

Red Stack

Magazin

DOAG

SOUG
swiss oracle
user group

AOUG
AUSTRIAN ORACLE USER GROUP

Jetzt inklusive BUSINESS NEWS

DATABASE



Aus der Praxis

Was mache ich als DBA bei der autonomen Datenbank noch?



Im Interview

Andre Lünsmann,
Barmenia
Versicherungen

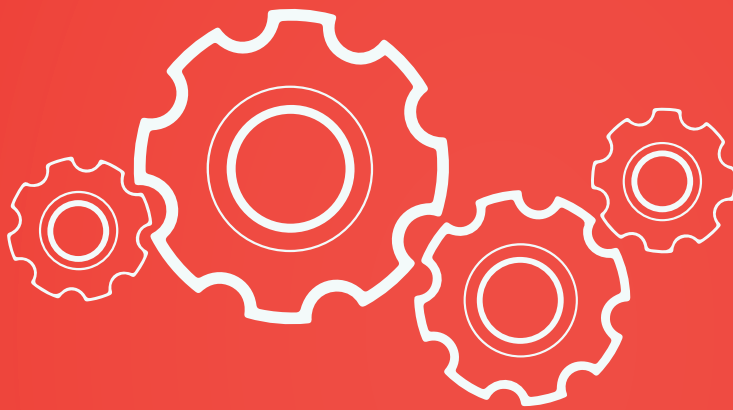
BUSINESS NEWS

Wie verändert
Blockchain
die Datenanalyse?

Werden Sie DOAG-Mitglied!

Ab 120 EUR/Jahr (zzgl. MwSt.)

„Gemeinsame Interessen gemeinsam vertreten“



+ 30 % Rabatt auf Veranstaltungen
+ Bezug der Zeitschriften

Red Stack Magazin inkl. Business News und Java aktuell

DOAG



Ingo Sobik-Weniger
Vorstand Next Generation,
Leitung Next Generation
Community

Liebe Mitglieder, liebe Leserinnen und Leser,

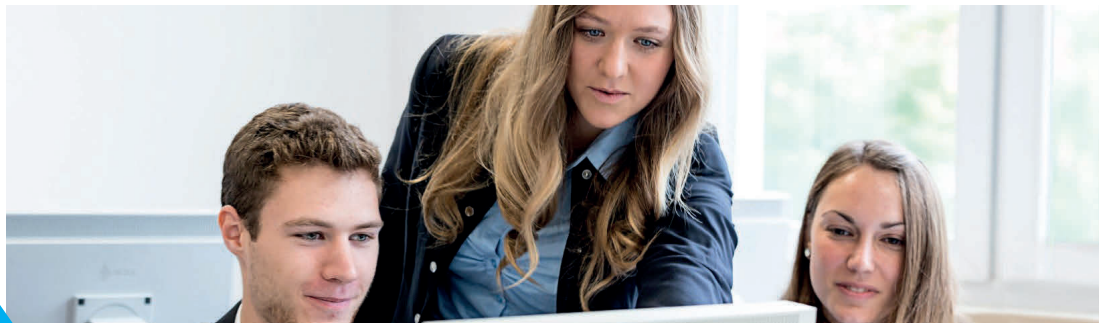
bei all den verrückten Ereignissen und Entwicklungen in unserer Welt, lernt man Beständigkeit wieder neu zu schätzen. Mit dieser Ausgabe des Red Stack Magazins greifen wir deshalb ein Steckenpferd der deutschsprachigen Oracle User Groups auf: Datenbanken. Wobei der Schein trügt. Denn natürlich bleibt auch bei diesem Thema die Welt nicht stehen. Im Gegenteil, während Oracle mit „autonomous“ hier auf ein völlig neues Konzept setzt, stellen sich viele DBAs die Frage, wie sich dadurch ihre Arbeit verändern wird. Lernen Sie deshalb in dieser Ausgabe, was auch mit „autonomous“ Bestand hat und wo sich neue Perspektiven öffnen. Wie auch zum Beispiel beim Thema Interoperabilität, die aufgrund immer heterogener werdender IT-Landschaften weiter an Bedeutung gewinnt.

Am Beispiel von PostgreSQL erhalten Sie gleich zwei spannende Ausblicke, wie das Zusammenspiel mit der Oracle-Datenbank zum Erfolg wird. Natürlich darf auch ein Blick in die neuesten Features von Oracle 20c nicht fehlen.

Die Business News ab Seite 74 widmet sich ganz dem Thema Blockchain aus der nicht-technischen Perspektive. Nach den großen Erwartungen an das neue technologische Paradigma und den Goldrausch erzeugenden Höhenflügen bei Kryptowährungen, ist das Thema zuletzt wieder aus dem Rampenlicht verschwunden. Der große Durchbruch blieb jedenfalls aus. Wie sich Szene und Technologie seither entwickelt haben und wo noch immer Potenzial liegt, haben wir für Sie zusammengetragen.

Wir freuen uns, Ihnen mit dieser Ausgabe ein prall gefülltes Magazin präsentieren zu dürfen und wünschen Ihnen viel Freude beim Lesen.

Ingo Sobik-Weniger



Training

Training

MUNIQSOFT

TRAINING

ORACLE
Silver Partner

20 Jahre Oracle-Datenbankschulungen von Experten, effizient und kundenorientiert!

Sie können an all unseren Schulungen auch **ONLINE** teilnehmen.

Im Livestream verfolgen Sie die gewünschten Kurse von zu Hause oder Ihrem Büro aus.

SQL I & Vertiefung	18.01.-22.01.2021	€1.880.-netto
SQL II Fortschritt	25.01.-28.01.2021	€1.790.-netto
DB Monitoring I	08.02.-12.02.2021	€1.990.-netto
PL/SQL I Februarspecial	22.02.-26.02.2021	€1.690.-netto

☎ 089 679090-40

Website: www.munisoft-training.de

Tipps: www.munisoft-training.de/tipps

Schulungszentrum

Munisoft Training GmbH

Grünwalder Weg 13a

82008 Unterhaching/München

Mehr Oracle Schulungstermine unter munisoft-training.de

Auf Anfrage bieten wir auch gerne individuelle Inhouse Schulung und Consultingleistungen an!



Interview mit Andre Lünsmann



Schlechteste Praktiken und ihre technische Schuld – Teil 2



Oracle ADW vs. Snowflake



Blockchain-Technologie in der Videospielebranche: Spielerei oder Innovationstreiber?

Einleitung

- 3 Editorial
- 5 Timeline
- 6 Aus der Ferne betrachtet: DOAG – Rückbesinnung auf das Wesentliche
- 7 „Als Fazit der letzten 10 Jahre würde ich konstatieren, dass der Ansatz, nur einen Technologiestack einzusetzen, nicht mehr zeitgemäß und auch nicht wirtschaftlich ist“
Interview mit Andre Lünsmann

Datenbank

- 10 Was mache ich als DBA bei der autonomen Datenbank noch?
Sinan Petrus Toma
- 16 Schlechteste Praktiken und ihre technische Schuld, Teil 2
Franck Pachot
- 20 Wie PostgreSQL von 20 Jahren Oracle Tuning profitieren kann
Hervé Schweitzer
- 25 Oracle Clusterware 19c ist zertifiziert für Oracle Linux 8 – ein Testbericht
Torsten Kaden
- 29 Oracle Database Indexing Best Practices, Teil 3
Randolf Geist
- 35 Getting data from Oracle to PostgreSQL and vice versa
Daniel Westermann
- 44 JSON Type in der Oracle-Datenbank und Autonomous JSON Database
Beda Hammerschmidt
- 48 The Song of Fire and Ice – Oracle ADW vs. Snowflake
Dr-Ing. Dipl.-Inform. Holger Friedrich

Cloud

- 53 Oracle Data Science Cloud Service
Alfred Schlaucher

APEX

- 58 Vulnerability-Management mit APEX
Christina Finck und Klaus Schuermann
- 64 Flows for APEX – Einfach Prozesse in APEX modellieren und ausführen
Niels de Bruijn
- 67 Retrospektive: Warum es sinnvoll ist, auch zurückzuschauen
Thimo Fußbroich

Data Warehouse

- 70 Data Warehouse und Business Intelligence im Mittelstand
Hans-Peter Weih

Wie verändert Blockchain die Datenanalyse?

BUSINESS NEWS

- 74 Analyse von Blockchain-Daten
Evgenia Julia Rosa und Detlef Egbert Schröder
- 81 „Die Blockchain zeigt ihre Schokoladenseite in der Logistik und Markierung von Waren und darin, die Integrität der Daten zu gewährleisten.“
Marcos López sprach mit Ekaterina Koschkarova
- 83 Public Blockchain Innovation – Wie die Evolution der Protokolle Branchen revolutioniert
Jonas Huck
- 85 Blockchain Gaming – Vom Spielplatz zur Revolution
Dennis Walz
- 91 Enterprise Blockchain am Beispiel eines internationalen Produktanforderungssystems
Tim Rüb
- 94 System Lifecycle Management – Auf dem Weg zur Digitalisierung des Engineerings, Teil 2
Martin Eigner
- 98 Das Blockchain-Manifest
Verfasst von Falk Wolsky

Intern

- 101 Neue Mitglieder + Termine
- 102 Impressum + Inserenten

Timeline

30. September 2020

Der SQL Developer steht im Fokus des DOAG Development Online-Events. Sabine Heimsath, selbstständige IT-Beraterin und Oracle ACE, gibt in ihrer Session viele hilfreiche Tipps und Tricks rund um den Oracle SQL Developer.

13. bis 14. Oktober 2020

Das Berliner Expertenseminar zum Thema „Stabil unterwegs - Automatisierte Tests in der Oracle-Datenbank“ mit Samuel Nitsche findet als Online-Event statt.

28. Oktober 2020

Das DOAG Development Online-Event zum Thema „Jenkins mit Oracle“ findet online statt. Moritz Reinwald (MT AG) beleuchtet die Zusammenarbeit mit Jenkins als Automatisierungswerkzeug für Oracle im Rahmen von DevOps. Dabei stellt er Best Practices und Oracle-spezifische Besonderheiten vor – von einfachen Jobs bis hin zu komplexen CI/CD-Pipelines.

13. November 2020

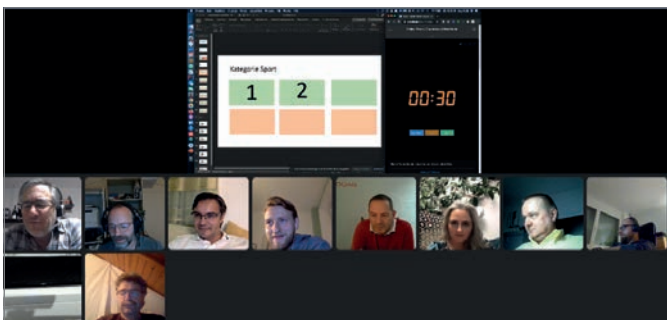
Das Datenbank Online-Event zum Thema „Oracle 12c R2 RAC Update nach 19c“ bietet einen Erfahrungsbericht zu einem Projekt. In seinem Vortrag berichtet Ralf Appelbaum von TEAM vom Ablauf des Updates sowohl der Grid-Infrastruktur als auch der Datenbanken von 12c Rel.2 auf 19c. Der Referent erläutert unter anderem auch, welche Hürden noch zu nehmen waren und wie das Ziel erreicht wurde.

17. bis 19. November 2020

Die 33. Konferenz + Ausstellung 2020 findet erstmals aufgrund des bundesweiten Teil-Lockdown während der Pandemie leider nur als moderierte Online-Konferenz statt.

Dennoch gelingt es, unseren Mitgliedern und Teilnehmern viele Gelegenheiten zum Erfahrungsaustausch und Networking zu geben. Rund 200 Vorträge, moderierte Expertengespräche und Workshops bieten umfangreiches Wissen. Neben den Nonstop-Sessions auf der virtuellen Hauptbühne gibt es Breakout-Varianten. In allen Vortragspausen finden zudem moderierte Expertengespräche statt.

Darüber hinaus laden Community Lounges zum virtuellen Treffen ein, bei denen spontan auch Vorträge und Workshops organisiert



Abendprogramm: Jeopardy (Quiz)

werden. Aussteller und Partner empfangen Besucher an virtuellen Ausstellungsständen zum Gespräch. Auch die DOAG ist mit einem Stand vertreten. Im Rahmen des Abendprogramms lädt die Development Community am ersten Tag zum "Apex Get Together" ein. Am Mittwochabend gibt es mit Jeopardy einen spannenden Spieleabend mit der Moderation der DOAG-Vorstände Armin Wildenberg und Ingo Sobik-Weniger und drei gegeneinander antretenden Teams. Das Team „Die DatenModellBauer“ geht vor den Teams „in-factory und „die Letzten werden die Ersten sein“ letzten Endes als Sieger hervor.

DOAG-Geschäftsführer Fried Saacke zieht nach drei Tagen ein positives Fazit und betont, dass die erste komplett virtuelle Jahreskonferenz der DOAG zunächst eine enorme Herausforderung und dann ein großer Erfolg war. „Dafür bedanke ich mich beim ganzen Team, allen engagierten Mitwirkenden aus dem Verein und den Mitarbeitern der Geschäftsstelle. Online ist in Zukunft ein Muss, am liebsten aber in Verbindung mit einer physisch vor Ort stattfindenden Konferenz. Hybride Veranstaltungen, die das Beste aus beiden Welten miteinander vereinen sind unser Ziel. Trotz des Erfolges wünsche ich mir, dass wir uns im nächsten Jahr wieder in Nürnberg sehen können und von dort live in die ganze Welt senden.“

26. November 2020

Das DOAG Development Online-Event mit Rainer Willems (Oracle) zum Thema „RESTful Services mit dem ORDS“ findet online statt. Mit dem ORDS lassen sich auch moderne REST-Schnittstellen für relationale Daten in der Datenbank oder auch für JSON-Dokumente in der Datenbank zur Verfügung stellen.

02. bis 03. Dezember 2020

Das Berliner Expertenseminar zum Thema „Java After Eight“ mit Nicolai Parlog findet statt. In diesem Kurs werden die Java-Versionen 9 bis 15 beleuchtet. Dabei liegt der Fokus auf neuen Sprachfeatures wie Sealed Classes, Records, Text Blocks, switch Expressions und var, aber auch neue und erweiterte APIs sowie JVM- und Performance-Verbesserungen werden theoretisch vorgestellt und mit praktischen Übungen untermauert.

11. Dezember 2020

Das DB Online-Event „Mit der richtigen Verbindung zum optimalen Betrieb: Ihre Datenbank in der Oracle Cloud (OCI)“ mit dem Cloud-Lösungs-Experten Frank Radde (Colt) findet statt. Behandelt werden die Herausforderungen bei der Migration einer Datenbank in die Oracle Cloud Infrastructure (OCI), die Auswahl der richtigen Netzwerkinfrastruktur, Trends zu Multi-Cloud-Implementierungen und Cloud-On-Demand-Lösungen.

18. Dezember 2020

Die DOAG schließt ihre Geschäftsstelle und verabschiedet sich in die Weihnachtsferien. Fröhliche Weihnachten und kommen Sie alle gut ins neue Jahr. Ab 4. Januar 2021 sind wir wieder für Sie da. Wir freuen uns auf die Zusammenarbeit in 2021 – es wird ein Jahr voller Herausforderungen, bleiben Sie gespannt und vor allem bleiben Sie gesund!

Aus der Ferne betrachtet: DOAG – Rückbesinnung auf das Wesentliche



*Dr. Dietmar Neugebauer
Ehemaliger DOAG-Vorstandsvorsitzender*

Ein neues Datenbank-Release steht vor der Tür. Inzwischen wohl mit der Bezeichnung 21c. Ich kann mich noch gut erinnern, mit welchem Marketingaufwand und großen Brimborium Oracle die neuen Versionen den Kunden früher vorgestellt hat. Das war immer ein Zeichen für neue technologische Errungenschaften. Und heute? Ist Ihnen schon etwas aufgefallen? Auch neue Modelle in der Automobilbranche oder neue Handy-Generationen erwecken nicht mehr unsere riesige Aufmerksamkeit. Liegt das nur an den Einschränkungen der Pandemie? Ich glaube, eher nicht. Unsere Schwerpunkte haben sich einfach geändert. Ein neues Widget auf dem Handy, ein neues Design und ein neuer Motor reißen uns jetzt nicht vom Hocker. Ob jetzt in der Datenbank der JSON-Datentyp existiert und die Lizenz drei PDBs beinhaltet – dafür dann der ganze Migrationsaufwand, brauche ich das wirklich? Sind meine Anforderungen heute nicht ganz andere?

Anwender erwarten heute, dass die Technologie so für sie zur Verfügung steht, wie sie gerade am Arbeitsplatz, in ihrem Projekt und auch in ihrer Mobilität benötigt wird. Dementsprechend wählen sie den Anbieter auf dem Markt aus. Die Technologie dazu muss nicht einmal mehr gekauft werden, sie lässt sich für bestimmte Zeit physisch mieten oder auch aus der Cloud benutzen. Das ist inzwischen vor allem bei der jüngeren Generation schnell selbstverständlich geworden. Eine Markenbindung zum Technologieprodukt gibt es nicht mehr. Das hat sich gewandelt, der Trend geht zum Anbieter von Services oder Mobilität.

Informationen und Bewertungen verbreiten sich dabei heute über die Medien wie Facebook, Twitter oder Instagram in kür-

zester Zeit auf dem weltweiten Wirtschaftsmarkt. Die DOAG hat sich sehr stark und überaus erfolgreich durch Wissensvermittlung auf Basis physischer Medien und Events ausgezeichnet. Dies ist natürlich in Corona-Zeiten so gut wie nicht mehr möglich und für reine Wissensvermittlung gibt es viel Content im Internet, der kostenfrei angeboten wird.

So muss die DOAG wieder mehr aus ihren Ursprüngen lernen. Erfahrungsaustausch unter den Anwendern war schon immer ein wichtiges Kriterium des Vereins. Wobei der Erfahrungsaustausch der Anwender heute über die reine Verwendung der Oracle-Produkte hinausgeht. Die Anwender wollen sich darüber austauschen, wie sie die besten Lösungen für ihre Herausforderungen am Markt finden und einsetzen können. Ein guter Schritt seitens der DOAG war, das Angebot online zu verstärken. Vielleicht könnte man dies, sobald es die Situation der Pandemie hergibt, auch mit kleinen, lokalen Clustern kombinieren, bei denen eine beschränkte Anzahl von Anwendern zu bestimmten Themen wieder physisch anwesend ist (vielleicht sogar in mehreren Clustern gleichzeitig) und weitere Teilnehmer online zugeschaltet werden können.

Ich bin überzeugt, trotz aller Digitalisierung, Automatisierung und künstlicher Intelligenz werden wir mit dem Abklingen der Pandemie wieder den persönlichen Austausch schätzen – vielleicht am Anfang einfach nur in kleineren Gruppen und mit ... **einer DOAG für Oracle-übergreifende Themen und einem Erfahrungsaustausch der Anwender auf virtueller und physischer Ebene.**



„Als Fazit der letzten 10 Jahre würde ich konstatieren, dass der Ansatz, nur einen Technologiestack einzusetzen, nicht mehr zeitgemäß und auch nicht wirtschaftlich ist“

Johannes Ahrends, DOAG-Themenverantwortlicher für die Oracle-Datenbank-Administration sowie Standard Edition, sprach mit Andre Lünsmann, Barmenia Versicherungen, über heterogene Datenbank-Landschaften.

Wie sieht Ihre heutige Datenbank-Landschaft aus?

Wir sind ein mittelständisches Versicherungsunternehmen mit Schwerpunkt in der privaten Krankenversicherung. Unsere Datenbank-Landschaft ist in den letzten 10 Jahren mit den Anforderungen des Marktes und agilen Entwicklungsmethoden sukzessive gewachsen.

Derzeit betreiben wir klassische, relationale Datenbanksysteme wie Oracle, Informix, MySQL und PostgreSQL sowie zunehmend auch MSSQL-Server. Darüber hinaus nutzen wir NoSQL-Datenbanken wie MongoDB und Netzwerk-hierarchische Datenbanken wie BerkeleyDB für Legacy-Anwendungen.

Der größte Teil unserer Anwendungslandschaft nutzt derzeit Oracle-Datenbanken, die wir im Rahmen unserer Mainframe-Ablösung seit 2009 einsetzen. Davor waren primär Netzwerk-hierarchische Datenbanken wie IIDS2 und DB2-Datenbanken im Einsatz.

Wie ist diese Datenbank-Landschaft entstanden?

Bis 2003 hatten wir primär für unsere Legacy-Anwendungen Netzwerk-hierarchische Datenbanken sowie Informix und MySQL im dezentralen Umfeld im Einsatz. Dann kamen übergangsweise DB2-Datenbanken im Rahmen des Einsatzes von

z/OS dazu. 2006/2007 entschieden wir uns, dass eine flächen-deckende Migration der Legacy (Cobol)- Anwendungen nach z/OS zu aufwendig und zu wenig gewinnbringend für das Unternehmen ist. So entstand der Plan für eine sukzessive Erneuerung der Anwendungslandschaft, mit dem gleichzeitig der Wechsel in das dezentrale Umfeld mit Java-Anwendungen eingeleitet wurde.

Die großen Bestands- und Leistungssysteme wurden gemeinsam mit einem Partner- Unternehmen neu entwickelt. Im Interesse einer zukunftsweisenden Ausrichtung wurde beschlossen, strategisch Java- und Oracle-Datenbanken für die neuen Systeme einzusetzen.

Im Rahmen der vorbereitenden Aktivitäten wurde zunächst ein Datenmigrationsprozess für alle Netzwerk-hierarchischen Datenbanken des Mainframe entwickelt, damit alle Daten in die dezentrale Welt migriert werden konnten. In diesem Schritt wurden MySQL- Datenbanken genutzt, da die Datentypen von MySQL besser mit den Netzwerk- hierarchischen Datenbanken korrespondierten.

2009 wurde die Unfallsparte dann als erste Anwendung in Java mit einer Oracle- Datenbank zum Einsatz gebracht, dabei musste zunächst die Kommunikation mit den zentralen Cobol- Anwendungen des Mainframe realisiert werden.



Zur Person: Andre Lünsmann

Andre Lünsmann ist seit seiner Ausbildung zum Versicherungskaufmann und der Weiterbildung zum Versicherungsfachwirt in den 80er-Jahren bei der Barmenia beschäftigt.

Er verantwortet seit über 10 Jahren die Bereiche Datenbanken und dezentrale Produktion bei den Barmenia Versicherungen. In über 30 Jahren hat er vom Entwickler über Architektur, Datenmodellierung bis hin zum Projektleiter und Teamleiter an einer Vielzahl von Themen in der IT gearbeitet und unter anderem auch die Ablösung des Mainframe durch eine virtualisierte Emulation eingeleitet. Strategie, Agilität und Change-Prozesse im Kontext der Mitarbeiterführung sind seine Stärken. Seit der Gründung im Jahre 2015 ist er Mitglied im DOAG-Anwenderbeirat und wurde 2020 zum Vorstand Infrastruktur & Middleware gewählt.

War trotz der strategischen Ausrichtung für Oracle auch MySQL gesetzt?

Ja. Schon früh stellte sich für uns die Frage, ob wir wirklich alle neuen Anwendungen unabhängig von ihren Anforderungen mit Oracle-Datenbanken betreiben wollten. Aufgrund der durchweg guten Erfahrungen mit MySQL im Rahmen der Datenmigrationen entschieden wir, dass Anwendungen mit geringeren Datenvolumen und geringerer Bedeutung für den Geschäftsbetrieb in MySQL betrieben werden sollten. Der Treiber waren hierfür die Ressourcen-schonende MySQL-Installation und die Lizenzfreiheit.

Wie hat sich die Oracle-Infrastruktur weiterentwickelt?

Nachdem wir 2011/2012 erste Anwendungen mit Oracle in die damalige Infrastruktur auf AIX-Maschinen in einem RAC-Cluster portiert hatten, wurde deutlich, dass wir für einen nennenswerten Anteil unserer Anwendungen die Vorteile eines RAC-Clusters nicht nutzen konnten, da immer wieder Anwendungen nicht clusterfähig waren. Andererseits verursachte die Pflege des RAC-Clusters nicht unerhebliche Aufwände in der Administration und die damals fünf großen Datenbanken bargen bei lastintensiven Verarbeitungen immer wieder das Problem, gleichzeitig alle Anwendungen in der Performance einzuschränken.

Wir begannen daher damit, Service-Level für Anwendungen zu definieren, zu denen wir auch konkrete Festlegungen für die Infrastruktur treffen konnten.

Passend zu den Themenbereichen „Verfügbarkeit“ und „(un) geplante Serviceausfälle“ legten wir auch Rahmenbedingungen für die Infrastruktur fest. So wurden beispielsweise zusätzliche Data-Guard-Instanzen eingerichtet, um bei den geschäftskritischen Anwendungen die Wiederherstellungszeit (RTO) einhalten zu können.

Wofür werden die weiteren DB-Systeme eingesetzt?

Informix-Datenbanken

haben wir schon in den 90er-Jahren für unsere Anwendungen für die Vertriebsschiene im dezentralen Umfeld eingesetzt. Auch heute ist dies noch ein wesentlicher Teil unsere Anwendungslandschaft. Mit der laufenden Migration auf virtuelle Linux-Maschinen modernisieren wir diesen alten Stack gerade.

BerkeleyDB

setzen wir seit 2012 für die komplette Mainframe-Infrastruktur ein, die wir auf eine emulierte Umgebung portiert haben. Bei dieser Portierung wurden für die Netzwerke hierarchische Datenbanken beschlossen.

MSSQL-Server

wurde im Windows-Umfeld in den letzten Jahren mit Kaufsoftware-Komponenten, die als Datenhaltung eine MSSQL-Server-Instanz benötigten, immer mehr eingesetzt.

MongoDB

wurde fast zeitgleich für das Bestandssystem der Lebensversicherung 2012 eingeführt. In diesem Zusammenhang wurde die NoSQL-Datenbank für die Speicherung der nicht mehr änderbaren Historienbestände genutzt. Ziel war es, die historischen

Zustände des Altsystems möglichst einfach zugreifbar in der neuen Anwendung zu machen, ohne dafür ein vollständiges relationales Modell designen zu müssen.

Wie lässt sich eine so vielfältige Datenbank-Landschaft managen?
Wir präferieren eine möglichst standardisierte Vorgehensweise unabhängig vom DB-System, jede Menge Automation und der Fokus liegt auf einer privaten Cloud-Strategie, um den Self-Service-Charakter zu unterstreichen.

Sowohl die emulierte Mainframe-Umgebung als auch die Mongo-Datenbanken wurden als virtuelle Maschinen mit automatisiertem Mechanismus in Puppet bereitgestellt.

Die virtuelle Infrastruktur mit darunterliegendem, synchronem Storage machte dabei durch die Möglichkeiten einer transparenten Verschiebung der virtuellen Maschine zwischen den Rechenzentren die Verfügbarkeit viel flexibler und einfacher. Wir entschieden uns daher auch, das Oracle RAC Cluster mit fünf großen Datenbanken durch knapp 200 virtuelle Maschinen mit logisch gruppierten Datenbanken zu ersetzen. 2016/2017 wurde das RAC-Cluster sukzessive abgelöst. Für den Install, das Patchen etc. gibt es für die Oracle-Instanzen nun automatisierte Prozesse. Die Bereitstellungszeiten und Downtimes haben sich so drastisch reduziert.

Auch die MSSQL-Server-Datenbanken laufen inzwischen entsprechend den Serviceklassen auf virtualisierten Maschinen, die aus Lizenzgründen (wie bei Oracle auch) auf separater Hardware laufen.

Wie wird sich das Datenbank-Umfeld Ihrer Meinung nach weiterentwickeln?

In den letzten Jahren nehmen nun auch die PostgreSQL-Instanzen zu, die ebenfalls öfter mit Kaufsoftware angefordert werden. Auch für PostgreSQL bauen wir derzeit eine virtuelle Infrastruktur mit automatisiertem Bereitstellungsprozess auf. PostgreSQL ist durch die von EDB angebotene Oracle-Kompatibilität auch ein ernstzunehmender Konkurrent für Oracle, den es sich zumindest unter dem Blickwinkel der Lizenzkosten zu betrachten lohnt.

Inzwischen verwenden wir für alle Datenbanksysteme die gleichen Ansätze, um Service-Level, Infrastruktur, Automation und Administration zu vereinheitlichen. Wir sehen das als zwingende Basis, um den Herausforderungen an Agilität, Flexibilität und Kostengesichtspunkten gerecht zu werden.

Als Fazit der letzten 10 Jahre würde ich konstatieren, dass der Ansatz, nur einen Technologiestack einzusetzen, nicht mehr zeitgemäß und auch nicht wirtschaftlich ist. In einer agilen DevOps-Welt mit Cloud-Ansätzen muss jede IT passende Lösungen in der Infrastruktur bereitstellen.

Das bedeutet auch, dass die Datenbank-Administration nicht mehr nach den einzelnen DB-Systemen getrennt werden sollte. Bei uns ist ein DBA gleichermaßen für Oracle, Informix, MySQL, MongoDB und MSSQL-Server verantwortlich. Natürlich hat jeder DBA da Stärken und Schwächen in dem einen oder anderen System, aber in der Rufbereitschaft muss er auch wissen, was zu tun ist, daher macht eine Spezialisierung keinen Sinn.

Für die Administratoren liegt eine große Herausforderung darin zu akzeptieren, dass die Zeit des händischen Konfigurierens mit der Comand Line nur noch für die Bereitstellung auto-

matisierter Prozesse zielführend und effizient genug ist. Gefragt sind Self-Service-Portale, wo man sich seine Datenbank selbst anfordern kann und die Bereitstellung in wenigen Minuten erledigt ist.

Die Herausforderungen an die Flexibilität der Beteiligten sind daher mindestens ebenso groß wie die der eingesetzten Infrastrukturen.

Was wünschen Sie sich von den Datenbank-Herstellern?

Im Oracle-Umfeld migrieren wir gerade auf 19c (Multitenant) und müssen feststellen, dass dort noch vieles nicht so funktioniert, wie man das aufgrund der schon länger verfügbaren Multitenant-Datenbanken erwarten würde. Wir müssen auch konstatieren, dass wir mit unseren eigenen Automatismen immer noch deutlich besser sind als das, was von Oracle hier auf Basis dieser „Standardisierung“ so angeboten wird. Wenn Oracle sich wirklich in Richtung autonome Datenbank bewegt, ist das eindeutig noch zu wenig.

Hier fehlen Service-Portale und Assistenten, die es dem Kunden erlauben, mit wenig Aufwand eine Individualisierung für sein Umfeld vorzunehmen.

In einer modernen Infrastruktur sind Virtualisierung/Containerisierung, Service-Orientierung und planbare Kostenentwicklungen unverzichtbare Bestandteile.

Für Oracle haben wir uns 2010 entschieden, da wir von einem Hersteller Java, den Application Server (WebLogic) und die Datenbank beziehen und mit Support versorgen konnten. Gerade das Thema Virtualisierung ist aus Lizenzgründen aber kritisch zu bewerten. Es kostet uns inzwischen monatliche Meetings mit dem Oracle-Vertrieb, um ein tragfähiges Vertragskonstrukt abzustimmen, obwohl wir schon ULA-Kunde mit einer Virtualisierungsvereinbarung sind. Die Vereinbarungen sind auch nicht so gestaltet, dass man als Kunde sich beruhigt zurücklehnt und sich sicher für die nächsten Jahre fühlt.

Im Kontext Support ist es leider auch seit Jahren so, dass man nur mit viel Mühe die Schallmauer des 1st-Level-Supports durchbrechen kann, damit man endlich einen fachkundigen Mitarbeiter aus dem Labor involviert, der das dokumentierte Problem auch versteht und lösen kann.

Zusammengefasst wünsche ich mir mehr Kundennähe bei Supportanfragen und Lizenz-Verhandlungen. Ich möchte das Gefühl haben, dass der Hersteller mich als Vertragspartner schätzt und mich nicht mehr oder weniger nötigt, Vereinbarungen zu treffen, die nicht wirklich unseren Bedarf abdecken und dabei unsere europäischen Rechtsgrundlagen nicht ausreichend berücksichtigen.

Es überrascht mich daher nicht, dass der Trend in Richtung Open Source ungebrochen ist.



Was mache ich als DBA bei der autonomen Datenbank noch?

Sinan Petrus Toma, Oracle Deutschland

„Ich bin Oracle DBA und unser Unternehmen stellt alle unsere Oracle-Datenbanken auf die autonome Datenbank um! Was mache ich denn nun?“ Administrative Abläufe werden in der autonomen Datenbank vollautomatisiert durchgeführt. Sie beinhalten sowohl täglich wiederkehrende Aufgaben wie das Ändern der Parameter oder das Erweitern von Tablespaces als auch zeitintensive Themen wie Patching und Upgrade. Backups werden automatisch erstellt. Hochverfügbarkeit und Disaster Recovery sind im Hintergrund implementiert. Auch die darauf basierende Exadata-Infrastruktur ist voll automatisiert. Daher stellen sich die Fragen „Für welche Aufgaben bin ich als DBA bei der autonomen Datenbank noch zuständig?“ und – bei so viel Automatisierung – „Womit verbringe ich die ersparte Zeit?“ sowie „Ergeben sich durch die neue Technologie neue Aufgaben und Verantwortlichkeiten?“ Im folgenden Artikel werden diese Fragen beantwortet. Sie werden feststellen, dass DBAs auch zukünftig noch immer gefordert sind, sich mit der Datenbank auseinanderzusetzen. Vor allem aber auch, wenn sie Veränderungen offen gegenüberstehen, sie als Chance sehen und gerne über den Tellerrand schauen.

Um ein gemeinsames Verständnis und eine Diskussionsgrundlage zu schaffen, werfen wir zunächst einen Blick auf die Aufgaben, die vollständig automatisiert und ohne jegliche Interaktion in der autonomen Datenbank ablaufen:

- Tablespaces werden automatisch erstellt und Datendateien erweitert.
- Full, Incremental und Archive Log Backups werden automatisch erstellt.
- Alle Daten werden mit Transparent Data Encryption (TDE) verschlüsselt.
- Backups und SQL*Net-Verbindungen sind standardmäßig verschlüsselt.
- Release Updates und Sicherheits-Patches werden im Hintergrund einge spielt.
- Major-Release Upgrades sind automatisiert.
- Unified Auditing und Audit-Policies sind standardmäßig aktiviert.
- Der Betrieb der darunter liegenden Exadata-Infrastruktur ist vollständig automatisiert.

Des Weiteren gibt es einige Restriktionen, um die Sicherheit und die Leistung der Datenbank zu gewährleisten:

- Es gibt keinen Zugriff auf die Hardware und keinen Zugang zum Betriebssystem.
- Benutzer mit SYSDBA-Privileg können nicht verwendet werden.
- Einschränkungen bei SQL-Statements bezüglich Tablespaces und Key Management.

Spätestens jetzt wird deutlich, dass die oben gestellten Fragen mehr als berechtigt sind und dass sich DBAs durchaus Sorgen und Gedanken machen sollten. Die gute Nachricht ist, die Arbeit wird ihnen trotzdem nicht ausgehen. Sehen wir uns im Folgenden einige Tätigkeitsbereiche eines DBA an.

Datenbank bereitstellen

Die Bereitstellung der autonomen Datenbank erfolgt mit sehr wenigen Klicks. Der DBA muss jedoch im Vorfeld einige Details kennen, um die dafür nötigen Entscheidungen zu treffen. Benötigt die Applikation ein Datawarehouse oder eine transaktionsverarbeitende Datenbank?

Diese unterscheiden sich beispielsweise hinsichtlich des defaultmäßig eingestellten Datenspeicherformats und der Größe der SGA und PGA, die für die entsprechenden Workload-Typen optimiert sind. Die Datenbank kann auf einer gemeinsam genutzten oder dedizierten Exadata-Infrastruktur in der Cloud bereitgestellt werden. Der Zugriff auf die Datenbank kann von überall oder nur von einem privaten Netzwerk gewährt werden. Hierzu ist ein Mindestmaß an Wissen über die geplante virtuelle Netzwerkkonfiguration notwendig. Der DBA muss sich über die Optionen und Unterschiede in der Netzwerkkonfiguration im Klaren sein, um die Applikation bei der richtigen Wahl der Datenbank zu unterstützen.

Benutzer und Rechte verwalten

Die Benutzer sowie deren Rollen und Zugriffsrechte zu verwalten, bleibt weiterhin vollständig in der Hoheit der DBAs. Ob es um Benutzererstellung, das Löschen und Entsperren von Benutzerkonten geht oder ob es sich um die Zuordnung von Rollen und Rechtevergabe handelt – es ist ein manueller Zugriff notwendig. In diesem Bereich bleibt alles beim Alten, außer dass das SYSDBA-Privileg nicht mehr vergeben und somit auch nicht verwendet werden kann.

Verbindung zur Datenbank aufbauen

Damit Benutzer und Anwendungen eine Verbindung zur autonomen Datenbank herstellen, müssen sie die Client-Zugangsdaten in einer ZIP-Datei herunterladen. Diese beinhaltet die sqlnet.ora-, tnsnames.ora- sowie die Wallet- und Keystore-Dateien.

Wenn Sie einen Blick auf die tnsnames.ora-Dateien werfen, werden Sie die vor-

definierten SQL*Net-Services vorfinden. Diese unterscheiden sich bezüglich Resource Management Plan Shares, Parallelisierung und der Anzahl der konkurrierenden SQL-Abfragen (siehe Tabelle 1).

Die Resource Management Plan Shares können vom DBA angepasst werden. Weiterhin können vom DBA pro Service Grenzwerte bezüglich Laufzeit und I/O definiert werden, um eine lang laufende Abfrage zu beenden und Ressourcen freizugeben. Anhand dieser Daten ist es Aufgabe des DBA, die Anwender bei der Auswahl des geeigneten Service zu beraten.

Datenbankzugriffe kontrollieren

Es liegt in der Verantwortung der DBAs, das Wallet an einem sicheren Ort zu bewahren und nur mit berechtigten Personen zu teilen. Das Wallet kann bei Bedarf rotiert werden. Ein Rotieren kann auch in gewissen Zeitabständen erfolgen, um bestimmte Sicherheitsrichtlinien zu erfüllen. Hierzu können die sogenannten Access Control Lists (ACLs) verwendet werden, um den Zugriff auf die Datenbank nur von bestimmten Client-IPs oder IP-Bereichen zu erlauben. Verbindungen aus allen anderen IP-Adressen, die nicht in der ACL eingetragen sind, werden nicht zugelassen. Ein Beispiel zeigt *Abbildung 1*.

CPU und Speicher skalieren

Wenn die Benutzeranzahl oder die Datenbankgröße wächst, können mehr Ressourcen allokiert werden. Da Tablespaces in der autonomen Datenbank automatisch verwaltet werden, muss nur die Gesamtmenge an nötigem Speicher angegeben werden, zum Beispiel 2 anstatt 1 TB. Diese Operation erfolgt online. Die CPU-Anzahl kann unabhängig

Service Name	Resource Management Plan Shares	Parallelisierung	Konkurrierende SQL-Abfragen
TP URGENT	12	Manuel	300x CPU-Anzahl
TP	8	1	300x CPU-Anzahl
HIGH	4	Anzahl CPUs	3
MEDIUM	2	Max. 4	1,25x CPU-Anzahl
LOW	1	1	300x CPU-Anzahl

Tabelle 1: SQL*Net-Services im Vergleich

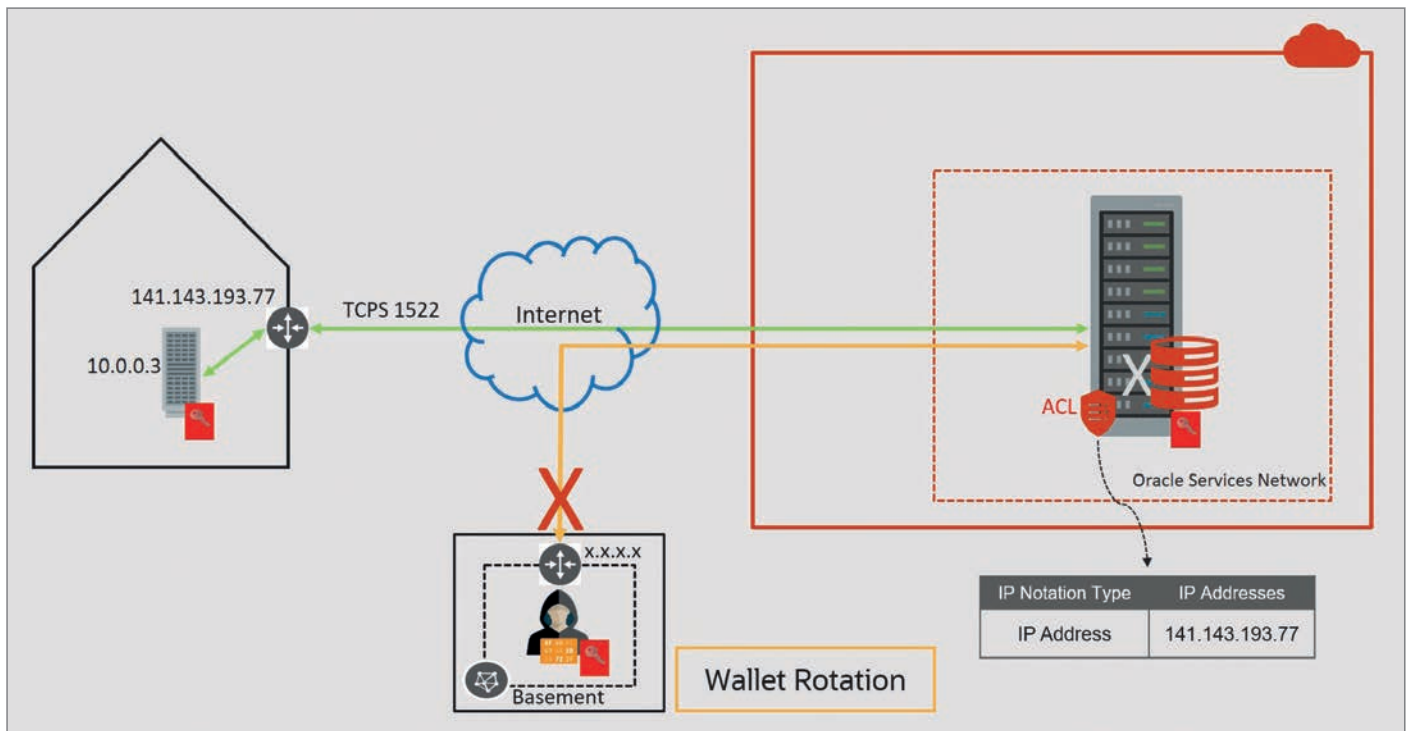


Abbildung 1: Verbindung zur autonomen Datenbank (Quelle: Sinan Petrus Toma)

davon ebenso ohne Service-Unterbrechung hoch- beziehungsweise herunter-skaliert werden. Falls erwünscht, kann die „Auto Scale“-Funktion aktiviert werden, um die CPU-Ressourcen bis auf das Dreifache automatisch je nach Auslastung zu skalieren.

Ressourcenverbrauch überwachen

Die Cloud-Konsole bietet eine Übersicht über den historischen und aktuellen CPU- und Speicherverbrauch. Hier können sich DBAs sehr schnell einen Überblick über die Datenbankaktivitäten, aktive Sessions und SQL-Abfragen verschaffen. Diese Daten bilden die Entscheidungsgrundlage, ob, wann und wie viele Ressourcen allokiert werden sollen. Lang laufende Abfragen lassen sich im Bereich „Service Konsole“ auflisten. Diese kann der DBA gemeinsam mit den Applikationsverantwortlichen überprüfen und gegebenenfalls optimieren, denn eine optimierte Abfrage kann mehr bewirken als das Hinzufügen zusätzlicher CPUs. ASH und AWR Reports können von der „Performance Hub“-Seite heruntergeladen werden.

Wenn Sie für Überwachung und Analysezwecke mehr Details aus dem Alert-Log oder den Trace-Dateien benötigen, ist

dies auch ohne Zugriff auf das Betriebssystem möglich. Dazu nutzen Sie die folgenden v\$-Views:

- v\$diag_alert_ext, um die Meldungen im Alert-Log anzuzeigen,
- v\$diag_trace_file, um die Trace-Dateien aufzulisten,
- v\$diag_trace_file_contents, um den Inhalt eines Trace-Files anzuzeigen.

Hier handelt es sich um die gleichen Views, die Sie aus Ihrer täglichen Arbeit möglicherweise bereits kennen.

Manuelles Backup erstellen

Ein wöchentliches Level-0- und ein tägliches Incremental-Level-1-Backup werden automatisch erstellt. Jedoch erhalten DBAs nicht selten eine Anfrage von der Anwendungsbetreuung, ein Full-Backup zu einem bestimmten Zeitpunkt – meist vor einem großen Deployment – zu erstellen. Dieses wird im Fall der Fälle für die Wiederherstellung verwendet, um auf das Einspielen von Level-1-Backups zu verzichten und damit die Wiederherstellungszeit zu beschleunigen. Um manuelle Backups von der autonomen Datenbank zu erstellen, muss ein separates Bucket im Object Storage erstellt

werden. Die Datenbank muss entsprechend konfiguriert werden, um den Zugriff auf das Object Storage zu erlauben. Hierfür benötigt der DBA neue Kenntnisse über Cloud Object Storage sowie das neue DBMS_CLOUD PL/SQL Package und seine Prozeduren.

Datenbank wiederherstellen

Soll die Datenbank zu einem früheren Zeitpunkt wiederhergestellt werden, kann dies menügeführt über die Auswahl eines bestimmten Backups oder eines bestimmten Zeitpunkts erfolgen. Dies kann der DBA mit wenigen Klicks über die Konsole starten. Der komplette Wiederherstellungsprozess ist jedoch automatisiert.

Der DBA kann die Flashback-Technologien auch dazu nutzen, um eine einzelne Tabelle oder Transaktion wiederherzustellen. Hierfür sind weiterhin eigens geschriebene SQL-Statements notwendig. Folgende Flashback Features sind in der autonomen Datenbank vorhanden:

- Flashback Drop
- Flashback Query
- Flashback Table
- Flashback Transaction Query
- Flashback Version Query

Datenbank klonen

Je nach Bedarf für Test- und Entwicklungszwecke kann der DBA die autonome Datenbank (die PDB) online klonen und damit eine neue Datenbank bereitstellen. Es können nur die Metadaten oder ein vollständiger Klon erstellt werden. Der Klon kann zum aktuellen oder aus einem früheren Backup erstellt werden. Ein Klon von einer früheren auf die aktuelle Datenbankversion ist ebenfalls möglich. Dieses Feature ist sehr nützlich und empfehlenswert, um eine neue Datenbankversion zu testen und somit die Applikationskompatibilität zu gewährleisten, bevor das tatsächliche Upgrade der Produktionsdatenbank durchgeführt wird.

Daten laden und migrieren

Zur Migration von einer On-Premises-Datenbank in eine autonome Datenbank kann Oracle Golden Gate für eine Echtzeitreplikation verwendet werden. Darüber hinaus kann auch weiterhin das traditionelle Data Pump Export und Import genutzt werden (siehe Abbildung 2). Die Dump-Dateien müssen hierfür in den Object Storage in der Cloud hochgeladen werden. Der Zugriff auf das Object Storage von der autonomen Datenbank muss ebenso erlaubt werden.

Um Daten aus einer autonomen Datenbank zu exportieren, können die Dump-Dateien auf der lokalen Exadata gespeichert werden und mittels DBMS_CLOUD PL/SQL Package auf das Cloud Object Storage kopiert oder direkt ins Object Storage exportiert werden. Von dort können die Dumps für Archivierungszwe-

cke, den Import in eine On-Premises- oder in eine weitere autonome Datenbank verwendet werden. Zudem können Sie mittels Datenbank-Link komplett auf den Zwischenspeicher verzichten und Data Pump mit dem Network Mode nutzen.

Um Datenbankmigrationen einfacher zu gestalten und den Export-Import-Prozess weiter zu automatisieren, bietet Oracle das kostenfreie Tool MV2ADB „move data to Autonomous Database in one-click“, das man von My Oracle Support unter der Doc ID 2463574.1 herunterladen kann.

Vor einer Migration kann das Schema in der Source-Datenbank überprüft werden, um gegebenenfalls Anpassungen für die autonome Datenbank vorzunehmen. Hierzu nutzen Sie das Tool „Autonomous Database Schema Advisor (von My Oracle Support Doc ID 2462677.1)“, das Ihnen einen Text-Report über die Objekte, die nicht oder mit Anpassungen migriert werden können, erstellt, inklusive Best-Practices, Empfehlungen und Anleitungen.

Datenbanksicherheit implementieren

Wenn es um die Cloud geht, spielt Sicherheit eine besonders große Rolle und ist eines der meist diskutierten Themen, wenn nicht sogar DAS Thema Nummer 1. Wie trägt die autonome Datenbank zur Sicherheit bei?

- Daten-, Backup-, und Netzwerkverschlüsselung sind standardmäßig aktiviert und können NICHT deaktiviert werden.
- Zugriff auf das Betriebssystem ist nicht möglich und das SYSDBA-Privileg kann nicht vergeben werden.

- Sicherheits-Patches werden automatisch eingespielt.
- Unified Auditing ist aktiviert und einige Standard-Policies sind vordefiniert.

Was kann der DBA in punkto Sicherheit noch tun? Die Antwort ist: sehr viel! Denn Sicherheit ist eine gemeinsame Verantwortung. Folgende Liste gibt einen Überblick über weitere Aufgabengebiete, um die Sicherheit zu erhöhen:

- Verwaltung der Wallets und der Netzwerkzugriffslisten.
- Verwendung von Database Vault, um den Zugriff bestimmter Benutzergruppen auf Applikationsdaten zu unterbinden.
- Definieren von Data Redaction Policies, um sensible Daten zur Laufzeit zu ersetzen.
- Definieren weiterer Unified Auditing Policies, um etwa Lesezugriffe mittels Data Pump Export zu auditieren.
- Nutzung von Data Safe, um die Sicherheitsrichtlinien zu überwachen und sensible Daten zu maskieren.

Data Safe ist ein webbasiertes Kontrollzentrum, um Sicherheits- und Benutzer-Assessments-Reports zu erstellen, ähnlich wie das DBSAT, jedoch mit viel mehr Funktionalität (siehe Tabelle 2).

Noch mehr automatisieren

Alle Interaktionen mit der autonomen Datenbank über die webbasierte Cloud-Konsole lassen sich ebenso per REST-API aufrufen oder über die OCI-CLI-Kommandozeile realisieren. Dadurch ergibt sich

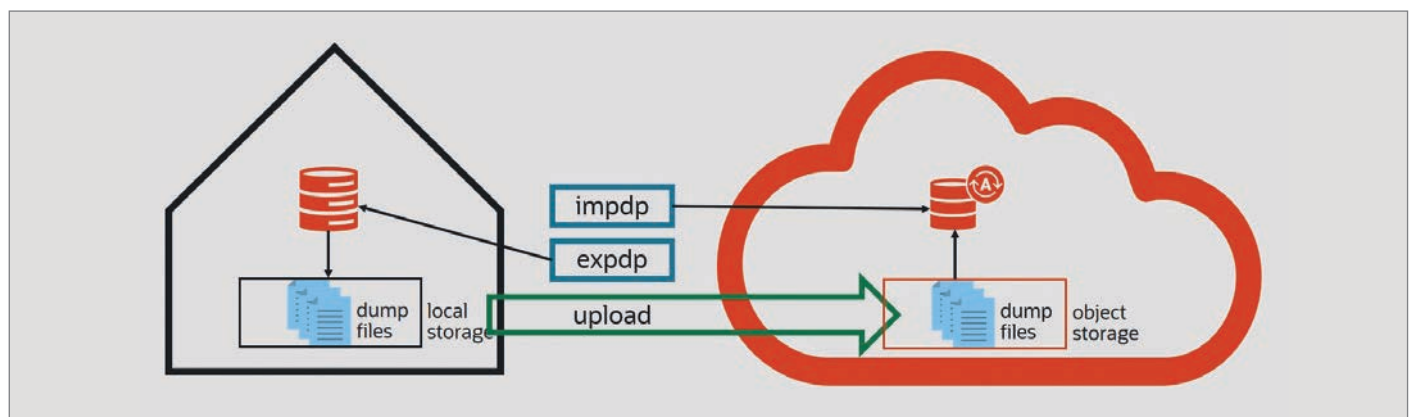


Abbildung 2: Data Pump Export und Import (Quelle: Sinan Petrus Toma)

die Möglichkeit, noch mehr Aufgaben zu automatisieren. Sie können vor einem nächtlichen Job, der mehr Ressourcen benötigt, programmatisch die CPU-Anzahl unmittelbar vor dem Jobstart hoch- und direkt nach Jobende herunterskalieren.

Test- und Entwicklungsdatenbanken, die nur während der regulären Arbeitszeit verwendet werden, können ebenfalls automatisch heruntergefahren und am nächsten Arbeitstag wieder gestartet werden. Die entsprechenden CLI-Kommandos sehen wie folgt aus:

```
oci db autonomous-database stop
--autonomous-database-id $ADB_OCID
oci db autonomous-database start
--autonomous-database-id $ADB_OCID
```

Mit wenig Bash-Programmierung schreiben Sie eine for-Schleife, führen diese Befehle für alle autonomen Datenbanken aus und sparen dabei, denn bei heruntergefahrener Datenbank werden keinerlei CPU-Kosten berechnet.

Planmäßige Data Pump Exports lassen sich mit sehr wenigen Codezeilen automatisieren, im Object Storage archivieren und bei Bedarf automatisch in eine weitere OCI-Region kopieren.

Hierzu können mit Terraform als „Infrastructure as Code“ die Datenbanken und weitere Cloud Resources mit einem Knopfdruck provisioniert werden. Damit wird sichergestellt, dass immer die gleiche und korrekte Konfiguration verwendet wird und damit manuelle Fehler weiter minimiert werden.

	DBSAT	Data Safe
Security Assessment	Ja	Ja
User Risk Assessment and Drill-Down to Audit Records	Nein	Ja
User Activity Auditing and Reporting	Nein	Ja
Sensitive Data Discovery	Limitiert	Ja
Data Masking	Nein	Ja
Multiple Targets	Nein	Ja
Modern GUI	Nein	Ja
Reports History	Nein	Ja

Tabelle 2: DBSAT und Data Safe im Vergleich

Über den Tellerrand schauen

Die autonome Datenbank auf dedizierter Exadata-Infrastruktur wird im eigenen privaten Netzwerk bereitgestellt. Eine autonome Datenbank auf gemeinsam genutzter Exadata-Infrastruktur kann über Private Endpoints und Network Security Groups (kurz NSGs) so konfiguriert werden, dass sie lediglich über eine private IP verfügt und aus einem privaten Netzwerk zugänglich ist. Die Applikation wird typischerweise ebenso in einem privaten Subnetz platziert und ist nur über einen Load-Balancer in einem Public-Subnetz zugänglich, wie in *Abbildung 3* zu sehen ist.

Damit der DBA die Netzwerkkonfiguration der Datenbank vornehmen und die Zugriffskontrolle implementieren kann, ist ein Verständnis der kompletten Architektur unabdingbar. Wenn der DBA sich

mit der Architektur beschäftigen muss, warum kann er dann nicht auch derjenige sein, der die Architektur erstellt und diese verantwortet? Damit wären wir bei der ersten Aufgabe, die über den Tellerrand hinausgeht: Cloud-Architekt.

Des Weiteren können DBAs mit der ersparten Zeit durch die Automatisierung die Applikation bei der Datenbankmodellierung unterstützen. Wer sonst mit bereits erworbenem Datenbankwissen wäre geeigneter dafür?

Wenn Sie sich für Applikationsentwicklung interessieren, können Sie mit der autonomen Datenbank sofort mit der APEX-Entwicklung starten, denn APEX ist bereits in der autonomen Datenbank konfiguriert und im wahrsten Sinne des Wortes „nur einen Klick entfernt“. Außerdem bieten die autonome Datenbank und die Cloud weitere Tools wie Oracle Analytics Desktop und Oracle Machine

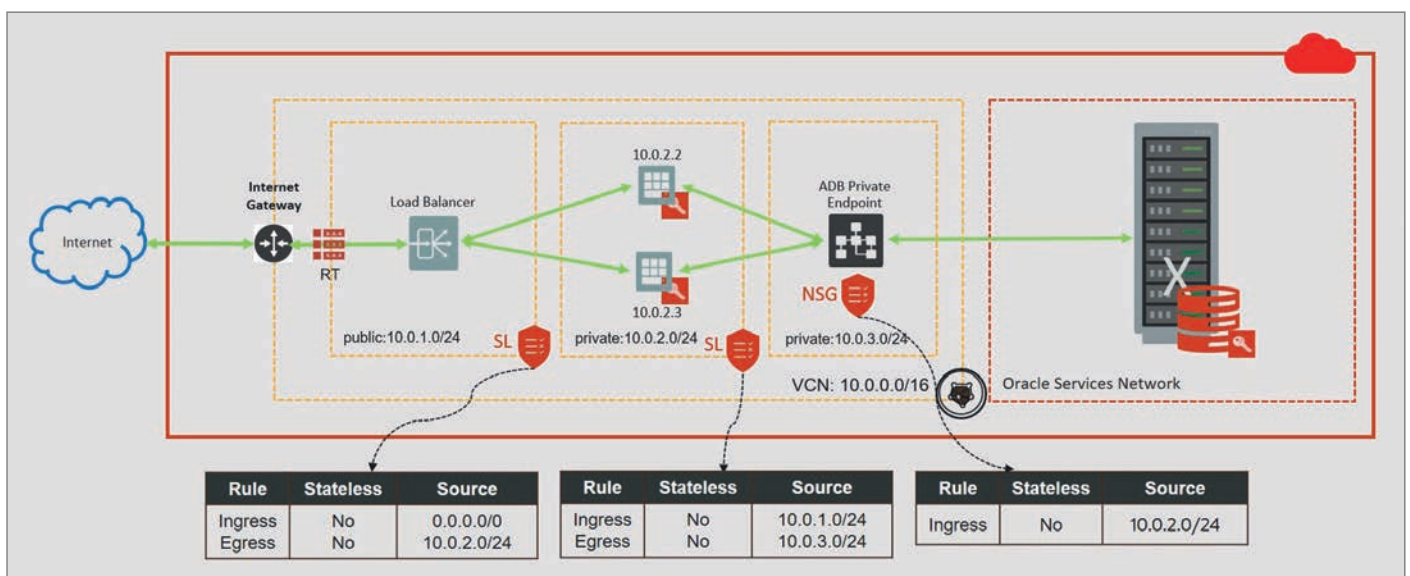


Abbildung 3: Autonome Datenbank mit Private Endpoint (Quelle: Sinan Petrus Toma)

Learning Notebooks, um ohne weiteren Aufwand Daten zu analysieren und ML-Algorithmen zu implementieren.

Zusammenfassung

Die Verantwortlichkeiten der DBAs verschieben sich von zeitaufwendigen, operativen Aufgaben zu übergeordneten und geschäftsbezogenen Prioritäten. Beispiele dafür sind: die Gewährleistung der Datensicherheit, der Schutz vertraulicher Informationen und die Unterstützung der Applikationsentwickler bei der Datenmodellierung und Leistungsoptimierung.

Dazu müssen DBAs neues Wissen erwerben, eine flexible und anpassungsfähige Denkweise haben und vor allem

offen und begeisterungsfähig für neue Technologien und Möglichkeiten sein.

Im deutschsprachigen Datenbank & Cloud-Technologie-Blog in der Kategorie Autonomous DB (siehe blogs.oracle.com/coretec/autonomous-db) finden Sie viele interessante Artikel und nützliche Tipps & Tricks zum Arbeiten mit der autonomen Datenbank.

Über den Autor

Sinan Petrus Toma begeistert sich für Datenbank- und Cloud-Technologien. Als Solution Engineer und Cloud-Architekt bei Oracle unterstützt er Unternehmen in Deutschland und Europa bei deren Transformation in die Cloud, insbesondere im Datenbank-Bereich.

Als ehemaliger Oracle DBA berichtet er von seinen eigenen Erfahrungen, den Unterschieden und den Vorteilen der Cloud und der autonomen Datenbank.



Sinan Petrus Toma
sinan.petrus.toma@oracle.com



MUNIQSOFT
CONSULTING



Performance-Tuning mit IQ

Mehr Power für Ihre Oracle Lösungen!

Nutzen Sie unseren proaktiven Datenbank-Healthcheck als Startschuss für die Optimierung Ihrer Oracle Datenbanken.

Ungebremst ans Ziel mit der Muniqsoft Consulting GmbH
www.muniqsoft-consulting.de

ORACLE Gold Partner

Specialized
Oracle Database



Jetzt Beratungstermin vereinbaren:
+49 89 62286789-39



Schlechteste Praktiken und ihre technische Schuld, Teil 2

Franck Pachot, dbi services

Häufig werde ich nach „besten Praktiken“ gefragt – eine Frage, die mir unangenehm ist, da die Antwort in der Regel vom Kontext abhängt. Ich kann natürlich verschiedene „Standardpraktiken“ empfehlen, die sinnvoll sind, wenn es keinen besonderen Grund gibt, anders vorzugehen. Meist geht es dem Fragesteller jedoch um einfache, allgemeingültige Regeln, die sich immer und überall anwenden lassen. Wenn es um generelle Empfehlungen geht, zähle ich lieber Dinge auf, die unter allen Umständen vermieden werden sollten. An dieser Stelle werde ich daher auf einige der „schlechtesten Praktiken“ eingehen. In meiner Tätigkeit als Berater sind mir Umgebungen mit erheblicher technischer Schuld begegnet. In der Vergangenheit auf die Schnelle getroffene Fehlentscheidungen, sei es aus Unkenntnis oder einfach aus Faulheit, zeigen in der Gegenwart oft unangenehme Folgen oder gefährden durch unnötige Komplexität oder schwierige Wartung und Weiterentwicklung zukünftige Projekte. Die Aufstellung erhebt keinen Anspruch auf Vollständigkeit, jeder einzelne Punkt zieht meiner Erfahrung nach jedoch problematische Konsequenzen nach sich.

In Teil 1 des Artikels widmeten wir uns vielen schlechten Praktiken wie beispielsweise fehlenden Informationen im DB-Namen. Im Folgenden zweiten (und letzten) Teil dieses Artikels fahren wir mit den schlechten Praktiken fort – wieder mit Namenskonventionen: Zuviel Information ist auch ein Problem.

Überzogene Namenskonventionen

Wie gesagt sind Namenskonventionen ein sehr guter Ersatz für die Dokumentierung von unveränderlichen Eigenschaften. Allerdings können sie sich als schlecht erweisen, wenn Sie darüber hinausgehende Informationen einbringen wollen.

Ich habe zum Beispiel Datenbanken gesehen, bei denen die Version in den Namen aufgenommen wurde, also DB12 für 12c. Irgendwann braucht diese Datenbank ein Upgrade und dann wollen Sie sie nicht umbenennen. Außerdem enthält dieser Zusatz keinen echten Informationswert, da die Version der Datenbank sehr einfach abgefragt werden kann. Gleiches gilt für den Anhang „_STANDBY“ für Standby-Datenbanken.

Irgendwann stellen Sie sie um und die Datenbank übernimmt eine andere Rolle. Sie können den geografischen Standort hinzufügen, beispielsweise „GE“ für Genf, wenn Sie sicher sind, dass beim Transfer zu einem anderen Datacenter auf jeden Fall eine neue Standby-Datenbank hinzugefügt wird und nicht die Server umgezogen werden. Verzichten Sie jedoch darauf, den Namen des Racks hinzuzufügen, in den die Server eingebaut sind, oder den Namen des Speichersystems, auf dem sich die Datenbank befindet, da dies Dinge sind, die sie möglicherweise ändern wollen (online im RAC durch Hinzufügen von Knoten und/oder ASM-Festplatten), ohne notwendigerweise alles umzubenennen.

Beim Anmelden bei einem Dienst sollte der Nutzer nur wissen, bei welcher Datenbank er sich anmeldet und welche Rolle ihr zukommt (Workload der Anwendung, Lesen/Schreiben, nur Lesen ...), nicht aber, wo sie physisch läuft. Dies ist Aufgabe von HA-Diensten, TAF, Application Continuity, dem Connection Manager, dem SCAN Listener ...

Speichern in script.old oder script.20200202 anstatt in der entsprechenden Versionskontrolle

Sie haben ein Skript geändert und wollen die frühere Version speichern für den Fall, dass ein Fehler auftritt? Sehr gut. Eine Umbenennung mit Datum, Ihrem Namen oder irgendeiner fantastischen Kodierung ist jedoch keine gute Idee. Zumal heute höchst unkompliziert Git für die Versionsverwaltung zur Verfügung steht. Damit können Sie mit einem einfachen Job Ihre Änderung vornehmen, sie auf einem oder mehreren Servern testen und überall einspielen – oder mit einem Rollback zur vorausgehenden Version wieder rückgängig machen. Ein weiterer Vorteil: Sie geben bei Freigabe einen Kommentar ein. Damit ist die Änderung selbstdokumentiert. Ihr nachfolgender Kollege wird sich über die Info, was warum geändert wurde, sehr freuen.

Keine Festlegung des allgemeinen Verzeichnisses für Konfigurationsdateien

Tnsnames.ora wird gemeinsam genutzt. Die Ablage unter \$ORACLE_HOME/network/admin ist keine gute Idee, da Sie mehrere \$ORACLE_HOME haben können und Sie bei einem Upgrade auf ein neues ORACLE_HOME wechseln werden. Standardmäßig wird als allgemeines Verzeichnis für Netzwerkdateien /etc. verwendet. Sie können aber auch ein TNS_ADMIN-Verzeichnis festlegen. Stellen Sie in diesem Fall sicher, dass die Einstellung für alle, die den Listener und die Datenbank möglicherweise starten, gleich ist. Entscheiden Sie gemeinsam mit Ihrem Team über die Vorgehensweise und halten Sie sich dann durchgehend daran. Und auch hier gilt: Lassen Sie sich nicht zu Bequemlichkeit hinreißen. Ich habe einen Server mit mehreren ORACLE_HOME gesehen (aufgrund mehrerer Versionen), wobei jede sqlnet.ora in \$ORACLE_HOME/network/admin einen symbolischen Link auf eine der vorausgehenden Versionen enthielt. Beim Upgraden der Datenbank erscheint dies als eine sehr einfache und wirkungsvolle Methode. Löscht ein Kollege jedoch irgendwann ein altes ORACLE_HOME, das nicht mehr genutzt wird, bricht die gesamte Kon-

struktion zusammen. Und wer trägt in diesem Fall die Schuld? Der Erste, der sein Upgrade nicht sauber durchgeführt hat.

Wenn Sie für eine bestimmte Datenbank spezielle Parameter benötigen, können Sie eine spezielle Datei einrichten, die die allgemeine Datei mit „ifile=“ einschließt.

Anhäufung von Prozessen auf toten Verbindungen

Um es kurz zu machen: Sie sollten in der serverseitigen Steuerdatei sqlnet.ora die SQLNET.EXPIRE_TIME hinzufügen, da bei Unterbrechung der Client-Session auf dem Server sonst ein nutzloser Prozess bestehen bleibt. Gründe, warum die TCP-Verbindung zwischen Client und Server unterbrochen werden kann, gibt es viele. Dabei werden zahlreiche Ressourcen genutzt: Prozess, Speicher, möglicherweise Locks und höchstwahrscheinlich ein DBA, um den Prozess irgendwann zu beenden. SQLNET.EXPIRE_TIME=10 ist mit Sicherheit eine gute Idee, da der Client so nach 10-minütiger Inaktivität geprüft wird.

Beenden blockierender Sessions ohne „Hanganalyze“

Es kann vorkommen, dass eine Session mehrere andere blockiert. In diesem Fall müssen Sie schnell handeln und die Session beenden, was keine Zeit für eine echte Analyse der Situation lässt. Ohne Analyse ist die Wahrscheinlichkeit jedoch hoch, dass das Problem erneut auftritt. Ein „Hanganalyze“ dauert nur wenige Sekunden und generiert genügend Informationen für eine Post-Mortem-Analyse (die von Ihnen oder dem Oracle Support durchgeführt werden kann). Mit diesem Thema beschäftigt sich auch mein Blogbeitrag: „Oracle bleibt hängen? Denken Sie an hanganalyze und systemstate!“ (siehe unter <https://blog.dbi-services.com/oracle-is-hanging-dont-forget-hanganalyze-and-systemstate/>)

Beschreibung der Infrastruktur in einem weiteren zusätzlichen Master-Repository

Sie könnten unter Umständen versucht sein, Ihre gesamte Infrastruktur (Server,

Datenbanken, Versionen ...) in einer Infrastrukturdatenbank zu beschreiben und alle Elemente (Bereitstellungsskripte, Backuppläne ...) auf dieser Basis zu erstellen. Es ist mir jedoch kein einziger Fall begegnet, in dem dies nicht zu einer punktbezogenen Abhängigkeit geführt hätte, die das Betriebsverfahren verkompliziert und sowohl die Automatisierungsmöglichkeiten als auch die Agilität einschränkt. Vor allem aber brauchen Sie das nicht, es gibt bereits eine ganze Reihe von Repositories, die jeweils einen ganz bestimmten Zweck erfüllen. Für die verschiedenen Server finden Sie in oratab oder Grid Infrastructure eine Liste der Datenbanken. Auf diesem Weg sind alle Informationen zugänglich. Über den RMAN-Katalog erhalten Sie alle Infos über Dateien und Backups. Über den Data Guard Broker erhalten Sie die Primary-/Standby-Konfiguration (und ja, den Broker nicht zu verwenden, zählt ebenfalls zu den „schlechtesten Praktiken“ ...). Die Enterprise Manager Cloud Control sammelt Informationen über Targets; die wenigen fehlenden Informationen (Abteilung, Geschäftsfeld, Lebenszyklus, Kontakt, Kostenstelle) können über die GUI oder die Befehlszeile ergänzt werden.

Verkomplizieren Sie die Dinge nicht. Es ist effizienter, vom Standard auszugehen und dann Informationen über die Konfiguration einzuholen, anstatt auf einer „Master-Meta-Datenbank“ aufzusetzen. Vermeiden Sie auf einen Punkt konzentrierte Abhängigkeiten zwischen Backups, Monitoring, Audit, Bereitstellung ... So können Sie Ihre Weiterentwicklung gezielt an einer Stelle vorantreiben und behindern sich nicht selbst durch wechselseitige Abhängigkeiten.

Für einen tieferen Einstieg in dieses Thema empfehle ich die Veröffentlichungen und Präsentationen von Ludovico Caldara (*siehe unter <http://www.ludovicocaldara.net/dba/>*) über Konvention vor Konfiguration und die Frage, warum die Verwaltung eines „Zoo“ vermieden werden sollte.

Falscher Konfigurationsansatz – wenn das Pferd von hinten aufgezäumt wird ...

Ich könnte unzählige Beispiele anführen, in denen mir ineffiziente Konfigurationen

begegnet sind. Einfach deshalb, weil bei der Definition am falschen Ende begonnen wurde. Stellen Sie sich vor, Sie kaufen zuerst die Hardware und überlegen sich dann, wie Sie am besten den RAC-Cluster konfigurieren – nur um festzustellen, dass Sie nicht über genügend Netzwerkarten für eine zuverlässige Verbindung verfügen. Oder Sie konfigurieren Hugepages als prozentualen Anteil des verfügbaren RAM-Speichers anstatt in Form von erforderlicher SGA-Größe. Oder Sie verwenden die In-Memory-Option für ein Datawarehouse, ohne sich zuvor Gedanken über die Partitionierung und Parallel Query zu machen. In diesem Fall war Parallel Query zwischen Instanzen sogar mit `parallel_force_local=true` deaktiviert und die Tabellen waren verteilt: Jeder Query stand nur die Hälfte der Zeilen im In-Memory Column Store zur Verfügung, die anderen bedurften eines Full Table Scan des Rowstore.

Verzicht auf den UTF-8-Zeichensatz

Im Jahr 2020 halte ich es nicht für zeitgemäß, die Namen von Menschen auf [A-Za-z -] zu beschränken oder in Textfeldern umgekehrte Fragezeichen anzuzeigen, wenn jemand typografische Anführungszeichen oder Ligaturen aus einer Unicode-Anwendung in ein Textfeld kopiert hat. Die Datenbank sollte für alle Zeichen, die in den Anwendungen und Java auftreten können, Unicode zulassen. Das geht auch ganz unproblematisch: Jede neue Anwendung sollte eine auf AL32UTF8 aufgebaute Datenbank ansprechen. Natürlich gibt es die Alternative, einen Single-Byte-Zeichensatz und NVARCHAR zu verwenden. Tatsächlich ist dies jedoch nicht sehr weit verbreitet.

Überstürzte Oneoff-Patches

Oracle ist eine komplexe Software und daher begegnen wir unweigerlich auch Problemen (Bugs). In der Regel gibt es in diesem Fall zwei „Lösungen“: Fehlerkorrekturen (wie Patches) und Ausweichlösungen (wie die Deaktivierung einer Funktion). Bei Implementierung in einem Release Update ist die Fehlerkorrektur natürlich die langfristig richtige Lösung.

Kurzfristig betrachtet ist ein Oneoff-Patch allerdings wohl die schlechteste Lösung. Sie werden sich der Konsequenzen schnell bewusst, wenn Sie einen Merge benötigen – und damit erhöhen Sie die technische Schuld noch weiter. Mir sind Datenbanken mit mehr als hundert Oneoff-Patches begegnet. Und natürlich kommen diese Patches dem Wunsch, immer auf dem neuesten Stand der Release Updates zu bleiben, nicht gerade entgegen, da Sie sie zunächst mit einem Rollback rückgängig machen, dann die RU einspielen und ein Merge für frühere und noch immer fortbestehende Fehlerkorrekturen anfragen müssen. Deshalb ist es immer vorzuziehen, eine Ausweichlösung zu finden und zu beantragen, dass der Fehler mit dem nächsten RU behoben wird. In meinem Artikel „Weshalb Sie immer das neueste Release Update nutzen sollten“ (*siehe <http://viewer.zmags.com/publication/568ef12d#/568ef12d/8>*) in OracleScene habe ich mich näher mit diesem Thema befasst.

Verzicht auf Kontrolle – seien Sie nicht zu clever

Funktionen, die sich auf die Gesamtkomplexität und den Gesamtbetriebsaufwand auswirken, sollten unter Kontrolle bleiben. Database Link ist eine fantastische Funktion, wenn Sie Daten teilen müssen oder Datenbankaufrufe zwischen zwei Datenbanken durchführen möchten. Allerdings sollte der Aufbau sorgfältig durchdacht und kontrolliert werden (lose Kopplung, keine zyklischen Abhängigkeiten, dokumentiert, sachgemäßes Update bei Aktualisierung der Testdatenbank aus der Produktion ...). Die Kontrolle zu behalten, bedeutet, dass sich der DBA um die Erstellung und Verwaltung kümmert. Mir sind Datenbanken untergekommen, in denen die Entwickler einfach schnell Datenbanklinks anlegen konnten, wenn sie einige Daten teilen wollten. Mir fiel dann die Aufgabe zu, Fehler mit verteilten Transaktionen über drei Datenbanken in verschiedenen Versionen zu beheben. Dies zuzulassen, ist kein schlauer Schachzug, sondern zeugt einfach von Faulheit. Natürlich zeigen sich die Konsequenzen später, wenn die Datenbanken migriert oder upgedatet werden sollen. Noch schlimmer wird es, wenn materialisierte Views (MV) als schnelle Lösung für Perfor-

manceprobleme eingesetzt werden. Ich habe eine „BI“-Datenbank gesehen, bei der nur wegen dieser häufigen MV-Refreshes täglich Redo-Logs im Terabyte-Bereich zusammenkamen. Nochmal, Replikation und Aggregate sind sehr sinnvolle Funktionen, wenn sie durchdacht eingesetzt und kontrolliert werden. Vergessen Sie dabei nicht die Auswirkungen auf die Infrastruktur. Bei der Konzeption ist ein Dev- und Ops-Ansatz unverzichtbar.

Und: Die Vergabe von höheren Privilegien mag in der Sandbox und in der Entwicklung ihren Platz haben. Nicht in der Produktion. In der Produktion sollte es undenkbar sein, dass versehentlich eine Tabelle gelöscht wird. Was in der Produktion läuft, sollte zuvor mit den gleichen automatisierten Skripten in der Entwicklung, im Test und in der Vorproduktion gelaufen und zwischen jeder Phase validiert worden sein.

Streitereien zwischen Dev und Ops

Wie bereits zuvor erwähnt: Dev und Ops müssen ein gemeinsames Konzept dafür entwerfen, was in der Produktion bereitgestellt und gewartet werden soll. Fehlt diese Kommunikation, so arbeiten diese Teams meiner Erfahrung nach gegeneinander. Der DBA weigert sich, Funktionen zu nutzen, die den Programmcode unterstützen würden. Oder der Entwickler gestaltet Ausweichlösungen unnötig komplex, was beim DBA wiederum auf Ablehnung stößt. Ich traf beispielsweise einmal auf einen zeitgesteuerten Job, der alle 15 Minuten eine umfangreiche Tabelle durchscannte, um das „Caching zu fördern“. Darin zeigt sich, wie wenig die Funktionsweise des Buffer Cache verstanden wird. Was als Verbesserung der Performance gedacht war, führte lediglich zu Ressourcenverschwendung für unnötige Vorgänge.

Reine Funktionsfähigkeit vor Qualität

Hier knüpfen wir an die Überlegungen zu technischer Schuld in der Einleitung an. Ein Nutzer bittet um die Kopie eines Schemas mit Data Pump. Sie kümmern sich sofort darum und sind zufrieden,

den Service Request schnell erledigen zu können. Haben Sie aber auch daran gedacht, dass die gleiche Anfrage wahrscheinlich bald wieder gestellt wird? Nehmen Sie sich einen Moment Zeit und dokumentieren Sie Ihre Schritte (in diesem Beispiel die expdp/impdp-Befehle), sodass die Erledigung beim nächsten Mal leichter fällt. Tauschen Sie sich mit Kollegen aus: Die Dokumentierung für Kollegen kommt im Grunde einem Code Review gleich. Der Kollege sagt ihnen möglicherweise, dass Sie ein „Flashback_Time“ vergessen haben und ihr schnelles Werk letztendlich inkonsistent ist. War dieses dokumentierte Verfahren erst ein paar Mal erfolgreich, ist der zusätzliche Schritt zu einem Skript nicht mehr groß. Und eines Tages hat Ihr Nutzer vielleicht sogar ein Portal, um es selbst laufen zu lassen. Diese Verkettung von Dokumentation, Skripterstellung und Automatisierung führt zu einer Verbesserung bei Qualität und Effizienz. Bei nachhaltiger Produktivität geht es nicht darum, die Dinge möglichst schnell zu erledigen, sondern um eine Reduzierung der technischen Schuld.

Im gleichen Sinne machen Sie Ihren Code mit der Ausführung von Aktionen in Form von Skripten durch die implizite Qualität auch für andere nutzbar. Durch die Namensgebung und Dokumentierung, die Überprüfung der Returncodes zur Fehlererkennung werden die Abläufe wiederholbar und alle temporären Anpassungen werden am Ende wieder zurückgesetzt. Ich habe beispielsweise eine login.sql gesehen, die mehrere Spalten mit ‚noprnt‘ formatierte. Sollten Sie oder ein Kollege irgendwann ein Skript laufen lassen, das eine gleichnamige Spalte enthält, kann viel Zeit mit der Suche nach dem Grund verloren gehen, warum diese Spalte einfach verschwindet. Wie viel Zeit werden Sie vergeuden, bis Sie dieses login.sql aufspüren? Jeder Verwendung von COLUMN in sqlplus sollte am Ende ein COLUMN CLEAR folgen. In der Informatik – wie auch überall sonst – sind Sie gut beraten, jeden Ort in dem Zustand zu verlassen, in dem sie ihn vorgefunden haben.

Meine Erfahrungen mit schlechtesten Praktiken, die die technische Schuld erhöhen und später ihre Konsequenzen zeigen, könnten noch viele Seiten füllen. Ich möchte jedoch mit einem wichtigen Rat schließen: Versetzen Sie sich gedank-

lich in die Zukunft und denken Sie an Ihre Kollegen. Machen Sie Nachhaltigkeit zu Ihrem obersten Prinzip.

Über den Autor

Franck Pachot ist Principal Consultant und Database Evangelist bei **dbi services** in der Schweiz, Oracle Certified Master, Oracle ACE Director und stolzes Mitglied des Oak Table Network.



Franck Pachot
franck.pachot@dbi-services.com



Wie PostgreSQL von 20 Jahren Oracle Tuning profitieren kann

Hervé Schweitzer, dbi services

Nach meinem Einstieg in die Welt des Oracle-Datenbank-Tunings vor 20 Jahren habe ich mich entschlossen, ein neues RDBMS anzuschauen. Mein Kollege Daniel Westermann, Team Leader des Open-Infrastructure-Teams, der schon länger mit PostgreSQL unterwegs ist, hat mich überzeugt, mir PostgreSQL anzugucken. Da das Tuning mein Lieblingsbereich ist, habe ich Daniel gebeten, für dbi services einen „dbi InSite PostgreSQL-Tuning-Workshop“ zu schreiben. Bei dbi services gibt es für jeden neuen Workshop einen Dryrun. Während dieses Dryruns wurde mir klar, dass ich das PostgreSQL-Tuning vertiefen werde, weil es sehr ähnlich wie Oracle aufgebaut ist, somit werde ich den „dbi InSite PostgreSQL-Tuning-Workshop“ auch selbst geben können.

PostgreSQL prompt

Am Anfang war das größte Hemmnis bei PostgreSQL, dass sqlprompt, bei dem per Default der # „Hashtag“ benutzt wird, was für mich der Linux Root User Prompt ist. Somit war meine erste Frage, wie ich diesen sqlprompt ändern kann, um einen sogenannten „Like Oracle SQL Prompt“ zu erhalten (siehe Abbildung 1 und 2).

Backslash „\“

Der „\“ ist die wichtigste Taste bei PostgreSQL. Ohne diese Taste werden Sie schnell ein Problem haben, denn alle Hilfe- oder Metadaten-Abfragen erfolgen immer mit „\“. Deswegen ist es wichtig, diese Taste schnell zu erreichen, ohne eine Tastenkombination.

Database Optimizer

Der PostgreSQL-Optimizer ist dafür zuständig, die Kosten einer SQL-Abfrage zu

```
postgres@dbi-pg :/home/postgres/ [PG11] psql -U postgres postgres
postgres=#
```

Abbildung 1: Standard PostgreSQL Prompt (Quelle: Hervé Schweitzer)

```
postgres@dbi-pg :/home/postgres/ [PG111] cat .psqlrc
. . .
\set PROMPT1 '%/ PSQL>'
\set PROMPT2 '%/ PSQL>'

postgres@dbi-pg :/home/postgres/ [PG11] psql -U postgres postgres
postgres PSQL> \c test
You are now connected to database "test" as user "postgres".
test PSQL>
test PSQL> first line of multiline code
(test PSQL> second line of multiline code
(test PSQL> ;
```

Abbildung 2: Angepasster PostgreSQL Prompt (Quelle: Hervé Schweitzer)

evaluieren, um den günstigsten Zugriffsplan zu finden und zu benutzen, genauso wie bei Oracle.

Hierfür braucht der Optimierer Statistiken. Auch hier sind die bei PostgreSQL und Oracle gesammelten Statistiken ähnlich.

Buffer Cache

Hier beginnt einer der größten Unterschiede zwischen PostgreSQL und Oracle. Bei Oracle spricht man von den Daten-Blocks, die wir in den Buffer Cache der Datenbank gespeichert haben wollen. Mit Oracle wird der Systemspeicher hauptsächlich von der Datenbank verwaltet, bis zu 80%. Somit weiß die Datenbank bis zu 80%, was genau im System gespeichert ist (siehe Abbildung 3).

Bei PostgreSQL nennt sich der Buffer Cache „Shared Buffer Cache“; diesen Cache setzt man bei PostgreSQL viel kleiner, wobei Standardwerte bei 25 bis 40% des Systemspeichers liegen. Dadurch weiß die PostgreSQL-Datenbank viel weniger, was

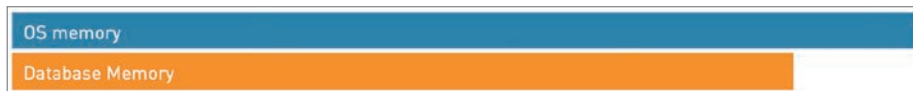


Abbildung 3: Oracle-Datenbank-Speicherverwaltung (Quelle: Hervé Schweitzer)



Abbildung 4: PostgreSQL-Datenbank-Speicherverwaltung (Quelle: Hervé Schweitzer)

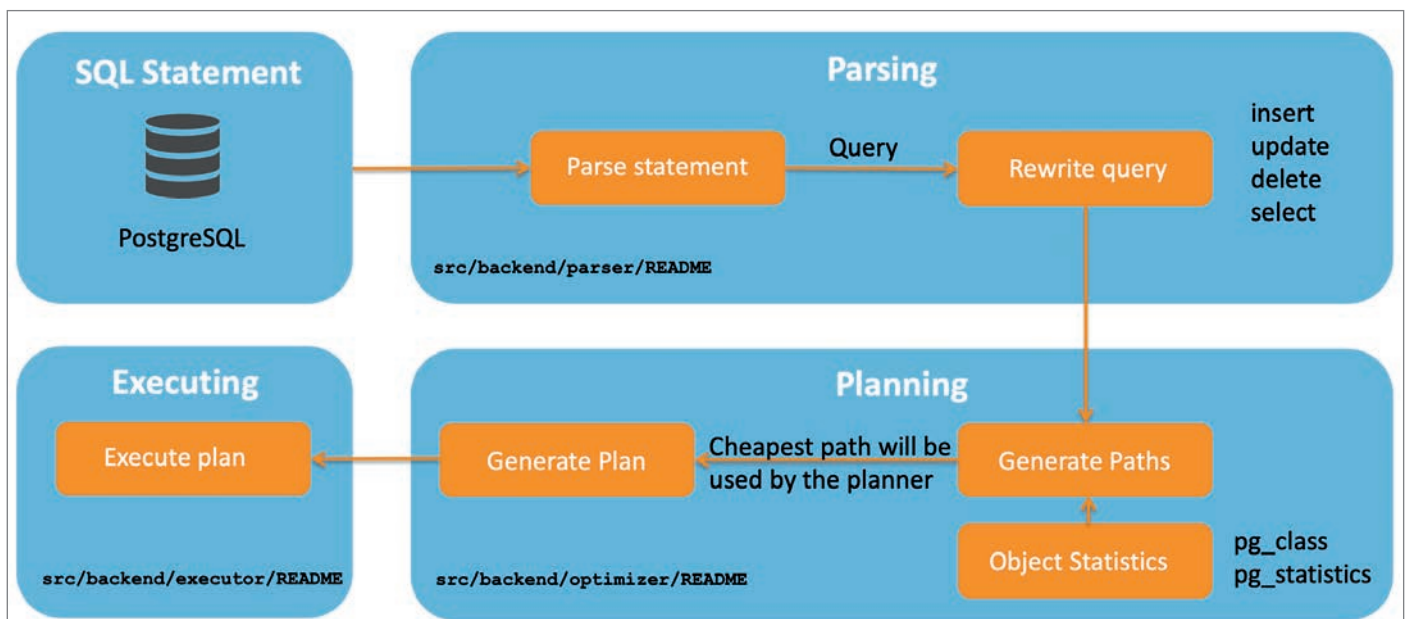


Abbildung 5: PostgreSQL-Optimierungsschritte (Quelle: Hervé Schweitzer)


```

pgbench PSQL> \h explain -- help page of all explain commands
pgbench PSQL> explain select * from t1 where a=1;

              QUERY PLAN
-----
Index Only Scan using index1 on t1 (cost=0.28..8.30 rows=1 width=4)
  Index Cond: (a = 1)
    
```

Abbildung 6: PostgreSQL Execution Plan (Quelle: Hervé Schweitzer)

```

pgbench PSQL> explain analyze select * from t1 where a=1;

              QUERY PLAN
-----
Index Only Scan using index1 on t1 (cost=0.28..8.30 rows=1) (actual time=1.625..1.626 rows=1)
  Index Cond: (a = 1)
  Heap Fetches: 1
  Planning Time: 0.092 ms
  Execution Time: 0.123 ms
    
```

Abbildung 7: PostgreSQL Execution Plan mit Ausführung (Quelle: Hervé Schweitzer)

im Systemspeicher liegt (siehe Abbildung 4). Vielleicht sind Daten schon im Systemspeicher, aber das ist nicht sichergestellt.

Shared Pool

Da gibt es nichts zu diskutieren über PostgreSQL und Oracle, denn PostgreSQL hat keinen Shared Pool und ist auch nicht dafür angelegt, einen zu implementieren. Bei PostgreSQL werden die Dictionary- und Executions-Plan-Information nur in der eigenen Session gespeichert. Sobald man eine neue Session eröffnet, ist alles verloren und alles muss wieder erst einmal in den Speicher geladen werden.

Optimierungsschritte

Die unterschiedlichen Optimierungsschritte zwischen Oracle und PostgreSQL sind ähnlich, der Hauptunterschied liegt beim Parsing (siehe Abbildung 5). Bei PostgreSQL wird standardmäßig jede neue Abfrage wieder geparkt. Das ist nicht der Fall bei Oracle, da es einen Shared Pool gibt (Soft-Parsing). Was es bei PostgreSQL gibt, ist der „PREPARE“-Befehl, um manchmal das Parsing zu umgehen. Wir werden später in diesem Artikel noch auf den „PREPARE“-Befehl zurückkommen.

Execution Plan

Mein Lieblings-Tool bei der Optimierung ist der Execution Plan, da ohne den Aus-

führungsplan das Problem nicht gefunden werden kann. Beim Tuning ist das Wichtigste, die Ursache des Problems zu finden, ansonsten werden Sie nur versuchen, das Problem zu lösen, ohne genau zu wissen, ob es funktionieren wird. Bei PostgreSQL nennt sich dieses Programm „explain“.

Abbildung 6 zeigt einen Execution Plan einer ganz einfachen Abfrage mit einem Index-Zugriff für eine Zeile. Hierbei wird die Abfrage nicht ausgeführt, es werden nur die Statistik-Informationen benutzt.

In Abbildung 7 wird die Abfrage ausgeführt und Sie werden nicht nur die Informationen von den Statistiken haben, sondern auch die Informationen über die Ausführung selbst, was viel mehr Informationen liefern wird. Beispielsweise die wirkliche Zeilenzahl und die Laufzeit der Abfrage.

In diesem Beispiel kann man sehen, dass geplant war, eine „rows“ zu retournieren, und beim Aufruf ist auch nur eine Zeile retourniert worden. Somit können wir sagen, dass der Plan gut war. Es ist ein guter Einstieg bei Problemen, bei denen

ein großer Unterschied zwischen den beiden „rows“-Werten besteht.

PREPARE-Befehl

Wie schon angesprochen, können Sie, wenn Sie die Parsing-Zeit bei PostgreSQL umgehen wollen, den „PREPARE“-Befehl benutzen. Mit PREPARE kann nach fünf Aufrufen der gleichen Abfrage das Parsing umgangen werden, allerdings nur in der eigenen Session. Zur Information, ein PREPARE verhält sich wie Bind-Variablen bei Oracle.

Damit ich Ihnen die Verwendung des PREPARE-Befehls am besten zeigen kann, gebe ich Ihnen ein kleines Beispiel (siehe Abbildung 8 bis 12) mit der unteren Tabelle SKEWED_DATA, die unregelmäßig verteilte Daten beinhaltet. Beabsichtigt ist, einen Full-Seq-Scan für a=1 zu verwenden und einen Index-Scan für alle andere Werte.

Ich führe einen ersten Aufruf der Abfrage mit a=1 aus. Wir sehen, dass kein Index-Scan gemacht wird.

```

pgbench PSQL> select a,count(*) from skewed_data group by a;
 a | count
---+-----
 1 | 99990
 2 |      1
 3 |      1
 4 |      1
 5 |      1
 6 |      1
 7 |      1
 8 |      1
 9 |      1
10 |      1
    
```

Abbildung 8: Tabelle SKEWED_DATA (Quelle: Hervé Schweitzer)


```

pgbench PSQL> prepare my_stmt as select * from skewed_data where a = $1;
pgbench PSQL> explain analyze execute my_stmt ('1');

```

QUERY PLAN

```

-----
Seq Scan on skewed_data (cost=0.00..29167.00 rows=99990 width=6) (rows=99991)
  Filter: (a = 1)
  Rows Removed by Filter: 9
  Planning time: 0.135 ms
  Execution time: 217.040 ms

```

Abbildung 9: Erster Aufruf mit a=1 (Quelle: Hervé Schweitzer)

```

pgbench PSQL> explain analyze execute my_stmt ('2');

```

QUERY PLAN

```

-----
Index Only Scan using i1 on skewed_data (cost=0.42..8.48 rows=3 width=6) (rows=1)
  Index Cond: (a = 2)
  Planning time: 0.155 ms
  Execution time: 0.083 ms

```

Abbildung 10: Zweiter Aufruf mit a=2 (Quelle: Hervé Schweitzer)

Dann versuchen wir es mit a=2; wie erwartet wird jetzt ein Index-Zugriff benutzt.

Dann führe ich bis zu drei Aufrufe der Abfrage, nochmal mit a=1, aus.

Wir können feststellen, dass der Optimierer für die drei weiteren Ausführungen mit a=1 den gleichen Plan wie im Schritt 1 wiederbenutzt und wir einen Full-Seq-Scan erhalten, da wir wieder 99% der Daten lesen wollen.

Jetzt machen wir noch einen sechsten Aufruf, und wir schauen, was passiert.

Warum haben wir jetzt plötzlich wieder einen Index-Scan, obwohl wir 99% der Daten lesen möchten? Dies ist eine klare Regel von PostgreSQL, was auch so in dem Code und in der Dokumentation beschrieben ist. Wenn nach fünf Aufrufen mit PRE-

PARE die Kosten von einem „Generic Plan“ billiger sind als die des „Custom Plan“, wird der „Generic Plan“ verwendet. „Generic Plan“ und „Custom Plan“ kann man anhand des Filters unterscheiden.

```

(a = 1) Custom Plan
(a = $1) Generic Plan

```

Standardmäßig generiert PostgreSQL bei der Verwendung des PREPARE-Befehls für die ersten fünf Aufrufe einen „Custom Plan“. Dieser übernimmt die Werte der Parameter. Ab der sechsten Ausführung werden die Kosten des „Custom Plan“ mit den Kosten von einem „Generic Plan“ verglichen, der die Werte der Parameter nicht übernimmt. Falls die Kosten des „Generic Plan“ billiger sind, wird die-

ser Plan übernommen und das wird auch für die Laufzeit der Session so bleiben.

Das bedeutet nicht, dass PostgreSQL schlecht ist, es heißt nur, dass Sie die Regeln ihres Optimierers kennen müssen, damit Sie ihn optimal nutzen können, und das gilt für alle RDBMS-Optimierer.

Seit PostgreSQL12 gibt es einen neuen Parameter, der es möglich macht, dieses Verhalten zu ändern.

PLAN_CACHE_MODE= AUTO (Default) heißt wie in PostgreSQL11.

Es gibt außerdem noch die beiden weiteren Werte

FORCE_GENERIC_PLAN und FORCE_CUSTOM_PLAN.

```

pgbench PSQL> explain analyze execute my_stmt ('1'); -- repeat that 3 times more --

```

QUERY PLAN

```

-----
Seq Scan on skewed_data (cost=0.00..29167.00 rows=99990 width=37)
  (actual time=0.014..231.884 rows=99991 loops=1)
  Filter: (a = 1)

```

Abbildung 11: Drei weitere Aufrufe mit a=1 (Quelle: Hervé Schweitzer)

```

pgbench PSQL> explain analyze execute my_stmt ('1');

```

QUERY PLAN

```

-----
Index Scan using i1 on skewed_data (cost=0.42..11300.93 rows=33333 width=37)
  (actual time=0.115..355.414 rows=99991 loops=1)
  Index Cond: (a = $1)

```

Abbildung 12: Sechster Aufruf mit a=1 (Quelle: Hervé Schweitzer)

Zusammenfassung

Bei der Datenbank-Optimierung ist es wichtig, die Probleme genau identifizieren zu können. Ansonsten werden Maßnahmen ergriffen, ohne zu wissen, ob diese überhaupt zum Erfolg führen, und das kann fatale Folgen haben. Um wirklich etwas zu erreichen, muss man wissen, wie der Optimierer genau funktioniert. Hat man das verstanden, wird es einfacher, die richtigen Maßnahmen zu treffen, um ein Performance-Problem zu lösen.

Bei der Datenbank-Optimierung sind Ihr Wissen und Ihre Erfahrung von unschätzbarem Wert. Erfahrung erwirbt man besonders mit Test-Umgebungen, damit man mit unterschiedlichen Optimierungs-Ansätzen ausprobieren kann, wie sich ein System genau verhält. Nur mit dieser Erfahrung kann man die entsprechenden Optimierungs-Maßnahmen dann auch fundiert einsetzen.

Als Schlusswort über PostgreSQL-Datenbank-Tuning: Welche Funktionalität fehlt in PostgreSQL am meisten gegenüber Oracle? PostgreSQL hat keinen „Shared Pool“, um die Datenbank-Session-Informationen mit anderen Sessions zu teilen. Wie könnte man dieses Problem bei PostgreSQL umgehen? Mit der Verwendung eines

„Connection Pool“, wobei die erforderlichen parallel notwendigen Verbindungen konfiguriert werden. Daher ist die Chance größer, Informationen im Cache einer bestehenden Datenbank-Sessions zu finden, da mehrere Datenbank-User sich eine kleine Anzahl von Sessions teilen müssen.

Über den Autor

Hervé Schweitzer hat mehr als 20 Jahre Erfahrung im Management, im Engineering und in der Optimierung von Datenbanken und Infrastrukturen. Er ist spezialisiert in Oracle-Technologien wie Standardisierung, Backup & Recovery und Tuning sowie in Hochverfügbarkeitslösungen wie Oracle Data Guard, Oracle Grid Infrastructure, Oracle Real Application Clusters (RAC), Oracle GoldenGate und Oracle Failsafe. Hervé Schweitzer ist „Oracle Certified Master 11g (OCM 11g)“.

Seit 2017 verbringt Hervé Schweitzer immer mehr Zeit auf PostgreSQL, was ihm ermöglicht hat, ein Expertenniveau im PostgreSQL-Tuning zu erreichen.

Hervé Schweitzer ist Referent für die dbi-Insite-Oracle-Data-Guard- und Oracle-GoldenGate-Workshops. Seit 2019 ist Hervé

auch als Referent für die dbi-Insite-PostgreSQL-DBA-Essential- und PostgreSQL-Tuning-Workshops zuständig. Zusätzlich zu seinen Erfahrungen auf Oracle und PostgreSQL unterstützt er derzeit Kunden beim Übergang ihrer Datenbank-Workloads zur AWS-Cloud.

Vor seinem Eintritt bei dbi services war Hervé Schweitzer Senior Consultant bei Trivadis in Basel. Zuvor arbeitete er als IT-Administrator und Oracle DBA bei Crossair in Basel.

Hervé Schweitzer verfügt über ein Informatik-Diplom (BTS, Frankreich).

Seine Branchenerfahrung umfasst Pharma, Transport & Logistik, Banken, Energie, Automotive etc.



Hervé Schweitzer
herve.schweitzer@dbi-services.com

Super-Power: Autonomous Database und Machine Learning

DOAG Online

Oracle ACE Director Jim Czuprynski zeigt in einer Artikel-Serie seine "Superkräfte" Autonomous Database und Machine Learning.

Der offene Ausgang der US-Präsidentenwahl 2020 bestimmte Anfang November die Medien. Die Auseinandersetzung mit der Vorhersage von US-Wahlergebnissen war einer der Gründe, warum Oracle-Datenbankadministrator Jim Czuprynski vor einigen Jahren damit begonnen hat, sich detaillierter mit den Möglichkeiten von Autonomous Database und Machine Learning auseinanderzusetzen.

Der US-amerikanische ACE Director ist mittlerweile ein Experte auf diesem Gebiet und teilt sein Wissen anhand konkreter Beispiele in seiner sehr lesenswerten vierteiligen Artikel-Serie "What's Your Super-Power? Mine is Autonomous Database and Machine Learning", erschienen in den Ausgaben 19 bis 22 des vom EOUC herausgegebenen eMagazines ORAWORLD (siehe <http://www.oraworld.org/home/>).

Die vier Artikel können Sie über die folgenden Links aufrufen: Teil 1 (siehe <http://www.oraworld.org/fileadmin/documents/19-ORAWORLD.pdf#page=31>), Teil 2 (siehe <http://www.oraworld.org/fileadmin/documents/20-ORAWORLD.pdf#page=43>), Teil 3 (siehe <http://www.oraworld.org/fileadmin/documents/21-ORAWORLD.pdf#page=44>) und Teil 4 (siehe <http://www.oraworld.org/fileadmin/documents/22-ORAWORLD.pdf#page=23>).



Oracle Clusterware 19c ist zertifiziert für Oracle Linux 8 – ein Testbericht

Torsten Kaden, Atos

Endlich ist es offiziell, die Oracle-Datenbank- und Grid-Version 19c wurden auch für die Plattform Oracle Linux / Red Hat Linux 8 freigegeben [1]. Bereits vor der Freigabe gab es hierfür in diversen Blogs eine Reihe inoffizieller Anleitungen, was am System alles angepasst werden musste, um eine 19c auf OL8 zum Laufen zu bringen. Jedoch mit dem offiziellen Segen von Oracle sollte die Installation ohne Probleme durchgeführt werden können. Da aktuell in einem der Kundenprojekte die Ablösung einer 12.1-Installation auf RHEL6 anstand und nichts gegen die neue Plattform sprach, habe ich mir die Installation erst auf einem Testsystem genau angeschaut. Im Folgenden beschreibe ich, wie eine Installation erfolgreich durchgeführt werden kann.

Seit Mai 2020 kann man in My Oracle Support (MOS) auf der Registerkarte „Certifications“ nachlesen, unter welchen Voraussetzungen Oracle die Version 19c auf der Plattform OL8 freigegeben hat. Mit der Suche nach:

- Product: Oracle Clusterware
- Release: 19.0.0.0.0
- Platform: Linux x86_64 Oracle Linux 8.x

bekommt man das Ergebnis (*siehe Abbildung 1*).

- 64-Bit-Installation
- Oracle Linux 8 Update 1 oder neuer
- Unbreakable Enterprise Kernel 6 Version 5.4.17-2011.0.7.el8uek.x86_64 oder neuer
- mindestens Clusterware Patchlevel RU 19.7 (oder RU 19.6 plus diverse OneOff Patches)

Zum Zeitpunkt meiner Tests ist OL8 U2 die aktuelle Version. Die Installation der Auswahl Minimal von der DVD V996906-01.iso (Oracle Linux Release 8 Update 2 for x86 64 bit, 7.8 GB) in einer virtuellen Maschine verläuft schnell und problemlos. Dabei sollte die Sprache des Systems auf en_US.UTF8 eingestellt bleiben. Andernfalls geben die Perl-Module aus dem Grid-Home Locale-Warnings aus. Danach

Certification Results

Back Share Link Print Preview

Oracle Clusterware 19.0.0.0.0 is certified on Linux x86-64 Oracle Linux 8.x Update 1+

Notes

Oracle Clusterware 19.0.0.0.0 with Linux x86-64 Oracle Linux 8.x

Product: For general information relating to certification for the Oracle Database product, including virtualization, interoperability, binary compatibility, general release and patch set information, see [Core Database Certification Information \(Doc ID 1306539.1\)](#).

Platform: For details about certifications of Oracle Database on Linux x86-64, see [Certification Information for Oracle Database on Linux x86-64 \(Doc ID 1304727.1\)](#).

For details about Oracle Linux Ksplice Support with the Oracle Database, see [Oracle Linux Ksplice Support with Oracle Database \(Doc ID 2312257.1\)](#).

Minimum RU: 19.7 (or 19.6 with patches, refer to aforementioned certification notes)

Minimum kernel versions:

- Oracle Linux 8.1 with the Unbreakable Enterprise Kernel 6: 5.4.17-2011.0.7.el8uek.x86_64 or later, or
- Oracle Linux 8.0 with the Red Hat Compatible kernel: 4.18.0-80.el8.x86_64 or later

ACFS: Please refer to [ACFS Supported on OS Platforms for Oracle ACFS certification details \(Doc ID 1369107.1\)](#)

Support Information

Product Release	End of			
	Premier Support	Error Correction	Extended Support	Sustaining Support
Oracle Clusterware 19.0.0.0.0	Not Set	Not Set	Not Set	Not Set

N [Need an explanation of support policies? Learn More..](#)

32/64 Bit Compatibility

Product Compatibility	32-bit	64-bit
Oracle Clusterware 19.0.0.0.0 64-bit		
Linux x86-64 Oracle Linux 8.x	✖	✔

Abbildung 1: Certification Results (© T. Kaden, My Oracle Support)

werden das UEK-Repository aktiviert, die neuesten Updates eingespielt und weitere benötigte Pakete inklusive kernel-uek, oracleasm-support und vor allem oracle-database-preinstall-19c nachinstalliert. Der Grid Infrastructure Installation and Upgrade Guide for Linux listet in Tabelle 4-1 (x86-64 Oracle Linux 8 Minimum Operating System Requirements) alle benötigten Pakete auf [2]. Am Ende der Tabelle 4-1 wird für „Patches and Known Issues“ auf die Oracle Database Release Notes

verwiesen. Hierin wird die bei OL8 nicht verfügbare compat-libcap1 auch unter einer eigenen Bug-Nummer 29772579 geführt [3]. Des Weiteren muss die Um-

gebungsvariable CV_ASSUME_DISTID auf den Wert OL7 gesetzt werden [3]. Da dies an verschiedenen Stellen in unterschiedlichen Shells benötigt wird, ist es sinn-

```
[root@ol8 ~]# vim /etc/profile.d/oracle.sh
export CV_ASSUME_DISTID=OL7
[root@ol8 ~]# vim /etc/selinux/config
SELINUX=permissive
[root@ol8 ~]# echo "192.168.245.128 ol8" >> /etc/hosts
```

Listing 1: Die Vorbereitungen für die Installation

```
[root@ol8 ~]# passwd oracle
[oracle@ol8 ~]$ mkdir -p /u01/app/oracle/product/19/grid /u01/app/oracle/product/19/dbhome_1
[oracle@ol8 ~]$ cd /u01/app/oracle/product/19/grid/
[oracle@ol8 grid]$ unzip -q /home/oracle/install/LINUX.X64_193000_grid_home.zip
[oracle@ol8 grid]$ rm -rf OPatch
[oracle@ol8 grid]$ unzip -q /home/oracle/patches/p6880880_190000_Linux-x86-64.zip
[oracle@ol8 grid]$ OPatch/patch version
OPatch Version: 12.2.0.1.21
OPatch succeeded.
[root@ol8 ~]# dnf install /u01/app/oracle/product/19/grid/cv/rpm/cvuqdisk-1.0.10-1.rpm [root@ol8 ~]# oracleasm
configure -e -u oracle -g oinstall -s y -v
[root@ol8 ~]# oracleasm init
[root@ol8 ~]# oracleasm createdisk -v DATA1 /dev/sdb1
[oracle@ol8 grid]$ ./runcluvfy.sh stage -pre hacfg
...
Vorabprüfung für Oracle Restart-Konfiguration war nicht erfolgreich.
ol8: PRVF-7532 : Package "compat-libcap1" fehlt auf Knoten "ol8"
[oracle@ol8 grid]$ ./gridSetup.sh -applyRU /home/oracle/patches/30899722/30869156
[root@ol8 ~]# /u01/app/oraInventory/orainstRoot.sh
[root@ol8 ~]# /u01/app/oracle/product/19/grid/root.sh
```

Listing 2: Die einzelnen Schritte zum gridSetup

```

...
Creating /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.
Using configuration parameter file: /u01/app/oracle/product/19/grid/crs/install/crsconfig_params
The log of current session can be found at:
  /u01/app/oracle/crsdata/ol8/crsconfig/roothas_2020-07-19_03-16-22PM.log
2020/07/19 15:16:25 CLSRSC-363: User ignored prerequisites during installation
LOCAL ADD MODE
Creating OCR keys for user 'oracle', privgrp 'oinstall'..
Operation successful.
LOCAL ONLY MODE
Successfully accumulated necessary OCR keys.
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
CRS-4664: Node ol8 successfully pinned.
2020/07/19 15:16:34 CLSRSC-330: Adding Clusterware entries to file 'oracle-ohasd.service'

ol8  2020/07/19 15:17:32  /u01/app/oracle/crsdata/ol8/olr/backup_20200719_151732.olr  3292847284
2020/07/19 15:17:33 CLSRSC-327: Successfully configured Oracle Restart for a standalone server

```

Listing 3: Die Ausgabe der Root-Skripte

voll, diese an zentraler Stelle einzutragen. Standardmäßig ist bei OL8 SELinux aktiviert. Damit kommt später die Clusterware jedoch nicht zurecht. Obwohl so nicht in der Oracle-Dokumentation angegeben, muss SELinux mindestens auf permissive herabgesetzt werden. Genauso müssen der Hostname und die IP-Adresse lokal eingetragen sein, ansonsten findet der Grid-Installer die Verbindung zum ohasd nach der root.sh nicht (siehe Listing 1).

Nach diesen Vorbereitungen muss der Rechner neu gestartet werden, damit der installierte UEK-Kernel genutzt und die SELinux-Einstellung wirksam wird. Jetzt geht die eigentliche Installation los. Die nächsten Schritte sind, wie in Listing 2 dargestellt, Grid-Patch und -Image entpacken, OPatch austauschen/aktualisieren und prüfen, ob alle Installationsvoraussetzungen erfüllt sind. Durch den oraclean-Aufruf wird eine Platte mit einem ASM-Label versehen, damit sie später als Kandidaten-Platte verwendbar ist. Eine interessante Feststellung am Rande. Zum Entpacken des Grid-Image sowie der Patches muss das Programm unzip verwendet werden. Das testweise Entpacken der Archive mit 7zip ließ den OPatch-Lauf mit Fehlern abbrechen und das Grid-Home war so beschädigt, dass der Patch nicht zurückgerollt werden konnte.

Wenn man alle Schritte dieser Anleitung befolgt, verläuft die Installation im Oracle Universal Installer fehlerfrei und

man erhält eine funktionierende Standalone-Clusterware (siehe Listing 3 und 4). Einzig die Verwendung des ASM Filter Driver ist nicht möglich (siehe Abbildung 2). Nach der Installation funktioniert auch opatchauto fehlerfrei. Es wird das Grid-Home gefunden und selbstständig die Clusterware herunter- und wieder hochgefahren. Ich erwähne dies deshalb, da bei diversen Installationsversuchen das Grid-Setup mit Fehlern abgeschlossen wurde und trotz laufender Clusterware-

Prozesse kein Patchen mit opatchauto möglich war.

Fazit

Für den produktiven Einsatz ist die Plattform OL8 mit Grid 19c weitestgehend nutzbar. Da bis dato Features, wie zum Beispiel ACFS, mit RU 19.7 auf dieser Plattform noch nicht zur Verfügung stehen [3], bleibt abzuwarten, ob mit einem

Name	Target	State	Server	State details
Local Resources				
ora.DATA.dg				
	ONLINE	ONLINE	ol8	STABLE
ora.LISTENER.lsnr				
	ONLINE	ONLINE	ol8	STABLE
ora.asm				
	ONLINE	ONLINE	ol8	Started, STABLE
ora.ons				
	OFFLINE	OFFLINE	ol8	STABLE
Cluster Resources				
ora.cssd				
1	ONLINE	ONLINE	ol8	STABLE
ora.diskmon				
1	OFFLINE	OFFLINE		STABLE
ora.evmd				
1	ONLINE	ONLINE	ol8	STABLE

Listing 4: Vom Installer angelegte Services

der nächsten Updates auch diese Features freigegeben werden. Des Weiteren sollte in der Dokumentation von Oracle auf die kleinen, aber entscheidenden Details der Konfiguration (SELinux, /etc/hosts-Eintrag) hingewiesen werden, ohne die eine erfolgreiche Installation nicht möglich ist. Nach mehreren Installationsversuchen auf dem Testsystem gelang es mir, mit dem oben beschriebenen Vorgehen ein Grid 19c erfolgreich aufzusetzen. Entsprechend lief die Installation auf dem Kundensystem auf Anhieb fehlerfrei durch, inklusive anschließendem Datenbank-19c-Setup.

(Dieser Artikel entstand bereits im Juni 2020)

Quellen

- [1] <https://mikedietrichde.com/2020/05/14/of-course-oracle-clusterware-is-certified-on-ol8-rhel8-as-well/>
 [2] <https://docs.oracle.com/en/database/oracle/oracle-database/19/cwlin/supported-oracle-linux-8-distributions-for-x86-64.html>

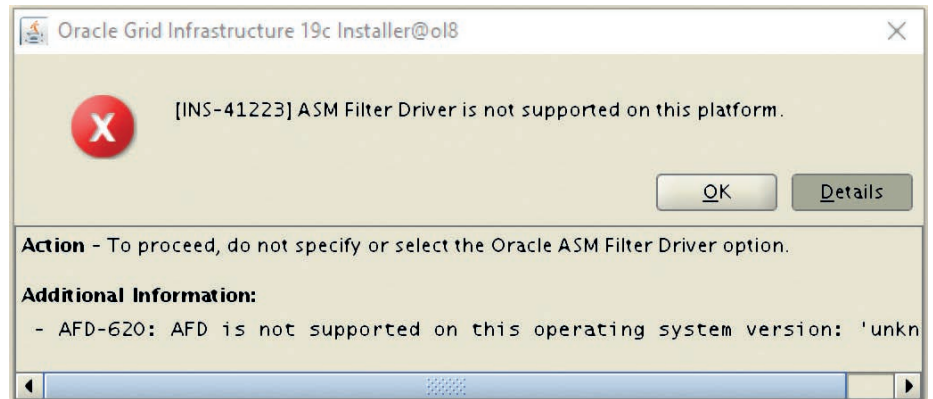


Abbildung 2: ASM-Filter-Driver-Fehlermeldung: os version is unknown (© T. Kaden, Oracle Universal Installer)

- [3] <https://docs.oracle.com/en/database/oracle/oracle-database/19/rnrdr/linux-platform-issues.html>

Über den Autor

Torsten Kaden arbeitet seit 16 Jahren bei Atos, früher science+computing AG, als Database Expert. Neben DBA-Tätigkeit und Datenbank-Consulting sind Unix-/Linux-Systeme ein weiterer Schwerpunkt.



Torsten Kaden
torsten.kaden@atos.net

DSGVO Vs. Cloud Act

DOAG Online

Das Verhältnis zwischen dem US CLOUD Act und der europäischen Datenschutz-Grundverordnung stellt Unternehmen vor ein Dilemma.

Die Datenschutz-Grundverordnung (DSGVO) findet seit dem 25. Mai 2018 Anwendung. Zwei Monate zuvor, am 23. März 2018, trat ein weiterer Rechtsakt in Kraft, der ebenfalls den Umgang mit Daten regelt: Der US-amerikanische CLOUD Act (Clarifying Lawful Overseas Use of Data Act), ein Gesetz zur Klärung der rechtmäßigen Verwendung von Daten in Übersee.

Gut zwei Jahre später ist das Verhältnis der beiden Gesetze weiterhin sehr umstritten und hat erhebliche Auswirkungen auf die Praxis. Der CLOUD Act gilt für Unternehmen auf der ganzen Welt mit mindestens einer Niederlassung oder Geschäftstätigkeit in den USA. Er soll den US-Strafverfolgungsbehörden den Zugriff auf die Kommunikationsdaten der Benutzer ohne Rückgriff auf internationale Rechtshilfeabkommen ermöglichen. Dies gilt insbesondere für den Fall, dass Daten von einer

US-Firma auf einem Server innerhalb der EU gespeichert werden. Dies ist aus der Sicht des europäischen Datenschutzrechts höchst problematisch.

Während der CLOUD Act die Offenlegung bestimmter Daten gegenüber den US-Strafverfolgungsbehörden verlangt, ist eine solche Offenlegung nach der DSGVO grundsätzlich verboten. Dies führt zu einer massiven Spannung zwischen CLOUD Act und dem europäischen Datenschutzrecht. Im zweiten Teil einer Artikelserie zum Thema CLOUD Act, erschienen in der Ausgabe #22 des eMagazines ORAWORLD (siehe <http://www.oraworld.org/fileadmin/documents/22-ORAWORLD.pdf#page=17>), beleuchtet Physiker und Rechtsanwalt Dr. David Bomhard das Dilemma und zeigt Möglichkeiten auf, wie dieses in der Praxis gelöst werden könnte.



Oracle Database Indexing Best Practices, Teil 3

Randolf Geist, Unabhängiger Berater

Auch wenn die Version 19c der Oracle-Datenbank die automatische Indizierung als neues Feature mit sich bringt, wird die Indizierung auch in Zukunft ein Dauerbrenner-Thema bleiben – auch schon aufgrund der Lizenzpolitik seitens Oracle, die das neue Feature in naher Zukunft nur einem Bruchteil der Kunden zukommen lassen wird.

Daher soll dieser Artikel mit einigen Mythen im Bereich Indizierung aufräumen und hilfreiche Tipps und Tricks beschreiben, die helfen, besser zu verstehen, was im Bereich Indizierung sinnvoll und nützlich ist. Im dritten und letzten Teil dieser Reihe schauen wir weiter auf Mehrspalten-Indizes und darauf, was es diesbezüglich noch weiter zu beachten gilt.

Index-Komprimierung

Durch das Voranstellen von Spalten mit wenigen Ausprägungen kann noch ein anderes Feature zum Einsatz kommen, das wahrscheinlich zu den am wenigsten genutzten, kostenlosen (Enterprise Edition) Features der Datenbank gehört: die sogenannte Index Compression. Diese gibt es schon seit der Version 8 von Oracle als „Basic Index Compression“, die keine weitere Lizenz über die Enterprise Edition hinaus benötigt. Seit der Version 12c gibt es zusätzlich eine „Advanced Index Compression“, die eine „Advanced Compression“-Lizenz voraussetzt.

Die „Basic Index Compression“ ersetzt in den Index Leaf Blocks führende Werte, die sich wiederholen, mit einem kürzeren Platzhalter/Symbol, und speichert den sich wiederholenden Wert nur einmalig im Block ab. Je größer also der Längenunterschied zwischen Wert und Platzhalter, desto effektiver der „Kompressionsfaktor“. Durch diese simple Implementierung erzeugt diese Art des Platzsparens so gut wie keinen messbaren Overhead beim Suchen und Pflegen des Index und

```
alter index CONC_INDEX1_CODE_ID rebuild compress 1;

SQL> SELECT index_name, blevel, leaf_blocks, clustering_factor
       2   from user_indexes
       3   where index_name = 'CONC_INDEX1_CODE_ID' or index_name = 'CONC_
INDEX1_ID_CODE';
INDEX_NAME          BLEVEL    LEAF_BLOCKS    CLUSTERING_FACTOR
-----
CONC_INDEX1_CODE_ID      2         2924           1000000
CONC_INDEX1_ID_CODE     2         3306           1000000
```

Listing 1: Kompressionseffekt nur mit Angabe der führenden Spalte

kann trotzdem die Größe einer Index-Struktur signifikant verkleinern.

Der maßgebliche Unterschied zwischen „Basic Index Compression“ und der kostenpflichtigen „Advanced Index Compression“ besteht darin, dass man bei der „Basic“-Variante die Anzahl der sinnvoll zu komprimierenden führenden Spalten selbst bestimmen muss und dies von Oracle bedingungslos umgesetzt wird – gibt man keinen Wert an, komprimiert Oracle alle Spalten (außer bei UNIQUE-Indizes, hier werden alle bis auf die letzte Spalte komprimiert). Wird hier eine unpassende Anzahl von Spal-

ten für die Kompression verwendet, kann sich der Effekt ins Gegenteil verkehren: Bei sich entsprechend wenig wiederholenden Werten wird so viel Platz für Platzhalter und Wertspeicherung verbraucht, sodass der Index größer anstatt kleiner werden kann. Insofern sollte man sich hier gut überlegen, welche Anzahl von Spalten Sinn ergibt. Oracle kann diesen Wert tatsächlich auch selbst ermitteln, leider ist dies aber umständlich: Mithilfe der Ausführung des Befehls ANALYZE INDEX ... VALIDATE STRUCTURE wird der spezielle View INDEX_STATS befüllt – und dieser beinhaltet neben vielen

```
select * from conc_index1 a where code = 2 and id2 = 4243;

  ID      CODE      ID2      FILLE
-----
  4242      2      4243      x
  4242      2      4243      x

-----
| Id | Operation                               | Name                | Rows | Bytes | Cost (%CPU) | Time      |
-----
|  0 | SELECT STATEMENT                         |                      |     2 |    38 |    593 (1)  | 00:00:01 |
|  1 | TABLE ACCESS BY INDEX ROWID BATCHED    | CONC_INDEX1         |     2 |    38 |    593 (1)  | 00:00:01 |
|*  2 | INDEX RANGE SCAN                        | CONC_INDEX1_CODE_ID |     2 |      |    591 (1)  | 00:00:01 |
-----

Predicate Information (identified by operation id):
-----
 2 - access("CODE"=2 AND "ID2"=4243)
    filter("ID2"=4243)

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

Listing 2: Kompressionseffekt mit Zugriff auf die erste und dritte Spalte

```

CREATE INDEX conc_index1_id_id2 ON conc_index1
(id, id2);
SQL> SELECT index_name, blevel, leaf_blocks, clustering_factor
  2   from user_indexes
  3   where table_name = 'CONC_INDEX1'
  4   ;

```

INDEX_NAME	BLEVEL	LEAF_BLOCKS	CLUSTERING_FACTOR
CONC_INDEX1_ID_ID2	2	2919	1000000

```
SQL> select * from conc_index1 a where id = 10000 and id2 = 10001;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	38	5 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	CONC_INDEX1	2	38	5 (0)	00:00:01
* 2	INDEX RANGE SCAN	CONC_INDEX1_ID_ID2	2		3 (0)	00:00:01

Predicate Information (identified by operation id):

```
2 - access("ID"=10000 AND "ID2"=10001)
```

Statistics

```

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

```

Listing 3: Index-Erzeugung und Ausführung einer Beispiel-Abfrage auf beide Spalten per Gleichheitsvergleich mittels SQL*Plus AUTOTRACE

anderen Statistiken über den untersuchten Index auch die Spalte OPT_COMPR_COUNT, welche die von Oracle ermittelte optimale Anzahl von führenden Spalten für die „Basic Index Compression“ darstellt.

Leider wird diese Information nur im OFFLINE-Modus von VALIDATE STRUCTURE ermittelt – und dieser wiederum sperrt den Index gegen Veränderungen während der Ausführung und ist somit auf aktiven Datenbanken nicht wirklich sinnvoll einsetzbar – wohl dem, der eine Kopie der Datenbank vorliegen hat, wo der Befehl ohne Auswirkungen ausgeführt werden kann.

Die „Advanced Index Compression“ löst dieses Problem auf sehr elegante Art und Weise – hier evaluiert Oracle für jeden Index Leaf Block automatisch,

welche führenden Werte sich für die Kompression eignen können – ist also in der Lage, das Verfahren flexibel anzupassen und damit unterschiedliche Einstellungen pro Block vornehmen zu können, je nachdem, wie die tatsächlichen Daten aussehen. Dies erlaubt dann auch, den Kompressionseffekt über die gesamte Index-Struktur gesehen zu maximieren.

Inzwischen, seit der Version 12.2, gibt es auch noch eine „Advanced High Index Compression“, die ein ganz anderes Konzept verfolgt und mit einem tatsächlichen Kompressionsverfahren nochmal ganz andere Kompressionsfaktoren für B*Tree-Indizes erreichen kann, wenn auch mit deutlich mehr CPU-Overhead verbunden.

In unserem konkreten Fall ist die Lage allerdings auch ohne das Ausführen des ANALYZE INDEX VALIDATE STRUCTURE-Befehls relativ offensichtlich – nur die führende CODE-Spalte wiederholt sich signifikant, die weiteren ID- und ID2-Spalten haben viele Ausprägungen und wiederholen sich wenig, wenn auch pro Wert zwei Mal. Insofern bekommen wir wahrscheinlich mit entweder nur der führenden Spalte oder allen dreien den besten Kompressionseffekt – hier in Listing 1 ein Beispiel dafür, was passiert, wenn ich nur die führende Spalte angebe.

Der Index ist jetzt ca. 10% kleiner geworden – mehr ist durch die Kürze der CODE-Spalte nicht drin, allerdings sparen wir uns ohne weitere Kosten dadurch eben 10% an Platz und bei entsprechen-


```
select * from conc_index1 a where id between 0 and 250000 and id2 = 10001;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	19	912 (3)	00:00:01
* 1	TABLE ACCESS FULL	CONC_INDEX1	1	19	912 (3)	00:00:01

Predicate Information (identified by operation id):

```
1 - filter("ID2"=10001 AND "ID"<=250000 AND "ID">=0)
```

Statistics

```

0 recursive calls
0 db block gets
3289 consistent gets
0 physical reads
0 redo size
555 bytes sent via SQL*Net to client
371 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
2 rows processed

```

Listing 4: Ergebnis bei Bereichssuche auf der ID-Spalte über die Hälfte des vorhandenen Wertebereichs

```
select /*+ index(a) */ * from conc_index1 a where id between 0 and 250000 and id2 = 10001;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		1	19	1472 (1)
1	TABLE ACCESS BY INDEX ROWID BATCHED	CONC_INDEX1	1	19	1472 (1)
* 2	INDEX RANGE SCAN	CONC_INDEX1_ID_ID2	1		1471 (1)

Predicate Information (identified by operation id):

```
2 - access("ID">=0 AND "ID2"=10001 AND "ID"<=250000)
   filter("ID2"=10001)
```

Statistics

```

0 recursive calls
0 db block gets
1467 consistent gets
0 physical reads
0 redo size
563 bytes sent via SQL*Net to client
372 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
2 rows processed

```

Listing 5: Index-Zugriff per Hint

den Operationen auch 10% an Arbeit – in *Listing 2* dargestellt am Beispiel des nicht mehr so effizienten Zugriffs auf die erste und dritte Spalte.

Die Operation benötigt jetzt also auch 10% weniger logisches I/O, da die Index-Struktur durch die Kompression kompakter geworden ist.

Grundsätzlich kann man sich also folgende Hinweise für die Reihenfolge von Spalten in B*Tree-Indizes merken:

- Solange man alle Spalten des Index bei der Suche angibt, spielt die Reihenfolge der Spalten für die Effizienz des Zugriffs keine Rolle.
- Soll der Index auch für andere Zugriffe verwendet werden können, sollten die Spalten an den Anfang gestellt werden, die den effizientesten Zugriff erlauben – bei denen also die Index-Struktur am effizientesten navigiert werden kann ohne Überspringen von Spalten oder Bereichs-abfragen (dazu gleich noch mehr).
- Die Einsatzmöglichkeiten eines Index können möglicherweise erhöht werden, wenn man Spalten mit wenig Ausprägungen an den Anfang des Index stellt, da Oracle dann mittels INDEX

SKIP SCAN-Zugriffen Abfragen verarbeiten kann.

- Das Voranstellen dieser Spalten mit wenig Ausprägungen erlaubt zusätzlich noch die bessere Nutzung der „Index Compression“, die bei der „Enterprise Edition“ von Oracle als „Basic Index Compression“ kostenfrei enthalten ist und sowohl den Platzbedarf verringern als auch die Effizienz steigern kann.

Mehrspalten-Indizes und Bereichsvergleiche

Abschließend noch ein Beispiel dafür, was passiert, wenn mit Bereichsvergleichen gearbeitet wird. Dazu erzeuge ich einen neuen Index auf der Tabelle, diesmal nur auf den Spalten ID und ID2. Solange ich mit einfachen Werten suche, funktioniert der Index-Zugriff effizient – hier wie in *Listing 3* dargestellt die Index-Erzeugung sowie die Ausführung einer Beispiel-Abfrage auf beide Spalten per Gleichheitsvergleich mittels SQL*Plus AUTOTRACE.

Mache ich aber eine Bereichssuche auf der ID-Spalte über die Hälfte des vorhan-

den Werteraums (Werteraum insgesamt von 0 bis 499.999, Suche von 0 bis 250.000), kann dies passieren (*siehe Listing 4*).

Hier wird nicht mehr der Index vom Optimizer herangezogen, sondern stattdessen ein „Full Table Scan“ durchgeführt – eine ineffiziente Operation, um zwei Zeilen aus einer Million zu identifizieren. Erzwingen wir den Index-Zugriff per Hint, wird auch klar, warum – wie sich in *Listing 5* ablesen lässt.

Durch Abfrage auf die Hälfte des Werteraums der führenden ID-Spalte kann nicht mehr effizient auf die ID2-Spalte innerhalb der Index-Struktur zugegriffen werden. Grund dafür ist der gleiche wie oben; er ergibt sich aus der Vorsortierung der Werte in der Index-Struktur – effektiv wird also der halbe Index durchsucht, darum sind die berechneten Kosten auch höher als beim „Full Table Scan“ und die logischen I/Os entsprechen ungefähr der Hälfte der Index-Größe (2.919 Leaf Blocks, 1.467 logische I/Os). Oracle zeigt das auch wieder in der „Predicate Information“ an – die Suchbedingung auf ID2 wird als FILTER aufgeführt.

Möchte ich also die gezeigte Abfrage effizienter gestalten, muss ich die Reihenfol-

```
CREATE INDEX conc_index1_id2_id ON conc_index1
(id2, id);

SQL> select * from conc_index1 a where id = 10000 and id2 = 10001;

-----
| Id | Operation                               | Name                | Rows | Bytes | Cost (%CPU)| Time     |
-----|-----|-----|-----|-----|-----|-----|
| 0 | SELECT STATEMENT                         |                     |      2 |      38 |      5 (0)| 00:00:01 |
| 1 | TABLE ACCESS BY INDEX ROWID BATCHED    | CONC_INDEX1         |      2 |      38 |      5 (0)| 00:00:01 |
|* 2 | INDEX RANGE SCAN                         | CONC_INDEX1_ID2_ID |      2 |      |      3 (0)| 00:00:01 |
-----

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

   2 - access("ID2"=10001 AND "ID"=10000)

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

Listing 6: Abfrage mit ID2 als führender Spalte

```
select * from conc_index1 where id between 0 and 250000 and id2 = 10001;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	19	4 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	CONC_INDEX1	1	19	4 (0)	00:00:01
* 2	INDEX RANGE SCAN	CONC_INDEX1_ID2_ID	1		3 (0)	00:00:01

Predicate Information (identified by operation id):

```
2 - access("ID2"=10001 AND "ID">=0 AND "ID"<=250000)
```

Statistics

```

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

```

Listing 7: Wiederholung der Bereichsabfrage

ge der Spalten im Index ändern und ID2 als führende Spalte angeben (siehe Listing 6).

Wie zuvor ändert sich an der Effizienz des Zugriffs über beide Spalten per Gleichheitsvergleich durch die Änderung der Spaltenreihenfolge nichts.

Wenn ich jetzt aber die Bereichsabfrage wiederhole, bekomme ich das in Listing 7 dargestellte Ergebnis.

Der Zugriff über den Index ist jetzt sehr effizient geworden, da zuerst über ID2 direkt in der Index-Struktur navigiert werden kann und auch die Bereichsabfrage durch die Sortierung abgedeckt ist – beide Bedingungen werden als Teil von ACCESS in der „Predicate Information“ angezeigt.

Bereichsabfragen bergen also bei der Verwendung von B*Tree-Indizes auch das Risiko, dass Zugriffe auf nachfolgende Spalten nicht mehr effizient über die Sortierung des Indizes abgebildet werden können, ganz ähnlich wie beim Überspringen von Spalten.

Dies ist ein weiteres Kriterium, das bei der Auswahl der Spaltenreihenfolge berücksichtigt werden sollte.

Quellen

- <http://www.oracle-performance.de>
- <http://oracle-randolf.blogspot.com>



Randolf Geist
randolf.geist@oracle-performance.de

Oracles Weblogic-Servern steht ein Angriff bevor

DOAG Online

Das Internet Storm Center (ISC) ist eine Organisation des mit Computersicherheit befassten SANS-Instituts. Das Institut überwacht die Anzahl bössartiger Aktivitäten und schädlicher Software im Internet. Forscher des ISC schlagen nun Alarm: Aufgrund einer als "kritisch eingestuften Sicherheitslücke" stehen Oracles WebLogic Server unmittelbar vor einer Attacke, so berichtete Heise kürzlich.

Aus Sicherheitsgründen wird Administratoren empfohlen, die laufenden WebLogic-Server so schnell wie möglich auf den aktuellen Stand zu bringen, da nach Angaben des SANS ISC die er-

kannten Sicherheitslücken relativ einfach auszunutzen seien. Betroffen sind folgende Versionen der WebLogic-Server:

10.3.6.0.0, 12.1.3.0.0, 12.2.1.3.0, 12.2.1.4.0, 14.1.1.0.0

Im quartalsweise erscheinenden Critical Patch Update von Oracle (DOAG Online berichtete) hat Oracle unter den insgesamt 402 Patches auch die entsprechenden Sicherheitsupdates für die gefährdeten WebLogic-Server ausgeliefert.



Getting data from Oracle to PostgreSQL and vice versa

Daniel Westermann, dbi services

In den letzten Jahren hat PostgreSQL in fast jedem Unternehmen Einzug gehalten und wird dabei vielseitig eingesetzt. Da aber in der heutigen Zeit zumeist mehrere Datenbank-Systeme zum Einsatz kommen, stellt sich schnell die Frage, wie Daten einfach zwischen diesen Systemen ausgetauscht werden können. PostgreSQL implementiert Teile der SQL/MED-Spezifikation, die sich genau mit diesem Thema beschäftigt – und genau darum soll es in diesem Artikel gehen: Wie schaffe ich es, PostgreSQL und Oracle so zu verbinden, dass ich einfach Daten in beide Richtungen austauschen und bearbeiten kann?

SQL/MED

Bei SQL/MED handelt es sich um „Management of external Data“. SQL/MED ist spezifiziert in ISO/IEC 9075-9:2008 und beschreibt Erweiterungen zu SQL, die sogenannten „Foreign Data Wrappers“ und „Data Link“-Typen. Ziel dieser Erweiterungen ist es, Daten in Fremdsystemen über SQL zu verwalten. Ein Fremdsystem muss dabei nicht unbedingt eine andere relationale Datenbank sein, sondern es kann sich hierbei um irgendein anderes System handeln, zum Beispiel Dateien, Internet-Seiten, NoSQL-Systeme oder auch Social-Media-Anbieter wie Facebook oder Twitter. Die Daten in einem Fremdsystem werden dabei als „External Data“ bezeichnet.

Ein Foreign Data Wrapper bietet die Möglichkeit, Daten in externen Systemen

über das SQL-Interface anzusprechen und diese bei Bedarf auch zu verbinden (join). Der SQL-Server (in diesem Fall PostgreSQL) ist dafür zuständig, ein SQL, das

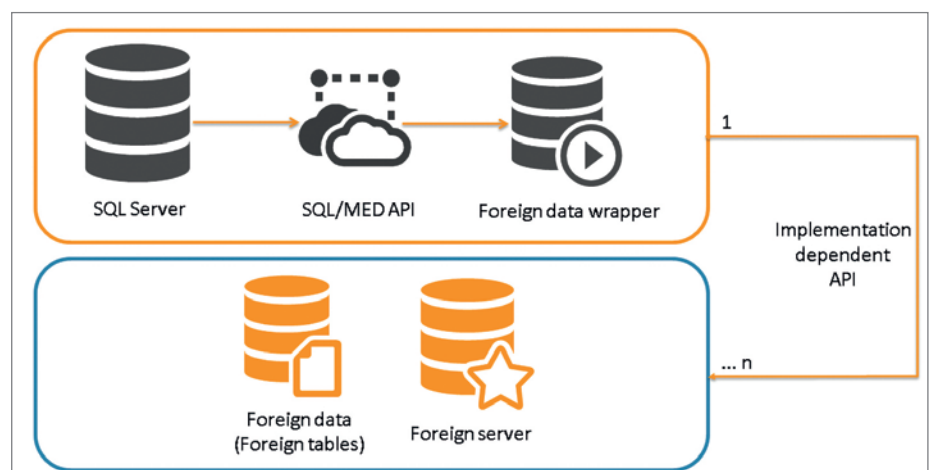


Abbildung 1: SQL/MED-Überblick (Quelle: Daniel Westermann)

NoSQL Database Wrappers

Data Source	Type	License	Code	Install	Doc	Notes
BigTable or HBase	Native Rust Binding (RPGFFI)	MIT	Github			
Cassandra	Multicorn	MIT	Github	Rankactive		
Cassandra2	Native	MIT	Github			
Cassandra	Multicorn	PostgreSQL	Github			
ClickHouse	Multicorn	BSD	Github		README	
ClickHouse	Native	Apache	Github		README	
CouchDB	Native	PostgreSQL	Github	PGXN		Original version
CouchDB	Native	PostgreSQL	Github			golgauth version (9.1 - 9.2+ compatible)
GridDB	Native	PostgreSQL	Github		README	
InfluxDB	Native	PostgreSQL	Github		README	
Kafka	Native	PostgreSQL	GitHub		README	
Kyoto Tycoon	Native	MIT	Github			

Abbildung 2: Überblick über vorhandene Foreign Data Wrappers (Quelle: Daniel Westermann)

gegen ein Fremdsystem abgesetzt wird, entsprechend auseinanderzunehmen und an den Foreign Data Wrapper zu übergeben. Dieser wiederum dient dazu, die Anfrage so umzuwandeln, dass sie vom Fremdsystem verstanden und verarbeitet werden kann. Grafisch dargestellt ergibt sich ein Bild wie in *Abbildung 1*.

Die Implementation des spezifischen Foreign Data Wrappers ist dabei abhängig vom Fremdsystem, in unserem Falle Oracle.

Foreign Data Wrappers in PostgreSQL

Die erste Version von PostgreSQL, die die Möglichkeit bietet, mit Foreign Data Wrappers zu arbeiten, ist PostgreSQL 9.1, das im September 2011 erschien. Seitdem hat es diverse Erweiterungen und Verbesserungen gegeben. Das grundsätzliche Vorgehen zum Anbinden von Fremdsystemen ist allerdings immer gleich geblieben und wird sich auch nicht ändern. Es braucht aufseiten von PostgreSQL immer:

1. den „Foreign Data Wrapper“ an sich
2. die Definition eines „Foreign Server“
3. die Definition(en) von „Foreign Table(s)“
4. ein User-Mapping, falls gegen den Foreign Server authentifiziert werden muss

Community PostgreSQL liefert standardmäßig zwei Foreign Data Wrappers mit:

- file_fdw für das Anbinden von Dateien

Table of Contents

- 56.1. Foreign Data Wrapper Functions
- 56.2. Foreign Data Wrapper Callback Routines
 - 56.2.1. FDW Routines for Scanning Foreign Tables
 - 56.2.2. FDW Routines for Scanning Foreign Joins
 - 56.2.3. FDW Routines for Planning Post-Scan/Join Processing
 - 56.2.4. FDW Routines for Updating Foreign Tables
 - 56.2.5. FDW Routines for Row Locking
 - 56.2.6. FDW Routines for EXPLAIN
 - 56.2.7. FDW Routines for ANALYZE
 - 56.2.8. FDW Routines for IMPORT FOREIGN SCHEMA
 - 56.2.9. FDW Routines for Parallel Execution
 - 56.2.10. FDW Routines for Reparameterization of Paths
- 56.3. Foreign Data Wrapper Helper Functions
- 56.4. Foreign Data Wrapper Query Planning
- 56.5. Row Locking in Foreign Data Wrappers

Abbildung 3: Eigene „Foreign Data Wrappers“ schreiben (Quelle: Daniel Westermann)

- postgres_fdw für das Anbinden von anderen PostgreSQL-Datenbanken

Zusätzliche Foreign Data Wrappers müssen nachinstalliert werden. Im PostgreSQL-Wiki (*siehe https://wiki.postgresql.org/wiki/Foreign_data_wrappers*) gibt es eine gute Übersicht über die vorhandenen Implementationen (*siehe Abbildung 2*).

Soll ein Fremdsystem angebinden werden, für das es noch keinen „Foreign Data Wrapper“ gibt, besteht immer die Möglichkeit, einen eigenen „Foreign Data Wrapper“ zu schreiben (*siehe Abbildung 3*). Auch das ist in der offiziellen PostgreSQL-Dokumentation beschrieben: <https://www.postgresql.org/docs/current/fdwhandler.html>

Die Installation eines „Foreign Data Wrapper“ erfolgt über das Kommando „create extension“, mit dem auch alle an-

deren Erweiterungen in PostgreSQL hinzugefügt werden.

Der „Foreign Data Wrapper“ für Dateien

Bevor wir mit den Anforderungen und der Installation des „Foreign Data Wrapper“ für Oracle beginnen, hier ein kleines Beispiel für die Installation und Anwendung des oben beschriebenen file_fdw. Dies soll einerseits verdeutlichen, dass alle „Foreign Data Wrappers“ die gleichen Aktionen in PostgreSQL benötigen, und andererseits die Vielfältigkeit der Anwendung von „Foreign Data Wrappers“ hervorheben.

Wie bereits angesprochen ist der file_fdw normalerweise schon vorhanden, falls PostgreSQL entsprechend installiert wurde. Bei der Installation aus Paketen (rpm

```

13:59:05 postgres@centos8pg:/home/postgres/ [pgdev] dnf search postgresql-
===== Name & Summary Matched: postgresql- =====
postgresql-jdbc-javadoc.noarch : API docs for postgresql-jdbc
===== Name Matched: postgresql- =====
postgresql-docs.x86_64 : Extra documentation for PostgreSQL
postgresql-jdbc.noarch : JDBC driver for PostgreSQL
postgresql-odbc.x86_64 : PostgreSQL ODBC driver
postgresql-test.x86_64 : The test suite distributed with PostgreSQL
postgresql-pltcl.x86_64 : The Tcl procedural language for PostgreSQL
postgresql-plperl.x86_64 : The Perl procedural language for PostgreSQL
postgresql-server.x86_64 : The programs needed to create and run a PostgreSQL server
postgresql-static.x86_64 : Statically linked PostgreSQL libraries
postgresql-contrib.x86_64 : Extension modules distributed with PostgreSQL
postgresql-upgrade.x86_64 : Support for upgrading from the previous major release of PostgreSQL
postgresql-plpython3.x86_64 : The Python3 procedural language for PostgreSQL
postgresql-odbc-tests.x86_64 : Testsuite files for psqlodbc
soci-postgresql-devel.x86_64 : PostgreSQL back-end for soci
postgresql-server-devel.x86_64 : PostgreSQL development header files and libraries
postgresql-upgrade-devel.x86_64 : Support for build of extensions required for upgrade process
postgresql-test-rpm-macros.x86_64 : Convenience RPM macros for build-time testing against PostgreSQL server

```

Abbildung 4: CentOS-8-PostgreSQL-Pakete (Quelle: Daniel Westermann)

oder deb) sollte man darauf achten, dass die Erweiterungen in separaten Paketen zur Verfügung gestellt werden. Bei CentOS 8 würde das wie in *Abbildung 4* aussehen.

Die Installation des file_fdw (siehe *Listing 1*).

Da file_fdw dafür da ist, um mit Dateien zu arbeiten, erstellen wir ein simples CSV mit einer Million Zeilen. Was hierbei zu beachten ist:

- «::» in PostgreSQL bedeutet einen Cast von einem Datentyp in einen anderen
- «||» funktioniert wie auch in Oracle: Verkettung von Strings
- generate_series ist eine PostgreSQL-Funktion, die eine Reihe von Integern generiert, in diesem Fall von 1 bis 1.000.000
- copy ist ein PostgreSQL-Kommando, um Daten aus Dateien in PostgreSQL zu lesen und diese auch wieder in Dateien rauszuschreiben (siehe *Listing 2*).

Um diese CSV-Datei nun über den „Foreign Data Wrapper“ ansprechen zu können, müssen der „Foreign Server“ und die „Foreign Table“ erstellt werden. Der „Foreign Server“ definiert, welcher „Foreign Data Wrapper“ verwendet werden soll; die „Foreign Table“ definiert die Schnittstelle zur Datei (siehe *Listing 3*).

Sobald beide Objekte erfolgreich erstellt sind, kann die CSV-Datei über das normale SQL-Interface angesprochen werden (siehe *Listing 4*).

Da es beim Ansprechen einer Datei keine Authentifizierung braucht, muss in diesem Fall kein „User Mapping“ erstellt werden.

Der „Foreign Data Wrapper“ für Oracle

Im Gegensatz zu file_fdw ist der „Foreign Data Wrapper“ für Oracle nicht in Community PostgreSQL enthalten und muss nachinstalliert werden. Als Vorbedingung muss ein Oracle Client auf dem PostgreSQL-System vorhanden sein, da oracle_fdw gegen diesen kompiliert werden muss.

Der Oracle Instant Client (<https://www.oracle.com/database/technologies/instant-client.html>) reicht hierbei völlig aus und es sollten folgende Instant-Client-Pakete auf dem PostgreSQL-Server vorhanden sein:

- Basic
- SQL*Plus
- SDK

```

postgres=# create extension file_fdw;
CREATE EXTENSION
postgres=# \dx

                List of installed extensions
   Name   | Version | Description
-----+-----+-----
file_fdw | 1.0     | foreign-data wrapper for flat file access
plpgsql  | 1.0     | PL/pgSQL procedural language
(2 rows)

```

Listing 1: Die Installation von file_fdw

```

postgres=# create table file ( dummy text );
CREATE TABLE
postgres=# insert into file
           select i::text||','
              ||i::text||','||i::text||','||i::text
           from generate_series(1,1000000) i;
INSERT 0 1000000
postgres=# copy file to '/var/tmp/dummy.csv';
COPY 1000000
postgres=# \! head -5 /var/tmp/dummy.csv
1,1,1,1,1
2,2,2,2,2
3,3,2,3,3
4,4,2,4,4
5,5,2,5,5
...

```

Listing 2: Erstellen einer simplen CSV-Datei

Sämtliche Pakete stellt Oracle sowohl als rpm-Paket wie auch als zip-Datei bereit. Um Distributions-unabhängig zu bleiben, werden wir mit den zip-Dateien arbeiten. Auf diese Weise kann alles auch auf Debian-basierten Systemen nachgestellt werden. Der generelle Überblick über die Systeme ist in *Abbildung 5* dargestellt.

Der erste Schritt ist nun, den Oracle Instant Client auf dem PostgreSQL-Server zu entpacken und die Verbindung zur Oracle-Datenbank mit SQL*Plus zu testen (*siehe Listing 5*).

Um spätere Probleme beim Verbindungsaufbau auszuschließen, empfiehlt es sich immer, die Verbindung zur Oracle-Datenbank mit SQL*Plus zu testen und vorab zu prüfen, ob SQL*Plus korrekt funktioniert (*siehe Listing 6*).

SQL*Plus hat Abhängigkeiten zu libnsl (Legacy support library for NIS). Da dieses Paket nicht installiert ist, muss dies nachgeholt werden (*siehe Listing 7*).

Sind alle Abhängigkeiten gegeben, kann die Verbindung zur Oracle-Datenbank getestet werden (*siehe Listing 8*).

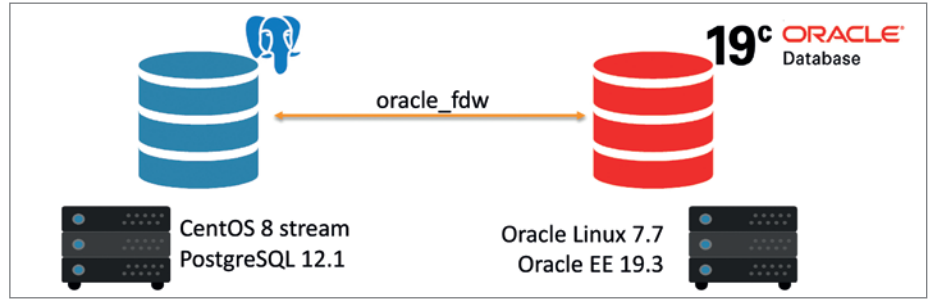


Abbildung 5: System-Übersicht (Quelle: Daniel Westermann)

```
postgres=# create server srv_file_fdw foreign data wrapper file_fdw;
CREATE SERVER
postgres=# create foreign table exttab ( a int, b int, c int, d int, e int)
server srv_file_fdw
options ( filename '/var/tmp/dummy.csv'
, format 'csv', header 'false' );
```

Listing 3: Erstellen des „Foreign Server“ und der „Foreign Table“

```
postgres=# select * from exttab limit 5;
 a | b | c | d | e
---+---+---+---+---
 1 | 1 | 1 | 1 | 1
 2 | 2 | 2 | 2 | 2
 3 | 3 | 3 | 3 | 3
 4 | 4 | 4 | 4 | 4
 5 | 5 | 5 | 5 | 5
(5 rows)
```

Listing 4: Ansprechen der CSV-Datei über SQL

```
postgres@pg:~$ pwd
/home/postgres
postgres@pg:~$ ls -l
total 75220
-rw-rw-r--. 1 postgres instantclient-basic-linux.x64-19.5.0.0.dbru.zip
-rw-rw-r--. 1 postgres instantclient-sdk-linux.x64-19.5.0.0.dbru.zip
-rw-rw-r--. 1 postgres instantclient-sqlplus-linux.x64-19.5.0.0.dbru.zip
postgres@pg:~$ unzip instantclient-basic-linux.x64-19.5.0.0.dbru.zip
postgres@pg:~$ unzip instantclient-sdk-linux.x64-19.5.0.0.dbru.zip
postgres@pg:~$ unzip instantclient-sqlplus-linux.x64-19.5.0.0.dbru.zip
postgres@pg:~$ ls instantclient_19_5/
adrci                libclntsh.so        libipcl.so          libocci.so.12.1
libsqlplusic.so     SDK_README          xstreams.jar       BASIC_LICENSE
libclntsh.so.10.1  libmq11.so         libocci.so.18.1    libsqlplus.so
Sqlplus             BASIC_README       libclntsh.so.11.1  libnnz19.so
libocci.so.19.1    network            SQLPLUS_LICENSE    genezi
libclntsh.so.12.1  libocci.so         libociei.so        ojdbc8.jar
SQLPLUS_README    glogin.sql         libclntsh.so.18.1  libocci.so.10.1
libocijdbc19.so   sdk                ucp.jar            libclntshcore.so.19.1
libclntsh.so.19.1  libocci.so.11.1   liboramysql19.so   SDK_LICENSE
uidrvci
postgres@pg:~$ mkdir -p instantclient_19_5/network/admin
```

Listing 5: Entpacken des Oracle Instant Client

```
postgres@pg:~$ export PATH=$PATH:~/instantclient_19_5/
postgres@pg:~$ export ORACLE_HOME=~/instantclient_19_5/
postgres@pg:~$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME:$ORACLE_HOME/sdk/include/
postgres@pg:~$ export TNS_ADMIN=$ORACLE_HOME/network/admin
postgres@pg:~$ sqlplus /nolog
sqlplus: error while loading shared libraries: libnsl.so.1: cannot open shared object file: No such file or directory
```

Listing 6: Funktionstest von SQL*Plus

```
postgres@pg:$ sudo dnf install -y libnsl
```

Listing 7: Installation von libnsl

```
postgres@pg:$ cat > ~/instantclient_19_5/network/admin/tnsnames.ora << EOF
ORATARGET.WORLD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.22.213)(PORT = 1521))
    )
  )
(CONNECT_DATA =
  (SERVICE_NAME = db1.it.dbi-services.com)
)
)
EOF

postgres@pg:$ cat > ~/instantclient_19_5/network/admin/sqlnet.ora << EOF
NAMES.DIRECTORY_PATH= (LDAP, TNSNAMES, HOSTNAME)
NAMES.DEFAULT_DOMAIN = WORLD
TRACE_LEVEL_CLIENT = OFF
SQLNET.EXPIRE_TIME = 30
EOF

postgres@pg:$ sqlplus system/manager@ORATARGET

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Feb 7 17:03:42 2020
Version 19.5.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Last Successful login time: Mon Feb 03 2020 11:27:11 +01:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL>
```

Listing 8: Verbindungstest zur Oracle-Datenbank

```
postgres@pg:$ git clone https://github.com/laurenz/oracle_fdw.git
postgres@pg:$ ls oracle_fdw/
CHANGELOG          Makefile           oracle_fdw--1.1--1.2.sql      oracle_fdw.control
oracle_utils.c     sql                expected                     msvc
oracle_fdw--1.2.sql oracle_fdw.h        README.md                     TODO
LICENSE            oracle_fdw--1.0--1.1.sql  oracle_fdw.c
oracle_gis.c       README.oracle_fdw
```

Listing 9: Herunterladen der oracle_fdw Sources

Grundsätzlich ist unser PostgreSQL-System nun bereit für die Installation des „Foreign Data Wrapper“ für Oracle. Wie viele andere PostgreSQL-Erweiterungen auch, kann oracle_fdw von GitHub bezogen werden – am einfachsten, indem das Repository direkt mit git bezogen wird (siehe Listing 9).

Da der „Foreign Data Wrapper“ für Oracle kompiliert und installiert werden muss, müssen die Umgebungsvariablen für Oracle (wie oben geschehen) gesetzt sein. Zusätzlich muss auch pg_config im PATH zu finden sein (siehe Listing 10).

pg_config gibt Konfigurationsparameter einer PostgreSQL-Installation zurück und

```
postgres@pg:$ which pg_config
/u01/app/postgres/product/DEV/db_1/bin/pg_config
```

Listing 10: pg_config

```
postgres@pg:$ cd oracle_fdw/
postgres@pg:$ make
postgres@pg:$ make install
```

Listing 11: Die Installation von oracle_fdw

jede Extension, die installiert wird, muss darauf zugreifen können, da ansonsten die PostgreSQL „header files“ und Bibliotheken nicht gefunden werden können.

Die Installation des „Foreign Data Wrapper“ für Oracle ist dann eine Sache von zwei Befehlen (siehe Listing 11). Bevor wir die Erweiterung nun in PostgreSQL in-

stallieren können, muss PostgreSQL einmal neu gestartet werden, damit die Umgebungsvariable LD_LIBRARY_PATH die richtigen Werte enthält, wenn PostgreSQL startet. Ansonsten wird PostgreSQL die Oracle Libraries nicht finden (siehe Listing 12).

Um sicherzustellen, dass die Umgebungsvariablen immer richtig gesetzt sind, wenn PostgreSQL startet, empfiehlt es sich natürlich, diese permanent zu setzen (z.B. im .bash_profile oder im PostgreSQL Startip-Skript). Ist dies gewährleistet, kann

die Erweiterung problemlos installiert werden (siehe Listing 13).

Das weitere Vorgehen ist ähnlich wie bei file_fdw: Man braucht die Definition des „Foreign Server“ und ein „User Mapping“, um gegen Oracle zu authentifizieren. Im Unterschied zu vorher werden hier nun Verbindungsinformationen mitgegeben (siehe Listing 14).

Von nun an ist alles bereit, um Daten aus Oracle abzufragen und diese auch wieder zurückzuschreiben. Die Definition(en) der

„Foreign Table(s)“ kann auf zwei Arten erfolgen: zum einen, wie hier im ersten Beispiel, Tabelle für Tabelle (siehe Listing 15).

Dieses Vorgehen würde allerdings initial einiges an Aufwand bedeuten. Vor allem dann, wenn es sich um viele Tabellen handelt, die man in der Oracle-Datenbank ansprechen möchte. Viel einfacher und auch schneller geht es wie in Listing 16 abgebildet.

Nicht jeder „Foreign Data Wrapper“ implementiert „import foreign schema“. Ist es allerdings implementiert, erleich-

```
postgres@pg:$ psql postgres
psql (12.1 dbi services build)
Type "help" for help.

postgres=# create extension oracle_fdw;
ERROR: could not load library "/u01/app/postgres/product/12/db_1/lib/oracle_fdw.so": libclntsh.so.19.1: cannot
open shared object file: No such file or directory
postgres=#
```

Listing 12: Falls die Umgebung nicht richtig gesetzt ist

```
postgres@pg:$ pg_ctl stop
postgres@pg:$ echo $LD_LIBRARY_PATH
:/home/postgres/instantclient_19_5/:/home/postgres/instantclient_19_5/sdk/include/:/home/postgres/instantcli-
ent_19_5/:/home/postgres/instantclient_19_5/sdk/include/
postgres@pg:$ pg_ctl start
postgres@pg:$ psql postgres
psql (12.1 dbi services build)
Type "help" for help.

postgres=# create extension oracle_fdw;
CREATE EXTENSION
postgres=# \dx

                List of installed extensions
  Name      | Version | Schema  | Description
-----+-----+-----+-----
 oracle_fdw | 1.2     | public  | foreign data wrapper for Oracle access
 plpgsql   | 1.0     | pg_catalog | PL/pgSQL procedural language
```

Listing 13: Neustarten von PostgreSQL mit gesetzten Umgebungsvariablen

```
postgres=# create server oratarget foreign data wrapper oracle_fdw
           options (dbserver '//192.168.22.213:1521/db1.it.dbi-services.com');
CREATE SERVER
postgres=# \des+

                List of foreign servers
  Name      | Foreign-data wrapper | FDW Options
-----+-----+-----
 oratarget  | oracle_fdw           | (dbserver '//192.168.22.213:1521/db1.it.dbi-services.com') |

postgres=# create user mapping for postgres server oratarget
           options (user 'system', password 'manager');
CREATE USER MAPPING

postgres=# \deu+

                List of user mappings
  Server  | User name | FDW options
-----+-----+-----
 oratarget | postgres | ("user" 'system', password 'manager')
(1 row)
```

Listing 14: Erstellen des „Foreign Server“ und des „User Mapping“

```

postgres=# create schema oracle;
CREATE SCHEMA
postgres=# create foreign table oracle.sales
  ( prod_id integer options (key 'true') not null
  , cust_id integer options (key 'true') not null
  , time_id timestamp options (key 'true') not null
  , channel_id integer options (key 'true') not null
  , promo_id integer options (key 'true') not null
  , quantity_sold numeric not null
  , amount_sold numeric not null
  ) server oratarget options (schema 'SH', table 'SALES');

postgres=# select count(*) from oracle.sales;
 count
-----
 918843
(1 row)

postgres=# select prod_id,cust_id from oracle.sales limit 5;
 prod_id | cust_id
-----+-----
      13 |     987
      13 |    1660
      13 |    1762
      13 |    1843
      13 |    1948

```

Listing 15: Erstellen einer „Foreign Table“

tert es das initiale Erstellen der „Foreign Tables“ erheblich.

Sind die „Foreign Tables“ erstellt, kann auf jede Tabelle in der Oracle-Datenbank zugegriffen werden. Man muss sich dabei allerdings bewusst sein, dass jeder Zugriff auf eine „Foreign Tabelle“ in PostgreSQL im Hintergrund über das Netzwerk auf die Oracle-Tabellen zugreift. Die Daten sind nur in der Oracle-Datenbank vorhanden, nicht aber in PostgreSQL.

Soll viel und vor allem performant mit den Daten gearbeitet werden, empfiehlt es sich, die Daten lokal nach PostgreSQL zu laden, beispielsweise über eine Materialized View (siehe Listing 17).

Sind die Daten dann verarbeitet, können sie auch wieder zur Oracle-Datenbank zurückgeschrieben werden (siehe Listing 18).

Der „Foreign Data Wrapper“ für Oracle bietet noch diverse andere Einstellungen, zum Beispiel kann auf Tabellen-Basis das Prefetching eingestellt werden, um den Datentransfer zu optimieren (siehe Listing 19).

Zudem können „Foreign Tables“ read only gestellt werden (siehe Listing 20).

Ein mächtiges Werkzeug ist der sogenannte Join-Pushdown. Einige „Foreign Data Wrappers“ (oracle_fdw ist einer von diesen) erlauben das Sammeln von Statistiken für den Optimizer von „Foreign Table(s)“ (siehe Listing 21).

Sind Statistiken vorhanden, kann ein join-pushdown angewendet werden: Wer-

```

postgres=# create schema oratarget;
CREATE SCHEMA
postgres=# import foreign schema "SH" from server oratarget
           into oratarget
           options (case 'lower');
IMPORT FOREIGN SCHEMA
postgres=# select a.relname
           , b.ftserver
           , b.ftoptions
           from pg_class a
           , pg_foreign_table b
           where b.ftrelid = a.oid
           order by a.relname;

postgres=# select prod_id,cust_id,time_id,amount_sold
           from oratarget.sales limit 5;

 prod_id | cust_id |          time_id          | amount_sold
-----+-----+-----+-----
      13 |     987 | 1998-01-10 00:00:00 |    1232.16
      13 |    1660 | 1998-01-10 00:00:00 |    1232.16
      13 |    1762 | 1998-01-10 00:00:00 |    1232.16
      13 |    1843 | 1998-01-10 00:00:00 |    1232.16
      13 |    1948 | 1998-01-10 00:00:00 |    1232.16
(5 rows)

```

Listing 16: Importieren des „Foreign-Schemas“

```

postgres=# create materialized view mv_my_customers as
           select cust_id, cust_first_name,cust_last_name,cust_gender
           from oratarget.customers;
SELECT 55500
postgres=# select cust_id,cust_first_name,cust_last_name,cust_gender
           from mv_my_customers limit 3;
 cust_id | cust_first_name | cust_last_name | cust_gender
-----+-----+-----+-----
   49671 | Abigail         | Ruddy          | M
   3228  | Abigail         | Ruddy          | M
   6783  | Abigail         | Ruddy          | M
(3 rows)

```

Listing 17: Daten lokal in PostgreSQL speichern


```
postgres=# insert into oratarget.customers
      values ( 1111111,'soug','soug','F',2020,'married','soug',2800
            , <bern>,51442,>bern>,52610,52770,>519-236-6123>
            , <G: 130,000 - 149,999>,1500,>soug@soug.ch>
            , 'total',52772,null,'1998-01-01 00:00:00',null,'I');
INSERT 0 1
postgres=# select count(*)
      from oratarget.customers
      where cust_id = 1111111;

 count
-----
      1
(1 row)
```

Listing 18: Daten in die Oracle-Datenbank zurückschreiben

```
postgres=# alter foreign table oratarget.customers options ( prefetch '1000' );
ALTER FOREIGN TABLE
```

Listing 19: Konfigurieren des Prefetching auf Tabellenbasis

den zwei oder mehr „Foreign Tables“ „gejoint“, so wird dafür gesorgt, dass der Join auf der Oracle-Seite und nicht auf der PostgreSQL-Seite stattfindet, was weniger effektiv wäre (siehe Listing 22).

Fazit

Dadurch, dass die PostgreSQL-Community Teile von SQL/MED implementiert hat, bietet sich PostgreSQL als Datenintegrationsplattform förmlich an. Über „Foreign Data Wrappers“ können unzählige Fremdsysteme angebunden sowie deren Daten über das Standard-SQL-Interface gelesen und (je nach Foreign Data Wrapper) auch wieder geschrieben werden.

Auch wenn es sich hierbei um keine logische Replikation handelt, so kann doch auf eine sehr einfache Art und Weise mit Daten aus anderen Systemen gearbeitet werden. Durch die Kombination von Oracle und PostgreSQL auf Datenebene können beide Systeme ihre Stärken ausspielen und einer Koexistenz steht nichts mehr im Wege.

Quellen

[1] Die PostgreSQL-Wiki-Seite zu SQL/MED: <https://wiki.postgresql.org/wiki/SQL/MED>

Über den Autor

Daniel Westermann ist Principal Consultant und Technology Leader Open Infrastructure bei dbi services.

```
postgres=# alter foreign table oratarget.customers options(readonly
      'true');
ALTER FOREIGN TABLE
postgres=# update oratarget.customers
      set cust_first_name = 'soug2'
      where cust_id = 1111111;
ERROR: foreign table "customers" does not allow updates
postgres=#
```

Listing 20: Read-only foreign tables

```
postgres=# analyze verbose oratarget.customers;
postgres=# select attname,avg_width,n_distinct
      from pg_stats
      where tablename = 'customers' and schemaname = 'oratarget';
 attname | avg_width | n_distinct
-----+-----+-----
 cust_id |          6 |          -1
 cust_first_name |          6 |         1254
 cust_last_name |          7 |          885
 cust_gender |          2 |           2
```

Listing 21: Sammeln von Optimizer-Statistiken

Als Senior Consultant hat Daniel Westermann mehr als zehn Jahre Erfahrung im Management, im Engineering sowie in der Optimierung von Datenbanken und Infrastrukturen, insbesondere mit Oracle und PostgreSQL. Daniel hat sich von Anfang an auf Oracle-Technologien spezialisiert, zum Beispiel für Performanceoptimierung und Tuning, Standardisierung, Backup & Recovery, in Hochverfügbarkeitslösungen wie mit Oracle Real Application Clusters (RAC), Oracle Data Guard, Oracle Grid Infrastructure sowie in Storage-Technologien wie

Oracle Automatic Storage Management (ASM). Daniel Westermann ist Oracle Certified Professional 10g/11g/12c.

Mit der Zeit hat sich Daniel immer mehr im Bereich der Open-Source-Technologien engagiert und sich zum „Technology Leader Open Infrastructure“ und PostgreSQL-Experten entwickelt. Auf Community Tools basierend oder mit denen der EnterpriseDB, entwickelt und installiert er komplexe Lösungen im Bereich der Hochverfügbarkeit mit PostgreSQL. Daniel ist zertifiziert für PostgreSQL Plus 9.0 Profes-

```

postgres=# explain (analyze,verbose)
           select oratarget.customers.cust_first_name
              , oratarget.customers.cust_last_name
              , oratarget.sales.prod_id
              , oratarget.sales.time_id
           from oratarget.customers
           join oratarget.sales
           on oratarget.customers.cust_id = oratarget.sales.cust_id
           where oratarget.customers.cust_id = 6783;

Foreign Scan  (cost=10000.00..20000.00 rows=1000 width=53)
  Output: customers.cust_first_name, customers.cust_last_name, sales.prod_id, sales.time_id
  Oracle query: SELECT /*6bc19f5c70777a6c7dd703f7f9cf9dfc*/ r1."CUST_FIRST_NAME", r1."CUST_LAST_NAME",
r2."PROD_ID", r2."TIME_ID" FROM ("SH"."CUSTOMERS" r1 INNER JOIN "SH"."SALES" r>
  Oracle plan:  SELECT STATEMENT
  Oracle plan:    HASH JOIN   (condition "R1"."CUST_ID"="R2"."CUST_ID")
  Oracle plan:      NESTED LOOPS
  Oracle plan:        NESTED LOOPS
  Oracle plan:          STATISTICS COLLECTOR
  Oracle plan:            TABLE ACCESS FULL CUSTOMERS
  Oracle plan:              PARTITION RANGE ALL
  Oracle plan:                BITMAP CONVERSION TO ROWIDS
  Oracle plan:                  BITMAP INDEX SINGLE VALUE SALES_CUST_BIX (condition "R1"."CUST_ID"="R2"."CUST_ID")
  Oracle plan:                    TABLE ACCESS BY LOCAL INDEX ROWID SALES
  Oracle plan:                      PARTITION RANGE ALL
  Oracle plan:                        TABLE ACCESS FULL SALES

```

Listing 22: Join pushdowns

sional und Postgres Advanced Server 9.4 Professional und darf regelmäßig bei PostgreSQL-Konferenzen in der Schweiz und in Europa referieren.

Vor seinem Eintritt bei dbi services war Daniel Westermann Management System Engineer bei LC SYSTEMS-Engineering AG in Basel. Zuvor arbeitete er als Oracle-Ent-

wickler und Projektleiter bei Delta Energy Solutions AG in Basel (heute Powel AG).

Daniel Westermann verfügt über ein Wirtschafts-Informatik-Diplom (DHBW, Deutschland).

Seine Branchenerfahrung umfasst Pharma, Banken, Energie, Lotterie, Telekommunikation.



Daniel Westermann
daniel.westermann@dbi-services.com

Microservices – Immer die richtige Wahl?

Lisa Damerow, DOAG-Redaktion

Microservices-Architekturen liegen voll im Trend. Dabei entstehen anstelle einer monolithischen Anwendung einzelne, unabhängige Prozesse, die über sprachunabhängige Programmierschnittstellen miteinander kommunizieren.

Jeder Prozess ist ein einzelner „Microservice“ und erfüllt eine bestimmte Aufgabe. Der Vorteil daran ist, dass jede Komponente einzeln aktualisiert, deploy und skaliert werden kann. So sind erhöhte Flexibilität, Resilienz und Agilität nur einige der vielen Vorteile, die einer Microservices-Architektur zugesprochen werden.

Das klingt sehr ansprechend und viele sind daran interessiert, eine solche Architektur umzusetzen. Dennoch sind Microservices nicht immer die beste Lösung. Jürgen Lampe gibt Ihnen

in seinem Artikel „Drum prüfe, wer sich bindet“ in der Java aktuell 5/20 (siehe <https://backoffice.doag.org/formes/pubfiles/12430289/docs/Publikationen/Java-Aktuell/2020/05-2020/05-2020-Java-aktuell-WEB.pdf>) ab Seite 12 eine Checkliste an die Hand, die Sie unbedingt vor der Umsetzung beachten sollten, damit der Traum von Microservices nicht zum Alptraum wird. Denn eine Zurückstellung auf eine monolithische Architektur ist nur mit erheblichem Aufwand nötig. Deshalb beleuchtet er in seinem umfangreichen Artikel wichtige Aspekte und Fallstricke.



JSON Type in der Oracle-Datenbank und Autonomous JSON Database

Beda Hammerschmidt, Oracle Deutschland

JSON wird bei der Entwicklung von neuen Anwendungen immer beliebter; was als Datenaustauschformat zwischen Webserver und Browser begann, findet immer häufiger den Weg in die Datenbank. Dieser Artikel beleuchtet, welche Anforderungen, Vorteile und Probleme sich daraus ergeben, Daten nicht normalisiert in Tabellen, sondern stattdessen im JSON-Format zu speichern und abzufragen.

JSON ist ein einfaches Format, um Objekte einer Programmiersprache in einen String zu serialisieren. JSON ist zwar syntaktisch anders als XML, diesem aber nicht unähnlich: Beides sind hierarchische Datenmodelle, menschenlesbar und selbstbeschreibend. JSON gilt jedoch als wesentlich einfacher und ersetzt somit XML in vielen Bereichen (z.B. JSON REST statt XML Webservice).

Was ist JSON?

Wir beschreiben JSON an folgendem Beispiel (*siehe Listing 1*). Wer JSON kennt, kann diesen Absatz gerne überspringen.

JSON basiert auf zwei Strukturen: dem Objekt {...} und dem Array [...]. In einem Objekt werden Key-Value-Paare ohne Reihenfolge zusammengefasst, der Key identifiziert den Wert (z.B. ‚age‘ hat den

Wert 25). Keys sind immer String, ihnen folgt ein Doppelpunkt und ein dem Key zugeordneter Wert – dieser ist entweder skalar oder komplex (Objekt oder Array). Bei skalaren Werten wird der Datentyp durch die Syntax festgelegt (25 ist eine Nummer, „25“ ein String). Die Datentypen sind auf Nummer, String, Boolean und null reduziert. Leider gibt es kein Datum oder Timestamp, sodass diese in der Re-

gel als String abgebildet werden müssen (siehe ‚lastUpdated‘). Innerhalb eines Arrays definiert die Position den Wert. Das Array gleicht somit einer Liste, das Object einer Hashtable/Map. Arrays und Objekte lassen sich schachteln, um beliebig komplizierte Hierarchien zu bilden.

Anders als bei XML gibt es in der Regel kein Schema, das JSON-Daten definiert. Auch bietet die JSON-Syntax mehr Informationen als XML. Der Wert [„1“] beschreibt eine Liste mit einem String. Beim XML-Äquivalent <x>1<x> müsste man ins XML-Schema schauen, um dieselben Informationen zu bestimmen.

Warum JSON in der Datenbank?

Das Speichern von JSON wurde durch NoSQL-Produkte wie MongoDB populär. Heute unterstützen viele relationale Datenbanken ebenfalls JSON, wenn auch in einem unterschiedlichen Maß. Die folgenden Gründe sprechen für JSON:

1. **Schema-Flexibilität:** Änderungen der Anwendung lassen sich leicht mit JSON realisieren: So lässt sich etwa ein ‚middleName‘ als Beispiel einfügen. Dies wäre mit einem `Alter Table Add Column` relational noch leicht durchzuführen – aber oft muss der Entwickler solch eine Änderung mit einem DBA koordinieren. JSON ermöglicht auch Kardinalitäts-Änderungen – sollen im Beispiel mehrere Adressen pro Person unterstützt werden, so wird aus dem Wert für ‚address‘ einfach ein Array, das beliebig viele Adressen beinhalten kann. Dies wäre relational vermutlich mit einer neuen Tabelle verbunden, sodass Anfragen und Updates angepasst werden müssten. JSON entkoppelt somit Änderungen an der Anwendung vom Datenbankschema.
2. **Keine Normalisierung:** Der Entwickler ‚denkt‘ häufig in Objekten der Anwendung, zum Beispiel ‚Kunde‘, ‚Vertrag‘, ‚Bestellung‘. Diese lassen sich meist leicht in ein JSON-Objekt überführen, während bei einer Normalisierung oft mehrere Tabellen benötigt werden. Folglich involvieren selbst einfache Operationen SQL über mehrere Tabellen, was von (jüngeren) Entwicklern häufig als unnötig kom-

plex wahrgenommen wird. ORM-Tools wie Hibernate mögen das Mapping vereinfachen, werden jedoch auch oft als Hindernis und inflexibel empfunden. JSON zur Vermeidung von Normalisierung kann auch zu Performanz-Verbesserungen führen, wenn ein JSON-Zugriff Joins über viele Tabellen erspart.

3. **Einfache APIs:** Entwickler mögen Datenbank-Operationen direkt von der Programmiersprache aufrufen, ohne erst SQL Strings zusammenzubauen. Diese sogenannten ‚JSON Document Store‘ APIs stammen von NoSQL-Produkten, die ihren Mangel einer deklarativen Anfragesprache wie SQL so erfolgreich als Vorteil verkaufen konnten.

oder analytische Anfragen sein. Gerade bei NoSQL-Datenbanken kann das dazu führen, dass Daten jetzt in ein anderes System geladen werden müssen.

- JSON kennt keine Referenzen – Primary/Foreign-Key-Constraints gibt es erst mal nicht, können in Oracle durch Virtual Columns jedoch nachgebildet werden. In einer NoSQL-Datenbank hat man diese Lösung nicht und muss versuchen, die Daten-Integrität im Applikationscode zu gewährleisten.
- NoSQL-Datenbanken haben oft weitere Einschränkungen: So fehlen oft Transaktionen, Joins und andere Datenbank-Basics, die dann von Hand im Applikationscode nachgebaut werden müssen.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": { "street": "21 2nd Street",
              "city": "New York", "state": "NY",
              "postalCode": "10021", "isBusiness": false },
  "phoneNumbers": [ { "type": "home", "number": "212 555-1234" },
                    { "type": "mobile", "number": "646 555-4567" } ],
  "lastUpdated": "2019-05-13T13:03:35+0000"
}
```

Listing 1: JSON-Objekt mit Daten einer Person

Wenngleich hier eine Menge Vorteile von JSON gegenüber klassischen Tabellen genannt wurden, so müssen die Konsequenzen und Nachteile jedoch auch klar benannt werden. Diese sind häufig dem Entwickler nicht umfänglich bewusst (oder zu Beginn der Entwicklung noch nicht relevant), können jedoch zu gravierenden Problemen im späteren Betrieb führen:

- Die Schema-Flexibilität von JSON bedeutet nicht, dass kein Schema vorhanden ist, sondern eher, dass das Schema implizit im Applikationscode verteilt ist. Schema-Änderungen werden jetzt nicht mehr von der Datenbank zentral ‚kontrolliert‘, sondern Applikationscode muss gegebenenfalls angepasst werden, was Wartung nicht vermeidet, sondern nur verschiebt.
- Ein Schema, das gut für einen Anwendungsfall geeignet ist (zum Beispiel alle Daten eines Kunden finden), mag schlecht für andere Fälle wie Reporting

Was bietet Oracle?

Oracle hat erkannt, dass JSON bei Entwicklern sehr beliebt ist, aber dass es wenig Sinn ergibt, sich die Flexibilität von JSON mit dem Verlust von wichtigen relationalen Features zu erkaufen. Bereits 2013 haben Oracle und IBM den offenen SQL/JSON-Standard entworfen, der später im SQL-2016-Standard aufging. Oracle brachte 2014 mit Release 12.1.0.2 die erste Datenbank mit SQL/JSON-Funktionen heraus. (Der Autor hatte nicht nur die Ehre am Standard mitzuarbeiten, sondern auch – wie es bei Oracle üblich ist – die neuen Funktionen wie `JSON_VALUE` oder `JSON_TABLE` in C zu implementieren und die Bugs zu beheben.) In der aktuellen Datenbankversion 19c gehen die Funktionalitäten weit über die SQL/JSON-Funktionen hinaus:

Oracles Ansatz ist es, JSON nicht als Alternative zum relationalen Modell zu etablieren, sondern die Vorteile beider Model-

le je nach Bedarf kombinieren zu können. Konkret bedeutet dies, a) dass JSON in normalen Tabellen gespeichert wird, b) dass relationale Spalten und JSON-Spalten beliebig kombinierbar sind und c) dass sich JSON-Daten ins relationale Modell umwandeln lassen und sich umgekehrt aus relationalen Daten JSON generieren lässt. *Listing 2* zeigt, wie das JSON-Beispiel mittels `JSON_TABLE` ins relationale Modell überführt wird. Man beachte, dass aus einer JSON-Eingabe zwei Zeilen generiert werden, da über das Array mit den Telefonnummern iteriert wird (Nested Path)

Wie leicht es ist, aus relationalen Daten JSON zu erzeugen, illustriert *Listing 3*, das zwei Zeilen der emp-Tabelle als zwei JSON-Objekte in einem Array zurückliefert:

JSON und Tabellen lassen sich also leicht von einer Darstellung in die andere und zurück verwandeln, sodass man von verschiedenen Sichten auf dieselben Daten sprechen kann. JSON und relational sind somit kein Entweder-oder, sondern lassen sich bedarfsgerecht kombinieren. Insbesondere ist der volle Umfang von SQL auf JSON-Daten anwendbar.

SODA Document Store API

Die im SQL/JSON-Standard definierten Funktionen wie `JSON_TABLE` oder `JSON_OBJECT` sind Erweiterungen der SQL-Sprache. Entwickler mit NoSQL-Hintergrund

```
select id, jt.*
from custData, JSON_TABLE(jcol, '$' columns (
  "first"          path '$.firstName',
  "age" number      path '$.age',
  "state" varchar2(2) path '$.address.state',
  NESTED PATH '$.phoneNumbers[*]' columns(
    phone" path '$.number') )) jt;
```

ID	first	age	state	phone
1	John	25	NY	212-555-1234
1	John	25	NY	646-555-4567

Listing 2: JSON_Table mit Ergebnis

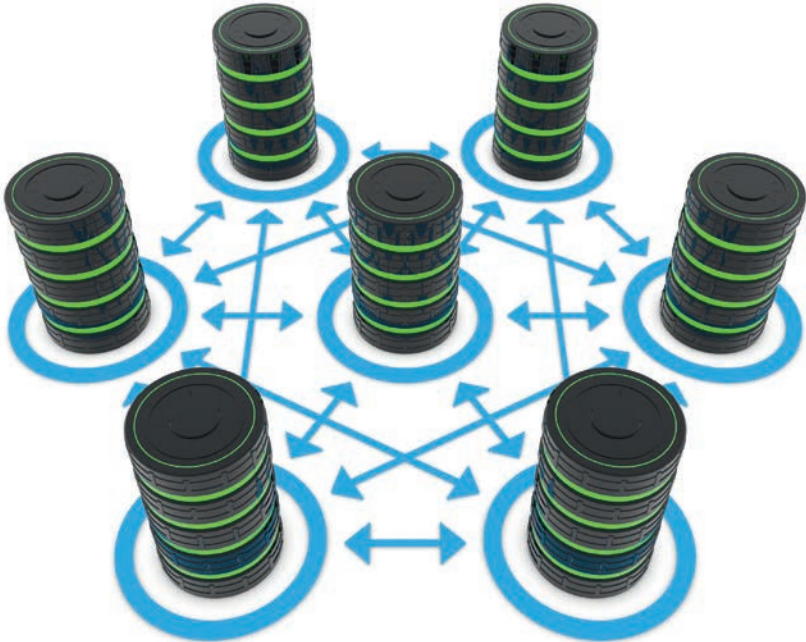
```
SELECT JSON_ARRAYAGG (JSON_OBJECT (*))
from emp WHERE empno > 7900;
```

```
[{"EMPNO":7902,"ENAME":"FORD","JOB":"ANALYST","MGR":7566,...},
 {"EMPNO":7934,"ENAME":"MILLER","JOB":"CLERK","MGR":7782,...}]
```

Listing 3: JSON-Erzeugung aus relationalen Daten

wissen Document Store APIs zu schätzen – insbesondere MongoDB hat hier das Bewusstsein für eine einfach zu nutzende Datenbank-Schnittstelle geschaffen. Oracle hat mit SODA (Simple Oracle Document Access) ein ähnliches API geschaffen, das direkt in Java, Python, JavaScript (nodeJs), PL/SQL, aber auch REST und Sql-Cl verfügbar ist. Die Idee von SODA ist es, einfache Operationen (Finden, Einfügen, Löschen, Ersetzen) direkt von der Programmiersprache aufzurufen – der SO-

DA-Treiber ersetzt diese dann mit einem äquivalenten SQL Statement, das an die Datenbank gesendet wird. SODA ist also ein NoSQL-API auf einer SQL-Datenbank. *Listing 4* zeigt, wie eine JSON-Collection mit dem Namen ‚cities‘ angelegt wird und wie zwei JSON-Datensätze eingefügt werden. Die ‚soda get‘-Operation hat einen Filterausdruck (`{„population“:{„$gt“:250000}}`) und selektiert alle Städte, deren Einwohnerzahl größer als 250.000 ist.



```
soda create cities;
soda insert cities {"name":"San Jose","population":1021795,"county":"Santa Clara"}
soda insert cities {"name":"Atlanta","population":506811,"county":["Fulton", "DeKalb"]}

soda get cities -f {"population":{"$gt":250000}}
```

Listing 4: SODA-Operationen zum Erzeugen einer Collection, Einfügen von JSON-Daten und Suche mit JSON-Filterausdruck

Hervorzuheben ist, dass SODA all diese Operationen automatisch in äquivalente SQL Statements übersetzt (CREATE TABLE, INSERT, SELECT.. WHERE). Das Beispiel nutzt SqlCl, die Syntax ist in anderen Sprachen wie Java oder JavaScript sehr ähnlich, sodass sich SODA-Kenntnisse leicht übertragen lassen. Die JSON-Collection ‚cities‘ ist nichts anderes als eine System-generierte Tabelle mit einer JSON-Spalte – sie lässt sich ohne Weiteres mit SQL (z.B. JSON_TABLE) abfragen.

JSON-Type mit Binärformat

JSON ist in der Regel ein String (UTF8 Unicode), der bei Datenbank-Zugriffen erst geparkt werden muss. Es gibt keine Möglichkeit, irrelevante Werte zu überspringen. Updates erfordern, dass der JSON-String neu erstellt werden muss. Aus diesem Grund hat Oracle in neueren Releases ein binäres JSON-Format und einen neuen SQL-Datentyp ‚JSON‘ erstellt. Das Binärformat ist für schnelle Anfragen und Updates optimiert: Offsets erlauben es, direkt zu selektierten Daten zu springen; Updates können partiell durchgeführt werden und reduzieren so IO und Logs. Aus Benutzersicht ändert sich wenig, da alle Optimierungen transparent stattfinden. Das Binärformat (Format OSON) ist in allen 19c-Cloud-Datenbanken verfügbar. JSON Type kann im 20c Preview Release evaluiert werden [1].

Autonomous JSON Database

Als neuestes Mitglied der Oracle Autonomous Databases gibt es seit August 2020 die Autonomous JSON Database (AJD) – einen für JSON Workloads optimierten Cloud-Service. AJD hat denselben Funktionsumfang wie Autonomous Transaction Processing (ATP), wird jedoch preislich deutlich günstiger angeboten, um

neue Entwickler anzuziehen. Im Gegensatz ist AJD auf 20GB Nicht-JSON-Daten begrenzt, sodass rein relationale Workloads weiterhin ATP oder ADW nutzen. Details und auch Performance-Vergleiche mit MongoDB Atlas finden sich in [2], [3] und [4].

Fazit

Während gerade (jüngere) Entwickler sich von Tabellen abwenden und JSON als primäres Datenmodell bevorzugen (JavaScript im Browser, Node.js im Server und JSON in der Datenbank), so geht es hingegen bei erfolgreichen Projekten vielmehr darum, die Stärken beider Modelle zu kombinieren. SQL mag für einfache Operationen den Document Store APIs unterlegen sein, spätestens bei komplexeren Anfragen ist es jedoch eindeutig besser (und dank Optimizer auch schneller). Bewährte relationale Konzepte wie (Materialized) Views, Partitionen, Transaktionen, Stored Procedures sind natürlich auch relevant, wenn die Daten im JSON-Format vorliegen. Mit Oracle als führende relationale JSON-Datenbank ist der Nutzer auf der sicheren Seite: SQL und NoSQL-API lassen sich gemeinsam mit denselben Daten nutzen – ein von SODA eingefügtes JSON-Dokument kann sofort mit SQL relational weiterverarbeitet werden und relationale Ergebnisse lassen sich per JSON-Erzeugung auch SODA-Clients zur Verfügung stellen. Ein Oracle-DBA kann gesammeltes Wissen auch mit JSON weiter anwenden.

Quellen

- [1] <https://blogs.oracle.com/jsondb/20cpreview>
- [2] <https://www.oracle.com/autonomous-database/autonomous-json-database/>
- [3] <https://blogs.oracle.com/jsondb/autonomous-json-database>
- [4] <https://wikibon.com/oracle-autonomous-json-takes-on-document-dbs/>

Über den Autor

Beda Hammerschmidt ist seit 2006 Entwickler im Oracle-Datenbank-Team und beschäftigt sich seit 2013 mit verschiedenen JSON-Funktionalitäten und dem SQL/JSON-Standard.



Beda Hammerschmidt
beda.hammerschmidt@oracle.com



The Song of Fire and Ice – Oracle ADW vs. Snowflake

Dr.-Ing. Holger Friedrich, sumIT

Nicht nur dank viralen Marketings, sondern auch durch beeindruckende Leistungsdaten und innovative Architektur ist Snowflake eine ernstzunehmende Konkurrenz für Oracles Autonomous Data Warehouse im Bereich des Cloud Data Warehousing. Beide Produkte haben spezifische Stärken und Schwächen bezüglich ihrer Funktionalität, des Handlings und der Einsatzszenarien. Diese zu kennen ist Voraussetzung für die richtige Wahl.

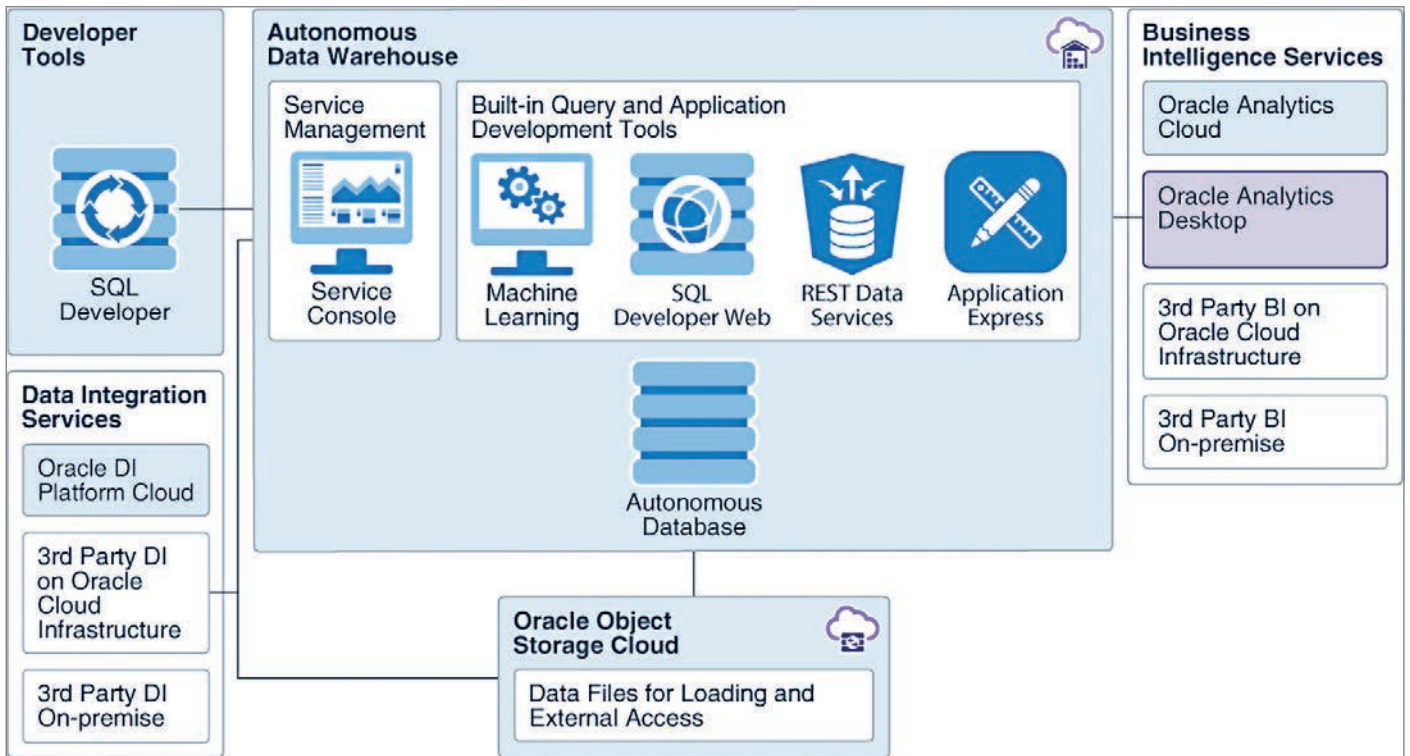


Abbildung 1: Architektur von ADW in der Oracle Cloud (© Oracle)

Oftmals pflegen Oracle-Kunden zu den genutzten Produkten eher eine Vernunft- denn eine Liebesbeziehung. Datenbank und Optionen werden als teuer wahrgenommen, aber als technisch überlegen geschätzt. Deshalb werden ihnen kritische Arbeiten und Daten anvertraut. Seit Beginn der Cloud-Ära treten jedoch neue Wettbewerber und Produkte in großer Zahl in den Markt ein. Sie versprechen nicht nur unbegrenzte Skalierbarkeit, Performanz und hohe Sicherheit, sondern sind auch preislich attraktiv. Herausragend unter diesen neuen Konkurrenten im Cloud-Warehousing ist Snowflake.

Die Kontrahenten teilen einige Gemeinsamkeiten, unterscheiden sich allerdings signifikant in ihrer Architektur, dem Tooling darum herum, Feature-Sets und Handhabung. Diese Unterschiede determinieren ihre Eignung für bestimmte Anwendungsfälle.

Abbildung 1 zeigt die Architektur von Oracle ADW und die Einbindung in die Oracle Cloud. Im Zentrum steht eine Oracle-Datenbank auf Exadata-Infrastruktur. Darum herum sind diverse zusätzliche Tools und Services wie Machine-Learning- Notebooks, Data Safe, ein Middle-Tier für APEX und REST-APIs und weitere dem Service zugeordnet. Im Bereich Datenintegration wurde Oracle DI

Platform Cloud kürzlich durch zwei Dienste, Oracle Data Integrator und OCI Data Integration ersetzt.

Abbildung 2 dagegen zeigt die Snowflake-Architektur. Im Gegensatz zu ADW sind hier Storage und Compute getrennt und vollständig unabhängig voneinander skalierbar. Alle Metadaten, Nutzerverwaltung und Management des Dienstes sind im übergreifenden Service-Layer pro Region zentralisiert.

Ease of Use, Verwaltung, Wartung und Pflege

Einfache Bedienung, sehr hohe Agilität, hoher Automatisierungsgrad und Self-Service-Funktionalität sind die großen Versprechen des Cloud-Data-Warehousing. Snowflake hat hier durch seine Cloud-native Architektur leichte Vorteile in Sachen Einfachheit und Komfort, während ADW durch die vollständige Integration in das Oracle-Cloud-Management punktet, obwohl die Basis-Engine bereits einige Jahrzehnte im Einsatz ist.

In der Oracle Cloud müssen mehrere verschiedene Technologien zur Verwaltung und für den Betrieb von ADW-Instanzen eingesetzt werden. Namentlich sind dies SQL, PL/SQL innerhalb der ADWs und REST-Calls (siehe Listing 1), zum

Beispiel mittels Terraform-Scripting für die Verwaltung der Instanzen. Dies ist einerseits etwas schwieriger, da mehrere Technologien gemeistert werden müssen, andererseits aber im Sinne einer Enterprise-weiten Bewirtschaftung praktisch, da die gleichen Sprachen und Tools für die gesamte Infrastruktur verwendet werden, also User Management, Netzwerke etc. Das DWH ist somit kein Sonderfall im Unternehmen.

Bei Snowflake dagegen ist SQL der Schlüssel für alle administrativen Operationen (siehe Listing 2). Erzeugung von Datenbanken, Compute-Nodes, Benutzermanagement und alle anderen administrativen Aufgaben werden per SQL ausgelöst. Dies erlaubt es einem DWH-Team, seine ganze Umgebung inklusive des Datenbank-Layers ohne Hilfe anderer Gruppen zu verwalten. Schnelleres und autonomeres Arbeiten wird so erleichtert. Auf der anderen Seite sind für die Absicherung, Datenintegration, Netzwerk etc. Kenntnisse über und Werkzeuge der jeweiligen Cloud zu nutzen, was im Gegensatz zu ADW eine vollständig andere Umgebung darstellt.

Neben dem administrativen Tooling gestaltet sich die Wartung und Pflege der beiden Services ähnlich. Bei beiden wird regelmäßig gepatcht und werden neue

Funktionen frei geschaltet. Während dies bei Snowflake ein stetiger Fluss ist, bei dem neue Features während recht kurzer Testphasen zur Verfügung stehen, geht Oracle eher einen traditionellen, aber Enterprise-tauglichen Weg. Patching und diverse Funktionen werden kontinuierlich eingeführt, große Veränderungen stehen den Kunden jedoch als Versionsupgrade einige Monate für Tests und selbstbestimmte Upgrades zur Verfügung, bevor sie verpflichtend ausgeführt werden.

Snowflake bietet größere Flexibilität, was Cloning und Sharing von Daten angeht, da beides auch auf Schema und Objektebene funktioniert und nicht wie bei ADW ausschließlich ganze Datenbanken geklont werden können. Auch ist das Cloning bei Snowflake in wenigen Augenblicken durchgeführt. Grund dafür ist die Cloud-native Implementierung auf einem Object-Store, bei dem keine Veränderungen an Datenbankdateien gemacht werden können. Ein Klon bedeutet damit zunächst nur das Anlegen administrativer Pointer für eine Gruppe von Dateiversionen im Snowflakes Service-Layer. Klon und Quelle divergieren erst ab den ersten

Änderungen in einer der Datenbanken, da dann neue Files geschrieben werden.

Skalierbarkeit

Skalierbarkeit wird bei beiden Plattformen auf unterschiedliche Art und Weise erreicht. Oracle ADW läuft als PDBs auf Standard-Exadata. Skalierbarkeit kann dort auf zwei Arten erfolgen, mittels einer Zuweisung von mehr Rechenleistung auf einem Exa-Server oder durch Hinzufügen neuer Datenbankinstanzen auf anderen Exa-Servern. Diese Architektur beschränkt die erreichbare Maximalgröße, die jedoch ohnehin kaum von einem Kunden mit einer einzelnen PDB erreicht werden dürfte. Andererseits ist die Skalierung elastisch, ohne spürbare Serviceunterbrüche. Das Auto-Scale-Feature des ADW stellt zudem eine Auto-Skalierung bis zur maximal dreifachen Rechenleistung bei sekundengenaue Abrechnung bereit.

Snowflake dagegen kann asymmetrisch mit einer beliebigen Menge von Compute-Shapes skalieren. Ein sogenanntes Virtual Warehouse kann dabei aus einem einzel-

nen Shape von bis zu 128 Nodes oder aus einem Cluster von Shapes gleicher Größe bestehen. Verschiedene Virtual Warehouses sind für verschiedene Workloads gedacht. Scale-up eines Shape zu mehr Nodes ist online möglich, um Querys mit mehr Rechenleistung zu versorgen, führt aber zuweilen zu spürbaren Verlängerungen gerade laufender Abfragen. Scale-out durch mehrere Shapes in einem Cluster dient der nebenläufigen Verarbeitung einer höheren Anzahl von Abfragen. Snowflake stellt Auto-Scale-out zur Verfügung, um Spitzen großer Abfragemengen zu bewältigen.

Leistung

Die Leistung beider Plattformen für analytische Workloads ist sehr gut. Snowflake profitiert dabei von drei Eigenschaften.

1. Spaltenorientierte Speicherung
2. Lokales Caching bei den Compute Nodes auf Flash-Storage
3. Keinerlei Aufwand für Indexmanagement, Constraint-Prüfung, Undo-Management oder Ähnliches

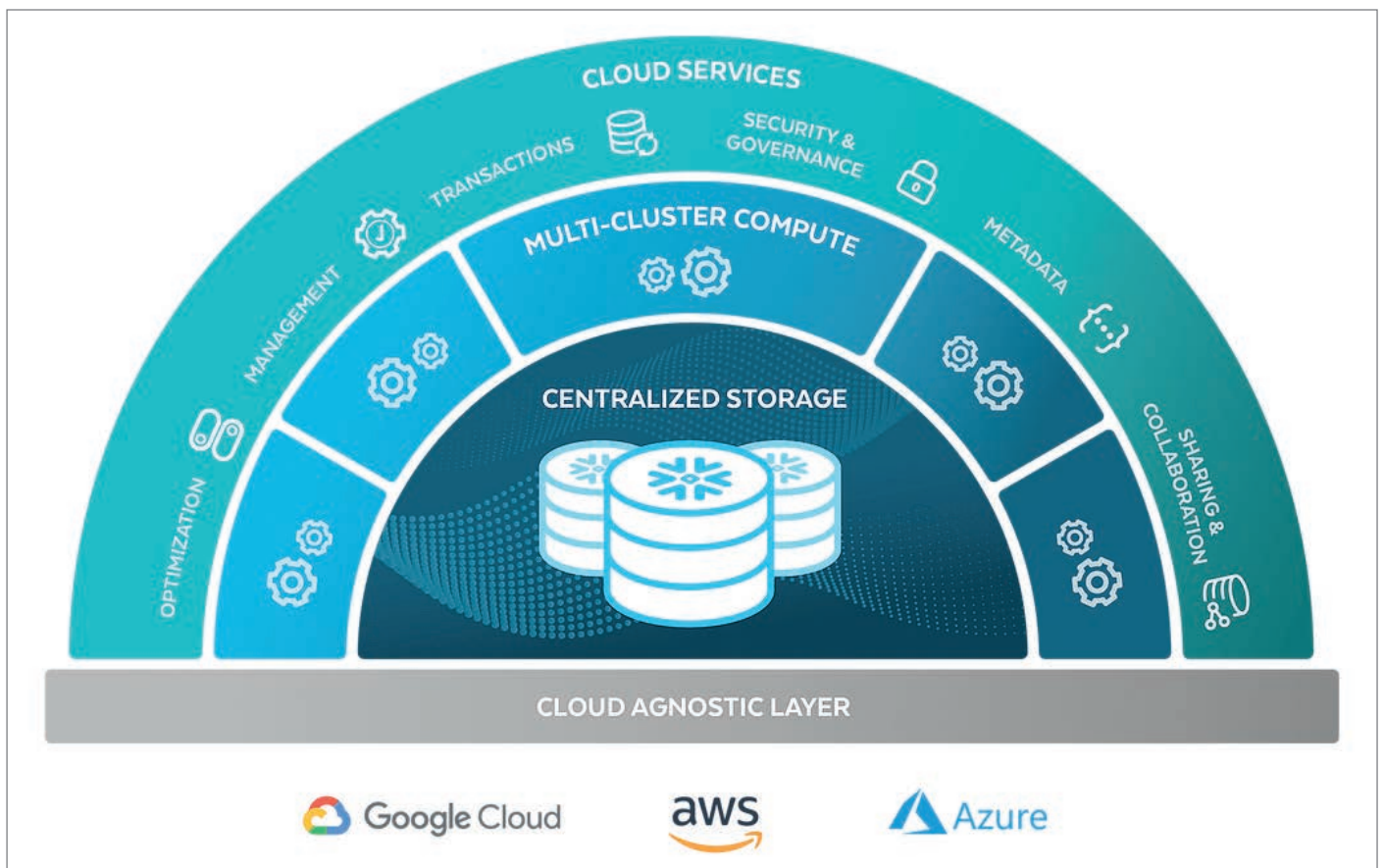


Abbildung 2: Architektur des Snowflake-Dienstes (© Snowflake)

Hier zahlt sich aus, dass Snowflake konsequent auf die Erstellung, Wartung und Prüfung von Primary-, Unique- und Foreign-Key-Constraints verzichtet. Werden Constraints jedoch gewünscht oder gar gebraucht, um Datenqualität, zum Beispiel für regulatorische Berichte, sicherzustellen oder für andere Zwecke, dann ist Snowflake ungeeignet. Fast alle "modernen" Cloud-DWH-Dienste, wie etwa MS Synapse und Google BigQuery, haben dieselbe Einschränkung bezüglich fehlender Constraints. Im Juni 2020 hat Snowflake ein "Search-Optimisation"-Feature angekündigt, um den Zugriff auf kleine Datenmengen zu beschleunigen. Die Implementierung ist verborgen, aber es sieht danach aus, als würden auf Knopfdruck Single-Column-Non-Unique-Indizes erzeugt. Welche Auswirkungen dieses Feature auf die Performance haben wird, bleibt abzuwarten.

Oracle ADWs Performance ist – wie zu erwarten – ebenfalls sehr gut. Es ist eine Oracle-Datenbank, die auf einer modernen Exadata-Generation läuft. Sie nutzt

- Hybrid-Columnar-Compression, somit auch spaltenorientierte Speicherung,
- große Mengen Flash-Storage im Storage-Layer und
- Query-Offloading zu Compute-Ressourcen im Storage-Layer.

Der Grad der Performance kann zudem vom Kunden beeinflusst werden. Werden aktive Constraints genutzt, muss der Rechenoverhead dafür geleistet werden, sonst nicht. Allerdings hat der Kunde bei ADW die Wahl, die ihm bei Snowflake nicht gegeben ist. Zudem ist es möglich, mit Indizes Zugriffe auf kleine Datenmengen gezielt zu beschleunigen.

Features

Im Bereich der Features, also der funktionalen Produkteigenschaften, gibt es große Unterschiede zwischen beiden Diens-

```
use role sysadmin;
create or replace database MYSF01_DB comment='meine SF db';
create or replace WAREHOUSE MYSF01_WH
  with WAREHOUSE_SIZE = 'XSMALL' WAREHOUSE_TYPE = 'STANDARD'
      AUTO_SUSPEND = 600 AUTO_RESUME = TRUE
      COMMENT = 'mein SF Virtual Warehouse';
```

Listing 2: Erstellen einer Snowflake-Datenbank und von Compute-Knoten mittels SQL

```
#Erstellen einer neuen autonomen Datenbank:
POST /20160918/autonomousDatabases
Host: database.us-phoenix-1.oraclecloud.com
<authorization and other headers>
{
  "compartmentId"      : "ocid.compartment.oc1..<unique_ID>",
  "displayName"       : "meine_autonome_db",
  "dbName"            : "MYADW01",
  "adminPassword"     : "<password>",
  "cpuCoreCount"      : 8,
  "dataStorageSizeInTBs" : 2,
  "dbWorkload"        : "DW"
}
```

Listing 1: Erstellen einer ADB mittels REST-Call

ten. Sie resultieren einerseits aus der Reife und dem Alter des Produkts und andererseits aus der grundlegenden Architektur.

Oracle Autonomous Data Warehouse bietet eine Reihe von Features, die Snowflake vermissen lässt oder bislang nur in recht eingeschränkter Form implementiert. Beispiele hierfür sind:

- Vollständige Enterprise-Level-Programmierungsumgebung mit PL/SQL, Edition-Based Redefinition und weiteren Funktionen
- Eine Vielzahl spezialisierter, integrierter Engines mit XML-DB, Machine Learning, Spatial, Graph etc.
- Feingranulare Access-Steuerung auf Datenebene mit Virtual Private Database und Data Masking
- Weitreichende Kontrolle über die Ausführung von Abfragen durch Hints
- Performancefeature und -strukturen wie Partitioning, Materialized und Analytic Views, Query Pattern Matching, Indizes und andere
- Application Express (APEX) Low-code-Applikationsentwicklungsumgebung
- REST-APIs für den Daten-Access und zu APEX-Applikationen mit enthaltenen Oracle Rest Data Services (ORDS)

Darüber hinaus kam im vergangenen Jahr auch eine vertiefte Integration mit ande-

ren Oracle-Cloud-Diensten hinzu. Ein Beispiel ist die übergreifende Verwaltung der Sicherheit einer Flotte von ADW-Instanzen mit dem Dienst Data-Safe. Die Anbindung solcher übergreifender und integrierender Clouddienste ist für Oracle mit ADW möglich, da der Dienst ausschließlich in der Oracle Cloud angeboten wird und Oracle selbst diese und die Dienste darin programmiert.

Snowflake implementiert im klassischen Feature-Bereich eifrig hinzu. Es gibt mittlerweile mit der „Search Optimization“ Indizes, mit Data Masking spaltenbasierte Sichtbarkeitsbeschränkung und mit JavaScript eine erste, noch recht primitive Programmierungsumgebung. Verbesserte Programmierbarkeit von DWH-Applikationen durch Unterstützung anderer Programmiersprachen ist mittlerweile angekündigt.

Hängt Snowflake im klassischen Datenbank-/Data-Warehouse-Bereich mit Features hinterher, so gibt es andererseits eine Reihe neuer Eigenschaften, die der Dienst ADW voraushat. Dazu gehören:

- Read-only Sharing von Strukturen, Datenbanken oder ganzen Schemata
- Daten(-bank)replikation über mehrere Cloud-Regionen und Clouds hinweg
- Time-Travel-Funktionalität für temporäre Querys über viele Tage hinweg
- Undrop von einzelnen Strukturen, Schemata und ganzen Datenbanken
- Kontinuierliche Datenpipelines mit Snow-Pipe zum automatischen Laden neu eintreffender Daten aus Data-Lakes
- Zentrale Nutzer- und Rechteverwaltung für alle Snowflake-Instanzen eines Kontos

Diese Alleinstellungsmerkmale von Snowflake gegenüber ADW sind der Nutzung

der Cloud-nativen Architektur mit zentralem administrativen Layer und Daten-Dictionary (siehe Abbildung 2) sowie der Speicherform in Object Storage zu verdanken. Grundsätzlich lohnt es sich, regelmäßig die Webseiten der Hersteller aufzusuchen, um den aktuellen Stand der vorhandenen Funktionen zu prüfen. [1, 2]

Kosten

Der Vergleich, die A-priori-Abschätzung der Kosten verschiedener Cloud-Dienste ist schwierig. Neben den Kosten des DWH-Service selbst spielen auch weitere Faktoren, wie zum Beispiel Netzwerkkosten zur Anbindung von Quellen, eine wichtige Rolle. Hierunter fallen beispielsweise die Anbindung des heimischen On-Premises-Netzes an die Cloud, aber auch Egress-Kosten, sollte eine Quelle in einer anderen Cloud liegen als der Data-Warehouse-Service selbst. Einen Vorteil kann hier in manchen Szenarien Snowflake für sich verbuchen, da man drei Clouds zur Auswahl hat. Geschickte Platzierung des Dienstes hilft, Egress-Kosten einzusparen. Für Oracle gilt dies in der eigenen Cloud und für Quellen in Azure-Rechenzentren, die per direktem Interconnect an die Oracle Cloud angeschlossen sind.

Ein weiterer Kostenpunkt stellt die Datenintegration dar, sollte hierfür ein metadatenbasiertes Werkzeug anstatt manueller Programmierung genutzt werden. Während Snowflake kein eigenes oder fremdes DI-Tooling gratis zur Verfügung stellt, kann man für Oracle ADW zum gegenwärtigen Zeitpunkt Oracle Data Integrator und damit ein Enterprise-Integrationswerkzeug kostenfrei vom Marketplace der Oracle Cloud beziehen und für ADW einsetzen. [3]

Schließlich stellt sich die Frage, wie viel Rechenleistung und Speicherplatz es für wie viel Geld gibt und welche Zeiträume abgerechnet werden. Snowflake rechnet in Nodes, dokumentiert allerdings nicht die Anzahl der vCPUs, der ein Node entspricht. Man handelt ganz nach dem alten Motto von Rolls Royce, dass eine genauere Leistungsangabe nicht notwendig sei, da immer eine überragende Performance bereitstünde. Es bleibt für den Kunden unsicher, wie viele Rechenressourcen er für einen Credit in welcher Cloud oder

Region bei Snowflake bekommt. Ebenso unklar ist, ob diese Ressourcen dann fest zugesichert sind oder Noisy-Neighbour-Effekte auftreten könnten. Preise und Shapes variieren zwischen Clouds und Regionen.

Oracle ist beim Thema Kosten transparent und klar. Eine OCPU in der Cloud entspricht einem Intel Core und damit 2 vCPUs in einer Exadata-Infrastruktur. Die Ressourcen werden per Resource Manager auf der genutzten Exadata-Infrastruktur zugesichert. Der Preis pro OCPU ist in allen Regionen der Welt gleich. Zum Einstieg bietet Oracle im Netz zudem das Cloud-Äquivalent der XE-Datenbankversion im Rahmen der kostenfreien Always-Free-Services an. Dieser Dienst ist bezüglich Leistung und Speicherplatz begrenzt, enthält aber darüber hinaus alle Komponenten des kostenpflichtigen Dienstes. [4]

Fazit

Beide Cloud-DWH-Dienste, Snowflake wie ADW, sind leistungsfähig, effizient, von der Kostenseite her attraktiv und hervorragend im agilen Umfeld einsetzbar. Welcher Dienst der passende für einen Kunden oder ein Projekt ist, hängt von den konkreten Anforderungen, Umssystemen und der gewählten Cloud-Strategie ab.

Ist Datenkonsistenzprüfung durch die Datenbank nicht wichtig, Programmierung in der Datenbank nicht erwünscht und sind stattdessen andere Dinge wie Long-Distance-Replikation, zentrale Rechteverwaltung über viele Instanzen oder flexibles Data Sharing gefragt, ist Snowflake ein hervorragendes Produkt. In diesem Bereich ist Snowflake auch eher in Konkurrenz zu Googles BigQuery zu sehen. Google hat mit BigQuery ebenso noch ein Feature-Defizit, hat jedoch kürzlich Multi-Cloud-Unterstützung mit BigQuery Omni [5] angekündigt.

Werden stattdessen starke Datenkonsistenzprüfung, eine komplette, reife Programmierumgebung, multiple Verarbeitungs-Engines in einem Service und komplexe Sicherheitsmechanismen benötigt, dann führt an Oracle ADW kaum ein Weg vorbei. Nicht zu unterschätzen ist dabei die Möglichkeit, in ADW mittels Hints und detaillierter Analyse von Ablaufplänen auf die Ausführungsstrategi-

en Einfluss zu nehmen. In der Cloud-Welt wird dem Benutzer zwar suggeriert, dass er sich darüber keine Gedanken machen müsse, aber in der Realität muss in der Regel das Maximale mit begrenztem Budget erreicht werden. Im Zweifelsfall ist es daher keine Option, mehr Leistung zu kaufen, wenn ein Verarbeitungsschritt nicht wie gewünscht performt. Stattdessen ist es ein entscheidender Vorteil, wenn man auch in einem voll gemanagten Service in der Lage ist, Detailanalysen von Ausführungsplänen durchzuführen und Ausführungen zu tunen.

Über den Autor

Holger Friedrich ist Geschäftsführer von sumIT AG [6] und Oracle ACE. Seit 20 Jahren ist er auf den Gebieten Data Warehousing, Datenintegration und Business Intelligence tätig.

Quellen

- [1] <https://www.oracle.com/database/technologies/datawarehouse-bigdata/adw-features.html>
- [2] <https://docs.snowflake.com/en/release-notes/new-features.html>
- [3] https://cloudmarketplace.oracle.com/marketplace/de_DE/listing/62627436
- [4] <https://www.oracle.com/cloud/free/>
- [5] <https://cloud.google.com/blog/products/data-analytics/introducing-bigquery-omni>
- [6] <https://www.sumit.ch/>



Dr.-Ing. Dipl.-Inform. Holger Friedrich
Holger.Friedrich@sumit.ch

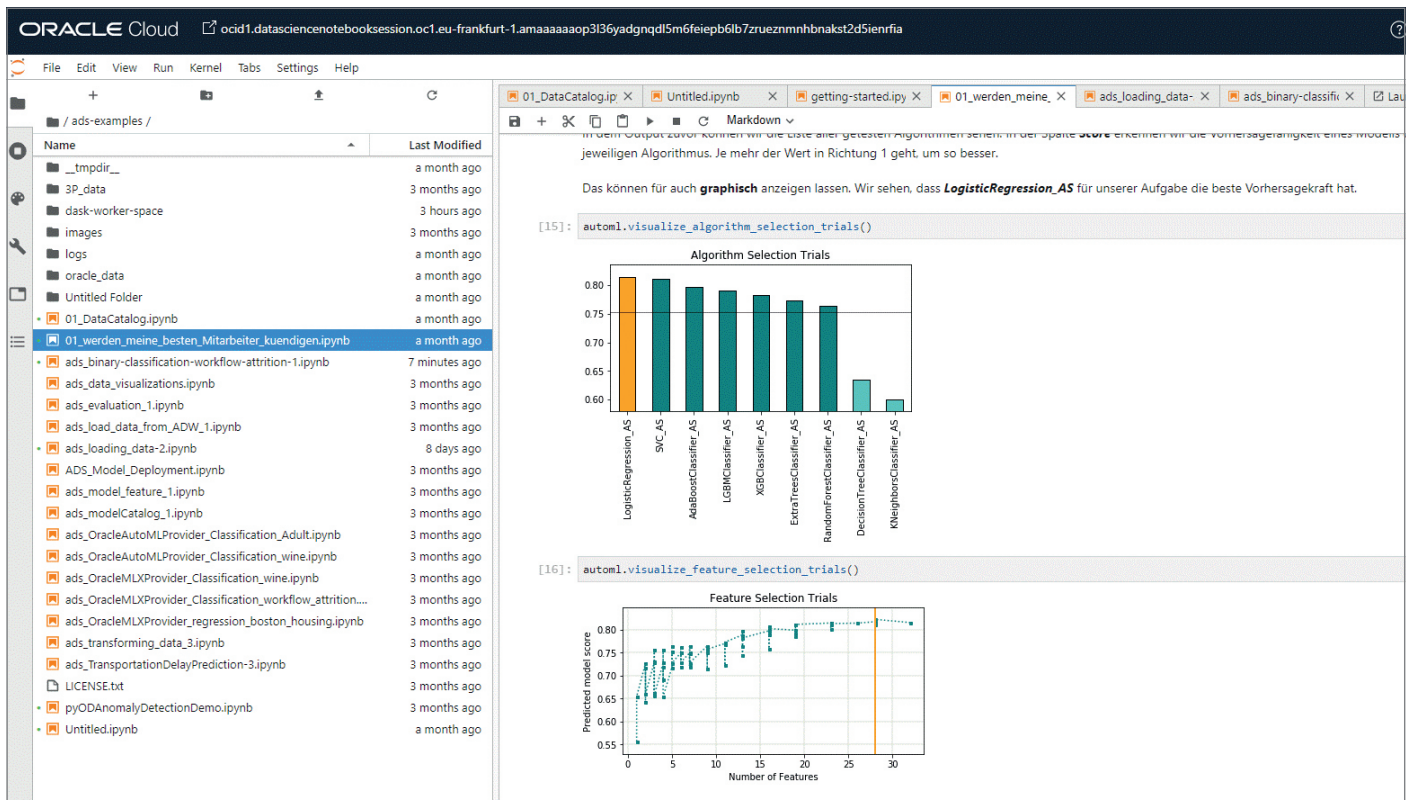


Abbildung 1: Gesamtsicht, Jupyter Notebook (Python) (Quelle: Alfred Schlaucher)

flow. Daten werden zunächst gesucht und wenn sie gefunden sind, werden sie eingelesen und man versucht die Struktur der Daten zunächst zu verstehen. Dann bereitet man die Daten für den eigentlichen Analyse-Vorgang auf, der dann in der Data Science Cloud automatisiert erfolgt. In einem letzten Schritt wird die Tauglichkeit der Ergebnisse überprüft. All diese Schritte können sehr zeitaufwendig sein. Hier setzt der Data Science Cloud Service an. Der Service automatisiert diese Schritte, steigert damit die Produktivität des Analysten und verkürzt die Analysezeit enorm.

Im Folgenden werden die einzelnen Workflow-Schritte dargestellt.

Schritt 1: Gewinnen von Daten

Die Data-Science-Cloud-Lösung hat alle Datenzugriffsanforderungen in dem Modul *DatasetFactory* zusammengefasst. Dieses Modul beinhaltet:

- Lesen und Schreiben der **wichtigsten Datenformate** wie CSV, TSV, Parquet, Libsvm, JSON, Excel, HDF5, SQL, XML, Log-Files usw. Interessant ist die Fä-

higkeit, Files mit einem einzigen Aufruf in ein anderes Format zu überführen, also Lesen von CSV und Schreiben nach Parquet oder Lesen von XML und Schreiben nach JSON. Um Missverständnissen vorzubeugen: Es geht nicht darum, dass man dieses mit Programmiermitteln überhaupt kann. *Es geht darum, dieses mit einem einzigen Aufruf und spontan zu erledigen.*

- Lesen von den diversen **physischen Ablageorten** wie lokales File-System, Google Cloud Storage, AWS S3, Object Storage, SQL-Server, Oracle DB, noSQL und vielen anderen mehr.
- Data Sampling. Große Dateien will man ungern auf einen Schlag einlesen. Über einen Schwellwert steuert man, ab welcher Anzahl von Sätzen ein Sampling beginnen soll. Es ist also auch möglich, mit riesigen Dateien zu arbeiten. Allerdings kostet das in der Erkundungsphase sehr viel Zeit und man wird einen Sampling-Schwellwert einstellen.
- Data Profiling. Die Daten werden bereits beim Lesen im Sinne von Data Profiling analysiert. Das heißt, unmittelbar nach dem Lesevorgang stehen die wichtigsten Metadaten der Datei und Strukturmerkmale der einzelnen

Spalten/Variablen zum Abfragen bereit. Kategoriale und metrische Attribute, Heterogenität der Attribute etc. werden erkannt und potenzielle Kandidaten für eine Ziel-Variable werden vermerkt.

Schritt 2: Verstehen von Daten

Auch wenn man sich in einem Python-Notebook befindet, arbeitet man dennoch interaktiv. Ein eingelesenes Daten-Objekt kann man jetzt interaktiv bezüglich Struktur und Werteverteilung betrachten. Dabei hilft eine automatisch aufbereitete Grafik.

- Über Funktionen wie „*type_of*“ fragt man die Struktur der Daten ab.
- Die Funktion „*show_in_notebook*“ zeigt mit einem Aufruf
 - ein Summary des kompletten Datensets,
 - AutoPlot der einzelnen Attribute/Features entsprechend der Datenstruktur (abhängig von continuous/categorical/binary/multi-class),
 - grafische Anzeige des Correlations-Grades zwischen Features,

- Daten-Sampling-Ausschnitt (Real-daten).

Schritt 3: Präparieren der Daten

Nicht alle Datenattribute sind Analyse-relevant. Indizes, starke Korrelationen, Attribute mit lediglich einem oder gar fehlenden Werten tragen in der Regel nicht zu einem sinnvollen Vorhersageergebnis bei. Auch dies wird automatisch bereinigt:

- Die Funktion „*get_recommendations*“ zeigt solche Schwachstellen auf:
 - Löschung von Konstanten-Spalten und Primary-Key-Felder
 - Ersetzen von NULLs durch „angenehmere“ Werte
 - Löschen von stark korrelierenden Spalten
 - Ausbalancieren von Skewness
- Die Funktion „*auto_transform*“ kann die Ergebnisse dieser Schwachstellenanalyse automatisch, also ohne weitere Interaktion, bereinigen. Will man das nicht, so klickt man sich interaktiv durch einen Dialog mit möglichen Reinigungsmaßnahmen durch.

Maßnahmen wie Skalieren oder Transponieren usw. bleiben einem späteren Schritt vorbehalten. Sie werden abhängig von den später eingesetzten Algorithmen im Rahmen des AutoML automatisch durchgeführt.

Schritt 4: Erstellen des Vorhersagemodells

Erst jetzt startet die eigentliche Maschine-Learning- beziehungsweise Deep-Learning-Analyse. Der Analyst kann mehrere Wege wählen:

1. Für einen ersten „Schnellschuss“ bietet sich AutoML an. AutoML wählt anhand der vorliegenden Daten eine Reihe von voreingestellten Algorithmen zur Modellerstellung, führt ein Scoring der Modellgüte durch und schlägt den für die Daten am besten geeigneten Algorithmus in einem Ranking vor. Dieser Weg wird für die meisten Anwendungsfälle ausreichende Ergebnisse erzielen. Gerade für Analysten mit

wenig langjähriger Erfahrung oder bei wenig zur Verfügung stehender Projektzeit ist dieser Weg vorzuziehen.

2. Alternativ dazu kann man natürlich mit Python selbst den Aufruf möglicher Algorithmen programmieren. Diese klassische Vorgehensweise ermöglicht das individuelle Setzen von Parametern für die Algorithmen. Aber auch hier ist es Praxis, nicht nur einen Algorithmus, sondern eine Auswahl unterschiedlicher Algorithmen durchzuspielen.
3. Selbstverständlich ist auch eine Kombination dieser beiden Wege sinnvoll: Zunächst lässt man AutoML eine Reihe von Algorithmen automatisch prüfen, um sich einen für seine Daten besonders geeigneten Algorithmus vorschlagen zu lassen. Diesen nimmt man dann und programmiert ihn mit den eigenen Wunschparametern aus. Ein anschließender Vergleich der Vorhersagegüte zeigt, wer „besser war“.

In allen Varianten wird man allerdings die Vorhersagegenauigkeit der gefundenen Modelle im Nachgang überprüfen. Das erledigt man in einem Folgeschritt.

Zu 1) AutoML:

AutoML besteht aus vier Modulen und bringt diese im Hintergrund automatisch über einen internen Pipe-Mechanismus zur Ausführung. Der Pipe-Mechanismus nimmt die Ergebnisse von Modul 1, schiebt sie zu Modul 2 usw.:

Modul 1: Zur Lösung einer Klassifizierungsaufgabe zieht ein erstes Modul beispielsweise die folgenden Algorithmen heran und erstellt für alle Algorithmen ein Modell:

- AdaBoostClassifier
- DecisionTreeClassifier
- ExtraTreesClassifier
- KNeighborsClassifier
- LGBMClassifier
- LinearSVC
- LogisticRegression
- RandomForestClassifier
- SVC
- XGBClassifier

Modul 2: Ein zweites Modul macht gegebenenfalls ein sogenanntes *Adaptiv Sampling*. Das ist gerade bei großen Datenbeständen wichtig, die man aufgrund ihrer Größe nicht komplett analysieren lassen will.

Modul 3: Das dritte Modul unternimmt das *Feature Selection*. Es rankt also die Attribute des Datensatzes nach ihrer Fähigkeit, zu dem Vorhersageergebnis beizutragen.

Modul 4: Im letzten Modul wird noch ein Hyperparameter-Tuning durchgeführt. Es sucht die optimale Parametereinstellung der einzelnen Algorithmen.

Die Module werden teilweise mehrfach und wechselseitig gestartet.

AutoML ist wahlweise parallelisierbar. Man kann also die genutzte Hardware

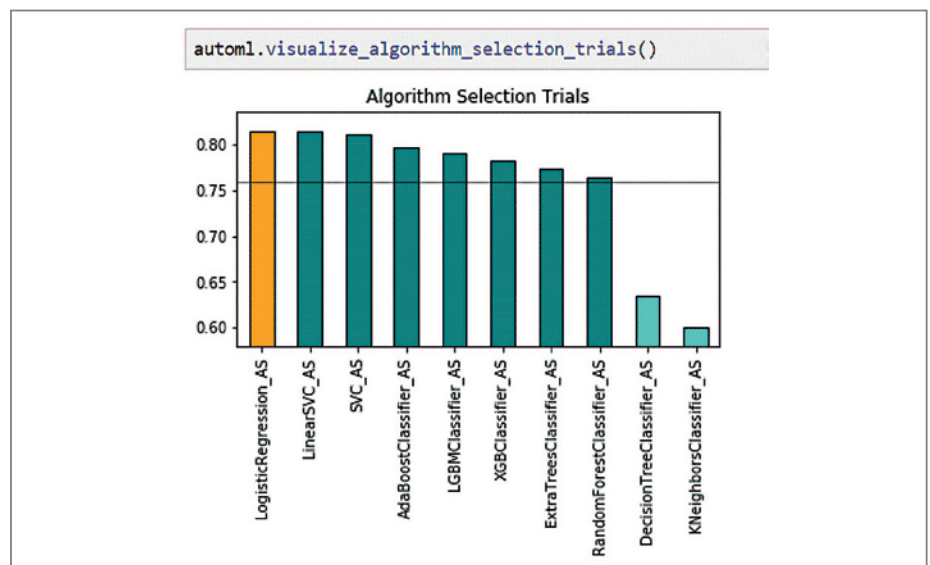


Abbildung 2: Visualisierung der Scores von mehreren durch AutoML getesteten Algorithmen. Logistic Regression zeigt mit 82% den besten Wert. Mit diesem Kandidaten würde man weiter analysieren. (Quelle: Alfred Schlaucher)

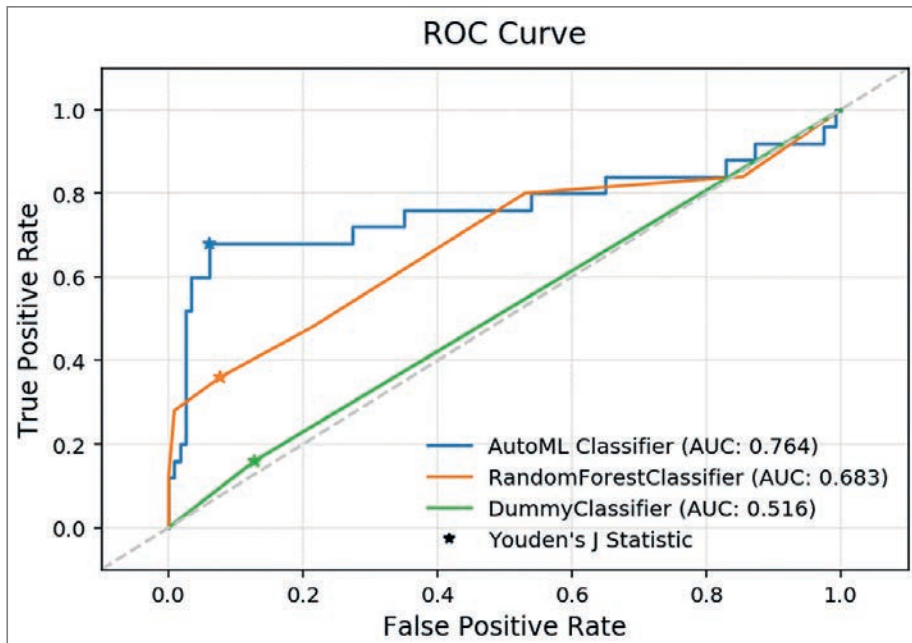


Abbildung 3: ROC-Kurve als Vergleich von mit 3 verschiedenen Algorithmen erstellten Modellen: ein von AutoML erstelltes Modell (blau), ein manuell erstelltes Modell (gelb) und ein Dummy-Modell mit keiner Vorhersagefähigkeit. (Quelle: Alfred Schlaucher)

optimal einsetzen. Bei vielen Prozessoren lässt sich eine hohe Parallelisierung einstellen. Schritte wie das Erstellen von Trainings-/Testdatenbeständen erfolgen zum Teil implizit.

Zu 2) Manuelle Algorithmen-Anwendung / Deep Learning

Bei der Durchsicht der Algorithmen des AutoML wird sicher aufgefallen sein, dass zum Beispiel Deep-Learning-Algorithmen fehlen. Diese sind natürlich optional wählbar, aber AutoML in Kombination mit Deep Learning kann, ganz pragmatisch betrachtet, in der Praxis zu Schwierigkeiten führen. Wählt man einen Deep-Learning-Ansatz beispielsweise mit TensorFlow (Keras), so hat man es oft mit extrem hohen Datenmengen beziehungsweise mit speziellen Datenformaten wie etwa aus Pixeln bestehenden Bildern zu tun.

Die neuronalen Netze des Deep Learning wenden unter Umständen wiederholt Abermilliarden von Schwellwertoperationen in ihren Knoten an. Hat man sich zur Anwendung eines Deep-Learning-Algorithmus entschieden, wird man auch die Hardware, beispielsweise GPUs als Rechenwerke, wählen. Das ist in der Cloud natürlich kein Problem und der Data Science Cloud Service von Oracle bietet diese Variante auch an. Diese Vorgehensweise ist allerdings nicht für alle

Aufgabenstellungen sinnvoll und je nach Anwendungsfall wird der Analyst zwischen den oben gezeigten klassischen Algorithmen und einem Deep-Learning-Ansatz entscheiden. Oder er wählt testweise beide, um die Ergebnisse zu vergleichen. Die Vorgehensweise lautet also eher: zunächst AutoML und dann Deep Learning – und nicht beides zur gleichen Zeit im Rahmen eines AutoML.

Damit haben wir einen Fall gefunden, in dem sich ein manuelles Anwenden eines Algorithmus empfiehlt. Der Oracle Data Science Cloud Service liefert für Deep-Learning-Verfahren fertig installierte TensorFlow-Libraries mit dem eingebetteten Keras. Je nachdem, welches Compute-Shape man in der Oracle Cloud gewählt hat, lassen sich die Berechnungen im Hintergrund auch auf GPUs ausführen und das ganz spontan ohne Vorbereitung der Infrastruktur (siehe Abbildung 2).

Schritt 5: Modelle evaluieren / Prüfen der Güte von Modellen

Einer der letzten Schritte des Workflows ist die Prüfung, ob ein Modell auch gut vorhersagen kann, und das Klären der Frage, welches von mehreren berechneten

Modellen eine Aufgabenstellung am besten löst. Nehmen wir etwa die oben beschriebene 3. Vorgehensweise, also die Kombination aus AutoML und zusätzlich manuell entwickeltem Modell, so will man möglichst bequem die Qualität aller gefundenen Modelle vergleichen.

Hier kommt das *ADSEvaluator*-Modul ins Spiel. Das ist ein einfacher Funktionsaufruf mit den Namen der zu prüfenden Modelle als List-Parameter (also beliebig viele Modelle können gleichzeitig geprüft werden). Als Ergebnis erhält man unter anderem:

- Einen Plot mit einer Precision-Recall-Kurve mit allen Modellen
- Eine ROC-Kurve mit allen Modellen (siehe Abbildung 3)
- Ein Lift Chart mit allen Modellen
- Ein Gain-Chart-Plot mit allen Modellen
- Für jedes Modell eine Confusion-Matrix
- Eine vergleichende Kosten-Analyse der jeweiligen Modelle
- Und vieles mehr

Schritt 6: Modelle erklären

Die Vorhersagequalität (Güte) eines Modells festzustellen, bedeutet noch nicht, dass man auch versteht, warum ein Modell eine bestimmte Vorhersage macht. Die Erklärung eines Modells (Model Explanation), ermöglicht erst Rückschlüsse auf die Ursachen von Ereignissen, also die besonderen Verhältnisse in den Unternehmensprozessen, die zu den Analyse-daten geführt haben. Hinzu kommt die fortschreitende Weiterentwicklung der zunehmend komplexeren Algorithmen:

- Welche Features tragen am meisten zum Entstehen eines Ereignisses bei?
- Wie weit ist ein Modell generalisierbar, das heißt, sind die Vorhersagemodelle auch noch passend, wenn neue, bislang unbekannte Input-Daten vorliegen, oder hat das Modell nur bekannte Zustände „auswendig gelernt“ (Overfitting-Problem)?

Den Prozess zum Erklären von Machine-Learning- und Deep-Learning-Modellen nennt man im Englischen „Machine Learning Explainability“ (MLX). Die Aufgabe des MLX wird in dem Data Science Cloud Service über den *ADSExplainer* gelöst.

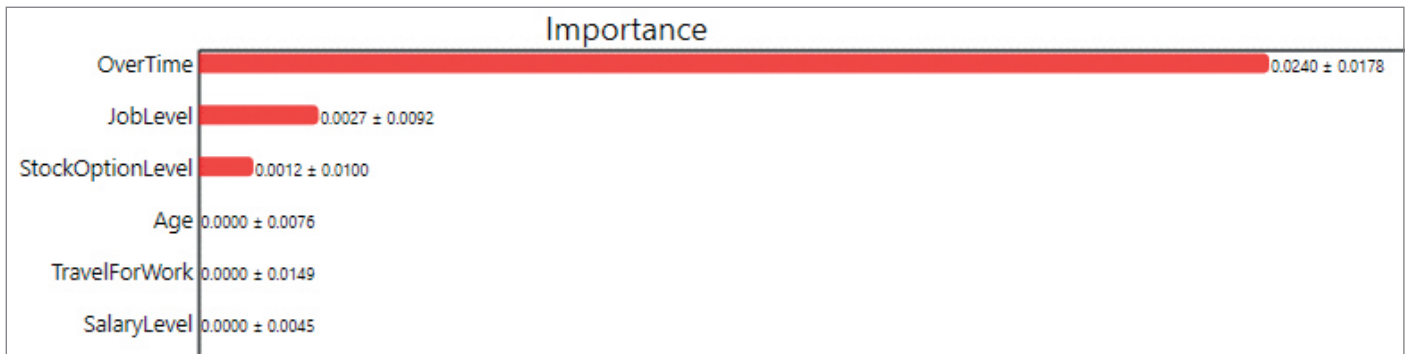


Abbildung 4: Visualisierung der Top-6-Attribute, also der Attribute, die am meisten zu einem Vorhersageergebnis beitragen (Feature-Importance-Analyse) (Quelle: Alfred Schlaucher)

Mehrere Optionen stehen zur Auswahl:

- In einer *Global Explanation* Phase wird die Bedeutung der einzelnen Attribute für das Vorhersageergebnis näher erklärt. Man untersucht, wieweit sich die Vorhersagefähigkeit eines Modells durch Weglassen und Hinzufügen von Attributen verändert.
- Das *Local Explanation* untersucht die Vorhersage von einzelnen Ereignissen aufgrund von Werteverteilungen in den Attributen und gibt damit punktgenaue Erklärungen für eine einzelne Vorhersage in einem konkreten Fall.

Eine Vielzahl von Einzelmethoden schließt sich hier an, die der ADSExplainer mit diversen weiteren Plots anbietet. In dieser Phase muss Domain- und Geschäftswissen die weitere Analyse begleiten, das heißt, Fachanwender sind wieder mit an Bord und müssen helfen, die Einzelergebnisse zu interpretieren (siehe Abbildung 4).

Schritt 7: Verwalten von Modellen / Model Catalog

Der letzte Schritt bei der Herstellung von Machine-Learning- und Deep-Learning-Modellen ist deren Verwaltung (Management) nach Abschluss der Modellentwicklung. Nach Abschluss der Modellentwicklung überführt man Modelle beispielsweise in operative Ablaufprozesse und nutzt sie dort zur Steuerung von betrieblichen Vorgängen. Den aktuellen Entwicklungsstand eines Modells, also den Stand, der sich aktuell in einem produktiven Einsatz befindet, speichert der Data Science Cloud Service im *Model Catalog*.

Spätestens an dieser Stelle sollte das interne *ADS-Modellformat* erwähnt werden. In diesem Format speichert der Data Science Cloud Service alle Modelle im Model Catalog ab. Dieses interne ADS-Format wurde bereits in den Schritten zuvor, also Modell-Evaluation und Modell-Explanation, verwendet. Verwaltungsoperationen sind unter anderem *save-*, *load-*, *publish-model*. Zur besseren Verwaltung ergänzt man gespeicherte Modelle mit weiteren Metadaten, wie Beschreibungen, Verwendungszweck, Identifikationsnummer, Autor, Versionen usw.

Solche Metadaten können schließlich in einem Browser oder in der OCI-Oberfläche publiziert werden.

Fazit

Der Data Science Cloud Service erleichtert nicht nur durch seine vielen Automatismen die Arbeit des Data Scientist, der Service beinhaltet auch einige Funktionen, die man in anderen Lösungen nicht findet. Dazu gehören *Dataset-Factory*, *ausgeprägtes AutoML* sowie das *Explanate-Modul*. Der Service hat das für viele gewohnte *Anaconda Jupyter Notebook* fertig konfiguriert im Gepäck und kostet auch nichts extra. Damit ist der Service ein kleines Highlight in der Oracle Cloud.

Über den Autor

Alfred Schlaucher ist seit 1986 in der IT tätig. Er startete mit der Programmierung von Datenbanken auf Großrechnern von Siemens und IBM. Zu Beginn der 1990er Jahre verlagerte sich der Schwerpunkt

seiner Arbeit auf Datenmodellierung und Metadatenmanagement. Seit Ende der 1990er Jahre berät er Oracle-Kunden bezüglich ihrer Data-Warehouse- und Analyse-Infrastruktur. In den letzten Jahren beschäftigt sich der Autor verstärkt mit Big Data, Advanced Analytics und Machine Learning sowie den Analysesprachen R und Python.



Alfred Schlaucher
Alfred.Schlaucher@oracle.com



Vulnerability-Management mit APEX

Christina Finck und Klaus Schuermann, Arvato Systems

Cyber-Kriminalität ist ein weltweites Problem. Im Zeitraum von 2017 bis Ende 2019 sind 75 Prozent der Unternehmen in Deutschland Opfer von Datendiebstahl, Industriespionage oder Sabotage geworden [1]. Die Folgen für betroffene Firmen können verheerend sein – von Reputations-Schäden bis hin zu massiven finanziellen Einbußen. Vor diesem Hintergrund ist IT-Security relevanter denn je. Viele Unternehmen sind bemüht, Security Incidents vorzubeugen, indem sie Schwachstellen detektieren und proaktiv schließen – und zwar, bevor Kriminelle diese ausnutzen. Das ständige Erheben, Analysieren, Abgleichen und Reporten von Schwachstellen-Daten ist jedoch eine große Herausforderung. Dabei gibt es Lösungen, mit denen sich Hürden bei der Datenverarbeitung einfach abbauen und IT-Schwachstellen effizient managen lassen.

Potenzielle Sicherheitslücken identifizieren

Schwachstellen-Management, auch Vulnerability-Management genannt, dient dazu, vorhandene Sicherheitslücken in der IT-Infrastruktur von Unternehmen aufzuspüren und zu schließen. Um Schwachstellen zu identifizieren, kommen typischerweise Schwachstellen-Scanner, wie etwa Nexpose, Nessus, OpenVAS oder Qualys, zum Einsatz. Diese Tools überprüfen Systeme innerhalb eines Netzbereichs regelmäßig auf mögliche Sicherheitslücken. Als Grundlage dient der Industriestandard

CVE (Common Vulnerabilities and Exposure) mit mehr als 100.000 Einträgen. Den Umfang der ohnehin umfangreichen Liste erweitern die Scanner-Hersteller fortlaufend [2]. Der Daten-Output erfolgt in Form einer Auflistung von Varianten, die aufzeigt, welche potenziellen Sicherheitslücken eine bestimmte Software-Installation oder -Konfiguration auf einem System eröffnet. Die Art der Schwachstelle und die Einschätzung, wie leicht sie sich ausnutzen lässt, beeinflussen die Priorität, mit der sie zu schließen ist. Dabei gilt: Je größer die zu bearbeitende IT-Landschaft ist und je älter sowie verwundbarer die Systeme sind,

desto schwieriger ist es, entsprechende Fälle zu identifizieren.

Ein fortlaufender Prozess aus Scannen und Lückenschließen

Ein typischer Schwachstellen-Report ist umfangreich und unübersichtlich. Um eine Schwachstelle zu schließen, müssen Unternehmen aus dem Report zunächst Handlungsmaßnahmen, sogenannte Remediation Action, ableiten, die ein System-Administrator im nächsten Schritt softwareseitig umsetzt. Ein erneuter Scan

bringt dann die nötige Gewissheit, ob es gelungen ist, die Schwachstelle tatsächlich zu schließen (siehe Abbildung 1).

Der Wille, durchzuhalten

Dabei ist zu beachten, dass ein Scanner nur meldet, welche Schwachstellen zum Zeitpunkt des Scans vorhanden waren – nicht aber, welche Schwachstellen bereits erfolgreich geschlossen wurden. Die Krux: Zu diesem Zeitpunkt hat der Schwachstellen-Manager meist schon mehrere Spreadsheets vorliegen, mit denen er versucht, das Delta zwischen Scan-Zeitpunkten zu ermitteln, umgesetzte Maßnahmen zu tracken oder False Positives sowie Exceptions nachzuhalten. Aufgrund der daraus resultierenden Komplexität ist das der Zeitpunkt, an dem viele Unternehmen kapitulieren: Sie geben das Schwachstellen-Management entweder komplett auf oder beschränken sich auf die allernötigsten Aufgaben – mit dem Ergebnis, dass die Angriffsfläche ihrer Systeme zunehmend größer wird.

Prototyping

Schwachstellen-Management mit herkömmlichen Mitteln und verfügbaren Produkten zu betreiben, ist extrem zeit- und ressourcenintensiv – insbesondere dann, wenn es sich um große Rechenzentren und heterogene IT-Landschaften mit mehreren zehntausend Hosts handelt. Um dem Abhilfe zu schaffen, haben wir mit einem ersten Proof of Concept, geschrieben in PHP unter Nutzung von MySQL, versucht, den Prozess, die mittels Scan gene-

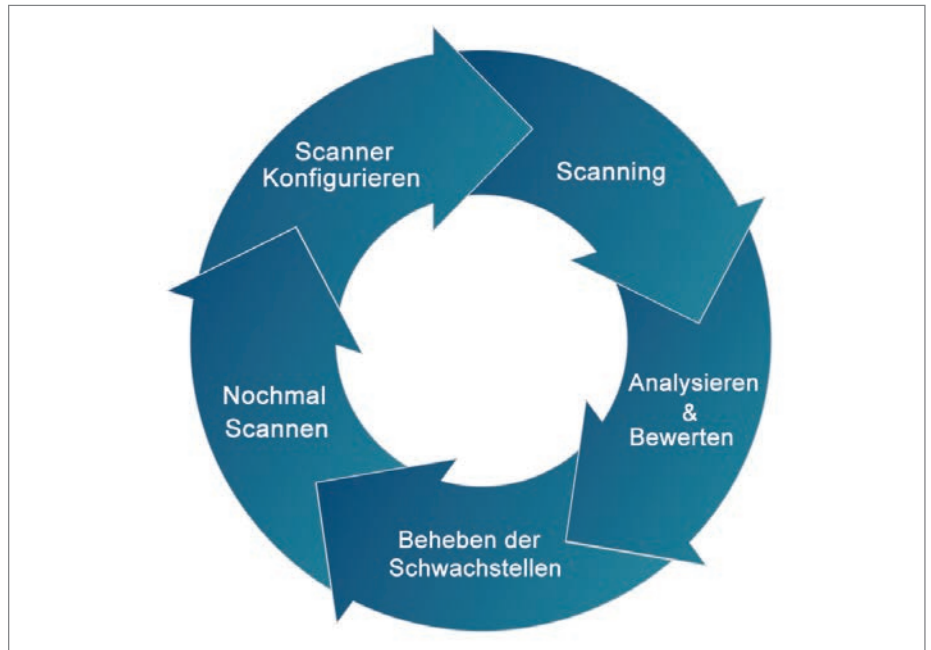


Abbildung 1: Der Schwachstellen-Management-Prozess (eigene Darstellung, © Arvato Systems)

rierten Daten in Handlungsempfehlungen zu übersetzen, zu automatisieren. Die Idee dahinter: die Scan-Daten um „Tags“, basierend auf Mustern in den Roh-Daten, anzureichern. Ein Tag entspricht dabei einer nötigen Maßnahme, um eine bis n mögliche Schwachstellen mit derselben Ursache zu beheben. Abbildung 2 zeigt ein Beispiel eines exemplarischen Prozesses.

Auf einem Host ist der „Adobe Acrobat Reader“ in der Version 19.010 installiert.

1. Schritt: Import von Findings

Der Scanner meldet 326 Findings, in der Form „acrobat-cve-2019-8221, acrobat-cve-2019-7039, acrobat-cve-2019-8017, acrobat-cve-2019-8016, acrobat-cve-2019-8025, acrobat-cve-2019-7066, ...“.

2. Schritt: Tagging/Gruppieren

Unsere Tagging-Logik vergibt das Tag „Adobe Acrobat Obsolete“.

3. Schritt: Aggregation

Durch die Aggregation der Daten ergibt sich das Ergebnis: ein Datensatz für 326 Findings.

4. Remediation

In der Datenbank ist für das Tag „Adobe Acrobat Obsolete“ die Remediation Action „Upgrade to newer version“ hinterlegt, für die im Idealfall automatisch ein Ticket erstellt und weitergeleitet wird.

Dieser Prozess bietet den Vorteil, dass anstelle von n Varianten, wie eine bestimmte Software-Version oder Fehl-Konfiguration ausgenutzt werden kann, nur die nötige Maßnahme zu reporten ist. Sie beugt möglichen diversen Exploits vor, wie beispielsweise dem Patchen einer Software oder einer Härtungs-Maßnahme auf einem System. Anhand der Tags lässt sich dann eine aggregierte Übersicht über die nötigen Maßnahmen erstellen, die ein Schwachstellen-Manager direkt bearbeiten kann. Dank einer automatisierten Daten-Aggregation steigert dieser Prototyp die Prozess-Effizienz signifikant.

Daneben gab es weitere Hürden zu nehmen, wie zum Beispiel:

- das Nachverfolgen der Daten im Zeitverlauf,

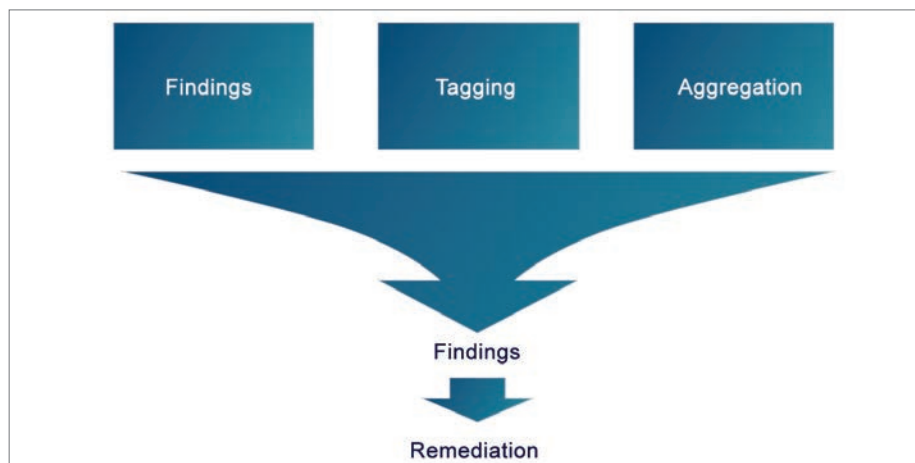


Abbildung 2: Aggregation der Schwachstellen (eigene Darstellung, © Arvato Systems)

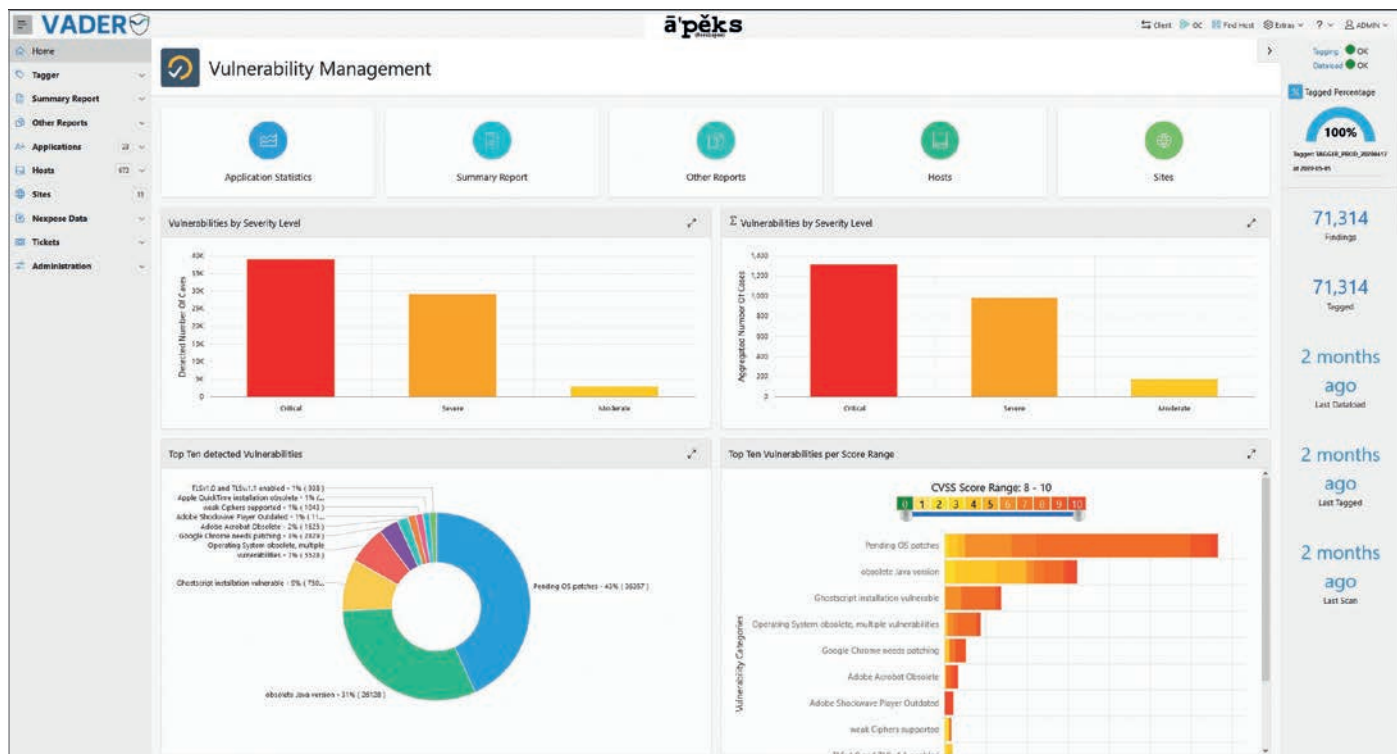


Abbildung 3: Das „Vader“ Dashboard (© Arvato Systems)

- das Einbinden von Reporting-Funktionalitäten, die flexibel genug sind, um den Berichtsanforderungen aller Anwender – vom System-Administrator bis hin zum CIO – gerecht zu werden,
- ein User-Management gemäß Need-to-know-Prinzip, das unterschiedliche User-Rollen unterstützt und
- das Anreichern der Scan-Daten um Kontext- oder Bearbeitungs-Informationen.

Um eine Anwendung zu entwickeln, die all diesen Anforderungen gerecht wird, haben wir auf Oracle APEX als Low-Code-Entwicklungsumgebung zurückgegriffen. Dank seiner vorgefertigten, leicht adaptierbaren Module konnten wir insbesondere das User-Management und flexible Reporting-Möglichkeiten in der Anwendung mit Arbeitstitel „Vader“ bedarfsgerecht umsetzen (siehe Abbildung 3). Mit einem anderen Framework hätten wir diese Features eigens programmieren müssen.

Anpassungen kurzfristig umsetzen und testen

Mit APEX haben wir die Methode des Prototyping fortgeführt. APEX ermöglicht es dem Entwickler, Anpassungen zeitnah

umzusetzen und zu präsentieren. So war es möglich, Layout-Wünsche kurzfristig zu realisieren, zu testen – sowie im Zweifel zu verwerfen und durch bessere Layouts zu ersetzen. Dieses Vorgehen führte zu einer hohen Akzeptanz und einer guten Usability.

Aufgrund dieses überzeugenden Ansatzes gab es wenige Monate nach dem ersten produktiven Einsatz eine neue Anforderung: Es galt, das System auf Mandantenfähigkeit umzubauen und um eine Zwei-Faktor-Authentifizierung zu erweitern (siehe Abbildung 4). Dabei war es sehr

hilfreich, auf verfügbare Plug-ins zurückzugreifen, um auch komplexere Funktionen in kurzer Zeit zu implementieren.

Der „Tagger“

Für die dynamische Konfiguration der Tagging-Logik haben wir in PL/SQL eine Rules-Engine implementiert, den sogenannten „Tagger“. Die vom Scanner gelieferten Findings werden nach einer umfangreichen Sequenz von Kriterien ausgewertet. Die in einem Katalog hinterlegten Krite-

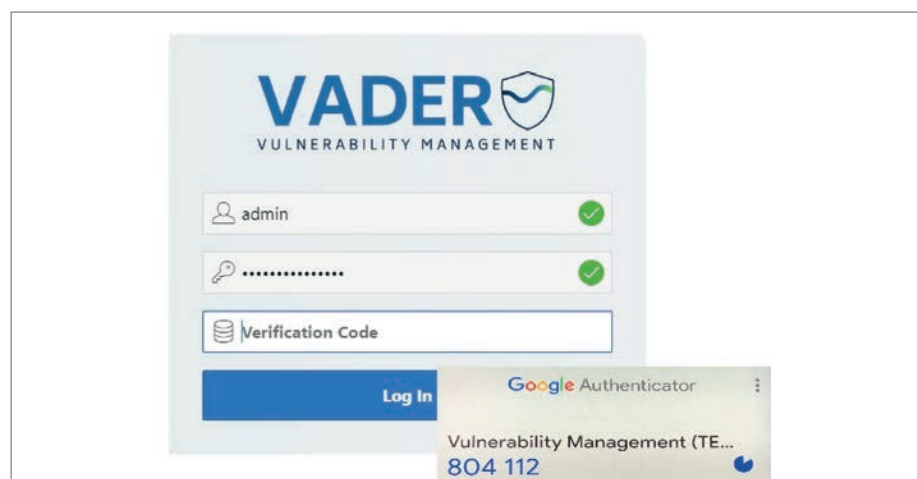


Abbildung 4: Login-Maske mit Zwei-Faktor-Authentifizierung (© Arvato Systems)

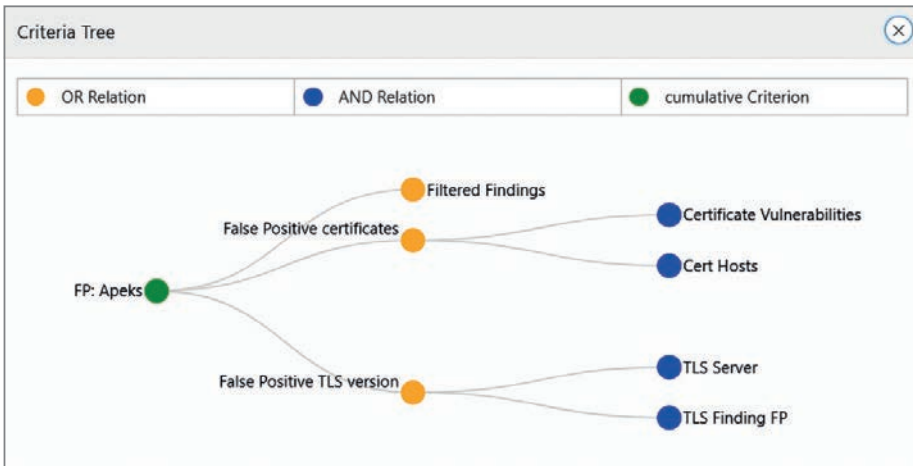


Abbildung 5: Tagger Treewiew (© Arvato Systems)

rien lassen sich nach unterschiedlichen Methoden modellieren, wie etwa Pattern Matchings, Regular Expressions, Datums-Intervalle, Listen-Lookups oder auch einfache Vergleiche. Um ein Tag zu vergeben, sind oft mehrere Kriterien mit booleschen Operationen verknüpft. Um die Kriterien zu definieren, kann der Nutzer angeben, ob sie systemweit oder nur für einzelne Mandanten gelten sollen. Zurzeit vergibt der produktive Tagger 240 verschiedene Tags, für die 361 Kriterien evaluiert werden. Um dabei nicht den Überblick zu verlieren, ha-

ben wir in APEX eine grafische Darstellung einzelner Kriterien und des gesamten Taggers erstellt (siehe Abbildung 5).

Ein Aufgabenschwerpunkt des Schwachstellen-Managers besteht in der Pflege des Taggers. Neuartige, vom Scanner gemeldete Schwachstellen sind über einen APEX-Dialog in die Tagger-Logik zeitnah einzu-arbeiten, sodass auch diese mit einem Tag versehen werden. Dabei ist es möglich, die Tagger-Logik in einer Test-Umgebung zu entwickeln und zu testen. Im Anschluss an einen erfolgreichen Test lässt sie sich als

gezippte JSON-Datei in die Produktions-Umgebung übertragen.

Reporting

Schwachstellen zu reporten, ist der eigentliche Zweck von Vader. Dabei sind die Anforderungen je nach Sichtweise und Rolle des Betrachters sehr unterschiedlich. Für das Reporting der Schwachstellen kommen drei verschiedene Techniken zum Einsatz:

- Interactive Reports für die Statusverarbeitung und die Ticket-Erstellung
- Jet Charts für Statistiken und die Darstellung in der Timeline
- JasperReport für HTML-E-Mails

Die von APEX out of the box verfügbaren Interactive Reports erfüllen einen Großteil der Anwenderwünsche. Besonders nützlich sind dabei die Möglichkeiten, Spalten auszuwählen, neu anzuordnen und als neuen Report unter einem eigenen Namen zu speichern (siehe Abbildung 6).

Das in APEX vorintegrierte JavaScript Extension Toolkit (JET) bekommt mit jeder neuen APEX-Version mehr deklarati-

Abbildung 6: Interactive Report mit den Funktionen zur Statusverarbeitung und Ticket-Erstellung (© Arvato Systems)

ve Funktionen, die Nutzer nur noch ein- oder ausschalten müssen. Die in unserer Anwendung erstellten Businessgrafiken ermöglichen es dem Anwender, den Betrachtungszeitraum durch die Auswahl der entsprechenden Parameter zu bestimmen und sich die Ergebnisse zum Beispiel auf Tages-, Wochen- oder Monatsebene anzeigen zu lassen (siehe Abbildung 7).

Reports als E-Mails verschicken

Der regelmäßige Versand von E-Mails mit Charts und Trend-Darstellungen hat sich als besonders effizient erwiesen, um System-Verantwortliche zu motivieren, tätig zu werden. Mit Bordmitteln war es jedoch nicht möglich, individuelle, adressatenabhängige E-Mails zu erzeugen. Daher haben wir die JasperReportsIntegration, ein Community-Tool von Dietmar Aust, dafür eingesetzt. Mit etwas Aufwand lassen sich damit recht komplexe Reports erzeugen, die sich im HTML-Format als E-Mail verschicken lassen (siehe Abbildung 8).

Technischer Hintergrund

Das System befindet sich im stetigen Wachstum. Aktuell arbeiten rund 140 Anwender für 23 Mandanten damit. Dabei werden die Scan-Daten von zurzeit mehr als 12.500 Hosts importiert. Das sind über eine Million Datensätze, die täglich zu tagen und zu aggregieren sind. Ein Mehrfaches davon ist geplant. Dazu setzen wir Oracle 19c in der Standard Edition ein. APEX haben wir kürzlich problemlos auf Release 20.1 umgestellt. Dabei schöpfen wir die verfügbaren technischen Möglichkeiten von Oracle und APEX umfangreich aus:

- Die Scan-Daten werden über ein REST/JSON API geladen. Dabei gewährleistet der Oracle REST Data Service eine ideale Kombination mit den SQL/JSON-Kommandos.
- Sämtliche Batchverarbeitung wird durch den Oracle Scheduler gesteuert.
- Zur Verarbeitung von CSV-Dateien nutzen wir das höchst effiziente Interface von APEX_DATA_PARSER.

Für einen möglichen Umstieg auf eine Enterprise Edition sind auch schon die Par-

tition Keys so weit vorgesehen, dass sich dieses Feature problemlos aktivieren ließe. Um einige komplexe Abfragen zu beschleunigen, ist kurzfristig geplant, auch Oracle Text, welches das Interactive Grid bereits heute deklarativ unterstützt, einzusetzen.

Die Vorteile von Vader im Überblick

- Signifikante Reduktion (90 Prozent von Zeit und Aufwand im Schwachstellen-Management)
- Aufbau eines „Remediation Wikis“, in dem sich für die vergebenen Tags beim ersten Auftreten die Handlungsmaßnahmen manuell ergänzen und in der Folge automatisiert anwenden lassen
- Automatisierte Übersetzung der Daten zu ausführbaren Arbeitspaketen (z. B.: Vader reportet, dass eine Java-1.7-Version zu aktualisieren ist – ein To-do; ein Scanner reportet 350 Varianten, wie sich diese Version ausnutzen lässt, um ein System zu kompromittieren)

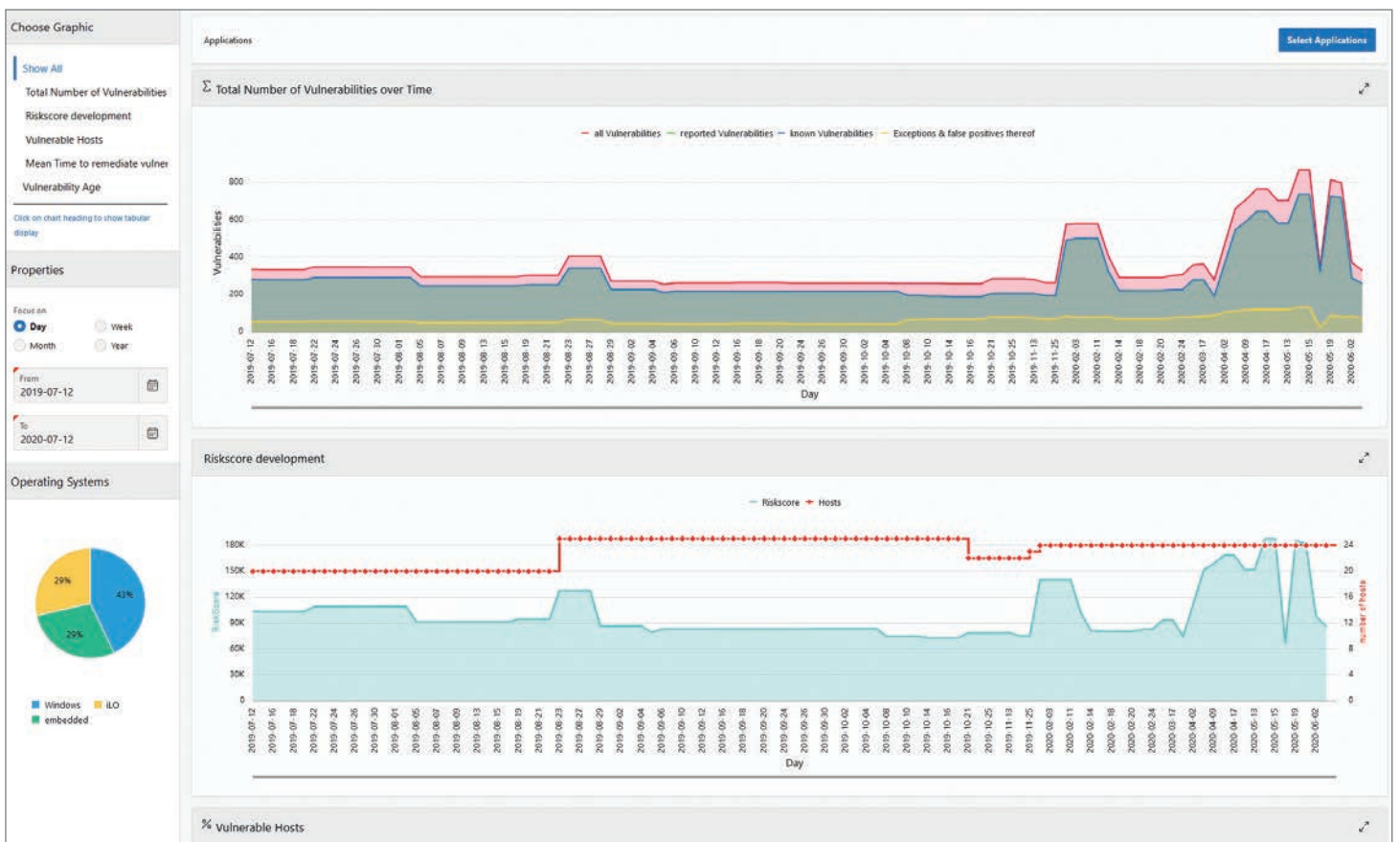


Abbildung 7: Statistics (© Arvato Systems)

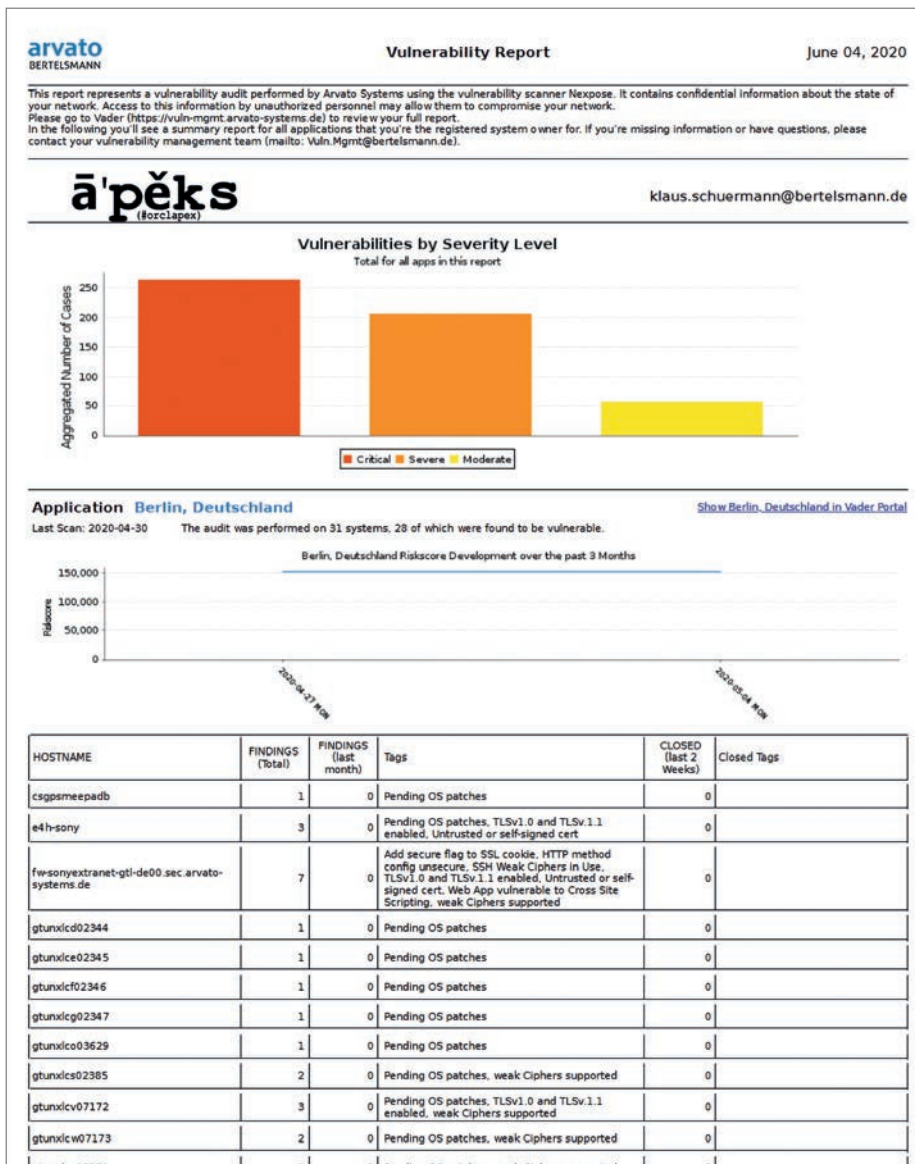


Abbildung 8: E-Mail-Report (© Arvato Systems)

- Einfaches Tracking von Maßnahmen zur Schwachstellen-Behebung im Zeitverlauf (Status und qualitative Informationen der Bearbeitung)
- Konsolidierung aller relevanten Schwachstellen-Management-Daten in einem System (Remediation Plan, Best Practices, Tickets etc.)
- Audit-sichere Dokumentation mit zwölfmonatiger Historie
- Flexibles Reporting, das den Bedürfnissen unterschiedlicher Stakeholder im Schwachstellen-Management-Prozess gerecht wird

Vader ist für uns selbst und unsere Kunden zum Werkzeug der Wahl im Schwachstellen-Management geworden. Bis Jahresende wird es für alle unsere Rechenzentren und in der Cloud

gehosteten Systeme verfügbar sein. Wir entwickeln die Anwendung kontinuierlich weiter, um das Schwachstellen-Management noch einfacher und effizienter zu gestalten. Aktuell arbeiten wir daran, Vader an unser Ticket-System anzubinden, sodass wir die Implementierung einer Maßnahme zur Beseitigung einer Schwachstelle direkt in Vader beauftragen und nachverfolgen können. Ein weiterer wichtiger Aspekt ist die Integration von Asset- und Lifecycle-Daten. Die Bandbreite an Software, die ein Scanner prüfen kann, ist limitiert. Mit der Anreicherung von Inventar-Daten ist es uns möglich, einen vollständigen Schwachstellen-Report zu erstellen, der alle nötigen Maßnahmen darstellt. Im weiteren Verlauf werden wir Vader in weitere Systeme integrieren.

Quellen:

- [1] <https://de.statista.com/statistik/daten/studie/150885/umfrage/anteil-der-unternehmen-die-opfer-von-digitalen-angriff-en-wurden/>
- [2] <https://cve.mitre.org>

Über die Autoren

Christina Finck ist Senior Project Manager bei Arvato Systems. Als Projektleiterin betreut sie diverse Projekte in den Bereichen IT-Infrastruktur und Security. Sie ist die konzeptionelle Urheberin und Produkt-Managerin von „Vader“, der Vulnerability-Management-Lösung von Arvato Systems.

Klaus Schuermann ist Senior Consultant bei Arvato Systems. Seit 1985 entwickelt er Oracle-basierte Anwendungen, überwiegend im Handels- und Medienbereich. Für die Entwicklung nutzte er zunächst Oracle Forms, seit 2006 setzt er ausschließlich und begeistert Oracle APEX ein. Er ist der Architekt der „Vader“-Infrastruktur.



Christina Finck
Christina.Finck@bertelsmann.de



Klaus Schuermann
Klaus.Schuermann@bertelsmann.de



Flows for APEX – Einfach Prozesse in APEX modellieren und ausführen

Niels de Bruijn, MT AG

Ein neues Open Source Tool erlaubt die Modellierung sowie die Ausführung von BPMN-2.0- Prozessen in Oracle Application Express (APEX).

APEX bietet keine Funktionalität für die Modellierung von Prozessen. Normalerweise wird in APEX zuerst das Datenmodell implementiert und anschließend die Oberfläche. Prozesse werden dabei meist außerhalb von APEX visuell erstellt und anschließend programmatisch mit

PL/SQL sowie einigen Tabellen implementiert. Ein Nachteil dabei ist, dass ein Endanwender nicht visuell sehen kann, wo sich der Prozess gerade befindet. Außerdem muss ein APEX-Entwickler die APEX-Umgebung verlassen, um die visuelle Prozessmodellierung vorzunehmen.

Von der Idee bis zur Realisierung

Die Idee für Flows for APEX kam dem Autor Ende 2019: Ein APEX-Entwickler sollte ein Tool für die Prozessmodellierung/-ausführung mit wenigen Klicks in APEX 20.1+

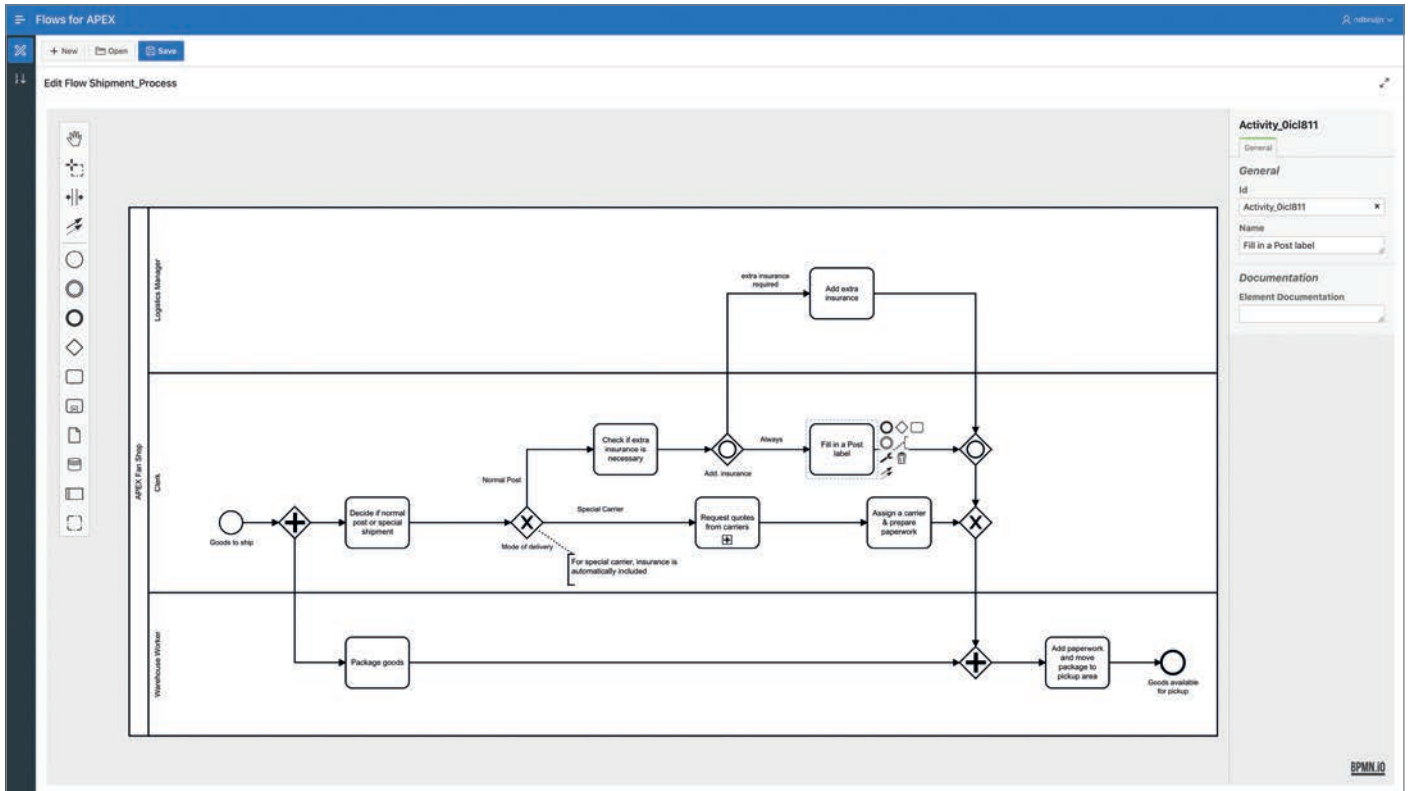


Abbildung 1: Prozessmodellierung in Flows for APEX (Quelle: Niels de Bruijn)

installieren können. Die Arbeit damit sollte den bekannten Low-Code-Ansatz verfolgen, aber dennoch eine programmatische Erweiterung mit PL/SQL an jeder Stelle erlauben.

Für die Realisierung wurde das Open-Source-JavaScript-Framework bpmn.io in APEX integriert, womit eine visuelle Prozessmodellierung ermöglicht wird. Die daraus entstandene XML-Datei wird nach dem Speichern des Prozessmodells durch PL/SQL verarbeitet, damit alle Objekte und Verbindungen in Tabellen relational vorliegen (siehe Abbildung 1).

Die Überwachung von laufenden Flows erfolgt über die Seite „Flow Control“. Hier kann bei Bedarf auch in einen laufenden Flow manuell eingegriffen werden (siehe Abbildung 2).

Wie verwende ich Flows for APEX in meiner APEX App?

Flows for APEX kommt als eine Packaged App und lässt sich in einer Minute installieren. Es stellt zunächst die Modellierungs-

und Ausführungsumgebung bereit. Anschließend können eigene Apps Flows for APEX integriert werden. So gibt es für die Darstellung von einem Flow das Region Plug-in „BPMN Viewer“ (siehe Abbildung 3). Es lässt sich bei apex.world kostenfrei herunterladen und kann in eine bestehende App importiert werden. Anschließend wird es auf einer Seite hinzugefügt. Damit es weiß, welche Prozessinstanz angezeigt werden soll, wird ein bestimmtes SQL-Statement gebraucht. Es wird dabei empfohlen, die Demo-App „Order Shipment“ aus der

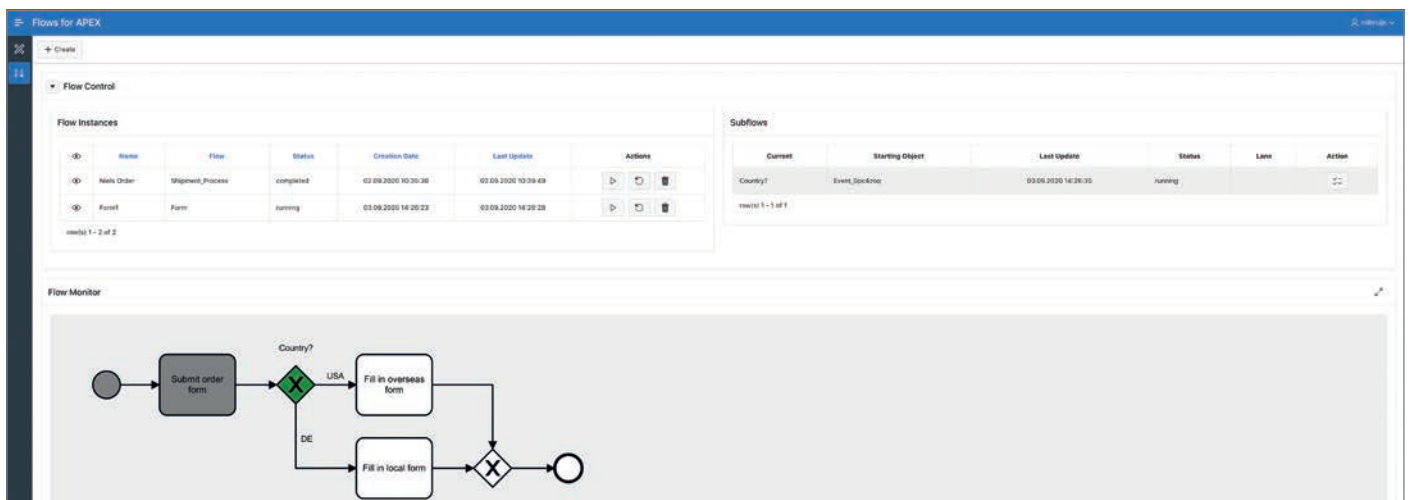


Abbildung 2: Überwachung von laufenden Flows in Flows for APEX (Quelle: Niels de Bruijn)

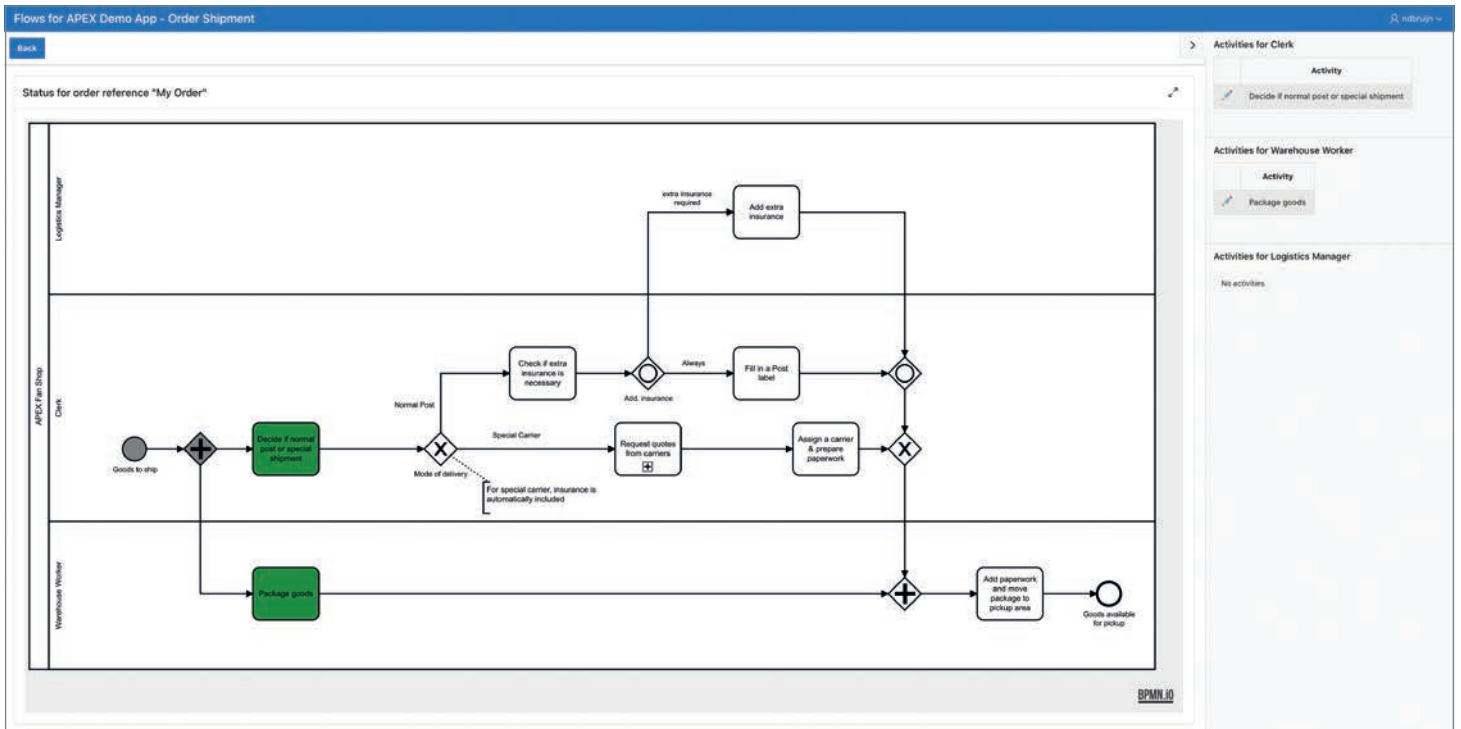


Abbildung 3: Eigene APEX-Applikation mit dem Plug-in „BPMN Viewer“ (Quelle: Niels de Bruijn)

```

procedure form_approve
( pi_process_id in number
, pi_subflow_id in number
)
is
begin
-- die aktuelle Aktivität betrifft eine Genehmigung und erfordert
-- eine Entscheidung (ja/nein).
flows_pkg.flow_next_branch
( p_process_id => pi_process_id
, p_subflow_id => pi_subflow_id
, p_branch_name => 'ja'
);
end;
    
```

Listing 1: Steuerung des Fortschritts im Flow mithilfe eines PL/SQL-API.

Distribution von Flows for APEX zu installieren, die ebenfalls das Plug-in verwendet, um das korrekte Format der Abfrage auf die eigene Anwendung zu übertragen.

Für die Steuerung eines Flow gibt es das PL/SQL Package flow_api_pkg. Ein APEX-Entwickler kann dank dieser seine Entwicklung wie gewohnt in APEX machen und steuert den Flow über das PL/SQL-API (siehe Listing 1).

You never walk alone

Bewusst wurde der Source Code auf github.com unter der MIT-Lizenz bereitgestellt. Von Anfang an wurde damit eine aktive Mitarbeit am Tool durch APEX-Entwickler weltweit er-

möglicht. Aktuell ist ein eingespieltes Team von vier Entwicklern an der Weiterentwicklung beteiligt. Der Austausch findet primär über Zoom und Slack statt. Wer Interesse hat, bei der Entwicklung mitzumachen, kann sich gerne beim Autor melden, aber wir freuen uns natürlich genauso auf Feedback über die Verwendung von Flows for APEX!

Quellen

- Demo App Order Shipment auf apex.oracle.com: <https://bit.ly/flowsforapexdemo>
- Demo Flows for APEX auf apex.oracle.com: <https://bit.ly/flowsforapex>
- Software-Download: <https://apex.mt-ag.com/flowsforapex>
- Dokumentation: <https://mt-ag.github.io/apex-flowsforapex>

- GitHub Repository: <https://github.com/mt-ag/apex-flowsforapex>
- Blog Post: <http://nielsdebr.blogspot.com/2020/06/flows-for-apex.html>
- Plugin „BPMN Viewer“: https://bit.ly/apexplugin_bpmnviewer



Niels de Bruijn
Niels.deBruijn@mt-ag.com

RETRO

Retrospektive: Warum es sinnvoll ist, auch zurückzuschauen

Thimo Fußbroich, Opitz Consulting Deutschland

Retrospektiven sind seit vielen Jahren fester Bestandteil in agilen Entwicklungsprojekten und doch nutzen viele Teams dieses Werkzeug zur Verbesserung ihrer Zusammenarbeit noch nicht. In diesem Artikel werden das Vorgehen und der Nutzen dieses Meetings beschrieben und damit hoffentlich ein Anstoß gegeben, selbst das Vorurteil zu entkräften, dass es einfach nur ein weiteres Meeting ist und uns wertvolle Zeit raubt.

Feedback ist ein wichtiger Bestandteil jeder (guten) Zusammenarbeit in einem Team. Häufig wird dieses Instrument durch den Projektleiter oder einen Vorgesetzten mit den Worten „Wenn was ist, sagt es mir. Ich habe ein offenes Ohr für euch.“ initiiert. Doch seien wir ehrlich: Meistens bleibt es bei diesen Worten oder es erfolgt eine (zu) späte Rückmeldung, wenn etwas nicht so funktioniert, wie erhofft.

Im Werkzeugkoffer der agilen Methoden findet sich eine Lösung für dieses Maniko: Aus einem losen „Wenn was ist; wenn

es Verbesserungsvorschläge gibt“ wird ein dedizierter Termin und aus dem Gespräch zwischen Tür und Angel ein systematisches und effektives Vorgehen. Das Team tauscht sich in einem regelmäßigen Rahmen über die Zusammenarbeit aus und kann dadurch frühzeitig Hemmnisse ausräumen sowie Verbesserungen umsetzen. Dies hat, aller Erfahrung nach, einen positiven Effekt auf die Arbeit des Teams.

Bei Scrum sind solche Retrospektiven sogar ein fester Bestandteil eines Sprints und können so von Iteration zu Iteration

zu einem immer besser werdenden Team beitragen. Außerhalb von Scrum ist diese Form des Rückblicks jedoch noch nicht weit verbreitet. Dabei sind Retrospektiven keineswegs an dieses Vorgehensmodell gebunden.

Eine Retrospektive ist dabei keine endlose Rückschau, bei der sich jeder Einzelne ungehemmt seinen Ärger von der Seele redet. Stattdessen richtet sie den Blick sehr gezielt auf einen bestimmten Zeitraum, wie den letzten Sprint oder die Entwicklung des letzten Meilensteins. Da-

bei sind Zeiträume von zwei Wochen bis maximal drei Monaten ein guter Rahmen. Neben einer Betrachtung von Zeiträumen können Retrospektiven aber auch für den Rückblick auf ein Ereignis – zum Beispiel die Durchführung einer Veranstaltung oder eine Marketingkampagne – genutzt werden, um daraus Anregungen für Wiederholungen oder ähnliche Veranstaltungen mitzunehmen.

Das Wesen einer Retrospektive

Wir neigen häufig dazu, bei Problemen die Schuld zuerst bei anderen zu suchen. Wir zeigen mit dem Finger auf die vermeintlichen Übeltäter, die ihrerseits zurückzeigen oder auf Dritte verweisen. Das schafft keine konstruktive Atmosphäre für ein Team, das lange zusammenarbeiten will und dabei idealerweise zusammenwächst und stärker wird. Retrospektiven sind „not a blame game“, sondern darauf ausgelegt, eine positive und konstruktive Atmosphäre zu schaffen, in der das Team die „Frage dahinter“ klären kann: Warum ist das Problem aufgetaucht und was müssen wir machen, damit es nicht wieder auftritt?

Neben dieser nach vorne gerichteten Sicht auf die Probleme lebt eine Retrospektive ebenso vom Aufzeigen der positiven Aspekte; also Dinge, die im Team gut laufen. Das Team soll sich die Frage stellen: Wie können wir sicherstellen, dass diese positiven Dinge erhalten bleiben und sich etablieren? Nach dem Motto: „Behalte das Gute und verbessere das, was nicht so gut funktioniert.“

Retrospektiven folgen immer zwei Regeln:

1. Die „Prime Directive“: Diese Regel greift auf die einfache Erkenntnis zurück, dass jeder im Team immer sein Bestmögliches tut – mit den Mitteln, die ihm in dem betreffenden Moment zur Verfügung stehen. Schuldzuweisungen sind also überflüssig.
2. Die „Vegas-Regel“: Diese Regel nimmt Bezug auf die berühmte Casino-Hochburg in Nevada mit dem Ausspruch „What happens in Vegas stays in Vegas!“ Sie schafft damit einen geschützten Raum, in dem sich jedes Teammitglied frei fühlen und offen sprechen kann. Nur so können auch eventuell schwieri-

ge Themen konstruktiv und wertschätzend angegangen werden. Die Themen einer Retrospektive bleiben also im Teilnehmerkreis; nur abgestimmte Inhalte gehen an Dritte heraus.

Der Facilitator

Ein weiterer Punkt spielt bei der Planung eine wichtige Rolle: In einer Retrospektive ist es wichtig, dass alle Teammitglieder auf Augenhöhe miteinander kommunizieren können. Dies kann schwierig sein, wenn ein Mitglied als Moderator die Zügel in die Hand nehmen muss und dadurch nur mit geteilter Aufmerksamkeit an der Diskussion teilnehmen kann – eventuell sogar in einem Zielkonflikt steht und damit vorschnell eine Lösung festlegt. Daher kann es empfehlenswert sein, eine externe Person mit der Moderation zu betrauen, einen sogenannten Facilitator. Der Facilitator kann insbesondere in schwierigen Diskussionen neutral agieren und somit die zielgerichtete Durchführung der Retrospektive sicherstellen.

Ebenso hat der Facilitator die Zeit im Blick, steuert das Team durch das Meeting und kann Abläufe, Themen und Entscheidungen für alle visuell deutlich machen. Je nach Länge des zu betrachtenden Zeitraums, der Erfahrung des Teams und dessen Redebedarfs sollten für eine Retrospektive eineinhalb bis drei Stunden

veranschlagt werden. Gerade bei Retrospektiven im Scrum-Umfeld, in dem Iterationen alle zwei oder drei Wochen stattfinden, kann auch eine Stunde oder weniger ausreichen.

Die fünf Phasen einer Retrospektive

Eine Retrospektive wird typischerweise in fünf Phasen unterteilt; sie kann aber beliebig an die Bedürfnisse des Teams oder – soweit vorab bekannt – an die zu besprechenden Themen angepasst werden. Das Team soll nicht der Methodik genügen, sondern die Methodik soll dem Team helfen, sich selbst und seine Arbeitsweise wahrzunehmen und zu verbessern.

Die Phasen der Retrospektive sind in *Abbildung 1* dargestellt und werden im Folgenden genauer erklärt. Dabei sind die Grenzen beziehungsweise Übergänge zwischen den einzelnen Phasen in den wenigsten Retrospektiven klar. Je nach Team und Thema kann der Facilitator hier variieren und unterschiedliche Vorgehensweisen nutzen. Jede Phase wird anhand einer Aktivität für die Teilnehmer interessant und manchmal auch unkonventionell durchgeführt, damit Retrospektiven spannend, kreativ, interaktiv und nicht wie alle anderen Meetings sind. Eine gute Sammlung von weit über 100 Aktivitäten bietet die Website www.retromat.org

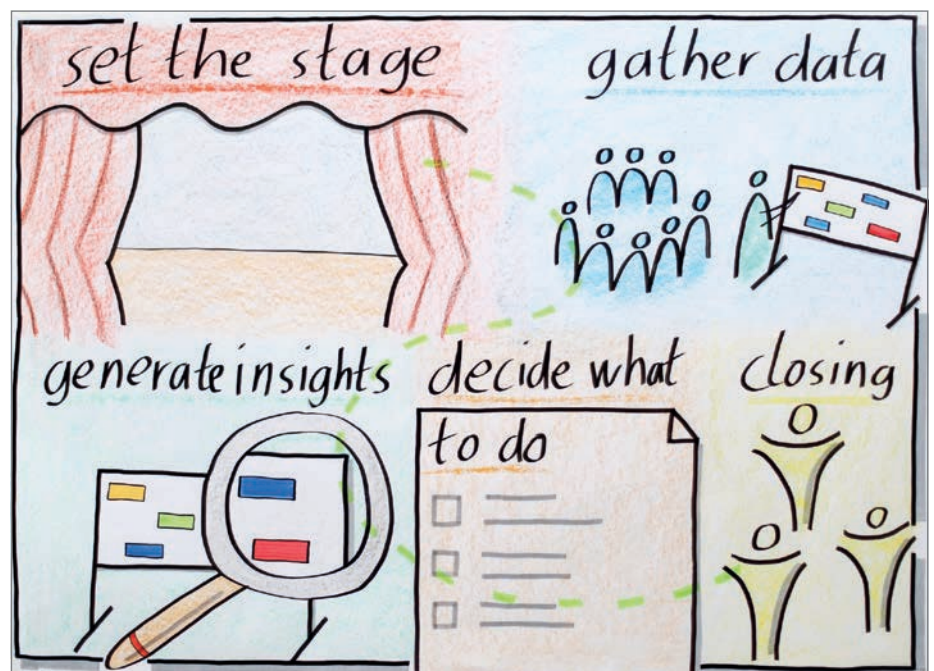


Abbildung 1: Phasen einer Retrospektive (Quelle: Thimo Fußbroich)

Phase 1: Set the stage

In dieser ersten Phase sollen die Teilnehmer im Meeting ankommen und werden vom Facilitator emotional abgeholt. Nachdem die wichtigsten Rahmenbedingungen und Regeln erklärt wurden, startet die erste Aktivität. Ziel ist es, in einem kurzen Blitzlicht die Stimmung im Team aufzunehmen und eine positive Ausrichtung für die Retrospektive zu schaffen. Hier kann auch nochmal das Ergebnis aus der letzten Retrospektive aufgenommen werden, um im weiteren Verlauf zu schauen, welche der vorgenommenen Themen wie umgesetzt werden konnten und ob dies den gewünschten Effekt hatte.

Zeitlich nimmt diese Phase etwa 5 % der veranschlagten Zeit in Anspruch. Dabei soll es nicht nur darum gehen, das Meeting zu starten, sondern jeder Teilnehmer erhält die Möglichkeit, sich auf die Retrospektive einzustellen. Zudem kommt jedes Teammitglied in dieser Phase zu Wort, auch um deutlich zu machen, dass jeder Einzelne – unabhängig von Position, Erfahrung, Hierarchie etc. – das Mandat hat, einen Beitrag zur Retrospektive zu leisten.

Phase 2: Gather data

In der zweiten Phase ist das Team aufgerufen, Eindrücke, Erfahrungen, Situationen etc. aus den vergangenen Wochen oder Monaten zu sammeln. Was ist gut gelaufen? Was war nicht so gut? Dabei geht es erst mal darum zu sammeln – nicht zu bewerten oder zu entscheiden, worüber gesprochen werden soll.

In diesen ca. 35 % des gesamten Meetings geht es darum, alle Themen zuzulassen, die die Teammitglieder in ihrer Arbeit beschäftigen. Das können technische, fachliche oder organisatorische Themen sein; Team-Interna oder Einflüsse von außen. Alles, was einen Einfluss – direkt oder indirekt – auf das Team hat, darf genannt werden, weil alle diese Themen Auswirkungen auf das Team haben.

Im Übergang zur dritten Phase werden die Themen zu größeren Themenblöcken zusammengefasst und priorisiert. Dabei ist darauf zu achten, dass die Blöcke nicht zu groß werden und nicht mehr zielführend in der Retrospektive besprochen werden können. Ebenso sollte im Folgen-

den nicht über Themen gesprochen werden, wenn das Team darauf keinen Einfluss hat – die wenigsten Teams können beispielsweise beschließen, mehr Gehalt zu bekommen, auch wenn dies ein guter Motivator sein kann.

Phase 3: Generate insights

Nachdem die Themen zu Blöcken geclustert und priorisiert wurden, geht es in dieser Phase darum, diese nacheinander zu besprechen. Dabei wird jedes Thema in einer im Vorfeld definierten Timebox besprochen, um zu verhindern, dass Diskussionen über einzelne Themen ausufern. Die nachfolgenden Themen haben zwar eine etwas niedrigere Priorität, aber auch sie sind wichtig und ihre Besprechung kann ebenfalls Verbesserungen für das Team bewirken.

Zu jedem Thema werden die Ursachen und Folgen für das Team diskutiert. Dabei sollten die Abhängigkeiten in der notwendigen Tiefe analysiert werden, um mit anschließenden Maßnahmen an die Ursache heranzukommen und nicht nur Symptome zu bekämpfen. Nur so kann die Arbeit des Teams nachhaltig verbessert werden.

Diese Phase nimmt damit etwa 45 % der Zeit in Anspruch und ist der zentrale Punkt der Retrospektive, in dem das Team seine Zusammenarbeit reflektiert und Entscheidungen für die Zukunft trifft.

Phase 4: Decide what to do

Nachdem das Team in der dritten Phase mögliche Maßnahmen identifiziert hat, um seine Arbeit und Zusammenarbeit künftig zu verbessern, geht es nun darum zu entscheiden, welche der diskutierten Maßnahmen umgesetzt werden sollen. Dabei ist es wichtig, dass sich das Team nicht zu viele Maßnahmen vornimmt. Jeder sollte die Möglichkeit haben, beschlossene Maßnahmen neben der eigentlichen Arbeit umzusetzen, da ansonsten der Nutzen der Retrospektive geschmälert wird.

Damit später nichts verloren geht, ist es wichtig, dass das Team für jede Maßnahme einen Verantwortlichen benennt. Damit ist klar, wer die Umsetzung betreut, und die Verbesserungsmaßnahmen gehen nicht in der täglichen Arbeit unter. Der Zeitanteil dieser Phase beträgt ungefähr 10 %.

Phase 5: Closing

Zum Abschluss blickt das Team auf die vergangenen Stunden zurück – es handelt sich also quasi um eine Retrospektive der Retrospektive. Was war gut an der Retrospektive? War es sinnvoll investierte Zeit? Was soll beim nächsten Mal anders gemacht werden?

Dieses Feedback gibt dem Facilitator die Möglichkeit, die nächste Retrospektive für das Team besser zu gestalten. Ebenso wichtig ist es in dieser Phase, die Ergebnisse der Retrospektive noch einmal kurz zusammenzufassen und das Team zu motivieren, die beschlossenen Maßnahmen zeitnah umzusetzen.

Fazit

Retrospektiven bieten als unkonventionelle Form von Meetings die Möglichkeit, dass sich ein Team offen und ehrlich über seine Arbeitsweise austauschen und sich selbst kontinuierlich verbessern kann. Auch wenn es kaum feste Kennzahlen gibt, um den Erfolg von Retrospektiven zu messen, haben sie in aller Regel einen positiven Effekt auf ein Team. Es gilt also: Einfach mal ausprobieren und selbst überzeugen lassen.

Über den Autor

Ich bin seit über zehn Jahren Entwickler im Oracle-Umfeld – SQL, PL/SQL und APEX. Seit einigen Jahren bin ich auch als Projektleiter tätig und befasse mich gerne mit Soft-Skill-Themen. Dazu zählt auch, dass ich verschiedene Teams als Facilitator in Retrospektiven begleite.



Thimo Fußbroich
Thimo.fussbroich@opitz-consulting.com

Data Warehouse und Business Intelligence im Mittelstand

Hans-Peter Weih, Hochschule Mainz

Vor 32 Jahren veröffentlichten Barry Devlin und Paul Murphy [3] im IBM Systems Journal erstmals eine Data-Warehouse-Architektur, die heute noch von jedem Data-Warehouse-Architekten als solche erkannt werden würde. In den Jahren danach haben sich Business-Intelligence-Systeme weiterentwickelt, mit einem Data Warehouse in der Mitte einer Hub-and-Spoke-Architektur, an deren Anfang die Datenintegration und am Ende die Analyse der Daten und der unmittelbare Beitrag zur Entscheidungsunterstützung innerhalb der Firma stehen. Mit dem Aufkommen von Big Data sowie der Renaissance von maschinellem Lernen und neuronalen Netzen verschwand plötzlich das Data Warehouse aus dem allgemeinen Interesse bzw. wurde mit immer wiederkehrender Regelmäßigkeit für tot erklärt; die BI Community fokussierte sich mehr auf Data Lakes, Analytics und Data Science. Doch damit ist das klassische Data Warehouse im Rahmen der BI nicht tot, sondern bildet nach wie vor für viele Aufgaben in den Firmen das Rückgrat des Reportings und vieler Datenanalysen, da es ein ausgereiftes und nutzbringendes Konzept ist. Es hat erfolgreich Big-Data-Ansätze in die Architektur mit aufgenommen und breitet sich vor allem im Mittelstand immer weiter aus. Business Intelligence wird in zunehmendem Maße an Universitäten und Hochschulen in der Breite gelehrt und trägt damit wesentlich dazu bei, die Ideen von Devlin und Murphy (*siehe oben*) in den Unternehmen zu etablieren.

Das Data Warehouse – lebendig wie eh und je

Vor 20 Jahren beschäftigte ich mich das erste Mal mit Data Warehousing, da ich für meinen damaligen Arbeitgeber, die deutsche Niederlassung einer Versicherungsgesellschaft, das Reporting über das Neu- und Bestandsgeschäft für das Controlling aufbauen sollte. Es gab kaum Literatur darüber, wie man so etwas am besten macht. Speicherplatz war rar und teuer, die Softwareprodukte im Bereich der Business Intelligence waren extrem teuer und wurden nur von Firmen, die ein entsprechendes IT-Budget hatten, eingesetzt. Durch den Besuch von Anwenderforen der Softwarehersteller sowie von Veranstaltungen des TDWI (The Data Warehouse Institute, ein herstellerunabhängiger Verein für BI und Analytics) kam man schnell in engen Kontakt mit anderen BI-Anwendern und stellte fest, dass DWH und BI in vielen Firmen gewinnbringend eingesetzt wurde und dass die Probleme in den anderen Firmen ähnlich waren und auf ähnliche Weise gelöst wurden. Auffallend war auch, dass BI in seiner

ganzen Breite vornehmlich von großen Firmen mit einem entsprechenden IT-Budget eingesetzt wurde. Kleine Firmen lösten ihre Reporting-Anforderungen häufig mit Excel oder setzten kleine Inselfösungen, zum Beispiel für die Planung der Budgetierung ein. Mit dem Aufkommen von Big Data war Data Warehousing in vielen Firmen bereits eine etablierte Technologie, sodass die Community sich gerne dem neuen Hype Big Data, dem aufkommenden, wiederbelebten maschinellen Lernen und den neuronalen Netzen zuwandte. Aus den Konferenzen und Tagungen entfielen zunehmend die Beiträge über klassisches BI und der Fokus wurde immer mehr auf Big Data, Analytics und Data Science gesetzt.

Trotz des Schwenks der Aufmerksamkeit ist das Data Warehouse und die klassische BI noch längst nicht abgeschrieben oder tot, wie häufig in diversen Artikeln proklamiert, sondern erfolgreich wie eh und je. Dies liegt vor allem daran, dass *„Big Data nicht neu ist, sondern nur die Fortschreibung bekannter Konzepte in Kombination mit neuen Technologien. Big Data ist insofern keine Revolution, sondern vielmehr*

eine Stufe in der Evolution zu einer stärker datenorientierten Unternehmensrealität.“ [2]

Aktuell entdeckt der Mittelstand zunehmend das Thema Data Warehouse und BI für sich und implementiert erfolgreich Lösungen. Während die großen Firmen versuchen, aus dem Thema „Big Data“ Gewinn zu schöpfen, legt der Mittelstand mit einem Data Warehouse die Grundlagen für eine einfache, aber effektiv funktionierende BI innerhalb des Unternehmens. Die Aktivität des Mittelstandes hat mehrere Gründe:

- BI-Software und benötigter Speicherplatz sind bezahlbar geworden. Vor allem Commercial-Open-Source-Produkte und Cloud-Anwendungen haben hier vielen Firmen den Weg in die BI geöffnet. Datenbanksysteme wie MySQL und PostgreSQL, Integrationstools wie Talend, Reporting wie Jasper Reports oder OLAP wie von Pentaho sowie inzwischen günstige kommerzielle Produkte erleichtern den Einstieg, ohne sich sofort in horrenden Lizenzkosten zu stürzen.
- Business Intelligence ist inzwischen Standard-Ausbildungsfach an den Universitäten

ten und Hochschulen; neben Informatik vor allem auch in den Wirtschaftsfächern. Kern der Lehre ist hier immer der Aufbau eines Enterprise Data Warehouse durch Datenintegration aus verschiedenen operativen Quellen, aus dem die analytischen Anwendungen gespeist werden. Somit werden die Ideen der BI auch in viele Firmen getragen und sind nicht nur Spielereien von IT-Fachleuten, sondern sie werden auch vom Management als notwendig erachtet und gefördert. Durch die Ausbildung sowie verschiedene Zertifizierungsprogramme diverser Organisationen stehen inzwischen auch viel mehr Fachkräfte mit entsprechenden Kenntnissen zur Verfügung.

- Viele Firmen haben schlicht und ergreifend kein „Big Data“ oder keine Problemstellung, die Big Data erfordert, und damit auch keine Notwendigkeit, sich damit auseinanderzusetzen. Ihre Probleme lassen sich mit einem Data Warehouse und angebundenen Tools einfach lösen.
- Nicht zuletzt gibt es jede Menge Literatur, Use Cases und Vorlagen zum Thema „Data Warehousing und BI“; viele Beratungsfirmen und Softwarehersteller fokussieren sich mit „BI out of the Box“-Angeboten auf den Mittelstand.

Was ist ein Data Warehouse?

Ein Data Warehouse ist eine vom operativen Datenbestand getrennte, logisch zentralisierte, dispositive Datenhaltung. Inmon [4] definierte ein Data Warehouse als eine Sammlung von Daten, die integriert, themenorientiert und nicht volatil sind sowie einen Zeitraumbezug besitzen („integrated, subject oriented, non volatile and time variant“). Dies bedeutet, dass die Daten im Data Warehouse aus verschiedenen internen und externen Quellen zu einer inhaltlich widerspruchsfreien Datensammlung zusammengefasst werden (Integration) und grundsätzlich an den Informationsbedürfnissen des Managements ausgerichtet sind (Themenorientierung). Im Gegensatz zu operativen Systemen sollen sich die Daten nicht verändern, sondern dauerhaft für Analysen gespeichert werden (Nicht-Volatilität) und die Daten sollen alle mit entsprechenden Zeitstempeln versehen sein (Zeitraumbezug), was neben einfachen Zeitstempeln zum Zeitpunkt der Erstellung der Daten auch Gültigkeitszeit-

räume umfassen kann (bitemporale Daten). *Abbildung 1* zeigt das grobe Schema einer typischen BI-Hub-and-Spoke-Architektur, in der das Data Warehouse als Hub dient, über den die gesammelten und integrierten Daten für die weiteren Anwendungen weiterverteilt werden. Im sogenannten ETL-Prozess (Extract Transform Load) werden die Daten aus den diversen operativen Quellsystemen in das Data Warehouse integriert. Data Marts bilden Teilausschnitte aus dem Data Warehouse für bestimmte Analyseziele, wie zum Beispiel Kostenanalyse oder Analyse der Vertriebskanäle.

Die Notwendigkeit, Reports für verschiedene Bereiche des Unternehmens zu erstellen, Geschäftsentwicklungen zu monitorieren und Planungsprozesse zu unterstützen sind auch für den Mittelstand nicht neu, und das oben beschriebene Data Warehouse ist eine ideale Grundlage, um vor allem das Controlling und den Vertrieb hierbei zu unterstützen. Der jährliche BARC BI Survey [1] zeigt auch, dass BI-Systeme im Mittelstand vor allem im Standard Reporting, Ad Hoc Reporting und in multidimensionaler Analyse weit verbreitet sind und nur wenige Firmen einen Einsatz als nicht erforderlich sehen. Tools zur weiteren Verarbeitung wie Planung und Budgetierung sind jedoch noch

nicht weit verbreitet. Controlling basiert hier in vielen Firmen vornehmlich auf Tabellenkalkulationsprogrammen, die aus operativen Quellen oft per manuellem Extrakt mit Daten gefüllt werden und dann umfangreiche Berechnungen vornehmen. Spreadsheets mit Gigabyte an Dateien und Berechnungszeiten von mehreren Minuten sind hierbei nicht ungewöhnlich.

Data Warehouse und Big Data

Mit Big Data werden sowohl besondere Datenbestände als auch Methoden, um diese Datenbestände zu verarbeiten, bezeichnet. Big Data liegt dann vor, wenn die Möglichkeiten der klassischen Datenhaltung, -verarbeitung und -analyse auf konventioneller Hardware überstiegen werden [2]. Sehr häufig wird Big Data durch die drei „V“ in der Literatur charakterisiert: Volume, Velocity und Variety. Volume bezieht sich auf die gestiegene Menge der auswertungsrelevanten Daten wie Messergebnisse von Sensoren, Protokolldateien und Transaktionen. Besonders das Internet hat hier zu einem Zuwachs beigetragen. Aber nicht nur die Datenmenge, sondern auch die Ge-

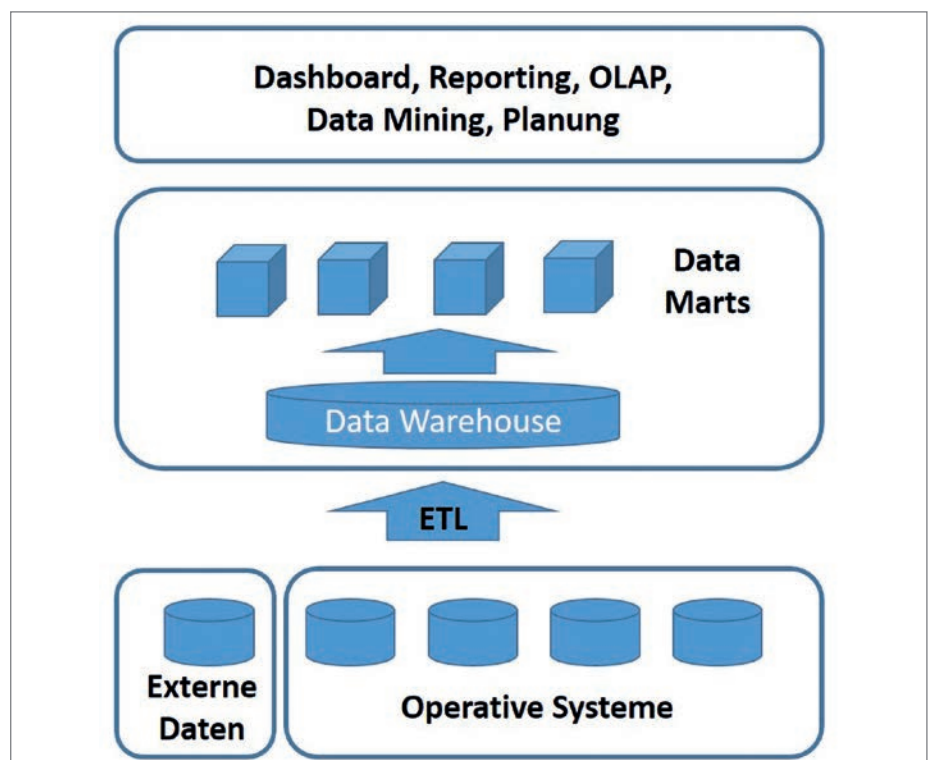


Abbildung 1: Data-Warehouse-Referenzarchitektur (Quelle: Hans-Peter Weih nach [5])

schwindigkeit (Velocity), mit der die Daten entstehen, ist gestiegen, sodass die Daten nicht immer zeitnah analysiert und integriert werden können. Das dritte V bezieht sich auf die Vielfalt der Datenformate (Variety), die zur Verfügung stehen und nicht mit den klassischen Methoden gespeichert und analysiert werden können. Die Daten sind oft polystrukturiert – im Gegensatz zu transaktionsorientierten Daten wie zum Beispiel in ERP-Systemen. Um den Anforderungen von Big-Data im Data-Warehouse-Umfeld gerecht zu werden, wird die Standardarchitektur um einen sogenannten „Data Lake“ erweitert, in dem die Daten kontinuierlich gesammelt werden und sowohl für eine Integration ins Data Warehouse als auch für Analysen in einem Analytic Lab zur Verfügung stehen. *Abbildung 2* zeigt die Integration von Big Data in einer BI-Umgebung.

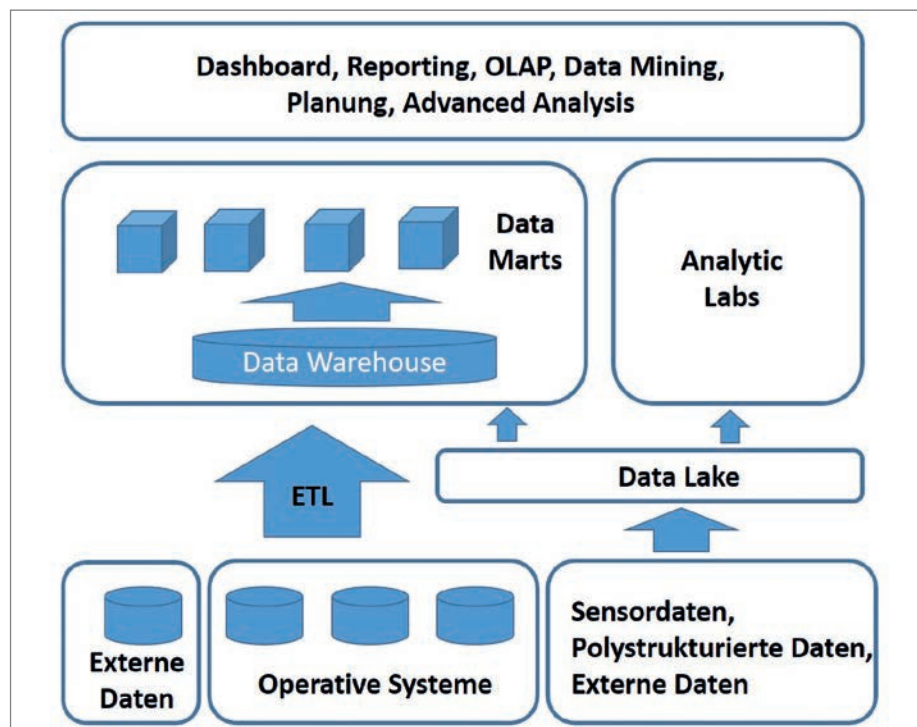


Abbildung 2: Data-Warehouse-Referenzarchitektur erweitert um Data Lake (Quelle: Hans-Peter Weih nach [1])

Data Warehouse und BI in der Ausbildung

Laut BARC BI Survey sind fehlende Ressourcen eines der Hauptprobleme bei der Implementierung von BI-Systemen im Mittelstand. Business Intelligence ist jedoch seit einigen Jahren Standardfach an vielen Universitäten und Hochschulen. An der Hochschule in Mainz ist Business Intelligence integraler Bestandteil der Pflichtvorlesung „Unternehmen und IT“ für alle BWL-Studiengänge inklusive praktischer Anwendung. Für den berufsintegrierten Studiengang Wirtschaftsinformatik ist ein komplettes Modul „Business Intelligence“ verpflichtend. Kern hierbei ist die Hub-and-Spoke-Architektur, die Datenmodellierung im Data Warehouse, der Integrationsprozess und die Möglichkeiten der Auswertungen und Analysen der Daten. Die Partner der Hochschule sind sehr häufig mittelständische Firmen, bei denen Business Intelligence, wenn überhaupt, nur rudimentär angewandt wird. Durch den berufsintegrierten Studiengang erleben wir es sehr häufig, dass Studenten die Ideen in die Firmen tragen, da sie die Notwendigkeit von Veränderungen sehen. Im Rahmen von Bachelor- oder Masterarbeiten werden häufig erste Ansätze umgesetzt und weitere Anwendungen innerhalb der Firma evaluiert. In den Firmen findet man häufig Situationen wie im nächsten Abschnitt beschrieben.

Data Warehouse im Mittelstand

Auf der TDWI-Tagung 2018 stellten Chameni und Gluchowski [1] eine kleine Studie über den Einsatz von BI bei mittelständischen Unternehmen vor. Hierbei konnte man sehr gut sehen, dass viele Unternehmen nur Teile einer BI-Architektur umsetzen, indem zum Beispiel Daten direkt aus den operativen Systemen gezogen werden, um Data Marts in OLAP-Systemen zu füllen, oder es wurden mehrere analytische Datenbanken parallel gefahren, was dem Integrationsgedanken entgegenläuft.

Das folgende Beispiel ist ein typischer Fall dafür, wie Data Warehousing und Business Intelligence im Mittelstand einfach und erfolgreich umgesetzt werden können. Es handelt sich hierbei um ein aktuelles Projekt mit einer Firma, die ausschließlich über das Internet Dienstleistungen (Content-Erstellung) vermittelt und abrechnet. Sie ist in den letzten Jahren stark gewachsen; aktuell werden pro Jahr ca. 500.000 Aufträge abgerechnet. Der Bestand an aktiven Kunden beträgt ca. 40.000. Die Aufträge werden in einem eigens entwickelten Verwaltungssystem in verschiedenen Mandanten bearbeitet. Da die Datenmenge noch überschaubar ist, wird das Reporting im Controlling ak-

tuell in Excel durchgeführt. Hierzu werden die Daten aus den Produktionssystemen auf Auftragsebene extrahiert und in Excel geladen. Dort beginnt nun die Verarbeitung mit der Berechnung von Kennzahlen und der anschließenden Aggregation nach verschiedenen Dimensionen, der sogenannten Pivotierung, um die Ergebnisse der Aufträge darüber analysieren zu können. Typische Fragestellungen sind hier die Mengen und Umsätze über die verschiedenen Vertriebskanäle, Länder und Kundengruppen. Mit den Monaten im laufenden Jahr werden auch die Excel Sheets immer unhandlicher und es erfordert jede Menge Zeit, um damit zu arbeiten. Problematisch ist ebenfalls die Integration von Daten aus anderen Systemen, etwa aus Helpdesk oder Mailing-Systemen, die ihre Daten in verschiedenen Formaten liefern. Die manuelle Erstellung der Datenauszüge führt häufig zu Fehlern, ebenso wie die Abgrenzung der Daten nach den Reporting-Zeiträumen.

Die Einführung eines Data Warehouse ist hier eine sinnvolle Lösung, da verschiedene Mandanten und Quellen integriert werden konnten, indem die Datenformate und Schlüssel vereinheitlicht, KPIs konsistent definiert und berechnet, Dimensionen wie Vertriebsstrukturen einheitlich erstellt und Zeitreihen gespei-

chert wurden. In einem nächsten Schritt werden die Daten aggregiert und können somit in einfacher Form an das Controlling zur Darstellung der Ergebnisse weitergegeben werden. Alle manuellen Schritte konnten mit entsprechenden Tools automatisiert werden, was auch die Fehlerquote deutlich senkte. Mit Talend als Datenintegrationstool in der Open-Source-Variante und MariaDB in der Cloud entfielen die Lizenz- und Wartungskosten für die Software; bezahlt werden müssen nur die aktuell ca. 10 GB Speicher in der Cloud. Eine externe Beratung im Umfang von 10 Tagen analysierte die Situation und entwickelte mit der Firma ein entsprechendes Konzept zur Umsetzung der ersten Aufgabe. Die Entwicklung konnte von einem internen Mitarbeiter gestemmt werden, der nach einer einwöchigen Einarbeitungszeit in die Tools mit der Umsetzung beginnen und nach etwa einem Monat die Ablösung des ersten Reports präsentieren konnte.

Das Data Warehouse dient jetzt als Single Point of Truth für das Reporting und konnte in einem zweiten Schritt als hervorragende Quelle zur Analyse des Kundenverhaltens mithilfe maschinellen Lernens genutzt werden, da die Daten über einen hohen Qualitätsgrad verfügen. Durch die Integration von weiteren Systemen wie Helpdesk und Mailing-Systemen kann nun Schritt für Schritt ein einheitliches Bild der Kunden aufgebaut werden. Als Front End zur Darstellung und Aufarbeitung der Ergebnisse dient immer noch Excel, da es den Benutzern vertraut ist und man damit einfache Auswertungen machen kann.

Durch die Einführung des Data Warehouse hat sich nicht nur der Prozess zur Erstellung der Daten deutlich beschleunigt, auch die Datenqualität hat sich enorm verbessert. Probleme der unterschiedlichen Systeme wurden identifiziert und bereinigt, die Daten können vielfältig genutzt und die Mitarbeiter sich mehr auf die Analyse als das Erstellen der Zahlen konzentrieren, was eine größere Mitarbeiterzufriedenheit bringt. In einem weiteren Schritt wurde versucht, mittels Data Mining Methoden eine Churn-Analyse der Kunden durchzuführen. Hierbei bildete das Data Warehouse eine zuverlässige und qualitätsgesicherte Quelle für die Analysen.

Die Probleme, die bei diesem Projekt auftraten, sind die typischen Probleme,

die auch im BARC BI Survey genannt wurden: fehlende Ressourcen im Projekt-Team, unklare Anforderungen und softwareseitige Probleme. Sehr häufig ist im Mittelstand die Personaldecke besonders bei IT-Fachkräften sehr dünn, sodass nicht einfach ein Vollzeit-Projektteam aus der Taufe gehoben werden kann. Vor allem die Fachkraft, die den Ansatz implementieren sollte, wurde immer wieder zu anderen Arbeiten abgezogen.

Der Einsatz der Open Source Tools läuft nicht immer reibungslos, so dass sich einige Problemlösungen trotz aktiver Community als sehr aufwändig gestalteten. Mit der Implementierung der Lösung musste auch sichergestellt werden, dass die Lösung einfach zu handhaben ist, um problemlos in den operativen Betrieb überführt zu werden. Dies sollte man beachten, um eine Lösung nach erfolgreicher Einführung dauerhaft zu etablieren.

Big Data ist bei der Firma momentan noch kein Thema. Weder die Größe des Geschäftes noch die Daten, die die Kunden über das Internet hinterlassen, sind so groß, dass sie mit konventionellen Mitteln nicht mehr zu beherrschen sind. Für die Firma ist es im ersten Schritt viel wichtiger, die klassischen Controlling-Aufgaben in den Griff zu bekommen und sich einen Einblick in die Struktur und das Verhalten der Kunden zu verschaffen, um so einen Mehrwert für das Geschäft zu generieren.

Fazit

Das Ende des Data Warehouse wurde schon öfter verkündet. Trotzdem zeigt das Data Warehouse eine bemerkenswerte Langlebigkeit und ist bei vielen Unternehmen immer noch das Rückgrat von Reporting und Datenanalyse, was die Stärke des Ansatzes verdeutlicht. Das Data Warehouse befriedigt die meisten Informationsbedürfnisse innerhalb einer Firma und breitet sich zusehends auch in mittelständischen Firmen aus, die die Vorteile des Data-Warehouse-Ansatzes schätzen und es gewinnbringend nutzen können.

Es dürfte jedoch klar sein, dass die Informationsbedürfnisse von Firmen im Rahmen der Digitalisierung nicht durch ein Data Warehouse allein gedeckt werden können. Hierfür wird der Data-Warehouse-Ansatz um das Konzept des

Data Lake erweitert, das Analysen im Big-Data-Umfeld zulässt. Das Herz jedoch bleibt immer das Data Warehouse mit seinen wesentlichen Geschäftsinformationen und integriertem Datenmodell.

Quellen

- [1] Prof. Dr. P. Chameni und Prof. Dr. P. Gluchowski 2018: Business Intelligence für den Mittelstand – grundlegende BI-Architekturen und Konzepte. - TDWI Tagung 2018 in München
- [2] C. Dittmar, C. Felden, R. Finger et al. 2016: Big Data – Ein Überblick – dpunkt Verlag, Heidelberg.
- [3] Dr. Barry Devlin 2018: Thirty Years of Data Warehousing – Business Intelligence Journal Vol. 23 No.1 Seite 12 – 24.
- [4] Inmon, William H. 1992: Building the Data Warehouse, QED Information Sciences.
- [5] Hahne, Michael 2014: Modellierung von Business Intelligence Systemen, dpunkt Verlag, Heidelberg

Über den Autor

Hans-Peter Weih war neunzehn Jahre bei der deutschen Niederlassung der Standard Life Versicherung tätig und dort als Teamleiter für den Aufbau des Data Warehouse und der Business Intelligence verantwortlich. Seit einigen Jahren unterrichtet er an der Hochschule Mainz Business Intelligence, Datenbanken und Data Science und berät freiberuflich mittelständische Unternehmen im Bereich Business Intelligence. Außerdem organisiert er zusammen mit weiteren Kollegen den Round Table der TDWI (The Data Warehouse Institute, www.tdwi.eu) in Frankfurt.

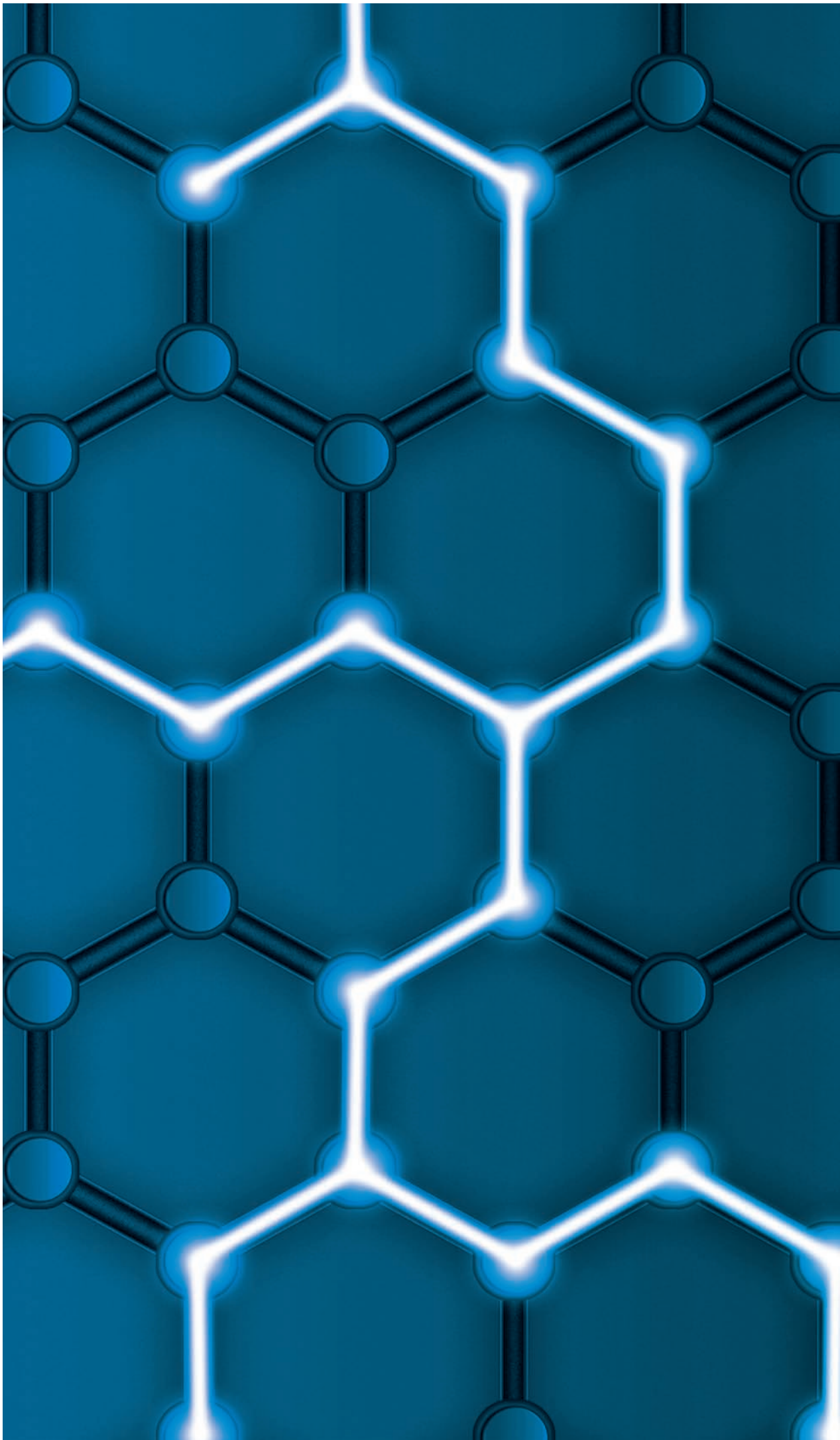


Hans-Peter Weih
hans-peter.weih@hs-mainz.de

USF — USCB

NEWS

01/2021



Analyse von Blockchain-Daten

Evgenia Julia Rosa und Detlef Egbert Schröder, Oracle Deutschland B.V. & Co KG

Die Blockchain-Technologie hat die Hype-Phase überwunden und sich in verschiedenen Bereichen bereits etabliert. Nun ist es an der Zeit, aus den in der Blockchain abgelegten Informationen die Datensätze zu heben und sie mit geeigneten Werkzeugen gewinnbringend fürs Geschäft zu analysieren.

Der Beitrag zeigt, wie die Oracle Analytics Cloud und die autonome Datenbank im Zusammenspiel mit der Oracle-Blockchain-Plattform visuelle Analyse der Blockchain-Daten möglich machen.

Nach einem anfänglichen Hype im Bereich Kryptowährungen hat die Blockchain-Technologie schon längst ihren Platz in Anwendungen verschiedener Industrien wie Logistik, Handel, Energie, Finanzen, Gesundheitswesen sowie in öffentlichen Verwaltungen gefunden.

Eigenschaften und Anwendungsbereiche der Blockchain-Technologie

Die Blockchain-Technologie weist Eigenschaften auf, die sie für den Einsatz in bestimmten Szenarien interessant machen.

- **Dezentraler Aufbau** ist eines der wesentlichen Merkmale der Blockchain. Daten werden auf viele Knoten im Blockchain-Netzwerk verteilt. Das Netzwerk wird aus mehreren gleichbe-

rechtigten Teilnehmern gebildet, die jeweils über eine vollständige Kopie der Blockchain verfügen. Das sorgt unter anderem für Ausfallsicherheit.

- **Unveränderlichkeit:** Die einzelnen Blöcke einer Blockchain enthalten in der Regel Transaktionsdaten und können nur angehängt, aber nicht verändert werden. So bleibt eine Transaktion in der Blockchain für immer bestehen und kann nicht geändert werden.
- Der dezentrale Aufbau einer Blockchain bildet die Grundlage für **Transparenz und Nachvollziehbarkeit**. Jeder Teilnehmer verfügt über eine immer aktuelle Kopie der Blockchain und kann so alle getätigten Transaktionen und den aktuellen Status nachvollziehen.
- **Kryptografie-basierte Sicherheit:** Die modernen, asymmetrischen Verschlüsselungstechnologien sorgen für einen hohen Grad an Sicherheit. Manipulationen an Daten in der Blockchain sind so gut wie unmöglich.

- **Konsensbildung:** Die Konsensmechanismen stellen sicher, dass nur zulässige Datensätze akzeptiert werden (Validierung) und dass die gespeicherten Versionen der Blockchain auf allen Knoten identisch sind.
- Die sogenannten **Smart Contracts** stellen die Business-Logik (bzw. den Programm-Code) der Blockchain dar. Durch sie werden die Geschäftstransaktionen an der Blockchain ausgeführt.

Die Anwendungsbereiche der Blockchain-Technologie sind sehr vielfältig und gehen von Nachvollziehbarkeit der Lieferketten über Herkunftsnachweise der Materialien bis hin zu Fälschungssicherheit von Medikamenten oder Identitätsnachweisen. Überall, wo mehrere Teilnehmer in einen Geschäftsprozess involviert sind und Transparenz und Nachvollziehbarkeit eine wichtige Rolle spielen, kann Blockchain-Technologie ihre Vorteile entfalten.

Es gibt mittlerweile viele Blockchain-Implementierungen auf dem Markt, die vom

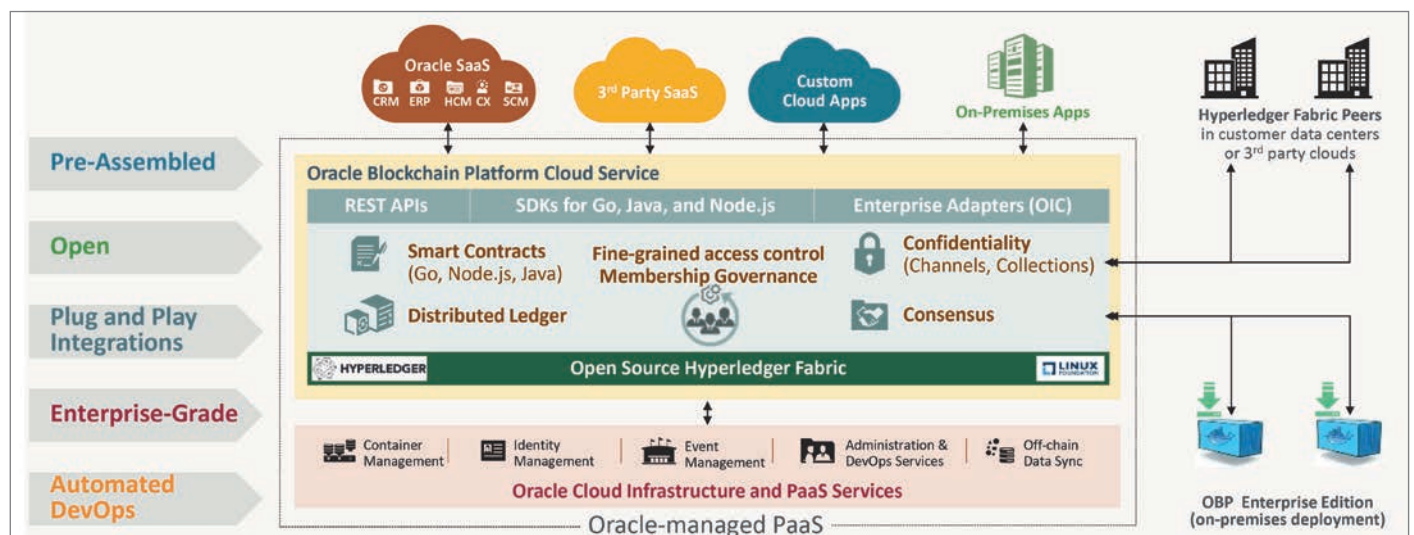


Abbildung 1: Oracle-Blockchain-Plattform (Quelle: Oracle)

Typ her öffentlich (**Public**: jeder kann teilnehmen und kann anonym bleiben) oder geschlossen (**Private**: Teilnehmer werden durch eine zentrale Instanz zugelassen und sind im Netzwerk bekannt) sein können – sie können auch branchenspezifisch sein.

Oracle-Blockchain-Plattform

Oracle bietet seinen Kunden eine Blockchain-Plattform an, die auf einer bewährten Open-Source-Implementierung basiert und durch zusätzliche Werkzeuge die Entwicklung und den Einsatz der Blockchain-Anwendungen effektiver und einfacher macht. Die Basis der **Oracle-Blockchain-Plattform** [1] bildet das **Hyperledger-Fabric**-Projekt (HLF) der Linux Foundation [2]. Diese Implementierung stellt eine Private Blockchain dar und hat sich bereits in vielen Bereichen bewährt. Die Zusatzfunktionalitäten, mit denen Oracle diese Kernimplementierung anreichert, stehen vor allem für Produktivität, Einfachheit der Bedienung und die Unternehmensreife. So wird die Oracle-Blockchain-Plattform sowohl als **Cloud-Lösung** als auch für den Einsatz im kundeneigenen Rechenzentrum (**On-Premises**) angeboten. Dabei ist auch ein **hybrider Ansatz** möglich: Die Teilnehmer des gleichen Blockchain-Netzwerks können ihre Knoten sowohl in der Cloud als auch On-Premises betreiben, die Hyperledger-Fabric-Native-Implementierung nutzen oder eine Hyperledger-Fabric-basierte Implementierung eines anderen Anbieters einsetzen (siehe Abbildung 1).

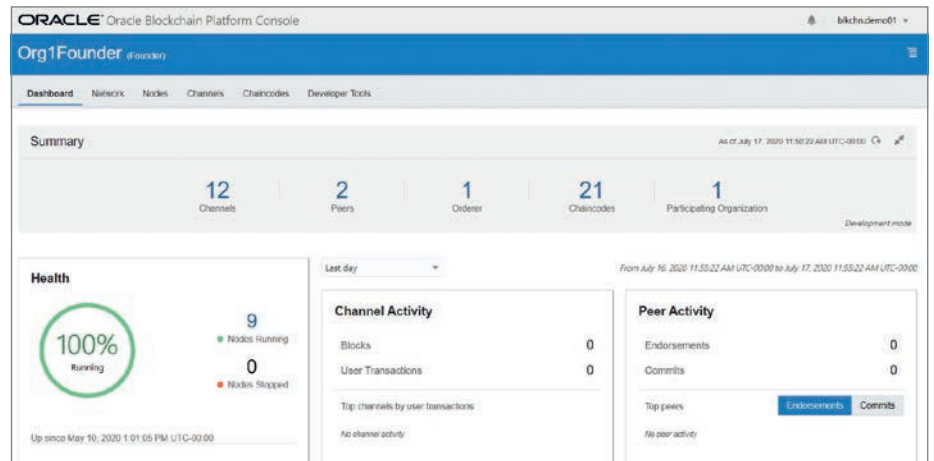


Abbildung 2: Grafische Administrationsoberfläche der Oracle-Blockchain-Plattform (Quelle: Oracle)

Die Cloud-Variante der Oracle-Blockchain-Plattform bietet alle Vorteile der Oracle-Cloud-Infrastruktur wie Performanz, Skalierbarkeit und Ausfallsicherheit bei entfallendem Aufwand für die Installation und Wartung. Außerdem können die Blockchain-Anwendungen sehr einfach mit anderen Unternehmensanwendungen integriert werden. Dazu steht die **Oracle Integration Cloud** mit der Vielzahl an Adaptoren und der **Visual Builder** für die grafische Oberflächenentwicklung zur Verfügung.

Die **Web-basierte Administrationsoberfläche** (siehe Abbildung 2) bietet den Nutzern auch ohne Entwickler-Hintergrund eine einfache Möglichkeit, ein Hyperledger-Fabric-Blockchain-Netzwerk aufzusetzen und zu konfigurieren. Über die grafische Oberfläche werden auch die Channels (die Blockchain Ledgers beinhalten) eingerichtet, der Chain-

code installiert sowie weitere HLF-spezifische Aufgaben durchgeführt. Über die Web-Oberfläche ist ebenfalls die Einsicht in die Inhalte einzelner Blöcke der Blockchain möglich.

Für die Entwickler des Chaincode (wie Smart Contracts in der HLF-Architektur genannt werden) und der Client-Applikationen stehen mehrere **Entwickler-Werkzeuge**, **Programmiersprachen** und **Software Development Kits (SDK)** zur Verfügung, die die Entwicklung schneller und produktiver machen. Entwickler können auf zahlreiche Code-Beispiele, Templates und Test-Werkzeuge zurückgreifen, die direkt über die Console heruntergeladen werden können.

Analyse von Transaktionsdaten

Die Transaktionsdaten, die in den einzelnen Blöcken der Blockchain gespeichert werden, können als Geschäftsdaten für

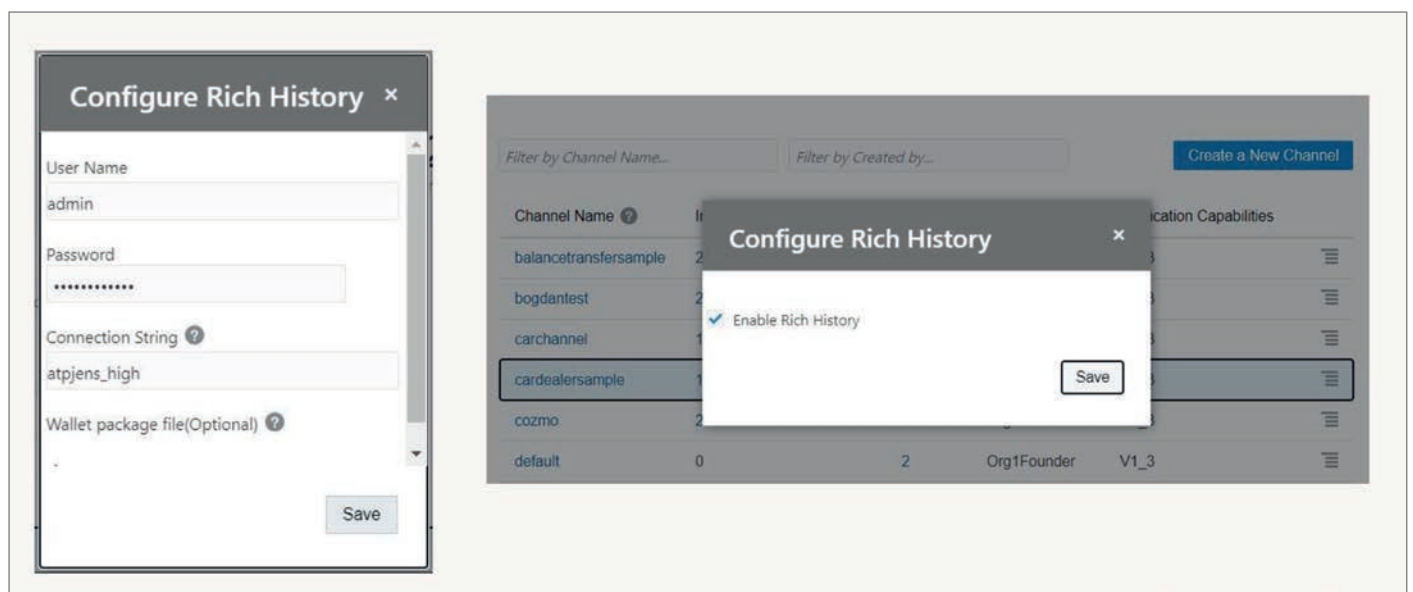


Abbildung 3: Einfache Konfiguration von Rich History Database. (Quelle: Oracle). Links: Einrichtung der Datenbankverbindung. Rechts: Konfiguration des Channels

Auswertungen interessant sein. Doch das Auslesen dieser Daten aus einer Blockchain per Chaincode kann bei größeren Datenmengen aufwendig und nicht performant genug sein. Für analytische Auswertungen von geschäftsrelevanten Daten gibt es auf dem Markt zwar viele Werkzeuge mit ausagekräftigen Dashboards und Reportfunktionen. Diese setzen jedoch in der Regel auf relationalen Datenbanken auf, die ja bekanntlich von ihrer Architektur her mit großen Datenmengen sehr performant umgehen können. Bei den Blockchain-Daten sind die Auswertungsmöglichkeiten technologiebedingt eingeschränkt.

Um Abhilfe hierfür zu schaffen, bietet Oracle mit dem Merkmal „Rich History Database“ eine Möglichkeit, die in der Blockchain gespeicherten Daten auf bewährte Art und Weise performant und nutzerfreundlich auszuwerten.

Mit einer sehr einfachen Konfiguration in der Web-Oberfläche (siehe Abbildung 3) kann der Anwender angeben, ob die Daten einer Blockchain (eines Channels) in eine relationale Oracle-Datenbank synchronisiert werden sollen.

Nach jeder stattgefundenen Transaktion werden die Daten folglich in einem dafür vorgesehenen Datenbank-Schema abgespeichert. Die Auswertung der Transaktionsdaten ist dann mit jedem für den Anwender geläufigen Analytics-Werkzeug, das auf die Oracle-Datenbank zugreifen kann, möglich. Die Abbildung 4 zeigt die Funktionsweise des Rich-History-Database-Merkmals.

Nachfolgend wird dargestellt, wie mit den Oracle-Analyse-Werkzeugen die Blockchain-Daten analysiert werden können.

Oracle Self Service BI

Oracle hat ein komplettes Spektrum an Auswertungswerkzeugen. Dies reicht von Standard Reporting, das immer noch in Print beziehungsweise PDF stattfindet, über Dashboard und Mobile sowie Scorecarding und Microsoft Office bis hin zu Self Service BI. Alle diese Komponenten aus der Oracle Analytics Cloud [3] (siehe Abbildung 5) basieren auf denselben Metadaten und einer einheitlichen Informations- und Technologie-Basis, sodass ein Wechsel des Zugangs zu den Daten, also egal ob Self Service oder

Print, die gleichen Ergebnisse liefert. Diese Flexibilität des Frontends wird durch die Möglichkeit, unterschiedliche Datenquellen als Federated Datasources zusammen zu analysieren, noch erweitert.

Im Folgenden wird die Self-Service-Komponente aus dieser Suite als Analysewerkzeug für die Auswertung von Blockchain-Daten verwendet.

Oracle Data Visualization zeichnet sich durch einen intuitiven Zugang aus, der es auf einfache Weise ermöglicht, Daten aus verschiedenen Quellen, bereits existierenden Schnittstellen und neuen, eventuell lokalen Daten zusammen zu analysieren und mit wenigen Schritten auch Machine Learning auf diese Daten anzuwenden.

Unterstützt wird dies durch eine große Vielfalt an grafischen Visualisierungsmöglichkeiten, die es in diesem Fall erlaubt, auch die Beziehungen und Zusammenhänge optisch darzustellen.

Zur Veranschaulichung betrachten wir ein Beispiel-Szenario aus der Fahrzeugherstellung und Verkauf. Dabei geht es um den Informationsaustausch über Autos und Autoteile zwischen einem Automobilgroß-

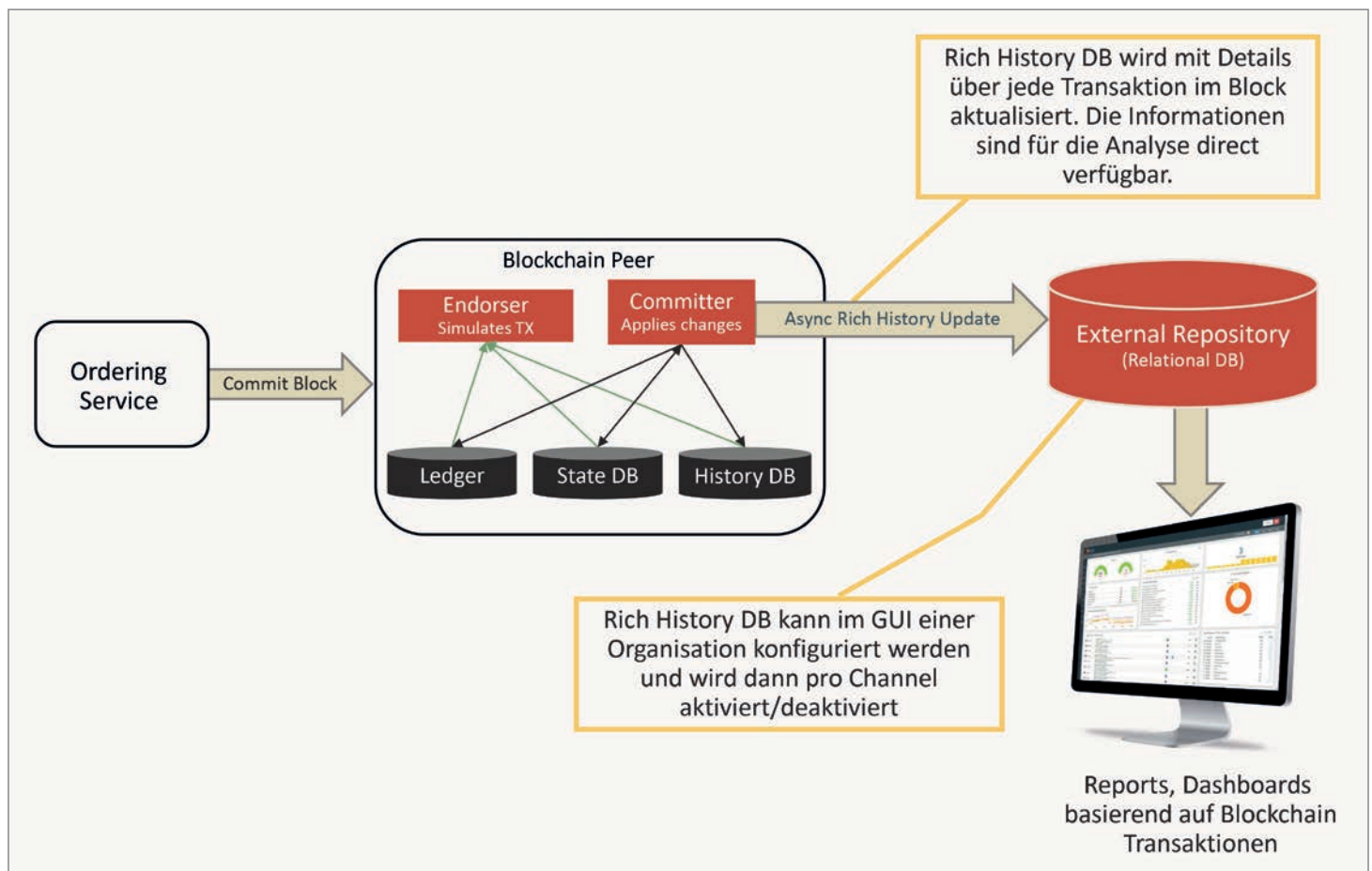


Abbildung 4: Rich-History-Database-Architektur (Quelle: Oracle)

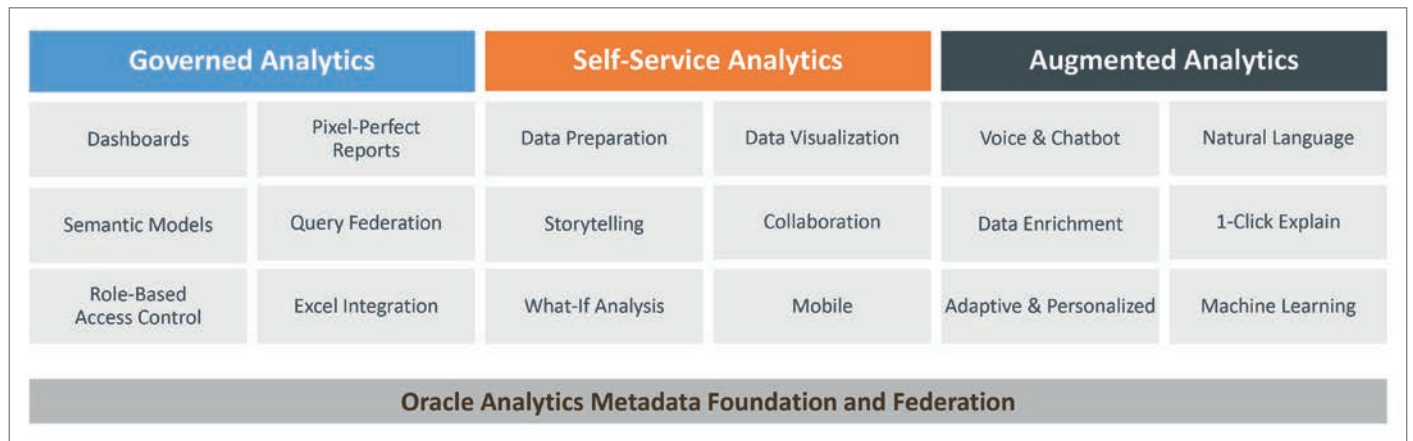


Abbildung 5: Oracle Analytics Cloud-Plattform – Überblick (Quelle: Oracle)

händler, Autoherstellern und Autoteile-Lieferanten. Die Informationen über Transaktionen zwischen den jeweiligen Partnern (wie zum Beispiel Bestellung oder Reklamation der Autoteile) werden in der Oracle-Blockchain festgehalten. Der Autohändler möchte visuell darstellen und auswerten können, wie die Geschäfte mit einzelnen Partnern laufen.

Die Daten der Blockchain-Transaktionen werden in der Rich-History-Datenbank – in diesem Fall einer Oracle-Autonomous-Data-Warehouse-Datenbank – in Tabellen abgelegt, deren Schema speziell für die Aufnahme der Blockchain-Daten vorgesehen ist. Die zentrale History-Tabelle, die wir in unserem Beispiel für die Analyse verwenden, heißt „Org1Founder_carchannel_hist“. In dieser Ta-

```

SELECT
  "BC"."CHAINCODEID"      "CHAINCODEID",
  "BC"."KEY"              "KEY",
  "BC"."TXNISVALID"      "TXNISVALID",
  "BC"."VALUE"           "VALUE",
  "BC"."VALUEJSON"       "VALUEJSON",
  "BC".VALUEJSON.serialNumber  serialNumber,
  "BC".VALUEJSON.assembler    assembler,
  "BC".VALUEJSON.assemblyDate assemblyDate,
  "BC".VALUEJSON.name        name,
  "BC".VALUEJSON.owner       owner,
  "BC".VALUEJSON.recall      recall,
  "BC".VALUEJSON.recallDate  recallDate,
  "BC"."BLOCKNO"           "BLOCKNO",
  "BC"."TXNNO"             "TXNNO",
  "BC"."TXNID"            "TXNID",
  "BC"."TXNTIMESTAMP"     "TXNTIMESTAMP",
  "BC"."TXNISDELETE"     "TXNISDELETE"
FROM
  "BC"." Org1Founder_carchannel_hist " "BC";
    
```

Listing 1: SQL zur JSON-Extraktion in einzelne Spalten der JSON Properties

CHAINCODEID	KEY	TXNISVALID	VALUEJSON	DOCTYPE	SERIALNUMBER	ASSEMBLER	ASSEMBLYDATE	NAME	TXNID
carTrace	abe1234	1	{\"docType\":\"vehiclePart\", \"serialN...\"}	vehiclePart	abe1234	panama-parts	1502688979	airbag 2020	8148dcdcf9ac4891f8c429b0cfd0
carTrace	abe1235	1	{\"docType\":\"vehiclePart\", \"serialN...\"}	vehiclePart	abe1235	panama-parts	1502688979	airbag 4050	3fcdf28cef0e355cbfb660723722a
carTrace	abf1234	1	{\"docType\":\"vehiclePart\", \"serialN...\"}	vehiclePart	abf1234	panama-parts	1502688979	airbag 2021	059f520c4b3820d27b6f59165d8
carTrace	abf1235	1	{\"docType\":\"vehiclePart\", \"serialN...\"}	vehiclePart	abf1235	panama-parts	1502688979	airbag 4051	90c65c6eef77041584a5da59933
carTrace	abf2022	1	{\"docType\":\"vehiclePart\", \"serialN...\"}	vehiclePart	abf2022	panama-parts	1502688979	airbag 2022	6708ca675c2444695083cd721f1
carTrace	abf4052	1	{\"docType\":\"vehiclePart\", \"serialN...\"}	vehiclePart	abf4052	panama-parts	1502688979	airbag 4052	28a0006396598efdb0ae67fd11d

Abbildung 6: Inhalt der Blockchain-Rich-History-Tabelle (Quelle: Oracle)

belle sind alle Informationen des Blockchain-Channels „carchannel“ und seine Transaktionen als JSON vorhanden und können nun für die Analyse verwendet werden.

Dafür lösen wir die Daten in der JSON-Spalte (VALUEJSON) auf und stellen sie für weitere Abfragen als Spalten zur Verfügung. Das dazu notwendige SQL ist im Listing 1 abgebildet.

Die extrahierten Properties sind die definierten Attribute, die für diesen Channel angegeben sind und bei den Transaktionen verwendet werden.

Diese Daten können nun als Übersicht, wie in Abbildung 6 dargestellt, visualisiert werden.

Natürlich lassen sich die Daten auch auf Mengen, Bestellhistorien, Zeitreihen

und Recalls hin abfragen und visualisieren. Dies verdeutlicht Abbildung 7 exemplarisch.

Da die Transaktionen immer zwischen mindestens zwei Partnern stattfinden, kann dieses Beziehungsgeflecht, das daraus entsteht, auch in einer Grafik veranschaulicht werden (siehe Abbildung 8).

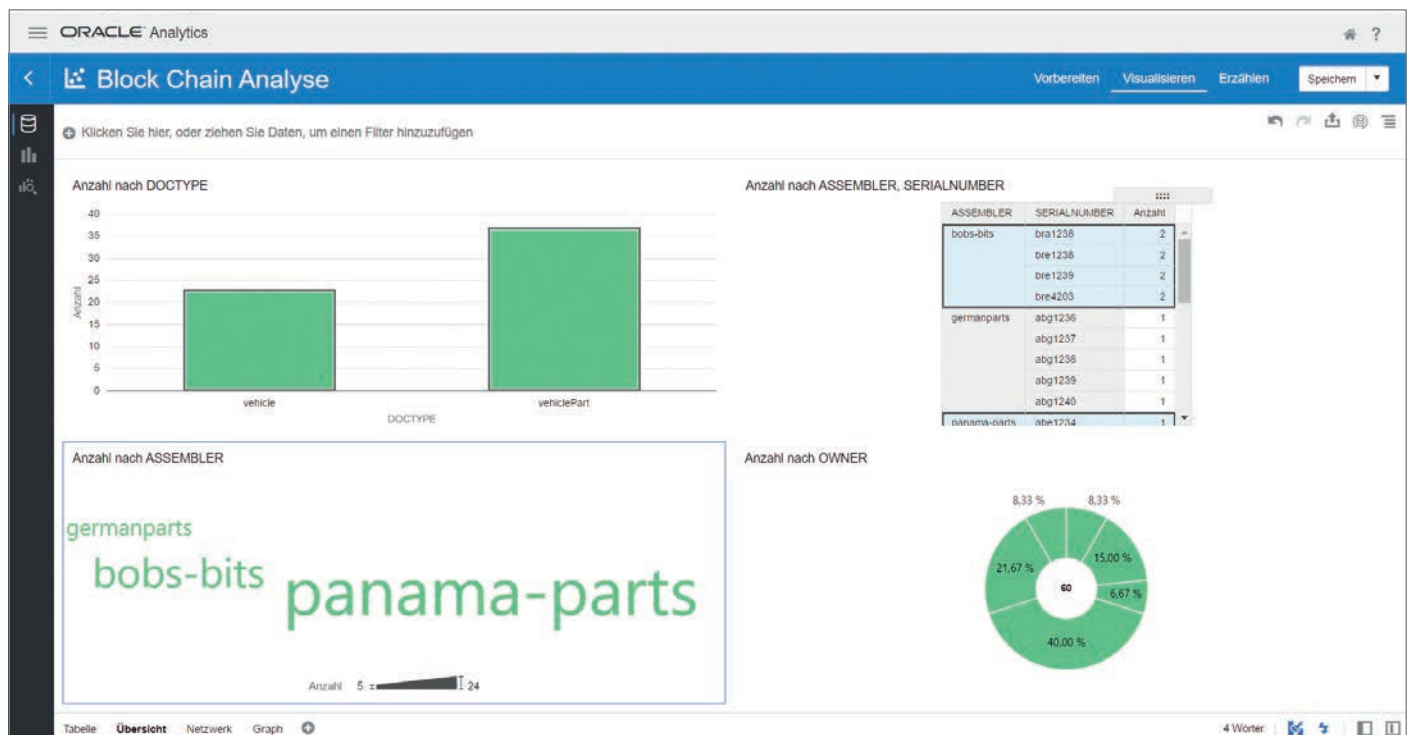


Abbildung 7: Analyse-Dashboard der bestellten Teile (Quelle: Oracle)

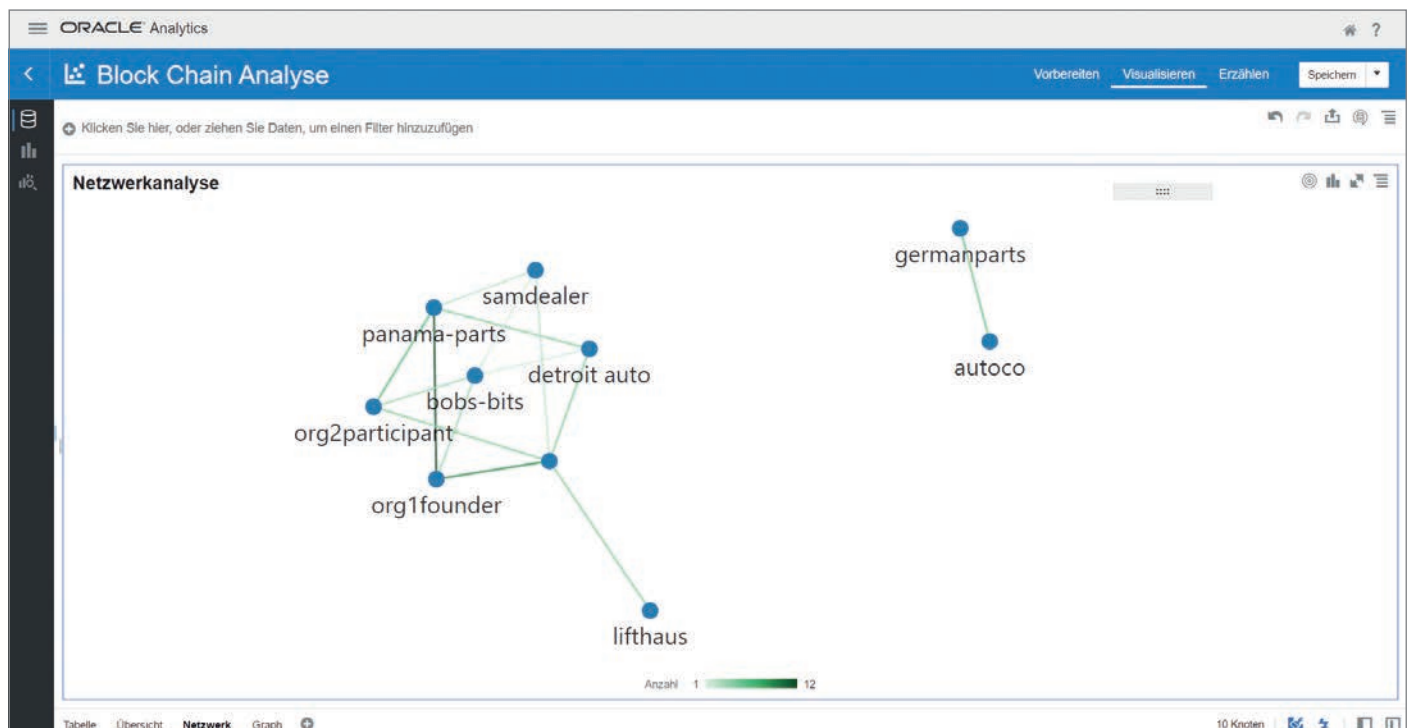


Abbildung 8: Netzwerkanalyse der beteiligten Partner (Quelle: Oracle)



Abbildung 9: Graph-Analyse der beteiligten Partner (Quelle: Oracle)

Damit können bei den verschiedenen Anwendungsfällen von Blockchain die Beziehungsmuster (zum Beispiel verwendete Bauteile eines Werkstücks, Transaktionspartner in einem finanziellen Netzwerk oder aber auch zentrale Figuren in einem verdächtigen Finanz-Transaktionsnetzwerk) herausgefunden und anhand einer einfachen Graph-Darstellung der zugrunde liegenden Daten der Blockchain History DB analysiert werden (siehe Abbildung 9).

Mithilfe von Graph SQL in der Datenbank könnten weitere Aspekte des Netzwerkes abgefragt und dann auch in Oracle Data Visualization analysiert werden.

Mit einfachen Mitteln und ohne technisches Verständnis für die Blockchain oder die Datenbankabfragen kann so jeder Anwender seine Sicht auf diese Daten gewinnen und diese spielend mit Bestandsdaten oder weiteren Informationen aus anderen Systemen abgleichen und die Analysen verdichten.

Die Transaktionen, die in der Blockchain gespeichert werden, lassen sich so sinnvoll aufbereiten und gewinnbringend analysieren.

Erkenntnisse

Die in der Blockchain abgespeicherten Transaktionsdaten lassen sich mit geeigneten Werkzeugen recht einfach visuell darstellen und analysieren. So können Fachanwender ohne technische Vorkenntnisse über Blockchain- und Datendank-Technologien die für

ihr Fachgebiet wichtigen Erkenntnisse aus den Daten ziehen und geschäftsrelevante Entscheidungen treffen.

Für diese Blockchain-basierten Anwendungen bietet Oracle eine Open-Source-basierte Plattform, mit der sowohl die Entwickler und der Betrieb als auch die Daten-Analysten geeignete Werkzeuge erhalten, um effizient im jeweiligen Aufgabenfeld arbeiten zu können.

Die einzelnen Komponenten lassen sich einfach konfigurieren und stellen damit eine ideale Umgebung dar, die eine Analyse der Blockchain-Daten ermöglicht und gleichzeitig über die autonome Datenbank in der Cloud eine administrationsarme Umgebung mit vielen weiteren Möglichkeiten, beispielsweise Graph SQL und Machine Learning, bereitstellt. Damit rücken Blockchain-Anwendungen weiter aus der Nische ins Licht der Daten-Analysten.

Quellen

- [1] Oracle-Blockchain-Plattform:
<https://www.oracle.com/de/blockchain/>
<https://developer.oracle.com/de/blockchain/>
<https://blogs.oracle.com/blockchain/>
- [2] Hyperledger-Projekt der Linux Foundation:
<https://www.hyperledger.org/>
- [3] Oracle Analytics Cloud:
<https://www.oracle.com/business-analytics/analytics-cloud.html>
<https://www.oracle.com/a/ocom/docs/solutions/business-analytics/oracle-analytics-cloud-ds.pdf>

<https://docs.oracle.com/en/cloud/paas/analytics-cloud/>



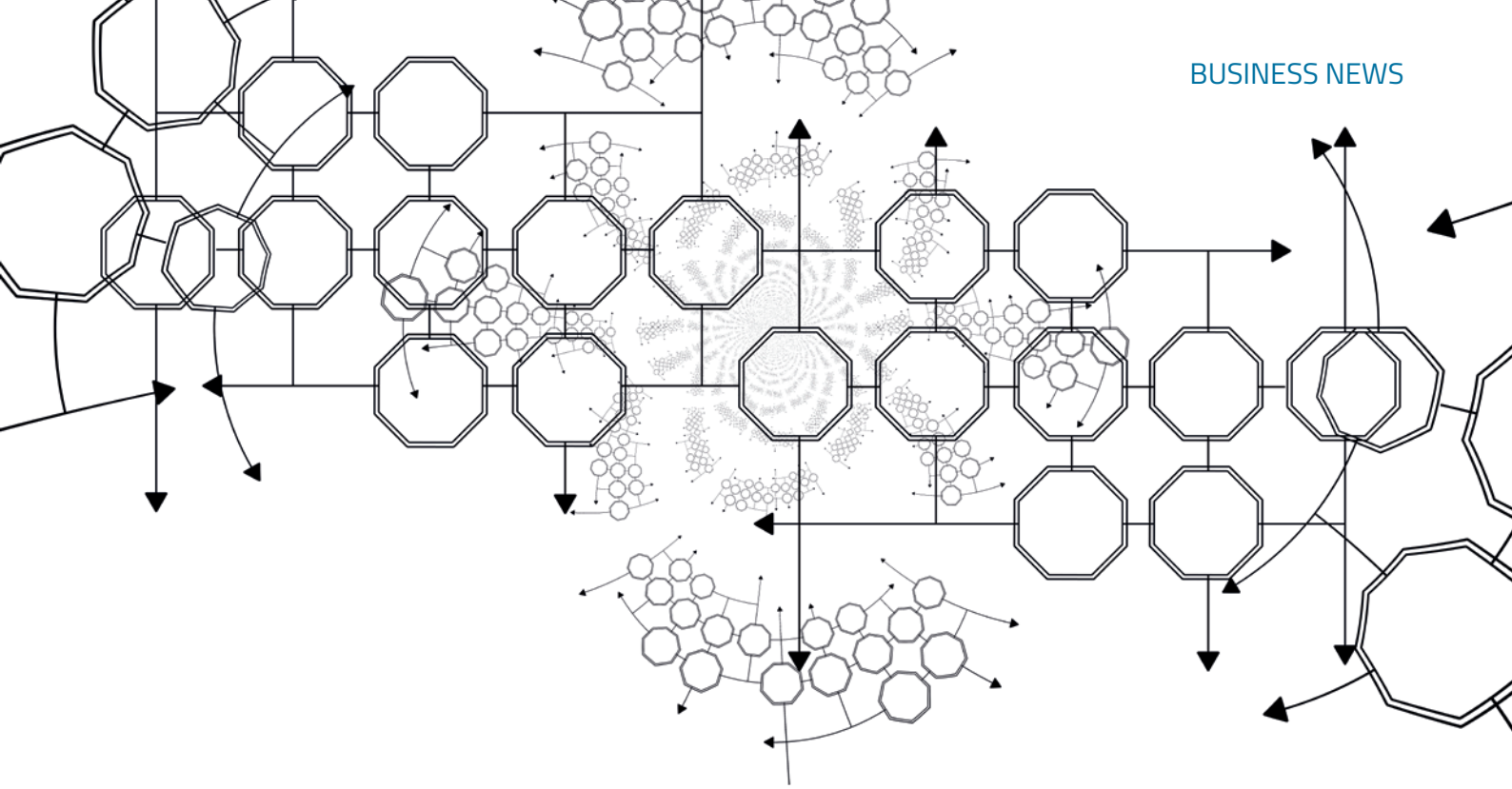
Evgenia Rosa
evgenia.rosa@oracle.com

Evgenia Rosa arbeitet als Principal Solution Engineer bei Oracle Deutschland B.V. & Co. KG. In dieser Rolle unterstützt sie Oracle-Kunden bei der Entwicklung von Cloud-basierten Lösungen mit den Schwerpunkten Integration, intelligente Prozessautomatisierung und Blockchain.



Detlef E. Schröder
detlef.e.schroeder@oracle.com

Detlef E. Schröder arbeitet als Master Principal Solution Engineer bei Oracle Deutschland B.V. & Co KG. In dieser Rolle unterstützt er Kunden bei analytischen Fragestellungen und KI-Aufgabestellungen sowie DWH-Architekturen.



„Die Blockchain zeigt ihre Schokoladen- seite in der Logistik und Markierung von Waren und darin, die Integrität der Daten zu gewährleisten.“

Interview mit Ekaterina Koschkarova, Boston SoftDesign

Die Zeit der Skeptiker ist vorbei, und die Blockchain-Technologie ist in vielen Branchen zum Standard geworden. Aber viele Fragen, Gerüchte und Spekulationen bleiben. Wir sprachen über die populäre Behauptung „Da gibt es keine Analytics in der Blockchain!“ mit Ekaterina Koschkarova, technische Expertin von Boston SoftDesign (Europa- und USA-orientiert) und DBI (in Russland und den ehemaligen GUS-Staaten tätig).

Frau Koschkarova, Datenanalyse und Blockchain: Was macht dieses „Pärchen“ so interessant?

Die Blockchain hat die Welt verändert. Ob grundlegend oder nicht, ist noch nicht klar. Ist es nur „hype“ oder werden die Blockchain-Technologien eine Gewohnheit für uns sein, so wie Big Data? Wie funktionieren Unternehmen, die die Blockchain verwenden? Was passiert in dieser seltsamen Welt der Blöcke? Ist es wirklich effektiv? Um etwas zu verwalten, um etwas zu verstehen, müssen wir es zuerst messen, analysieren.

Daher ist die Idee der Analyse in der Blockchain eine sehr relevante Frage. *Smart Contracts, Assets, Tokens...* Das alles ist ein Teil eines Ökosystems. Erfahren, wie es funktioniert, ist wahnsinnig interessant.

Wie sehen die technischen Anforderungen dafür aus?

Blockchain-Netzwerke können öffentlich und nichtöffentlich sein. Wenn wir über offene Netzwerke sprechen, dann können wir Informationen über Blöcke und Transaktionen

erhalten, zum Beispiel mit APIs. Außerdem kann man die gesamte Blockkette für das Mining oder die Transaktionsauthentifizierung herunterladen. Sieht gut aus, nicht wahr? Leider ist das alles nicht so einfach. Blockchain ist eine Folge von Blöcken. Die Blöcke speichern Transaktionsdaten und Metadaten für die Integrität. Um etwas in der Blockchain zu finden, muss man die Blocknummer oder den Hash der Transaktion oder die Adresse kennen. Die Blockchain ist anonym, weil sie für Anonymität und Privatsphäre geschaffen wurde.

Mit Kryptowährungen ist es anders. Es gibt Kryptowährungsbörsen, zum Beispiel Binance [1], mit den notwendigen Standard-Tools: Der Benutzer kann Wechselkurse, DOM („Depth of Market“) und andere Parameter untersuchen.

Was können wir bekommen?

Wenn wir über Kryptowährungen sprechen, sind die beliebtesten Use Cases die Kapitalisierung der Währungen des Netzwerks: Wie wird die Währung gegen andere Währungen und Kryptowährungen gehandelt? Welche Trends, welche Prognosen gibt es?

Wenn es jedoch um die Anwendung von Blockchain-Netzwerken in anderen Bereichen wie der Logistik geht, dann gibt es Use Cases wie Track&Trace, das bedeutet Tracking aller Dokumente und Metadaten in der Lieferkette (wann wurde die Fracht von wem wohin geliefert). Diese Produkte lassen sich problemlos auch mit IoT integrieren, sodass man nicht nur die Metadaten der Sendungen verwalten kann, sondern auch die Integrität der Verpackung überwacht. Da alle Mitglieder des Netzwerks den Zugriff auf die Blockkette haben, ist es unmöglich, die Daten durch Entfernen oder Hinzufügen von Blöcken zu verfälschen. Aber die Statistiken in verschiedenen Ansichten zu sehen und zu verstehen, wie sich die Frachten bewegen, das ist einfach! Gute Ergebnisse zeigt Oracle Track&Trace, basierend auf der Blockchain. Eine ähnliche Idee kann den Banken angeboten werden, um internationale Transaktionen zu implementieren.

Worin unterscheidet sich die Datenanalyse mithilfe von Blockchain von der „klassischen“ Datenanalyse? Können Sie uns ein Beispiel dafür geben?

Was ist die „klassische“ Datenanalyse? Es ist der Erhalt und die Konsolidierung von Daten aus Datenquellen, aus Datenbanken, APIs; die Reinigung der Daten und ihre Anreicherung. Dann folgt die Analyse: die Formierung der Graphen, die Suche der Regelmäßigkeit, die Prognostizierung, die Visualisierung. Die gleichen Schritte werden von den Blockchain-Daten erwartet. Das Schwierigste, wie ich bereits gesagt habe, ist, die notwendigen und nützlichen Daten zu erhalten. Eine große Rolle in der Blockchain-Analyse spielt prädikative Analyse: Sie hilft, das Verhalten des gesamten Netzwerks vorherzusagen. Die Daten aus der Blockchain werden normalerweise geparkt und in der Datenbank gespeichert, wo sie mit Standardmethoden analysiert werden können. Es ist auch eine gute Idee, Blockchain-Daten mit den Daten der Web-Analyse (zum Beispiel die Statistik der Client-Nutzer) anzureichern.

Aber der größte Unterschied zur klassischen Analyse ist die Anonymität von Transaktionen. Und natürlich eine Reihe von analysierten Parametern.

Welche grundlegenden Metriken können analysiert werden?

Die Metriken sind netzwerkabhängig. Schauen wir zum Beispiel auf *Ethereum*, eine Plattform für dezentralisierte Dienste, und seine „forks“. In jedem Netzwerk wird in erster Linie die Anzahl der Netzwerkadressen analysiert. Wichtig ist, daran zu erinnern: Adresse und Benutzer sind nicht dasselbe. Theoretisch kann ein Benutzer mehrere Adressen haben. Die Dynamik der Erstellung neuer Adressen und deren Verhalten werden analysiert. Ein guter Indikator sind die ausgegebenen *Token*: In Verbindung mit der Analyse von Adressen kann man hochwertige Daten von der Token-Ausgabe mit bestimmten Adressen erhalten. Dieser Ansatz wird aktuell für *Smart Contracts* verwendet. Alle diese erhaltenen Daten werden von den Analytikern aggregiert, Grafiken werden erstellt, Durchschnittswerte werden ausgegeben. Wie ich schon gesagt habe, hilft immer die Anreicherung der Blockchain-Daten mit Web-Analyse-Reports: In welchem Land, zu welcher Zeit werden die meisten *Token* ausgegeben, wie verhält sich der Benutzer. Alle diese Daten helfen, ein vollständiges Bild der Interaktion zu schaffen. Es ist auch sehr wichtig, die Kapitalisierung des Netzwerks zu überwachen. Man muss jede Stunde die Daten über die Börsen erhalten. Aber hier hilft schon die „klassische“ Analytik.

Sehen Sie auch Gefahren im Umgang mit Blockchain?

Die Hauptgefahr der Blockchain ist gleichzeitig ihr Hauptvorteil: die Anonymität von Transaktionen. Wer steckt hinter der Adresse? Es gibt kaum eine Chance, es herauszufinden. Daher kann die Blockchain leicht zu einem Geldwäsche-Tool werden, auch weil viele Länder und Finanzinstitute die Verwendung von Kryptowährungen noch nicht regulieren. Es gibt allerdings eine noch größere Gefahr: die Finanzierung des Terrorismus. Transaktionen, die zu illegalen Waffenkäufen führen, können nicht verfolgt und gestoppt werden. Das ist eine neue Herausforderung für die finanzielle Compliance.

Wer nutzt hauptsächlich Blockchain?

Theoretisch kann Blockchain fast überall angewendet werden, weil es die Architektur der Datenübertragung ist. Die führenden

Verbraucher dieser Technologie sind Finanzinstitute, Banken: Sie sind an neuen Funktionen und Technologien interessiert, zum Beispiel an Kryptowährungen und Kryptobörsen. Die Datenübertragung macht es möglich, die Dauer der Transaktionen von Tagen bis auf Stunden zu reduzieren. Die Blockchain zeigt ihre Schokoladenseite in der Logistik und Markierung von Waren und darin, die Integrität der Daten zu gewährleisten. Wie ich bereits erwähnte habe, sehen alle Teilnehmer des Netzwerks eine Kopie der Blockkette. Mit anderen Worten, die Blockchain wird überall benötigt, wo Daten schnell übertragen werden müssen und dabei nicht manipuliert werden können.

Blockchain ist noch eine recht junge Technologie, die im Jahr 2009 bei der ersten Implementierung der Bitcoin-Software offiziell veröffentlicht wurde. Seitdem wächst die Anhängerschaft von „Blockchain als verteiltes Datenbankmanagementsystem mittels kryptographischer Verfahren“ [2] kontinuierlich. Wohin geht die Reise? Und wird man in Zukunft eher Kryptographie-Experten als DBAs benötigen?

Schwer zu sagen, aber ich denke: ja. DBAs und andere Experten, die mit Datenbanken und Datenstrukturen arbeiten, müssen verstehen, wie die Daten gespeichert und verwaltet werden. Deswegen glaube ich, dass wir bald Jobs wie „CRYPTO-DBA“ sehen. So dringt die Kryptographie in unser Leben ein und nicht nur mit den IT-Sicherheitsprodukten.

Frau Koschkarova, herzlichen Dank für das Gespräch.

Quellen

- [1] <https://www.binance.com>
- [2] <https://de.wikipedia.org/wiki/Blockchain>



Ekaterina Koschkarova
ekoshkarova@bostonds.ru

Ekaterina ist technische Expertin von Boston SoftDesign (Europa- und USA-orientiert) und DBI (in Russland und ehemaligen GUS-Staaten tätig) mit der Expertise in den klassischen IT-Bereichen Datenbanken, Softwareentwicklung und IT-Sicherheit ebenso wie im Bereich Blockchain.

Public Blockchain Innovation – wie die Evolution der Protokolle Branchen revolutioniert

Jonas Huck, esentri AG

Sogenannte Open Public Blockchains wie Ethereum bilden die Grundlage für einen rasant wachsenden Innovationsprozess. Sie sind im Begriff, den Finanzsektor zu revolutionieren. Hier bilden sich immer neue Protokolle, die mittels Smart Contracts als funktionale Bausteine komplexe Finanzdienstleistungen und Funktionalität abbilden und diese in die Hände von Endbenutzern bringen. Durch ihre verteilte Natur nicht zu stoppen und ohne zentrale Entität, gehen sie an sämtlichen Regularien vorbei und entwickeln dadurch eine innovative Geschwindigkeit, mit der herkömmliche Dienstleistungen traditioneller Firmen nicht mehr mithalten können. Wie kam es dazu und wo geht die Reise hin?

Warum ist Blockchain nicht gleich Blockchain?

Der Begriff Blockchain wird oft sehr dehnbar verwendet. Das führt an vielen Stellen zu fundamentalen Missverständnissen. Anfangs wurde der Begriff Blockchain als das Bitcoin unterliegende technologische Konzept gesehen. Jedoch wurde der Begriff Blockchain bald auch separat von Bitcoin verwendet. Oft werden dabei die spieltheoretischen sowie ökonomischen Aspekte übersehen, die fundamentaler Bestandteil einer Blockchain sind. Ohne die unterliegende Währung wie beispielsweise Bitcoin fällt das komplizierte Konstrukt aus Technologie, Spieltheorie und Ökonomie in sich zusammen und die Blockchain verliert ihre revolutionären Eigenschaften von barrierefreier, autarker Funktionalität ohne eine verwaltende Entität.

Im Folgenden wird der Bereich der Public Blockchains betrachtet. „Public“ bedeutet in diesem Fall: ohne verwaltende und regulierende Einheit, konträr zu Enterprise Blockchains, die den Aspekt der zentralen Kontrolle beibehalten. Damit verbunden ist auch eine juristische Bindung an eine Person oder Firma. Bei Open Public Blockchains verwaltet sich das Netzwerk als Teil des Protokolls selbst und gibt ökonomische Anreize in Form der jeweiligen Kryptowährung. Dies soll die Teilnehmer des Netzwerks dazu bewegen, im Sinne des Protokolls zu handeln. Es handelt sich dabei um ein hochkomplexes technisches sowie soziales Konstrukt, das im Falle von Bitcoin seit

über zehn Jahren autark und hochverfügbar funktioniert und expandiert.

Evolutionsstufe 1

Inspiziert durch Bitcoin, jedoch dessen Einschränkungen und Grenzen erkennend, wurde 2015 die Public Blockchain *Ethereum* ins Leben gerufen. Bitcoin nutzt aufgrund sicherheitsrelevanter Aspekte eine sehr limitierte Skriptsprache und ist deshalb nicht zur Realisierung komplexer Programme geeignet. *Ethereum* hingegen bietet eine Turing-vollständige Programmiersprache. Turing-Vollständigkeit ist ein Konzept aus der Informatik und bedeutet, dass sich sämtliche definierbare Logik auf Basis der Programmiersprache abbilden lässt. Das ist keinesfalls ein Merkmal, das *Ethereum* zu einer besseren Blockchain als Bitcoin macht. Es ist eine andere Ausrichtung. Im Bereich der Blockchains besteht ein konstantes Gewichten von Aspekten wie Sicherheit, Geschwindigkeit, Dezentralisierung sowie Flexibilität. Auf *Ethereum* oder anderen Turing-vollständigen Blockchains geschriebene Programme werden als funktionale Bausteine auch *Smart Contracts* genannt. Der Vertrag besteht in der absoluten Bindung an die Funktionalität des Source-Codes. Transaktionen lassen sich nicht rückgängig machen. Sowohl technisch als auch rechtlich gibt es keine Möglichkeit dazu. Dadurch entsteht eine hohe Verantwortung für fehlerfrei und sicher geschriebenen Code. Die erste populäre Anwendung fanden *Smart Contracts* auf *Ethereum* in sogenannten *Initial Coin Offerings*

(ICOs). Diese stellen entweder handelbare *Tokens* dar, vergleichbar einer Währung, oder auch eindeutige digitale *Assets*, sogenannte *Non-Fungible Tokens*. 2017 erreichte der ICO-Hype seine Hochphase und unzählige, zum größten Teil wertlose *Tokens* wurden kreiert, die als spekulatives Konstrukt die Emotionalität unerfahrener Spekulanten ausnutzte. In dieser Phase entstand nicht viel Substanz in Form sinnvoller Anwendungen, die dem Nutzer über den Use Case als reines spekulatives Instrument hinaus einen Mehrwert bot. Dennoch zeigte es die Möglichkeiten von *Smart Contracts* auf. Diese sind zumindest in der Theorie vielfältig. Als digitales, unveränderbares und handelbares Pendant verknüpfen *Non-Fungible Tokens* klassische *Assets*, wie Wertpapiere oder auch Immobilien, oder sie stellen in ihrer einfachsten Form rein digitale, handelbare *Assets* dar. Daraus ergeben sich zahlreiche Möglichkeiten für innovative Anwendungen sowie eine Automatisierung und Beschleunigung vieler Prozesse, die heute immer noch von manuellen und trägen Arbeitsschritten dominiert werden. Momentan scheitert die Umsetzung meist noch an regulatorischen Einschränkungen, die ein schnelles Fortschreiten hier behindern.

DeFi – Finanzprodukte werden dezentralisiert

In den letzten Monaten ist das Wort DeFi immer präsenter geworden. DeFi steht für *Decentralized Finance*. Das Protokoll Bitcoin gibt dem Anwender alle Funktionalität in

die Hand sowie jegliche Verantwortung im richtigen Umgang mit dem Protokoll. Es gibt keine zentrale Entität, die Fehler rückgängig machen kann, wie man es im Falle einer Bank zumindest teilweise gewohnt ist. Dadurch ist in Bitcoin-Kreisen die Redewendung „be your own bank“ entstanden. DeFi bringt dieses Konzept auf die nächste Stufe. Bisher haben die meisten Nutzer von Kryptowährungen wie Bitcoin und *Ethereum* ihre *Coins* und *Tokens* entweder ausgegeben oder in den meisten Fällen als spekulatives Instrument gehalten und nichts damit unternommen. DeFi-Protokolle ermöglichen nun Endbenutzern, das digitale Geld für sich arbeiten zu lassen, anstatt nur auf den Wert des jeweiligen *Tokens* zu spekulieren. So werden nach und nach neue Protokolle angeboten, die klassische Finanzinstrumente nachbilden und bisher nur Banken vorbehalten waren. In den meisten Fällen sind das bislang der Verleih von *Tokens* und die damit verbundene Einnahme von Zinsen sowie das Bereitstellen von Liquidität bestimmter *Tokens* und die Beteiligung an Transaktionsgebühren.

Ähnlich dem ICO-Hype 2017 sind vor allem Spekulanten bereit, große Risiken einzugehen, und der in *DeFi Smart Contracts* gehaltene Wert wächst rasant im mehrstelligen Millionenbereich. Das Risiko für den Endbenutzer liegt bei den meisten Protokollen in der Validität und Fehlerfreiheit der *Smart Contracts*, die jedoch frei einsehbar und überprüfbar sind. Allerdings steigt deren Komplexität rasant, auch durch das Einbeziehen externer Datenquellen als Teil der *Contracts*, sogenannter *Oracles*, sowie der Verknüpfung mehrerer *Contracts* zu komplexen technologischen sowie funktionalen Konstrukten. Im Gegenzug bieten viele DeFi-Protokolle sehr hohe Renditen, keine regulatorische Einstiegshürde sowie das Wegfallen von Mittelsmännern und den damit verbundenen Gebühren.

Ersetzen der Mittelsmänner durch Protokolle

Ebenso wie mit ICOs zeigen DeFi-Protokolle erneut, was über *Smart Contracts* möglich ist, und es ist nur eine Frage der Zeit, bis auch andere Branchen über dezentrale Protokolle abgebildet werden. Wichtig dabei ist zu verstehen, dass die herkömmliche Denkweise in Firmenkonstrukten mit festgelegten Regulierungen und Auflagen nicht mehr funktioniert. Neue Produkte und Dienstleistungen in Form von dezentralen Protokollen werden den Markt durch-

dringen. Traditionelle Firmen werden sich zukünftig mit der innovativen Kraft und Geschwindigkeit dieser Protokolle messen müssen. Zusätzlich erschwerend wirken Ertragsmodelle sowie Auflagen auf Firmen ein, um rentabel sowie rechtskonform zu bleiben. Es ist davon auszugehen, dass Protokolle in sämtliche Branchen dringen werden, in denen automatisierbare Produkte und Dienstleistungen angeboten werden und Gelder in Form von Gebühren und Transaktionskosten an Mittelsmänner fließen. Blockchain-Protokolle beteiligen Endbenutzer an diesen Gebühren als Teil des Netzwerks, anstatt sie in Firmenkonstrukten zu binden. Darüber hinaus entfällt die Einstiegshürde von Regularien und Auflagen auch für den Anwender.

Nutzen der Blockchain im Energiesektor

Auch in anderen Bereichen beginnen Firmen sowie Individuen, auf kreative Art und Weise Blockchain-Technologien zu ihrem Vorteil zu nutzen. Bitcoin wird in populären Kreisen oftmals für seine Energieverschwendung angeprangert. Über das Wort Verschwendung lässt sich hier jedoch streiten, da die Aufwendung der Energie ein fundamentaler Bestandteil des Protokolls ist, um die Sicherheit des Netzwerks sicherzustellen. Demnach wird die Energie nicht verschwendet, sondern transformiert in die Sicherheit des Bitcoin-Netzwerks. Genau diesen Sachverhalt machen sich erste Energieerzeuger zunutze. Wie allgemein bekannt ist, stellt gerade das Speichern von Energie eine große Herausforderung dar. So ist es in vielen Fällen für Energieerzeuger ökonomisch sinnvoller, Energie zu verbrennen als beispielsweise die Maschinen zur Energiegewinnung bei Überschuss herunterzufahren. Diese überschüssige Energie wird nun von einigen Energieerzeugern zum sogenannten Minen von Bitcoin verwendet. Minen ist der Prozess der Validierung von Bitcoin-Transaktionen und des damit verbundenen Erbringens von Arbeit durch mathematische Berechnungen, die Energie benötigen. Als Entlohnung für die erbrachte Arbeit und das damit verbundene Betreiben des Netzwerks werden Miner mit Bitcoin entlohnt. So kann überschüssige Energie wiederum in Bitcoin transformiert werden, anstatt sie verbrennen zu müssen. Als positiver Nebeneffekt kommt das auch dem Bitcoin-Netzwerk zugute, das durch neue Miner weiter dezentralisiert und dadurch sicherer wird.

Wohin geht die Reise?

Es ist schwer vorhersehbar, wie sich offene Protokolle entwickeln werden, welche Branchen als Nächstes von dezentralen Protokollen durchdrungen werden und wo der Platz von zentralisierten Firmen in Zukunft sein wird. Klar ist, auch dezentrale Protokolle machen nicht alles besser. In vielen Bereichen stellen eine zentrale Entität als Ansprechpartner, ein menschliches Gegenüber sowie die menschliche Kreativität, Auffassungsgabe und die Imperfektion ein Alleinstellungsmerkmal dar, das nicht so einfach ersetzt werden kann. Jedoch gibt es in genug Branchen Mittelsmänner, die für das Produkt selbst oder die Dienstleistung keinen Mehrwert darstellen. Viele Leistungen von Banken, Versicherungen oder auch Plattformen wie Streamingdiensten, Lieferdiensten oder zur Vermittlung von Jobs lassen sich über Protokolle realisieren und verhindern somit das Abschöpfen der Erlöse durch Dritte. Genau dort werden Protokolle weiter vordringen und den Markt nachhaltig verändern. Daher müssen sich Firmen mit Weitblick orientieren und auf die Stärken zentralisierter Strukturen setzen, wie ein höherer Transaktionsdurchsatz, leichtere Änderbarkeit von IT-Infrastrukturen, eine einfachere Skalierbarkeit, schnelle Release-Zyklen, Kontrolle der Datenhoheit sowie Serviceleistungen. Ebenso können auch traditionelle Firmen Vorteile neuer Protokolle nutzen, um die Schwächen zentralisierter Strukturen zu reduzieren.

Quellen

The Original Let's Talk Bitcoin Show
(<https://soundcloud.com/mindtomatter>)
<https://soundcloud.com/mindtomatter/ltb-345-bitcoin-miners-us-energy-producers-and-moores-law>



Jonas Huck
jonas.huck@esentri.com

Jonas Huck ist seit 2017 Consultant bei der esentri AG und beschäftigt sich in seiner Freizeit mit Open-Public-Blockchain-Technologien.



Blockchain Gaming – Vom Spielplatz zur Revolution

Dennis Walz, Code Alliance Berlin

Videospiele sind mit einem jährlichen Umsatz von über 140 Milliarden US-Dollar ein relevanter Wirtschaftssektor. Rund fünf Milliarden davon sind allein im deutschen Markt entstanden (Stand 2018, mit einem jährlichen Wachstum von etwa 9% [1]). Neben der rein ökonomischen Relevanz kommt der Videospielebranche obendrein eine innovationstreibende Rolle zu: Neue Technologien werden im Rahmen von Spielen erprobt, weiterentwickelt und einer breiten Öffentlichkeit zugänglich gemacht. Beispiele hierfür sind die rapide Verbesserung von Prozessoren, aber auch schillernde neue Technologien, die geeignet sind, unseren Alltag wieder einmal nachhaltig zu verändern, beispielsweise durch Virtual und Augmented Reality. In diesem Beitrag wollen wir uns daher einmal anschauen, wo und wie Blockchain-Technologie in Videospielen bisher genutzt wird, was wir daraus lernen können und welche Entwicklungen zu erwarten sind.

Gaming als Selbstzweck der Blockchain

Zunächst betrachten wir den am häufigsten anzutreffenden Spiel-Typ mit Blockchain-Bezug: Spekulationsspiele und Glücksspiel. Spekulation ist ein inhärentes Thema der Blockchain, insbesondere von Kryptowährungen [2]. Durch *Smart Contracts* können Wetten digital gesetzt und erfüllt werden, ohne dass hierfür Vertrauen in die Teilnehmenden nötig ist.

Ein *Smart Contract* ist ein Programm, das innerhalb der Blockchain adressierbar ist. Dieses Programm dient im Kern der Steuerung von Transaktionen auf Grundlage festgesetz-

ter Konditionen. Da es ein Programm ist, können diese Konditionen recht komplex werden sowie andere, nicht-transaktionsbezogene Logiken umsetzen. Im Beispiel der Wette muss ein *Smart Contract* abbilden, welche Parteien die Wette abschließen, was passiert, wenn die Wette positiv oder negativ ausgeht, und, sofern das möglich ist, welche Logik im Fall eines Unentschiedens greift. „Smart“ wird der Vertrag, indem er die Konditionen selbst prüfen und die daran geknüpften Folgen veranlassen kann. Im genannten Beispiel stellt der „Contract“ also den digitalen Buchmacher dar, der auf das Wettergebnis reagiert und das Geld transferiert. Der Beispielcode eines

Smart Contract in *Listing 1* zeigt eine simple Version eines Wettvertrags, bei dem Spieler durch Nutzung der `bet()`-Funktion auf ein `$outcome` wetten. Wenn das Ergebnis feststeht, bekommen alle Spieler die korrekt gewettet haben, den doppelten Betrag durch die `payout()`-Funktion ausgezahlt, die nur durch den Eigentümer des Wett-*Smart Contract* ausgelöst werden kann.

Earnbet ist mit rund 2000 Nutzern am Tag das größte Online-Casino im Markt der dezentralen Apps. Es nutzt die WAX-Blockchain, um alle Spiele in Form von *Smart Contracts* durchzuführen und die Nutzer-Interaktion als Transaktionshistorie transparent zu machen.

Die kryptografischen Eigenschaften des Hash in der Blockchain werden als Zufallsgenerator genutzt, sodass die Blockchain den Ausgang des Glücksspiels bestimmt [3]. Durch den *Smart Contract* kann ein Spieler zum einen darauf vertrauen, dass der Gewinn ausbezahlt wird, und zum anderen ist die Fairness des Spiels durch die Blockchain nachprüfbar. Das Vertrauen beruht also auf dem Verfahren, nicht auf dem Wettpartner beziehungsweise Anbieter.

Die Rolle von Gambling-Anwendungen für die Blockchain ist gewissermaßen eine selbsterhaltende: Auch wenn es sonst nichts zu tun gibt, gewettet wird immer. So wundert es nicht, dass sich Gambling-Anwendungen schnell auf jeder neuen Blockchain (mit Kryptowährung) etablieren; Casino-Anwendungen sind also Teil des initialen Anreizes dafür, neue Blockchains zu beleben.

Digitale Güterindividualität und Knappheit

Bisher gab es für digitale Anwendungen ein grundsätzliches Problem: „Wie kann es eine Knappheit von virtuellen Gütern geben, wenn sich Digitalität gerade durch Reproduzierbarkeit auszeichnet?“ Der Wert einer Sammelkarte hängt beispielsweise davon ab, wie viele Kopien davon erstellt wurden. Der Wert eines Sport-Trikots, das von einem populären Sportler im Finalspiel eines großen Turniers getragen wurde, ist signifikant höher als eine frisch produzierte Version des gleichen Trikots. Mittels *Non Fungible Tokens* (NFT) werden diese Konzepte von Eigentümerschaft und Güterindividualität in digitale (Spiel-)Welten übertragen.

Ein *Token* ist eine kryptografisch abgesicherte Repräsentation eines Wertes, einer Entität oder Funktion innerhalb von Blockchains. Die bekannteste Anwendung für ein *Token* sind *Payment Tokens*: Sie repräsentieren die Kryptowährungen, wie Bitcoin, und werden daher gerne auch „Coin“ genannt. Im Gegensatz zu Währungs-Token ist ein *Non Fungible Token* nicht austauschbar (= non fungible), wodurch es geeignet ist, einen individuellen, unteilbaren, aber veräußerbaren Eigentumsanspruch im Netzwerk darzustellen. Während ich also beispielsweise 0.01 Bitcoin transferieren kann, kann ich beim NFT immer nur das gesamte *Token* übergeben. Im Ethereum-Protokoll wurde dieses *Non Fungible Token* mit ERC-721 definiert und durch ERC-1155, bewusst mit Blick auf Blockchain-Gaming, nochmals optimiert [4]. NFT finden sich aber auch in vielen anderen Protokollen, zum Beispiel *EOS*, *TRON* oder *Loom*.

Das erste Videospiel, was NFT zur Grundlage des Spielgeschehens hatte, ist *CryptoKitties* [5]. Der Spieler sammelt, „züchtet“, kauft und verkauft hier digitale Katzen. Der Wert einer Katze entsteht durch die Seltenheit ihrer Attribute. Welche Attribute eine Katze ausprägt, hängt wiederum von den „Genen“ der Eltern-Katzen ab, sodass bestimmte Attribute nur durch bestimmte Katzen gezüchtet werden können. *CryptoKitties* erzeugt, wie die meisten NFT-Collectible-Games, *Ethereum-Tokens*, wodurch bei jeder Blockchain-relevanten Aktion tatsächliche Kosten für den Spieler entstehen. Während *CryptoKitties* das erste Spiel dieser Art war, sind andere *Collectible Games*, nicht zuletzt wegen eines größeren Spielumfangs, mittlerweile deutlich beliebter: *Splint-*

terlands, *Axie Infinity*, *CryptoBewMaster* und *Upland* sind die derzeit meistgespielten dezentralen Apps mit *Crypto Collectibles* [6].

Zugegebenermaßen ist die „Besonderheit durch Seltenheit“ von virtuellen Objekten, ebenso wie Eigentum an virtuellen Gütern, kein komplett neues Konzept in Videospielen. Im Gegenteil ist das Sammeln von besonderen virtuellen Gegenständen ein gewichtiger Teil von nahezu allen aktuell beliebten Onlinespielen: *Fortnite*, *Counter Strike*, *Fall Guys*, *Hearthstone* – bei all diesen Spielen gibt es seltene Gegenstände, die zumeist durch sogenannte *Lootboxen* oder andere Zufallsereignisse erlangt werden können. Die essenzielle Veränderung besteht zum einen in der Dokumentation der Seltenheit: Ein Spieler sieht in der Blockchain, dass es das vermeintlich seltene Item tatsächlich nur in begrenzter Zahl gibt. Zum anderen kann der Spieler seine *Token* auch außerhalb der Spielwelt veräußern und behalten, selbst wenn die Spiel-Server abgeschaltet werden, das sorgt für tatsächliche Eigentümerschaft am virtuellen Objekt.

Panini, der Hersteller der vor allem bei Kindern beliebten Sticker-Sammelkarten-Päckchen mit Fußball-, Star-Wars- und anderen Lizenz-Motiven, hat den Trend zum digitalen *Collectible* erkannt und probiert sich derzeit am Handel mit Motiven amerikanischer Sport-Ligen. Die Preise dieser digitalen Sammelkarten reichen bereits von zehn Dollar bis zu mehreren Zehntausend Dollar [7]. Im deutschen Markt hat der FC Bayern München in Kooperation mit *Stryking* [8] einen Versuch gelauncht, Sammelkarten in Form von NFT, gekoppelt an ein Wettspiel, auf der Plattform

```
contract ExampleBet {
    struct Bet {
        address player;
        uint amount;
    }
    mapping(uint256 => mapping(address => Bet[])) public betAmounts;
    uint256 public total;
    // ...
    function bet(uint256 outcome) public payable {
        betAmounts[outcome].push(Bet(msg.sender, msg.value));
        total += msg.value;
        require(total < 2 ** 128); // avoid overflow
    }
    function payout(uint256 realoutcome) onlyowner {
        for(uint x; x < betAmounts[realoutcome].length; x++) { // ! potential overflow
            betAmounts[realoutcome][x].player.send(betAmounts[realoutcome][x].amount * 2);
        }
    }
}
```

Listing 1: Beispiel Auszug Smart Contract in Programmiersprache Solidity

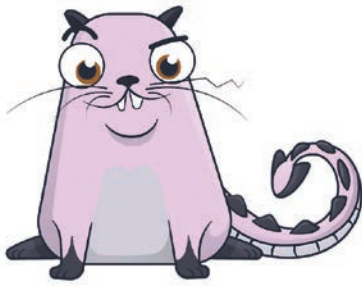


Abbildung 1: Cryptokitty #896775 Dragon (© Cryptokitties.co <https://www.cryptokitties.co/kitty/896775>)

zu etablieren. Borussia Dortmund lässt seine Spieler als digitale Karte auf der Plattform Swap [9] verbreiten, weitere Vereine werden auf die ein oder andere Art folgen. Der Ansatz dieser „Akteure aus der Offline-Welt“ ist etwas anders als der von *CryptoKitties* und Co.: Die Blockchain ist nicht-öffentlich, sie wird als Datenbank und Buchhaltungs-Technologie eingesetzt, nicht als Marktplatz. Panini erklärt dazu in den FAQ zum Marktplatz, dass man den Wert der Karte an eine stabile Währung binden wolle und deshalb keinen Handel in Crypto-Netzen unterstütze.

Virtuelle Vergesellschaftung

Der besondere Mehrwert von virtuellen Objekten, die durch ein NFT manifestiert sind, liegt allerdings gerade nicht in der Insel-Lösung einzelner Anbieter, sondern in der potenziellen Interoperabilität der *Tokens* mithilfe dezentraler, öffentlicher Blockchains: Wenn die erlangten Items, beziehungsweise deren *Token*, auf einer öffentlichen, etablierten Kryptowährungs-Blockchain liegen, können sie dezentral gehandelt, es kann damit spekuliert und sie können in anderen Anwendungen inhaltlich wiederverwendet werden. So ist beispielsweise denkbar, dass Videospiele ihre Assets wiederverwendbar machen oder aus anderen (beliebten) Spielen importieren, sei es als tatsächlich verwendbares Item der virtuellen Welt oder als Trophäe im Spielerprofil – die meisten Spieler lieben es zu zeigen, dass sie Veteranen oder besonders große Fans der Spiele sind [10].

Für den Nutzer von Videospiele ergibt sich aus solchen öffentlichen NFT-Collectibles also ein doppelter Mehrwert: Bei der Beendigung des Spielbetriebs bleibt das Invest (sei es in Form von Zeit oder Geld) digital erhalten, es hat über die Grenzen des Ursprungskontextes hinaus Verwendungsmöglichkeiten und es entsteht ein direkter monetärer Mehrwert durch die Möglichkeit des Handels und der Spekulation. Die Betreiber der Spiele

und damit auch Herausgeber der *Tokens* profitieren ebenfalls doppelt: Einerseits ist der gesamte Handel rund um die *Tokens* Content, der das Spiel attraktiver macht, und andererseits steigert die Wiederverwendung von Assets die Kundenbindung. Peer to Peer Economies sind die Harmonisierung von Spieler- und Entwickler-Incentives [10].

Der größte Marktplatz für NFT-Collectibles ist derzeit *OpenSea* [11]. Das Collectible mit dem höchsten Letztverkaufspreis ist eine Drachen-Cryptokitty für 600 Ether (siehe Abbildung 1), das entsprach zum Zeitpunkt der Transaktion etwa 172.000 US-Dollar. Ein so hoher Preis ist allerdings die Ausnahme und, da dieses Drachen-Kitty keine seltenen Eigenschaften besitzt, im konkreten Fall schwer zu erklären. Der durchschnittliche Handelspreis von *CryptoKitties* liegt bei 40,91 US-Dollar, der Median bei 2,71 US-Dollar [12]. Zu den Blockchain Games mit dem höchsten Transfervolumen gehören auch zwei Vorzeige-Projekte, was Vergesellschaftung und Dezentralisierung der Videospiele(assets) angeht: *Decentraland* [13] und *Cryptovoxels* [14]

Decentraland und *Cryptovoxels* sind beides virtuelle Welten, in denen Spieler sich bewegen und mit der Welt interagieren können. Der Clou ist, dass alle Parzellen dieser Welten eine Repräsentation als *Token* in der Blockchain haben, sodass die Welt von Nutzern verwaltet werden kann. Jedes *Crypto Token* verweist auf Assets, die wiederum in einer dezentralen Datenbank gehalten und vom Eigentümer verändert werden können. Neben den statischen Assets, wie 3D-Objekten, Texturen und Sounds, kann auch Programmlogik erstellt werden, sodass die Eigentümer von virtuellem Land dieses nach Belieben in eigene Spiele-im-Spiel umbauen können. Der so geschaffene User-Content sorgt wiederum dafür, dass der Wert der Parzelle steigt, während gleichzeitig *Decentraland* insgesamt attraktiver wird, wodurch auch der Wert anderer Parzellen steigt. Das Vertrauen in diese Werte kann inhärent durch die dezentrale Struktur und das Vertrauen in die Blockchain nachvollzogen werden – *Decentraland* kann nicht verschwinden, solange es Menschen gibt, die ein Interesse an dessen Erhalt haben und die Blockchain am Leben halten. Die Welt und ihre Werte werden demnach immer sicherer, je mehr Menschen zum Stakeholder werden.

Cheating, Bugs, Limitationen und deren Verhinderung

In einer Studie zur Sicherheit von Videospiele auf der Blockchain zeigen Min et al.

auf, dass eine Vielzahl der Spiele offensichtliche Sicherheitsprobleme hat [15]. Besonders interessant sind hierbei Sicherheitslücken und Bugs, die in *Smart Contracts* gegossen wurden: Da *Smart Contracts* auf der Blockchain geschrieben sind, ist auch der Inhalt des Contract Teil der unveränderlichen Kette. Was einerseits Vertrauen in die Authentizität der Programmlogik schafft, führt bei Sicherheitsmängeln zu großen Problemen. Nur 11,63 Prozent der untersuchten 1.311 *Contracts* waren frei von Bugs (siehe Abbildung 2). Im *Smart-Contract-Code* aus Listing 1 habe ich beispielhaft einen kleinen, aber fatalen Fehler eingebaut, der so auch in der echten Welt häufig auftritt: Transaktionen erzeugen in aller Regel Transaktionskosten, in *Ethereum* wird zum Beispiel „Gas“ verbraucht. Das verbrauchbare Gas pro Transaktion ist wiederum durch das Protokoll fest limitiert. Da die Beispiel-Funktion „*payout()*“ versucht, jedem Spieler Geld auszuzahlen, steigt die Menge an verbrauchtem Gas pro User. Sobald mehr Iterationen stattfinden, als Gas verbrauchbar ist, scheitert der *Smart Contract*, eine Teiltransaktion findet nicht statt. In diesem Fall wäre die korrekte Implementation, die Auszahlung der Wettsumme durch einen Pull der Teilnehmer auszulösen (= ein Teilnehmer pro Transaktion), anstatt proaktiv alle Transaktionen anzustoßen (= n Transaktionen bei n Teilnehmern).

Transaktionen auf der Blockchain sind oft zu langsam für Videospieleverhältnisse und der Speicher auf der Blockchain ist begrenzt, während Videospiele-Assets dazu neigen, sehr groß zu werden. Das hat beispielsweise zur Folge, dass die Bildrepräsentationen der *CryptoKitties* (wie in Abbildung 1) auf zentralen Servern des Spielbetreibers gelagert werden. Die *Crypto Tokens* selbst beinhalten lediglich eine URL zum zentral gespeicherten Bild, was der dezentralen Struktur der Blockchain widerspricht. Interessanter sind daher Lösungen, die auch Datenbanken beziehungsweise Speicher dezentral verteilen. *Decentraland* setzt zum Beispiel auf BitTorrent- und Kademia-Netze zur Verteilung der Assets. Andere mögliche Verteilungsarten sind das *Interplanetary File System* [16] oder *Solid Pods*. *Solid Pods* [17] sind ein Akronym für „personal online data storage“. Es handelt sich dabei zwar nicht um dezentralen Speicher, aber *Pods* stellen eine Lösung für das Problem dar, dass nicht alle Daten auf einer dezentralen Struktur gehalten werden sollen. Gerade bei persönlichen Daten kann durch *Pods* die Hoheit

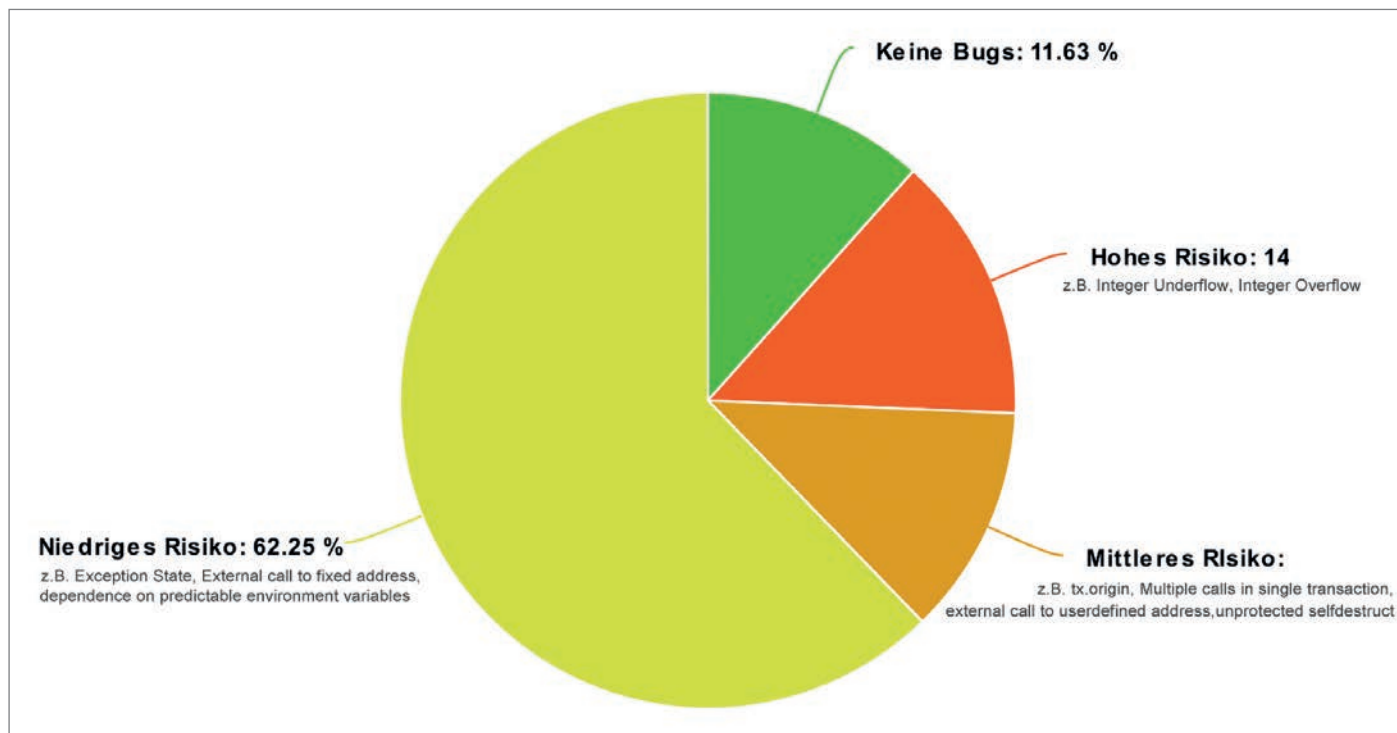


Abbildung 2: Piechart der Daten von Min et al. [15] (© GPL Dennis Walz)

über eigene Daten bei gleichzeitiger Einbindung in dezentrale Strukturen gewährleistet werden [18].

Zur Verbesserung der Performance müssen Aufgaben aufgeteilt werden – nicht alles kann auf der Haupt-Chain stattfinden. Eine denkbare Struktur sähe so aus, dass die Spielteilnehmer eine Session-Blockchain führen, in der alle Interaktionen ausgeführt und geloggt werden. Am Ende der Session werden die Spielergebnisse komprimiert in die Haupt-Chain geschrieben. Für solche Szenarien schlagen Yuen et al. vor, einen neuartigen Konsensmechanismus namens „Proof of Play“ einzusetzen [19]. Die Grundidee von Proof of Play ist analog zum üblichen Proof-of-Work-Consensus, dass Teilnehmer der Blockchain sich das Recht zum Schreiben eines Blocks verdienen müssen. Die These lautet, dass jeder Teilnehmer des Spiels grundsätzlich ein Interesse daran hat, dass das Spiel fair abläuft. Das „Mining“ soll in Form des natürlichen Spielens, ausgedrückt durch angemessen komplexe, nicht durch einen Bot durchführbare Aufgaben abgebildet werden.

Kleine, performante, Session-basierte *Sidechains* bieten außerdem die Möglichkeit, das Spiel effektiv gegen Betrug abzusichern, indem jede Spielertransaktion geloggt wird und zu jedem Zeitpunkt als Interaktionskette überprüft werden kann. In einem Feldversuch wurde ein Konzept erarbeitet, das die Überprüfung der User-Interaktionen mittels *Smart*

Contracts auf Session-basierten *Fabric Blockchains* durchführt [20]. Getestet wurde die Implementation auf Grundlage von „Doom“ (siehe Abbildung 3), einem „fast-paced Shooter“. Diese Art Videospiel zeichnet sich durch besonders schnelle Aktionsabläufe und hohe Anforderung an Synchronizität und Genauigkeit aus. Das Problem, dass nicht jeder Spieler zu jedem Zeitpunkt den Status der anderen Spieler kennen soll, wird durch sogenannte *Enclaves* in Kombination mit den passenden *Smart Contracts* gelöst. Im Falle des Shooters darf der Spieler beispielsweise keine Information über die Position des Gegners bekommen. Diese Daten müssen daher gekapselt im *Smart Contract* evaluiert werden, ohne dass unehrliche Spieler die Information aus der Blockchain extrahieren können. Im Versuch gelang es, das Anti-Cheat-System mit einer Latenz von unter 150 Millisekunden bei 32 Teilnehmern zu etablieren. „Doom“ ist in der Originalversion 1993 erschienen und hat dadurch vergleichsweise wenig Umfang bezüglich zusätzlicher Elemente, die neben dem Spieler und Projektilen womöglich ebenso in der Blockchain berechnet werden müssten. Dennoch zeigt der Versuch, dass ein Einsatz der Blockchain zur Cheat-Verhinderung jetzt schon möglich ist.

Was uns erwartet

CryptoKitties, als erstes großes Blockchain-basiertes Spiel, zeigt zugleich die Limitie-

rungen von Token-basierten *Collectible Games* auf: Kein noch so seltenes *Collectible* hat einen Wert, wenn es keine Spieler gibt, die Interesse daran haben, und gleichzeitig hängt der auszahlbare Wert des *Tokens* vom Kurs der Kryptowährung ab, auf der das *Token* basiert [2]. Für den Erfolg digitaler *Collectibles* ist daher entscheidend, dass es für den Nutzer mehr Wert gibt als die reine Partizipation an einem spekulativen Drittmarkt innerhalb des Crypto-Ökosystems.

Anhand der Beispiele *Decentraland* und *Cryptovoxels* zeigt sich gut, wie öffentliche Blockchains den Nutzer von Anwendungen noch essenzieller zum Stakeholder der Anwendung machen, als das bisher bereits durch zeitliches, monetäres und emotionales Invest in die Lieblingsspiele passiert. Durch die dezentrale Struktur hängen die Verfügbarkeit und Lebensdauer der Anwendung genauso essenziell von den Nutzern ab wie der Wert virtueller Güter und das Regelwerk der virtuellen Welt. Dadurch, dass Blockchains per Definition keine nachträgliche Änderung der Einträge im Register erlauben, wird die effektive Kontrolle der Aktivitäten von Nutzern möglich, was einen optimalen Schutz gegen Betrug für alle Parteien darstellen kann. Die automatisch entstehende Geschichte einer virtuellen Welt kann obendrein weiteren Wert erzeugen: Wenn die Besitzhistorie eines Gegenstandes – zum Beispiel eines Hutes – nachvollziehbar

ist, dann liegt es nahe, dass dieser Hut besonderen Wert erlangt, wenn er von einem populären Spieler getragen wurde. Dies gilt auch rückwirkend, so wie im Beispiel des Sport-Trikots aus der analogen Welt.

Verteilte, dezentrale Datenbanken und passende Storage-Lösungen für Daten, die aufgrund des Datenschutzes oder technischer Bedingungen wie der Dateigröße keinen Platz auf der Blockchain finden, ebenso wie die Interoperabilität verschiedener Services und unterschiedlicher Blockchain-Protokolle sind relevante Themen, deren Lösung direkten Einfluss auf nicht-videospielbezogene Anwendungen hat. Im Rahmen einer Standardisierung werden wir hier eine Entwicklung von Protokollen und Architekturen sehen: Besançon et al. schlagen zum Beispiel eine Schichten-Struktur ähnlich dem OSI-Modell vor [21]. Zu einem Blockchain-Software-Stack wird in einigen Anwendungen auch die Kombination von (mehreren) öffentlichen und privaten Blockchains gehören, da sich durch Sidechains Skalierungsprobleme lösen lassen und gegebenenfalls weiterhin existierende „Softwaregeheimnisse“ abgebildet werden sollen.

Zugleich wächst der Bedarf, einzelne Komponenten durch neuere, verbesserte beziehungsweise dem Anwendungsfall angemessenere Technik zu ersetzen, wodurch auch der Markt für Blockchain-Middleware größer wird, um genau das zu ermöglichen. Bestrebungen diesbezüglich werden beispielsweise von *Bifrost* [22] entwickelt: Durch ein vereinheitlichtes Middleware-API können Blockchain-Mechanismen gewissermaßen „as a service“ genutzt und die darunterliegende Blockchain weitestgehend ignoriert werden. Durch diese Entwicklungen wird die Integration von Blockchains in komplexere Anwendungen, mit für die Blockchain eigentlich untypischen Anforderungen wie großen Datenmengen oder Real-Time-Kommunikation, unkompliziert möglich. Das wiederum bringt die Verbreitung der Blockchain-Technologie voran.

Die Ideen, eine digitale Wertschöpfungskette in die Blockchain zu gießen und Eigentum durch *Token* auszudrücken, werden wir aufgrund der genannten Vorteile in Zukunft öfter sehen. Videospiele haben sich in breitem Feld bereits von Pay-to-Play nach Free-To-Play-Modellen entwickelt. Der nächste logische Schritt steht vor uns: Play-To-Earn bietet dem Spieler die Möglichkeit, seine erspielten Fortschritte aus dem Spiel „mitzunehmen“ oder zu veräu-

ßern. Und selbst wenn sie keinen monetären Wert haben: Ein wertloses *Token* ist immer noch besser, als alles zu verlieren, wenn der Spielbetrieb eingestellt wird, und schließlich gibt es immer noch die theoretische Chance, dass ein *Token* doch nochmal an Wert gewinnt.

Cheating, oder vielmehr Betrug, ist kein exklusives Thema von Videospiele. Techniken, die darauf setzen, eine Ereigniskette und deren Teilnehmer gegeneinander abzusichern, können auch in Produktions- und Lieferketten Anwendung finden. So können für Produzenten und Verbraucher gleichermaßen transparente, manipulationsichere Blockchains geführt werden, die den Weg eines Produkts von den Rohstoffen bis zum Kauf abbilden. Das funktioniert besonders gut, wenn die Güter selbst digital nachverfolgbar sind: Ein gutes Beispiel hierfür ist der Strommarkt, der sich durch erneuerbare Energien dezentralisiert und diversifiziert. Die Blockchain ist geeignet, die Teilnehmer zu synchronisieren und Betrug zu verhindern. Dieser und andere Anwendungsfälle sind nahezu identisch zu den Abläufen in der virtuellen Welt. Demzufolge sind die Entwicklungen von Sicherheitsmechanismen für (dezentrale) Videospiele vielversprechend.

Das Konzept der Interoperabilität und Marktfähigkeit von NFT wird auch außerhalb von Videospiele weiter Beliebtheit erlangen. Die Blockchain ist perfekt geeignet, den Wert von Sammelobjekten zu manifestieren. Betrachtet man Blockchain als digitales Grundbuch, liegt der Einsatz von NFT zur Repräsentation von anderen Eigentümern, wie Immobilien oder Kraftfahrzeugen, nahe. Die Versuche hierzu sehen wir in den obigen Ausführungen. Mithilfe entsprechender *Smart Contracts* könnte so auch der Eigentumswechsel von Gütern, die aufgrund ihres (gesellschaftlichen) Werts durch Grundbuch, Fahrzeugbrief oder andere Papiere abgesichert werden müssen, reibungsfreier durchgeführt werden, ohne die gewünschten notariellen Sicherheiten zu verlieren. Noch näher liegt die Verwendung von NFT als Abbildung von (digitalen) Lizenzen inklusive der Regulation von Vielfältigungs- und Veräußerungsrechten.

Activision Blizzard, mit rund 6.5 Milliarden Euro Jahresumsatz einer der Top-10-Videospiel-Publisher, hat ein Patent angemeldet, in dem von einer Blockchain-basierten Plattform zur Aggregation von normalisierten Statistiken und vom Austausch von



Der grüne Faden für Ihre Digitale Evolution

Wir bei PROMATIS folgen einem selbst entwickelten grünen Faden:

Mit professioneller Beratung und innovativen Digitalisierungslösungen schaffen wir exzellente Geschäftsprozesse: agil, bedarfsgerecht, intelligent und zukunftssicher. Nachhaltige Qualität und Wirtschaftlichkeit sichern wir durch kontinuierliche Verbesserung der eingesetzten Verfahren, Produkte und Services.

Mit unserer Digitalisierungskompetenz und unseren Best Practice-Lösungen begleiten wir Sie auf Ihrer Reise in die Oracle Cloud.

PROMATIS Gruppe
Pforzheimer Str. 160
76275 Ettlingen
+49 7243 2179-0
www.promatis.de

Ettlingen | Hamburg | Berlin | Wien | Zürich | Denver

Videospielinhalten zwischen verschiedenen Videospielen die Rede ist [23]. Damit ist klar: Blockchain-Technologie wird in der Spielebranche in naher Zukunft nicht mehr wegzudenken sein.

Power to the People

Die Revolution entsteht durch die Kombination der genannten Features der Blockchain-Lösungen. Der Nutzer erhält durch Dezentralität, Kollektivität und Persistenz der Blockchain Mündigkeit gegenüber dem Entwickler oder Anbieter. Er bekommt die Chance, die Regeln des Systems anhand von *Smart Contracts* zu kontrollieren und selbst mitzugestalten. Er kann als Teilnehmer der Blockchain zu deren Verlässlichkeit beitragen und direkt am Erfolg partizipieren. Er kann Eigentum erwerben, plattformunabhängig veräußern und selbst gestalten. Je populärer diese Möglichkeiten (durch Videospiele) werden, desto mehr werden die Nutzer gegen klassische Strukturen (auch in anderen Bereichen) revolütieren und ihre Emanzipation, als wichtigste Stakeholder des Produkts, einfordern.

Mit Blick auf die Bugs in *Smart Contracts* ist es begrüßenswert, dass Blockchain-Technologie weitläufig im Kontext von „harmlosen Videospielen“ erprobt wird, ehe wir uns als Gesellschaft abhängig von dieser Technologie machen. Lassen wir sie erst mal erwachsen werden, lange wird das nicht mehr dauern. Die Aussichten sind, nicht nur für Videospiele, verheißungsvoll.

Quellen

Alle Links wurden das letzte Mal am 04.10.2020 abgerufen.

- [1] Jahresreport der deutschen Games-Branche 2019
<https://www.game.de/guides/jahresreport-der-deutschen-games-branche-2019/>
- [2] Serada, Alesja & Sihvonen, Tanja & Harviainen, J. Tuomas (2020). CryptoKitties and the New Ludic Economy: How Blockchain Introduces Value, Ownership, and Scarcity in Digital Gaming. Games and Culture. 10.1177/1555412019898305.
- [3] <https://blog.earnbet.io/en/earnbet-announced-as-top-decentralized-casino/>
- [4] <https://eips.ethereum.org/EIPS/eip-721> & <https://eips.ethereum.org/EIPS/eip-1155>
- [5] <https://www.cryptokitties.co/>
- [6] Dapp Toplisten:
<https://www.dapp.com/dapps> & <https://dappradar.com/>
- [7] Panini Blockchain FAQ
<https://www.paniniamerica.net/blockchain/>
- [8] <https://stryking.io/>



Abbildung 3: Doom Chocolate (© <https://www.chocolate-doom.org/wiki/index.php/File:Chocolate-doom.png>)

- [9] <https://fantastec-swap.io/>
- [10] QIAO, D. (2020). This is Not a Game: Blockchain Regulation and Its Application to Video Games.
- [11] <https://opensea.io/>
- [12] Statistik über Crypto Kitty - Verkäufe
<https://kittysales.co/>
- [13] <https://decentraland.org/>
- [14] <https://cryptovoxels.com/>
- [15] Min, T., & Cai, W. (2019). A Security Case Study for Blockchain Games. 2019 IEEE Games, Entertainment, Media Conference (GEM), 1-8.
- [16] <https://ipfs.io/>
- [17] <https://solidproject.org/>
- [18] Manoharan Ramachandran, Niaz Chowdhury, Allan Third, John Domingue, Kevin Quick, and Michelle Bachler. 2020. Towards Complete Decentralised Verification of Data with Confidentiality: Different ways to connect Solid Pods and Blockchain. In Companion Proceedings of the Web Conference 2020 (WWW '20). Association for Computing Machinery, New York, NY, USA, 645–649. DOI: <https://doi.org/10.1145/3366424.3385759>
- [19] Ho Yin Yuen, Feijie Wu, Wei Cai, Henry C.B. Chan, Qiao Yan, and Victor C.M. Leung. 2019. Proof-of-Play: A Novel Consensus Model for Blockchain-based Peer-to-Peer Gaming System. In Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI '19). Association for Computing Machinery, New York, NY, USA, 19–28. DOI: <https://doi.org/10.1145/3327960.3332386>
- [20] Sukrit Kalra, Rishabh Sanghi, and Mohan Dhawan. 2018. Blockchain-based real-time cheat prevention and robustness for multi-player online games. In Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '18). Association for Computing Machinery, New York, NY, USA, 178–190. DOI: <https://doi.org/10.1145/3281411.3281438>
- [21] Besançon, L., Silva, C., & Ghodous, P. (2019). Towards Blockchain Interoperability: Improving Video Games Data Exchange. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 81-85.
- [22] <https://thebifrost.io/>
- [23] US Patent #10765948
<http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetathtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=10765948.PN.&OS=PN/10765948&RS=PN/10765948>



Dennis Walz

dennis@code-alliance.de

Dennis Walz ist Medieninformatiker und Anwendungsentwickler bei Code Alliance Berlin mit dem Schwerpunkt Mobile Development und Schnittstellenarchitektur. Als „Videospiel-Native“ begleitet er progressiv den Entwicklungsprozess von Enterprise- und Consumer-Anwendungen.



Enterprise Blockchain am Beispiel eines internationalen Produktanforderungssystems

Tim Rüb, esentri AG

Blockchain ist nicht nur Bitcoin und kann auch Aufgaben abseits von öffentlichen Finanztransaktionen lösen – das hat sich mittlerweile herumgesprochen. Doch wie genau unterscheidet sich Blockchain im Unternehmenskontext (Enterprise Blockchain) von Kryptowährungen und Co.? Welche Probleme kann ich mit Blockchain und Smart Contracts (Verträgen auf der Blockchain) lösen und was gibt es dabei zu beachten? Anhand eines Proof of Concept für die METRONOM, die Digitaleinheit des Handelskonzerns METRO, folgt ein kurzer Überblick über die Grundbausteine von Enterprise Blockchain und ein konkretes Anwendungsbeispiel dafür, wie Vertrauen entlang von Lieferketten wiederhergestellt werden kann.

Was genau macht eine Blockchain?

Um festzustellen, ob eine Blockchain im Unternehmensumfeld helfen kann, Vorsprung zu generieren, ist vorab zu klären, was eine Blockchain ist. Ursprünglich wurde als Blockchain der ersten Generation (Blockchain 1.0) nur der Anwendungsfall Kryptowährungen genannt, die bekannteste – Bitcoin – allen voran. In Generation 2.0 kamen sogenannte *Smart Contracts* hinzu, Verträge, mit denen sich die Finanztransaktionen der ersten Generation an Bedingungen knüpfen ließen.

Im Gegensatz zu klassischen Verträgen auf Papier waren diese *Smart Contracts* jedoch in Form von Programmcode auf der Blockchain gespeichert. So wäre beispielsweise ein Vertrag denkbar, der pro gefahrenem Kilometer meines Mietwagens automatisch einen Betrag x einer Kryptowährung von meinem Konto abbucht. Dafür benötigt der *Smart Contract* Zugriff auf den Kilometerstand des Autos und eine Abbuchungsermächtigung meines Kontos, der Rest geschieht automatisiert.

Intelligente Verträge lassen sich jedoch nicht nur auf Finanztransaktionen anwenden, sondern auch auf andere zu automatisierende Unternehmensprozesse, in denen verschiedene Parteien, Geschäftspartner und andere Externe beteiligt sind. Man spricht hier von Blockchain 3.0 [1]. Ein klassisches Beispiel stellt die Lieferkettennachverfolgung dar, bei der die einzelnen Bestandteile eines Produkts bis zu ihrem Ursprungsort lückenlos rückverfolgbar sind. Dafür ist es notwendig, dass jede Person, die

die Ware verarbeitet, empfängt oder weitertransportiert, diese Statusänderung auf der Blockchain speichert.

Der METRO-Anwendungsfall im Detail

Ein interessanter Anwendungsfall für die Blockchain ist eine Proof-of-Concept-Anwendung, die für die METRONOM entwickelt wurde. Es bildet ein Produktsystem für die METRO ab, das Eigenmarkenprodukte (beispielsweise einen Schokoriegel oder eine Streichcreme wie in *Abbildung 1* dargestellt) bei potenziellen Herstellern und Lieferanten anfragen kann. Verkompliziert wird dieser Prozess, da jede Landesgesellschaft und damit jedes Land, in dem das Produkt verkauft werden soll, unterschiedliche Produktkriterien von den Lieferanten fordert. Die Anforderungen an die Produkte sind dabei zahlreich, von A wie Anzahl Produkte pro Packungseinheit bis Z wie Zuckeranteil. Bevor das Produkt in dem jeweiligen Land vertrieben werden darf, muss eine Zustimmung für jedes dieser Kriterien von der jeweiligen Landesgesellschaft erfolgt sein. Ebenso muss der Hersteller garantieren, dass er die Produkte in der gewünschten Qualität herstellen und liefern kann.

Digitalisierung mit der Blockchain

Diese Prozesse der Abstimmung galt es zu digitalisieren. Bisher wurden die oben genannten Prozesse zu großen Teilen manuell und mit Medienbrüchen abgewickelt. Die dazugehörigen Daten wurden wenig transparent in einem Dokumentenmanagementsystem abgelegt. Insbesondere durch Unstimmigkeiten und Produktionsfehler seitens der Lieferanten, die aus diesen Unklarheiten entstehen, kann es wiederholt zu zeitaufwendiger und kostenintensiver Nacharbeit kommen.

Product	Supplier	Country Status	Supplier Status
Schokoriegel	Nestle	100%	50%
Streichcreme	Mondelez	66%	100%
Lemon Drink	PepsiCo	33%	0%

Abbildung 1: Übersicht über Produktanforderungsprofile (© esentri AG)

Hier kommt nun die Blockchain-Anwendung ins Spiel. Mit ihr ist es möglich, lückenlos und unveränderbar die Anforderungen an die Produkte durch die einzelnen Landesgesellschaften zu erfassen und auf der Blockchain zu speichern. Anschließend können andere Landesgesellschaften, wie in *Abbildung 2* dargestellt, abstimmen, ob diese Produktanforderungen mit ihren eigenen Anforderungen übereinstimmen. So ist ein Land beispielsweise durch Gesetzesvorgaben mit der Packungsgröße nicht einverstanden oder die Kunden des Landes präferieren einen anderen Farbton des Produkts. Der Lieferant kann nun ebenfalls zustimmen, falls er das Produkt zu den angegebenen Kriterien produzieren kann. Dafür stellt (im einfachsten Fall) der Betreiber der Blockchain einen Account für den Lieferanten bereit, mit dem nur dieser sich auf der Plattform einloggen und abstimmen kann. Handelt es sich um einen größeren Lieferanten oder regelmäßigen Kunden, ist ebenfalls ein eigener Knoten und somit die vollständige Teilnahme am Netzwerk denkbar.

Vorteil des Systems ist dabei die Transparenz bei der Abstimmung von Produktanforderungen. Hat ein Hersteller zugestimmt, das Produkt zu den gegebenen, oben genannten Konditionen (beispielsweise Größe oder Gewicht) herzustellen, wird dies auf der Blockchain gespeichert und kann nachträglich nicht mehr verändert werden. Die Konditionen werden dabei von den Ländern vorab festgelegt. Die Zustimmung durch die Hersteller ist durch die Blockchain verbindlich und unveränderlich. Dies gilt ebenfalls für die entsprechenden Landesgesellschaften. Wurde dort eine Produkteigenschaft als „akzeptiert“ gekennzeichnet, so kann diese nicht im Nachhinein beanstandet werden. Die Blockchain bringt somit Vertrauen in die Geschäftswelt. Sichtbar erscheint die vollständige Änderungshistorie eines Produktanforderungsprofils in tabellarischer Form in einer Timeline, die jede Änderung eines Nutzers zeitlich aufgeschlüsselt darstellt (siehe *Abbildung 3*).

Das erstellte System überzeugt vor allem mit dem ermöglichten Performance-Gewinn in Form von kürzerer Bearbeitungszeit für ein Produktanforderungsprofil. Da die Anzahl von Telefonaten und E-Mails mithilfe des Systems drastisch reduziert werden kann und nun lediglich in Sonderfällen nötig ist, findet die Abstimmung und Freigabe deutlich effizienter und effektiver statt. Geld und Ressourcen werden gespart!

Vorteile der Private Permissioned Blockchain

Bei der eingesetzten Blockchain handelt es sich um die Oracle-Blockchain-Plattform, die auf *Hyperledger Fabric* basiert. Durch die Cloud-Lösung ist dabei eine einfache Einrichtung der Blockchain garantiert, an-

No.	Name	Value	Approve
1	thickness	8mm	<input checked="" type="checkbox"/>
2	cooking-time	4	<input type="checkbox"/>
3	broken	3	<input checked="" type="checkbox"/>

Options selected: 1 Publish

Abbildung 2: Ein Anforderungsprofil im Detail (© esentri AG)

dere Hyperledger-Fabric-Instanzen können ebenfalls am Netzwerk teilnehmen. Dies ist beispielsweise der Fall, wenn ein großer Lieferant mit einer internen IT einen eigenen Hyperledger-Fabric-Knoten betreiben will. Er setzt intern oder ebenfalls in der Cloud ein weiteres Netzwerk auf und verknüpft dieses mit dem METRO-Netzwerk.

Der Vorteil von *Hyperledger Fabric* im Gegensatz zu öffentlichen, bekannten Blockchains wie Bitcoin und Co. ist, dass diese nur auf Einladung genutzt werden kann. Bei der Erstellung des Netzwerks ist dabei festzulegen, wer andere Nutzer zum Netzwerk hinzufügen darf. Im METRO-Proof-of-Concept ist dies die METRO selbst. Es handelt sich also um eine private Blockchain. Insbesondere im Geschäftsumfeld ist es oft nicht gewünscht, dass externe Teilnehmer, in diesem Fall Drittfirmen, am Konsensprozess teilhaben können. Eine weitere Besonderheit betrifft die Rollenverteilung im System selbst. So können beispielsweise die Lieferanten nicht den Status der Konkurrenz einsehen; die Daten sollen je nach Rolle des eingeloggteten Nutzers sichtbar sein. Um das zu ermöglichen, ist eine *Permissioned Blockchain* erforderlich, in der Teilnehmer unterschiedliche Rechte besitzen. Bei *Hyperledger Fabric* handelt es sich folglich um eine private *Permissioned Blockchain*. Die Nutzer werden dabei von der Oracle Cloud über den Oracle-Identity-Cloud-Service verwaltet, wenn die Cloud-Variante der Oracle Blockchain verwendet wird. Als vollwertiges Identitätssystem ist die Benutzerverwaltung über die Weboberfläche komfortabel möglich.

Smart Contracts – die Technik dahinter

Wie genau funktioniert jedoch die Digitalisierung von Geschäftsprozessen im Blockchain-Umfeld? Im Normalfall besteht eine Anwendung aus einer Datenbank zur Persistierung der Daten, einem Backend zum Verarbeiten dieser Daten und einem Frontend, um die Daten anzuzeigen. Der Oracle-Blockchain-Service setzt am Frontend an und stellt eine REST-Schnittstelle zur Verfügung, über die Daten aus der Blockchain gelesen und gespeichert werden können. Dies geschieht über vorab in einem *Smart Contract* definierte Funktionen, die mit Parametern ausgeführt werden. Der *Smart Contract*, er wird *Chaincode* bei *Hyperledger Fabric* genannt, ersetzt also das Backend einer klassischen Anwendung. Trotzdem kann zur

Streichcreme			
Mondelez			
Status		Timeline	Attribute Details
Filter			
thickness			
No.	Country	Date	Attribute
3	Germany	Wed Jan 02 2019 00:00:00 GMT+0100 (Mitteleuropäische Normalzeit)	Thickness
6	Austria	Thu Jan 03 2019 00:00:00 GMT+0100 (Mitteleuropäische Normalzeit)	Thickness
9	Switzerland	Tue Jan 15 2019 00:00:00 GMT+0100 (Mitteleuropäische Normalzeit)	Thickness
10	Great Britain	Fri Jan 18 2019 00:00:00 GMT+0100 (Mitteleuropäische Normalzeit)	Thickness

Abbildung 3: Die Transaktionshistorie eines Produkts (© esentri AG)

zusätzlichen Datenverarbeitung eine weitere Schicht (Backend) zwischen die Oracle Blockchain und der Benutzerschnittstelle geschaltet werden. Dies hat den Vorteil, dass die Datenverarbeitung und eventueller Zugriff auf Schnittstellen auch außerhalb der Blockchain erfolgen können.

Die Daten werden dabei in Form eines Transaktionslogs auf der Blockchain gespeichert, indem jede Änderung von Daten vermerkt wird. In *Abbildung 3* ist zu sehen, dass vier Länder jeweils über das Attribut „Thickness“ der Streichcreme abgestimmt haben. Die Attributzustimmung ist die Voraussetzung, um das Produkt in dem jeweiligen Land zu verkaufen. Zusätzlich wird der aktuelle Datenstand in einer sogenannten State-Datenbank gespeichert, die schnelle Abfragen ermöglicht [2]. Die State-Datenbank lässt sich jedoch jederzeit aus der vollständigen Transaktionshistorie (dem Ledger) erneut erstellen.

Fazit

Enterprise Blockchain ermöglicht eine transparente und unveränderliche Art der Digitalisierung von bestehenden Geschäftsprozessen. Es wird das Vertrauen in Geschäftspartner mithilfe der Blockchain-Technologie gesichert und gestärkt, Daten werden manipulationsicher gespeichert. Das Misstrauen entlang wachsender Lieferketten und die daraus folgenden komplexen Kontrollprozesse lassen sich effektiv durch *Smart Contracts* ersetzen und bringen Sicherheit, Geschwindigkeit und Kostenreduktion in diese Prozesse. Cloud-Dienste wie der Oracle-Blockchain-Service unterstützen dabei den schnellen Aufbau eines Blockchain-Netzwerks durch das Bereitstellen der notwendigen Infrastruktur. Durch

diese vorgegebene Konfiguration wird die Produktentwicklungszeit verkürzt.

Quellen

- [1] Blockchain Technologien, Forschungsfragen und Anwendungen - Fraunhofer Gesellschaft:
https://www.aisee.fraunhofer.de/content/dam/aiese/Dokumente/Publikationen/Studien_TechReports/deutsch/FhG-Positionspapier-Blockchain.pdf
- [2] Oracle Blockchain Dokumentation - Oracle:
<https://docs.oracle.com/en/cloud/paas/blockchain-cloud/usingoci/query-state-database.html>



Tim Rüb

tim.rueb@esentri.com

Tim Rüb ist seit 2019 bei der esentri AG beschäftigt und studiert Informatik (M. Sc.) an der Hochschule Karlsruhe – Technik und Wirtschaft. Er unterstützt die esentri AG im Bereich DevOps und Blockchain.

System Lifecycle Management – Auf dem Weg zur Digitalisierung des Engineerings, Teil 2

Martin Eigner, EIGNER Engineering Consult

Die Einsatzbedingungen der Produktdatenverwaltung haben sich seit 1985 permanent verändert. Anfangs war 2D M-CAD der Schwerpunkt von Product Data Management (PDM). Bereits durch die 3D-M-CAD-Systeme hat sich die Komplexität der Datenverwaltung drastisch erhöht. Der nächste große Evolutionsschritt wurde durch die schnelle Zunahme mechatronischer Produkte und den Wunsch von Anwendern, die Anwendung von PDM über das Kerngebiet der Entwicklung und Konstruktion zu erweitern, ausgelöst (Product Lifecycle Management, PLM). IoT (Internet of Things), IoS (Internet of Service) und die daraus resultierende globalisierte interdisziplinäre Entwicklung cybertronischer Produkte und Systeme, permanent zunehmende Software- und Elektronikanteile am Produkt, höhere Anforderungen an die unternehmensinterne und -externe Zusammenarbeit sowie sich parallel dazu entwickelnde interdisziplinäre Engineering-Prozesse und -Methoden (Model Based Systems Engineering, MBSE) führten in den letzten Jahren zur Erweiterung des PLM-Ansatzes um das System Lifecycle Management (→ SysLM). Dieses bildet das Rückgrat [1].

Anm. d. Red.: Teil 1 behandelte, neben der genauen Definition des Begriffs SysLM, die Geschichte der Produktverwaltung, den Weg zum System Lifecycle Management durch Interdisziplinarität, moderne Engineering-Prozesse, neue Anwendungsfunktionen, Softwaretechnologie und innovative Geschäftsmodelle der Systemanbieter. Dabei wurden sowohl die typischen Funktionen von SysLM abgebildet, unterteilt in Anwendung und Plattform, als auch eine IT-Infrastruktur auf fünf Ebenen mit einem zentralen SysLM-Backbone.

Korrektur: In Teil 1 haben wir aus Versehen die *Abbildung 2* ausgelassen, womit auch die Beschriftungen für die folgenden Bilder durcheinander geriet. Wir haben das im entsprechenden Web-PDF nachträglich korrigiert und bitten um Entschuldigung. Das korrigierte Web-PDF können Sie unter dieser URL aufrufen: https://backoffice.doag.org/formes/pubfiles/12579009/docs/Publikationen/DOAGNews/2020/05-2020/05_2020-Business_News-Martin_Eigner_System_Lifecycle_Management_Auf_dem_Weg_zur_Digitalisierung_des_Engineerings_Teil_1.pdf

SysLM – das Rückgrat der Digitalisierung des Engineerings

Das Internet der Dinge und Services (IoT/IoS) sowie Industrie 4.0 gehen in der Zukunft von vernetzten Produkten, Systemen und Dienstleistungen aus. Der wertmäßige Anteil an Elektronik und Software wird bei dieser Art von Produkten und eingebetteten Dienstleistungen kontinuierlich steigen. Kommunizieren Produkte miteinander über das Internet, wird von Cyber-Physical Systems beziehungsweise cybertronischen Systemen gesprochen. Die Entwicklung dieser neuen Systeme wird mehrere Konsequenzen nach sich ziehen: interdisziplinäre, regional und organisatorisch verteilte sowie integrierte Produktentwicklung, ein Überdenken heutiger Konstruktionsmethoden, Prozesse, IT-Lösungen und Organisationsformen sowie die For-

derung nach durchgängigen Prozessketten, basierend auf digitalen Modellen in der Anforderungsdefinition, Systemarchitektur, Produktentwicklung, Simulation, Produktionsplanung, Produktion und Service. Weiterhin müssen Planungs- und Entwurfsmethoden aller Disziplinen – Mechanik, Elektronik und Software – auf den Prüfstand gestellt und auf ihre Tauglichkeit für ein neues Vorgehensmodell der Produkt- und Serviceentwicklung überprüft werden, um diese in einen gemeinsamen, integrierten und interdisziplinären Methoden-, Prozess- und IT-Lösungsansatz zu überführen. Dieser Ansatz der Digitalisierung der Produktentwicklung wird Engineering 4.0 genannt. Die Grundlagen bilden Methoden des Model Based Systems Engineering (MBSE). MBSE basiert auf der Erzeugung, Vernetzung und Interaktion von Lebenszyk-

lusphasen-spezifischen digitalen Modellen. Aus dem Produktionsmodell entsteht als „as built“-Stückliste die Basis des Digital Twin, der wiederum die Grundlage für Service-orientierte Geschäftsmodelle bildet. Durch die Vernetzung der Partialmodelle wird der Digital Thread ermöglicht, die Verbindung der in den Lebenszyklus-spezifischen Partialmodellen abgelegten Informationen (*siehe Abbildung 5*).

Ein durchgängiges Digital Model, Digital Thread und Digital Twin erlauben neben der Optimierung von Änderungsmanagement und Schadensanalyse über verschiedene Wertschöpfungsprozesse hinweg auch die Ausschöpfung verschiedenster Formen digitaler Geschäftsmodelle und über Produkte angebotene Dienstleistungen. Fällt eine Komponente im Betrieb durch Schadenshäufigkeit auf, so muss

durch rückwärtiges Abarbeiten des Digital Thread die Ursache des Problems ermittelt werden.

Die Digitalisierung der Produkte und der Engineering-Prozesse bedeutet einen Transformationsprozess, der die klassischen Gren-

zen einer fragmentierten und konkurrierenden IT-Lösungswelt neu ordnet. Weg vom Silodenken zu einem durchgängigen integ-

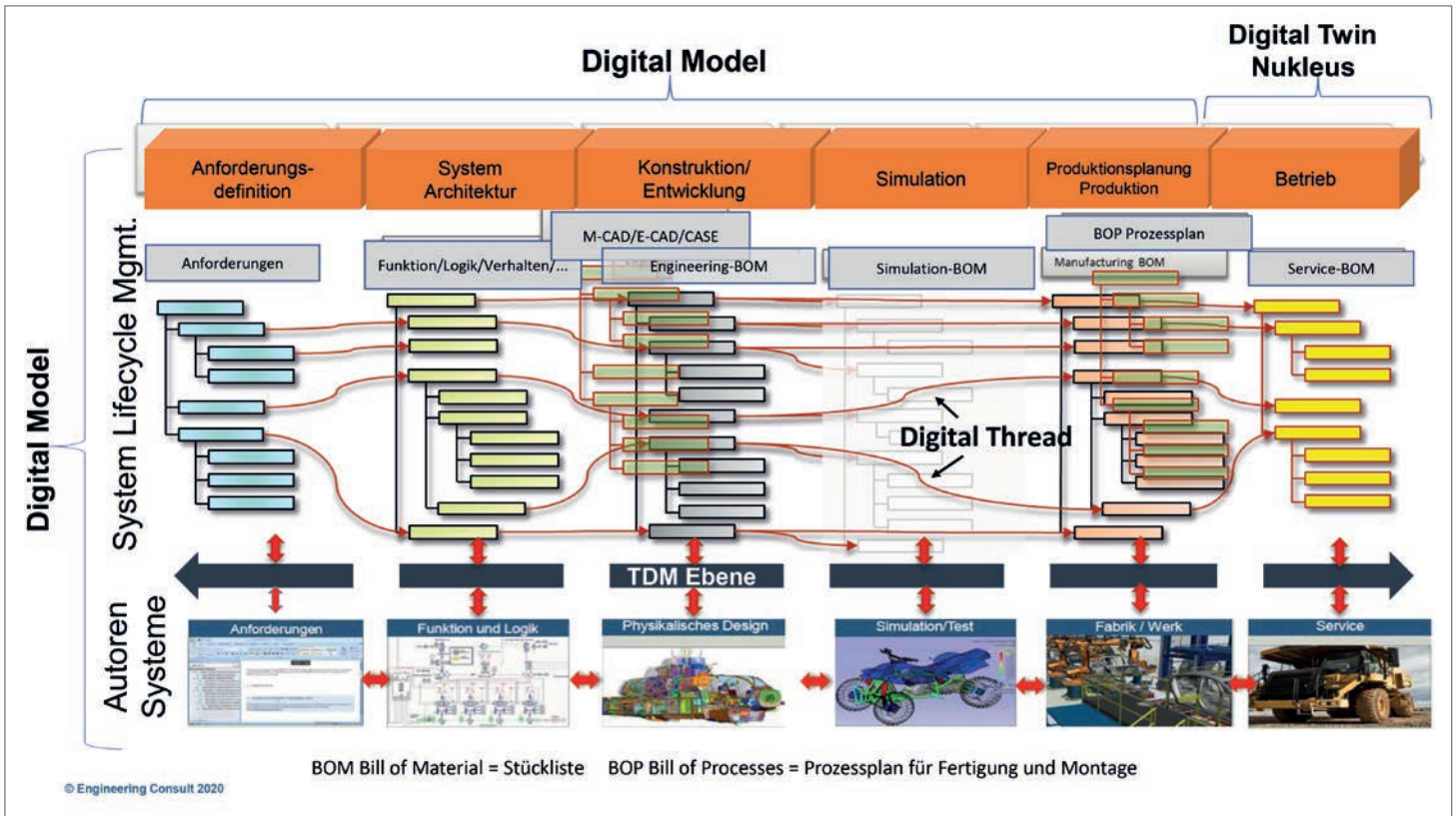


Abbildung 5: SysLM als Grundlage des Digital Model, des Digital Twin und des Digital Thread [1]

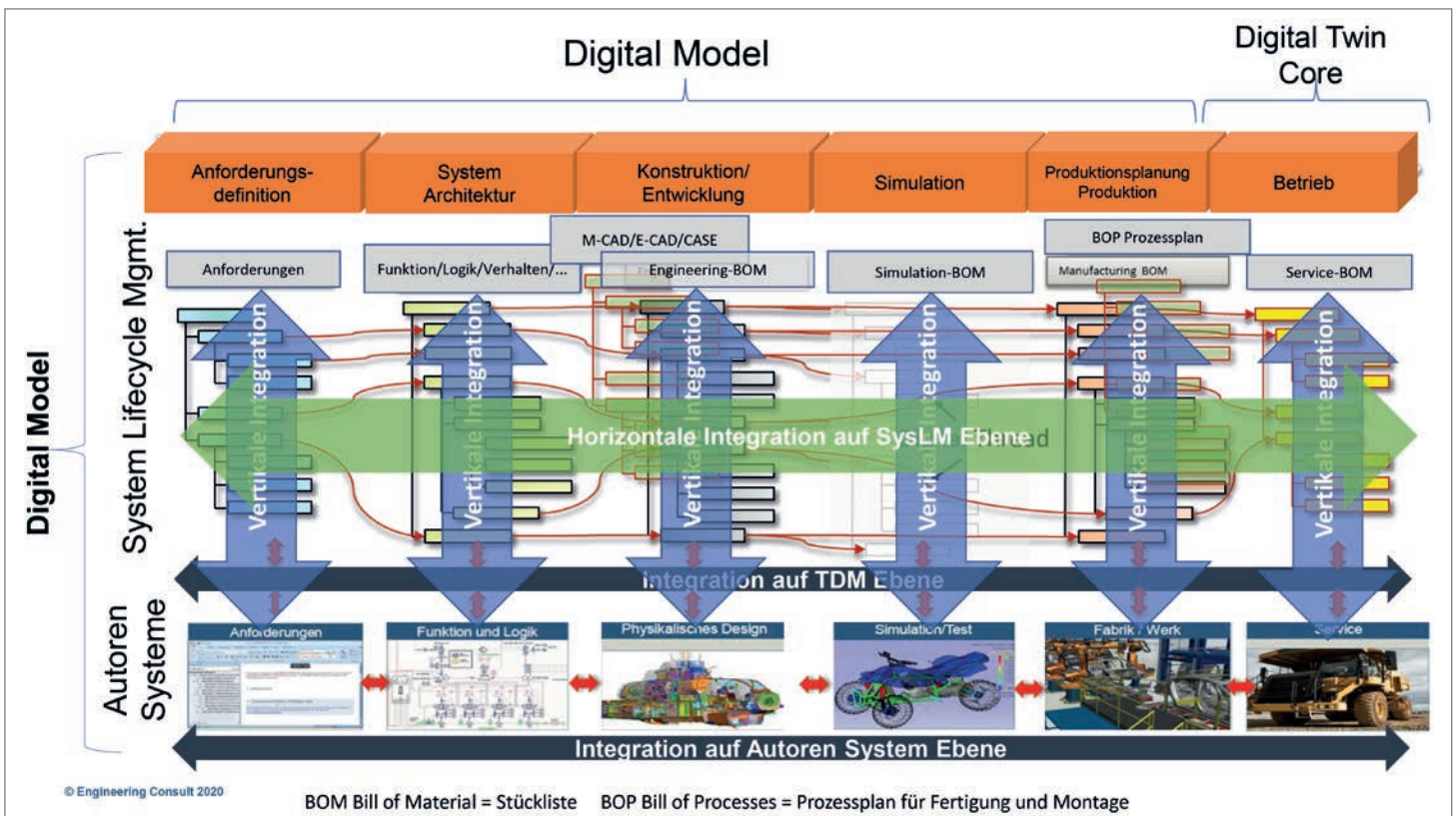


Abbildung 6: Horizontale und vertikale Integration der Engineering-Prozesse [1]

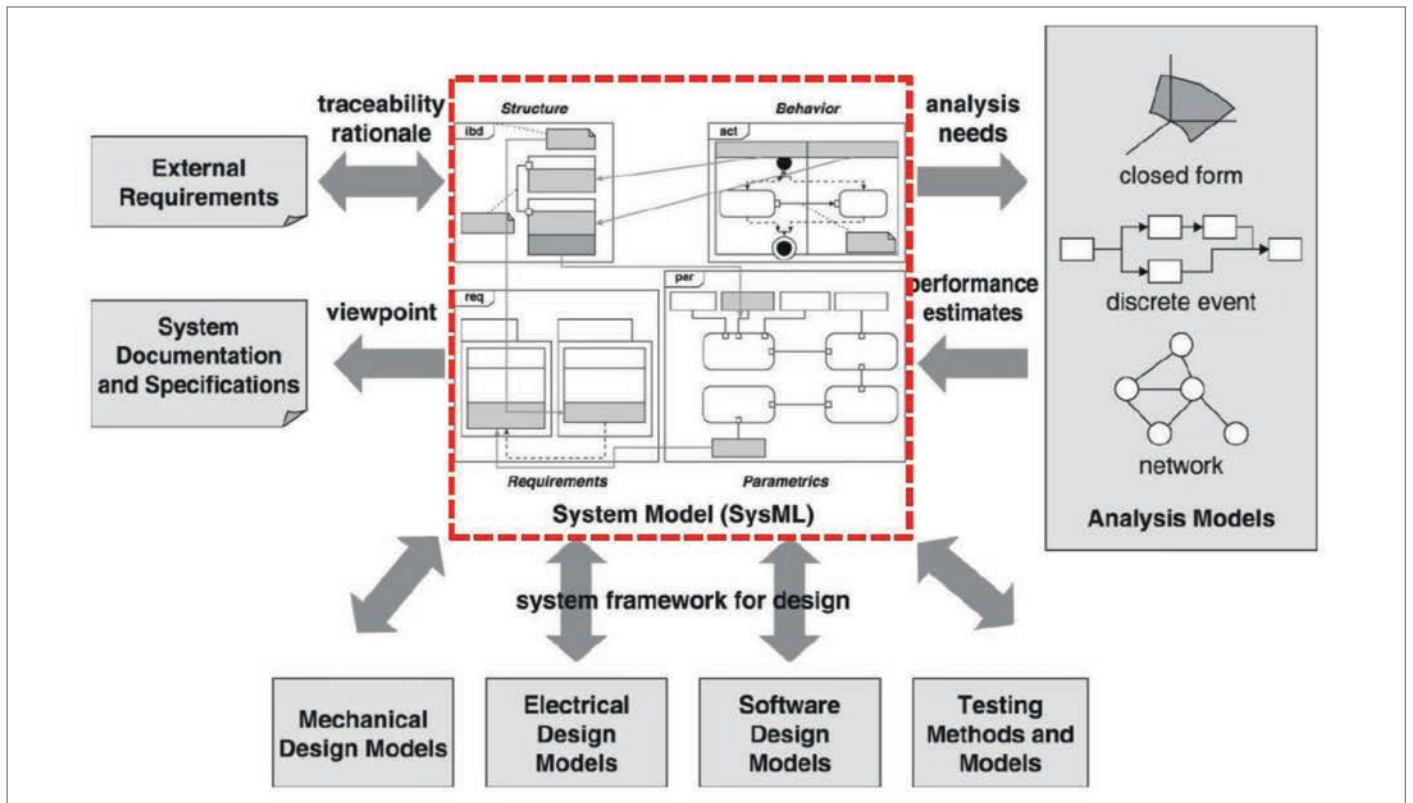


Abbildung 7: Bedeutung der Systemmodellierung für die Integration der Engineering-Prozesse [2]

rativen Lösungsansatz für das Engineering. Ein leichtgewichtiger und föderierter Engineering-Backbone wird die Rolle der Daten- und Prozessintegration über den gesamten Produktlebenszyklus inklusive des operativen Betriebes einnehmen. Dabei wird zwischen vertikaler und horizontaler Prozessintegration unterschieden (siehe Abbildung 6).

Unter vertikaler Integration soll hier entsprechend Abbildung 6 die Integration von Autorenssystemen über die TDM-Lösungen zu einem SysLM-Backbone und umgekehrt verstanden werden. Gerade die Potenziale in den ganz frühen Lebenszyklusphasen, also die Anforderungs- und Systemarchitekturmodellierung (→ Sys-

temmodellierung) (siehe Abbildung 7) inklusive der Anbindung an die administrative Ebene (→ SysLM), werden heute industriell noch wenig genutzt. Sie haben noch sehr starken akademischen und industriellen Forschungscharakter, sind aber wesentlich sowohl für die vertikale als auch für die horizontale Integration.

- Software Konfigurationsmanagement (SCM)**
 - Optimiert für das Management von Source Code, SW Modulen und lauffähigen Programmen
 - Revisionierung durch Major, Minor, Patch Revision (Semver.org)
 - Methoden: Branching, Tagging (Baseline), Merging
- Hardware (M/E/EE) Konfigurationsmanagement**
 - Basiert auf Stücklisten = Produktstrukturen (hierarchische Objekte) + zusätzliche Dokumente
 - Revisionierung durch Revision (=Änderungsindex) + ev. Iteration
 - Die konfigurierten Objekte haben einen Lebenszyklus und werden durch Revisionen/Versionen (D: Änderungsindex) gesteuert
 - Das Änderungsmanagement, dass zu einer Konfiguration führt besteht aus Regeln, z.B. für Kompatible und inkompatible Änderungen
- Architektur Konfigurationsmanagement (...und Prozess Plan und BPMN)**
 - Kombination von Hierarchie und Netzwerk
 - Intelligente Information: Ports, mit Energie-, Material-, Informationsfluss,
 - Revisionierung ?

© Engineering Consult 2020

Abbildung 8: Verschiedene Änderungsprozesse in verschiedenen Disziplinen

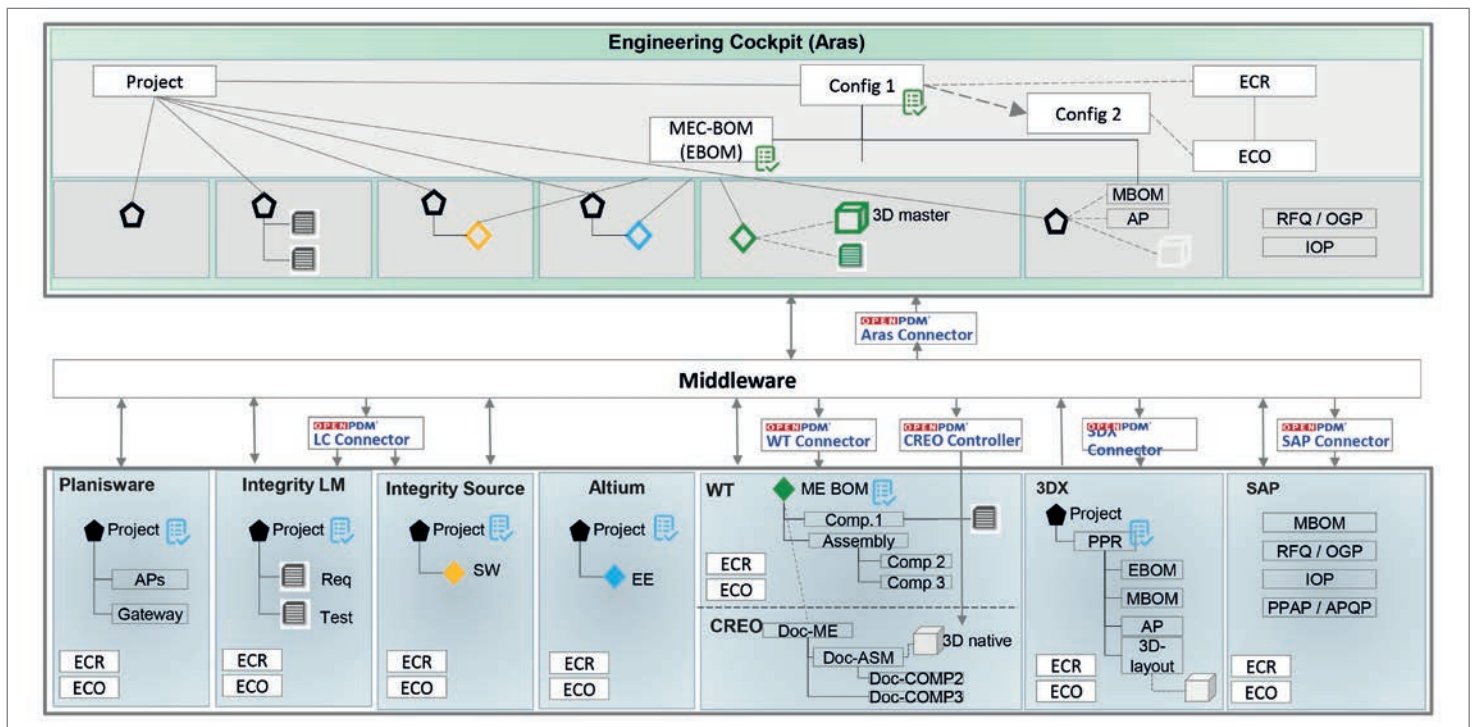


Abbildung 9: Konzept einer vertikalen und horizontalen Integration der Engineering-Prozesse [3]

Die Relevanz dieser Integration in der sehr frühen Phase des Produktlebenszyklus ist von Fachleuten anerkannt und beginnt, sich in den Bereichen High Tech, Medical, Automotive, Aerospace und Defense durchzusetzen. Dagegen ist die Integration von M-CAD- und E-CAD-Systemen in TDM und PLM/SysLM seit vielen Jahren Standard. Die heute noch eher wenig genutzten Bereiche der Integration sind die Software-Entwicklung und die Simulation. Die Simulation ist ebenso wie die Systemmodellierung zentraler Bestandteil einer betrieblichen V&V-Strategie (→ Validierung und Verifizierung). Die Integration in die Produktionsplanung und -steuerung (PPS = Produktionsplanung und -steuerung/MES = Manufacturing Execution System) ist weniger technisch eine Herausforderung, sondern eher politisch, da sich dort die Anwendungen von SysLM, PPS und MES stark überschneiden.

Die horizontale Integration wird wesentlich von SysLM orchestriert. Hier laufen die zentralen Prozesse des unternehmensweiten Freigabe-, Änderungs- und Konfigurationsmanagements, der Visualisierung und der Zusammenarbeit. Hier liegt die Schwierigkeit insbesondere im Aufbau einer unternehmensweiten technischen Organisation (→ Nummernsysteme, Revisionierung/Versionierung, Stücklisten, ...) und in der Harmonisierung der Disziplin-spezifischen Änderungsprozesse (siehe Abbildung 8).

Aber auch auf der Autoren- und TDM-Ebene laufen Teilprozesse der Integration, zum Beispiel Kopplung M-CAD/E-CAD zur Bauraumuntersuchung oder Übergabe von Fertigungs- und Montagedaten aus E-CAD. Zum Abschluss zeigt *Abbildung 9* ein Konzept für eine vertikale und horizontale Integration der Engineering-Prozesse, das die vorgestellten Ideen zur Digitalisierung auf der Basis des MBSE-Ansatzes und eines Engineering-Backbones gekoppelt über redundanzfreie REST-Kopplung zusammenfassend darstellt.

Sie werden jetzt sicher als Leser denken: „Aha, wie bei PLM wieder viele Visionen, aber wo sind die Lösungen?“ Sie haben im Jahr 2020 sicher Recht, doch Visionen waren schon immer der Ansporn für Ingenieure, diese auch umzusetzen. Wir stehen erst am Anfang einer weiteren Evolutionsstufe, die allerdings disruptiver sein wird, als alle vorherigen. Durch die starke Verknüpfung von SysLM mit neuen Geschäftsmodellen und der betrieblichen Digitalisierungsstrategie wird SysLM sich nicht nur auf die rein technische Ebene reduzieren lassen. Als Teil dieser Umsetzung sind die teilhabenden Menschen, Organisationen, Geschäfts- und Engineering-Prozesse sowie die Wertschöpfungsketten betroffen.

Quellen

[1] Eigner, M.: System Lifecycle Management - Strategische Komponente von

Engineering 4.0, 2020, Springer Berlin, Heidelberg (erscheint E 2020)

[2] Friedenthal, S.; Greigo, R.; Mark S.; INCOSE MBSE Roadmap, in "INCOSE Model Based Systems Engineering (MBSE) Workshop Outbrief" (Presentation Slides), presented at INCOSE International Workshop 2008, Albuquerque, NM, pg. 6, Jan. 26, 2008.

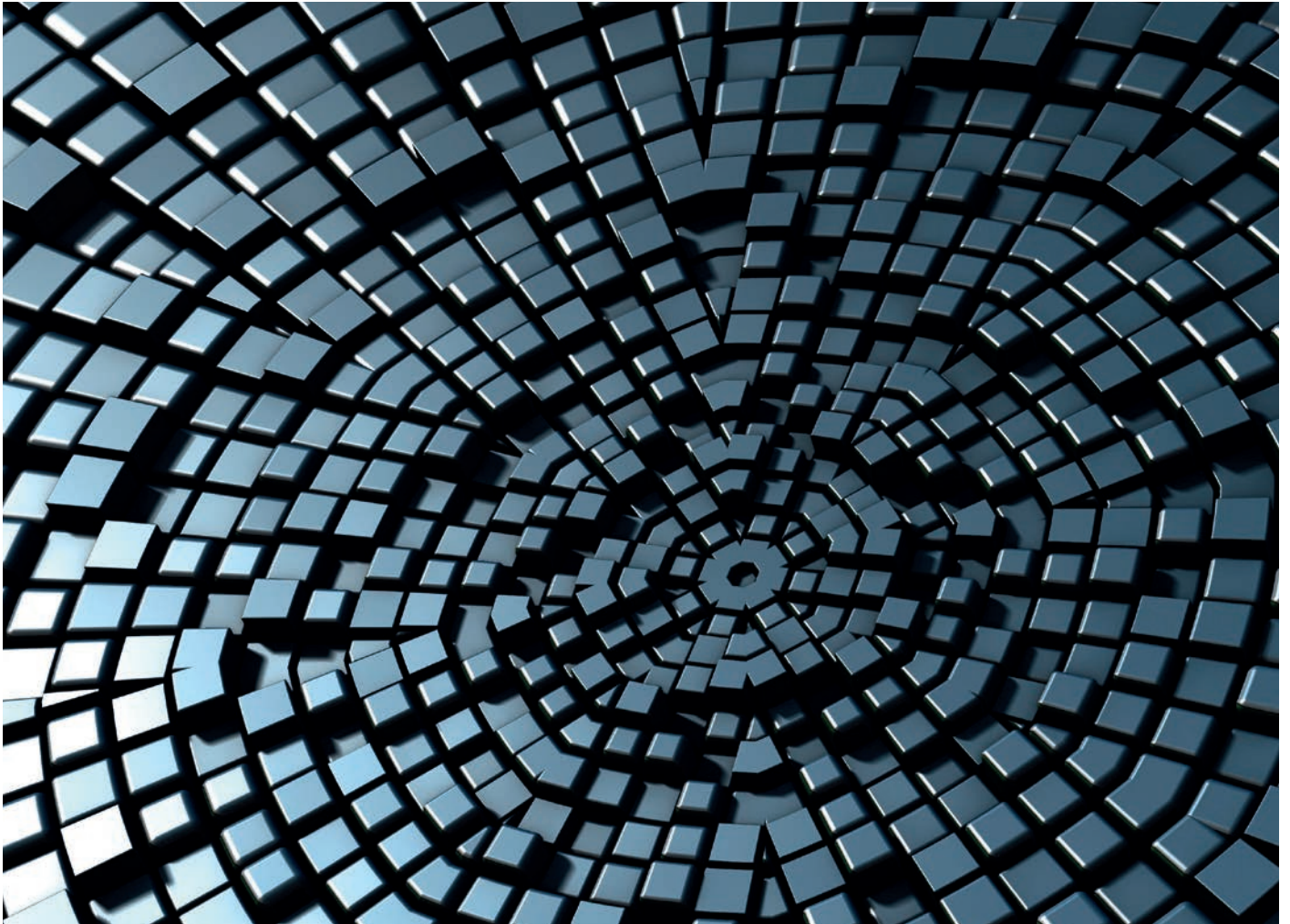
[3] Eigner, M., Stürken, M.: Integrated model-based systems: Engineering solution to support digital transformation, ProSTEP Symposium, Stuttgart, 2019



Martin Eigner

eigner.engineeringconsult@gmail.com

Nach dem Abschluss an der Universität Karlsruhe im Jahr 1980 war Martin Eigner Leiter der Technischen Datenverarbeitung und Organisation sowie der elektronischen Vorentwicklung in einem Geschäftsbereich der Robert Bosch GmbH. 1985 gründete er die weltweit erste PLM-Firma EIGNER + PARTNER AG, die 2001 mit Agile fusionierte und 2007 an ORACLE verkauft wurde. 2003 gründete er die EIGNER Engineering Consult, ein Beratungsunternehmen zur Optimierung von Engineering-Prozessen mit den Schwerpunkten PLM, MBSE und Digitalisierung. Martin Eigner war von 10/2003 bis 10/2017 Leiter des Lehrstuhls für Virtuelle Produktentwicklung an der Universität Kaiserslautern.



Das Blockchain-Manifest

Verfasst von Falk Wolsky, 01/2020, aktualisiert 10/2020

Während Blockchain an sich noch immer zu großen Teilen ein Buzzword ist, fehlt es an guten Produktideen mit dieser spannenden, neuen Technologie. Um den Blickwinkel von der Technologie auf die Nutzung in Form von nützlichen Produkten zu wechseln, habe ich dieses Manifest geschrieben.

Nähert man sich dem Phänomen „Blockchain“, so kann man zunächst eine soziologische Prägung erkennen: „Die junge Generation leidet unter dem Fehlen der Kernfamilie und lehnt daher zentrale Institutionen ab.“ Seit Tausenden von Jahren gibt es Verträge zwischen Menschen über Werte. Durch die Ablehnung der zentralen Institutionen und die Anwendung moderner kryptografischer und verteilter Datenbanktechnologien

schafft diese junge Generation Rahmenbedingungen, um den Vertrauens-Bund zwischen Menschen zu ändern.

Mit dem Aufkommen von PCs in den 1980er Jahren versteht damals nur eine Gemeinschaft junger Hacker den Wert von Computern zu Hause. Und wie eine Rückkehr der Geschichte haben wir heute eine nicht allzugroße Gemeinschaft junger Entwickler, die eine Revolution des „externa-

lisierten Vertrauens“ zwischen Menschen weltweit vorantreibt.

Derzeit befinden wir uns mit Blockchains am Ende des „Tales der Enttäuschung“ im Gartner-Hype-Zyklus und vermissen immer noch die Killer-App für Blockchain, da die „alten Jungs“ in der IT oft zu Recht ausdrücken, dass man für die jeweilige vorgeschlagene Lösung nicht zwangsweise eine Blockchain benötigt.

Dieses Manifest beschreibt die Motivation, die Kernideen und die Vision hinter der Idee, eine Blockchain zu entwickeln und zu verwenden.

Die Entwicklung von Gesellschaft und IT heizt sich auf und wird jeden Tag schneller. In dem Manifest werden wir Mega-Trends, Kunden- und Gesellschaftstrends, aber auch IT- und vor allem Blockchain-Trends einbeziehen.

Relevante Mega-, Kunden- und Gesellschaftstrends

Algorithmisierung – Verwendung mathematischer, statistischer und künstlicher Intelligenzalgorithmien in Software zur Optimierung von Prozessen, Erkenntnissen und der Verwendung von Daten für jeden Lebensaspekt.

Individualisierung – Die Marketingwelt schafft Bedürfnisse, die immer individueller auf Kundenprofile zugeschnitten sind, bis hin zur 1-zu-1-Individualisierung (Segmentation to one). Dies gilt sowohl für digitale als auch für physische Produkte.

Attention-Ökonomie – Wenn früher das Paradigma „Ich denke – also bin ich“ galt, so wechseln wir inzwischen zu „Jemand hat an mich gedacht – also bin ich“. Die Aufmerksamkeitsökonomie ist eine Zusammenfassung von Veränderungen des psychologischen Verhaltens in individuellen und gesellschaftlichen Aspekten, die durch die Mechanismen von sozialen Netzwerken beschleunigt wird.

Bedürfnis nach Privatsphäre – Da immer mehr Daten von jedem Einzelnen verfügbar sind, wird mehr Privatsphäre gefordert. Datenverletzungen und Manipulation von Personen formen ein dystopisches Bild. Zugleich ist die gefühlte und gelebte Forderung nach Privatsphäre gänzlich außer Balance. Wir rufen zwar nach Datensicherheit und Privatsphäre, tun dies aber auf Plattformen wie Facebook, YouTube oder TikTok.

Mensch-Maschine-Dialog – Es wird eine Verlagerung von Benutzereingaben und der Verwendung von Computern stattfinden – von Stiften, Tastaturen und digitalen Formularen hin zu sprachbasierter Kommunikation oder Kommunikation über Gehirnverbindungen.

Trends in der IT

Virtuelle und erweiterte Realität – Das führt zu einer veränderten Nutzung von IT-Systemen und -Inhalten auf beiden Seiten: B2B und Back-End sowie für B2C auf Kundenseite. Es ist normal, dass wir auf Displays schauen. Das sind zweidimensionale Flächen. Zwar kann man dreidimensionale Inhalte darstellen, aber nicht immersiv in diese „eintauchen“. Virtual-Reality- und Augmented-Reality-Anwendungen nehmen an Fahrt auf und bald wird es zum Beispiel normal sein, ein Computer-Programm in einer dreidimensionalen Umgebung zu gestalten, statt in einer zweidimensionalen Entwicklungsumgebung (IDE integrated development environment) mit Code-Zeilen. Auch auf der Kundenseite wird es als normale Realität wahrgenommen werden, dass immersive Produkterlebnisse angeboten werden.

Künstliche Intelligenz – Das Bedürfnis nach Just-in-Time-Erkenntnissen prägt den Markt für In-Memory-Daten und schnelle Datenzugriffslösungen. Gepaart mit Mustererkennung, Algorithmen und künstlicher Intelligenz wird dies zu großen Veränderungen auf dem IT-Markt führen. Die Gewinner fügen dem Kern ihrer Produkte Datenströme und die Aufwertung der Daten mittels künstlicher Intelligenz hinzu. Auf diese Weise entstehen neue Services, Produkte und neuer Nutzen.

Internet der Dinge – Von der Logistik über das Lager und die Produktion bis zum Haus des Endkunden: Sensoren und Zeitreihendaten werden überall sein. Die Nutzung dieser Daten wird verschiedenste neue Kundendienstleistungen ermöglichen.

Hochsprachen – Von Assembler über C/C++ und Java bis hin zu JavaScript, von CSV zu XML und JSON: Höhere Layer sparen Zeit und versprechen einfaches Lernen sowie schnelles Prototyping. Man sollte trotz der Vorteile die Sicherheitsrisiken der Paketierung und des Fehlens formeller Leitfäden nicht vergessen.

Blockchain – Die Blockchain bietet digitale Verträge und verteilte Ledger. Zwei Zutaten, die die IT und die Gesellschaft in den nächsten Jahrzehnten grundlegend prägen werden. Die Gewinner werden Transaktionen außerhalb der Blockchain-Nodes bereitstellen, um die Geschwindigkeit zu er-

höhen; ebenso ein sehr gutes statistisches/analytisches Framework und eine gute Konnektivität zu realen Diensten sowie eine einfache Verwendung für „Nicht-Entwickler“.

Wir glauben

Menschen wie Sie und ich, die die Blockchain in der Zukunft verwenden, müssen in der Lage sein, auf einfache, App-ähnliche Weise mit den digitalen Verträgen und den digitalisierten Assets zu interagieren. Die Verlagerung von „Entwicklerwissen zum Umgang mit einer Blockchain“ hin zu leicht verständlichen Benutzeroberflächen ist ein Schlüsselement für den allgemeinen Erfolg einer Blockchain.

Was getan werden sollte: Es scheint wichtig, in Konzeptions- und Interaktionsdesign zu investieren, um die Kernobjekte einer Blockchain greifbar zu machen, etwa um einen visuell besseren Ausdruck für einen Smart Contract, einen Token oder eine Wallet zu finden. Ebenso für die Prozesse und Funktionen eines bereitgestellten, aktiven Smart Contract. Nur durch eine einfache visuelle Sprache – die auch für Nicht-Entwickler verständlich ist – kann eine Blockchain eine breitere Verwendung finden.

Klassische Papierverträge, digitale Verträge und intelligente Verträge

Was getan werden sollte: Durch einen AI/OCR- und Dokumentenmanagement-gesteuerten Ansatz mit Blockchain-basierten Services (Hashing von Dokumenten, Ursprungsnachweis etc.) sollte eine Brücke zum Handling klassischer Dokumente und zu Verträgen ermöglicht werden. Außerdem sollten Lösungen wie DocuSign integriert werden. Dies wird Kunden helfen, ihr klassisches Handling von Dokumenten nicht direkt zu einem Blockchain-basierten Handling zu „revolutionieren“, sondern in realistischem Tempo zu „evolutionieren“.

Für Millennials und ihre Nachfolger – aufgewachsen vor allem mit digital-dialogbasierter Kommunikation – braucht es eine moderne und einfache Art und Weise der dialogbasierten Interaktion mit Blockchains.

Was getan werden sollte: Social Media Messenger und sprachbasierte Assistenten wie Alexa oder Siri forcieren ein spezielles Nutzungsverhalten. So wie jeder bereits erwar-

tet, „kostenloses WLAN“ zu bekommen, werden die Erwartungen der zukünftigen Benutzer einer Blockchain auf beiden Seiten (B2B und B2C) eine dialogbasierte Interaktion beinhalten. Dialoge im Sinne von Chats für Wallet und Vertragsaktionen oder Voice-basierte Interfaces werden der Standard sein. Mit anderen Worten, Benutzer einer Blockchain beginnen die tägliche Arbeit manchmal mit der einfachen Frage: „Hallo Blockchain, öffnen Sie die Vertragsstatistik von gestern für mich.“

Direkt nutzbare Erkenntnisse und mögliche Interaktionen mit einer Blockchain sollten durch Mechanismen der künstlichen Intelligenz unterstützt werden.

Was getan werden sollte: Die Blockchain der Zukunft muss Hook-Points an Daten und Prozessen haben, um sie zu verknüpfen – mit Mechanismen der künstlichen Intelligenz. Es beginnt mit der Musteranalyse in den historischen Daten, mit Smart Wallets und wirklich „intelligenten“ Smart Contracts, um Entscheidungen in Geschäfts- und Anwendungsprozessen zu treffen. Diese künstliche Intelligenz kann durch Benutzerinteraktion genutzt werden, sich aber auch geschickt in Bedienoberflächen integrieren. Beispiel: eine intelligente Geldbörse mit Ausgabenverwaltung.

Distributed Ledger und IoT werden in Kombination bedeutend. Geräte arbeiten basierend auf Smart Contracts.

Was getan werden sollte: An jedem geeigneten Punkt (Smart Contract, Wallet, Analysis) müssen Hook-Points für sogenannte Digital Twins vorhanden sein. Dadurch lässt sich Blockchain-Technologie für vordefinierte IoT-Geschäftsprozesse und automatisch abarbeitbare Verträge verwenden.

Mashup ist ein älterer Begriff, wird aber bedeutsam, sobald eine Blockchain oder App keine „Integrationen“ zu angesagten Diensten wie Dropbox, Mailchimp usw. hat. Dadurch schrumpft die Akzeptanz einer Blockchain.

Was getan werden sollte: Man sollte wichtige Dienste und Plattformen nach Priorität der Nutzung integrieren. Sie bereichern nicht

nur das Serviceangebot für das eigene Produkt der Blockchain, sondern senken auch die Barriere für Nutzungsentscheidungen.

Prinzip „Verständlichkeit zuerst“

Die beste Multiplikationsstrategie basiert auf einer guten Benutzerakzeptanz und einem starken Netzwerk von Lösungspartnern. Jede Komponente und App, die entwickelt wird, muss nicht nur von den Entwicklern und ihrer Community, sondern auch von Produktmanagement-Teams überprüft und dokumentiert werden. Beschreibungen, die für „jeden“ gleichermaßen verständlich sind – auch wenn es sich nicht um einen Entwickler handelt –, müssen die Normalität sein.

Prinzip „Digital Society Solutions“

Estland erhält mittlerweile international große Aufmerksamkeit, da fast alle staatlichen Dienste digital online verfügbar und über einen Chipkarten-basierten, einfachen Sicherheitsmechanismus zugänglich sind. Am Horizont erscheint eine digitale Gesellschaft. Digitale Nomaden sind nur ein Teil davon.

Diese digitale Gesellschaft braucht aber auch klassische Dienste. Und bei diesen Diensten dreht sich alles um Bezahlung, Wertverfolgung, Eigentumsnachweis, Abstimmung und so weiter. Eine moderne Blockchain muss Minimal Valuable Prototypes (MVPs) und Lösungen für eine solche digitale Gesellschaft liefern, beispielsweise für

- Aufzeichnungen über Eigentum (zum Beispiel über Grundstücke und Immobilien)
- Aufzeichnungen über Zertifikate (Führerschein, Wohnsitz)
- Mietverträge für Wohnungen, Autos etc.
- Zahlung von öffentlichen und geschäftlichen Dienstleistungen und Produkten
- Automatische Steuerberechnung und Zahlung
- Abstimmung für Infrastrukturprojekte
- Crowd Investing für Infrastrukturprojekte
- Geld verleihen
- Charity

Resümee

Ändert man den Blickwinkel von der Technologie hin zu Nutzern, die Produkte, gute

Interaktion und Werthaltigkeit erwarten, dann erkennt man, dass wir beim Thema Blockchain noch einen weiten Weg vor uns haben.

Das Digital Banking öffnet den Weg zum Grundverständnis einer „Wallet“, die sich doch ganz erheblich von einem klassischen Konto mit Auszügen am Kontoauszugsdrucker unterscheidet. Einfach zu bedienende Wallets, ein gutes Ökosystem für Entwickler und gute Produktideen sind der Weg, auf dem Blockchain-Technologien erfolgreich werden.



Falk Wolsky
falk.wolsky@gmx.at

Falk Wolsky ist für Unternehmen in den Bereichen Professional Innovation & Product Development tätig. Er unterstützt und entwickelt Projekte als CTO und Chief Innovation Officer international unter anderem in den Themen AI & ML, IoT, Big Data und Blockchain. Für große Unternehmen begleitet Falk Wolsky ganzheitlich Projekte für die digitale Transformation. Daneben ist er Autor sowie Sprecher und Dozent für Konferenzen und Universitäten.

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- Fredo David
- Henrik Bachmann
- Gertrud Abram
- Friedhelm Gibbels
- Michael Vinnemeier
- Aleexander Kornbrust

Firmenmitglieder DOAG

- Amprion GmbH,
Mathias Körbs
- Evotec SE,
Rachid Chaib



Termine

Januar

01

08.01.2021

DOAG Datenbank Online-Event: Durch die Dornen zu den Sternen: Migration von Oracle nach Postgres

Borys Neselovskyi
Online

21.01.2021

**DOAG Online-Event:
DOAG CC Support**

Online

26.01.2021

**Netsuite Users Days
(Online-Veranstaltungsreihe)**

Online

27.01.2021

**DOAG Development Online-Event:
Polymorphic Table Functions**

Online

Februar

02

02.02.2021

**Netsuite Users Days
(Online-Veranstaltungsreihe)**

Online

09.02.2021

Regionaltreffen Bremen

Ralf Kölling
Bremen

09.02.2021

**Netsuite Users Days
(Online-Veranstaltungsreihe)**

Online

16.02.2021

**Netsuite Users Days
(Online-Veranstaltungsreihe)**

Online

24.02.2021

**DOAG Development Online-Event:
EBR oder Exception Handling**

Online

März

03

10.03.2021

Java aktuell 2/2021
Erscheinungstermin für Abonnenten

10.03.2021

Red Stack Magazin 2/2021
Erscheinungstermin für Abonnenten

16. - 17.03.2021

JavaLand 2021
Hybride Veranstaltung On-Site und Online

19.03.2021

Java aktuell 2/2021
Erscheinungstermin am Kiosk

Impressum

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

Red Stack Magazin inkl. Business News ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin inkl. Business News wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Björn Bröhl. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führen einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
ViSdP: Mylène Diacquenod
Redaktionsleitung Red Stack Magazin:
Martin Meyer
Redaktionsleitung Business News:
Marcos López
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Johannes Ahrends, Niels de Bruijn, Martin Eigner, Christina Fink, Dr. Holger Friedrich, Thimo Fußbroich, Randolph Geist, Beda Hammerschmidt, Jonas Huck, Thomas Kaden, Ekaterina Koshkarova, Marcos López, Andre Lünsmann, Martin Meyer, Dietmar Neugebauer, Franck Pachot, Sinan Petrus Toma, Evgenia Rosa, Tim Rüb, Alfred Schlaucher, Detlef Schröder, Klaus Schuermann, Hervé Schweitzer, Ingo Sobik-Weniger, Dennis Walz, Hans-Peter Weih, Daniel Westermann, Falk Wolsky

Titel, Gestaltung und Satz:

Alexander Kermas
DOAG Dienstleistungen GmbH
(Anschrift s.o.)

Fotonachweis:

Titel: © frannyanne | www.123rf.com
S. 02: © Vidoslava | <https://stock.adobe.com>
S. 05: © Martin Meyer | www.doag.org
S. 07: © alphaspirt | www.123rf.com
S. 10: © Paurielaki | www.123rf.com
S. 16: © Aleksandr Davydov | www.123rf.com
S. 20: © kzenon | www.123rf.com
S. 25: © Olivier Le Moal | www.123rf.com
S. 29: © tashatuvango | www.123rf.com
S. 35: © Zackery Blanton | www.123rf.com
S. 44: © stylephotographs | www.123rf.com
S. 46: © conceptw | www.123rf.com
S. 48: © Christos Georghiou | <https://stock.adobe.com>
S. 53: © wrightstudio | www.123rf.com
S. 58: © Khoon Lay Gan | <https://stock.adobe.com>

S. 64: © vikayatskina | www.123rf.com

S. 67: © Valentyn Akhnovskyi | www.123rf.com

Titel S. 74: © thedigitalartist | www.pixabay.com

S. 81: © geralt | www.pixabay.com

S. 85: © Painter06 | www.pixabay.com

S. 91: © geralt | www.pixabay.com

S. 98: © TheDigitalArtist | www.pixabay.com

S. 101: © gmast3r | www.123rf.com

Anzeigen:

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

Druck:

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

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags.

Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

B4Bmedia.net AG https://e-3.de	U 3	MuniQsoft Consulting GmbH www.muniqsoft-consulting.de	S. 15	PROMATIS software GmbH www.promatis.de	S. 89
DOAG e.V. www.doag.org	U 2, U 4	MuniQsoft Training GmbH www.muniqsoft-training.de	S. 3		



Das E-3 Magazin

Information und Bildungsarbeit von und für die SAP-Community

**Wir waren zwar nicht die Ersten
auf dem Mond,
dafür sind wir die Ersten,
die unabhängig
über SAP® berichten.**



JavaLand

16. - 18. März 2021
in Brühl bei Köln

Programm jetzt
online

Hybride Veranstaltung

Was die JavaLand als Plattform für Wissenstransfer und Networking ausmacht, kannst du vor Ort im Phantasialand oder online erleben. Als Teilnehmer entscheidest du selbst!

