Nr. 4 | September 2018 | ISSN 2366-7915 | www.doag.org

Red Stack Magazin

DOAG SOUG swiss oracle user group AOUG AUSTRIAN ORACLE USER GROUP

Security

Aus der Praxis

Effiziente Delivery mit APIs, Microservices und DevOps

Im Interview

Alexander Kornbrust, Red-Database-Security GmbH

Apex 18.1

Page Designer New Features


Early Bird
bis zum
28. Sep.
2018
DOAG
Konferenz + Ausstellung
20. - 23. November
in Nürnberg
2018.doag.org
PROGRAMM
ONLINE
mit rund 450 Vorträgen
Eventpartner:
AOUG
SOUG
swiss oracle user group
iJUG
Verbund
ORACLE

Bruno Cirone
DOAG Themenverantwortlicher Security

Liebe Mitglieder, liebe Leserinnen und Leser,

das Thema „Security" hat in den letzten Jahren eine immer wichtigere Rolle in unserer Arbeit eingenommen. Vor gar nicht allzu langer Zeit wurde ein neuer Benutzer noch auf Zuruf angelegt. Die Rechte wurden einfach von einem anderen User kopiert. Noch schlimmer, manchmal wurden sie nach dem Motto „gib dem User einfach DBA-Rechte, dann ist es gut" vergeben.

Heute haben wir Datenschutzgesetze, interne Regularien und verschiedene Nachweispflichten. Das Anlegen, Löschen und Ändern der User sowie die Vergabe der Rechte muss im Unternehmen nach einem Regelprozess mit entsprechenden Genehmigungen erfolgen. Die Revalidierung jedes Benutzers und dessen Rechte erfolgt in regelmäßigen Abständen.

Auf einer der letzten Security-Panels der DOAG Konferenz in Nürnberg kam allerdings aus dem Publikum die Frage: „Was kann ich tun, damit die Geschäftsführung die Security ernst nimmt und Ressourcen wie Personal, Lizenzen, Geld für benötigte Tools etc. zur Verfügung stellt?" Im Regelfall resultiert diese Fragestellung daraus, dass die entsprechenden Entscheider den Wert ihrer Daten nicht kennen oder nicht einschätzen können. Daher wird auch die Notwendigkeit der Investitionen nicht gesehen. Meine Antwort ist: „In den Datenbanken werden Millionenwerte oder gar Milliardenwerte verwaltet. Es ist zwar nicht immer sofort ersichtlich, aber die immateriellen Werte wie Kundendaten, Patente, Rezepturen etc. stellen einen hohen Wert für ein Unternehmen dar. Man stelle sich vor, ein Mitbewerber erhält Einblick in Forschungsergebnisse, Finanzdaten, Angebote, Patente ..."

Security ist nicht Selbstzweck oder ein Nice-to-have, sondern ein wichtiger Baustein für die Zukunftssicherung des Unternehmens. Der Schutz der Daten und Datenbanken schützt das Unternehmen und viele Arbeitsplätze.

In diesem Sinne wünsche ich Ihnen, dass Ihre Daten sicher sind.

Ihr

Bruno Cirone

In vielen Skripten rund um die tägliche Wartung müssen Passwörter sicher hinterlegt sein

Der sichere Betrieb der Oracle-Datenbank durch sogenannte „Gewaltenteilung“

Seit dem Jahr 2017 bietet Oracle verschiedene Cloud Services in einem Frankfurter Rechenzentrum an

Einleitung

Security

Aktuell

Spatial

Entwicklung

Intern

Timeline

9. Mai 2018

Fried Saacke, DOAG-Vorstand und Geschäftsführer, und Mylène Diacquenod von der DOAG-Geschäftsstelle besuchen auf Einladung von Carsten Diercks, stellvertretendem Vorsitzenden des de'ge'pol Deutsche Gesellschaft für Politikberatung e.V., in Berlin die Veranstaltung „DSGVO in der Praxis – Public Affairs und die neuen Anforderungen". Zwei Fachanwälte und Experten im Datenschutzrecht stellen die Eckdaten der Verordnung vor und diskutieren praxisbezogen die Fragen der Teilnehmer. Das Thema beschäftigt die DOAG-Geschäftsstelle enorm, um alle Vorgaben korrekt einzuhalten.

14./15. Mai 2018

Rund 250 Teilnehmer treffen sich in Düsseldorf zur DOAG 2018 Datenbank, um wissenswerte Neuigkeiten rund um die Oracle-Datenbank zu erfahren und sich untereinander auszutauschen. Damit setzt sich das erfreuliche, stetige Wachstum der Veranstaltung auch in diesem Jahr fort. Die wissenshungrigen Besucher haben zwei Tage lang die Qual der (Aus-)Wahl zwischen knapp vierzig Vorträgen in vier parallelen Vortragsreihen. Neben klassischen Datenbank-Themen gibt es in diesem Jahr auch Vorträge zu den Themen „Datenschutz" und „Hybride Architekturen". Auch Datenbank-Neulinge kommen dabei nicht zu kurz. Wer lieber die Theorie in die Praxis umsetzen und sein Wissen zu RMAN erweitern möchte, hat an beiden Tagen die Möglichkeit, den Workshop „RMAN Duplicate" der MT AG zu besuchen. Ein weiteres Highlight ist der interaktive Vortrag „Sein oder Nichtsein – ist das die Frage für den DBA?" von Markus Flechtner und Johannes Ahrends, der sich mit der Zukunft des Datenbank-Administrators befasst. Auch die lockere Open-Mic-Session unter der provokanten Überschrift „DBAs sind unfehlbar?!" lädt die Teilnehmer zum Mitwirken ein. Dabei hat jeder Freiwillige fünf Minuten Zeit, einen besonders amüsanten Projektbericht zu teilen. Die rege Beteiligung und die vielen Anekdoten sorgen für einige Lacher und machen die Session zu einem sehr gelungenen Ausklang des ersten Tages.

Gute Stimmung auf der DOAG 2018 Datenbank

25. Mai 2018

Der DOAG-Vorstand ist in Hamburg zu seiner regelmäßigen Sitzung. Im Mittelpunkt stehen die offenen Punkte aus der Delegiertenversammlung, insbesondere der Auftrag zur Gründung einer neuen Arbeitsgruppe mit Vertretern aus Delegiertenversammlung und Vorstand. Diese soll die bestehenden Strukturen des Vereins hinsichtlich der Themenausrichtung überprüfen und eine Strategie für die zukünftige Ausrichtung der DOAG erarbeiten, um neuen Technologie-Themen in Entwicklung und Infrastruktur zu begegnen.

29. – 31. Mai 2018

Die Mitarbeiterinnen und Mitarbeiter der DOAG-Geschäftsstelle treffen sich zum diesjährigen Team-Workshop. Das Thema unter der Leitung von Leander Greitemann von Institut für angewandte Kreativität lautet „Perspektivenwechsel. Vom reaktiven zum kreativen Mindset". Das Team sammelt wertvolle neue Erkenntnisse und tankt viel Energie, um die anstehenden Aufgaben optimal zu lösen.

1. Juni 2018

Der DOAG-Vorstand stimmt dem Vorschlag für das NextGen-Programm auf der DOAG 2018 Konferenz + Ausstellung zu. Bis zu 80 Studierende und Auszubildende werden kostenfreien Zutritt zur Konferenz erhalten. Im Gegenzug werden sie die Organisation der Veranstaltung im kleinen Rahmen unterstützen. Die DOAG möchte es damit dem IT-Nachwuchs ermöglichen, im geschützten Rahmen bereits früh wertvolle Kontakte zu knüpfen und die meist recht frisch erworbenen Fähigkeiten auszuprobieren.

Motiviert bei der Sache: Studierende und Auszubildende auf der DOAG Konferenz + Ausstellung

4. Juni 2018

Der EOUC Oracle User Group Summit startet im kroatischen Zagreb mit vielen Repräsentanten von Oracle-, MySQL- und Java-Anwendergruppen aus ganz Europa. Als Vertreter der DOAG nehmen Fried Saacke, Christian Trieb, Ralf Kölling und Ingo Sobik teil. Im Mittelpunkt steht die Zusammenarbeit der Usergroups untereinander und mit Oracle. Zu Beginn stellt sich mit Bianca Green die neue Ansprechpartnerin für die Usergroups bei Oracle vor. Ihr langjähriger Vorgänger Tom Scheirsen soll künftig als Director Community Experience and Global Initiatives weltweit die Usergroups animieren, um mehr mit der Cloud zu machen.

8. Juni 2018

Zum Abschluss des EOUC Oracle User Group Summit lädt die DOAG alle Vertreter der europäischen Usergroups nach Nürnberg ein, um wie bereits vor zwei Jahren im Rahmen der DOAG 2018 Konferenz + Ausstellung die Zusammenarbeit der Usergroups zu intensivieren.

11. Juni 2018

Die DOAG kann den profilierten Redner Lars Vollmer als Keynote-Speaker für die DOAG 2018 Konferenz + Ausstellung gewinnen. Der promovierte Ingenieur, Honorarprofessor und Begründer eines Thinktanks für moderne Unternehmensführung erhellt in seinen Publikationen neue Perspektiven auf Wirtschaft und Unternehmen. Im Mittelpunkt stehen die Thesen von Lars Vollmer, dessen Bestseller „Zurück an die Arbeit" vom Spiegel Potenzial als Innovationshandbuch attestiert wird: „Eine fulminante Abrechnung mit gängigen Managementmethoden und ein Manifest für echte Arbeit."

Lars Vollmer wird Keynote-Speaker der DOAG 2018 Konferenz + Ausstellung

11. Juni 2018

Am Vortag zur Anwenderkonferenz der Austrian Oracle User Group findet im Konferenzhotel „Austria Trend Savoyen Vienna" der „AOUG Master Class Monday" statt. In mehreren parallelen Halbtags-Workshops können sich die Besucher über aktuelle Themen wie „Oracle-Datenbank-Anwendungsentwicklung" und „Adaptive Query Optimization" bei den Experten Christian Antognini und Gerald Venzl informieren. Am späten Nachmittag geht es dann mit einem historischen Postbus zu einer Sightseeing-Tour auf den Wiener Kahlenberg. Leider schafft der Bus wegen eines Defekts die Anreise zum Hotel nicht, sodass die Teilnehmer zunächst auf Taxis umsteigen müssen. Bei schönster Abenddämmerung gibt es dann dennoch im „Häuserl am Roan" am Kahlenberg kulinarische Köstlichkeiten wie Wiener Schnitzel, Palatschinken und gutes Bier. Spät abends kommt schließlich überraschend ein Ersatzbus, der alle Gäste in einer sehr abenteuerlichen Nachtfahrt mit etwa 25 Stundenkilometern wieder zurück zum Hotel bringt.

12. Juni 2018

In Wien findet die AOUG-Anwenderkonferenz 2018 der Österreichischen User Group (AOUG) statt. Es ist zugleich der dreißigste Geburtstag des Vereins. 150 Teilnehmer können sich einen Tag lang zu zahlreichen Themen informieren. Nach einem kurzen, nostalgischen Ausflug in die Vergangenheit der AOUG eröffnet Neil Sholay mit seiner Keynote zum Thema „Digitale Transformation" die Konferenz. In weiterer Folge zeigt Christopher Clewes, wie man dazu mit Oracle innerhalb von nur ein paar Wochen eine Idee ins Leben rufen kann. Im Anschluss gibt es dann 28 spannende Fachvorträge mit Sprechern aus zahlreichen Nationen in vier parallelen Tracks.

Martin Winkler, Managing Director Oracle Österreich; Neil Sholey, Vice President of Digital, Oracle EMEA; Ingrid Kriegl, AOUG-Vorstand; Christopher Clewes, Programme Director for Oracle Industry Innovation Advisors and Cloud Enterprise Architects across EMEA and JAPAC, und Michael Hatzinger, AOUG-Präsident (von links nach rechts)

14. Juni 2018

Rund 50 Teilnehmer kommen bei der DOAG 2018 Logistik + IT im GS1 Germany Knowledge Center in Köln zusammen, um sich über die Chancen der Digitalisierung in der Logistik-Branche auszutauschen. Ein intelligenter Kühlschrank, der darüber Auskunft gibt, welche Nahrungsmittel gebraucht werden und in welchem Supermarkt man diese bekommt – vielleicht ist das schon bald Realität. Bei einer Führung können die Teilnehmer das digitale Einkaufserlebnis der Zukunft schon einmal live erleben. Ausgestattet mit einem iPad und ihrem persönlichen Avatar erkunden sie den zukünftigen Supermarkt samt unsichtbaren Barcodes und personifizierten Produkt-Hinweisen wie zum Beispiel Coupons oder Allergie-Informationen. Der Keynote-Speaker Dennis Schenkel, Projektmanager und Consultant bei neuland, hat einige Denkanstöße im Gepäck. Seine Mission: Die Digitalisierung so zu nutzen, dass „wir keinen zweiten Planeten brauchen". Dank der fortschreitenden Dematerialisierung scheint dies nun möglich: Wenn beispielsweise Bahntickets mehr und mehr in digitaler Form in der Handy-App genutzt werden oder Apps plötzlich Schlüssel ersetzen, sinkt der Ressourcen-Verbrauch drastisch. Daher sein Appell: „Nutzen wir die Digitalisierung und gestalten wir die Erde 5.0!" Im Anschluss an seinen Vortrag entwickelt sich eine angeregte Diskussion, was nicht zuletzt den vorgestellten drei Grundsätzen der Digitalisierung und deren möglichen sozialen, politischen und ökonomischen Folgen geschuldet ist. In zwei Streams dreht sich einen Tag lang alles rund um „Digitale Transformation und Logistik 4.0" sowie Oracle-basierte Digitalisierungslösungen für Supply Chain und Logistik. Industrie 4.0, neue Technologien wie Amazons Alexa, Blockchain und Cloud sowie Echtzeitanalysen sind nur einige Themen der zahlreichen Fachbeiträge der Veranstaltung und zeigen anschaulich die Aktualität der Branche. Bei der zweiten Führung am Mittag machen die Teilnehmer Bekanntschaft mit dem Roboter „Ava", der die sechs Stufen der Wertschöpfungskette vom Erzeuger zum Shopper mit multimedialer Unterstützung zeigt. Das innovative GS1 Germany Knowledge Center bietet damit das ideale Umfeld der Veranstaltung; selbst in kleinen Details des Hauses, wie etwa digitalen Raumschildern, ist der Funke der Digitalisierung zu entdecken.

18./19. Juni 2018

Die Middleware- und „Engineered Systems"-Community trifft sich im Flemings Conference Hotel in Frankfurt am Main, um Neuigkeiten, Wissen und Gedanken auszutauschen. Zum ersten Mal liegt einer der Schwerpunkte in diesem Jahr bei Middleware. Experten präsentieren ihre Erfahrungen zu Betrieb und Tuning von Oracle Middleware, Virtualisierung (Cloud), Migration oder hy-

briden Umgebungen. Im Bereich der Engineered Systems liegt der Schwerpunkt auf Exadata und ODA. Von der Implementierung über den Betrieb bis hin zum Tuning und Upgrade erfahren die Teilnehmer alles Wissenswerte über den Lifecycle der Maschinen.

24. Juni 2018

Die Swiss Oracle User Group organisiert mit dem SOUG Day ein Highlight für alle Mitglieder im einzigartigen Ambiente des Trafo in Baden. An dieser ganz besonderen Location trifft Industriekultur auf ein breites Programm mit Themen rund um die Welt der Oracle-Infrastruktur, Oracle-Datenbank und natürlich der Oracle Cloud. Bereits die Keynote von Thomas Würthinger, Senior Research Director Oracle Labs, zeigt einen Trend, den auch die Swiss Oracle User Group aufgreift: Einsatz der Open-Source-Technologien in Kombination mit etablierten Produkten und Plattformen. Dabei wird deutlich, wie die Oracle Labs mit GraalVM eine Runtime für unterschiedliche Sprachen entwickeln, die in Zeiten des Cloud-Native-Development die Entwicklung und die dafür genutzten Werkzeuge den Entwicklungsteams überlässt, dabei aber gleichzeitig sicherstellt, die bestmögliche Performance und Skalierung von Anwendungen in der Cloud zu ermöglichen. Dass es am Ende keine Frage ist, ob Open Source oder Herstellerprodukt zum Einsatz kommen, sondern am Ende meist eine Kombination aus beiden Welten die beste Lösung hervorbringt, ist ein „Key-Learning" für alle Teilnehmer. In insgesamt 24 Sessions wird ein breites Themenspektrum angeboten. Auch wenn die Oracle-Datenbank nach wie vor ein elementares Thema für die SOUG ist und bleiben wird, zeigt sich auch in der Agenda des SOUG Day ein eindeutiger Trend. Mehr als 100 Teilnehmer bestätigen dabei, dass sich die Rollen der IT in den Unternehmen und dadurch auch die Anforderungen an die Mitarbeiter in einem Wandel befinden. Durch die Etablierung von Dev-Ops-Konzepten und „Autonomous Cloud"-Produkten ändert sich, wie Thomas Teske von Oracle in seinem Vortrag darstellt, das Aufgabenfeld der Datenbank-Experten. Development und Infrastruktur rücken enger zusammen und die IT selbst wird zum „Innovator" im Unternehmen. Auch die Swiss Oracle User Group stellt sich diesem Wandel und spricht damit neue Zielgruppen an, sowohl im Bereich „Innovation" als auch in der Community der Developer und agilen Entwicklungsteams. So werden am SOUG Day zum Beispiel das „Projekt FN" für Serverless Functions von Oracle vorgestellt, aber auch eine Übersicht über den Markt der Blockchain-Technologien und konkrete praktische Anwendungsfälle dargestellt. Abgerundet wird das Programm mit „Soft Skill"-Vorträgen rund um das Thema „Agiles Projektmanagement" und Use-Cases für den Bereich „Cloud-Integration". Das „Experiment" der Swiss Oracle User Group, in einem Ganztages-Event die etablierten Datenbank-Themen um wertvolle Inhalte aus den Bereichen „Development" und „Innovation" zu erweitern, wird als voller Erfolg betrachtet. Mit neuen Formaten, die sowohl regional als auch inhaltlich in den bestehenden Mitgliedsunternehmen ein breiteres Publikum adressieren, wird die SOUG weiter ein attraktives Programm anbieten und somit für Unternehmen eine essenzielle Plattform sein, rund um die Oracle-Cloud und alle Prozesse, die den Wandel der IT in Unternehmen aktuell begleiten.

Dr. Dietmar Neugebauer
Ehemaliger DOAG-Vorstandsvorsitzender

Aus der Ferne betrachtet: Meine Daten gehören nur mir – wirklich?

Bestimmt viele von Ihnen haben wie ich vor dem 25. Mai 2018 zum Inkrafttreten der EU-Datenschutz-Grundverordnung eine Vielzahl von Mails erhalten mit der Nachfrage, ob man über diverse Verteiler noch Mitteilungen erhalten möchte. Was mich erstaunte: dass hier bei verschiedenen Verteilern durchaus verschiedene Ansätze vorgeschlagen wurden. Die Vorsichtigen, bei denen man sich aufs Neue anmelden musste – was man dann natürlich geflissentlich ignoriert hat – oder die etwas Forscheren, die einen nur darauf hinwiesen, dass man den Newsletter abbestellen könnte – diese Möglichkeit hatte man bisher ja auch schon. Scheinbar ein Gesetz, das unterschiedliche Auslegungen zulässt.

Was mir allerdings mehr zu denken gibt, ist der Aufwand, der hier an vielen Stellen betrieben werden musste, ohne dass ein Mehrwert generiert worden ist. Bei den Newslettern, bei denen ich mich angemeldet habe, habe ich das mit der Absicht getan, für mich interessante Informationen zu erhalten. Die Spam-Mails von anderen Verteilern sind deshalb nicht weniger geworden.

Wenn man dann liest, welchen Aufwand gerade kleinere Unternehmen betreiben müssen, um den Anforderungen der DSGVO gerecht zu werden, und wie sie in ihrer Produktivität und in ihren Innovationen durch solchen Bürokratismus gelähmt werden, stellt sich mir die Frage, ob meine Daten für mich so wichtig sind, dass sie so behandelt werden müssen.

Basieren nicht viele unserer Errungenschaften auf der Möglichkeit eines schnellen Daten-Austausches und -Zugriffs? Wollen wir nicht in Zukunft den Austausch unserer Daten beim Allgemeinarzt, Facharzt und im Krankenhaus, um eine schnellere und effektivere Behandlung zu ermöglichen? In der Medizin der Zukunft mit personalisierten Therapien und der Auswertung von großen Datenmengen zur rechtzeitigen Erkennung der globalen Ausbreitung von Infektionskrankheiten werden die Daten jedes Einzelnen gesammelt, um den Patienten umfassender zu behandeln sowie schnellere und bessere Diagnosen zu stellen.

Aufgrund der steigenden Kosten im Gesundheitswesen haben natürlich auch die Versicherungen Interesse, ihren Mitgliedern Anreize zu einer gesünderen Lebensweise zu geben. Wenn das über die Höhe der Beiträge gemacht werden soll, geht das natürlich nur, wenn sie auch Informationen über die Lebensweise ihrer Kunden bekommen. Zweimal Sport pro Woche und weniger Krankenkassenbetrag?

Hier kann die Zukunft noch viele Ideen bringen – etwa geringere Kfz-Versicherung gekoppelt an das Fahrverhalten (oder an das Punktekonto in Flensburg) oder die Entwicklungen zur Smart City mit Vereinfachung der Verwaltungsabläufe (Gerhard Schröder: „Die Daten sollen laufen, nicht der Bürger").
Wollen wir nicht diese Weiterentwicklungen durch eine bessere Datenkommunikation? Oder sehe ich das falsch? Gibt es ein persönliches Eigentumsrecht an seinen Daten? Ich sehe es nicht so!

Alexander Kornbrust (links) im Gespräch mit Dr. Dietmar Neugebauer

„Die Oracle-Datenbank bietet viele Security-Features …“

Das Thema „Sicherheit“ ist in der IT ein Dauerbrenner. Dr. Dietmar Neugebauer, ehemaliger Vorstandsvorsitzender der DOAG, und Wolfgang Taschner, Chefredakteur des Red Stack Magazin, sprachen darüber mit Alexander Kornbrust, Geschäftsführer der Red-Database-Security GmbH.

Sie sind auf vielen DOAG-Veranstaltungen als Sicherheitsexperte vertreten. Was fällt Ihnen spontan zum Thema „Sicherheit“ ein?
Alexander Kornbrust: Sicherheit ist für mich ein Prozess, an dem man ständig arbeiten muss. Das Problem im Unternehmen ist, dass jeder etwas anderes darunter versteht. Es ist wichtig, diese Sichtweisen alle unter einen Hut zu bringen: Datenbank-Administratoren sollten sich um Patches sowie um Rechte und Privilegien kümmern, die Entwickler müssen ihre Anwendungen sauber programmieren und die Compliance- und Audit-Verantwortlichen kümmern sich um die Einhaltung der Vorschriften.

Was bedeutet Sicherheit für Sie?
Alexander Kornbrust: Sicherheit lässt sich nur dann erreichen, wenn sie in allen Bereichen umgesetzt ist.

Was sind die größten Fehler, die Unternehmen in Bezug auf Sicherheit machen können?

Alexander Kornbrust: Ein Unternehmen muss immer alle Komponenten im Blick haben. Es darf sich also nicht nur auf Verschlüsselung oder Zugangskontrolle konzentrieren, sondern muss eine umfassende Grundsicherheit herstellen. Sobald überall ein gewisses Niveau erreicht ist, kann man die nächsthöheren Stufen anstreben.

Was sind die wichtigsten Punkte, die Unternehmen hinsichtlich Sicherheit beachten müssen?

Alexander Kornbrust: Eine enorme Lücke ist die Passwort-Vergabe, selbst bei großen Unternehmen. Viel zu häufig sind Benutzername und Passwort identisch. Zuallererst sind also die ganz offensichtlichen Dinge zu erledigen; sonst kann man sich alle anderen Maßnahmen fast sparen. Ich empfehle, dass jeder im Unternehmen – abhängig von seiner Rolle – eine entsprechende Sensibilität entwickeln sollte. Das heißt, Mitarbeiter im Büro entsprechend zu schulen, dass sie nicht auf Angriffe über Social Media hereinfallen, die Entwickler umfassen auszubilden, um sichere Applikationen zu schreiben, und die Datenbank-Administratoren bei der Umsetzung eines sicheren Zugangskonzepts zu unterstützen. Die Oracle-Datenbank bietet viele Security-Features, die bei den Administratoren gar nicht bekannt sind beziehungsweise von ihnen nicht genutzt werden.

"Eine enorme Lücke ist die Passwort-Vergabe, selbst bei großen Unternehmen ..."

Gibt es eine hundertprozentige Sicherheit?

Alexander Kornbrust: Nein, denn erstens kann man das gar nicht messen. Zum anderen sind ja nie alle Sicherheitslücken bekannt, wie man gerade erst bei den Lecks in den Intel-Chips gesehen hat. Zudem dauert es ziemlich lange, bis die Chip-Hersteller überhaupt eine Lösung anbieten. Hinzu kommt, dass es extrem schwierig ist, Probleme zu identifizieren, wenn mehrere Systeme parallel laufen und sich gegenseitig beeinflussen können.

Bei sehr hohen Sicherheitsstandards besteht die Gefahr, dass der Aufwand für die Mitarbeiter enorm groß ist. Wie viel Sicherheit sollte man einführen, ohne das Tagesgeschäft zu stören?

Alexander Kornbrust: Sicherheit bedeutet immer einen bestimmten Aufwand. Auf der anderen Seite sollen die Mitarbeiter in Ruhe arbeiten können. Wenn man das geschickt anstellt, geht das auch. Gerade Oracle bietet hier zahlreiche Features an, ohne die Leute groß zu nerven.

Sind irgendwelche gezielten Hackerangriffe auf Oracle-Datenbanken bekannt?

Alexander Kornbrust: Die gibt es schon, wobei die meisten Angriffe immer von innerhalb des Unternehmens kommen. Wenn also ein Mitarbeiter eine Woche vor seiner Kündigung eine Kopie der Produktionsdatenbank auf seinen Arbeitsplatzrechner macht oder ein Benutzer Daten anschaut, die ihn nichts angehen, dann ist das immer sehr verdächtig. Es ist in jedem Fall ratsam, mit entsprechenden Tools alle Datenzugriffe zu protokollieren und diese entsprechend zu untersuchen. Hackerangriffe von außen hingegen gehen in den meisten Fällen über die Anwendung, häufig durch SQL-Injection.

Worauf ist bei den Protokollen der Datenzugriffe besonders zu achten?

Alexander Kornbrust: Wenn der Login eines normalen Benutzers einmal schiefläuft, hat er sich in der Regel bei der Eingabe seines Passworts vertippt. Wenn allerdings der Login eines bestimmten Users immer wieder erfolglos ist oder ein Login zu ungewöhnlichen Zeiten stattfindet, lohnt es sich schon, dem nachzugehen. Je genauer man das Verhalten seiner Benutzer kennt, desto eher lassen sich Anomalien aufspüren. Ein oft unterschätztes Tool ist der Oracle-Error-Trigger. Da die meisten SQL-Injection-Angriffe mit einem Fehler beginnen, kann man bei bestimmten Fehlermeldungen sehr schnell auf einen Angriff schließen und beispielsweise die IP-Adresse blocken.

"Entsprechende Kenntnisse über die Datenbankzugriffe sind unentbehrlich ..."

Wer im Unternehmen sollte diese Aufgabe übernehmen?

Alexander Kornbrust: Leider werden diese Audit-Daten aufgrund mangelnden Know-hows viel zu selten analysiert. Entsprechende Kenntnisse über die Datenbankzugriffe sind unentbehrlich, sodass in jedem Fall ein DBA hinzugezogen werden sollte. Eine andere Sache in diesem Zusammenhang ist die, dass beim Erkennen eines unzulässigen Zugriffs die Daten meist schon weg sind. Hier sind unbedingt automatische Vorgehensweisen einzurichten, um schnell genug auf einen Angriff reagieren zu können.

Wie gehen Sie vor, wenn Sie zu einem Unternehmen gerufen werden, das Sicherheitsprobleme hat?

Alexander Kornbrust: Ich schaue mir immer zuerst alle Log-Daten an, um bestimmte Muster zu finden. Dabei helfen mir auch eigenentwickelte Tools.

Am 25. Mai tritt die EU-Datenschutz-Grundverordnung in Kraft. Wie schätzen Sie diese ein?

Alexander Kornbrust: Die EU-Datenschutz-Grundverordnung ist sehr interessant und wird vielfach unterschätzt. Jeder Bürger kann durch Nachfragen herausfinden, wo personenbezogene Daten von ihm gespeichert sind. Auf der anderen Seite deuten viele Unternehmen und Organisationen diese Verordnung so um, als würde es nur um Kundendaten gehen, und betrachten nur einige ausgewählte Datenbanken, während in der Realität oft Hunderte von Tabellen davon betroffen sind. Viele Unternehmen sind mit der Umsetzung zu spät gestartet und haben auch die Komplexität unterschätzt, wo überall Kundendaten abgespeichert sind. Zudem hat meist jede Abteilung im Unternehmen eine andere Definition von Kundendaten.

Auf welche Punkte sollte ein Unternehmen bei der Umsetzung der EU-Datenschutz-Grundverordnung besonderen Wert legen?

Alexander Kornbrust: Der erste Schritt ist die exakte Festlegung, was unter personenbezogenen Daten zu verstehen ist. Danach sollte man am besten automatisiert in allen strukturierten Datenbanken suchen.

Ist die EU-Datenschutz-Grundverordnung ein Schritt in die richtige Richtung?
Alexander Kornbrust: Ja, in jedem Fall, weil sie dazu führt, dass die Unternehmen und Organisationen in Zukunft eine sauberere Architektur bezüglich der Speicherung von Kundendaten erstellen können. Die EU-Datenschutz-Grundverordnung kommt nach meiner Ansicht allerdings zu früh, weil die ganzen Zielvorgaben in der vorgegebenen Zeit kaum realisierbar sind. Große Probleme sehe ich vor allem in der Änderung der Daten, weil in vielen Anwendungen dafür gar keine Möglichkeit vorgesehen ist. Der Aufwand für diese Umstellung ist enorm.

"Kritische Daten sind in jedem Fall beim Auslagern in die Cloud zu verschlüsseln ..."

Worauf sollte ein Unternehmen achten, wenn es seine Daten in die Cloud verlagert?
Alexander Kornbrust: Ich bin kein großer Anhänger der Cloud, weil die Unternehmen hier zu viel aus der Hand geben. Wenn eine Firma einen Cloud-Anbieter kontaktiert, empfehle ich, als Erstes die Information zu bekommen, wie viele Administratoren Zugriff auf die Daten in der Cloud haben, wie viele auf das Betriebssystem und an welchem Ort diese Administratoren sitzen. Die Antwort darauf lässt bereits erste Rückschlüsse auf die Sicherheit der Daten zu. Sollte es zu einem Vertragsabschluss kommen, ist in jedem Fall die Security als Service Level Agreement aufzunehmen. Dazu zählt beispielsweise, wie gepatcht wird oder wie die Audit-Logs behandelt werden und ob man diese Auswertung auch zu sehen bekommt. Kritische Daten sind in jedem Fall beim Auslagern in die Cloud zu verschlüsseln. Ein weiteres Problem entsteht, wenn man die Provider wechseln möchte, was ohne Downtime kaum möglich ist. Zudem stellt sich die Frage, ob der alte Provider dann die Daten löscht beziehungsweise bei seiner Backup-Strategie überhaupt dazu in der Lage ist.

Manche Cloud-Provider sagen, die Sicherheit der Daten in der Cloud sei höher als On-Premises auf den eigenen Rechnern?
Alexander Kornbrust: Diese Aussage würde ich jetzt so nicht unterschreiben. Auf dem Papier kann natürlich jeder Provider sagen, dass er diesen und jenen Standard erfüllt, aber in der Praxis lässt es die große Menge der Systeme gar nicht zu, eine große Sicherheit zu erreichen. Ein gut geschultes Datenbank-Team im Unternehmen wird in jedem Fall eine bessere Qualität liefern. Natürlich bietet die Cloud beispielsweise für den schnellen Test eines neuen Systems Vorteile, aber das würde ich nur mit Dummy-Daten empfehlen.

Welche Rolle spielt Oracle als Cloud-Provider?
Alexander Kornbrust: Oracle macht das schon gut und professionell.

Zur Person: **Alexander Kornbrust**

Alexander Kornbrust ist Gründer und Geschäftsführer der Red-Database-Security GmbH, einer auf Datenbank-Sicherheit spezialisierten Sicherheitsfirma. Er arbeitet seit dem Jahr 1992 mit Oracle-Produkten (Datenbank, Application Server, Entwicklungstools) und hat in der Zeit mehr als fünfhundert Sicherheitslücken in Oracle-Produkten an den Hersteller gemeldet. Alexander Kornbrust spricht regelmäßig auf international bekannten Konferenzen und ist Co-Author des Buchs „SQL Injection Attacks and Defense“. Seit einigen Jahren hat er sich neben Datenbank-Sicherheit auf die Implementierung von großen Audit-Systemen und automatische Daten-Klassifikation spezialisiert. Neben der Entwicklung von Software-Lösungen (für Datenk-Kassifikation, DSGVO-Werkzeuge etc.) führt die Red-Database-Security GmbH Security-Audits von relationalen Datenbanken durch. Im Zuge der neuen EU-Datenschutz-Grundverordnung hilft die Firma Kunden bei der Implementierung von Datenschutz-relevanten Prozesses mithilfe von Software und Automation.

Wie schätzen Sie das Oracle-Cloud-Rechenzentrum in Deutschland ein?
Alexander Kornbrust: Ich hatte noch nicht die Gelegenheit, das Rechenzentrum anzuschauen, sehe es aber in jedem Fall als Vorteil, wenn der Provider in Deutschland sitzt. Aber auch hier ist es wichtig, Auskunft darüber zu erhalten, ob die Systeme auch vor Ort und nicht von irgendwo anders auf der Welt administriert werden.

LOGICALIS
Business and technology working as one

Ihr digitaler Wegbereiter

- Clevere Lösungen für Networking, Collaboration & Security
- Durchgängiges Application & Information Management
- Zukunftsweisende Data Center Technologien & nahtlose Cloud Integration

www.logicalis.de

Oracle 18c mit neuen Datenbank-Security-Features

Norman Sibbing, ORACLE Deutschland B.V. & Co. KG

Oracle stellt mit der Änderung der Release-Strategie, beginnend im Jahr 2018, jedes Jahr neue Funktionalitäten zur Verfügung. Das aktuelle Release 18c bietet auch hinsichtlich Datenbank-Security ein paar Neuerungen. Der Artikel zeigt die interessantesten neuen Security-Features anhand von Beispielen.

Sämtliche Lisitings zu dem Artikel finden Sie als Datei unter dem folgenden Link:

www.doag.org/go/red_stack/201804/sibbing/listings

Schema ohne Passwort

Warum muss eigentlich jedes Schema in einer Oracle-Datenbank eine Authentifizierungsmethode haben? Es gibt doch auch Anwendungsfälle, bei denen es nur darauf ankommt, Datenbank-Objekte abzulegen. Ein schlichter Objekt-Container würde dabei ausreichen, ohne dass die Möglichkeit besteht, sich direkt dagegen verbinden zu können. Bisher musste bei der Erstellung eines Schemas zwingend eine Authentifizierungsmethode angegeben werden. Folgende Methoden stehen zur Verfügung:

- *Authentifizierung durch das Betriebssystem (OS-Authentifizierung, Kerberos, PKI)*
  CREATE USER Benutzer IDENTIFIED EXTERNALLY;

- *Authentifizierung durch die Datenbank (Passwort)*
  CREATE USER Benutzer IDENTIFIED BY 'Geheim';

- *Authentifizierung über ein Identity Management System (Enterprise User Security)*
  CREATE USER Benutzer IDENTIFIED GLOBALLY;

Doch was ist, wenn man keine Authentifizierung für dieses Schema haben möchte oder darf? Dafür gab es bislang nur Workarounds. Einer ist die Verwendung eines sehr langen und kryptischen Passworts „-Jm44!!j2h(7hsdtxl*psub+JTDl-„, das dann auch nach der Vergabe direkt wieder vergessen werden sollte. Eine andere Möglichkeit bestand darin, über die „CREATE USER HR IDENTIFIED BY VALUES"-Klausel einen ungültigen HASH-Wert anzugeben. Diese Variante geht allerdings ab der Datenbank-Version 12c nicht mehr, da hier der HASH-Wert überprüft wird. Wird es dennoch versucht, erhält man die Fehlermeldung „ORA-02153: invalid VALUES password string". Auch das Sperren des Schemas ist keine zufriedenstellende Lösung, da ein gesperrtes Schema kein Proxy-Log-in mehr zulässt.

Zum Beispiel sollte auf ein Applikationsschema nicht direkt zugegriffen werden dürfen – weder durch einen Applikationsserver noch durch einen Entwickler oder Datenbank-Administrator. Für den Zugriff der Applikation auf die Applikationsobjekte ist es empfohlen, einen separaten technischen Applikationsbenutzer in der Datenbank anzulegen, der durch Rollen und gegebenenfalls Synonyme Zugriff auf das Applikationsschema erhält. Nur so ist eine Steuerung der Zugriffsrechte möglich. In diesem Fall ist es unerheblich, ob das Applikationsschema gesperrt oder mit einem unbekannten Passwort geschützt ist oder gar kein Passwort besitzt. Bei der letzteren Variante (ohne Passwort) ist, wie erwähnt, kein Proxy-Log-in möglich.

Für Entwickler und Datenbank-Administratoren bietet sich hingegen der Einsatz eines Proxy-Benutzers an. Der wesentliche Vorteil dabei zeigt sich beim Versuch, Datenbank-Benutzer zu personalisieren, obwohl sie letztendlich Shared-Schemata wie „SYSTEM" oder das Applikations-Schema verwenden. Neben dem vereinfachten Rollen-Management – die Rollen werden im Prinzip vererbt – wird der reale Datenbank-Benutzer im Auditing-Trail eingetragen. Und genau darauf zielt das neue 18c-Datenbank-Feature „Schema Only Accounts" ab. Damit besteht die Möglichkeit, keine Authentifizierungsmethode angeben zu müssen. Mit der Klausel „NO AUTHENTICATION" beim „CREATE USER"- und „ALTER USER"-Statement wird mit „•CREATE|ALTER USER Benutzer NO AUTHENTICATION;" ein Schema Only Account erstellt. Das funktioniert fast mit jedem Datenbank-Benutzer. Ausnahme: Die Klausel „NO AUTHENTICATION" ist bei Datenbank-Benutzern mit Administrations-Privilegien wie „SYSDBA", „SYSOPER", „SYSBACKUP", „SYSKM", „SYSASM", „SYSRAC" und „SYSDG" sowie natürlich bei der Verwendung von Datenbank-Links nicht zulässig.

Das folgende Beispiel zeigt das Feature für die Datenbank-Administration mit dem User „SYSTEM". Aus Sicherheits- und Compliance-Gründen sollte dieser nicht zur Administration verwendet werden. Der Hauptgrund dafür ist, dass entsprechende Aktivitäten keiner realen Person zugeordnet werden können und demzufolge nicht nachvollziehbar sind. Hier bietet sich das neue Feature sehr gut an. Zunächst wird der „SYSTEM"-Benutzer zum Schema Only Account geändert. Er besitzt also kein Passwort mehr und folglich kann sich niemand direkt mit ihm anmelden *(siehe Listing 1)*. Nun wird ein personalisierter Benutzer mit eigenem Passwort angelegt. Der neue Benutzer erhält maximal das „Create Session"-Privileg; zudem ist es ihm erlaubt, den SYSTEM-Benutzer als Proxy-Benutzer zu verwenden *(siehe Listing 2)*. Mehr ist im Prinzip nicht zu tun. Die Anmeldung als personalisierter DBA mit den Rechten des SYSTEM-Benutzers erfolgt dann entsprechend einem Proxy-Log-in *(siehe Listing 3)*.

Schema Only Accounts bieten also eine geniale Möglichkeit, Datenbank-Benutzer auf einfache Weise zu personalisieren. Es werden hiermit gleich mehrere Themen adressiert:

- Benutzer mit „NO AUTHENTICATION" sind nicht mehr direkt nutzbar
- Kein Passwort erforderlich
- Kein Passwort-Profil mehr notwendig
- Jeder auf diese Art personalisierte DBA oder Entwickler verwaltet sein eigenes Passwort
- Im Audit-Trail ist der reale Benutzer enthalten, trotz Verwendung des Shared Account

Das Feature steht in allen Editionen ab Oracle-Datenbank 18c ohne zusätzliche Lizenzen zur Verfügung.

Unified Auditing in SYSLOG und Windows Event Viewer

Die Überwachung spezieller Aktivitäten in IT-Systemen wie Netzwerken, Datenbanken und Betriebssystemen gehört zu den wichtigsten Sicherheitsmaßnahmen in der IT-Sicherheit. Hierbei geht es im Wesentlichen immer um den Nachweis darüber, wer innerhalb der IT-Infrastruktur was wann wo und womit gemacht hat. Die meisten aktiven Komponenten einer IT-Infrastruktur sind in der Lage, bei richtiger Konfiguration Aktivitäten in den von ihnen zur Verfügung gestellten Diensten zu protokollieren. Die Herausforderung ist dabei, die Aktivitätsprotokolle aus diesen unterschiedlichen IT-Komponenten zu konsolidieren, um sie miteinander korrelieren zu können.

Eine dieser IT-Komponenten ist die Oracle-Datenbank; sie war schon immer in der Lage, Aktivitäten zu protokollieren. Dabei werden die durch das Oracle-Auditing generierten Daten in Tabellen und Betriebssystem-Dateien geschrieben. Es besteht auch die Möglichkeit, Datenbank-Protokolldaten in das Unix-SYSLOG beziehungsweise in den Microsoft Event Viewer zu schreiben. Das funktioniert noch bis heute – zumindest für das traditionelle Oracle-Datenbank-Auditing. Mit der Version 12c wurde ein neues Auditing eingeführt, das Oracle Unified Auditing, ein Datenbank-Auditing neben dem traditionellen (alten) Oracle Auditing. Oracle Unified Auditing ist moderner, flexibler und sicherer als das klassische Oracle-Datenbank-Auditing. Hierzu existiert bereits ein Tipp, der das Thema genauer darstellt. Das, was Oracle Unified Auditing bisher nicht konnte, war, die Auditdaten in das Unix-SYSLOG beziehungsweise in den Microsoft Event Viewer zu schreiben.

Diese fehlende Integration führte teilweise dazu, Oracle Unified Auditing nicht zu verwenden, obwohl es wesentlich besser zu kontrollieren ist. Ein Grund dafür ist,

SYSLOG/Event Viewer Name	Spalten im UNIFIED_AUDIT_TRAIL	Beschreibung
TYPE	AUDIT_TYPE	Audit-Typ (Standard, RMAN etc.)
DBID	DBID	Datenbank-ID
SESID	SESSION_ID	Session-ID
CLIENTID	CLIENT_IDENTIFIER	Identifikation des Session-Clients
ENTRYID	ENTRY_ID	Laufende Audit-ID pro Session
STMTID	STATEMENT_ID	Laufende Statement-ID pro Session
DBUSER	DB_USERNAME	Datenbank-Benutzer, dessen Aktionen überwacht wurden
CURUSER	CURRENT_USERNAME	Effektiver Benutzer
ACTION	ACTION	Action-Code (Statement Typ)
RETCODE	RETURN_CODE	ORA-Fehlercode
SCHEMA	OBJECT_SCHEMA	Name des Objekt-Schemas
OBJNAME	OBJECT_NAME	Name des betroffenen Objekts

Tabelle 1: Spalten-Mapping zwischen SYSLOG und Unified Audit Trail

dass viele Kunden Werkzeuge zur Überwachung von Aktivitäten sogenannte „System Information and Event Management"-Tools (SIEM) einsetzen, um die Protokolldaten sämtlicher IT-Komponenten und Services zentral aus dem Unix-SYSLOG beziehungsweise Microsoft Event Viewer zu sammeln, Oracle Unified Auditing aber dort die Daten nicht speichern konnte. Diese sinnvolle Integration ist nun ab der Oracle-Datenbank 18c für Unified Auditing möglich.

Einen kleinen Wermutstropfen gibt es: Nicht alle Informationen, die mit Unified Auditing gesammelt werden, werden auch ins SYSLOG beziehungsweise in den Event Viewer geschrieben. *Tabelle 1* zeigt die Informationen, die zur Verfügung stehen.

Um die Daten des Unified Auditing in das SYSLOG oder in den MS Event Viewer übertragen zu bekommen, sind lediglich zwei Konfigurationen notwendig, eine in der Oracle-Datenbank durchgeführt und eine im Betriebssystem. Zunächst wird die Datenbank vorbereitet. Der ab Oracle DB 18c verfügbare Parameter „UNIFIED_AUDIT_SYSTEMLOG" wird je nach Ziel (SYSLOG oder MS Event Viewer) wie folgt eingestellt:

- *SYSLOG*
  Auswahl der „SYSLOG"-Facility „user: user-level messages" oder „local[0-7]" sowie SYSLOG Severity Level (NOTICE, INFO, DEBUG, WARNING, ERR, CRIT, ALERT, und EMERG)

- *MS Event Viewer*
  TRUE (statt des Defaults FALSE)

Im *Listing 4* (ein UNIX-System) wird der Parameter „UNIFIED_AUDIT_SYSTEMLOG" auf „LOCAL6.NOTICE" gesetzt. Ist das erledigt, muss der System-Administrator („Root") die Konfiguration des SYSLOG durchführen. Im Beispiel wird die Einstellung „local6.notice" mit der LOG-Datei „/var/log/audit_oracle.log" gekoppelt. Dies wird in der Konfigurationsdatei „/etc/rsyslog.conf" beziehungsweise „/etc/syslog.conf" eingetragen. Nach der Änderung nur noch den SYSLOG-Daemon durchstarten und fertig ist die Konfiguration im Betriebssystem *(siehe Listing 5)*. Zum Schluss die Datenbank durchstarten und Einstellungen überprüfen *(siehe Listing 6)*. Nun sollte auch die entsprechende Auditdatei „/var/log/audit_oracle.log" vorhanden sein *(siehe Listing 7)*. Ab sofort werden die oben beschriebenen Audit-Informationen, neben der Speicherung im Unified Audit Trail, auch ins SYSLOG geschrieben *(siehe Listing 8)*. Das Ergebnis lässt sich durch einen privilegierten Betriebssystembenutzer (Root) überprüfen *(siehe Listing 9)*.

In einer Oracle-Multitenant-Umgebung ist bei Bedarf der Parameter „UNIFIED_AUDIT_SYSTEMLOG" in jeder Pluggable Database zu setzten. Hier besteht auch die Möglichkeit, unterschiedliche SYSLOG-Einstellungen zu verwenden, etwa dieselben wie bei der Container-Datenbank *(siehe Listing 10)*. In Beispiel von *Listing 11* schreiben jetzt beide Datenbanken (CDB und PDB1) in dieselbe Auditdatei.

Wie bereits erwähnt, werden nur bestimmte Informationen ins SYSLOG geschrieben. Diese Informationen reichen aber aus, um den gesamten Auditeintrag zu identifizieren. Möchte man zum Beispiel wissen, welche ausgeführte Aktion sich hinter dem Action-Code „227" verbirgt, lässt sich das über die Datenbank-View „audit_actions" abfragen *(siehe Listing 12)*. Detaillierte Informationen über die protokollierte Aktivität stehen im Unified Audit Trail der entsprechenden Datenbank *(siehe Listing 13)*.

Der Action-Code „ACTION:100" zeigt bereits, dass es sich hier um einen Anmeldeversuch (LOGON) handelte. Der Return-Code „RETCODE:1017" gibt an, dass dieser missglückt war. Um nun alle Informationen dieser Aktivität zu erhalten, benötigen wir die Session-ID „SESID:2439550631" und die Statement-ID „STMTID:1", um den Unified Audit Trail der Datenbank abzufragen *(siehe Listing 14)*.

Mit dieser Unified-Auditing-SYSLOG-Integration ist eine Lücke geschlossen, die so manchen davon abgehalten hat, Unified Auditing einzusetzen. Vermutlich sind es dem einen oder anderen noch zu wenig Audit-Informationen, die ins SYSLOG geschrieben werden (wie „SQL_TEXT"). Auch aus diesem Grund ist und bleibt es unumgänglich, die gesamten Informationen aus den Unified Audit Trails der Datenbanken zentral einzusammeln. Eine gute Lösung dafür ist die Verwendung von Oracle Audit Vault und Database Firewall. Im Übrigen besitzen Oracle Audit Vault und Database Firewall eine hervorragende SYSLOG-Integration, die sich über ein Regelwerk gut steuern lässt.

Dieses Feature steht in allen Editionen ab Oracle-Datenbank 18c ohne zusätzliche Lizenzen zur Verfügung.

Verschlüsselung von Anmelde-Informationen im Data Dictionary

Immer häufiger kommen sogenannten „Datenbank-Schwachstellen-Scanner" (Database Vulnerability Scanner) zum Einsatz, um potenzielle Sicherheitsrisiken in Datenbanken aufzuspüren. Die durch diese Werkzeuge entdeckten Schwachstellen und Fehlkonfigurationen lassen sich meist durch entsprechende Sicherheitsmaßnahmen mildern beziehungsweise komplett beheben. Es gibt allerdings auch Dinge, die sich nicht ohne Weiteres mildern oder beheben lassen, da sie systembedingt sind oder waren. Dazu gehören die im Oracle-Dictionary gespeicherten Anmelde-Informationen für Datenbank-Links und -Scheduler. Die hier verwendeten Kennwörter lassen sich über die Data-Dictionary-Tabellen „SYS.LINK$" und „SYS.SCHEDULER$_CREDENTIAL" auslesen. Zur Demonstration dieses Features wird zunächst ein Database Link angelegt *(siehe Listing 15)*. Nun folgt die Abfrage der „LINK$"-Tabelle, um das Kennwort zu erhalten *(siehe Listing 16)*.

Man darf sich von der Darstellung des „PASSWORDX"-Werts nicht täuschen lassen. Er sieht zwar verschlüsselt aus, ist es aber nicht. Zur Speicherung der Anmelde-Informationen von Datenbank-Links und Scheduler-Jobs sind die Kennwörter vielmehr verschleiert (obfuscated) gespeichert. Dabei werden Informationen durch einen rückrechenbaren Algorithmus augenscheinlich unkenntlich gemacht. Es handelt es sich allerdings nicht um einen Verschlüsselungs-Algorithmus im klassischen Sinne, sondern um ein vom entsprechenden Entwickler konzipiertes Verfahren. Ist also das Verfahren einem Dritten bekannt, kann er die über die Data-Dictionary-Tabellen „SYS.LINK$" und „SYS.SCHEDULER$_CREDENTIAL" ausgelesenen Kennwörter lesbar machen. Aus diesem Grund sollte der Zugriff auf diese sensiblen Tabellen weiterhin sehr restriktiv gehandhabt bleiben, obwohl es ab der Version 18c die Möglichkeit gibt, diese sensiblen Informationen mit einem echten Verschlüsselungs-Algorithmus AES256 (Advanced Encryption Standard) zu verschlüsseln. In einer Standard-Installation der Datenbank 18c sind diese Werte weiterhin nur verschleiert.

Verschlüsselung der Anmelde-Informationen von Datenbank-Links und Scheduler-Jobs

Dieses Feature verwendet Funktionalitäten der Transparent Data Encryption (TDE) und steht in allen Editionen ab Oracle 18c zur Verfügung. Eine Advanced-Security-Option-Lizenz ist hier jedoch nicht erforderlich. Der „COMPATIBLE"-Parameter der Datenbank muss dabei auf mindestens 12.2.0.2 gesetzt sein.

Die Verschlüsselung der Anmelde-Informationen in den „SYS.LINK$"- und „SYS.SCHEDULER$_CREDENTIAL"-Tabellen ist ähnlich der Verschlüsselung von Spalten und Tablespaces mittels TDE. Es existiert jeweils ein Schlüsselpaar, bestehend aus einem Master Encryption Key (MEK) und jeweils einem Data Encryption Key (DEK) für die Tabellen „SYS.LINK$" und „SYS.SCHEDULER$_CREDENTIAL". Anders als beim MEK, der extern gespeichert ist, sind die DEKs innerhalb der Datenbank verschlüsselt gespeichert. Der MEK ist zur Ver- und Entschlüsselung der DEKs erforderlich. Mit den so entschlüsselten DEKs können dann die Anmelde-Informationen in den „SYS.LINK$"- und „SYS.SCHEDULER$_CREDENTIAL"-Tabellen gelesen werden. Wie beschrieben, benötigt die Datenbank für diese Funktionalität einen MEK. Das ist derselbe, der beim Einsatz von Transparent Data Encryption erforderlich ist.

Achtung: Wer Transparent Data Encryption bereits einsetzt, muss die nächsten Schritte bis einschließlich der Erstellung des Master Encryption Key überspringen, da für die Datenbank bereits ein Keystore und ein Master Encryption Key existieren. Beim Oracle-Datenbank-Cloud-Service in der Oracle Public Cloud sind der Keystore und der Master Encryption Key ebenfalls bereits vorhanden und es sind keine weiteren Vorbereitungen durchzuführen. Trifft das alles nicht zu, ist zunächst ein Keystore für die Speicherung des Master Encryption Key zu erstellen. Im Standard ist der Keystore eine „PKCS12"-Datei („ewallet.p12"), die in einem Verzeichnis

auf dem Datenbank-Server gespeichert ist. In der Regel liegt der Keystore unter „$ORACLE_BASE/admin/$ORACLE_SID/wallet". Das „wallet"-Verzeichnis ist vorab anzulegen. Es besteht natürlich die Möglichkeit, ein alternatives Verzeichnis (etwa „/u01/oracle/db/orcl/wallet") zu verwenden. Dieses ist in der „SQLNET.ORA" des Datenbank-Servers anzugeben *(siehe Listing 17)*.

Nachdem die „SQLNET.ORA" entsprechend angepasst ist, lässt sich der Keystore erstellen und öffnen. Dafür ist ein „SYSKM"- oder „SYSDBA"-Administrations-Privileg erforderlich *(siehe Listing 18)*. Anschließend kann der Master Encryption Key erstellt werden *(siehe Listing 19)*. Das Ergebnis lässt sich über die View „V$ENCRYPTION_KEYS" überprüfen *(siehe Listing 20)*. Nun sind alle Voraussetzungen zum Verschlüsseln der Anmelde-Information in der „SYS.LINK$"- und „SYS.SCHEDULER$_CREDENTIAL"-Tabelle gegeben.

Damit alle aktuellen und alle zukünftigen Anmelde-Informationen verschlüsselt gespeichert werden, ist einmalig ein neues „ORACLE DDL"-Statement auszuführen. Ab der Oracle Datenbank 18c steht dafür das Statement „ALTER DATABASE DICTIONARY" zur Verfügung. Die Ausführung des Statements ist überschaubar; es existieren lediglich drei Ausführungsvarianten *(siehe Abbildung 1)*.

Das neue Statement lässt sich nur mit „SYSKM"-Administrations-Privilegien ausführen. Dieser Befehl verschlüsselt nun alle Anmelde-Informationen in den „SYS.LINK$"- und „SYS.SCHEDULER$_CREDENTIAL"-Tabellen *(siehe Listing 21)*. Das Ergebnis kann der „SYSDBA" durch Abfrage der „SYS.LINK$"-Tabelle überprüfen *(siehe Listing 22)*. Vergleicht man nun die Werte von „PASSWORDX" mit dem originalen Wert (ohne Verschlüsselung), lässt sich ein Unterschied erkennen *(siehe Listing 23)*.

Besteht ein Grund, die Anmelde-Informationen in den Tabellen „SYS.LINK$" und „SYS.SCHEDULER$_CREDENTIAL" mit einem neuen Data Encryption Key neu zu verschlüsseln, kann dies durch einen einfachen Befehl erfolgen *(siehe Listing 24)*. Durch eine wiederholte Abfrage der „SYS.LINK$"-Tabelle lässt sich die Neu-Verschlüsselung überprüfen *(siehe Listing 25)*. Wenn man die beiden „PASSWORDX"-Werte vergleicht, erkennt man die Unterschiede *(siehe Listing 26)*. Zu beachten ist, dass der Keystore der Datenbank geöffnet sein muss, bevor entsprechende Database-Links beziehungsweise Scheduler-Jobs benutzt oder gestartet werden können *(siehe Listing 27)*. Ist der Keystore geschlossen, erscheint eine Fehlermeldung bei der Verwendung eines Database-Links *(siehe Listing 28)*. Nach Öffnung des Keystore durch den „SYSDBA" oder „SYSKM" funktioniert alles wieder wie gewohnt *(siehe Listing 29)*.

Um alles wieder rückgängig zu machen, reicht ein Befehl *(siehe Listing 30)*. Beim Versuch, zum Beispiel einen Database-Link zu verwenden, erscheint eine Fehlermeldung *(siehe Listing 31)*. Dies lässt sich durch das Neusetzen des Kennworts der Database-Links korrigieren *(siehe Listing 32)*. Dieses Feature schließt eine Sicherheitslücke, die ständig bei Datenbank-Sicherheitsüberprüfungen als Risiko angezeigt wurde und jetzt mit der 18c geschlossen ist. Es steht in allen Editionen ab Oracle-Datenbank 18c ohne zusätzliche Lizenzen zur Verfügung.

Keystore pro Pluggable Database

Keystores werden zur sicheren Speicherung von Krypto-Schlüsseln jeglicher Art verwendet. Auch die Oracle-Datenbank verwendet einen Keystore zur Speicherung des externen Schlüssels, sobald Transparent Data Encryption (TDE) zum Verschlüsseln der Oracle-Datenbank verwendet wird. Neben der Speicherung des TDE Master Key werden weitere Keystores von der Oracle-Datenbank verwendet – zum Beispiel zur Speicherung von Passwörtern, Zertifikaten und anderen „Geheimnissen". Als Standard-Keystore wird das Oracle Wallet verwendet. Bisher war es so, dass jede verschlüsselte Datenbank einen dedizierten Keystore verwenden musste. Dies galt auch für eine Multitenant-Umgebung, in der jede Pluggable Database (PDB) den gemeinsamen Keystore der Container Database (CDB) benutzte. Das widersprach der strikten Trennung zwischen den PDBs untereinander, aber auch zwischen der CDB und den PDBs. Es war also nicht möglich, dass ein PDB-Administrator beziehungsweise der Mandant über seinen eigenen Keystore verfügte. Dabei ist der Zugriff auf Krypto-Schlüssel die „Achillessehne der Datensicherheit".

Derjenige, der Zugriff auf die Schlüssel hat, hat auch Zugriff auf die Daten. Mit der Oracle-Datenbank 18c wird diese Problematik nun gelöst: Jede PDB kann ihren eigenen Keystore besitzen. Das Feature – Keystore for Each Pluggable Database – ermöglicht es in einer Multitenant-Umgebung, dass jede PDB einen eigenen Keystore verwenden kann. Wird diese Eigenschaft genutzt, erhalten die Administratoren der PDB die volle alleinige Kontrolle über ihren Keystore. Hieraus folgt aber auch Verantwortung: Sollte der PDB-Administrator zum Beispiel das Passwort seines Keystore vergessen, gibt es niemanden mehr, der ihm helfen kann.

Um das Feature zu verwenden, wurden zwei neue Parameter und Keystore-Eigenschaften eingeführt, die in den folgenden Abschnitten erklärt werden.

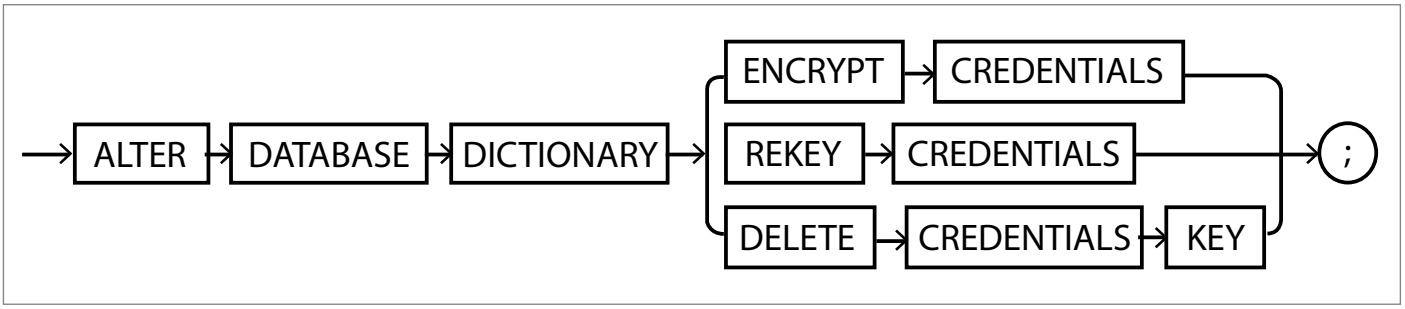


Abbildung 1: Ausführungsvarianten

Der Datenbank-Initialisierungs-Parameter „WALLET_ROOT“

Mit der Version 18c wurde der Datenbank-Initialisierungs-Parameter „WALLET_ROOT“ eingeführt. Dieser ersetzt den „SQLNET.ORA“-Parameter „SQLNET.ENCRYPTION_WALLET_LOCATION“, der ab 18c abgekündigt wurde, aber noch eine gewisse Zeit funktioniert. Ist der Parameter nicht gesetzt, bezieht die Datenbank das Wallet-Verzeichnis weiterhin über den „SQLNET.ENCRYPTION_WALLET_LOCATION“-Parameter aus der „SQLNET.ORA“. „WALLET_ROOT“ wird in der CDB gesetzt. Der Parameter beschreibt das Keystore-Basis-Verzeichnis aller der CDB zugeordneten PDBs. Darauf aufbauend dient die eindeutige „PDB-GUID“ als weiteres Verzeichnis zur Trennung der Keystores. *Listing 33* zeigt die Struktur der „/wallet-root/pdb-guid/tde“.

Der Parameter „WALLET_ROOT“ steht auch in der klassischen Oracle-Datenbank-Betriebsart der sogenannten „NON-CDB“ zur Verfügung. Im Standard ist diese nicht gesetzt und es handelt sich um einen statischen Parameter. Demzufolge ist die Datenbank nach dem Setzen neu zu starten. Das folgende Beispiel verwendet das Verzeichnis „/home/oracle/keystore_root“ als Keystore-Basis-Verzeichnis. Dafür wurde eine Oracle Database Cloud Service Enterprise Edition verwendet *(siehe Listing 34)*.

Der Datenbank-Initialisierungs-Parameter „TDE_CONFIGURATION“

Zudem wurde der Initialisierungs-Parameter „TDE_CONFIGURATION“ eingeführt. Dieser ist nur in einer Oracle Database Enterprise Edition auf Engineered Systems und ab Oracle Database Cloud Service Enterprise Edition aufwärts verwendbar. Bevor dieser Parameter gesetzt werden kann, muss das Keystore-Basis-Verzeichnis mittels „WALLET_ROOT“- Parameter gesetzt sein. „TDE_CONFIGURATION“ sorgt dafür, dass die PDBs wissen, welche Art von Keystore verwendet werden soll. Zur Auswahl stehen folgende Typen von Keystores:

- Das klassische Wallet FILE
- Hardware Security Modul (HSM)
- Oracle Key Vault (OKV)

Gesetzt wird dieser Parameter sowohl in der CDB als auch in jeder PDB. Das Beispiel im *Listing 35* verwendet den Keystore-Typ „FILE“, also ein Wallet als Keystore. Bei einer Multitenant-Umgebung mit zwei Pluggable Databases „PDB1“ und „PDB2“ (ohne „PDB$SEED“) sieht das wie in *Listing 36* aus. Die CDB-Root („CON_ID 1“) ist zu diesem Zeitpunkt immer im Keystore-Modus „NONE“ und alle weiteren PDBs („CON_ID 3“ und „4“), inklusive der „PDB$SEED“ („CON_ID 2“), sind standardmäßig auf „UNITED“ festgelegt. Diese Einstellung bedeutet, dass momentan alle PDBs einen gemeinsamen Keystore verwenden (Standard-Einstellung).

Neu mit 18c ist auch, dass zwei Betriebsvarianten der Keystores möglich sind: ein United- und ein Isolation-Modus. Im United-Modus sind alle TDE-Master-Keys wie gewohnt in einem der CDB zugeordneten Keystore gespeichert. Im Isolation-Modus hingegen erhält jede PDB einen eigenen Keystore. Der entscheidende Vorteil ist hier, dass ausschließlich der Administrator der PDB das Passwort des Keystore kennt. Wird der Keystore im United-Modus für die PDBs verwendet, hat im Gegensatz dazu der CDB-Administrator Zugriff auf alle dort gespeicherten TDE-Master-Keys.

Um jetzt einzelne PDBs in den Keystore-Isolation-Modus zu bringen, ist es notwendig, den bereits beschriebenen Parameter „TDE_CONFIGURATION“ in den entsprechenden PDBs zu setzen *(siehe Listing 37)*. Dies kann bei Bedarf auf andere PDBs angewandt werden. Es besteht die Möglichkeit, für jede PDB individuell einen anderen Keystore-Typ zu verwenden *(siehe Listing 38)*. Wie hier zu sehen ist, besitzt jetzt jede PDB ihren eigenen isolierten, File-basierten Keystore. Im Filesystem sieht das dann wie in *Listing 39* aus.

Dieses Feature ist ein konsequenter und wichtiger Schritt in Richtung „Mandantenfähigkeit“. Es erweitert die Möglichkeiten zur Isolierung von PDBs bezüglich des Schlüsselmanagements und bietet dem Mandanten somit eine bessere Zugriffskontrolle auf seine Daten. Dieser Parameter kann nur in einer Oracle Database Enterprise Edition auf Engineered Systems und ab Oracle Database Cloud Service Enterprise Edition aufwärts verwendet werden.

Centrally Managed Users

Centrally Managed Users (CMU) ermöglicht es, dass Datenbank-Benutzer und Rollen jetzt direkt in einem Microsoft Active Directory verwaltet werden können – und das, ohne einen Oracle-LDAP-Server dazwischen zu betreiben. Damit kann ein Benutzer aus dem Microsoft Active Directory exklusiv einem Datenbank-Benutzer zugeordnet sein oder es teilt sich eine Microsoft-Active-Directory-Gruppe einen Benutzer in der Datenbank. Microsoft-Active-Directory-Gruppen können auch direkt entsprechenden Datenbank-Rollen (Global Roles) zugeordnet sein. Die so zentral verwalteten Benutzer können sich durch Passwörter, Kerberos-Tickets und PKI-Zertifikate authentifizieren.

Für die Passwort-Authentifizierung müssen ein sogenannter „Passwortfilter“ auf dem Active Directory Server installiert und das Active Directory Schema erweitert sein. Dazu liefert Oracle ein Konfigurationswerkzeug, um den Kennwortfilter zu installieren und das Active-Directory-Schema zu erweitern.

Insbesondere bei neuen Identity-Management-Projekten ist dies eine gute Möglichkeit, sowohl Komplexität als auch die Betriebskosten im Hinblick auf Wartung und Entwicklung zu verringern. Dieses Feature steht in allen Enterprise-Editionen ab Oracle-Datenbank 18c ohne zusätzliche Lizenzen zur Verfügung.

Fazit

Wie gezeigt, enthält die Oracle-Datenbank 18c einige nützliche Sicherheits-Features. Jedes einzelne davon macht die IT ein wenig sicherer. Es muss allerdings auch eingesetzt werden.

Norman Sibbing
norman.sibbing@oracle.com

Wohin geht Security bei Oracle?

Michael Fischer, ORACLE Deutschland B.V. & Co. KG

Security ist eine der Kompetenzen von Oracle. Bestehende Produkte werden um Sicherheitsfunktionen weiterentwickelt, neue Security-Services geschaffen und Firmen auch im Bereich „Security" akquiriert. Ziel ist die kontinuierliche Verstärkung der Absicherungsmöglichkeiten von Daten und Services. Um Silo-Landschaften entgegenzuwirken, erfolgt eine Vorintegration in eine Art Sicherheitsplattform, die optional genutzt werden kann. Dieser Artikel beschreibt den aktuellen Stand im Technologie-Portfolio von Oracle mit einem kurzen Ausblick in die Zukunft.

„Security needs automation" war eine der Schlüssel-Botschaften von Larry Ellison auf der Oracle Open World 2017. In seiner Keynote hatte er das plakativ mit dem Krieg der Systeme von Hackern gegen Menschen begründet, die ihre Systeme quasi manuell verteidigen. In diesem Wettrüsten kann nur Schritt gehalten werden, wenn auch bei der Verteidigung automatisiert wird. Weitere Gründe für eine Automatisierung sind:

- Mit der stetig wachsenden Zahl von Systemen in On-Premises- und Cloud-Umgebungen steigt der Aufwand, alle diese Umgebungen zu überwachen.
- Durch die zunehmende Intelligenz in Cyber-Attacken im Gegensatz zu den einfachen Datenabgriffen der Vergangenheit reicht die Funktionalität bisheriger Log-Auswertungen nicht mehr aus.
- Mit neuen Regularien wie der EU-Datenschutz-Grundverordnung und auch einem genaueren Blick auf die Einhaltung von Regularien sind mehr Sicherheitsnachweise erforderlich.

Neben der Automatisierung findet sich die Weiterentwicklung im Bereich „Security" in den einzelnen Technologien und in Security-Komponenten wie dem Monitoring. Monitoring-Komponenten haben zwei Entwicklungen Rechnung zu tragen; zum einen müssen Aktivitäten in und Konfigurationen von On-Premises-Systemen und Cloud-basierten Diensten überwacht werden. Zum anderen, bedingt durch immer intelligentere Cyber-Attacken, müssen Aktivitäten über Systeme hinweg und abweichend vom normalen Verhalten erkannt werden.

Laut Gartner-Analysten (siehe unter anderem Gartner CARPA 2017) wird das bisherige Security-Monitoring heute auf den Prüfstand gestellt, klassische „Security Information and Event Management"-Systeme (SIEM) erfüllen diese erweiterten Anforderungen nicht. Andere Umfragen von Sicherheitsverantwortlichen unterstreichen dies, indem die Mehrzahl angibt, mit den vielen separaten Sicherheitstools frustriert zu sein sowie Korrelationen und entsprechende Automatisierungzu vermissen (siehe auch Ponemon Institute Report, Challenges 2017).

Ist das eine rein akademische Betrachtung oder besteht hier tatsächlich Handlungsbedarf? Hilfestellungen bei dieser Überlegung liefern beispielsweise der „Oracle und KPMG Cloud Threat Report 2018", der Maßnahmen darstellt, die aktuell zum Schutz von On-Premises- oder Cloud-basierten Datenbanken eingesetzt werden. Diese zeigen, dass ein stärkerer Schutz durchaus möglich wäre. Die geschilderten Angriffsszenarien sind je nach Branche unterschiedlich, sodass unterschiedliche Schwerpunkte bei der Verteidigung sinnvoll sind. Der Report zeigt auch, dass die Mehrzahl der Kunden eine weitergehende Automatisierung bereits umgesetzt haben oder zumindest planen. Mit Bezug auf den aufgeführten Report sind im Folgenden die Punkte „Automatisierung" und „Security-Technologien" betrachtet.

Automatisierung

Automatisierung hilft bei der Sicherstellung von Security, sowohl beim Betrieb der Systeme als auch beim Incident beziehungsweise Security & Threat Monitoring. Die Automatisierung beim Betrieb lässt sich exemplarisch an der Oracle-Datenbank zeigen. Eine Weiterentwicklung bei der Oracle-Datenbank bezüglich eines möglichst automatisierten Betriebes umfasst automatisiertes Patching und eine automatisierte Aktualisierung beziehungsweise Anpassung verfügbarer Security und Betriebseinstellungen. Unterstützungen dazu gibt es seit Längerem unter anderem mit dem Oracle Enterprise Manager, der Basis-Mechanismen bereitstellt.

Ziel bei der Automatisierung ist es, einen möglichst hohen Grad von Autarkie des Systems zu ermöglichen, die weit über die Basis-Mechanismen hinausreicht. Automatisierung ist keine reine Software-Lösung, sondern umfasst neben der Technologie auch die beteiligten Personen und festgelegten Prozesse. Oracle hat dies mit der Bereitstellung der Autonomous Database erstmalig eingeführt. Aufgabenstellungen, die dabei beispielsweise im Falle von Regularien adressiert werden, sind berücksichtigt. So sind hier ein „least privilege"-Management oder der Zugriff auf Audit-Daten zu nennen.

Bei Autonomous gelten die Rahmenbedingungen der Services, die als Teil der Unternehmenspolicies akzeptabel sein müssen. Die Verantwortlichkeiten des

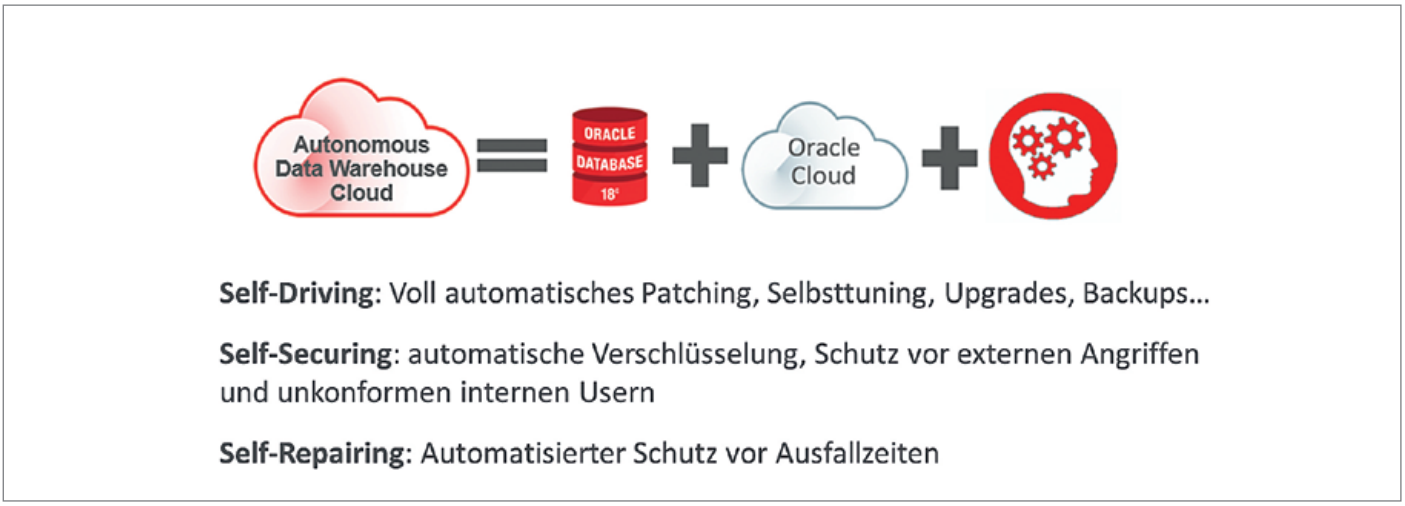


Abbildung 1: Oracle Autonomous Data Warehouse

Kunden sind gemäß den Unternehmensvorgaben umzusetzen, etwa Zugriffsberechtigungen, Schutz der Daten oder die Verarbeitung des Audit-Trails. Weitere Services sind zurzeit Oracle Autonomous Analytics Cloud, Oracle Autonomous Integration Cloud und Autonomous Visual Builder Cloud Service *(siehe Abbildung 1)*.

Die Automatisierung beim Security Monitoring kann durch ein Zusammenspiel verschiedener Oracle-Werkzeuge umgesetzt werden. Es kommen vier Komponenten zum Einsatz, die einzeln verwendet oder untereinander beziehungsweise in 3rd-Party-Systeme integriert werden können. Oracle stellt mit dem Oracle Enterprise Manager seit Längerem ein Werkzeug zum Monitoring der On-Premises- und Cloud-Installationen zur Verfügung. Der hier vorgestellte Ansatz legt andere, Cloud-basierte Werkzeuge von Oracle zugrunde, die weit über den Enterprise Manager hinausgehen. Zentrale Elemente des Monitorings bezüglich Security und Compliance sind:

- Die Überprüfung hinsichtlich der Aktionen in einzelnen Systemen und über Systeme hinweg.
- Eine fortwährende Prüfung der Konfiguration bezüglich der Unternehmens- und Compliance-Vorgaben.

Dieses Monitoring wird meist toolbasiert in SIEM-Systemen durchgeführt und durch ein Network Operation Center/Security Operation Center (NOC/SOC) betrieben. Traditionelle Ansätze sind typischerweise regelbasiert und arbeitsaufwendig. Dies birgt Herausforderungen, denen durch Automatisierung mithilfe von Machine Learning begegnet wird. Damit können viele der bisherigen manuellen Schritte im SOC automatisiert werden, sodass den SOC-Spezialisten mehr Zeit für komplexere Bewertungen und Analysen bleibt.

Automatisierung erfolgt beispielsweise beim Auffinden von Abweichungen durch automatisiertes User and Entity Behavior Analytics (UEBA), beim Aufspüren von Threats beziehungsweise Kill Chains oder bei Vorhersagen oder der Erkennung von allgemeinen Korrelationen und Mustern. Diese Funktionen werden von verschiedenen Oracle-Lösungskomponenten bereitgestellt. Sie bieten vorgefertigte Integrationen zu Systemen und nutzen einen unterliegenden Big-Data-Ansatz, um die beliebig wachsenden Daten in den Griff zu bekommen. Die Komponenten sind im Einzelnen:

- Oracle Management Cloud Security Monitoring Analytics (OMC SMA) zum Monitoring der Plattformen, ob On-Premises oder in der Cloud
- Oracle Management Cloud Compliance Control (OMC CC) zur Prüfung der Konfigurationen gegen Unternehmensvorgaben oder externe Regelwerke
- Oracle Cloud Access Security Broker (Oracle CASB) zum Entdecken einer Schatten-IT in der Cloud, dem Monitoring von Unternehmensvorgaben bezüglich Nutzung und Konfiguration von Oracle und 3rd-Party-Cloud-Diensten wie AWS, Salesforce, Azure, Box etc. sowie eine Data-Loss-Prevention-Funktionalität (DLP)
- Oracle Identity Cloud Service (Oracle IDCS) für das Bereitstellen eines Benutzerverwaltungs- und Authentifizierungssystems für On-Premises oder Cloud Services

Alle Komponenten zusammen bilden das im Jahr 2016 bei Gartner vorgestellte Oracle Identity Aware Security Operations Center (Identity SOC). Es vereint Identity, CASB, Security Monitoring & Analytics sowie Configuration Compliance und ermöglicht intelligente, risikobewusste Sicherheit in komplexen, hybriden Multi-Cloud-Umgebungen. In Kombination mit der Oracle Management Cloud bietet Oracle mit dem Security Monitoring ein einheitliches Dashboard für Management, Betrieb, Performance und Reporting *(siehe Abbildung 2)*.

Security-Best-Practices aus dem Report

Im „Oracle und KPMG Cloud Threat Report 2018" sind Security-Best-Practices aufgestellt. Diese tragen zwar Cloud-Security in der Überschrift, passen aber auch für On-Premises. On-Premises-Systeme haben die gleichen Problemstellen, sei es durch Mitarbeiter, eingeschleuste Maleware oder Zugriffsmöglichkeiten von außen. Die folgende Aufstellung aus dem Report ist keine umfassende Darstellung von Security-Best-Practices wie beispielsweise die CIS Controls, sondern dient zur Darstellung der von Oracle weiterentwickelten Security. Dabei kommen neue und

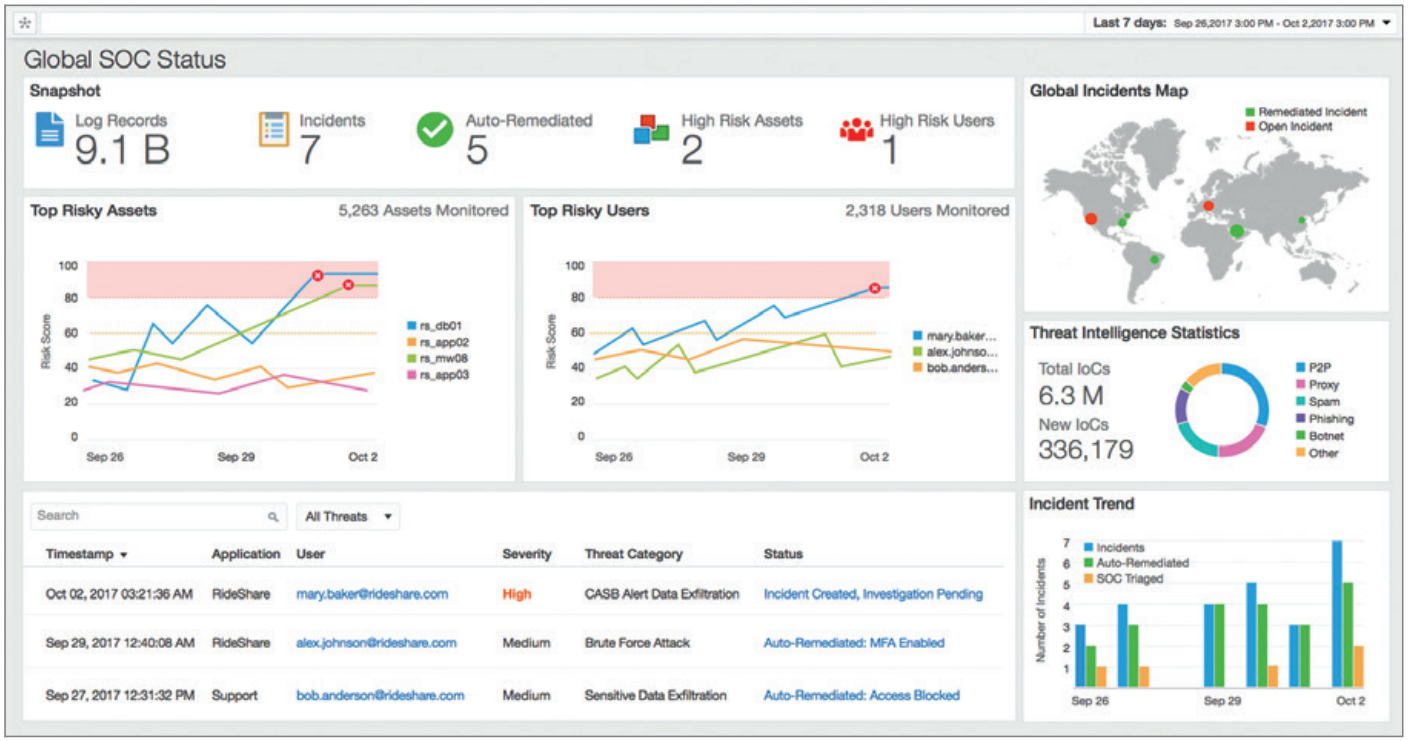


Abbildung 2: Oracle-Dashboard für das Security Monitoring

bekannte Mechanismen zum Tragen: Erstens müssen Policies für alle Umgebungen gelten und gesetzt werden – bei eigenen, externen oder Cloud-basierten Systemen oder Diensten. Um Policies in alle Umgebungen zu propagieren und durchzusetzen, können Provisionierungen und Enforcement Gateways/Proxys eingesetzt werden. Klassisches Identity und Access Management provisioniert Berechtigungen und stellt entsprechende Autorisierungen und SSO sicher. Werkzeuge von Oracle auch für hybride Umgebungen sind im IAM Portfolio. Erweitert wurde das IAM Portfolio um die Cloud-basierte Komponente, den Identity Cloud Service.

Zweitens bestimmt der Kontext der Daten deren Schutzbedarf mit, ein Aufruf von einem nicht registrierten Device beispielsweise ist unterschiedlich zu einem vertrauenswürdigen Device zu sehen. Kontextbasierte Autorisierung oder auch im Schritt vorher die kontextbasierte Anmeldemöglichkeit wird entweder in den Systemen umgesetzt (etwa während der Datenbank-Anmeldung) oder von einem zentralen Access-System durchgeführt (zum Beispiel mit Proxies). Werkzeuge von Oracle sind sowohl die eingangs gelisteten Komponenten als auch Funktionalitäten in den Produkten wie der Oracle-Datenbank.

Drittens sind Data Discovery und Klassifikation notwendig, um zu wissen, welche Daten im Unternehmen vorliegen und wie hoch der Schutzbedarf ist. In diesem Prozess wird die Anwendungslandschaft mit Hinblick auf verwendete Daten analysiert. Oracle bietet Hilfestellungen mit Werkzeugen bei der Inventarisierung und mit einem automatisierten Datenabgleich (Konsistenz der Datenquellen). Eine Analyse kann über das DBSat-Tool *(siehe "http://www.oracle.com/technetwork/database/security/dbsat/overview/index.html")* oder Application Data Modelling vom Enterprise Manager erfolgen, wenn neben den Dictionary-Funktionen auch Daten analysiert werden sollen. Die Data-Integration- und Enterprise-Quality-Data-Management-Werkzeuge können eingesetzt werden, wenn es um das Unternehmensdatenmodell und Datenabgleich geht. Die Werkzeuge von Oracle heißen wie in der Beschreibung in diesem Absatz aufgelistet.

Viertens ist die Verlässlichkeit des Schutzes durch Monitoring der gesetzten Konfiguration von Systemen und Einstellungen sicherzustellen. Die Konformität hinsichtlich der Konfiguration ist die Domäne von IT-Compliance. Werkzeuge von Oracle sind dazu der Enterprise Manager oder das „Configuration & Compliance"-Modul des Oracle Management Cloud Service. Vorhandene Regelwerke (STIG, CIS, Best Practices) können ausgewählt und auch angepasst werden. „Configuration & Compliance" hilft auch, ähnliche Datenbank-Umgebungen über Clustering zu erkennen, um Regelwerke individuell anzupassen.

Fünftens ist ein Schutz gegen Angriffe notwendig, um neben dem zuvor beschriebenen Schutz auch die Ausnutzung von Lücken in Systemen und Missbrauch von Accounts zu verhindern oder zumindest zu entdecken. Für Threat Prevention, hier durch das aktive, systemübergreifende Monitoring von Oracle und 3rd-Party-Komponenten, stellt Oracle Security Monitoring und CASB-Service zur Verfügung. Beide können Threats erkennen sowie die einzelnen Stufen von Kill Chains wie Lateral Movements.

Sechstens braucht es Data Loss Prevention, um einen ungewünschten Abfluss von Daten zu verhindern. Data Loss Prevention (DLP) wird vorrangig durch Unternehmenspolicies adressiert. Vergebene und regelmäßig überprüfte Berechtigungen beschreiben den Kreis der legalen Zugriffe. Unrechtmäßige Versuche werden im Vorfeld verhindert (Fehlkonfigurationen entdecken, bevor sie genutzt werden) oder beim Zugriff geblockt. Für dieses Blocken kann die starke Authentifizierung der Datenbank genutzt werden, die die Umgebungspa-

rameter auswertet, Tools auf dem Client oder ein Zugriff über einen Proxy. Im Fall von Datenbank-Zugriffen über Database Activity Monitoring (DAM) und im Falle von webbasierten Services über einen DLP-CASB-Proxy.

Siebentens ist Monitoring des tatsächlichen Verhaltens eines Benutzers oder Systems (UEBA) erforderlich, um herauszufinden, inwieweit Abweichungen vom normalen Verhalten vorhanden sind, um damit erkennen zu können, dass beispielsweise ein Account in fremde Hände gelangt ist oder der Mitarbeiter nicht unternehmenskonform agiert. Cloud Access Security Broker und Security Monitoring enthalten Policy-Monitoring, Threat-Erkennungen und UEBA-Komponenten. Alle drei Funktionen tragen zur Verhaltensanalyse bei. Threat-Erkennung kommt beispielsweise bei Brute-Force-Log-in-Versuchen zum Tragen oder beim Zugriff von wechselnden Orten in kürzester Zeit. UEBA vergleicht den Benutzer oder das System mit seinem normalen Verhalten oder dem Verhalten einer Vergleichsgruppe. So können Unregelmäßigkeiten bei SQL-Befehlen erkannt werden oder Systeme, die für andere Aufgaben zweckentfremdet werden.

Status quo der Installation bei Datenbanken

Laut dem Status quo bei Datenbanken („Oracle und KPMG Cloud Threat Report 2018“) nutzt lediglich die Hälfte der Befragten Datenbank-Firewalls, Verschlüsselung, Web-Application-Firewalls (WAF) und übergreifendes Monitoring. Etwa ein Viertel der Befragten nutzt beim Monitoring auch weiterentwickelte Technologien wie Machine Learning. Eine Annäherung an die Einführung des automatisierten Betriebs, hier Security Automation, ist erfolgt, aber noch nicht umgesetzt.

Bestehende Mechanismen oder Funktionalitäten werden noch nicht vollständig ausgeschöpft und bieten, auch wenn sie systemübergreifend eingesetzt werden, viel Potenzial für eine stärkere Absicherung. Dies ist jedoch nicht das Ende der Entwicklung der Datenbank; die aktuelle Version 18c bringt weiterentwickelte Security-Mechanismen, die in dieser Ausgabe im vorangegangenen Artikel vorgestellt sind.

Die Cloud

Cloud ist nochmal ein anderer Blickwinkel. Die bisher beschriebenen Mechanismen, Services oder Technologien können alle ebenfalls in einem Cloud-Szenario genutzt werden. Darüber hinaus bietet die Cloud weitere Sicherheitsmerkmale, die eigenständig weiterentwickelt werden. Beginnend mit dem Security-Paradigma „secure by default“ werden instanziierte Komponenten wie Datenbanken oder Storage verschlüsselt aufgesetzt und die Berechtigungen nach dem „least privilege“-Prinzip initial so zugewiesen, dass die Nutzung durch den Cloud-Verantwortlichen des Kunden erst freigeschaltet werden muss.

Services nutzen die Basis-Infrastruktur der Cloud, die unter Sicherheitsgesichtspunkten aufgesetzt und stetig weiterentwickelt wird. Nur zwei Merkmale davon sind die Off-Box-Netzwerk-Virtualisierung, um das Netzwerk zu definieren und abzusichern (SDN ohne zentralen Controller), und die exklusiven Nutzungsmöglichkeiten von dedizierten Servern. Mehr Informationen dazu stehen im Artikel zum Frankfurter Oracle Datacenter in dieser Ausgabe auf Seite 58.

Fazit und Ausblick

Security von, mit und bei Oracle bleibt ein spannendes Thema. Oracle erweitert im Security-Umfeld sein Portfolio kontinuierlich mit Eigenentwicklungen und Zukäufen, zuletzt mit der Akquise von Dyn (DNS-Service) und dem noch nicht vollständig durchgeführten Zukauf von Zenedge, einer Web-Application-Firewall inklusive Service-Management.

Die vergangenen Jahre brachten einen Ausbau des Portfolios um Sicherheitsfunktionen wie im Datenbank-Bereich, Sicherheits-Komponenten im Identity und Access Management mit speziellen Cloud Security Services sowie durch den Aufbau von Cloud-Rechenzentren der nächsten Generation. Der Portfolio-Ausbau ist natürlich nicht abgeschlossen, kommende Services bringen Erweiterungen in DLP oder im Bereich „Governance in der Cloud“.

In die Weiterentwicklung eingebracht wurde eine weitere Automatisierung sowohl für den Betrieb von Komponenten (wie Datenbank) als auch für das Monitoring. Entscheidenden Beitrag dazu leistet das jeweils integrierte Machine Learning. Die Hürde für potenzielle unrechtmäßige oder unbeabsichtigte Zugriffe wird höher gelegt, Unregelmäßigkeiten werden weitgehend automatisiert erkannt. Fehlkonfigurationen, vernachlässigtes Einspielen von Patches oder Probleme bei Verfügbarkeit, Backup und Restore werden unwahrscheinlicher. Die Automatisierung wird weiter vorangetrieben und ist für weitere Services geplant.

Dem Wandel des Betriebsmodells wurde ebenfalls Rechnung getragen. DevOps und Cloud-Native-Development werden unterstützt, ebenso wie der Umbau der SIEM- oder Netzwerk-Monitoring-Teams mit der Einführung von Security Operation Centers (SOC). Hier ist der SOC-Spezialist nicht mehr allein, um die Drehscheibe um SIEM-Informationen beziehungsweise Alarme zu bewerten. Algorithmen aus dem Bereich „Machine Learning“ oder aus der Daten-Analyse übernehmen nun das Aussortieren relevanter Daten und leiten gegebenenfalls eine automatisierte Behandlung ein.

Die Weiterentwicklung beim Thema „Sicherheit“ erfolgt bei Oracle unter verschiedenen Aspekten: mehr Automatisierung, mehr künstliche Intelligenz, Erweiterung der Sicherheitsfunktionen in vorhandenen Komponenten und weiteren neuen Komponenten sowie Cloud Services. Um hier Silo-Landschaften entgegenzuwirken, erfolgt parallel die Integration in eine Art Sicherheits-Plattform, die optional genutzt werden kann.

Weitere Informationen

- Securitykomponenten von Oracle: *https://www.oracle.com/security*
- Autonomous Services: *https://www.oracle.com/autonomouscloud/index.html*

Michael Fischer
michael.fischer@oracle.com

Passwörter in Skripten verschlüsselt hinterlegen

Gunther Pippèrr, Freiberufler

Das Problem: In vielen Skripten rund um die tägliche Wartung unserer Systemumgebungen müssen Passwörter hinterlegt sein und nicht immer kann mit dem Oracle-Wallet oder mit SSL-Zertifikaten beziehungsweise Betriebssystem-Rechten ganz auf Passwörter verzichtet werden.

Wie lassen sich Passwörter in Skripten so schützen, dass nicht jeder diese sofort auslesen und verwenden kann? Wie kann man die Skripte in einer Sourcecode-Verwaltung so hinterlegen, dass dort sicher keine Passwörter mehr vorkommen? Wie muss man Passwörter auf gehosteten Server-Umgebungen hinterlegen, um den Sicherheits-Regularien des Unternehmens zu genügen?

Gerade in Umgebungen, die in die Jahre gekommen sind, wimmelt es nur so von Passwörtern. Jede Änderung bedingt meist umfangreiche Anpassungen an vielen Stellen. Werden Skripte kopiert und weitergegeben, besteht immer die Gefahr, dass ein Passwort in die falschen Hände gerät. Besonders in gehosteten Umgebungen, auf denen das Betriebssystem von einem Hoster mit Root-Rechten verwaltet wird, sollte man sich der Gefahr mit den offenen Passwörtern in Skripten sehr bewusst sein. Diese stellen trotz aller Bemühung auch heute noch eines der höchsten Sicherheitsrisiken für die meisten Systeme dar. Um dieses Risiko zu bekämpfen, sollten folgende Vorgaben für alle Umgebungen gültig sein:

- Kein Passwort kommt im Skript vor; es wird nur eine Variable verwendet, die mit dem Passwort gefüllt wird.
- Bei einer Passwort-Änderung muss das Skript nicht angepasst werden.
- Die Passwörter sind auf dem System verschlüsselt hinterlegt.
- Die verschlüsselten Passwörter können nur auf der Zielmaschine entschlüsselt gelesen werden.
- Alles was für die Umsetzung des Konzepts benötigt wird, muss auch in veralteten beziehungsweise gehosteten Umgebungen ohne besondere Systemrechte möglich sein.

Nebenbei werden so ohne großen Aufwand meist schon die wichtigsten Sicherheits-Regeln eingehalten, ohne den Betrieb mit zu hohen Aufwänden zu belasten. Die Umsetzung des Konzepts schützt vor folgenden Szenarien:

- Skripte können nun problemlos per Mail/Git oder Web verteilt werden, keine Passwörter gehen aus Versehen verloren.
- Die verschlüsselten Passwörter sind keinem von Nutzen, der keinen Zugriff auf den Ziel-Server hat.
- Unangenehme Gespräche mit dem Security-Officer in der Kantine beim Mittagessen.

In folgenden Szenarien schützt das Konzept nicht:

- Vor den neugierigen Blicken der Kollegen mit Root-Zugriff auf das System: Während das Skript läuft, ist im Speicher oder in der „/proc“-Umgebung das Passwort mit etwas Geschick meist auffindbar.
- Das Passwort lässt sich auf der Maschine per Skript auslesen, der Schlüssel für das Passwort lässt sich dort allerdings nie so verstecken, dass keiner ihn findet, denn das Skript benötigt diesen ja auch.

Wie bei jeder Verschlüsselung ist es im Grunde egal, wie komplex oder sicher der Algorithmus ist, meist kann durch ein wenig Nachdenken und Ausprobieren der Schlüssel im System selber gefunden werden. Den Schlüssel zu verstecken, gelingt den wenigsten wirklich, denken wir nur an unsere Eltern, da liegt der Hausschlüssel auch immer hinten am Gartenzaun unter dem Blumentopf.

Unser Ziel ist mit dieser Lösung nicht eine 100-prozentige Sicherheit, sondern das Begrenzen von Schaden und das Erfüllen von Sicherheitsvorgaben. Werden diese konsequent umgesetzt, bedeutet dies für das Unternehmen meist schon eine dramatische Verbesserung der Sicherheitslage – ohne großen Aufwand und ohne viele Schwierigkeiten im Betrieb. Wie lässt sich das Ganze nun aber unter Windows und Linux so bequem wie möglich umsetzen? Unser Werkzeugkasten besteht aus:

- Microsoft Credential Objects in der Windows PowerShell
- Verschlüsseltes Hinterlegen von Passwörtern auf Linux-Systemen mit „openssl“
- Oracle-Features wie Oracle Wallet

Im ersten Schritt ist die ideale Lösung für das Problem, gar keine Passwörter mehr zu verwenden. Wir delegieren diese Auf-

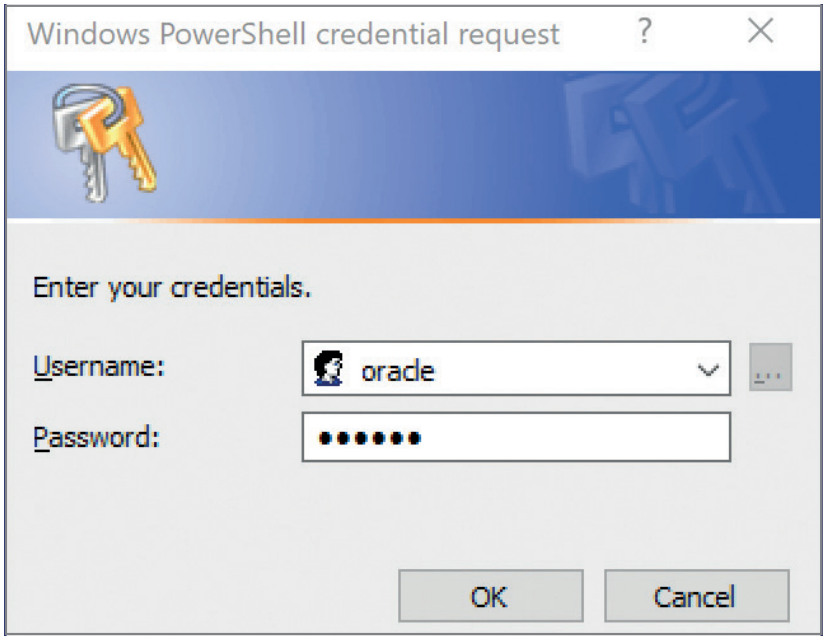


Abbildung 1: Der Password-Dialog vom „get-Credential“

gabe einfach an das Betriebssystem der Datenbank und überlassen diesem beziehungsweise dem System-Administrator der Umgebung die ganze Verantwortung dafür, dass alles sicher betrieben wird. Der Nachteil: Wer sich am System anmelden kann, kommt nun ganz ohne Passwort aus. Sehr einfach lässt sich das umsetzen, wenn der Betriebssystem-User in der DBA-Gruppe ist und alle Aufgaben mit dem SYS-User durchgeführt werden können. Etwas sicherer ist es, „External authentification“ für die Datenbank-Zugänge einzusetzen. Sehr komfortabel ist das Oracle-Wallet als Secure-External-Passwort-Store-Lösung *[1]*.

Passwörter unter Windows schützen

Unter Windows ist das verschlüsselte Hinterlegen der Passwörter so trivial, dass jeder, der es nicht nutzt, sich eigentlich fast grob fahrlässig verhält. Das Erstellen eines Passwort-Containers inklusive des Aufrufs der Pflege-Oberfläche sind im Prinzip nur zwei Zeilen Code *(siehe Abbildung 1)*. Wer lieber das Passwort über die Konsole setzt, kann mit dem Registry-Eintrag „(HKLM:\SOFTWARE\Microsoft\PowerShell\1\ShellIds\ConsolePrompting)=$True“ den grafischen Dialog vermeiden *(siehe Abbildung 2)*.

Listing 1 zeigt den Sourcecode. Ein praktisches Beispiel dazu ist, den Apex-Sourcecode automatisch zu exportieren und einzuchecken mit Git unter Windows mit der PowerShell *[2]*.

Wie das Ganze im Detail funktioniert

Das Passwort wird über den originalen Windows-Passwort-Dialog einmalig über den Aufruf von „GET-CREDENTIAL“ eingegeben und in einer serialisierten Form als XML-Datei auf der Festplatte hinterlegt. Im Code erzeugt man mit „$user_credential=GET-CREDENTIAL -credential "$db_user“ in der Variablen „user_credential“ das Credential-Objekt mit dem Passwort. Der Schlüssel ist die eindeutige ID des aktuell installierten Betriebssystems, dies ist für jeden Windows-Rechner auf dieser Welt eindeutig.

Um nun den Passwort-Store auf der Platte abzulegen, um nicht bei jedem Skript-Aufruf das Passwort neu eingeben zu müssen, wird das Objekt mit „export-clixml -InputObject $user_credential -Path $oracle_credential” serialisiert. Es entsteht eine XML-Datei mit dem Objekt in einer Art „BASE64“-Codierung.

Beim nächsten Lauf ist das Passwort bereits hinterlegt und das Objekt wird wieder mit „$user_credential=Import-Clixml -Path $oracle_credential” in den Speicher geladen. Nun lässt sich das Passwort im Skript in Klarschrift über „$db_password=$user_credential.GetNetworkCredential().Password“ auslesen.

Passwörter unter Linux sicher verwahren

Unter Linux ist das Ganze etwas schwieriger. Gerade in gehosteten Umgebungen muss der Kunde mit dem Vorlieb nehmen, was der Dienstleister unter Sicherheit versteht. Also meistens rein kostenoptimiert zu arbeiten und wenig flexibel auf besondere Software-Wünsche wie ein aktuelles Java oder einen „gcc“ einzugehen.

SSH und damit „openssl“ steht hier jedoch überwiegend zur Verfügung beziehungsweise es spricht sehr wenig dagegen, das kostenfrei installieren zu lassen. Damit steht mit „openssl“ auf Linux ein Werkzeug zur Verfügung, das die kompliziertesten Verschlüsselungs-Verfahren sehr sicher beherrscht. Jetzt ist nur noch einen Schlüssel zu finden, der möglichst

```
PS C:\> Get-Credential

cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential
User: oracle
Password for user oracle: ******


UserName                       Password
--------                       --------
oracle   System.Security.SecureString
```

Abbildung 2: Der Password-Dialog vom „get-Credential“ in der Konsolea

```
# Parameter belegen
$db_user     = "system"
$oracle_credential = "ORACLE_CREDENTIAL.xml"

#
# Prüfen ob bereits ein serialisiertes Password gefunden werden kann
# wenn nicht den Dialog dazu öffnen und das Password als XML speichern (verschlüsselt!)
#

if (!(test-path -path $oracle_credential)) {
     #  setzen des Passworts über einen Dialog
   $user_credential=GET-CREDENTIAL -credential "$db_user"
   export-clixml -InputObject $user_credential -Path $oracle_credential
}
else {
   # Objekt wieder einlesen
   $user_credential=Import-Clixml -Path $oracle_credential
}

#Password als text wieder auslesen
$db_password=$user_credential.GetNetworkCredential().Password
```

Listing 1

länger ist als das Passwort, um das Entschlüsseln für den Angreifer spannender zu gestalten. Der Schlüssel muss Folgendes erfüllen:

- Auf jedem Server dieser Welt eindeutig sein
- Länger als das Passwort sein
- Sich nicht auf den ersten Blick erkennen lassen
- Einfach unter Linux zu erzeugen/auszulesen sein

Solche Schlüssel könnten sein:

- WWN oder UUID eines Device, das sich auf dem Server nicht so schnell ändert
- Hardware-ID wie die MAC-Adresse oder die Prozessor-ID
- Eine eigene Routine in C, die eine eindeutige ID erzeugt

Der Autor verwendet meist einfach die UUID von „/dev/sda1“ als Schlüssel oder die Seriennummer des OS unter HP UX. Es ist der Phantasie des Entwicklers überlassen, durch Obfuscation das Ganze noch etwas intransparenter zu gestalten. Die Kollegen vom Support aus fernen Ländern sollten das am Ende allerdings noch bedienen und warten können.

Im Detail wird das Passwort zu Beginn in einer Datei in Klarschrift hinterlegt und dann verschlüsselt. Das kann im Skript gleich beim nächsten Aufruf oder mit einem kleinen Hilfsskript vorab erfolgen. „openssl“ dient im Folgenden dazu, die Konfigurationsdatei mit dem Passwort-Store zu verschlüsseln, was den kompliziertesten Teil des Ganzen darstellt. Über den Aufruf „openssl des3 -salt -in ${PWDFILE} -out ${PWDFILE}.des3 -pass pass:"${SYSTEMIDENTIFIER}"“ wird die Passwort-Datei verschlüsselt. *Listing 2* zeigt den Sourcecode.

Um die Sicherheit noch weiter zu erhöhen und um Spuren in der Umgebung so weit wie möglich zu verschleiern, kann das Passwort in der Datei zuvor noch mit einem symmetrischen Algorithmus so verschlüsselt werden, dass erst zur Laufzeit im Skript an den jeweiligen Stellen das echte Passwort extrahiert wird. Dieses sollte dann ohne Umwege in das aufzurufende Programm hinein „gepiped“ werden. Dies ist zwar nicht deutlich sicherer, dient aber dazu, auf den ersten Blick den Angreifer etwas mehr zu verwirren.

Fazit

Auch mit diesem Konzept lässt sich keine endgültige Sicherheit herstellen. Es ist je-

Über „/proc“ die Laufzeit-Umgebung eines Linux-Prozesses auslesen

Mit entsprechenden Rechten lässt sich über das „/proc“-Dateisystem und die „environ“-Datei die gesetzte Umgebung eines Skripts auslesen. Sensitive Daten wie Passwörter sollten also nur so kurz wie möglich in diesem Bereich sichtbar sein. Auf diesem Weg lässt sich auch sehr einfach überprüfen, welche Umgebungs-Variablen ein Prozess (wie zum Beispiel ein Oracle-Hintergrund-Prozess) wirklich sieht und verwendet. Dazu ein Anwendungsbeispiel:

- Prozess-ID mit „ps uafx | grep sqlplus” ermitteln
- Mit „cd /proc/<processid>/“ in das Proc-File-System wechseln
- Über die Datei „environ“ kann nun die Umgebung des Prozesses mit „strings environ | grep NLS_LANG“ ausgelesen werden

```
# password.conf.des3
PWDFILE=.password.conf
export PWDFILE

# etwas Eindeutiges aus der Umgebung der Maschine auslesen
# Linux
SYSTEMIDENTIFIER=`ls -l /dev/disk/by-uuid/ | awk '{ print $9 }'  | tail -1`
export SYSTEMIDENTIFIER

#
# Password verschlüsseln
encryptPWDFile () {
    /usr/bin/openssl des3 -salt -in  ${PWDFILE} -out ${PWDFILE}.des3 -pass pass:"${SYSTEMIDENTIFIER}" > /dev/
null
    # Remove orignal file
    rm ${PWDFILE}
}

# Password wieder auslesen
dencryptPWDFile() {
    /usr/bin/openssl des3 -d -salt -in ${PWDFILE}.des3 -out ${PWDFILE} -pass pass:"${SYSTEMIDENTIFIER}" > /
dev/null
}


# Password in den Speicher laden
# Falls verschlüsselte Datei vorliegt
if [ -f "${PWDFILE}.des3" ]; then
    dencryptPWDFile
    # in die Umgebung einlesen
    . ${PWDFILE}
    # Klarschrift Datei wieder entfernen
    rm ${PWDFILE}
else
     # Falls unverschlüsselt vorliegt, Datei verschlüsslen
   if [ -f "${PWDFILE}" ]; then
        . ${PWDFILE}
       encryptPWDFile
    else
      echo "no preconfiguration file =>password.conf<= found"
      echo "export DB_PWD=" > ${PWDFILE}
      echo "no preconfiguration password.conf found - edit the file =>password.conf<= and set password and start
again"
      exit 1
    fi
fi

# Password auf eine interne Variable kopieren und überschreiben
# siehe Kasten zu den environ Problem
INTERNAL_PWD=${DB_PWD}
export DB_PWD="DU_SOLLST_DAS_NICHT_LESEN"

echo "Info --  read encrypted password  =>> ${INTERNAL_PWD} <<=="
```

Listing 2

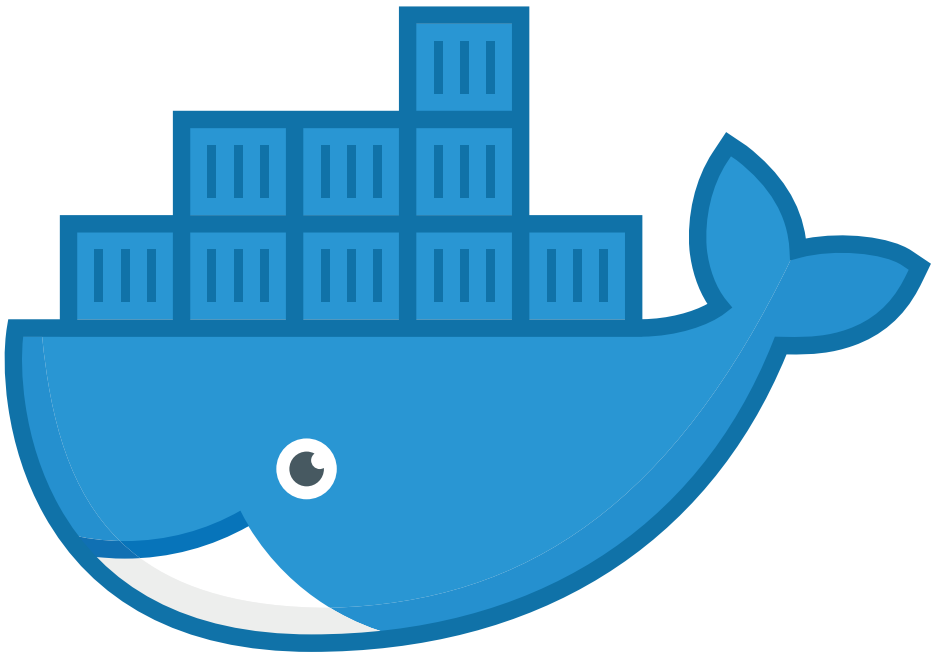
doch ein wichtiger Schritt, sensitiver und vor allem proaktiv mit dem Passwort-Problem umzugehen. Und das Ganze ohne besonderen Aufwand für den Betrieb mit dem Vorteil, die Passwort-Änderungen zu zentralisieren. Der vollständige Sourcecode steht unter „*https://github.com/gpipperr/RedStack_Magazin_2018_encrypt_password*“. Besonders möchte sich der Autor bei Martina Pippèrr und Sebastian Geddes bedanken, die geduldig geholfen haben, diesen Text zu überabeiteten.

Weiterführende Links

[1] *https://www.pipperr.de/dokuwiki/doku.php?id=dba:passwort_schuetzen*

[2] *https://www.pipperr.de/dokuwiki/doku.php?id=dba:oracle_secure_external_password_store*

Gunther Pippèrr
gunther@pipperr.de



Oracle Unified Directory in Docker

Stefan Oehrli, Trivadis AG

Virtualisierung, Container beziehungsweise Docker sind seit Längerem aktuelle Themen der modernen IT. Viele Hersteller stellen entsprechend Docker-Images ihrer Produkte zur Verfügung. Oracle Unified Directory, ein schlanker, Java-basierter Verzeichnisserver, ist prädestiniert für den Betrieb in einer Container-Umgebung. Dieser Artikel zeigt, wie man Oracle Unified Directory in einem Docker-Image installiert, konfiguriert und anschließend als Container betreibt.

Für viele Oracle-Produkte wie Datenbanken, Fusion Middleware, Java etc. gibt es in der Oracle Container Registry *[1]* fertige Docker-Container, die sich zumindest für Entwicklungsarbeiten direkt nutzen lassen. Benötigt man eine speziellere Version oder möchte den Container an die eigenen Bedürfnisse anpassen, findet man im offiziellen Oracle Docker GitHub Repository *[2]* Docker-Files sowie Build- und Konfigurationsskripte – nur für Oracle Unified Directory (OUD) bietet Oracle aktuell keinen fertigen Docker-Container oder geeignete Build-Skripte an. Dies ist schade, zumal es den einen oder anderen Anwendungsfall gibt, in dem man einen LDAP-Server auch in einer Container-Umgebung nutzen möchte. Da Installation und Konfiguration von OUD keine Hexerei ist, steht dem Vorhaben „OUD Docker Image“ nichts im Weg.

Voraussetzungen

Oracle Unified Directory ist relativ genügsam, wenn es um die System-Anforderungen geht. Es sind lediglich ein entsprechendes Betriebssystem, ein aktuelles Oracle-JDK sowie rund 500 MB Platz für die Software und die OUD-Instanz erforderlich. Ein Betrieb von OUD ist auf jeder Umgebung möglich, in der ein Oracle-JDK installiert werden kann; also auch auf einem einfachen Raspberry Pi *[3]* mit Debian Linux, wobei es sich dabei nicht um eine zertifizierte Umgebung für OUD handelt. Informationen zu den zertifizierten Komponenten entnimmt man am besten der Zertifizierungsmatrix auf My Oracle Support *[4]*. Im Rahmen dieses Artikels werden wir uns auf folgende Versionen beschränken:

- Betriebssystem Oracle Enterprise Linux 7.5 (slim)
- Oracle Server JRE 8 Update 172
- Oracle Unified Directory 12.2.1.3.0
- Oracle Fusion Middleware Infrastructure 12.2.1.3.0
- Aktuelle Patch Set Updates vom April 2018
- OUD-Docker-Build-Skripte *[5]*
- OUD-Umgebungs-Skripte *[6]*

Weitere Versionen sind durch Oracle unterstützt. Im GitHub Repository des Autors *[5]* finden sich weitere Docker-Files für OUD 12c, OUD 11g, OUDSM 12c sowie für den etwas älteren Oracle Directory Server Enterprise Edition 11.1.1.7.0. Als Entwicklungsumgebung wird die Docker Community Edition 18.05.0 auf einem Laptop mit MacOS 10.13 verwendet. Es ist ohne Weiteres möglich, das vorgestellte Setup in aktuellen Docker-Umgebungen auf anderen Betriebssystemen nachzubauen.

Neben der lauffähigen Docker-Umgebung wird für die folgenden Arbeiten vorausgesetzt, dass sowohl die Docker-Build-Skripte als auch die Software lokal vorhanden sind. Die einfachste Methode dafür ist ein Git Clone vom GitHub Repository des Autors *[5]* zu erstellen. Die Links für die Downloads der oben aufgeführten Software sind, wie man es vom offiziellen Oracle Docker GitHub *[2]* Repository gewohnt ist, als „*.download"-Testdateien vorhanden.

OUD-Installationsmethoden

Mit dem neuesten Release von OUD hat Oracle verschiedene Varianten der Implementierung eingeführt.

- *Stand-alone OUD Server*
  Bei dieser Implementierungsmethode wird OUD als einfacher LDAP-Server mit geringem Platzbedarf eingesetzt. Die Administration erfolgt über die Kommandozeile („dsconfig", „ldapmodify" etc.) oder, wenn möglich, mit einem LDAP-Browser. Diese Methode wird für das OUD-Docker-Image verwendet.

- *Collocated OUD Server*
  Funktioniert mit OUD und Oracle Unified Directory Services Manager (OUDSM) in separaten Domains. Bei dieser Methode wird neben OUD zusätzlich Fusion-Middleware-Infrastruktur für den Oracle Unified Directory Services Manager (OUDSM) installiert. Diese Methode wird für das OUDSM-Docker-Image verwendet, wobei im OUDSM-Docker-Image lediglich die OUDSM-Komponente zum Einsatz kommt.

- *Collocated OUD Server*
  Funktioniert mit OUD und OUDSM in einer Domain. Bei dieser Methode wird neben OUD zusätzlich Fusion-Middleware-Infrastruktur für OUDSM installiert. OUD und OUDSM sind in einer WLS-Domain konfiguriert und werden gemeinsam administriert. Diese Methode ist komplexer, benötigt zusätzliche Installationsschritte und wird im Folgenden nicht weiter erläutert.

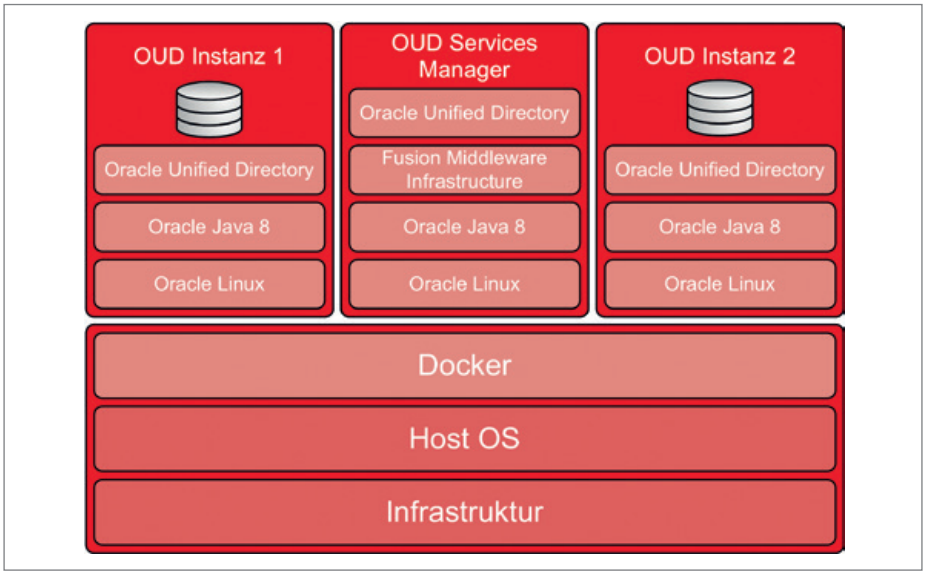


Abbildung 1: Docker-Image

Es ist möglich, sowohl OUD als auch OUDSM in einem Docker-Image zu installieren und zu nutzen. Doch auf diese Weise laufen mehrere Prozesse beziehungsweise Services in einem Container. Dies widerspricht dem Grundsatz, dass pro Container nur ein Service zu konfigurieren ist. Zudem lassen sich mit einem OUDSM mehrere OUD-Instanzen auf unterschiedlichen Hosts beziehungsweise Containern administrieren. Aus diesem Grund trennen wir die OUD- und OUDSM-Instanzen auf und erstellen zwei separate Docker-Images. *Abbildung 1* zeigt schematisch eine Docker-Umgebung mit mehreren OUD-Containern und einem OUDSM-Container.

Das Basis-Docker-Image

Bevor wir mit der eigentlichen Installation von OUD beginnen können, ist ein Docker-Basis-Image erforderlich. Im Docker-File von OUD und OUDSM wird mit „FROM oracle/serverjre:8" auf ein entsprechendes Basis-Image referenziert. In der Regel ist ein Image mit diesem Tag nicht vorhanden. Daher ist ein bestehendes Docker-Image, das Oracle Java 8 enthält, mit einem dazugehörigen Tag zu versehen.

Das Kommando „docker tag oracle/serverjre:1.8.0_172 oracle/serverjre:8" zeigt, wie ein bestehendes Docker-Image „oracle/serverjre:1.8.0_172" mit dem benötigten Tag versehen werden kann. Alternativ kann man das offizielle Oracle-Java-8-Image aus der Oracle Container Registry *[1]* laden. Nach dem Log-in kann das Image mit „docker pull" geladen und mit einem Tag versehen werden *(siehe Listing 1)*.

Als weitere Variante erstellt man ein eigenes Basis-Image mit Oracle Java 8.

```
docker login container-registry.oracle.com
docker pull container-registry.oracle.com/java/serverjre:8
docker tag container-registry.oracle.com/java/serverjre:8 \ oracle/serverjre:8
```

Listing 1

Im GitHub-Repository des Autors *[5]* sind ein entsprechendes Docker-File und eine kurze Beschreibung vorhanden. Mit dem Skripts „build.sh“ oder „docker build“ wird das gewünschte Image erstellt.

OUD-Image

Nachdem das Basis-Image vorhanden ist, muss die Software, also die OUD-Binaries„p26270957_122130_Generic.zip“ sowie der aktuellen PSU „p27742743_122130_Generic.zip“, in den Build-Ordner von OUD („$HOME/docker/OracleOUD/12.2.1.3.0“) kopiert werden. Anschließend lässt sich das OUD-Docker-Image mit den Kommandos „cd OracleOUD/12.2.1.3.0“ und „docker build -t oracle/oud:12.2.1.3.0“ erstellen. Zusammenfassend werden mit dem „docker build“-Kommando folgende Schritte aus dem Docker-File durchgeführt:

- Setzen der verschiedenen Umgebungsvariablen, abhängig von allfälligen Build-Argumenten für „ORACLE_ROOT“, „ORACLE_DATA“ oder „ORACLE_BASE“. Mit diesen Parametern lassen sich bei Bedarf die Verzeichnisstruktur der OUD-Installation, insbesondere das Verzeichnis für die OUD-Instanz, sowie das Oracle-Base anpassen. In der Regel ist dies aber nicht nötig.
- Erstellen des Benutzers und der Gruppen für die OUD-Installation sowie Anlegen der verschiedenen Verzeichnisse. Optional werden bei Bedarf noch fehlende Pakete aus dem Oracle Public YUM Repository nachinstalliert („tar“, „gzip“ und „libaio“).
- Kopieren der Docker-Hilfe-Skripte sowie, falls vorhanden, der Software-Pakete.
- Installation von OUD mit dem Hilfe-Skript „setup_oud.sh“. Es stellt sicher, dass die OUD-Software entpackt und mit einer Silent-Installation ein Standalone-OUD installiert wird. Im Anschluss an die Installation wird zusätzlich der aktuelle PSU installiert. Bei Bedarf kann das Skript die Software auch herunterladen. Dazu später mehr.
- Installation von OUD-Base mit dem Hilfe-Skript „setup_oudbase.sh“. Es lädt die aktuellste Version von OUD-Base direkt von GitHub.
- Festlegen der OUD-Ports, die später exportiert werden.
- Definition des Volume-Pfads für die persistenten Daten. Die OUD-Instanz sowie Log- und Konfigurationsdateien sind in einem separaten Verzeichnis „ORACLE_DATA“ beziehungsweise standardmäßig in „/u01“ abgelegt.
- Abschließend werden das Health-Check-Skript sowie das OUD-Start-Skript festgelegt.

Mit dem so erstellten Docker-Image lässt sich nun ein OUD-Container erstellen und starten. Das folgende Kommando *(siehe Listing 2)* startet im Hintergrund einen OUD-Container mit dem Namen „oudeng“ sowie dem Volume-Pfad „/Data/vm/docker/volumes/oudeng“.

Zusätzlich wird mit diesem Aufruf explizit das Port-Mapping angegeben. So

```
docker run --detach \
--volume /Data/vm/docker/volumes/oudeng:/u01 \
-p 1389:1389 -p 1636:1636 -p 4444:4444 \
-e OUD_INSTANCE=oud_test \
-e BASEDN=dc=example,dc=com \
--hostname oudeng --name oudeng \
oracle/oud:12.2.1.3.0
```

Listing 2

```
soe@gaia:~/ [ic12201] docker exec -it oudeng bash --login
Source environment for OUD Instance oud_docker
------------------------------------------------------------
 Instance Name       : oud_docker
 Instance Home (ok)  : /u01/instances/oud_docker
 Oracle Home         : /u00/app/oracle/product/fmw12.2.1.3.0
 Instance Status     : up
 LDAP Port           : 1389
 LDAPS Port          : 1636
 Admin Port          : 4444
 Replication Port    : 8989
------------------------------------------------------------
oracle@oudeng:/u00/app/oracle/ [oud_docker]
```

Abbildung 2: Interaktiver Zugriff auf „oudeng“ via „docker exec“

ist der Container von außen, also vom Host System, erreichbar. Auf diesen Container kann nun via Kommandozeile mit „docker exec“ interaktiv oder mit einem LDAP-Browser auf einen der exportierten Ports zugegriffen werden. *Abbildung 2* zeigt ein Beispiel, wie mit „docker exec“ eine „bash“-Shell im Container „oudeng“ gestartet wird. Auf diese Weise lässt sich die OUD-Instanz wie gewohnt mit der Kommandozeile verwalten.

Abbildung 3 zeigt den Directory Information Tree (DIT) der OUD-Instanz, angezeigt durch das Apache Directory Studio. Der Zugriff erfolgte via „localhost“ auf Port 1389 beziehungsweise mit der LDAP-URL „ldap://localhost:1389“.

Wie ersichtlich, enthält die Instanz bereits den Suffix „dc=postgasse,dc=org“ mit rund 100 Einträgen unter „ou=People, dc=postgasse,dc=org“. Dies liegt daran, dass durch das Startskript beim ersten Start des OUD-Containers automatisch eine Instanz angelegt wird. Dabei werden verschiedene Standartwerte verwendet, die sich durch das Setzen von Umgebungsvariablen steuern lassen. Dreh- und Angelpunkt ist das OUD-Startskript „start_oud_instance.sh“. Es prüft, ob eine OUD-Instanz vorhanden ist. Falls nicht, wird mit dem Skript „create_oud_instance.sh“ eine solche erstellt. Bei Bedarf werden dabei spezifisch Konfigurationsskripte ausgeführt, um die OUD-Instanz zu konfigurieren. *Abbildung 4* zeigt schematisch den Ablauf beim Start beziehungsweise Erstellen des OUD-Containers.

Neben „start_oud_instance.sh“ und „create_oud_instance.sh“ werden folgende Hilfe-Skripte im Docker-Container genutzt:

- „check_oud_instance.sh“ wird für den Health Check des Docker-Containers genutzt und prüft den Status der OUD-Instanz. Es ist verantwortlich für die Angabe eines Status „healthy/unhealthy/starting“ beim Aufruf von „docker p“.
- „config_oud_instance.sh“ prüft, ob entsprechende Konfigurationsdateien („*.sh“, „*.conf“ oder „*.ldif“) im Verzeichnis „${OUD_INSTANCE_ADMIN}/create“ vorhanden sind. Falls ja, werden diese sequenziell ausgeführt.
- „create_oud_instance.sh“ erstellt bei Bedarf eine OUD-Instanz. Basis-Instanz-Parameter lassen sich mit Umgebungsvariablen anpassen. Das Skript generiert explizit ein Passwort für den Root-Directory-Benutzer und legt dieses in der Datei „${OUD_INSTANCE_ADMIN}/etc/${OUD_INSTANCE}_pwd.txt“ ab.
- „setup_oud.sh“ wird nur beim Build des Docker-Image zur Installation von OUD benötigt.

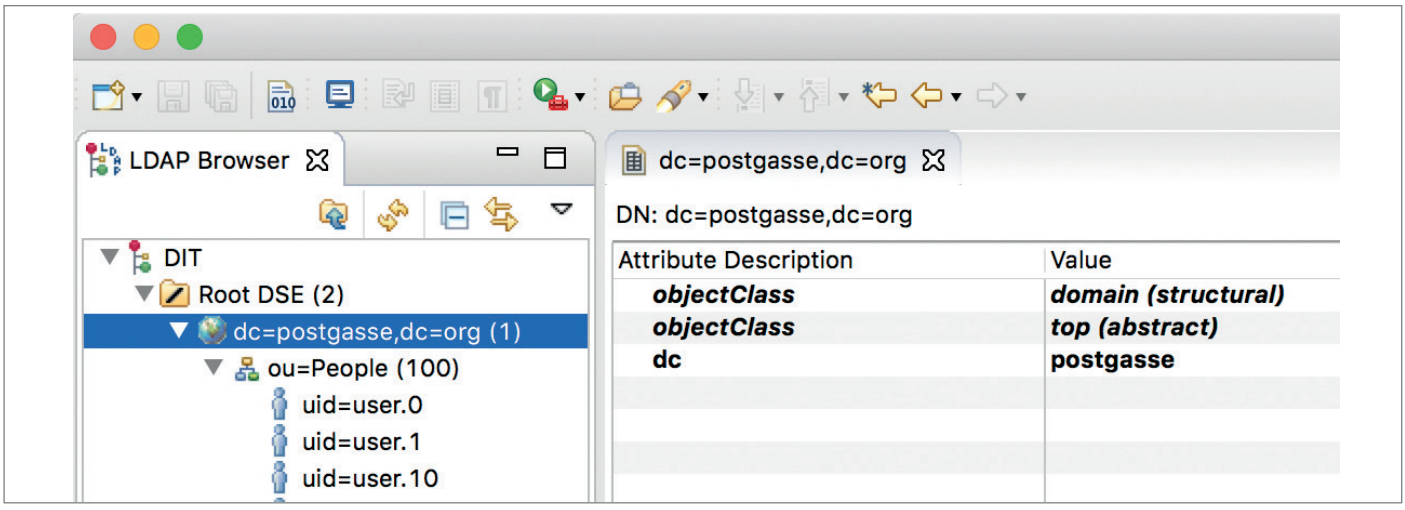


Abbildung 3: Zugriff mit Apache Directory Studio

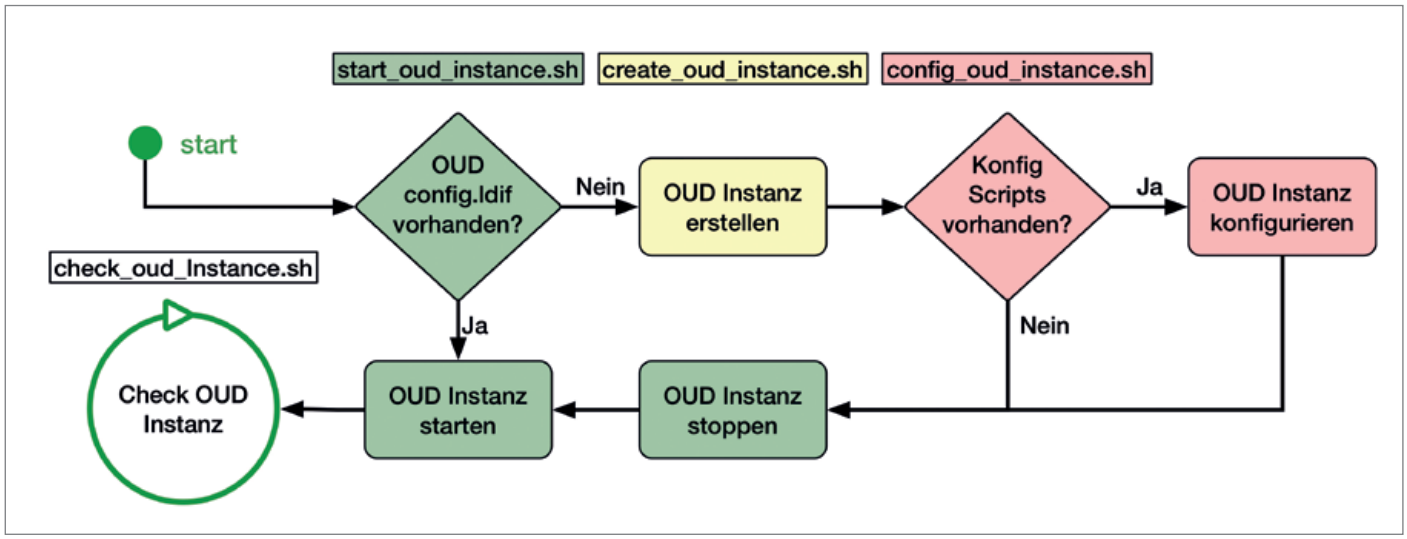


Abbildung 4: Ablauf beim Start der OUD-Instanz

- „setup_oudbase.sh“ wird beim Build des Docker-Image zur Installation der OUD-Base-Umgebungsskripte genutzt.
- „start_oud_instance.sh“ startet die OUD-Instanz und definiert Signal-Handler für „SIGINT, „SIGTERM“ und „SIGKILL“. Fehlt die OUD-Instanz, wird „create_oud_instance.sh“ aufgerufen.

Die Parametrisierung der Skripte erfolgt grundsätzlich via Umgebungsvariable bei der Instanziierung des Docker-Containers mit dem Parameter „-e“. Mit wenigen Ausnahmen ist eine nachträgliche Anpassung nicht sinnvoll. Ein Beispiel für diese Parametrisierung ist im folgenden Code-Beispiel ersichtlich. Dieser Aufruf von „docker run“ setzt explizit einen OUD-Instanz-Namen und einen alternativen BaseDN *(siehe Listing 3)*.

Neben „OUD_INSTANCE“ und „BASEDN“ können noch weitere Parameter zur Konfiguration des OUD-Containers beziehungsweise der OUD-Instanz genutzt werden. *Tabelle 1* gibt eine Übersicht.

Mit der Kombination der Umgebungsvariablen und den Konfigurationsskripten lassen sich verschiedene OUD-Instanzen für unterschiedliche Use-Cases anlegen. Fehlen die gewünschten Konfigurationsskripte, kann man mit „CREATE_INSTANCE=FALSE“ das automatische Erstellen der Instanz umgehen und manuell eine OUD-Instanz anlegen. Diese wird bei einem späteren Starten des Containers durch „start_oud_instance.sh“ automatisch gestartet. Läuft dabei etwas schief, hat man die Möglichkeit, direkt im Docker-Container die OUD-Log-Dateien zu prüfen oder mit „docker logs“ die OUD-Instanz-Log-Datei anzuzeigen. Mit „docker logs <CONTAINER>“ sieht man zudem immer auch den „STDOUT“ der Hilfe-Skripte beim Erstellen, Starten oder Stoppen einer OUD-Instanz.

Das OUDSM-Image

Der Aufbau des OUDSM-Image erfolgt analog zum OUD-Image. Neben den OUD-Binaries müssen zusätzlich noch Oracle Fusion Middleware Infrastructure „p26269885_122130_Generic.zip“ sowie das aktuelle WebLogic-PSU „p27342434_122130_Generic.zip“ in den Build-Ordner kopiert werden. Anschließend kann das OUDSM-Docker-Image mit den Kommandos „cd OracleOUDSM/12.2.1.3.0“ und „docker build -t oracle/oudsm:12.2.1.3.0“ erstellt werden.

Der Ablauf beim Build ist ähnlich wie beim OUD-Docker-Image. Der wesentliche Unterschied ist das Installationsskript „setup_oudsm.sh“, das beim OUDSM-Image eine „Collocated“-Installation von Fusion Middleware Infrastructure und Unified Directory erstellt. Zudem sind die oben beschriebenen Hilfe-Skripte für die Bedürfnisse von WebLogic beziehungsweise dem OUDSM angepasst und entsprechend umbenannt worden. Aus einem „start_oud_instance.sh“ wurde dem zufolge ein „start_oudsm_domain.sh“. Zusätzlich gibt es für das Erstellen der OUDSM-WebLogic-Domain das Python-Skript „create_oudsm_domain.py“. Das OUDSM-Image wird ausschließlich für den Betrieb eines Oracle Unified Directory Services Manager verwendet. Daher ist eine weitere Konfiguration grundsätzlich nicht nötig. Der gewünschte OUDSM-Container kann direkt mit dem Kommando in *Listing 4* erstellt werden.

Bei diesem Container wurde durch die Verwendung von „--volume“ ebenfalls explizit ein Volume angegeben. Die Zuweisung der Ports wurde mit dem Parameter „-P“ jedoch Docker überlassen. Mit dem Kommando „docker port“ können die verwendeten Ports nach dem Start des Containers angezeigt werden. Für den oben gestarteten Container wurden folgende Ports gewählt *(siehe Listing 5)*. *Abbildung 5* zeigt, wie man sich via „localhost“ auf den Port 32769 verbindet und mit OUDSM auf den zuvor erstellten OUD-Container „oudeng“ zugreift.

```
docker run --detach \
--volume /Data/vm/docker/volumes/oudeng:/u01 \
-p 1389:1389 -p 1636:1636 -p 4444:4444 \
--hostname oudeng --name oudeng \
oracle/oud:12.2.1.3.0
```

Listing 3

Variable	Verwendung
OUD_INSTANCE	OUD-Instanz-Name, Standardwert „oud_docker“
CREATE_INSTANCE	Flag zum Erstellen der OUD-Instanz beim Starten, Standardwert „TRUE“
OUD_INSTANCE_BASE	Basis-Verzeichnis für die OUD-Instanzen, Standardwert „$ORACLE_DATA/instances“
OUD_INSTANCE_HOME	OUD-Instanz-Home-Verzeichnis, Standardwert „${OUD_INSTANCE_BASE}/${OUD_INSTANCE}“
OUD_INSTANCE_ADMIN	OUD-Instanz-Admin-Verzeichnis, Standardwert „${ORACLE_DATA}/admin/${OUD_INSTANCE}“
ADMIN_USER	OUD-Directory-Root-Benutzer, Standardwert „cn=Directory Manager“
ADMIN_PASSWORD	Password für den OUD-Directory-Root-Benutzer, wird automatisch generiert und in „$PWD_FILE“ abgelegt. Durch Setzten von „ADMIN_PASSWORD“ kann explizit ein Passwort angegeben werden, das allerdings im Container als Umgebungsvariable gesetzt bleibt.
PWD_FILE	Password-File mit dem Clear Text Password vom Directory-Root-Benutzer, Standardwert „${OUD_INSTANCE_ADMIN}/etc/${OUD_INSTANCE}_pwd.txt}“
SAMPLE_DATA	Flag für das Erstellen der Beispieldaten, mögliche Werte sind „TRUE“, „FALSE“ oder Anzahl der Datensätze, Standardwert „TRUE“
OUD_INSTANCE_INIT	Verzeichnis für die Instanz-Konfigurationsdateien, Standardwert „$ORACLE_DATA/scripts“

Tabelle 1

```
docker run --detach \
--volume /Data/vm/docker/volumes/oudsm:/u01 \
-P --hostname oudsm --name oudsm \
oracle/oudsm:12.2.1.3.0
```

Listing 4

```
docker port oudsm
7002/tcp -> 0.0.0.0:32768
7001/tcp -> 0.0.0.0:32769
```

Listing 5

Die korrekte IP-Adresse des Containers „oudeng“ erhält man durch das Kommando „docker inspect <CONTAINER>“. Wenn man nicht alle Informationen zum Container anzeigen will, kann man dies, wie im Beispiel in *Listing 6*, direkt auf die IP-Adresse filtern.

OUD Base

Wie angemerkt, wird bei den Docker-Images für OUD und OUDSM sowie ODSEE immer OUD Base mit installiert. Es ist eine kleine Sammlung von Skripten, Umgebungsvariablen und Aliasen für die Kommandozeilen-Administration der Oracle Directory Server. Für den Betrieb der Docker-Container ist OUD Base nicht zwingend erforderlich. Gleichwohl erleichtert es einem die Arbeit auf der Kommandozeile.

Speziell wenn mehrere OUD-Instanzen konfiguriert wurden, liefern die Aliase „oud_status“ oder „oud_up“ immer Informationen wie Instanz-Name, Status, Oracle-Home, Ports etc. *Abbildung 2* weiter oben zeigt eine entsprechende Ausgabe von „oud_status“ beim Log-in. Weitere Informationen sowie Installationspakete findet man im OUD Base GitHub Repository des Autors *[5]*.

Die Größe der Docker-Images optimieren (für Fortgeschrittene)

Bei dem aufgezeigten Vorgehen für die Erstellung der Docker-Images für OUD und OUDSM wurde nicht speziell auf die Größe geachtet und die Software einfach im Build-Ordner abgelegt. Das „COPY“-Kommando im „Dockerfile“ stellt bei der Ausführung von „docker build“ sicher, dass die Software in das dazugehörige Image kopiert wird. Dies führt dazu, dass die Images für OUD und OUDSM beträchtlich größer werden als eigentlich nötig, was an der Art und Weise liegt, wie Docker die Images schichtweise aufbaut.

Ein Löschen auf einem nachfolgenden Layer bewirkt nur ein logisches Löschen, physisch bleibt die Software weiterhin Bestandteil des Layers. Eine einfache Lösung dieses Problems ist der Parameter „—squash“ bei neueren Docker-Versionen. Der Parameter wird zusätzlich bei „docker build“ angegeben und stellt sicher, dass alle neu erstellten Layer am Ende zusammengeführt werden. So können die oben aufgeführten „docker build“-Kommandos einfach mit „—squash“ ergänzt werden, um kleinere Images zu erhalten.

Alternativ dazu bietet es sich an, die Software beim Erstellen des Image direkt in einem „RUN“-Kommando herunterzuladen, auszupacken, zu installieren und am Schluss auch gleich zu löschen. Die verwendeten Setup-Skripte „setup_oud.sh“ und „setup_oudsm.sh“ sind entsprechend vorbereitet und können die Software nach dieser Methode installieren. Dazu muss beim Build mit dem Parameter „--add-host“ lediglich ein Host

```
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' oudeng
```

Listing 6

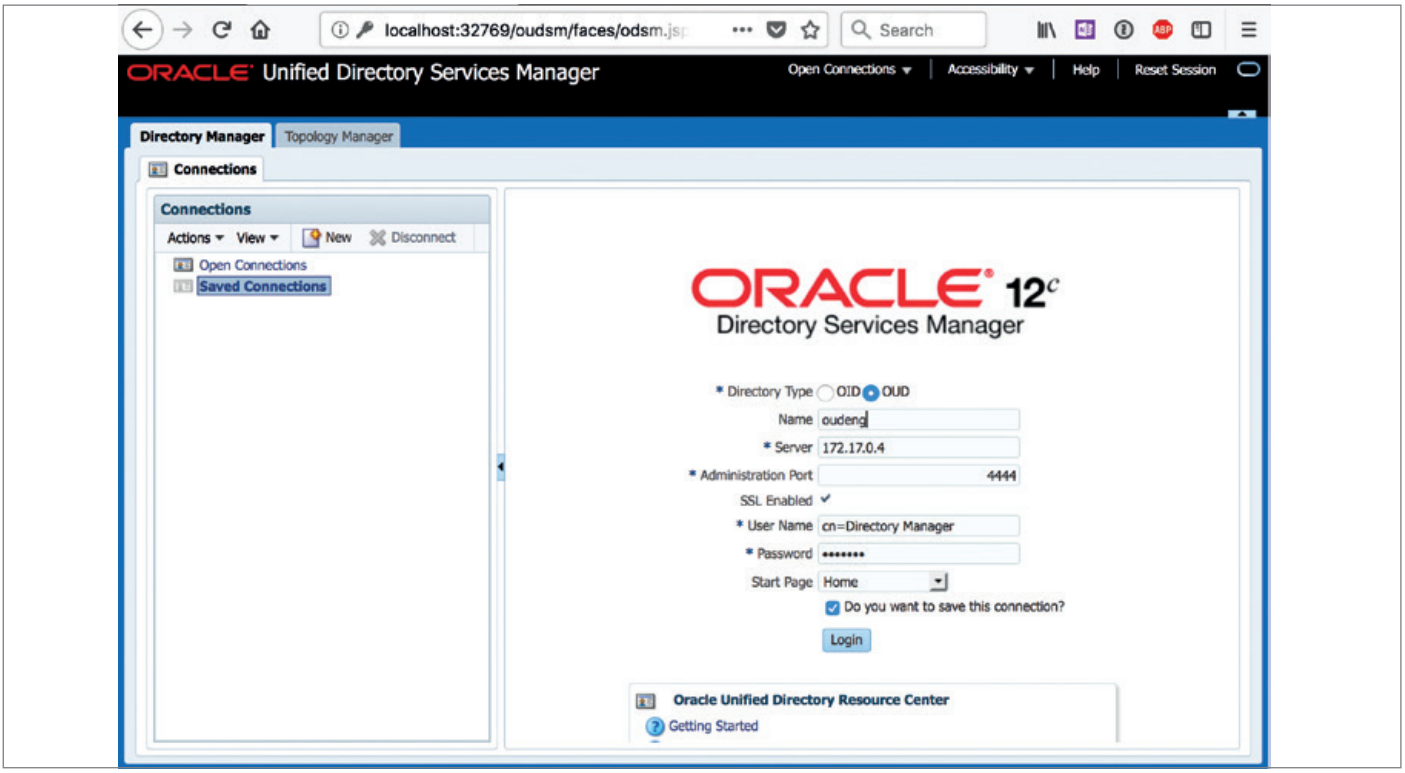


Abbildung 5: Log-in im Oracle Unified Directory Services Manager

```
cd OracleOUD/12.2.1.3.0
rm *.zip
docker build --add-host=orarepo:172.17.0.2 \
-t oracle/oud:12.2.1.3.0 .
```

Listing 7

beziehungsweise dessen IP-Adresse angegeben werden, von wo die Software heruntergeladen werden kann. Das Beispiel in *Listing 7* zeigt ein angepasstes Build-Kommando mit der Angabe einer IP-Adresse für den Host „orarepo“.

Im Beispiel handelt es sich ebenfalls um einen Docker-Container mit einem Apache Webserver, der nichts anderes macht, als lokal via HTTP die Software zur Verfügung zu stellen. Mehr dazu im Blog-Post „Smaller Oracle Docker Images“ *[7]*.

Fazit

Egal, ob man Oracle Unified Directory nur einmal anschauen möchte oder einen konkreten Anwendungsfall verfolgt – mit der hier vorgestellten Methode lässt sich Oracle Unified Directory schnell und einfach in einem Docker-Container aufsetzen und betreiben. Die Möglichkeit, dass die OUD-Instanzen manuell oder mit vordefinierten Skripts konfiguriert werden können, liefert die nötige Flexibilität, spezifische Bedürfnisse abzudecken.

Die verschiedenen Images für OUD, OUDSM und ODSEE wurden im Rahmen von Kunden-Projekten bereits mehrfach genutzt, um konkrete Problemstellungen zu lösen, etwa eine Ad-hoc-Test-Instanz für die Analyse von OUD-Problemen im Rahmen eines Service-Request, Test-Migration von ODSEE nach OUD, Entwicklung eines OUD-Proxy für die Integration weiterer Verzeichnisse, Einsatz von Oracle Enterprise User Security mit OUD und vieles mehr.

Mit einem „docker run“ ist ein neuer OUD-Container schnell erstellt und man hat Zeit, sich auf das Wesentliche zu konzentrieren, ohne sich zuerst mit der Installation zu befassen. Zudem steht einer produktiven Nutzung von OUD auf Docker grundsätzlich nichts im Weg. Auch wenn Oracle bis heute noch keine offiziellen OUD-Docker-Images zur Verfügung stellt, wird gemäß Oracle-Support OUD auf Docker grundsätzlich unterstützt.

Quellen

[1] Oracle Container Registry: *https://container-registry.oracle.com*
[2] Oracle Docker GitHub: *https://github.com/oracle/docker-images*
[3] Blog Post OUD to go on Raspberry Pi Zero: *http://url.oradba.ch/2AHzfbi*
[4] My Oracle Support OUD 12c Zertifizierung: *https://url.oradba.ch/MOSoud12c*
[5] Stefan Oehrli GitHub Docker Repository: *https://github.com/oehrlis/docker*
[6] OUD-Umgebungs-Skripte: *https://github.com/oehrlis/oudbase*
[7] Blog Post Smaller Oracle Docker Images: *https://url.oradba.ch/2ut6jEH*

Stefan Oehrli
stefan.oehrli@trivadis.com

Administration der Oracle-Datenbank mit Gewaltenteilung

Matthias Mann, DOAG Competence Center Security

Zentrales Thema vieler rechtlicher Vorschriften und Regularien zum sicheren Betrieb der Oracle-Datenbank sind die Risiko-Minimierung mittels Durchsetzung von „Least-Privilege" sowie die „Vier-Augen"-Prinzipien. Beides kann durch sogenannte „Gewaltenteilung" erreicht werden. Der Artikel erläutert an einem einfachen Beispiel, wie dies mit Oracle Database Vault umgesetzt werden kann.

In Wikipedia *[1]* lesen wir über „Gewaltentrennung": „Separation of duty, as a security principle, has as its primary objective the prevention of fraud and errors. This objective is achieved by disseminating the tasks and associated privileges for a specific business process among multiple users." Diese Definition bringt das Prinzip auf den Punkt: Der „Business Prozess" ist bei uns ein bestimmter Vorgang in der Datenbank-Administration. Die erzwungene Zusammenarbeit bei der Durchführung dieses Vorgangs von zwei oder mehreren funktionalen Administratoren realisiert ein Vier-Augen-Prinzip. Die Oracle-Datenbank setzt dies durch das Produkt „Database Vault" um.

Database Vault auf einen Blick

Database Vault ist eine kostenpflichtige Option zusätzlich zur Enterprise Edition. Die Grundelemente zur Durchsetzung des „least-privilege"-Prinzips sind „Realms", „Command Rules", „Rules", „Rulesets" sowie „Factors". Zum Erzwingen des Vier-Augen-Prinzips sind zentrale Datenbank-Rollen um wesentliche Rechte beschränkt. Das wichtigste Beispiel ist die Herauslösung der User- und Datenbankprofil-Managementrechte aus der Rolle „DBA". Oracle Database Vault ist im Kernel realisiert und garantiert dadurch eine Rechtekontrolle zur Laufzeit. Es kann nur mit einem Datenbank-Restart deaktiviert werden.

„Realms" sind abgeschottete Bereiche innerhalb der Rechte-Struktur des Datenbank-Katalogs, die die mächtigen „ANY"-Privilegien blockieren. Dies gilt sowohl beim Zugriff eines Users (wie des DBA) von außerhalb auf Daten im „Realm", aber genauso auch in der umgekehrten Richtung. Es sei aber betont, dass ungeachtet dieser Beschränkungen die Vergabe von Objekt-Privilegien genauso wie in einer normalen Datenbank funktioniert.

Durch „Command-Rules" lassen sich ganz gezielt einzelne Befehle kontrollieren oder deaktivieren. „Factors" erlauben, die Kontrolle des Datenbank-Zugriffs mit Umgebungs-Parametern zu kombinieren. Man nennt den Eingriff und die ständige Kontrolle des Kernels im Rechte-Management „mandatory access control". Um einen nahezu vollständigen Schutz der Datenbank zu gewährleisten, ergibt der Einsatz von Database Vault nur wirklich Sinn, wenn gleichzeitig die Daten mit Transparent Data Encryption verschlüsselt sind.

Die Administrator-Accounts mit Database Vault

Die *Tabelle 1* gibt einen Überblick über die wesentlichen Administratoren in der

Datenbank-Rollen	Management von	Administrations-Verantwortung
DV_OWNER DV_ADMIN	Rulesets Rules Command Rules Factors	Security-Administrator
DV_ACCTMGR	User, Profile	User-Administrator
DBA, sysdba	System-Privilegien Startup/Shutdown Backup/Recovery Tuning Space Management Datenbank-Patching und -Upgrades Tablespace Encryption	Datenbank-Administrator
DV_REALM_OWNER	Applikationsschema-Upgrades Rechte-Verwaltung für Applikation	Schema-Administrator

Tabelle 1: Administratoren in der mit Database Vault geschützten Datenbank

Administrationsbereich	verantwortlich / beteiligt	Einschränkung für den DBA
physisches Datenbank-Backup	DBA	keine
Objekt-Statistiken	Schema-Administrator, DBA	keine
Performance-Tuning	Schema-Administrator, DBA	kein Datenzugriff
Space Management	DBA	keine
Upgrades/Patching	DBA, Security-Administrator	nur mit Rolle DV_PATCH_ADMIN
Userverwaltung	User-Administrator	nicht möglich
Database-Vault-Administration	Security-Administrator	nicht möglich
Schema-Verwaltung	Schema-Administrator	nicht möglich
logisches Schema-Backup (Datapump)	Schema-Administrator	nicht möglich

Tabelle 2: Einschränkungen für den DBA in einer mit Database Vault geschützten Datenbank

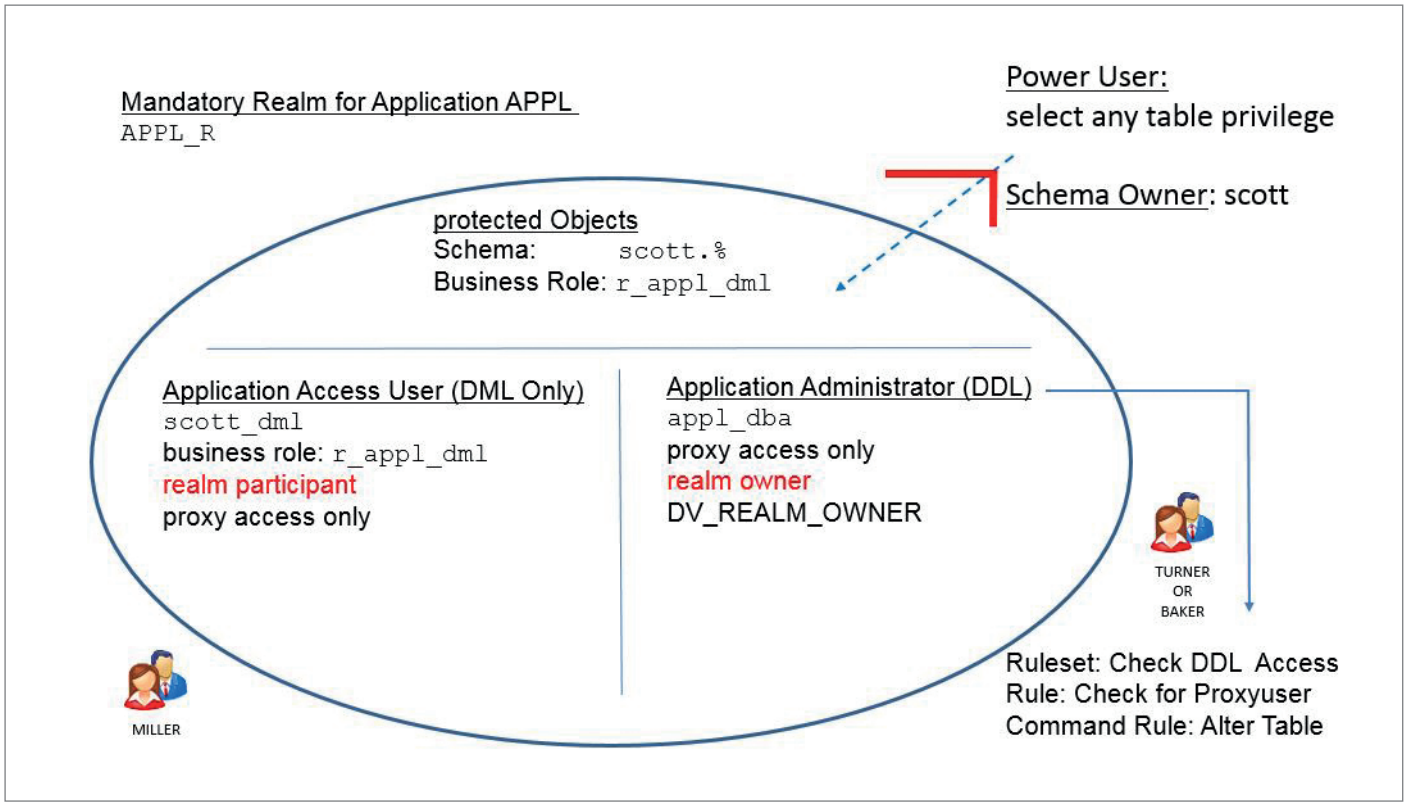


Abbildung 1: Schutz eines Applikationsschemas mit Database Vault

Username	Privilegien	Funktion
DB_ADMIN	DBA	Database-Administrator
DVOWNER	DV_OWNER	Security-Administrator
DVACCTMGR	DV_ACCTMGR	User-Administrator
SCOTT	RESOURCE	Schema-Eigentümer
SCOTT_DML	DML-Rechte auf Objekte von SCOTT	Applikations-Zugriffsuser
APPL_DBA	DV_REALM_OWNER	Schema-Administrator
MILLER		Proxy-User für „scott_dml“
TURNER	Verbot von „alter table“	Proxy-User für „appl_dba“
BAKER	„alter table“ via Ruleset	Proxy-User für „appl_dba“

Tabelle 3: Beteiligte Administratoren beim Schutz eines Applikationsschemas mit Database Vault

Listing	Schritt	Administrator
1	Erzeugen eines Datenbank-Profils und Anlegen der Funktions-Accounts „scott”, „scott_dml” und „appl_dba”	dvacctmgr
2	Vergabe der System-Privilegien	db_admin
3	Laden des Schemas „scott“ und Autorisierung von „scott_dml“	scott
4	„realm”-Konfiguration des Realm „appl_r”	dvowner
5	Konfiguration und Realm-Schutz einer Rolle mit Objekt-Privilegien	scott, dvowner
6	Konfiguration der Proxyprivilegien	dvowner, dvacctmgr
7	Dem Schema-Owner „scott“ wird die Realm-Autorisierung entzogen und das Realm „appl_r“ wird als „mandatory“ geflaggt	dvowner
8	Konfiguration der Command Rule „Check DDL Access” und der Database-Vault-Rule „Check for Proxy User”	dvowner
9	Prüfung der Realm-geschützten Objekte und Autorisierungen	dvowner

Tabelle 4: Schrittweiser, arbeitsteiliger Aufbau des Test-Setups

```
connect db_admin
-- authorize schema owner
grant resource to scott;
-- application role management in charge of schema owner
grant create role to scott;
grant drop any role to scott;
-- not mandatory, but recommended for application DBA
grant DV_REALM_OWNER to appl_dba;
```

Listing 2: Vergabe der System-Privilegien

Oracle-Datenbank, ihre Verantwortungsbereiche und die daraus resultierenden Unterschiede zur Administration in einer ungeschützten Datenbank. Ein komplementärer Blick auf die Datenbank-Administration ergibt sich, wenn man die Verantwortlichkeiten und Einschränkungen für den DBA bei den wesentlichen Prozessen im Vergleich zu einer Datenbank ohne Database Vault in *Tabelle 2* betrachtet.

Beispiel: Schutz eines Applikationsschemas

Wir erstellen nun Schritt für Schritt eine Konfiguration, die folgende Merkmale beinhaltet *(siehe Abbildung 1):*

- Der DBA soll keine Applikationsdaten mehr einsehen können
- Ein Schema-Administrator (Applikations-DBA) wird etabliert
- Der DML-Zugriff auf das Schema soll nur per Proxy Connect möglich sein
- Der Schema-Owner soll seine Daten nicht einsehen können
- DML- und DDL-Rechte sollen getrennt werden
- Ein Ruleset soll das „alter table"-Kommando und eine Rule dies nur durch Proxy-Anmeldung autorisieren.

Tabelle 3 gibt eine Zusammenfassung der beteiligten Administratoren und User. Dazu einige Bemerkungen:

- Database-Vault-Rechte sind von den Standard-Datenbank-Rechten zu unterscheiden. Im Beispiel sehen wir die Database-Vault-Rechte „realm owner" und „realm participant". Ein „realm owner" ist im Gegensatz zu einem „realm participant" berechtigt, Objekt-Privilegien auf seine eigenen Objekte zu vergeben (im Gegensatz zu einer unge-

```
connect dvowner
-- create Realm structure appl_r
begin
  dbms_macadm.create_realm(
            realm_name    => 'appl_r'
           ,description   => 'Application APPL'
           ,enabled       => dbms_macutl.g_yes
           ,audit_options => dbms_macutl.g_realm_audit_fail);
  end;
/
-- protect objects of schema scott
begin
  dbms_macadm.add_object_to_realm(
              realm_name   => 'appl_r'
             ,object_owner => 'scott'
             ,object_name  => '%'
             ,object_type  => '%');
  end;
/
-- authorize scott as "realm owner"
begin
  dbms_macadm.add_auth_to_realm(
              realm_name   => 'appl_r'
             ,grantee      => 'scott'
             ,auth_options => dbms_macutl.g_realm_auth_owner);
  end;
/
-- protect objects of access user scott_dml
begin
  dbms_macadm.add_object_to_realm(
              realm_name   => 'appl_r'
             ,object_owner => 'scott_dml'
             ,object_name  => '%'
             ,object_type  => '%');
  end;
/
-- authorize scott_dml as "realm participant"
begin
  dbms_macadm.add_auth_to_realm(
        realm_name   => 'appl_r'
       ,grantee      => 'scott_dml'
       ,auth_options => dbms_macutl.g_realm_auth_participant);
  end;
/
-- authorize appl_dba as "realm owner"
begin
  dbms_macadm.add_auth_to_realm(
              realm_name   => 'appl_r'
             ,grantee      => 'appl_dba'
             ,auth_options => dbms_macutl.g_realm_auth_owner);
  end;
/
-- authorize appl_dba for DDL on schema scott
exec dbms_macadm.authorize_ddl('appl_dba','scott');
```

Listing 4: Realm-Konfiguration des Realm „appl_r"

```
connect scott
create role r_appl_dml not identified;
grant r_appl_dml to scott_dml;
grant select, insert, update, delete on dept, emp, … to r_appl_dml;
create synonym scott_dml.dept for scott.dept;
…
connect dvowner
-- add role to realm
begin
  dbms_macadm.add_object_to_realm(
              realm_name   => 'appl_r'
             ,object_owner => '%'
             ,object_name  => 'R_APPL_DML'
             ,object_type  => 'ROLE');
  end;
/
```

Listing 5: Konfiguration und Realm-Schutz einer Rolle mit Objekt-Privilegien

```
connect dvowner
-- new with 12c:
-- authorize proxy permissions via Database Vault
exec dbms_macadm.authorize_proxy_user('MILLER','SCOTT_DML');
exec dbms_macadm.authorize_proxy_user('BAKER','APPL_DBA');
exec dbms_macadm.authorize_proxy_user('TURNER','APPL_DBA');
connect dvacctmgr
-- configure proxy permissions
alter user scott_dml proxy only connect;
alter user scott_dml grant connect through miller;
alter user appl_dba proxy only connect;
alter user appl_dba grant connect through baker;
alter user appl_dba grant connect through turner;
```

Listing 6: Konfiguration der Proxy-Privilegien

```
connect dvowner
-- unauthorize schema owner
begin
  dbms_macadm.delete_auth_from realm(
              realm_name => 'appl_r'
             ,grantee    => 'SCOTT');
end;
/
begin
  dbms_macadm.update_realm(
              realm_name    => 'appl_r'
             ,description   => 'Realm for Application APPL'
             ,enabled       => dbms_macutl.g_yes
             ,audit_options => dbms_macutl.g_realm_audit_fail
             ,realm_type    => 1);
  end;
/
```

Listing 7: Mandatory-Realm

```
connect dvowner
begin
  dbms_macadm.create_rule_set(
  rule_set_name   => 'Check DDL Access',
  description     => 'Check DDL Access',
  enabled         => 'Y',
  fail_message    => 'Cannot access SCOTT Objects',
  fail_code       => -20100);
  end;
/
-- connect the command "alter table" with the ruleset
begin
  dbms_macadm.create_command_rule
             (command        => 'ALTER TABLE'
            ,rule_set_name   => 'Check DDL Access'
            ,object_name     => '%'
            ,object_owner    => 'SCOTT'
            ,enabled         => 'Y');
end;
/
-- create a Database Vault Rule and attach to ruleset
begin
  dbms_macadm.create_rule(
  rule_name => 'Check for Proxyuser',
  rule_expr => 'SYS_CONTEXT(''USERENV'',''PROXY_USER'') = ''TURNER''');
  COMMIT;
end;
/
begin
  dbms_macadm.add_rule_to_ruleset(
              rule_set_name      => 'Check DDL Access'
             ,rule_name          => 'Check for Proxyuser');
  end;
/
```

Listing 8: Konfiguration der Command-Rule „Check DDL Access" und der Database-Vault-Rule „Check for Proxy User"

```
Realm appl_r Protected Objects
appl_r      SCOTT      %            %
            %          ROLE         R_APPL_DML
            SCOTT_DML  %            %

Realm appl_r Authorizations
appl_r      APPL_DBA   Owner
            SCOTT_DML  Participant
```

Listing 9: Prüfung der Realm-geschützten Objekte und Autorisierungen

schützten Datenbank). Das Database-Vault-Recht „realm owner" ist nicht mit der Datenbank-Rolle „DV_REALM_OWNER" zu verwechseln.

- Bei einem „mandatory realm" können Rechte auf Daten-Inhalte von geschützten Objekten nur auf „realm"-autorisierte Benutzer vergeben werden.
- Rollen, die Objekt-Privilegien auf „realm"-geschützte Objekte beinhalten, sind separat im „realm" zu schützen. Der Erzeuger einer Datenbank-Rolle ist, wenn kein „revoke" vorgenommen wird, automatisch Grantee dieser erzeugten Rolle. Aus diesen beiden Gründen empfiehlt sich, das Rollen-Management der applikationsbezogenen Rollen mit dem Schema-Owner durchzuführen sowie das Erzeugen und Löschen solcher Rollen im Normalbetrieb durch Database-Vault-Rulesets zu unterbinden, um ein versehentliches Anlegen von Rollen durch den DBA zu verhindern.
- Es wird deutlich, dass die einzelnen Schritte des Setups mit den verschiedenen Administrations-Usern durchgeführt werden müssen. Hier zeigt sich in der Praxis, ob die Prozesse im Unternehmen daran angepasst sind.

Aufbau des Setups

Tabelle 4 zeigt den schrittweisen und arbeitsteiligen Aufbau des Test-Setups. Die wichtigsten Schritte sind in den abgedruckten Listings in gekürzter Form wiedergegeben. Die vollständigen Listings stehen Ihnen unter *www.doag.org/go/red_stack/201804/mann/listings* zum Download zur Verfügung.

```
connect db_admin
select * from scott.dept;
ERROR at line 1:
ORA-01031: insufficient privileges
```

Listing 10: Schutz der Daten vor DBA-Zugriff

```
connect scott
select * from scott.dept;
              *
ERROR at line 1:
ORA-01031: insufficient privileges
```

Listing 11: Entzug der DML-Rechte vom Schema-Owner

Test	Ziel	Listing
A	Unterbindung des DBA-Zugriffs auf das Schema „scott“	10
B	Entzug der DML-Rechte vom Schema-Owner „scott“ durch den Entzug der Realm-Autorisierung und das Flaggen des Realm „appl_r“ als „mandatory“	11
C	Test des Ruleset „Check DDL Access“ Der „alter table“-Befehl darf nur von Benutzern ausgeführt werden, die durch die Rule „Check for Proxy User“ dazu autorisiert sind. In unserem Setup ist das der User „turner“.	12

Tabelle 5: Test des aufgebauten Setups

```
connect baker[appl_dba]
select sys_context('USERENV','SESSION_USER') SessionUser,
       sys_context('USERENV','PROXY_USER') ProxyUser,
       sys_context('USERENV','CURRENT_SCHEMA') CurrentSchema from dual;
SESSIONUSER     PROXYUSER       CURRENTSCHEMA
--------------- --------------- ---------------
APPL_DBA        BAKER           APPL_DBA
alter table scott.emp modify (deptno number(3));
*
ERROR at line 1:
ORA-47306: 20100: Cannot Access SCOTT Objects

connect turner[appl_dba]
select sys_context('USERENV','SESSION_USER') SessionUser,
       sys_context('USERENV','PROXY_USER') ProxyUser,
       sys_context('USERENV','CURRENT_SCHEMA') CurrentSchema from dual;
SESSIONUSER     PROXYUSER       CURRENTSCHEMA
--------------- --------------- ---------------
APPL_DBA        TURNER          APPL_DBA
alter table scott.emp modify (deptno number(3));
Table altered.
```

Listing 12: Test des Ruleset „Check DDL Access“

Test des Setups

Die Tests in *Tabelle 5* zeigen die Wirkung der Database-Vault-Konstruktion (siehe dazu auch *Abbildung 1*).

Fazit

Dieses einfache Beispiel zeigt wesentliche Unterschiede im Management einer durch Database Vault geschützten Datenbank im Gegensatz zum bekannten Verhalten in einer normalen Datenbank. Viele andere Einsatz-Szenarien sind mit Rulesets und Database-Vault-Faktoren (Umgebungsvariablen) denkbar.

Database Vault ist in der aktuellen Version 12c ein ausgereiftes technisches Produkt, das bei anderen Datenbank-Systemen seinesgleichen sucht und nach Erfahrung des Autors ohne Konkurrenz dasteht. Vor dem Einsatz ist es jedoch unausweichlich, sich über die im Unternehmen bei der Datenbank-Administration zu befolgenden Prozesse ein genaues Bild zu verschaffen sowie ein Konzept zu erstellen und umzusetzen, das diese Prozesse mit den technischen Erfordernissen des Produkts in Einklang bringt.

Referenz

[1] *https://en.wikipedia.org/wiki/Separation_of_duties*

Dr. Matthias Mann
matthias.mann@doag.org

Entspricht die Exadata den höchsten Sicherheitsanforderungen?

Borys Neselovskyi, OPITZ CONSULTING Deutschland GmbH

Die Oracle-Exadata ist für den Betrieb von hochkritischen, durchsatzintensiven Datenbanken nicht mehr wegzudenken. Mit einer hervorragenden Hardware-Ausstattung und eigenen Software-Lösungen wie etwa Smart Scans findet sie Einsatz in unternehmenskritischen Umgebungen, die gesondert behandelt werden sollen. Was die Perfomance betrifft, ist die Exadata also die beste Lösung für den Betrieb von Oracle-Datenbanken. Aber wird sie auch modernen Sicherheitsanforderungen gerecht? Diese Frage ist zu komplex, um in einem Satz mit „ja“ oder „nein“ beantwortet zu werden.

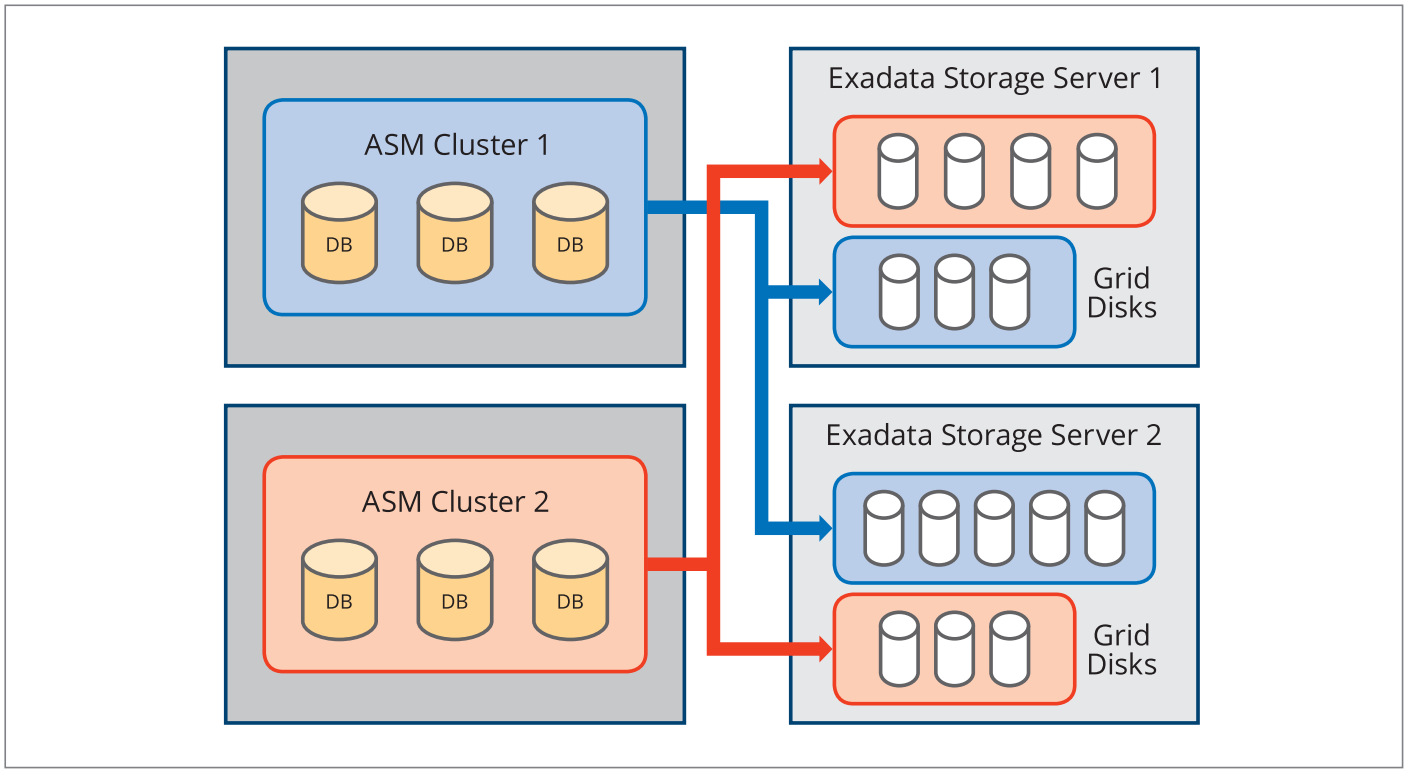


Abbildung 1: ASM-Scoped Security

Sicherheit hat viele Fassetten. Folgende Aspekte wurden für die Bewertung des Sicherheitsstandards einmal genauer angesehen:

- Ist eine Isolierung von unterschiedlichen Umgebungen innerhalb einer Exadata-Maschine möglich? Mit anderen Worten: Ist die Exadata mandantenfähig?
- Ist die Exadata nach der Installation sicher genug?
- Wie sieht es mit der Härtung der Betriebssysteme aller Komponenten aus?
- Sind Netzwerk-Komponenten sicherheitskonform installiert und konfiguriert?
- Entspricht die IT-Infrastruktur rund um die Exadata den Sicherheitsanforderungen?

Ist die Exadata mandantenfähig?

Die Isolierung verschiedener Umgebungen innerhalb der Exadata ist in vielen Fällen durchaus sinnvoll. So können zum Beispiel Datenbank-Anwendungen von unterschiedlichen Firmen (oder Abteilungen) komplett voneinander getrennt auf einer Exadata betrieben werden. Ein anderes Anwendungsszenario wäre die Trennung von Test- und Produktions-Umgebungen innerhalb einer Exadata: Da sich viele Unternehmen keine Exadata leisten können, die nur zu Testzwecken dient, ist die Trennung unterschiedlicher Umgebungen durchaus sinnvoll. Doch die technische Umsetzung ist aufwendig und kompliziert. Tiefgreifendes Know-how und exzellentes Verständnis der Arbeitsweise einer Exadata sind absolut notwendig, um die Isolierung zu vollziehen.

Eine Exadata besteht aus mehreren Komponenten und Modulen, die bei der Konfiguration der Isolierung einzeln betrachtet werden sollten. Die folgenden Begriffe sind für das Verständnis der Konfiguration der Mandantenfähigkeit sehr wichtig:

- *Datenbank-Server (engl. „Compute Node“)*
  Dieser besteht aus mindestens zwei physikalischen Maschinen, die für den Betrieb von Datenbanken verantwortlich sind. Die Datenbanken werden hier im Cluster betrieben. Die Cluster-Komponenten sind auf jedem Datenbank-Server installiert.

- *Storage-Server (engl. „Storage Cell“)*
  Dieser besteht aus mindestens drei Speicher-Systemen, die den Plattenplatz für die Speicherung von Daten in Datenbanken bereitstellen.

- *InfiniBand-Switches*
  Hierbei handelt es sich um Netzwerk-Switches für die Netzwerk-Kommunikation zwischen Datenbank- und Storage-Servern. In der Standard-Ausstattung sind zwei InfiniBand-Switches vorhanden, die als eine Einheit namens „InfiniBand Fabric“ für die Cluster-Kommunikation verantwortlich sind.

- *Automatic Storage Management (ASM)*
  Oracle ASM besteht aus einem Volume-Manager und einem Dateisystem für Oracle-Datenbank-Dateien, die den Inhalt des Storage aufbereiten und ihn Datenbanken zur Verfügung stellen.

- *Oracle-Clusterware*
  Alle Exadata-Hardware-Komponenten sind redundant ausgelegt und vor Ausfällen ausreichend geschützt. Die Datenbanken, Listener und andere notwendigen Dienste werden auf mehreren Datenbank-Servern in einem (oder mehreren) Real Application Clustern (RAC) betrieben. Wenn eine Datenbank-Instanz auf einem Datenbank-Server ausfällt, übernimmt die Instanz auf einem anderen

Cluster-Mitglied die Arbeit. Mit der Cluster-Funktionalität erhöht sich die Ausfallsicherheit der Datenbanken.

- *Interconnect (Cluster-Kommunikation)*
  Alle Cluster-Mitglieder kommunizieren miteinander über das sogenannte „Cluster Interconnect". Diese Kommunikation erfolgt über ein InfiniBand-Netzwerk. Alle Server haben InfiniBand-Netzwerkkarten und sind darüber mit beiden InfiniBand-Switches redundant verbunden.

- *Public- oder Client-Netzwerk*
  Über dieses Netzwerk erfolgt die Kommunikation zwischen Datenbanken und Anwender (beziehungsweise Anwendungen).

- *Virtualisierte Plattform/Bare Metal*
  Exadata kann als „Blech" (Bare Metal) oder virtualisiert betrieben werden. Die Virtualisierung bringt mehr Komplexität bei der Konfiguration und Lifecycle-Operationen mit sich. Auf der anderen Seite erlaubt die Virtualisierung mehr Flexibilität bei der Exadata-Konfiguration.

Nachdem die Begriffe klar sind, können wir die Konfiguration der Mandantenfähigkeit auf allen Ebenen besprechen.

Exadata: virtualisiert oder Bare Metal?

Die Trennung von Umgebungen innerhalb einer Exadata kann auf der Cluster-Ebene realisiert werden. In der Regel sollen zwei Cluster für die Isolierung sorgen. In der virtualisierten Exadata-Variante können auf jedem physikalischen Datenbank-Server beliebig viele virtuelle Server aufgesetzt sein, die in separate Cluster segregiert sind. In der Bare-Metal-Variante müssen physikalische Datenbank- und Storage-Server in zwei Cluster aufgeteilt sein. Dabei ist wichtig zu wissen, dass zu einem Cluster mindestens zwei Datenbank- und drei Storage-Server gehören. Dem entsprechen die Exadata-Modelle Half- und Full-Rack. Die kleineren Modelle Eight- und Quarter-Rack eignen sich dagegen nicht für die physikalische Isolierung. In diesem Fall bleibt nur die Option, die Exadata zu virtualisieren.

Trennung der Clusterkommunikation

In einer Zwei-Cluster-Umgebung muss auch die Cluster-Kommunikation getrennt stattfinden. Für diese Kommunikation ist die InfiniBand Fabric zuständig. Deren Hardware lässt sich nicht pro Cluster isolieren; lediglich die Netzwerk-Kommunikation kann in zwei Ströme unterteilt werden. Das Verfahren nennt sich „InfiniBand Partitionierung" und wird seitens Oracle nur in der virtualisierten Exadata-Variante unterstützt.

Isolierung von Storage-Bereichen pro Cluster beziehungsweise pro Datenbank

Die Isolierung von Clustern sollte auch auf der Storage-Ebene durchgeführt werden. Das Tool ASM gruppiert die physikalischen Platten logisch in sogenannte „Grid-Disks". Diese werden in ASM-Gruppen zusammengefasst und stellen Plattenplatz für die Datenbanken zur Verfügung. Der Zugriff auf den Storage wird durch folgende ASM-Modi geregelt:

- *Open Mode*
  Alle Datenbanken können auf alle vorhandenen Grid-Disks zugreifen; dabei gibt es keine Einschränkungen. Das ist die Standard-Einstellung, sie bietet keine Sicherheit auf der Storage-Ebene. Diese ASM-Konfiguration sollte daher nur für Test- oder Entwicklungs-Umgebungen verwendet werden.

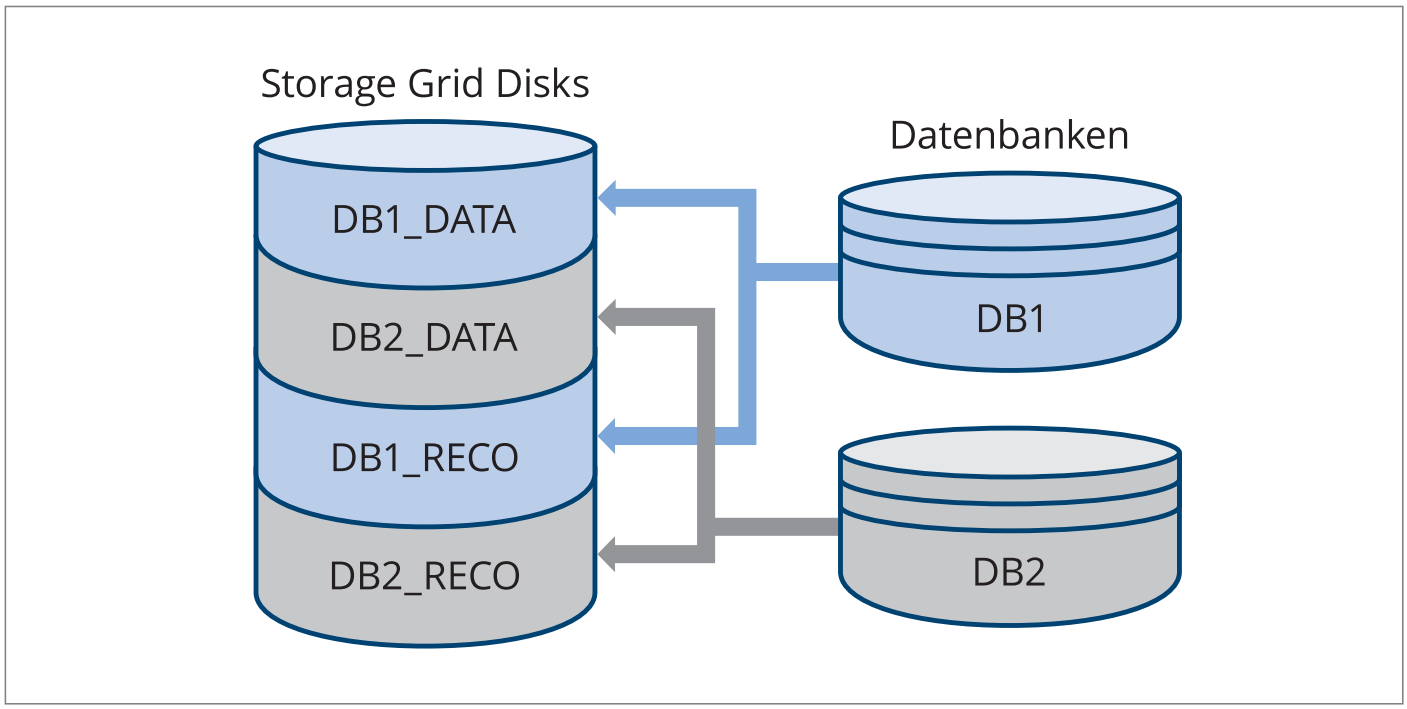


Abbildung 2: Database-Scoped Security

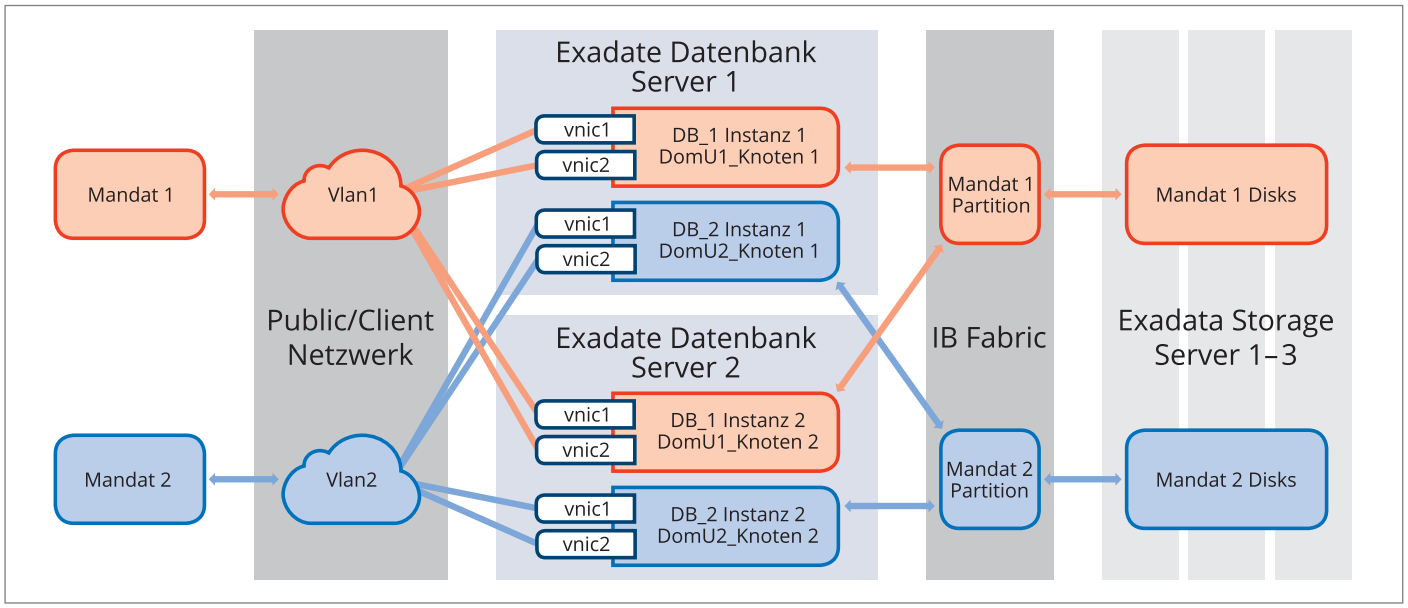


Abbildung 3: Exadata-Isolierung

- *ASM-Scoped Security-Mode*
  Grid-Disks werden einem Cluster zugeteilt und können nur von Datenbanken verwendet werden, die zu diesem Cluster gehören. Datenbanken aus einem anderen Cluster bekommen keinen Zugriff auf diesen Storage-Bereich *(siehe Abbildung 1)*.

- *Database-Scoped Security-Mode*
  Dieser Modus entspricht dem höchsten Sicherheitsstandard. Dabei wird jeder Datenbank ein eigener Plattenbereich zugeteilt. Dieser Modus ist eine Stufe härter als der ASM-Scoped Security-Mode *(siehe Abbildung 2)*.

Partitionierung des Public-Netzwerks

Der Zugriff auf Datenbanken in einer Exadata erfolgt über ein sogenanntes „Public- oder Client-Netzwerk". Wenn mehrere Cluster innerhalb einer Exadata betrieben werden, kann auch das Client-Netzwerk pro Cluster separiert werden, was durch das VLAN-Tagging-Verfahren realisiert wird. Dieses ermöglicht, ein bestehendes physikalisches Netzwerk zu partitionieren und mehrere virtuelle Netzwerke zu kreieren, die voneinander getrennt sind. VLAN Tagging basiert auf der 802.1Q-Technologie. Es ist nicht Exadata-spezifisch, sondern findet auch im Linux/Unix-Umfeld Verwendung. Die Partitionierung des Public-Netzwerks kann ab Exadata-Version 12.1.2.1.1 bei der initialen Exadata-Konfiguration mittels Oracle Exadata Deployment Assistant (OEDA) durchgeführt werden.

Abbildung 3 zeigt eine virtualisierte Exadata. Zwei Cluster (farblich rot und blau gekennzeichnet) wurden konsequent voneinander isoliert und zwar auf allen Ebenen: Der Client-Zugriff auf jedes Cluster erfolgt über eine virtuelle Partition des Public-Netzwerks. Die Cluster-Kommunikation wurde auch auf der InfiniBand-Ebene partitioniert und verläuft somit getrennt. Auf der Storage-Ebene hat jedes Cluster einen eigenen Plattenbereich.

Fazit: Eine Exadata ist mandantenfähig. Die konsequente Isolierung mehrerer Umgebungen innerhalb einer Exadata setzt aber folgende Maßnahmen voraus:

- Die Exadata muss virtualisiert sein.
- Das InfiniBand-Netzwerk muss partitioniert sein.
- ASM-Scoped Security-Mode oder Database-Scoped Security-Mode muss konfiguriert sein.
- Das Client-Netzwerk muss durch das VLAN-Tagging partitioniert werden.

Wie sicher ist die Exadata?

Die Software-Ausstattung eines Exadata-Datenbank-Servers besteht aus Betriebssystem, Cluster- und Datenbank-Software. Das Betriebssystem wird durch den Benutzer „root" administriert. Die Administration von Oracle Cluster Software und Datenbank erfolgt standardmäßig über den Benutzer „oracle". Für Umgebungen mit erhöhten Sicherheitsanforderungen sollten sogenannte „Separations of Duties" konfiguriert sein. Dabei wird die Administration von Cluster und Datenbank getrennt. Der Benutzer „grid" administriert den Cluster. Die Datenbank-Administration auf der Betriebssystem-Ebene erfolgt weiterhin über den Benutzer „oracle". Somit sind Cluster- und Datenbank-Bereiche voneinander getrennt. Der Nachteil dieser Lösung ist die erhöhte Komplexität der Umgebung und der größere Aufwand bei Administration und Lifecycle-Operationen.

Die initiale Installation einer Exadata beinhaltet eine Härtung aller Komponenten nach den besten Sicherheitsstandards. So sind sowohl auf Datenbank- als auch auf Storage-Servern nur die notwendigen Betriebssystem-Pakete installiert. Vergeblich sucht man Kommandos wie „telnet" oder „nc", um die Netzwerk-Konnektivität zu prüfen. Diese und viele andere Binaries sind per Default nicht installiert. Exadata-Administratoren können natürlich weitere Pakete nachinstallieren, müssen aber sicherstellen, dass dabei keine Sicherheitslücken geöffnet werden.

Überflüssige Linux-Dienste und -Protokolle sind per Default deaktiviert. So werden die Internet-Kommunikationsdienste „inetd"/„xinetd" auf der Exadata nicht gestartet. Die für den Exadata-Betrieb notwendigen Dienste wie Secure Socket Lay-

er (SSH) und Network Time Protokol (NTP) bieten erweiterte Möglichkeiten für Sicherheitskonfigurationen. Alte und unsichere SSH-Versionen werden nicht unterstützt.

Alle Verzeichnisse und Dateien sind mit den minimal notwendigen Berechtigungen ausgestattet. Daten, die auf der Exadata gespeichert und transportiert werden, können sicher verschlüsselt werden: Die Exadata-Linux-Server unterstützen den höchsten Verschlüsselungsstandard 140 of FIPS (Federal Information Processing Standards).

Auf allen Storage-Servern ist die Linux-Firewall „iptables" aktiviert. Direkte Verbindungen zu den Storage-Servern von außen sind per Default nie erlaubt. Die Datenbank-Server sind hingegen nicht mit einer aktiven Firewall ausgestattet. Für die Einhaltung von Sicherheits-Richtlinien sollten die Firewall-Regeln realisiert und der Dienst „iptables" aktiviert werden. Zudem sollte man die Daten über das Netzwerk verschlüsselt übertragen. Für die Kontrolle der Netzwerk-Zugriffe ist die Zugriffs-Tabelle („Access Control Lists") zu pflegen. Die Netzwerk-Switches sollten folgendermaßen nach Best Practices konfiguriert sein:

- AAA-Prinzip einhalten (Authentifizierung/Autorisierung/Accounting)
- Port Mirroring aktivieren: Netzwerk-Daten werden zusätzlich an eine Prüfstelle (Intrusion Detection System) für die Sicherheits-Analyse weitergeleitet (Intrusion Detection System)
- Auto-Trunking deaktivieren

Die Standard-Passwort-Richtlinien sind bereits sicher. Ein neues Passwort darf dem alten nicht ähnlich sein. Die Laufzeit eines Passworts beträgt 90 Tage. Die Komplexität ist wie folgt geregelt:

- Erlaubte Zeichen
  - Ziffern, Klein- bzw. Großbuchstaben, Sonderzeichen
- Passwortlänge
  - Bei der Verwendung der drei Zeichenklassen muss ein Passwort mindestens zwölf Zeichen lang sein
  - Bei der Verwendung der vier Zeichenklassen muss ein Passwort mindestens acht Zeichen lang sein

Fehlgeschlagene Anmeldungen an der Exadata werden wie folgt geregelt:

- Nach einem fehlgeschlagenen Log-in-Versuch wird der Account für zehn Minuten gesperrt.
- Nach fünf fehlgeschlagenen Log-ins wird der Benutzer dauerhaft gesperrt.

Als Betriebssystembenutzer „root" können gesperrte Benutzer mit dem Kommando „pam_tailly2" angezeigt und entsperrt werden:

- *Gesperrte Benutzer anzeigen*
  „/sbin/pam_tally2"

- *Benutzer entsperren*
  „/sbin/pamm_tally2 -user <username> --reset"

Die Exadata-Storage- und Datenbank-Server werden über das Tool „Integrated Lights Out Manager" (ILOM) administriert. Darüber können die Hardware-Komponenten geprüft und verwaltet werden. Das Tool kann über eine Web-Oberfläche oder über ein Command Line Interface (CLI) bedient werden: Der Zugriff auf ILOM über das Web-Interface erfolgt über einen Internet-Browser. Die Kommunikation ist mit SSL verschlüsselt. Der Zugriff auf ILOM CLI erfolgt entweder über Secure Socket Shell (SSH) oder Intelligent Platform Management Interface (IPMI). Sowohl bei SSH als auch bei IPMI wird die zweite Version verwendet, die erweiterte Sicherheitseinstellungen unterstützt.

Die Funktion „Standard-Betriebssystem-Auditing" kann mit Linux-Mitteln aktiviert werden. Auf dem Storage-Server gibt es die Möglichkeit, die erweiterte Auditing-Funktion zu aktivieren, indem man den Parameter „sysLogConf" setzt.

Sicherheit von Datenbank und Infrastruktur

Die Härtung der gesamten Infrastruktur sowie der Datenbanken ist eine wichtige Aufgabe. Diese Maßnahmen können sehr umfangreich sein. Die Härtung der Exadata sollte Teil des gesamten Sicherheitskonzepts sein, bei dem alle Elemente der Infrastruktur berücksichtigt werden.

Tipp 1: Der Zugriff auf die Exadata-Komponenten über privilegierte Betriebssystem-Benutzer sollte deaktiviert werden. Der Zugriff erfolgt mit einem nicht privilegierten Benutzer, der mithilfe des Werkzeugs „sudo" zu einem System-Account (wie etwa „root"/„oracle"/„grid") wechseln kann.

Tipp 2: Der direkte Zugang zu Exadata-Systemen mittels SSH sollte für Anwendungsbenutzer unterbunden werden.

Tipp 3: Für die Überwachung von Storage-Servern durch den Enterprise Manager sollte der Betriebssystembenutzer „cellmonitor" eingesetzt werden.

Tipp 4: Die direkte SSH-Verbindung zum Storage-Server kann deaktiviert werden. Alle administrativen Aufgaben können vom Datenbank-Server via ExaCLI oder REST-API ausgeführt werden.

Fazit

Die Exadata entspricht grundsätzlich den höchsten Sicherheitsanforderungen. Konfiguration und Implementierung sind allerdings sehr aufwendig und setzen einen hohen Kompetenzgrad des Personals voraus. Erforderlich ist zudem ein Konzept, das es ermöglicht, Exadata immer auf dem aktuellsten Stand zu halten. Unter anderem sollte darauf geachtet werden:

- Die Installation von Sicherheitspatches dauerhaft zu planen und zu terminieren
- Monitoring von neuen Sicherheitslücken in regelmäßigen Abständen durchzuführen
- Zeitnah die dafür vorgesehenen Lösungen zu implementieren, wenn Sicherheitslücken entdeckt werden.

Diese und weitere Maßnahmen, kontinuierlich geplant und umgesetzt, gewährleisten die Sicherheit der Exadata während des gesamten Lebenszyklus.

Borys Neselovskyi
borys.neselovskyi@opitz-consulting.com

Durchgecheckt

Bruno Cirone, DOAG Themenverantwortlicher Security

Es ist schon erstaunlich, was man als Oracle-Datenbank-Berater im Lauf seines Berufslebens so alles erlebt. Gerade im Security-Bereich gibt es immer wieder Aha-Erlebnisse, die einem die Sprache verschlagen. Zwei exemplarische Beispiele.

Beim ersten Beispiel wurde ich vom Deutschland-CIO eines amerikanischen, multinationalen Chemiekonzerns beauftragt, einen Security-Check bei einem seiner Tochterunternehmen vorzunehmen. Es sollte nur ein Tag dauern und einfach nur eine Geschmacksrichtung der Security wiedergeben. Also es war klar, er wollte von mir ein pauschales Okay erhalten – und dies natürlich, ohne größere Investitionen zu tätigen.

Aber was sollte ich überhaupt prüfen? Die Antwort war einfach und klar: „Natürlich alles." Selbstverständlich ist dies von einer Person gar nicht zu bewältigen. Insbesondere an nur einem Tag ist das absolut unmöglich und eher absurd. Daher hatte ich den Auftrag zunächst abgelehnt. Der CIO bat mich trotz meiner Bedenken, den Auftrag anzunehmen, was ich dann wegen meiner Neugier tat.

Das Tochterunternehmen war ein Forschungslabor und forschte unter anderem nach Additiven, um die Stoffe länger haltbar, kälteresistenter und geschmeidiger zu machen. Es war klar, dass diese Forschungsergebnisse (Patente) einen erheblichen Wert für das Unternehmen darstellten.

Nach der Beauftragung musste ich ein polizeiliches Führungszeugnis vorlegen und meine finanzielle Situation erklären. In dieser Zeit bekam ich verschiedene Anrufe von Auskunfteien, die alle mehr oder weniger das Gleiche wissen wollten: „Seit wann existiert Ihre Firma?", „Haben Sie Referenzkunden?" etc.

Etwa zwei Wochen vor dem Termin erhielt ich eine DVD mit einem Belehrungsfilm, der die Sicherheitsmaßnahmen erklärt und ebenso, an wen man sich wenden muss, falls man das Gefühl hat, dass etwas nicht in Ordnung ist. Ich musste diesen Film ansehen und danach online einen Prüfungsfragebogen fehlerfrei ausfüllen. Erst danach wurde der Termin final bestätigt.

Am Beratungstag musste ich meinen Pass beim Empfang hinterlegen und be-

kam einen Besucherausweis, danach wurde ich von einem Mitarbeiter abgeholt. Beim Laborkomplex angekommen, wurde mein Besucherausweis durch einen spezielleren Ausweis ausgetauscht.

Endlich ging es an dem eigenen Rechenzentrum vorbei in einen Besprechungsraum. Auf mich warteten etwa fünfzehn Wissenschaftler und Laborleiter. Ich fragte meinen Begleiter, ob die Server von der zentralen Administration betreut werden. Er meinte: „Natürlich nicht, wir sind hier in einem Hochsicherheitsbereich, da haben auch die internen Administratoren keinen Zugriff drauf." Ich war von den bisherigen Sicherheitsmaßnahmen beeindruckt. So einfach konnte keiner hier reinkommen.

Die Besprechung lief wie erwartet. Keiner der Teilnehmer hatte meinen Besuch gefordert oder gar gewünscht. Da galt es zunächst, die Angst vor dem Security-Thema zu nehmen. Die Besprechung wurde im Lauf des Tages immer angenehmer, aber auch anstrengender.

Nach rund zwölf sehr intensiven Stunden war die Besprechung kurz vor dem Ende. Ich hatte mein Notebook runtergefahren, die Unterlagen zusammengestellt und meinen Aktenkoffer schon fast geschlossen. Da hatte einer der Wissenschaftler doch noch eine Frage: „Können wird die Datenbank weiterhin über eine ODBC-Anbindung komplett auf das Notebook kopieren und dann mit Microsoft Access weiterarbeiten?" Ich sackte in meinem Stuhl zusammen und konnte nur noch ein „Wie bitte?" von mir geben. Der Wissenschaftler meinte: „Ich muss ja weiterarbeiten können, wenn ich im Zug unterwegs bin oder auf dem Flughafen auf meinen Flug warte." Ich dachte noch an einen letzten Strohhalm: „Haben Sie auf dem Notebook eine Verschlüsselungssoftware und wann wird der Bildschirm gesperrt?" Die Antwort war für mich nicht überraschend. „Die Verschlüsselungssoftware haben wir mal bestellt, aber irgendwie war die wohl zu teuer und zu aufwendig, daher nutzen wir sie nicht. Ich schalte meine Bildschirmsperre ab; es nervt, wenn ich dann alle paar Minuten mein Passwort eingeben muss." Ich konnte eine gewisse Frustration nicht verbergen.

Mein Abschlussbericht hatte danach durchaus etwas Positives für die Mitarbeiter. Alle bekamen innerhalb kürzester Zeit neue Hochsicherheits-Notebooks mit eigenen Telefonkarten. Auf jedem Notebook ist eine Oracle-Datenbank installiert, die auch nur ihre eigenen Daten beinhalten durfte.

Bei den Erlkönigen

Das zweite Beispiel stammt von einem Automobilzulieferer. Auch hier sollte ich einen schnellen Überblick über die Security-Situation liefern. Im Vorhinein wurde mir mitgeteilt, dass ich keine Kamera, Smartphone oder Telefon mit Kamera, Werkzeuge etc. mitbringen durfte. Am Werkseingang wurden mein Wagen und meine Unterlagen untersucht.

Während ich auf meine Abholung wartete, fuhren zwei LKW auf dem Werksgelände vor. Einer mit „Pampers"-Logo, der andere mit einem Logo einer Fernsehmarke. Ein Automobilzulieferer mit einem „Pampers"-LKW machte mich schon sehr neugierig, deshalb habe ich den Mitarbeiter, der mich abgeholt hat, danach gefragt. Er sagte, dass alle Erlkönige, also Fahrzeuge, die in etwa zwei bis drei Jahren auf den Markt kommen, in solchen Fahrzeugen zu ihnen gebracht werden. Es wäre ja nicht auszudenken, wenn Fotos, Videos etc. von den Erlkönigen auf ihrem Gelände gemacht werden würden. Sollte so etwas passieren, kündigen die Fahrzeughersteller den Vertrag mit den Zuliefererfirmen.

Im Werk selber gab es für diese Erlkönige einen speziellen, abgesicherten Bereich. Dort sind besondere Zutrittsausweise notwendig, die Fensterscheiben sind bleiverglast und die Techniker müssen Bleimatten über die Fahrzeuge legen, wenn sie nicht daran arbeiteten. Die LKW wurden entladen und ich konnte die Fahrzeuge sehen, die in zwei Jahren auf den Straßen fahren werden. Diese Fahrzeuge wurden mit Messtechnik vollgestopft.

Dieser Bereich hatte einen eigenen Oracle Server für etwa fünfzig Mitarbeiter. In der dazugehörigen Datenbank sind alle Daten wie Messergebnisse, Optimierungen, Anforderungen an Werkzeugen etc. festgehalten. Der Standort des Servers war allerdings nicht in diesem Hochsicherheitsbereich, sondern in einer Abstellkammer ohne besondere Sicherheitsvorkehrungen. Der Putzfrauenschlüssel diente zum Öffnen der Tür und war auch sinnvollerweise neben der Tür aufgehängt.

Nach einer kurzen Einführungsphase durfte ich meine Arbeit beginnen. Man hatte mir ein kleines Büro zur Verfügung gestellt. Die Login-Daten vom Oracle User wurden mir in einem versiegelten Umschlag übergeben. Das Passwort würde nach meinem Besuch wieder geändert. Sehr gut, bis auf die Abstellkammer war das vollkommen okay.

Im Laufe des Tages wollte ich mich als „root"-User auf dem System anmelden. Da ich das Passwort nicht wusste, habe ich einen der Techniker danach gefragt. Die Antwort war: „Das Passwort ist toor, also root rückwärts geschrieben". Was soll man dazu noch sagen? Dafür war aber der Oracle-User (auf dem Papier) bestens geschützt.

Fazit

Diese beiden Beispiele waren etwas ungewöhnlich, aber sie zeigen die Kernprobleme sehr deutlich. Die Investition in Security-Maßnahmen wird nicht immer als sinnvoll angesehen. Security kann ja nicht mit einem Return on Invest (ROI) schöngerechnet werden. Eine fehlende Sensibilisierung und (menschliche) Nachlässigkeiten führen zu erheblichen Security-Problemen. Letztlich kann man die notwendigen Maßnahmen auf diese Formel bringen: „Security kostet Geld, Performance und Ressourcen. Keine Security kostet noch mehr Geld und gefährdet das Unternehmen".

Bruno Cirone
bruno.cirone@doag.org

DSGVO heißt erst einmal: Ran an die Prozesse!

Die neue Datenschutz-Grundverordnung (DSGVO) nimmt vor allem massiv Einfluss auf die gesamten IT-Prozesse. Dazu ein Interview mit Sabine Rudolf, Head of Marketing & Product Sales, PROMATIS software GmbH

Wie war Ihre Herangehensweise für die Umsetzung der neuen DSGVO?

Eigentlich befasst sich die DSGVO ja nur mit der Verarbeitung von personenbezogenen Daten. Das hört sich nicht so gewaltig an, kratzt man jedoch ein wenig an der Oberfläche, zeigt sich, wo überall in dem enormen und verzweigten Netzwerk der Unternehmens-IT diese Daten zu finden sind. Um die Anforderungen gesetzeskonform umzusetzen, gibt es grundsätzlich zwei Möglichkeiten: entweder selbst die unzähligen Stellen in der umfassenden digitalen Unternehmenswelt mühsam zu suchen, um danach die Änderungen individuell zu realisieren, oder die Aufgabe an einen Experten zu übertragen. Da sich PROMATIS schon lange und intensiv mit den Datenschutz-Themen beschäftigt und Unternehmen bei der Implementierung und Einhaltung der europaweiten DSGVO-Richtlinien unterstützt, war es für uns ein Leichtes, die Realisierung selbst in die Hand zu nehmen. Hierfür konnten wir auf ein tiefes Verständnis der Gesetze sowie der Anwendungsbereiche gepaart mit fundiertem Know-how der gesamten Unternehmensprozesse und Datenstrukturen zurückgreifen. So haben wir uns bei der Vorgehensweise an der klassischen Handhabung orientiert: Analyse – Konzeption – Umsetzung. Eigentlich nichts Neues, wenn der Fokus des Verfahrens auf der Analyse liegt, denn je detaillierter und systematischer die Untersuchungen der Systeme durchgeführt werden, desto schneller und somit auch effizienter kann dann die Umsetzung erfolgen.

Waren Sie mit der Analyse zufrieden?

Ja, ein Vorteil unserer umfassenden Analyse lag in dem Erkennen von Schwachstellen, unnötigen Prozessen und Daten, redundanten Vorgängen und vielen weiteren Punkten, die eine IT-Landschaft belasten. Aus unserer täglichen Arbeit wussten wir, dass für die pragmatische und valide Erstellung der Konzepte eine systematische Intelligenz erforderlich ist, die sowohl die Gesetzesvorgaben als auch die Prozess-Strukturen innerhalb des Unternehmens kennt. Diese Methodik spiegelt sich auch in der Umsetzung wider, praxisorientierte Lösungen mit minimalem Aufwand zu implementieren. Das sahen wir als große Chance, die durch die Einführung der DSGVO bestand. Denn aufgrund der detaillierten und vor allem systematischen Betrachtung konnten Prozesse optimiert, Daten minimiert und die Transparenz erhöht werden.

Ist für die Umsetzung ein Datenschutzbeauftragter erforderlich?

Datenschutzbeauftragte hatten vermutlich so kurz vor Ablauf der Übergangsregelung der DSGVO Hochkonjunktur. Allerdings sehen weder die DSGVO noch das deutsche Datenschutzgesetz (BDSG) zwangsläufig einen Datenschutzbeauftragten in jedem Unternehmen vor. Es sei denn, Art und Umfang der Datenverarbeitung machen es notwendig; in der Regel bei mehr als neun Personen, die persönliche Daten verarbeiten. Bei PROMATIS war dies der Fall, deshalb war für uns schnell klar, dass wir eine Person mit diesen speziellen Fähigkeiten benötigten. Die fachlichen

Anforderungen an die Position bezogen sich nicht nur auf die tiefen Kenntnisse in der Gesetzeslandschaft, sondern auch auf ein grundlegendes Verständnis der Prozess-Strukturen innerhalb des Unternehmens sowie der Realisierungen mittels geeigneter Software-Tools. Wir hatten Glück und konnten solch einen Experten für uns gewinnen, der uns aktiv und mit viel Know-how und Engagement sowohl bei der internen Umsetzung als auch in Kundenprojekten unterstützt.

Welche Prozess-Anpassungen waren bei Ihnen im Unternehmen notwendig?

Da gab es eine Vielzahl, die eigentlich das gesamte Unternehmen betraf. Die erste Gruppe umfasst den Umgang mit personenbezogenen Mitarbeiterdaten. Die betroffenen Abteilungen, allen voran HR, aber auch Buchhaltung und System-Administration, überprüften, wie die Prozesse effizient und konform zu gestalten sind. Wir hatten den Vorteil, ein Prozess-Modellierungs-Tool unseres strategischen Partners Horus einsetzen zu können, das die neue und verschlankte Prozess-Landschaft visualisiert darstellt, wodurch die Umsetzung für alle Beteiligten recht einfach war. Die andere Gruppe betraf die Verarbeitung der Kundendaten. In diesem Bereich, der insbesondere Marketing und Vertrieb zufiel, hatten wir größere Anpassungen vorzunehmen, denn der Umgang mit diesen sensiblen Daten wird in der DSGVO sehr streng reglementiert. Allerdings stellte dies für uns auch eine Chance dar: Durch die Vorgaben waren wir gezwungen, uns aktiv mit der Quantität und Qualität unserer Kunden- und Interessenten-Datenbanken auseinanderzusetzen, Bereinigungen vorzunehmen und Prozesse wie beispielsweise die Informationsvermittlung innerhalb des Unternehmens oder mit Partnern zu verschlanken. Eine weitere Chance entwickelte sich aus der Anforderung der Transparenz. Wir waren gezwungen, zentralisiert und in Systemen, die den DSGVO-Anforderungen entsprachen, die Prozesse abzubilden und berechtigungsgesteuert zu nutzen.

Wie hat das Zusammenspiel zwischen den einzelnen Abteilungen funktioniert?

Am Anfang, als das Thema der DSGVO im Raum stand, war die Zusammenarbeit ein wenig holprig, denn für jede Abteilung bedeutete dieser Prozess ein Mehraufwand, was insbesondere in unserem Projektgeschäft eine Herausforderung darstellte. Dank unseres Datenschutzbeauftragten, der mit viel Geduld und abteilungsbezogenen Schulungen die Thematik verdeutlichte, arbeiteten die Abteilungen eng zusammen. Dies zeigte sich unter anderem an der Schnittstelle zwischen Marketing und Vertrieb. Gemeinsam wurden Verantwortlichkeiten geklärt, Schnittstellen definiert, Scoring-Tabellen zur Lead-Qualifizierung entwickelt und eine intensivere Nutzung der Marketing- und Vertriebs-Tools vorangetrieben – also weit mehr, als die DSGVO fordert. Wir sahen dies als Möglichkeit, uns intensiv mit strategischen Fragen zu beschäftigen sowie die Basis für eine innovative Kundenansprache zu schaffen.

Inwiefern gilt europäisches Datenschutzrecht auch für außereuropäische Unternehmen?

Die neue DSGVO richtet sich nicht nur an alle europäischen Unternehmen und Behörden, sondern an jede Organisation, die in Europa oder mit europäischen Kunden Geschäfte macht. PROMATIS besitzt Ländergesellschaften in Österreich, Schweiz und USA, die sehr eng mit dem Headquarter in Ettlingen zusammenarbeiten. Zentrale Abteilungen wie beispielsweise HR und Marketing betreuen die gesamte Gruppe und hier greift die Vorgabe, dass bei einer Auftragsverarbeitung in einem europäischen Land die Richtlinien der DSGVO gelten.

Waren Ihre Marketing-Prozesse von der Umsetzung der DSGVO tangiert?

Marketing war sehr stark in diese gesamte DSGVO-Thematik involviert, denn der Umgang mit Kundendaten, der ja das Kernbusiness von Marketing und Vertrieb darstellt, ist streng reglementiert. Diese grundlegenden Änderungen betrafen sowohl die internen Prozesse der Marketing-Abteilung als auch die Zusammenarbeit mit anderen Abteilungen und Partnern. Doch die größte Herausforderung stellte eine systematische, transparente und valide Struktur der Vorgehensweisen im Umgang mit den Kunden- und Interessentendaten dar. Durch den Einsatz von unterstützenden Software-Tools wie „Oracle Eloqua“ als Marketing-Automation und dem CRM-System „Oracle Sales Cloud“ konnten diese Herausforderungen in wirklich innovativen Prozessen etabliert werden.

Was raten Sie anderen Unternehmen, die sich mit der DSGVO bis jetzt noch nicht beschäftigt haben?

Zunächst sollte ein breit aufgestelltes Projekt-Team zusammengestellt werden. Beim Datenschutz reicht es nicht, wenn die IT im stillen Kämmerchen konzipiert, da muss die Geschäftsführung ebenso mit am Tisch sitzen wie die Fachabteilungen, die Personal- und Rechtsabteilung und natürlich auch der Datenschutzbeauftragte. Wer ganz sicher gehen will, kann auch externe Partner hinzuziehen wie beispielsweise Fachjuristen. So ist gewährleistet, dass alle Perspektiven Berücksichtigung finden. Wenn das Team steht, ist es gar nicht mehr so schwierig, einen weiteren Projektfahrplan zu definieren. Mit dieser strukturierten Vorgehensweise stießen wir durchgängig auf positive Resonanz bei unseren Kunden, die wir während dieser außergewöhnlichen Phase aktiv und mit unserem Know-how hinsichtlich Gesetzesvorgaben und Prozessen unterstützten.

Sie haben die Anforderungen umgesetzt, wie sieht Ihr Fazit heute aus?

Wenn Sie mir diese Frage Anfang des Jahres gestellt hätten, wäre meine Antwort eher negativ ausgefallen. Doch heute sehe ich es aus einer anderen Perspektive. Durch die DSGVO waren wir gezwungen, uns mit unseren Prozessen, den Datenbergen, den vielen Kleinigkeiten, die sich im Laufe eines Geschäftslebens festsetzten, auseinanderzusetzen – mit dem Resultat, dass wir uns von Altlasten trennten, neue, schlankere Prozesse aufsetzten und mehr Transparenz in unsere Unternehmenslandschaft brachten. Dies ist nicht nur für unsere weitere Vorgehensweise ein Vorteil, auch unsere Kunden und Interessenten profitieren davon. Denn heute sind wir in der Lage, genau diese Themen zielorientiert zu platzieren, die wirklich von Interesse sind.

Sabine Rudolf
sabine.rudolf@promatis.de

Security im Oracle-Rechenzentrum in Frankfurt

Michael Fischer, ORACLE Deutschland B.V. & Co. KG

Oracle bietet seit mehreren Jahren Cloud Services in den Bereichen „IaaS", „PaaS" und „SaaS" an. Seit dem Jahr 2017 wird ein Teil der Services auch in Deutschland angeboten. Der Artikel beschreibt Sicherheitsmerkmale dieser Oracle Cloud Services im Frankfurter Rechenzentrum.

Der Anteil von Unternehmen, die Cloud-Angebote nutzen, steigt seit Jahren kontinuierlich. Sie benötigen dafür skalierbare, hybride Cloud-Lösungen, die die Sicherheits-, Datenschutz- und Compliance-Anforderungen erfüllen. Um diesen Anforderungen gerecht zu werden, hat Oracle unter anderem die Oracle Cloud Infrastructure entwickelt, eine Cloud-Plattform, die den Kunden ein virtuelles Rechenzentrum in der Cloud bietet. Die Oracle Cloud Infrastructure stellt eine Vielzahl von Cloud-Services bereit *(siehe Abbildung 1)*.

Unter „Compute" finden sich dedizierte Server (auch „Bare Metal") sowie virtuelle Maschinen (VM). Der Speicher steht mit „Storage" (in Form von Block, File, Object, Archive-Speicher) zur Verfügung. Oracle-Datenbanken gibt es in den verschiedenen Ausprägungen: Shared, dediziert, Exadata und Autonomous. Zugrunde liegen Software-definierte, virtuelle Cloud-Netzwerke (VCN), optional ergänzt um „Edge"-Services wie Load Balancing, DNS und weitere Services.

Für die Verwaltung werden in den Services Identitäts- und Zugriffsmanagement (engl. Identitiy and Access Management), Verschlüsselung, Monitoring und Auditierung bereitgestellt sowie ein Cloud-Tooling zur Unterstützung bei den typischen Tätigkeiten wie Erzeugen, Starten, Stoppen, Löschen von Services ebenso wie Backups, Restores oder Patching.

Für Unternehmenskunden, die eine Public Cloud nutzen wollen, sind die Datensicherheit und der Aufwand für die Migration bestehender Anwendungen von zentraler Bedeutung. Angesichts der Einschränkungen herkömmlicher öffentlicher Clouds migrieren Unternehmen normaler-

weise nicht-kritische Anwendungen in die Cloud und beschränken geschäftskritische Produktionsanwendungen und Daten weiterhin auf ihre lokalen Rechenzentren.

Oracle hat seine Cloud-Infrastruktur so aufgebaut, dass Unternehmen auch unternehmenskritische Anwendungen und Daten unter Berücksichtigung der Sicherheit migrieren und den Overhead beim Aufbau und Betrieb der Rechenzentrums-Infrastruktur reduzieren können. Mit der Oracle Cloud Infrastructure erhalten Unternehmenskunden die gleiche Kontrolle und Transparenz über ihre Workloads wie in ihren eigenen Rechenzentren.

Für Kunden, die eine vollständig isolierte und kontrollierte Umgebung benötigen, bietet Oracle Cloud Infrastructure sogenannte „Bare-Metal-Instanzen", also Maschinen, die vollständig vom Kunden verwaltet werden, ohne dass eine Software von Oracle auf der Instanz läuft. Kunden haben in diesem Fall sogar vollständigen Root-Zugang zu diesen Maschinen.

Seit dem Jahr 2017 ist die Oracle Cloud Infrastructure auch in den Rechenzentren in Frankfurt bereitgestellt. Dort werden drei örtlich getrennte Rechenzentren (in 5 bis 15 Kilometern Entfernung) genutzt, um eine entsprechende Ausfallsicherheit und Disaster-Recovery-Funktionen bereitzustellen.

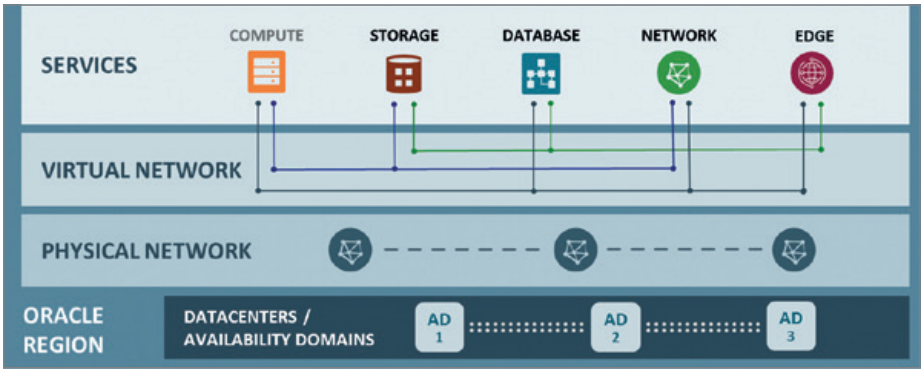


Abbildung 1: Funktionalitäten in der Infrastructure Cloud von Oracle

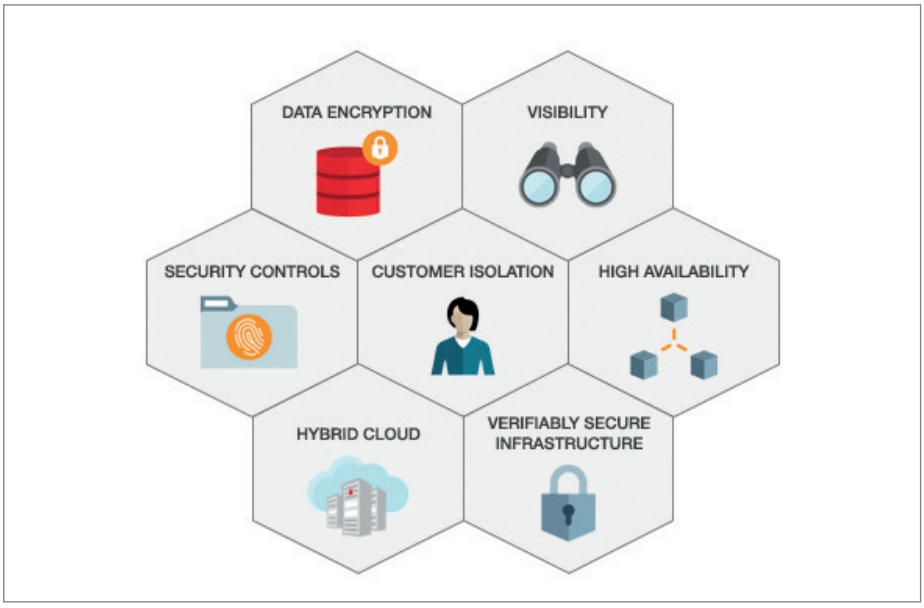


Abbildung 2: Die sieben Sicherheitsprinzipien

Die Sicherheitsprinzipien der Oracle Cloud Infrastructure

Oracle-Cloud-Infrastructure-Sicherheit umfasst folgende Prinzipien *(siehe Abbildung 2):*

- *Kundenisolierung*
  Anwendungs- und Datenbestände werden in einer Umgebung bereitgestellt, die von anderen Kunden und dem Oracle-Betrieb isoliert ist.

- *Verschlüsselung*
  Schutz der Daten am Speicherort und während der Übertragung durch Verschlüsselung. Transparenz bezüglich kryptografischer Algorithmen und Schlüsselverwaltung.

- *Security Controls*
  Funktionen zur Konfiguration von Sicherheit, die es ermöglichen, den Zugang zu Diensten einzuschränken (engl. „least privilege") und operative Verantwortlichkeiten zu trennen (engl. „segregation of duties").

- *Sichtbarkeit*
  Über Logdaten und Sicherheitsanalysen lassen sich die eigenen Ressourcen überprüfen und überwachen. Damit können Audit-Anforderungen erfüllt sowie Sicherheits- und Betriebsrisiken reduziert werden.

- *Sichere hybride Umgebungen*
  Vorhandene Sicherheitsmechanismen wie Benutzerkonten und Richtlinien sowie Sicherheitslösungen von Drittanbietern beim Zugriff auf Cloud-Ressourcen können weiterhin genutzt werden, um Daten in der Cloud abzusichern.

- *Hohe Verfügbarkeit*
  Bereits in der Standard-Konfiguration bieten die dreifach redundanten Rechenzentren hochverfügbare Scale-out-Architekturen und Resistenz gegen Netzwerk-Angriffe.

- *Überprüfbarkeit*
  Nachweis der Einhaltung der strengen Sicherheits-Standards von Oracle durch Audits, Zertifizierungen und Zertifikate von Drittanbietern. Eigene Cloud-Audits können beantragt werden.

- *Off-Box-Virtualisierung*
  Das virtuelle Netzwerk wird ausschließlich Software-basiert innerhalb der Cloud Infrastructure realisiert und nicht auf den auf den Maschinen laufenden Hypervisoren. Neben höherer Performance beim Netzwerk-Durchsatz bedeutet das auch den Wegfall von Sicherheitslücken durch potenzielle Exploits des Hypervisors auf der Netzwerk-Ebene.

Durch den Einsatz der Oracle Cloud Infrastructure profitieren Kunden direkt von der umfassenden Expertise von Oracle und den kontinuierlichen Investitionen in die Sicherheit. Die Entwicklung von Cloud Services erfolgt unter ISO-27001-Standards unter Berücksichtigung der ISO-27002-Controls für Information Security Management.

Die Rechenzentren, die die Cloud Services betreiben, sind zertifiziert nach ISO 9001, ISO 14001, OHSAS 18001, ISO 27001, ISO 50001 und PCI-DSS. Vom Typ sind sie ANSI/TIA-942-A Tier-III- oder Tier-IV-Standards des „Uptime Institute" und der Telecommunications Industry Association (TIA). Die Oracle Cloud Services in Frankfurt besitzen SSAE16/ISAE3402 (SOC-1 Typ 2), AT101 (SOC-2 Typ 2), SOC-3, PCI-DSS und HIPAA-Attestations sowie ISO/IEC-27001:2013-Zertifizierungen. Weitere (auch C5) sind in Planung.

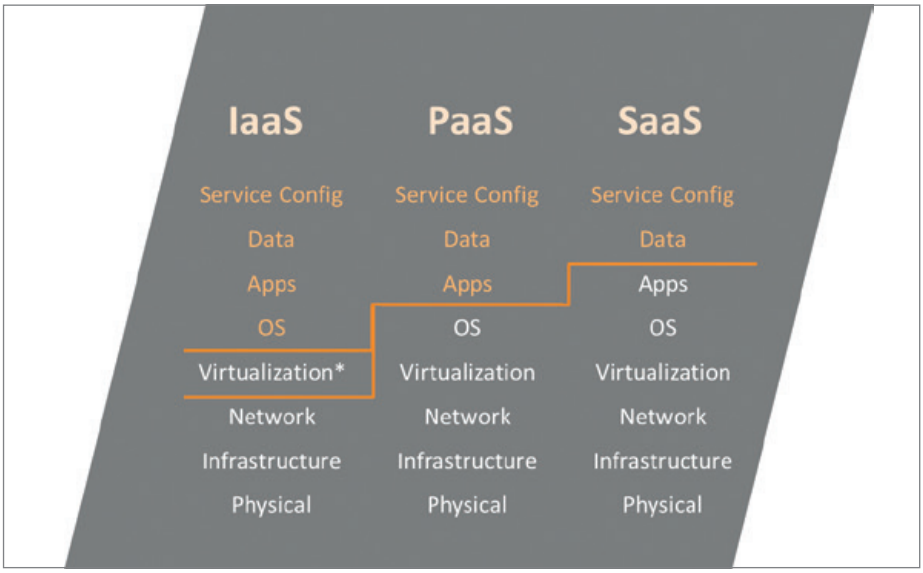


Abbildung 3: Die Aufteilung der Verantwortlichkeiten: Der weiß eingefärbte Teil liegt beim Cloud-Anbieter

Gemeinsame Verantwortung

Oracle stellt standardmäßig Cloud-Sicherheitstechnologien und betriebliche Prozesse zur Absicherung der Cloud Services bereit. Zusätzlich muss der Kunde darüber hinaus bei der Nutzung der Oracle Cloud auch selbst Verantwortung für Sicherheit und Compliance übernehmen, etwa in der sicheren Konfiguration der Cloud Services. Sicherheit in der Cloud ist somit immer eine gemeinsame Verantwortung, aufgeteilt zwischen dem Kunden und Oracle.

In der Cloud-Umgebung ist Oracle für die Sicherheit der zugrunde liegenden Cloud-Komponenten (wie Rechenzentrums-Einrichtungen, Hard- und Software-Systeme) verantwortlich, die Kunden hingegen tragen die Verantwortung für die Absicherung ihrer Workloads und die Konfiguration ihrer Services (wie Compute, Network, Storage und Database). Auf einem exklusiv genutzten Bare-Metal-Server erweitert sich die Verantwortung der Kunden auf den gesamten Software-Stack. In der Abbildung ist der Sonderfall mit Bare-Metal, dem Wegfall der Server-Virtualisierung durch den Cloud Provider, mit „*)" markiert *(siehe Abbildung 3)*.

Im Einzelnen lassen sich die Verantwortlichkeiten von Kunden und Oracle in die folgenden Bereiche unterteilen: Der Zugriff auf Oracle Cloud Services ist mit einem Identitäts- und Zugriffsmanagement (IAM) geschützt. Die Kunden sind für die Verwaltung und Überprüfung ihrer Accounts und für alle Aktivitäten im Rahmen der Nutzung verantwortlich. Oracle stellt lediglich das IAM zum Identitätsmanagement, zur Authentifizierung (inklusive Single-Sign-on über Federation), Autorisierung und Auditierung. IAM erlaubt über einen fein granularen Mechanismus, Zugriffsrechte auf Ressourcen-Anwendergruppen zu geben. Hierbei verfährt Oracle nach dem Least-Privilege-Grundsatz: Ein Nutzer hat per Default keine Rechte. Alle Rechte müssen explizit gesetzt werden.

Absicherung des Workloads: Kunden sind für den Schutz und die Sicherung des Betriebssystems und der Anwendungsschichten verantwortlich. Dieser Schutz umfasst, abhängig von den Services, das Patchen von Anwendungen und Betriebssystemen, die Konfiguration des Betriebssystems sowie den Schutz vor Malware und Netzwerkangriffen. Oracle bietet sichere Images mit aktuellen Patch-Levels an und ermöglicht im Rahmen der Kompatibilität mit den Services, Sicherheitslösungen von Drittanbietern, beispielsweise SIEM oder Firewall-Software-Appliances, aufzuspielen.

Datenklassifizierung und Compliance: Der Kunde ist für die korrekte Klassifizierung und Kennzeichnung seiner Daten und die Einhaltung der Compliance-Anforderungen verantwortlich. Auch die Überprüfung der Gesamtlösung hinsichtlich der Compliance ist in der Verantwortung des Kunden.

Sicherheit der Host- und Storage-Komponenten: Kunden sind für die sichere Konfiguration und Verwaltung ihrer Bare-Metal-, Compute- (virtuelle Hosts, Container), Storage- (Objekt-, lokale Storage-, Block-Volumes, File und Archive) und Plattform-Services (wie Datenbank-Konfiguration) verantwortlich. Oracle stellt die Basissicherheit dieser Elemente zur Verfügung.

Netzwerksicherheit: Kunden sind für die sichere Konfiguration von Netzwerk-Elementen wie virtuellen Netzwerken, Load Balancing, DNS und Gateways (Internet, VPN, Peering) zuständig. Oracle ist für die Bereitstellung einer sicheren Netzwerk-Infrastruktur verantwortlich. Oracle Cloud Infrastructure bietet dabei Kunden mit Security-Lists eine sehr feingranulare Konfigurationsmöglichkeit (welcher Traffic kann von wo nach wo über welchen Port erfolgen) ihrer Sub-Netze an.

Client- und Endpoint-Protection: Kunden nutzen verschiedene Hard- und Software-Systeme, wie mobile Geräte und Browser, um auf ihre Cloud-Ressourcen zuzugreifen. Kunden sind für die Absicherung aller Clients und Endgeräte verantwortlich, die sie für den Zugriff auf Oracle Cloud Infrastructure Services verwenden.

Physische Sicherheit: Oracle ist verantwortlich für den Schutz der globalen Infrastruktur, die alle in der Oracle Cloud Infrastructure angebotenen Services betreibt. Diese Infrastruktur besteht aus

Hardware, Software, Netzwerken und Einrichtungen, die Oracle Cloud Infrastructure Services betreiben. Oracle betreibt seine Cloud-Dienste innerhalb von Rechenzentren namhafter externer Betreiber, die mindestens ANSI/TIA-942-A Tier-3- oder Tier-4-Standards und einer N2-Redundanz-Methodologie folgen. Die Betreiber haben weder administrativen noch physischen Zugang zu den in Cages betriebenen Rechnern, es wird nur Housing, Energie, Kühlung und Brandschutz zur Verfügung gestellt.

Die Konzepte, Regionen, Redundanzen und Verfügbarkeit

Oracle Cloud Infrastructure bietet dem Kunden die Möglichkeit, sich sein Cloud-Datacenter aus den verschiedenen Cloud-Service-Lokationen auszusuchen, so zum Beispiel Frankfurt. Jede Region besteht aus drei Standorten mit eigenen Datacentern, den sogenannten „Availability Domains". Availability Domains innerhalb der gleichen Region sind durch ein sicheres, schnelles und latenzarmes Netzwerk verbunden, die Verbindungen sind verschlüsselt und je nach Servicetyp erfolgt transparent eine redundante Speicherung an mehreren Standorten.

Die Availability Domains sind physisch getrennte, von unterschiedlichen Anbietern gestellte Rechenzentren (jeweils 5 – 15 km voneinander getrennt). Es handelt sich also nicht nur um Availability Zones, die lediglich innerhalb eines Rechenzentrums durch Brandschutztüren voneinander getrennt sind. In Deutschland existiert als Region Frankfurt. Eine weitere EU-Region ist noch London, ein Ausbau weiterer EU-Regionen ist geplant.

Die Availability Domains können auch explizit vom Kunden genutzt werden, um hochverfügbare Anwendungen aufzubauen. Entsprechende technische Hilfsmittel wie Load Balancer oder Data Guard stehen zur Verfügung *(siehe Abbildung 4)*.

Zugriffsverwaltung und Auditing

Wird ein Cloud Account bei Oracle angelegt, bekommt der Kunde die notwendigen administrativen Accounts. Damit können Berechtigungsstrukturen definiert und weitere Accounts angelegt und verwaltet werden. Benutzer können entweder im Cloud Service über die Oberfläche angelegt, von außen provisioniert oder im Rahmen des Single-Sign-on (SSO) übernommen werden. Diese SSO-„on-the-fly"-Übernahme erfolgt optional basierend auf dem Federation Standard via SAML, etwa von einem Active Directory Federation Service aus. Passwort-Policies können bei Benutzern mit direkter Anmeldemöglichkeit definiert werden. Starke Authentifizierung und kontextbasierte Authentifizierung lassen sich im jeweiligen führenden Anmeldesystem konfigurieren.

Das Zugriffsmanagement ermöglicht so die Umsetzung des „least privilege"-Prinzips und optional die direkte Steuerung der Rechte vom lokalen oder fremden Identity Management System, etwa die Deaktivierung beim Verlassen des Unternehmens. Über die Steuerungsmöglichkeit können auch Abteilungen und/oder Test- und Produktions-Systeme abgebildet werden.

Alle Ressourcen sind standardmäßig geschützt und können nur mit entsprechenden Berechtigungen angelegt, geändert oder gelöscht werden. Dazu werden Gruppen definiert, Policies erstellt und die Gruppen den Benutzern oder Services zugewiesen. Die Architektur lässt den angemeldeten Account oder Service nur diejenigen Operationen ausführen, für die er berechtigt wurde. Das Berechtigungsmanagement wirkt übergreifend über Regionen.

Neben der Verwaltung der Cloud über die Web-Konsole bietet Oracle Cloud Infrastructure auch die Möglichkeit, über ein REST-API oder durch bereitgestellte SDKs (wie eine CLI für Shell-Skripte, Python, Java) Ressourcen zu verwalten. Zudem werden auch deklarative Infrastructure-as-Code-Technologien (IaC) wie Terraform unterstützt.

Alle Zugriffe der Web-Konsole über IaC oder API/SDKs laufen am Ende über das gleiche API und werden daher identisch und vollständig durch den Oracle Cloud Infrastructure Audit Service protokolliert. Der Audit-Service stellt die Auditdaten über eine grafische Konsole und als JSON-Daten über ein authentifiziertes, filterbares Abfrage-API zur Integration in einem übergeordneten System bereit. Der Inhalt des Audit-Protokolls umfasst die

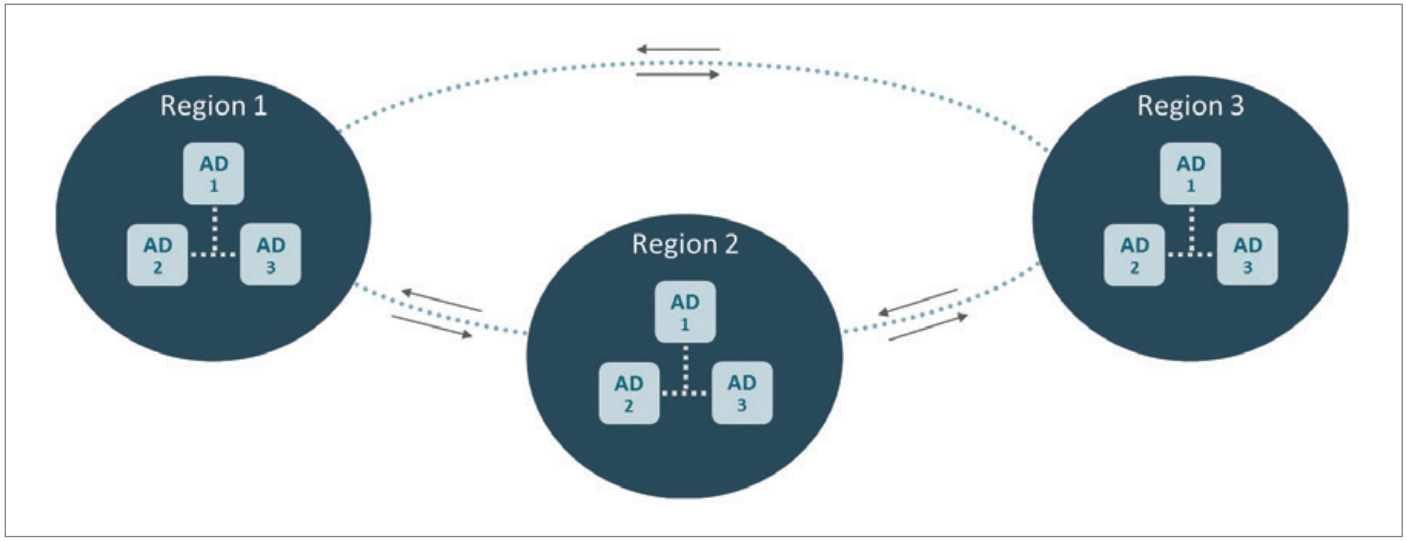


Abbildung 4: Das Konzept mit Regionen und Availability Domains

aufgetretene Aktivität, den Benutzer, der sie initiiert hat, das Datum und die Uhrzeit des Requests sowie die Quell-IP, den User-Agent und die HTTP-Header des Requests. Auditdaten von Services wie DB Auditlog oder Syslog sind wie bei On-Premises-Systemen über die entsprechenden Service-Schnittstellen verfügbar.

Instanzen

Die Oracle Cloud Infrastructure bietet Bare-Metal-, Virtual-Machine- und Datenbank-Instanzen in verschiedenen Ausprägungen:

- Bare-Metal-Instanzen (BM) sind dedizierte physische Server für einen einzelnen Kunden, der die volle Kontrolle über den Server hat. Es gibt keinen von Oracle verwalteten Hypervisor; Oracle-Mitarbeiter haben keinen Zugriff auf Speicher oder Storage. Die gesamte Netzwerk-Virtualisierung erfolgt off-box und nur der Oracle Integrated Lights Out Manager (ILOM) wird zum Hardware-Monitoring und -Reboot verwendet.
- Virtuelle Maschinen (VMs) sind mandantenfähige Kunden-VMs auf Oracle Hypervisor, der die Isolierung zwischen den Kunden bietet.
- Oracle-Datenbanken sind verfügbar als Exadata, Exclusive DB auf Bare-Metal-Instanzen, VM-DB-Instanzen und Autonomous DWH. DB-Systeme unterliegen der gleichen Zugriffsregelung wie die Instanzen beziehungsweise der Konfiguration der zugrunde liegenden Services (wie ADWC). Standardmäßig werden die Daten am Speicherort mit Oracle-TDE (AES) verschlüsselt, wobei die Hauptschlüssel in einem Oracle-Wallet auf jedem DB-System gespeichert sind. Wallets können ausgelagert oder zentral über Oracle Key Vault verwaltet werden. RMAN-Backups von DB-Systemen werden verschlüsselt und in kundeneigenen Buckets im Object Storage abgelegt.

Oracle-Cloud–BM-, -VM- und -DB-Instanzen verwenden als Standard schlüsselbasiertes SSH für die Verwaltung. Die Schlüssel werden vom Kunden festgelegt. Im Falle der Datenbank vergibt der Kunde zusätzlich die DB-Credentials. Andere Zugänge sind nicht vorkonfiguriert. Es können eigene oder von Oracle bereitgestellte gepatchte Images verwendet werden. Alle von Oracle bereitgestellten Images verfügen über sichere Standard-Einstellungen einschließlich Firewalls auf Betriebssystem-Ebene, die standardmäßig aktiviert sind.

Netzwerk und Storage

Im Oracle-Cloud-Infrastructure-Netzwerk gibt es keine Noisy-Neighbor-Probleme auf dem Netz und an den Instanzen. SLAs werden sowohl für Netzwerk-Bandbreiten als auch für Latenzzeiten zugesagt. Es erfolgt eine Isolierung der Kunden-Netzwerke auf Layer 3. Dies wird durch Software-Defined-Networks (SDNs) ermöglicht, die physisch auf einer speziellen Architektur (sogenannte „Clos-Netzwerke“) basieren und Off-Box-Netzwerk-Virtualisierung ohne zentralen Netzwerk-Hypervisor nutzen. Es erfolgt keine Over-Subscription.

Der Kunde definiert sein virtuelles Cloud-Netzwerk (VCN) unter Verwendung aller bekannten Sicherheits-Funktionalitäten wie IPs (public, private), Sub-Netzen (public, private), Routing-Tabellen und Security Lists (Firewalls). Durch Kunden konfigurierte Gateways kontrollieren die Netz-Übergänge mit dem Internet, einem VPN (IPSec), zwischen den Sub-Netzen und zwischen Regionen. Es kann eine private Verbindung zwischen dem Rechenzentrum des Kunden und der Oracle Cloud aufgeschaltet werden. Damit ist das Netzwerk vollständig unter der Kontrolle des Kunden.

Storage steht in verschiedenen Ausprägungen zur Verfügung. Sowohl physisch in den einzelnen Maschinen für High-Performance-Datenzugriffe (IOPS) als auch als externer Speicher. Externer Speicher und Boot Volumes sind immer AES-verschlüsselt. Externer Speicher ist für den Kunden transparent redundant. Weitere Redundanz kann konfiguriert werden. Die Storage-Typen sind im Einzelnen:

- *Lokaler Storage*
  NVMe-gestützter Speicher in den Dense-IO-Instanzen für maximale IOPS

- *Boot und Block Volumes*
  Über das Netzwerk angeschlossener Speicher (iSCSI) mit redundanter Speicherung in der jeweiligen Availability Domain

- *Object-Storage*
  Speicher aus Buckets und Objects, redundant über Availability Domains. Eine Authentifizierung ist beim Zugriff notwendig, es sei denn, es wird ein öffentlicher Zugriff festgelegt. Zugriffsschlüssel für andere Protokolle (wie Amazon S3) können über das integrierte Berechtigungsmanagement ebenso verwaltet werden wie vorauthentifizierte Benutzerzugriffe.

- *File-Storage*
  NFSv3-Endpunkt als Mount-Ziel im VCN-Subnetz jedes Kunden. Das Mount-Ziel wird durch einen DNS-Namen identifiziert und auf eine IP-Adresse abgebildet.

- *Archive-Storage*
  Speicherung langlebiger Daten mit nur seltenem Zugriff. Der Zugriff erfolgt wie bei Object-Storage (API, SDK, CLI), aber im Gegensatz zum Object-Storage stehen Daten nach einem Daten-Abruf erst mit einer Verzögerung von mehreren Stunden bereit.

Weitere Services

Weitere Services wie Load Balancer und Managed Domain Name Server (DNS) stehen zur Verfügung. Load Balancer können kundeneigene Zertifikate verwenden und unterstützen End-to-End-SSL, SSL-Tunneling oder SSL-Terminierung. Im Standard werden TLS 1.2 und Forward-Secrecy-Chiffre verwendet. Der Oracle-Cloud-Infrastructure-DNS-Service bietet dynamische, statische und rekursive DNS-Lösungen. Der DNS-Dienst arbeitet in einem globalen Anycast-Netzwerk mit 18 Points of Presence (PoP) auf fünf Kontinenten und bietet vollständig redundante DNS-Konstellationen sowie mehrere Tier-1-Transit-Anbieter pro PoP.

Sicherheits-Design und -Kontrollen

Das Oracle-Sicherheitsmodell basiert auf Menschen, Prozessen, Werkzeugen und

einer gemeinsamen Sicherheits-Plattform mit Methoden und Ansätzen *(siehe Abbildung 5)*. Einige Aspekte daraus sind:

- *Benutzer-Authentifizierung und Zugriffskontrolle*
  Der Zugang zu den Produktionssystemen unterliegt dem „least privilege"-Prinzip. Die Berechtigungen werden regelmäßig überprüft und sind auch mit dem HR-System gekoppelt. Der Zugriff auf Produktions-Umgebungen erfordert eine Multi-Faktor-Authentifizierung (MFA). Zugriffe auf Produktions-Systeme werden protokolliert und die Protokolle zur Sicherheits-Analyse gespeichert.

- *Change Management*
  Die Oracle Cloud Infrastructure folgt definierten Change-Management- und Deployment-Prozessen. Alle Änderungen, die in der Produktions-Umgebung vorgenommen werden, folgen einem Test- und Freigabe-Prozess. Die Integrität kritischer System-Konfigurationen wird überwacht, um sicherzustellen, dass sie mit dem erwarteten Zustand übereinstimmen.

- *Schwachstellen-Management*
  Sowohl interne Penetrationstests als auch Tests durch externe Branchen-Experten werden durchgeführt, um potenzielle Schwachstellen zu identifizieren. Oracle Cloud Infrastructure Hosts durchlaufen periodische Scans mit branchenüblichen Scannern.

- *Sicherheits-Protokollierung und -Überwachung*
  Sicherheitsrelevante Ereignisse (wie API-Aufrufe und Netzwerk-Ereignisse) werden protokolliert und die Protokolle auf anomales Verhalten überwacht. Alarme, die ein Überwachungsmechanismus auslöst, werden vom Sicherheitsteam verfolgt und ausgewertet.

- *Netzwerk-Sicherheit*
  Standardmäßig erfolgt die Kundenkommunikation mit Oracle Cloud Infrastructure Services unter Verwendung aktueller TLS-Chiffren und Konfigurationen, um Kundendaten während der Übertragung zu schützen und Man-in-the-Middle-Angriffe zu verhindern. Aufrufe von Diensten sind mit öffentlichen Schlüsseln digital signiert, um Manipulationen zu verhindern. Es werden Tools und Mechanismen eingesetzt, um verteilten Denial-of-Service-Angriffen (DDoS) zu begegnen und eine hohe Verfügbarkeit zu erreichen.

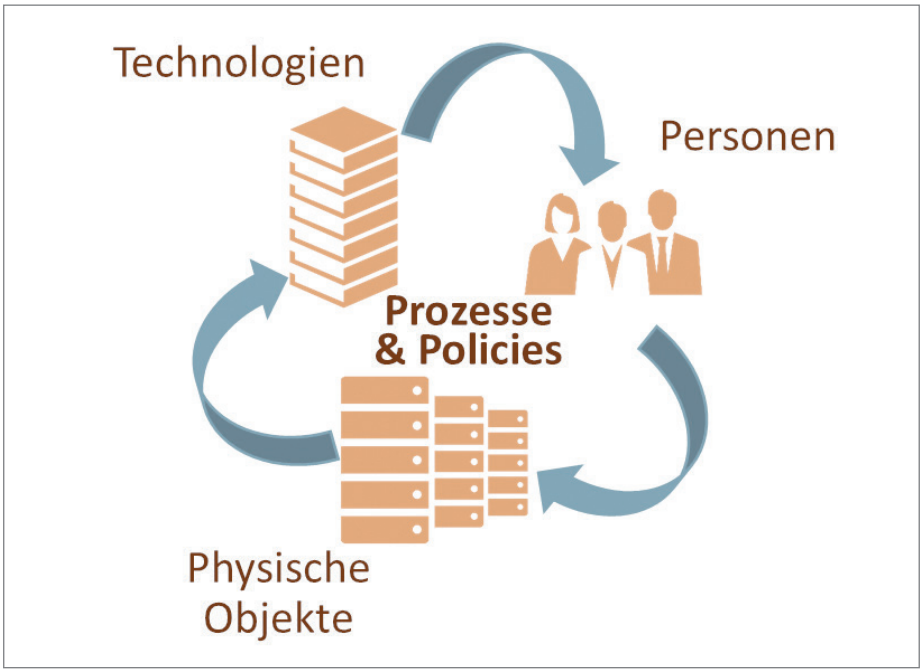


Abbildung 5: Sicherheitsmodell Oracle Cloud

- *Reaktion auf Vorfälle/Incidents*
  Innerhalb der Oracle Cloud wird ein Monitoring eingesetzt, um Incidents (dt. Vorfälle) zu erkennen. Abhängig von der Art des Vorfalls sind Eskalationspfade und Reaktionsteams definiert, um den Vorfall zu beheben. Oracle arbeitet mit dem Kunden, mit den entsprechenden technischen Teams und gegebenenfalls mit externen Strafverfolgungsbehörden zusammen, um auf den Vorfall zu reagieren. Das Ziel ist es, die Vertraulichkeit, Integrität und Verfügbarkeit der Cloud-Services-Umgebung wiederherzustellen und Ursachen und Abhilfemaßnahmen zu ermitteln. Mit dem sogenannten „CAPA-Prozess" (Corrective Action/Preventative Action) erfolgt die Ursachen-Analyse, um bei Bedarf Änderungen für die Produktion, technische Maßnahmen oder Prozess-Änderungen abzuleiten und sicherzustellen, dass das Problem nicht erneut auftritt. Verantwortlich dafür ist ein 24/7-Incident-Response-Team, die Kommunikation zum Kunden übernimmt die Oracle Global Information Security (GIS). Der Auftragnehmer wird den Auftraggeber im Falle eines Sicherheitsverstoßes, in dem der Auftraggeber gemäß geltendem Recht eine Benachrichtigung erhalten muss, innerhalb von 24 Stunden oder früher informieren. Wenn Informationen gesammelt oder anderweitig verfügbar sind, wird Oracle dem Auftraggeber eine Beschreibung des Sicherheitsverstoßes zur Verfügung stellen, sofern dies nicht gesetzlich untersagt ist.

- *Trennung der Cloud-Entwicklung von der Produktion*
  Die Vorproduktions-Umgebungen (wie Entwicklung, Test und Integration) sind von den Produktions-Umgebungen getrennt, sodass die Entwicklungs- und Test-Aktivitäten keine Auswirkungen auf die Produktivsysteme haben.

- *Löschung und Medienvernichtung*
  Die Oracle-Cloud-Infrastructure-Instanzen werden nach der Freigabe der Hardware durch den Kunden sicher gelöscht und neu initialisiert. Wenn die zugrunde liegende Hardware das Ende ihrer Lebensdauer erreicht hat, wird sie sicher zerstört. Vor dem Verlassen der Oracle-Rechenzentren werden die entsprechenden Laufwerke durch den Einsatz branchenführender Medienvernichtungsgeräte unbrauchbar gemacht.

Datenschutz

Der Oracle Cloud liegt die Auftrags-Datenverarbeitung zugrunde. Bei der Nutzung der Oracle Cloud akzeptiert der Kunde diese und weitere Verträge zur Nutzung der Oracle Cloud. Oracle ist Auftrags-Datenverarbeiter („Processor") und der Kunde bleibt der Auftrags-Datenverantwortliche („Controller"). Die Oracle Cloud Infrastructure hat zwei Kategorien von kundenbezogenen Daten:

- *Informationen zum Kundenkonto*
  Dies sind Informationen, die für den Betrieb des Oracle-Cloud-Infrastructure-Kontos des Kunden erforderlich sind und in erster Linie zur Kontaktaufnahme und Abrechnung verwendet werden. Die Verwendung der persönlichen Daten, die Oracle vom Kunden zum Zwecke der Kontoführung sammelt, wird durch die Oracle-Datenschutz-Richtlinie geregelt. Die Oracle Cloud Infrastructure fungiert in diesem Fall als Controller.

- *Durch den Kunden gespeicherte Daten*
  In Daten, die Kunden in der Oracle-Cloud-Infrastruktur speichern, wozu auch personenbezogene Daten gehören können, hat Oracle keinen Einblick. Zusätzlich bestehen keine Einflussmöglichkeiten auf die Entscheidungen des Kunden über deren Erhebung und Verwendung. Oracle hat keine direkte Beziehung zu den betroffenen Benutzern, der Kunde ist der Auftrags-Datenverantwortliche und verwaltet die Daten. Oracle Cloud Infrastructure ist nur der Auftrags-Datenverarbeiter.

Oracle als amerikanischer Anbieter hält sich zum Zeitpunkt der Erstellung des Dokuments an das EU-U.S. Privacy Shield Framework beziehungsweise das U.S.-Swiss Safe Harbor Agreement des US-Handelsministeriums bezüglich der Erfassung, Verwendung und Speicherung von personenbezogenen Daten von Bürgern der Europäischen Union beziehungsweise der Schweiz. Oracle ist auch dafür verantwortlich, dass Dritte, die im Auftrag von Oracle handeln, dies auch tun.

Bezüglich auf beispielsweise in der DSGVO aufgeführte Anforderungen wird in den Cloud Policies (Data Processing Agreement) Transparenz geschaffen. Ergänzende Informationen wie Datenbank-Funktionen, Konfigurationen und Komponenten, die bei der Umsetzung helfen können, finden sich in verschiedenen Oracle Whitepapers. Das Oracle Cloud Infrastructure GDPR Whitepaper beispielsweise konzentriert sich auf Kundenservice-Daten und alle persönlichen Informationen im IaaS Service.

Fazit

Oracle stellt mit der Oracle Cloud Infrastructure in Frankfurt eine Cloud-Umgebung bereit, mit der unternehmenskritische Workloads unter Wahrung der Sicherheits-Anforderungen in die Cloud verlagert werden können. Kunden erhalten Kontrolle und Absicherung unter anderem durch:

- Security per Default: Verschlüsselung am Speicherort und beim Zugriff; Zugriffsbeschränkung im Standard ausgehend von einem Nichts-ist-erlaubt-Ansatz.
- Wahlmöglichkeit bezüglich Datenhosting: weltweit, EU, Deutschland oder On-Premises.
- Isolation der Kunden durch Layer3-Off-Box-virtualisierte Netzwerke und Einsatzmöglichkeit dedizierter Maschinen. Dadurch gibt es kein Noisy-Neighbor-Problem im Netzwerk oder auf dem Server.
- Compliance: Security-Zertifizierungen der Cloud durch Unabhängige, Hilfestellungen für GDPR und andere Regularien. Protokolldaten für Monitoring und Auditierungen sind zugreifbar.
- Cloud Security Services: Zusätzliche Services von Oracle, um Oracle oder 3rd-Party-Clouds/-Umgebungen abzusichern. Einsatzmöglichkeit von Software-Lösungen von Drittanbietern zum Schutz der Daten und Ressourcen in der Cloud.
- Redundante Rechenzentren, die hochverfügbare Scale-out-Architekturen ermöglichen und Netzwerk-Angriffe abwehren.

Oracle investiert weiterhin in die Entwicklung von Cloud Services und Sicherheits-Technologien. Funktionale Erweiterungen auch im Bereich „Security" finden in der Cloud permanent statt, beispielsweise die Erweiterung einer Web-Application-Firewall.

Weitere Informationen

- Oracle IaaS Security Security and Compliance, Informationen und Whitepaper: *https://cloud.oracle.com/iaas_compliance*
- Cloud-Verträge: *http://www.oracle.com/us/corporate/contracts/cloud-services/index.html*
- GDPR Whitepaper: *https://cloud.oracle.com/iaas/whitepapers/oci-gdpr.pdf und oracle.com/goto/gdpr* und *oracle.com/goto/gdpr-newsletter*
- Services Security Documentation: *https://docs.oracle.com/en/cloud*
- Blogeinträge: *https://blogs.oracle.com/cloud-infrastructure*, *https://blogs.oracle.com/coretec* und *https://www.oraclecloud.de*

Michael Fischer
michael.fischer@oracle.com

Neu

Page Designer New Features in Apex 18.1

Tobias Arnhold, Tobias Arnhold – IT Consulting

Mit Apex 18.1 kommen viele große Neuerungen. Dieser Artikel konzentriert sich auf die kleinen Features im Page Designer und darauf, wie Entwickler davon in ihrer täglichen Arbeit profitieren können.

In dem Moment, in dem der Pager Designer in Apex 18.1 das erste Mal geöffnet wird, sieht die Oberfläche leicht anders aus, neben neuen Icons und anderen Farb-Nuancen hat sich allerdings der Aufbau nicht drastisch verändert *(siehe Abbildung 1)*.

Die typische Interaktion im Page Designer funktioniert weiterhin wie in Apex 5.0 beziegunsweise 5.1. Ein Verschieben der Tabs ist wie gewohnt möglich, aber auch weiterhin kann kein Sperren beziegunsweise Festpinnen der Tabs durchgeführt werden. Als Workaround dazu dient weiterhin die Chrome Erweiterung „APEX Page Designer Tab Lock". Damit wird über den Browser die „Drag and Drop"-Funktionalität der Apex-Regionen deaktiviert.

Die Schnellnavigations-Leiste

Einige kleine Änderungen betreffen die Schnellnavigations-Leiste. Das Menü „Settings" ist in das Menü „Utilities" eingegliedert; die Menüs „Team Development" und „Developer Comments" wurden beide in das Menü „Create" übernommen. Die Menü-Aufrufe selbst wurden zum Teil

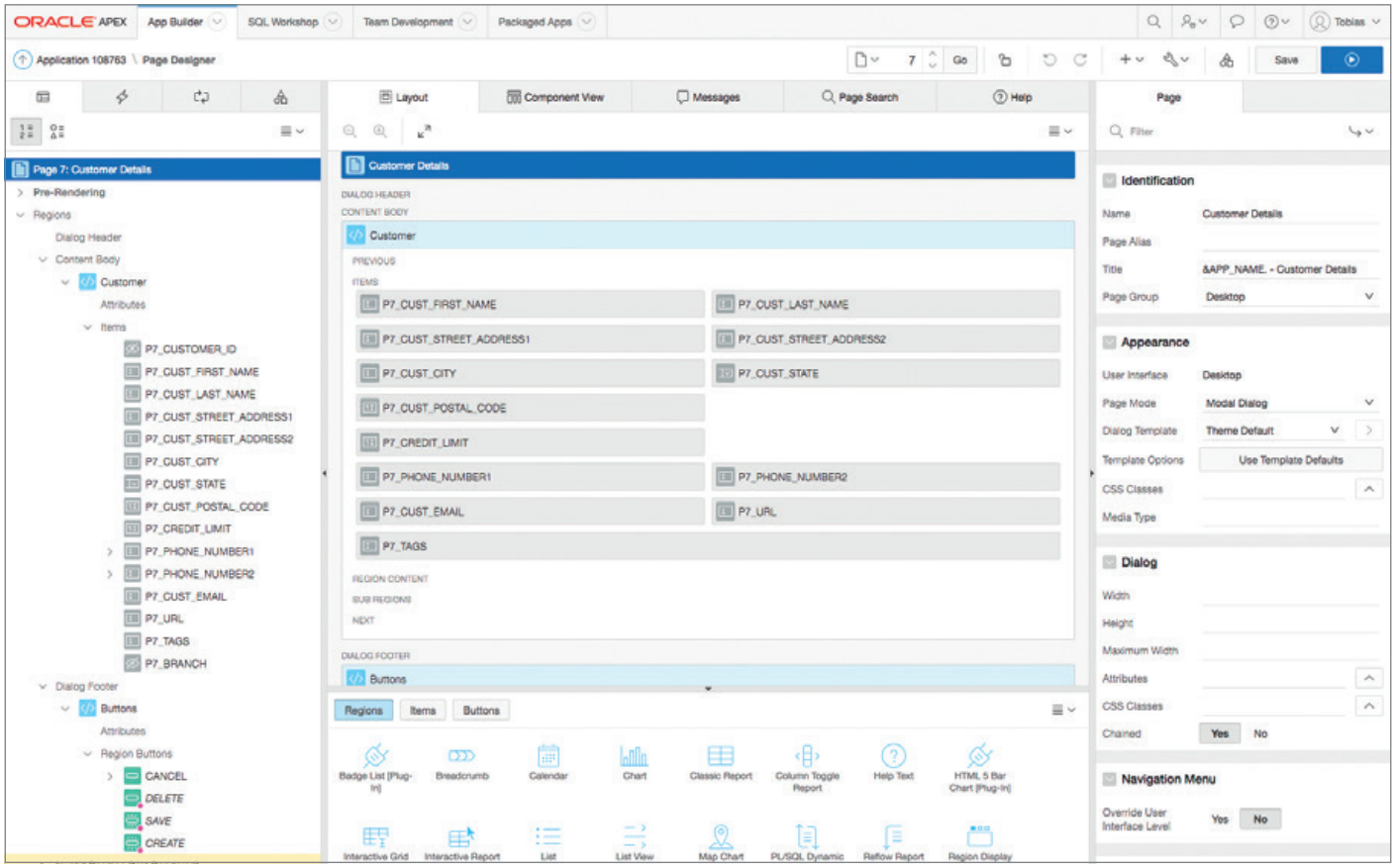


Abbildung 1: Der Page Designer in Apex 18.1

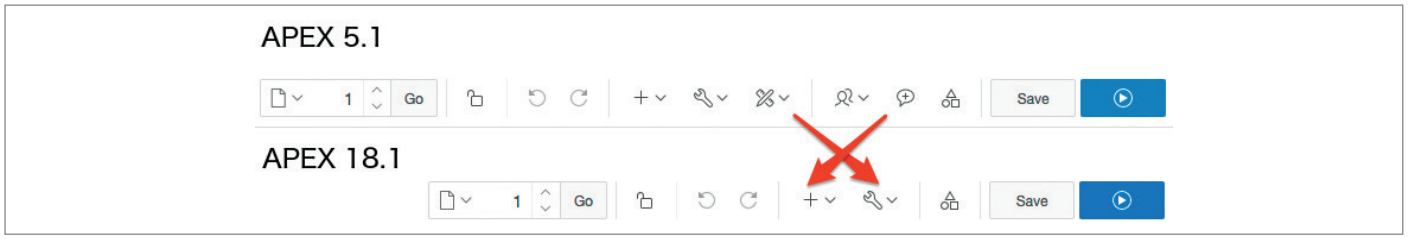


Abbildung 2: Aufgeräumte Schnellnavigationsleiste

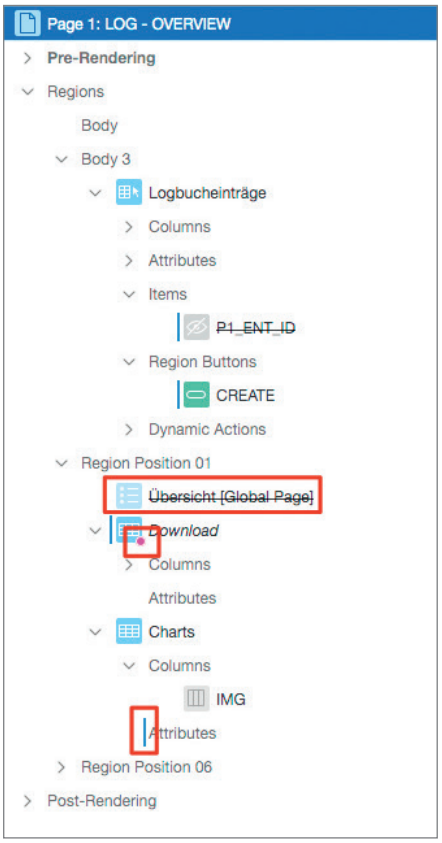


Abbildung 3: Visuelle Hervorhebung bei „Server-side Condition" gleich „Never"

durch Tastenkürzel erweitert. Neu im Bereich „Settings" ist die Option „Show Tooltips", mit der sich die Anzeige der kleinen, schwarzen Tool-Tipps deaktivieren lässt *(siehe Abbildung 2)*.

Seitenobjekte und Attribute

Objekte mit Bedingung („Server-side Condition") erhalten nun einen markanteren roten Punkt im Objekt-Icon. Objekte wiederum mit Bedingung „Never" sind in Apex 18.1 durchgestrichen und mit verringerter Deckkraft angezeigt. Dies bringt insbesondere aus Sicht der Code-Bereinigung einen großen Vorteil. Generell erhalten alle geänderten Objekte vor dem Speichern einen blauen Balken, wodurch die Änderung auch im Page-Rendering-Menü sichtbar wird *(siehe Abbildung 3)*.

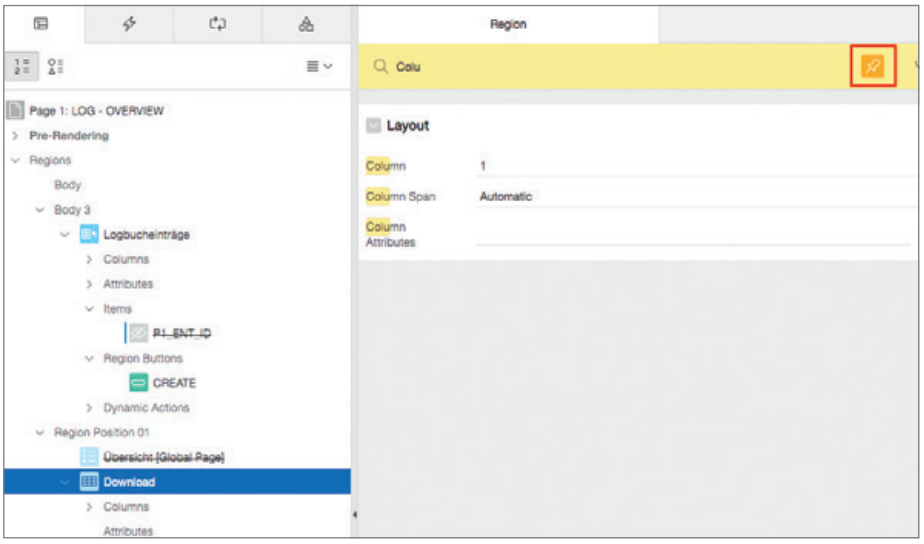


Abbildung 4: „Pinned" Attribut-Suche

Die Suche in den Objekt-Attributen kann nun gepinnt werden, dadurch bleibt der Filter solange erhalten, bis er wieder freigegeben ist. Dies ist insbesondere hilfreich, wenn Objekte nacheinander abgearbeitet werden müssen und die Inhalte variieren können (beispielsweise „Custom Attributes", siehe *Abbildung 4*).

Generell wurde die Visualisierung in der Attribut-Bearbeitung leicht modifiziert, wodurch die einzelnen Blöcke besser auseinandergehalten werden können. Über das „Go to Group"-Icon werden nun alle anderen Gruppen automatisch eingeklappt und die Inhalte der ausgewählten Gruppe hervorgehoben dargestellt. Dies bringt in der täglichen Arbeit ebenfalls wieder einen kleinen Performance-Vorteil gegenüber älteren Apex-Versionen *(siehe Abbildung 5)*.

Laufzeit-Fehler

Ein ebenfalls sehr nützliches neues Feature ist die Visualisierung von JavaScript-Fehlern während der Laufzeit. Im Fehlerfall erscheint in der Developer-Toolbar (links) ein Ausrufezeichen, das auf einen JavaScript-Fehler hinweist *(siehe Abbildung 6)*. Die genaue Fehler-Analyse muss

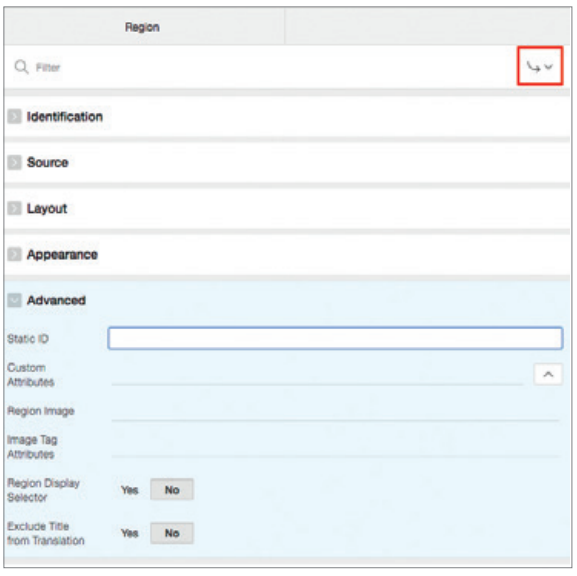


Abbildung 5: Schnellnavigation nach Gruppe

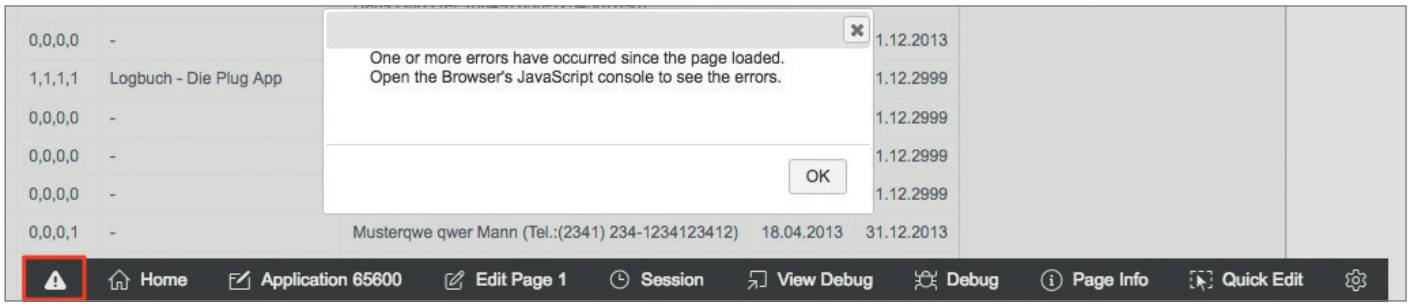


Abbildung 6: JavaScript-Error

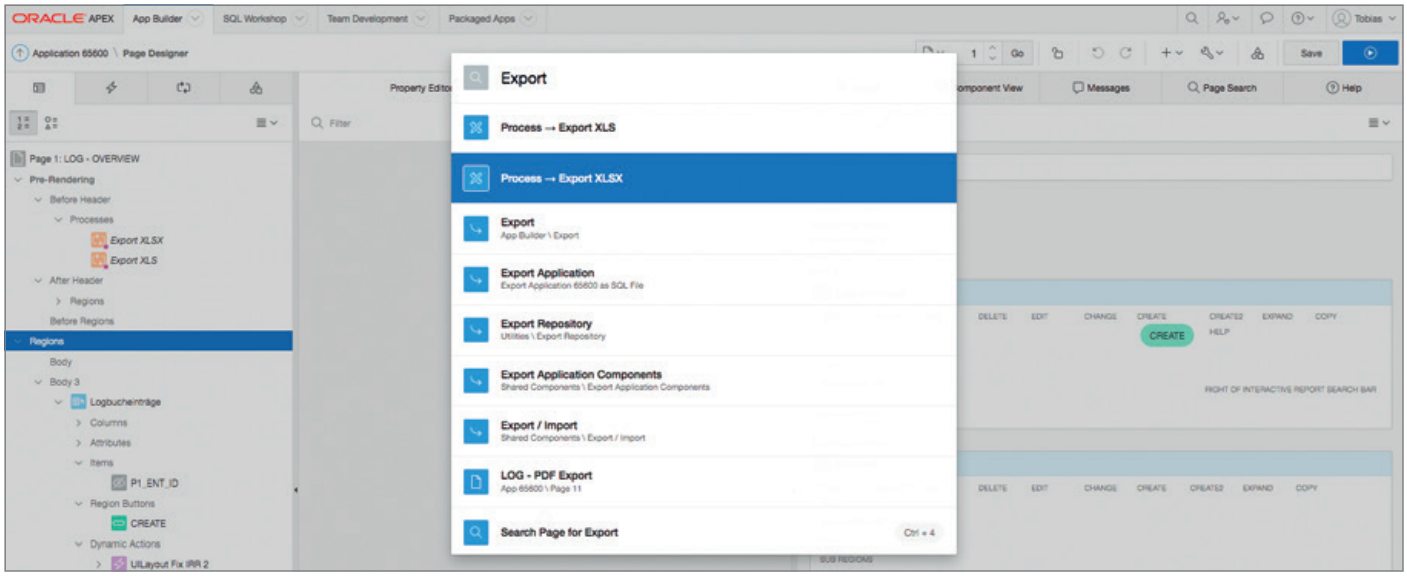


Abbildung 7: Spotlight Search

dann über die Developer-Tools des Web-Browsers erfolgen.

Component View

Die Component View ist tot, lang lebe die Component View. In Apex 18.1 gibt es davon eine neue Variante. Diese besteht nur noch aus zwei Bereichen: „Page Rendering" und „Page Processing". „Shared Component" und „Page Attributes" fallen weg.

Spotlight-Suche

Ein größeres neues Feature innerhalb des Page Designer und der Apex-Entwicklungsumgebung selbst ist die Spotlight-Suche, früher bekannt als „Search Application". Dieses Feature ist stark erweitert, da viel mehr Anwendungsinformationen in das Ergebnis integriert sind. Das Ergebnis selbst wird in der Suchleiste vorgelistet, ähnlich dem „Text Field with autocomplete". Die Tastenkombination „CTRL + Hochkomma" beziehungsweise auf deutschen Tastaturlayouts „CTRL + ä" startet die Spotlight-Suche direkt. So kann man nach einem Seitenobjekt suchen und direkt dahin navigieren, ohne die Maus verwenden zu müssen *(siehe Abbildung 7)*.

Fazit

Apex 18.1 bringt viele Detail-Änderungen im Page Designer mit sich. Dem Autor gefallen vor allem das neue Highlighting von Änderungen und Bedingungen im „Page Rendering" sowie die Spotlight-Suche und die Attribut-Navigation/-Suche. Das Apex-Team zeigt einmal mehr, was es heißt, Community-Feedback richtig zu verarbeiten.

Wichtige Links

- Software/Dokumentation: *https://apex.oracle.com*
- New Features: *https://docs.oracle.com/database/apex-18.1/HTMRN/toc.htm#HTMRN-GUID-D6D545CB-3ECD-468A-9E7F-8CC09F7F478B*
- Blog: *apex-at-work.com*
- Community Portal: *https://apex.world*

Tobias Arnhold
tobias-arnhold@hotmail.de

Ja, wo laufen sie denn? Tracking und Tracing in 2D, 3D und 4D

Hans Viehmann, ORACLE Deutschland B.V. & Co. KG

Die Erfassung und Auswertung der Bewegung von Personen, Fahrzeugen und Gegenständen spielt in den unterschiedlichsten Bereichen unseres täglichen Lebens eine wichtige Rolle. Egal, ob es sich um Verkehrsprojekte in Smart Cities, um das Verhalten von Kunden in Einkaufszentren oder um die Auslieferung von Bauteilen in Produktionsketten im Umfeld von Industrie-4.0-Projekten handelt. Überall haben Standort-Daten und die Verfolgung von Objekten einen Anteil an Entscheidungsprozessen.

Oracle bietet seit Jahren für diese Einsatzfelder Technologien zur Verwaltung, Analyse und Visualisierung raumbezogener Daten an, die zu großen Teilen bereits in der Standard Edition enthalten sind. Dieser Artikel stellt die aktuellen Möglichkeiten in Oracle 12.2 Spatial and Graph dar und erläutert die Grundlagen der Anwendungsentwicklung On-Premises und in der Oracle Cloud. Dazu gehören neben der Visualisierung auf Karten auch weitergehende Analysen, die mit der Verwendung des Straßennetzes in der Datenbank einhergehen.

Raumbezogene Daten sind Bestandteil praktisch jeder Datenbank. Sie kommen in verschiedenen Formen vor und beschreiben beispielsweise topografische Strukturen wie Flüsse oder Parks oder auch Positionen von Fahrzeugen oder Smartphones. Aber nicht nur Positionsangaben in Form von geografischen Koordinaten enthalten einen Raumbezug. Auch Umsatzzahlen können räumlich verortet sein, wenn sie einer (geografischen) Vertriebsregion zugeordnet sind. Ebenso stellen Adressen oder Ortsbezeichnungen, wie etwa Sehenswürdigkeiten, raumbezogene Daten dar. Allen diesen Daten kommt eine spezielle Bedeutung zu, weil sie einen Bezug herstellen, wo anderweitig keine Beziehung vorhanden wäre.

Grundlagen

Um in der Datenbank mit raumbezogenen Daten umgehen zu können, sind vier Basis-Funktionalitäten erforderlich. Zunächst braucht man einen Datentyp zur Speicherung räumlicher Daten als Punkte, Linien oder Flächen samt dem zugehörigen Koordinatensystem. Für diese Objekte benötigt man topologische Operatoren, die räumliche Beziehungen bestimmen können; außerdem räumliche Funktionen beispielsweise zur Berechnung von Flächen, Abständen oder Puffzonen. Schließlich braucht es räumliche Indizes, um schnell und gezielt auf Geometrien zugreifen zu können.

Entsprechende Funktionalitäten sind in der Oracle-Datenbank seit dem Jahr 1995 vorhanden und seither sukzessive verfeinert worden. So enthalten alle Editionen den Datentyp „SDO_GEOMETRY“, Operatoren wie „SDO_INSIDE“, Funktionen wie „WITHIN_DISTANCE“ oder die Domain-Indizes „MDSYS.SPATIAL_INDEX“ beziehungsweise „MDSYS.SPATIAL_INDEX_V2“, die im Gegensatz zum klassischen B-Tree auf R-Baum-Strukturen basieren.

Abbildung 1 zeigt den Aufbau von „SDO_GEOMETRY“. Er besteht aus Metadaten wie etwa der Anzahl der Dimensionen, dem Koordinaten-System oder der Geometrie-Klasse (Punkt, Linienzug, Fläche etc.) sowie den eigentlichen Koordinaten, die im Falle eines Punkts in einem „SDO_POINT_TYPE“ beziehungsweise allgemein in einem „VARRAY“ abgelegt sind.

Um beispielsweise eine Position 10 Grad östlicher Länge und 53 Grad nördlicher Breite zu erzeugen, genügt das Statement „select sdo_geometry('POINT (10 53)', 4326) from dual;“. Mit diesen Datenstrukturen muss man sich im Allgemeinen nicht näher befassen, sie werden meist von Tools generiert.

Durchaus gängig sind dagegen topologische Abfragen in SQL. Möchte man etwa feststellen, welche Bundesländer an NRW angrenzen, kann man den Operator „SDO_RELATE“ einsetzen, der überprüft, welche geometrischen Objekte gemeinsame Grenzpunkte besitzen, ohne dass die Geometrien überlappen. Das zugehörige SQL-Statement könnte wie in *Listing 1* aussehen.

Sind in einer Tabelle ausschließlich punktförmige Daten enthalten, lässt sich eine Reihe von Optimierungen nutzen, die die Ausführungszeit minimieren. In dem Falle ist es auch nicht zwingend erforderlich, eine Spalte vom Typ „SDO_GEOMETRY“ anzulegen. Man kann eben-

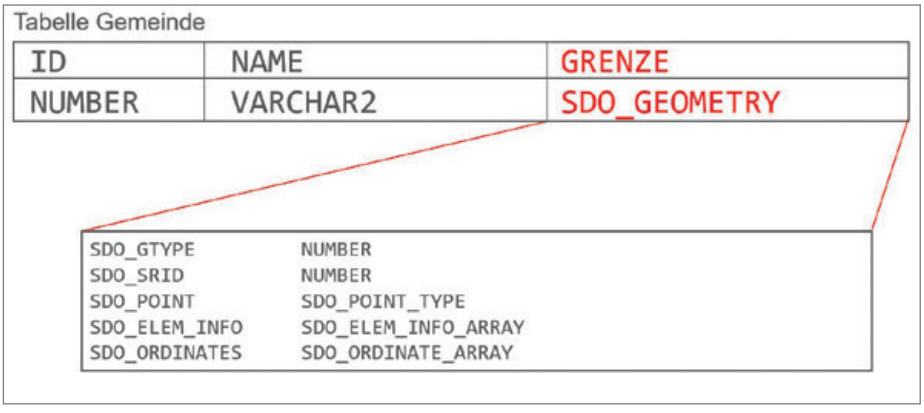


Abbildung 1: Struktur und Einsatz des Datentyps „SDO_GEOMETRY“

```
select l1.name from laender l1, laender l2
where l2.name='NRW' and
sdo_relate(l1.grenze, l2.grenze,'mask=touch')='TRUE';
```

Listing 1

so je eine Spalte für x- und y-Koordinate verwenden und einen „function based"-Index nutzen, der auf „SDO_GEOMETRY" basiert und dessen Funktionswerte dann für die räumliche Abfrage eingesetzt werden.

Neben diesen grundlegenden Technologien umfasst die „Spatial and Graph"-Option zur Datenbank zusätzlich Funktionen zur Umsetzung von Adressen zu Koordinaten (Geokodierung), Routenplanung oder weitere Datentypen speziell zur Verwaltung von Daten in drei Dimensionen. Seit Oracle 12c ist auch eine Reihe von Performance-relevanten Optimierungen mit der Option verfügbar, die für den Einsatz im Umfeld von Data-Warehouse-Projekten oder anderen unternehmensweiten Implementierungen von Bedeutung sind.

Zum Lizenzumfang gehört seit Oracle 12.2 ebenfalls eine Java-Komponente zur Visualisierung von Geodaten aus der Datenbank. Die „Spatial and Graph"-Option ist Bestandteil verschiedener Cloud Services, etwa der Database Cloud Services (High Performance und Extreme Performance Edition). Inzwischen sind unter der Produktbezeichnung „Oracle Big Data Spatial and Graph" auch Funktionalitäten zur räumlichen Analyse auf den Big-Data-Plattformen (Hadoop, Spark, Oracle NoSQL) verfügbar, die auf den gleichen Konzepten basieren.

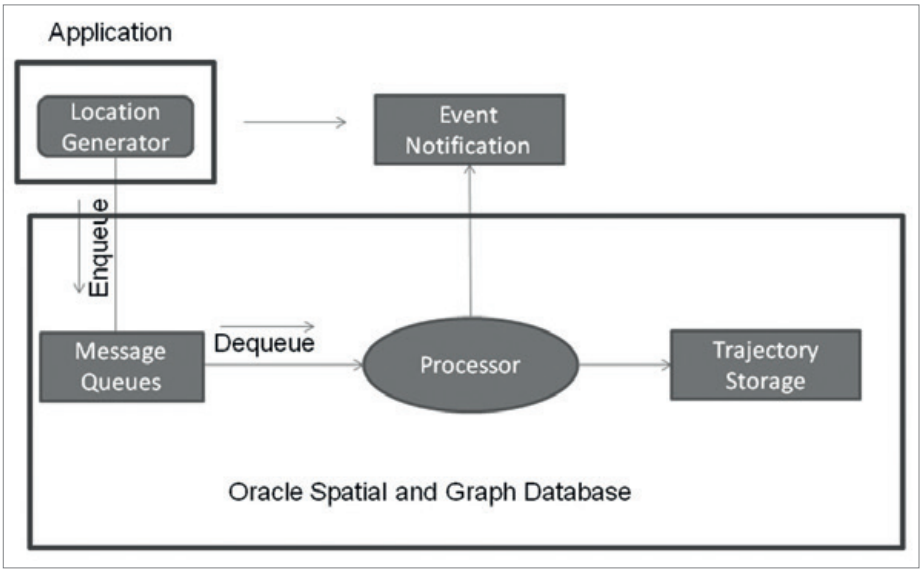


Abbildung 2: Schematischer Aufbau des Tracking-Servers

Location Tracking

Für die Überwachung und Rückverfolgung von Objekten, die mit einem Positions-Sensor verbunden sind, bietet Spatial and Graph seit Release 12.2 einen Satz von Programmier-Schnittstellen in Java und PL/SQL an, der auf den oben beschriebenen Basis-Funktionalitäten aufsetzt. Dieser Tracking-Server ist dafür gedacht, zahlreiche Objekte, die fortlaufend ihre Koordinaten senden, darauf zu überprüfen, ob sie vorgegebene Bereiche verlassen (Geofencing) oder ein vordefiniertes Zielgebiet erreichen. Die Implementierung basiert auf Advanced Queueing, um eine asynchrone Verarbeitung der Positionsdaten zu ermöglichen. Jede Position wird mithilfe des entsprechenden topologischen Operators darauf überprüft, ob sie die für das Objekt relevante Flächengeometrie berührt, wobei natürlich zur räumlichen Suche der Spatial-Index zum Einsatz kommt *(siehe Abbildung 2)*.

Technisch ist die Vorgehensweise wie folgt: Zunächst müssen die zu überwachenden Flächen definiert, im nächsten Schritt der Tracking-Server initialisiert,

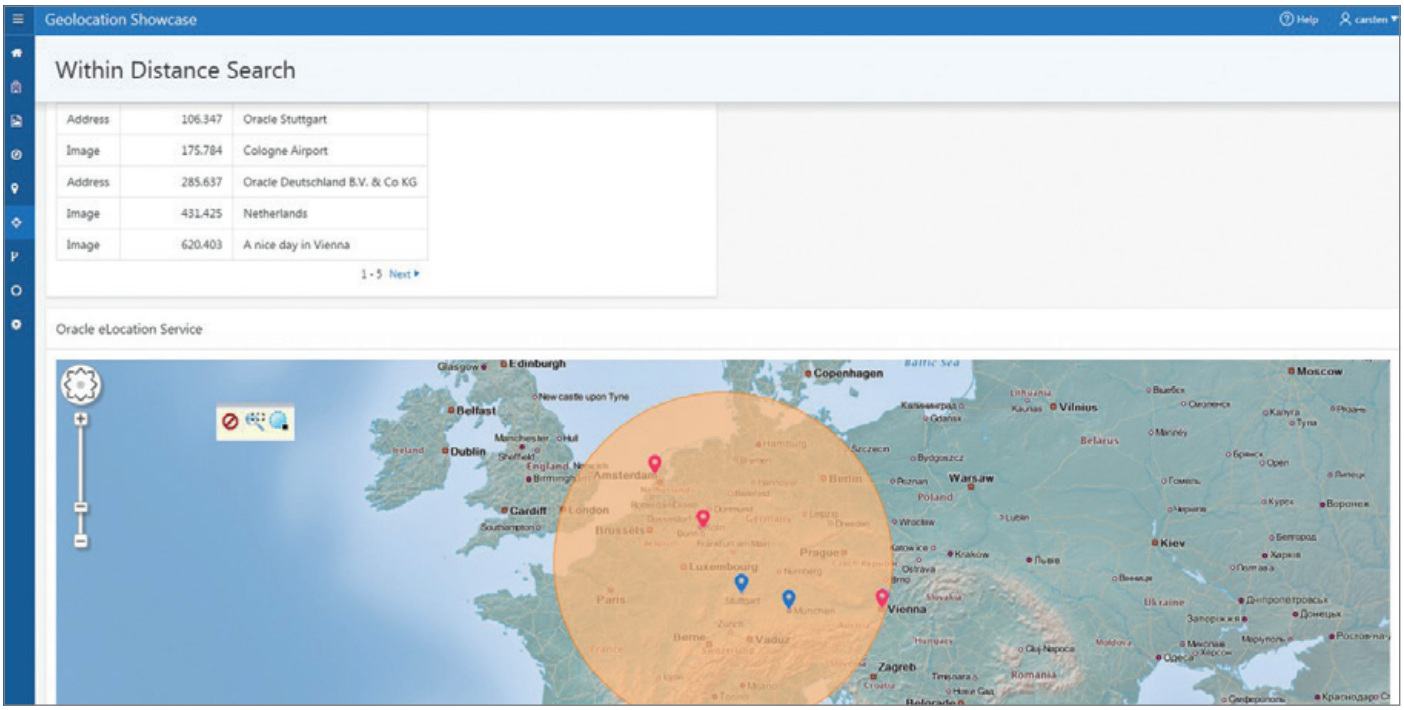


Abbildung 3: Kartendarstellung in Oracle-Apex (Geolocation Showcase)

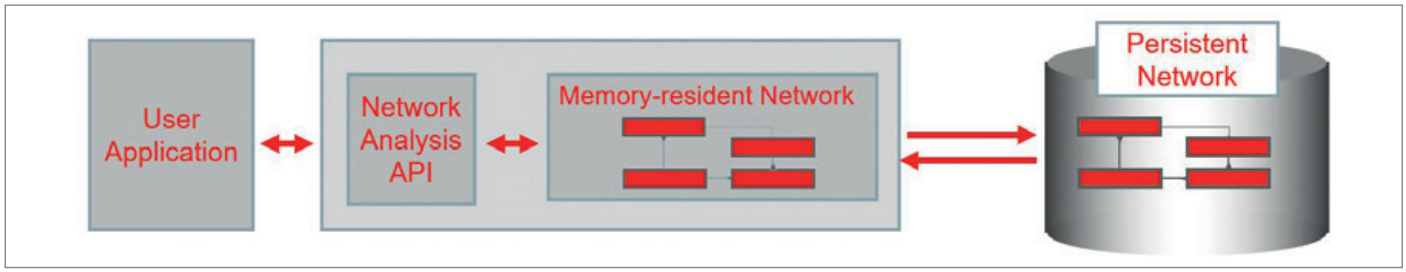


Abbildung 4: Schematische Darstellung der Routing Engine in Oracle Spatial and Graph

anschließend die zu überwachenden Objekte definiert und den jeweiligen Flächen zugeordnet und ab dann Positionsmeldungen je Objekt über die entsprechende Queue an den Tracking-Server gesendet werden. Das „SDO_TRKR"-Package stellt die erforderlichen APIs bereit. Tritt ein relevantes Ereignis ein, verlässt also ein Objekt die vorgegebene Region beziehungsweise erreicht das Zielgebiet, wird in der Notification Queue eine entsprechende Meldung eingestellt, die dort ausgelesen werden kann. Hier stehen alle Möglichkeiten zur Verfügung, die Advanced Queueing anbietet.

Grafische Darstellung auf Karten

Für die Darstellung der Ereignisse im räumlichen Zusammenhang ist es hilfreich, die relevanten Objekte auf einer Karte zu visualisieren. Zu dem Zweck ist in der „Spatial and Graph"-Lizenz eine Java-Komponente enthalten, die geometrische Objekte aus der Datenbank anzeigen und bei Bedarf auch statische Karten von „maps.oracle.com", von Google Maps oder Bing Maps als Hintergrund-Daten laden kann. Die Komponente basiert auf HTML5 und steht beispielsweise als Plug-in für Apex zur Verfügung, sodass eine sehr einfache Möglichkeit existiert, die zu überwachenden Flächen aus dem Tracking-Server gemeinsam mit Ereignissen aus der Notification Queue auf einer Karte anzuzeigen.

Die Nutzung von „maps.oracle.com" für die Hintergrundkarten über das Apex-Plug-in ist im Übrigen ohne Lizenzkosten möglich. Für die Entwicklung funktional anspruchsvollerer Anwendungen mit dynamischen Echtzeitdaten gibt es am Markt auch kommerzielle Tools wie beispielsweise LuciadRIA, die etwa in Anwendungen auf Basis von OracleJET integriert werden können und ebenfalls direkt auf die raumbezogenen Daten in der Datenbank zugreifen *(siehe Abbildung 3)*.

Tracking und Tracing auf Straßennetzen

Je nach Anwendungsfall ist es unter Umständen nicht ausreichend, in der Verfolgung von Objekten nur die gesendete Position zu betrachten, weil diese, je nach Methode der Positionsbestimmung, eine mehr oder minder große Ungenauigkeit mit sich tragen kann. Außerdem sagt die Position nicht notwendigerweise etwas über die Wegstrecke zum Ziel aus, wenn man nicht gerade mit dem Hubschrauber unterwegs ist. Hier kann es erforderlich sein, die Bewegungen der Objekte auf dem Straßennetz abzubilden. Zu diesem Zweck ist seit Oracle 10g die Unterstützung von Netzwerk-Datenmodellen in Oracle Spatial and Graph enthalten. Damit ist es möglich, Straßennetze, Versorgungsnetze etc. als abstrakten Graphen in der Datenbank abzulegen und darauf Analysen durchzuführen.

Für Straßennetze bedeutet dies, dass jeder Straßenabschnitt als Kante und jede Abzweigung beziehungsweise Kreuzung als Knoten abgebildet und samt der zugehörigen Eigenschaften wie Fahrtrichtung der Einbahnstraße, Abbiegevorschriften, Geschwindigkeits- oder Gewichtsbegrenzungen in der Datenbank verwaltet werden kann. Entsprechende Referenz-Datenbestände sind von kommerziellen Anbietern wie HERE oder TomTom für Oracle Spatial and Graph fertig aufbereitet als Transportable Tablespaces verfügbar. Auch für OpenStreetMap existiert ein Konverter von CISS TDI, der die Daten für Oracle Spatial and Graph in ein routingfähiges Modell umwandelt *(siehe Abbildung 4)*.

Damit wird es möglich, ungenaue Positionsdaten von Fahrzeugen auf das nächstgelegene Straßensegment zu projizieren und über den Routenverlauf trotz unscharfer Standorte zu einer exakten Routen-Geometrie zu kommen. Außerdem steht auf diesem Netzwerk-Datenmodell eine Routing-Engine zur Verfügung, die bei gegebener Position beispielsweise die kürzeste oder schnellste Strecke zum Ziel berechnen kann. Dafür nutzt sie eine Java-Laufzeitumgebung, in der die relevanten Teile des Netzwerks im Hauptspeicher vorgehalten werden. Seit Oracle 12.2 ist die Routing-Engine sogar in der Lage, für die Berechnungen Wochentag und Tageszeit sowie davon abhängige Fahrzeiten auf Basis historischer Durchschnittsgeschwindigkeiten zu berücksichtigen.

Event Processing Engines

Die oben beschriebene Implementierung des Tracking-Servers skaliert mit der Datenbank und ist durchaus in der Lage, eine große Anzahl bewegter Objekte zu überwachen. Für anspruchsvollere Anwendungen, die nicht nur raumbezogene Analysen auf Sensordaten und ähnlichen Datenströmen zum Ziel haben, reicht eine Datenbank-basierte Lösung allerdings irgendwann nicht mehr aus. Hier bietet sich konzeptionell eine ereignisgesteuerte Architektur („Event-driven Architecture") an, die im Kern auf einer Event Processing Engine basiert. Unter dem Produktnamen „Oracle Stream Analytics" findet sich eine entsprechende Plattform im Fusion-Middleware-Portfolio. Diese ist ebenfalls für die Analyse räumlicher Daten sehr gut geeignet, da sie ähnlich der Datenbank geometrische Objekte kennt, topologische Operatoren und geometrische Funktionen umfasst und eine räumliche Indizierung auf Basis von R-Trees enthält.

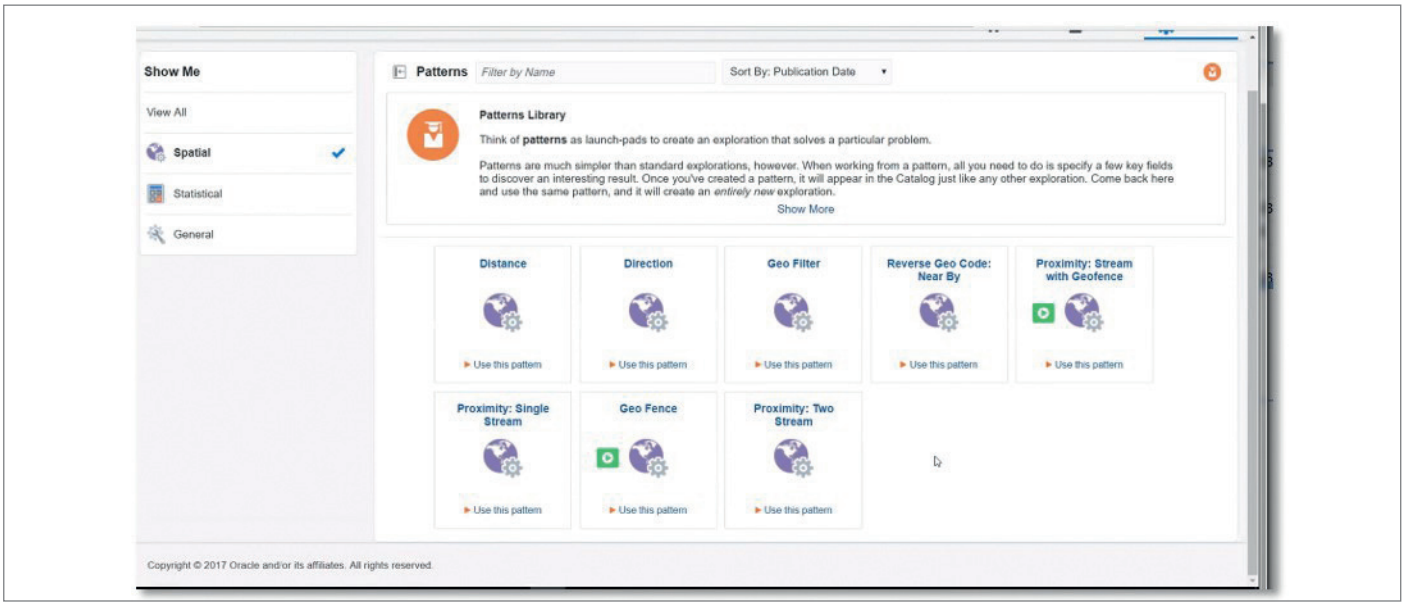


Abbildung 5: Screenshot der Design-Patterns für räumliche Daten in Oracle Stream Analytics

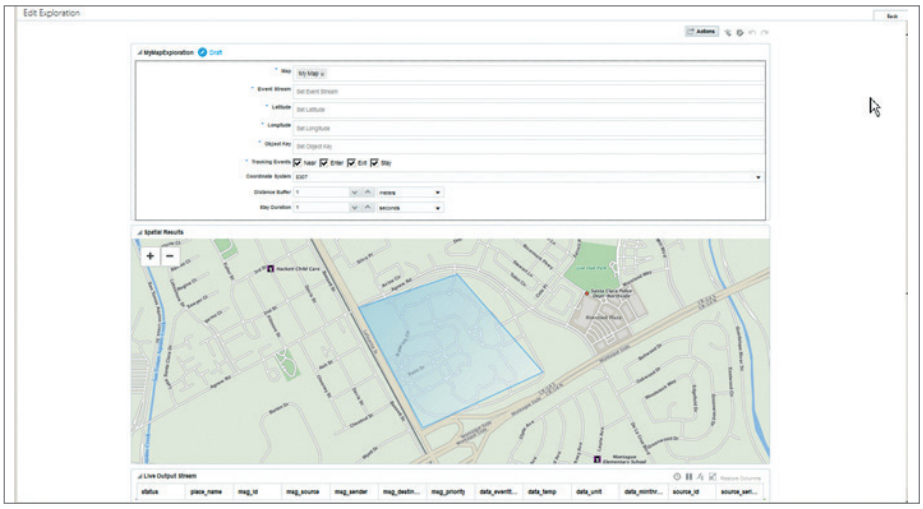


Abbildung 6: Screenshot einer Flächen-Definition im Oracle IoT Cloud Service

Oracle Stream Analytics ist in der Lage, auf Datenströmen aus verschiedensten Quellen in Echtzeit Analysen auszuführen. Über eine grafische Oberfläche lassen sich Verarbeitungsschritte auf den Datenströmen entwerfen, weitestgehend ohne dass Codierung notwendig wäre. Die Engine bietet zahlreiche vorgefertigte Design-Patterns, die Korrelation, Aggregation, Filterung, Mustererkennung etc. bereits abdecken. Unter den Design-Patterns speziell für Positionsdaten befinden sich auch mehrere Varianten von geografischen Abstandsberechnungen zwischen Objekten, Geofencing, räumliche Filterung etc. Die Verarbeitung der eingehenden Meldungen erfolgt Event-basiert in Verarbeitungs-Pipelines mit dem Ziel, aus potenziell riesigen Datenmengen nur die relevanten Ereignisse zu extrahieren und an einen Empfänger weiterzuleiten *(siehe Abbildung 5)*.

Oracle Stream Analytics wird beispielweise von Telematik-Dienstleistern zum Management von Millionen von Fahrzeugen eingesetzt. Die Engine kommt aber auch im Oracle Internet of Things Cloud Service zum Einsatz. Dank der grafischen Benutzeroberfläche lassen sich damit sehr schnell komplette Anwendungen entwickeln, die von der Anbindung unterschiedlicher Geräte an den Cloud Service über den Umgang mit verschiedenen Sensoren oder Meldungsformaten bis hin zur Integration diverser Unternehmensanwendungen alle notwendigen Funktionalitäten umfassen *(siehe Abbildung 6)*.

Die Definition von flächenhaften Objekten für Geofencing mithilfe einer Karte ist ebenso enthalten wie die Vorgabe der unterschiedlichen raumbezogenen Ereignisarten (Objekt bewegt sich in die vorgegebene Region hinein, Objekt bewegt sich aus der Region heraus, Objekt unter- oder überschreitet einen vorgegebenen Abstand oder behält einen der Zustände eine längere Zeit bei).

Fazit

Cloud Services wie dieser sind sicher erst der Anfang einer Entwicklung, im Rahmen derer die Nutzung von raumbezogenen Daten in verschiedenen Einsatzfeldern wesentlich einfacher wird als in der Vergangenheit.

Hans Viehmann
hans.viehmann@oracle.com

Objekte machen das Leben leichter – reloaded

Jürgen Sieben, ConDeS GmbH & Co. KG

In der letzten Folge wurden Objekte in sehr einfacher Form eingesetzt, um den Code eleganter zu machen. Eine Erweiterung dieser Strategie setzt ein komplett neues Denken über Code in Gang, das zwar in objektorientierten Programmier-Umgebungen seit Langem bekannt ist, im Umfeld von PL/SQL allerdings nur sehr selten eingesetzt wird: das Interface.

Als Beispiel nehmen wir an, in einem Programm müssten die Metadaten von Bankanweisungen verwaltet werden. Diese können unterschiedlichen Typs sein, wie zum Beispiel eine Überweisung, ein Dauerauftrag oder ein Bankeinzug. Diesen Bankanweisungen gemein sind einige Attribute, wie etwa die Bankverbindung, der Betrag, der Adressat etc. Andere Angaben sind auf den jeweiligen Typ beschränkt, so ist beispielsweise nur bei einem Dauerauftrag die Angabe eines Zeitraums erforderlich, vielleicht könnte ein Dauerauftrag auch für eine bestimmte Zeit ausgesetzt werden.

In unserer Anwendung benötigen wir nun Code, der mit diesen verschiedenen Bankanweisungs-Typen umgehen kann. Nehmen wir an, wir müssten eine Überweisung erfassen, signieren und ausführen können; je nach Zustand, in dem sich die Überweisung befindet, könnte es möglich sein, sie zu stornieren. Wir benötigen also Packages, die diese Methoden implementieren. Da die unterschiedlichen Bankanweisungen jedoch unterschiedliche Geschäftslogik erfordern (der Dauerauftrag unterliegt vielleicht anderen Freigabeverfahren als eine Überweisung, Bankeinzüge dürfen nur von bestimmten Mitarbeitern erfasst werden etc.), endet das Ganze mit mehreren Packages für jede Art der Banküberweisung, die zum Teil extrem ähnliche Arbeiten verrichten.

Auf der anderen Seite werden wir Code schreiben, der mit allen Typen von Bankanweisungen umgehen können soll, etwa in zentralen Such-Dialogen oder in zentralisierten Packages zur Signatur von Bankanweisungen. In diesen Packages benötigen wir nun „CASE“-Anweisungen, die – je nach Typ – unterschiedliche Packages aufrufen. Da sich die Funktionalität so stark ähnelt, wird man eventuell

```
create or replace type bankanweisung
  authid definer
as object(
  id number,
  iban char(22 byte),
  bic varchar2(11 byte),
  empfaenger varchar2(50 char),
  betrag number,
  -- ...
  member function get_iban(
    self in out nocopy bankanweisung)
    return varchar2,
  member function get_bic_11
    return varchar2,
  member procedure speichern(
    self in out nocopy bankanweisung),
  member procedure signieren(
    self in out nocopy bankanweisung),
  member procedure anweisen(
    self in out nocopy bankanweisung),
  member procedure widerrufen(
    self in out nocopy bankanweisung,
    p_grund in varchar2)
  -- ...
) not final not instantiable;
/
```

Listing 1: Basistyp der Bankanweisung

darüber nachdenken, die „CASE"-Anweisungen in den Fällen, in denen mehrere Typen von Bankanweisungen gleich arbeiten, auf den gleichen Code zeigen zu lassen, was dazu führt, dass nun die einfache Zuordnung „Ein Typ - ein Package" nicht mehr durchgehalten wird. Das Chaos ist komplett.

Um dieses Chaos zu umgehen, wird bei objektorientierten Sprachen ein Interface verwendet: Ein Objekt beinhaltet alle Attribute, die alle Bankanweisungen enthalten sowie die Methoden, die unsere Anweisungen aufweisen können sollen. Dieses Objekt dient anschließend als „Blaupause" zur Erstellung der unterschiedlichen Arten von Anweisungen. Dieses Objekt nennen wir „BANKANWEISUNG" und definieren es wie in *Listing 1*.

Der Code ist wie eine Package-Spezifikation zu lesen. Die Attribute „IBAN", „BIC" etc. sind mit öffentlichen Package-Variablen vergleichbar, die Methoden entsprechen den Package-Methoden. Zusätzlich enthalten all diese Methoden einen Verweis auf sich selbst („SELF"). Es ist darauf zu achten, dass dieser Parameter genau so heißen muss und nicht etwa „P_SELF" oder ähnlich. Zum Schluss folgen noch zwei Klauseln, die festlegen, dass wir von diesem Objekt andere Objekte ableiten können („NOT FINAL") und dass es nicht möglich ist, dieses Objekt selbst zu benutzen. Dazu im Folgenden mehr.

Zu dieser Deklaration gehört nun noch, wie auch bei Packages, eine Implementierung des Objekts, in der festgelegt wird, was das Objekt tun soll, wenn eine der gezeigten Methoden aufgerufen wird. Dabei orientieren wir uns an folgender Regel: Methoden, die für alle Typen von Bankanweisungen gleich sind, werden komplett implementiert; Methoden, die je nach Typ unterschiedlich sind, werden nur als Stub angelegt („begin null; end;"). Im Beispiel können die Zugriffsmethoden „GET_IBAN" und „GET_BIC_11" (die Methoden liefern eine lesbare Version der IBAN gemäß DIN 5008 mit Leerzeichen sowie eine auf elf Zeichen erweiterte BIC) direkt im Objekt implementiert werden, während die Methoden zum Speichern, Signieren, Ausführen und Widerrufen wohl eher als Stub implementiert werden.

Aufbauend auf diesem Objekt können nun weitere Bankanweisungs-Typen erstellt werden. Das Besondere: Diese Objekte basieren auf dem oben erstellten Objekt und erweitern beziehungsweise überschreiben es. Die bereits implementierten Methoden müssen nicht neu implementiert werden, sondern können bei den abgeleiteten Objekten weggelassen werden. Benötigen wir also ein Objekt „UEBERWEISUNG", wird dies von „BANKANWEISUNG" abgeleitet und überschreibt die bislang nicht implementierten Methoden. *Listing 2* zeigt ein solches Objekt.

Dass dieser Typ von „BANKANWEISUNG" abgeleitet ist, erkennen Sie an der Klausel „UNDER" in der Deklaration. Da dieser Typ „INSTANTIABLE" ist (die Klausel „NOT INSTANTIABLE" fehlt), kann von ihm ein konkretes Objekt abgeleitet werden. Dieses Objekt wird über eine Methode erzeugt, die als „Konstruktor" bekannt ist und genauso heißt wie der Typ, der durch die Methode erzeugt wird. Dieses Prinzip hatten wir schon bei den Nested Tables kennengelernt. Der Konstruktor erwartet einige der Attribute wie etwa die IBAN, eventuell eine BIC, den Empfänger sowie den Betrag und belegt, nach entsprechender Prüfung, die internen Attribute mit den ermittelten Werten. Eine Sequenz könnte in dieser Methode eine ID ermitteln, unter der das Objekt später in einer Tabelle abgelegt werden kann.

Auch dieser Typ wird einen Objektkörper erhalten. Er benötigt allerdings eine komplexere Implementierung und ist es sinnvoll, nicht in den Objektkörpern die Logik zu implementieren, sondern diese in entsprechende Packages auszulagern. Der Grund dafür ist, dass Packages über mächtigere Möglichkeiten verfügen als Objektkörper (zum Beispiel private Hilfsmethoden und Packagevariablen). Wie das funktioniert, steht im Code zum Artikel unter *„https://github.com/j-sieben/Oracle_OO/blob/master/Objekte%20als%20Interface/Script.sql"*. Nun kann also für das Objekt „UEBERWEISUNG" ein Package „UEBERWEISUNG_PKG" angelegt werden, das die Implementierung der Methoden übernimmt. Bei eher trivialem Code lassen sich jedoch auch die Methoden direkt im Objekt implementieren. Auch dieses Objekt ist „NOT FINAL", sodass zum Beispiel ein Dauerauftrag von diesem Typ abgeleitet werden könnte: Er erbt alle Methoden der Überweisung, ergänzt um die Dinge, die bei einer Dauerauszahlung anders gemacht werden müssen.

Was bringt uns das nun? Oberhalb der Packages, die die Logik für die verschiedenen Arten der Bankanweisungen implementieren, sitzt nun eine Objekthierarchie mit aufeinander aufbauenden Objekttypen. Dieser zusätzliche Over-

```
create or replace type UEBERWEISUNG under BANKANWEISUNG(
  overriding member procedure speichern(
    self in out nocopy ueberweisung),
  overriding member procedure signieren(
    self in out nocopy ueberweisung),
  overriding member procedure anweisen(
    self in out nocopy ueberweisung),
  overriding member procedure widerrufen(
    self in out nocopy ueberweisung,
    p_grund in varchar2),
  constructor function ueberweisung(
    self in out nocopy ueberweisung,
    p_iban in varchar2,
    p_bic in varchar2 default null,
    p_empfaenger in varchar2,
    p_betrag in number)
    return self as result
) not final;
/
```

Listing 2: Implementierung des Typs „UEBERWEISUNG“

```
declare
  l_anweisung BANKANWEISUNG;
  l_betrag number := 123.45;
begin
  l_anweisung := UEBERWEISUNG('DE…', 'ABCDEF…', 'Willi Mueller', l_betrag);
end;
/
```

Listing 3

head bringt uns mehrere entscheidende Vorteile: Zunächst ist es möglich, eine Variable vom Typ „BANKANWEISUNG“ zu erstellen und in dieser ein beliebiges, von diesem Typ abgeleitetes Objekt zu speichern. In der Variablen dieses Typs könnte also auch eine Überweisung gespeichert werden *(siehe Listing 3)*.

Der Vorteil: Wir können nun Code schreiben, der mit beliebigen Bankanweisungen umgehen kann, auch mit solchen, die wir bislang noch gar nicht definiert haben: Eine Methode mit einem Parameter vom Typ „BANKANWEISUNG“ kann mit Instanzen aller abgeleiteten Typen aufgerufen werden.

Dann stellt dieses Verfahren sicher, dass eine Überweisung unmittelbar die IBAN ausgeben kann, obwohl der Typ „UEBERWEISUNG“ die Funktion gar nicht implementiert, und zwar einheitlich nach dem Verfahren, dass der Obertyp implementiert. Der Vorteil hier: Gleiche Funktionalität muss nicht mehrfach erstellt oder in Utility-Packages ausgelagert werden. Wenn der Obertyp etwas bereits kann, muss ein abgeleiteter Typ dies nicht neu implementieren. Erst, wenn er etwas anders machen muss als sein hierarchischer Vorgänger, überschreibt er die Implementierung in einer eigenen Methode.

Nebenbemerkung: Im obigen Code-Abschnitt kommt eine Hilfsvariable „L_BETRAG“ vor, um den Betrag zu übergeben. Hier, wie auch an anderen Stellen in SQL und PL/SQL (XML und JSON im Beispiel), tritt ansonsten ein Problem überschießender Internationalisierungsfreude zutage: Wird ein Betrag mit einem Dezimalpunkt übergeben, wird dieser nicht als gültige Zahl erkannt, weil dieser in den NLS-Einstellungen als Tausender-Trenner angesehen wird. Wenn irgendjemand von Oracle dies liest: Kann man dieses Problem nicht endlich einmal lösen?

Durch die Referenz auf einen Vorgänger ist zudem sichergestellt, dass alle Methoden aller hierarchischen Vorgänger im aktuellen Objekt vorhanden sind. Der Vorteil ist hier, dass Sie keine Fallunterscheidungen beim Ansprechen eines Objektes mehr benötigen. Hierzu als Beispiel: Der Typ „DAUERAUFTRAG“ benötigt eine Methode zum Aussetzen von Überweisungen für einen bestimmten Zeitraum. Daher implementiert der Obertyp „BANKANWEISUNG“ diese Methode als Stub. Nun kann jedes beliebige Objekt angewiesen werden, eine Aussetzung durchzuführen, ohne dass dies zu einem Kompilierfehler führt. Wenn Sie mögen, werfen Sie in der Oberklasse einen Fehler, dass für diesen Typ der Aufruf dieser Methode nicht zulässig sei, um diesen Weg zu sichern, aber der aufrufende Code muss nun nicht mehr zwischen den verschiedenen Ausprägungen der Bankanweisung unterscheiden können. Kann ein konkreter Typ eine Zahlung aussetzen, wird er implementieren, wie dies geschehen soll, ansonsten fällt die Ausführung auf den Stub des Obertyps zurück und macht entweder nichts oder wirft einen Fehler – ganz, wie Sie wollen.

Fazit

Code wird durch dieses Vorgehen deutlich einfacher zu schreiben, wenn auch zu Beginn nicht notwendigerweise einfacher zu verstehen. Dennoch ist die Flexibilität bestechend, die darin besteht, dass Sie einen neuen Bankanweisungs-Typ nachträglich ganz einfach integrieren können, ohne bestehenden Code ändern zu müssen. Schön ist auch, dass wir Funktionalität dort implementieren können, wo sie am sinnvollsten ist, in unserem Beispiel in dem Package, das die Methode des jeweiligen Typs implementiert. Hat man sich an diese Systematik erst einmal gewöhnt, wird komplexer Code deutlich eleganter.

Hinweis: Das Skript zur Dokumentation der Objekt-Hierarchie finden Sie unter *„https://github.com/j-sieben/Oracle_OO/blob/master/Objekte%20als%20Interface/Script.sql“*.

Jürgen Sieben
j.sieben@condes.de

Effiziente Delivery mit APIs, Microservices und DevOps

Sven Bernhardt, OPITZ CONSULTING Deutschland GmbH

Traditionelle IT-Systemlandschaften bestehen oft aus monolithischen Applikationen, die häufig langwierigen, formalisierten Release-Zyklen unterliegen. Aus Sicht der Gesamtstabilität ist das auch sinnvoll, da monolithische Applikationen in der Regel eine Sammlung von eng miteinander verzahnten Business Capabilities abbilden – fachlich bedingte Änderungen sind so allerdings nur innerhalb der Release-Zyklen möglich – und somit wenig agil sind. Konträr dazu steht heute die zentrale Herausforderung der IT, Agilität im Unternehmen sicherzustellen. Änderungen an bestehenden Komponenten sollen schnell durchgeführt und bereitgestellt werden, ohne dass dies Auswirkungen auf bestehende Systeme und Services hat. Definitiv eine Herausforderung! Mit der richtigen Architektur-Idee und geeigneten Werkzeugen jedoch keine unlösbare ...

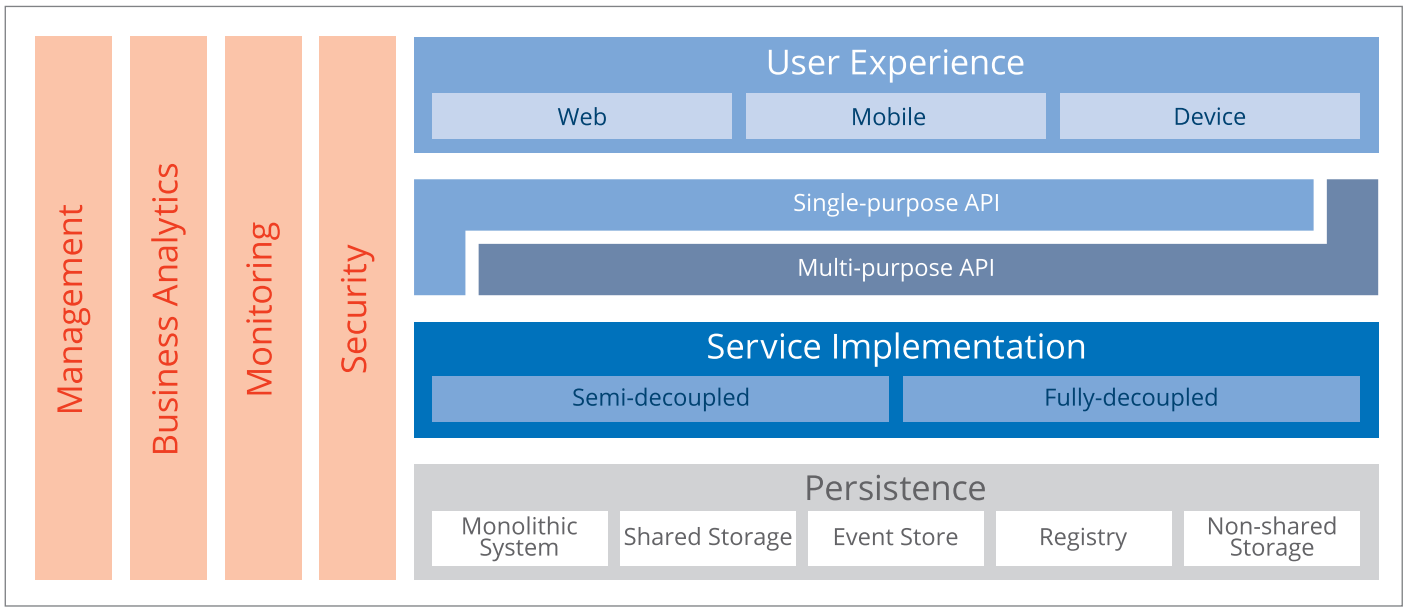


Abbildung 1: OMESA-Referenz-Architektur

Moderne Ansätze wie Microservices-Architekturen *[1]* setzen auf die strikte Trennung unterschiedlicher Business Capabilities, die unabhängig voneinander implementiert, bereitgestellt und betrieben werden können. Ist eine Kommunikation zwischen den Services erforderlich, erfolgt diese in der Regel eventbasiert über einen Event-Hub. Einzelne Microservices sind somit vollständig voneinander entkoppelt. Änderungen, bedingt durch neue oder geänderte Anforderungen, können also ohne Beeinträchtigung bereits vorhandener Funktionalitäten erfolgen; soweit die Theorie.

In der Praxis geht es allerdings nicht ohne ein Umdenken in der Organisation. Betrieb und Entwicklung müssen enger zusammenrücken. Die konsequente Umsetzung eines DevOps-Ansatzes ist unabdingbar für den Erfolg von Microservices, um Vorteile wie die Steigerung von Agilität und Effizienz realisieren zu können. Die Einführung eines DevOps-Ansatzes bedingt neue Denk- und Arbeitsweisen innerhalb der IT-Organisation. Dazu zählen unter anderem:

- Das Aufbrechen getrennter Entwicklungs- und Betriebsbereiche
- Die Formung neuer, heterogener Teams
- Die Automatisierung großer Teile des Entwicklungsprozesses

Das neue Mantra, das die neu geformten Teams in diesem Zusammenhang verinnerlichen müssen, lautet: „You build it, you run it!" Ein Prozess, der nicht von heute auf morgen abgeschlossen ist. Die erfolgreiche Einführung von Microservices ist also mit nicht zu unterschätzenden organisatorischen Herausforderungen verbunden, insbesondere für gewachsene IT-Organisationen. Aber auch technologisch beziehungsweise architektonisch ergeben sich diverse Herausforderungen – denn in den seltensten Fällen startet ein Unternehmen auf der grünen Wiese.

Referenz-Architektur für flexible Anwendungs-Architekturen

Das Projekt der Open Modern Enterprise Software Architecture (OMESA) *[2]* beschäftigt sich mit verschiedenen Fragestellungen, zum Beispiel damit, wie bewährte architektonische Grundprinzipien und Architekturmuster in modernen Software-Architekturen zu verankern sind. Ziel ist es, alte und neue Welt sinnvoll miteinander zu kombinieren. Anstelle kompletter Restrukturierung und Refaktorisierung heißt die Devise „sinnvolle Koexistenz". Ein solches Vorgehen ist gerade in Bezug auf langjährig gewachsene IT-Systemlandschaften sinnvoll.

Eine der zentralen Botschaften von OMESA lautet: „Microservices are no silver bullets!" OMESA definiert zu diesem Zweck eine Referenz-Architektur sowie ein mehrstufiges Capability Model *(siehe Abbildung 1)*. Sie zeigt die zentralen Ebenen, wobei Microservices im Bereich der „Service Implementation" *[3]* einzuordnen sind, als sogenannte „Fully-decoupled Services". Auf der Ebene der „Persistence" befinden sich die bestehenden und zumeist monolithischen IT-Systeme, die über „Semi-decoupled Services" in die Gesamt-Architektur eingebunden sind.

Oberhalb der Ebene „Service Implementation" ist die API-Ebene verortet, die sich wiederum in die Bereiche „Single-Purpose API" und „Multi-Purpose API" aufteilt. Auf der obersten Ebene, der „Delivery Experience", geht es schließlich um die Interaktion mit der Außenwelt; in vielen Fällen handelt es sich hierbei um eine Mensch-Maschine-Interaktion. Dabei geht es vor allem um Themen wie moderne Client-Applikationen oder die Möglichkeit, über verschiedene Kanäle wie Chatbots mit den Services eines Unternehmens interagieren zu können.

APIs verbinden Microservices und UIs

In der OMESA-Referenz-Architektur ist die API-Ebene ein grundlegender Baustein moderner Software-Architekturen. Aber warum sind APIs so essenziell? Um möglichst unabhängig voneinander zu bleiben, interagieren Microservices untereinander hauptsächlich asynchron beziehungsweise eventbasiert. Für die Kommunikation mit der Außenwelt, bei-

spielsweise über Benutzeroberflächen, ist dieser Kommunikationsstil allerdings im Sinne einer guten User Experience (UX) nicht geeignet. Ein synchrones Kommunikationsverhalten, das sich durch zeitnahe Reaktionen auszeichnet, fühlt sich für menschliche Akteure natürlicher an und sollte deshalb auch hier das bevorzugte Kommunikationsmuster sein. Das bedeutet, dass Microservices, deren Funktionalitäten extern verfügbar gemacht werden, ein synchrones Service-Interface, also ein API, bereitstellen müssen.

Der in OMESA propagierte, zweischichtige API-Ansatz, der Single- und Multi-Purpose-APIs unterscheidet, macht die Gesamt-Architektur flexibler und agiler *[4]*. Multi-Purpose-APIs sind allgemeiner, bieten einen breiteren Funktionsumfang und sind somit potenziell wiederverwendbar; Single-Purpose-APIs hingegen können auf den Multi-Purpose-APIs aufbauen und bilden dabei UI- oder Device-spezifische Logik ab. Single-Purpose-APIs stellen im Grunde eine Implementierungsvariante des „Backends For Frontends"-Pattern (BFF) *[5]* dar.

Die API-Ebene dient also vor allem dazu, die Service-Implementierung von der Benutzeroberfläche zu abstrahieren und die von den Services bereitgestellten Funktionalitäten sicher nach außen zu exponieren. „Sicher" bedeutet hier, dass die API-Ebene übergreifende Aspekte wie grundlegende Sicherheitsmechanismen (wie Authentifizierung und Autorisierung), Origin-Controls (Cross-Origin Sharing Resources, CORS *[6]*) oder Threat-Protection-Maßnahmen (wie Rate Limits) zentral und konsistent definiert werden, ohne diese explizit in jedem Service ausimplementieren zu müssen. Das hat den Vorteil, dass sich Backend-Entwickler voll und ganz auf die Implementierung der Geschäftslogik konzentrieren können.

Zwischenfazit der wichtigsten Fakten

Zusammenfassend kann bislang festgehalten werden, dass die Umsetzung moderner Architekturen auf der Basis von Microservices sowohl organisatorische als auch architektonische sowie technische Herausforderungen mit sich bringt. Zudem ist die Etablierung von DevOps essenziell für den Erfolg von Microservices. APIs sind in diesem Zusammenhang wichtig, um Service-Funktionalitäten nach außen anbieten zu können und so Mehrwerte wie die Etablierung neuer digitaler Economies zu generieren.

Im zweiten Teil dieses Artikels sollen die angesprochenen Aspekte kurz anhand eines praktischen Beispielprojekts näher beleuchtet werden. Es handelt sich dabei um eine gemeinschaftliche Entwicklung der Oracle-ACEs Lonneke Dikmans, Lucas Jellema, Luis Weir, Guido Schmutz, José Rodrigues und Sven Bernhardt, die sich in Vorbereitung auf einen Vortrag für das jährlich stattfindende PaaS-Forum im März 2018 als Team zusammenfanden, um einen Showcase auf Basis der Oracle Cloud zu entwickeln. Die Ideen der OMESA-Referenz-Architektur dienten dem Team dabei als architektonische Grundlage für die Implementierung.

Beispielszenario: eine Webshop-Lösung

Als Beispielszenario diente dem Team ein fiktiver Webshop, der auf einer Microservices-Architektur basiert. Jedes

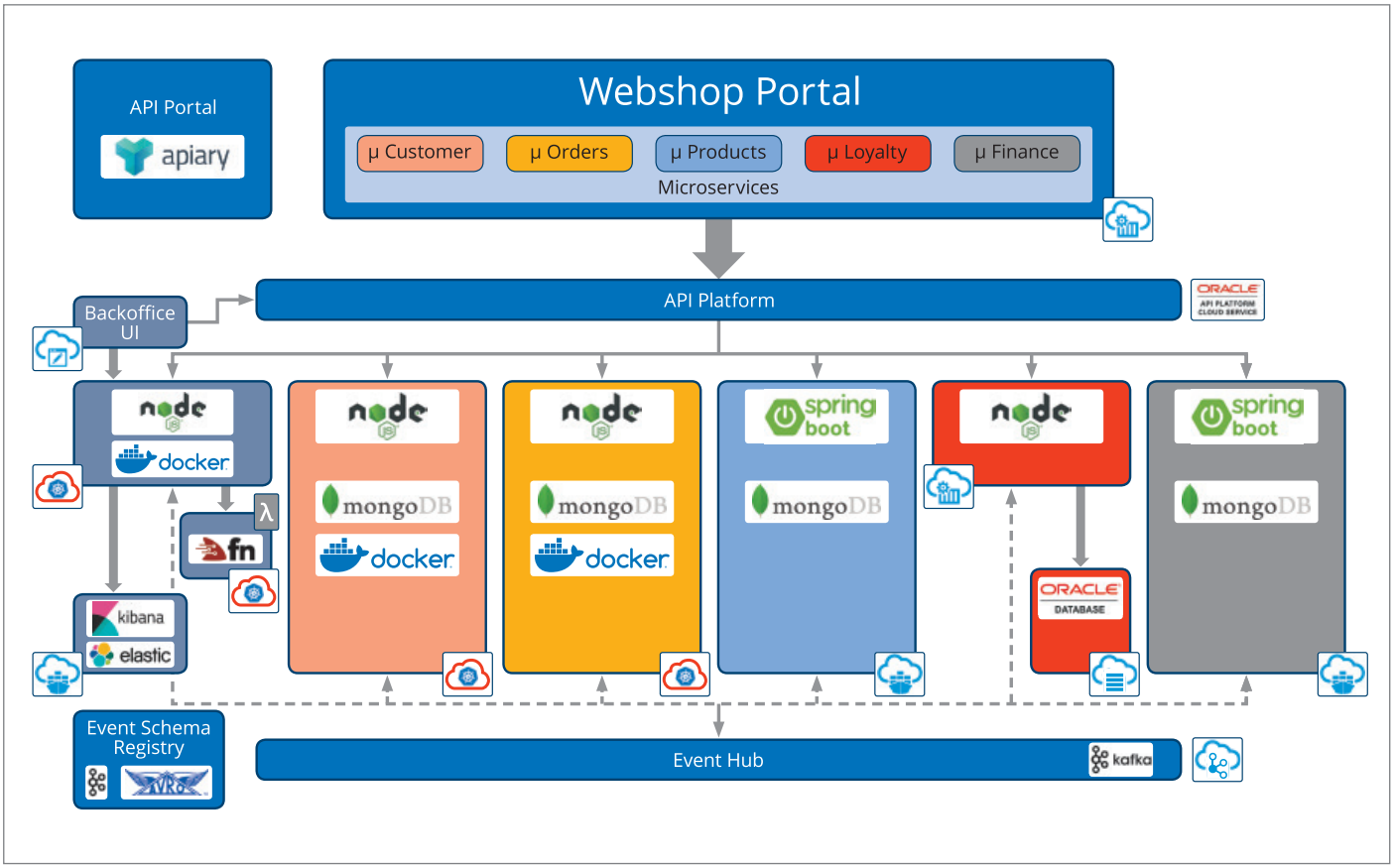


Abbildung 2: Gesamt-Architektur der Webshop-Lösung

Teammitglied war für die Umsetzung einer bestimmten Business Capability, also für jeweils einen Microservice, zuständig. *Abbildung 2* zeigt die Gesamt-Architektur sowie die verwendeten Technologien im Überblick.

Wie der *Abbildung 2* entnommen werden kann, sind bei der Umsetzung viele verschiedene Technologien zum Einsatz gekommen, um den Webshop mit seinen sechs Microservices „Logistics", „Customers", „Orders", „Products", „Loyalty" und „Finance" umzusetzen. Als Laufzeit-Umgebung für die Microservices dienten verschiedene Cloud Services: Oracle Application Container Cloud Service, Oracle Container Cloud Classic und Oracle Container Engine for Kubernetes.

Die Interaktion zwischen den Microservices erfolgt eventbasiert über den Oracle Event Hub Cloud Service. Um die Verbindlichkeit der Event-Definitionen sicherzustellen und die Abstimmungen zwischen den Teams zu vereinfachen, kommt eine Avro Event Schema Registry zum Einsatz *[7]*.

Da die Microservices von UIs verwendet werden sollen, müssen sie REST-APIs anbieten, die über den Oracle API Platform Cloud Service (APIP CS) nach außen exponiert werden. Der Einfachheit halber und da im ersten Schritt nur eine Web UI bedient werden musste, wird auf API-Ebene nicht zwischen Multi- und Single-Purpose-APIs unterschieden.

Beim Webshop-Portal, das auf Oracle JET basiert, handelt es sich um keine klassische, monolithische Web-Applikation. Vielmehr stellt es nur den Rahmen zur Verfügung, der die übrigen Microservice-spezifischen UIs einbindet, was maximale Flexibilität bei Änderungen bedeutet. Wenn beispielsweise aufgrund technischer Anpassungen ein Deployment des Microservice „Finance" notwendig wird, kann dieses jederzeit durchgeführt werden, ohne die übrigen Services zu beeinträchtigen. Benutzer können also weiterhin den Produktkatalog durchsuchen oder Bestellungen durchführen; nur im „Finance"-Bereich kommt es kurzfristig zu Einschränkungen. Die Gesamt-Architektur ist also sehr flexibel aufgebaut. Sie besteht aus Einzelkomponenten, die auf horizontaler Ebene voneinander unabhängig sind. Auf diese Weise kann sehr agil auf sich ändernde Fachanforderungen reagiert werden.

„API first"-Entwicklung

Intuitive APIs sind kritische Erfolgsfaktoren moderner Software-Architekturen. Sie sollten einfach zu bedienen, schwer zu missbrauchen, benutzer- sowie wartungsfreundlich und konsistent definiert sein. Kurzum: Gutes API-Design ist wichtig für die Akzeptanz der Anwender und damit äußerst relevant für die Nutzung eines API. Wie bei der UX für UIs muss man sich also auch hier genauer ansehen, wie das API vom Konsumenten verwendet wird. Deshalb starten wir die Entwicklung nicht etwa mit der Implementierung der Backend-Logik, sondern mit dem Design des API.

Ein „API first"-Design-Ansatz ist für die Flexibilität und Agilität während des gesamten API-Lebenszyklus unerlässlich. Darüber hinaus ermöglicht „API first" die kollaborative Zusammenarbeit verschiedener Stakeholder an einer API-Definition und entkoppelt so API-Implementierung, UI- und Backend-Service-Entwicklung. Dieser Zusammenhang wird in *Abbildung 3* gezeigt.

Für die Microservices des Webshops wurde also zunächst das API beschrieben. Das Team setzte dafür die Oracle Apiary Plattform ein, über die das zugehörige API wahlweise mit Swagger oder API Blueprint beschrieben werden kann. Der Vorteil von Apiary ist, dass das API nach Fertigstellung der Beschreibung durch die Plattform direkt in einer Mock-

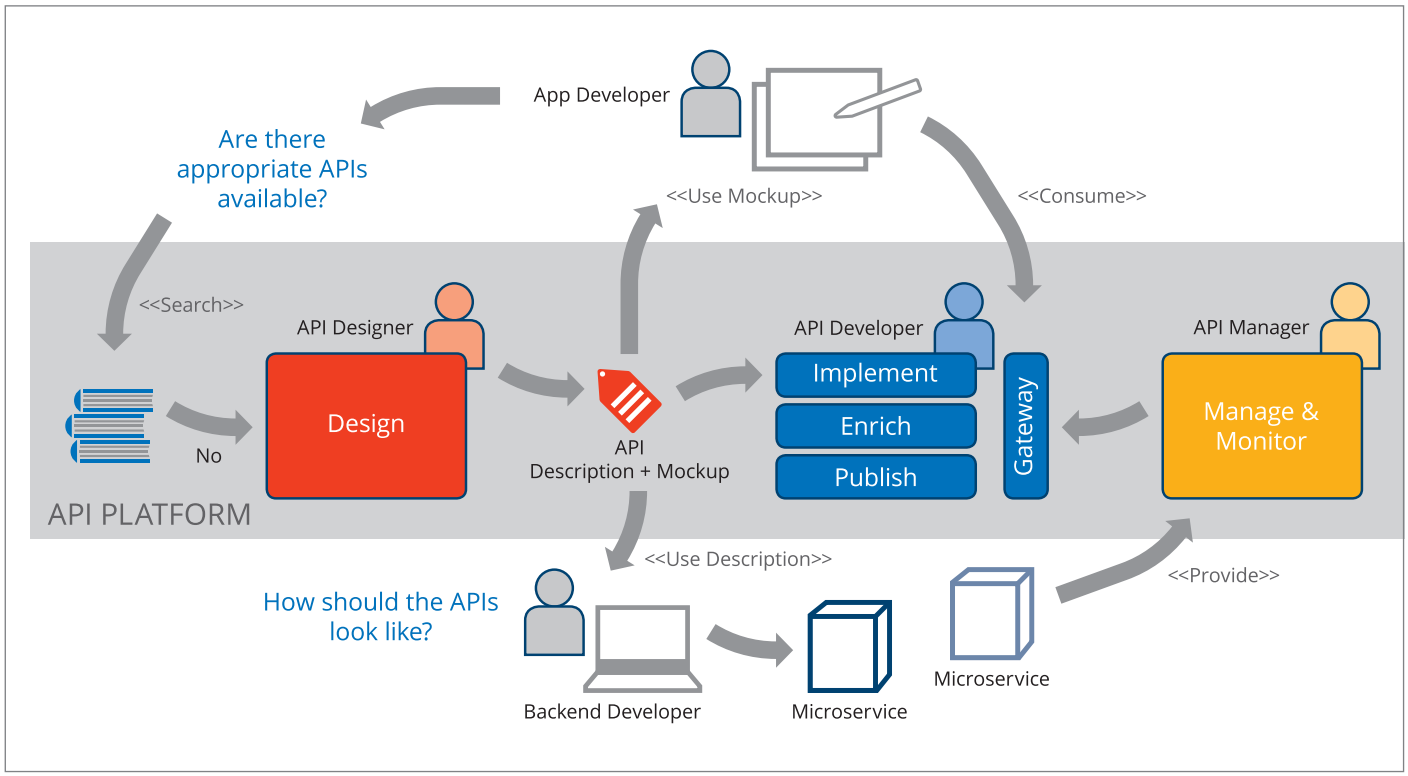


Abbildung 3: „API first"-Design-Ansatz

Variante zur Verfügung gestellt wird und verwendet werden kann.

Wie in *Abbildung 3* dargestellt, können nun App Developer, API Developer und Backend Developer direkt und völlig unabhängig voneinander mit der Implementierung der Web-UI, des API und der Backend-Service-Implementierung starten. Da die API-Beschreibung so allen Stakeholdern bereits zu einem sehr frühen Zeitpunkt zur Verfügung steht, sind Änderungen am API einfach und ohne großen Aufwand möglich. Dank der kurzen Feedbackzyklen kommt man schnell zu einer guten API-Usability.

Test-Automatisierung

Das Thema „Test-Automatisierung" spielt im Kontext von DevOps eine wichtige Rolle. Um Änderungen möglichst schnell produktiv zur Verfügung stellen zu können, ist eine hohe Test-Abdeckung notwendig. Bei der Entwicklung der Backend-Logik der Microservices schrieb das Team Unit-Tests, die dann zum Build-Zeitpunkt mithilfe entsprechender Build-Management-Tools wie Apache Maven automatisiert ausgeführt wurden. In diesem Zusammenhang stellte sich allerdings die Frage, ob und wie es möglich sein würde, Tests für API-Definitionen zu automatisieren. Denn schließlich soll ja auch sichergestellt werden, dass sich ein Backend Service konform zur API-Definition verhält.

Genau solche Tests lassen sich mit Dredd *[8]* automatisieren, einem HTTP-API-Testing-Framework, das Tests gegen ein API ausführen kann. Die lokale Installation des Frameworks erfolgt über NPM. Im Anschluss daran können per „dredd init"-Kommando eine „dredd.yml" Datei erzeugt und darin die Test-Details für das entsprechende API definiert werden, etwa der HTTP-Endpunkt des zu testenden API. Output ist dann ein entsprechender Test-Report auf der Kommandozeile. Darüber hinaus können die Test-Ergebnisse auch in der Apiary Console eingesehen werden.

Build- und Delivery-Automatisierung

Als Build- und Delivery-Plattform kam bei der Webshop-Lösung Oracle Wercker *[9]* zum Einsatz, eine Automatisierungsplattform für den Build und die Bereitstellung von Microservice- und Container-basierten Applikationen. Um Wercker für den Build und das Deployment einer Container-basierten Applikation nutzen zu können, muss die Plattform mit dem GitHub-Repository der zu bauenden Applikation verbunden sein. Im Anschluss daran kann der Build und Delivery Workflow über eine Sequenz sogenannter „Pipelines" definiert werden.

Was innerhalb der einzelnen Pipelines passiert, wird in einer YAML-Datei definiert, der „wercker.yml", die im Git-Repository abzulegen ist. Es gibt einen Workflow, der zunächst den Container baut, diesen dann in einer Registry registriert und ihn im Anschluss einrichtet. Darüber hinaus können noch weitere Pipelines in den Workflow aufgenommen werden, zum Beispiel um die Dredd-Tests auszuführen, bevor der Service ausgerollt wird.

Fazit

Wie der Artikel zeigt, sind die Herausforderungen bei der Entwicklung von Microservices-basierten Applikationen mannigfaltig. Eine zentrale Rolle spielen APIs, um die externe Kommunikation abzubilden, sowie ein konsistenter DevOps-Ansatz. Die Vorgehensweise bei der Entwicklung ist ein entscheidender Faktor; der „API first"-Ansatz hilft dabei, die Entwicklung möglichst effizient zu gestalten.

Anhand einer beispielhaften Webshop-Entwicklung wurde skizziert, worauf es bei der Umsetzung moderner Applikationen in Bezug auf Microservices ankommt. Da die Teammitglieder über fünf verschiedene Länder verteilt waren, konnten wir zudem einen ersten Eindruck bezüglich der organisatorischen Aspekte bekommen, die mit einer Microservices-Implementierung einhergehen. Die Team-Kommunikation lief teilweise über Slack; im Laufe des Projekts, das sich über zwei Monate erstreckte, wurden mehr als dreitausend Nachrichten ausgetauscht! Ein Indikator dafür, wie groß der Abstimmungsbedarf selbst in einem solch kleinen Use Case sein kann.

Weitere Herausforderungen technischer Natur werden uns zukünftig noch weiter beschäftigen. Eine Frage, die intensiv diskutiert wurde, war zum Beispiel: „Wie gestalte ich eine Choreographie für die unterschiedlichen Microservices, um einen Bestellprozess abbilden zu können?" Gedanken hierzu finden sich unter *[10]*.

Quellen

[1] James Lewis, Martin Fowler: „Microservices – A Definition of this New Architectural Term", 2014: *https://www.martinfowler.com/articles/microservices.html*

[2] OMESA Group: „Open Modern Software Architecture Project", OMESA Website, 2017: *http://omesa.io*

[3] OMESA Group: „Capabilities Service Implementation", OMESA Website, 2017: *http://omesa.io/serviceimplementation*

[4] OMESA Group: „Capabilities API", OMESA Website, 2017: *http://omesa.io/apilayer*

[5] Sam Newman: „Pattern: Backends For Frontends", Blog des Autors, 11/2015: *https://samnewman.io/patterns/architectural/bff*

[6] Anne van Kesteren: „Cross-Origin Resource Sharing", W3C Recommendation, 1/2014: *https://www.w3.org/TR/cors*

[7] Apache Software Foundation: „Welcome to Apache Avro!", Projektdokumentation, 2012: *https://avro.apache.org*

[8] „Dredd – HTTP API Testing Framework", Projektdokumentation 5/18, *http://dredd.readthedocs.io/en/latest*

[9] Oracle + Wercker: „Increase developer velocity ...", Oracle 2018: *http://www.wercker.com*

[10] Luis Weir: „Is BPM Dead, Long Live Microservices?", Blog des Autors, 2/2018: *http://www.soa4u.co.uk/2018/02/is-bpm-dead-long-live-microservices.html*

Sven Bernhardt
sven.bernhardt@opitz-consulting.com

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- Benjamin Belghith
- Joachim Sand
- Matthias Runte
- Christopher Trumpf
- André Meier
- Jörg Doppelreiter
- Bastian Gdowiok
- Nikolaus Eichler
- Jerome Below
- Udo Scholz
- Matthias Reimann
- Stefan Kinnen
- Manfred Schweizer
- Walter Rausch
- Irmgard Henn
- Frank Preikschat

Termine

18.09.2018
SOUG DAY
Schweiz

20.09.2018
Regionaltreffen München/Südbayern
Andreas Ströbel
München

20.09.2018
Regionaltreffen Nürnberg/Franken
Martin Klier & Thomas Köppel
Nürnberg

20.09.2018
Vorstandssitzung
Telfs

10.10.2018
Regionaltreffen Berlin/Brandenburg
Michel Keemers & Mylène Diacquenod

11.10.2018
Regionaltreffen NRW
Martin Schmitter
Düsseldorf

12.10.2018
DOAG Datenbank Webinar
online

18.10.2018
Regionaltreffen Nürnberg/Franken
Martin Klier & Thomas Köppel
Nürnberg

18.10.2018
Regionaltreffen Stuttgart
Jens-Uwe Petersen & Anja Stollberg
Stuttgart

22.10.2018
Regionaltreffen München/Südbayern
Andreas Ströbel
München

25.10.2018
Regionaltreffen Karlsruhe
Reiner Bünger
Karlsruhe

25.10.2018
Regionaltreffen Dresden/Sachsen
Helmut Marten

09.11.2018
DOAG Datenbank Webinar
online

15.11.2018
Regionaltreffen Nürnberg/Franken
Martin Klier & Thomas Köppel
Nürnberg

20.11.2018
DOAG 2018 Konferenz + Ausstellung
Nürnberg

Impressum

Red Stack Magazin 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 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 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: Stefan Kinnen. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führen einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:
Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
Chefredakteur (ViSdP): Wolfgang Taschner
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Lisa Damerow, Mylène Diacquenod, Marina Fischer, Klaus-Michael Hatzinger, Sanela Lukavica, Yann Neuhaus, Fried Saacke

Fotonachweis:
Titel: © aimage/123RF
S. 12: © gmast3r/123RF
S. 22: © Maksim Kabakou/123RF
S. 27: © Docker
S. 34: Original: © Andriy Popov/123RF
S. 41: © Oracle
S. 46: © texelart/123RF
S. 50: © scanrail/123RF
S. 60: © ammentorp/123RF
S. 65: © alphaspirit/123RF
S. 68: © Aleksandr Velichko/123RF

Anzeigen:
Simone Fischer,
DOAG Dienstleistungen GmbH
(verantwortlich, Anschrift s.o.)
Kontakt: anzeigen@doag.org
Mediadaten und Preise unter:
www.doag.org/go/mediadaten

Druck:
adame Advertising and Media GmbH,
www.adame.de

Titel, Gestaltung und Satz:
Caroline Sengpiel,
DOAG Dienstleistungen GmbH
(Anschrift s.o.)

Inserentenverzeichnis

20. - 21. September 2018 in Dresden

DOAG BIG DATA Days

Daten, der Treibstoff der digitalen Gesellschaft

Bis **10. August** Frühbucher-Rabatt

BIG DATA

Informieren Sie sich über die Rolle der Oracle-Datenbank-Technologie und die Verarbeitung von großen Datenmengen durch spezielle Features der Datenbank-Version 18c.

VISUALISIERUNG/ REPORTING

Werkzeuge für die Visualisierung (Oracle Business Intelligence, APEX, ADF, JET) und das Reporting (Oracle BI Publisher, Reports) bieten vielfältige Möglichkeiten.

GEODATEN

Nutzen Sie die Möglichkeit, unter anderem an Talks aus den Themenbereichen Geoinformation in der Cloud und hybriden Umgebungen, Linked Open Geodata oder Location Intelligence teilzunehmen.

Weitere Informationen und Anmeldung unter:
www.doag.org/go/bigdatadays