DWH-Modernisierung mithilfe eines Data Lake – die verschiedenen Umsetzungsmöglichkeiten in der Praxis

Fabian Hardt, OPITZ CONSULTING Deutschland GmbH

Inmitten von Digitalisierung und Industrie 4.0 haben sich die Anforderungen für die Speicherung und die anschließende Analyse an klassische, seit vielen Jahren etablierte Data-Warehouse-Systeme maßgeblich geändert. Bleibt dennoch der klassische Ansatz weiterhin das Mittel der Wahl oder sind Unternehmen gezwungen, die bestehende Data-Warehouse-Architektur zu modernisieren oder langfristig sogar zu ersetzen? Der Artikel geht dieser Frage nach und beleuchtet die jeweiligen Vor- und Nachteile anhand eines Praxisbeispiels.

Für Unternehmen kann es im Vorfeld einer Modernisierungsentscheidung hilfreich sein, sich die Möglichkeiten einmal genauer anzusehen. Wie sollte etwa eine Modernisierung aus technischer Sicht aufgebaut sein, um die stetig steigenden Anforderungen weiterhin angemessen erfüllen zu können? Im weiteren Verlauf werden einige Beispiel-Architekturen vorgestellt und deren Vor- und Nachteile betrachtet. Dabei wird deutlich, welche verschiedenen Möglichkeiten es bei der Modernisierung in der Praxis gibt und in welcher Konstellation Data Lakes zu Kosteneinsparungen in einer bestehenden Systemlandschaft führen können. Bevor es um die Frage geht, inwiefern ein Data Lake ein klassisches Oracle Data Warehouse (DWH) ersetzen oder vielleicht eher eine perfekte Ergänzung darstellen kann, ist es wichtig, beide Systeme klar voneinander abzugrenzen.

Mit Data Lakes große Datenmengen flexibel verarbeiten

Von einer hohen Abstraktionsebene aus betrachtet, handelt es sich bei einem Data Lake um eine alternative Datenspeicherungsmethode, die mithilfe sogenannter „Big-Data-Frameworks" arbeitet. In der Praxis kommen in einem Data Lake häufig Hadoop-Systeme zum Einsatz, kombiniert mit Software-Tools wie Apache Spark und Apache Kafka sowie mit einer Lambda-Architektur, um Daten auch in Echtzeit in angemessener Latenz verarbeiten zu können. Ein Data Lake erfüllt damit deutlich besser die typischen Big-Data-Anforderungen als ein klassisches DWH, da er aufgrund der hohen Skalierbarkeit deutlich besser für die Verarbeitung großer Datenmengen geeignet ist. Zudem ist ein Data Lake deutlich flexibler, was die Integration neuer Datenstrukturen und -typen angeht. Das ist dem „Schema on Read"-Paradigma geschuldet, auf das im nachfolgenden Abschnitt näher eingegangen wird.

„Schema on Read" vs. „Schema on Write"

Ein klassisches DWH beziehungsweise eine relationale Datenbank folgt dem sogenannten „Schema on Write". Hier ist das Zielschema im DWH vorgegeben und die Daten werden mit den Schritten „Extrahieren, Transformieren, Laden" (ETL) in die Form dieses Schemas gebracht. Das vorliegende Datenmodell und die darunterliegenden Strukturen orientieren sich vor allem an den Anforderungen der Nutzer. Der Vorteil dieses Paradigmas ist die automatisch hohe Datenqualität, die festgelegte Tabellenstrukturen und technische Hilfsmittel wie Constraints gewährleisten. Daten, die im Aufbau von diesen Strukturen abweichen, können nicht eingefügt werden. Der Nachteil: Das Vorgehen bei diesem Schema-Typ ist wenig agil. Bei jeder Veränderung der Quelldaten ist eine Anpassung der gesamten ETL-Strecke nötig.

Ein Data Lake hingegen folgt dem sogenannten „Schema on Read"-Paradigma. Hier erfolgt die Schematisierung erst beim Auslesen der Daten. Die Quelldaten liegen ohne Datenverlust vor und können zu einem späteren Zeitpunkt verarbeitet werden. Die Reihenfolge wäre jetzt also „Extrahieren, Laden, Transformieren" (ELT). In der Folge besteht bei Änderungen an den Quellsystemen keine direkte Abhängigkeit. Somit droht kein Datenverlust, wenn die Struktur der Quelldaten von der Struktur der Zieldaten abweicht. Lediglich bei der nachfolgenden Verarbeitung muss die Logik entsprechend angepasst werden, damit alle Daten ohne großen Aufwand nachträglich verarbeitet werden können. Ein Nachteil dieser „Schemalosigkeit" ist die fehlende Prüfung auf Datenintegrität. Nicht gefüllte Attribute oder falsche Datentypen in einer Spalte werden erst beim Auslesen der Daten erkannt und sind somit in der Ladelogik entsprechend zu berücksichtigen [1].

Mithilfe des „Schema on Read"-Vorgehens ist ein Erkenntnisgewinn aus den vorliegenden Daten jederzeit möglich, auch wenn zunächst keine Zusammenhänge vermutet wurden [2]. In der Praxis macht ist eine Kombination aus „Schema on Read" und „Schema on Write" sinnvoll. Das Resultat sind Architekturen, wie sie in *Abbildung 1* dargestellt sind. Es gibt einen „Rohdaten-Bereich" (Raw Data), in dem die Quelldaten zunächst im Ausgangszustand abgelegt werden. Parallel dazu wird eine „Data Refinery" betrieben. Diese dient als Preprocessing Area, ähnlich der Staging Area im DWH. Als letzter Baustein ist der „qualitätsgesicherte Bereich" (Refined Data) zu sehen. Dieser ist essenziell; er wird gesondert kontrolliert und verwaltet. In diesen Bereich gelangen nur Daten, die durch Data Cleansing – teilweise in Echtzeit – aufbereitet wurden. Häufig wird dieser Bereich auch „Data Reservoir" genannt und als separater Datenbereich betrachtet [3].

Damit ein Data Lake zu einem Data Reservoir wird, sind folgende Dinge zu beachten:

- Es ist ein Katalog erforderlich, der den Inhalt des Data Lake klassifiziert und Metadaten zu den Objekten bereitstellt.

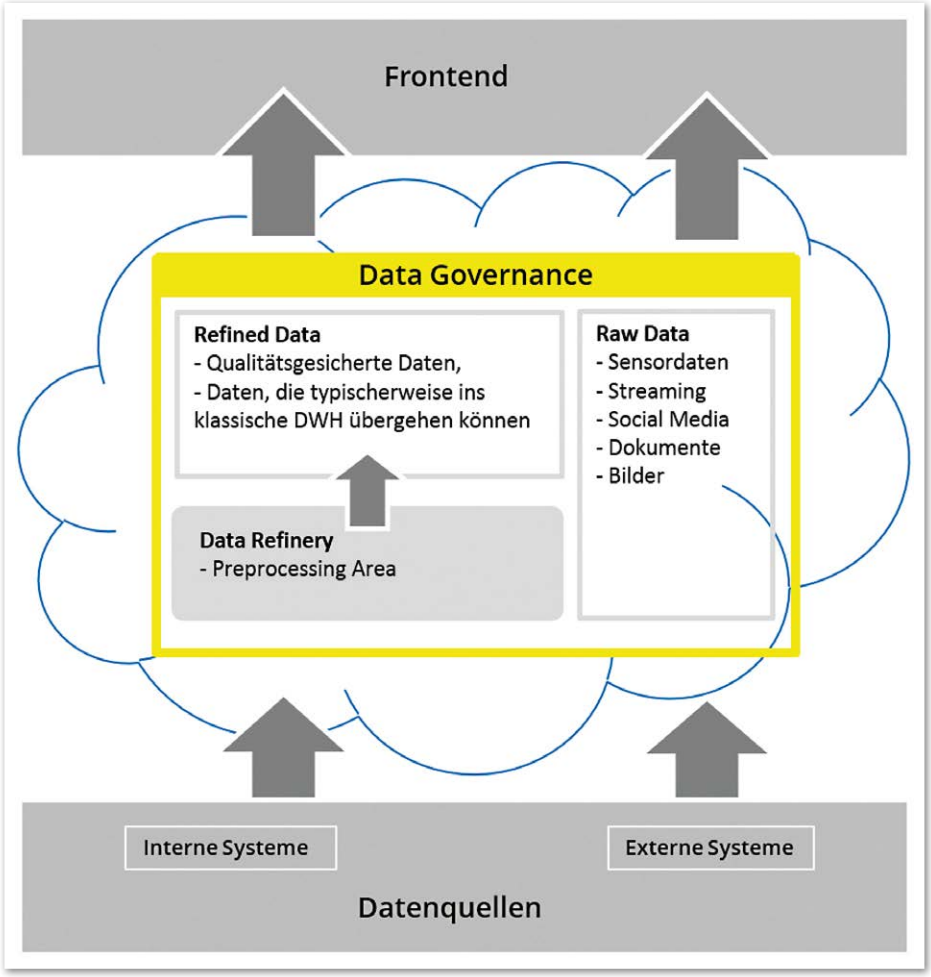

Abbildung 1: Data-Lake-Detailansicht

- Es gibt innerhalb des Reservoirs ein Berechtigungskonzept, analog zum DWH. Auch hier darf die Fachabteilung nur ihre eigenen Daten einsehen. Innerhalb des Reservoirs ist sowohl die zeitliche als auch die inhaltliche Konsistenz sichergestellt.
- Es besteht die Möglichkeit, ein Data Reservoir fachlich orientiert anzulegen; dies geschieht wie bei fachlich angelegten Data Marts im DWH. Diese sind für einen speziellen Zweck implementiert und bilden zum Beispiel einen Kernprozess des Unternehmens ab.

Das hybride Architektur-Szenario

Unter einer hybriden Architektur versteht man eine parallele Datenverarbeitung in einem Data Lake oder DWH *(siehe Abbildung 2)*. Hierbei wird die Kernkompetenz des jeweiligen Systems zu einem Maximum ausgeschöpft. Der hybride Ansatz nutzt somit die Vorteile beider Architekturen und ermöglicht auf diese Weise schnellere Innovationszyklen. Trotzdem lassen sich das DWH und sein teurer Speicher verkleinern, indem man einige Daten wie zum Beispiel unverdichtete Massendaten in den Data Lake verschiebt. Das bestehende DWH muss bei diesem Ansatz nicht direkt architektonisch verändert werden; alle Systeme für die Datenbeladung, Auswertungen und betriebliches Berichtswesen können weiterhin existieren.

Ein klassisches Beispiel wäre an dieser Stelle das Exportieren von berechneten KPIs, also sogenannter „Faktendaten". Diese sind im DWH-Prozess bereits vom Fachbereich abgenommen und somit diversen Qualitätssicherungsmaßnahmen unterzogen worden. Mithilfe eines Offloading-Verfahrens stehen diese Kennzahlen auch in einem Data Lake zur Verfügung.

Offloading als Kernelement der DWH-Modernisierung

Das Offloading-Vorgehen gewinnt im Zuge der Digitalisierung zunehmend an Bedeutung, weil an digitalen Geschäftsprozessen stets verschiedenste IT-Systeme beteiligt sind. Es wird künftig also nicht mehr ausreichen, ein DWH als die „Single Source of Truth" zu betrachten, zumindest nicht, solange es nicht eine umfassende Modernisierung erfahren hat. Im Zusammenspiel mit einem Data Lake kann das DWH befähigt werden, auch Informationen aus neuartigen Datenquellen zu verarbeiten und diese Informationen ganzheitlich zu ver-

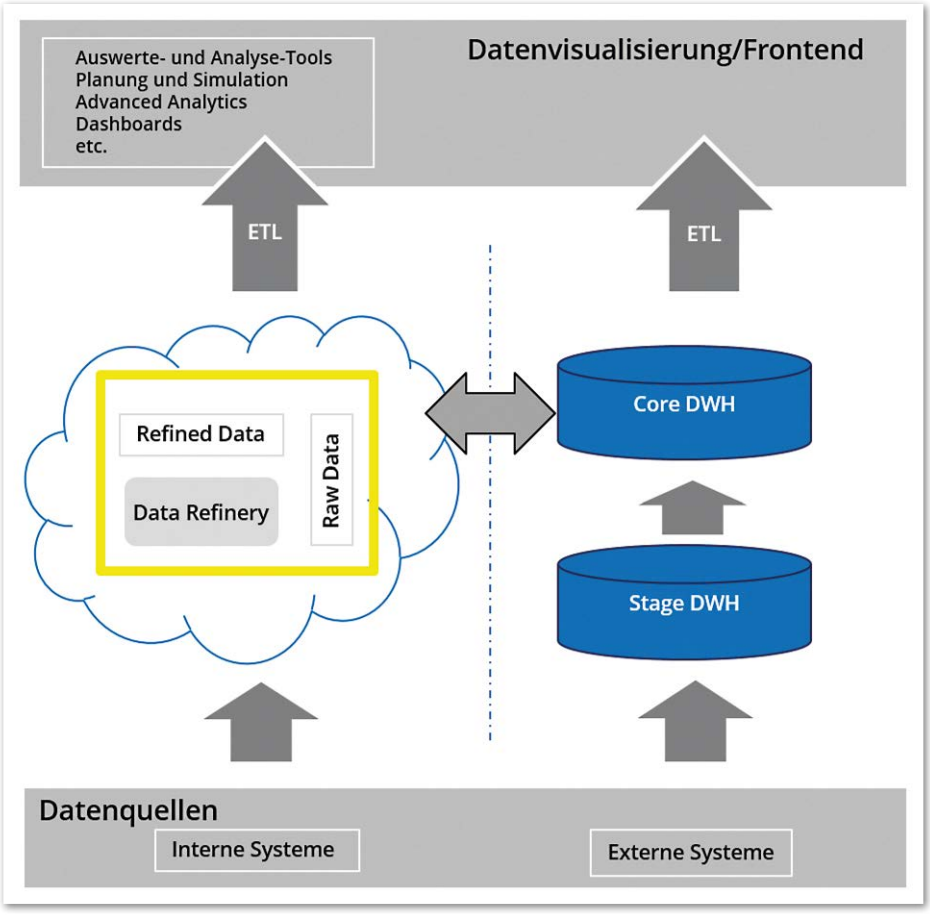

Abbildung 2: Hybride Architektur aus DWH und Data Lake

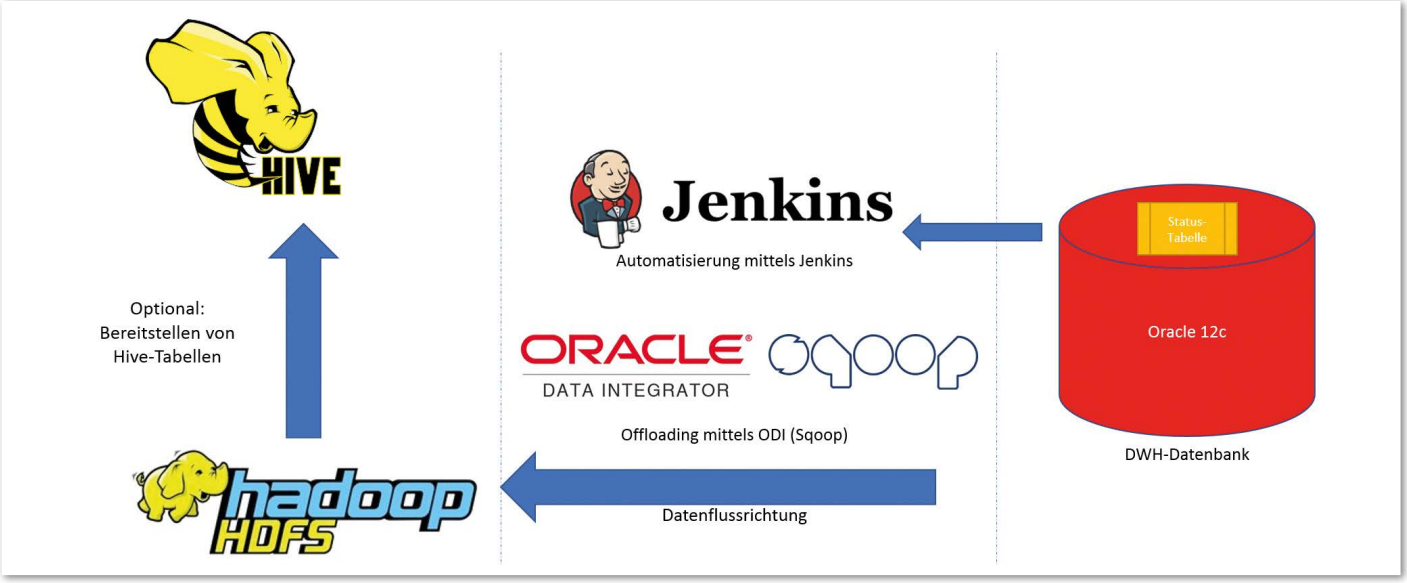

Abbildung 3: Unabhängiges Mastersystem

einen. Alternativ ist auch die Bildung von Datensilos möglich, um die Spezialisierung der verschiedenen Systeme zu nutzen und die Daten erst in einem gemeinsamen Frontend wieder zusammenzuführen. Es gibt verschiedene Möglichkeiten, ein Offloading-Vorgehen zu realisieren, und mehrere strategische Ansätze, um die technischen Hürden eines DWH-Offloading zu meistern:

- *Konzept für ein Mastersystem*
  An erster Stelle steht ein Konzept, aus dem ein sogenanntes „Mastersystem" hervorgeht. In vielen aktuellen Unternehmensinitiativen wird ein Big-Data-System als neues strategisches Mastersystem definiert, oftmals in Form eines Data Lake. In diesem Fall ist also ein DWH-Offloading zu implementieren, um die ganzheitlichen Unternehmensdaten in den Data Lake zu überführen. Dieses Architektur-Szenario bietet ein hohes Maß an Flexibilität, da ein Data Lake deutlich besser mit Echtzeit- und semistrukturierten Daten umgehen kann als das bestehende DWH-System. In diesem Fall findet ein Offloading der Daten vom DWH in den Data Lake statt.

- *Datenfluss vom Data Lake ins DWH*
  Als zweite Variante können Daten aus dem Data Lake ins DWH fließen. Es findet also ein Offloading in Richtung DWH statt. Dies kann vor allem dann sinnvoll sein, wenn eine Massendaten-Verarbeitung auf dem Big-Data-System durchgeführt werden soll und die hochgradig verdichteten Daten (Fakten) langfristig im DWH zu Auswertungszwecken erforderlich sind.

- *Neues Mastersystem*
  Eine dritte Möglichkeit kann sein, ein ganz neues System als Mastersystem zu etablieren, das die Metadaten der beiden Systeme hält und auch das Offloading in eine oder sogar beide Richtungen entsprechend überwacht und automatisiert. *Abbildung 3* zeigt einen technischen Vorschlag zur Realisierung einer solchen Systemlandschaft. Es wird ein neuer Master-Server aufgesetzt, der aus einer Oracle-Datenbank und einem Jenkins-Server besteht. Die Datenbank verwaltet sämtliche Metadaten, also Berechtigungen, technische Monitoring-Daten sowie fachliche Ladestände – genau die Metadaten, die eine Aussage darüber treffen, bis zu welchem Zeitraum die Daten erfolgreich zwischen den Systemen synchronisiert wurden.

Ein Praxisbeispiel für das Zusammenspiel von Data Lake und DWH

Dieser Abschnitt stellt ein hybrides Architektur-Szenario vor, das für einen Kunden des Autors aus der Telekommunikationsbranche entwickelt wurde und dort im produktiven Einsatz ist. Es besteht aus einem klassischen Data Mart und einem Data Lake als DWH-Core-Ersatz.

Die Kombination dieser beiden Systeme wurde gewählt, da die Berechnung von KPIs auf Call Data Records (CDRs) bisher ein technisch aufwendiger Prozess war, verbunden mit teuren Lizenzen und sehr langen Laufzeiten. Um den Mehrwert eines Data Lake in

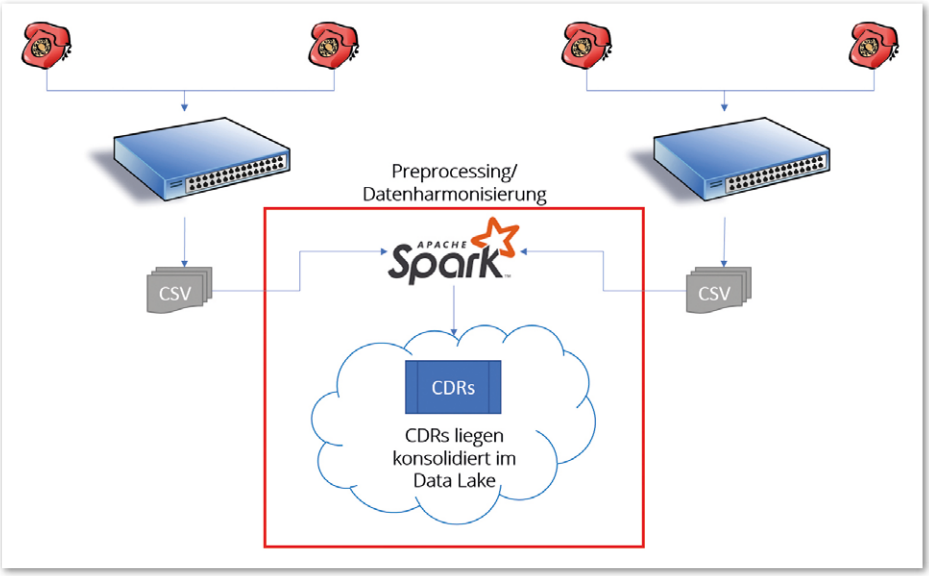

Abbildung 4: Der Erzeugungsprozess von Call Data Records

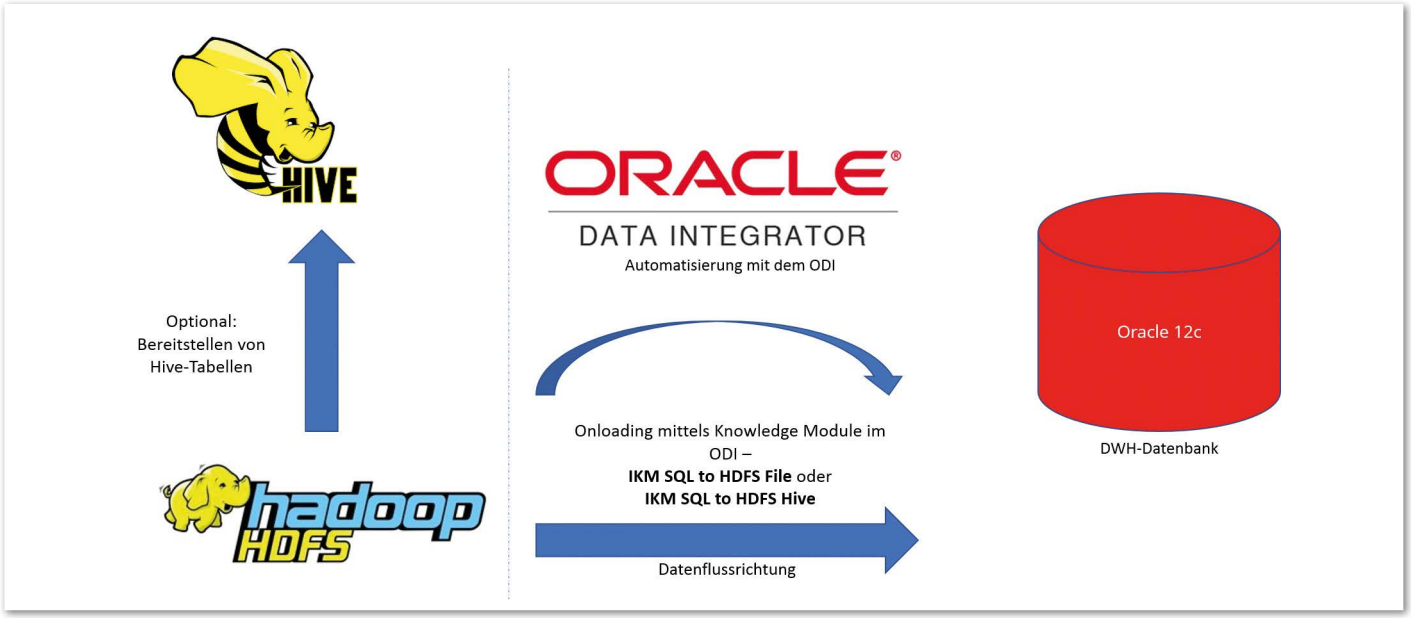

Abbildung 5: DWH-Onloading-Prozess

diesem Anwendungsfall nachzuvollziehen, ist zunächst ein gewisses Grundverständnis um den Erzeugungsprozess von CDRs vonnöten.

Abbildung 4 stellt den Prozess grafisch dar. Bei jedem Anruf werden die Detail-Informationen zur aufgebauten Verbindung von sogenannten Switches protokolliert und in Dateien abgespeichert. Da ein Telekommunikationsanbieter in der Regel Switches von diversen Herstellern im Einsatz hat, liegen die Daten in sehr unterschiedlichen Formaten vor. Dies betrifft sowohl die Struktur als auch den fachlichen Inhalt der Dateien. Daher müssen diese für eine Weiterverarbeitung harmonisiert und zusammengeführt werden. In der Regel findet dieser Vorgang in einer sogenannten „Mediationsphase" statt.

Das Ergebnis dieses Prozesses ist normalerweise eine sehr große Tabelle in einem relationalen Datenbank-System, die sämtliche Rohdaten zu den protokollierten Verbindungen enthält. Aufgrund der sehr hohen Datenmenge, die hier anfällt, handelt es sich um einen relativ kostenintensiven Verarbeitungsschritt. Diese Kosten werden durch teure Enterprise-Lizenzen und hohe Hardware-Anforderungen verursacht.

Die bereits angesprochene Mediationsphase mündet in einen sogenannten „Billing-Prozess". Dort werden die CDR-Daten um weitere Kundendaten angereichert, um die anfallenden Kosten einem bestimmten Kundenkonto zuordnen zu können. Erweiterte Auswertungsmöglichkeiten auf CDR-Basis sind an dieser Stelle meist nicht vorgesehen, weshalb viele Unternehmen auf die Idee kommen, diese Daten zusätzlich im Unternehmens-DWH zu verarbeiten, mit dem Ziel, weitergehende Kennzahlen zur Netzqualität oder zum allgemeinen Nutzerverhalten zu ermitteln. Sowohl aus Performance- als auch aus Datenschutzgründen werden die Daten dann oft nur temporär ins DWH geladen und nur so lange dort gespeichert, bis die Kennzahlen für den entsprechenden Tag erfolgreich berechnet wurden.

Genau an dieser Stelle war es für den Kunden sinnvoll, die Konsolidierung der Daten an eine zentrale Stelle auszulagern. Ein Big-Data-System eignete sich in diesem Fall sehr gut, da die Verarbeitung über dieses hochgradig parallelisiert und auf viele Rechenknoten verteilt werden kann. Die Dateien der Switches können direkt im Big-Data-System, beispielsweise mit einem Software-Tool wie Spark, verarbeitet und abschließend in einem Data Lake als zentralem Datenspeicherort abgelegt werden. Um Redundanzen bei der Verarbeitung dieser großen Datenmengen zu vermeiden, ist es empfehlenswert, sowohl die Billing-Daten als auch die weiteren KPIs, die zuvor im DWH berechnet wurden, im Big-Data-System zu berechnen. Anschließend können diese mit einem geeigneten Offloading/Onloading-Verfahren ins Zielsystem transferiert werden *(siehe Abbildung 5)*.

Sofern ein klassisches Oracle DWH inklusive des Oracle Data Integrator (ODI) im Einsatz ist, ist es auf recht einfach Weise möglich, ein Onloading (aus Sicht des DWH-Systems) durchzuführen und die Kennzahlen in die bestehende Datenlandschaft zu integrieren. Die aggregierten CDRs, die im Data Lake vorliegen, können mit dem Software-Tool Scoop, das bereits in den Knowledge-Modulen des ODI zum Einsatz kommt, ins DWH übertragen werden.

Auf diese Weise lassen sich die Tabellen im Data Lake ganz einfach in die Datentransformations-Logik im DWH integrieren und im regelmäßigen DWH-Batchprozess mitverarbeiten. So können Faktendaten, die bereits im Data Lake berechnet wurden, im DWH weiterverarbeitet werden. Ein Beispiel dafür ist die weitere Aggregation der Daten. Typischerweise werden die Verbrauchsdaten von Einzelkunden auch auf Monatswerte aggregiert und mit den Planumsätzen in Beziehung gebracht.

Quellen

[1] Sandmann, Big Data im Banking: Data Lake statt Data Warehouse?: *https://bankinghub.de/banking/technology/big-data-im-banking-data-lake-statt-data-warehouse*

[2] Pasupuleti/Purra, Data Lake Development with Big Data, Packt Verlag, Birmingham, 2015

[3] Reiss/Reimann, Das Data Lake Konzept: Der Schatz im Datensee: *https://www.it-daily.net/it-management/big-data-analytics/11222-das-data-lake-konzept-der-schatz-im-datensee*

Fabian Hardt
fabian.hardt@opitz-consulting.com