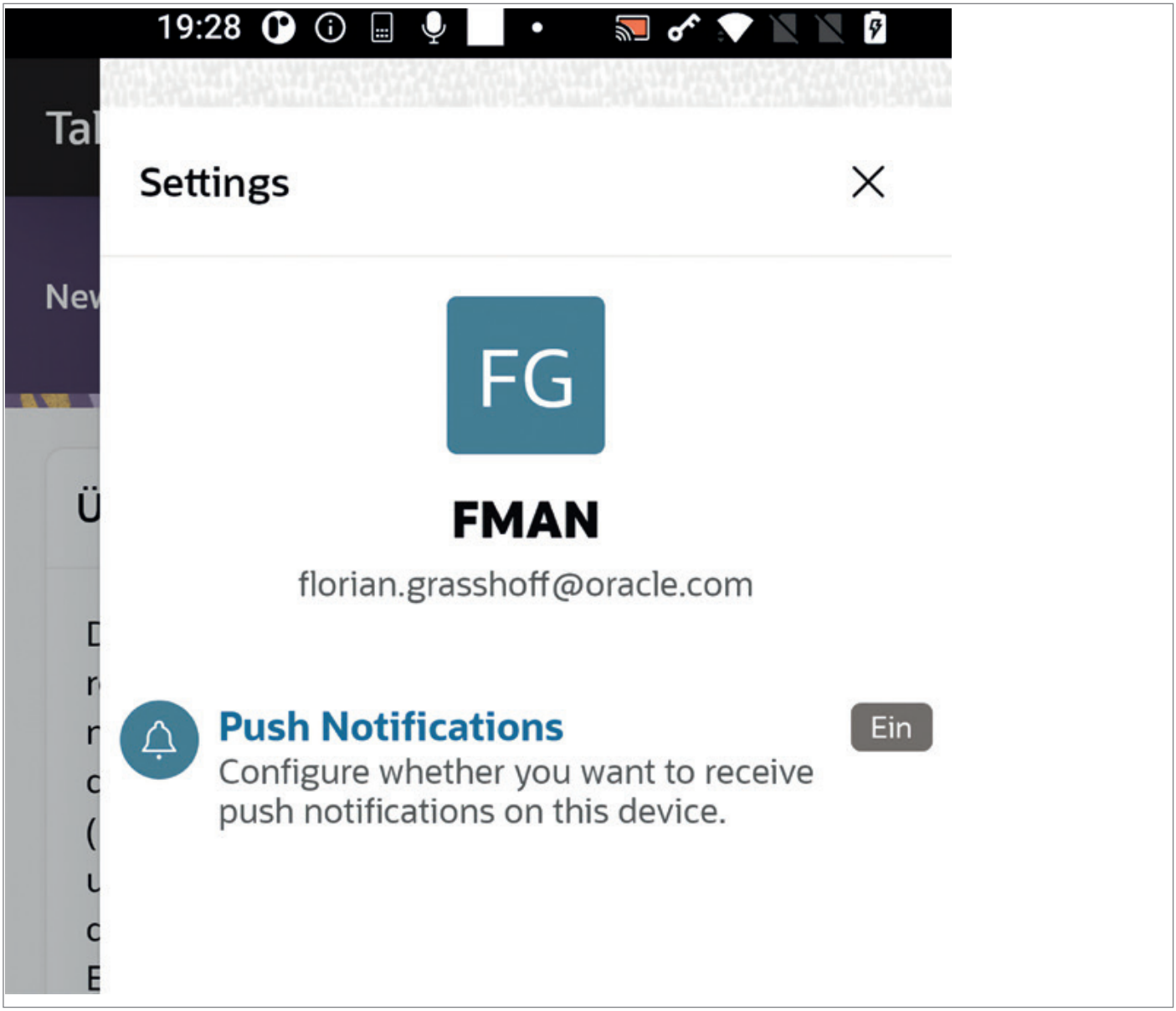


Abbildung 5: PWA App Screenshot 2 – Push Notification Settings (Quelle: Florian Graßhoff)

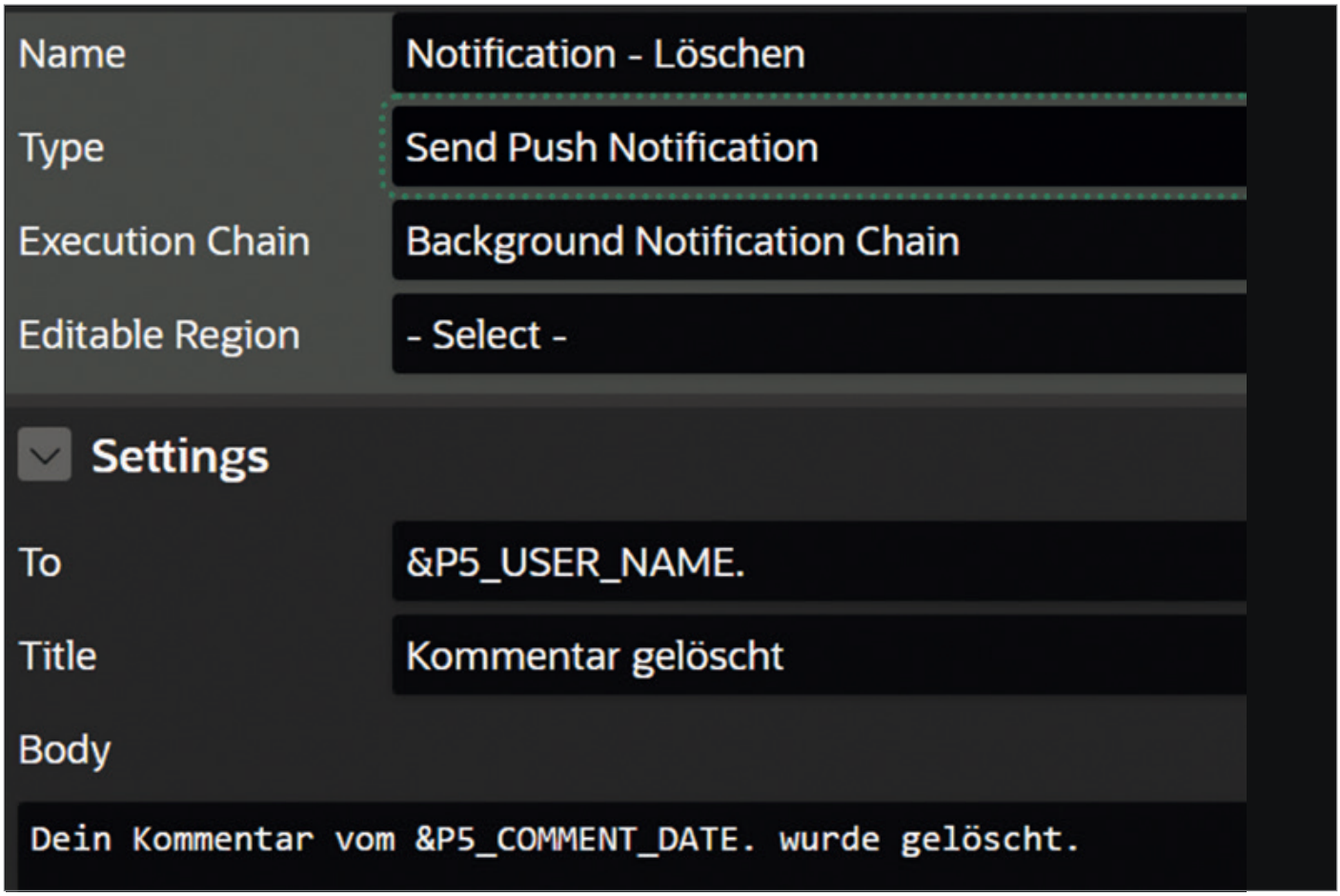


Abbildung 6: Page Designer – Push-Notification-Prozess (Quelle: Florian Graßhoff)

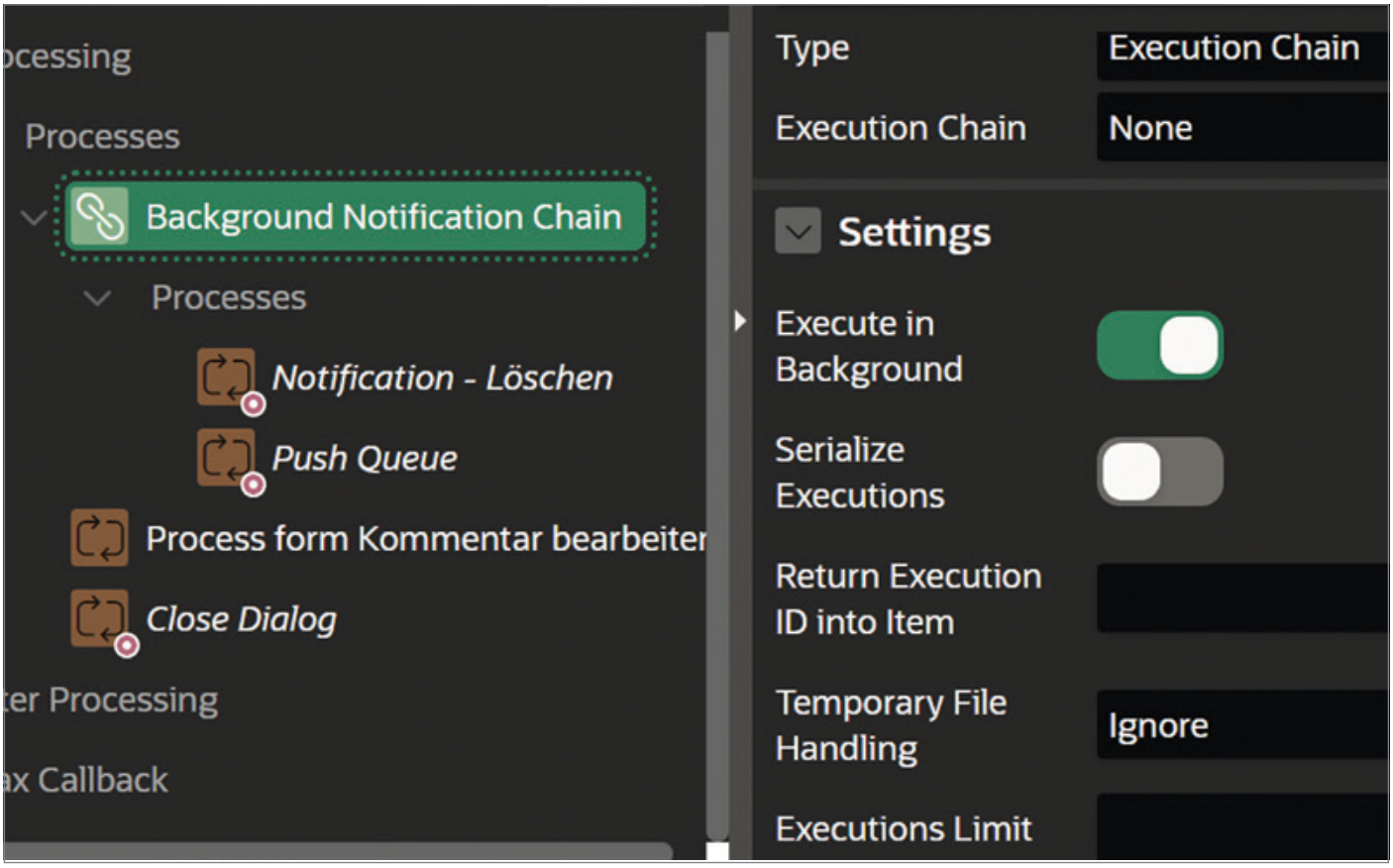


Abbildung 7: Page Designer – Execution-Chain-Prozess-Typ (Quelle: Florian Graßhoff)

```
begin
    for l_subscription in (
        select distinct user_name
          from apex_appl_push_subscriptions
         where application_id = :APP_ID )
    loop
        apex_pwa.send_push_notification (
            p_application_id => :APP_ID,
            p_user_name      => l_subscription.user_name,
            p_title          => 'Neuer Kommentar',
            p_body           => 'Einer neuer Kommentar…' );
    end loop;
end;
```

Listing 1: Code-Beispiel, Push-Benachrichtigung an alle Subscriber

MENT & APEX_LANG.APPLY_XLIFF_DOCUMENT“ möglich.

Die maximale Anzahl der Attribute für Item-Type Plug-Ins wurde von 15 auf 25 erhöht.

Fazit

Eine kleine Reise durch die APEX 23.1-Feature-Landschaft ist nicht ausreichend, es bedarf einer groß angelegten Expedition, um allen Neuerungen gerecht zu werden. Zeit ist Geld, jeder ist von Effizienz und Selbstoptimierung getrieben. Sich auf dem Laufenden zu halten, kann sich aber durchaus lohnen, denn einerseits können uns Eigenimplementierungen erspart bleiben und andererseits sind wir doch genau deshalb hier, wegen unseres gemeinsamen technischen Interesses, das uns in diesen Beruf oder Hobby geführt hat.

Weitere Informationen

- Informationen zu APEX und Demo-Umgebung
  https://apex.oracle.com
- Oracle APEX Blog
  https://blogs.oracle.com/apex/
- Oracle Forums
  https://forums.oracle.com
- Oracle APEX Release 23.1 Dokumentation
  https://docs.oracle.com/en/database/oracle/apex/23.1
- APEX PWA Reference
  https://apex.oracle.com/pls/apex/r/apex_pm/apex-pwa-reference/home
- Oracle Developers YouTube Channel
  https://www.youtube.com/@oracledevs

Über den Autor

Florian Graßhoff hat bereits in der Fachinformatiker-Ausbildung hauptsächlich mit Oracle-Datenbanken gearbeitet. Seitdem ist er der Oracle-Technologie treu geblieben. Zuletzt hat er als Lead Developer und stellvertretender Teamleiter, im Datenbank-Team, für einen IT-Dienstleister im Polizeiumfeld gearbeitet und ist im Juli 2022 in das APEX Development Team gewechselt. Im APEX-Team beschäftigt er sich hauptsächlich mit dem Application Search Feature und REST Data Sources.

Florian Graßhoff
Florian.Grasshoff@Oracle.com

Holen Sie als APEX-Entwickler das Beste aus Oracle Autonomous Databases heraus

Timo Herwix, MT

Suchen Sie nach einer sicheren, vollständig verwalteten Datenbankumgebung für die Entwicklung von Low-Code-Anwendungen? Mit Oracle Autonomous Database in der Oracle Cloud Infrastructure sind Sie bestens bedient. Als APEX-Entwickler können Sie das Potenzial dieser Datenbanken leicht maximieren. Lesen Sie mehr über Ansätze, wie eine „Always FREE Tier" für eine produktionsreife Architektur verwendet werden kann, wie Sie Ihre Identität mit benutzerdefinierten URLs stärken können und wie Sie Single Sign-on aktivieren. Und das ganz ohne Bedenken bezüglich der Cloud-Sicherheit!

Ist es möglich eine „Always FREE Tier" in einer produktionsbereiten Architektur zu verwenden?

Bevor Sie mit der Entwicklung beginnen, sollten Fragen zur Infrastruktur und Sicherheit geklärt werden: Wer soll auf die Anwendung zugreifen können? Soll der Zugriff auf eine bestimmte Gruppe von Nutzern beschränkt werden oder soll die Anwendung öffentlich zugänglich sein? Wie kritisch sind die Daten? Ist eine Multi-Faktor-Authentifizierung erforderlich?

Darüber hinaus ist es wichtig, den erforderlichen Speicherbedarf zu bestimmen und einen Plan für die Sicherung und Wiederherstellung der Daten zu entwickeln. Dafür stellt die Oracle Cloud Infrastructure eine Reihe von Ressourcen zur Verfügung, die Sie die ganze Zeit kostenlos nutzen können. Sicherlich ist in bestimmten Bereichen ein gewisser Mehraufwand erforderlich, aber das Endergebnis wird umso zufriedenstellender sein.

Können benutzerdefinierte URLs verwendet werden, um die Identität der Anwendung zu stärken?

Als Nutzer der Oracle Cloud habe ich mich oft gefragt, ob ich die automatisch generierte URL durch eine meiner eigenen Domains ersetzen kann. Die Antwort lautet: ja! OCI hat eine Möglichkeit geschaffen, die von Ihnen gewünschte benutzerdefinierte URL (Vanity URL) durch die Verwendung eines Load Balancers zu verwenden. Der Konfigurationsprozess ist einfach und kann schnell durchgeführt werden. Jedoch ist dies nicht in einer kostenlosen Instanz möglich, da eine autonome Datenbank einen privaten Endpunkt benötigt und der Prozess der Zertifikatserneuerung nicht automatisiert werden kann, was wiederum zu Mehraufwand führen würde.

Eine andere Möglichkeit ist die Verwendung eines NginX-Reverse-Proxy-Servers. Dazu benötigen Sie eine „Compute-Instanz", die die Anfragen an APEX auf

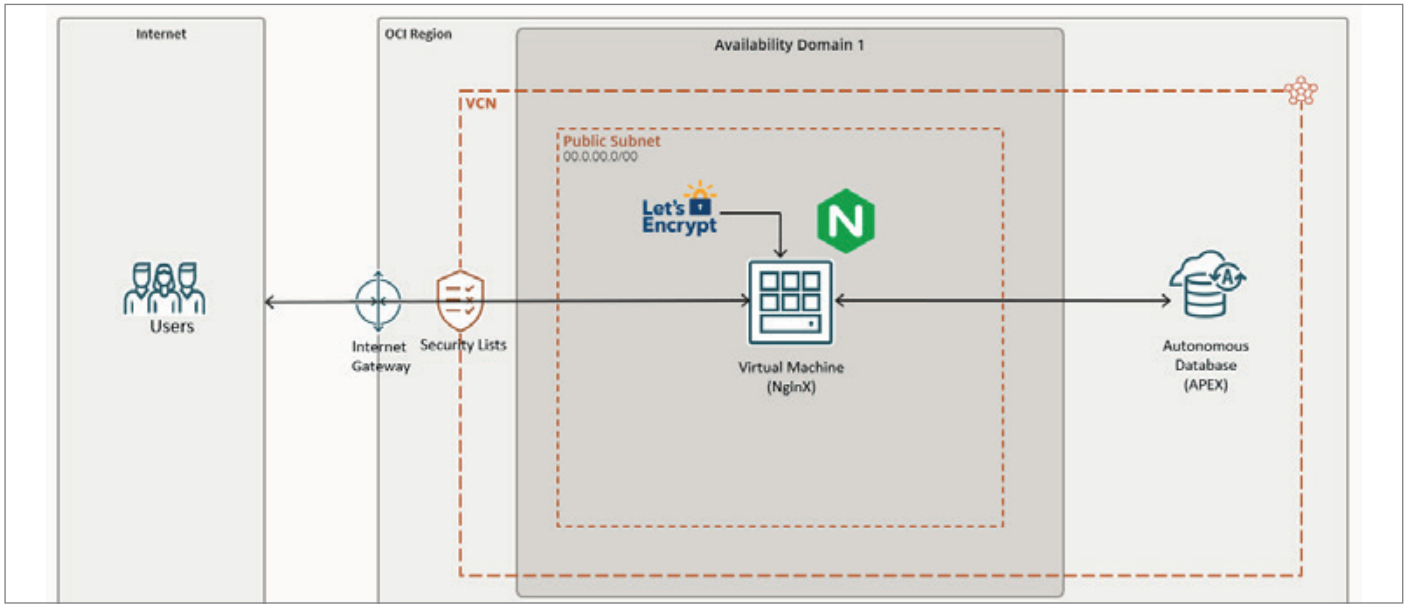


Abbildung 1: Architektur-Diagramm – NginX reverse proxy server (Quelle: Timo Herwix)

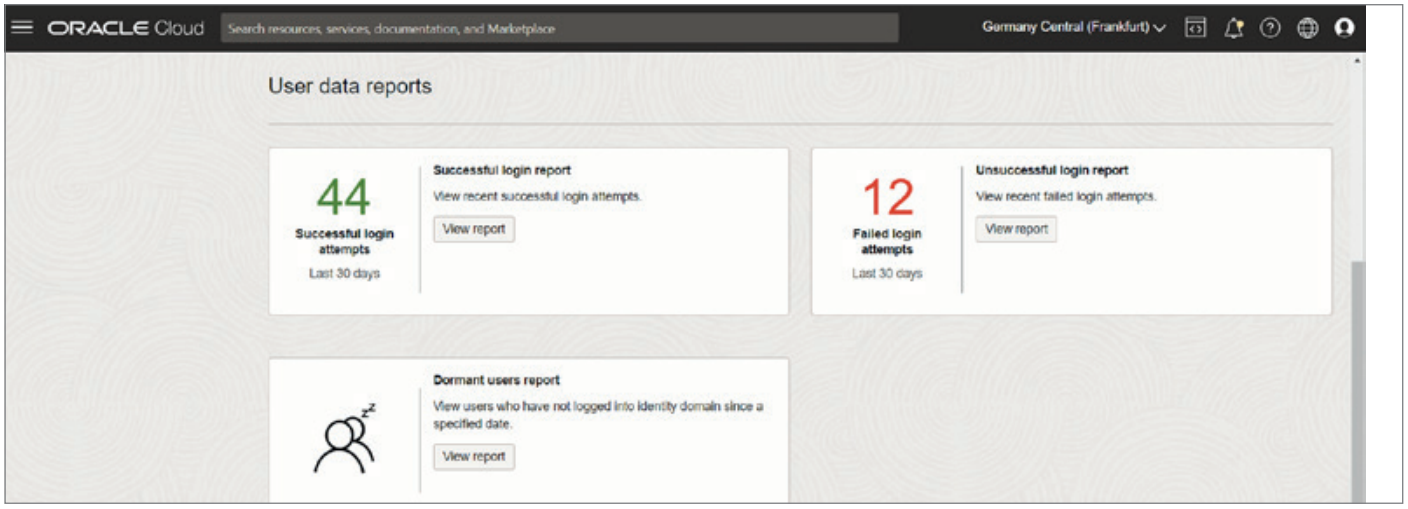


Abbildung 2: Beispiel Audit-Report (Quelle: Timo Herwix)

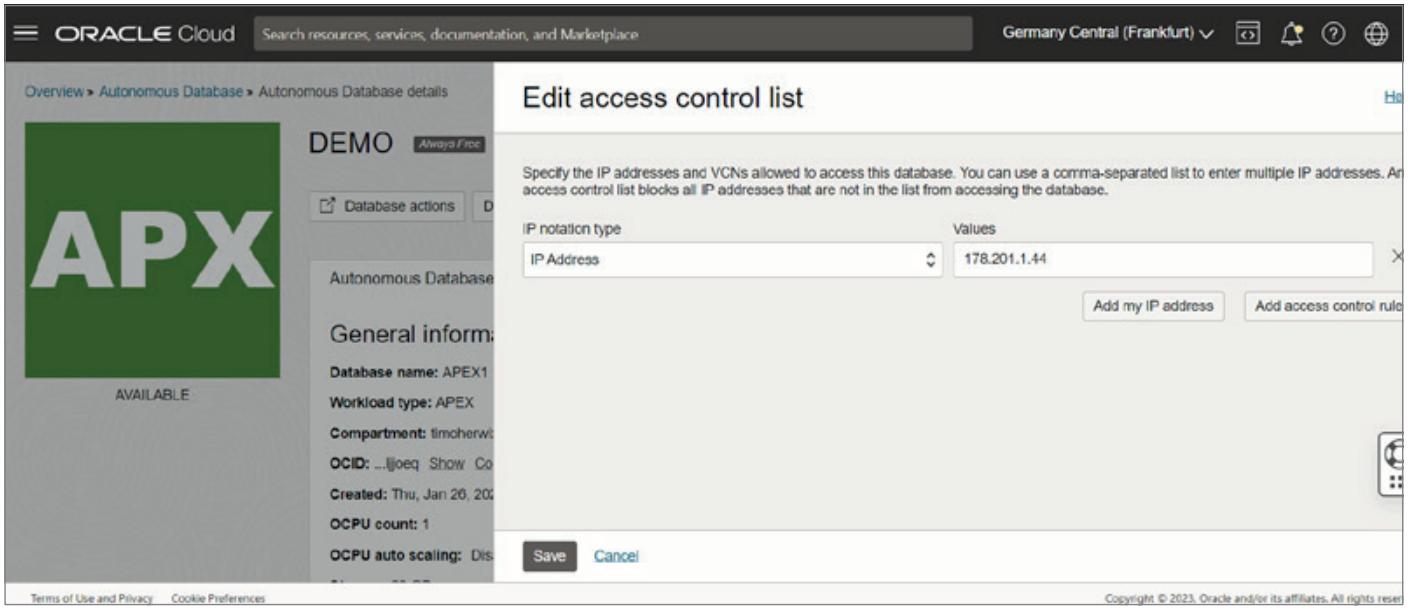


Abbildung 3: Nur bestimmten IP-Adressen den Zugriff auf die Datenbank erlauben (Quelle: Timo Herwix)

der ADB weiterleitet und „Let's Encrypt" verwendet, um eine kostenlose, automatisierte und offene Zertifizierungsstelle für die HTTPS(SSL/TLS)-Verschlüsselung zu ermöglichen. Im Grunde ist diese Methode nicht viel schwieriger als das, was OCI vorgeschlagen hat. Es gibt jedoch einige Einschränkungen. So können Sie beispielsweise kein Social Sign-On (SSO) verwenden. Dafür müssen Sie noch einen Schritt weiter gehen und Ihr eigenes benutzerdefiniertes ORDS auf der „Compute-Instanz" installieren.

Kann Single Sign-On verwendet werden?

SSO minimiert das Risiko von Angriffen, indem es die Anzahl der Einstiegspunkte begrenzt. Mit SSO authentifizieren sich die User beispielsweise nur einmal am Tag und greifen mit einem einzigen Satz von Anmeldedaten auf alle Anwendungen zu. Dieser Ansatz stärkt die Unternehmenssicherheit, indem er den Anmeldeprozess rationalisiert. Denn viele User nutzen oft dasselbe Passwort für verschiedene Anwendungen, was sie anfälliger für Sicherheitsverletzungen macht.

Die OCI-Authentifizierung ermöglicht eine schnelle und einfache Implementierung von SSO-Anmeldungen sowie die Option der Multifaktor-Authentifizierung (MFA). Außerdem können Anmeldungsrichtlinien erstellt werden, in denen festgelegt wird, welche Provider zugelassen sind und welche Benutzergruppen wie oft MFA benötigen. Ein weiterer wesentlicher Vorteil von SSO über OCI sind die umfassenden Audit-Berichte. Den größten Nutzen bietet jedoch die Möglichkeit, mehrere Anbieter (Azure etc.) zu unterstützen, ohne dass zusätzlicher Code erforderlich ist.

Ist die Cloud sicher und kann darauf vertraut werden, dass Daten sicher aufbewahrt sind?

Diese Frage kann man nicht seriös mit einem einfachen „Ja" beantworten. Das Sicherheitsniveau einer Cloud-Infrastruktur hängt von den Bemühungen des Entwicklers ab sowie von den Sicherheitsmaßnahmen, die bei der Erstellung der darauf ausgeführten Anwendungen getroffen werden. Zum Glück gibt es verschiedene Möglichkeiten, eine sichere Cloud-Umgebung zu schaffen.

Mit Hilfe von Kontroll-Listen können Sie beispielsweise Regeln für den Netzwerkzugriff festlegen, die Ihre autonome Datenbank schützen, indem sie nur bekannten IP-Adressen den Zugriff erlauben. Alle unbekannten IP-Adressen werden blockiert. Darüber hinaus lassen sich SSO und MFA leicht einrichten, um die Cloud-Sicherheit zu erhöhen. Mit OCI kann die Web-Application Firewall verwendet werden, um bösartigen Datenverkehr oder gefälschte Anfragen herauszufiltern. Wenn Sie einen eigenen ORDS-Server haben, können Sie auch das Modul „event MPM" verwenden, um die Anfragen in Ihrem Proxy-Server zu drosseln.

Gibt es einen Notfallplan?

Um eine FREE-Tier-Instanz in einer produktionsfähigen Architektur zu verwenden, ist es wichtig, eine schnelle und einfache Sicherungsoption für Ihre Datenbank zu haben. OCI bietet zwar inkrementelle Backups für FREE-Tier-ADB-Instanzen an, aber es ist nicht möglich, von diesen Backups wiederherzustellen. Um eine Datenbankwiederherstellung durchzuführen, ist ein Upgrade auf eine kostenpflichtige Instanz nötig. Das bedeutet, dass Sie für die Sicherung und Wiederherstellung Ihrer Umgebung selbst verantwortlich sind, wenn Sie eine kostenlose Instanz in einer Produktionsumgebung verwenden – was leider ein großer Nachteil ist.

Es gibt jedoch einen Hoffnungsschimmer: Auch wenn es nicht die beste Praxis ist, haben Sie die Möglichkeit, Ihre Daten und APEX-Anwendungen zu sichern und sie in Ihrer zweiten FREE-Tier-ADB-Umgebung durch Klonen wiederherzustellen. Sie können einfach die von OCI bereitgestellte REST-API verwenden und diesen täglichen Prozess automatisieren. Eine andere Möglichkeit besteht darin, ein eigenes Backup zu erstellen, indem Sie jede Nacht alle Daten exportieren. Dazu können Sie Data-Pump-Schemaexporte für Daten und DDLs in das DB-Dateisystem planen und dann die Data-Pump-Dateien in einen OCI Object Storage übertragen. Sie können Ihre APEX-Anwendung(en) sichern, indem Sie die API „apex_export.get_application" für jede Anwendung verwenden und sie ebenfalls in den Object Storage kopieren. Mit diesen Voraussetzungen kann eine neue Instanz in Kürze provisioniert werden.

Fazit

Zusammenfassend lässt sich sagen, dass autonome Datenbanken in der Oracle Cloud Infrastructure APEX-Entwicklern großartige Möglichkeiten zur Erstellung sicherer Low-Code-Anwendungen bieten. Es gibt zwar einige Einschränkungen mit einer kostenlosen Instanz, wie zum Beispiel das Verwenden von benutzerdefinierten URLs und die Notwendigkeit eines Backup-Plans, aber es gibt auch Umgehungsmöglichkeiten. Durch den Einsatz von Zugriffskontrolllisten, SSO und der Web-Application-Firewall können Entwickler eine sichere Cloud-Umgebung einrichten. Insgesamt können Sie das Potenzial maximieren, indem Sie die Infrastruktur- und Sicherheitsanforderungen sorgfältig berücksichtigen und die verfügbaren Ressourcen nutzen.

Über den Autor

Timo Herwix ist seit 2019 als Senior Consultant bei der MT in Ratingen tätig. In seinem Tagesgeschäft arbeitet er mit Oracle-Datenbanken und der Entwicklung von Oracle-APEX-Anwendungen. Vor seinem Einstieg bei der MT arbeitete Timo Herwix als Data-Warehouse-Spezialist, wo er Erfahrungen als Datenbankadministrator, Performance Tuning und bei der SQL-Entwicklung sammelte. Neben Datenbankmodellierung, PL/SQL-Entwicklung und dem Entwickeln von APEX-Anwendungen, interessiert er sich für Webentwicklung und deren Architektur. 2023 hielt Timo Herwix auf der „APEX World" einen Vortrag zum Thema „Oracle Autonomous Databases for APEX Developer".

Timo Herwix
Timo.herwix@mt-itsolutions.com

Mut zur Veränderung: Change Data Capture (CDC) von PostgreSQL über NiFi in die Oracle Cloud (OCI)

Janis Ax, Hatice Sen; Ordix

Immer mehr Unternehmen setzen auf die Cloud, um ihre Daten effizienter zu speichern und zu verwalten. Doch wie können Sie Ihre bestehenden Datenbanken nahtlos in die Cloud integrieren? Eine vielversprechende Lösung ist Change Data Capture (CDC), ein Verfahren, das Änderungen in Echtzeit erfasst. Bei PostgreSQL-Datenbanken wird hierzu die Funktion Logical Replication genutzt. Das ermöglicht es, Änderungen an ausgewählten Tabellen oder Schemata in Echtzeit auf andere Systeme, wie Apache NiFi weiterzuleiten. Sie erfahren in diesem Artikel, wie Sie CDC von PostgreSQL über Apache NiFi in die Oracle Cloud (OCI) integrieren können und somit Datenänderungen direkt in die Cloud streamen. Schritt für Schritt zeigen wir Ihnen, wie die einzelnen Komponenten funktionieren und zeigen den gesamten Prozess auf, damit Sie Ihre Daten schnell und sicher in die Cloud übertragen können.

Sollen Daten in die Cloud gelagert werden, handelt es sich **nicht** immer um eine ganze Datenbankmigration. Manchmal sollen nur für einzelne Use-Cases bestimmte Daten in der Cloud zur Verfügung stehen, um darauf beispielsweise Analysen durchzuführen. Außerdem eignet sich die Cloud hervorragend als Archiv, da Cloud-Speicher, wie beispielsweise S3-Buckets, sehr günstig sind.

So oder so: Wenn Daten in der Cloud benötigt werden, stellen sich folgende Fragen: Wie kommen die Daten dorthin und viel wichtiger, wenn die Daten dort sind, wie bleiben sie aktuell? Wie können Änderungen an On-Premises-Daten in die Cloud gespiegelt werden?

Für die erstmalige Migration von On-Premises in die Cloud gibt es einige Tools, wie den Database Migration Service *(siehe: https://cloud.google.com/database-migration?hl=de)* von Google. Mit solchen Tools lassen sich einmalig ganze Datenbank-Management-Systeme in die äquivalenten Cloud-Komponenten laden. Hierbei ist das Ziel klar: Die Migration der Systeme in ihrer Gesamtheit in die jeweilige Cloud.

Ziel dieses Artikels ist es allerdings, die Änderungen an den Daten zu protokollieren und, wenn nötig, in Cloud-Komponenten oder andere Systeme einzuspeisen.

Wir möchten dazu Apache NiFi vorstellen und aufzeigen, wie NiFi die Datenänderungen mitbekommt und mit einfachen **No-Code** Dataflows diese an die verschiedensten Ziele schicken und vor allem aktuell halten kann.

Dafür nutzen wir die Logical-Replication-Funktion, die PostgreSQL bietet. Normalerweise wird diese Technik genutzt, um zum Beispiel ein hochverfügbares PostgreSQL-Cluster aufzubauen. Jegliche Änderung an den Daten, also Insert-, Update- und Delete-Operationen werden dabei protokolliert und, in diesem Beispiel, Apache NiFi, zur Verfügung gestellt.

Sind die Daten in NiFi können wir sehr einfach Dataflows erstellen, die die Daten dann je nach Bedarf routen, um beispielsweise Transformationen vorzunehmen. Schlussendlich ist unser Ziel aber die Cloud und wir werden die jeweiligen Änderungen durch einen einfachen Dataflow in near-real-time-Geschwindigkeit in verschiedene Cloud-Speichersysteme, wie Amazon S3-Buckets oder Google Cloud Storage, laden und zusätzlich in eine Cloud-Datenbank.

Die Basics

Apache NiFi und was die NSA damit zu tun hat

Apache NiFi ist ein Open-Source-Tool für Datenintegration, das von der National Security Agency (NSA) entwickelt wurde. Das Projekt wurde 2014 an die Apache Software Foundation übergeben und wird seitdem als Toplevel-Projekt geführt und weiterentwickelt.

Apache NiFi ist ein Web-UI-basiertes Tool, das eine einfache und skalierbare Möglichkeit zur Integration von Daten verschiedener Typen und Quellen bietet. Es ermöglicht Benutzern, Daten in Echtzeit zu erfassen, zu übertragen und zu transformieren.

Ein besonderes Merkmal von Apache NiFi im Vergleich zu anderen ETL-Tools ist die Fähigkeit zur Integration einer Vielzahl von Datentypen, einschließlich strukturierter, semistrukturierter und unstrukturierter Daten. Es unterstützt auch die Integration von Daten aus verschiedenen Quellen wie Datenbanken, Dateisystemen und IoT-Geräten.

Die Funktionsweise von Apache NiFi basiert auf dem Konzept von Datenflüssen. Diese bestehen aus einer Reihe von Prozessoren, die Daten verarbeiten, und Controller Services, die Einstellungen und Informationen bereitstellen, die von den Prozessoren verwendet werden. Die Erstellung eines Datenflusses erfolgt durch das Drag-and-Drop von Prozessoren auf eine Arbeitsfläche und das Verbinden von Prozessoren durch Connections. Jeder Prozessor kann konfiguriert werden, um bestimmte Aufgaben an den Daten auszuführen. Dabei ist eine parallele Verarbeitung von Daten durch die Verteilung von Datenflüssen auf mehrere Knoten in einem Cluster möglich.

NiFi kann man sich wie eine magische Röhre vorstellen, durch die alle möglichen Arten von Daten fließen können. Dabei hilft NiFi Daten von einem Ort zum anderen zu bewegen, sie zu organisieren und zu verarbeiten.

Von Datenflüssen & Prozessoren: Die Grundlagen von Apache NiFi

Allgemeiner Aufbau und Definitionen

NiFi verwendet einen Flow-basierten Ansatz für die Datenintegration. Ein Flow besteht aus einer Sammlung von Prozessoren, die jeweils eine spezifische Aufgabe ausführen. Es gibt Prozessoren, die Daten aus verschiedenen Quellen einlesen können, andere transformieren Daten und wieder andere schreiben Daten in verschiedene Zielsysteme. Werden diese einfachen Bausteine verkettet, lassen sich komplexe und mächtige Dataflows realisieren.

Wie werden Flows gebaut? Es gibt eine intuitive Web-UI, mit der Flows, oder genauer Dataflows, per Drag-and-Drop-Komponenten erstellt werden.

Prozessoren

Prozessoren sind die Hauptkomponenten in NiFi. Sie führen die eigentliche Datenverarbeitung durch, das heißt, sie sind die eigentlichen Arbeiter. Es gibt eine Vielzahl von Prozessoren, die verschiedene Funktionen ausführen können, wie zum Beispiel das Lesen von Daten aus einer Datenbank, das Konvertieren von Daten in verschiedene Formate oder das Schreiben von Daten in ein Ziel. Hierbei ist wichtig zu erwähnen, dass ein Prozessor immer nur eine bestimmte Aufgabe zu erledigen hat.

Controller Services

Controller Services sind Komponenten, die von Prozessoren verwendet werden, um auf externe Ressourcen wie Datenbanken oder Cloud-Dienste zuzugreifen. Sie laufen als Deamons im Hintergrund und stellen Dienstleistungen, wie Datenbanken und Cloud-Zugänge bereit. Sie werden einmalig konfiguriert und können von beliebig vielen Prozessoren genutzt werden.

Connections

Damit sich die Daten durch den Flow bewegen können, gibt es Connections.

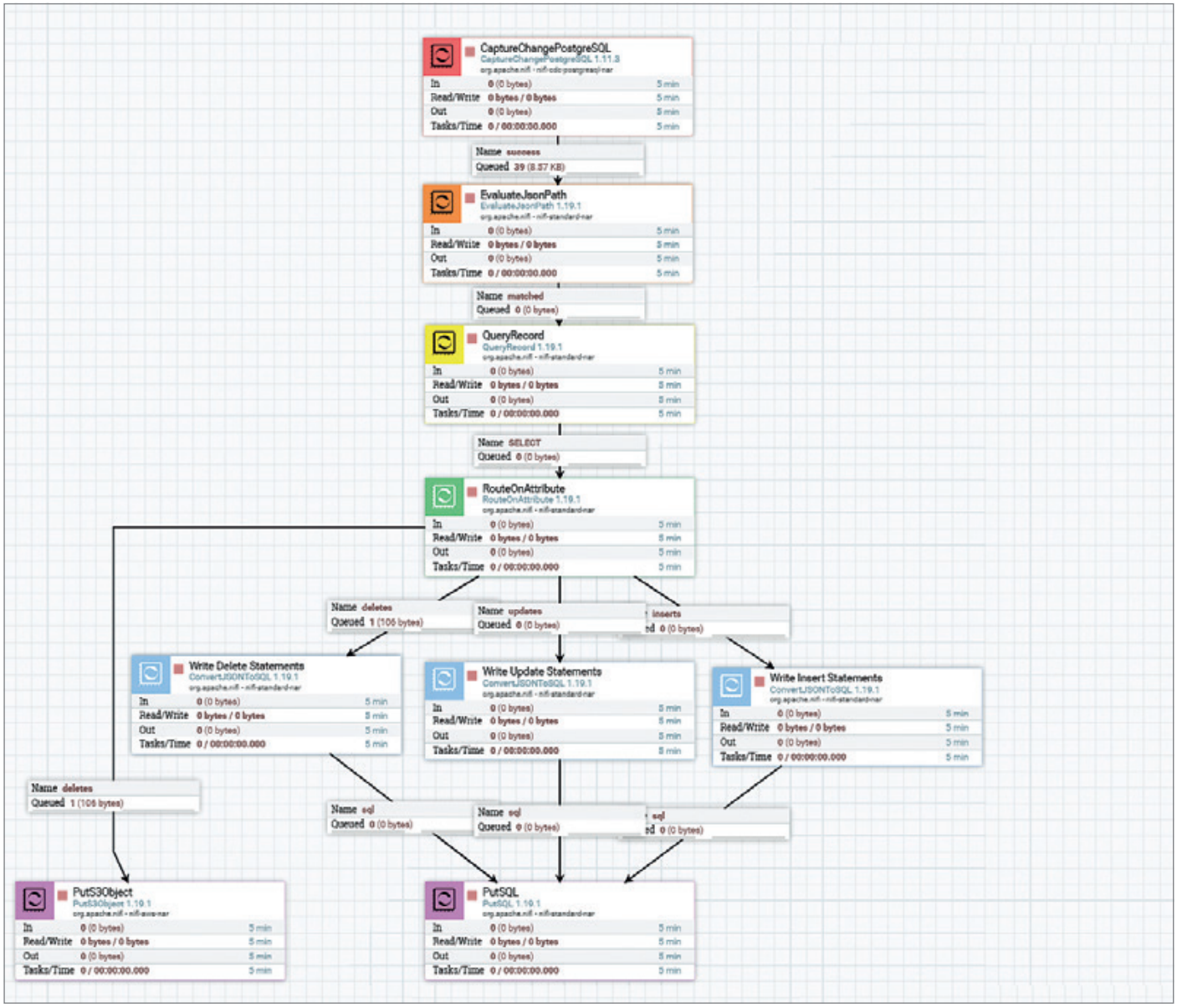


Abbildung 1: Aufbau unseres Flows (© [ORDIX AG, Flow aus Apache NiFi])

Mit ihnen werden Verbindungen zwischen Prozessoren hergestellt. Auf den Connections bewegen sich sogenannte FlowFiles, so werden Datenobjekte in NiFi bezeichnet, von einem in den nächsten Prozessor.

PostgreSQL und die Logical Replication

In diesem Kapitel erklären wir kurz die Verwendung der logischen Replikation in PostgreSQL. Dies wird benötigt, damit 3rd-Partytools wie Apache NiFi und dessen Change-Data-Capture-Prozessor die Datenänderungen in Echtzeit mitbekommen.

Die logische Replikation unterscheidet sich von der physischen Replikation und Streaming-Replikation in verschiedenen Aspekten. Während physische Replikation auf Blockebene arbeitet und eine vollständige Kopie der Datenbank erfordert, repliziert die logische Replikation nur spezifische Datenänderungen. Im Gegensatz zur Streaming-Replikation, die Daten im binären Format überträgt, werden Daten bei der logischen Replikation im logischen Format übermittelt. Dies ermöglicht selektive Replikation und granulare Kontrolle, wodurch Benutzer effizienter arbeiten und auf spezifische Anforderungen reagieren können.

Einige der Vorteile der logischen Replikation sind:

- **Selektive Replikation:** Benutzer können auswählen, welche Tabellen, Spalten oder Schemata repliziert werden sollen. Dies trägt dazu bei, den Speicherplatzbedarf zu reduzieren und die Leistung zu verbessern.
- **Effizienz:** Da nur die Änderungen an den Daten repliziert werden, die aufgezeichnet werden sollen, ist die logische Replikation viel effizienter als die physische Replikation.
- **Flexibilität:** Die logische Replikation bietet granulare Kontrolle darüber, welche Daten repliziert werden sollen. Dies ermöglicht es Benutzern, unterschiedliche Replikationsszenarien zu implementieren und auf spezifische Anforderungen zu reagieren.

Durch die Verwendung von Replikations-Slots, Publikationen und Abonne-

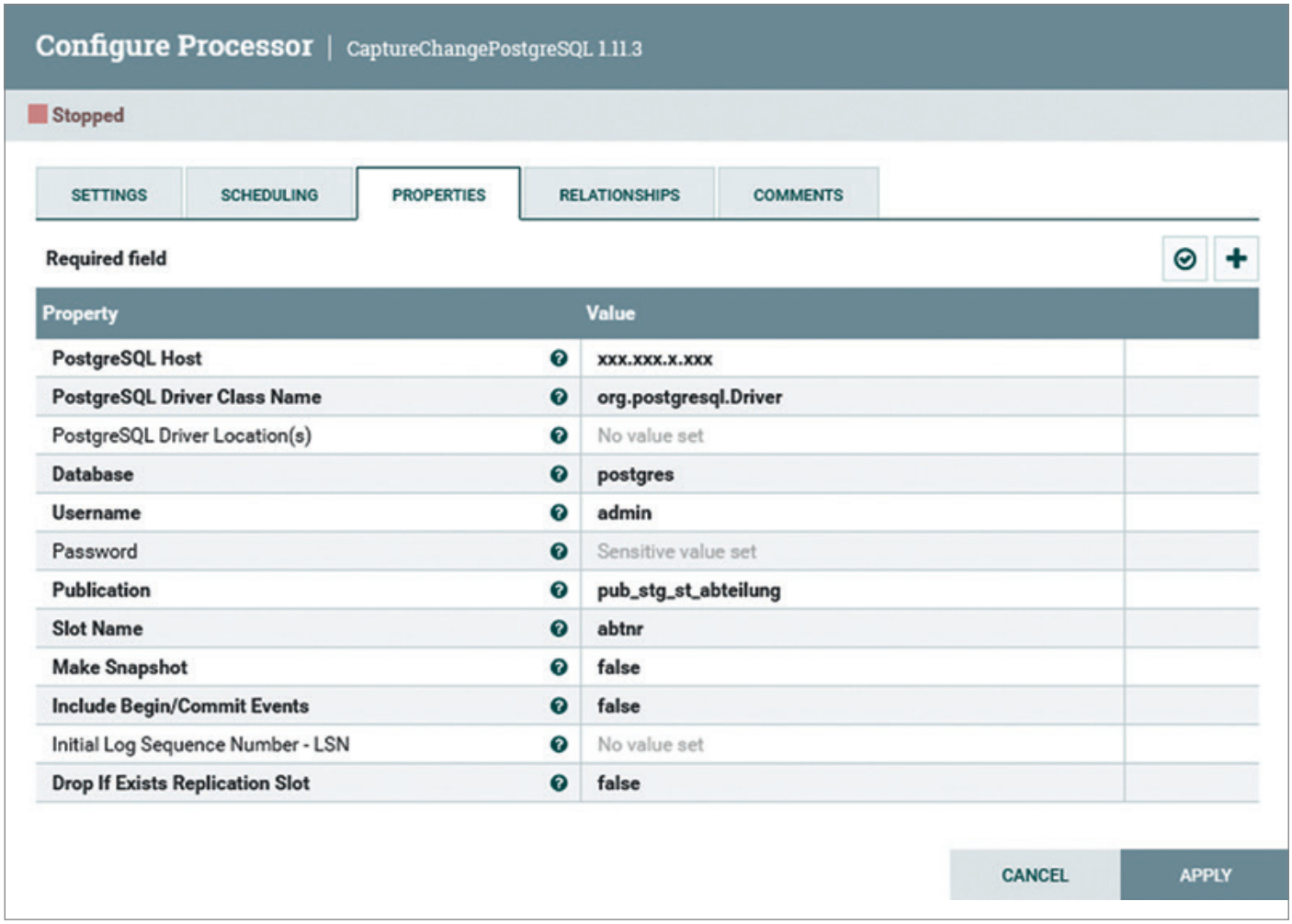


Abbildung 2: Konfiguration des CaptureChangePostgreSQL-Prozessors (© [ORDIX AG, Flow aus Apache NiFi])

ments in PostgreSQL kann die logische Replikation Datenänderungen sicher und zuverlässig an Apache NiFi senden. Replikation Slots sind ein zentraler Mechanismus in PostgreSQLs Logical Replication. Sie stellen sicher, dass der Datenbank-Server niemals mehr Änderungsdaten sendet, als der Replikationsempfänger verarbeiten kann. Ein Replikationsslot ist eine Art Puffer, in dem der Datenbankserver Änderungsdaten speichert, bis sie vom Replikationsempfänger abgerufen werden. Durch die Verwendung von Replikationsslots wird sichergestellt, dass die Datenreplikation nicht aufgrund von Netzwerkproblemen oder anderen Engpässen ins Stocken gerät.

Subskriptionen sind das Gegenstück zu Replikationsslots. Eine Subskription ist eine Verbindung von einem Replikationsempfänger (Subscriber) zu einem Replikationsquellserver (Publisher), die Daten in Echtzeit zur Verfügung stellt. Sie kann auf eine einzelne Datenbank, eine Gruppe von Tabellen oder auf die gesamte Datenbank ausgerichtet sein.

Die Verbindung zwischen Subscriber und Publisher wird über Abonnements hergestellt. Ein Abonnement definiert, welche Tabellen oder Schemata eines Publishers von einem Subscriber repliziert werden sollen. Dabei wird jedem Abonnement ein Replikationsslot zugeordnet, in dem die vom Publisher gesendeten Änderungen gespeichert werden.

Ein weiterer Vorteil von Subskriptionen besteht darin, dass der Subscriber selbst entscheiden kann, welche Daten er replizieren möchte. So können beispielsweise nur bestimmte Spalten einer Tabelle oder nur Änderungen, die ein bestimmtes Filterkriterium erfüllen, repliziert werden.

Um eine Subskription zu erstellen, muss der Subscriber die benötigten Informationen wie den Namen des Publishers, den Namen des Abonnements und die Tabellen oder Schemata angeben, die repliziert werden sollen. Sobald die Subskription erstellt wurde, beginnt der Publisher, Änderungen an den Subscriber zu senden. Der Subscriber empfängt diese Änderungen und wendet sie auf seine eigene Datenbank an.

Dies ermöglicht es Benutzern, Daten aus verschiedenen Quellen zu sammeln und zu analysieren und sie in Echtzeit an andere Systeme weiterzugeben, was für Anwendungen wie Echtzeitanalysen oder Streaming von Vorteil ist.

Von der Community zur Integration: Der CDC-PostgreSQL-Prozessor

Da es sich bei Apache NiFi um eine Open-Source-Software handelt, arbeiten viele verschiedene Personen daran, NiFi zu erweitern und zu verbessern. So war auch der CaptureChangePostgreSQL-Prozessor ein Projekt der Community, das größtenteils von Gerdan Rezende dos Santos (GitHub: gerdansantos – *siehe https://github.com/gerdansantos*) und Davy Alva-

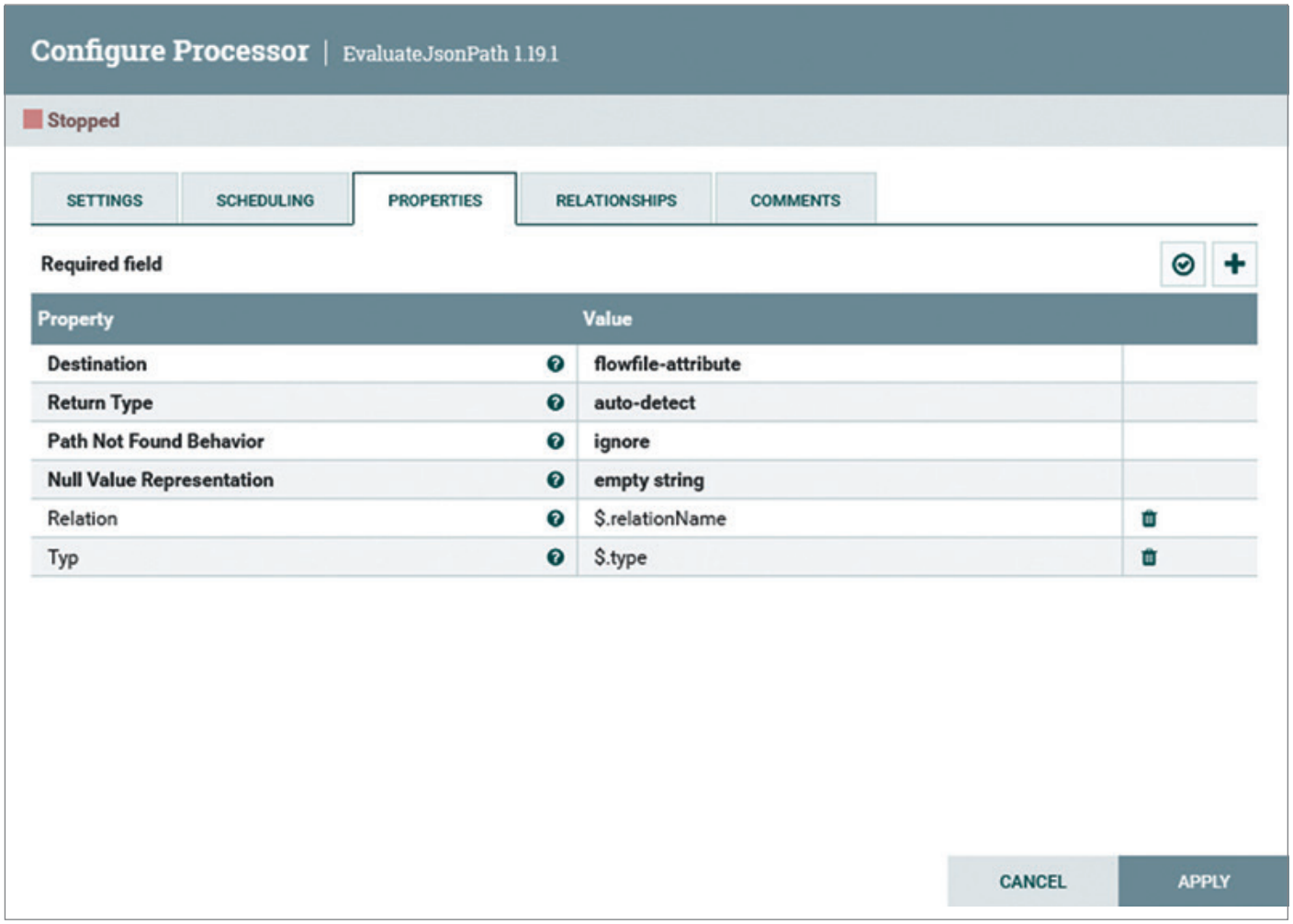


Abbildung 3: Konfiguration des EvaluateJsonPath-Prozessors (© [ORDIX AG, Flow aus Apache NiFi])

renga Machado (GitHub: davyam –*siehe https://github.com/davyam*) geführt wurde. *[1]* Mit diesem Prozessor können Änderungen an der PostgreSQL-Datenbank detektiert und an NiFi gestreamt werden.

Eine der Stärken von NiFi ist die Erweiterbarkeit, das heißt, wir können unsere eigenen Prozessoren in NiFi anbinden. Wie genau das geht, erklärt Janis Ax detailliert in einem Blogartikel „Geht nicht? Gibt's nicht? Mein eigener Apache-NiFi.Prozessor" im ORDIX Blog. *[2]*

Obwohl Apache NiFi eine Vielzahl von Prozessoren bereitstellt, kann es manchmal erforderlich sein, einen eigenen Prozessor zu entwickeln und zu integrieren, um spezifische Anforderungen anzupassen, komplexe Workflows umzusetzen und eine nahtlose Integration mit externen Systemen zu gewährleisten.

Der CaptureChangePostreSQL-Prozessor wurde von der Community entwickelt und ist derzeit nicht Teil des offiziellen NiFi-Prozessorkatalogs. Trotzdem können wir ihn ganz einfach zu unserem eigenen Prozessorkatalog hinzufügen und nutzen. Hierfür müssen wir die **nar**-Datei des erstellten Prozessors in einem Ordner ablegen und auf diese in der **nifi.properties**-Datei verweisen. Dadurch haben wir die Möglichkeit, unsere NiFi-Instanz um beliebig viele Prozessoren zu erweitern. Mehr dazu findet sich auch im Ordix Blog *(siehe https://blog.ordix.de/)*.

In NiFi existierte bereits der CaptureChangeMySQL-Prozessor, der, wie der Name bereits erraten lässt, CaptureDataChange auf einer MySQL-Datenbank ausführt. Der CaptureChangePostgreSQL-Prozessor wurde zu einer beliebten Anfrage, sodass sich die Community selbst an die Arbeit gemacht hat, um diesen zu erstellen. Damit ein Prozessor zum offiziellen Katalog von NiFi aufgenommen wird, wird dieser einigen Tests unterzogen und es muss eine gute Dokumentation zur Nutzung vorgelegt werden. Das Hinzufügen des CaptureChangePostgreSQL-Prozessors ist seit einigen Jahren ein offenes Projekt und soll, wenn alles gut läuft, im nächsten Versionsrelease mitenthalten sein.

Doch wie bereits erklärt, müssen wir den Release nicht abwarten, um den Prozessor nutzen zu können. Im nächsten Kapitel besprechen wir einen Use-Case, in dem wir den Prozessor bereits jetzt benötigt haben.

Der Datenfluss

In unserem aktuellen Szenario arbeiten wir mit einer produktiven PostgreSQL-Datenbank, von der wir fortlaufend Abfragen für Analyse- und Reporting-Zwecke durchführen. Um die Leistung der Datenbank nicht zu beeinträchtigen, haben wir uns dazu entschieden, die Daten in eine Cloud-Umgebung zu streamen, in der wir über erweiterte Ressourcen verfügen, um umfangreiche Abfragen auszuführen und die Ergebnisse in entsprechenden Dashboards zu präsentieren. Mithilfe des

```
{
  "tupleData" : {
    "process_id" : "48791dfsa4gev91",
    "abtnr" : "1",
    "bereich_nr" : "5",
    "abtname" : "Marketing",
    "abtleiter" : "38",
    "its" : "2023-05-08"
  },
  "relationName" : "staging.st_abteilung",
  "type" : "delete"
}
```

Listing 1: Ein FlowFile aus dem CaptureChangePostgreSQL-Prozessor: So sieht der Aufbau des FlowFiles aus, den wir aus dem CaptureChangePostgreSQL erhalten. Am Anfang des FlowFiles steht der Datensatz und am Ende die Relation, an der die Änderung vorgenommen wurde. Zusätzlich erhalten wir die Art der Änderung.

CaptureChangePostgreSQL-Prozessors erhalten wir die Datensätze als FlowFiles unmittelbar nach erfolgten Änderungen in der Datenbank. Nun fehlt uns lediglich der restliche Datenfluss, um die Daten effizient in die Cloud zu streamen – dazu im Folgenden mehr.

Das große Ganze

Unser Ziel ist es, Datenänderungen, die in unserer Datenbank vorgenommen werden, in eine Cloud-Umgebung zu übertragen. Zu diesem Zweck haben wir einen Dataflow entwickelt, der in *Abbildung 1* dargestellt ist. Grundsätzlich werden die Daten in Storage Buckets sowie in einer Datenbank in der Cloud gespeichert.

Was passiert hier genau?

Registriert der CaptureChangePostgreSQL-Prozessor (rot) eine Datenänderung, wird in NiFi ein korrespondierendes FlowFile erstellt. Im FlowFile finden sich einige Details als JSON. Dazu gehören die Datenänderungen, aber auch Meta-Informationen, wie der Tabellenname und der Typ der Änderung.

Mit diesen Informationen können wir die Daten filtern (grün). So ist es möglich nach Tabellen zu filtern, oder nach Statement-Typen wie **„delete“** oder **„insert“**.

Letzteres machen wir, um die Daten zu filtern und alle **delete**-Statements in einem Storage Bucket zu archivieren (violett, links).

Zusätzlich lassen sich mit dem QueryRecord-Prozessor (gelb) SQL-Statements auf strukturierten Daten, wie JSON, CSV oder Avro ausführen. So können wir unsere FlowFiles umstrukturieren oder sogar SQL-Berechnungen auf diesen ausführen.

Mit dem ConvertJSONtoSQL-Prozessor (blau) lässt sich die vorher bearbeitete JSON-Datei so transformieren, dass darauf ein neues SQL-Statement entsteht, das wir in der PostgreSQL-Datenbank oder jeder anderen Datenbank in der Cloud ausführen können. Dies ist nicht auf Cloud-Datenbanken beschränkt.

Die Ausführung geschieht dann im PutSQL-Prozessor (violett, rechts). Durch diesen Prozessor werden die Statements in der Cloud ausgeführt. Die Datenbanken bleiben so synchron.

Von der Datenänderung zur Cloud-Speicherung: Schritt für Schritt durch den Dataflow

Tauchen wir nun tiefer in den Dataflow ein und werfen einen detaillierten Blick auf die Schritte, die es ermöglichen, Datenänderungen in die Cloud zu übertragen. Wir werden Schritt für Schritt durch den Ablauf navigieren und die Datenintegration erleben, die diesen Dataflow auszeichnet.

CaptureChangePostgreSQL-Prozessor: Erfassung von Datenänderungen in Echtzeit

Um den CaptureChangePostgreSQL-Prozessor *(siehe Abbildung 2)* in Apache NiFi zu verwenden, sind folgende Konfigurationswerte erforderlich:

- PostgreSQL Host: Die Adresse des PostgreSQL-Servers.
- PostgreSQL Driver Class Name: Der Name der Treiberklasse für die PostgreSQL-Datenbank.
- Database: Der Name der zu erfassenden Datenbank.
- Username: Der Benutzername für den Datenbankzugriff.
- Publication: Der Name der Logical-Replication-Publikation.
- Slot Name: Der Name des Replikation-Slots.

Da der CaptureChangePostgreSQL-Prozessor die Funktion der Logical Replication nutzt, müssen wir sowohl die Publikation als auch den Slot-Namen angeben. Der Slot-Name fungiert als Identifikator für den Replikations-Slot.

Nachdem wir alle erforderlichen Eigenschaften konfiguriert haben, können wir den Prozessor starten. Dadurch erhalten wir die Datenänderungen in Echtzeit als FlowFiles in Apache NiFi.

Die Daten, die wir aus dem Prozessor erhalten, haben folgendes Format *(siehe Listing 1)*.

Diese FlowFiles enthalten den Datensatz, an dem die Änderung stattgefunden hat, den Tabellennamen sowie den Typ der Änderung (z. B. **"delete"**).

Um diese FlowFiles weiter zu verarbeiten, verwenden wir den EvaluateJsonPath-Prozessor *(siehe Abbildung 3)*. Dieser ermöglicht es uns, Werte aus der JSON-Datei als Attribute eines FlowFiles zu speichern. Dieses Vorgehen ist ein gängiges Pattern in NiFi, da die Arbeit mit FlowFiles deutlich einfacher ist als die Arbeit mit dem Content.

Wir konfigurieren den Prozessor so, dass er die JSON-Werte als FlowFile-Attribute speichert. Der Prozessor liest hier den Tabellennamen und den Statement-Typ aus der JSON-Datei *(siehe Abbildung 3)*.

Wir möchten alle Delete-Statements archivieren und zusätzlich alle Statements – also Insert-, Update- und Delete-Statements – in die Cloud-Datenbank streamen.

Dazu müssen wir die FlowFiles (Daten) filtern, beziehungsweise routen. Das können wir über die zuvor neu erstellten Attribute tun.

```
{
     "process_id" : "48791dfsa4gev91",
    "abtnr" : "1",
    "bereich_nr" : "5",
    "abtname" : "Marketing",
    "abtleiter" : "38",
    "its" : "2023-05-08"
  }
```

Listing 2: Ein FlowFile aus dem QueryRecord-Prozessor: Die Datenstruktur des FlowFiles wurde so umgeformt, dass er nun dem ConvertJSONtoSQL-Prozessor übergeben werden kann. Mit dem neuen FlowFile kann der Prozessor nun SQL-Insert-, Update- oder Delete-Statements schreiben.

Die benötigten Datensätze befinden sich unter „tupleData“. Da kommt der Prozessor QueryRecord ins Spiel. Mit diesem können wir SELECT-Statements auf unseren FlowFiles definieren und sogar gängige SQL-Berechnungen ausführen.

Hier nutzen wir die SQL-Abfragen, um unsere FlowFiles in das benötigte Format für den nächsten Prozessor zu bringen *(siehe Abbildung 4)*.

Der QueryRecord-Prozessor kann Statements auf beliebigen Dateiformaten ausführen und diese in beliebigen Formaten ausgeben. Mit Hilfe des QueryRecord-Prozessors ändern wir die Struktur unserer JSON-Datei, sodass wir auf alle relevanten Werte der Datenänderung zugreifen *(siehe Abbildung 4)*.

Unser FlowFile hat nun folgende Struktur *(siehe Listing 2)*.

Wir haben unser JSON-Dokument nun so transformiert, dass es nur noch aus den „tupleData“ besteht und nicht wie bisher noch Meta-Informationen, wie den Tabellennamen, beinhaltet.

Dabei sind die anderen Metadaten, „relationName“ und „type“, nicht verloren gegangen, weil wir sie zuvor mit Hilfe des EvaluateJsonPath-Prozessors als Attribute mitgegeben hatten. Sie finden sich nur nicht mehr im Content des FlowFiles wieder.

Als nächstes können wir anhand dieser Attribute unsere FlowFiles filtern und routen.

Dafür nutzen wir den RouteOnAttribute-Prozessor.

Unsere Delete-Statements legen wir zunächst in ein S3-Bucket und leiten diese dann, zusammen mit unseren Inserts und Updates, weiter an den ConvertJSONtoSQL-Prozessor. Dieser kann JSON-Dateien in SQL-Statements konvertieren. Der ConvertJSONtoSQL-Prozessor kann die Metadaten von den zu befüllenden Tabellen abgreifen und die Statements entsprechend verfassen.

Schlussendlich erzeugt dieser Prozessor anhand der Metadaten der Zieltabelle und anhand der Daten, die als JSON vorliegen, ein für den Datensatz spezifisches SQL-Statement. Dieser Datensatz wird dann einfach durch den PutSQL-Prozessor ausgeführt.

Somit haben wir erfolgreich die Änderungen einer Datenbank getrackt, diese Daten transformiert, geroutet und in unterschiedlichen Zielsystemen abgelegt. Hier ist wichtig zu erwähnen, dass wir nur einen kleinen Einblick in die Möglichkeiten von Apache NiFi zeigen konnten. Wir könnten auch Daten aus anderen Quellen einbinden und in beliebig vielen Zielsystemen ablegen.

Vom Rinnsal zum Strom: Ein abschließender Blick auf unseren Dataflow

In diesem Artikel haben Sie einen kurzen, aber detaillierten Einblick in die Verwendung von Apache NiFi bekommen. Wir haben gezeigt, wie Sie Datenänderungen in Echtzeit erfassen, transformieren und in verschiedene Zielorte übertragen können. Dabei haben wir einige Key-Features kennengelernt, die NiFi zu einer leistungsstarken und flexiblen Plattform machen.

Ein wichtiges Feature, das uns bei der Datenverarbeitung geholfen hat, ist der QueryRecord-Prozessor. Mit diesem können wir SQL-Statements gegen strukturierte Daten wie JSON ausführen. Dadurch erhalten wir die Möglichkeit, die Daten gezielt zu filtern, zu transformieren und Manipulationen vor der Weiterverarbeitung durchzuführen.

Ein weiteres leistungsstarkes Feature ist der CovertJSONtoSQL-Prozessor. Dieser generiert dynamische SQL-Statements basierend auf den Metadaten der Zieltabelle und den Informationen in der JSON-Datei. Dadurch können wir die Daten nahtlos in eine Cloud-Datenbank übertragen und dort effizient verarbeiten.

Der Artikel zeigt uns auch die allgemeine Power von NiFi und Open-Source-Software. Mit nur sieben speziellen Prozessoren haben wir einen beeindruckenden Datenfluss entwickelt, der nicht nur Daten archiviert, sondern sie auch in eine Cloud-Datenbank streamt. Dies verdeutlicht die Flexibilität und Anpassungsfähigkeit von NiFi, um komplexe Datenintegrations- und Verarbeitungsworkflows zu realisieren.

Insgesamt bietet Apache NiFi eine Vielzahl von Prozessoren und Funktionen, die es ermöglichen, Datenänderungen effizient zu erfassen, zu transformieren und in verschiedenen Zielsystemen zu nutzen. Mit seiner benutzerfreundlichen Web-UI und der Unterstützung von Open-Source-Software ist NiFi eine wertvolle Plattform für Datenintegration und -verarbeitung in einer Vielzahl von Anwendungsfällen. Mit den vorgestellten Features haben wir einen Einblick in das Potenzial von NiFi erhalten und können nun diese leistungsstarke Plattform nutzen, um unsere eigenen datengetriebenen Projekte voranzutreiben.

Quellen

[1] Gerdan Santos und Davy Machado (2020): NIFI-4329 – Adding CaptureChangePostgreSQL processor to capture data changes (INSERT/UPDATE/DELETE) in PostgreSQL tables via Logical Replication., github.com, *www.github.com/apache/nifi/pull/4065*

[2] Janis Ax (2023): Geht nicht? Gibt's nicht? Mein eigener Apache-NiFi-Prozessor: *https://blog.ordix.de/mein-eigener-apache-nifi-prozessor*

[3] ORDIX Blog: *https://blog.ordix.de/*

Über die Autoren

Hatice Sen ist seit dem 01.01.2023 bei der Ordix AG als Junior IT-Consultant tätig. Zuvor hat sie Mathematik an der Johannes-Gutenberg-Universität Mainz studiert. Bei der Ordix AG arbeitet sie im DWH- und

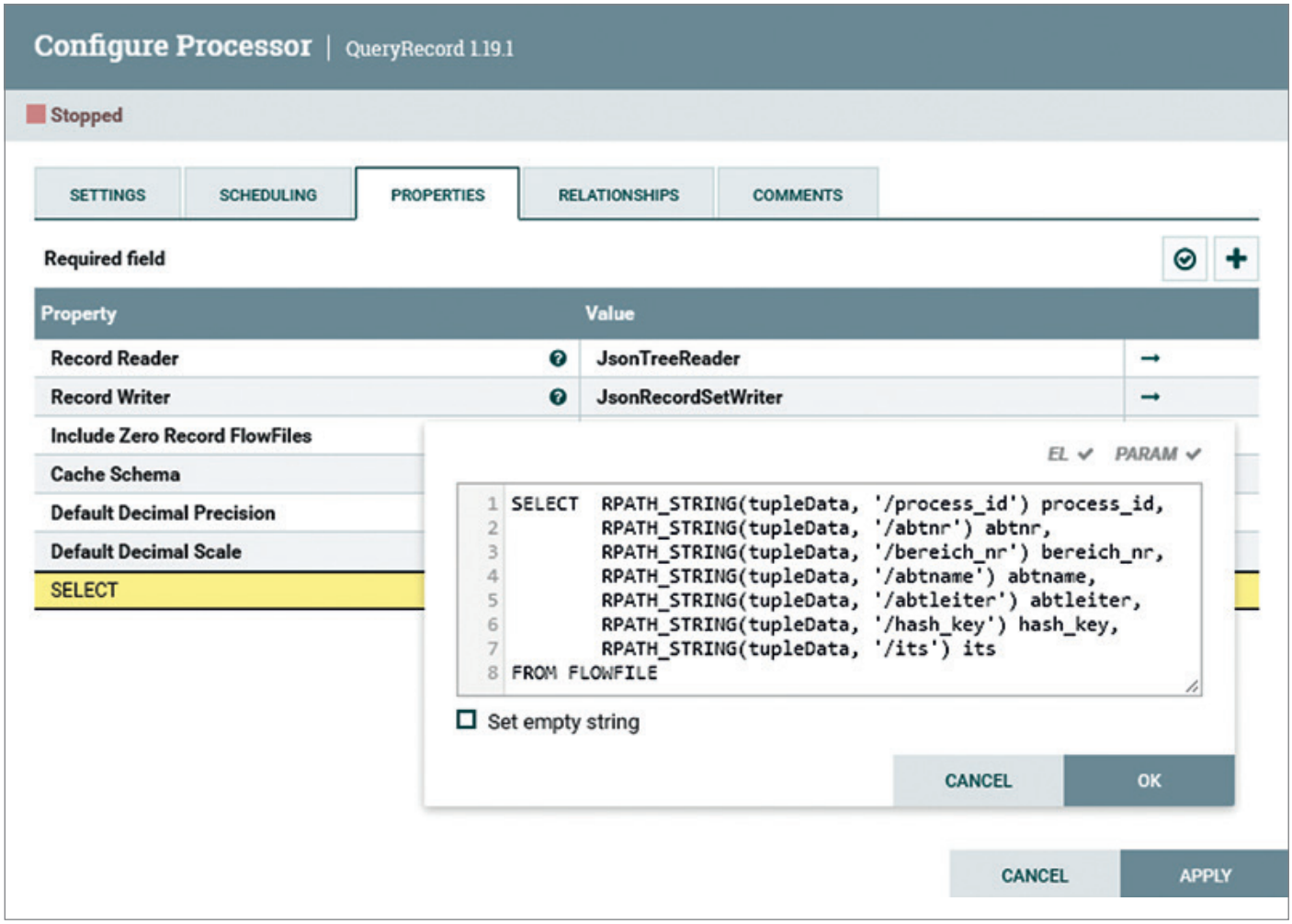
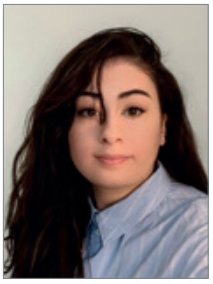


Abbildung 4: Konfiguration des QueryRecord-Prozessors (© [ORDIX AG, Flow aus Apache NiFi])

ETL-Team und spezialisiert sich derzeit in Apache NiFi.

Janis Ax ist seit 2017 bei der Ordix AG und hat dort im Rahmen eines dualen Studiums erfolgreich seinen Bachelor absolviert. Seit 2020 ist Janis als Consultant für die Ordix AG tätig und dort als Berater im Big Data- sowie Cloud-Umfeld unterwegs. Er berät dort die Kunden primär in Big-Data-Dataflow-Komponenten und gibt sein Wissen regelmäßig bei Seminaren und auf Konferenzen weiter.

Er ist als Autor für das Fachmagazin Informatik Aktuell tätig und war bereits Sprecher der IT-Tage 2021 und 2022.



Hatice Sen
hts@ordix.de

Janis Ax
jax@ordix.de

Vulnerability Management für Datenbank-Plattformen

Daniel Steiger, Accenture

Das Ziel von Vulnerability Management ist es, Sicherheitsschwachstellen in der IT-Infrastruktur und in der Systemlandschaft – sowohl von On-Premises als auch cloudbasierten Lösungen – frühestmöglich zu erkennen und angemessen darauf zu reagieren. In diesem Artikel beschreibe ich die wichtigsten Elemente für ein wirksames Vulnerability Management im Kontext einer Datenbankplattform, zeige eine bewährte Vorgehensweise auf und – last but not least – teile ich die Erfahrung eines Product Owners, der sich unverhofft in der Rolle des Security Champions seines DevOps-Teams wiederfindet.

Einleitung

Die zentrale Frage beim Vulnerability Management lautet: „Was kann in meiner Systemumgebung schiefgehen? "

- Wo liegen die Schwachstellen in meiner Systemarchitektur?
- Woher weiß ich, dass jemand Daten „in-transit", „in-use" oder „at-rest" nicht ändern kann?
- Was passiert, wenn jemand die Daten in der Datenbank ändert?
- Könnte ein Angreifer Befehle mit erweiterten Rechten ausführen?
- Ist es in Ordnung, wenn Informationen unverschlüsselt von einer Box zur nächsten wandern?
- Und vieles mehr

Mit Hilfe der Threat-Modelling-Methode untersuchen wir die Systemumgebung im Hinblick auf potenzielle Schwachstellen und mögliche Angriffsvektoren. Die so identifizierten Schwachstellen werden anschließend auf ihre Kritikalität geprüft, bewertet und entsprechend ihrer Priorität behoben.

In der Regel handelt es sich bei den untersuchten Systemen um komplexe Umgebungen. Das heißt, Dinge ändern sich ständig und deshalb ist die Zeit für die Identifikation und die Behebung von Sicherheits-Schwachstellen ein entscheidender Faktor. Je kürzer das „Window of Opportunity" gehalten werden kann, desto geringer ist das Risiko, Opfer eines automatisierten Angriffs zu werden *(siehe Abbildung 1)*. Die zeitnahe Identifikation von Schwachstellen ist deshalb zentral. Der Threat- Modelling-Ansatz ist dabei von großem Nutzen, weil er auf einem iterativen Ansatz basiert.

Im folgenden Abschnitt sehen wir, wie Schwachstellen systematisch identifiziert, priorisiert, mitigiert und rapportiert werden können.

Was braucht es?

Neben einer bewährten Vorgehensweise, guter Kenntnis des eingesetzten Technologie-Stacks, Verfügbarkeit der notwendigen Ressourcen (Experten) und effizienten Werkzeugen, erfordert nachhaltiges Vulnerability Management vor allem eines: Kontinuität. Der Vulnerability Management Lifecycle soll die kontinuierliche Erkennung und Priorisierung von Bedrohungen, rechtzeitige Maßnahmen und eine gezielte Berichterstattung an alle Beteiligten im Unternehmen ermöglichen.

Wie häufig der Vulnerability-Management-Zyklus *(siehe Abbildung 2)* durchlaufen werden muss, hängt stark vom Technologie Stack, der Veränderungsrate und der Exponiertheit des Systems ab. Faktoren wie Schutzbedarf, Security Policies und Compliance-Anforderungen sind ebenfalls zu berücksichtigen. In der Praxis hat sich ein jährlicher bis halbjährlicher Review bewährt. Zusätzlich ist eine Prüfung auf neue Schwachstellen jeweils bei Systemänderungen zu empfehlen.

Referenzbeispiel „Managed Database Platform"

Beim Referenzbeispiel handelt es sich um eine DBaaS-Plattform innerhalb einer umfangreichen Enterprise-Cloud-Lösung (Private Cloud). Die Managed-Database-Plattform *(siehe Abbildung 3)* besteht aus einem Orchestration-Layer für das automatische Deployment und die Verwaltung der Datenbank-Services sowie einem Platform-Layer für den eigentlichen Kunden-Workload. Das DevOps-Team ist für Entwicklung, Betrieb und Sicherheit des Gesamtsystems verantwortlich. Die Systemgrenze (Platform Perimeter) umfasst den Orchestration- und den Platform-Layer. Die Orchestration-Services werden in einem Management-Tenant betrieben; der Database Platform-Layer basiert auf einer Oracle Private Cloud Appliance.

Threat Modeling

„The best use of threat modeling is to improve the security and privacy of a system

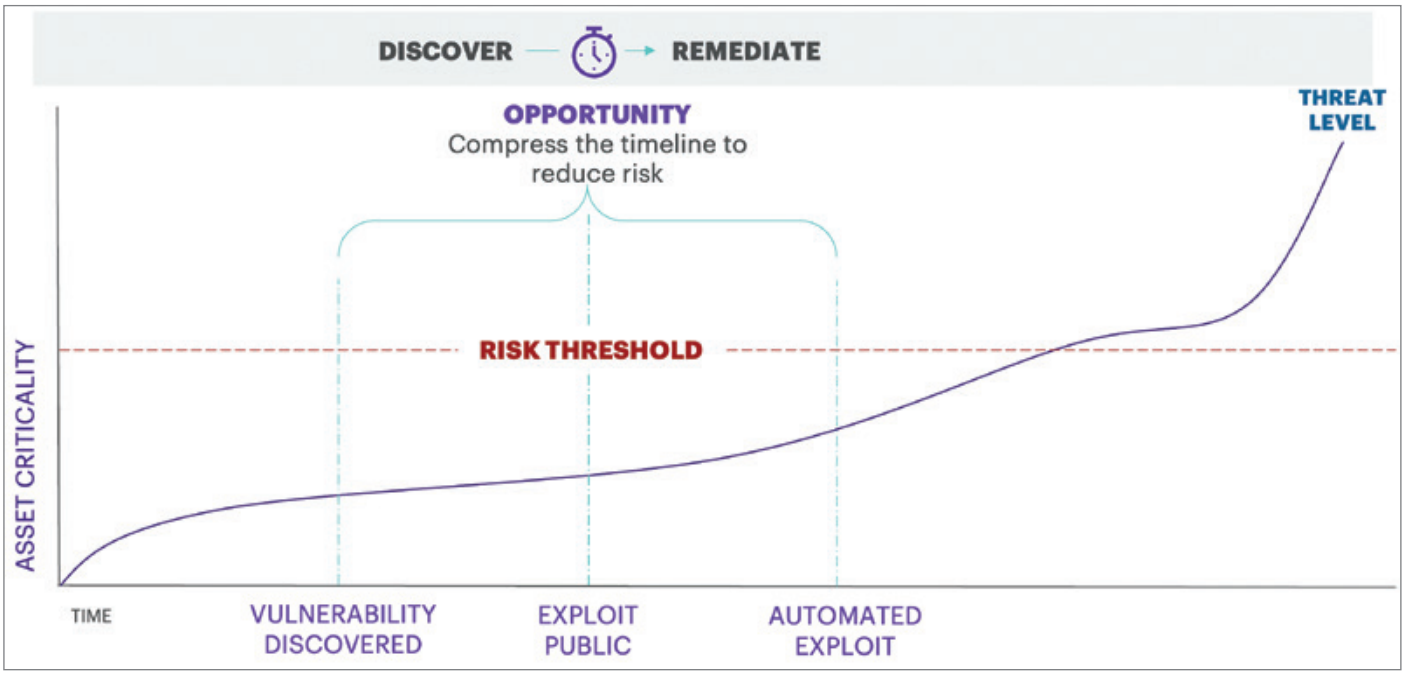


Abbildung 1: Window of Opportunity (© Accenture)

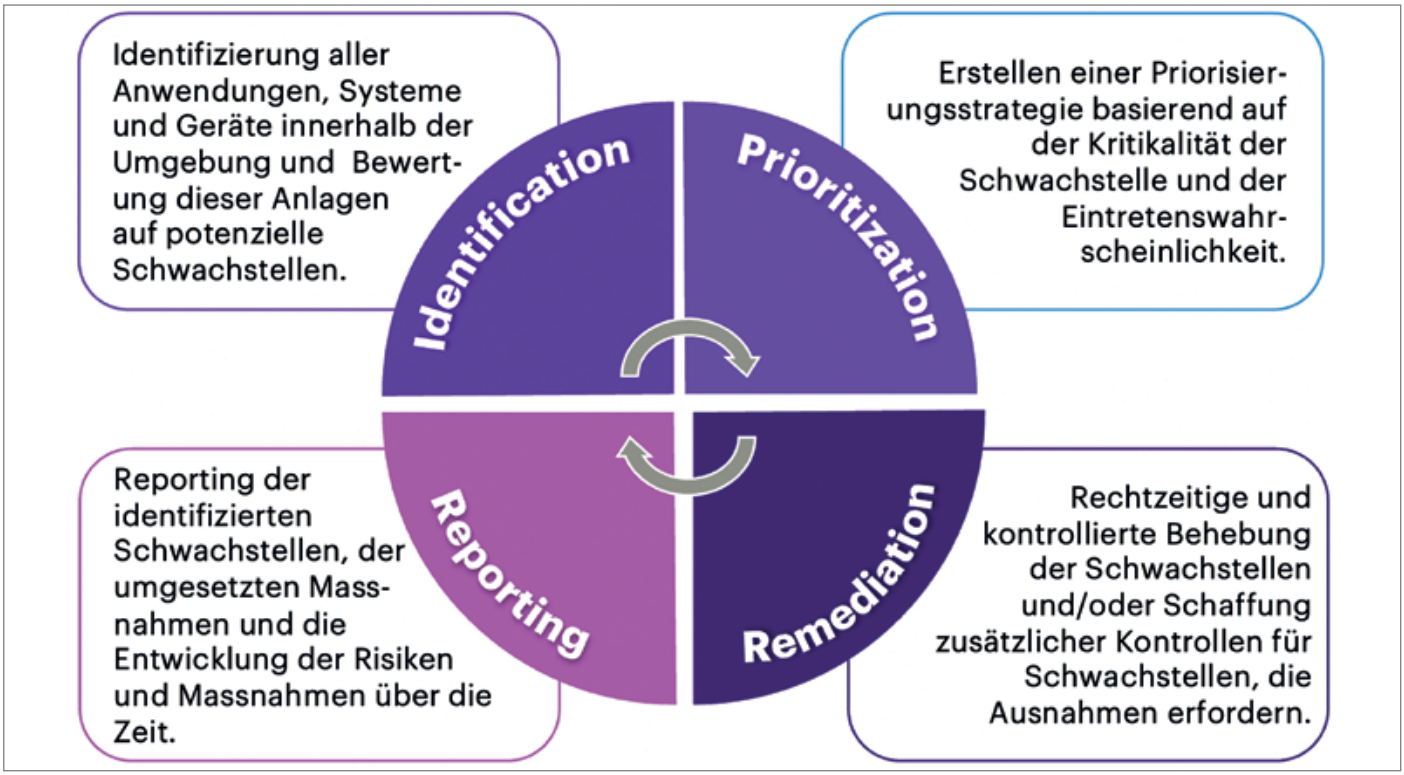


Abbildung 2: Vulnerability Management Lifecycle (© Accenture)

through early and frequent analysis." Threat Modeling Manifesto *[3]*

Beim „Threat Modeling" – also der Schwachstellen-Analyse – geht es um vier zentrale Fragen:

- Was willst du bauen? Was hast du gebaut?
- Was kann schief gehen?
- Was solltest du gegen die Dinge tun, die schief gehen können?
- Hast du eine gute Analyse durchgeführt?

Im ersten Schritt zeichnet man ein Diagramm des zu analysierenden Systems. Dabei handelt es sich um eine stark vereinfachte Darstellung der Systemarchitektur *(siehe auch Abbildung 3)*. Die verwendete Symbolik beschränkt sich auf Services, Schnittstellen/Verbindungen und Systemgrenzen.

Auch wenn ein möglichst komplettes Bild der Systemlandschaft angestrebt wird: Vollständigkeit ist nicht das oberste Ziel. Es geht vielmehr darum, kritische Schwachstellen mit hohem Risikopotenti-

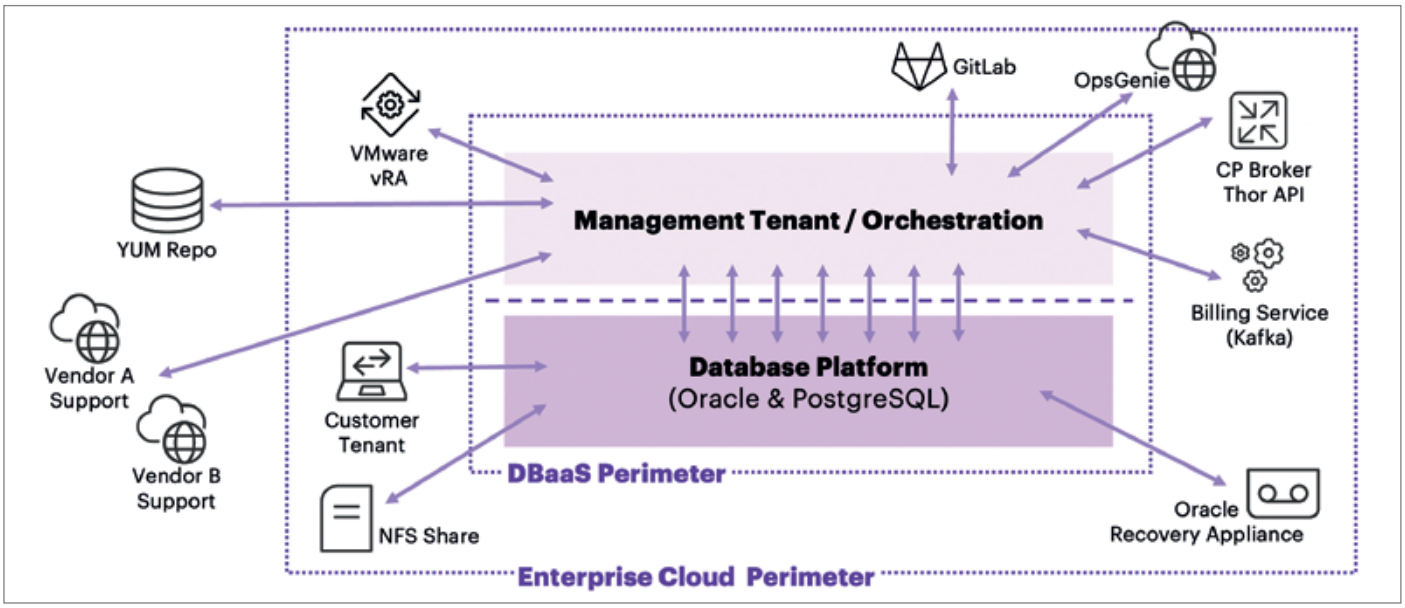


Abbildung 3: Referenzbeispiel Managed Database Platform (Quelle: Daniel Steiger)

al rasch zu erkennen und entsprechende Maßnahmen zeitnah umzusetzen.

Mit jeder Iteration wird das Modell vollständiger und umfangreicher. Zentral für eine belastbare System- und Schwachstellenanalyse ist dabei, die Mitwirkung von Experten aus den Bereichen Netzwerk, Datenbank, Storage, Backup, Automation, Betrieb, Systemadministration, Applikation und Security. Threat Modelling ist eine Teamleistung aller am System beteiligten Experten – nicht nur der Security-Experten!

Eine ausführlichere Beschreibung der oben genannten Prinzipien ist im Threat Modelling Manifest *[3]* festgehalten.

Trust Boundaries

Für die Schwachstellenanalyse wird das Diagramm mit so genannten „Trust Boundaries" (Vertrauensgrenzen) ergänzt. Trust Boundaries sollten überall dort eingezeichnet werden, wo verschiedene Personen verschiedene Dinge kontrollieren *(siehe Abbildung 4)*. Beispiel einer solchen Trust Boundary ist die Grenze zwischen dem Kunden-Tenant und einem Datenbankservice auf der Datenbank-Plattform (TB 7). Der Kunden-Tenant ist unter der Kontrolle des Endkunden, während die Datenbankplattform von der Plattformbetreiberin kontrolliert wird. In unserem Referenzbeispiel sind alle zu analysierenden Trust Boundaries eingezeichnet *(TB 1 bis 11)*.

Sind die relevanten Trust Boundaries definiert, folgt der anspruchsvollste Teil der Analyse: die Beschreibung möglicher Threats (Bedrohungen) für alle Trust Boundaries. Dazu muss man sich in die Rolle eines Angreifers versetzen und realistische Bedrohungsszenarien formulieren. An diesem Punkt kommt das STRIDE-Framework ins Spiel.

STRIDE-Framework

STRIDE hilft relevante „Was-kann-schiefgehen"-Fragen *(siehe auch Einleitung)* zu formulieren und damit Schwachstellen systematisch und gezielt zu identifizieren.

STRIDE beschreibt sechs Bedrohungs-Kategorien. Das STRIDE-Akronym steht für **S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service und **E**levation of Privilege:

- **Spoofing** – Vorgeben, etwas oder jemand zu sein, der man nicht ist.
- **Tampering** – Etwas verändern, das man nicht verändern darf. Dazu zählen Daten im Netzwerk, auf Festplatten oder im Arbeitsspeicher.
- **Repudiation** – Etwas abstreiten, respektive zu behaupten, dass man etwas nicht getan hat (unabhängig davon, ob man es getan hat oder nicht).
- **Information Disclosure** – Offenlegung und Weitergabe von Informationen an Personen, die nicht berechtigt sind diese einzusehen.
- **Denial of Service** – Angriffe, die darauf abzielen, ein System zum Absturz zu bringen, es unbrauchbar langsam zu machen, oder den gesamten Speicher zu füllen.
- **Elevation of Privilege** – Erhöhung der Berechtigung, so dass ein Programm oder ein Benutzer technisch in der Lage ist, Dinge zu tun, die es/er nicht machen darf.

Die sechs Bedrohungskategorien von STRIDE beschreiben ziemlich genau das Gegenteil der typischerweise erwünschten Systemeigenschaften *(siehe Abbildung 5)*. Um die erwünschten Systemeigenschaften zu schützen, gilt es also mögliche Threats zu identifizieren und abzuwehren.

In unserem Referenzbeispiel konnten wir mit Hilfe der STRIDE-Methode zahlreiche Schwachstellen identifizieren und beheben. Dies ist beispielhaft anhand der beiden folgenden Threats illustriert.

Beispiel 1: *Trust Boundary Jumphost*

Threat „Repudation": Kann ein DevOps-Engineer abstreiten, dass sie/er eine Aktion ausgeführt hat? Ja, das wäre möglich, weil aktuell kein Command-Level-Logging auf den Jumphosts aktiviert ist und zudem haben die Engineers Admin-Rechte auf den Jumphosts.

Mitigation: dieser Threat lässt sich durch eine zentrale Syslog-Lösung eliminieren. Zudem sollten die Admin-Rechte auf den Jumphosts entfernt werden.

Beispiel 2: *Trust Boundary GitLab Runner*

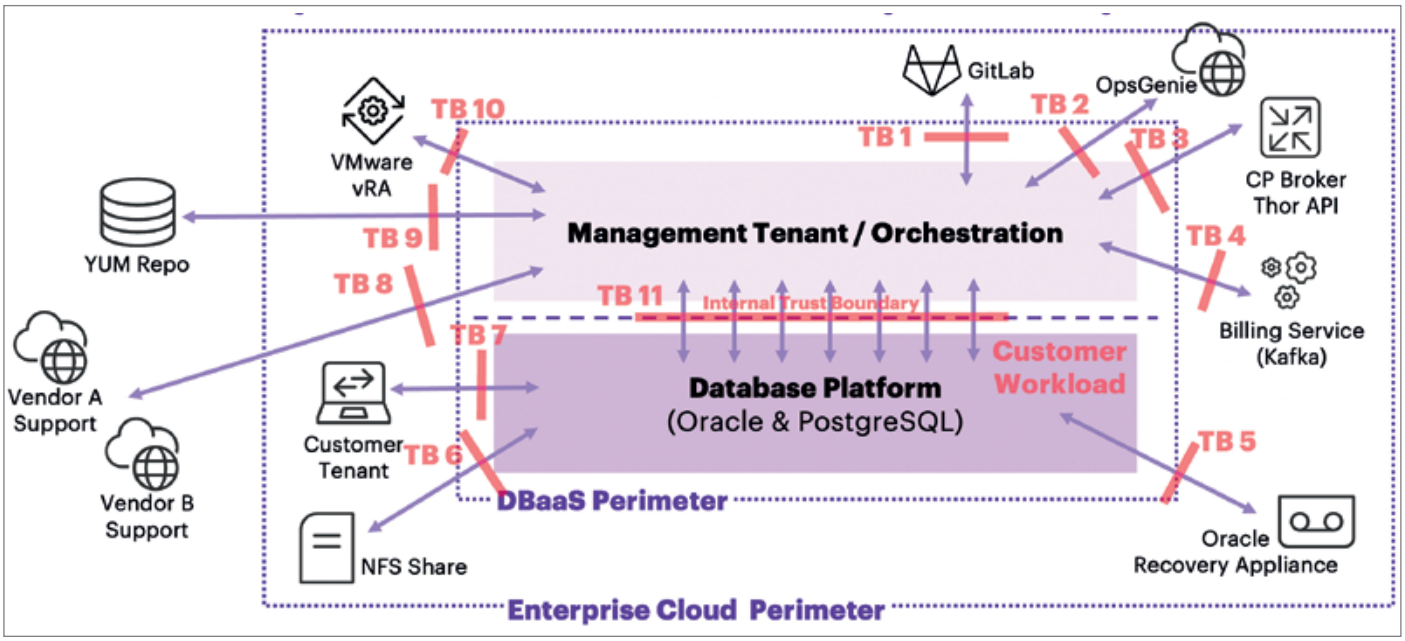


Abbildung 4: Diagramm mit Trust Boundaries (Quelle: Daniel Steiger)

Threat	Desirable Property
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-Repudiation
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Abbildung 5: STRIDE – Threats vs. Systemeigenschaften (Quelle: [1] Adam Shostack)

Threat „Elevation of Privilege“: Kann ein DevOps-Engineer Befehle mit erweiterten Rechten ausführen? Ja, das wäre möglich, weil die GitLab-Runner-Prozesse als Root ausgeführt werden. Zudem können Personen mit Write Access auf das Git-Repository die CI/CD-Pipeline ändern.

Mitigation: dieser Threat lässt sich einfach eliminieren, indem man die GitLab-Runner- Prozesse im User-Mode ausführt.

Das „Elevation of Privilege“-Kartenspiel – Threat Modelling the fun way

„Elevation of Privilege“ (EoP) ist ein Kartenspiel *[2]*, welches beim Threat Modelling hilft, auf spielerische Art und Weise STRIDE Security-Threats zu identifizieren *(siehe Abbildung 6)*. „Elevation of Privilege“ fördert zudem den interdisziplinären Austausch, weil es idealerweise im erweiterten DevOps-Team gespielt wird. Eine EOP-Runde kann auch mal zwischendurch gespielt werden. Das EOP-Kartenset besteht im Wesentlichen aus je 14 Situationskarten pro STRIDE-Kategorie. Jede Karte beschreibt eine mögliche Bedrohungssituation. In der Kategorie Information Disclosure ist beispielsweise folgende Situation beschrieben: „An attacker can see error messages with security-sensitive content“. Die Situationskarten erleichtern den Einstieg in die Threat-Modelling-Methode, weil damit bereits ein Großteil typischer Bedrohungssituationen gut beschrieben ist.

Risk Register und Vulnerability Dashboard

Die nach der STRIDE-Methode beschriebenen Schwachstellen werden nun systematisch auf ihre Eintretenswahrscheinlichkeit hin bewertet. Die Bewertung basiert auf der Einschätzung entlang der Frage, wie einfach die Schwachstelle entdeckt werden kann, wie verbreitet und wie bekannt diese ist und welche Folgen die Ausnutzung der Schwachstelle haben könnte. Je höher das daraus resultierende Rating ist, desto größer ist das Risiko (Low, Medium, High) für das Unternehmen. Wir haben für die Ermittlung der Risiken ein Excel-File verwendet, mit dem für jedes Threat das Risiko aufgrund von vier Faktoren berechnet wird *(siehe Abbildung 7)*.

Im einfachsten Fall dient die Risikobewertung wie in *Abbildung 7* dargestellt auch gleich als Risk Register und Vulnerability Dashboard. In Projekten mit mehreren DevOps-Teams empfiehlt es sich jedoch, die Ergebnisse aus der Risikobewertung in ein zentrales Gefäß, wie ein Wiki, respektive ein ITSM-Tool zu überführen, wo die Threats entspre-

Abbildung 6: „Elevation of Privilege" (EoP) Kartenspiel (Quelle: [2] agilestationery.com)

chend ihrer Kritikalität priorisiert und die Umsetzung (Mitigation, Remediation) getrackt wird.

Weitere Informationsquellen für die Risikobewertung sind in *Abbildung 8* ersichtlich. Viele Hersteller stellen Informationen zu Sicherheitslücken und Bugs ihrer Produkte in Form von Security Reports und Security Alerts zur Verfügung. Generische Security Scans, wie die CIS-Benchmarks, sind für alle verbreiteten Betriebssysteme und Datenbanken verfügbar und liefern sehr detaillierte Reports. Die CIS-Scans lassen sich auch einfach automatisieren. Ebenfalls kann ein Pentest von unabhängigen externen Spezialisten wertvolle Informationen über den Sicherheitszustand der Systemumgebung liefern.

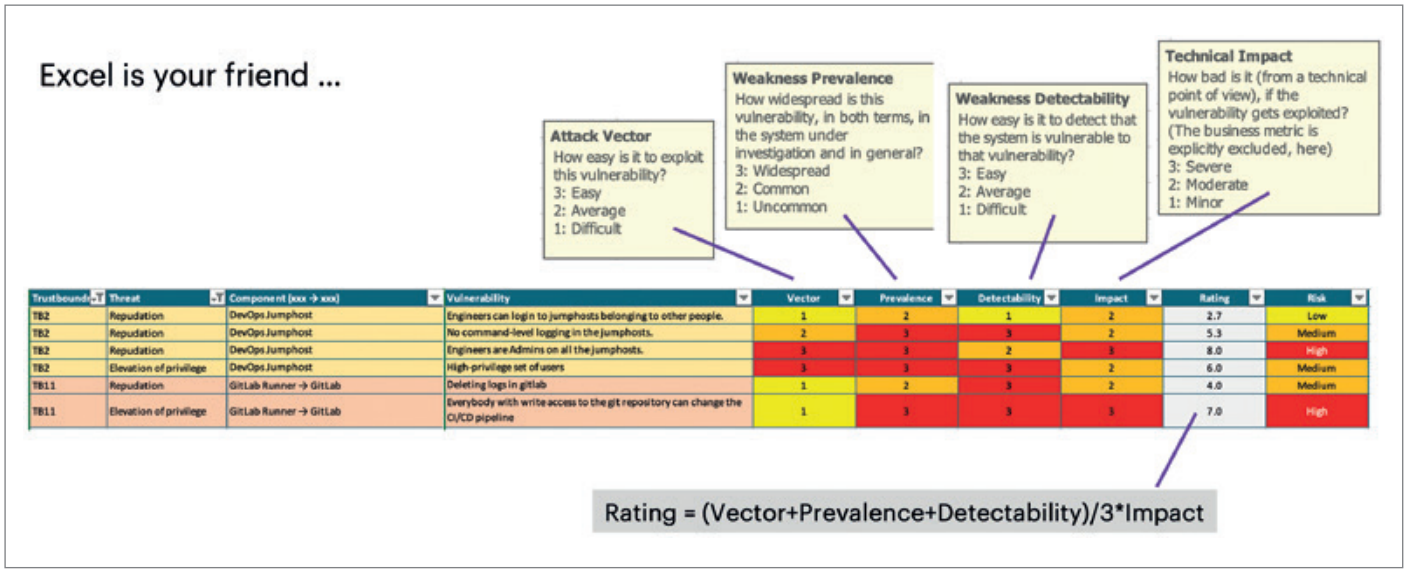
Was ist gegen die Dinge zu tun, die schief gehen können?

Als nächster Schritt erfolgt das so genannte Risk Treatment, also die Behandlung der Risiken, frei nach dem Credo: *„Mitigate it, eliminate it, transfer it, or accept it"*. In vielen Fällen können die Risiken durch Einspielen aktueller Patches, Firmware oder SW-Updates eliminiert werden. Auch das Hardening von Betriebssystem und Datenbanken, sowie die Re-Konfiguration von Firewalls und Netzwerkkomponenten sind typische Entschärfungsmaßnahmen. Lässt sich ein Risiko nicht auf diese Weise eliminieren oder entschärfen, muss gegebenenfalls ein Re-Design, respektive eine Architekturänderung, ins Auge gefasst werden.

Hast du eine gute Analyse durchgeführt?

Die Validierung des Threat-Modells ist der letzte Schritt, welcher im Rahmen des Threat Modellings erfolgt. Ist das Modell vollständig? Ist es hinreichend genau? Deckt es alle Sicherheitsentscheidungen ab, die getroffen wurden? Kann die nächste Version mit diesem Diagramm ohne Änderungen erstellt werden?

Im Vulnerability Management Lifecyle gibt es keine „Definition of Done". Das heißt, die Validierung des Threat-Modells markiert immer auch den Beginn der nächsten Iteration: Das Diagramm muss aktualisiert werden, Diagramm-Details sind aufgrund neuer Gegebenheiten zu



Trustbound	Threat	Component (xxx → xxx)	Vulnerability	Vector	Prevalence	Detectability	Impact	Rating	Risk
TB2	Repudation	DevOps Jumphost	Engineers can login to jumphosts belonging to other people.	1	2	1	2	2.7	Low
TB2	Repudation	DevOps Jumphost	No command-level logging in the jumphosts.	2	3	3	2	5.3	Medium
TB2	Repudation	DevOps Jumphost	Engineers are Admins on all the jumphosts.	3	3	2	3	8.0	High
TB2	Elevation of privilege	DevOps Jumphost	High-privilege set of users	3	3	3	2	6.0	Medium
TB11	Repudation	GitLab Runner → GitLab	Deleting logs in gitlab	1	2	3	2	4.0	Medium
TB11	Elevation of privilege	GitLab Runner → GitLab	Everybody with write access to the git repository can change the CI/CD pipeline	1	3	3	3	7.0	High

Abbildung 7: Risikobewertung von Threats (Quelle: Daniel Steiger)

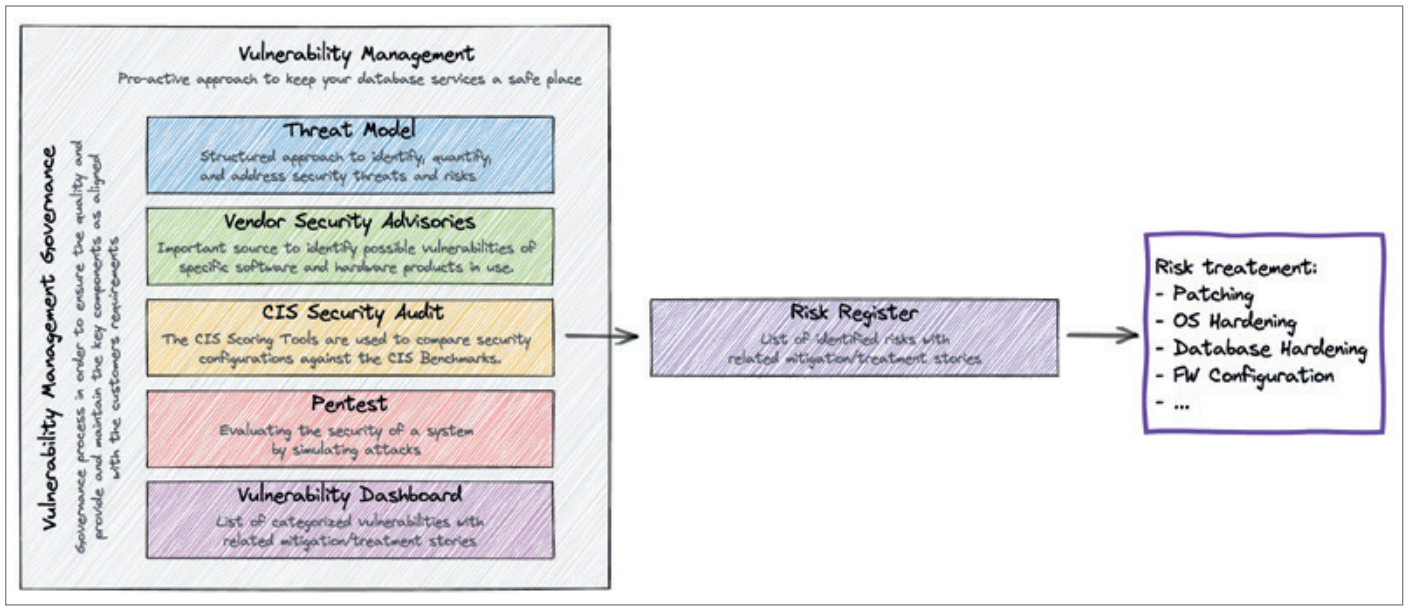


Abbildung 8: Vulnerability Management Building-Blocks (Quelle: Daniel Steiger)

bereinigen, Threats sind zu überprüfen und gegebenenfalls neu zu beschreiben, Tests sind zu überprüfen und gegebenenfalls anzupassen.

Threat-Modelling-Tipps

In der praktischen Umsetzung der Threat-Modelling-Methode haben sich nach meiner Erfahrung ein paar Dinge bewährt:

- „Get the right people in the room“: Bring Lösungsarchitekt, Entwickler mit einem guten Verständnis des Codes und der Datenflüsse, Produkt Owner, und Security Champion an einen Tisch (resp. vor das Flip-Chart).
- „Just start, and don’t try to be perfect“ Man muss keine Security-Spezialistin und kein Security-Spezialist sein, um mit Threat Modelling zu beginnen.
- Fokussiere dich bei der Automatisierung auf einfache und repetitive Aufgaben wie Vulnerability Scanning – nicht alle Aufgaben lassen sich sinnvoll automatisieren.
- Projekt- resp. gesamtsystemübergreifende Guidelines für das Risiko-Rating und das Reporting definieren (inkl. zentrale Reporting-Plattform).
- Methoden-Unterstützung und der Austausch zwischen den Technologie-Teams beschleunigt das Ramp-Up für Threat-Modell-Beginner.

Nützliche Tools

Im Rahmen unserer Threat-Modellling-Aktivitäten haben wir im Team folgende Tools verwendet:

- Whiteboard für Skizzen und Entwürfe
- draw.io für das Threat-Model-Diagramm und für Detailskizzen
- Excalidraw für die Building-Block-Abbildung
- Microsoft Excel für die Risikobewertung

- Wiki (Dokumentation, Vulnerability Dashboard, Risk Matrix, Risk Tracking)
- Elevation of Privilege Game [2]
- Thread Modeller (u. a. Microsoft SDL Threat Modeller) für das Selbststudium und als Referenz für eigene Zeichnungen in draw.io

Im Weiteren haben wir CIS-CAT Pro für die CIS Security Audits und OpenSCAP für den Benchmark für das Oracle RDBMS (Security Content Automation Protocol *(https://www.open-scap.org/)* verwendet. Zusätzlich haben wir periodisch die produktspezifischen Security Advisories und Critical Security Alerts ausgewertet. Für OCI-Kunden bietet Oracle einen OCI Vulnerability Scanning Service (OCI VSS).

Fazit, oder: Welchen Mehrwert bringt Threat Modelling?

Auch wenn sich der Mehrwert schwer quantifizieren lässt: Vulnerability Management ist ein zentraler Bestandteil der Security-Architektur. Nach zwei Jahren praktischer Anwendung der Threat-Modelling-Methode konnten wir diverse Schwachstellen identifizieren und die Systemsicherheit im Sinne eines iterativen Vorgehens kontinuierlich verbessern. Die eine oder andere Schwachstelle war intern bereits bekannt. Schlussendlich ist es aber dem Vulnerability-Management-Vorgehen zuzuschreiben, dass das DevOps-Team wiederkehrend die eigene Lösungsarchitektur kritisch hinterfragt und konkrete Maßnahmen zur Verbesserung der Systemsicherheit konsequent umgesetzt hat. Die Stärke der Methode liegt ohne Zweifel in der interdisziplinären Zusammenarbeit und dem Einbezug verschiedenster Perspektiven bei der Analyse.

Das Threat Modelling Manifest *[3]* bringt den Mehrwert auf den Punkt:

- A culture of finding and fixing design issues over checkbox compliance.
- People and collaboration over processes, methodologies, and tools.
- A journey of understanding over a security or privacy snapshot.
- Doing threat modeling over talking about it.
- Continuous refinement over a single delivery.

(Quelle: https://www.threatmodelingmanifesto.org/#values)

In diesem Sinne: Viel Erfolg und Happy Threat Modelling!

Quellen/Referenzen

[1] Adam Shostack 2014: Threat Modeling: Designing for Security. John Wiley & Sons, Inc., Indianapolis, Indiana

[2] Elevation of Privilege (EoP) Card Deck, by *https://agilestationery.com/*

[3] Threat Modeling Manifesto: *https://www.threatmodelingmanifesto.org*
Adam Shostack Blog: *https://shostack.org/*

Über den Autor

Daniel Steiger ist Tech Architecture Manager bei Accenture Data&AI (vormals Trivadis) und Database Platform Infrastructure Architect mit mehr als 25 Jahren Erfahrung als Berater und Projektleiter. In den letzten Jahren war Daniel Steiger Product Owner einer Database Service Platform für ein großes Schweizer Telekommunikationsunternehmen. In dieser Rolle leitete er ein interdisziplinäres Team von zehn Ingenieuren innerhalb einer großen agilen Organisation (SAFe). Neben seiner Rolle als PO, agiler Coach und Architekt, war Daniel Steiger auch für die Compliance Delivery verantwortlich.

Daniel Steiger
daniel.steiger@accenture.com

Oracle Datenbanken Monthly News

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

DOAG Online

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

Alles wird wieder in einem Video präsentiert.

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

https://www.doag.org/de/home/news/oracle-datenbanken-monthly-news-24/

Thementag zu Oracle Forms und APEX auf der DOAG 2023 Konferenz & Ausstellung in Nürnberg am 21.11.2023

Frank Hoffmann und Carolin Krützmann

Eine gute Nachricht für alle Forms-Kunden ist, dass das DOAG-Development-Team dieses Jahr entschieden hat, Forms – neben Webinaren und Artikeln im „Red Stack Magazin" – auch auf der diesjährigen Konferenz einen Platz am Thementag einzuräumen. Eine gute Möglichkeit fundiertes Wissen zu erlangen, Fragen direkt an den Produktmanager zu stellen oder sich mit anderen Experten zu vernetzen, um Fachwissen zu teilen.

Alle **Forms-Interessenten** aus Österreich, der Schweiz und Deutschland sind ganz herzlich nach Nürnberg eingeladen, um sich gemeinsam zu vernetzen und mit den Referenten und Besuchern ins Gespräch zu kommen.

Warum investieren wir weiter in Oracle Forms? Weil es noch eine ganze Reihe von DOAG-Kunden gibt, die das Produkt einsetzen. Viele komplexe Lösungen wie WWS, ERP und Fachanwendungen wurden vor 20 bis 25 Jahren für viel Geld entwickelt und laufen bis heute sehr zuverlässig im 24/7 Betrieb. Eine Neuentwicklung würde je nach Tool den zwei bis dreifachen Aufwand der Erstentwicklung kosten, viele Ressourcen binden und ein neues Entwicklerteam erfordern. Da entscheidet man sich oft, wenn die Software weiter ihren Zweck gut erfüllt, bei Forms zu bleiben.

Eine weitere sehr gute Nachricht erreichte mich letzte Woche von **Michael Kilimann, dem Geschäftsführer des IT-Machers:** Mit dem langjährigen Forms Top Referenten **Dr. Jürgen Menge** werden wieder nach Bedarf Oracle-Forms-Schulungen zu fairen Preisen angeboten. Weitere Experten stehen dabei auch für Spezialthemen zur Verfügung. Nach den vielen mühsamen Corona-Jahren sollen die Schulungen in kleinen Gruppen (5-10 Personen) im schönen Wolfratshausen stattfinden. Der IT-Macher wird auch aktiv den Thementag mitgestalten.

Mit der aktuellen Version **12.2.1.19** und dem für 2024 offiziell (laut weblogic sod Juni 2023) angekündigten neuen Long-Term Release **14.1.2** (Forms and Reports – Support 2032++), was auch maßgeblich durch die Petition der DOAG 2019 zustande gekommen ist, gibt es einige gute Perspektiven für den weiteren Betrieb der mehr als 40 Jahre alten Backend Software von Oracle und Chancen zur nativen UI/UX-Modernisierung (u. a. materialized Items, Roll-Over-Funktionen, Current Record Indicator, Rest Services, LOV Autocomplete, Responsive Multiblock), ohne den Programmcode ändern zu müssen.

Als Themen dieses Jahr sind unter anderem zwei Vorträge des Oracle-Forms-Produktmanagers **Michael Ferrante** eingereicht worden. Ein Vortrag zum aktuellen und ein weiterer zu den zukünftigen Releases von Forms, daneben ein Vortrag zum Hybridbetrieb von Forms und APEX des **IT-Machers**, sowie ein Vortrag zur automatisierten Anpassung von Forms von **Jan Peter Timmermann** mit dem Forms API Master. Ein weiterer spannender Vortrag wird über ein Modernisierungsprojekt „Forms:Forms" der **Universitätsklinik Bonn** gehalten. Der Referent, geht der Frage nach, warum selbst nach 25 Jahren der Ambulanzbetrieb mit einer Oracle-Forms-Lösung so erfolgreich ist und nun nach reiflicher Überlegung auf das neueste Forms-Release und die weiteren angekündigten Forms-Versionen migriert und modernisiert werden soll.

Am Abend zuvor ist eine Forms-Happy-Hour geplant, um ausführlich über die Anekdoten von Forms ins Gespräch zu kommen, die Zuneigung für das Produkt zum Ausdruck zu bringen, und schon mal den goldenen fünfzigsten Produkt-Geburtstag mit dem Forms-Erfinder Bill Friend (1979), dem größten Förderer Sohaib Abbasi (1985-2003) und natürlich Michael Ferrante (Forms PM 12,14) für 2029 zu planen. ☺ Bei Interesse an der Happy-Hour oder Fragen zum Forms-Thementag bitte eine E-Mail an: *Frank.Hoffmann@doag.org* (DOAG Themenverantwortlicher Oracle Forms)

Der APEX-Thementag auf der DOAG 2023 Konferenz + Ausstellung

Ein bahnbrechender Thementag erwartet alle APEX-Enthusiasten auf der diesjährigen DOAG-Konferenz in Nürnberg. Mit einer Vielzahl von innovativen Vorträgen, interaktiven Barcamp-Sessions und praxisorientierten Workshops wird der APEX-Thementag ein absolutes Highlight für alle, die sich für die Zukunft dieser leistungsstarken Low-Code-Entwicklungsplattform interessieren. Der Tag wird mit einer spannenden Keynote von Marc Sewtz, dem Senior Director of Software Development (APEX), eröffnet. In seiner Präsentation wird er die neuesten Entwicklungen und Funktionen von APEX im Jahr 2023 vorstellen.

Nach der inspirierenden Keynote folgt am Vormittag ein interaktives Barcamp-Format, das den Teilnehmenden die Möglichkeit gibt, ihre eigenen Themen auf die Agenda zu bringen. In kleinen Gruppen können sie ihre Ideen diskutieren, ihr Wissen teilen und wertvolle Einblicke von anderen APEX-Anwendern gewinnen. Das Barcamp fördert den offenen Austausch und ermöglicht es den Teilnehmern, individuelle Fragen zu stellen und konkrete Herausforderungen bei der Anwendungsentwicklung zu diskutieren. Es ist eine großartige Gelegenheit, von Good Practices zu lernen und das eigene Netzwerk mit anderen APEX-Enthusiasten zu erweitern.

Am Nachmittag stehen verschiedene praxisorientierte Workshops auf dem Programm, die den Teilnehmern die Möglichkeit bieten, sich intensiv mit spezifischen Themen auseinanderzusetzen.

Die Workshops sind darauf ausgerichtet, den Teilnehmern praktische Fähigkeiten zu vermitteln und ihnen die Gelegenheit zu geben, Themen zu erkunden, die normalerweise auf einer Konferenz zu kurz kommen.

Während der Workshops sind die Teilnehmer explizit dazu aufgerufen, Fragen zu stellen oder Anforderungen zu stellen.

Im ersten Workshop werden Niels de Bruijn und Moritz Klein „Flows for APEX" anhand von praktischen Beispielen vorstellen. Die Teilnehmer werden lernen, komplexe Workflows in APEX zu erstellen und zu verwalten.

Sie werden praktische Einblicke erhalten und wertvolle Techniken kennenlernen, um effiziente Arbeitsabläufe in ihren Anwendungen zu implementieren.

Im zweiten Workshop zeigen Carsten Czarski und Moritz Klein, was man alles mit den RESTful-Services in APEX machen kann.

Die Teilnehmer werden die Möglichkeit haben, die Potenziale von RESTful-Services in APEX voll auszuschöpfen und zu erfahren, wie sie diese effektiv in ihre Anwendungen integrieren können. Sie werden lernen, wie Sie Daten mit anderen Systemen austauschen, APIs nutzen und nahtlose Integrationen schaffen können.

Abgerundet wird das Ganze mit einem Workshop über die Anpassung einer APEX-Applikation auf das Corporate Design von Alli Pierre Yotti und Andreas Upgang.

Hier werden die Teilnehmer lernen, wie sie das Erscheinungsbild und die Benutzererfahrung ihrer APEX-Anwendungen an die individuellen Anforderungen ihres Unternehmens anpassen können.

Von der Gestaltung von Benutzeroberflächen bis hin zur Erstellung einer konsistenten Markenidentität wird dieser Workshop den Teilnehmern wertvolle Einblicke in die Gestaltung ansprechender und professioneller Anwendungen geben.

Der APEX-Thementag wird eine einzigartige Gelegenheit bieten, die neuesten Funktionen von APEX kennenzulernen, praktische Fähigkeiten zu erwerben und sich mit anderen APEX-Enthusiasten auszutauschen.

Seien Sie dabei und lassen Sie sich von der Zukunft von APEX inspirieren!

BERLINER EXPERTENSEMINARE

Die Berliner Expertenseminare sind Expertenschulungen und Weiterbildungen der DOAG, die mit einer Hands-on-Mentalität über zwei Tage geballtes Fachwissen mit praxisnahen Übungen vermitteln. Profis geben in kleiner Runde ihr großes Know-how weiter und sorgen für einen optimalen Wissenstransfer, der unmittelbar danach angewendet und in die täglichen Aufgaben und Herausforderungen fließen kann. Für ein exquisites Buffet ist während des gesamten Seminars ebenfalls gesorgt. Die Schulungen dauern täglich bis 17 Uhr. Im Anschluss an den ersten Seminartag wartet eine Abendveranstaltung auf die Teilnehmerinnen und Teilnehmer.

30. - 31.08.2023

Oracle Cloud Infrastructure – von der Konsole zur Automation – Kickstart!

Berliner Expertenseminar mit Stefan Oehrli und Martin Berger

Es wird gemeinsam eine Umgebung bestehend aus Compute-Instanzen und Datenbanken aufgebaut. Danach konfigurieren Sie in praktischen Übungen Terraform und deployen anschließend unterschiedliche Ressourcen mithilfe von Terraform und OCI Stacks. Dazu gehören Compute Instances, Database Services, Autonomous Database, Load Balancer und mehr.

27. - 28.09.2023

Oracle Datenbank Indexing

Berliner Expertenseminar mit Randolf Eberle-Geist

In diesem Seminar werden die wichtigsten Themen bezüglich Indizierung mit B*Tree und Bitmap Indizes in der Oracle-Datenbank behandelt – Text / XML / JSON / Domain-Indizes werden zwar je nach verfügbarer Zeit erwähnt und beschrieben, der Schwerpunkt liegt eindeutig auf den B*Tree / Bitmap Indizes.

DOAG

DOAG Datenbank mit Exaday 2023

ON DEMAND

DATENBANK 2023 VERPASST?

Jetzt On-demand-Ticket buchen und Vortragsaufzeichnungen anschauen!

ALLE ANGEBOTE IM TICKETSHOP

BUSINESS

NEWS

05/2023

REQUIREMENTS ENGINEERING

Requirements Engineering – das Fundament der Softwareentwicklung

Stephan Tönnies, DOAG – Softskills Initiative

Das erfolgreiche Abschließen von Softwareentwicklungsprojekten stellt immer wieder eine Herausforderung dar. Oftmals scheitern sie nicht an technischen Umsetzungen, sondern an Fehlentwicklungen oder ungenauen Aufgabenstellungen. Im Folgenden wird ausgeführt, was Requirements Engineering (Anforderungserhebung) ist und wie es in Softwareprojekten eingebunden werden kann.

Bau mir einfach ein Haus!

Stell dir folgende Situation vor… Eine Person möchte gerne etwas bauen. Sie geht zu einem Handwerker und sagt: „Hier hast du ein Grundstück! Ich möchte gerne ein Haus drauf bauen."

Der Handwerker antwortet: *„Ok, und wie soll es aussehen?"*

„Wie ein Haus halt! Eine Eingangstür, ein paar Fenster und ein Dach. Und hier hast du schonmal das Geld. Du kannst bestimmt jetzt anfangen. Es ist ja sicherlich nicht zu schwer ein Haus zu bauen."

„Ok, dann fange ich einfach an und du erklärst mir in den nächsten Wochen wie es konkret aussehen soll."

Ein paar Wochen später treffen sie sich wieder und der Kunde stellt fest: „Fangen wir mit der Küche an. Ich möchte gerne darin zwei Fenster."

„Ok, und sollen diese irgendwie bestimmt aussehen?"

„Wie Fenster halt. Glasig und durchsichtig."

„Ok, ich weiß genau, was du haben möchtest!"

Einige Zeit später treffen sie sich erneut und der Kunde stellt fest:

„Was sind denn das für Fenster?"

„Ganz normale Standardfenster."

„Ich wollte aber bodentiefe! Das ist doch glasklar. Das hat unter meinen Bekannten jeder! Und was ist das hier eigentlich für ein Raum? Hier passt ja gar nicht meine Küche rein?"

So oder so ähnlich wird heutzutage immer noch Softwareentwicklung betrieben. Dabei ist ein Softwareprojekt kein physisches Haus. Man muss nicht alle Aspekte schon vorab geklärt haben, jedoch sollte die Grundrichtung feststehen. Es gibt Gründe, warum Bauunternehmen erst mit dem Bau beginnen, wenn die Pläne stehen. Warum fangen wir in der Softwareentwicklung dann so häufig an, die Software zu entwickeln, obwohl das Konzept noch nicht steht? Dabei muss nicht unbedingt gleich ein komplettes Pflichtenheft erwartet werden. Im Agilen sollte eine feste Vision als erster Startpunkt genügen. Warum fällt es uns so schwer, dem Kunden diese Vision zu entlocken? Warum lassen wir unsere Kunden mit diesen vagen Aussagen davonkommen, obwohl dies zum Nachteil aller Beteiligten ist?

Argumente dafür können sein, dass es bequem ist. Kunden sind gelegentlich nicht entscheidungsfreudig; und wenn sie es sind, kann man nicht sicher sein, dass ihre Aussagen tatsächlich das eigene Problem beschreiben. Weniger entscheidungsfreudige Kunden fühlen sich schnell genervt und bedrängt, wenn konkrete Vorschläge erwartet werden. Dies kann dazu führen, dass alle Beteiligten sich unwohl fühlen und keine konkreten Ergebnisse zustande kommen. Konfliktscheue und Bequemlichkeit hindern uns daran, die Vision zu schärfen und über unsere Erwartungen zu sprechen. Das ist nur menschlich. Es hält uns jedoch davon ab, kunden- und zielorientiert zu arbeiten.

Im Gegenteil: Mit unpräzisen Informationen schaffen wir uns „Freiräume", die wir für die Umsetzung der Software ausfüllen müssen. Wir füllen diese mit den besten Absichten und Ideen, die den Kunden gefallen könnten. Diese reagieren dann oft nur oberflächlich auf die Vorschläge, da sie gelegentlich keine eigenen Ideen mitbringen. Der Kunde sagt daraufhin ein vorläufiges „Ja" zu den Ansätzen der Softwareentwickelnden. Die Softwareentwicklung fühlt sich darauf bestätigt und der Kunde freut sich, dass Fortschritte erzielt werden. Der Kunde setzt sich meist ernsthaft mit dem Thema auseinander, wenn die Lösung fertig entwickelt ist. Vorher hat dieser oft nicht die Kapazität, sich mit der Thematik genauer auseinanderzusetzen. Diese beiden Situationen, wohlwollende Entwicklung und oberflächlicher Kunde, stellen ein ideales Szenario für eine Fehlentwicklung dar. Wie der Handwerker in der Geschichte, der nach bestem Wissen und Gewissen die Wünsche des Kunden erfüllen wollte und der Kunde, der sich mit der Thematik erst am Ende auseinandersetzt, wenn schon das Gesamtprodukt fertiggestellt ist.

Die Lösung für dieses Problem ist einfach. Man müsste den Kunden nur genauer befragen. Er würde dann genauer antworten, sodas die Gefahr einer Fehlentwicklung sinkt. Natürlich stellt sich das in der Praxis als nicht so einfach dar. Ein Versuch, belastbarere Informationen vom Kunden zu erhalten, ist das Requirements Engineering (Anforderungserhebung).

Wer plant mein Hausbau?

Eine Erwartung von agilen Entwicklungsmethoden ist die Kundenzentrierung und der Value der eigenen Entwicklung im Zentrum des eigenen Handelns im Team. Dabei gehört es dazu, den Kunden nahe an der Entwicklung zu halten, um kurze Feedbackschleifen zu ermöglichen. Dabei soll beispielsweise ein Product Owner (PO) als stellvertretende Instanz die Kundenanforderungen abbilden, während sich die Entwicklung mit den Anforderungen des PO

auseinandersetzt. In manchen Fällen sind auch Endanwender direkt in den Reviews vertreten, welche Feedback und neue Impulse in die Entwicklung tragen. Die Fähigkeit, eine Sprache zu finden, in der die Entwicklung die Resultate auf verständlicher Art kommuniziert und gleichzeitig die Impulse richtig einordnet und auf diese eingeht, ist daher eine Schlüsselkompetenz für das agile Vorgehen.

Aus dieser Motivation heraus sind manche davon ausgegangen, dass dedizierte Requirements Engineers oder ähnliche Tätigkeiten nicht mehr benötigt werden, da diese von den allgemeinen Aufgaben eines agilen Entwicklungsteams übernommen werden. In der Theorie mag das stimmen. Die Praxis zeigt jedoch einen entgegengesetzten Trend. In den letzten Jahren ist der Einsatz von sogenannten Business Analysten (wie ich einer bin) immer mehr ein fester Bestandteil eines agilen Entwicklungsteams geworden.

Business Analysten übernehmen in diesen Teams die Aufgaben der fachlichen Klärungen und eine adäquate Übersetzung in Entwicklungsaufgaben. Sie übernehmen damit Aufgaben und Kompetenzen, die theoretisch von dem PO und von dem Entwicklungsteam bereits abgedeckt sein sollten. Ein Business Analyst hat daher theoretisch keine Daseinsberechtigung, da diese Aufgaben bereits von den beteiligten Personen abgedeckt sein sollten.

Dies ist in der Praxis offensichtlich nicht der Fall. Business Analysten haben gerade Hochkonjunktur und werden sukzessiv in immer mehr Entwicklungsteams eingesetzt. Dies liegt daran, dass den eingesetzten POs und Entwicklern nicht unbedingt die richtigen Skillsets mitgegeben werden, um ihre Aufgaben im Sinne des agilen Mindsets zu erfüllen. Adäquates Requirements Engineering ist eine dieser benötigten Fähigkeiten.

Was ist ein Haus?

Die Aufgabenbereiche des Requirements Engineering umfassen dabei folgende Tätigkeiten:

- Ermittlung von Anforderungen
- Analyse und Spezifikation von Anforderungen
- Validierung von Anforderungen
- Management von Anforderungen

Die Qualität des Requirements Engineering leitet sich dabei aus dem größten Engpass der jeweiligen Punkte. Das bedeutet, man kann ein Meister in der Ermittlung von Anforderungen sein. Wenn die Fähigkeit eingeschränkt ist, diese Anforderungen nach ihren ihrer Legitimität und Wertigkeit zu validieren, überflutet man die Entwicklung mit vielen „identifizierten" Anforderungen, wodurch die schiere Masse an neuen Anforderungen die Entwicklung in ihren Entscheidungen überfordern kann.

Die Fähigkeit des *Requirements Engineering* ist nur so gut, wie die Fähigkeit mit der geringsten Ausprägung. In diesem Text wurden vier Tätigkeiten beschrieben, die gewisse Fähigkeiten für eine gelungene Anwendung bedürfen. Es bietet sich daher an, die Fähigkeit weiterzubilden, welche aktuell der Flaschenhals ist.

Was erwarte ich von einem Haus?

Der erste Schritt beim Requirements Engineering ist dabei das Erfragen der Bedürfnisse des Kunden. Dabei ist die Art der Fragestellung wichtig. Der Kunde antwortet im besten Falle nur auf die Frage, welche man gestellt hat. Formuliert man eine Frage zu allgemein, antwortet der Kunde nicht konkret auf das gewünschte Szenario. Stellt man eine Detailfrage, erhält man keine Antwort über die große Vision.

Finde eine Sprache, die auch der Kunde versteht. Halte die Fragen einfach und überschaubar. Stelle nur eine Frage gleichzeitig. Diese Aussagen wirken vielleicht trivial. Beobachte dich mal selbst in Gesprächen mit dem Kunden. Hältst du dich wirklich an alle Kriterien in dem Gespräch? Warum laufen manche Gespräche gut? Warum laufen manche Gespräche schlecht? Diese Regeln gelten nicht nur für die Ermittlung von Anforderungen. Dies sind generelle Regeln der zwischenmenschlichen Kommunikation.

Wichtig ist auch, einen Raum zu schaffen, in dem sich der Kunde in Bezug auf Anforderungen wohl und sicher fühlt. Dies beinhaltet sowohl die Räumlichkeit des Treffens, wie beispielsweise das Format und das Medium, als auch den emotionalen Raum. Fühlen sich die Kunden in einer sicheren Umgebung, in der sie beispielsweise Ideen frei äußern können, ohne dafür verurteilt oder mit Häme belohnt zu werden? Die Kunden müssen sich sicher fühlen. Wenn den Kunden dieser Raum nicht eingeräumt wird, antworten sie eher (firmen-)politisch korrekt, aber nicht ehrlich, oder sie thematisieren Randthemen, statt die relevanten und wichtigen Themen zu adressieren.

Unterscheide in dem Gespräch zwischen Bedürfnissen und Lösungen. Oft haben sich Kunden vorab Gedanken gemacht, welche Lösung ihr Problem löst. Sie kommunizieren daraufhin die Lösung, nicht das Problem. Achte mal selbst darauf. Es ist gefährlich, die Lösung des Kunden als Anforderung aufzunehmen, da diese vielleicht nicht das Problem (für alle) löst. Die Lösung einer Aufgabe liegt in der Hand der Entwicklung, nicht des Kunden. Vielleicht gibt es deutlich bessere Lösungen (fachlich wie technisch). Wenn man Lösungen in den Anforderungen verankert, nimmt man sich die Innovationskraft und lädt sich zusätzliche Komplexität auf.

Wie sehen die Pläne aus?

Die nächste Aufgabe besteht in der Analyse der einzelnen Anforderungen. Es ist wichtig zu wissen, was sich der Kunde wünscht. Ebenso wichtig ist es festzustellen, ob diese Anforderungen und Wünsche zum zu entwickelnden Produkt passen. Wenn der Kunde in einem geschlossenen Raum schwimmen möchte, muss man nicht zwingend ein Hallenbad an ein Haus bauen. Es könnte vielleicht ausreichen, darauf hinzuweisen, dass es in der nächsten Stadt ein öffentliches Hallenbad gibt.

Ein Softwareprodukt kann nicht alle Probleme des Kunden lösen. In den meisten Fällen reicht es aus, wenn die Software ein Problem löst. Daher ist es nach dem Gespräch mit dem Kunden wichtig, dass der Input des Kunden mit dem Gesamtkonzept des Produkts übereinstimmt. Wenn ein geschildertes Problem zu weit von der Produktidee entfernt ist, sollte man die Idee verwerfen (und dies dem Kunden kommunizieren).

Es ist wichtig, Ursache und Wirkung des geschilderten Problems zu verstehen. Das Problem muss verstanden werden, damit die Problembeschreibung in Form einer Story oder Spezifikation an die Entwicklung weitergegeben werden kann. Wenn das Problem nicht verstanden ist, erzeugt dies Unsicherheiten in der Entwicklung. Unsicherheiten führen zu ungenauer Arbeit. Ungenaue Arbeit erhöht das Risiko, dass das eigentliche Problem nicht gelöst wird. Wenn der Kunde sich die Mühe macht, seine Probleme zu erklären, erwartet er eine Lösung dafür. Erhält er daraufhin eine ungenaue Lösung, führt das zu berechtigter Enttäuschung. Es ist unerlässlich, dass das Problem und die Implikationen verstanden sind.

Das Format, in dem die Anforderungen dargestellt werden, kann frei gewählt werden. Wichtig ist, dass sowohl man selbst, der PO, als auch die Entwickler dieses Format verstehen. Eine konkrete Gestaltung und Anpassung des Formats ändert sich meist während der Entwicklung, da sich die Bedürfnisse der Entwickler im Laufe der Zeit ändern. Die Aufgabe des Requirements Engineering ist die Verschriftlichung und Übersetzung der Kundenwünsche für die Entwicklung. Dies erfordert ein gewisses Maß am Experimentieren, um eine geeignete Darstellung der Anforderungen für die Entwicklung, POs und Kunden zu finden.

Sind das wirklich Anforderungen an ein Haus?

Wir haben vom Kunden verstanden, was er sich wünscht und welche Probleme er hat. Allerdings können wir nicht alle Kundenwünsche ungefiltert in die Entwicklung einfließen lassen. Ansonsten besteht die Gefahr, dass das Produkt aufgebläht wird und seinen Fokus auf den tatsächlichen Mehrwert verliert. Daher ist es wichtig, die ermittelten Anforderungen hinsichtlich ihrer Wertigkeit für das Produkt zu analysieren.

Es ist durchaus möglich, dass einige Anforderungen das Produkt in eine ganz andere Richtung lenken. Um zu bestimmen, ob dies eine positive oder negative Veränderung darstellt, muss eine Kosten-Nutzen-Analyse durchgeführt werden. Dabei beschreibt der Nutzen den Mehrwert, den die Anforderung für das Produkt hat. Nach Evidence-Based Management (EBM), einem Konzept zur Messbarkeit von Mehrwert von scrum.org, gibt es vier Nutzen-Kategorien, die eine Anforderung erfüllen sollte:

Current Value: Einen Mehrwert für den aktuellen Nutzenden des Produkts
Unrealized Value: Erhöhung des Nutzenpotenzials durch Erschließung neuer Kundengruppen
Ability to Innovate: Ermöglichung neuer technologischer oder geschäftlicher Möglichkeiten durch einen flexibleren Unterbau/Geschäftsmodell
Time to Market: Beschleunigung der Entwicklung/des Geschäftsmodells

Trägt die neue Anforderung zu einem oder mehreren dieser Ziele bei? Wenn nicht, sollte die Anforderung, unabhängig von den entstehenden Kosten, abgelehnt werden.

Neben dem Nutzen müssen auch die zu erbringenden Aufwände betrachtet werden. Die einzige sinnvolle Interpretation besteht darin, den Nutzen immer im Kontext der entstehenden Aufwände zu sehen. In dieser Phase reicht eine grobe Schätzung aus. Wenn bereits bei der Grobschätzung klar wird, dass die Anforderung einen enormen Umbau erfordert und dabei nur einen mittleren Nutzen bietet, dann stehen die Aufwände in keinem Verhältnis zum Nutzen. Die Konsequenz wäre die Ablehnung des Antrags oder eine entsprechend niedrige Priorisierung.

Wer macht die Bauaufsicht?

In der letzten Phase haben wir uns aufgrund einer Analyse dazu entschieden, die Anforderung umzusetzen. Es genügt jedoch nicht, diese wahllos der Entwicklung zu überlassen. Bei einem Produkt kommen viele verschiedene Anforderungen zusammen, die es „wert" sind, umgesetzt zu werden. Aber welche genau sind das? Und in welcher Reihenfolge sollten sie umgesetzt werden? Wir benötigen einen zentralen Ort, an dem die Anforderungen gesammelt werden. Wenn eine Anforderung für das Projekt relevant ist, dann findet man sie dort. Es ist wichtig, dass es nur einen Ort für die Anforderungen gibt. Dieser stellt die Single-Source-Of-Truth dar. Wenn man sich nicht darauf verlassen kann, dass alle Anforderungen an einem Ort gesammelt sind, entsteht Unsicherheit bei allen Beteiligten.

Von Zeit zu Zeit sollten die einzelnen Anforderungen in dieser Sammlung überprüft und ihre aktuelle Relevanz überprüft werden. Aufgrund von Veränderungen in der Umgebung oder durch die Umsetzung von Anforderungen kann sich die Relevanz der noch nicht umgesetzten Anforderungen ändern. Wenn diese weiterhin relevant sind, sollten sie in der Sammlung verbleiben. Wenn nicht, sollten sie entfernt werden. Es entstehen Risiken, wenn die Anforderungssammlung nicht aktuell ist. Veraltete Anforderungen können die Sammlung verstopfen und bergen das Risiko umgesetzt zu werden. Wenn veraltete Anforderungen gefunden werden, sollten sie sofort aus der Sammlung entfernt werden.

Nun haben wir eine Sammlung von Anforderungen, die wir umsetzen wollen und von denen wir wissen, dass sie zum aktuellen Zeitpunkt relevant sind. Doch welche konkrete Anforderung sollte als nächstes entwickelt werden? Dafür ist eine Priorisierung der Anforderungen notwendig. Die Priorisierung ist ein eigenes Thema. In diesem Text wird nur kurz darauf eingegangen. Vereinfacht gesagt, entsteht die Priorisierung durch das Zusammenspiel von Aufwand/Nutzen (EBM) und den Kundenwünschen.

Das Fundament der Softwareentwicklung

Viele Probleme in der Entwicklung können bereits bei der Erhebung und Verwaltung der Anforderungen vermieden werden. Es erfordert viel Aufwand und Konsequenz, die Anforderungen sauber und relevant zu halten, aber es ist es wert.

Jede beteiligte Partei fühlt sich respektiert und abgeholt, wenn wir diesen Schritt des konsequenten Requirements Engineerings gehen. Unsere Aufgaben werden relevanter und unsere Kunden zufriedener. Auch die Softwareentwicklung wird entlastet, die oft unter Unsicherheit und nicht wertschöpfender Entwicklung leidet. Requirements Engineering ist nur ein Teilaspekt im Gesamtkontext einer erfolgreichen Produktentwicklung. Es stellt jedoch ein solides Fundament dar. Es ist unsere Aufgabe dieses Fundament zu stärken, denn wir alle profitieren davon.

Stephan Tönnies
stephan.toennies@gmail.com

Stephan arbeitet als Business Analyst bei Capgemini vordergründig mit der Anforderungserhebung. Gemeinsam mit ihnen spezifiziert und entwickelt er in einer agilen Entwicklung die Produkte weiter. Als Speaker und in der DOAG als Themenverantwortlicher für Digitale Transformation teilt er sein Wissen mit der Community.


WORK HARDER

Jobs to be Done – voller Fokus auf den Job

Benjamin Klatt, viadee

Das Leben ist zu kurz, um etwas zu entwickeln, dass niemand haben möchte… Dieser Satz gilt sowohl im Konsumenten- als auch im Unternehmensbereich. Nicht ohne Grund scheitern auch 99% der Startups – weil sie etwas bauen, das niemand will [1]. „Jobs to be Done" fokussiert sich daher auf Kunden und ihre Jobs, die sie erledigen möchten. Der Ansatz liefert Methoden, das zugrunde liegende "„Warum"" der Kunden zu verstehen. Was treibt sie an, den Job zu erledigen? Welche Ziele verfolgen sie? Dabei werden sowohl funktionale als auch emotionale und soziale Bedürfnisse berücksichtigt. Für den Einsatz im eigenen Alltag werfen wir einen pragmatischen Blick auf einige Konzepte und Methoden, um diese direkt anwenden zu können.

Was ist Jobs to be Done?

Jobs to be Done, oder auch JTBD ist eine Methode zur radikalen Kundenfokussierung *[2]*. Sie wurde bereits 1999 von Anthony W. Ulwick und Clayton M. Christensen entwickelt. Durch die wachsende Bedeutung von User Experience und Innovationen hat sie in den letzten Jahren nochmal einen deutlichen Schub erlebt. Die grundlegende Idee von JTBD: Kunden möchten einen Job erledigen und engagieren ein Produkt oder eine Dienstleistung, wenn sie ihnen helfen, ihr Ziel zu erreichen. Man legt den Fokus also auf den Job und betrachtet Kunden und Kontext immer im Bezug zu diesem. JTBD liefert dazu Methoden und Strukturen, um die nicht immer offensichtlichen Jobs sowie zugrundeliegende Bedürfnisse und den Kontext der Kunden zu verstehen, beziehungsweise zu analysieren.

Was ist ein Job?

Typischerweise hat man schnell einen Job im Kopf. Beispielsweise möchte man mit Freunden einen gemütlichen Abend verbringen. Eine Möglichkeit dazu ist ein leckeres Essen zu kochen und in dem Zuge dessen muss man unter anderem einen Topf auf den Herd stellen. Wir haben also einen sogenannten Main Job („Einen Abend verbringen"), der unser eigentliches Ziel beinhaltet. Dazu kommt ein **Related Job** („Essen kochen"), der mit dem **Main Job** eng verbunden ist, aber auch für sich allein oder in einem anderen Kontext stehen könnte. Und als drittes haben wir noch einen **Job-Teilschritt** oder Job-Step („Topf bereitstellen"), der für sich allein wenig Mehrwert liefert, aber einen Schritt auf dem Weg zur Job-Erfüllung darstellt.

Wir suchen den Main Job, da dieser das eigentliche Ziel unserer Kunden ist. Den Main Job zu lösen, stellt den größten Mehrwert für sie dar. Um ihn klar herausarbeiten zu können, ist es hilfreich auch die Related Jobs, die manchmal auch alternative Realisierungen des Main Jobs, sind, sauber abzugrenzen. Im Produktdesign lässt sich dann auch überlegen, mit welchen Related Jobs man gegebenenfalls konkurriert oder, welche man gegebenenfalls sogar unterstützen könnte. Statt zu kochen, könnte man auch einfach essen Essen bestellen.

Zwischen Weltfrieden und Arbeitsschritt

Theodore Levitt hat mal schön gesagt: „Menschen wollen keinen Bohrer, sie wollen ein Loch in der Wand". Hier würden wir uns aus der JTBD-Perspektive direkt die Frage stellen, ob Menschen wirklich ein Loch in der Wand wollen. Ein Loch an sich wird in der Regel noch keinen großen Mehrwert bringen. Ein Bohrer-Hersteller wird jedoch genau den „Job" des Lochbohrens adressieren. Aus diesem Grund machen wir uns auf die Suche nach dem richtigen Job-Level, das wir adressieren möchten. Hierzu

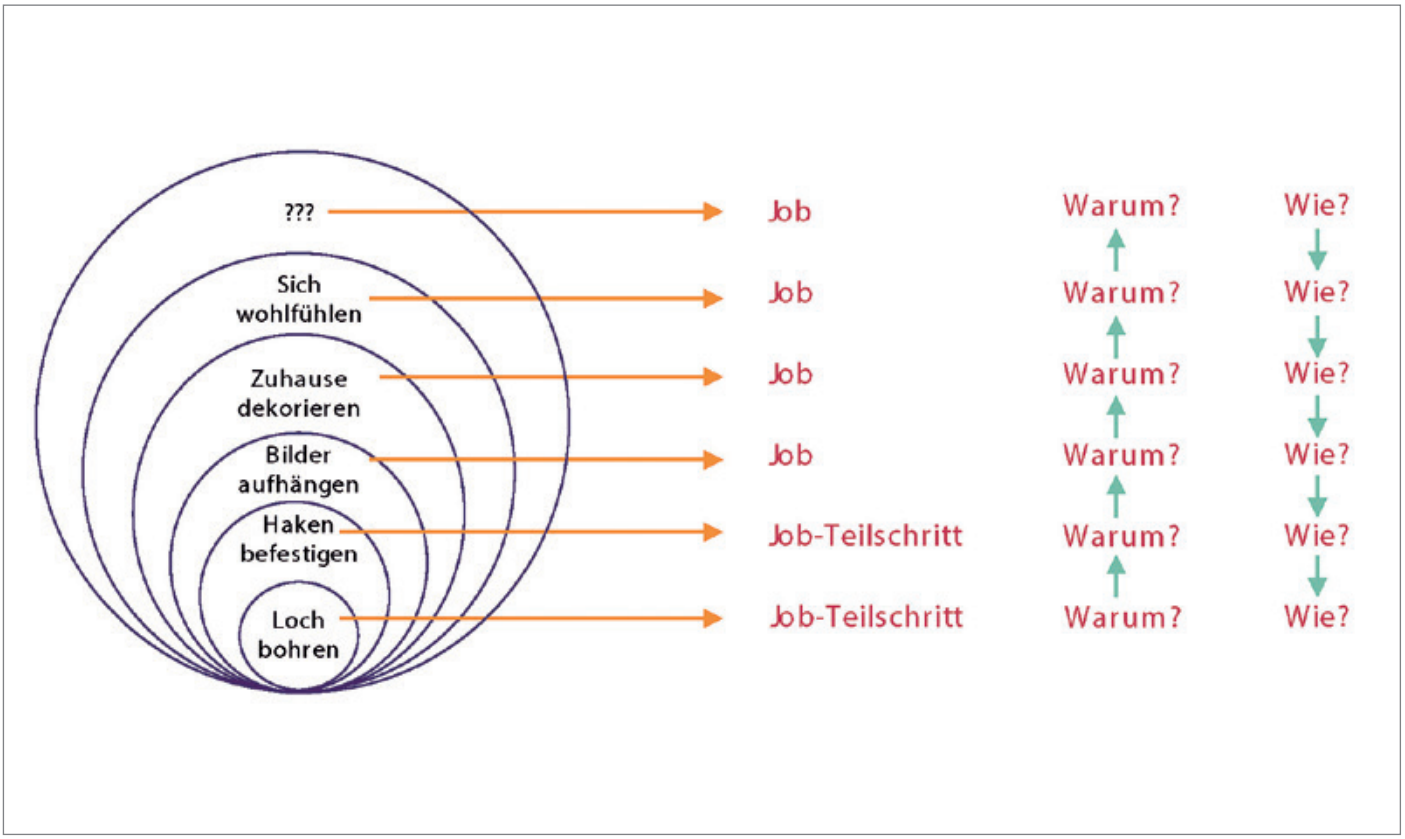


Abbildung 1: Job Level (@viadee angelehnt an Ash Maurya Running Lean [3])

hilft uns das in *Abbildung 1* gezeigte Modell der Job Level. So können wir beispielsweise mit Warum-Fragen nach dem Grund für ein Job fragen. In dem Beispiel aus *Abbildung 1* finden wir heraus, dass das Loch für einen Haken gedacht ist, mit dem wir Bilder aufhängen wollen und die wiederum zur Dekoration unseres Zuhauses dienen sollen. Sind wir zu generisch unterwegs, helfen uns Wie-Fragen, konkretere Jobs zu identifizieren. Am Ende haben Powerstrips einen großen Marktanteil, wenn es um Bildbefestigungen geht, da Menschen in der Regel auch kein Loch in der Wand möchten.

Jobs sind stabil

Das spannende an Jobs ist, dass sie recht stabil gegenüber zeitlicher Veränderung sind. Das, was sich ändert sind die Produkte und Dienstleistungen, mit denen sie gelöst werden. Nehmen wir als Beispiel unseren Job Essen für einen gemütlichen Abend zuzubereiten. Während wir in der Steinzeit auf die Jagd gehen mussten, haben wir im letzten Jahrhundert eingekauft und gekocht. Daraufhin folgten Pizzabringdienste, bis es zu einer Trendwende kam und wieder Lebensmittel geliefert wurden bis hin zu Diensten wie Hello Fresh, die Rezepte mit passenden Zutaten zur Haustür bringen. Seit einigen Jahren ist bereits 3D Food Printing *[4]* möglich, wobei sich zeigen muss, ob dies die Bedürfnisse für eine ausreichend große Kundenmenge erfüllt. Die Quintessenz ist, dass der eigentliche Job gleichbleibt. Nur wie und mit welchen Gütekriterien er gelöst wird, verändert sich. Letztere greifen wir daher später wieder auf.

Wie beschreibt man Jobs?

Nachdem wir Main und Related Jobs kennengelernt haben, stellt sich die Frage, wie sich ein Job beschreiben lässt. Hier bringt das Job- to- be- Done- Framework Strukturen mit, da man in der Regel viele Jobs erfasst, variiert und sich immer weiter dem oder den eigentlichen Jobs nährt. Daher hier die drei Grundelemente zur Job-Beschreibung:

Job Performer sind ganz einfach diejenigen, die einen Job erledigen möchten. Man verwendet bewusst nicht den Begriff „Kunde", da man beispielsweise Anwender und Käufer eines Produktes differenziert. So möchte ein Anwender mit einer Textbearbeitungssoftware Briefe schreiben, während ein Käufer eine Software wirtschaftlich sinnvoll anschaffen möchte, um seine Kollegen zufrieden zu stellen.

Job Statement ist eine kompakte konkrete Benennung eines Jobs. Hier empfiehlt sich eine schematische Formulierung, um mit einer großen Fülle von Statements immernoch den Überblick zu behalten und variieren zu können. Ein Standardschema wäre „Ein Bild an einer Wohnzimmerwand aufhängen". Hier zeigen sich schon zwei Variationsmöglichkeiten und man könnte Bild und Wohnzimmer abändern und sich Fragen, ob es noch der gleiche Job für den gleichen Job Performer ist: „Ein Regal an einer Badezimmerwand aufhängen". Schon stellt sich die Frage, ob unsere Powerstrips auch hier helfen.

Needs (Bedürfnisse) geben uns einen Einblick, was Job Performern bei der Job-Erfüllung wichtig ist und wie sie beurteilen, ob ein Job gut erledigt wurde. Das Job to be Done JTBD-Framework kennt drei verschiedene Kategorien von Needs, um diese ganzheitlich erfassen zu können:

- Funktionale Bedürfnisse beschreiben ob und wie das Ziel eines Jobs erreicht wurde. Beispielsweise möchte man den gemütlichen Abend mit Freunden schnell oder mit wenig Aufwand vorbereiten können.

Abbildung 2: JTBD Workshop Foto (@viadee)

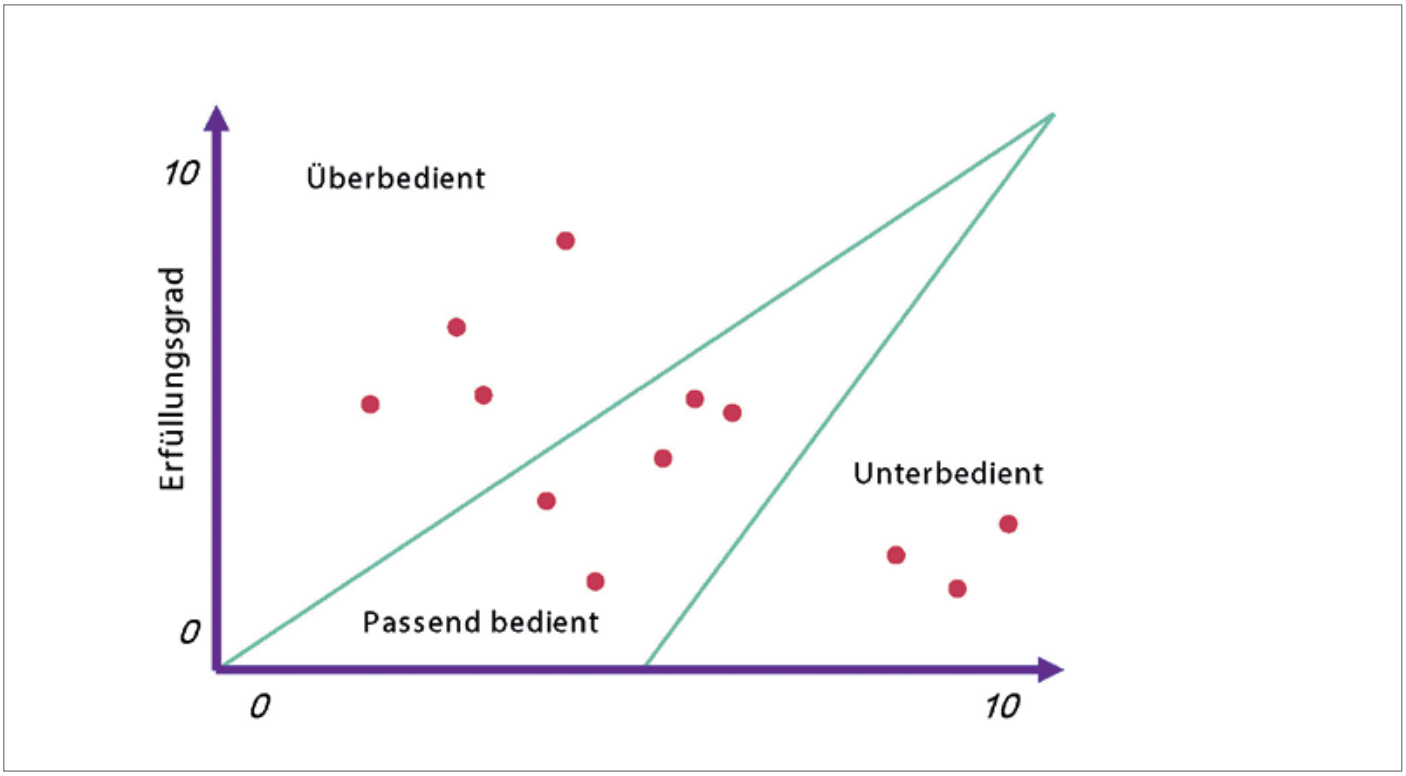


Abbildung 3: Quadrant zur Bedürfnisbedienung (@viadee angelehnt an Ulwick)

- Emotionale Bedürfnisse beschreiben die persönlichen Gefühle bei der Job-Erfüllung. So möchte man sich beispielsweise nicht gehetzt oder gestresst fühlen, wenn man den gemütlichen Abend vorbereitet.
- Soziale Bedürfnisse betrachten die Gefühle im Kontext der Umwelt bei der Job-Erfüllung. So ist es einem möglicherweise wichtig, dass alle Freunde das zubereitete Essen mögen werden.

Mit den drei Elementen Job Performer, Job Statement und Needs sowie den drei Arten von Needs können wir nun sehr gut Jobs identifizieren, beziehungsweise erarbeiten.

Wie erarbeitet man sich Jobs?

Ganz essenziell ist das Gespräch mit Kunden beziehungsweise Job Performern, um herauszufinden, was ihre eigentlichen Jobs sind und, was sie dabei bewegt. Zur Orientierung starten wir daher mit ersten Interviews mit potenziellen Job Performern, sammeln Wissen, identifizieren gegebenenfalls weitere Job Performer und kommen dann mit allen Beteiligten für einen Workshop zusammen. Man kann natürlich auch ohne Interviews starten, um zunächst die eigenen Gedanken zu ordnen.

Das Foto in *Abbildung 2* zeigt die Ergebnisse eines solchen Workshops, in diesem Fall zu unserem Arbeitsplatzbuchungssystem Buuky. Prinzipiell sammelt man in der ersten Spalte die Job Performer, von denen man vermutet, dass sie in irgendeiner Form mit dem eigenen Produkt interagieren. In der nächsten Spalte sammelt man die Jobs je Job Performer als Job Statements. In dem Beispiel ist aufgeschlüsselt, dass sich ein Mitarbeiter einen Arbeitsplatz im Büro bucht, während ein Einkäufer eine Lösung zur Büro-Arbeitsplatzbuchung für die Mitarbeiter beschaffen möchte. In der dritten Spalte sammeln wir die Bedürfnisse je Job. Beispielsweise möchten Mitarbeitende ihren Arbeitsplatz mit möglichst wenig Zeitaufwand buchen. Die Bedürfnisse erhalten Bewertungen, um sie als Gütekriterien nutzen zu können.

Bedürfnisse als Gütekriterien

Die identifizierten Bedürfnisse haben in Abhängigkeit von den Job Performern eine unterschiedliche Relevanz. Hieraus lassen sich Gütekriterien ableiten, mit denen die eigenen und existierende Lösungsalternativen bewertet werden können. Um dies zu tunerfassen, erhalten alle Bedürfnisse zwei Bewertungen von 0 bis 10:

1. Wie wichtig ist Bedürfnis X?
2. Wie gut wird Bedürfnis X mit aktuellen Angeboten bedient?

Hieraus lässt sich ein Quadrant wie in *Abbildung 3* aufzeichnen, in dem die Bedürfnisse mit besonders hohem Potential zu finden sind. Alle Bedürfnisse werden hier entsprechend ihrer Bewertung bezüglich Erfüllungsgrad und Relevanz eingetragen. Die Bedürfnisse mit hoher Relevanz und geringem Erfüllungsgrad sind natürlich spannend. Die derzeit passend bedienten Bedürfnisse stellen erwartete Basisfunktionalität dar. Dagegen sind die überbedienten Bedürfnisse oft nur durch einen günstigeren Preis oder disruptive Innovationen adressierbar.

Die Spitze des Job-Eisbergs

Auch wenn dies nur ein kurzer Flug über die grundlegenden Elemente von Jobs to be Done war, so kann man hiermit schon hilfreiche Erkenntnisse für die eigene Produktgestaltung gewinnen. Das Framework insgesamt geht noch deutlich weiter. Beispielsweise können mit einer Job Map Jobs in ihre Teilschritte und deren Relevanz für die Job Performer strukturiert werden. Wenn der Einstieg Interesse geweckt hat, können wir nur empfehlen nach und nach auch weitere Bestandteile des JTBD-Frameworks zu ergründen. Kennt man die Jobs und relevanten Bedürfnisse, lassen sich hiermit übrigens auch deutlich bessere Pitches für die eigenen Lösungen entwickeln. Gerade die emotionalen und sozialen Bedürfnisse helfen, packende Pitches zu formulieren. In unserer Innovationsarbeit nutzen wir zu dem oft JTBD in Kombination mit einem Lean Canvas. Dank JTBD können wir hier die Kundensegmente und ihre Probleme, sowie die Unique Value Proposition und Features besser identifizieren und benennen.

Zu guter Letzt möchten wir aber noch eine Empfehlung mit auf den Weg geben: Sprechen Sie mit ihren Kunden und Job Performern und führen Sie Interviews. Nur auf diesem Weg können Sie wirklich lernen, was diese Menschen bewegt. Ein Job, den man sich nur auf dem Papier überlegt und nicht prüft, bleibt eine Annahme. In diesem Sinne: Viel Freude auf der Job-Suche und Entdeckungsreise.

Quellen

[1] *https://medium.com/swlh/why-90-of-startups-fail-ea41ed0a2f8c*
[2] *https://jobs-to-be-done.com/outcome-driven-innovation-odi-is-jobs-to-be-done-theory-in-practice-2944c6ebc40e*
[3] Running Lean, Ash Maurya
[4] *https://en.wikipedia.org/wiki/3D_food_printing*

Dr. Benjamin Klatt
benjamin.klatt@viadee.de

Benjamin ist für die viadee als IT-Architekt und Coach im Bereich Organisationsentwicklung unterwegs. Er begleitet Unternehmen bei der Digitalisierung von Produkten und Prozessen, sowie der Erschließung neuer Technologien, Arbeitsweisen und Innovationen. Darüber hinaus arbeitet er an und in der Forschung und Entwicklung der viadee.

APEX connect

by DOAG

on demand

APEX 2023 VERPASST?

Jetzt On-demand-Ticket buchen und
Vortragsaufzeichnungen anschauen!

Alle Angebote im
On-demand-Ticket-Shop

apex.doag.org DOAG

Interview mit Katharina Schraft, Vorstand Business Solutions und Leiterin Business Solutions Community: „Requirements schaffen eine gemeinsame Basis."

Nach drei Jahren legte Dr. Thomas Karle seinen Posten als DOAG Vorstand Business Solutions und Leiter der Business Solutions Community nieder und übergab den Stab an Katharina Schraft. Sie wurde im Februar dieses Jahres von den Delegierten auf der Versammlung im Berliner Hotel Estrel zur Nachfolgerin gewählt. Marcos López, Redaktionsleitung Business News, nutzte die Gelegenheit, um mit Frau Schraft, die Vice President NetSuite bei der PROMATIS software GmbH ist, eine „Standortbestimmung der Business Solutions Community" zu bewerkstelligen, ihre Ideen als neues Mitglied im Vorstand in Erfahrung zu bringen und über ihr Selbstverständnis als Frau in einem von Männern dominierten Business zu reden.

Frau Schraft, Herr Karle trat als Vorstand zurück, die Wahl fiel im Februar dieses Jahres auf Sie. Was verbinden Sie mit diesem Ehrenamt für die DOAG?

Ich freue mich sehr, dass ich den Vorstand überzeugen konnte, mich für diese Position einzusetzen. Die DOAG begleitet mich schon mein ganzes Arbeitsleben – bereits 2012 durfte ich auf der ehemaligen DOAG Applications in München einen Vortrag über Accounting in Oracle Applications halten, das war dann auch nicht der einzige. Der Bereich Business Solutions hat sehr viel Potential – denn jedes Unternehmen benötigt sie. Meiner Meinung nach fügen die Business Solutions die anderen Bereiche zusammen und bringen am Ende die Anwender in Kontakt mit der IT.

Woraus besteht zukünftig Ihre Arbeit beziehungsweise wie sieht Ihre Arbeit in dieser Position konkret aus? Geht es mehr ums „Verwalten" von etablierten Standards oder ums „kreative Gestalten"?

Es geht vor allem um den Ausbau von Dingen, die sehr gut funktionieren, wie zum Beispiel die Arbeit der Kollegen rund um NetSuite mit den European NetSuite User Days, die im Herbst zum zweiten Mal stattfinden werden, wie auch um das Erschließen neuer Bereiche in denen Bedarf und Potenzial vorliegt. Es ist also eher ein „kreatives Gestalten" – ich möchte aber darauf achten, dass die Community die Chance hat, ergonomisch zu wachsen. Es ist eine große Verpflichtung, Aktive zu finden und diese dann aber auch zu „bespielen".

Die Business Solutions Community wurde vor nunmehr 13 Jahren gegründet. Haben Sie schon eine Idee, welche Akzente Sie setzen wollen?

Vor allem steht akut eine kleine Umstrukturierung an, um die gewachsene Struktur etwas zurechtzurücken und dann im Anschluss stabil ausbauen zu können. Die Fokusthemen der Community und ihre Positionierung innerhalb der DOAG sind formuliert, nun ist der Ausbau dran. Zentral ist weiterhin die digitale Transformation wie auch das Thema Cloud zu betrachten – das ist nichts, was groß überrascht. Es bringt aber, neben der Auswahl eines ERP-Systems, auch weitere Fragestellungen wie Datenschutz, Compliance oder die optionale Systemarchitektur mit sich. Soweit zur groben Ausrichtung. Zur Umsetzung und der Arbeit in der Community ist mir die Vernetzung sehr wichtig. Mit Kunden, Partnern und auch weiteren Communities – innerhalb wie auch außerhalb der DOAG. Das wird ein bisschen Zeit brauchen – Netzwerke müssen aufgebaut und am Leben gehalten werden. Die Aktiven wollen sich gerne einbringen – je nach den individuellen Möglichkeiten.

Inwieweit werden Sie Ihre Netsuite-Expertise einbringen? Sie sind Vizepräsidentin dieser Business Unit bei der PROMATIS software GmbH.

In den letzten Jahren konnte ich vielfältige Expertise in der ganzen Breite von Business Applications sammeln. Angefangen habe ich als Finance Consultant mit der Oracle E-Business Suite, momentan habe ich im Bereich NetSuite auch Berührungspunkte mit vielen Kunden und Systemen (auch außerhalb der Oracle-Produktfamilie) und begleite Kunden vom Vertrieb bis hin zum Support. Es hilft mir natürlich, auch bei den aktuell „heißen" Themen mitreden zu können, aber wir sind hier in der Community bereits sehr gut aufgestellt. Ich sehe meinen Schwerpunkt also eher in der Vernetzung und dem Ausbau von ERP-Systemen, um eine vernünftige und robuste Infrastruktur zu gewinnen.

Sie sind ebenso Expertin für Compliance-Fragestellungen. Wie hat sich dieses Thema im Lichte von DSGVO, Nachhaltigkeit, Corona und Krieg verändert? Gibt es eine gesteigerte Achtsamkeit für dieses Thema oder drängt zum Beispiel die aktuelle Inflation regelrecht dazu mit Blick auf die Betriebswirtschaftlichkeit und Konkurrenzfähigkeit den Gewinn wieder mehr in den Fokus zu nehmen und nicht die Einhaltung von Gesetzen, Richtlinien und „guten Absichten"?

Die verschiedenen Ansätze gibt es natürlich immer – Fokus auf Gewinn oder eher traditionelle Ansätze. Ich denke, dass die Unternehmen, welche sich eine ERP Lösung im mittleren bis höheren Preissegment anschaffen, sich eine reine Gewinnorientierung ohne die Basics nicht leisten können. Mit der Digitalisierung und dem globalen Einsatz von Systemen ergeben sich gezwungenermaßen neue Compliance-Fragestellungen für Unternehmen. Wer heutzutage zum Beispiel im eCommerce aktiv ist, kann sehr schnell in vielen Ländern außerhalb Deutschlands steuerpflichtig sein. Und jedes Land hat sehr individuelle Anforderungen was Compliance und Berichterstattung betrifft. Zudem gibt es viele neue Anforderungen an Digitalisierung. Ermöglicht wird das auch durch moderne Systeme in der Cloud, welche die „Internationalität" auch in der DNA haben. Zu bedenken ist nur: automatisch geht das alles nicht. Und insbesondere bei neuen Firmen liegt oft auch der Fokus zunächst nicht auf Finanzrahmen und Compliance. Hier sind Vorsicht und ein erfahrener Partner das A und O. Auch hier würde sich die DOAG Business Solutions Community gerne als vertrauenswürdiger Partner anbieten.

Corona scheint vorbei. Die App wurde ganz offiziell abgeschaltet, das heißt sie wird nicht mehr weiterentwickelt, Begegnungsmitteilungen wurden eingestellt und sie soll aus den App Stores verschwinden. Wie sieht Ihr Blick zurück aus? Was haben wir aus der Pandemie gelernt? Was können wir verbessern?

Die Pandemie hat Potenziale in verschiedensten Bereichen aufgezeigt. Neben sehr offensichtlichen Themen wie Digitalisierung denke ich, dass auch diese Phase der Isolation eine starke persönliche Wirkung auf jeden hatte. Konzepte wie Work-Life-Balance, Geschäftsreisen, Homeoffice und Arbeitsplatzattraktivität stehen nun oft in einem anderen Licht da – sowohl von Arbeitgeber – als auch Arbeitnehmerseite. Unternehmen arbeiten an ihren Grundwerten. Zudem war es eine Zeit, in der man gegebenenfalls seine aktuelle Situation überdacht und sich umorientiert hat, vielleicht auch, weil sich manche Arbeitgeber und Geschäftsfelder als sicher herausgestellt haben. Für mich selbst war die Pandemie eine sehr intensive Zeit, in der tatsächlich weiterhin viel zu tun war – auch dank einem Job im stark digitalisierten Umfeld. Ich würde mir wünschen, dass wieder mehr Wert auf bedeutungsvolle Begegnungen gelegt wird im Kontrast zu den Arbeitsta-

gen voller Zoomcalls, die sicher viele von uns erlebt haben.

Während sich die DOAG öffnet und das O für Oracle in der Namensgebung ganz bewusst und offiziell „verschwunden" ist, sind die Business Solutions weiterhin eine Domäne von Oracle. Bleibt Oracle auch weiterhin der entscheidende Treiber dieses Segments? Wie sieht es mit der „Konkurrenz" aus?

Die Zukunft und eigentlich auch schon gelebte Realität, ist eine Architektur, die sich aus „best of breed"-Lösungen von verschiedenen Anbietern zusammensetzt. Das kann zum Beispiel so aussehen, dass eine Oracle Cloud Lösung mit lokalen Payroll-Systemen integriert werden, oder auch eine Oracle HCM-Lösung welche mit SAP-Systemen zusammenarbeitet – da sie nun mal best of breed ist ☺. Im Kern werden wir uns weiterhin mit Oracle-Systemen beschäftigen und die Community auch in diese Richtung weiter ausbauen. Dem offenen Austausch mit nicht-Oracle Gruppen steht das nicht im Wege, ganz im Gegenteil. Es macht zum Beispiel sehr viel Sinn, sich mit anderen Communities zu gemeinsamen Themen zusammenzufinden und voneinander zu lernen.

Das Thema der aktuellen Ausgabe der vorliegenden Business News ist ja Requirements Engineering. Gerade bei Business Solutions ist dies ein zentraler Punkt, da durch die Standardsoftware-Pakete viele Requirements „vorimplementiert" sind und auf der anderen Seite Einschränkungen bezüglich Erweiterbarkeit solcher Pakete zu berücksichtigen sind. Wie können hier Lösungsstrategien für Kunden bei einer Implementierung aussehen? Sind gegebenenfalls unterschiedliche Ansätze je nach verwendeter Oracle Produktfamilie zu empfehlen?

In den Grundzügen muss man sehen, dass es in der Implementierung und dem Betrieb von Systemen wichtig ist, eine Art des Requirements Engineerings zu betreiben. In meinen Projekten habe ich festgestellt, dass die Art, die Detailtiefe und die Durchführung oft nicht vom System abhängt, sondern von der Konstellation Berater – Kunde. Berater bringen Ansätze und im Idealfall auch Materialien mit, diese müssen aber insbesondere vom Kunden angewandt werden.

Warum ist denn Requirements Engineering so wichtig?

Requirements schaffen eine gemeinsame Basis. Sie sind in Summe die knackigere Version eines Lastenheftes, in denen der Kunde ganz klar seine Vorstellungen ausdrückt. Ohne diese Requirements findet man sich meiner Erfahrung nach sehr schnell in verfahrenen Situationen wieder, da man zu einem, oft späten, Zeitpunkt feststellt, dass Wunsch und Realität vielleicht gar nicht auf einen Nenner zu bringen sind. Um das Requirements Engineering effektiver zu machen, wird oft mit Entwürfen, fertigen Paketen oder Modellen gearbeitet, um schneller und effektiver zu möglichst treffsicheren Requirements zu gelangen. Der ideale Prozess ist oft so: Kunde stellt Beschreibung in Vertriebsphase bereit, Anbieter mappt das auf Systemfunktionalität und stellt so dar, dass das System prinzipiell auf einem (gegebenenfalls hohen) Level geeignet ist diese Anforderungen zu erfüllen. Das Projekt beginnt, dann stellt der Partner eine Basis für detaillierte Requirements bereit, zum Beispiel durch Listen oder Abfrage in Fragebögen. Diese können dann in einen Initial-Setup einfließen oder diskutiert werden, wenn diese Wünsche nicht im System umsetzbar sind. Dann, nach der Implementierung, kann der Kunde die Requirements, welche nun in Form von Testfällen vorliegen, nutzen, um das System zu validieren. Zum Ende des Projekts sind dann die Requirements eine gemeinsame Basis für Kunde und Partner, um das Projekt abzuschließen.

Die Gender-Diskussion hat mittlerweile viele Ebenen der gesellschaftlichen Wahrnehmung erreicht, von der Verteilung und Besetzung von Posten über die Unterschiede der Entlohnung bis zu Rente, da Frauen mit Kindern durch Schwangerschaft, Geburt und Stillzeiten Einbußen zu beklagen haben. Auch der Vorstand der DOAG ist mehrheitlich männlich besetzt, sie sind eine von zwei weiblichen Vorständen. Nehmen Sie ihre Rolle als Frau sowohl bei Ihrem Arbeitgeber als auch im DOAG Vorstand als eine besondere wahr?

Vor allem denke ich, dass das zentrale Element die Diversität ist, nicht unbedingt eine Quote bezüglich Frauen. Organisationen wie Vereine leben davon, Mitglieder zu gewinnen. Firmen möchten Bewerber anziehen. Eine diverse Belegschaft sorgt dafür, dass sich auch vielfältigere Mitarbeiter oder Mitglieder angesprochen fühlen. Mit der Diversität kommen dann auch Anforderungen und Herausforderungen auf die Unternehmen zu, für diese dann Lösungen gefunden werden können. Ich fühle mich als Frau bei der DOAG wie auch bei meinem Arbeitgeber sehr wohl, da ich bei beiden nicht das Gefühl habe, aufgrund meines Geschlechts anders behandelt zu werden. Somit hoffe ich, dass das auch sichtbar ist für andere junge Frauen und diese sich bestärkt fühlen den Schritt in die IT, ins Management und auch in Vereine zu gehen und sich zu engagieren.

Frau Schraft, herzlichen Dank für das Interview!

Katharina Schraft
katharina.schraft@doag.org

Katharina Schraft ist Vorstand der Business Solutions der DOAG. Sie ist seit 2010 bei der PROMATIS Unternehmensgruppe am Hauptsitz in Ettlingen (TechnologieRegion Karlsruhe) tätig und hat ihren Schwerpunkt in der ERP-Beratung mit Fokus auf Finance. Seit 2018 ist sie Vice President und Leiterin der Business Unit NetSuite bei PROMATIS und berät Kunden unter anderem in strategischen Buchhaltungs- und Compliance-Fragestellungen.

Quadratur des Kreises – Attribute Clustering, ein weithin unterschätztes Feature der Oracle-Datenbank – Teil 3

Randolf Eberle-Geist, Unabhängiger Berater

Im zweiten Teil des Artikels wurden die grundlegenden Features des „Attribute Clusterings“ gezeigt und wie damit bestimmte Zugriffswege auf Tabellendaten effizienter gestaltet werden können. Insbesondere ermöglicht es, das „Attribute Clustering“ im Rahmen bestimmter Grenzen nicht nur einen Zugriffsweg zu optimieren, sondern die Daten so abzulegen, dass unterschiedliche Zugriffswege effizienter werden können („Interleaved Ordering“) – ein Alleinstellungsmerkmal dieses Features. Im letzten Teil des Artikels wird nun auf eine weitere Möglichkeit des „Attribute Clusterings“ eingegangen – die Optimierung von Joins, also der Zugriff auf eine Tabelle über eine andere Tabelle – im zweiten Teil wurde am Ende gezeigt, wie der direkte Zugriff per Fremdschlüssel auf eine Faktentabelle effizienter gestaltet werden konnte.

„Join Attribute Clustering“

Nun verwenden reale Abfragen in einem typischen „Star Schema“ selten Filter direkt auf den Fremdschlüsseln der Faktentabelle, sondern filtern meistens auf den Dimensionen. Hier würden also Joins zwischen Dimensions- und Faktentabelle stattfinden und entsprechend auf den Dimensionen die Filter angewendet werden.

Selbst dieses Szenario kann mittels „Attribute Clustering“ optimiert werden und dies wiederum sogar für mehrere Zugriffswege unabhängig voneinander mittels der Kombination von „Join Attribute Clustering“ und „Interleaved Order“.

Das heißt, es ist sogar möglich, Joins zu optimieren, also den Zugriff auf eine Tabelle per Join von einer anderen Tabelle durch „Attribute Clustering“ zu optimieren. Dies ist der Funktionalität von Oracle-Clustern ähnlich, in dem Sinne, dass die Daten so abgelegt werden, dass Joins zwischen den Tabellen möglichst effizient durchgeführt werden können. Nur werden hier nicht die Daten von unterschiedlichen Tabellen mit dem gleichen Cluster-Key-Wert in den gleichen Blöcken abgelegt (Oracle Table Cluster mit mehreren Tabellen), sondern nur entsprechend innerhalb einer „Heap“-Tabelle organisiert.

Als Voraussetzung dafür müssen zumindest Unique Indizes auf den per Join verknüpften Tabellen existieren, die garantieren, dass der Join keine Duplikate erzeugt – idealerweise geht Oracle von einem Star- oder Snowflake-Schema mit einer Fakten- und mehreren Dimensionstabellen aus. Das Feature ist also auch an dieser Stelle an Data-Warehouse-Umgebungen angelehnt.

Wie würde das nun in dem konkreten Beispiel aussehen können? Anstatt das „Attribute Clustering“ direkt auf den Fremdschlüsseln zu definieren, wird nun ein entsprechend zu optimierender „Join“ im Clustering festgelegt *(siehe Listing 1)*.

Die Definition des „Attribute Clusterings” wurde also wieder auf der Tabelle SALES_AC entsprechend geändert – es wird nun zu den Dimensionstabellen LOCATIONS und PRODUCTS jeweils ein Join definiert und der Zugriff auf Filtern auf den folgenden Attributen der Dimensionstabellen optimiert:

(locations.state, locations.county)
products.product_name

Und zwar unabhängig voneinander – es werden jetzt also Abfragen optimiert, die die SALES_AC-Tabelle mit der LOCATIONS-Tabelle per Join auf der LOCATION_ID verknüpfen und auf der LOCATIONS-Dimension auf STATE oder STATE und COUNTY filtern, als auch Abfragen, die die SALES_AC-Tabelle mit der PRODUCTS-Tabelle per Join auf der PRODUCT_ID verknüpfen und auf der PRODUCTS- Dimension auf PRODUCT_NAME filtern.

Das Beispiel beinhaltet der Vollständigkeit halber auch die Definition einer sogenannten „Zonemap“ – dies ist allerdings nur auf Exadata-basierten Umgebungen („Exadata on premise“, „Exadata Cloud at Customer“, „Autonomous Database on Exadata“) aus lizenztechnischen Gründen unterstützt, was sehr schade ist, da gerade auf Nicht-Exadata-Umgebungen dieses Feature sehr nützlich wäre. Damit könnten dann unter anderem auch Full-Table-Scan-Zugriffe mittels „Attribute Clustering“ optimiert werden, indem die Datenbank dabei nur auf die Blöcke der Tabelle zugreift, in denen laut „Zonemap“-Information auch Zeilen stehen, die die gesuchten Werte beinhalten – der Rest kann einfach übersprungen werden. Effizient wird das dann, wenn die Daten eben gemäß des Suchkriteriums zusammenhängend in der Tabelle mittels „Attribute Clusterings“ organisiert sind.

Zurück zu den Abfragen mit optimierten Joins. Zuerst werden wieder passende Indizes auf der Faktentabelle erzeugt *(siehe Listing 2)*.

Nun können die Beispiel-Abfragen ausgeführt werden und zwar wieder auf

```
SQL> REM Drop the current attribute clustering definition
SQL>
SQL> ALTER TABLE sales_ac DROP CLUSTERING
  2  /

Table altered.

SQL> TRUNCATE TABLE sales_ac
  2  /

Table truncated.

SQL> --
SQL> REM Enable interleaved join attribute clustering
SQL> REM on SALES_AC table.
SQL> REM For the sake of example, create
SQL> REM the zone map manually.
SQL> --
SQL> ALTER TABLE sales_ac
  2  ADD CLUSTERING sales_ac
  3  JOIN locations ON (sales_ac.location_id = locations.location_id)
  4  JOIN products  ON (sales_ac.product_id = products.product_id)
  5  BY INTERLEAVED ORDER ((locations.state, locations.county),products.product_name)
  6  --BY LINEAR ORDER (locations.state, locations.county)
  7  WITHOUT MATERIALIZED ZONEMAP
```

```
  8  /

Table altered.

SQL> --
SQL> REM Manually create the zone map
SQL> --
SQL>
SQL> CREATE MATERIALIZED ZONEMAP sales_ac_zmap
  2  AS
  3  SELECT SYS_OP_ZONE_ID(s.rowid),
  4         MIN(l.state) min_state,
  5         MAX(l.state) max_state,
  6         MIN(l.county) min_county,
  7         MAX(l.county) max_county,
  8         MIN(p.product_name) min_prod,
  9         MAX(p.product_name) max_prod
 10  FROM sales_ac s,
 11       locations l,
 12       products p
 13  WHERE s.location_id = l.location_id(+)
 14  AND   s.product_id = p.product_id(+)
 15  GROUP BY SYS_OP_ZONE_ID(s.rowid)
 16  /
GROUP BY SYS_OP_ZONE_ID(s.rowid)
                               *
ERROR at line 15:
ORA-31969: ZONEMAP not supported for table stored in tablespace of this storage type

SQL> --
SQL> REM Insert data and observe that a
SQL> REM sorts and joins are performed to cluster
SQL> REM data in the SALES_AC table.
SQL> REM The direct path insert operation will maintain
SQL> REM the zone map for us.
SQL> --
SQL> INSERT /*+ APPEND */ INTO sales_ac SELECT * FROM sales_source
  2  /

1952104 rows created.

SQL> SELECT * FROM TABLE(dbms_xplan.display_cursor)
 2  /

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------
SQL_ID  8wzqwqgwwynya, child number 0
-------------------------------------
INSERT /*+ APPEND */ INTO sales_ac SELECT * FROM sales_source

Plan hash value: 2808338825

-----------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------
|   0 | INSERT STATEMENT                   |              |       |       |       | 47632 (100)|          |
|   1 |  LOAD AS SELECT                    | SALES_AC     |       |       |       |            |          |
|   2 |   OPTIMIZER STATISTICS GATHERING   |              |  1952K|   180M|       | 47632   (1)| 00:00:02 |
|   3 |    SORT ORDER BY                   |              |  1952K|   180M|   203M| 47632   (1)| 00:00:02 |
|*  4 |     HASH JOIN RIGHT OUTER          |              |  1952K|   180M|       |  4408   (2)| 00:00:01 |
|   5 |      TABLE ACCESS FULL             | LOCATIONS    |  3143 | 94290 |       |    11   (0)| 00:00:01 |
|*  6 |      HASH JOIN RIGHT OUTER         |              |  1952K|   124M|       |  4388   (1)| 00:00:01 |
|   7 |       TABLE ACCESS FULL            | PRODUCTS     |    28 |   364 |       |     2   (0)| 00:00:01 |
|   8 |       TABLE ACCESS FULL            | SALES_SOURCE |  1952K|   100M|       |  4378   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------
```

```
Predicate Information (identified by operation id):
---------------------------------------------------

   4 - access("SALES_SOURCE"."LOCATION_ID"="LOCATIONS"."LOCATION_ID")
   6 - access("SALES_SOURCE"."PRODUCT_ID"="PRODUCTS"."PRODUCT_ID")


26 rows selected.

SQL> COMMIT
  2  /

Commit complete.

SQL> EXECUTE dbms_stats.gather_table_stats(ownname=>NULL,tabname=>'sales_ac');

PL/SQL procedure successfully completed.
```

Listing 1: Definition des „Join Attribute Clusterings" auf der Faktentabelle und Neubefüllung

```
SQL> REM The full potential of attribute clusters are realised
SQL> REM when used in conjunction with zone maps, Exadata storage indexes
SQL> REM and In-Memory min/max pruning. However, they also improve
SQL> REM index clustering. This is demonstrated here.
SQL>
SQL> REM Create indexes on location id for the standard SALES
SQL> REM table and the attribute clustered SALES_AC table
SQL>
SQL> CREATE INDEX sales_loc_i ON sales (location_id)
  2  /

Index created.

SQL> CREATE INDEX sales_ac_loc_i ON sales_ac (location_id)
  2  /

Index created.

SQL> CREATE INDEX sales_prod_i ON sales (product_id)
  2  /

Index created.

SQL> CREATE INDEX sales_ac_prod_i ON sales_ac (product_id)
  2  /

Index created.
```

Listing 2: Erzeugen der passenden Indizes für die Beispielabfragen

```
SQL> SET AUTOTRACE ON
SQL>
SQL> REM Conventional
SQL>
SQL> set termout off
SQL>
SQL> SELECT SUM(amount)
  2  FROM   sales
  3  JOIN   locations  ON (sales.location_id = locations.location_id)
  4  WHERE  locations.state  = 'California'
  5  AND    locations.county = 'Alpine County'
  6  /
```

```
SUM(AMOUNT)
-----------
  643823,06

1 row selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 1895572434

-------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name        | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |             |     1 |    39 |   623   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE               |             |     1 |    39 |            |          |
|   2 |   NESTED LOOPS                |             |    17 |   663 |   623   (0)| 00:00:01 |
|   3 |    NESTED LOOPS               |             |   621 |   663 |   623   (0)| 00:00:01 |
|*  4 |     TABLE ACCESS FULL         | LOCATIONS   |     1 |    30 |    11   (0)| 00:00:01 |
|*  5 |     INDEX RANGE SCAN          | SALES_LOC_I |   621 |       |     2   (0)| 00:00:01 |
|   6 |    TABLE ACCESS BY INDEX ROWID| SALES       |   621 |  5589 |   612   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------

   4 - filter("LOCATIONS"."COUNTY"='Alpine County' AND
              "LOCATIONS"."STATE"='California')
   5 - access("SALES"."LOCATION_ID"="LOCATIONS"."LOCATION_ID")

Note
-----
   - this is an adaptive plan


Statistics
----------------------------------------------------------
          0  recursive calls
          0  db block gets
        655  consistent gets
          0  physical reads
          0  redo size
        352  bytes sent via SQL*Net to client
        372  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed

SQL> REM With attribute clustering
SQL>
SQL> set termout off
SQL>
SQL> SELECT SUM(amount)
  2  FROM   sales_ac
  3  JOIN   locations  ON (sales_ac.location_id = locations.location_id)
  4  WHERE  locations.state  = 'California'
  5  AND    locations.county = 'Alpine County'
  6  /

SUM(AMOUNT)
-----------
  643823,06

1 row selected.


Execution Plan
----------------------------------------------------------
```

```
Plan hash value: 918547280

-------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                |     1 |    39 |    46   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE               |                |     1 |    39 |            |          |
|   2 |   NESTED LOOPS                |                |    17 |   663 |    46   (0)| 00:00:01 |
|   3 |    NESTED LOOPS               |                |   621 |   663 |    46   (0)| 00:00:01 |
|*  4 |     TABLE ACCESS FULL         | LOCATIONS      |     1 |    30 |    11   (0)| 00:00:01 |
|*  5 |     INDEX RANGE SCAN          | SALES_AC_LOC_I |   621 |       |     2   (0)| 00:00:01 |
|   6 |    TABLE ACCESS BY INDEX ROWID| SALES_AC       |   621 |  5589 |    35   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   4 - filter("LOCATIONS"."COUNTY"='Alpine County' AND "LOCATIONS"."STATE"='California')
   5 - access("SALES_AC"."LOCATION_ID"="LOCATIONS"."LOCATION_ID")

Note
-----
   - this is an adaptive plan


Statistics
----------------------------------------------------------
          0  recursive calls
          0  db block gets
         72  consistent gets
          0  physical reads
          0  redo size
        352  bytes sent via SQL*Net to client
        372  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

Listing 3: AUTOTRACE-Ergebnisse der Abfrage auf LOCATIONS-Dimension

```
SQL> REM Since interleaved ordering was used to cluster the table,
SQL> REM predicates can be used in various combinations. In particular
SQL> REM pruning is still effective if product_name is used alone.
SQL> REM Predicates for location dimensions do not need to be included.
SQL>
SQL> REM Conventional
SQL>
SQL> set termout off
SQL>
SQL> SELECT SUM(amount)
  2  FROM   sales
  3  JOIN   products  ON (sales.product_id = products.product_id)
  4  WHERE  products.product_name  = 'DATEPALM'
  5  /

SUM(AMOUNT)
-----------
   36235437

1 row selected.


Execution Plan
----------------------------------------------------------
```

```
Plan hash value: 2961462735

-------------------------------------------------------------------------------
| Id  | Operation           | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |          |     1 |    21 |  4397   (1)| 00:00:01 |
|   1 |  SORT AGGREGATE     |          |     1 |    21 |            |          |
|*  2 |   HASH JOIN         |          | 69718 |  1429K|  4397   (1)| 00:00:01 |
|*  3 |    TABLE ACCESS FULL| PRODUCTS |     1 |    13 |     2   (0)| 00:00:01 |
|   4 |    TABLE ACCESS FULL| SALES    |  1952K|    14M|  4386   (1)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("SALES"."PRODUCT_ID"="PRODUCTS"."PRODUCT_ID")
   3 - filter("PRODUCTS"."PRODUCT_NAME"='DATEPALM')


Statistics
----------------------------------------------------------
          4  recursive calls
          0  db block gets
      16102  consistent gets
          0  physical reads
          0  redo size
        353  bytes sent via SQL*Net to client
        372  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed

SQL> REM With attribute clustering
SQL>
SQL> set termout off
SQL>
SQL> SELECT SUM(amount)
  2  FROM   sales_ac
  3  JOIN   products  ON (sales_ac.product_id = products.product_id)
  4  WHERE  products.product_name  = 'DATEPALM'
  5  /

SUM(AMOUNT)
-----------
   36235437

1 row selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 1999596888

-----------------------------------------------------------------------------------------------
| Id  | Operation                     | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                 |     1 |    21 |   745   (1)| 00:00:01 |
|   1 |  SORT AGGREGATE               |                 |     1 |    21 |            |          |
|   2 |   NESTED LOOPS                |                 | 69718 |  1429K|   745   (1)| 00:00:01 |
|   3 |    NESTED LOOPS               |                 | 69718 |  1429K|   745   (1)| 00:00:01 |
|*  4 |     TABLE ACCESS FULL         | PRODUCTS        |     1 |    13 |     2   (0)| 00:00:01 |
|*  5 |     INDEX RANGE SCAN          | SALES_AC_PROD_I | 69718 |       |   137   (1)| 00:00:01 |
|   6 |    TABLE ACCESS BY INDEX ROWID| SALES_AC        | 69718 |   544K|   743   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
```

```
   4 - filter("PRODUCTS"."PRODUCT_NAME"='DATEPALM')
   5 - access("SALES_AC"."PRODUCT_ID"="PRODUCTS"."PRODUCT_ID")

Note
-----
   - this is an adaptive plan


Statistics
----------------------------------------------------------
          0  recursive calls
          0  db block gets
        407  consistent gets
          0  physical reads
          0  redo size
        353  bytes sent via SQL*Net to client
        372  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

Listing 4: AUTOTRACE-Ergebnisse der Abfrage auf PRODUCTS-Dimension

beiden Tabellen exemplarisch – *Listing 3* zeigt den Zugriff über die LOCATIONS-Dimension, *Listing 4* den per PRODUCTS-Dimension.

Wie beschrieben, verknüpfen die Beispiel-Abfragen die Faktentabelle mit der LOCATIONS- oder PRODUCTS-Tabelle und filtern auf den entsprechenden Attributen der Dimension. Alle gezeigten Szenarien benötigen wieder deutlich weniger logisches I/O („consistent gets"-Statistik) auf der Tabelle SALES_AC mit dem definierten „Join Attribute Clustering" und zwar auch wieder für unterschiedliche Zugriffswege, sowohl der Join zu der LOCATIONS-Dimension und dem Filter auf der STATE- oder STATE- und COUNTY-Spalte, als auch unabhängig davon der Join zu der PRODUCTS-Dimension und dem Filter auf der PRODUCTS_NAME-Spalte.

Zusammenfassung

Es ist also mittels „Attribute Clusterings" möglich, sowohl einfache Zugriffe auf einer Tabelle per Index zu optimieren, als auch komplexere Szenarien, die Joins zwischen Fakten- und Dimensionstabellen beinhalten.

Eine weitere Besonderheit des „Attribute Clusterings" ist die Möglichkeit, mehr als einen Zugriffsweg auf eine Tabelle optimieren zu können – im Rahmen gewisser Grenzen.

Außerdem sei angemerkt, dass mittels der „Zonemaps" nicht nur Index-Zugriffe per „Attribute Clusterings" optimiert werden können, vorausgesetzt, man befindet sich auf einer unterstützten „Exadata"-basierten Plattform.

Durch das „Attribute Clustering" verbessern sich potenziell noch andere Effekte – wie zum Beispiel die Komprimierung von Daten bei Verwendung der verschiedenen Kompressionsoptionen von Oracle – insbesondere bei der Verwendung der „Inmemory Column Store"-Option von Oracle, bei der die Daten auch komprimiert im Speicher abgelegt werden können.

Ebenso kann das sogenannte „Inmemory Pruning" – das im Grunde auf der gleichen Information beruht wie die erwähnte „Zonemap" – mittels „Attribute Clustering" verbessert werden. Oracle muss also beim „Inmemory Scan" effektiv nur die Daten im Speicher scannen, die das gesuchte Kriterium beinhalten, alles andere kann wieder übersprungen werden.

Über den Autor

Randolf Eberle-Geist ist seit über 25 Jahren als Freiberufler aktiv und hat sich auf Oracle-Datenbank-Performance-Themen spezialisiert. Im Bereich der SQL-Performance-Analyse und der Oracle-Optimizer-Technologie gehört er zu den Top-Experten weltweit und gibt sein Wissen regelmäßig auf Konferenzen sowie in Seminaren und Veröffentlichungen weiter. Er steht für kurzfristige Einsätze im Bereich Performance-Troubleshooting zur Verfügung, bietet aber auch eine Reihe von Workshops für Datenbank-Entwickler und DBAs an. Eine ausführliche Leistungsbeschreibung und eine Auswahl von Workshops finden Sie unter *http://www.oracle-performance.de*.

Randolf Eberle-Geist
randolf.geist@oracle-performance.de


piment fort entier
3,50€/100gr
estragon
4.50/100g
carvi
2.10/100g
mélange cajun
4.40/100g
piment doux
2.20/100g
cumin entier
2.10/100g

Wann PL/SQL nicht verwendet werden sollte

Jürgen Sieben, ConDeS

Dieser Artikel basiert auf einer Fundstelle in einem Fachbuch, die mich bei der Lektüre zu Widerspruch gereizt hat. Mir geht es im Folgenden nicht darum, mit dem Finger auf andere Autoren zu zeigen, sondern der gelesene Artikel dient als Ausgangspunkt, um ein Thema zu erläutern, von dem ich hoffe, dass es für viele Leser interessant sein könnte. Die Fundstelle behandelt ein Rezept in PL/SQL und wirft die Frage auf: Wann sollte PL/SQL nicht verwendet werden?

Die Fundstelle

In einem Kochbuch für PL/SQL-Code habe ich eine sehr schöne PL/SQL-Prozedur für das Problem gefunden, doppelte Einträge aus einer Tabelle zu löschen. Die Prozedur liest sich so *(siehe Listing 1)*.

Wunderbar!

Das Problem

Die Tabelle `EMPLOYEES` gehört zum Schema `HR` und enthält 107 Zeilen. Das ist nichts. Daher machen wir uns eine realistischere Simulation und verwenden die Zeilen der Tabelle `ALL_OBJECTS` mit etwa 100.000 Zeilen für unsere Beispiele. Zunächst kopieren wir die Daten in eine neue Tabelle und fügen die Objekte des Benutzers SH doppelt ein, um Verstöße gegen einen zu schaffenden Primärschlüssel zu erzeugen *(siehe Listing 2)*.

Dann portieren wir das Rezept auf unsere Daten *(siehe Listing 3)*.

Da ich gerade ohnehin Pause machen wollte, rufe ich die Prozedur auf *(siehe Listing 4)*.

Dieser Code hat offensichtlich eine ganze Menge Probleme und eines davon ist die Laufzeit, ein anderes die Tatsache, dass jede Dublette doppelt ausgegeben wird. Lassen Sie uns damit beginnen, zu analysieren, auf welche Weise die Dubletten in der Tabelle aufgespürt werden. Wir iterieren über alle Zeilen der Tabelle `TEST_TABLE` und öffnen für jede ermittelte Zeile einen weiteren Cursor auf `TEST_TABLE`, um die Anzahl der Zeilen zu zählen, die für diese ID vorhanden sind. Um allem die Krone aufzusetzen, ist die Tabelle `TEST_TABLE` des äußeren Cursors zudem noch sortiert, obwohl nicht ganz klar ist, wozu das dienen soll. Der äußere Cursor wird also etwa 95.000 Zeilen lesen und im Arbeitsspeicher des Session-Kontextes sortieren, um anschließend für jede Zeile eine Auswertung auf die gleiche Tabelle durchzuführen. Sollte sich dann herausstellen, dass eine Primärschlüsselinformation doppelt gefunden wurde, habe ich meine Dublette gefunden und gebe sie aus.

Ein Problem besteht darin, dass nicht sichergestellt ist, dass die weiteren `select`-Abfragen in einem lesekonsistenten Zustand zur äußeren Abfrage beantwortet werden. Eine `select`-Abfrage sperrt keine Daten und solange der Session-Isolation-Level nicht gerade auf `SERIALIZABLE` gestellt wurde, wird jede `select`-Abfrage die Daten der Tabelle so sehen, wie sie zum Zeitpunkt dieser entsprechenden Abfrage sichtbar waren. Die gesamte Abfrage wird in über drei Minuten ausgeführt, das heißt, dass die letz-

```
declare
  cursor emp_cur is
    select *
      from employees
     order by employee_id;
  emp_count number := 0;
  total_count number := 0;
begin
  for emp in emp_cur loop
    select count(*)
      into emp_count
      from employees
     where employee_id = emp.employee_id;
    if emp_count > 1 then
      total_count := total_count + 1;
      -- do_something
    end if;
  end loop;
end;
```

Listing 1: PL/SQL-Prozedur für das Problem, doppelte Einträge aus einer Tabelle zu löschen

```
SQL> create table test_table(
  2    owner varchar2(30),
  3    object_id number,
  4    object_name varchar2(30),
  5    object_type varchar2(30));

Tabelle wurde erstellt.

SQL> insert into test_table(owner, object_id, object_name, object_type)
  2  select owner, object_id, object_name, object_type
  3    from all_objects
  4   union all
  5  select owner, object_id, object_name, object_type
  6    from all_objects
  7   where owner = 'SH';

92553 Zeilen erstellt.
```

*Listing 2: Kopieren der Daten in eine neue Tabelle und doppeltes Einfügen der Objekte des Benutzers **SH***

```
SQL> create or replace procedure remove_duplicates
  2  as
  3    cursor object_cur is
  4      select *
  5         from test_table
  6       order by object_id;
  7    l_obj_count number := 0;
  8    l_total_count number := 0;
  9  begin
 10    for obj in object_cur loop
 11      select count(*)
 12        into l_obj_count
 13        from test_table
 14       where object_id = obj.object_id;
 15      if l_obj_count > 1 then
 16        dbms_output.put_line(
 17          obj.object_type || ' ' ||
 18          obj.object_name || ' existiert doppelt.');
 19      end if;
 20    end loop;
 21  end remove_duplicates;
 22  /

Prozedur wurde erstellt.
```

Listing 3: Portierung des Rezepts auf unsere Daten

```
SQL> set serveroutput on;
SQL> call remove_duplicates();

TABLE COSTS existiert doppelt.
TABLE COSTS existiert doppelt.
TABLE PARTITION COSTS existiert doppelt.
TABLE PARTITION COSTS existiert doppelt.
TABLE PARTITION COSTS existiert doppelt.
TABLE PARTITION COSTS existiert doppelt.
TABLE PARTITION COSTS existiert doppelt.
...

Aufruf wurde abgeschlossen.
Abgelaufen: 00:03:14:51
```

Listing 4: Aufruf der Prozedur

ten `select`-Abfragen die gleiche Tabelle über drei Minuten später nach Daten gefragt haben, die durch die erste Abfrage vorab abgefragt worden ist. Zudem werden immerhin etwa 100.000 `select`-Abfragen abgeschickt.

Dieser Ansatz atmet prozeduralen Geist. Das ist nichts Ehrenrühriges, aber in Datenbanken eben völlig falsch. Das Mantra lautet:

- Wenn Du ein Problem hast, löse es in SQL.
- Wenn das gar nicht geht, löse es in PL/SQL.
- Wenn das gar nicht geht, löse es in Java.
- Wenn das gar nicht geht, löse es in C.
- Wenn das gar nicht geht – überlege, was für ein Problem Du hast.

Das Wichtige an diesem Mantra ist die Reihenfolge der einzusetzenden Programmiersprachen. Und das Problem ist, dass die Lösung hier einen entscheidenden Fehler macht.

Lösungsansätze

Versuchen wir, den Code zu retten, indem wir das Problem der Lesekonsistenz und der vielen `select`-Anweisungen angehen. Inhaltlich sollen doch wohl die Daten, die nicht als Dublette vorhanden sind, ignoriert werden. Wir wollen also lediglich die Daten sehen, die doppelt auftauchen. Das klingt doch sehr nach folgender `select`-Anweisung *(siehe Listing 5)*.

```
SQL> select object_id, object_type, object_name, count(*) anzahl
  2    from test_table
  3   group by object_id, object_type, object_name
  4  having count(*) > 1;

 OBJECT_ID OBJECT_TYPE               OBJECT_NAME                    ANZAHL
---------- ------------------------ ------------------------- ----------
     92618 TABLE PARTITION          COSTS                              2
     92613 TABLE PARTITION          COSTS                              2
     92660 TABLE PARTITION          SALES                              2
     92694 TABLE                    TIMES                              2
     92850 INDEX                    CHANNELS_PK                        2
     92891 INDEX PARTITION          COSTS_TIME_BIX                     2
...
310 Zeilen ausgewählt.

Abgelaufen: 00:00:00.48
```

*Listing 5: **select**-Anweisung zur Ansicht doppelter Daten*

Eine erste Verbesserung, das ist klar (achten Sie einmal auf die Zeit). Diese Abfrage würde sich gut für die Verwendung im Cursor der Prozedur eignen, zumal sich dadurch die innere Abfrage komplett erledigt hätte *(siehe Listing 6)*.

Doch ist die Frage, was wir eigentlich erreichen wollten. Wollten wir nur das Ergebnis auf den Bildschirm schreiben? Dann hätten wir den Aufruf von `dbms_output.put_line()` auch als Textkonkatenation in der `select`-Abfrage ersetzen können. Wahrscheinlicher ist jedoch, dass wir die Dubletten einfach loswerden möchten. Welche Optionen hier zur Verfügung stehen, hängt vom Szenario ab. Spielen wir einmal einige durch.

```
SQL> create or replace procedure remove_duplicates
  2  as
  3    cursor object_cur is
  4      select object_id, object_type, object_name, count(*) anzahl
  5        from test_table
  6       group by object_id, object_type, object_name
  7      having count(*) > 1;
  8    l_total_count number := 0;
  9  begin
 10    for obj in object_cur loop
 11      dbms_output.put_line(
 12        obj.object_type || ' ' ||
 13        obj.object_name || ' existiert doppelt.');
 14    end loop;
 15  end remove_duplicates;
 16  /

Prozedur wurde erstellt.

SQL> call remove_duplicates();
TABLE PARTITION COSTS existiert doppelt.
TABLE PARTITION COSTS existiert doppelt.
TABLE PARTITION SALES existiert doppelt.
TABLE TIMES existiert doppelt.
...
Aufruf wurde abgeschlossen.

Abgelaufen: 00:00:00.45
```

Listing 6: Neue Abfrage

Wir haben Daten aus einer CSV importiert

Dann wäre das wahrscheinliche Szenario, dass wir die guten Daten in eine Zieltabelle kopieren und die schlechten verwerfen möchten. Das ist aber auch ganz gut ohne PL/SQL machbar *(siehe Listing 7)*.

Die `insert`-Anweisung kopiert nur die Daten ohne Dubletten. Die Lösung nutzt die analytische Funktion `rank()`, die eine Reihenfolge über ein Gruppierungskriterium nach einem Sortierkriterium bildet. Da die doppelten Zeilen gleich sind, benötigen wir ein Kriterium, um uns zwischen den beiden zu entscheiden. Hier bietet sich zum Beispiel die `ROWID` an. In der äußeren Abfrage werden anschließend alle Dubletten ausgefiltert, denn diese haben einen `Rang > 1`. Die Liste der `ROWID` der verbleibenden Zeilen wird dann zum Identifizieren der Zeilen verwendet, die eingefügt werden sollen.

Das Problem der PL/SQL-Anweisung wäre ja, dass die „guten" Zeilen durch die prozedurale Logik zeilenweise in die Zieltabelle überführt werden müssten, das ist durch diese Anweisung nicht nötig.

Wir möchten Daten bereinigen, um einen Primärschlüssel einzurichten

Eine Löschanweisung direkt in der Tabelle ist etwas aufwändiger, weil eine Entscheidung getroffen werden muss, welche von zwei Zeilen mit gleichen Werten gelöscht werden soll.

Auch diese Lösung lässt sich ohne PL/SQL realisieren *(siehe Listing 8)*.

```
SQL> create table target_table
  2  as select *
  3    from test_table
  4   where null is not null;

Tabelle wurde erstellt.

SQL> alter table target_table add constraint pk_target_table
  2    primary key(object_id);

Tabelle wurde geändert.

SQL> insert into test_table(owner, object_id, object_name, object_type)
  2  select owner, object_id, object_name, object_type
  3    from test_table
  4   where rowid in (
  5         select row_id
  6           from (select rowid row_id,
  7                        rank() over (
  8                          partition by object_id order by rowid) rang
  9                   from test_table)
 10          where rang = 1);

92243 Zeilen erstellt.
Abgelaufen: 00:00:00.17
```

Listing 7: Die guten Daten werden in eine Zieltabelle kopiert und die schlechten verworfen

```
SQL> delete from test_table
  2   where rowid in (
  3          select row_id
  4            from (select rowid row_id,
  5                         rank() over (
  6                           partition by object_id order by rowid) rang
  7                    from test_table)
  8           where rang > 1);

310 Zeilen gelöscht.
Abgelaufen: 00:00:00.06
```

Listing 8: Löschanweisung

```
SQL> call dbms_errlog.create_error_log('TARGET_TABLE', 'TARGET_ERR');

Aufruf wurde abgeschlossen.

SQL> insert into target_table
  2  select *
  3    from test_table
  4     log errors into target_err
  5  reject limit unlimited;

92243 Zeilen erstellt.

Abgelaufen: 00:00:00.23

SQL> select count(*)
  2    from target_err;

  COUNT(*)
----------
       310
```

*Listing 9: **insert**-Anweisung in der Zieltabelle mit einer **log errors**-Klausel*

```
SQL> insert first
  2    when rang = 1 then
  3    into target_table(owner, object_id, object_name, object_type)
  4         values(owner, object_id, object_name, object_type)
  5    else into target_err(owner, object_id, object_name, object_type)
  6         values(owner, object_id, object_name, object_type)
  7  select t.*, rank() over (partition by object_id order by rowid) rang
  8    from test_table t;

92553 Zeilen erstellt.

Abgelaufen: 00:00:00.14
```

Listing 10: multi-table-insert-Anweisung

Achten Sie auch hier gern auf die Ausführungszeit. Die Lösung nutzt wieder die analytische Funktion `rank()`. In der äußeren Abfrage werden nur Dubletten gefiltert, denn diese haben einen `Rang > 1`. Die Liste der `ROWID` dieser Zeilen wird anschließend zum Bereinigen der Tabelle verwendet.

Die guten sollen ins Töpfchen, die schlechten ins Kröpfchen

Ist das Ziel die Separierung der beiden Datenmengen, stehen ebenfalls mehrere Wege zur Verfügung. Zum Beispiel könnten Sie eine `insert`-Anweisung in die Zieltabelle mit einer log `errors`-Klausel ausstatten, die bei einem Primärschlüsselverstoß die verstoßende Zeile in eine Fehlertabelle schreiben wird *(siehe Listing 9)*.

Alternativ könnte der Weg über eine `multi-table-insert-Anweisung` laufen. Bei dieser Variante würden die Daten durch eine entsprechende `select`-Anweisung mit einem Rang ausgestattet und beim Einfügen der Daten eine Fallunterscheidung nach dem Rang ausgeführt *(siehe Listing 10)*.

Die Aufbereitung der Daten haben Sie schon gesehen, hier entfällt nur die Filterung über die Dubletten, da diese für die Fallunterscheidung benötigt werden.

Zusammenfassung

Das Hauptproblem der Programmierung von Datenbanken ist es, der Datenbank möglichst wenig im Weg zu stehen. Genau das aber passiert gerade routinierten Anwendungsprogrammierern häufig. Zu sehr ist man den prozeduralen Ansatz gewohnt, als dass man sich auf die Suche nach einem mengenorientierten Ansatz macht. SQL hat viele Anstrengungen unternommen, prozedurale Programmierung unnötig zu machen und Standardprobleme direkt aus SQL heraus zu lösen.

Sollten Ihnen einige der eingesetzten Techniken unbekannt gewesen sein (`log errors`-Klausel, multi-table-insert, analytische Funktionen), verstehen Sie das doch bitte als Einladung, sich wieder einmal mit SQL auseinanderzusetzen. Die Lösungen sind im Regelfall kürzer, schneller, besser zu warten und benötigen keine weitreichenden Unittests. Genug Argumente also, PL/SQL auch einmal im Schrank zu lassen.

Jürgen Sieben
j.sieben@condes.de

BEST OF DOAG ONLINE

Eine Auswahl der besten DOAG News Juni/Juli 2023

CloudLand 2023: Die zweite Ausgabe des Cloud Native Festivals wächst auf 500 Besucherinnen und Besucher

Mit über 140 Beiträgen, zahlreichen Workshops und interaktiven Formaten sowie kurzweiligen Themennächten und einem innovativen Sponsorenkonzept weiß die CloudLand erneut zu überzeugen.

Engineered Systems: Oracle kündigt neue Exadata X10M an

Oracle hat jetzt die Spezifikationen der nächsten Generation seiner Exadata-Database-Machine-Plattform, der Exadata X10M vorgestellt.

Interview mit Dr. Julia Freudenberg, Keynote Speakerin der CloudLand 2023

"Kinder und Jugendliche für die Möglichkeiten der digitalen Welt zu begeistern und das Leuchten in den Augen zu sehen, wenn sie am Ende eines Kurses ihr eigenes Projekt programmiert haben, ist die beste Motivation überhaupt."

DOAG Datenbank Kolumne: Der neue SQL-Standard ist da!

Wer hat nicht schon einmal etwas vom SQL-Standard gehört? Ein gemeinsames Gremium von ISO und IEC standardisiert SQL unter Mitwirkung nationaler Normungsgremien wie ANSI oder DIN.

Java-Lizenzfallen bei Oracle?

Wie jetzt auch von deutschsprachigen Portalen berichtet wird, versucht Oracle laut einer Meldung des Online-Portals The Register, Kunden zum Wechsel ihrer Lizenz für Java SE zu motivieren.

DOAG 2023 Datenbank mit Exaday: Erfolgreiche Konferenz in Düsseldorf!

Es war mal wieder Zeit, für internationale Referentinnen und Referenten in Düsseldorf Bühnenluft zu schnuppern und einem richtigen Publikum im Saal gegenüberzustehen.

Wir begrüßen unsere neuen Mitglieder

Natürliche Mitglieder:

- Marko Mosch
- Phillip Ansorge
- Fatih Karabacak

Korporative Mitglieder:

- Provinzial Nord Brandkasse AG, Repräsentantin: Nina Rosin
- Sparkassen DirektVersicherung AG, Repräsentant: Jost Müller
- Provinzial Versicherung AG, Repräsentant: Michele Turco
- Lippische Landesbrandversicherung AG, Repräsentant: Christian Peters
- SINTEC Informatik Gmbh, Repräsentantin: Beate Schiler

Termine

August 08

11.08.2023
Oracle Datenbank auf ARM Prozessor-Technologie
DB WebSession mit Clemens Bleile
Online

17.08.2023
Regionaltreffen Hamburg
APEX und Oracle Datenbank
Hamburg

22.08.2023
Regionaltreffen NRW
Thema wird zeitnah benannt
Dortmund

24.08.2023
APEX 23.x
DevTalk Summer Special mit Carsten Czarski + APEX-Team
Moderation: Carolin Krützmann
Online

30.08.2023 - 31.08.2023
Expertenseminar: Oracle Cloud Infrastructure – von der Konsole zur Automation – Kickstart!
Berliner Expertenseminar mit Martin Berger und Stefan Oehrli
Berlin

September 09

01.09.2023
DOAG Webinar Oracle Support: AHF - Autonomous Health Framework
Online

08.09.2023
DBA_INDEXE_USAGE – Speed up your DML DB WebSession mit Christian Pfundtner
Online

21.09.2023
Oracle Forms
DevTalk mit Jürgen Menge & Frank Hoffmann
Moderation: Carsten Czarski
Online

26.09.2023 - 27.09.2023
Noon2Noon: Datenbank-Security
Wiederholungstermin aufgrund der hohen Nachfrage beim ersten Noon2Noon-Workshop
Hannover

27.09.2023 - 28.09.2023
Expertenseminar: Oracle Datenbank Indexing. Berliner Expertenseminar mit Randolf Eberle-Geist
Berlin

Oktober 10

13.10.2023
Oracle 23c DB Nest im praktischen Einsatz
DB WebSession mit Stefan Oehrli
Online

19.10.2023
Back-end Testing bei DB-Entwicklungs-projekten – Teil 2
DevTalk mit Oliver Lemm und Samuel Nitsche
Online

26.10.2023
Regionaltreffen NRW
Thema wird zeitnah benannt
Wuppertal

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): Janis Ax, Randolf Eberle-Geist, Florian Graßhoff, Timo Herwix, Frank Hoffmann, Benjamin Klatt, Torsten Kleiber, Carolin Krützmann, Oliver Lemm, Marcos López, Martin Meyer, Moritz Reinwald, Katharina Schraft, Hatice Sen, Jürgen Sieben, Daniel Steiger, Stephan Tönnies.

Titel, Gestaltung und Satz:
Diana Tkach
DOAG Dienstleistungen GmbH
(Anschrift s.o.)

Fotonachweis:
Titel: © freepik | *www.freepik.com*
S. 6: © Brooke Cagleg | *www.unsplash.com*
S. 14: © analogicus | *www.pixabay.com*
S. 24: © Starship | *www.maxpixel.net*
S. 32: © Kanenori | *www.pixabay.com*
S. 36: © geralt | *www.pixabay.com*
S. 44: © Tama66 | *www.pixabay.com*
S. 52: © Martin Meyer | *www.doag.org*
Titel S. 52: © freepik | *www.freepik.com*
S. 60: © Jordan Whitfield | *www.unsplash.com*
S. 70: © PublicDomainPictures | *www.pixabay.com*
S. 78: @ Pexels | *www.pixabay.com*
S. 85: © freepick | *www.freepik.com*

Anzeigen:
sponsoring@doag.org

Mediadaten und Preise:
www.doag.org/go/mediadaten

Druck:
WIRmachenDRUCK GmbH,
www.wir-machen-druck.de

Inserentenverzeichnis

Early Bird
bis
9.10.

European NetSuite User Days

20. – 21. NOVEMBER 2023

IN NÜRNBERG

netsuite.doag.org

DOAG

JavaLand

www.javaland.eu

IM PHANTASIALAND BEI KÖLN

27. - 29.2.2024

Präsentiert von: DOAG Heise Medien Community Partner: iJUG Verbund