Data Engineering

Wintersemester 25/26

Vorstellung

- Jannik Heyl (31)
- Master Informatik
- 5 Jahre Dozent an der TH Bingen, 1 Jahr in Worms
- CTO & Business Unit Lead Data & AI

Stellt euch vor!

Name, Hintergrund & Erwartungshaltung

Fragen an euch.

- Wer hat schon mal mit Data Warehouses gearbeitet?
- Kennt jemand NoSQL-Datenbanken?
- Hat jemand Erfahrungen mit Docker oder Kubernetes?
- Wer hat schon mal mit Python programmiert?
- Wer hat von ETL (Extract, Transform, Load) gehört?
- Kennt jemand den Begriff CDC (Change Data Capture)?

Einführung in Data Engineering

Definition

„Data engineering is the development, implementation, and maintenance of systems and processes that take in raw data and produce high-quality, consistent information that supports downstream use cases, such as analysis and machine learning. Data engineering is the intersection of security, data management, DataOps, data architecture, orchestration, and software engineering. A data engineer manages the data engineering lifecycle, beginning with getting data from source systems and ending with serving data for use cases, such as analysis or machine learning.“

Fundamentals of Data Engineering

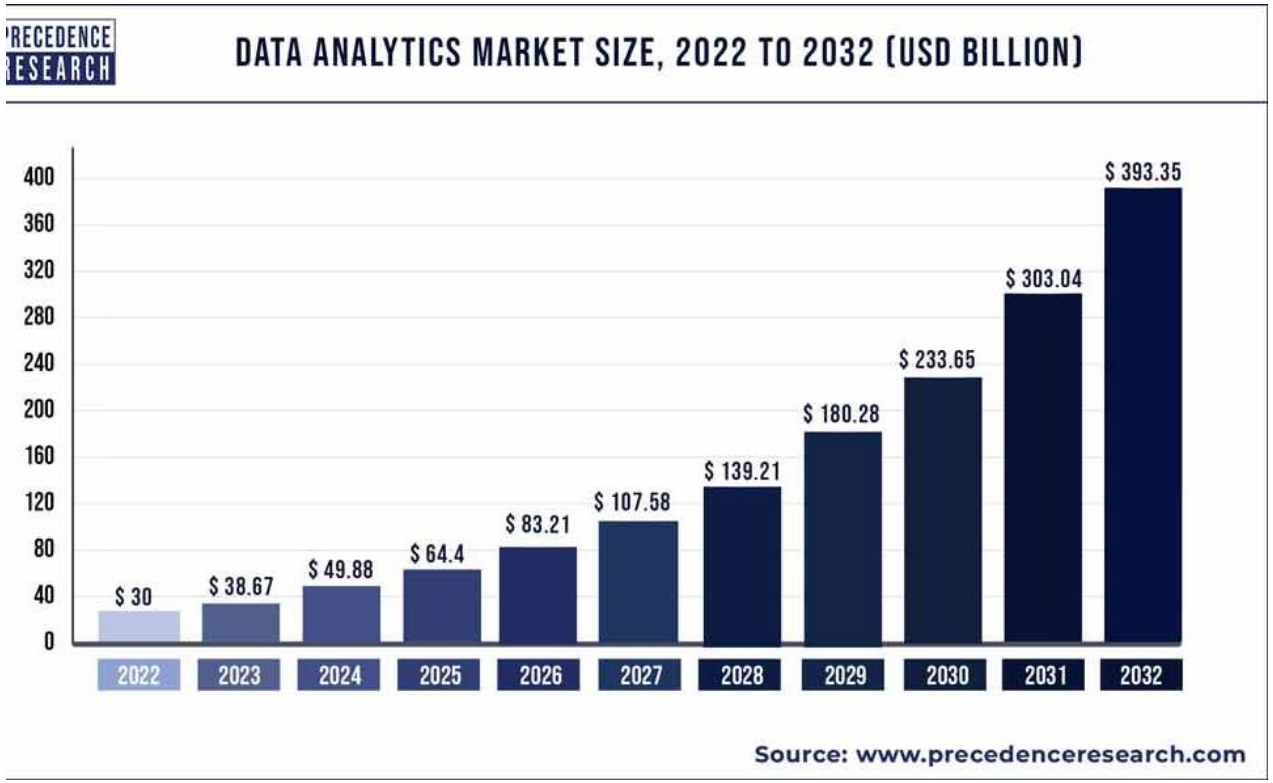
Joe Reis

Definition #2

„Data engineering is all about the movement, manipulation, and management of data.“

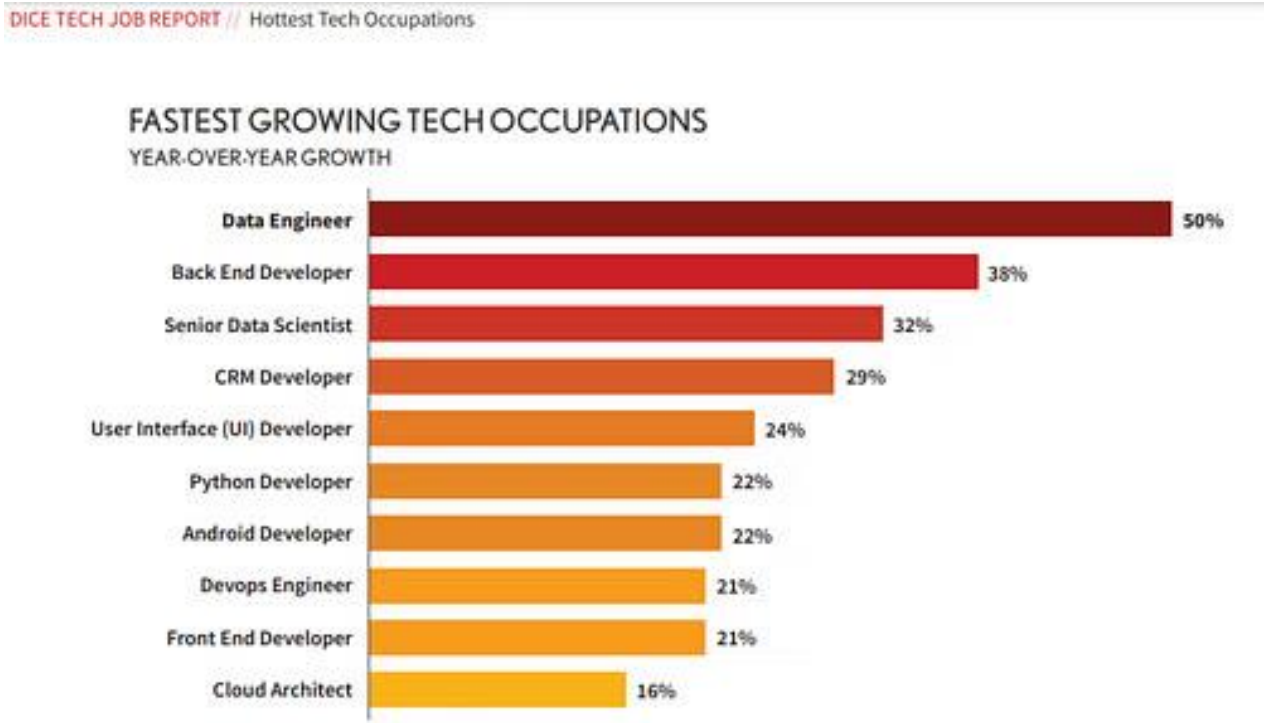
Lewis Gavin

Wachstum



- **Hottest Field in Technology**: Data Engineering ist die Grundlage für Data Science und eine der gefragtesten Fähigkeiten im Technologiebereich.

Nachfrage



- *„Dice Tech Job Report stated Data engineering as the fastest-growing career in technology in 2019, with a 50% increase in open positions.“*

Einführung

- **Zwei Typen des Data Engineering**:
  - **SQL-basiertes Data Engineering** (klassische Datenbanken, Abfragen in SQL)
  - **Big Data-orientiertes Data Engineering** (Verarbeitung großer Datenmengen mit verschiedenen Tools)
- **Der Data Engineer verwaltet den gesamten Lebenszyklus der Datenverarbeitung**
  - Vom Dateneingang bis zur Bereitstellung für analytische und operative Systeme

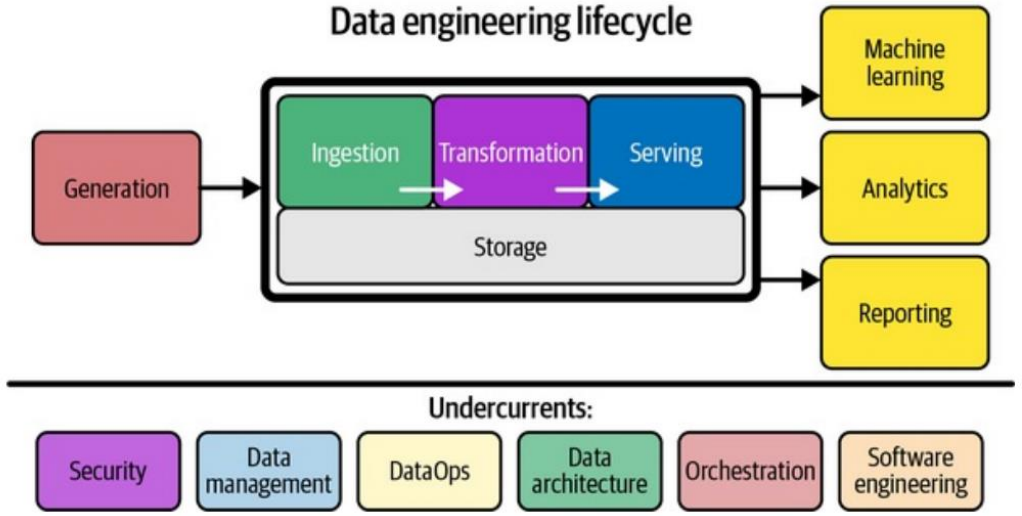
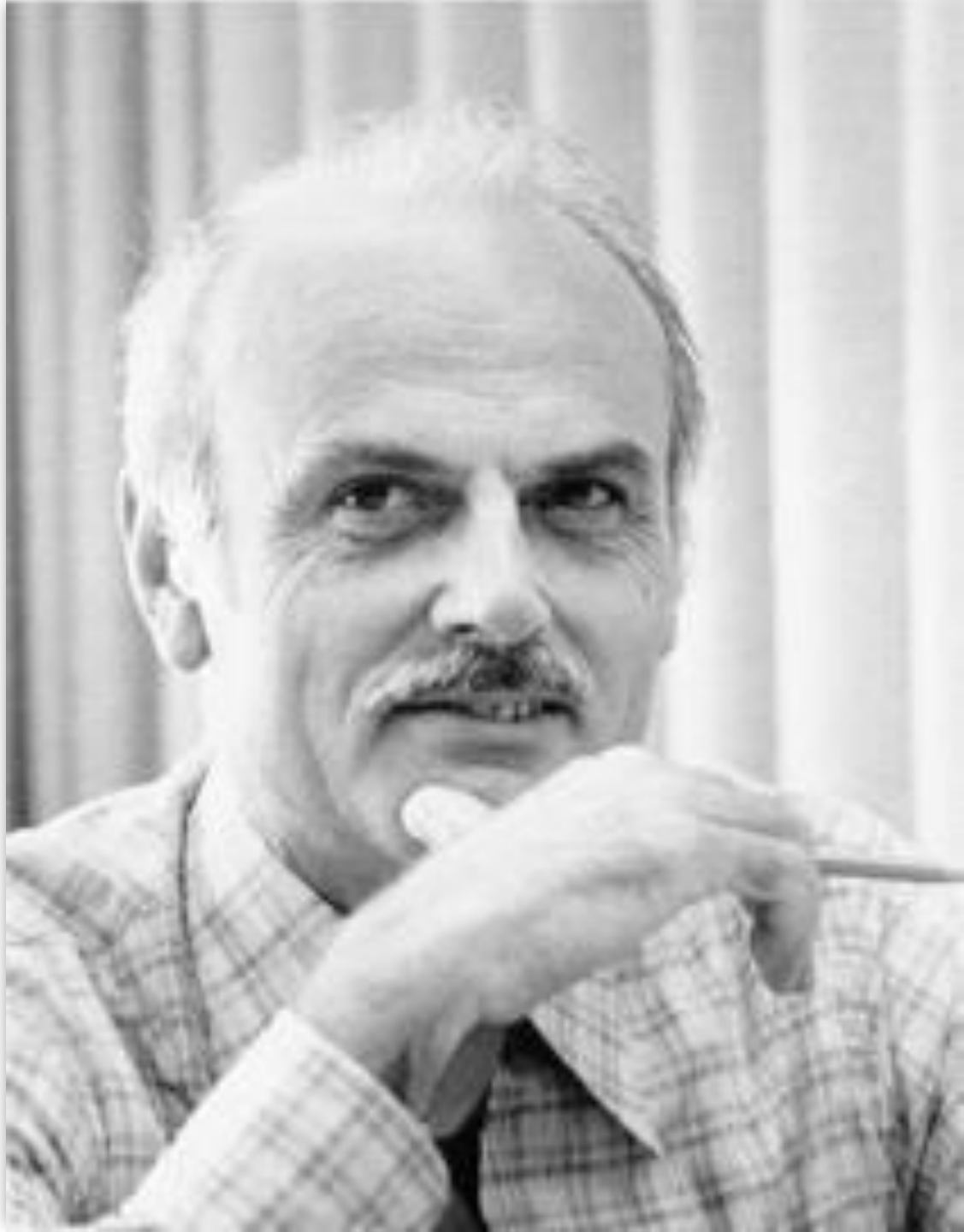


Figure 1-1. *The data engineering lifecycle*

Historie

Anfänge 1970 – Relationale Datenbanken

- Konzept erfunden von E. F. Codd bei IBM
- Publiziert im Paper "*A Relational Model of Data for Large Shared Data Banks*“
- Ursprünglich definiert durch 12 Regeln (Codd‘s 12 Rules)
- Diese wurden als zu umfangreich angesehen, weshalb sie sich heute auf zwei Regeln beschränken

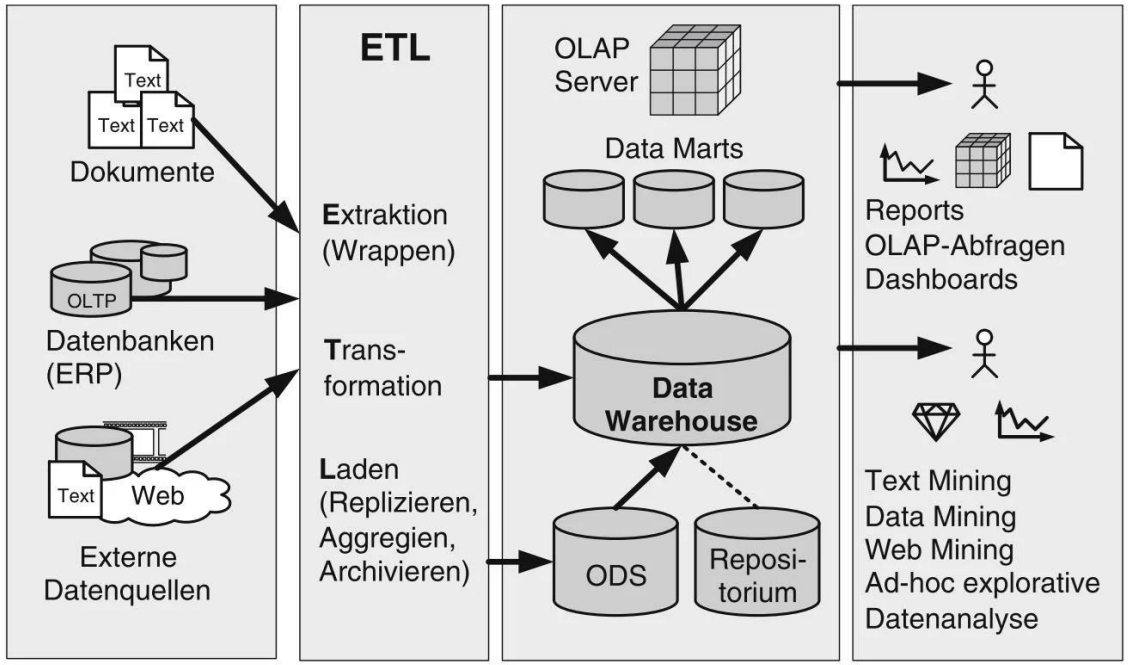


Anfänge 1970 – Relationale Datenbanken

- Die zwei wesentlichen:
  - Daten müssen in Relationen präsentiert werden (Tabellen mit einem Set aus Zeilen und Reihen)
  - Relationale Operatoren um Daten in tabellarischer Form zu manipulieren
- Gegründet von IBM, populär gemacht durch Oracle
- Anfänge des strukturierten Datenmanagements

Data Warehouse (80s)

- Data Warehouse (DWH) oder Enterprise Data Warehouse (EDW)
- System für **Berichterstellung und Datenanalyse**
- (Zentrale) Komponente der **Business Intelligence (BI)**



Funktion

- **Zentrales Repository** für integrierte Daten aus verschiedenen Quellen
- Speicherung von **aktuellen und historischen Daten**
- Dient der Erstellung von Berichten und **Datenanalyse**
- **Unterstützt datengetriebene Entscheidungsfindung**

Warum macht man Analysen in einer gesonderten Datenbank?

Aufstieg von Big Data (2000er/2010er)

- **Gründe für den Aufstieg**:
  - Wachstum der Datenmengen und der Bedarf an neuen Verarbeitungsstrategien mit Standard-Hardware
- **Die 3 V’s von Big Data**:
  - **Velocity** (Geschwindigkeit der Daten), **Variety** (Vielfalt der Daten), **Volume** (Datenmenge)

Meilensteine in der Big Data Entwicklung

- **Google File System (2003)** und **MapReduce (2004)**: Grundlage für skalierbare Datenverarbeitung
- **Yahoo und Hadoop (2006)**: Erste Implementierung eines Data-Lake-Ansatzes
- **Amazon Web Services (AWS)**: Einführung von EC2, S3, NoSQL-Datenbanken (DynamoDB) und flexiblem Pay-as-you-go-Modell
- **Hadoop gewinnt schnell an Popularität**

Warum ist die Public Cloud elementar für die Entwicklung von Big Data?

Data Lake in a nutshell

Probleme relationaler DWHs

DWHs halten keine Rohdaten

- Daten werden für spezifische Anaylsen ausgewählt, verdichtet und strukturiert
- Anders artige Analysen auf den selben Quelldaten lassen sich nicht mehr durchführen
- Design von Datenstrukturen und Aufbereitung der Daten ist sehr zeitaufwendig

Big Data = Big Money

- Menge der Compute Ressourcen steigt linear mit der Menge der Daten (bei allen Datenbanken der Fall)
- Ungenutzte Daten (Cold Data) treiben die kosten der Systeme in die Höhe
- Datenmengen werden systematisch reduziert um Kosten im Griff zu behalten

Probleme relationaler DWHs

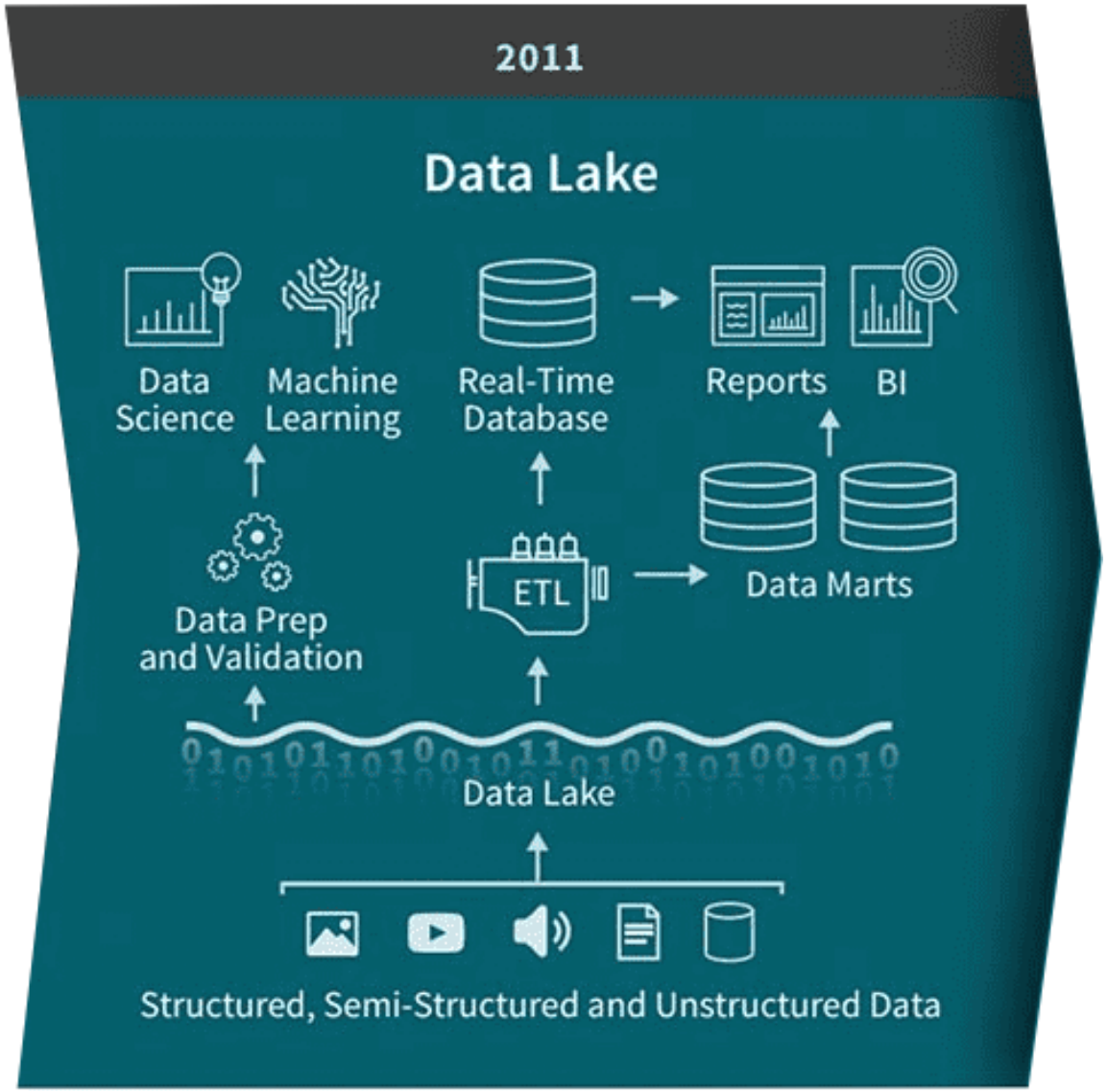
- Unstrukturierte und Semistrukturierte Daten sind für relationale DWHs ungeeignet
- Andere DBMs werden benötigt
- Analysen über Daten aus mehreren Quellen hinweg nur sehr schwer möglich
- Als zentrale Single Source of Truth für alle Daten im Unternehmen ungeeignet

Herausforderungen

- **Von Batch- zu Stream-Processing**: Echtzeit-Datenverarbeitung als nächste Revolution
- **Zunehmende Komplexität**:
  - Immer mehr Tools, die für den Data Engineer erforderlich wurden: BI/Data Science Wissen, Softwareentwicklung, Infrastruktur, Automatisierung und verteilte Systeme

Idee eines Data Lakes

- Komponenten einer Datenbank werden in **eigenständige**, voneinander unabhängige und **hochgradig skalierbare Systeme unterteilt**
- Keine Datenbank sondern **verteilter, skalierbarer Dateispeicher**
- Keine Abhängigkeit mehr zwischen Compute und Storage (bei Hadoop teilweise anders)
- Komponenten sind je nach Anwendungsfall austauschbar
- Keine Restriktion bezüglich Datenstruktur
- Zentrale Plattform für alle Daten einer Organisation



Zurück zur Historie ☺

Big Data - Übermäßige Komplexität

- Unternehmen bauten komplexe Architekturen, die oft nicht den erwarteten Mehrwert brachten
- Tatsächliche Datenmengen sind oft sehr klein
- Hadoop und das Ökosystem sind sehr komplex zu betreiben
- Der klassische Data Lake Ansatz hat konzeptionelle Probleme

Big Data in a nutshell

„Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it.”

Dan Ariely

Ende des Hypes

- Big Data ist in sehr vielen Unternehmen gescheitert
- Massive Investitionen, Datensümpfe, nicht nutzbare Daten, schwer zu betreibende Systeme
- Hadoop ist tot

Die 2020er – Evolution der Systeme

- Modularisierung und Abstraktion von Frameworks
- Weniger low-level Arbeiten, mehr Management und Orchestrierung
- Das Data Lakehouse – Fusion aus Data Lake und Data Warehouse – funktionierende Data Lake Evolution
- Aufkommen des KI Hypes benötigt neue Datengrundlage

Neue Herausforderungen und Aufgaben:

- Datenqualität, Datenschutz, Sicherheitsanforderungen
- **DataOps** und **Data Lifecycle Management** werden zentral
- Konzepte wie das „*Data Mesh*“ kommen auf

Zunehmende Bedeutung von Data Engineering

- **Data Engineer** als zentrale Rolle, um Dateninfrastrukturen für Analytik, KI und maschinelles Lernen bereitzustellen

Unterschied zwischen Data Engineering und Data Science

- **Ergänzend, aber unterschiedlich**:
  - Data Engineers arbeiten "upstream" (Datenvorbereitung, Infrastruktur)
  - Data Scientists arbeiten "downstream" (Modellierung, Analyse)
- **Data Engineering als Basis für erfolgreiche Data Science**:
  - 70-80% der Arbeit von Data Scientists und ML-Ingenieuren besteht aus der Datenaufbereitung
  - Solide Datenfundamente sind nötig, bevor man sich mit KI und maschinellem Lernen beschäftigt

Aufgaben eines Data Engineers

- Datenaufbereitung, Bereinigung
- Datenmodellierung
- Schreiben und Orchestrieren von Datenpipelines
- Architektur verteilter Systeme, verteilte Speicherung und Verarbeitung

Organisatorisches

Prüfungsleistung: Projektarbeit

- **Projektaufgabe:**
  - Daten aus einer beliebigen Quelle laden (z.B. **Web Scraping**)
  - Daten **verarbeiten** und in einer geeigneten **Datenstruktur** speichern
  - Speichern der Daten in einem **System Ihrer Wahl**
- **Analyseteil:**
  - Führen Sie **Analysen** auf den verarbeiteten Daten durch
- **Auswahlmöglichkeiten:**
  - Klassischer Analytics Use Case
  - Such-Optimierung
  - Verarbeitung von Echtzeitdaten

Studienleistung

- **Keine…**

Abwesenheit

- 09.10.2025
- 30.10.2025
- 21.11.2025