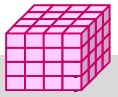
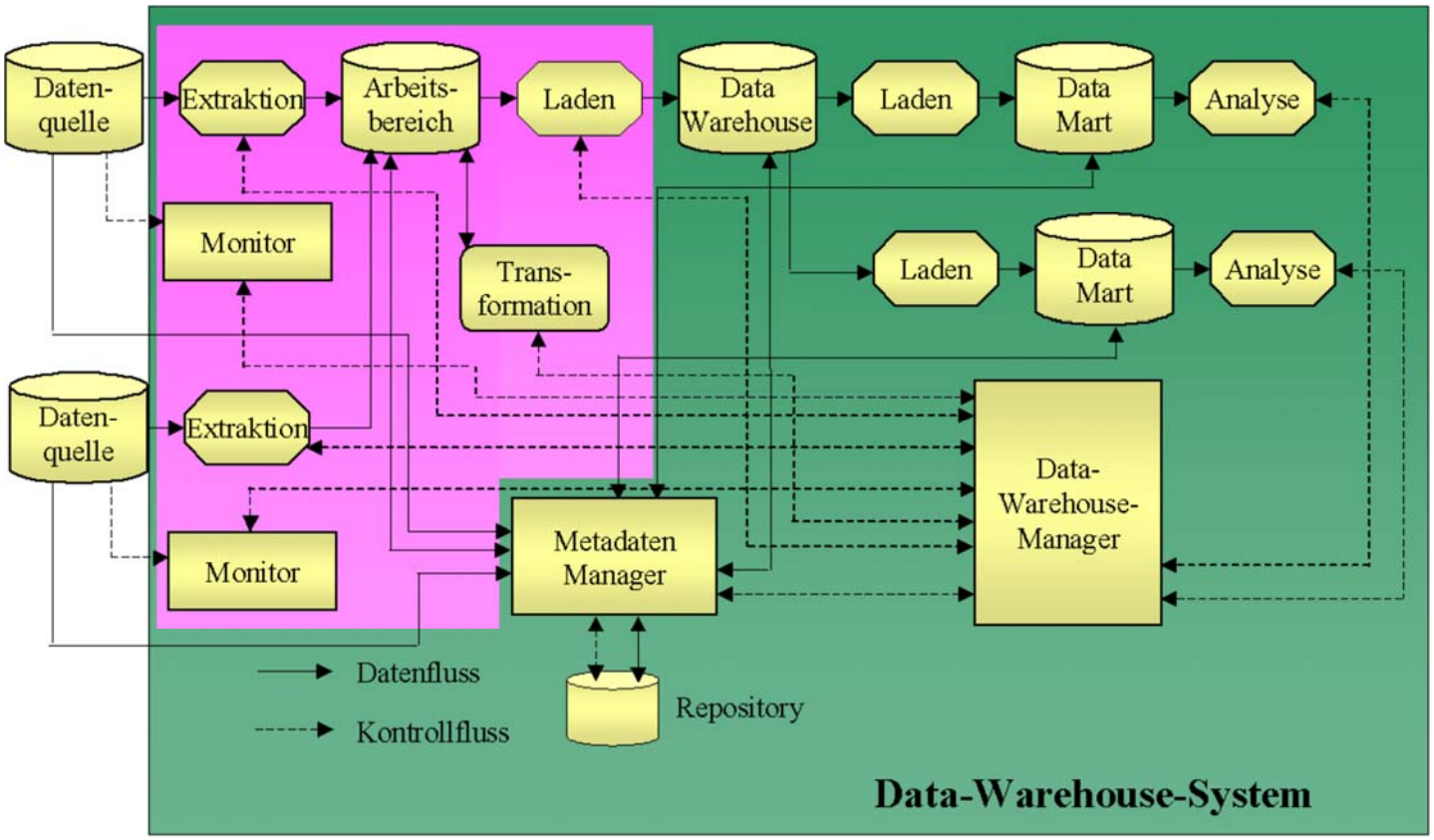
2. Architektur von Data Warehouse-Systemen

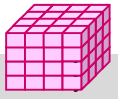
- Referenzarchitektur
  - Scheduler, Datenquellen, Datenextraktion, Transformation und Laden
- Abhängige vs. unabhängige Data Marts
- Operational Data Store (ODS)
- Metadatenverwaltung
  - technische vs. fachliche Metadaten
- DWH und Big Data
  - Data Lake
  - Kombinationen DWH + Lake (Data Fabric, Data Lakehouse)



DW-Referenzarchitektur

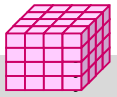


Quelle: Bauer/Günzel



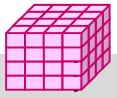
Phasen des Data Warehousing

1. Überwachung der Quellen auf Änderungen durch Monitore
2. Extrahieren/Kopieren der relevanten Daten in temporären Arbeitsbereich
3. Transformation der Daten im Arbeitsbereich (Bereinigung, Integration)
4. Laden der Daten ins Data Warehouse (DW)
5. Laden der Daten in Data Marts (DM)
6. Analyse: Operationen auf Daten des DW oder DM



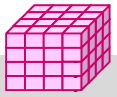
Datenquellen

- Lieferanten der Daten (gehören nicht direkt zum DW)
- Merkmale
  - intern (Unternehmen) oder extern (z.B. von Datenmarktplätzen /Web)
  - ggf. kostenpflichtig
  - i.a. autonom
  - i.a. heterogen bzgl. Struktur, Inhalt und Schnittstellen (Datenbanken, Dateien)
- Qualitätsforderungen:
  - Verfügbarkeit von Metadaten
  - Konsistenz (Widerspruchsfreiheit), Korrektheit (Übereinstimmung mit Realität)
  - Vollständigkeit (z.B. keine fehlenden Werte oder Attribute)
  - Aktualität, Verständlichkeit
  - Verwendbarkeit: Zugriffsmöglichkeiten auf Daten und Änderungen



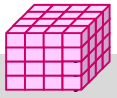
Data-Warehouse-Manager/Scheduler

- Ablaufsteuerung: Initiierung, Steuerung und Überwachung der einzelnen Prozesse
- Initiierung des Datenbeschaffungs-/Extraktionsprozesses und Übertragung der Daten in Arbeitsbereich
  - in regelmäßigen Zeitabständen (jede Nacht, am Wochenende etc.)
  - bei Änderung einer Quelle
  - auf explizites Verlangen durch Administrator
- Fehlerfall
  - Dokumentation von Fehlern
  - Wiederanlaufmechanismen
- Zugriff auf Metadaten aus dem Repository
  - Steuerung des Ablaufs
  - Parameter der Komponenten



Datenextraktion

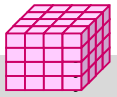
- Monitore: Entdeckung von Änderungen in einer Datenquelle
  - interne Datenquellen: aktive Mechanismen
  - externe Datenquellen: Polling / periodische Abfragen
- Extraktionskomponenten: Übertragung von Daten aus Quellen in Arbeitsbereich
  - Periodisch, auf Verlangen durch Admin oder ereignisgesteuert (z.B. sofort nach jeder Änderung oder nach bestimmter Zahl von Änderungen)
- Kriterien
  - Online- vs- Offline Zugang zu Datenquellen
  - inkrementelle Updates oder vollständige Kopien
- unterschiedliche Funktionalität der Quellsysteme
  - Nutzung von Standardschnittstellen (z.B. ODBC/JDBC) oder Eigenentwicklung
  - Nutzung spezieller Funktionalität, z.B. von DBS-Quellen
  - ggf. nur Offline-Bereitstellung von Dateien



Datenextraktion: Strategien

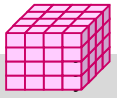
- Snapshots: periodisches Kopieren der Daten in Datei
  - Abgleich in Staging Arera mit Vorversion (z.B. mit SQL MERGE Statement)
- Nutzung von Change Data Capture (CDC) von Quell-DBS
  - DBS erzeugt „Change tables“
  - ggf. Nutzung von Log-Dateien
- Nutzung von (Last-Change-) Timestamps in Quell-Datenbanken
- Trigger
  - Auslösen bei Datenänderungen und Kopieren der geänderten Tupel in Change Tables

	Autonomie	Performanz	Nutzbarkeit
Snapshot			
CDC/Log			
Timestamps			
Trigger			



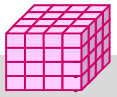
Datentransformation und Laden

- *Arbeitsbereich* (engl.: *Staging Area*)
  - temporärer Zwischenspeicher zur Integration und Bereinigung
  - Laden der Daten ins DW erst nach erfolgreichem Abschluss der Transformation
  - keine Beeinflussung der Quellen oder des DW
  - keine Weitergabe fehlerbehafteter Daten
- *Transformationskomponente*: Vorbereitung der Daten für Laden
  - Data Auditing/Profiling: Datenüberprüfung, Aufspüren von Abweichungen
  - **Data Cleaning**: Beseitigung von Verunreinigungen, fehlerhafte oder fehlende Werte, Redundanzen, veralteten Werte
  - Konsolidierung: Vereinheitlich von Datentypen, Datumsangaben, Maßeinheiten, Kodierungen etc.
- *Ladekomponente*: Übertragung der bereinigten und aufbereiteten (z.B. aggregierten) Daten in DW
  - Nutzung spezieller Ladewerkzeuge (z.B. Bulk Loader)
  - Historisierung: zusätzliches Abspeichern geänderter Daten anstatt Überschreiben
  - Offline (batch) vs. Online-Laden (Verfügbarkeit des DW während des Ladens)



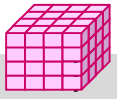
Data Warehouse

- relationale, mehrdimensionale oder kombinierte Speicherung der Daten (ROLAP, MOLAP, HOLAP)
- oft Trennung zwischen
  - relationalem Basis-DB (Warehouse) mit Detaildaten und
  - mehreren abgeleiteten Datenwürfel (Cubes) bzw. Data Marts mit aggregierten Daten
- Änderungen im (Basis-) Data Warehouse nach Laden müssen auf Cubes/Data Marts angewandt werden



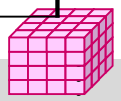
Data Marts

- Was ist eine Data Mart?
  - eine Teilmenge des Data Warehouse
  - inhaltliche Beschränkung auf bestimmten Themenkomplex oder Geschäftsbereich
- führt zu verteilter DW-Lösung
- Gründe für Data Marts
  - Performance: schnellere Anfragen, weniger Benutzer, Lastverteilung
  - Eigenständigkeit, Datenschutz
  - ggf. schnellere Realisierung
- Probleme
  - zusätzliche Redundanz
  - zusätzlicher Transformationsaufwand
  - erhöhte Konsistenzprobleme
- Varianten
  - abhängige Data Marts
  - unabhängige Data Marts

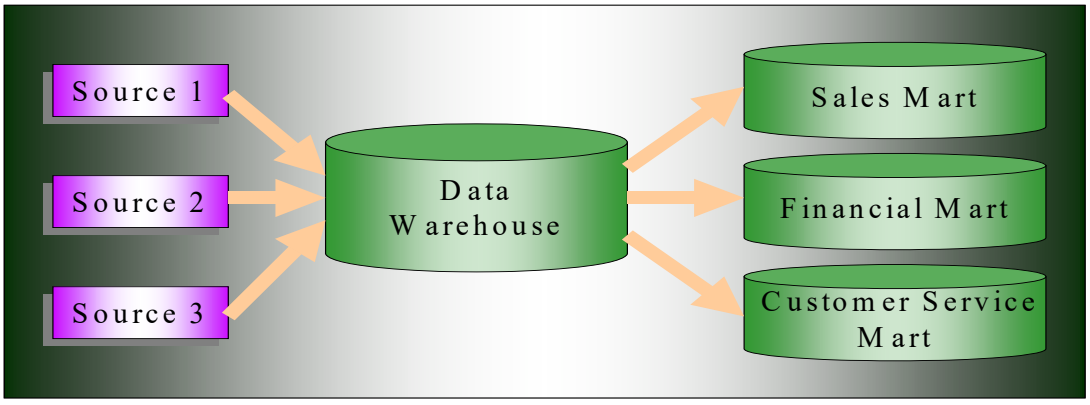


Data Warehouse vs. Data Mart

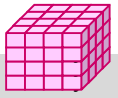
	Data Warehouse	Data Mart
Philosophie	anwendungsneutral	anwendungsbezogen
Adressat der Datenbereitstellung	Unternehmen	Abteilung
Datenmenge / Detaillierungsgrad	hoch	gering
Umfang historischer Daten	hoch	gering
Optimierungsziel	Datenmenge	Antwortzeiten
Anzahl	eins (wenige)	mehrere
Typische DB-Technologie	relational	multidimensional



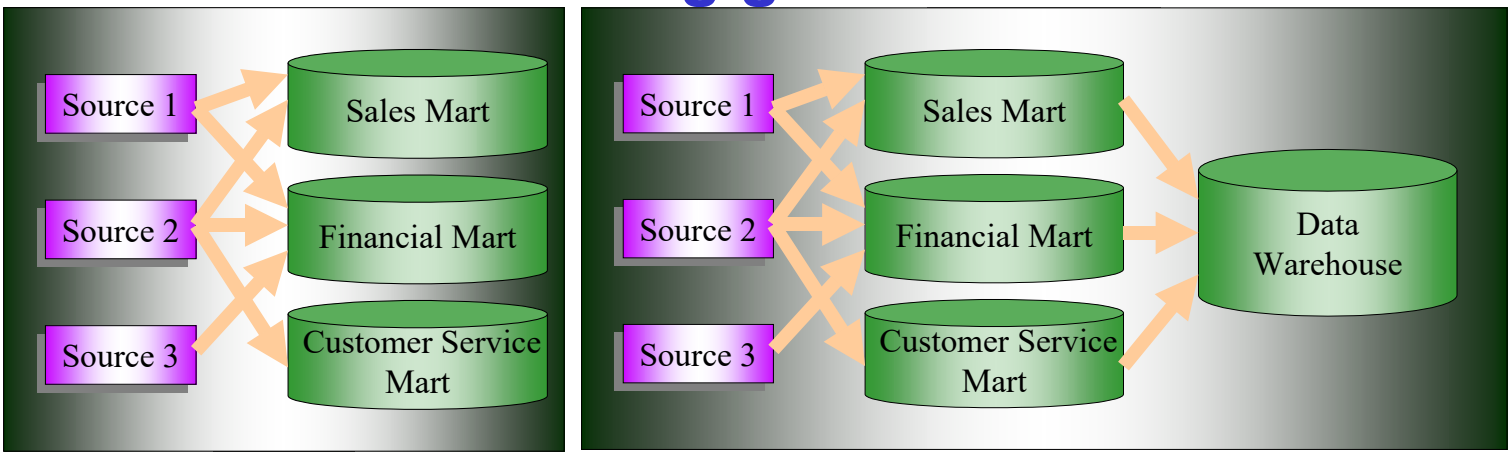
Abhängige Data Marts



- Hub-and-Spoke-Architektur („Nabe- und Speiche“)
- Data Marts sind Extrakte aus dem zentralen Warehouse
  - strukturelle Ausschnitte (Teilschema, z.B. nur bestimmte Kennzahlen)
  - inhaltliche Extrakte (z.B. nur bestimmter Zeitraum, bestimmte Filialen ...)
  - Aggregierung (geringere Granularität), z.B. nur Monatssummen
- Vorteile:
  - relativ einfach ableitbar (Replikationsmechanismen des Warehouse-DBS)
  - Analysen auf Data Marts sind konsistent mit Analysen auf Warehouse
- Nachteil: Entwicklungsdauer (Unternehmens-DW zunächst zu erstellen)



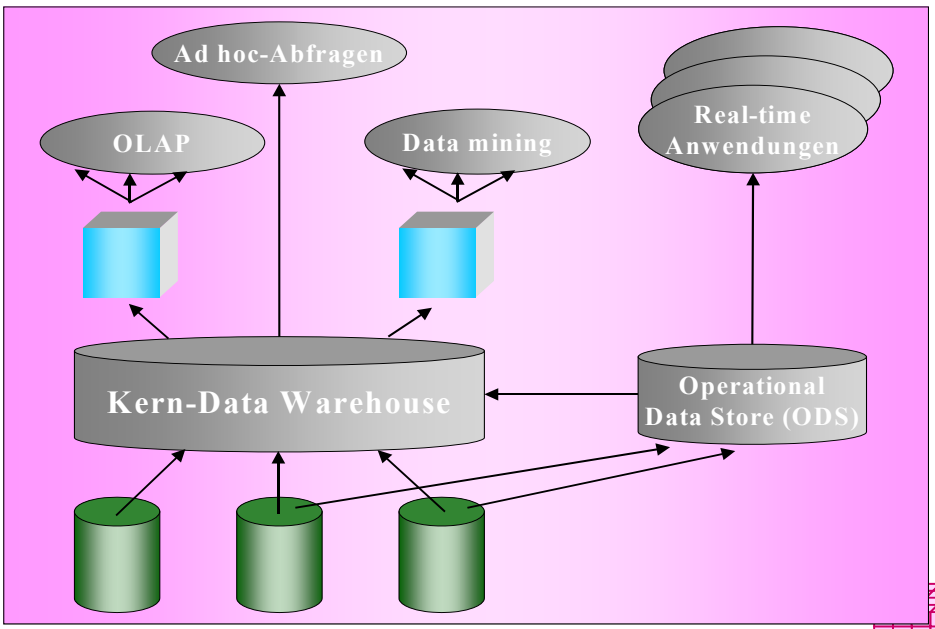
Unabhängige Data Marts



- Variante 1: kein zentrales, unternehmensweites DW
  - wesentlich einfachere und schnellere Erstellung der DM verglichen mit DW
  - Datenduplizierung zwischen Data Marts, Gefahr von Konsistenzproblemen
  - Aufwand wächst proportional zur Anzahl der DM
  - schwierigere Erweiterbarkeit
  - keine unternehmensweite Analysemöglichkeit
- Variante 2: unabhängige DM + Ableitung eines DW aus DM
- Variante 3: unabhängige DM + Verwendung gemeinsamer Dimensionen

Operational Data Store (ODS)

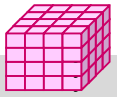
- optionale Komponente einer DW-Architektur zur Unterstützung operativer (Realzeit-) Anwendungen auf integrierten Daten
  - größere Datenaktualität als Warehouse
  - direkte Änderbarkeit der Daten
  - geringere Verdichtung/Aggregation, da keine primäre Ausrichtung auf Analysezwecke
- Probleme
  - weitere Erhöhung der Redundanz
  - geänderte Daten im ODS



Vergleich ODS - DW

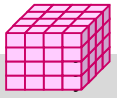
	Operational data store	Data warehouse
Best suited for	Granular, low-level queries against detailed operational data	Complex queries against summary or aggregated analytical data
Purpose	Operational reporting; current or near real-time reporting	Historical and trend analysis reporting on a large volume of data
Data duration	Contains a short window of data	Contains the entire history of the organization's data
Decision making	Supports operational and tactical decisions on current or near real-time data	Provides feedback on strategic decisions, leading to overall system improvements
Data load frequency	Might load data every few minutes or hourly	Might load data daily, weekly, monthly, or quarterly

Serra: Deciphering data architectures, 2024

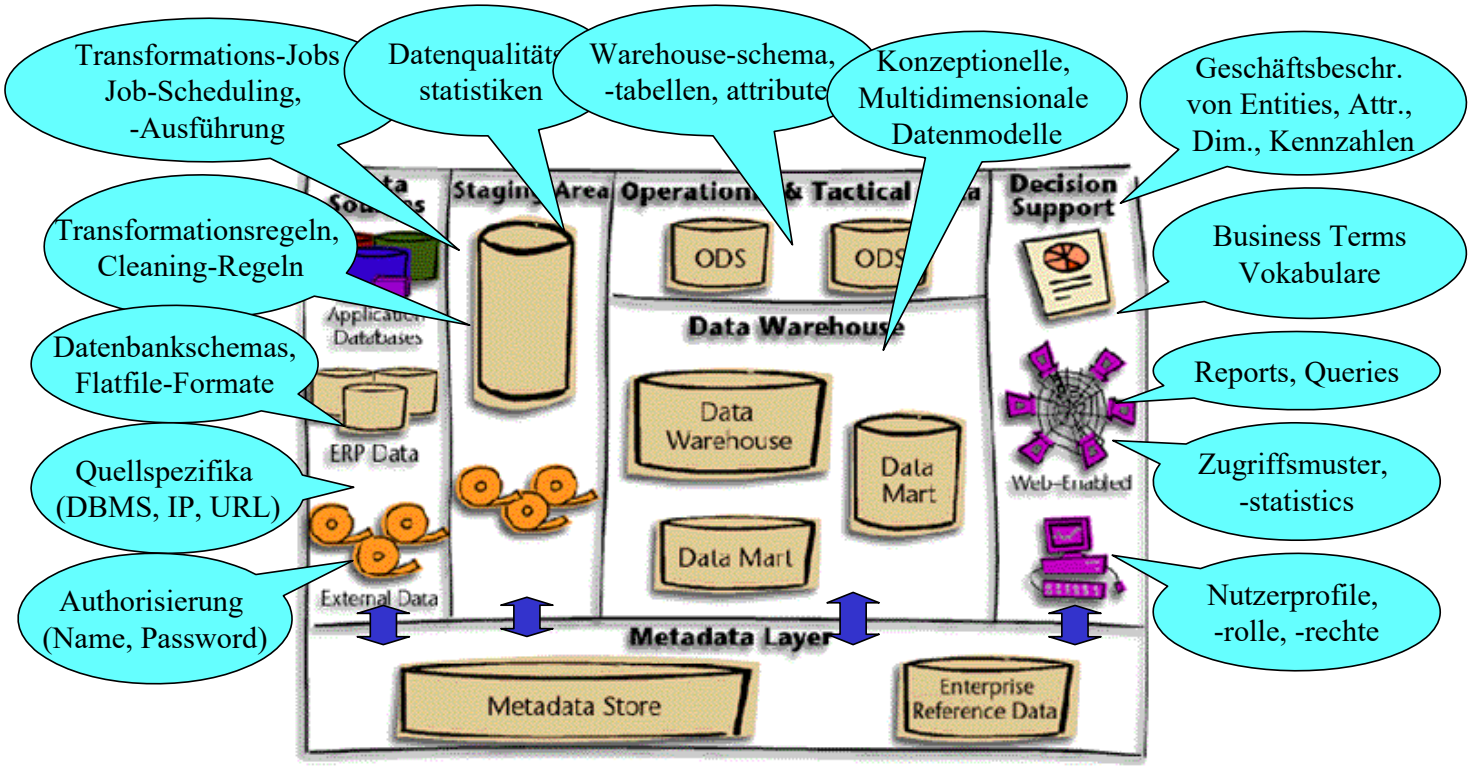


Metadaten-Verwaltung

- Anforderungen an Metadaten-Verwaltung / Repository
  - Bereitstellung aller relevanten Metadaten auf aktuellem Stand
  - flexible Zugriffsmöglichkeiten (DB-basiert) über mächtige Schnittstellen
  - Versions- und Konfigurationsverwaltung
- Unterstützung für technische und fachliche Aufgaben und Nutzer
  - Technische Metadaten vs. Business-Metadaten
- Realisierungsformen
  - werkzeugspezifisch: fester Teil von Werkzeugen (Design-Tool, ETL, Analyse-Tool)
  - allgemein einsetzbar: generisches und erweiterbares Repository-Schema (Metadaten-Modell)
- zahlreiche proprietäre Metadaten-Modelle
  - erfordert häufigen Austausch sowie Transformation/Integration von Metadaten
- Standardisierungsbemühungen mit begrenztem Erfolg
  - Open Information Model (OIM) - wurde 2000 eingestellt
  - Common Warehouse Metamodel (CWM) der OMG (Object Management Group)



Metadaten im Data Warehouse-Kontext

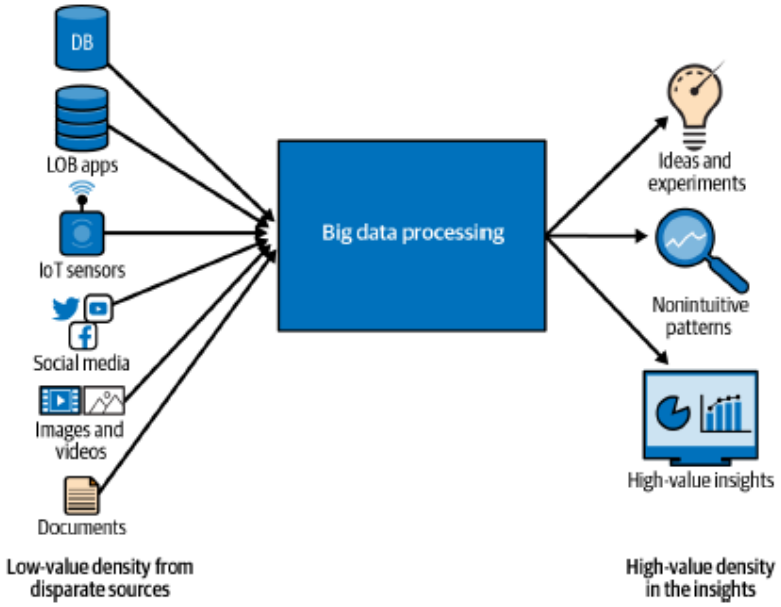


Technische vs. fachliche Metadaten

- technische Metadaten
  - Quell-, Ziel-Systeme (technische Zugriffsoptionen etc.)
  - Warehouse-Administration (Datenaktualisierung, -archivierung, Optimierung)
    - Job-Ausführungsstatus, Auslastungsstatitiken, …
  - Schemata: Datenbank-Schemata, Dateiformate
  - Datenabhängigkeiten: (technische) Mappings zur Datentransformation
    - operationale Systeme <-> Data Warehouse<->Data Marts<-> Analysetools (Queries, etc.)
- fachliche Metadaten (Business-Metadaten)
  - Informationsmodelle, konzeptuelle Datenmodelle
  - Unternehmens-/Branchen-spezifische Vokabulare (Business terms)
  - Mapping Business Terms <->DWH-Elemente (Dimensionsattribute, Fakten)
  - Nutzermerkmale (Rollen, Interessensgebieten …)
  - Geschäftsbeschreibung von Kennzahlen (Key Performance Indicators), Queries, Reports
  - Datenqualität
    - Herkunft (lineage): aus welchen Quellen stammen die Daten? Besitzer?
    - Richtigkeit (accuracy): welche Transformation wurden angewendet?
    - Aktualität (timeliness): wann war der letzte Aktualisierungsvorgang?

Data Warehouse vs Big Data

- Data Warehouse
  - Analyse strukturierter Daten auf Basis eines Datenbankschemas
  - Schema first / Schema on Write (Speichern von Daten muss Schema befolgen)
- Big-Data: zusätzliche Datenarten mit weiteren Anforderungen
  - un-/teilstrukturierte Daten sowie Datenströme
  - oft schemalose Speicherung der Daten (zB in Data Lake) bzw. Schema on Read für Analysen



Serra: Deciphering data architectures, 2024

Stufen der Datenanalyse

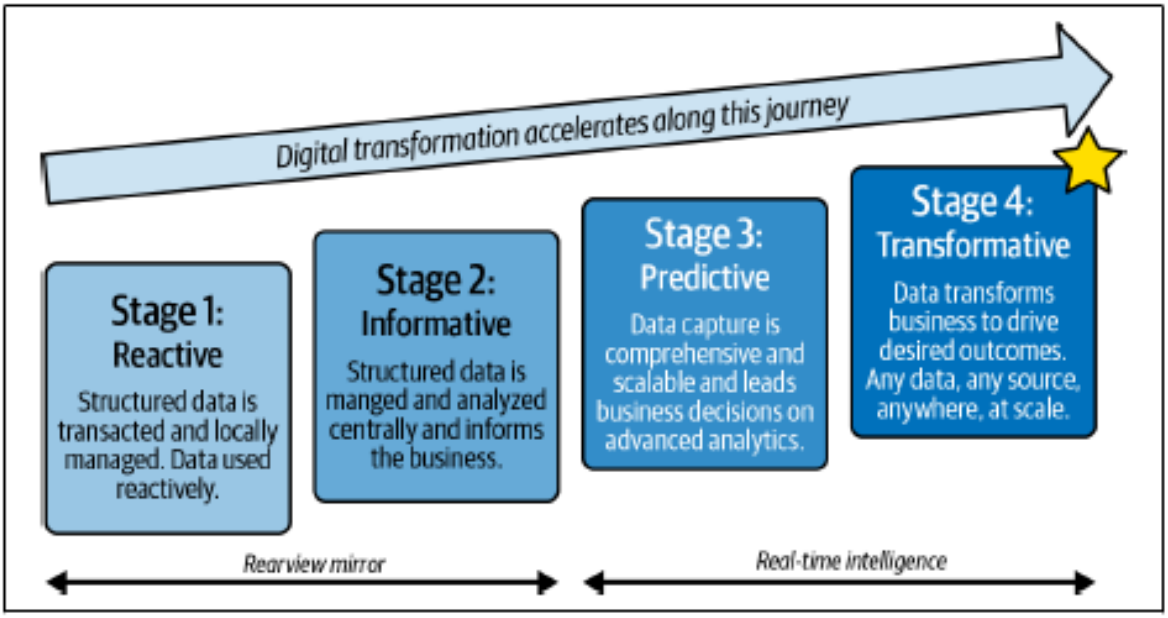
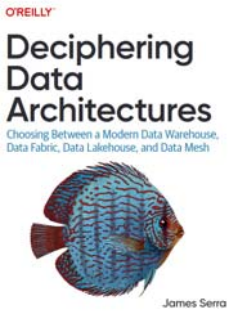


Figure 1-3. Enterprise data maturity stages

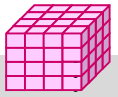
Serra: Deciphering data architectures, 2024

- Unternehmen sind derzeit meist auf Stufe 2 mit Data Warehouse, insbesondere für On-Premise-Datenhaltung
- Stufen 3 und 4: cloud-basierte Datenverwaltung mit Einsatz von Machine Learning

Datenarchitekturen (Serra 2024)

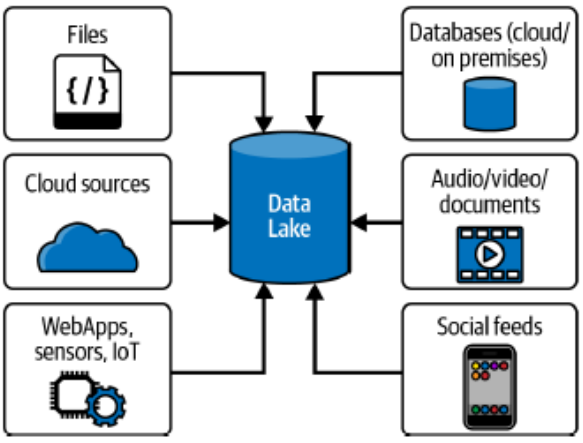


Characteristic	Relational data warehouse	Data lake	Modern data warehouse	Data fabric	Data lakehouse	Data mesh
Year introduced	1984	2010	2011	2016	2020	2019
Centralized/ decentralized	Centralized	Centralized	Centralized	Centralized	Centralized	Decentralized
Storage type	Relational	Object	Relational and object	Relational and object	Object	Domain-specific
Schema type	Schema-on-write	Schema-on-read	Schema-on-read and schema-on-write	Schema-on-read and schema-on-write	Schema-on-read	Domain-specific
Data security	High	Low to medium	Medium to high	High	Medium	Domain-specific
Data latency	Low	High	Low to high	Low to high	Medium to high	Domain-specific
Time to value	Medium	Low	Low	Low	Low	High
Total cost of solution	High	Low	Medium	Medium to high	Low to medium	High
Supported use cases	Low	Low to medium	Medium	Medium to high	High	High
Difficulty of development	Low	Medium	Medium	Medium	Medium to high	High
Maturity of technology	High	Medium	Medium to high	Medium to high	Medium to high	Low
Company skill set needed	Low	Low to medium	Medium	Medium to high	Medium to high	High

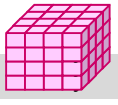


Data Lakes

- Speicherung heterogener Daten ohne gemeinsames Schema
  - Schema on Read
- initiale Nutzung v.a. für Hadoop File System
- Nutzung der Daten problematisch und aufwändig, z.B. mit Frameworks wie MapReduce, Apache Spark, Jupiter Notebooks
- fehlende Datenintegration
- nützlich als Teil einer umfassenderen Datenarchitektur, v.a. für Staging / Datenaufbereitung

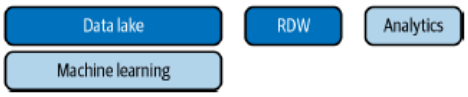


Serra: Deciphering data architectures, 2024

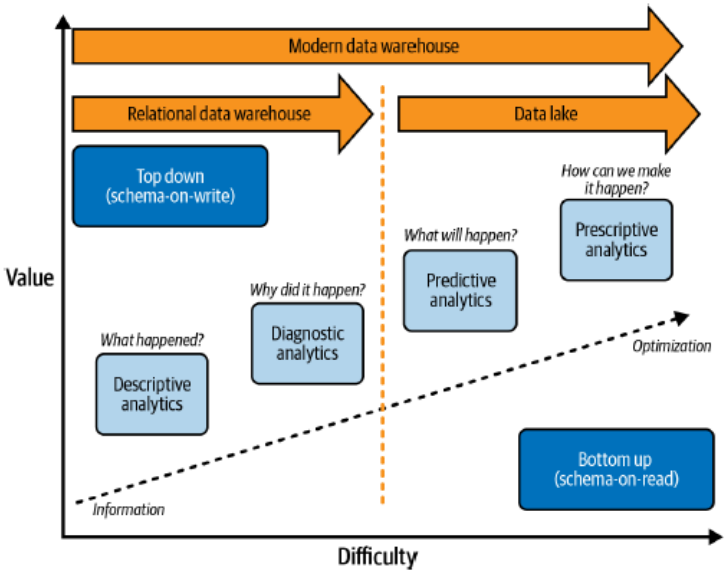


Kombination Warehouse / Data Lake

- „Modern Data Warehouse“ nach Serra 2024, v.a. für weitergehende Analysen



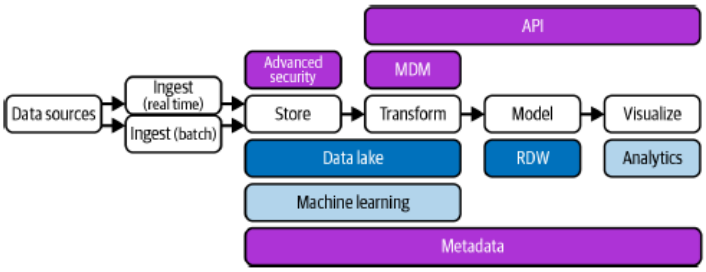
- oft kein monolithisches System, sondern Kombination verschiedener Tools
  - unterschiedliche Pipelines zur Datenaufbereitung je nach Daten- und Analyseart



Data Fabric

- Weiterentwicklung der hybriden Data Lake/Warehouse-Architektur
  um nahezu beliebige Datenquellen und Zugriffsmöglichkeiten „fabrikartig“ zu unterstützen

- keine allgemein akzeptierte Definition
- stärkere Unterstützung für
  - einheitliche verwaltete Metadaten
  - Masterdaten-Management (MDM)
  - Real-time data ingestion
  - virtuelle Datenanbindung (statt Datenkopien)

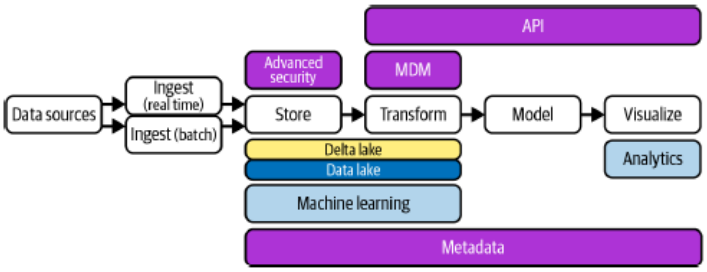


Serra: Deciphering data architectures, 2024

- Zukunft: Verwendung von Wissensgraphen zur semantischen Beschreibung und Verknüpfung unterschiedlicher Datenobjekte

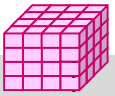
Data Lakehouse

- Entwicklung von Fa. Databricks (2020)
- engere Kombination von Data Lake / Data Warehouse
  - alle Daten im Lake ohne zusätzliches Warehouse für strukturierte Daten
  - dafür transaktionale Zusatzkomponente im Lake, um Inhalte relational nutzen zu können (SQL-Zugang, CRUD Operationen …)

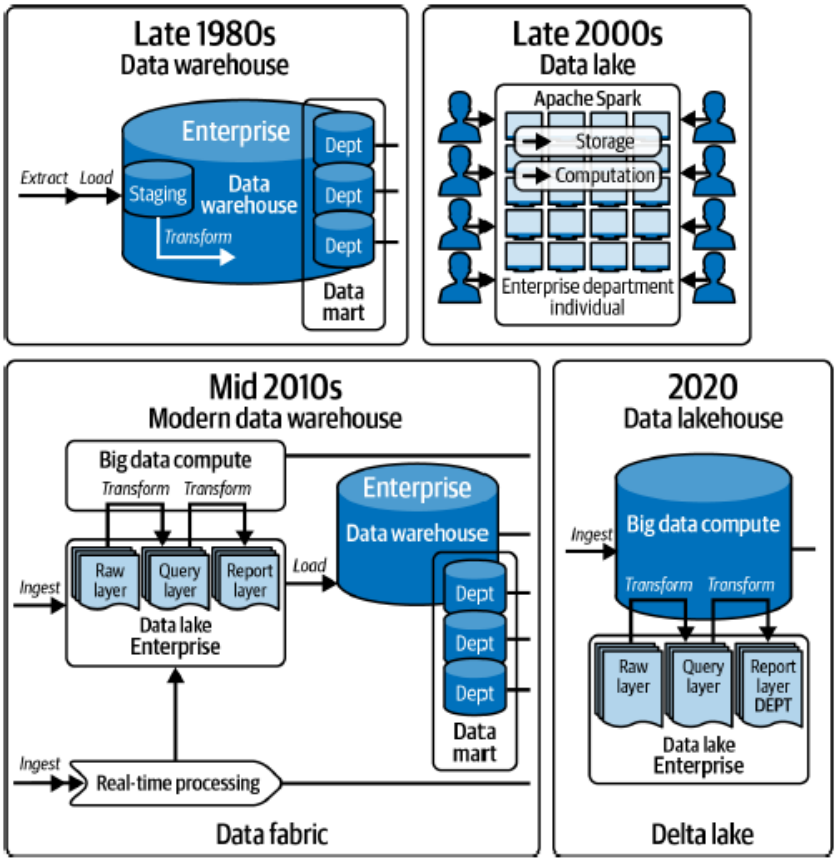


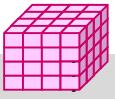
Serra: Deciphering data architectures, 2024

- alternative Realisierungen
  - Delta Lake
  - Apache Iceberg
  - Apache Hudi
- Tradeoffs
  - Anfragen typischerweise langsamer als mit DWH
  - stark limitierte semantische Datenintegration (Schema on read, i.a. Views über mehrere Dateien)
  - bessere Skalierbarkeit auf viele Datenquellen
  - Unterstützung heterogener Datenarten



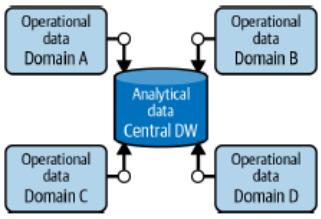
Historische Entwicklung

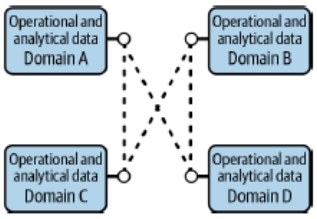




Data Mesh

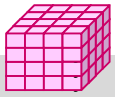
- Konzept einer dezentralen Architektur mit mehreren domänen-spezifischen Datensammlungen innerhalb eines Unternehmens





Serra: Deciphering data architectures, 2024

- Probleme zentraler Ansätze sollen umgangen werden
  - Skalierbarkeit, Abhängigkeit zu zentraler IT …
  - besseres Self-Serving für Analysten
- verschiedene Varianten
- kontroverse Einschätzungen



Zusammenfassung

- wesentliche Komponenten der Referenzarchitektur
  - ETL-Komponenten inklusive Monitoring und Scheduling
  - Arbeitsbereich (Staging Area)
  - Data Warehouse und Data Marts / Cubes
  - Metadaten-Verwaltung
- Extraktionsansätze:
  - Snapshot
  - DBMS-Verfahren: CDC, Timestamps, Trigger
- abhängige vs. unabhängige Data Marts
- ODS (Online Data Store): Unterstützung operativer Anwendungen auf integrierten Daten
- Ko-Existenz DWH und Big-Data-Technologien / Data Lakes
  - flexible Unterstützung unstrukturierter und hoch-dynamischer Daten
  - parallele ETL-Verarbeitung und Datenanalyse u.a. mit Data Mining / Machine Learning
  - Varianten u.a. Data Fabric , Data Lakehouse