

programmierung und datenbanken

Joern Ploennigs

Datenbanktypen

MIDJOURNEY: DATABASE TREE, REF. GUSTAV KLIMT

WIEDERHOLUNG: HÖRSAALFRAGE

Was ist Vererbung, Generalisierung, Kapselung, Polymorphismus?



Midjourney: Object oriented man

WIDERHOLUNG: OBJEKTEIGENSCHAFTEN

Vererbung

Attribute und Methoden von Elternklassen werden an Kinder vererbt. Das hilft Redundanzen und Fehler zu vermeiden.

Generalisierung

Gemeinsamkeiten werden in generalisierten Eltern-Klassen implementiert

Polymorphismus

Kinderklassen können Methoden überschreiben und somit umdefinieren.

Kapselung

Kapselung von Daten und Methoden in Objekten ist ein Schutzmechanismus, um schadhafte Änderungen einzuschränken.

WIEDERHOLUNG: HÖRSAALFRAGE

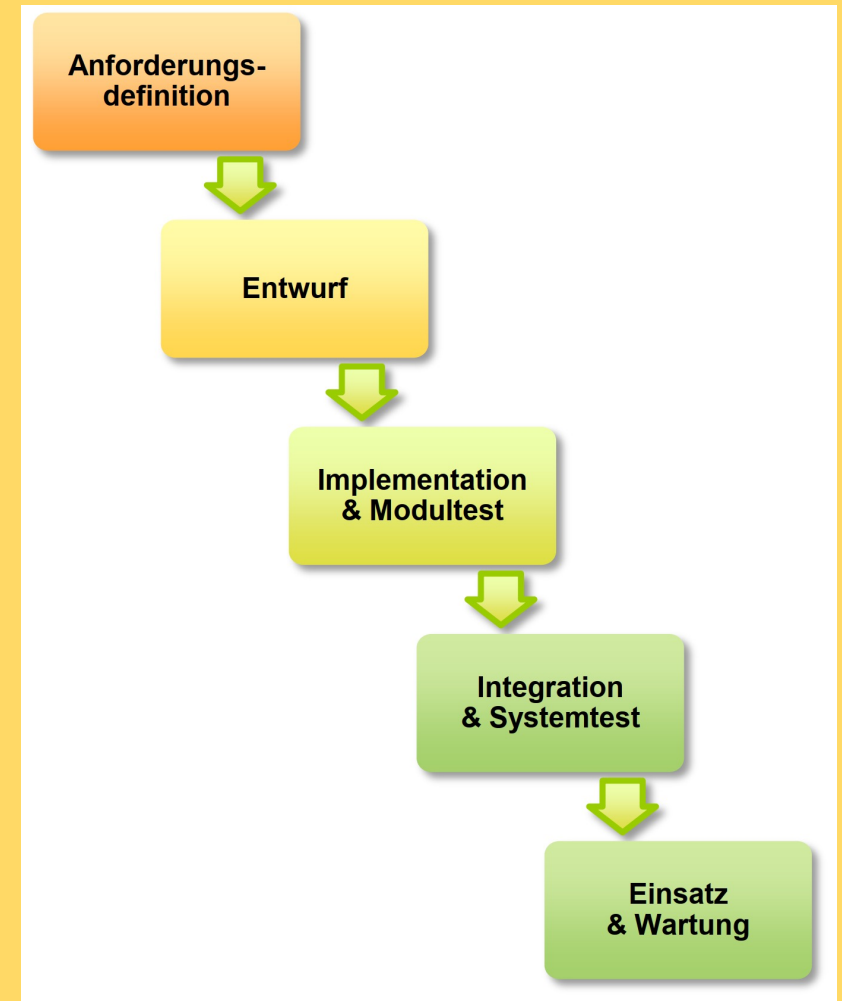
Was ist die Wasserfallmethode?



Midjourney: Waterfall

LINEARE METHODE - WASSERFALLMETHODE

- Traditionelles Modell das häufig noch in der Ausschreibung großer Systeme gefordert wird
- Entwicklung ist in mehrere sequenzielle Schritte aufgeteilt und jeder Schritt muss vor dem nächsten beendet werden
- Benutzerbeteiligung nur in der Anforderungsdefinition
- Jeder Aktivität wird dokumentiert → Gut geeignet für Ausschreibungen (nach der Anforderungsdefinition oder dem Entwurf, ISO 9000)



WIEDERHOLUNG: HÖRSAALFRAGE

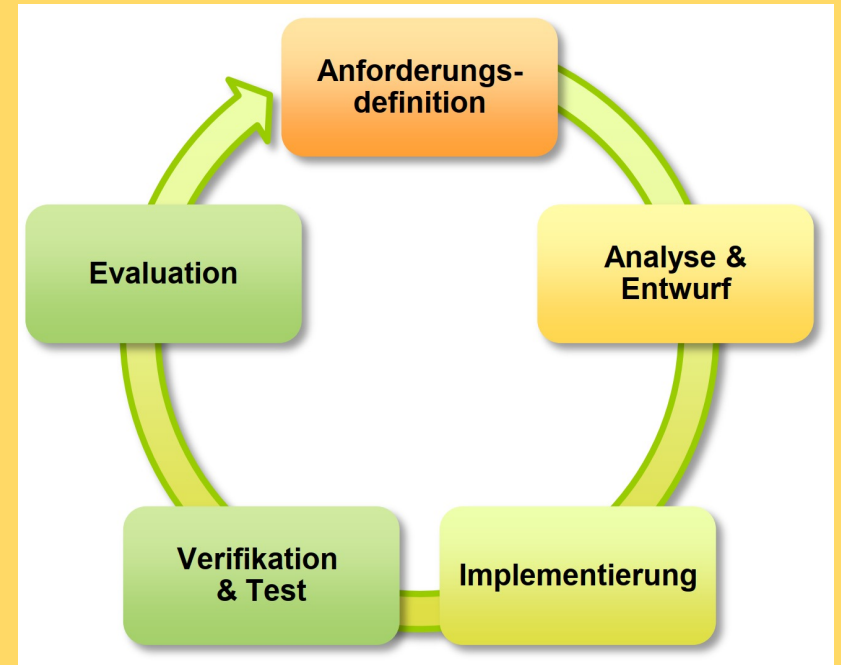
Was ist die Agile Methode?



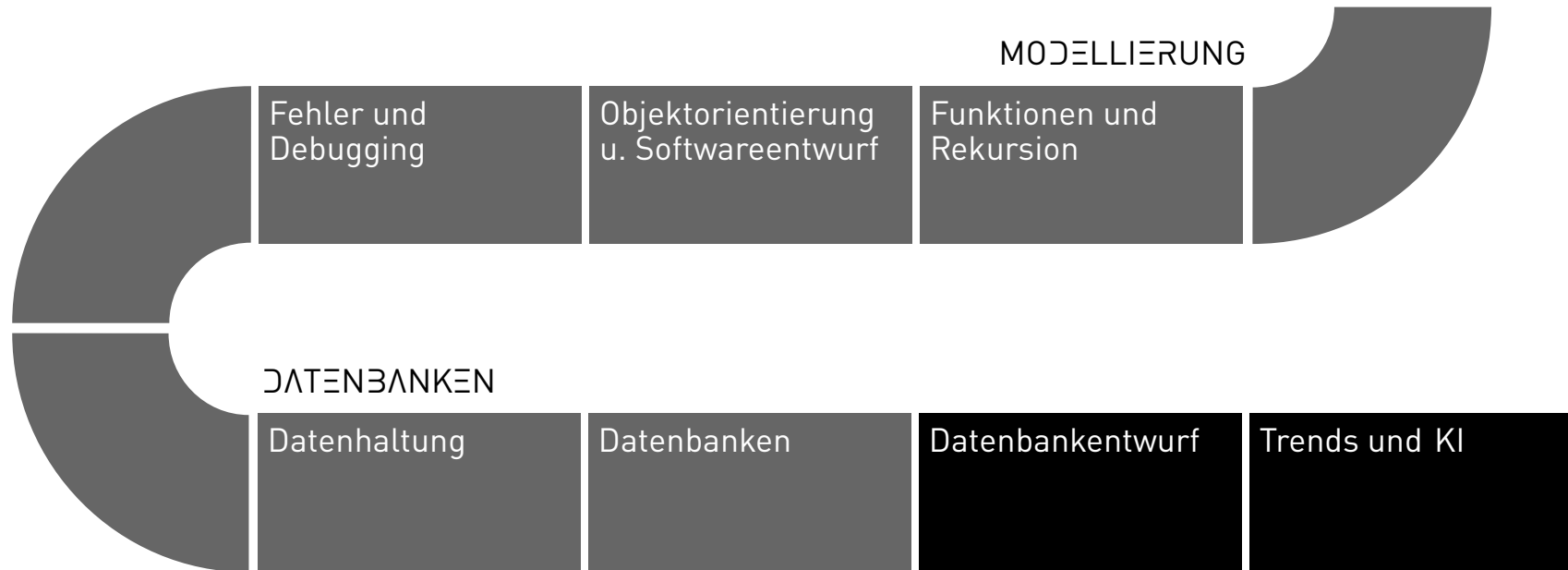
Midjourney: Sprinter and Waterfall

AGILE METHODE

- Moderne Methode um mit sich an ständig ändernden Anforderungen anpassen
- Ziel der schrittweisen Entwicklung der Lösung, um Aufwand und Komplexität einzelner Schritte in Grenzen zu halten
- Startet mit einer einfachen und ausbaufähigen Implementierung (MVP – Minimal Viable Produkt)
- Schrittweise Erweiterung und Verbesserung des Produktes mit regelmäßigen Releases (meist alle 3 Monate)
- Entwurfsfehler der ersten Iterationen können zu kompletten Neuentwurf führen



ABLAUF



DATENBANKEN DEFINITION

Definition: Datenbank

Unter einer *Datenbank (DB)* versteht man die logisch zusammengehörenden Daten, die von einem *DBMS (Datenbankmanagementsystem)* verwaltet werden

Definition: Datenbanksystem

Datenbank und Datenbankmanagementsystem zusammen bezeichnet man als *Datenbanksystem*

Zur Datenbank gehören neben den reinen 'Nutzdaten' auch die für die Verwaltung vom Datenbankmanagementsystem angelegten Objekte (beispielsweise Indizes und Logdateien)

WARUM BRINGEN WIR GELD „AUF DIE BANK“?

- Zentrale und langfristige Aufbewahrung
- Sicherung vor Verlusten
- Effizienz durch spezielles Dienstleistungsangebot (Daueraufträge, Portfolio, ...)
- Übersicht behalten
- Vernetzung mit dem globalen Finanznetzwerk



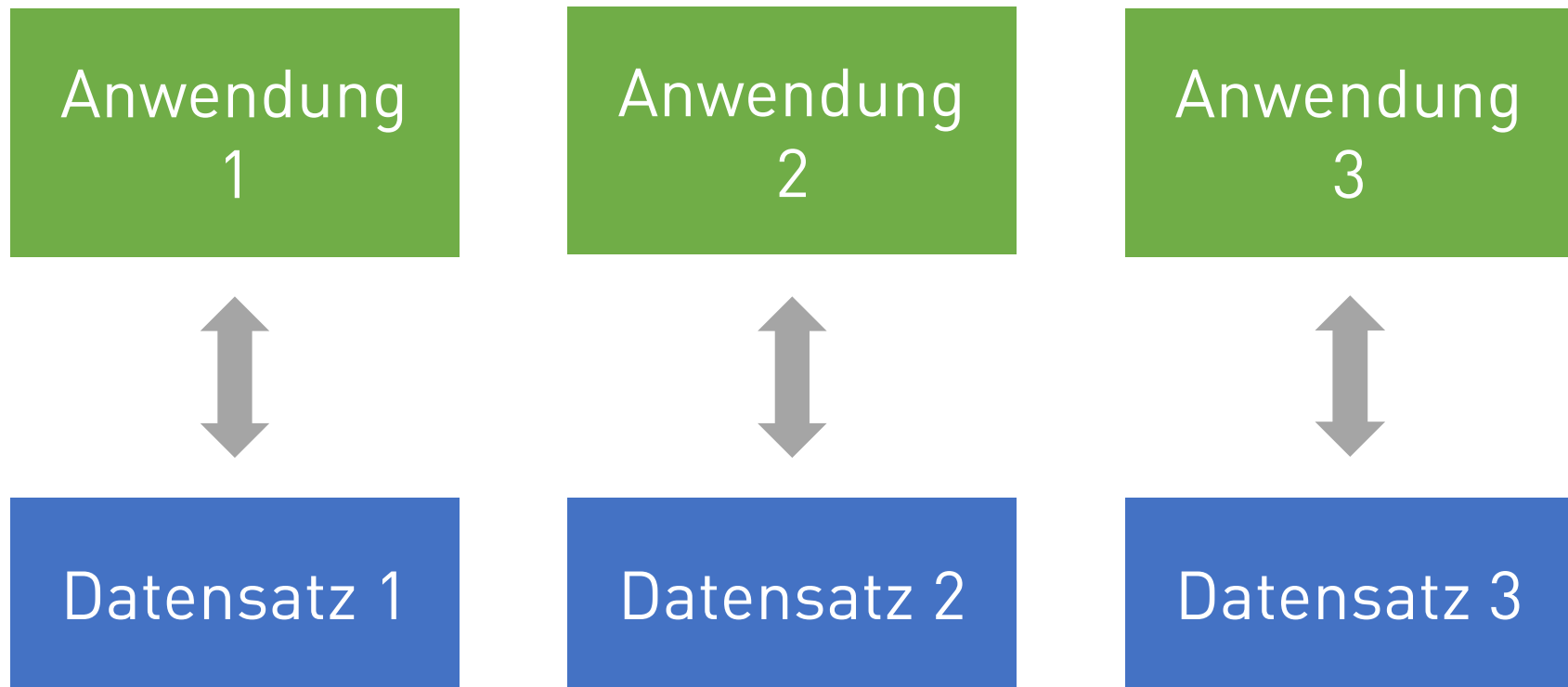
Midjourney: thief breaking into a bank safe

WARUM BRINGEN WIR DATEN AUF DIE „BANK“?

- *Effizienter* speichern und laden von Daten durch verschiedenen Clients (Webserver, Geräte, etc.)
- *Verwaltung* von sehr (sehr) großen Datenmengen (Skalierbarkeit)
- *Organisation* der Daten in vorgegebene Datenstrukturen (Normalisierung)
- *Langfristiges* speichern der Daten (Persistierung)
- *Schnelles* suchen der Daten durch Indexierung
- *Abgesicherte* Prozesse zum Ändern der Daten (Transaktionen)
- *Nachvollziehbares* Verändern von Daten durch Transaktions-Logs
- *Automatische* Datenanalyse (OLAP)

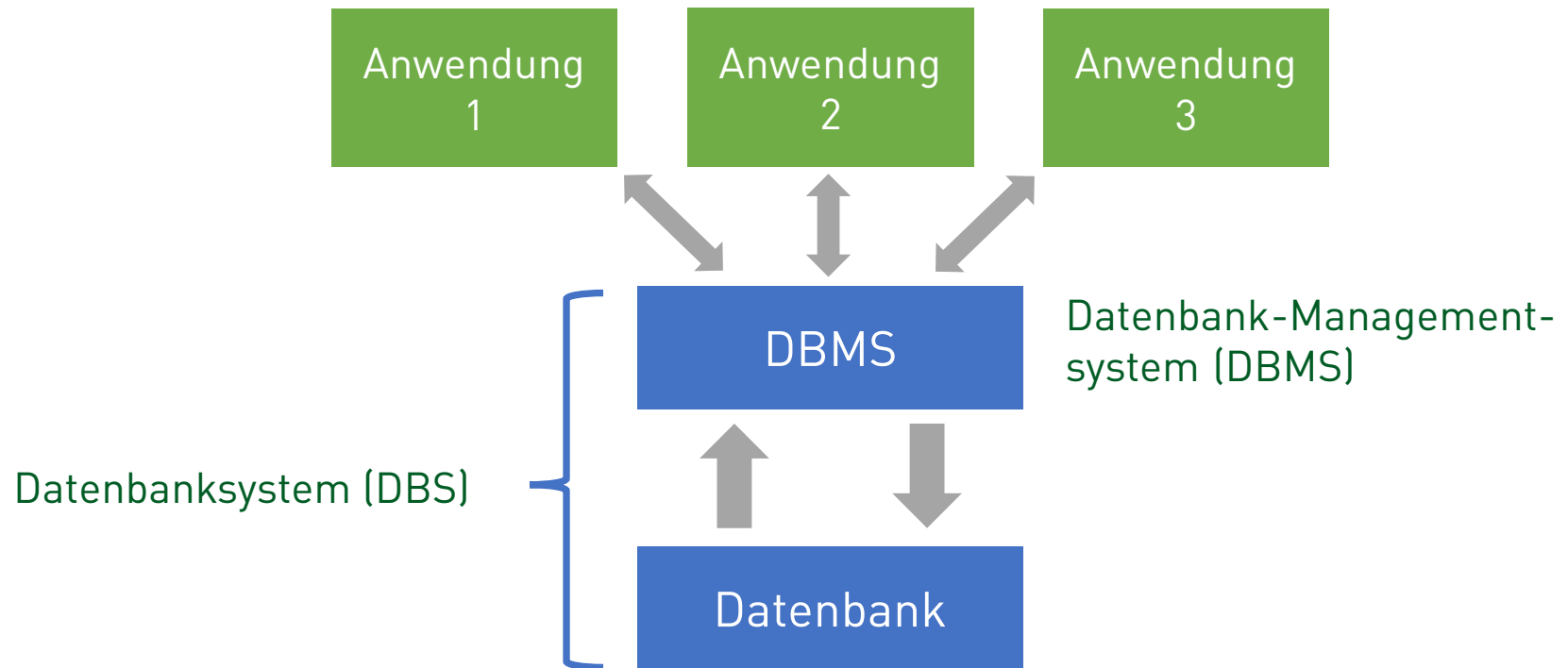
GRUNDKONZEPTE DATEIEN

- Struktur: Jede Anwendung strukturiert die Daten entsprechend der darin vorkommenden Datenarten (Format, Struktur ...).
- Dateisystem: Jede Anwendung legt die Daten entsprechend seiner Anforderungen (Zugriff, Endung, Ort ...) ab.



GRUNDKONZEPTE DATENBANKEN

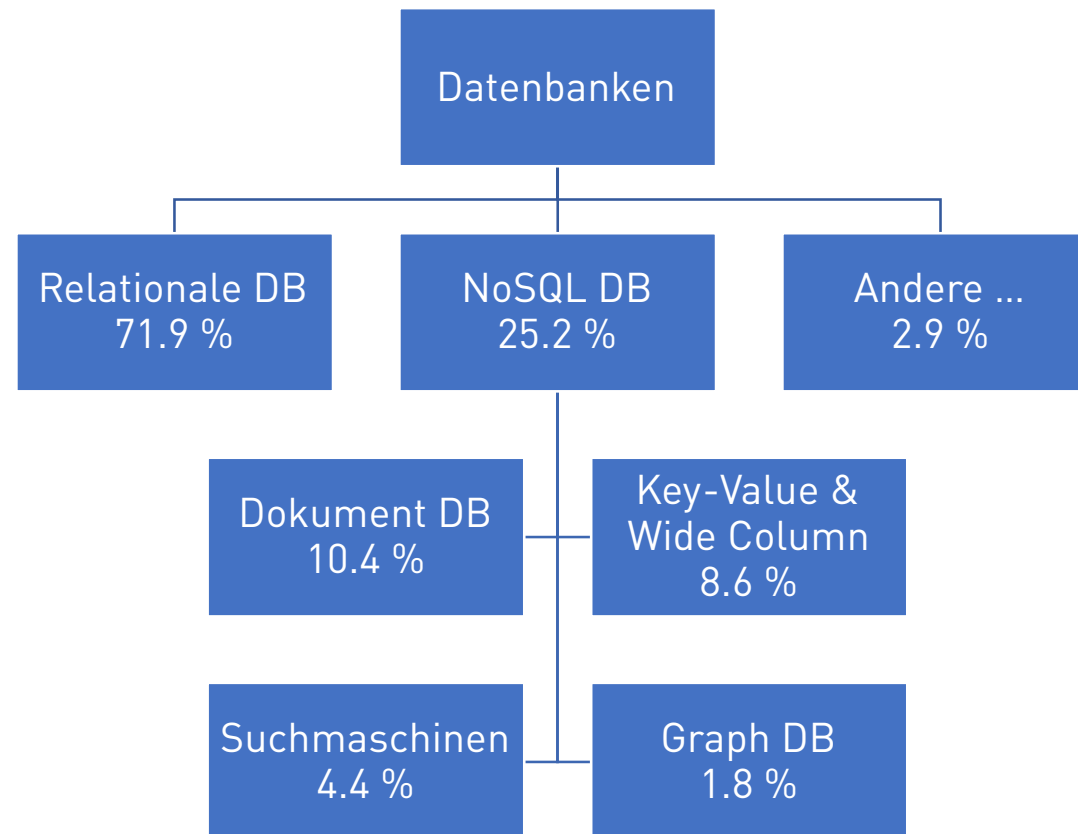
- Struktur: Alle Anwendung nutzen die gleiche Struktur die in der Datenbank modelliert wurde
- Dateisystem: Alle Anwendungen greifen auf die gleichen Daten zu. Zugriffe werden durch Transaktionen synchronisiert und protokolliert.



DIE CODD'SCHEN REGELN (CODD, 1985, 1990)

- *Integration*: einheitliche, nichtredundante Datenverwaltung
- *Operationen*: Speichern, Suchen, Ändern
- *Katalog*: Zugriffe auf Datenbankbeschreibungen im Data Dictionary
- *Benutzersichten*: Jeder Nutzer sieht die Daten die er sehen darf in der Art wie er sie sehen möchte
- *Integritätssicherung*: Korrektheit des Datenbankinhalts
- *Datenschutz*: Ausschluss unauthorisierter Zugriffe, nur berechtigte Nutzer
- *Transaktionen*: mehrere DB-Operationen als Funktionseinheit (ganz oder gar nicht)
- *Synchronisation*: parallele Transaktionen koordinieren
- *Datensicherung*: Wiederherstellung von Daten nach Systemfehlern

DATENBANKEN TYPEN - ÜBERSICHT



RELATIONALE DATENBANKEN

RDBMS werden bereits seit Anfang der 1980er Jahre verwendet und basieren auf dem relationalen (=tabellenorientierten) Datenmodell

- Das Schema einer Tabelle (=Relationenschema) ist definiert durch den Tabellennamen und eine fixe Anzahl von Attributen (=Spalten) mit entsprechenden Datentypen
- Da Daten in Tabellen organisiert werden, sind sie stark strukturiert mit einer durch die Tabelle definierten Struktur (Normalisierung)
- Die Standardsprache zum Aufbau/Änderung/Löschen ist *SQL*
- Populäre Systeme: Oracle, MySQL, Microsoft SQL Server, PostgreSQL, IBM Db2

NoSQL DATENBANKEN

NoSQL Datenbankmanagementsysteme sind Datenbanken die kein relationales (=tabellenorientierten) Datenmodell verwenden und damit in der Regel auch kein SQL unterstützen

- Sie finden seit etwa 2009 zunehmend Verbreitung
- Populäre Systeme: MongoDB, CouchDB, Cassandra, Redis, Neo4j, Amazon DynamoDB, HBase, OrientDB
- *Hauptgründe:*
 - Hohe Anforderungen an Skalierbarkeit
 - Fehlertoleranz moderner Web-Applikationen
 - Big Data Szenarien
 - Daten sind oft nur semi-strukturiert (lassen sich nicht in ein Schema pressen)

DOKUMENTENORIENTIERTE DATENBANKEN

Document Stores zeichnen sich durch eine *schemafreie* Organisation der Daten aus:

- Datensätze haben keine einheitliche Struktur
- Die Typen der Werte einzelner Spalten können pro Datensatz unterschiedlich sein
- Spalten können mehr als einen Wert haben (Arrays)
- Datensätze können eine verschachtelte Struktur haben
- Zur Darstellung der „Dokumente“ wird meist *JSON* verwendet
- Populäre Systeme: MongoDB, Amazon DynamoDB, Databricks, Azure Cosmos DB, Couchbase

KEY-VALUE DATENBANKEN

Key-Value Stores sind die wohl einfachste Form von Datenbankmanagementsystemen

- Sie können lediglich Paare von Schlüsseln und Werten abspeichern, sowie die Werte anhand des Schlüssels wieder zurückliefern
- Damit ähneln sie dem *dict* in Python
- Diese Einfachheit macht sie attraktiv für:
 - Ressourcenbegrenzte Systeme wie embedded PCs
 - Entwicklung von Web-Interfaces
- Populäre Systeme: Redis, Amazon DynamoDB, Azure Cosmos DB, Memcached, Hazelcast

SUCHMASCHINEN DATENBANKEN

Suchmaschinen sind NoSQL DBMS spezialisiert auf die Suche nach Dateninhalten wie Text:

- Features
 - Unterstützung komplexer Suchbegriffe
 - Volltextsuche
 - Stemming (Stammformreduktion eines Wortes)
 - Ergebnisreihung
 - Gruppierung von Suchergebnissen
 - Verteilte Suche für hohe Skalierbarkeit
- Populäre Systeme: Elasticsearch, Splunk, Solr, OpenSearch, MarkLogic

GRAPHDATENBANKEN

Graph DBMS stellen Datensätze in Form von *Knoten (Nodes)* und *Beziehungen (Edges)* zueinander dar

- Sie ermöglichen insbesondere die *Modellierung von Verbindungen*
- Ideal für Netzwerkanalysen, soziale Netzwerke, Empfehlungssysteme
- Populäre Systeme: Neo4j, Microsoft Azure Cosmos DB, Virtuoso, IBM KITT

DATENBANKEN TYPEN IN DER VORLESUNG

- Fokus auf: *Relationale DB* (71.9%)
 - Einfach zu benutzen
 - Meist verbreitet
 - Solide Grundlage
- Auch: Key-Value DB:
 - Einfache Konzepte
 - Praktische Übungen

DATENBANKEN IN PYTHON

- Zugriff über jeweils passende Bibliothek der Datenbank
- Oracle, Redis, MongoDB, etc.

```
from replit import db

# Daten speichern
db["entry"] = 5

# Daten abrufen
value = db["entry"]
print(value)  # Output: 5
```


fragen?

