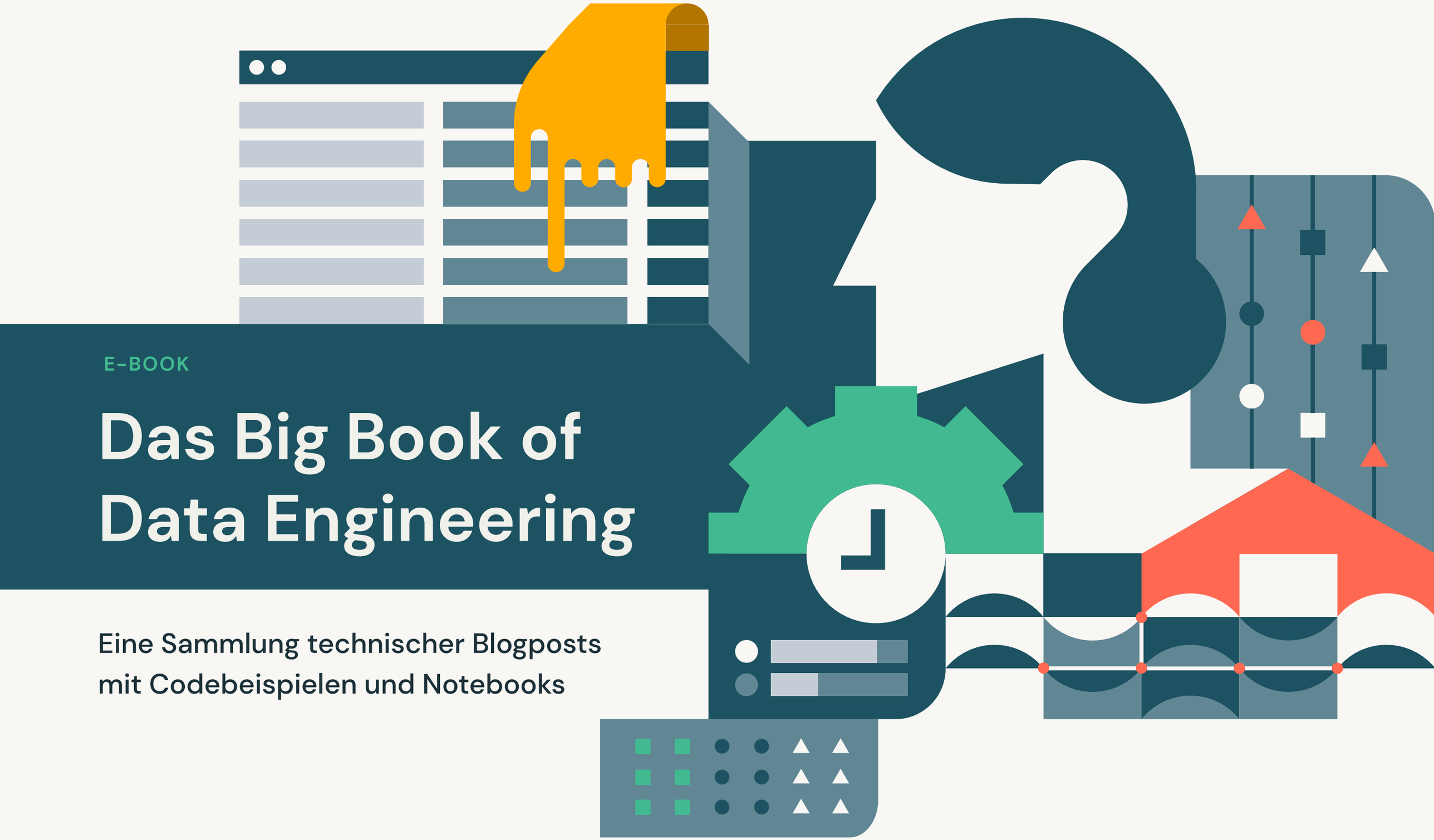
E-BOOK

Das Big Book of Data Engineering

Eine Sammlung technischer Blogposts
mit Codebeispielen und Notebooks



databricks

Inhalt

databricks

KAPITEL

01

Einführung in das Data Engineering auf Databricks

databricks

Unternehmen wissen inzwischen, welchen Wert Daten als strategische Ressource für verschiedene geschäftsbezogene Initiativen haben. So lassen sich mit Daten beispielsweise Umsätze steigern, das Kundenerlebnis optimieren, der Betrieb effizienter gestalten oder Produkte und Services verbessern. Allerdings ist die Komplexität beim Zugriff auf diese Daten und ihre Verwaltung deutlich gestiegen. Dies ist vor allem auf die explosionsartige Zunahme der Datenvolumina und -typen zurückzuführen: Die bei den Unternehmen angehäuften Daten liegen Schätzungen zufolge zu 80 % in unstrukturierter und halbstrukturierter Form vor. Und da stetig mehr und mehr Daten erfasst werden, bleiben 73 % davon für Analysen und Entscheidungen schlicht ungenutzt. Data-Engineering-Teams sind nun gefordert, Datenpipelines zur effizienten und zuverlässigen Bereitstellung von Daten zu erstellen und diesen brachliegenden Anteil zu reduzieren und mehr Daten nutzbar zu machen. Allerdings ist der Aufbau solcher komplexer Datenpipelines mit einer Reihe von Herausforderungen verbunden:

- Damit die Daten in den Data Lake kommen, müssen Data Engineers extrem viel Zeit aufwenden, um bei der Datenaufnahme regelmäßig durchzuführende Tasks manuell zu programmieren.
- Da auch Datenplattformen stetigem Wandel unterworfen sind, verbringen Data Engineers viel Zeit mit Aufbau und Wartung einer komplexen skalierbaren Infrastruktur, nur um diese am Ende wieder neu erstellen zu müssen.
- Mit der zunehmenden Bedeutung von Echtzeitdaten werden latenzarme Datenpipelines benötigt, die noch schwieriger zu erstellen und zu warten sind.
- Schließlich müssen, sobald alle Pipelines fertig geschrieben sind, die Data Engineers die Leistung ständig im Blick behalten und Pipelines und Architekturen so justieren, dass die SLAs eingehalten werden.

Wie kann Databricks helfen?

Mit der Databricks Lakehouse-Plattform erhalten Data Engineers Zugang zu einer datentechnischen End-to-End-Lösung, die alle Phasen abdeckt: Erfassung, Transformation, Verarbeitung, Zeitplanung und Bereitstellung der Daten. Die Lakehouse-Plattform automatisiert die komplexen Abläufe bei Aufbau und Pflege von Pipelines wie auch bei der direkten Ausführung von ETL-Workloads für einen Data Lake. Damit erhalten Data Engineers endlich die Möglichkeit, den Schwerpunkt auf Qualität und Zuverlässigkeit legen und so wertvolle Erkenntnisse gewinnen zu können.

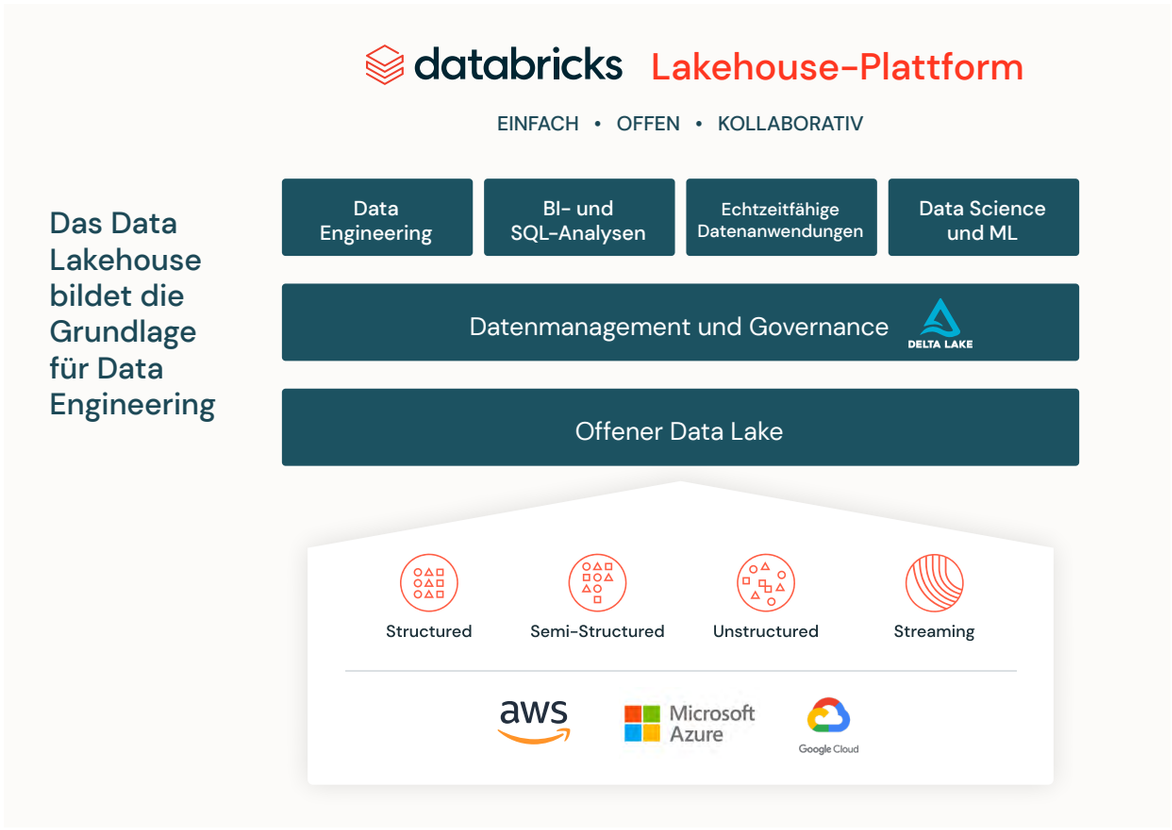


Abbildung 1
Die Databricks Lakehouse Plattform führt Ihre Daten, Analysen und KI auf einer gemeinsamen Plattform für alle datentechnischen Anwendungsfälle zusammen

Wichtige Unterscheidungsmerkmale für erfolgreiches Data Engineering mit Databricks

Da die Lakehouse-Architektur eine Vereinfachung darstellt, brauchen die Data Engineers einen Ansatz zum Erstellen von Datenpipelines, der den Anforderungen von Unternehmen gerecht wird. Ein Data-Engineering-Lösungsteam, das erfolgreich sein will, muss die folgenden acht Schlüsselkompetenzen vorweisen:

Fortlaufende oder geplante Datenerfassung

Data Engineers, die viele Petabyte Daten mit sich automatisch entwickelnden Schemata erfassen können, sind in der Lage, Daten für Analysen, Data Science oder Machine Learning schnell, zuverlässig, skalierbar und automatisch bereitzustellen. Dies umfasst folgende Vorgänge:

- Inkrementelles und effizientes Verarbeiten von Daten, die aus Dateien oder Streaming-Quellen wie Kafka, DBMS und NoSQL empfangen werden
- Automatisches Ableiten des Schemas und Erkennen von Spaltenwechseln bei strukturierten und unstrukturierten Datenformaten
- Automatisches und effizientes Überwachen der eingehenden Daten ohne manuelle Eingriffe
- Beibehalten der Datenspalten zur Vermeidung von Datenverlust

Deklarative ETL-Pipelines

Data Engineers können Entwicklungszeit und -aufwand reduzieren und den Schwerpunkt stattdessen auf die Implementierung von Geschäftslogik und Datenqualitätsprüfungen innerhalb der Datenpipeline mit SQL oder Python legen. Dies lässt sich auf folgende Weise erreichen:

- Durch absichtsgesteuerte deklarative Entwicklung, um das „Wie“ zu vereinfachen und das zu lösende „Was“ zu definieren.
- Durch automatisches Erstellen hochwertiger Datenherkunftsinformationen und das Verwalten von Tabellenabhängigkeiten in der gesamten Datenpipeline
- Durch automatisches Überprüfen auf fehlende Abhängigkeiten oder Syntaxfehler und das Verwalten der Datenpipeline-Wiederherstellung

Validierung und Überwachung der Datenqualität

Erhöhen Sie mit den folgenden Maßnahmen die Zuverlässigkeit der Daten im gesamten Data Lakehouse, damit die Datenteams sich bei nachgelagerten Initiativen auf die Qualität der Informationen verlassen können:

- Definieren von Kontrollmechanismen für Datenqualität und -integrität innerhalb der Pipeline mit definierten Datenerwartungen
- Behandeln von Datenqualitätsfehlern auf Basis vordefinierter Richtlinien (Qualitätsmangel, Qualitätsabfall, Warnung, Quarantäne)
- Einsetzen der Datenqualitätsmetriken, die für die gesamte Datenpipeline erfasst, beobachtet und gemeldet werden

Fehlertoleranz und automatische Wiederherstellung

Bewältigen vorübergehender Fehler und Wiederherstellen nach Auftreten der beim Pipelinebetrieb am häufigsten auftretenden Fehlerzustände mit schneller, skalierbarer und automatischer Wiederherstellung, die Folgendes umfasst:

- Fehlertolerante Mechanismen zur konsistenten Wiederherstellung des Datenzustands
- Möglichkeiten der automatischen Fortschrittsverfolgung beginnend bei der Quelle mit Checkpoint-Prüfung
- Möglichkeit zur automatischen Wiederherstellung des Zustands der Datenpipeline

Beobachtbarkeit von Datenpipelines

Überwachen Sie den Gesamtzustand der Datenpipeline mithilfe eines Dashboards mit einem Datenflussdiagramm und verfolgen Sie die End-to-End-Integrität der Pipeline in den Bereichen Leistung, Qualität und Latenz. Die Beobachtung der Datenpipeline umfasst:

- Ein hochwertiges und wirklichkeitsgetreues Herkunftsdiagramm, das den Datenfluss zwecks Folgenabschätzung transparent macht
- Differenzierte Protokollierung von Leistung und Status der Datenpipeline auf Zeilenebene
- Ständige Überwachung der Datenpipeline-Jobs zur Gewährleistung eines kontinuierlichen Betriebs

Batch- und Stream-Datenverarbeitung

Ermöglichen Sie Data Engineers eine Feinabstimmung der Datenlatenz mit Kostenkontrolle auch dann, wenn sie sich nicht mit komplexer Streaming-Verarbeitung auskennen, und ohne Implementieren einer Wiederherstellungslogik.

- Ausführen von Datenpipeline-Workloads auf automatisch bereitgestellten elastischen Rechenclustern auf Apache Spark™-Basis für Skalierung und Leistung
- Nutzen von Clustern zur Leistungsoptimierung, die Jobs parallelisieren und die Datenbewegung minimieren

Automatische Bereitstellung und Betrieb

Sorgen Sie für eine zuverlässige und vorhersehbare Bereitstellung von Daten für Analysen und maschinelles Lernen. Hierzu ermöglichen Sie Bereitstellung und Rollbacks von Datenpipelines auf einfache und automatische Weise, um Ausfallzeiten zu minimieren. Dies bietet folgende Vorteile:

- Vollständige, parametrisierte und automatisierte Implementierung für die fortlaufende Bereitstellung von Daten
- End-to-End-Orchestrierung, -Prüfung und -Überwachung der Datenpipeline-Implementierung bei allen wichtigen Cloud-Anbietern

Geplante Pipelines und Workflows

Einfache, klare und zuverlässige Orchestrierung von Datenverarbeitungsaufgaben für Daten- und Machine-Learning-Pipelines mit der Möglichkeit, mehrere nicht interaktive Tasks als Directed-Acyclic-Graph (DAG) auf einem Databricks-Compute-Cluster auszuführen.

- Einfaches Orchestrieren von Tasks in einem DAG unter Verwendung der Databricks-UI und -API
- Erstellen und Verwalten mehrerer Tasks in Jobs über UI oder API sowie Funktionen wie E-Mail-Warnungen für die Überwachung
- Cloud-übergreifendes Orchestrieren beliebiger Aufgaben, die über eine API außerhalb von Databricks verfügen

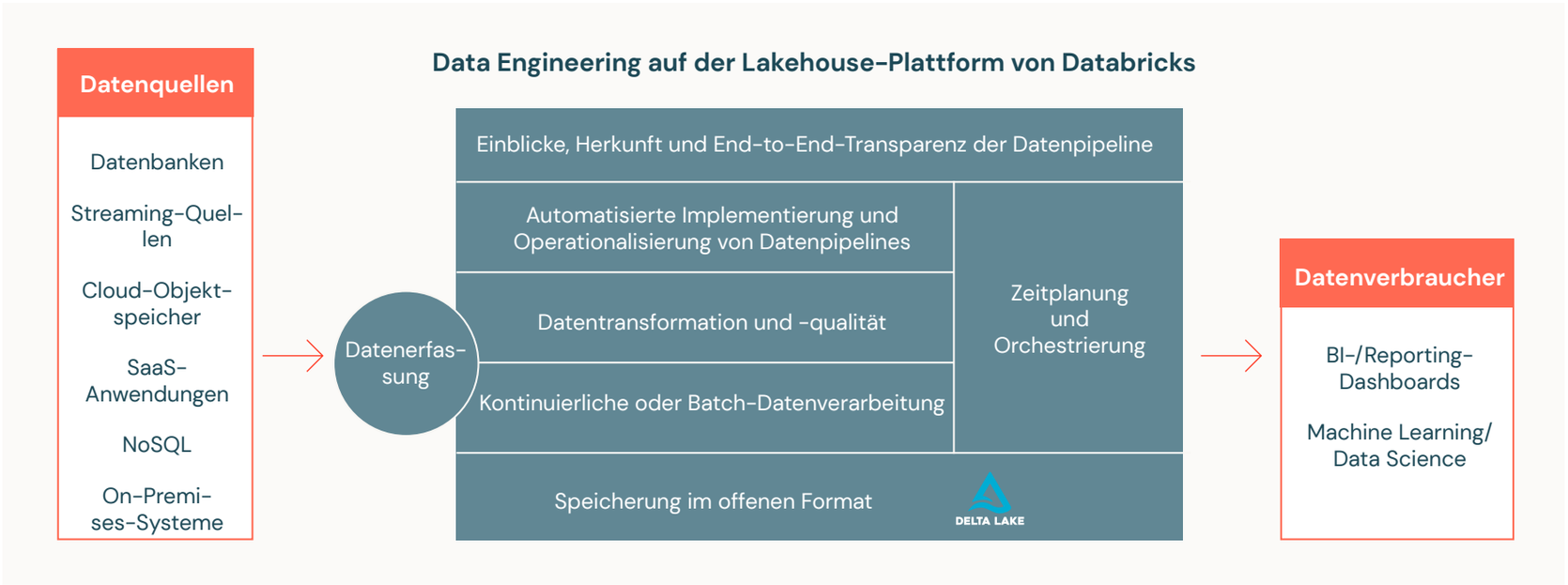


Abbildung 2
Data Engineering auf der Databricks-Referenzarchitektur

Fazit

Immer mehr Unternehmen arbeiten intensiv daran, datengesteuert zu werden. Dabei ist Data Engineering ein zentraler Erfolgsfaktor. Data Engineers, die zuverlässige und vertrauenswürdige Daten bereitstellen wollen, sollten keine Zeit für die manuelle Entwicklung und Pflege eines End-to-End-ETL-Lebenszyklus aufwenden müssen. Data-Engineering-Teams benötigen eine effiziente und skalierbare Methode, um die ETL-Entwicklung zu vereinfachen, die Belastbarkeit von Daten zu verbessern und ihren Betrieb zu verwalten.

Wie erwähnt, sorgen die acht wichtigsten Funktionen dafür, dass alle Datenabhängigkeiten automatisiert und gepflegt, integrierte Mechanismen zur Qualitätskontrolle mit Überwachung genutzt und dank automatischer Wiederherstellung umfassende Erkenntnisse aus dem Pipeline-Betrieb gewonnen werden, und vereinfachen auf diese Weise den ETL-Lebenszyklus. Data-Engineering-Teams können sich jetzt auf die einfache und schnelle Erstellung zuverlässiger und produktionsfertiger End-to-End-Datenpipelines konzentrieren. Dabei setzen sie ausschließlich SQL oder Python für Batch und Streaming ein und liefern hochwertige Daten für Analysen, Data Science oder maschinelles Lernen.

Anwendungsfälle

Im nächsten Kapitel werden Best Practices für End-to-End-Anwendungsfälle im Data Engineering beschrieben. Dabei stützen wir uns auf Beispiele aus der Praxis. Wir betrachten den gesamten Ablauf von der Datenerfassung und -verarbeitung bis hin zu Analysen und Machine Learning und zeigen anhand dessen, wie Sie aus Rohdaten verwertbare Informationen gewinnen. Außerdem statten wir Sie mit Datasets und Codebeispielen aus, damit Sie alle Aspekte des Datenlebenszyklus auf der Databricks Lakehouse-Plattform gleich in der Praxis erkunden können.

databricks

KAPITEL

02

Praktische Anwendungsfälle auf der Databricks Lakehouse-Plattform

Echtzeitanalysen am Point of Sale mit dem Data Lakehouse

Aufbau eines Cybersicherheits-Lakehouse für CrowdStrike Falcon-Events

Erschließen des Potenzials von Gesundheitsdaten mit einem modernen Data Lakehouse

Aktualität und Zuverlässigkeit beim Übermitteln von aufsichtsrechtlichen Meldungen

Umfassende Lösungen zur Geldwäschebekämpfung mit der Databricks Lakehouse-Plattform

Aufbau eines KI-Modells zur Echtzeiterkennung von toxischem Verhalten in Spielen

Fördern des Wandels bei Northwestern Mutual (Insights Platform)
durch den Umstieg auf eine skalierbare und offene Lakehouse-Architektur

So erstellte das Databricks-Datenteam ein Lakehouse über drei Clouds und mehr als 50 Regionen

KAPITEL 2.1

Echtzeitanalysen am Point of Sale mit dem Data Lakehouse

von **BRYAN SMITH** und **ROB SAKER**

9. September 2021

Störungen in der Lieferkette – verursacht durch reduziertes Produktangebot und verringerte Lagerkapazitäten – sowie die sich rasant verändernden Erwartungen der Verbraucher im Hinblick auf nahtlose Omnichannel-Erlebnisse zwingen Händler dazu, sich Gedanken über die Nutzung von Daten zur Verwaltung ihrer Abläufe zu machen. Bereits vor der Pandemie gaben 71 % der Einzelhandelsunternehmen an, dass das größte Hindernis bei der Umsetzung ihrer Omnichannel-Ziele der fehlende Echtzeiteinblick in den eigenen Warenbestand sei. Insofern hat die Pandemie die Nachfrage nach integrierten Online- und In-Store-Erlebnissen nur verstärkt und den Druck auf die Händler, die Produktverfügbarkeit korrekt anzugeben und Änderungen bei Bestellungen ohne Verzögerung einzupflegen, weiter erhöht. Ein besserer Zugang zu Echtzeitinformationen ist der Schlüssel zur Erfüllung der Verbraucherwünsche in der neuen Normalität.

In diesem Beitrag befassen wir uns mit dem Bedarf an Echtzeitdaten im Handel sowie mit der Frage, wie man die Herausforderungen des Echtzeit-Streamings von Point-of-Sale-Daten in großem Maßstab mit einem Data Lakehouse bewältigen kann.

Das Point-of-Sale-System

Das Point-of-Sale-System (oder PoS-System) ist seit Langem das Herzstück der Ladeninfrastruktur. Es dient der Aufzeichnung des Austauschs von Waren und Dienstleistungen zwischen Händler und Kunde. Zur Unterstützung dieses Austauschs erfasst das PoS-System normalerweise die Produktbestände und ermöglicht so eine Auffüllung, sobald die Stückzahlen unter ein kritisches Niveau fallen. Die Bedeutung des PoS-Systems für die Abläufe in den Ladengeschäften kann gar nicht hoch genug eingeschätzt werden, und als Aufzeichnungssystem für verkaufs- und bestandsbezogene Vorgänge ist der Zugriff auf seine Daten für Business Analysts von größtem Interesse.

Früher machte es die begrenzte Konnektivität zwischen den einzelnen Ladengeschäften und der Unternehmenszentrale notwendig, dass sich das gesamte PoS-System (und nicht nur die zugehörigen Terminalschnittstellen) physisch im Laden befanden. Nach Geschäftsschluss konnten diese Systeme dann eine Datenverbindung herstellen, um zusammengefasste Daten an ein Data Warehouse zu übermitteln, die nach der Konsolidierung einen tagesaktuellen Überblick über die Leistung der jeweiligen Filiale boten, jedoch bis zum Beginn der nächsten Datenübertragung am nächsten Abend veraltet waren.

Abbildung 1
Bestandsverfügbarkeit bei herkömmlichen, Batch-orientierten ETL-Mustern

Dank der mit moderner Konnektivität einhergehenden Verbesserungen können immer mehr Händler auf ein zentralisiertes, Cloud-basiertes PoS-System umsteigen, und viele andere entwickeln währenddessen Quasi-Echtzeit-Integrationen zwischen den Systemen in den Läden und dem Back-Office des Unternehmens. Die Verfügbarkeit von Informationen in Quasi-Echtzeit hat zur Folge, dass Händler ihre Schätzungen zur Verfügbarkeit von Artikeln fortlaufend aktualisieren können. Das Unternehmen steuert die Vorgänge nicht mehr wie zuvor basierend auf der Kenntnis der Bestandslage vom Vortag, sondern kann Maßnahmen auf Grundlage seines Wissens über den Bestand in diesem Moment ergreifen.

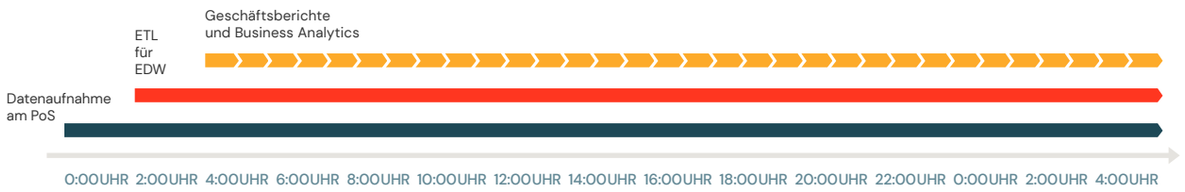


Abbildung 2
Inventarverfügbarkeit bei Streaming-ETL-Mustern

Erkenntnisse in Quasi-Echtzeit

So wirkungsvoll Erkenntnisse in Quasi-Echtzeit auch sein mögen: Der Übergang von der nächtlichen Verarbeitung zum kontinuierlichen Informationsfluss bringt besondere Herausforderungen mit sich – und dies nicht nur für den Data Engineer, der einen ganz neuartigen Workflow für die Datenverarbeitung entwickeln muss, sondern auch für den Informationsverbraucher. In diesem Beitragberichten wir von Erfahrungen einiger Kunden, die sich vor Kurzem auf diese Reise begeben haben, und untersuchen, wie wichtige Muster und Funktionen, die über das Lakehouse-Muster implementiert wurden, zum Erfolg geführt haben.

LEKTION 1

Den Rahmen ausloten

PoS-Systeme beschränken sich oft nicht nur einfach auf die Umsatz- und Bestandsverwaltung, sondern bieten eine breite Palette von Funktionen, z. B. Zahlungsabwicklung, Verwaltung von Salden, Rechnungsstellung und Auftragserteilung, Verwaltung von Treueprogrammen, Mitarbeiterplanung, Zeiterfassung und sogar Gehaltsabrechnung. So werden solche Systeme zum echten Universalwerkzeug im Ladenbetrieb.

Aus diesem Grund jedoch verteilen sich die im PoS-System gespeicherten Daten in der Regel über eine große und komplexe Datenbankstruktur. Wenn man Glück hat, stellt die PoS-Lösung eine Datenzugriffsschicht zur Verfügung, die diese Daten über einfacher zu interpretierende Strukturen zugänglich macht. Andernfalls muss der Data Engineer die mitunter unübersichtlichen Tabellen aufräumen, um festzustellen, welche Daten wertvoll sind und welche nicht.

Unabhängig davon, wie die Daten zugänglich gemacht werden, gilt die klassische Faustregel: Benennen Sie einen überzeugenden geschäftlichen Grund, um Ihre Lösung zu rechtfertigen, und begrenzen Sie auf dessen Grundlage den Umfang der anfänglich genutzten Datenbestände. Eine solche Rechtfertigung stammt häufig von einem einflussreichen Sponsor im Unternehmen, der mit der Bewältigung einer bestimmten geschäftlichen Aufgabe betraut ist und die Verfügbarkeit aktueller Informationen als entscheidend für seinen Erfolg ansieht.

Zur Veranschaulichung wollen wir eine wesentliche Herausforderung für viele Handelsunternehmen genauer unter die Lupe nehmen: die Implementierung von Omnichannel-Lösungen. Solche Lösungen, mit denen BOPIS-Transaktionen (Buy Online, Pickup In-Store) und filialübergreifende Transaktionen ermöglicht werden, sind auf hinreichend genaue Informationen über den Warenbestand im Ladengeschäft angewiesen. Müssten wir unseren anfänglichen Umfang auf diesen einen Bedarf beschränken, dann würde dies die Informationsanforderungen unseres Überwachungs- und Analysesystems drastisch verringern. Sobald eine echtzeitfähige Warenbestandslösung verfügbar ist und der Mehrwert vom Management anerkannt wird, können wir unseren Spielraum erweitern, um zusätzliche Anforderungen wie etwa die Beobachtung von Werbeaktionen und die Betrugserkennung einzubeziehen. So lässt sich das Ausmaß der Nutzung von Informationsressourcen mit jeder Iteration erweitern.

LEKTION 2

Ausrichten der Übertragung an Datenerzeugungsmustern und zeitlicher Sensibilität

Verschiedene Prozesse erzeugen unterschiedliche Daten im PoS. Umsatztransaktionen hinterlassen im Zweifelsfall eine Spur neuer Datensätze, die an die entsprechenden Tabellen angehängt werden. Retouren können mehrere Pfade durchlaufen, die Korrekturen in den Daten zur Umsatzentwicklung, das Einfügen neuer (also stornierter) Umsatzdatensätze und/oder das Ergänzen von Informationen in retourenspezifischen Strukturen nach sich ziehen. Die Dokumentation des Anbieters, undokumentiertes Wissen und womöglich sogar unabhängige Nachforschungen können erforderlich sein, um herauszufinden, wie und wo genau ereignisspezifische Informationen im PoS landen.

Das Verstehen solcher Muster kann hilfreich sein, um eine Datenübertragungsstrategie für bestimmte Informationstypen zu entwickeln. Für fortlaufendes Streaming etwa eignen sich hochfrequente und hochgradig differenzierte Muster mit Einfügecharakter möglicherweise ideal. Weniger häufig auftretende Ereignisse größeren Ausmaßes eignen sich wahrscheinlich am besten für eine Batch-gesteuerte Massendatenübertragung. Wenn aber diese beiden Arten der Datenübertragung zwei Enden eines Spektrums darstellen, dann liegen die meisten Ereignisse, die am PoS erfasst werden, wahrscheinlich irgendwo dazwischen.

Das Schöne am Data-Lakehouse-Ansatz für die Datenarchitektur ist, dass mehrere Datenübertragungsmodi parallel implementiert werden können. So kann für Daten, die sich ihrem Wesen nach für eine kontinuierliche Übertragung eignen, das Streaming verwendet werden. Dagegen sind Batch-Prozesse im Zweifelsfall die bessere Wahl für Daten, die sich am Prinzip der Massenübertragung orientieren. Und bei Daten, die zwischen diesen beiden Polen liegen, können Sie den Schwerpunkt auf die Frage legen, wie aktuell die Daten für die Entscheidungsfindung sein müssen, und sich von diesem Aspekt leiten lassen. Alle diese Modi können mit einem stimmigen Ansatz für die ETL-Implementierung angegangen werden. Genau dieses Problem hat viele frühere, häufig als Lambda-Architekturen bezeichnete Implementierungen verhindert.

LEKTION 3

Daten etappenweise erfassen

Die Daten werden von den PoS-Systemen in den Ladengeschäften mit unterschiedlicher Häufigkeit, in unterschiedlichen Formaten und mit unterschiedlichen Erwartungen an die rechtzeitige Verfügbarkeit übermittelt. Durch die Nutzung des bei Lakehouses beliebten Bronze/Silver/Gold-Entwurfsmusters können Sie die anfängliche Bereinigung, Umformatierung und Persistenz der Daten von den komplexeren Transformationen trennen, die für bestimmte geschäftsspezifische Arbeitsergebnisse erforderlich sind.

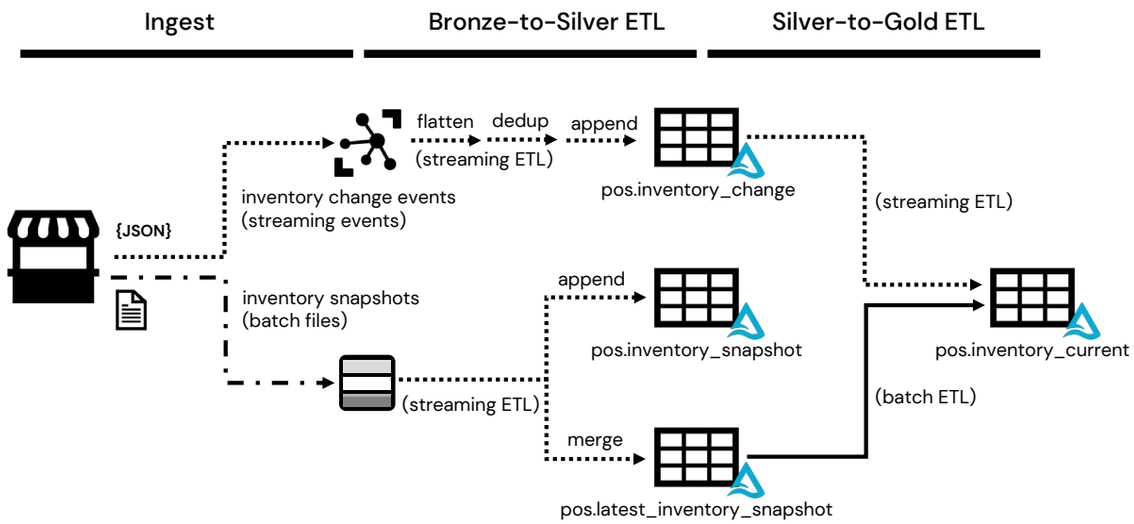


Abbildung 3
Data-Lakehouse-Architektur zur Berechnung des aktuellen Warenbestands unter Verwendung des Bronze/Silver/Gold-Datenpersistenzmusters

LEKTION 4

Erwartungen verwalten

Die Umstellung auf Echtzeitanalysen erfordert auch Veränderungen im Unternehmen. Gartner beschreibt dies durch das Streaming Analytics Maturity-Modell, innerhalb dessen die Analyse von Streaming-Daten in das Tagesgeschäft integriert wird. Dies geschieht aber nicht über Nacht.

Vielmehr brauchen Data Engineers Zeit, um die Herausforderungen zu erkennen, die mit dem Daten-Streaming von physischen Filialen zu einem zentralisierten, Cloud-basierten Backoffice verbunden sind. Dank besserer Konnektivität und höherer Systemzuverlässigkeit, verbunden mit immer robusteren ETL-Workflows, gelangen Daten schneller und mit höherer Zuverlässigkeit und Konsistenz zum Zielort. Dies erfordert häufig den Ausbau von Partnerschaften mit System Engineers und Anwendungsentwicklern, um die Integration in einem Maß zu unterstützen, wie es in den Tagen der reinen Batch-ETL-Workflows nicht die Regel war.

Business Analysts müssen sich mit dem Rauschen vertraut machen, das für fortlaufend aktualisierte Daten so typisch ist. Sie müssen das Diagnostizieren und Validieren von Datensätzen neu erlernen, weil beispielsweise zwei identische Anfragen – ausgeführt in einem Abstand von nur wenigen Sekunden – zu geringfügig unterschiedlichen Ergebnissen führen können. Sie müssen ihre Wahrnehmung von Problemen in den Daten schulen, die bei der Darstellung tagesaktueller Aggregationen oft verborgen bleiben. All dies erfordert Korrekturen sowohl bei der Analyse von in den Ergebnissen erkannten Signalen als auch bei der Reaktion darauf.

Diese Vorgänge finden bereits in den frühen Phasen des Reifeprozesses statt. Später dann kann die Fähigkeit des Unternehmens, aussagekräftige Signale im Datenstrom zu erkennen, die Möglichkeiten der automatisierten Erkennung und Reaktion verbessern. Hier erschließt sich der maximale Mehrwert in den Datenströmen. Doch bevor ein Unternehmen seine Tätigkeiten diesen Technologien anvertraut, müssen Überwachung und Governance implementiert und ihre Tauglichkeit festgestellt werden.

Implementieren des PoS-Streamings

Um zu veranschaulichen, wie die Lakehouse-Architektur auf PoS-Daten angewendet werden kann, haben wir einen Demo-Workflow entwickelt, um einen Warenbestand in Quasi-Echtzeit zu berechnen. Wir stellen uns dabei vor, dass zwei getrennte PoS-Systeme bestandsrelevante Informationen über Verkäufe, Wiederauffüllungen und Schwunddaten zusammen mit BOPIS-Transaktionen (die vom einem System eingeleitet und vom anderen abgewickelt werden) als Teil eines Streaming-Feeds zur Warenbestandsänderung übertragen. Regelmäßige, als Momentaufnahmen ausgeführte Zählungen der ab Lager verfügbaren Produkteinheiten werden vom PoS-System erfasst und en gros übermittelt. Diese Daten werden für einen Zeitraum von einem Monat simuliert und mit zehnfacher Geschwindigkeit wiedergegeben, um einen besseren Einblick in Änderungen des Warenbestands zu erhalten.

Die ETL-Prozesse (entsprechend Darstellung in Abbildung 3) stellen eine Mischung aus Streaming- und Batch-Techniken dar. Ein zweistufiger Ansatz mit minimal transformierten Daten, die in Deltatabellen erfasst werden, die unsere Silver-Ebene darstellen, trennt unseren zuerst beschriebenen, eher technisch ausgerichteten ETL-Ansatz von dem Ansatz, der für Berechnungen des aktuellen Bestands zuständig ist und eher geschäftlichen Charakter hat. Die zweite Phase wurde unter Verwendung traditioneller strukturierter Streaming-Funktionen implementiert. Auf diesen Aspekt werden wir ggf. im Zusammenhang mit den Delta Live-Tabellen später noch einmal zurückkommen, wenn die allgemeine Verfügbarkeit dieser neuen Funktionen anstehen.

Die Demo nutzt Azure IoT Hub-Instanzen sowie Azure Storage für die Datenerfassung, würde aber in ähnlicher Weise auch in den AWS- und GCP-Clouds funktionieren, wenn die jeweiligen Technologien entsprechend ersetzt würden.

Experimentieren Sie mit den folgenden kostenlosen Databricks-Notebooks

- **PoS 01: Einrichtung der Umgebung**
- **PoS 02: Datengenerierung**
- **PoS 03: ETL der Datenaufnahme**
- **PoS 04: Aktueller Bestand**

KAPITEL 2.2

Aufbau eines Cybersicherheits-Lakehouse für CrowdStrike Falcon-Events

von **AEMRO AMARE , ARUN PAMULAPATI, YONG SHENG HUANG** und **JASON POHL**

20. Mai 2021

Endpunktdaten werden von Sicherheitsteams für die Bedrohungserkennung, das Threat Hunting, die Untersuchung sicherheitsrelevanter Vorfälle und die Erfüllung von Compliance-Anforderungen benötigt. Die Datenmengen können viele Terabyte pro Tag bzw. mehrere Petabyte pro Jahr betragen. Die meisten Unternehmen tun sich schwer mit der Erfassung, Speicherung und Analyse von Endpunkt-Protokolldateien, da die mit derart großen Datenmengen einhergehenden Kosten und die Komplexität sehr hoch sind. Aber das muss nicht so sein.

In dieser zweiteiligen Blogpostserie erfahren Sie, wie Sie mit Databricks Endpunktdaten auch in einer Größenordnung von Petabytes operationalisieren können, um Ihr Sicherheitsniveau mithilfe fortschrittlicher Analysen kostengünstig zu steigern. Teil 1 (dieser Post) behandelt die Architektur der Datensammlung und die Integration mit einem SIEM (Splunk). Am Ende des Posts werden Sie dank der bereitgestellten Notebooks in der Lage sein, die Daten zur Analyse zu verwenden. In Teil 2 werden konkrete Anwendungsfälle, die Erstellung von ML-Modellen und automatisierte Anreicherungen und Analysen behandelt. Am Ende von Teil 2 können Sie die Notebooks zur Erkennung und Untersuchung von Bedrohungen mithilfe von Endpunktdaten implementieren.

Wir werden Logdateien aus CrowdStrike Falcon als Beispiele verwenden. Für den Zugriff auf diese Logdateien können Sie den Falcon Data Replicator (FDR) verwenden, um Ereignisrohdaten von der CrowdStrike-Plattform in einen Cloud-Speicher wie Amazon S3 zu übertragen. Diese Daten können mit der Lakehouse-Plattform von Databricks zusammen mit den weiteren Sicherheitstelemetriedaten erfasst, transformiert, analysiert und gespeichert werden. Kunden können CrowdStrike Falcon-Daten einlesen, eine Echtzeiterkennung auf Python-Basis anwenden, Verlaufsdaten mit Databricks SQL durchsuchen und Abfragen von SIEM-Tools wie Splunk mit dem Databricks Add-On für Splunk durchführen.

Probleme bei der Operationalisierung von CrowdStrike-Daten

Zwar stellen die CrowdStrike Falcon-Daten umfassende Details zur Ereignisprotokollierung zur Verfügung, doch ist es eine gewaltige Aufgabe, riesige Mengen komplexer Cybersicherheitsdaten in Quasi-Echtzeit und auf kosteneffiziente Weise zu erfassen, zu verarbeiten und zu operationalisieren. Dies sind nur einige der bekannten Herausforderungen:

- **Datenerfassung im großen Maßstab in Echtzeit:** Es ist schwierig, den Überblick über die verarbeiteten und unverarbeiteten Rohdatendateien zu behalten, die von FDR in Quasi-Echtzeit in den Cloud-Speicher geschrieben werden.
- **Komplexe Transformationen:** Das Datenformat ist halbstrukturiert. Jede Zeile einer Logdatei enthält Hunderte geradezu kriminell unterschiedlicher Arten von Nutzdaten. Zudem kann sich die Struktur der Ereignisdaten im Laufe der Zeit ändern.
- **Data Governance:** Solche Daten können sensibel sein, und der Zugriff darauf muss sich auf diejenigen Benutzer beschränken, die sie tatsächlich brauchen.

- **Vereinfachte End-to-End-Sicherheitsanalysen:** Für Data Engineering, ML und die Analyse dieser schnelllebigen und umfangreichen Datensätze werden skalierbare Tools benötigt.
- **Zusammenarbeit:** Durch eine effektive Zusammenarbeit kann das Fachwissen von Data Engineers, Cybersicherheitsanalysten und ML Engineers sinnvoll eingesetzt werden. Daher verbessert das Vorhandensein einer kollaborativen Plattform die Workload-Effizienz bei Analysen und Abhilfen im Cybersicherheitsbereich.

Infolgedessen befinden sich die Security Engineers in den Unternehmen in einer schwierigen Lage, denn Sie müssen die Kosten ebenso im Griff haben wie die betriebliche Effizienz. Dabei müssen sie entweder akzeptieren, an sehr teure proprietäre Systeme gebunden zu sein, oder einen enormen Aufwand betreiben, um eigene Sicherheitstools für Endgeräte zu entwickeln, die zudem ausreichende Skalierbarkeit und Leistung bieten.

Cybersicherheits-Lakehouse von Databricks

Databricks bietet Sicherheitsteams und Data Scientists neue Möglichkeiten, ihre Aufgaben effizient und effektiv zu erfüllen, und zudem eine Reihe von Tools, um die wachsenden, mit Big Data und immer ausgefeilteren Bedrohungen einhergehenden Herausforderungen zu bewältigen.

Lakehouse ist eine offene Architektur, die die besten Elemente von Data Lakes und Data Warehouses vereint. Ein Lakehouse vereinfacht den Aufbau einer Data-Engineering-Pipeline mit mehreren Hops, in deren Verlauf die Daten nach und nach an Struktur gewinnen. Der Vorteil einer Architektur mit mehreren Hops besteht darin, dass Data Engineers eine Pipeline entwickeln können, an deren Anfang die Rohdaten als „Single Source of Truth“ stehen, von denen alle nachfolgenden Abläufe abhängig sind. Die halbstrukturierten Rohdaten in CrowdStrike können jahrelang gespeichert werden, und nachgelagerte Transformationen und Aggregationen können nach Art eines End-to-End-Streamings durchgeführt werden, um die Daten zu verfeinern und kontextspezifische Strukturen einzuführen, die Erkennung und Analyse von Sicherheitsrisiken in verschiedenen Szenarien ermöglicht.

- **Datenerfassung:** Auto Loader (für AWS, Azure, GCP) hilft dabei, Daten unverzüglich zu lesen, sobald eine neue Datei von CrowdStrike FDR in den Rohdatenspeicher geschrieben wird. Das Programm nutzt Cloud-Benachrichtigungsservices, um neue Dateien schrittweise zu verarbeiten, sobald sie in der Cloud eintreffen. Außerdem konfiguriert Auto Loader den Benachrichtigungsdienst für neue Dateien automatisch, lauscht auf diesen und kann auf einen Durchsatz von mehreren Millionen Dateien pro Sekunde skalieren.
- **Einheitliche Stream- und Batch-Verarbeitung:** Delta Lake ist ein offener Ansatz, um Datenmanagement und Governance für Data Lakes zu implementieren und dabei die verteilte Rechenleistung von Apache Spark™ für große Datenmengen und Metadaten zu nutzen. Die Databricks Delta Engine ist eine hochoptimierte Engine, die in der Lage ist, Millionen von Datensätzen pro Sekunde zu verarbeiten.
- **Data Governance:** Mit Databricks Table Access Control (für AWS, Azure, GCP) können Administratoren verschiedene Ebenen für den Zugriff auf Deltatabellen basierend auf der geschäftlichen Funktion des betreffenden Benutzers definieren.

databricks

- **Tools für die Sicherheitsanalyse:** Databricks SQL hilft beim Erstellen eines interaktiven Dashboards mit automatischer Alarmierung bei Erkennung ungewöhnlicher Muster. Ebenso lässt es sich problemlos in gängige BI-Tools wie Tableau, Microsoft Power BI und Looker integrieren.
- **Zusammenarbeit an Databricks-Notebooks: Mit den** kollaborativen Notebooks von Databricks können Sicherheitsteams in Echtzeit zusammenarbeiten. Mehrere Benutzer können im selben Arbeitsbereich Abfragen in mehreren Sprachen ausführen, Visualisierungen gemeinsam nutzen und Kommentare abgeben, um Untersuchungen unterbrechungsfrei voranzutreiben.

Lakehouse-Architektur für CrowdStrike Falcon-Daten

Wir empfehlen die folgende Lakehouse-Architektur für Cybersicherheits-Workloads wie z. B. die CrowdStrike Falcon-Daten. Auto Loader und Delta Lake vereinfachen das Lesen von Rohdaten aus dem Cloud-Speicher und das Schreiben in eine Delta-Tabelle, machen es kostengünstig und minimieren den DevOps-Aufwand.

Bei dieser Architektur werden halbstrukturierte CrowdStrike-Daten in den Cloud-Speicher des Kunden in der Zielzone geladen. Dann nutzt Auto Loader Cloud-Benachrichtigungsservices, um die Verarbeitung und Erfassung neuer Dateien in die Bronze-Tabellen des Kunden automatisch auszulösen. Diese Tabellen fungieren als einzige vertrauenswürdige Quelle für alle nachgelagerten Jobs. Auto Loader behält verarbeitete wie unverarbeitete Dateien anhand von Kontrollpunkten im Blick und vermeidet so eine Doppelverarbeitung.

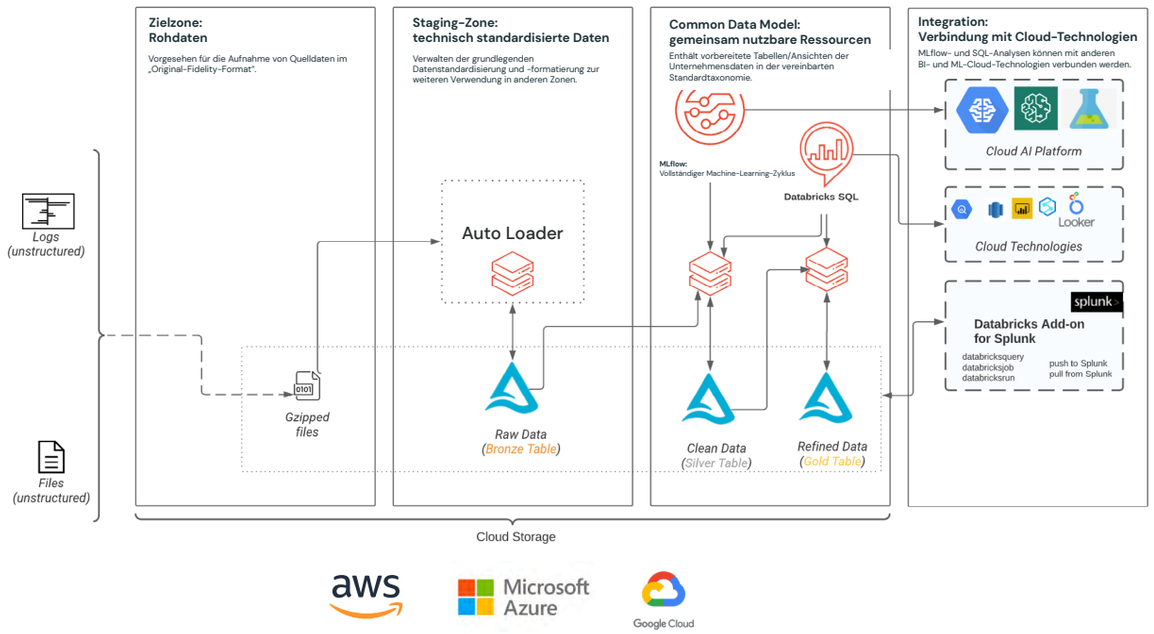


Abbildung 1
Lakehouse-Architektur für CrowdStrike Falcon-Daten

Beim Fortschreiten von der Bronze- zur Silver-Stufe wird ein Schema hinzugefügt, um die Daten zu strukturieren. Da wir aus einer einzigen vertrauenswürdigen Quelle lesen, können wir die vielen verschiedenen Ereignistypen verarbeiten und das passende Schema erzwingen, während sie in die jeweiligen Tabellen geschrieben werden. Die Möglichkeit, Schemata auf der Silver-Ebene durchzusetzen, vermittelt eine solide Grundlage für den Aufbau von ML- und Analyse-Workloads.

Die Gold-Phase, in der die Daten für schnellere Abfragen und höhere Leistung in Dashboards und BI-Tools aggregiert werden, ist optional und hängt vom jeweiligen Anwendungsfall und Datenvolumen ab. Werden unerwartete Tendenzen erkannt, dann können zuvor konfigurierte Warnungen ausgelöst werden.

Eine weitere optionale Funktion ist das Databricks-Add-On für Splunk. Hiermit können Sicherheitsteams die Vorteile des kosteneffizienten Databricks-Modells und der KI nutzen, ohne auf die Annehmlichkeiten von Splunk verzichten zu müssen. Kunden können mit dem Add-On aus einem Splunk-Dashboard oder einer Suchleiste heraus Ad-hoc-Abfragen an Databricks stellen. Außerdem können Benutzer Notebooks oder Jobs in Databricks über ein Splunk-Dashboard oder ausgelöst durch eine Splunk-Suche starten. Die Databricks-Integration ist bidirektional und ermöglicht es Kunden, verrauschte Daten zusammenzufassen oder Erkennungen in Databricks durchzuführen, die in Splunk Enterprise Security angezeigt werden. Kunden können sogar Splunk-Suchanfragen aus einem Databricks-Notebook heraus ausführen, um das Duplizieren von Daten zu vermeiden.

Die Integration von Splunk und Databricks ermöglicht es Kunden, Kosten zu senken, die zu analysierenden Datenquellen zu erweitern und die Ergebnisse einer robusteren Analyse-Engine bereitzustellen, ohne dass hierfür ein Wechsel der von den Mitarbeitern täglich verwendeten Tools erforderlich wäre.

Code-Analyse

Da Auto Loader den komplexesten Teil der dateibasierten Datenerfassung abstrahiert, kann eine Überführung der Rohdaten in die Bronze-Phase mit wenigen Codezeilen erstellt werden. Nachfolgend sehen Sie ein Codebeispiel für eine Delta-Erfassungspipeline in Scala. CrowdStrike Falcon-Ereignisdatensätze haben einen gemeinsamen Feldnamen: „event_simpleName“.

```
val crowdstrikeStream = spark.readStream
  .format(“cloudFiles”)
  .option(“cloudFiles.format”, “text”)    // Textdatei erfordert kein Schema
  .option(“cloudFiles.region”, “us-west-2”)
  .option(“cloudFiles.useNotifications”, “true”)
  .load(rawDataSource)
  .withColumn(“load_timestamp”, current_timestamp())
  .withColumn(“load_date”, to_date($”load_timestamp”))
  .withColumn(“eventType”, from_json($”value”, “struct”, Map.empty[String, String]))
.selectExpr(“eventType.event_simpleName”,”load_date”,”load_timestamp”, “value” )
  .writeStream
  .format(“delta”)
  .option(“checkpointLocation”, checkPointLocation)
  .table(“demo_bronze.crowdstrike”)
```

Bei der Überführung der Rohdaten auf die Bronze-Ebene wird nur der Ereignisname aus den Rohdaten extrahiert. Durch Hinzufügen eines Ladezeitstempels und der Datumsspalten speichern die Benutzer die Rohdaten in der Bronze-Tabelle. Die Bronze-Tabelle ist nach Ereignisname und Ladedatum unterteilt. So werden nachgelagerte Jobs für die Überführung auf die Silver-Ebene insbesondere dann leistungsfähiger, wenn nur eine begrenzte Anzahl von Ereignisdatumsbereichen relevant ist. Als Nächstes liest ein Bronze-

Silver-Streaming-Job Ereignisse aus einer Bronze-Tabelle, implementiert ein Schema und schreibt auf der Grundlage des Ereignisnamens in Hunderte von Ereignistabellen. Nachfolgend sehen Sie ein Codebeispiel in Scala:

```
spark
  .readStream
  .option("ignoreChanges", "true")
  .option("maxBytesPerTrigger", "2g")
  .option("maxFilesPerTrigger", "64")
  .format("delta")
  .load(bronzeTableLocation)
  .filter($"event_simpleName" === "event_name")
  .withColumn("event", from_json($"value", schema_of_json(sampleJson)) )
  .select($"event.*", $"load_timestamp", $"load_date")
  .withColumn("silver_timestamp", current_timestamp())
  .writeStream
  .format("delta")
  .outputMode("append")
  .option("mergeSchema", "true")
  .option("checkpointLocation", checkPoint)
  .option("path", tableLocation)
  .start()
```

Jedes Ereignisschema kann in einer Schemaregistrierung oder einer Deltatabelle gespeichert werden, falls ein Schema von mehreren datengesteuerten Services gemeinsam genutzt werden soll. Beachten Sie, dass der obige Code einen exemplarischen JSON-String verwendet, der aus der Bronze-Tabelle ausgelesen wird, und dass das Schema aus dem JSON-Code mit schema_of_json() abgeleitet wird. Später wird der JSON-String mit from_json() in eine Struct umgewandelt. Anschließend wird die Struct geglättet, was das Hinzufügen einer Zeitstempelspalte erforderlich macht. Diese Schritte liefern ein DataFrame mit allen erforderlichen Spalten, die an eine Ereignistabelle angehängt werden können. Schließlich schreiben wir diese strukturierten Daten im Append-Modus in eine Ereignistabelle.

Mit foreachBatch lassen sich Ereignisse alternativ mit nur einem Stream auf mehrere Tabellen verteilen. Hierzu wird eine Funktion definiert, die Microbatches verarbeitet. Mit foreachBatch() ist es möglich, vorhandene Batch-Datenquellen zum Filtern und Schreiben in mehrere Tabellen wiederzuverwenden. ForeachBatch() bietet jedoch nur eine at-least-once-Schreibgarantie. Daher ist eine manuelle Implementierung notwendig, um die exactly-once-Semantik zu erzwingen.

In diesem Stadium können die strukturierten Daten mit einer der Sprachen abgefragt werden, die in Databricks-Notebooks und -Jobs unterstützt werden: Python, R, Scala oder SQL. Die Daten der Silver-Ebene lassen sich bequem für ML- und Cyberangriffsanalysen verwenden.

Die nächste Streaming-Pipeline verliefe dann von Silver zu Gold. In diesem Stadium ist es möglich, Daten für Dashboards und Warnmeldungen zu aggregieren. Im zweiten Teil dieser Blogpostreihe werden wir Ihnen etwas ausführlicher zeigen, wie wir Dashboards mit Databricks SQL erstellen.

So geht es weiter

Besuchen Sie uns regelmäßig, um weitere Blogposts zu lesen, die durch den Einsatz von ML und Databricks SQL zusätzlichen Mehrwert für diesen Anwendungsfall generieren.

Sie können diese Notebooks in Ihrer eigenen Databricks-Implementierung verwenden. Jeder Notebook-Abschnitt ist mit Kommentaren versehen. Sie dürfen uns gerne eine E-Mail senden: cybersecurity@databricks.com. Wir freuen uns auf Ihre Fragen und Vorschläge, um die Verständlichkeit und den Einsatz dieses Notebooks noch einfacher zu gestalten.

Experimentieren Sie mit den folgenden kostenlosen Databricks-Notebooks.

KAPITEL 2.3

Erschließen des Potenzials von Gesundheitsdaten mit einem modernen Data Lakehouse

von MICHAEL ORTEGA , MICHAEL SANKY und AMIR KERMANY

19. JULI 2021

So lassen sich die Herausforderungen von Data Warehouses und Data Lakes im Gesundheitswesen und den Biowissenschaften überwinden

Ein einziger Patient produziert etwa 80 MB medizinische Daten pro Jahr. Schreibt man diesen Wert für Tausende von Patienten über ihr jeweils gesamtes Leben fort, dann kommt man auf Patientendaten in einer Größenordnung von mehreren Petabyte, die wertvolle Erkenntnisse enthalten. Das Erschließen dieser Informationen kann dazu beitragen, den Klinikbetrieb zu optimieren, die Arzneimittelforschung zu beschleunigen und die Patientengesundheit zu verbessern. Doch zunächst müssen die Daten für nachgelagerte Analysen und KI aufbereitet werden. Leider benötigen die meisten Unternehmen im Gesundheitswesen und in den Biowissenschaften bereits für das Erfassen, Bereinigen und Strukturieren ihrer Daten übermäßig viel Zeit.

Ein einziger Patient produziert mehr als 80 MB medizinische Daten pro Jahr

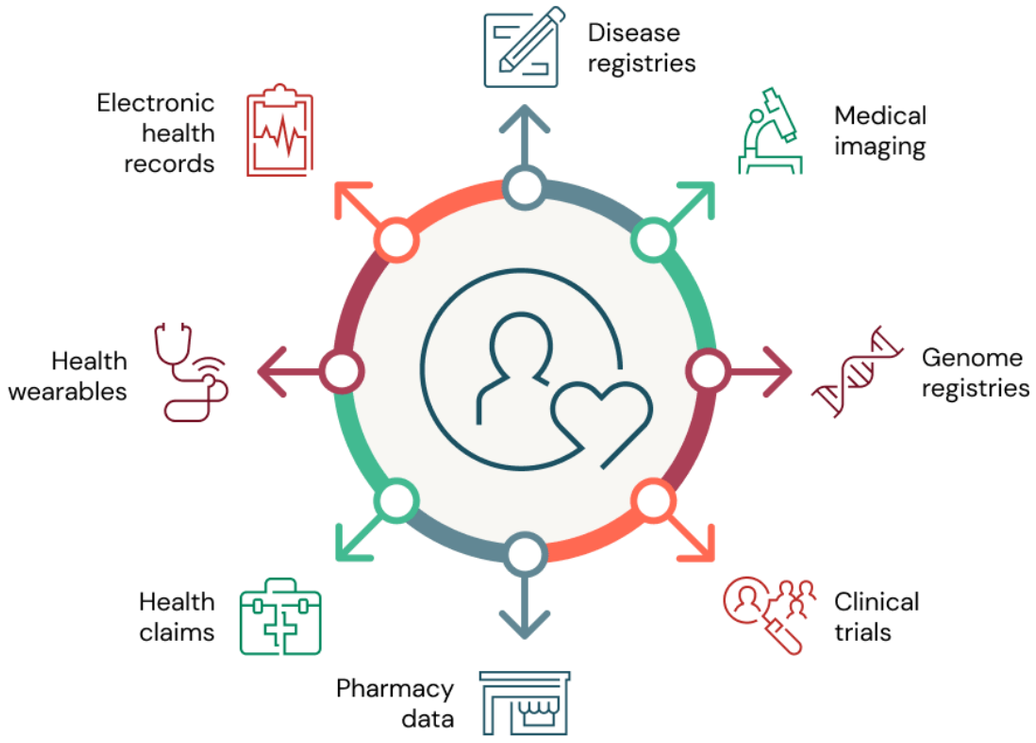


Abbildung 1
Gesundheitsdaten nehmen exponentiell zu: Bereits ein einziger Patient produziert über 80 MB Daten pro Jahr.

databricks

Herausforderungen für Datenanalysen im Gesundheitswesen und den Biowissenschaften

Es gibt eine ganze Reihe von Gründen dafür, warum Datenaufbereitung, Analysen und KI für Unternehmen im Gesundheitswesen eine Herausforderung darstellen. Viele davon stehen im Zusammenhang mit Investitionen in veraltete Datenarchitekturen, die auf Data Warehouses aufsetzen. Nachfolgend aufgeführt sind die vier häufigsten Herausforderungen, die in der Branche zu verzeichnen sind:

HERAUSFORDERUNG NR. 1: VOLUMINA

Skalieren für rapide zunehmende Gesundheitsdatenbestände

Die Gentechnik ist das wahrscheinlich beste Beispiel für das explosionsartige Wachstum der Datenvolumina im Gesundheitswesen. Die erstmalige Sequenzierung eines Genoms kostete über 1 Mrd. US-Dollar. Angesichts dieser enormen Kosten konzentrierten sich die ersten Bemühungen auf die Genomtypisierung, ein Verfahren zur Suche nach spezifischen Varianten in einem sehr kleinen Teil des Genoms eines Menschen – in der Regel etwa 0,1 %. (Übrigens widmet sich auch heute noch ein Großteil der Bemühungen dieser Aufgabe.) Daraus entwickelte sich die Whole-Exome-Sequenzierung, bei der die proteinkodierenden Teile des Genoms betrachtet werden, die immer noch weniger als 2 % des Gesamtgenoms ausmachen. Heute bieten Unternehmen dagegen WGS-Direkttests (Whole-Genome-Sequenzierung) für Verbraucher zu Kosten von unter 300 US-Dollar für 30 WGS-Prozesse an. Auf Bevölkerungsebene gibt die Biobank im Vereinigten Königreich in diesem Jahr mehr als 200.000 vollständige Genome für die Forschung frei. Aber es geht nicht nur um Genomik. Auch in Bereichen wie Bildgebung, Wearables mit Gesundheitsfunktionen und elektronischen Krankenakten ist ein enormes Wachstum zu verzeichnen.

Die Skalierung ist das A und O für Initiativen in Bereichen wie der Analyse der Bevölkerungsgesundheit und der Arzneimittelforschung. Leider sind viele Legacy-Architekturen On-Premises-Systeme und für Spitzenkapazitäten ausgelegt. Dieser Ansatz führt in Zeiten geringer Nutzung zu ungenutzter Rechenleistung (und damit am Ende zu Geldverschwendung) und lässt sich zudem nicht schnell skalieren, wenn Upgrades erforderlich sind.

HERAUSFORDERUNG NR. 2: VIELFALT

Analysieren verschiedenartiger Gesundheitsdaten

Unternehmen im Gesundheitswesen und in den Biowissenschaften haben es mit einer enormen Datenvielfalt zu tun, die jeweils eigene Nuancen aufweist. Es ist allgemein anerkannt, dass mehr als 80 % der medizinischen Daten unstrukturiert sind. Dennoch legen die meisten Unternehmen ihren Schwerpunkt immer noch auf Data Warehouses, obwohl diese eigentlich für strukturierte Daten und traditionelle SQL-basierte Analysen konzipiert sind. Unstrukturierte Daten umfassen beispielsweise Bilddaten, die für Diagnose und Messung von Krankheitsverläufen in Bereichen wie Onkologie, Immunologie und Neurologie (den am schnellsten wachsenden Kostenbereichen) von entscheidender Bedeutung sind, sowie fortlaufende Texte in ärztlichen Berichten, die für das Verständnis der vollständigen Krankheits- und Sozialgeschichte des Patienten unverzichtbar sind. Diese Datentypen zu ignorieren oder beiseitezulassen, kommt nicht infrage.

Erschwerend kommt hinzu, dass das Ökosystem im Gesundheitswesen immer stärker vernetzt ist und die Beteiligten sich mit neuartigen Datentypen auseinandersetzen müssen. So benötigen die Versicherungsunternehmen beispielsweise Schadensfalldaten, um Vereinbarungen zu Risikobeteiligungen zu verwalten und zu beurteilen, und Kostenträger brauchen klinische Daten, um etwa Maßnahmen vorab genehmigen und Qualitätsmaßnahmen fördern zu können. Oft fehlen in diesen Unternehmen Datenarchitekturen und Plattformen zur Unterstützung solcher neuer Datentypen.

Einige Unternehmen haben in Data Lakes investiert, um unstrukturierte Daten und erweiterte Analysen zu unterstützen, aber dies hat eine Reihe neuer Probleme aufgeworfen. In einer solchen Umgebung müssen Datenteams nun gleich zwei Systeme – nämlich Data Warehouses und Data Lakes – verwalten, in die von verschiedenen Tools Daten übertragen werden. Dies hat Probleme bei Datenqualität und Datenverwaltung zur Folge.

HERAUSFORDERUNG NR. 3: GESCHWINDIGKEIT

Erfassen von Daten-Streams für Echtzeiteinblicke in Patientendaten

In vielen Bereichen ist die Gesundheitsversorgung eine Frage von Leben und Tod. Der Zustand eines Patienten kann extrem dynamisch sein, weswegen eine Batch-Datenverarbeitung – selbst auf täglicher Basis – häufig nicht ausreicht. Der Zugang zu aktuellsten und sekundengenauen Informationen ist entscheidend für eine therapeutische Intervention. Um Leben zu retten, werden Streaming-Daten von Krankenhäusern und nationalen Gesundheitssystemen für alles Mögliche verwendet – von der Sepsisprognose bis zur Implementierung einer Echtzeitbedarfsprojektion für Intensivbetten.

Zudem ist die Datengeschwindigkeit ein wesentlicher Bestandteil der digitalen Revolution im Gesundheitswesen. Jeder Einzelne hat Zugang zu mehr Informationen als je zuvor und kann seine Versorgung in Echtzeit beeinflussen. Zum Beispiel können tragbare Geräte – wie etwa solche von Livongo zur fortlaufenden Messung des Blutzuckers – Daten in Echtzeit in mobile Anwendungen übertragen, die dann maßgeschneiderte Verhaltensempfehlungen geben.

Trotz einiger dieser ersten Erfolge sind die Datenarchitekturen in den meisten Unternehmen nicht für die für Streaming-Daten erforderliche Geschwindigkeit ausgelegt. Probleme bei der Zuverlässigkeit und der Integration von Echtzeit- und Verlaufsdaten weisen sich als Innovationsbremse.

HERAUSFORDERUNG NR. 4: AUTHENTIZITÄT

Vertrauen in Gesundheitsdaten und KI schaffen

Nicht zuletzt verlangen medizinische und gesetzliche Standards im Gesundheitswesen ein Höchstmaß an Datenkorrekheit. Unternehmen im Gesundheitswesen müssen im Bereich der öffentlichen Gesundheit hohe Compliance-Anforderungen erfüllen. Governance ist für die Demokratisierung von Daten in Unternehmen unverzichtbar.

Außerdem benötigen Unternehmen ein angemessenes Governance-Modell, wenn sie künstliche Intelligenz (KI) und maschinelles Lernen (ML) im klinischen Umfeld einsetzen wollen. Leider nutzen die meisten Unternehmen separate Plattformen für Data-Science-Workflows, die nicht mit ihrem Data Warehouse verbunden sind. Hierdurch entstehen erhebliche Herausforderungen, wenn es darum geht, bei KI-gestützten Anwendungen Vertrauen und Reproduzierbarkeit zu gewährleisten.

Erschließen von Gesundheitsdaten mit einem Lakehouse

Die Lakehouse-Architektur hilft Unternehmen im Gesundheitswesen und in den Biowissenschaften, diese Herausforderungen mithilfe einer modernen Datenarchitektur zu meistern, die Bezahlbarkeit, Skalierbarkeit und Flexibilität eines Cloud Data Lake mit der Leistung und Governance eines Data Warehouse kombiniert. Mit einem Lakehouse können Unternehmen Daten aller Art speichern und unterschiedlichste Analysen und ML in einer offenen Umgebung umsetzen.

Aufbauen eines Lakehouse für Gesundheitswesen und Biowissenschaften

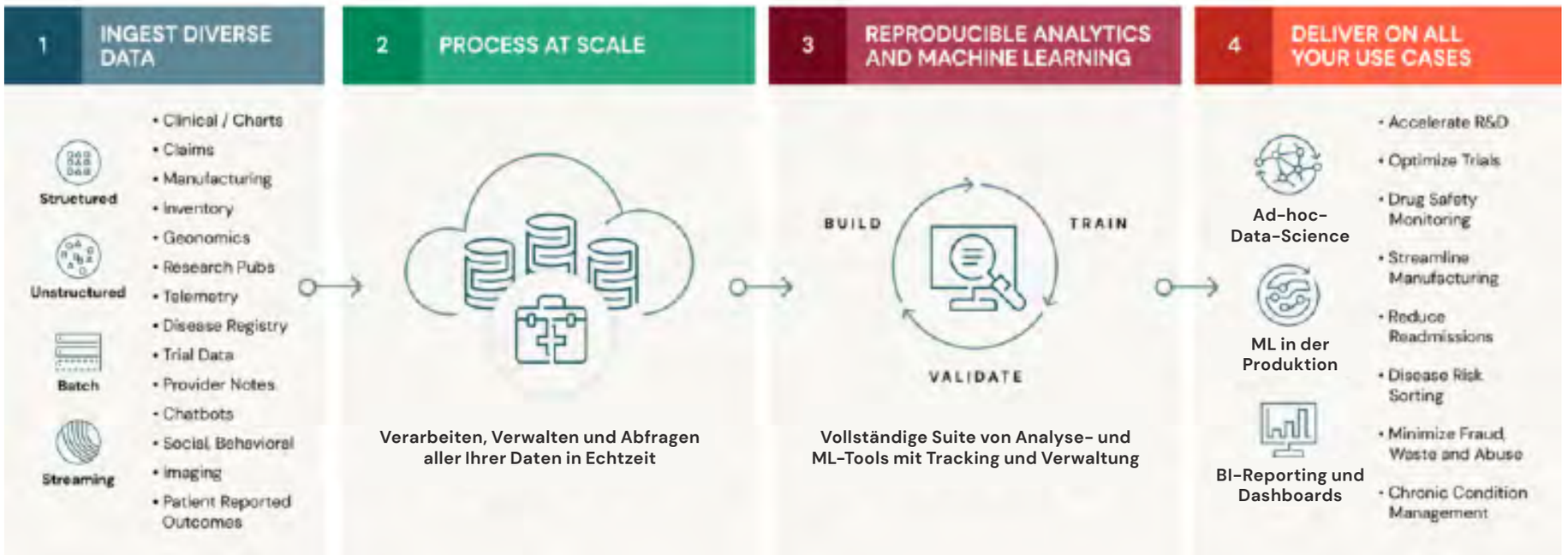


Abbildung 2
Implementieren aller denkbaren Anwendungsfälle für die Datenanalyse im Gesundheitswesen und den Biowissenschaften mit einer modernen Lakehouse-Architektur

Im Einzelnen bietet ein Lakehouse folgende Vorteile für Organisationen im Gesundheitswesen und den Biowissenschaften:

- **Organisieren aller Ihrer Gesundheitsdaten im großen Maßstab.** Das Herzstück der Lakehouse-Plattform von Databricks ist Delta Lake. Hierbei handelt es sich um eine Open-Source-Datenmanagementebene, die Zuverlässigkeit und Leistung für Ihren Data Lake bietet. Im Gegensatz zu einem herkömmlichen Data Warehouse unterstützt Delta Lake alle Arten strukturierter und unstrukturierter Daten. Um die Erfassung von Gesundheitsdaten zu erleichtern, hat Databricks Konnektoren für anwendungsspezifische Datentypen wie elektronische Krankenakten und Genomik entwickelt. Diese Konnektoren werden zusammen mit Datenmodellen nach Branchenstandard als Bestandteil von Solution Accelerators bereitgestellt. Darüber hinaus bietet Delta Lake integrierte Optimierungen für das Zwischenspeichern und Indizieren von Daten, um die Geschwindigkeit der Datenverarbeitung deutlich zu erhöhen. Mit diesen Funktionen können Teams alle Rohdaten an einem zentralen Ort zusammenführen und sie dann so aufbereiten, dass eine ganzheitliche Darstellung der Patientengesundheit entsteht.
- **Umsetzen aller Patientenanalysen und der gesamten KI.** Da alle Daten in einem Lakehouse zentral zusammengeführt sind, können Teams darauf aufbauend leistungsstarke Patientenanalysen und Prognosemodelle erstellen. Zu diesem Zweck bietet Databricks kollaborative Arbeitsbereiche mit einem Komplettpaket von Analyse- und KI-Tools sowie Unterstützung für eine breite Palette von Programmiersprachen wie SQL, R, Python und Scala. So kann eine vielfältige Gruppe von Anwendern, wie z. B. Data Scientists, Engineers und Medizininformatiker, alle Ihre Gesundheitsdaten gemeinsam analysieren, modellieren und visualisieren.

- **Bereitstellen von Echtzeiterkenntnissen zu Patienten.** Das Lakehouse bietet eine einheitliche Architektur für Streaming- und Batch-Daten. Es ist nicht mehr notwendig, zwei separate Architekturen vorzuhalten oder sich mit Problemen der Zuverlässigkeit auseinanderzusetzen. Durch die Ausführung der Lakehouse-Architektur auf Databricks haben Unternehmen außerdem Zugang zu einer Cloud-nativen Plattform, die je nach Arbeitsauslastung automatisch skaliert. Dies erleichtert das Erfassen von Streaming-Daten und das Zusammenführen mit Verlaufsdaten im Petabyteumfang und erlaubt es, auf Bevölkerungsebene Erkenntnisse in Quasi-Echtzeit zu gewinnen.
- **Sicherstellen von Datenqualität und Compliance.** Um die Authentizität der Daten zu gewährleisten, bietet das Lakehouse Funktionen, die in herkömmlichen Data Lakes fehlen, z. B. Schemadurchsetzung, Auditing, Versionierung und eine differenziert konfigurierbare Zugriffskontrolle. Ein wesentlicher Vorteil des Lakehouse ist die Möglichkeit, Analysen und ML basierend auf derselben, vertrauenswürdigen Datenquelle durchzuführen. Darüber hinaus bietet Databricks Funktionen für Tracking und Verwaltung von ML-Modellen, die es den Teams erleichtern, Ergebnisse in verschiedenen Umgebungen zu reproduzieren und die Einhaltung von Compliance-Standards zu gewährleisten. Alle diese Funktionen werden in einer HIPAA-konformen Analyseumgebung bereitgestellt.

Dieses Lakehouse ist die für die Verwaltung von Daten aus dem Gesundheitswesen und den Biowissenschaften am besten geeignete Architektur. Durch die Kombination dieser Architektur mit den Funktionen von Databricks können Unternehmen eine breite Palette wirkungsstarker Anwendungsfälle von der Arzneimittelforschung bis hin zu Disease-Management-Programmen unterstützen.

Aufbauen eines Lakehouse für Gesundheitswesen und Biowissenschaften

Wie bereits erwähnt, freuen wir uns, eine Reihe von Solution Accelerators anbieten zu können, die Unternehmen im Gesundheitswesen und den Biowissenschaften beim Aufbau eines Lakehouse für ihre spezifischen Anforderungen unterstützen. Unsere Solution Accelerators enthalten Beispieldaten, einsatzfertigen Code und Schritt-für-Schritt-Anleitungen in einem Databricks-Notebook.

Neuer Solution Accelerator: Lakehouse für den Praxiseinsatz Daten aus der Praxis liefern den Pharmaunternehmen abseits von Studien neue Erkenntnisse über die Patientengesundheit und die Wirksamkeit von Medikamenten. Dieser Solution Accelerator hilft Ihnen, auf der Grundlage von Databricks ein Lakehouse für den Praxiseinsatz aufzubauen. Wir zeigen Ihnen, wie Sie Stichprobendaten aus elektronischen Gesundheitsakten für eine Patientenpopulation einlesen, die Daten auf Basis des OMOP Common Data Model strukturieren und dann umfassende Analysen durchführen, um beispielsweise Muster für die Medikamentenverschreibung zu untersuchen.

Experimentieren Sie mit den folgenden kostenlosen Databricks-Notebooks.

Erfahren Sie mehr zu allen unseren Lösungen für das Gesundheitswesen und die Biowissenschaften.

databricks

KAPITEL 2.4

Aktualität und Zuverlässigkeit beim Übermitteln von aufsichtsrechtlichen Meldungen

von **ANTOINE AMEND** und **FAHMID KABIR**

17. September 2021

Risikosteuerung und regulatorische Compliance sind zunehmend komplexe und kostspielige Unterfangen. Seit der globalen Finanzkrise von 2008 hat der Umfang regulatorischer Änderungen um 500 % zugenommen und die Regulierungskosten in die Höhe getrieben. Die Bußgelder, die bei der Nichteinhaltung von Vorschriften und bei SLA-Verstößen verhängt werden, sind enorm: Allein 2019 mussten Banken über 10 Milliarden US-Dollar an Strafen für Verstöße gegen Geldwäschegesetze zahlen. In Anbetracht dessen muss die Verarbeitung von Berichten zweifelsohne auch dann erfolgen, wenn die Daten unvollständig sind. Andererseits wird auch eine mindere Datenqualität wegen „unzureichender Kontrolle“ mit Geldstrafen belegt. Infolgedessen finden sich viele Finanzdienstleister häufig im Spannungsfeld zwischen schlechter Datenqualität und strengen SLAs wieder und müssen zwischen Datenbelastbarkeit und Datenaktualität abwägen.

In diesem Solution Accelerator für aufsichtsrechtliche Meldungen zeigen wir Ihnen, wie Delta-Live-Tabellen die Erfassung und Verarbeitung regulatorischer Daten in Echtzeit garantieren, um regulatorische SLAs zu erfüllen. Mit der Kombination aus Delta Sharing und Delta-Live-Tabellen können sich Analysten in Echtzeit auf die Qualität der übermittelten Regulierungsdaten verlassen. In diesem Blogpost veranschaulichen wir die Vorteile der Lakehouse-Architektur, die Datenmodelle aus der Finanzdienstleistungsbranche mit der Flexibilität des Cloud-Computings kombiniert und so hohe Governance-Standards bei geringem Entwicklungsaufwand ermöglicht. Im Folgenden wird erläutert, was ein FIRE-Datenmodell ist und wie Delta Live Tables zum Aufbau robuster Datenpipelines integriert werden können.

FIRE-Datenmodell

Der FIRE-Datenstandard (Financial Regulatory) definiert eine allgemeine Spezifikation für die Übermittlung granularer Daten zwischen Regulierungssystemen im Finanzwesen. Regulatorische Daten sind dabei solche, die regulatorischen Eingaben, Anforderungen und Berechnungen zugrunde liegen und für politische, Überwachungs- und Aufsichtszwecke verwendet werden. Der FIRE-Datenstandard wird von der Europäischen Kommission, dem Open Data Institute und dem Open Data Incubator for Europe über das Förderprogramm Horizont 2020 unterstützt. Als Teil dieser Lösung haben wir ein PySpark-Modul beigesteuert, das FIRE-Datenmodelle interpretiert und in Apache Spark™-Betriebspipelines überführt.

databricks

Delta Live Tables

Databricks hat kürzlich ein neues Produkt für die Orchestrierung von Datenpipelines angekündigt: Delta Live Tables reduziert den Entwicklungs- und Verwaltungsaufwand für zuverlässige Datenpipelines auf Unternehmensebene. Da das FIRE-Datenmodell die Möglichkeit bietet, mehrere Erwartungen auszuwerten und ungültige Datensätze zu verwerfen oder in Echtzeit zu überwachen, liegen die Vorteile seiner Integration in Delta Live Tables auf der Hand. Wie in der folgenden Architektur veranschaulicht, **erfasst** Delta Live Tables granulare regulatorische Daten, die im Cloud-Speicher abgelegt werden, **schematisiert** Inhalte und **validiert** Datensätze gemäß der FIRE-Datenspezifikation auf Konsistenz. Lesen Sie weiter, um zu erfahren, wie wir unter Verwendung von Delta Sharing den sicheren, skalierbaren und transparenten Austausch granularer Informationen zwischen Regulierungssystemen demonstrieren.

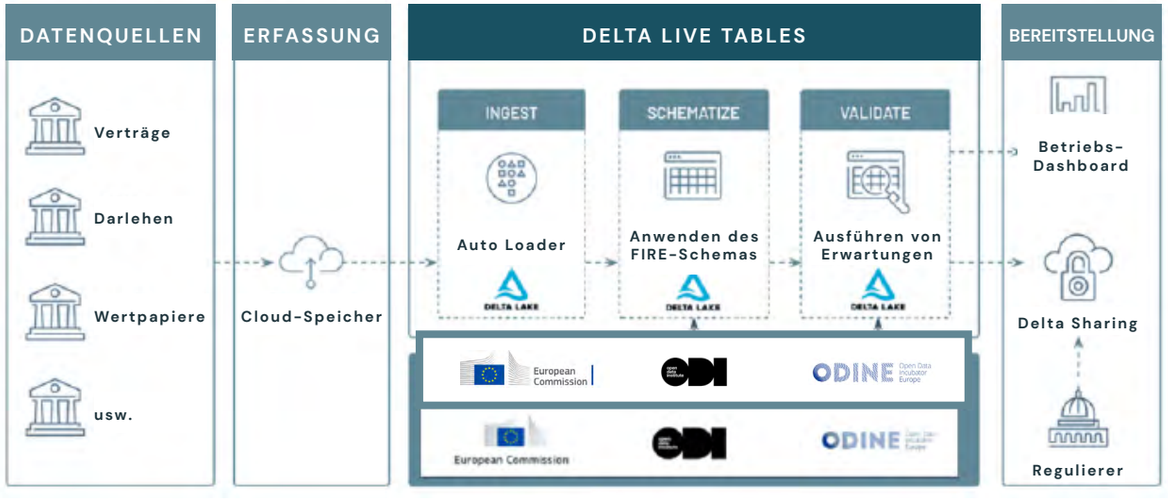


Abbildung 1

Schemaerzwingung

Auch wenn einige Datenformate strukturiert aussehen (z. B. JSON-Dateien), ist die Erzwingung eines Schemas nicht nur eine gute technische Praxis, sondern garantiert in Unternehmen und insbesondere im Hinblick auf die Einhaltung gesetzlicher Vorschriften, dass alle fehlenden Felder erwartet, unerwartete Felder verworfen und Datentypen vollständig ausgewertet werden (so muss beispielsweise ein Datum als Datumsobjekt und nicht als Zeichenfolge behandelt werden). Außerdem werden Ihre Systeme auf möglichen Datendrift geprüft. Mithilfe des FIRE-PySpark-Moduls rufen wir programmgesteuert das Spark-Schema ab, das für die Verarbeitung einer gegebenen FIRE-Entität (in diesem Beispiel eine Sicherheitsentität) erforderlich ist, die wir auf einen Rohdatenstrom anwenden.

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_schema = fire_model.schema
```

Im nachstehenden Beispiel wird das Schema für eingehende CSV-Dateien erzwungen. Durch Auszeichnen dieses Prozesses mit der @dlt-Annotation definieren wir unseren Einstiegspunkt in unsere Delta Live Table. Hierzu lesen wir CSV-Dateien mit Rohdaten aus einem eingebundenen Verzeichnis aus und schreiben schematisierte Datensätze auf eine Bronze-Ebene.

```
@dlt.create_table()
def collateral_bronze():
  return (
    spark
      .readStream
      .option("maxFilesPerTrigger", "1")
      .option("badRecordsPath", "/path/to/invalid/collateral")
      .format("csv")
      .schema(fire_schema)
      .load("/path/to/raw/collateral")
```

Auswerten von Erwartungen

Das Anwenden eines Schemas ist die eine Sache, das Erzwingen seiner Beschränkungen aber eine ganz andere. Der Schemadefinition einer FIRE-Entität (siehe unser Beispiel für die Definition des Sicherheitsschemas) können wir entnehmen, ob ein Feld erforderlich ist oder nicht. Bei einem Enumerationsobjekt vergewissern wir uns, dass seine Werte konsistent sind (z. B. der Währungscode). Zusätzlich zu den technischen Einschränkungen aus dem Schema meldet das FIRE-Modell auch geschäftliche Erwartungen, wie z. B. Minima, Maxima, monetäre Werte und maxItems. Alle diese technischen und geschäftlichen Einschränkungen werden programmgesteuert aus dem FIRE-Datenmodell abgerufen und als Folge von Spark SQL-Ausdrücken interpretiert.

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_constraints = fire_model.constraints
```

Mit Delta Live Tables können Benutzer mehrere Erwartungen gleichzeitig auswerten und so ungültige Datensätze verwerfen, die Datenqualität problemlos überwachen oder eine Pipeline komplett abbrechen. In unserem konkreten Szenario möchten wir Datensätze verwerfen, die keine unserer Erwartungen erfüllen, und später in einer Quarantänetabelle speichern, wie in den Notebooks in diesem Blog angegeben.

```
@dlt.create_table()
@dlt.expect_all_or_drop(fire_constraints)
def collateral_silver():
  return dlt.read_stream("collateral_bronze")
```

Mit nur wenigen Codezeilen haben wir sichergestellt, dass unsere Silver-Tabelle sowohl syntaktisch (gültiges Schema) als auch semantisch (gültige Erwartungen) korrekt ist. Wie unten gezeigt, erhalten die Compliance-Beauftragten in Echtzeit einen vollständigen Überblick über die Anzahl der verarbeiteten Datensätze. In diesem konkreten Beispiel haben wir dafür gesorgt, dass unsere Sicherheitsentität zu exakt 92,2 % vollständig ist (die verbleibenden 7,8 % werden in die Quarantäne überführt).

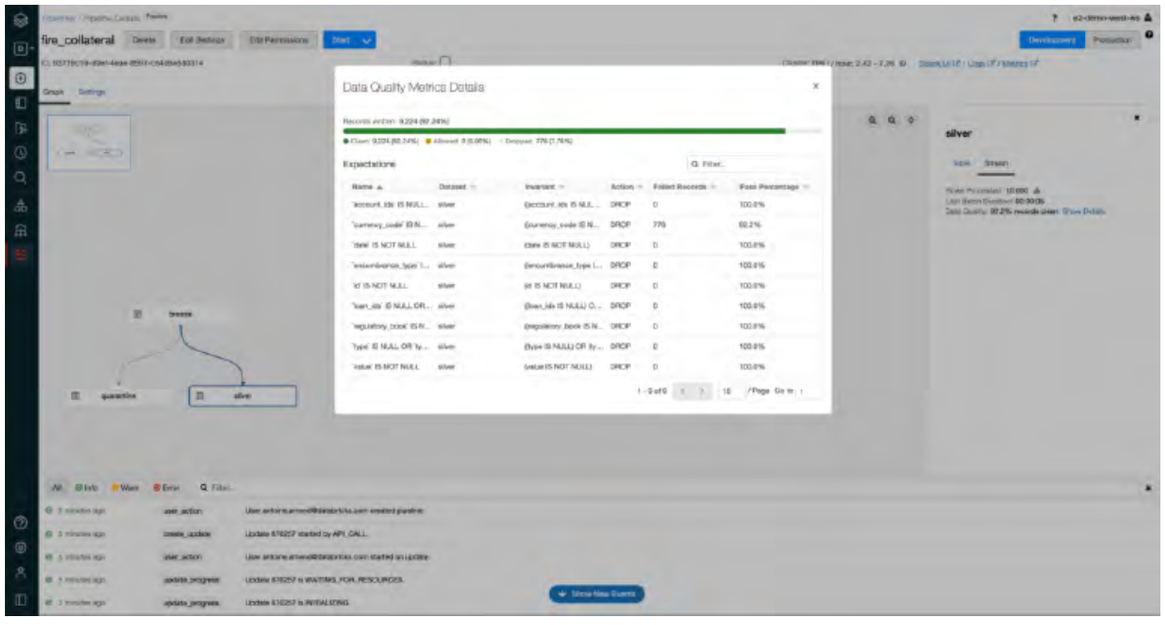


Abbildung 2

Operationsdatenspeicher

Zusätzlich zu den eigentlichen Daten, die als Deltadateien gespeichert werden, speichert Delta Live Tables auch Betriebsmetriken in einem „Deltaformat“ unter system/events. Indem wir neue Operationsmetriken mit Auto Loader „abonnieren“ und Systemereignisse im Batch oder in Echtzeit verarbeiten, wenn sich neue Metriken zeigen, folgen wir einem Standardmuster der Lakehouse-Architektur. Dank des Transaktionsprotokolls von Delta Lake, das jede Datenänderung protokolliert, können Unternehmen auf neue Metriken zugreifen, ohne eine eigene Checkpointprüfung entwickeln und pflegen zu müssen.

```
input_stream = spark \
    .readStream \
    .format(“delta”) \
    .load(“/path/to/pipeline/system/events”)

output_stream = extract_metrics(input_stream)

output_stream \
    .writeStream \
    .format(“delta”) \
    .option(“checkpointLocation”, “/path/to/checkpoint”) \
    .table(metrics_table)
```

Da alle Metriken zentral in einem Operationsspeicher verfügbar sind, können Analysten mit Databricks SQL einfache Dashboarding-Funktionen oder komplexere Warnmechanismen erstellen, um Datenqualitätsprobleme in Echtzeit zu erkennen.

	entity	expectation_name	expectation_value
1	adjustment	[id] is mandatory	`id` IS NOT NULL
2	adjustment	[date] is mandatory	`date` IS NOT NULL
3	adjustment	[col] is mandatory	`col` IS NOT NULL
4	adjustment	[contribution_amount] is mandatory	`contribution_amount` IS NOT NULL
5	adjustment	[currency_code] is mandatory	`currency_code` IS NOT NULL
6	adjustment	[currency_code] not allowed value	(`currency_code` IS NULL) OR (`currency_code` IN ('AED', 'AFN', 'ALL', 'AMD', 'ANG', 'AOA', 'ARS', 'AUD', 'AWG', 'AZN', 'BAM', 'BBD', 'BDT', 'BGN', 'BHD', 'BIF', 'BMD', 'BND', 'BOB', 'BOV', 'BRL', 'BSD', 'BTN', 'BWP', 'BYN', 'BZD', 'CAD', 'CDF', 'CHE', 'CHF', 'CHW', 'CLF', 'CLP', 'CNY', 'COP', 'COU', 'CRC', 'CUC', 'CUP', 'CVE', 'CZK', 'DJF', 'DKK', 'DOP', 'DZD', 'EGP', 'ERN', 'ETB', 'EUR', 'FJD', 'FKP', 'GBP', 'GEL', 'GHS', 'GIP', 'GMD', 'GNF', 'GTQ', 'GYD', 'HKD', 'HNL', 'HRK', 'HTG', 'HUF', 'I...

Abbildung 3

Der Aspekt der Unveränderlichkeit des Delta Lake-Formats in Verbindung mit der von Delta Live Tables gebotenen Transparenz bei der Datenqualität ermöglicht Finanzinstituten eine „Zeitreise“ zu bestimmten Versionen ihrer Daten, die sowohl dem Volumen als auch der Qualität nach den regulatorischen Anforderungen entsprechen. In unserem konkreten Beispiel hat das erneute Zuführen der 7,8 % ungültigen Datensätze aus der Quarantäne das Anhängen einer anderen Delta-Version an unsere Silver-Tabelle zur Folge – einer Version, die an Regulierungsstellen weitergegeben werden kann.

```
DESCRIBE HISTORY fire.collateral_silver
```

Übermittlung regulatorischer Daten

Mit uneingeschränktem Vertrauen in Datenqualität und -volumen können Finanzinstitute mithilfe von Delta Sharing, einem offenen Protokoll für den Austausch von Unternehmensdaten, sicher Informationen zwischen regulatorischen Systemen austauschen. Aufgrund des Open-Source-Charakters von Delta Lake sind die Endnutzer für die Nutzung der Daten (z. B. Zugriff auf Datendateien über einen SFTP-Server) weder auf die gleiche Plattform beschränkt noch auf komplexe ETL-Pipelines angewiesen, sondern können über Python, Spark oder direkt über MI/BI-Dashboards (wie Tableau oder Power BI) nativ auf schematisierte Daten zugreifen.

Wir könnten unsere Silver-Tabelle zwar in unveränderter Form freigeben, doch sollten wir trotzdem Geschäftsregeln verwenden, die eine Freigabe regulatorischer Daten nur dann gestatten, wenn ein vordefinierter Schwellenwert für die Datenqualität erreicht ist. Im folgenden Beispiel erstellen wir einen Klon unserer Silver-Tabelle in einer anderen Version und an einem bestimmten Speicherort, der von unseren internen Netzwerken getrennt, für Endbenutzer dagegen zugänglich ist. (Einen solchen Speicherort bezeichnet man auch als „Demilitarized Zone“, kurz „DMZ“.)

```
from delta.tables import *

deltaTable = DeltaTable.forName(spark, “fire.collateral_silver”)
deltaTable.cloneAtVersion(
  approved_version,
  dmz_path,
  isShallow=False,
  replace=True
)

spark.sql(
  “CREATE TABLE fire.colleral_gold USING DELTA LOCATION ‘{}’”
    .format(dmz_path)
)
```

Während die Delta Sharing-Open-Source-Lösung zum Verwalten der Zugriffsrechte auf einen Freigabeserver zurückgreift, nutzt Databricks Unity Catalog, um Richtlinien für die Zugriffssteuerung zu zentralisieren und zu erzwingen, den Benutzern vollständige Auditprotokolle bereitzustellen und die Zugriffsverwaltung über die SQL-Schnittstelle zu vereinfachen. Im folgenden Beispiel erstellen wir eine Freigabe („SHARE“), die unsere regulatorischen Tabellen enthält, und einen Empfänger („RECIPIENT“), für den wir unsere Daten freigeben.

```
-- DEFINE OUR SHARING STRATEGY
CREATE SHARE regulatory_reports;

ALTER SHARE regulatory_reports ADD TABLE fire.collateral_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.loan_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.security_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.derivative_gold;

-- CREATE RECIPIENTS AND GRANT SELECT ACCESS
CREATE RECIPIENT regulatory_body;

GRANT SELECT ON SHARE regulatory_reports TO RECIPIENT regulatory_body;
```

Jede Regulierungsbehörde und jeder Nutzer, der über eine entsprechende Berechtigung verfügt, kann mithilfe eines privaten Zugangstokens, das im Verlauf dieses Vorgangs ausgetauscht wird, auf unsere zugrunde liegenden Daten zugreifen. Weitere Informationen zu Delta Sharing finden Sie auf unserer Produktseite, oder Sie setzen sich mit Ihrem Databricks-Vertreter in Verbindung.

Compliance auf dem Prüfstand

Mit den Notebooks dieser Reihe und Delta Live Tables-Jobs haben wir die Vorteile der Lakehouse-Architektur beim Erfassen, Verarbeiten, Validieren und Übertragen regulatorischer Daten veranschaulicht. Konkret ging es um die Notwendigkeit für Unternehmen, Konsistenz, Integrität und Aktualität regulatorischer Pipelines zu gewährleisten. Dies ließ sich mit einem allgemeinen Datenmodell (FIRE) in Verbindung mit einer flexiblen Orchestrierungs-Engine (Delta Live Tables) leicht umsetzen. Mit den Delta Sharing-Funktionen haben wir schließlich gezeigt, wie die Finanzdienstleistungsbranche vollständige Transparenz und Vertrauen in die zwischen verschiedenen Regulierungssystemen ausgetauschten regulatorischen Daten schaffen und gleichzeitig Meldeanforderungen erfüllen, Betriebskosten senken und neue Standards implementieren kann.

Machen Sie sich mithilfe der beigefügten Notebooks mit der FIRE-Datenpipeline vertraut und besuchen Sie unseren Solution Accelerators Hub, um sich über unsere aktuellen Lösungen für Finanzdienstleister auf dem Laufenden zu halten.

Experimentieren Sie mit den folgenden kostenlosen Databricks-Notebooks.

databricks

KAPITEL 2.5

Umfassende Lösungen zur Geldwäschebekämpfung mit der Databricks Lakehouse-Plattform

von SRI GHATTAMANENI , RICARDO PORTILLA und ANINDITA MAHAPATRA

16. JULI 2021

Beseitigen der zentralen Herausforderungen beim Aufbau einer Lösung zur Bekämpfung von Finanzkriminalität

Die Einhaltung der Vorschriften zur Bekämpfung von Geldwäsche gehört zweifellos zu den wichtigsten Punkten auf der Agenda von Regulierern, die Finanzinstitute in aller Welt beaufsichtigen. So wie sich die Geldwäschebekämpfung im Laufe der Jahrzehnte weiterentwickelt hat und immer ausgeklügelter wurde, so sind auch die rechtlichen Anforderungen bei der Bekämpfung moderner Systeme zur Geldwäsche und Terrorismusfinanzierung gestiegen. Der US-amerikanische Bank Secrecy Act von 1970 gab den Finanzinstituten Leitlinien und einen Rahmen vor, um angemessene Kontrollmechanismen zur Überwachung von Finanztransaktionen und zur Meldung verdächtiger Steueraktivitäten an die zuständigen Behörden zu implementieren. Dieses Gesetz bildet den Rahmen für die Bekämpfung von Geldwäsche und Terrorfinanzierung durch Finanzinstitute.

Warum Geldwäschebekämpfung so komplex ist

Die aktuellen Maßnahmen zur Bekämpfung von Geldwäsche haben wenig Ähnlichkeit mit denen des letzten Jahrzehnts. Der Übergang zum Online-Banking, bei dem Finanzinstitute täglich viele Milliarden Transaktionen abwickeln, hat dazu geführt, dass das Ausmaß der Geldwäsche trotz strikterer Systeme zur Transaktionsüberwachung und solider Lösungen zur Legitimationsprüfung (Stichwort „Know Your Customer“) stetig zunimmt. In diesem Blogpost berichten wir über unsere Erfahrungen bei der gemeinsamen Entwicklung unternehmensweiter Lösungen zur Geldwäschebekämpfung auf der Lakehouse-Plattform mit unseren Kunden aus der Finanzbranche. Diese Plattform ermöglicht strikte Kontrolle und liefert gleichzeitig innovative und skalierbare Lösungen, die fortlaufend an die sich ändernden Gegebenheiten der Online-Geldwäsche und die zugehörigen Risiken angepasst werden können.

Aufbau einer Lösung zur Bekämpfung von Geldwäsche mit Lakehouse

Die mit der Verarbeitung von Milliarden von Transaktionen pro Tag einhergehende operative Belastung ergibt sich aus der Notwendigkeit, Daten aus vielen verschiedenen Quellen zu speichern und anspruchsvolle Next-Gen-Lösungen zur Geldwäschebekämpfung zu betreiben. Diese Lösungen bieten leistungsstarke Risikoanalysen und -berichte und unterstützen den Einsatz fortschrittlicher Machine-Learning-Modelle, um Fehlalarme zu reduzieren und die Effizienz nachgelagerter Untersuchungen zu verbessern. Die Finanzinstitute haben bereits Schritte unternommen, um die Infrastruktur- und Skalierungsprobleme zu lösen. Konkret erfolgte ein branchenweiter Umstieg von On-Premises-Architekturen in die Cloud, um Sicherheit und Agilität zu verbessern und sich die Skaleneffekte zunutze zu machen, die die notwendige Speicherung riesiger Datenmengen mit sich bringt.

Allerdings stellt sich auch die Frage, wie die enormen Mengen strukturierter und unstrukturierter Daten, die in kostengünstigen Objektspeichern erfasst und vorgehalten werden, sinnvoll zu nutzen sind. Cloud-Anbieter bieten zwar eine kostengünstige Möglichkeit, die Daten zu speichern, doch erst ihre Speicherung in hochwertigen und leistungsfähigen Formaten macht eine sinnvolle nachgelagerte Nutzung etwa beim Management von Geldwäscherisiken und für Compliance-Aktivitäten möglich. Und genau das macht die Lakehouse-Plattform von Databricks. Durch Kombination der niedrigen Speicherkosten von Data Lakes mit den robusten Transaktionsfunktionen von Data Warehouses können Finanzinstitute eine moderne Plattform zur Geldwäschebekämpfung aufbauen.

Neben den oben beschriebenen Herausforderungen bei der Datenspeicherung stehen auf diesen Bereich spezialisierte Analysten vor einigen weiteren Problemen, die für diesen Bereich spezifisch sind. So sollen Sie:

- die Time-to-Value bei der Analyse unstrukturierter Daten (z. B. Bilder, Textdaten und Netzwerklinks) verbessern,
- den DevOps-Aufwand für die Unterstützung kritischer ML-Funktionen wie die Auflösung von Entitäten, maschinelles Sehen und Graph-Analysen von Entitätsmetadaten verringern und
- durch Einführung von Analysetechniken und einer Dashboarding-Ebene für Transaktionen und erweiterten Tabellen zur Geldwäschebekämpfung Silos aufbrechen.

Zum Glück unterstützt Databricks die Lösung dieser Probleme: Mit Delta Lake werden unstrukturierte wie strukturierte Daten gespeichert und kombiniert, um Entitätsbeziehungen aufzubauen. Außerdem ermöglicht Databricks Delta Engine mit den neuen Photon-Serverressourcen zur Beschleunigung von BI-Abfragen auf Tabellen einen effizienten Zugriff auf die Daten. Hinzu kommt, dass Machine Learning im Lakehouse ein First-Class-Objekt ist, d. h., Analysten und Data Scientists müssen keine Zeit mit Subsampling oder dem Verschieben von Daten verschwenden, um Dashboards gemeinsam zu nutzen, und sind Kriminellen somit einen Schritt voraus.

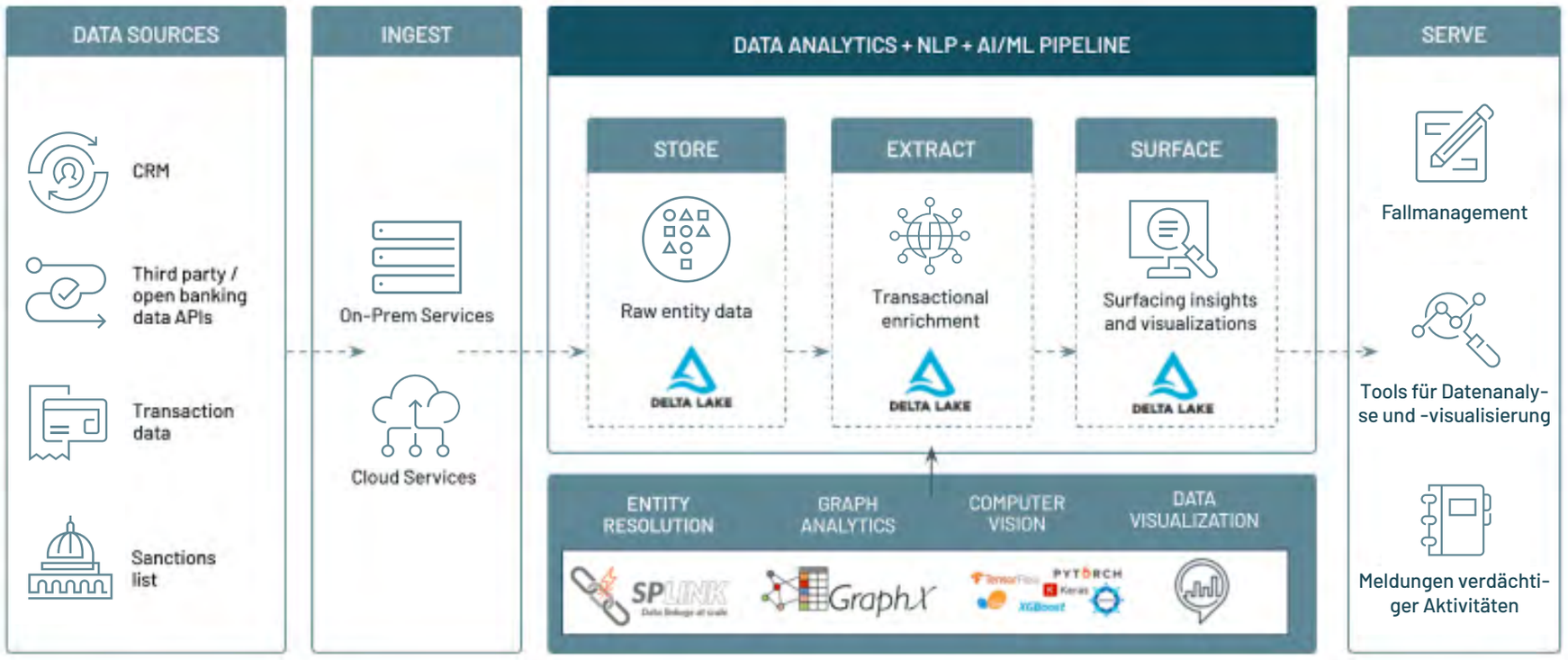


Abbildung 1

Erkennen von Geldwäschemustern mit Graph-Funktionen

Eine der wichtigsten Datenquellen, die auf Geldwäschebekämpfung spezialisierte Analysten bei einem Fall nutzen können, sind *Transaktionsdaten*. Zwar liegen diese Daten in Tabellenform vor und sind mit SQL leicht zugänglich, doch ist es ausgesprochen mühselig, Transaktionsketten, die drei oder mehr Ebenen tief sind, mit SQL-Abfragen nachzuvollziehen. Aus diesem Grund ist es wichtig, Sprachen und APIs flexibel handhaben zu können, um einfache Konzepte auszudrücken, wie z. B. ein Netzwerk verdächtiger Personen, die miteinander illegal Geschäfte abwickeln. Zum Glück ist dies mit GraphFrames, einer in der Databricks Runtime for Machine Learning vorinstallierten Graph-API, einfach zu bewerkstelligen.

In diesem Abschnitt zeigen wir, wie Graph-Analysen zur Aufdeckung von Geldwäschesystemen wie Synthetic Identities und Layering/Structuring verwendet werden können. Wir werden hierfür einen Datensatz, der aus Transaktionen sowie aus Transaktionen abgeleiteten Entitäten besteht, verwenden, um das Vorhandensein solcher Muster mit Apache Spark™, GraphFrames und Delta Lake zu erkennen. Die fortbestehenden Muster werden in Delta Lake gespeichert, sodass Databricks SQL auf die aggregierten Versionen dieser Ergebnisse auf der Gold-Ebene angewendet werden kann, um Endnutzern die Möglichkeiten der Graph-Analyse zu bieten.

Szenario 1: Synthetic Identities

Wie erwähnt, sollten bei Vorhandensein von Synthetic Identities die Alarmglocken schrillen. Mithilfe der Graph-Analyse können alle Entitäten aus unseren Transaktionen in einem Rutsch analysiert werden, um das Risikoniveau zu ermitteln. In unserer Analyse wird dies in drei Phasen durchgeführt:

- Zuerst werden auf der Grundlage der Transaktionsdaten die Entitäten extrahiert.
- Dann werden Verknüpfungen zwischen Entitäten basierend auf Adressen, Telefonnummern oder E-Mail-Adressen erstellt.
- Schließlich wird mithilfe von durch GraphFrames verbundene Komponenten festgestellt, ob mehrere Entitäten (die durch eine ID und weitere der oben genannten Attribute ausgewiesen werden) über eine oder mehrere Verknüpfungen miteinander verbunden sind.

Je nachdem, wie viele Verbindungen (d. h. gemeinsame Attribute) zwischen den Entitäten bestehen, können wir eine niedrigere oder höhere Risikobewertung zuweisen und für Gruppen mit hoher Bewertung Warnungen erstellen. Nachstehend sehen Sie eine schematische Darstellung dieses Konzepts.

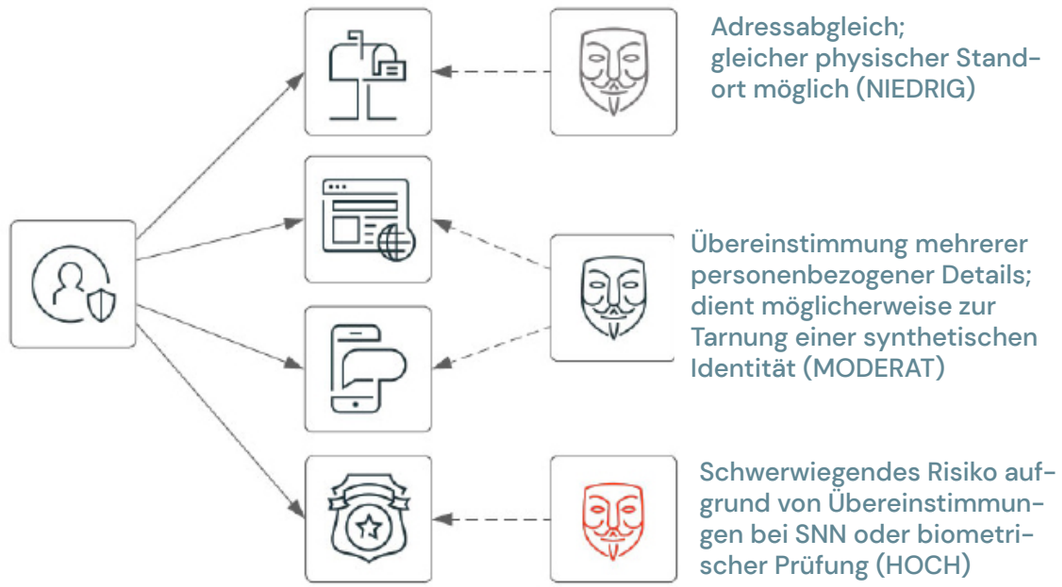


Abbildung 2

Zunächst erstellen wir einen Identitätsgraph mit Adresse, E-Mail-Adresse und Telefonnummer, um Personen zu verknüpfen, bei denen mindestens eines dieser Attribute übereinstimmt.

```
e_identity_sql = '''
select entity_id as src, address as dst from aml.aml_entities_synth  where address is not
null
UNION
select entity_id as src, email as dst from aml.aml_entities_synth  where email_addr is not
null
UNION
select entity_id as src, phone as dst from aml.aml_entities_synth  where phone_number is not
null
'''
```

```
from graphframes import *
from pyspark.sql.functions import *
aml_identity_g = GraphFrame(identity_vertices, identity_edges)
result = aml_identity_g.connectedComponents()

result \
 .select("id", "component", 'type') \
 .createOrReplaceTempView("components")
```

Als Nächstes führen wir Abfragen aus, um festzustellen, wann zwei Entitäten Überschneidungen bei personenbezogenen Daten und Bewertungen aufweisen. Ausgehend von den Ergebnissen dieser abfragenden Graphkomponenten würden wir eine Kohorte erwarten, die aus einem nur übereinstimmenden Attribut (z. B. der Postadresse) besteht, was kein allzu großer Grund zur Sorge wäre. Je mehr Attribute jedoch übereinstimmen, desto stärker sollten wir mit Problemen rechnen. Wie unten gezeigt, können wir Fälle markieren, in denen alle drei Attribute übereinstimmen, damit SQL-Analysten Tagesergebnisse aus der Graph-Analyse für alle Entitäten erhalten.

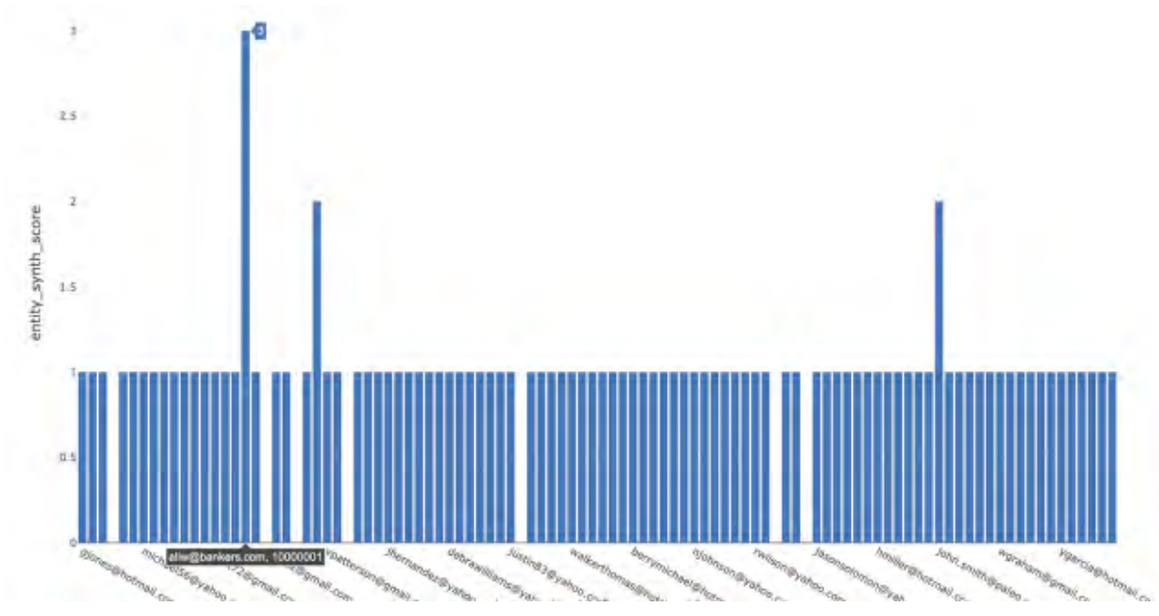


Abbildung 3

Szenario 2: Structuring

Ein weiteres häufiges Muster ist das sogenannte *Structuring*. Es tritt auf, wenn mehrere Entitäten zusammenwirken und kleinere Zahlungen an eine Anzahl von Banken senden, die unter dem Radar bleiben. Diese Banken überweisen anschließend größere Gesamtbeträge an ein einziges Institut (wie ganz unten rechts dargestellt). In diesem Szenario sind alle Parteien unter dem Schwellenwert von 10.000 Dollar geblieben, bei dem staatliche Stellen normalerweise alarmiert würden. Dies ist mit der Graph-Analyse nicht nur recht leicht zu bewerkstelligen, sondern das *Motivsuchverfahren* kann automatisiert werden, um auf andere Permutationen von Netzwerken ausgedehnt zu werden und weitere verdächtige Transaktionen auf gleiche Weise zu finden.

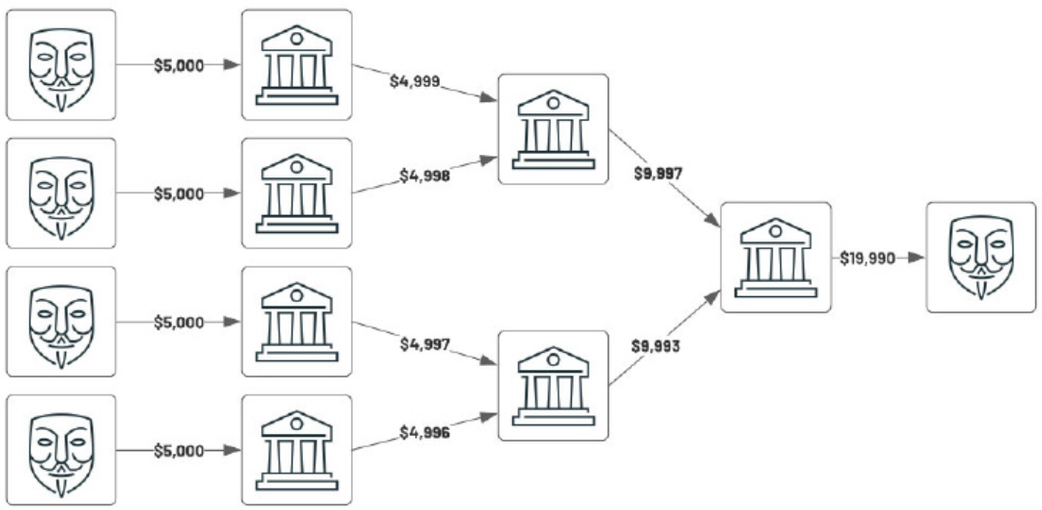


Abbildung 4

Nun wollen wir den Basiscode für die Motivsuche erstellen, um das oben beschriebene Szenario mithilfe der Graphfunktionen zu erkennen. Beachten Sie, dass es sich bei der folgenden Ausgabe um halbstrukturiertes JSON handelt. Alle Datentypen – auch unstrukturierte Typen – sind im Lakehouse leicht zugänglich. Die konkreten Ergebnisse werden wir dann für SQL-Berichte speichern.

```
motif = “(a)-[e1]->(b); (b)-[e2]->(c); (c)-[e3]->(d); (e)-[e4]->(f); (f)-[e5]-
>(c); (c)-[e6]->(g)”
struct_scn_1 = aml_entity_g.find(motif)

joined_graphs = struct_scn_1.alias(“a”) \
 .join(struct_scn_1.alias(“b”), col(“a.g.id”) == col(“b.g.id”)) \
 .filter(col(“a.e6.txn_amount”) + col(“b.e6.txn_amount”) > 10000)
```

Mit der Motivsuche konnten wir interessante Muster ermitteln, bei denen Geld über vier verschiedene Einrichtungen fließt und dabei jeweils unter dem Grenzwert von 10.000 Dollar bleibt. Wir verknüpfen unsere Graph-Metadaten mit strukturierten Datensätzen, um Erkenntnisse zu gewinnen, die ein Fachanalyst für Geldwäsche dann weiter untersuchen kann.

	top_entity_id	first_entity	second_entity	third_entity	fourth_entity
1	1	Brenda Thomas	Teresa Gibson	Mary Strong	Robert Wilkinson
2	3	Lindsey Barber	Joshua Harris	Mary Strong	Robert Wilkinson
3	5	Bruce White	Kathleen Elliott	Victor Arias	Robert Wilkinson
4	7	Jeffrey Lara	Amy Campbell	Victor Arias	Robert Wilkinson

Abbildung 5

Szenario 3: Ausbreitung von Risikobewertungen

Die ermittelten Hochrisikoentitäten haben – durch einen Vernetzungseffekt – Einfluss auf ihren Umkreis. Daher ist die Risikobewertung aller Entitäten, mit denen sie interagieren, entsprechend anzupassen, um die Einflusszone widerzuspiegeln. Mithilfe eines iterativen Ansatzes können wir den Transaktionsfluss bis zu jeglicher gegebenen Tiefe verfolgen und die Risikobewertungen weiterer betroffener Entitäten in diesem Netzwerk korrigieren. Wie bereits erwähnt, werden durch die Ausführung von Graph-Analysen mehrere wiederholte SQL-Joins und komplexe Geschäftslogik vermieden, die sich aufgrund von Beschränkungen beim Arbeitsspeicher auf die Leistung auswirken könnten. Für genau diesen Zweck wurden die Graph-Analyse und die Pregel-API entwickelt. Ursprünglich von Google stammend, ermöglichst es Pregel den Nutzern, Nachrichten von einem beliebigen Knotenpunkt rekursiv an seine entsprechenden Nachbarn „weiterzuverbreiten“, wobei der Zustand des Knotenpunkts (in diesem Fall die Risikobewertung) bei jedem Schritt aktualisiert wird. Wir können unseren dynamischen Risikoansatz mit der Pregel-API wie folgt darstellen.

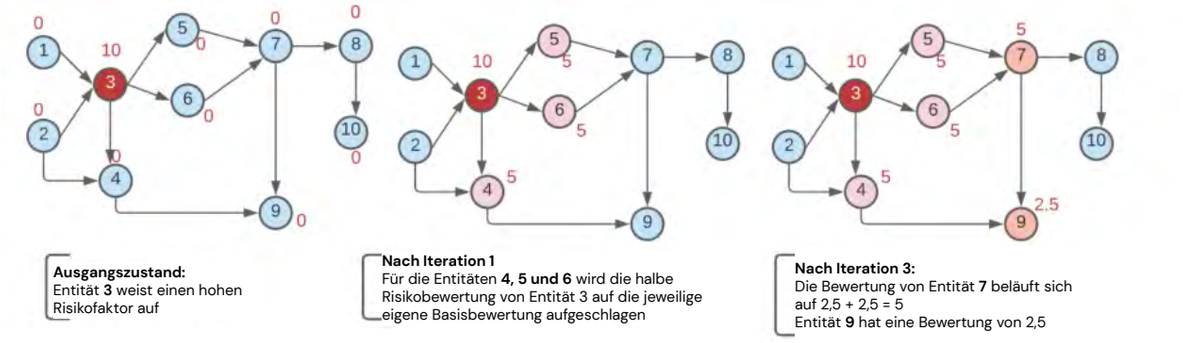


Abbildung 6

Das Diagramm unten links zeigt den Ausgangszustand des Netzes und zwei nachfolgende Iterationen. Angenommen, wir beginnen mit einem Übeltäter (Knoten 3), der eine Risikobewertung von 10 hat. Nun sollen alle Personen, die mit diesem Knoten Transaktionen durchführen und dabei Gelder erhalten (d. h. die Knoten 4, 5 und 6), sanktioniert werden. Zu diesem Zweck könnten wir beispielsweise die halbe Risikobewertung des Täters auf die Basisbewertung der verbundenen Personen aufschlagen. In der nächsten Iteration werden dann die Bewertungen aller Knoten angepasst, die den Knoten 4, 5 und 6 nachgelagert sind.

Knoten-Nr.	Iteration 0	Iteration 1	Iteration 2
1	0	0	0
2	0	0	0
3	10	10	10
4	0	5	5
5	0	5	5
6	0	5	5
7	0	0	5
8	0	0	0
9	0	0	2,5
10	0	0	0

Mit der Pregel-API aus GraphFrame können wir diese Berechnung durchführen und die geänderten Werte für weitere Anwendungen persistieren, die im Nachgang genutzt werden.

```
from graphframes.lib import Pregel

ranks = aml_entity_g.pregel \
    .setMaxIter(3) \
    .withVertexColumn(
       “risk_score”,
       col(“risk”),
       coalesce(Pregel.msg()+ col(“risk”),
       col(“risk_score”))
    ) \
    .sendMsgToDst(Pregel.src(“risk_score”)/2 )  \
    .aggMsgs(sum(Pregel.msg())) \
    .run()
```

Adressabgleich

Ein Muster, auf das wir noch kurz eingehen wollen, ist der Adressabgleich von Text mit echten Street View-Bildern. Häufig müssen Analysten, die im Bereich der Geldwäschebekämpfung tätig sind, die Echtheit von Adressen überprüfen, die mit den hinterlegten Entitäten verknüpft sind. Handelt es sich bei der Adresse um ein Geschäftsgebäude, einen Wohnkomplex oder nur einen Briefkasten? Allerdings ist die Analyse von Bildern oft ein manuell durchzuführender und daher mühsamer und zeitaufwendiger Vorgang, denn sie müssen gesucht, bereinigt und dann ausgewertet werden. Mit einer Lakehouse-Datenarchitektur können wir einen Großteil dieser Aufgabe mit Python und ML-Runtimes mit PyTorch und vortrainierten Open-Source-Modellen automatisieren. Nachstehend sehen Sie ein Beispiel für eine Adresse, die nach menschlichem Ermessen gültig ist. Zur Automatisierung der Validierung verwenden wir ein vortrainiertes VGG-Modell, für das es Hunderte von gültigen Objekten gibt, die wir zur Erkennung einer Immobilie verwenden können.

Abbildung 7

Mit dem nachstehenden Code, der täglich automatisiert ausgeführt werden kann, haben wir nun alle unsere Bilder beschriftet. Zur Vereinfachung der Abfrage haben wir alle Bildverweise und -beschriftungen in eine SQL-Tabelle geladen. Beachten Sie im unten stehenden Code, wie einfach es ist, mehrere Bilder nach den darauf gezeigten Objekten abzufragen. Die Möglichkeit, solche unstrukturierten Daten mit Delta Lake abzufragen, bedeutet für Analysten eine enorme Zeitersparnis und beschleunigt den Validierungsprozess von Tagen oder gar Wochen auf wenige Minuten.

```
from PIL import Image
from matplotlib import cm

img = Image.fromarray(img)
...

vgg = models.vgg16(pretrained=True)
prediction = vgg(img)
prediction = prediction.data.numpy().argmax()
img_and_labels[i] = labels[prediction]
```

Wenn wir beginnen, die Daten zusammenzufassen, erkennen wir, dass einige interessante Kategorien auftauchen. Wie in der nachstehenden Aufschlüsselung zu sehen ist, gibt es einige offensichtliche Markierungen wie *Terrasse*, *Wohnmobil* und *Roller*, bei denen wir erwarten würden, sie an einer Wohnadresse vorzufinden. Andererseits hat das CV-Modell in einem Bild eine Solarschüssel bei umliegenden Objekten markiert. (*Hinweis: Da wir uns auf ein Open-Source-Modell beschränken, das nicht auf einem individuellen Bildsatz trainiert wurde, ist die Kennzeichnung der Solarschüssel nicht korrekt.*) Bei näherer Betrachtung des Bildes erkennen wir sofort, dass es sich hier nicht um eine Solarschüssel handelt und – was noch wichtiger ist – die Adresse kein echtes Wohnhaus bezeichnet (wie in unserem Direktvergleich in Abbildung 7 zu sehen). Das Delta Lake-Format ermöglicht es uns, einen Verweis auf unsere unstrukturierten Daten zusammen mit einer Beschriftung für eine einfache Abfrage in unserer nachstehenden Klassifizierungsaufschlüsselung zu speichern.

Image Name	Rendered Image	Main Object	Risk Level
img_0.jpg		Patio	Low
img_1.jpg		Solar Dish	High

Abbildung 9

Address Validation

```
%sql

select label, count(1) from sml.address_tags group by label
```

▸ (2) Spark Jobs

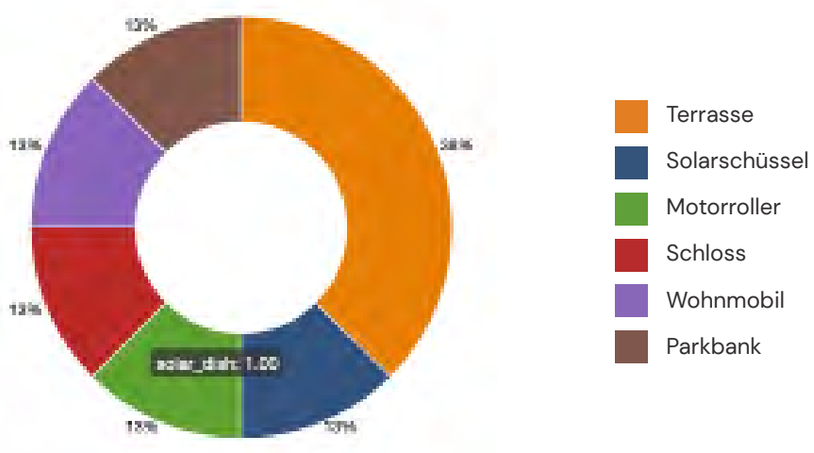


Abbildung 8

Auflösung von Entitäten

Die letzte Problemkategorie bei der Geldwäschebekämpfung, die wir betrachten wollen, ist die Auflösung von Entitäten. Für dieses Problem stehen viele Open-Source-Bibliotheken zur Verfügung. Daher entscheiden wir uns bei der einfachen unscharfen Suche für Splink. Splink implementiert eine solche Zuordnung auch im großen Maßstab und bietet Konfigurationsoptionen zur Angabe übereinstimmender Spalten sowie Sperrregeln.

Im Zusammenhang mit den Entitäten, die aus unseren Transaktionen abgeleitet werden, ist das Einfügen unserer Delta Lake-Transaktionen in den Kontext von Splink eine einfache Übung.

```
settings = {
  “link_type”: “dedupe_only”,
  “blocking_rules”: [
      “l.txn_amount = r.txn_amount”,
  ],
  “comparison_columns”: [
      {
          “col_name”: “rptd_originator_address”,
      },
      {
          “col_name”: “rptd_originator_name”,
      }
  ]
}

from splink import Splink
linker = Splink(settings, df2, spark)
df2_e = linker.get_scored_comparisons()
```

Splink gibt Übereinstimmungswahrscheinlichkeiten an, mit denen Transaktionen erkannt werden können, bei denen die Attribute der Entitäten – etwa gemeldete Postanschrift, Entitätsname oder Transaktionsbetrag – sehr ähnlich sind. In einem solchen Fall könnte dann ein Alarm ausgelöst werden. Angesichts der Tatsache, dass die Auflösung von Entitäten für den Abgleich von Kontoinformationen potenziell viel manuellen Aufwand erfordert, können Open-Source-Bibliotheken, die diese Aufgabe automatisieren und die Informationen in Delta Lake speichern, die Produktivität der Ermittler bei der Falllösung erheblich steigern. Zwar stehen für den Entitätsabgleich mehrere Optionen zur Verfügung, wir empfehlen aber den Einsatz von Locality-Sensitive Hashing (LSH), um den passenden Algorithmus für die jeweilige Aufgabe zu finden. Weitere Informationen zu LSH und seinen Vorteilen finden Sie in diesem Blogpost.

Wie bereits berichtet, haben wir bei der Adresse der NY Mellon Bank gleich einige Ungereimtheiten gefunden: „Canada Square, Canary Wharf, London, United Kingdom“ ähnelt stark „Canada Square, Canary Wharf, London, UK“. Wir können unsere de-duplizierten Datensätze in einer Delta-Tabelle speichern, die dann für Untersuchungen im Rahmen der Geldwäschebekämpfung verwendet werden kann.

unique_id_l	unique_id_r	rptd_originator_address_l	rptd_originator_address_r
223254	223256	Canada Square, Canary Wharf, London, United Kingdom	Canada Square, Canary Wharf, London, UK

Abbildung 10

Lakehouse-Dashboard für die Geldwäschebekämpfung

Mit seinem vereinfachten Datenmanagement, der beeindruckenden Leistung der neuen Photon-Abfrage-Engine und den Nebenläufigkeitseigenschaften für Nutzer schließt Databricks SQL bei Lakehouses eine Lücke. Dies ist wichtig, da viele Organisationen nicht über das Budget für überteuerte proprietäre Software zur Geldwäschebekämpfung verfügen, um die unzähligen Anwendungsfälle bei der Bekämpfung der Finanzkriminalität – wie z. B. die Bekämpfung der Terrorfinanzierung – zu unterstützen. Auf dem Markt gibt es Speziallösung für alles: für die Durchführung der oben genannten Graph-Analysen, für BI in einem Warehouse und auch für Machine Learning. Das Lakehouse-Design zur Bekämpfung der Geldwäsche führt alle drei Aspekte zusammen. Teams, die eine solche Datenplattform einsetzen, können Delta Lake zu den geringeren Kosten eines Cloud-Speichers nutzen und gleichzeitig problemlos Open-Source-Technologien integrieren, um kuratierte Berichte auf Grundlage von Graph-Technologie, maschinellem Sehen und SQL-Analytics-Technik zu erstellen. In Abbildung 11 zeigen wir eine mögliche Umsetzung für das Reporting bei der Geldwäschebekämpfung.

Die angehängten Notebooks erstellten ein Transaktionsobjekt, ein Entitätsobjekt und Übersichten zu potentiellen Structuring-Nutzern, Synthetic-Identities-Ebenen und Adressklassifizierungen unter Verwendung vortrainierter Modelle. In der nachstehenden Databricks SQL-Visualisierung haben wir unsere Photon-SQL-Engine verwendet, um auf Basis der erstellten Zusammenfassungen und der integrierten Visualisierung innerhalb weniger Minuten ein Reporting-Dashboard anzulegen. Für beide Tabellen und das Dashboard selbst gibt es vollständige ACLs, damit die Nutzer den Bericht mit Führungskräften und Datenteams teilen können, und eine Zeitplanungsfunktion zur regelmäßigen Ausführung dieses Berichts ist ebenfalls integriert. Das Dashboard ist die Krone der in die Geldwäschebekämpfungslösung integrierten KI-, BI- und Analysetechnik.

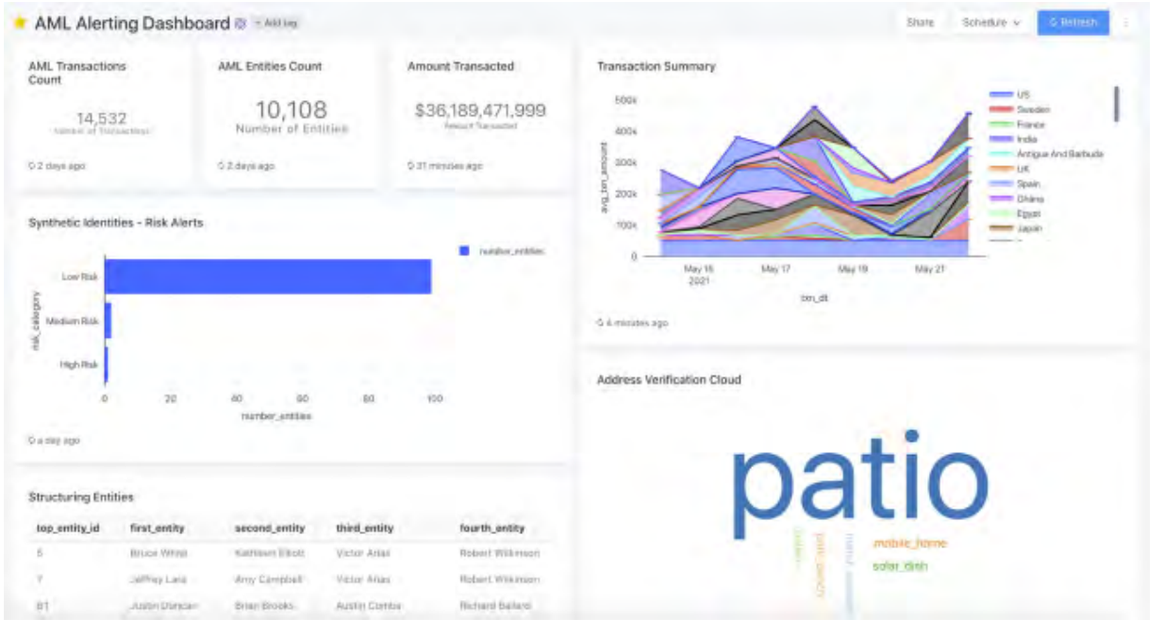


Abbildung 11

Die Transformation des Open Banking

Das Aufkommen des Open Banking ermöglicht es Finanzinstituten, durch den Austausch von Daten zwischen ihnen, den Verbrauchern, und Drittanbietern mithilfe von APIs ein besseres Kundenerlebnis zu bieten. Ein Beispiel hierfür ist die Zahlungsdiensterichtlinie (PSD2), die Finanzdienstleistungen im EU-Raum im Rahmen der Open Banking Europe-Initiative transformiert hat. Infolgedessen haben die Finanzinstitute Zugriff auf mehr Daten von vielen Banken und Dienstleistern erhalten, unter anderem auch auf Kundenkonto- und Transaktionsdaten. Dieser Trend hat sich aufgrund der jüngsten Leitlinien des US-amerikanischen Financial Crimes Enforcement Network (FinCEN) gemäß Abschnitt 314(b) des USA Patriot Act auch auf den Bereich von Betrug und Finanzkriminalität ausgedehnt. Die erfassten Finanzinstitute können Informationen über Personen, Entitäten, Unternehmen usw., bei denen der Verdacht auf mögliche Beteiligung an Geldwäsche besteht, nun mit anderen Finanzinstituten sowie zwischen in- und ausländischen Vertretungen austauschen.

Die Bestimmungen zum Informationsaustausch tragen zwar zur Transparenz bei und schützen die Finanzsysteme der USA vor Geldwäsche und Terrorismusfinanzierung, doch muss der Informationsaustausch unter Verwendung von Protokollen mit geeigneten Schutzmechanismen für Datenschutz und Datensicherheit erfolgen. Um das Problem des sicheren Informationsaustauschs zu lösen, kündigte Databricks kürzlich mit Delta Sharing ein offenes und sicheres Protokoll für die Datenfreigabe an. Unter Verwendung vertrauter Open-Source-APIs wie Pandas und Spark können Datenerzeuger und -verbraucher Daten nun über sichere und offene Protokolle austauschen, und alle Datentransaktionen können zwecks Einhaltung der Vorschriften des FinCEN einem vollständigen Audit unterzogen werden.

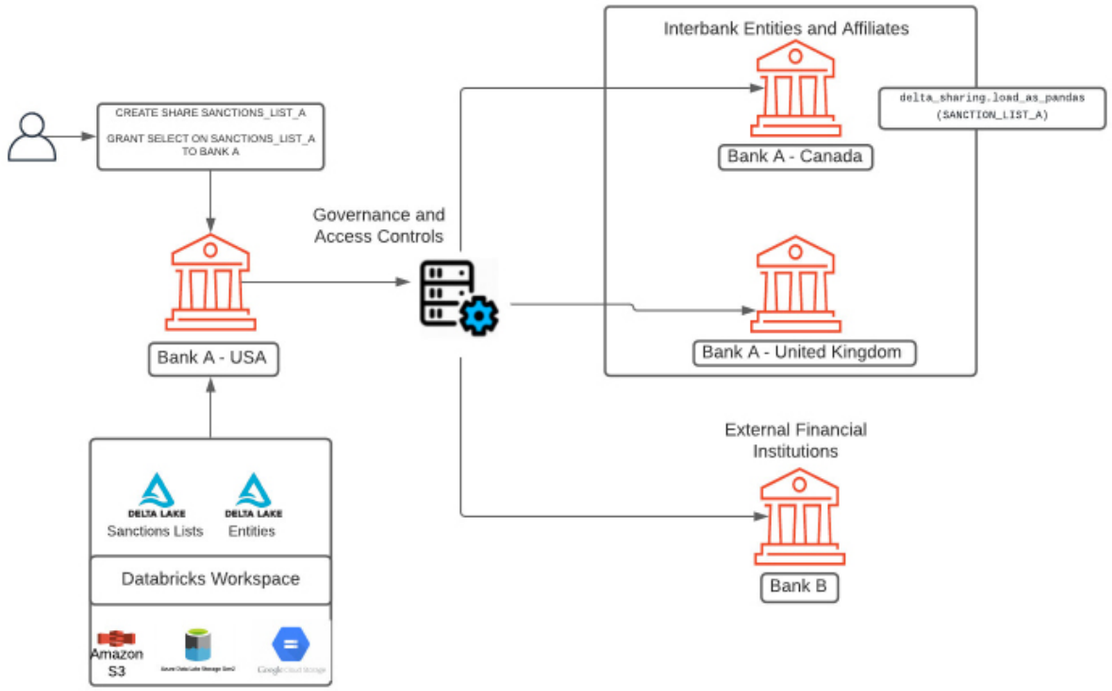


Abbildung 12

Fazit

Die Lakehouse-Architektur ist die am besten skalierbare und vielseitigste Plattform, wenn es darum geht, Analysten bei der Geldwäschebekämpfung zu unterstützen. Lakehouse unterstützt Anwendungsfälle, die von der unscharfen Suche über die Bildanalyse bis hin zu BI mit integrierten Dashboards reichen. Dieser Funktionsumfang ermöglicht es Unternehmen, die Gesamtbetriebskosten verglichen mit einschlägigen proprietären Lösungen zu senken. Das auf Finanzdienstleistung spezialisierte Databricks-Team arbeitet gegenwärtig an Lösungen für eine Vielzahl geschäftlicher Probleme im Finanzdienstleistungsbereich und ermöglicht es Data-Engineering- und Data-Science-Profis, ihre Databricks-Journey in Bereichen wie der Geldwäschebekämpfung mithilfe von Solution Accelerators anzutreten.

Experimentieren Sie mit den folgenden kostenlosen Databricks-Notebooks

- **Einführung in die Graphentheorie für die Geldwäschebekämpfung**
- **Einführung in maschinelles Sehen für die Geldwäschebekämpfung**
- **Einführung in die Entitätenauflösung für die Geldwäschebekämpfung**

databricks

KAPITEL 2.6

Aufbau eines KI-Modells zur Echtzeiterkennung von toxischem Verhalten in Spielen

von **DAN MORRIS** und **DUNCAN DAVIS**

16. Juni 2021

In MMO-Games (Massively Multiplayer Online), MOBAs (Multiplayer Online Battle Arenas) und Online-Spielen anderer Typen interagieren die Spieler ständig in Echtzeit miteinander, um sich zu koordinieren oder zu messen, während sie alle auf dasselbe Ziel hinarbeiten: den Sieg. Diese Interaktivität ist ein wesentlicher Bestandteil der Spieldynamik, gleichzeitig aber auch ein idealer Nährboden für toxisches Verhalten – ein Problem, das in der Welt der Online-Videospiele allgegenwärtig ist.

Toxisches Verhalten manifestiert sich in vielen Formen, wie z. B. in den verschiedenen Graden von Griefing, Cybermobbing und sexueller Belästigung, die im von Behaviour Interactive stammenden Diagramm oben rechts dargestellt sind, wo die verschiedenen Interaktionen im Multiplayer-Spiel „Dead by Night“ gezeigt werden.

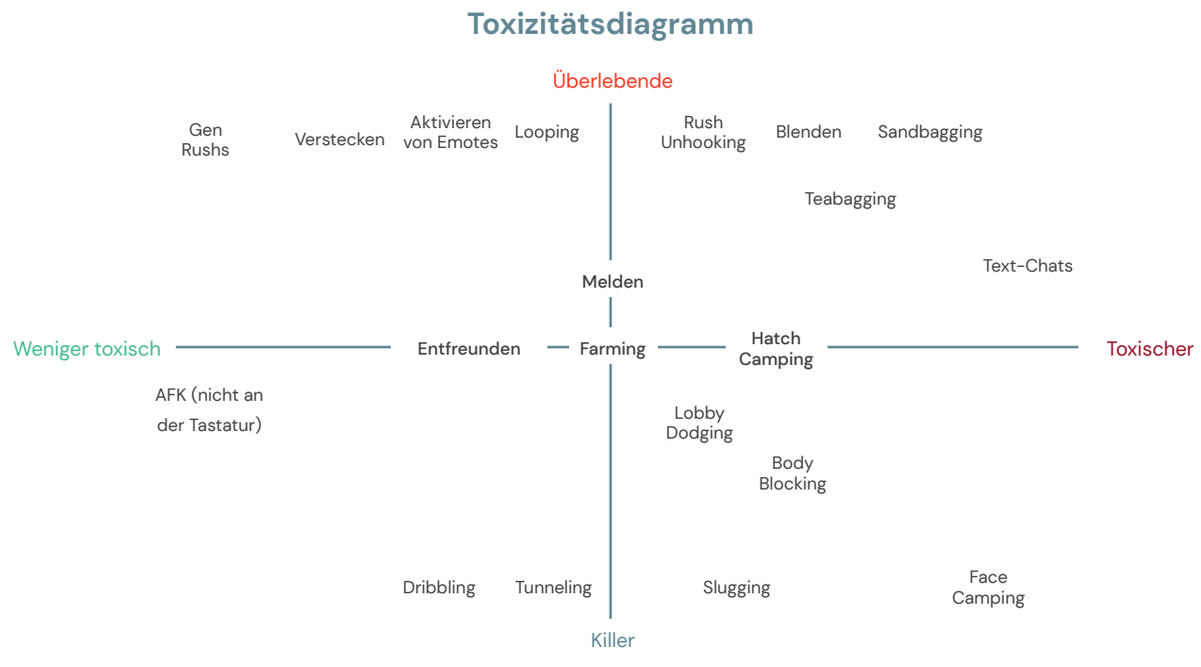


Abbildung 1
Diagramm der toxischen Interaktionen, die Spieler erleben

Neben dem persönlichen Leid, das bei Spielern und der Community verursacht werden kann – ein Problem, das gar nicht hoch genug eingeschätzt werden kann –, - beeinträchtigt toxisches Verhalten auch den finanziellen Gewinn vieler Spielestudios. So hat eine Studie der Michigan State University ergeben, dass 80 % der Spieler in letzter Zeit Opfer toxischen Verhaltens geworden sind, und 20 % von ihnen gaben an, dem betreffenden Spiel aufgrund dessen den Rücken gekehrt zu haben. In ähnlicher Weise hat eine Studie der Universität Tilburg gezeigt, dass eine störende oder toxische Begegnung beim erstmaligen Spielen zur Folge hatte, dass Spieler mehr als dreimal so häufig nicht mehr weiterspielten. Angesichts der Tatsache, dass die Spielerbindung für viele Studios oberste Priorität hat – vor allem, weil sich die Bereitstellung von Spielen immer stärker von physischen Medien auf langlebige Dienste verlagert –, liegt es nahe, dass Toxizität eingedämmt werden muss.

databricks

Zum Abwanderungsproblem kommt erschwerend hinzu, dass einige Unternehmen schon früh in der Entwicklung, teils sogar vor der Markteinführung, mit Toxizitätsproblemen konfrontiert werden. So war beispielsweise Crucible von Amazon zum Testen nur ohne Text- oder Voice-Chat freigegeben worden, weil kein System für Überwachung und Management toxischer Spieler und Interaktionen vorhanden war. Dies zeigt, dass die Dimensionen des Gaming-Bereichs die Fähigkeit der meisten Teams, solches Verhalten durch Reporting oder Eingriff in Störinteraktionen einzugrenzen, bei Weitem übersteigen. Aus diesem Grund ist es für die Studios unerlässlich, Analysen schon frühzeitig im Entwicklungszyklus in die Spiele zu integrieren und Konzepte für den Umgang mit toxischem Handeln zu entwerfen.

Toxizität in Spielen ist ganz klar ein vielschichtiges Problem, das mittlerweile Teil der Videospielkultur geworden ist. Daher gibt es keinen Universalansatz, mit dem sie angegangen werden könnte. Trotzdem kann die Bekämpfung von toxischem Verhalten in Game-Chats erhebliche Auswirkungen haben. Das gilt vor allem aufgrund der Häufigkeit toxischen Handelns und der Möglichkeit, es mithilfe von NLP-Verfahren (Natural Language Processing) automatisch zu erkennen.

Der Solution Accelerator „Toxicity Detection in Gaming“ von Databricks

Mit Daten zu toxischen Kommentaren von Jigsaw und Spieldaten aus Dota 2 führt dieser Solution Accelerator den Nutzer durch die Schritte, die erforderlich sind, um toxische Kommentare in Echtzeit zu erkennen. Hierbei kommen NLP und Ihr bestehendes Lakehouse zum Einsatz. Für NLP verwendet dieser Solution Accelerator Spark NLP von John Snow Labs, eine quelloffene Lösung auf Unternehmensniveau, die nativ auf Apache Spark™ aufsetzt.

Sie werden in diesem Solution Accelerator die folgenden Schritte durchführen:

- Jigsaw- und Dota 2-Daten mithilfe von Delta Lake in Tabellen laden
- Toxische Kommentare mit der Multilabel-Klassifizierung (Spark NLP) klassifizieren
- Mit MLflow Experimente beobachten und Modelle registrieren
- Rückschlüsse aus Batch- und Streaming-Daten anwenden
- Auswirkungen von Toxizität auf die Spieldaten untersuchen

Erkennen von Toxizität im Game-Chat in der Produktion

Mit diesem Solution Accelerator können Sie die Toxizitätserkennung jetzt noch einfacher in Ihre eigenen Spiele integrieren. Die unten stehende Referenzarchitektur zeigt beispielsweise, wie Chat- und Spieledaten aus verschiedenen Quellen wie etwa Streams, Dateien, Sprach- oder Betriebsdatenbanken übernommen und mithilfe von Databricks erfasst, gespeichert und in Feature-Tabellen für Machine-Learning-Pipelines, In-Game-ML, BI-Tabellen für Analysen und sogar für die direkte Interaktion mit Tools, die in der Community-Moderation verwendet werden, kuratiert werden können.

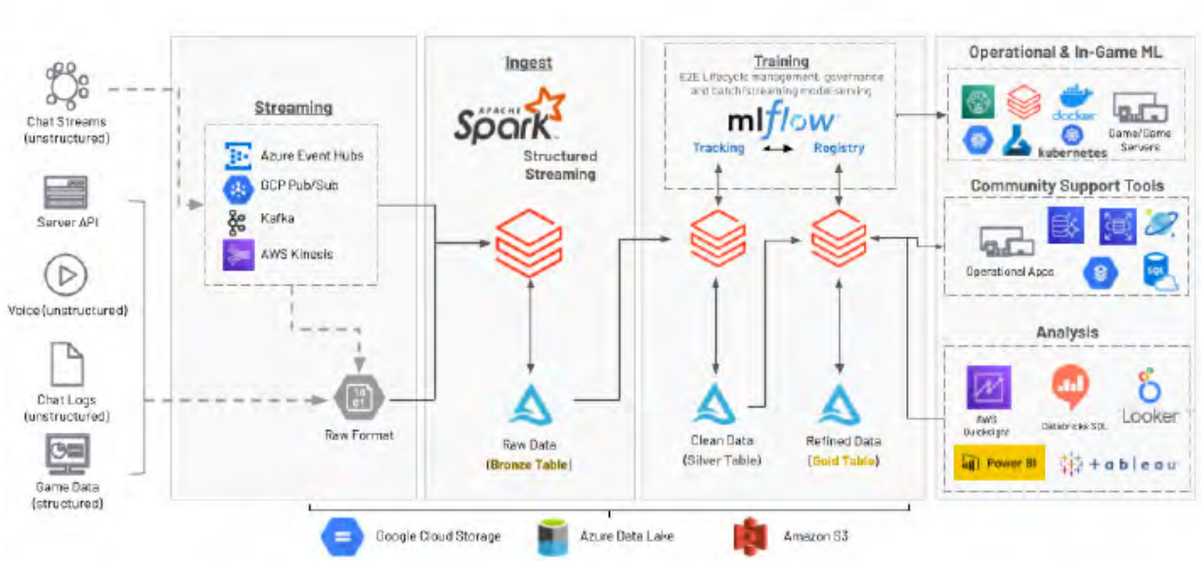


Abbildung 2
Referenzarchitektur zur Toxizitätserkennung

Eine skalierbare Echtzeitarchitektur zur Toxizitätserkennung in der Community bietet die Möglichkeit, die Workflows für Community Relationship Manager (CRMs) zu vereinfachen und Millionen von Interaktionen in überschaubare Workloads zu filtern. Ebenso kann sich die Möglichkeit, bei schwerwiegenden Toxizitätsereignissen in Echtzeit Alarm auszulösen oder sogar eine direkte Reaktion – wie etwa das Stummschalten von Spielern oder die schnelle Benachrichtigung eines CRM über den Vorfall – zu automatisieren, direkt auf die Spielerbindung auswirken. Ferner kann eine Plattform, die in der Lage ist, große Datensätze aus unterschiedlichen Quellen zu verarbeiten, die Überwachung der Markenwahrnehmung über Berichte und Dashboards ermöglichen.

Erste Schritte

Ziel dieses Solution Accelerators ist es, durch Echtzeiterkennung von toxischen Kommentaren im Game-Chat das laufende Management toxischer Interaktionen in Online-Spielen zu unterstützen. Legen Sie jetzt los und importieren Sie diesen Solution Accelerator direkt in Ihren Databricks-Arbeitsbereich.

Nach dem Import verfügen Sie über Notebooks mit zwei Pipelines, die direkt in die Produktion überführt werden können.

- Eine ML-Pipeline mit Multilabel-Klassifizierung und Training an echten englischsprachigen Datensätzen von Google Jigsaw. Dieses Modell klassifiziert und markiert Formen der Toxizität im Text.
- Inferenzpipeline für das Echtzeit-Streaming, die das Toxizitätsmodell nutzt. Die Pipelinequelle kann leicht modifiziert werden, um Chat-Daten aus allen gängigen Datenquellen zu erfassen.

Mit diesen beiden Pipelines können Sie Toxizität mit minimalem Aufwand nachvollziehen und analysieren. Dieser Solution Accelerator bietet außerdem eine Grundlage für Erstellung, Anpassung und Optimierung des Modells mit relevanten Daten zu Spielmechanismen und Communitys.

Experimentieren Sie mit den folgenden kostenlosen Databricks-Notebooks.

KAPITEL 2.7

Fördern des Wandels bei Northwestern Mutual (Insights Platform) durch den Umstieg auf eine skalierbare und offene Lakehouse-Architektur

von **MADHU KOTIAN**

15. Juli 2021

Die digitale Transformation steht bei den meisten aktuellen Big-Data-Initiativen von Unternehmen im Vordergrund. Das gilt vor allem bei Unternehmen mit einem hohen Anteil an Altlasten. Einer der wesentlichen Faktoren der digitalen Transformation sind Daten und die zugehörigen Datenspeicher. Seit über 160 Jahren unterstützt Northwestern Mutual Familien und Unternehmen bei der finanziellen Vorsorge. Angesichts eines Umsatzes von über 31 Mrd. US-Dollar, mehr als 4,6 Mio. Kunden und über 9.300 Finanzexperten gibt es wohl kaum ein anderes Unternehmen, das über ein derartiges Datenvolumen aus einer Vielzahl von Quellen verfügt.

In der heutigen Zeit stellt die Datenerfassung eine Herausforderung dar, denn Unternehmen haben es mit Millionen von Datenpunkten in unterschiedlichsten Formaten zu tun, die sich über verschiedene Zeiträume erstrecken und aus verschiedenen Richtungen kommen – und dies in beispielloser Anzahl. Wir wollen Daten für die Analyse vorbereiten, damit sie sinnvoll genutzt werden können. Ich freue mich, Ihnen heute unseren neuen Ansatz zur Transformation und Modernisierung unserer Dateneingabe- und Planungsprozesse sowie unserer Pläne für Datenspeicher vorstellen zu dürfen. Wir haben gelernt, dass ein effektiver Ansatz vielschichtig ist, deshalb werde ich neben den technischen Vorkehrungen auch unseren Plan für das Onboarding unseres Teams erläutern

Die Herausforderungen

Bevor wir uns an die Transformation machten, versuchten wir, gemeinsam mit unseren Geschäftspartnern unsere technischen Beschränkungen richtig nachzuvollziehen und ihre Hilfe bei der Formulierung der Problemstellung für unseren Business Case in Anspruch zu nehmen.

Der von uns identifizierte Schwachpunkt im Unternehmen war das Fehlen integrierter Daten, wobei Kunden- und Geschäftsdaten von verschiedenen internen und externen Teams und Datenquellen stammten. Wir realisierten zwar den Mehrwert von Echtzeitdaten, hatten aber zu solchen Produktions- und Echtzeitdaten, die es uns ermöglichen würden, zeitnah geschäftliche Entscheidungen zu treffen, nur in begrenztem Maße Zugang. Wir erfuhren auch, dass die von der Verwaltung erstellten Datenspeicher zur Entstehung von Datensilos geführt hatten, die ihrerseits Latenzprobleme, Mehrkosten für die Datenverwaltung und unnötige Beschränkungen der Sicherheit verursachten.

Darüber hinaus existierten Herausforderungen im Hinblick auf unseren aktuellen technischen Stand. Angesichts der gestiegenen Nachfrage und des zusätzlichen Datenbedarfs hatten wir Schwierigkeiten bei der Skalierbarkeit der Infrastruktur, der Datenlatenz, den Kosten für die Verwaltung der Datensilos, Beschränkungen bei Datengrößen und -volumina sowie Fragen der Datensicherheit. Angesichts dieser Herausforderungen erkannten wir, dass eine Menge Arbeit vor uns lag und wir Partner finden mussten, die uns bei der Transformation in geeigneter Weise unterstützen konnten.

Lösungsanalyse

Wir mussten also datengesteuert werden, um wieder wettbewerbsfähig werden, unsere Kunden besser betreuen und interne Prozesse optimieren zu können. Wir nahmen also verschiedene Optionen unter die Lupe und führten mehrere Proofs-of-Concept durch, um uns schließlich für eine endgültige Empfehlung zu entscheiden. Die folgenden Aspekte waren dabei für unsere künftige Strategie unverzichtbar:

- Komplettlösung für unsere Anforderungen in den Bereichen Datenerfassung, Datenmanagement und Datenanalyse
- Moderne Datenplattform, die unsere Entwickler und Business Analysts bei der Durchführung ihrer Analysen mit SQL effektiv unterstützt
- Daten-Engine, die auf S3 aufsetzende ACID-Transaktionen und rollenbasierte Sicherheit unterstützt
- System mit wirksamem Schutz für unsere personenbezogenen Daten und geschützten Gesundheitsinformationen
- Plattform mit automatischer Skalierung basierend auf Datenverarbeitung und Analysebedarf

Unsere vorhandene Infrastruktur basierte auf dem MSBI-Stack. Wir haben SSIS für die Datenerfassung, SQL Server als Datenspeicher, Azure Analysis Service für das tabellarische Modell und Power BI für Dashboard-Erstellung und Reporting verwendet. Auch wenn die Plattform die Anforderungen des Unternehmens zunächst erfüllte, bekamen wir aufgrund von steigendem Datenvolumen und Datenverarbeitungsbedarf Probleme mit der Skalierung und schränkten unsere Erwartungen an die Datenanalyse ein. Mit dem zusätzlichen Datenbedarf verursachten unsere durch verzögertes Laden entstehenden Datenlatenzprobleme und ein Datenspeicher für spezielle Geschäftsanforderungen die Entstehung von Datensilos und förderten zudem die Unübersichtlichkeit.

Außerdem waren wir durch die Verteilung der Daten auf mehrere Datenspeicher mit Sicherheitsproblemen konfrontiert. Wir hatten etwa 300 ETL-Jobs, die mehr als 7 Stunden unserer täglichen Arbeitszeit in Anspruch nahmen. Die Time-to-Market jeder Änderung oder Neuentwicklung betrug je nach Komplexität etwa 4 bis 6 Wochen.

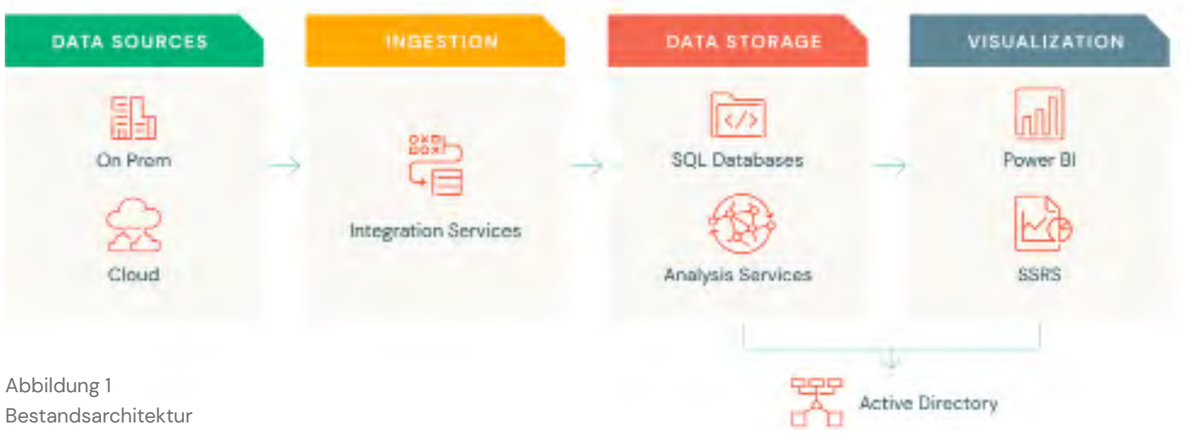


Abbildung 1
Bestandsarchitektur

Nachdem wir mehrere Lösungen auf dem Markt evaluiert hatten, entschieden wir uns für Databricks, um uns auf die Implementierung einer einzigen integrierten Datenverwaltungslösung beschränken zu können, die auf einer offenen Lakehouse-Architektur basiert.

databricks

Da Databricks auf der Grundlage von Apache Spark™ entwickelt wurde, konnten wir für die Erstellung unseres benutzerdefinierten Frameworks für die Dateneingabe und das Metadatenmanagement Python verwenden.
Dies bot uns die Flexibilität, Ad-hoc-Analysen und weitere Datenermittlungen mithilfe des Notebooks durchzuführen. Databricks Delta Lake (die Speicherebene, die auf unserem Data Lake aufbaut) bot uns die Flexibilität, verschiedene Funktionen für das Datenbankmanagement (ACID-Transaktionen, Metadaten-Governance, Zeitreisen usw.) umzusetzen, und ermöglichte auch die Implementierung der erforderlichen Sicherheitskontrollmechanismen. Databricks vereinfachte die Verwaltung und Skalierung des Clusters und reagierte effektiv auf den Nachholbedarf unserer Techniker und Geschäftsanwender.

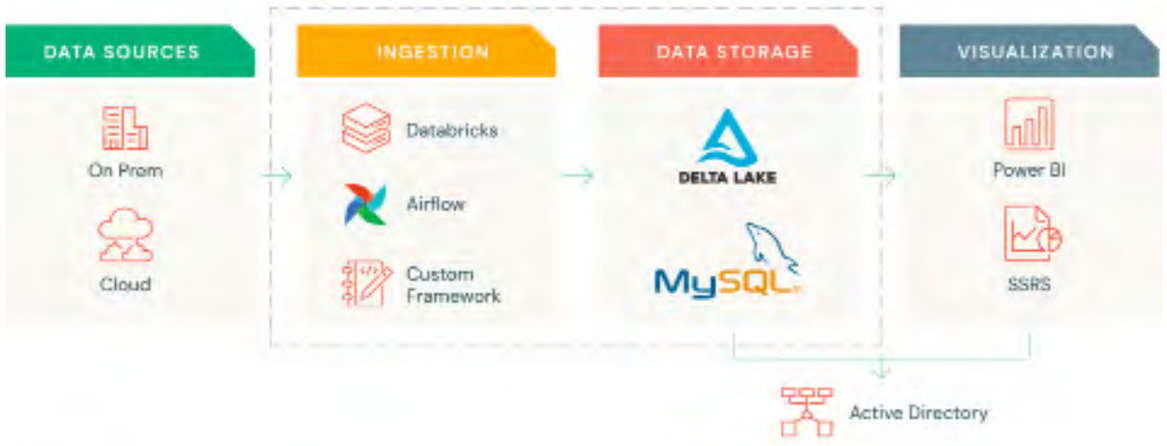


Abbildung 2
Architektur mit Databricks

Migrationsansatz und Onboarding-Ressourcen

Am Anfang stand eine kleine Gruppe von Engineers, die wir einem virtuellen Team aus unserem bestehenden Scrum-Team zuwiesen. Ihr Ziel bestand darin, verschiedene Proofs-of-Concept umzusetzen, die Entwicklung ausgehend von der empfohlenen Lösung vorzunehmen, Best Practices zu entwickeln und zum Schluss in ihre jeweiligen Teams zurückzukehren, um dort das Onboarding zu unterstützen. Der Rückgriff auf vorhandene Teammitglieder war für uns vorteilhafter, da sie das Bestandssystem kannten, den aktuellen Dateneingabefluss und die entsprechenden Geschäftsregeln verstanden und Kompetenzen in mindestens einem Bereich der Programmierung (Data Engineering und Software Engineering) besaßen. Dieses Team bildete sich zunächst in Python fort, konnte die komplizierten Details von Spark und Delta nachvollziehen und arbeitete eng mit dem Databricks-Team zusammen, um die Lösung bzw. den Lösungsansatz zu validieren. Während das Team an der Gestaltung des zukünftigen Zustands arbeitete, kümmerte sich der Rest unserer Entwickler um die Umsetzung geschäftlicher Prioritäten.

Da die meisten Entwickler MSBI-Stack-Engineers waren, lautete der Plan, eine Datenplattform aufzusetzen, die für unsere Entwickler, Geschäftsanwender und Außendienstberater reibungslos funktionieren würde.

- Wir haben zunächst ein Framework für die Datenerfassung entwickelt, das alle unsere Anforderungen an das Laden und Transformieren von Daten abdeckt. Es verfügte über eingebaute Sicherheitsfunktionen, die alle Metadaten und Geheimnisse unserer Quellsysteme zuverlässig schützten. Beim Erfassungsvorgang wurde eine JSON-Datei entgegengenommen, die die Quelle, das Ziel und die erforderliche Transformation angab. Dies ermöglichte sowohl einfache als auch komplexe Transformationen.
- Für die Planung verwendeten wir Airflow, aber angesichts der Komplexität der DAG bauten wir unser eigenes maßgeschneidertes Framework auf Grundlage von Airflow. Dieses nahm eine YAML-Datei mit Jobinformationen und zugehörigen Abhängigkeiten entgegen.
- Für die Verwaltung von Änderungen auf Schemaebene mit Delta haben wir ein weiteres eigenes Framework entwickelt, das verschiedene Operationen am Datenbanktyp (DDL) automatisierte, ohne dass die Entwickler einen „gläsernen Datenspeicher“ benötigten. Dies erleichterte zudem die Implementierung verschiedener Kontrollfunktionen für Datenspeicher-Audits.

Parallel dazu arbeitete das Team mit unserem Sicherheitsteam zusammen, um dafür Sorge zu tragen, dass wir alle Datensicherheitskriterien (Verschlüsselung von Datenübertragungen und ruhenden Daten sowie Verschlüsselung auf Spaltenebene zum Schutz personenbezogener Informationen) verstanden und umgesetzt haben.

Nach der Einrichtung dieser Frameworks implementierte das Kohortenteam einen End-to-End-Datenfluss (also von der Quelle zum Ziel mit allen Transformationen) und erstellte in Power BI eine Anzahl neuer Berichte und Dashboards, die auf Delta Lake verweisen. Das Ziel bestand darin, die Leistung unseres End-to-End-Prozesses zu testen, die Daten zu validieren und Feedback von unseren Anwendern zu erhalten. Basierend auf dem Feedback und den Ergebnissen unserer Leistungs- und Validierungstests konnten wir das Produkt schrittweise verbessern.

Gleichzeitig erstellten wir Schulungsleitfäden und Praxisanleitungen für die Fortbildung unserer Entwickler. Bald darauf beschlossen wir, die Mitglieder des Kohorten-Teams in ihre jeweiligen Teams zu versetzen, behielten aber einige von ihnen für die künftige Unterstützung der Plattforminfrastruktur (DevOps). Jedes Scrum-Team war für die Verwaltung und Bereitstellung der jeweiligen Kompetenzen und Funktionen für das Unternehmen verantwortlich. Nachdem die Mitglieder in ihre jeweiligen Teams zurückgekehrt waren, machten sie sich daran, die Arbeitsgeschwindigkeit im Team so zu erhöhen, dass der Migrationsmehraufwand bewältigt werden konnte. Den Teamleitern wurden konkrete Leitlinien und realistische Ziele vorgegeben, um die Migrationsziele für die verschiedenen Sprint- und Programmstufen zu erreichen. Die Teammitglieder in der Kohortengruppe waren nun die zuständigen Experten und halfen ihrem Team bei der Einführung der neuen Plattform. Sie standen für alle spontanen Fragen wie auch zur Unterstützung zur Verfügung.

Während wir unsere neue Plattform schrittweise aufbauten, behielten wir die alte Plattform zur Validierung und Überprüfung bei.

Der Beginn des Erfolgs

Für die gesamte Umstellung haben wir etwa anderthalb Jahre gebraucht, was eine ziemliche Leistung darstellt, wenn man bedenkt, dass wir alle Frameworks entwickeln, geschäftliche Prioritäten verwalten, die Sicherheitserwartungen managen, unser Team umrüsten und die Plattform migrieren mussten. Die Gesamtdauer für das Laden sank deutlich von 7 auf nur noch 2 Stunden. Unsere Time-to-Market betrug nur noch etwa 1 bis 2 Wochen – ein deutlicher Rückgang gegenüber den vorherigen 4 bis 6 Wochen. Das stellte eine erhebliche Verbesserung dar, die sich meiner Ansicht nach in mehrfacher Hinsicht auf unser gesamtes Unternehmen auswirken wird.

Unsere Reise ist noch nicht zu Ende. Während wir unsere Plattform weiter verbessern, steht bereits unsere nächste Aufgabe an: die Erweiterung des Lakehouse-Musters. Wir arbeiten gegenwärtig daran, unsere Plattform auf E2 zu migrieren und Databricks SQL zu implementieren. Wir arbeiten an unserer Strategie, unseren geschäftlichen Anwendern eine Self-Service-Plattform zur Verfügung zu stellen, damit sie ihre Ad-hoc-Analysen durchführen können. Außerdem wollen wir ihnen die Möglichkeit bieten, eigene Daten einzubringen und Analysen mit unseren integrierten Daten durchzuführen. Wir haben festgestellt, dass die Nutzung einer offenen, einheitlichen und skalierbaren Plattform für uns von großem Nutzen ist. Unsere Anforderungen und Fähigkeiten werden weiterwachsen, und daher ist es gut zu wissen, dass wir mit Databricks einen starken Partner haben.

Erfahren Sie mehr über den Weg von Northwestern Mutual zum Lakehouse.

ÜBER MADHU KOTIAN

Madhu Kotian ist Vice President of Engineering (Investment Products Data, CRM, Apps and Reporting) bei Northwestern Mutual. Er ist seit über 25 Jahren auf dem Gebiet der IT tätig. Seine Erfahrung und sein Fachwissen erstrecken sich auf die Bereiche Data Engineering, Personalmanagement, Programmmanagement, Architektur und Design sowie Entwicklung und Wartung unter Verwendung agiler Praktiken. Ferner ist er Fachmann für Verfahren und die Implementierung von Datenintegration und -analyse.

KAPITEL 2.8

So erstellte das Databricks-Datenteam ein Lakehouse über drei Clouds und mehr als 50 Regionen

von **JASON POHL** und **SURAJ ACHARYA**

14. Juli 2021

Die interne Protokollierungsinfrastruktur bei Databricks hat sich im Laufe der Jahre weiterentwickelt, und wir haben dabei einige Lektionen zu der Frage gelernt, wie man eine hochverfügbare Protokollierungspipeline Cloud- und standortübergreifend pflegen kann. In diesem Blogpost erfahren Sie, wie wir mit unserer Lakehouse-Plattform Echtzeitmetriken erfassen und verwalten und mehrere Clouds nutzen, um Ausfälle der Public Cloud abzufedern.

Als Databricks gegründet wurde, unterstützte es zunächst nur eine einzige Public Cloud. Mittlerweile jedoch bietet der Service in über 50 Regionen weltweit Unterstützung für die drei großen Public Clouds (AWS, Azure, GCP). Jeden Tag bringt Databricks im Auftrag seiner Kunden Millionen von virtuellen Maschinen zum Laufen. Unser Datenplattformteam mit seinen noch nicht einmal zehn Engineers ist für Aufbau und Wartung der Telemetrieinfrastruktur für die Protokollierung zuständig, die täglich ein halbes Petabyte an Daten verarbeitet. Die Orchestrierung, Überwachung und Nutzung wird über Servicelogs erfasst, die von unserer Infrastruktur verarbeitet werden, um aktuelle und korrekte Metriken zu generieren. Am Ende werden diese Daten dann in unserem eigenen Petabyte-großen Delta Lake gespeichert. Unser Datenplattformteam nutzt Databricks für die Cloud-übergreifende Datenverarbeitung. So können wir bei Bedarf Datenverbünde erstellen, die Wiederherstellung nach einem regional begrenzten Cloud-Ausfall abmildern und Störungen unserer aktiven Infrastruktur minimieren.

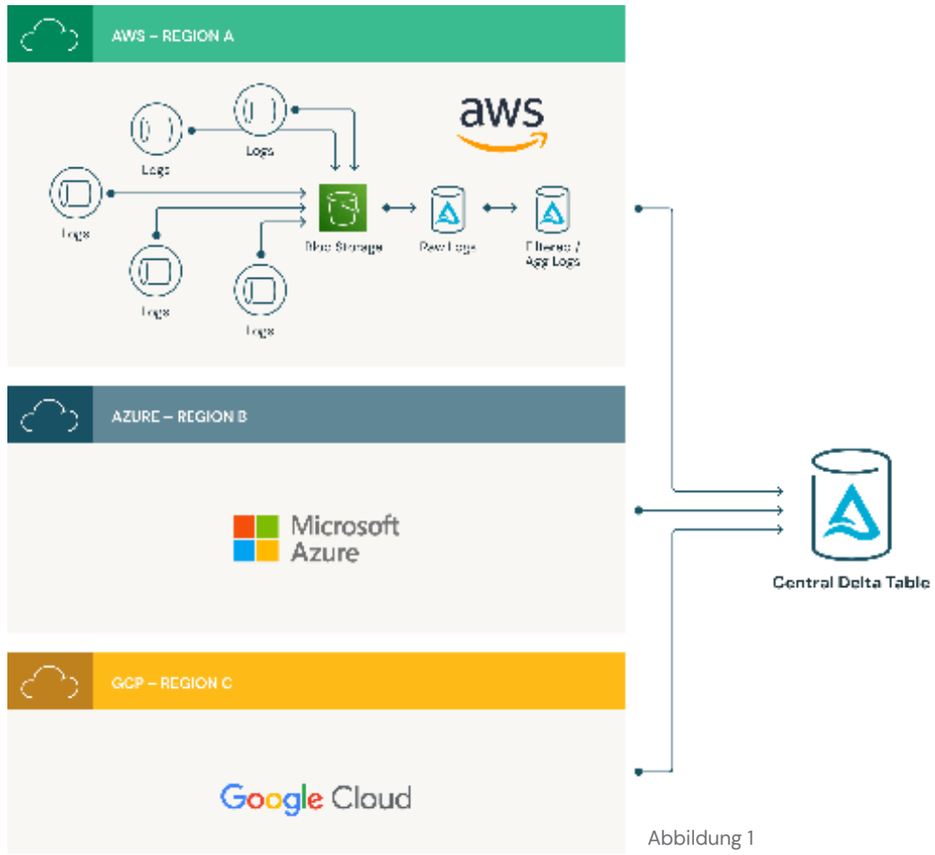


Abbildung 1

Pipelinearchitektur

Jede Cloud-Region verfügt über eine eigene Infrastruktur sowie Datenpipelines zum Erfassen, Zusammenführen und Speichern von Protokolldaten in einem regionalen Delta Lake. Produkttelemetriedaten werden produktübergreifend und in unseren Pipelines mit demselben Prozess erfasst, der in jeder Cloud-Region repliziert wird. Ein Log-Daemon erfasst die Telemetriedaten und schreibt die Protokolle in einen regionalen Cloud-Speicher-Bucket (S3, WASBS, GCS). Von dort aus erfasst eine zeitgesteuerte Pipeline die Protokolldateien mithilfe von Auto Loader (AWS, Azure, GCP) und schreibt die Daten in eine regionale Delta-Tabelle. Eine andere Pipeline liest Daten aus der regionalen Delta-Tabelle, filtert sie und schreibt sie in eine zentrale Delta-Tabelle in einer einzigen Cloud-Region.

databricks

Vor der Einführung von Delta Lake

Vor Delta Lake schrieben wir die Quelldaten in eine eigene Tabelle im zentralisierten Lake und erstellten dann eine Ansicht, die eine Union aller beteiligten Tabellen war. Die folgende Sicht musste zur Laufzeit berechnet werden und wurde immer ineffizienter, je mehr Regionen wir hinzufügten:

```
CREATE OR REPLACE VIEW all_logs AS
SELECT * FROM (
  SELECT * FROM region_1.log_table
  UNION ALL
  SELECT * FROM region_2.log_table
  UNION ALL
  SELECT * FROM region_3.log_table
  ...
);
```

Nach der Einführung von Delta Lake

Heute haben wir nur eine einzige Delta-Tabelle, die nebenläufige Schreibanweisungen aus über 50 verschiedenen Regionen entgegennimmt. Gleichzeitig bewältigt sie auch noch Datenabfragen. Das Abfragen der zentralen Tabelle ist extrem einfach:

```
SELECT * FROM central.all_logs;
```

Die Transaktionalität wird von Delta Lake verwaltet. Wir haben die einzelnen Regionaltabellen aus unserem zentralen Delta Lake geworfen und die UNION ALL-Sicht gleich dazu. Der folgende Code ist eine vereinfachte Darstellung der Syntax, die ausgeführt wird, um die Daten, deren Versand aus den regionalen Delta Lakes genehmigt wurde, in den zentralen Delta Lake zu laden:

```
spark.readStream.format(“delta”)
  .load(regional_source_path)
  .where(“egress_approved = true”)
  .writeStream
  .format(“delta”)
  .outputMode(“append”)
  .option(“checkpointLocation”, checkpoint_path)
  .start(central_target_path)
```

Disaster Recovery

Einer der Vorteile des Betriebs eines Cloud-übergreifenden Service besteht darin, dass wir für bestimmte Disaster-Recovery-Szenarien gut aufgestellt sind. Auch wenn es selten vorkommt, ist es eigentlich nicht ungewöhnlich, dass der Compute-Service einer bestimmten Cloud-Region ausfällt. In diesem Fall ist der Zugriff auf den Cloud-Speicher zwar möglich, aber die Möglichkeit, neue VMs zu starten, ist beeinträchtigt. Da wir unseren Datenpipelinecode so entwickelt haben, dass er Konfigurationen für die Quell- und Zielpfade akzeptiert, können wir Datenpipelines schnell in einer anderen Region als der, in der die Daten gespeichert sind, bereitstellen und ausführen. Es ist dabei unerheblich, in welcher Cloud der Cluster erstellt wird und aus welcher Cloud die Daten gelesen bzw. in welche sie geschrieben werden.

Es gibt einige wenige Datensätze, die wir durch fortlaufende Replikation der Daten zwischen den Cloud-Anbietern gegen einen Ausfall des Speicherservice absichern. Dies kann einfach durch die Nutzung der Delta-Deep Clone-Funktionalität erledigt werden. (Weitere Informationen hierzu finden Sie in diesem Blogpost.) Jedes Mal, wenn der Clone-Befehl für eine Tabelle ausgeführt wird, aktualisiert er den Klon nur inkrementell, d. h. mit den seit der letzten Befehlsausführung aufgetretenen Änderungen. Dies ist eine effiziente Möglichkeit, Daten regions- und sogar Cloud-übergreifend zu replizieren.

Störungsminimierung bei aktiven Datenpipelines

Unsere Datenpipelines sind das Lebenselixier unseres Managed Service und Teil eines globalen Unternehmens, das nie zur Ruhe kommt. Wir können es uns nicht leisten, die Pipelines über einen längeren Zeitraum hinweg für Wartungsarbeiten, Upgrades oder Daten-Backfills anzuhalten. Kürzlich mussten wir einen Pipeline-Fork implementieren, um einen Teil der Daten, die normalerweise in unsere Haupttabelle geschrieben werden, zu filtern, weil sie in eine andere Public Cloud geschrieben werden sollten. Dies war möglich, ohne den laufenden Betrieb zu stören.

Durch Befolgen der nachstehenden Schritte konnten wir die Änderungen an unserer Architektur ohne Unterbrechung in unser Live-System umsetzen.

Zuerst haben wir einen Deep Clone der Haupttabelle an einen neuen Ort in der anderen Cloud durchgeführt. Hierbei werden sowohl die Daten als auch das Transaktionsprotokoll so kopiert, dass die Konsistenz gewährleistet bleibt.

Dann haben wir die neue Konfiguration für unsere Pipelines freigegeben, sodass der Großteil der Daten weiterhin in die zentrale Haupttabelle geschrieben wird, während der gewünschte Teilbereich der Daten in die neue geklonte Tabelle in der anderen Cloud geschrieben wird. Diese Änderung kann ganz einfach durch die Bereitstellung einer neuen Konfiguration vorgenommen werden. Die Tabellen erhalten Updates dann nur für die vorgesehenen neuen Änderungen.

Als Nächstes führten wir denselben Deep-Clone-Befehl erneut aus. Delta Lake erfasst nun nur die inkrementellen Änderungen aus der ursprünglichen Haupttabelle und kopiert sie in die neue geklonte Tabelle. Hierdurch erfolgt im Wesentlichen ein Abgleich der neuen mit allen Datenänderungen, die zwischen Schritt 1 und Schritt 2 aufgetreten sind.

Abschließend können die Teilmenge der Daten aus der Haupttabelle und der Großteil der Daten aus der geklonten Tabelle gelöscht werden.

Nun enthalten beide Tabellen die gewünschten Daten mit vollständigem Transaktionsverlauf, und zwar live, ohne die Aktualität der Pipelines zu beeinträchtigen.

Fazit

Databricks abstrahiert die Details der einzelnen Cloud-Services, sei es für das Starten neuer Infrastrukturelemente mit unserem Cluster-Manager, das Erfassen von Daten mit Auto Loader oder die Durchführung transaktionaler Schreibvorgänge in den Cloud-Speicher mit Delta Lake. Dies bietet uns den Vorteil, eine einheitliche Codebasis verwenden zu können, um Rechen- und Speicherkapazitäten übergreifend für Public zu überbrücken, und zwar sowohl für den Datenverbund als auch bei der Notfallwiederherstellung. Diese Cloud-übergreifende Funktionalität bietet uns genügend Flexibilität, um Rechen- und Speicherkapazitäten dorthin verlagern zu können, wo sie uns und unseren Kunden am besten dienen.

KAPITEL

03

Kundenberichte

Atlassian

ABN AMRO

J.B. Hunt

KAPITEL 3.1

Atlassian

Atlassian gehört zu den führenden Anbietern von Software für Zusammenarbeit, Entwicklung und Fehlerverfolgung für Teams. Mit über 150.000 Kunden weltweit (darunter 85 der Fortune 100) befeuert Atlassian mit Produkten wie Jira, Confluence, Bitbucket, Trello und weiteren eine schlagkräftige Zusammenarbeit.

ANWENDUNGSFALL

Atlassian nutzt die Lakehouse-Plattform von Databricks, um Daten im gesamten Unternehmen zu demokratisieren und die Betriebskosten zu senken. Zurzeit sind bei Atlassian eine ganze Reihe von Anwendungsfällen implementiert, bei denen vor allem das Kundenerlebnis im Vordergrund steht.

Kundensupport und Serviceerfahrung
Da die Mehrheit der Kunden serverbasiert arbeitet (und dabei Produkte wie Jira und Confluence verwendet), entschloss sich Atlassian zur Auslagerung dieser Kunden in die Cloud, um umfassende Erkenntnisse zu gewinnen, die den Kundensupport bereichern sollten.

Personalisiertes Marketing
Dieselben Erkenntnisse konnten auch für personalisierte Marketing-E-Mails genutzt werden, um die Kunden mit neuen Funktionen und Produkten vertraut zu machen.

Erkennung von Missbrauch und Betrug
Atlassian ist nun dank Anomalieerkennung und prädiktiven Analysen in der Lage, den Missbrauch von Lizenzen sowie betrügerisches Verhalten vorherzusagen.

Bei Atlassian müssen wir sicherstellen, dass Teams funktionsübergreifend gut zusammenarbeiten können, um die im stetigen Wandel begriffenen Ziele umzusetzen. Eine vereinfachte Lakehouse-Architektur würde uns in die Lage versetzen, große Mengen an Benutzerdaten zu erfassen und die erforderlichen Analysen durchzuführen, um Kundenbedürfnisse besser prognostizieren und die Nutzungsqualität für unsere Kunden verbessern zu können. Mit der zentralen und benutzerfreundlichen Cloud-Analyseplattform können wir Kollaborations-Tools auf Grundlage handfester Erkenntnisse kurzfristig optimieren bzw. neu entwickeln.

Rohan Dhupelia
Data Platform Senior Manager, Atlassian

LÖSUNG UND NUTZEN

Mithilfe der Lakehouse-Plattform von Databricks kann Atlassian seine Daten intern wie extern umfassend demokratisieren. Von einem Data-Warehousing-Paradigma kommend, hat das Unternehmen auf eine Standardisierung auf Databricks umgestellt. Hierdurch ist es nun in der Lage, die Datenorientierung im gesamten Unternehmen zu verbessern. Über 3000 interne Nutzer aus den Bereichen Personalwesen, Marketing, Finanzen und Forschung und Entwicklung – das ist in der Summe mehr als der halbe Personalbestand des Unternehmens – greifen monatlich über offene Technologien wie Databricks SQL auf die mit der Plattform gewonnenen Erkenntnisse zu. Atlassian nutzt die Plattform außerdem, um seinen Kunden ein personalisiertes Support- und Serviceerlebnis zu bieten.

- Delta Lake bildet die Grundlage für ein zentrales Lakehouse mit vielen Petabyte Daten, auf die mehr als 3000 Nutzer aus den Bereichen HR, Marketing, Finanzen, Vertrieb, Support und F&E zugreifen.
- BI-Workloads auf der Basis von Databricks SQL ermöglichen Dashboard-Reporting für mehr Benutzer.
- MLflow rationalisiert MLOps für schnellere Auslieferung
- Die Vereinheitlichung der Datenplattform erleichtert die Governance, und selbstverwaltete Cluster ermöglichen autonomes Handeln.

Mit der Cloud-Architektur, der verbesserten Produktivität durch teamübergreifende Zusammenarbeit und der Möglichkeit, für Analyse und ML auf sämtliche Kundendaten zuzugreifen, werden die Auswirkungen bei Atlassian mittelfristig voraussichtlich erheblich sein. Bereits jetzt konnte das Unternehmen

- die Kosten für den IT-Betrieb (und insbesondere für Rechenkapazitäten) durch Verlagerung von mehr als 50.000 Spark-Jobs von EMR zu Databricks mit minimalem Aufwand und geringen Code-Änderungen um 60 % senken,
- durch kürzere Entwicklungszyklen die Fertigstellungdauer um 30 % reduzieren und
- die Abhängigkeiten von Datenteams dank verstärktem Self-Service im gesamten Unternehmen um 70 % verringern.

Weitere Informationen

databricks

KAPITEL 3.2

ABN AMRO

Das traditionsreiche Bankhaus ABN AMRO wollte sein Geschäft modernisieren, sah sich aber durch eine veraltete Infrastruktur und Data Warehouses behindert, die den Zugriff auf Daten aus verschiedenen Quellen erschwerten und für ineffiziente Datenprozesse und Workflows sorgten. Heute ermöglicht Azure Databricks ABN AMRO die Demokratisierung von Daten und KI für ein Team von mehr als 500 Engineers, Wissenschaftlern und Analysten, die gemeinsam an der Optimierung von Geschäftsabläufen und der Einführung neuer marktreifer Funktionen im gesamten Unternehmen arbeiten.

ANWENDUNGSFALL

ABN AMRO nutzt die Lakehouse-Plattform von Databricks, um durch Automatisierung und Einblicke in alle Abläufe Finanzdienstleistungen auf globaler Ebene zu transformieren.

Personalisiertes Finanzwesen
ABN AMRO nutzt Echtzeitdaten und Kundenerkenntnisse, um Produkte und Dienstleistungen anzubieten, die auf Kundenbedürfnisse zugeschnitten sind. So wird beispielsweise maschinelles Lernen für zielgerichtete Kommunikation in automatisierten Marketingkampagnen eingesetzt, um die Kundenbindung zu stärken und die Konversion zu fördern.

Risikomanagement
Mithilfe datengestützter Entscheidungsfindung legt ABN AMRO den Schwerpunkt darauf, Risiken für das Unternehmen wie auch seine Kunden zu mindern. So werden beispielsweise Berichte und Dashboards erstellt, die von internen Entscheidungsträgern und Führungskräften genutzt werden, um Risiken besser zu verstehen und Beeinträchtigungen des Geschäfts von ABN AMRO durch sie zu verhindern.

Betrugserkennung
Mit dem Ziel, schädliche Aktivitäten zu verhindern, setzt das Bankhaus Predictive Analytics ein, um Betrug zu erkennen, bevor Kunden Schaden nehmen. Zu den Aktivitäten, gegen die Maßnahmen ergriffen werden sollen, gehören etwa Geldwäsche und gefälschte Kreditkartenanträge.

databricks

Databricks hat unsere geschäftlichen Abläufe grundlegen verändert. Es hat die Möglichkeiten unserer Datenexperten mit fortschrittlichen Funktionen kontrolliert und skalierbar ergänzt und uns so in eine Position versetzt, in der wir unsere unternehmensweite Daten- und KI-Transformation erfolgreicher gestalten können.

Stefan Groot
Head of Analytics Engineering,
ABN AMRO

LÖSUNG UND NUTZEN

Heute ermöglicht Azure Databricks ABN AMRO die Demokratisierung von Daten und KI für ein Team von mehr als 500 Engineers, Wissenschaftlern und Analysten, die gemeinsam an der Optimierung von Geschäftsabläufen und der Einführung neuer marktreifer Funktionen im gesamten Unternehmen arbeiten.

- Delta Lake gestattet die Bereitstellung schneller und zuverlässiger Datenpipelines zur Einspeisung korrekter und vollständiger Daten für nachgelagerte Analysen.
- Die Integration mit Power BI erlaubt einfache SQL-Analysen und liefert mehr als 500 Geschäftsanwendern Erkenntnisse durch Berichte und Dashboards.
- MLflow beschleunigt die Implementierung neuer Modelle, die das Kundenerlebnis verbessern. Neue Anwendungsfälle werden in weniger als zwei Monaten umgesetzt.

10-mal
schnellere Time-to-Market – Implementierung von Anwendungsfällen innerhalb von nur zwei Monaten

Mehr als 100
Anwendungsfälle innerhalb der kommenden zwölf Monate zu implementieren

Mehr als 500
Geschäfts- und IT-Anwender mit ganz neuen Möglichkeiten

Weitere Informationen

databricks

KAPITEL 3.3

J.B. HUNT

Mit Databricks haben wir eine Basis für den innovativsten digitalen Marktplatz für Gütertransportleistungen erhalten, denn hiermit können wir unseren Spediteuren dank KI ein optimales Kundenerlebnis bieten.

Joe Spinelle
Director, Engineering and Technology,
J.B. Hunt

Weitere Informationen

databricks

Angesichts des selbstgesteckten Ziels, das effizienteste digitale Transportnetz Nordamerikas aufzubauen, beabsichtigte J.B. Hunt, die Güterlogistik zu rationalisieren und seinen Kunden optimale Speditionsleistungen zu bieten, jedoch gab es eine Reihe beträchtlicher Hindernisse, darunter eine veraltete Architektur, fehlende KI-Funktionen und das Unvermögen, Big Data sicher zu verarbeiten. Nach der Implementierung der Lakehouse-Plattform von Databricks und Immuta ist J.B. Hunt hingegen nun in der Lage, operative Lösungen anzubieten, die von der Verbesserung der Lieferketteneffizienz bis hin zur Steigerung der Fahrerproduktivität reichen. Diese Ergebnisse wiederum hatten erhebliche Einsparungen bei der IT-Infrastruktur und beträchtliche Umsatzsteigerungen zur Folge.

ANWENDUNGSFALL

J.B. Hunt nutzt Databricks, um über seine Carrier 360-Plattform branchenführende Analysen für Speditionsdienstleister zu liefern, Kosten zu senken und gleichzeitig Sicherheit und Produktivität der Fahrer zu steigern. Zu den Anwendungsfällen gehören Güterlogistik, Customer 360°, Personalisierung und vieles mehr.

LÖSUNG UND NUTZEN

J.B. Hunt nutzt die Lakehouse-Plattform von Databricks, um den sichersten und effizientesten Marktplatz für Gütertransporte in Nordamerika aufzubauen und so die Logistik zu optimieren, die Abläufe für die beauftragten Spediteure zu verbessern und Kosten zu senken.

- Delta Lake bündelt und demokratisiert Daten für die Echtzeitoptimierung von Routen und Fahrerempfehlungen über die Carrier 360-Plattform.
- Notebooks steigern die Produktivität des Datenteams und ermöglichen eine schnellere Implementierung von mehr Anwendungsfällen.
- MLflow beschleunigt die Implementierung neuer Modelle, die das Fahrererlebnis verbessern.

2,7 Mio. $
Einsparungen bei der IT-Infrastruktur und dadurch höhere Rentabilität

5 %
Mehreinnahmen durch verbesserte Logistik

99,8 %
schnellere Empfehlungen für ein besseres Kundenerlebnis der Spediteure

Databricks

Databricks ist das Unternehmen für Daten und KI. Mehr als 5.000 Unternehmen weltweit – darunter Comcast, Condé Nast, H&M und mehr als 40 Prozent der Fortune 500 – setzen bei der Konsolidierung ihrer Daten, Analysen und KI auf die Databricks Lakehouse-Plattform. Databricks hat seinen Hauptsitz in San Francisco im US-Bundesstaat Kalifornien und betreibt Niederlassungen auf der ganzen Welt. Databricks wurde von den ursprünglichen Entwicklern von Apache Spark™, Delta Lake und MLflow gegründet und hat es sich zur Aufgabe gemacht, Datenteams in aller Welt bei der Lösung schwierigster Probleme zu unterstützen. Wenn Sie mehr erfahren möchten, folgen Sie Databricks auf Twitter, LinkedIn und Facebook.

JETZT KOSTENLOS TESTEN

Kontaktieren Sie uns für eine personalisierte Demo:
databricks.com/contact

databricks