Bachelorarbeit

Entwicklung eines modernen Data Warehouse für den Vertrieb

zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

vorgelegt dem

Fachbereich Mathematik, Naturwissenschaften und Informatik
der Technischen Hochschule Mittelhessen

Dennis Vaupel

im September 2021

Referent:	Prof. Dr. Frank Kammer
Korreferent:	Prof. Dr. Harald Ritz

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Gießen, September 2021

Dennis Vaupel

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Die Digitalisierung des Arbeitslebens ist ein offenkundiges Phänomen, das in praktisch allen Branchen zu beobachten ist. Es spielt dabei keine Rolle, ob von Produktion, Marketing, Vertrieb, Logistik oder anderen Unternehmenszweigen die Rede ist.

Die konkreten Ausgestaltungen der „Vierten Industriellen Revolution“ sind aus gewissen Bereichen besonders prominent: Die Automatisierung von Produktionsabläufen mit Industrierobotern oder der Einsatz von autonomen Fahrzeugen in der Logistik sind vielbeachtete Beispiele.

Im Grunde noch wichtiger, aber weniger prominent sind unternehmensinterne Modernisierungen, die bürokratische Prozesse wie das Controlling betreffen. Geschäftsanalysen (englisch: Business Intelligence) und Datenanalysen allgemein sind zunehmend wichtige Faktoren, die in jedem Unternehmen stattfinden, egal ob es sich um Dienstleister oder produzierende Unternehmen handelt. Eine zentrale Rolle nehmen darunter jene Analysen ein, die Vertriebsprozesse betreffen.

Es ist heute zu beobachten, dass beispielsweise in mittelständischen Unternehmen Geschäftsanalysen auf einem teilweise primitiven Niveau durchgeführt werden. Dabei kommen oftmals altbekannte Werkzeuge wie Microsoft Excel zum Einsatz, mit denen eher einfache Unternehmenskennzahlen berechnet werden. Zwar bietet Excel mit Pivot-Tabellen etwas mächtigere Werkzeuge und es können auch je nach Unternehmen gewisse Analysen direkt in ERP-Systemen u.Ä. durchgeführt werden, doch um zeitgemäße Business Intelligence handelt es sich dabei nicht.

Bestehende Analysesysteme sind in ihren Möglichkeiten oft eingeschränkt, unflexibel und operieren nur auf relativ kleinen Datensätzen. Die Erstellung detaillierter und optisch ansprechender Geschäftsberichte ist somit einigermaßen aufwändig.

Die Praxis zeigt, dass fortgeschrittene Analysen in einigen Unternehmen heute noch gar nicht möglich sind. Vertriebsmitarbeiter müssen sich bei Planungen und strategischen Entscheidungen mangels harter Daten nicht selten an Erfahrungswerten orientieren und sich auf ihr Bauchgefühl verlassen. Besonders kritisch sind Auswertungen, bei denen komplexe Daten aus verschiedenen Systemen ausgewertet werden sollen. Dazu gehören beispielsweise die Analysen von Produktlebenszyklen, detaillierte Planungen von Projektkosten und sämtliche Auswertungen zum Thema Customer Relationship

Management.

Diese Situation ist auch deswegen ärgerlich, weil viele der erforderlichen Daten leicht zu erfassen wären oder bereits im Unternehmen vorhanden sind. So müssen Unternehmen alleine schon aus rechtlichen Gründen bestimmte Geschäftsunterlagen bis zu zehn Jahre aufbewahren[1]. Mit der richtigen Infrastruktur ließen sich viele Analysen gut umsetzen.

Eine solche Infrastruktur wird von einem Data Warehouse geboten. Ein Data Warehouse ist ein „physisches Informationssystem, das eine integrierte Sicht auf beliebige Daten zu Auswertungszwecken ermöglicht“[2]. Aus diesem Grund besitzen bereits viele Unternehmen ein Data Warehouse. Allerdings unterscheiden sich diese in Umfang, Qualität und somit auch in der tatsächlichen Nützlichkeit. Da derzeit vielerorts Infrastruktur und Unternehmensprozesse digitalisiert werden, wird der Wunsch nach moderner Business Intelligence seitens der Unternehmensleitung größer.

Es stellt sich also die Frage, wie eine Modernisierung in den Bereichen Data Analytics und Business Intelligence aussieht bzw. wie sie effizient und zukunftssicher umgesetzt werden kann. Teilweise wird dabei sogar grundsätzlich in Frage gestellt, ob Data Warehouses in dem Kontext noch zeitgemäß sind.

Diese Arbeit soll klären, wie Data Warehouses entworfen werden müssen, um zukunftssicher zu sein und auf zukünftige Herausforderungen adäquat reagieren zu können. Zu diesem Zweck werden die Anforderungen an ein solches System erläutert und eine entsprechende Referenzarchitektur entworfen. Anhand dieser werden bestehende Softwareprodukte auf ihre Eignung geprüft.

Kapitel 2

Historische Entwicklung

Die Vereinfachung von Routinearbeiten sowie die Unterstützung von Analysen und Vorhersagen gehören seit jeher zu den wichtigsten Softwareanwendungen. Mit dem Erscheinen der ersten Hochsprachen und der Verfügbarkeit des Speichermediums Magnetband entstanden schon in den 1960er Jahren Informationssysteme zum Zwecke betriebswirtschaftlicher Auswertungen.

Die Anwendungen von damals haben allerdings wenig mit der heutigen Softwarelandschaft gemein. Sowohl die eingesetzte Hardware als auch Software haben sich über die Jahrzehnte im Zuge neuer technischer Innovationen und wirtschaftlicher Anforderungen stark gewandelt.

Um die aktuelle Situation des Data-Warehouse-Sektors besser zu verstehen und aus Entwicklungen der jüngeren Vergangenheit Erkenntnisse für die nähere Zukunft gewinnen zu können, ist es angebracht, markante Entwicklungen zu untersuchen.

2.1 Evolution operativer Systeme

Geschichtlich sind den analytischen Systemen verschiedene operative Systeme vorausgegangen. In den 1970er Jahren ebneten das neue Speichermedium Festplatte und die Erfindung von Datenbankmanagementsystemen (DBMS) den Weg für leistungsstarke operative Systeme, die Anfragen interaktiv und in Echtzeit bearbeiten konnten (OLTP-Systeme)[3].

Diese operativen Systeme wurden in erster Linie von Personal aus der Buchhaltung bedient und waren darauf ausgelegt, bürokratische Routinearbeiten umzusetzen. Im Laufe der Zeit änderten sich jedoch sowohl die Anforderungen als auch die Architekturen dieser Systeme. Während sich frühe Werkzeuge lediglich auf die Erfassung von Lagerbeständen (*Inventory Control*) und die Planung von Materialbeschaffungen (*Material Requirements Planning*) beschränkten, erweiterte sich der Funktionsumfang der Systeme in den 80er Jahren enorm, sodass praktisch alle Unternehmensprozesse abgebildet werden konnten, von der Materialbeschaffung über die Produkion bis zur Finanzbuchhaltung[4].

Die neuen Anforderungen spiegelten sich auch in der technischen Umsetzung wider: Die komplexen und wartungsintensiven Eigenentwicklungen wurden in den 90er Jahren durch kommerzielle ERP-Systeme abgelöst. Diese vereinten alle bisherigen Funktionalitäten in einer Softwaresuite und boten dank ihres modularen Aufbaus die Möglichkeit, Funktionen anzupassen bzw. hinzuzufügen[4].

Das Aufkommen des Internets hat sich ab den 2000ern in der Form auf die ERP-Systeme ausgewirkt, dass einerseits die Instanzen innerhalb eines Unternehmens stärker integriert und somit konsistenter wurden, andererseits aber auch der direkte Datenaustausch mit externen Systemen ermöglicht wurde. Zu den neuen Funktionen zählten unter anderem Supply-Chain-Managenment mittels EDI-Schnittstelle oder die Integration von Shopsystemen zwecks E-Commerce.

Einige Softwarehersteller, wie etwa SAP, integrierten immer mehr Funktionen in ihre Produkte, sodass Unternehmen zunehmend vor eine Frage gestellt wurden: Ist ein solches monolithisches ERP-System (*Extended ERP*) die richtige Wahl oder doch besser eine Kombination verschiedener, spezialisierter Werkzeuge (*Best-of-Breed*-Ansatz)[5]? Diese Frage wurde umso dringender, da eine komplett neue Anforderungskategorie Relevanz gewann: Die systematische Analyse operativer Daten.

2.2 Aufkommen von Analysesystemen

Im Zuge der Verbreitung von OLTP-Systemen ergab sich natürlich auch der Wunsch, die anfallenden operativen Daten unter bestimmten Gesichtspunkten zu analysieren. Zu diesem Zweck begannen Unternehmen bereits in den 60er und 70er Jahren, Datenanalyseprogramme (OLAP-Systeme) zu entwickeln. Diese bestanden im Wesentlichen aus einem selbstentwickelten Extraktionsprogramm, das relevante Daten aus den Datenbanken der operativen Systeme kopierte, um diese dann in einer separaten Analysedatenbank zusammenzufassen und betriebliche Kennzahlen (*Key Performance Indicators*) zu berechnen. Diese Analyselösungen wurden *Management Information Systems* (MIS) genannt und richteten sich in erster Linie an Mitarbeiter in Führungspositionen. Probleme von MIS waren unter anderem die aufwendige Weiterentwicklung der Extraktionsprogramme und die damit einhergehende mangelnde Flexibilität.

Ab der Mitte der 80er Jahre kamen Weiterentwicklungen dieser MIS auf den Markt, nämlich *Executive Information Systems* (EIS). Diese waren in der Lage, große Datenmengen zu aggregieren und daraus Berichte zu erstellen. Endanwender waren Geschäftsleitungen, die EIS im Zuge größerer strategischer Entscheidungen zu Rate zogen.

Während MIS fachbereichs- bzw. abteilungsspezifische Auswertungen ermöglichten, trugen EIS Datenbestände aus verschiedenen operativen Quellen zusammen. Sie stellten die Analyseergebnisse graphisch in Berichten dar und konnten dank interaktiver Benutzeroberflächen auch vom technisch ungeschultem Managementpersonal bedient werden. Die Mehrheit der Unternehmen setzte in den frühen 90ern auf EIS-Produkte, die komplett oder größtenteils von Herstellern fertig entwickelt waren[6]. So konnten mehr Ressourcen auf die eigentliche Berichterstellung anstatt auf Entwicklung und Mitarbeiterschulung verwendet werden.

Trotz des erhofften Mehrwerts sind die traditionellen EIS-Lösungen der 90er Jahre als gescheitert zu betrachten. Die Gründe dafür waren sowohl sozialer bzw. unternehmenskultureller als auch technisch-architektureller Natur[7].

Fehlende Gesamtstrategie: Der Datenanalyse wurde im Unternehmen nicht die nötige Priorität zugewiesen. Die Geschäftsanalyse war bloß Beiwerk und es erfolgte keine adäquate Einordnung in die Gesamtorganisation des Unternehmens. Da infolgedessen nicht genügend Ressourcen alloziert wurden, mangelte es an verlässlichen Ansprechpartnern, qualifizierten Mitarbeitern und verbindlichen Standards.

Mangelnde Flexibilität: Es gab keine ernsthafte Strategie, wie mit neuen Anforderungen und technischen Entwicklungen umzugehen sei. Veraltete, starre Projektmanagementmethoden passten außerdem nicht mehr zu den dynamischen Anforderungen, mit denen Datenanalysten neuerdings konfrontiert waren[1].

Widersprüchliche Daten: EIS-Entwickler wurden den verschiedenen Fachabteilungen untergeordnet und erstellten Analysen nach Bedarf. Dazu extrahierten sie nach eigenem Ermessen, und ohne sich mit anderen Analysten abzusprechen, relevante Daten aus unterschiedlichen Quellsystemen. Aufgrund fehlender Koordination wurden oftmals redundante Daten gesammelt, die allerdings teilweise unterschiedlich in Berechnungen einflossen und so zu inkonsistenten Berichtsergebnissen führen konnten.

Politische Widerstände: Widersprüchliche Analyseergebnisse, teilweise extrem eskalierende Entwicklungskosten und schlecht in Zahlen ausdrückbarer Mehrwert führten zu Demotivation bei den Anwendern und Skepsis in der Führungsetage. Ohne zentrale Ansprechpartner konnten Ergebnisse schlecht evaluiert und Probleme schlecht reflektiert werden.

Die beschriebenen organisatorischen Probleme stammen sicherlich teilweise daher, dass damals noch unterschätzt wurde, welche herausragende Bedeutung die computergestützte Analyse von Unternehmensprozessen (*Business Intelligence*, kurz: BI) langfristig spielen würde. Sie hätten mit besserer Planung und Priorisierung vielleicht vermieden werden können. Anders verhält es sich mit den tiefgreifenden technischen Problemen; diese bedurften grundlegender Änderungen.

2.3 Notwendigkeit von Data Warehouses

Die Architektur, die sich durch die damalige Laissez-Faire-Einstellung bildete, war chaotisch und schwierig zu warten. Es gab keine Koordination zwischen den Datenanalysten, sodass im Laufe der Zeit hunderte von Extraktionsprogrammen im Unternehmen entstanden (vgl. Abb. 2.1). Zum einen entstanden dadurch große Mengen unnötig redundanter Daten im Unternehmen, die zusätzliche Kosten verursachten. Zum anderen wurden diese Daten zumeist von Anfang an stark aggregiert, sodass Details und vor allem die Nachvollziehbarkeit verloren ging. Die Entstehung dieses komplexen „Spinnennetzes“[3] ist der Hauptgrund für die Komplikationen der damaligen OLAP-Systeme.

[1] Inkrementelle bzw. agile Methoden der Softwareentwicklung steckten noch in den Kinderschuhen. Das Spiralmodell wurde erstmals 1988 systematisch beschrieben[8], das „Agile Manifest“ erschien 2001[9].

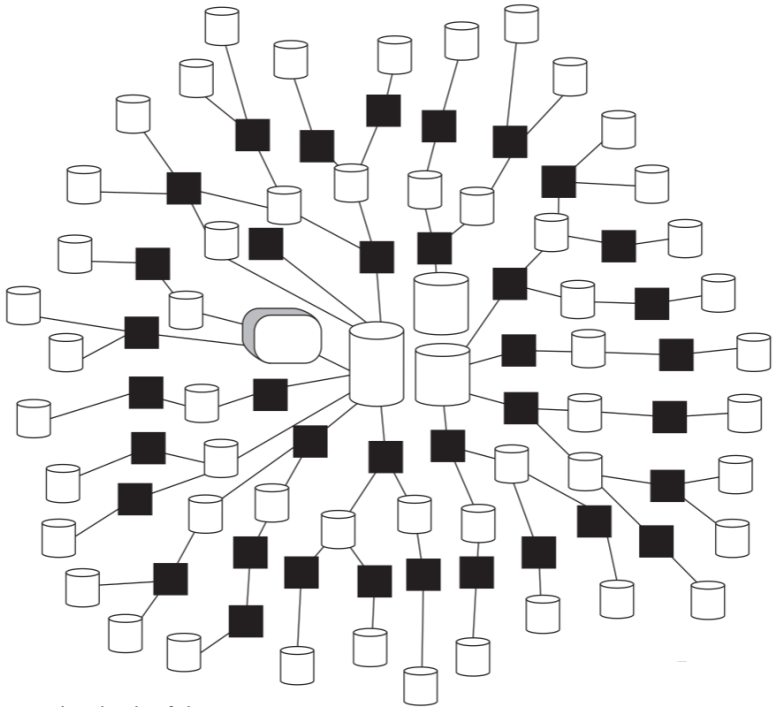

Abbildung 2.1: Das Ergebnis unkoordinierter Entwicklung ohne gemeinsame Datenbasis: Das „Spinnennetz“. Entnommen aus [3].

Der Gegenentwurf zu dieser Architektur ist das Data Warehouse (DW). Dabei handelt es sich um eine unternehmensweit einheitliches, leicht zu wartendes und flexibles System, das zwischen den ursprünglichen operativen Systemen und den *Data Access Tools* sitzt, mit denen schließlich Analysen und Monitoring realisiert werden (vgl. Abb. 2.2).

Heute kommen Data Warehouses in vielen Konzernen und zunehmend auch in mittelständischen Unternehmen zum Einsatz. Sie laufen zumeist teilweise oder komplett in der Cloud und bieten eine verlässliche Grundlage für modere Anforderungen wie Business Intelligence und Data Mining.

Der Data-Warehouse- und BI-Markt sind angesichts der Datenlage schwer zu quantifizieren. Es kann jedoch mit großer Sicherheit gesagt werden, dass er derzeit stark wächst. Der Data-Warehouse-Markt wird von 2017 bis 2022 um ca. 8% auf 20 Milliarden Dollar anwachsen[11]. Viele Unternehmen sind daran interessiert, ihre *Decision Support Systems* zu modernisieren und etwa jedes zweite Unternehmen setzt dahingehend bereits heute auf die Cloud.

Das Interesse an cloud-basierten Data-Analytics-Lösungen wird nicht zuletzt dadurch verdeutlicht, dass der Börsengang des Cloud-Analytics-Dienstleisters „Snowflake“ im vergangenen Jahr der bisher größte im gesamten Softwaresektor war[12].

Die Umsetzung neuer Analysesysteme bzw. die Migration bestehender Infrastruktur

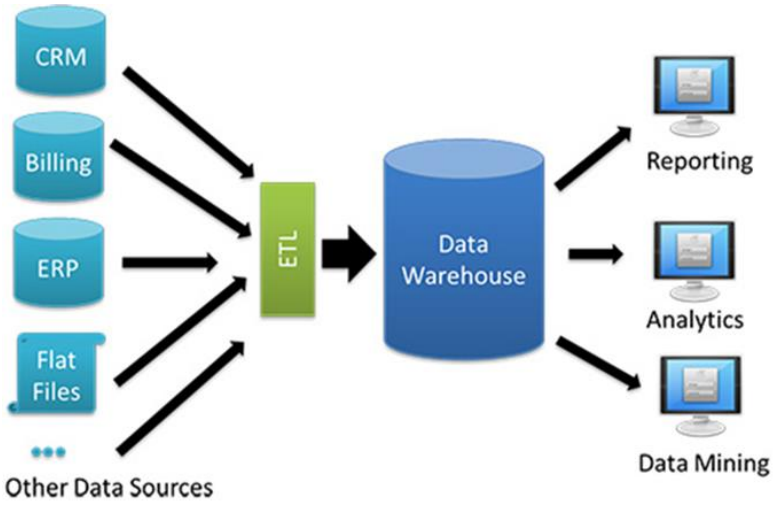

Abbildung 2.2: Klassisches Data Warehouse[10].

sind in jedem Fall enorme Unterfangen. Solche Projekte können nur gelingen, wenn die Anforderungen umrissen, die notwendigen Ressourcen abgeschätzt die technischen Möglichkeiten umfänglich evaluiert worden sind.

Kapitel 3

Anforderungen an moderne Data Warehouses

Viele Unternehmen sind an modernen Analyse- und Reporting-Systemen interessiert und bereit, in dafür notwendige Data Warehouse-Software zu investieren. In der Tat können moderne DSS heute ein signifikanter Wettbewerbsvorteil sein, müssen dafür aber auch in die allgemeine IT-Strategie des Unternehmens eingebunden sein und bestimmte Kriterien erfüllen. Die Auswahl und Implementierung einer BI-Infrastruktur ist eine ernstzunehmende Herausforderung. Ein solches Projekt kann ohne klare Anforderungsdefinition leicht zum Risiko für das Unternehmen werden.

Im Folgenden wird eine Referenzarchitektur umrissen, die auf die Bedürfnisse mittelständischer und größerer Unternehmen zugeschnitten ist und dabei möglichst flexibel und zukunftssicher sei.

3.1 Cloud-native Architektur

Eine grundlegende Entscheidung ist die Frage, ob und inwieweit eine solche Architektur cloudbasiert sein soll. Der Trend zur Migration in die Cloud ist bereits seit Jahren in vielen Bereichen der IT-Branche zu beobachten und bietet auch im Hinblick auf Data Warehouses zahlreiche Stärken.

Eine erste grundlegende Abwägung ist die, ob das Data Warehouse vor Ort im Unternehmen („On-Premises“) oder in der Cloud betrieben werden soll. Diese Entscheidung hat weitreichende Auswirkungen für das Unternehmen, beispielsweise im Hinblick auf die Anforderungen an das Personal oder die Kooperation mit Herstellern und Dienstleistern.

Der Hang zum Umstieg auf die Cloud lässt sich zwar in vielen Bereichen des IT-Sektors beobachten – bei ERP-Systemen hat dieser bereits vor 15 Jahren begonnen[13] – doch auch im Falle der Data Warehouses müssen Vor- und Nachteile abgewogen werden.

Durch den Einsatz cloudbasierter DWs kann eine **erhebliche Kostenreduktion** erzielt werden. Sie erlauben eine vergleichsweise günstige und schnelle Umsetzung, da

keine eigene Infrastruktur im Unternehmen geschaffen werden muss. Die Anschaffung von Hardware und der Arbeitsaufwand für Administratoren entfällt.

Wenn Hersteller nicht nur die Infrastruktur, sondern auch die fertige DW-Software anbieten, spricht man auch von *cloud-nativen* Lösungen oder auch von *Data Warehouse as a Service* (DWaaS). DWaaS-Produkte verfolgen oftmals einen Low-Code-Ansatz, sodass auch die Kosten für Entwickler auf ein Minimum reduziert werden können. Da es sich in der Regel um weit verbreitete und oft verwendete Systeme handelt, können bei Bedarf sehr leicht externe Dienstleister beauftragt werden, um spezielle Anpassungen vorzunehmen. Outsourcing ist in der Branche heute sehr weit verbreitet, etwa ein Viertel aller Unternehmen „lässt seine Cloud- und BI-Umgebung extern betreiben und verwalten, sogar 16 % lassen die Anbieter die Cloud-BI-Anwendung entwickeln.“[14].

Ein weiterer Grund für die Kosteneffizienz von Cloud-DWs ist das Prinzip der Trennung von Rechenleistung (*computation*) und Datenspeicherung (*storage*). Bei eigenen, lokalen Rechenzentren müssen sich die verfügbaren Ressourcen (Prozessoren und Speichermedien) immer an der maximal erwartbaren Last ausrichten und dauerhaft abrufbar sein. Bei gemanagten Cloudinstanzen hingegen können diese beiden Größen unabhängig voneinander beansprucht und skaliert werden.

Wenn die Zugriffshäufigkeit der vorhandenen Daten („Datentemperatur“) geschätzt bzw. bestimmt wurde, können die Daten dermaßen partitioniert werden, dass selten abgerufene Daten in günstigen serverlosen Data Lakes residieren und häufig abgerufene Daten in schnellerem, teurerem Speicher angesiedelt werden.

Bei der Rechenleistung kommt es schlicht dadurch zu Einsparungen, dass nötige Ressourcen nur dann alloziert werden, wenn sie benötigt werden. Im Falle des Vertriebs könnte es beispielsweise reichen, wenn jede Nacht neue Daten ins DW geladen werden und Rechenzeit sonst nur beansprucht wird, wenn a) der Entwickler Jobs manuell anstößt oder b) es zeitweise erhöhten Bedarf an Analysen oder Berichten gibt. Typische Szenarien, bei denen das Data Warehouse kurzfristig stark belastet wird, sind beispielsweise Analysen im Vorfeld des Weihnachtsgeschäfts, vierteljährliche Erstellungen von Quartalsberichten oder jährliche Umsatzplanungen fürs Folgejahr.

Die dynamische Bereitstellung von Rechenressourcen kommt dem Nutzungsverhalten von Data Warehouses sehr entgegen, da die Last üblicherweise viel ungleichmäßiger verteilt ist wie etwa bei OLTP-Systemen.

Es können aber nicht nur Spitzenlasten der Prozessoren abgefangen werden. Einige Bereiche des DW sind einem hohen Abfragevolumen ausgesetzt. Beispielsweise muss die Staging Area eine hohe Zahl gleichzeitiger Lese- und Schreibzugriffe bedienen können. Um zu verhindern, dass hier ein Flaschenhals entsteht, kann ein Read-Scale-Out realisiert werden[15]. Dazu wird eine Replika der Staging-Area-Datenbank erstellt, die nur Leseanfragen bearbeitet und somit das System entlastet (vgl. Abb. 3.1).

Eine weitere Stärke cloudbasierter Data Warehouses ist die hohe Verfügbarkeit und Sicherheit. Die meisten Cloud-Anbieter garantieren eine Verfügbarkeit von weit über 99%[1]. Unternehmensdaten sind zudem stärker geschützt als bei eigenen Rechenzentren,

[1]Hier einige Angaben zur Verfügbarkeit (Uptime) ausgewählter Cloud-Provider laut Dienstleistungs (*Service Level Agreement*): Amazon Redshift: 99,9%[16], Snowflake: 99,9%[17], SAP Cloud Services:

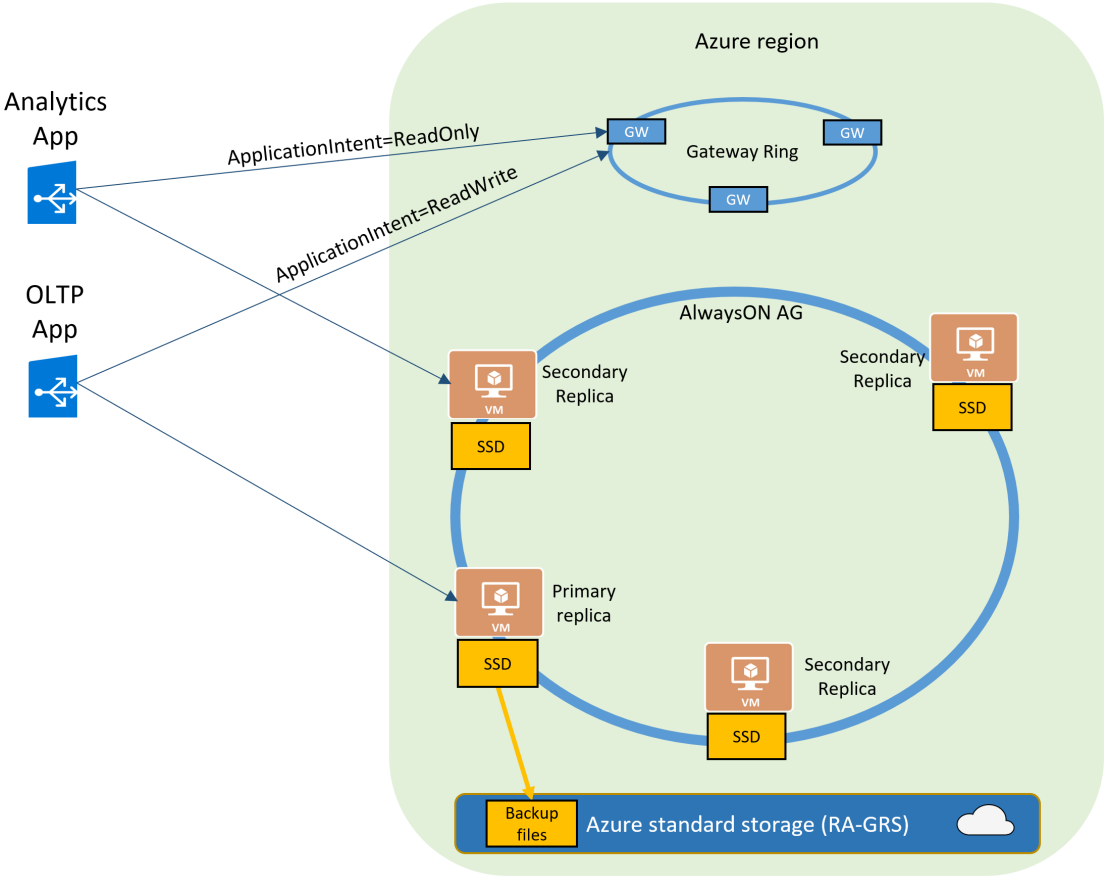

Abbildung 3.1: OLTP- und OLAP-Anwendungen greifen über ein Gateway auf eine Cloud-Datenbank zu. Die OLTP-Anwenung erhält Schreibzugriff auf eine Primärreplik, die OLAP-Anwenung Lesezugriff auf mehrere Sekundärrepliken[15].

da große Hersteller Sicherheitslücken tendenziell schneller finden und schließen.

Für multinationale Konzerne ist der cloudnative Ansatz auch deswegen interessant, weil sich damit vergleichsweise einfach *verteilte Data Warehouses* umsetzen lassen (vgl. Abb. 3.2). An allen Standorten des Unternehmens steht den Analysten eine einheitliche Infrastruktur zur Verfügung und es gelten gemeinsame Standards. Üblicherweise werden für verschiedene Standorte (konzeptionell) eigene DWs geschaffen, die große Mengen feingranularer Daten speichern und verarbeiten. Im Falle des Vertriebs können hier Details zu Angeboten, Rechnungen und Reklamationen erfasst werden. Lokale DWs können Preise in lokalen Währungen erfassen und sonstige Eigenheiten berücksichtigen. Das unternehmensübergreifende zentrale DW hingegen übernimmt nur für die Geschäts-

99,7%[18], Google BigQuery: 99,99%[19], Azure Synapse Analytics: 99,9%[20]. Diese Angaben können unterschritten werden, haben dann aber i.d.R. finanzielle Entschädigungen zur Folge.

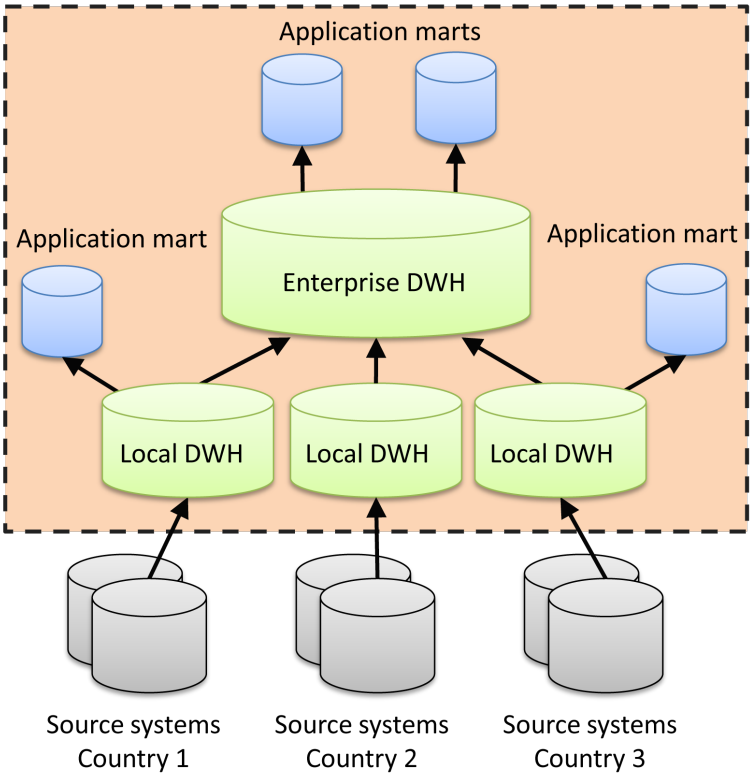

Abbildung 3.2: Verteiltes Data Warehouse[21]

leitung relevante, zumeist aggregierte Daten, die bei großen, strategischen Fragen helfen können.

Zu den Nachteilen cloudbasierter DWs zählen vor allem juristische Probleme. Datenschutzrichtlinien der BRD und der EU können Datenanalysten dazu zwingen, Daten entweder gar nicht oder nur in anonymisierter Form zu speichern. Diese Problematik besteht zwar allgemein bei DWs, ist aber in der Cloud noch verschärft. Aus diesem Grund sind in vielen Fällen spezielle Vereinbarungen mit den Betroffenen[2] nötig, damit diese Einschränkungen umgangen werden.

Insgesamt liegen die Vorteile des cloudnativen DWs klar auf der Hand. Richtig eingesetzt sind sie stabiler, mächtiger und günstiger als On-Premise-Lösungen. Falls bestimmte sensible Daten nicht auf fremden Servern gespeichert werden dürfen, ist immernoch eine hybride Architektur in Erwägung zu ziehen. Für die allermeisten Unternehmen liegt die Zukunft von Data Warehousing, Business Intelligence und Data Analytics in

[2]Dazu gehören einerseits Geschäftskunden (Zulieferer und Abnehmer), deren Daten aus EDI-Schnittstellen, Verträgen und Korrespondenzen gespeichert werden. Andererseits sind auch Privatkunden betroffen, bei etwa das Kaufverhalten oder die Interaktionen mit Webseiten und Newslettern analysiert wird.

der Cloud.

3.2 Unterstützung von Big Data

Seit den späten 2000er Jahren ist das Phänomen der zunehmend riesigen und heterogenen Datenmengen (*Big Data*) im Fokus der IT-Industrie. Daten, die im Unternehmen anfallen, werden immer stärker durch die drei Vs charakterisiert: *Volume, Velocity, Variety*[22] (zu deutsch: Menge, Schnelllebigkeit, Verschiedenartigkeit). Data Warehouses, die seit jeher mit genau diesen Herausforderungen konfrontiert waren, müssen in besonderem Maße auf diese Entwicklung reagieren.

Während frühe Cloudsysteme lediglich *cloud-enabled* waren, also bloß ohne große Anpassung in die Cloud migriert wurden, haben moderne Architekturen Lösungen entwickelt, um den neuen Anforderungen gerecht zu werden. Auch neue Anforderungen in Richtung Data Mining und KI müssen berücksichtigt werden.

Modulare, cloudbasierte Microservice-Architekturen können mit diesen neuen Anforderungen am besten Umgehen. Im Folgenden werden einige Techniken vorgestellt, die bei solchen Architekturen zum Einsatz kommen.

Data Lakes bieten sehr große und gleichzeitig güstige Speicherkapazitäten. Sie können sowohl aktuelle als auch historische Daten halten und stellen somit die Basis der Data-Warehouse-Infrastruktur. Neben strukturierten Daten werden auch immer mehr unstrukturierte Daten in Data Lakes gespeichert, z.B. Bilder, PDFs, Tonaufzeichnungen und so weiter.

Zu den kommerziell angebotenen Data Lakes gehören neben *Amazon S3* auch *Apache Hadoop File System* und *Microsoft Azure Data Lake*. Hadoop und Azure Data Lake sind auf Big-Data-Anwendungen ausgelegt und integrieren sich gut in entsprechende Data-Warehouse-Software (Apache Hive, Azure-Synapse-Analytics u.Ä.).

In-Memory-Datenbanken (IMDBs) bieten hochperformante Speicherzugriffe. Diese, meist spaltenorientierten, Datenbanken halten ihre Daten im Hauptspeicher und werden zunehmend in operativen Systemen, aber auch in Data Warehouses eingesetzt. Da sie im Gegensatz zu herkömmlichen Datenbanken höhere Betriebskosten haben, sollten sie ggf. nur für bestimmte kritische Daten genutzt werden.

Von den deutlich beschleunigten Speicherzugriffen profitieren mehrere Komponenten des DW: der DW-Kern und die Data Marts. Im Kern können Datentransformationen schneller durchgeführt werden. In den Data Marts können Daten schneller abgefragt und analysiert werden.

Die immer geringeren Kosten von IMDBs stellen die Existenz von Data Marts potenziell in Frage. Traditionell werden sie multidimensionel modelliert, um langsame JOIN-Operationen zu minimieren. Da diese ursprünglichen Hardwarebeschränkungen allerdings durch In-Memory-Techniken entfallen, könnten physische Data Marts in Zukunft nur noch virtuell im DW modelliert werden[23].

Zu Datenbanken, die In-Memory-Techniken unterstützen, gehören Redis[3], SAP HA-

[3] https://aws.amazon.com/redis/

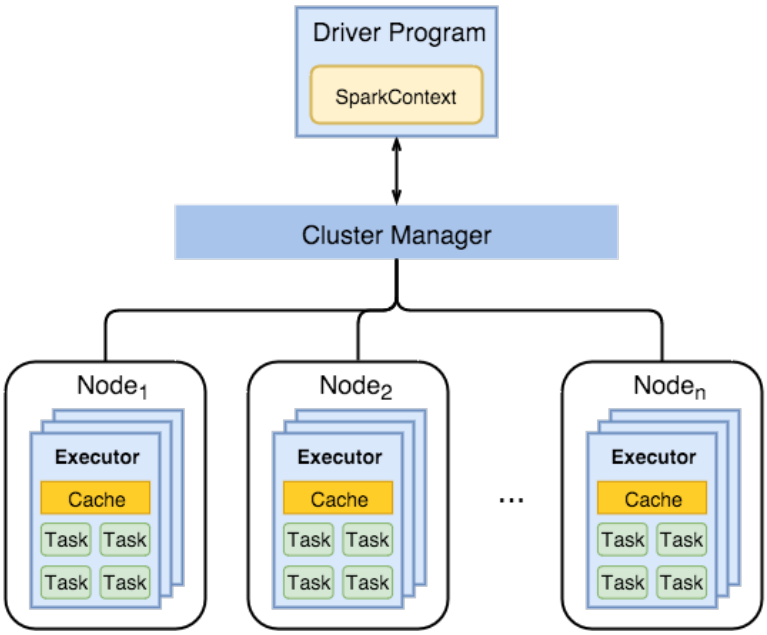

Abbildung 3.3: Grober Aufbau eines Apache Spark-Clusters[25]

NA[4] und Microsoft SQL Server[5].

Big-Data-Analytics erfordert clusterbasierte, massiv **parallele Datenverarbeitung**. Pionierarbeit leistete in diesem Bereich Google, das 2004 mit *MapReduce* ein Programmiermodell vorstellte[24], mit dem Berechnungen mit relativ wenig Aufwand verteilt auf Clustern ausgeführt werden können.

Die Kombination aus verteiltem Speicher und verteilter Berechnung ist essentiell für viele neue Anwendungsfälle wie künstlicher Intelligenz und maschinellem Lernen.

Prominente Vertreter solcher Lösungen sind Apache MapReduce in Apache Hadoop sowie das sehr ausgereifte Big-Data-Analytics-Framework Apache Spark (s. Abb. 3.3). Spark beinhaltet Komponenten zum verteilten Speichern, eine Abfragesprache, eine umfangreiche Bibliothek und Unterstützung für Machine-Learning-Anwendungen.

3.3 Flexible Schnittstellen

Ein Grund für die Datenmenge und -vielfalt in Data Warehouses ist die Vielzahl der Quellen, aus denen Daten für Analysen herangezogen werden können. DWs müssen also

[4] https://help.sap.com/viewer/fc5ace7a367c434190a8047881f92ed8/2.0.04/en-US/c39e5936ab9240a28cc85e1086315737.html

[5] https://cloudblogs.microsoft.com/sqlserver/2012/03/08/introducing-xvelocity-in-memory-technologies-in-sql-server-2012-for-10-100x-performance/

vielfältige Schnittstellen bieten, über die Daten in das Warehouse gelangen können (*data ingestion*).

Die klassische und weiterhin häufigste Datenquelle sind SQL-Datenbanken. Doch auch andere Datenbanken wie NoSQL-Datenbanken oder (proprietäre) Eigenentwicklungen sollten unterstützt werden. Zu den Standardschnittstellen gehören unter anderem ODBC- bzw. JDBC-Treiber sowie APIs wie SOAP, OData oder HTTP-Endpunkte. Alternativ können Daten auch vorerst aus inkompatiblen Quellen in eine gewöhnliche SQL-Datenbank exportiert werden, bevor sie dann in die Staging Area des Data Warehouse gelangen.

Weiterhin sollten Daten auch aus halbstrukturierten und unstrukturierten Quellen (Excel-Tabellen, XML-Dokumente, JSON-Dateien usw.) importierbar sein.

Flexible Schnittstellen sind gewissermaßen das Rückgrat moderner Data Warehouses: Egal wo sich Daten ursprünglich befanden, es muss immer einen Weg geben, sie einfach in das DW zu integrieren. Genauso wichtig sind Schnittstellen nach außen, über die *Data Access Tools* auf das DW zugreifen können. Denn Daten, die im DW gefangen sind oder nur mit großer Mühe ausgewertet werden können, sind praktisch nutzlos und werden die Adoption des Data Warehouses behindern.

3.4 Einfachheit und Erweiterbarkeit

Heute ist es besonders wichtig, dass Data Warehouses schnell angepasst und weiterentwickelt werden können, um neue Anforderungen zu erfüllen. In 70% aller Unternehmen ist laut eignen Aussagen heute in besonderem Maße Agilität, also erhöhte Geschwindigkeit und Produktivität, bei der Weiterentwicklung der Data-Warehouse-Lösungen gefordert[26]. Die Mehrheit der Unternehmen stimmt auch der Aussage zu, dass Datenlösungen gegenwärtig so schnell eingefordert werden wie noch nie zuvor.

Auf diese Forderungen reagieren DW-Entwickler mit agiler Projektdurchführung und dem Einsatz von Rapid Prototyping und ähnlicher Techniken. Anstatt einzelne Komponenten des Data Warehouses aufwendig schrittweise zu entwickeln, wird auf fertige Lösungen zurückgegriffen, um in kürzester Zeit einen vertikalen Prototypen zu entwickeln. Auf diese Weise kann den Auftraggebern vergleichsweise kurzer Zeit ein *Minimal Valuable Product* vorgeführt werden. Der Einsatz bestimmter Techniken kann so schnell getestet und evaluiert werden.

Dieses agile Vorgehen muss vom Data Warehouse allerdings auch technisch unterstützt werden. Idealerweise sollte ein einfach zu bedienendes, modulares Plug-and-Play-System benutzt werden, das sich leicht um neue Datenquellen, Datenflüsse und sonstige Komponenten erweitern lässt.

Hier können Cloud-Lösungen abermals ihre Stärke ausspielen: Da für experimentelle Projekte keine eigenen Testumgebungen physisch eingerichtet, sondern bloß hinzugebucht und konfiguriert werden müssen, können neue Techniken ohne große Hürden und ohne hohe Einstiegskosten getestet werden. Falls sie sich in der Testumgebung als passend erweisen, können sie anschließend in eine Produktivumgebung überführt und entsprechend hochskaliert werden.

Um die Produktivität bei der Implementierung neuer Lösungen zu steigern, sollten die Werkzeuge rund um das Data Warehouse intuitiv mittels graphischer Benutzeroberflächen bedienbar sein. Wo es geht sollten Produkte mit einem Low-Code- oder No-Code-Ansatz verwendet werden. In den meisten Bereichen, sei es Dateningestion, -transformation, Reporting und auch maschinellem Lernen gibt es Produkte, die per Drag-and-Drop funktionieren und automatisch Code erzeugen. Analog sollten auch weitere Aufgaben wie Tests oder Deployment automatisiert werden.

3.5 Umfangreiche Metadaten

Data Warehouses müssen nicht nur die zu analysierenden Daten selbst speichern, sondern auch Metadaten zu Umfang, Ursprung, Aktualität und Kontext der Daten. Die Metadaten im Data Warehouse können in verschiedene Kategorien unterteilt werden.

Technische Metadaten: Diese betreffen alle formalen Informationen bezüglich der vorliegenden Schemata bzw. der Überführungen in gewünschte Schemata und Mapping-Regeln. Auch Informationen zu konkret eingesetzten ETL-Werkzeugen und deren Konfiguration fallen hierunter, z.B. zugeteilte Ressourcen (Anzahl CPU-Kerne, maximaler Speicherverbrauch) oder Timeouts von ETL-Vorgängen.

Operative Metadaten: Zu diesen Metadaten gehören unter anderem Statusmeldungen von ETL- und Transformationsprozessen, Statistiken über verbrauchte Ressourcen und möglichst detaillierte Fehlermeldungen.

Metadaten zum Ressourcenverbrauch helfen Projektleitern, die Projektkosten zu kontrollieren und optimieren. Je nachdem wie genau das Monitoring ist, können verbrauchte Ressourcen je Abteilung, Team oder einzelnem Mitarbeiter analysiert werden.

Benutzerrollen und Zugriffsberechtigungen sind ebenfalls Metadaten, die gerade bei DWaaS eine wichtige Rolle spielen. Sie regeln den Zugriff auf Data Marts und Berichte, tragen zum Datenschutz bei und reflektieren firmeninterne Aufgaben und Befugnisse. Derartige Rollen und Berechtigungen sollten möglichst zentral verwaltet werden können, damit Authentifizierungen unkompliziert dienstübergreifend von statten gehen. Im besten Fall hat ein DW-Entwickler mit einem einzigen Konto gleichzeitig Zugriff auf Datenquellen, Data-Warehouse-Kern und Data Marts sowie BI-Tools.

Fachliche Metadaten: Die Entwicklung eines Data Warehouses ist ein interdisziplinäres Unterfangen, bei dem Entwickler auf Informationen aus den verschiedenen Fachbereichen des Unternehmens angewiesen sind.

Einerseits müssen mit Ansprechpartner aus der Buchhaltung, Administratoren der ERP-Systeme und ähnlichen Fachleuten relevante Datenbestände aus operativen Systemen identifiziert und interpretiert werden, damit vollständige und korrekte Daten in das DW geladen werden können.

Andererseits muss mit den Endanwendern aus dem Vertrieb eruiert werden, welche Analysen langfristig erwünscht sind. Beispielsweise sollten angedachte Analyseszenarien und dafür nötige Daten, Aggregationen und KPIs samt Berechnungsvorschriften dokumentiert werden. Richtig umgesetzt hilft diese Dokumentation den Endanwendern bei

der Interpretation der Analyseergebnisse, erhöht die Nachvollziehbarkeit und ermöglicht teilweise eigenständige Fehlersuche.

3.6 Niedrige Kosten

Data Warehouses müssen genau wie andere Softwareprodukte immer auch an ihrer Wirtschaftlichkeit gemessen werden. Selbst die perfekte Data-Warehouse-Lösung wird sich nicht durchsetzen können, wenn die Gesamtkosten unverhältnismäßig hoch sind.

Bei den Kosten für ein Data Warehouse können verschiedene Faktoren identifiziert werden, die in der Summe die Gesamtbetriebskosten bilden:

Lizenzen und Hardware: Hierbei handelt es sich um die naheliegendsten Kosten, nämlich jene, die vom Anbieter für verschiedene Dienstleistungen vorgeben werden. Dazu gehören Kosten für Speicher, Rechnerkapazitäten, Software-Lizenzen und weitere Dienste.

Gerade bei DWaaS gibt es zwei verschiedene Preismodelle. Zum einen können Dienste nach Verbrauch abgerechnet werden (*Pay As You Go*). Dies betrifft vor allem den Verbrauch von Hardwareressourcen, z.B. die stundengenaue Erfassung von CPU-Leistung oder die Gigabyte-genaue Erfassung von Speicher und Datenübertragungen. Der Kunde hat in der Regel die Möglichkeit seinen Ressourcenverbrauch mit Dashboards zu überwachen. Diese Art der Abrechnung ist zwar tendenziell teurer, eignet sich aber sehr gut für Testphasen, bei denen Technik neu eingeführt wird und es noch sehr schwer ist, den Ressourcenverbrauch realistisch abzuschätzen.

Zum anderen können feste Pauschalen vereinbart werden, die monatlich oder jährlich entrichtet werden müssen. Hierzu gehören einerseits Softwarelizenzen, aber auch die Provisionierung fester Ressourcen für das Data Warehouse ist möglich. Diese Möglichkeit ist für Szenarien interessant, wo der Ressourcenbedarf gut abschätzbar ist, da solche Fixpreise tendenziell günstiger sind.

In der Praxis handelt es sich oft um eine Mischform aus fester und verbrauchsorientierter Abrechnung. Die Berechnung der tatsächlichen Kosten gestaltet sich teilweise schwierig, da Preise gestaffelt sind und es viele verschiedene Optionen gibt. Unterm Strich sollten die effektiven Preise transparent und möglichst niedrig sein. Dort, wo die Preiszusammensetzung recht komplex ist, sollten Hersteller Preisrechner anbieten, die wenigstens Schätzungen ermöglichen.

Entwicklungskosten: Allgemein gilt in der IT-Industrie: „Zeit ist Geld.“ Dieser Spruch gilt auch für das Arbeiten im Data Warehouse. Je komplizierter und langwieriger Anpassungen am Data Warehouse sind, desto mehr Mannstunden müssen darin investiert werden. Die Personalkosten fließen ebenfalls in die Gesamtbetriebskosten ein.

Doch nicht nur Mitarbeiter sind teuer. Je nach Preismodell steigen beim Experimentieren und Debuggen auch zusätzliche Kosten für Datenzugriffe bzw. Datenübertragungen. Wenn der Anbieter dies ebenfalls in Rechnung stellt, schlägt die Entwicklung folglich doppelt zu Buche.

Um die Entwicklungskosten zu minimieren ist es demnach besonders wichtig, dass keine unnötigen menschlichen oder maschinellen Ressourcen auf einfache Routinearbei-

ten, wie das Anbinden externer Datenquellen, entfallen, sondern dass solche Prozesse „Out of the Box“, also mit minimalem Konfigurationsaufwand, funktionieren.

Personalkosten: Bei der Auswahl der Data-Warehouse-Lösung ist auch wichtig, wie die Verfügbarkeit qualifizierter Fachkräfte auf dem Arbeitsmarkt ist. Je spezieller und komplexer ein Produkt ist, desto schwieriger und teurer ist es, Personal zu finden. Bei Nischenprodukten ist es gleichermaßen schwierig, bei Bedarf externe Dienstleister zu finden.

Aus Managerperspektive ist auch abzuwägen, wie einfach es möglich ist, neue Mitarbeiter in eine DW-Software einzuschulen und weiterzubilden und ob es eventuell sogar möglich ist, einen bestehenden Mitarbeiter aus dem Vertreib, der Buchhaltung o.Ä. zum DW-Entwickler umzuschulen.

3.7 Dokumentation und Weiterbildungsmöglichkeiten

Eine gute Dokumentation der eingesetzten Data-Warehouse-Software ist für den Data-Warehouse-Entwickler unentbehrlich. Gerade bei einer agilen Projektdurchführung, bei der in kurzen Abständen neue Funktionen implementiert werden müssen, bedarf es leicht verständlicher, aktueller und praxisbezogener Dokumentation seitens des Anbieters.

Die Dokumentation sollte möglichst in verschiedenen Sprachen angeboten werden, etwa Deutsch und Englisch, und sprachlich wie inhaltlich stets korrekt sein. Minderwertige Maschinenübersetzungen sind nicht nur unleserlich, sondern hindern den Entwickler gewissermaßen bei der Arbeit.

Der Data-Warehouse-Anbieter sollte seine Dokumentation immer aktuell halten und ggf. kennzeichnen, für welche Produktversion die Dokumentation gilt. Darüber hinaus sollten neue Features, größere Änderungen und neue Produkte offen kommuniziert werden, beispielsweise über Blogs, Newsletter oder auf Konferenzen.

Wünschenswert sind weiterhin Möglichkeiten zur Weiterbildung. Von verschiedensten Bildungsangeboten wie virtuellen Video-Kursen oder persönlichen Workshops profitieren sowohl neue als auch erfahrene Mitarbeiter.

Kapitel 4

Vergleich von DW-Lösungen

Im vorangegangenen Kapitel[1] wurde das moderne Arbeiten im Data Warehouse auf Basis von Microsoft Azure umrissen und die praktischen Vorzüge eines modernen Cloud-Produktes verdeutlicht. Doch Microsoft ist bei weitem nicht der einzige Anbieter eines solchen Produktes. Heute ist eine Vielzahl an cloudbasierten Data-Warehouse-Produkten auf dem Markt erhältlich.

Im Folgenden werden drei moderne Data-Warehouse-Lösungen, nämlich „Microsoft Azure Synapse Analytics“, „Amazon Redshift“ und „SAP Data Warehouse Cloud“, im Hinblick auf die in Kapitel 3 beschrieben Anforderungen verglichen.

Diese genannten Produkte verkörpern eine neue Generation von Data-Warehouse-Lösungen. Sie werden von großen, etablierten Softwareunternehmen angeboten, sind Teil eines größeren Ökosystems und zeichnen sich auf ihre Weise durch besonders innovative oder kompetitive Angebote aus. Es handelt sich bei allen dreien um DWaaS-Produkte.

4.1 Mirosoft Azure Synapse Analytis

Microsoft Azure ist die Cloud-Computing-Sparte des amerikanischen Softwarekonzerns. Mit über 280 verschiedenen Diensten in über 20 Kategorien[2] bietet Microsoft ein vielfältiges Sortiment, das sich gegenseitig komplementiert und fortlaufend erweitert wird. In Data-Warehouse-Kontext sind vor allem die Produkte im Bereich Cloud-Speicher, Analytics, Big Data und Machine Learning interessant.

Konkret bietet Microsoft mit „Azure Synapse Analytics“ eine sehr ausgereifte und kosteneffiziente Data-Warehouse-Lösung, die besonders mit Flexibilität und Benutzerfreundlichkeit überzeugt. Nicht umsonst hat der renommierte IT-Marktanalyst Gartner Microsoft zum klaren Marktführer in den Bereichen „Analytics“ und „Business Intelligence“ erklärt[27]. Im Zusammenspiel mit anderen Azure-Diensten ergibt sich tatsächlich eine solide, erweiterbare Architektur (vgl. Abb. 4.1).

[1] Das ursprüngliche vorangangene Kapitel wurde aus dieser Fassung entfernt, da es firmeninterne Daten enthielt, die nicht veröffentlicht werden dürfen.

[2] https://azure.mirosoft.om/en-us/servies/

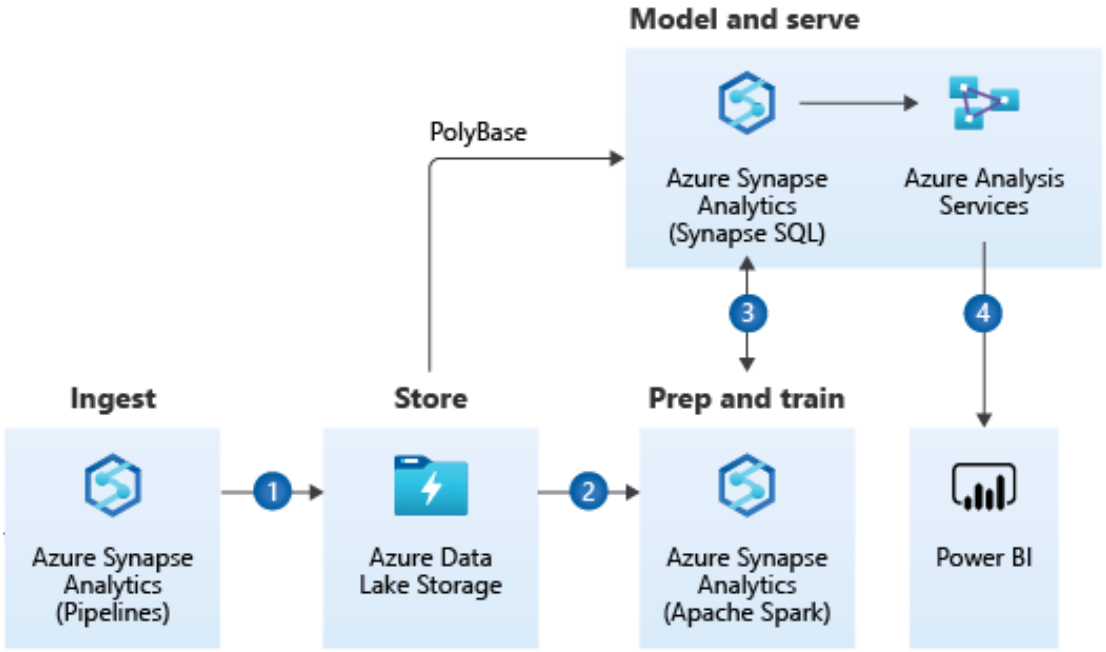

Abbildung 4.1: Microsofts Referenzarchitektur eines Enterprise Data Warehouses[28]: Laden, Transformation und Analysen mit Synapse Analytics, Speicherung im Azure Data Lake Storage und Berichterstellung mit Power BI.

Cloudnative Architektur

Bei Azure Synapse Analytics handelt es sich um eine voll gemanagte, serverlose Data-Warehouse-Lösung in der Cloud. Über eine zentrale Weboberfläche namens „Azure Synapse Studio“ kann von überall auf das Data Warehouse zugegriffen werden. Synapse Studio enthält eine integrierte Entwicklungsumgebung sowie verschiedenste Administrationswerkzeuge.

Als zugrundeliegende Speichertechnologie kommen Data Lakes („Azure Data Lake Gen2“) zum Einsatz. Obwohl Daten in den Data Lakes physisch als Dateien vorlegen (z.B. im komprimierten Parquet-Format), werden sie wie eine relationale Datenbank mit T-SQL abgefragt.

Diese Virtualisierung wird durch „PolyBase“, einer Funktion von SQL Server[29], ermöglicht: Mithilfe von PolyBase kann Synapse Analytics transparent auf jene Data Lakes zugreifen, als wären es reguläre SQL-Datenbanken. Aus diesem Grund spricht Microsoft hier auch von „serverlosen SQL-Pools“[30].

Für Datentransformationen und Analysen setzt Synapse Analytics Apache Spark ein. Wie in Kap. 4 gezeigt wurde, muss jedoch kein nativer Spark-Code geschrieben werden – dieser wird über Umwege generiert.

Unterstützung von Big Data

Massive Datenmengen können sehr leicht in das Data Warehouse integriert werden. Eine Möglichkeit ist es, externe Daten direkt mit Synapse Analytics in das DW zu übertragen. Da intern ohnehin Data Lakes genutzt werden, ist die Speicherung verhältnismäßig günstig.

Eine andere Möglichkeit ist es, massive Daten aus separaten Systemen anzubinden ohne sie in das DW zu laden. Es ist beispielsweise möglich, Azure Cosmos DB, Microsofts verteilte NoSQL-Datenbank für Echtzeitanwendungen, mit einer Technik namens „Azure Synapse Link“ mit dem DW zu verbinden. Bei dieser hybriden Architektur stehen operative Echtzeitdaten aus CosmosDB direkt, also ohne ETL-Prozess, für Analysen mit Synapse Analytics zur Verfügung. Microsoft bezeichnet dies als „hybrid transactional and analytical processing“ (HTAP)[31].

Microsoft bietet mit „Data Lake Analytics“ eine weitere Möglichkeit, Analysen auch abseits des Data Warehouses auf Data Lakes durchzuführen.

Flexible Schnittstellen

Die Konnektivität mit unterschiedlichsten Datenquellen wird im Azure-Ökosystem sehr stark betont. Unter dem Namen „Azure Data Factory“ bietet Azure eine Vielzahl an vorgefertigten Schnittstellen („Connectoren“), die alle in Synapse Analytics zur Verfügung stehen[32]. Es existieren Schnittstellen zu SQL- und NoSQL-Datenbanken, Fileservern und verschiedenen Webservices wie Salesforce oder Paypal.

Falls es keinen fertigen Connector gibt, können alternativ generische Schnittstellen genutzt werden. Auf Datenbanken kann stattdessen per ODBC-Connector zugegriffen werden und für externe Software (z.B. BI-Tools) gibt es weitere generische Schnittstellen wie SOAP, OData oder HTTP.

Die Interoperabilität zwischen verschiedenen Diensten gestaltet sich in der Praxis recht einfach, da solche Schnittstellen einfach in Synapse Analytics als „Linked Services“ eingerichtet werden und so unkompliziert verwaltet und kombiniert werden können.

Benutzerfreundlichkeit

Das zentrale Modellierungswerkzeug in Synapse Analytics sind *Pipelines*. Diese werden im Baukastensystem mit einer graphischen Oberfläche aus einzelnen *Activities* zusammengebaut (vgl. Kap. 4). Bei diesem Low-Code-Ansatz werden grundsätzlich bloß SQL-Kenntnisse vorausgesetzt. Pipelines, die Ladevorgänge, Aufbereitungen und Analysen beschreiben, werden im Hintergrund als JSON-Dokumente gespeichert und können auf diese Weise automatisch mit Azures Versionskontrolle erfasst werden. Synapse Analytics' Baukastensystem ist intuitiv, leicht zu bedienen und ermöglicht sehr schnelle Anpassungen.

Fortgeschrittene Analysetechniken, die über bloße SQL-Abfragen hinausgehen und fortgeschrittene Programmierung erfordern, können ebenfalls umgesetzt werden. Hierfür lassen sich beispielsweise „Azure Data Bricks“, also selbstgeschriebene Programme in

Python oder anderen Sprachen, in Pipelines integrieren. Ein entsprechender Editor ist direkt in Synapse Studio enthalten.

Die Benutzerfreundlichkeit und Softwareergonomie hat bei Azure-Produkten allgemein einen hohen Stellenwert. Azure-Produkte wie Synapse Analytics und verwandten Dienste werden größtenteils als Webanwendung mit ansprechender Oberfläche angeboten. Insbesondere bei Synapse Analytics können alle Funktionen direkt im Browser mit minimalem Programmieraufwand erledigt werden.

Die noch existenten Desktopanwendungen werden ebenfalls modernisiert. So wird beispielsweise die Windows-exklusive Datenbankentwicklungsumgebung „Microsoft SQL Server Management Studio“, die auch beim Arbeiten am Data Warehouse ausgiebig zum Einsatz kommt, durch die fortgeschrittenere Cross-Plattform-Anwendung „Azure Data Studio“ ersetzt.

Metadaten

Technische Metadaten zu Schemata, Mappings und eingesetzten Laufzeitumgebungen werden in den entsprechenden Pipeline-Editoren angezeigt. Das Deployment der gebauten Pipelines ist unkompliziert: CI/CD-Pipelines können genau wie alle anderen Pipelines einfach im graphischen Editor angelegt und überwacht werden (Azure Pipelines). Operative Metadaten zum Pipeline-Scheduling, Ressourcenverbrauch und auch Fehlermeldung können in Echtzeit im Bereich „Monitoring“ von Synapse Studio betrachtet werden.

Die Verwaltung fachlicher Metadaten, also die Dokumentation der Datenquellen, Kennzahlen und Hinweise zur Dateninterpretation, muss jedoch mit zusätzlichen Werkzeugen erledigt werden. Dazu bietet Microsoft die Softwaresuite Azure DevOps an, die in Synapse Analytics integriert werden kann. In Azure DevOps besteht die Möglichkeit, die Projektdokumentation in Form eines Wikis zu pflegen.

Überhaupt ist die Integration mit Azure DevOps in der Praxis äußerst hilfreich. Es werden Werkzeuge zum agilen Projektmanagement geboten (Azure Boards) und zur Versionsverwaltung mit Git (Azure Repos).

Für die Nutzerverwaltung bietet Microsoft den Dienst „Azure Active Directory“ an, mit dem feingranulare Berechtigungen vergeben werden können. Es lassen sich etwa unternehmensweit gültige Rollen für Data-Warehouse-Entwickler, Datenanalysten und Berichtskonsumenten einrichten.

Dokumentation und Weiterbildung

Microsofts Dokumentation ist sehr ausführlich und praxisbezogen. Es ist stets gekennzeichnet, wann die Dokumentation zuletzt aktualisiert wurde und für welche Produkte sie gilt. Die Hauptsprache ist zwar Englisch, die Maschinenübersetzung ist aber in Ordnung.

Die Dokumentation nimmt den Entwickler sehr gut an die Hand: Es werden nicht nur Details zu einzelnen Softwareprodukten erklärt, sondern auch typische Anwendungsfälle

und Standardarchitekturen beschrieben und diskutiert. Auch als neuer DW-Entwickler findet man sich darin gut zurecht.

Weiterhin werden von Microsoft in Zusammenarbeit mit zertifizierten Partnerunternehmen regelmäßig kostenlose Weiterbildungen angeboten[3]. Diese finden derzeit hauptsächlich online statt und werden meist interaktiv als Workshops angeboten.

Kosten

Eine große Stärke von Azure Synapse Analytics ist die klare Trennung der Ressourcen für Massenspeicher und Rechenleistung.

Bei ETL-Prozessen kommen spezielle *Data Integration Units* (DUIs) zum Einsatz, also Laufzeitumgebungen, deren Bandbreite, Parallelität und CPU-Ressourcen nach Bedarf konfiguriert werden kann. Bei Transformationen und Analysen hingegen werden nach Bedarf Apache Spark-Cluster alloziert, deren Laufzeit ebenfalls stundengenau abgerechnet wird.

Die effektiven Speicherkosten setzen sich also aus dem Volumen der Data Lakes sowie der Laufzeit der DUIs zusammen.

Ein Preisrechner mit verschiedenen vorgefertigten Szenarien ist auf Microsoft Azures Internetseite zu finden[4].

4.2 Amazon Redshift

Redshift ist Amazons Data-Warehouse-Lösung und Teil der Amazon Web Services (AWS), der Cloud-Computing-Sparte des amerikanischen Konzerns. Obwohl Amazon Redshift ein eigenes Produkt ist, integriert es sich sehr gut in das restliche AWS-Ökosystem. Die Einrichtung und Administration eines Data Warehouses mit Amazon Redshift geschieht beispielsweise in der vereinheitlichten AWS Managementkonsole.

Cloudnative Architektur

Bei Amazon Redshift handelt es sich um ein Cloud-natives, voll gemanagtes Data Warehouse. Redshift basiert auf einer proprietären verteilten Datenbank, die eine clusterbasierte *Massive-Parallel-Processing-Architektur* (MPP) aufweist: Daten werden verteilt in verschiedenen *Compute Nodes* gespeichert und verarbeitet. Dank dieser *Shared-Nothing-Architektur* können Abfragen und Ladevorgänge parallel im Cluster ausgeführt werden.

Als Abfragesprache kommt Redshift SQL zum Einsatz. Dies ist eine Abwandlung von PostgreSQL und enthält ergänzende Anweisungen zur Datenpartition innerhalb des Clusters u.Ä.

Redshifts Datenbank ist spaltenbasiert und setzt sehr aggressiv *Kompression* ein. Spalten werden individuell mit der effizientesten Methode komprimiert, wodurch eine

[3] Einen Überblick gibt es hier: https://events.microsoft.com/?language=Deutsch
[4] https://azure.microsoft.com/en-us/pricing/calculator/

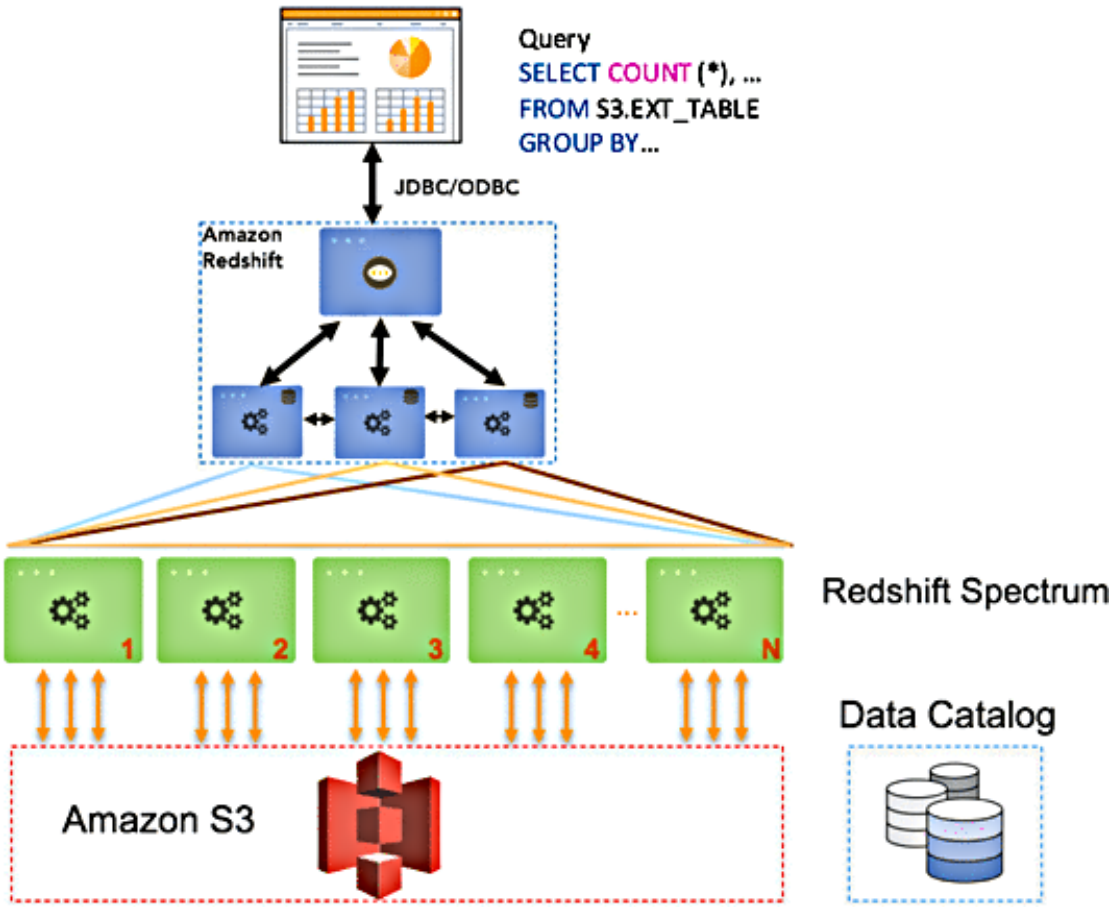

Abbildung 4.2: Der Amazon Redshift Cluster (blau) kommuniziert per ODBC-Connector mit Analyseprogrammen. Der Kern des Data Warehouses kann über Redshift Spectrum weitere Daten abfragen oder laden[33].

insgesamte Reduktion des Speicherbedarfs von über 70% möglich ist[34]. Redshift bietet darüber hinaus den eingebauten Befehl „ANALYZE COMPRESSION“, mit dem weitere Optimierungsmöglichkeiten erkannt werden können.

Die starke Komprimierung beschleunigt die Ausführungsgeschwindigkeit im Cluster, da Daten schneller auf Compute Nodes aufgeteilt werden. Redshift setzt darüber hinaus weitere Optimierungstechniken ein, um eine hohe Geschwindigkeit der Datenbank zu erreichen.

Es werden automatisch *Zone Maps* angelegt, die Metadaten über vorkommende Spaltenwerte anlegen. Außerdem kann eine automatische Sortierung von Spalten mit dem Schlüsselwort „SORTKEY“ erzwungen werden. Dieses *Data Sorting* wie auch die Informationen aus den Zone Maps ermöglichen es, bei vielen Abfragen den nötigen Suchbereich stark zu verkleinern und somit die durchgeführten Scans zu vermindern.

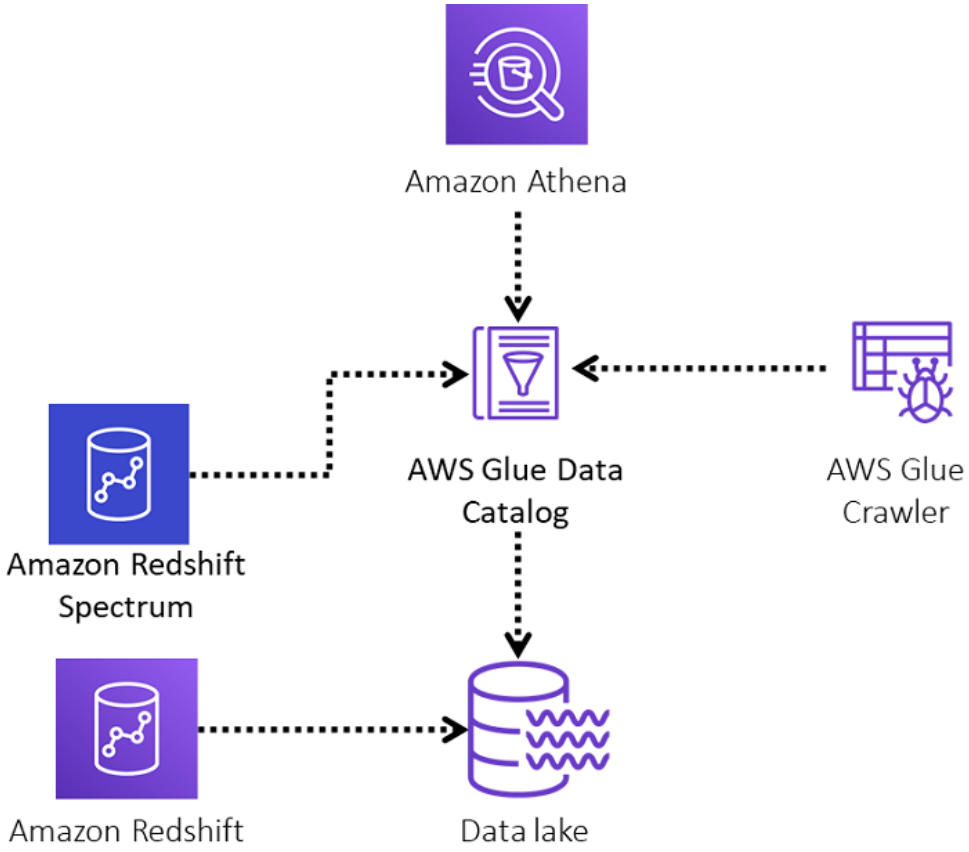

Abbildung 4.3: Data-Lake-Anbindung in Amazon Redshift: Nachdem ein S3-Data-Lake mit AWS Glue Crawler indiziert wurde, kann dessen Inhalt mit Redshift Spectrum oder anderen Methoden abgefragt werden. Entnommen aus [35].

Unterstützung von Big Data

Für Big-Data-Anwendungen besteht die Möglichkeit, einen S3 Data Lake Storage zu benutzen. Es können sowohl neue Daten standardmäßig in diesen Data Lake geladen werden als auch bestehende Daten aus dem Redshift-Cluster in diesem umgelagert werden[5]. Dieser kann mit Redshift Spectrum ebenfalls an das DW angebunden werden[35]. Voraussetzung dafür ist, dass der zuvor mit AWS Glue Crawler ein „Data Catalogue“ erstellt wurde. Dieser enthält Informationen zu Ort und Struktur der Daten, sodass Redshift diese verwenden kann.

Abseits von Redshift bietet AWS ein breites Spektrum an Diensten, um auf diesem Data Lake anderweitig Analysen durchzuführen. Dazu gehören Athena (einfache SQL-Abfragen), Amazon EMR (Big-Data-Analytics) und Amazon SageMaker (Machi-

[5]Zu diesem Zweck gibt es in Redshift SQL den Befehl „UNLOAD“.

ne Learning).

Flexible Schnittstellen

Es gibt verschiedene Möglichkeiten wie Daten in das DW geladen werden können. In vielen Fällen werden Daten aus Amazon Simple Storage Service (S3) geladen; um diesen Prozess zu vereinfachen, wurde Redshift Spectrum geschaffen. Mit Redshift Spectrum können Daten in verschiedensten Formaten wie Apache OCR, Apache Parquet oder JSON mit SQL-Befehlen aus S3 geladen werden. Alternativ können Daten auch mit Hilfe von AWS Glue, Amazons dediziertem ETL-Werkzeug, aus externen Quellen geladen werden.

BI-Anwendungen und andere externe Software greift mittels ODBC-/JDBC-Treibern oder Redshifts „Data API“ auf das Data Warehouse zu. Da Redshifts SQL-Dialekt fast identisch zu PostgreSQL ist, kann bei der Entwicklung externer Anwendungen unter Umständen auch auf PostgreSQL-Bibliotheken zurückgegriffen werden.

Benutzerfreundlichkeit

Redshifts Weboberfläche ist eher minimalistisch und bietet einen begrenzten Funktionsumfang. Der eingebaute „Query Editor“ ermöglicht das interaktive Schreiben und Testen von SQL-Abfragen. Mit dem „Query Scheduler“ wird der automatische Ablauf von SQL-Skripten geplant[36]. Diese Grundfunktionen sind größtenteils textbasiert und erfordern das manuelle Bearbeiten von Skripten und JSON-Konfigurationsdateien.

Um auf einfachere Weise komplexe ETL-Prozesse zu konfiguieren, muss der Dienst „AWS Glue“ eingesetzt werden. Mit dem graphischen Transformationseditor „Glue DataBrew“ ist die Datenverarbeitung mit Hilfe vorgefertigter Bausteine nach dem Drag-and-Drop-Prinzip möglich (s. Abb. 4.4). AWS Glue ist nicht Teil von Redshift und muss daher zusätzlich provisioniert und bezahlt werden.

Metadaten

Die Metadaten in Amazon Redshift beziehen sich hauptsächlich auf Ressourcenverwaltung, Scheduling und Datenherkunft. Fachliche Metadaten können leider nicht abgebildet werden. Der Fokus liegt hier ganz klar auf dem Clustermanagement und den Sicherheitseinstellungen.

Der Zugriff auf Redshift-Cluster ist auf verschiedenen Ebenen gesichert. Einerseits wird das Data Warehouse in einer eigenen *Virtual Private Cloud* betrieben, sodass der Zugriff nur über gesichterte Kanäle wie IPsec-VPNs erfolgen kann. Andererseits werden Zugriffsberechtigungen im Rahmen des „AWS Identity and Access Management“ mit Rollen geregelt. Diese Verwaltung geschieht AWS-übergreifend mit der graphischen Benutzeroberfläche namens „IAM Management Studio“.

Benutzerrollen sind auch aus einem anderen Grund von Belang. Anfragen an Redshift werden stets in Warteschlangen (Queues) eingereiht. Mit Hilfe des „Redshift Workload Management“ (WLM) können diese Warteschlangen unterschiedlich priorisiert werden.

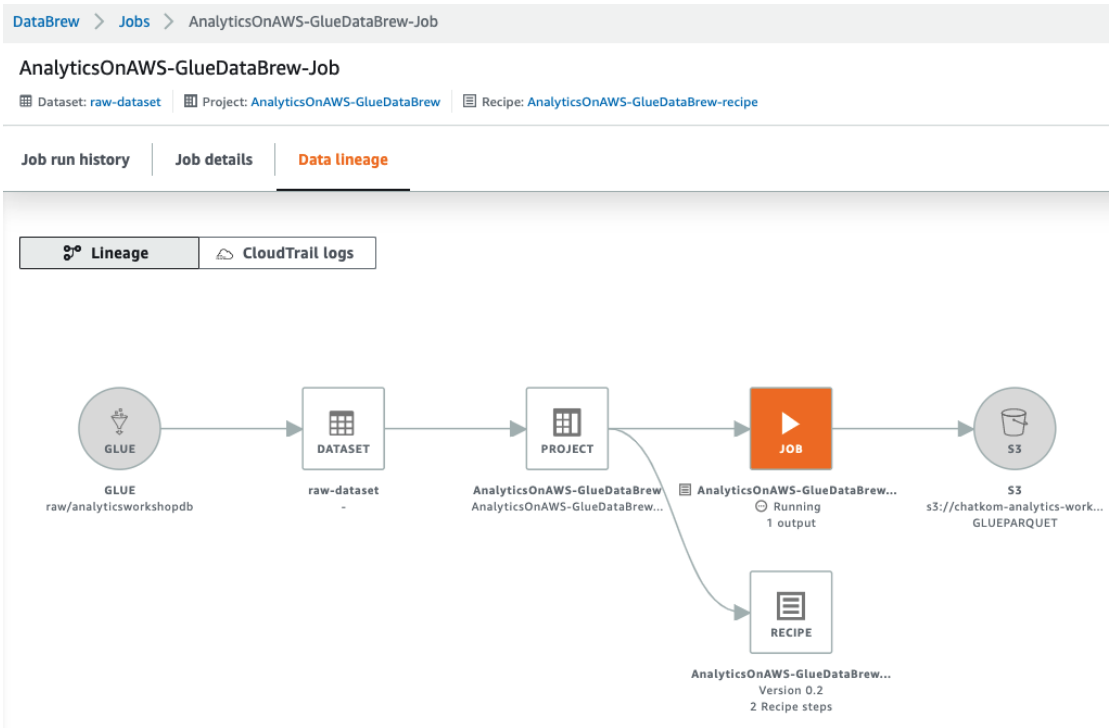

Abbildung 4.4: ETL-Prozess mit AWS Glue[37].

Es können verschiedene Parameter eingestellt werden wie maximaler Speicherverbrauch, maximal gleichzeitige Jobs und Timeouts. So können beispielsweise analytische Lasten zu Geschäftszeiten gegenüber ETL-Prozessen priorisiert werden.

Dokumentation und Weiterbildung

Die Dokumentation von Amazon Redshift ist recht umfangreich, aber leider in mancher Hinsicht etwas unzulänglich. Erstens ist nicht klar erkennbar, wann die Dokumentation geändert wurde. Zweitens ist zwar eine deutsche Übersetzung für Redshift verfügbar, Amazon warnt jedoch selbst, dass sie fehlerhaft sein könnte und im Zweifel die englische Fassung gültig ist. Drittens sind die Erklärungen oft sehr trocken und ohne Screenshots. Das ist schade, weil Amazon Arbeitsabläufe für viele typische Anwendungsfälle ansonsten eigentlich recht praxisnah erklärt.

Amazon leistet intensive Öffentlichkeitsarbeit und bietet Entwicklern auf verschiedenen Plattformen Möglichkeiten zur Weiterbildung. Besonders hervorheben kann man hier das Programm „AWS Events and Webinars“[6], in dessen Rahmen regelmäßig Webinare, Workshops und Vorträge zu verschiedenen AWS-Themen angeboten werden, sowie

[6]https://aws.amazon.com/events/

den Youtube-Kanal von Amazon Webservices[7], auf dem neue Technologien und Anwendungsmöglichkeiten vorgestellt werden.

Kosten

Amazon wirbt mit schneller und kostengünstiger Speicherung großer Datenmengen bis in den Petabyte-Bereich. Konkret sollen durchschnittliche Betriebskosten von unter 1000$ pro Terabyte pro Jahr möglich sein, d.h. eine Ersparnis von ca. 90% im Vergleich zu einem selbst betriebenen Data Warehouse[34]. Diese Angabe geht allerdings von optimaler Kompression aus und variiert je nach Konfiguration des Redshift-Clusters.

Dazu kommt, dass die Entwicklungskosten bei Redshift höher als bei anderen Anbietern ausfallen können. Denn um das volle Potential der MPP-Architektur auszuschöpfen, müssen Workflows entsprechend angepasst werden. Beispielsweise sollten Quelldateien vor dem Ladevorgang passend aufgeteilt werden, um tatsächlich parallel geladen werden zu können.

Der Wartungsaufwand des Clusters ist zudem etwas höher. Da Daten bei DELETE-Anweisungen aus Performance-Gründen nur logisch als gelöscht markiert werden, müssen in regelmäßigen Abständen ANALYZE- und VACUUM-Befehle durchgeführt werden, um Speicherplatz freizugeben. Da der Cluster zu diesen Zeiten belegt ist, geht dies nur außerhalb der Betriebszeiten.

Ein grundlegendes Problem ist, dass es keine klare Trennung zwischen Speicher und Rechenleistung gibt. Dies macht die Skalierung schwieriger und teurer. Es sollte also sorgfältig geprüft werden, ob die gesamten Betriebs- und Mitarbeiterkosten tatsächlich so niedrig ausfallen.

4.3 SAP Data Warehouse Cloud

SAP ist Deutschlands größtes Softwareunternehmen und Hersteller verschiedener Geschäftsanwendungen. Zu der größten und weltweit bekanntesten Produktsparte gehören SAPs ERP-Systeme. Laut eigenen Aussagen berühren 77% aller weltweiten erwirtschafteten Umsätze auf irgendeine Weise SAP-Systeme[38].

Seit einigen Jahren konzentriert sich SAP stärker auf Cloud-Dienste und hat mit der „SAP Business Technology Platform“ (BTP) eine neue Sparte geschaffen, die moderne, cloudbasierte Anwendungen aus den Bereichen Data Management, Data Analytics und Machine Learning Umfasst.

Das Produkt „Data Warehouse Cloud“ (DWC) ist SAPs Versuch einer webbasierten Data-Warehouse-Lösung. Es umfasst Werkzeuge zum Monitoring, zur Dateningestion, Datenmodellierung und -visualisierung. Potentiell könnte DWC langfristig SAPs bestehende On-Premise-Data-Warehouse-Produkte wie BW/4HANA ersetzen, in welchem Zeitrahmen das passiert ist allerdings derzeit ungewiss[39].

[7] https://www.youtube.com/user/AmazonWebServices

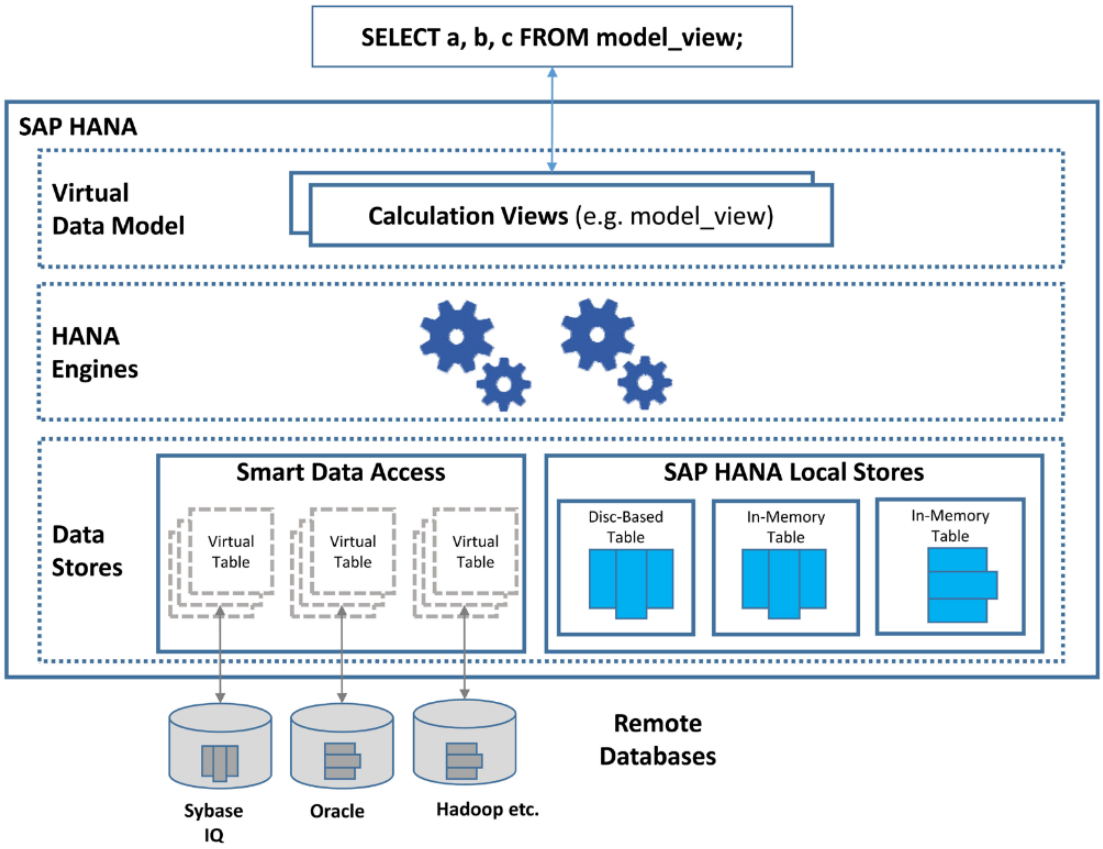

Abbildung 4.5: Architektur von SAP HANA[40].

Cloudnative Architektur

Als Speichertechnik kommt in SAP Data Warehouse Cloud die proprietäre verteilte Datenbank SAP HANA zum Einsatz. Prinzipiell handelt es sich um eine spaltenbasierte In-Memory-Datenbank, die dementsprechend performante Operationen ermöglicht.

Da es aber natürlich äußert kostspielig sein kann, ein komplettes Data Warehouse im Hauptspeicher zu halten, unterstützt HANA intern auch weitere Speichertechniken („Local Stores“) wie die günstigere Speicherung auf Festplatten bzw. SSDs (vgl. Abb. 4.5).

SAP HANAs zugrundeliegende Speichertechnologien können nach Bedarf hierarchisiert werden („Dynamic Tiering“). Für die Speicherung großer, statischer („kalter“) Daten können per „Smart Data Access“ auch Data Lakes virtuell in HANA verwendet werden[40].

SAP ist zwar ein Softwarehersteller, aber kein Infrastrukturprovider. Die Data Warehouse Cloud wird bei SAP lizensiert, aber über dritte „Hyper Scaler Provider“ gehostet. Standardmäßig übernimmt AWS das Hosting der HANA-Instanz.

Bei Bedarf können HANA-Instanzen auch eigenständig bei verschiedenen Anbietern gehostet werden[8,9,10].

Unterstützung von Big Data

Die Unterstützung von Big-Data-Anwendungen scheint derzeit noch ein blinder Fleck in der DWC zu sein. Seitens SAP existiert diesbezüglich keine allgemeine Referenzarchitektur.

Zwar sind SAP HANA Cloud Data Lakes in DWC konfigurierbar und auf Data Lakes kann virtuell zugegriffen werden, aber es gibt im Bezug auf die Data Warehouse Cloud keine dedizierten Lösungen für Echzeit-Architekturen, machinelles Lernen u.Ä.

Vermutlich werden entsprechende Lösungen in Zukunft entwickelt und in DWC integriert. SAP arbeitet derzeit mit großen Industriepartner wie Google Cloud[41] an neuen Lösungen. Eine interessante neue Technik in der Richtung ist SAP Vora: eine verteilte NoSQL-Datenbank, die in Kubernetes-Pods auf Hadoop-Clustern läuft und die parallele Verarbeitung mit Apache Spark unterstützt. Leider gibt es bisher kein gängiges Modell, um Vora in die DWC zu integrieren.

Flexible Schnittstellen

Die Data Warehouse Cloud bietet Schnittstellen zu anderen SAP-Diensten (On-Premise und in der Cloud) sowie zu verschiedenen externen Datenquellen. Es gibt diverse vorkonfigurierte Schnittstellen zu populären Datenbanken und anderen Datenspeichern wie Amazon Redshift oder Azure Data Lake Stores[11].

Des Weiteren können beliebige Datenquellen mittels generischer OData- oder JDBC-Schnittstelle angebunden werden. Somit können BI-Werkzeuge und sonstige Software auf DWC-Daten zugreifen.

SAP-Datenquellen genießen eine gewisse Sonderstellung: Dank „SAP HANA Smart Data Integration“ gibt es eine SAP-interne Schnittstelle zum Datenaustausch. Diese setzt lediglich voraus, dass auf dem gewünschten Quellsystem SAPs „Data Provisioning Agent“ installiert ist.

Benutzerfreundlichkeit

Was die Benutzerfreundlichkeit angeht, ist DWC zwischen Azure Synsapse Analytics und Redshift angesiedelt. Die Weboberfläche ist aufgeräumt und übersichtlich. Die ver-

[8] „SAP HANA on AWS“: https://aws.amazon.com/sap/solutions/saphana/

[9] „SAP HANA on GCP“: https://cloud.google.com/solutions/sap/docs/overview-of-sap-on-google-cloud

[10] „SAP HANA on Azure“: https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/sap/hana-overview-architecture

[11] Die vollständige Übersicht über vorhandene Schnittstellen findet sich hier: https://help.sap.com/viewer/9f804b8efa8043539289f42f372c4862/cloud/en-US/eb85e157ab654152bd68a8714036e463.html

schiedenen integrierten Funktionen lassen sich zu einem großen Teil ohne Programmieraufwand mithilfe graphischer Oberflächen erledigen.

Der „Data Builder“ ist ein simples Datenbank-Entwicklungswerkzeug, mit dem sich einfach Tabellen erstellen, bearbeiten und mit fachlichen Metadaten annotieren lassen. Außerdem können einfach CSV-Dateien hochgeladen und daraus neue Tabellen erstellt werden.

Bei der Erstellung von Views lässt DWC dem Benutzer die Wahl, ob er sie händisch im „SQL View Builder“ oder nach dem Baukastensystem mit vorgefertigten Bausteinen im „Graphical View Builder“ zusammenklicken möchte.

„Data Flow“ ist ähnlich wie der Graphical View Builder ein visueller Editor nach dem Drag-and-Drop-Prinzip, der die einfache Konfiguration von ETL-Prozessen und Datentransformationen ermöglicht.

Metadaten

In DWC gibt es vielseitige Möglichkeiten, Metadaten zu bearbeiten und anzeigen zu lassen. Grundsätzlich wird immer in einem „Space“, also einem virtuellen Arbeitsbereich, gearbeitet. Nutzer haben verschiedene Berechtigungen in einem Space, vom Administrator über den Entwickler bis zum Konsumenten. So sind die Daten vor unbefugten Zugriffen geschützt.

Zu jedem Space wird der Ressourceverbrauch erhoben. DWC zeigt dem Administrator automatisch auf, wo heiße und kalte Daten im DW vorliegen, sodass er erwägen kann, die jeweiligen Daten in andere Speichertechniken umzuschichten.

Im „Business Catalogue“ werden automatisch alle Tabllen und Views aufgeführt. Dort können Ursprung und Struktur der Daten schnell in Erfahrung gebracht werden. Um ein noch besseres Verständnis des Unternehmensdatenmodells zu bekommen und die Zusammenhänge zwischen Tabellen im Data Warehouse zu dokumentieren, gibt es in DWC einen graphischen ER-Diagramm-Editor.

Dokumentation und Weiterbildung

SAPs Dokumentation ist ausführlich und größtenteils auch auf Deutsch verfügbar. Sie ist nach Produkten aufgeteilt und klar strukturiert.

Für weitere Informationen zu neuen Entwicklungen, Trends und Produktankündigungen lohnt sich das Verfolgen des SAP-Blogs[12], auf dem Mitarbeiter eine große Menge hochwertiger Artikel veröffentlichen.

Auf der Seite von „SAP DWC Events“[13] werden regelmäßig neue Workshops angekündigt,die sowohl live im Netz oder on-demand konsumiert werden können.

[12] https://blogs.sap.com/

[13] https://www.sap.com/products/data-warehouse-cloud/events.html

	Synapse Analytics	**Redshift**	**DWC**
Cloudnativ	++	++	++
Big Data	+	+	o
Schnittstellen	++	+	+
Benutzerfreundlichkeit	++	-	+
Metadaten	++	o	++
Kosten	+	++	- -
Dokumentation	++	+	+

Tabelle 4.1: Bewertung der DW-Lösungen

Kosten

Die Betriebskosten sind eine Schattenseit von SAP DWC. Zuerst einmal gibt es nur eine sehr begrenzte Testversion, die nach 30 Tagen abläuft[14]. Ein unbegrenztes kostenloses Kontingent, wie man es mittlerweile von anderen Anbietern gewohnt ist, gibt es nicht. Immerhin sind bei einem DWC-Vertrag auch automatisch 5 Lizenzen für SAP Analytics Cloud enthalten.

Die Betriebskosten setzen sich aus fixen Kosten für gemieteten Speicher und aus den verwendeten „Capacity Units“, also den verbrauchten Hardwareressourcen, zusammen. Die Schätzung der Betriebskosten ist etwas kompliziert, deswegen bietet SAP einen Kostenrechner für DWC an[15].

Selbst bei der niedrigsten Konfiguration ohne Data Lake und mit der schwächsten Hardware ergeben sich geschätzte monatliche Betriebskosten von etwa 4.300€. Im Vergleich zu anderen Anbieter sind die Kosten damit verhältnismäßig hoch.

4.4 Abschließender Vergleich

Nachdem die DWaaS-Lösungen von Azure, AWS und SAP im Detail untersucht wurden, soll nun eine abschließender Vergleich vorgenommen werden.

Von den vorgestellten Produkten ist Azure Synapse Analytics die insgesamt ausgereifteste Lösung. Dank des einfachen visuellen Baukastensystems, das vielfältig genutzt werden kann, und der hervorragenden Dokumentation ist die Einarbeitung und das Erstellen eines ersten Prototypen relativ einfach. Selbst Vertriebsmitarbeiter könnten mit überschaubarem Aufwand zu DW-Entwicklern umgeschult werden.

Amazon Redshift ist weniger intuitiv und keine bequeme All-in-One-Lösung, bietet dafür aber eine äußerst effiziente und stabile Infrastruktur. Redshift selbst ist eine sehr minimalistische Data-Warehouse-Software, die erst in Kombination mit weiteren Werkzeugen aus dem AWS-Ökosystem ihr volles Potential entfaltet. Der Einsatz von Amazon

[14] Details zur Testversion: https://www.sap.com/products/data-warehouse-cloud/trial.html
[15] https://www.sap.com/dmc/exp/2020-01-datawarehousecloudcalculator/index.html

Redshift liegt nahe, wenn bereits AWS-Dienste wie S3, Kinesis oder AWS Lamda genutzt werden und AWS-Fachwissen im Unternehmen vorhanden ist.

SAP Data Warehouse Cloud ist die in diesem Vergleich noch am wenigsten ausgereifte Data-Warehouse-Lösung. DWC wird heute vor allem für hybride Lösungen genutzt, bei denen ein Teil der Unternehmensdaten in einem On-Premise-Data-Warehouse wie SAP BW/4HANA liegt und nun die Vorzüge der neuen Cloudplattform ausgetestet werden. Für Unternehmen, die bisher kaum in SAP-Produkte investiert sind, ist DWC jedoch alleine schon aufgrund des Kostenpunkts eher uninteressant.

Kapitel 5

Zukünftige Trends

Es zeichnen sich im Bereich Data Warehousing auf verschiedenen Ebenen neue Entwicklungen ab. Auf der Hardwareebene profitieren Data Warehouses naturgemäß besonders stark von schnelleren und/oder günstigeren Speichertechnologien. Aber auch Neuerungen der Prozessor- und Netzwerktechnik können sich positiv auf Durchsatz und Latenz auswirken. Wie in Kapitel 2 dargelegt, können bestimmte Technikspränge ganze Paradigmenwechsel einleiten.

Auf der Softwareebene gab es in den letzten Jahren ebenfalls Fortschritte hinsichtlich paralleler Verarbeitung, verteilter Speicherung und damit verbundenen Datenzugriffsarten.

DWs sind immer nur ein Mittel zum Zweck und müssen daher stets aktuellen und sich andeutenden Anforderungen gewachsen sein. Da es sich um große, komplexe Systeme handelt, ist es besonders wichtig, langfristig zu planen und mögliche Szenarien zu berücksichtigen. Verspätete oder unüberlegte Umstrukturierungen können Ressourcen verschwenden, zum Wettbewerbsnachteil werden und dem Unternehmen unterm Strich schaden.

Es werden nun einige Entwicklungen vorgestellt, die möglicherweise das Potential haben, zukünftig weite Verbreitung zu finden. Es wird diskutiert, für welche Anwendungsfälle diese interessant sind und inwieweit alte Technologien dadurch abgelöst werden könnten.

5.1 Data Warehouses mit NoSQL

NoSQL-Datenbanken sind heutzutage sehr beliebt und eignen sich aufgrund verschiedener Faktoren sehr gut für Big-Data-Anwendungen.

Ein großer Vorteil von NoSQL-Datenbanken im Vergleich zu relationalen DBMS ist die Tatsache, dass sie sehr stark skalieren können. Alle gängigen NoSQL-Lösungen wie Apache Cassandra, MongoDB oder Amazon DynamoDB unterstützen Skalierung mittels automatischer horizontaler Partitionierung (*Sharding*). Zudem ist die Zugriffsgeschwindigkeit bei nichtrelationalen Datenbanken tendenziell höher[42]. Beides durchaus wünschenswerte Eigenschaften im Hinblick auf Data Warehousing.

Da die Daten in NoSQL-Datenbanken im Gegensatz zu relationalen Datenbanken nicht an starre Schemata gebunden sind, ist NoSQL perfekt für die Speicherung unstrukturierter Daten gemacht. Genau diese nehmen immer größeren Platz in DWs ein.

Die Schemalosigkeit führt auch dazu, dass die Entwicklung im DW einfacher und schneller ablaufen kann, da dokumentbasierte Modellierung oft intuitiver und flexibler ist[43]. Änderungen an relationalen Tabellen beispielsweise sind vergleichsweise mühsam.

Die Flexibilität von NoSQL ist allerdings gleichzeitig ein Nachteil: Nichtrelationale Datenmodelle weisen redundante Daten bzw. duplizierte Datensätze auf. Dies ist alleine schon dann unvermeidbar, wenn relationale Daten im Zuge der Ingestion denormalisiert werden müssen. Dadurch erhöht sich nicht nur der Speicherbedarf. Kennzahlen können verfälscht werden und das Ändern von Datensätzen wird ebenfalls aufwändiger.

NoSQL-Systeme sind also nicht uneingeschränkt für Data Warehousing geeignet. Sie lassen sich vor allem dann gut einsetzen, wenn es darum geht, hohe Volumen einfacher Transaktionen zu verarbeiten. So nutzt die Bundesagentur für Arbeit seit 2018 Apache Cassandra zur hochverfügbaren Speicherung interner Betriebsdaten[44]. In dem Fall werden computergenerierte Statusdaten mit Cassandra gespeichert, sodass sie mit Apache Solr und Grafana überwacht werden können.

Komplette NoSQL-DWs sind in der Praxis zwar ungewöhnlich, bei Bedarf lassen sie sich jedoch umsetzen, beispielsweise mit der DW-Software Apache Hive, die auf das spaltenbasierte DBMS Apache HBase[1] aufbaut.

Allgemein ist NoSQL in kleinen und mittleren Unternehmen einsetzbar, in denen die Daten eher einfach strukturiert sind und der erhöhte, redundante Speicherbedarf nicht all zu sehr ins Gewicht fällt. Außerdem ist der Einsatz von NoSQL-Technologien dann sinnvoll, wenn es sich um sehr große Datenmengen handelt, bei denen ein beträchtlicher Teil auf semi- oder unstrukturierte Daten entfällt. Im Zuge von Industrie 4.0 werden viel deutsche Firmen mit ebendiesen Anforderungen konfrontiert werden.

5.2 Real Time Analytical Processing

Derzeit ist es üblich, dass DWs in regelmäßigen Abständen Datenaktualisierungen durchführen, wobei sie neue Daten aus den Quellsystemen aufnehmen. Diese inkrementellen Ladevorgänge finden oft alle 24 Stunden außerhalb der Geschäftszeiten statt. Auf dieser Datengrundlage sind folglich höchstens tagesgenaue Auswertungen möglich, was im Vertrieb und anderen Abteilungen für viele Szenarien akzeptabel ist.

In anderen Fällen hingegen besteht der Wunsch nach Echtzeit-Analysen, wobei im Idealfall alle Daten aus den operativen Systemen ohne Verzug analytischen Systemen zur Verfügung stehen. Zu den Anwendungsfällen, bei denen Echtzeitanalysen erforderlich sind, gehören unter anderem Social-Media-Marketing, die Auswertung von Telekommunikationsdaten und die Überwachung von Produktionsabläufen. Echtzeitanalysen gewinnen vor allem durch die rasche Verbreitung eingebetteter Systeme und IoT-Geräte stark an Bedeutung.

[1] https://hbase.apache.org/

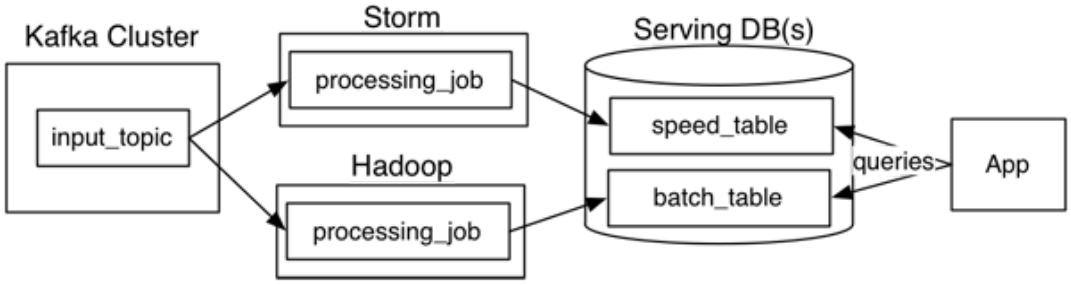

Abbildung 5.1: Beispielhafte Lambda-Architektur mit Apache Kafka, Apache Storm und Apache Hadoop. Entnommen aus [45].

Eine Lösung bietet die sogenannte Lambda-Architektur (vgl. Abb. 5.1): Hierbei speist ein Messagebroker, wie etwa Apache Kafka, eingehende Daten in Echtzeit in zwei verschiedene Subsysteme ein: einen *Hot Path* und einen *Cold Path*. Im Hot Path stehen Daten zum Monitoring mit niedriger Latenz zur Verfügung, dafür aber auch nur in einem eingeschränkten Zeitfenster. Im Cold Path werden sie hingegen dauerhaft verlustfrei gespeichert, sodass sie für spätere Auswertungen zur Verfügung stehen.

Bei den Echtzeitdaten handelt es sich meist um einen Strom kleiner, automatisch erzeugter Daten (*Logs*), z.B. aus Sensoren oder anderen Softwaresystemen. Diese können bei Bedarf als historische Daten in das DW integriert werden, um zu einem späteren Zeitpunkt weitergehende Analysen im Batch-Betrieb darauf durchzuführen.

Nachteil der Lambda-Architektur ist der hohe Implementierungsaufwand[45]. Mehrere unterschiedliche (Speicher-)Systeme müssen miteinander integriert werden, wodurch nachträgliche Änderungen gleichermaßen komplexer werden. Abhilfe kann hier die etwa vereinfachte Kappa-Architektur schaffen.

5.3 Logical Data Warehouse

Technische Umsetzungen von Data Warehouses werden seit jeher stark vom jeweiligen Stand der Technik geprägt. So ist die bis heute übliche Trennung von operativen und analytischen Systemen maßgeblich hardwarebedingt. Ebenso wie frühere Architekturen auf die Einführung bahnbrechender Technologien wie der Festplatte oder des Multicore-Prozessors regierten, müssen moderne DWs neue Entwicklungen berücksichtigen.

Soweit es heute absehbar ist, werden kukünftige Architekturen auf der Hardwareebene von nichtflüchtigem Hauptspeicher (NVM-Geräte) sowie verschiedenen spezialisierten Recheneinheiten (Manycore-Prozessoren, Feldrechnern etc.) geprägt sein[46]. Rechnersysteme werden in Kombination mit entsprechend angepasster Software in der Lage sein, bei niedriger Latenz deutlich höhere, parallele Lese- und Schreibgeschwindigkeiten zu ermöglichen und somit heutige Beschränkungen aufzuheben. In der Praxis könnten somit in den kommenden Jahren und Jahrzenten OLTP- und OLAP-Systeme zu so genannten *OLTAP-Systemen* verschmelzen.

Falls diese Vorhersagen eintreffen, werden sie sich aber auf absehbare Zeit vermutlich nur in bestimmten Subsystemen durchsetzen. Konkret würde das bedeuten, dass Data-Warehouse-Architekturen insgesamt heterogener werden. Dies bestätigt die bereits heute vorherrschende Meinung, dass die Zukunft von DWs nicht im starren, monolithischen Enterprise Data Warehouse liegt, sondern vielmehr in einer „logischen Informationsbereitstellungsplattform“[47], also einem *Logischen Data Warehouse* (LDW).

Beim LDW handelt es sich um eine intern heterogene Analyseinfrastruktur, die extern über einfache Analysewerkzeuge abgefragt werden kann, sodass auch betriebliche Endanwender eigenständig Datenanalysen durchführen können (*Self-Service Analytics*). Ob die zugrundeliegenden Daten in dem Fall aus verteilten MapReduce-Berechnungen, Data Mining oder klassischen SQL-Abfragen stammen, ist für den Endanwender irrelevant.

Das Konzept des Logischen Data Warehouse wurde von Inmon[48] bereits 2004 im Kontext föderierter Data Warehouses erwähnt und 2011 erneut von Beyer[47] im erweiterten Sinn aufgegriffen. LDWs spiegeln verschiedene Aspekte der modernen Cloud-Computing-Philosophie wider, etwa Virtualisierung, Heterogenität und Verteiltheit. Der LDW-Ansatz wird von DW-Anbietern bereits umgesetzt, wie Microsofts Azure Data Factory oder Amazons Redshift Spectrum beweisen.

Für Endanwender bergen LDWs das Potential, einen einfacheren, demokratischen Zugang zu Daten zu erhalten[49]. Wenn die Vision der Self Serving Analysis tatsächlich weitgehend umgesetzt werden kann, hat dies sowohl Auswirkungen auf die Unternehmensorganisation als auch auf betroffene Hierarchien und Berufsbilder.

Kapitel 6

Fazit

Die Behauptung, Data Warehousing sei veraltet oder nicht mehr zeitgemäß ist schlicht falsch. Im Gegenteil, die immer heterogeneren Datenquellen und noch nicht angezapften Datenbestände zeigen erst recht, dass moderne Data Warehouses unverzichtbar sind.

Die technische Umsetzung der Data Warehouses muss mit dieser Entwicklung mithalten. Die Vergleiche der verschiendenen kommerziellen DW-Lösungen haben gezeigt, dass es zwar verschiedene Schwerpunkte bei der Implementierung gibt, die grundsätzliche Stoßrichtung aber stets die selbe ist: Ein modernes Data Warehouse muss cloudbasiert und modular sein und mit möglichst vielen verschiedenen Speichertechniken und Werkzeugen kombinierbar sein. Der DW-Entwickler muss in der Lage sein, seine Ressourcennutzung zu kontrollieren und überwachen, bei Bedarf Daten aus externem Speicher zu laden oder weniger relevante Daten in externen Speichersystemen auf Halde zu legen.

Der Trend geht dahin, dass aufwändige bzw. teure Datenmigrationen immer seltener nötig sind. Es kommen zunehmend Virtualisierungstechniken zum Einsatz. Es ist immer seltener nötig sich um Speicherzugriffszeiten Sorgen zu machen.

Die technischen Aspekte des Data Warehouse sind allgemein immer weniger für den einfachen DW-Entwickler von Belang. Da in einem beträchtlichen Teil der Fälle keine besonderen Programmierkenntnisse mehr zur Anpassung des Data Warehouse nötig sind, werden Vertriebsmitarbeiter oder Vetreter anderer nicht-technischer Fachbereiche zunehmend in das Data Warehousing miteinbezogen. Das Arbeiten am Data Warehouse ist agil und interdisziplinär.

Nur weil Data Warehouses in der Cloud besonders flexibel sind und unter Umständen deutlich günstigere Betriebskosten aufweisen, heißt das aber noch lange nicht, dass die Migration auf ein solches System automatisch zum Erfolg wird.

Wie der historische Rückblick gezeigt hat, spielen weiche Faktoren auch eine entscheidende Rolle beim Erfolg solcher Umstellungen. Dazu gehören beispielsweise realistische Erwartungen an neue DW-Software. Inmon sprach schon davon, dass ein Data-Warehouse-Prototyp bereits ein Erfolg ist, wenn er zu 50% richtige Ergebnisse liefert[50]. Diese Faustregel gilt bis heute.

Gerade heute ist es für Unternehmen wichtig, Kompetenzen in ihren eigenen IT-

Abteilungen in Sachen Data Warehousing und Datenmanagement aufzubauen, um nicht ständig von externen Dienstleistern abhängig zu sein.

Dadurch, dass moderne DW-Lösungen den Entwicklern noch mächtigere Werkzeuge an die Hand geben, wird die Position des DW-Entwicklers aufgewertet. Er bekommt mehr Mitspracherecht bei strategischen Entscheidungen[51] und wirkt aktiv an der Verbesserung der eigenen IT-Landschaft mit.

Literatur

[1] Handelskammer Hamburg. "Aufbewahrungsfristen von Geschäftsunterlagen." (2021), Adresse: https://www.hk24.de/produktmarken/beratung-service/recht_und_steuern/steuerrecht/abgabenrecht/aufbewahrungsfristen-geschaeftsunterlagen/1157174 (besucht am 19. 07. 2021).

[2] A. Bauer und H. Günzel, *Data-Warehouse-Systeme: Architektur, Entwicklung, Anwendung*. dpunkt.verlag, 2013.

[3] W. H. Inmon, *Building the Data Warehouse*. John Wiley & Sons, 2005.

[4] M. A. Rashid, L. Hossain und J. D. Patrick, "The evolution of ERP systems: A historical perspective," in *Enterprise resource planning: Solutions and management*, IGI global, 2002, S. 35–50.

[5] S. Whittle. "Extended ERP: The path of least resistance?" ZDNet. (12. Dez. 2003), Adresse: https://www.zdnet.com/article/extended-erp-the-path-of-least-resistance/ (besucht am 19. 07. 2021).

[6] R. K. Rainer Jr, C. A. Snyder und H. J. Watson, "The evolution of executive information system software," *Decision Support Systems*, Jg. 8, Nr. 4, S. 333–341, 1992.

[7] N. McBride, "The rise and fall of an executive information system: a case study," *Information Systems Journal*, Jg. 7, Nr. 4, S. 277–287, 1997. DOI: https://doi.org/10.1046/j.1365-2575.1997.00021.x. Adresse: https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-2575.1997.00021.x.

[8] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, Jg. 21, Nr. 5, S. 61–72, 1988, Publisher: IEEE.

[9] J. Highsmith. "History: The agile manifesto," Agile Alliance. (2001), Adresse: http://agilemanifesto.org/history.html (besucht am 21. 07. 2021).

[10] S. Choo Quan. "Basic terminology of data warehousing (DW) for business intelligence (BI)," Design & Execute. (16. Sep. 1999), Adresse: http://www.designandexecute.com/designs/basics-of-data-warehouse-dw/ (besucht am 15. 09. 2021).

[11] T. Hafen. "Das Data Warehouse wandert in die Cloud," com! professional. (2. Juli 2018), Adresse: https://www.com-magazin.de/praxis/business-it/data-warehouse-wandert-in-cloud-1550048.html (besucht am 22. 07. 2021).

[12] C. Goldman. “Snowflake debuts at $245 a share in biggest software IPO ever.” (16. Sep. 2020), Adresse: https://www.thestreet.com/markets/ipos/snowflake-debuts-at-tk-a-share-in-biggest-software-ipo-ever (besucht am 03. 09. 2021).

[13] SAP. “What Is Cloud ERP?” SAP Insights. (2021), Adresse: https://insights.sap.com/what-is-cloud-erp/ (besucht am 26. 07. 2021).

[14] C. Bange und W. Eckerson. “BI und Datenmanagement in der Cloud: Treiber, Nutzen und Herausforderungen.” (Feb. 2017), Adresse: https://barc.de/docs/bi-und-datenmanagement-in-der-cloud-treiber-nutzen-und-herausforderungen (besucht am 26. 07. 2021).

[15] Microsoft. “Use read-only replicas to offload read-only query workloads,” Microsoft Docs. (6. Juli 2021), Adresse: https://docs.microsoft.com/en-us/azure/azure-sql/database/read-scale-out (besucht am 26. 07. 2021).

[16] Amazon Web Services, *Amazon redshift service level agreement*, 19. März 2019. Adresse: https://aws.amazon.com/redshift/sla/ (besucht am 08. 09. 2021).

[17] Snowflake, *Support policy and service level agreement*, 20. Feb. 2019. Adresse: https://www.snowflake.com/wp-content/uploads/2017/07/Snowflake-Computing-Standard-Support-Policies-and-SLA-2015-05-26.pdf (besucht am 08. 09. 2021).

[18] SAP, *Service level agreement for SAP cloud services*, 2021. Adresse: https://www.sap.com/docs/download/agreements/product-use-and-support-terms/cls/en/service-level-agreement-for-sap-cloud-services-english-v8-2021.pdf (besucht am 08. 09. 2021).

[19] Google Cloud, *BigQuery service level agreement (SLA)*, 1. Juli 2021. Adresse: https://cloud.google.com/bigquery/sla (besucht am 08. 09. 2021).

[20] Microsoft, *SLA for azure synapse analytics*, Aug. 2021. Adresse: https://azure.microsoft.com/en-us/support/legal/sla/synapse-analytics/v1_1/ (besucht am 08. 09. 2021).

[21] T. Priebe, A. Reisser und D. Hoang, “Reinventing the Wheel Why Harmonization and Reuse Fail in Complex Data Warehouse Environments and a Proposed Solution to the Problem.,” Jan. 2011, S. 38.

[22] E. Dumbill. “What is big data? an introduction to the big data landscape.,” O’Reilley Radar. (11. Jan. 2012), Adresse: http://radar.oreilly.com/2012/01/what-is-big-data.html (besucht am 15. 09. 2021).

[23] A. Schlaucher. “Einsatz von In-Memory-Datenbanken zur Virtualisierung im Data Warehouse,” Informatik Aktuell. (11. Aug. 2015), Adresse: https://www.informatik-aktuell.de/betrieb/virtualisierung/einsatz-von-in-memory-datenbanken-zur-virtualisierung-im-data-warehouse.html (besucht am 17. 08. 2021).

[24] J. Dean und S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, San Francisco, CA, 2004, S. 137–150.

[25] A. Kirillov. "Apache spark: Core concepts, architecture and internals." (3. März 2016), Adresse: http://datastrophic.io/core-concepts-architecture-and-internals-of-apache-spark/ (besucht am 03. 09. 2021).

[26] P. Russom, *TDWI pulse report: Accelerating data warehouse development*, 2021.

[27] A. Ulagaratchagan. "Microsoft named a leader in the 2021 gartner magic quadrant for analytics and BI platforms," Microsoft Power BI Blog. (18. Feb. 2021), Adresse: https://powerbi.microsoft.com/en-us/blog/microsoft-named-a-leader-in-2021-gartner-magic-quadrant-for-analytics-and-bi-platforms/ (besucht am 02. 09. 2021).

[28] Microsoft. "Data warehousing and analytics," Microsoft Docs. (2021), Adresse: https://docs.microsoft.com/en-us/azure/architecture/example-scenario/data/data-warehouse (besucht am 06. 09. 2021).

[29] ——, "Introducing data virtualization with PolyBase," Microsoft Docs. (23. März 2021), Adresse: https://docs.microsoft.com/en-us/sql/relational-databases/polybase/polybase-guide?view=sql-server-ver15.

[30] ——, "Serverless SQL pool in azure synapse analytics," Microsoft Docs. (15. Apr. 2020), Adresse: https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/on-demand-workspace-overview.

[31] ——, "What is Azure Synapse Link for Azure Cosmos DB?" Microsoft Docs. (12. Juli 2021), Adresse: https://docs.microsoft.com/en-us/azure/cosmos-db/synapse-link.

[32] ——, "Azure data factory and azure synapse analytics connector overview," Microsoft Docs. (9. Sep. 2021), Adresse: https://docs.microsoft.com/en-us/azure/data-factory/connector-overview.

[33] M. Kleider. "Amazon redshift spectrum extends data warehousing out to exabytes—no loading required," AWS Big Data Blog. (21. Juli 2017), Adresse: https://aws.amazon.com/blogs/big-data/amazon-redshift-spectrum-extends-data-warehousing-out-to-exabytes-no-loading-required/ (besucht am 14. 09. 2021).

[34] ——, "Data Compression Improvements in Amazon Redshift Bring Compression Ratios Up to 4x," AWS News Blog. (6. Apr. 2017), Adresse: https://aws.amazon.com/blogs/aws/data-compression-improvements-in-amazon-redshift/ (besucht am 04. 09. 2021).

[35] Y. Dolan, A. Gendler und V. Gromakowski. "Extend your Amazon Redshift Data Warehouse to your Data Lake," AWS Big Data Blog. (24. Juni 2020), Adresse: https://aws.amazon.com/blogs/big-data/extend-your-amazon-redshift-data-warehouse-to-your-data-lake/ (besucht am 04. 09. 2021).

[36] S. Das, A. Dias und V. Parampally Vijayakumar. "Scheduling SQL queries on your amazon redshift data warehouse," AWS Big Data Blog. (22. Dez. 2020), Adresse: https://aws.amazon.com/blogs/big-data/scheduling-sql-queries-on-your-amazon-redshift-data-warehouse/ (besucht am 15. 09. 2021).

[37] Amazon Web Services. "Transform data with AWS glue DataBrew," Analytics on AWS. (2021), Adresse: https://intro-to-analytics-on-aws.workshop.aws/en/lab-guide/transform_glue_databrew.html (besucht am 16. 09. 2021).

[38] SAP. "SAP Company Information," About SAP. (2021), Adresse: https://www.sap.com/about/company.html (besucht am 13. 09. 2021).

[39] C. Kampmann. "SAP data warehouse cloud – the end of the SAP BW?!" SAP Blogs. (29. Sep. 2020), Adresse: https://blogs.sap.com/2020/09/29/sap-data-warehouse-cloud-the-end-of-the-sap-bw/ (besucht am 15. 09. 2021).

[40] A. Pattanayak, "Data Virtualization with SAP HANA Smart Data Access," *Journal of Computer and Communications*, Jg. 05, S. 62–68, Jan. 2017. DOI: 10.4236/jcc.2017.58005.

[41] Google Cloud. "Google cloud and SAP partner to accelerate business transformations in the cloud," Google Cloud Press Releases. (29. Juli 2021), Adresse: https://cloud.google.com/press-releases/2021/0729/sap-google-cloud (besucht am 15. 09. 2021).

[42] R. Wang und Z. Yang, *SQL vs NoSQL: A performance comparison*, 2017. Adresse: https://www.cs.rochester.edu/courses/261/fall2017/termpaper/submissions/06/Paper.pdf.

[43] I. Oditis, Z. Bicevska, J. Bicevskis und G. Karnitis, "Implementation of NoSQL-based data Warehouses," *Baltic Journal of Modern Computing*, Jg. 6, Nr. 1, S. 45–55, 2018, Publisher: University of Latvia.

[44] E. Sultanow, O. Weiß und M. Seßler, "Echtzeitmonitoring-Architektur mit Cassandra und Solr," *OBJEKTspektrum*, Nr. 6, S. 40–45, 2018.

[45] J. Kreps. "Questioning the lambda architecture," O'Reilley Radar. (2. Juli 2014), Adresse: https://www.oreilly.com/radar/questioning-the-lambda-architecture/.

[46] W. Pan, Z. Li, Y. Zhang und C. Weng, "The New Hardware Development Trend and the Challenges in Data Management and Analysis," *Data Science and Engineering*, Jg. 3, Nr. 3, S. 263–276, 1. Sep. 2018, ISSN: 2364-1541. DOI: 10.1007/s41019-018-0072-6. Adresse: https://doi.org/10.1007/s41019-018-0072-6.

[47] M. Adrian. "Mark beyer, father of the logical data warehouse, guest post," Gartner Blog Network. (3. Nov. 2011), Adresse: https://blogs.gartner.com/merv-adrian/2011/11/03/mark-beyer-father-of-the-logical-data-warehouse-guest-post/ (besucht am 01. 09. 2021).

[48] B. Inmon, "The Logical Data Warehouse," *Information Management*, Jg. 14, Nr. 6, S. 67, 2004, Publisher: SourceMedia.

[49] M. Iurillo. "The next generation logical data warehouse: It's time to democratize the data," Dataversity. (27. Feb. 2018), Adresse: https://www.dataversity.net/next-generation-logical-data-warehouse-time-democratize-data/ (besucht am 01. 09. 2021).

[50] W. H. Inmon, *Building the Data Warehouse,3rd Edition*, 3rd. USA: John Wiley & Sons, Inc., 2002, ISBN: 0-471-08130-2.

[51] M. Albertson. "Competitive race in the cloud data warehousing market gets interesting," SiliconANGLE. (21. Dez. 2020), Adresse: https://siliconangle.com/2020/12/21/competitive-race-in-the-cloud-data-warehousing-market-is-beginning-to-get-interesting-cubeconversations/ (besucht am 03. 09. 2021).