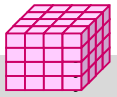
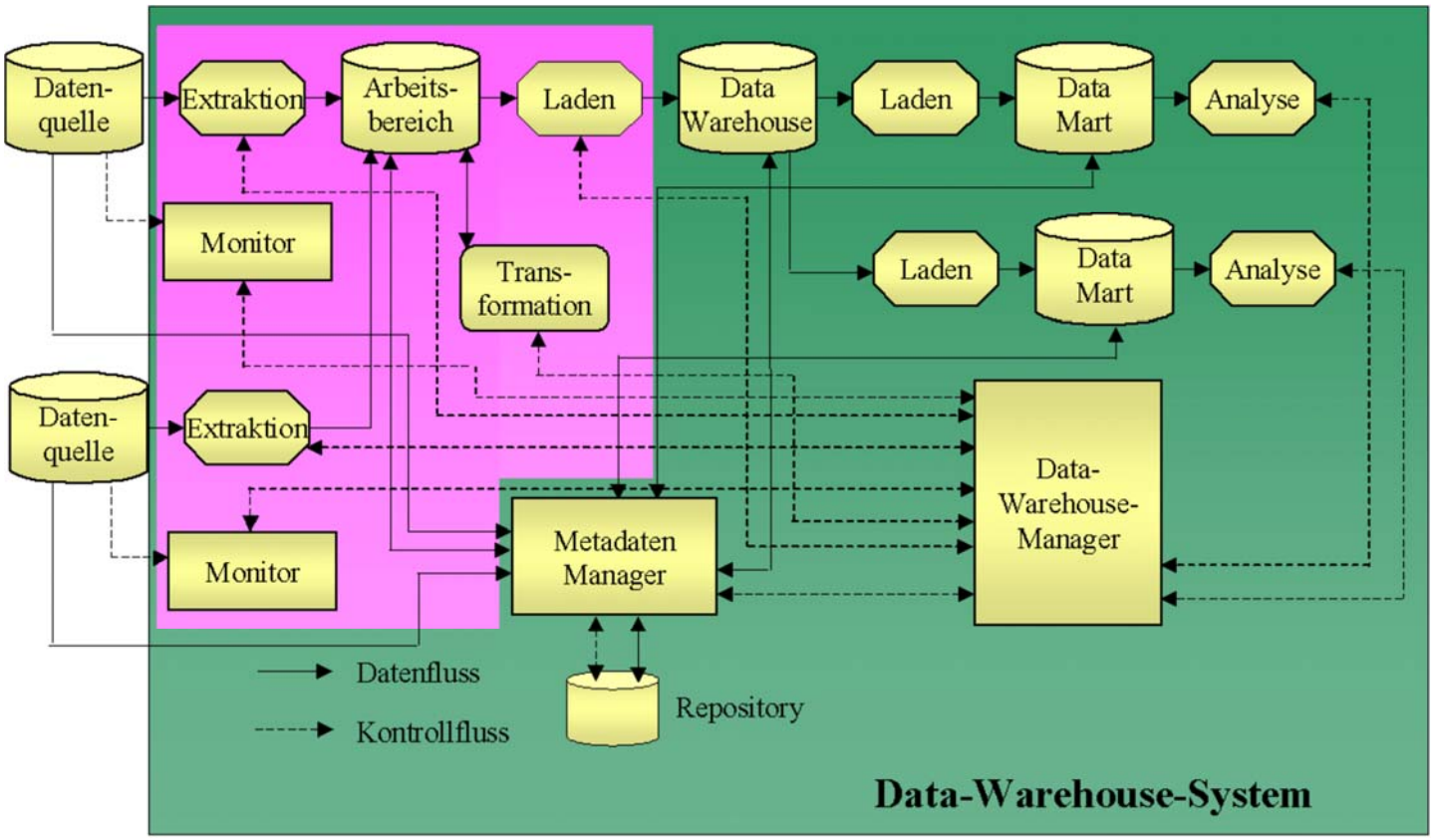
2. Architektur von Data Warehouse-Systemen

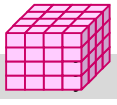
- Referenzarchitektur
  - Scheduler, Datenquellen, Datenextraktion, Transformation und Laden
- Abhängige vs. unabhängige Data Marts
- Operational Data Store (ODS)
- Metadatenverwaltung
  - technische vs. fachliche Metadaten
- Data Warehouse Appliances
- DWH und Big Data / Data Lake



DW-Referenzarchitektur

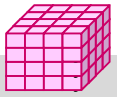


Quelle: Bauer/Günzel



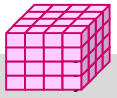
Phasen des Data Warehousing

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



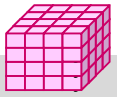
Datenquellen

- Lieferanten der Daten (gehören nicht direkt zum DW)
- Merkmale
  - intern (Unternehmen) oder extern (z.B. Internet)
  - 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



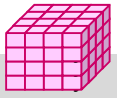
Data-Warehouse-Manager/Scheduler

- 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

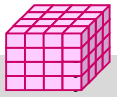
- Monitore: Entdeckung von Änderungen in einer Datenquelle
  - interne Datenquellen: aktive Mechanismen
  - externe Datenquellen: Polling / periodische Abfragen
- Extraktionskomponenten: Übertragung von Daten aus Quellen in Arbeitsbereich
  - periodisch
  - auf explizites Verlangen durch Administrator
  - ereignisgesteuert (z.B. sofort bei jeder Änderung oder bei Erreichen einer definierten Anzahl von Änderungen)
- Randbedingungen
  - Autonomie der Quellsysteme ist zu wahren
  - hohe Performance-Anforderungen bei großen Datenmengen
- unterschiedliche Funktionalität der Quellsysteme
  - Nutzung von Standardschnittstellen (z.B. ODBC/JDBC) oder Eigenentwicklung
  - Nutzung spezieller Funktionalität, z.B. von DBS-Quellen



Datenextraktion: Strategien

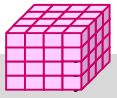
- Snapshots: periodisches Kopieren des Datenbestandes in Datei
- Trigger
  - Auslösen von Triggern bei Datenänderungen und Kopieren der geänderten Tupel
- Log-basiert
  - Analyse von Transaktions-Log-Dateien der DBMS zur Erkennung von Änderungen
- Nutzung von DBMS-Replikationsmechanismen

	Autonomie	Performanz	Nutzbarkeit
Snapshot			
Log			
Trigger			
Replikation		realisierungs-abhängig	



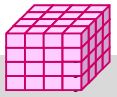
Datentransformation und Laden

- *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)



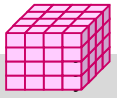
Data Warehouse

- 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



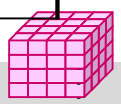
Data Marts

- 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

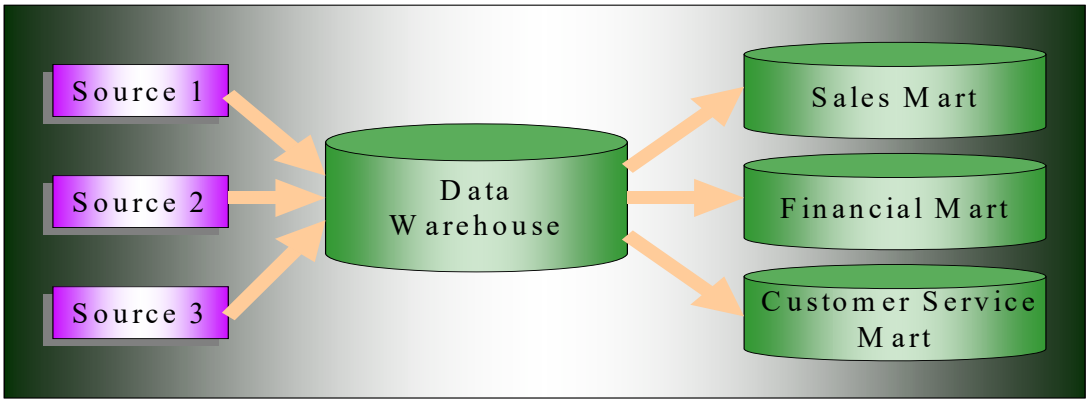


Data Warehouse vs. Data Mart

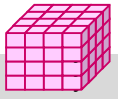
	Data Warehouse	Data Mart
Philosophie	anwendungsneutral	anwendungsbezogen
Adressat der Datenbereitstellung	Unternehmen	Abteilung
Datenmenge / Detaillierungsgrad	hoch	gering
Umfang historischer Daten	hoch	gering
Optimierungsziel	Datenmenge	Antwortzeiten
Anzahl	eins (wenige)	mehrere
Typische DB-Technologie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



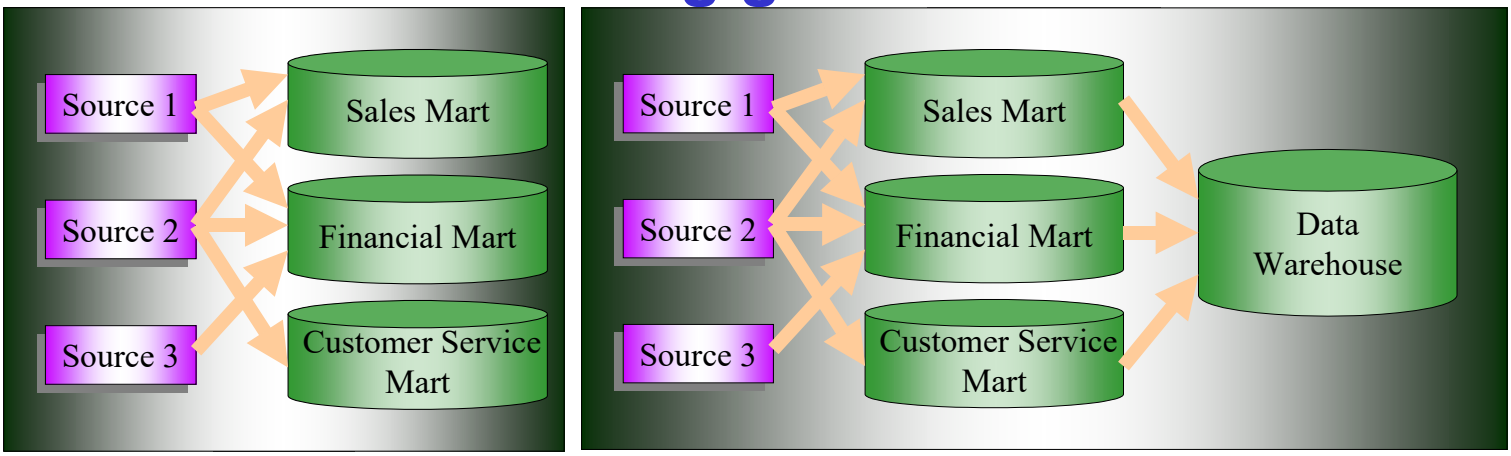
Abhängige Data Marts



- 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)



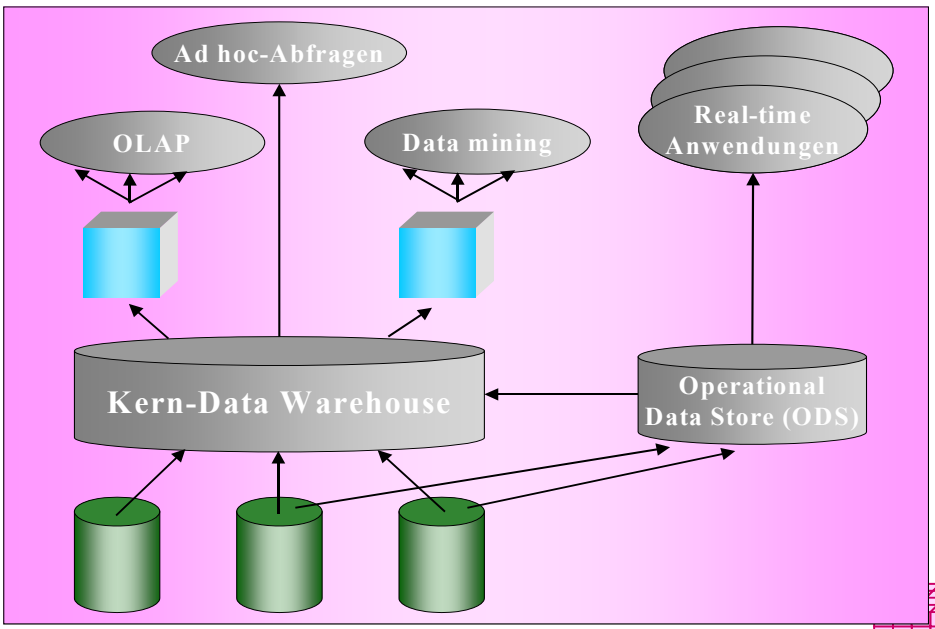
Unabhängige Data Marts



- 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

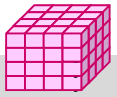
Operational Data Store (ODS)

- 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

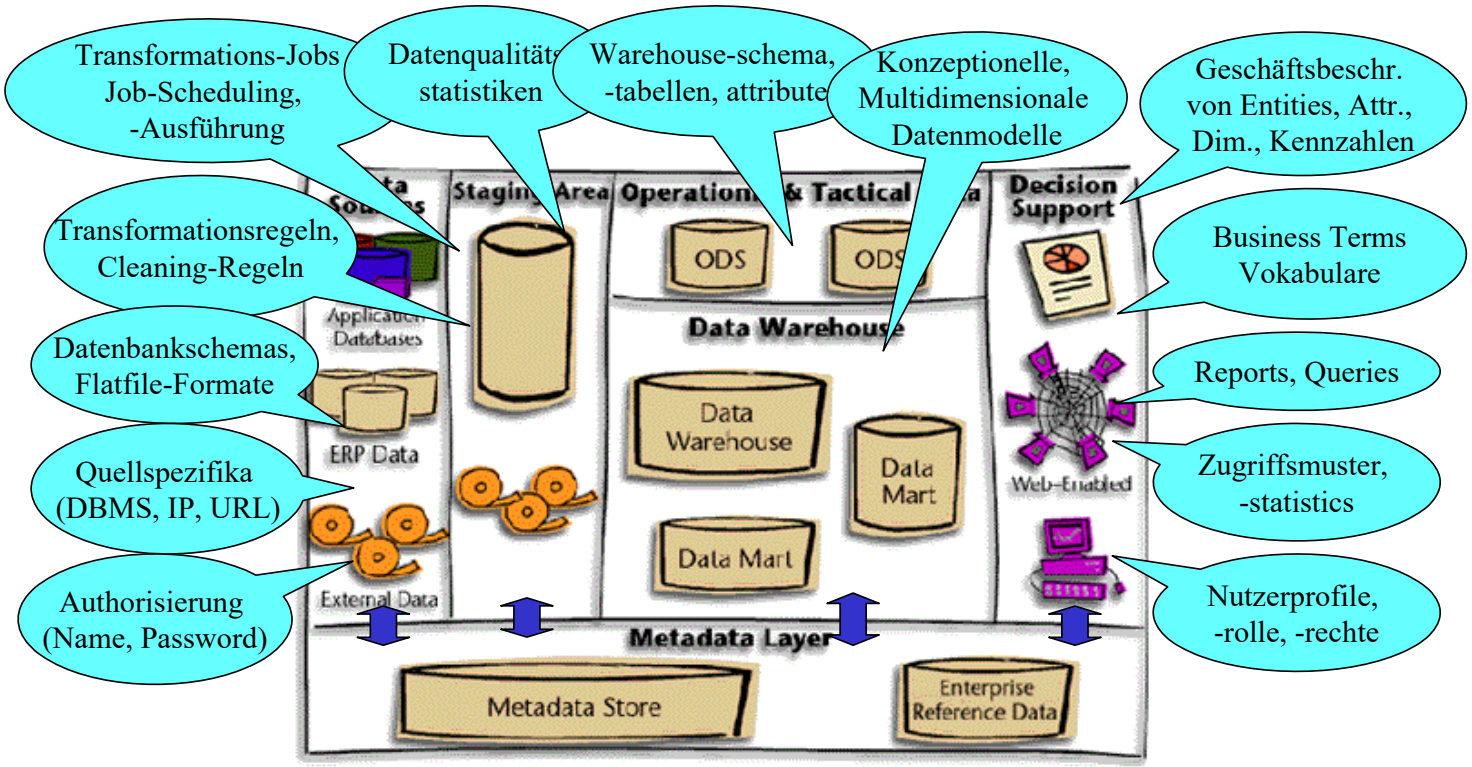


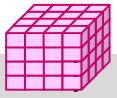
Metadaten-Verwaltung

- 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)



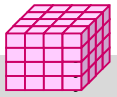
Metadaten im Data Warehouse-Kontext





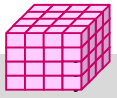
Technische vs. fachliche Metadaten

- 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?



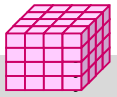
Data Warehouse Appliances

- vorkonfigurierte, komplette Data Warehouse-Installation
  - mehrere Server, Externspeicher, Software, etc.
  - Pricing nach Datenumfang (nicht Hardware)
- Vorteile/Ziele von Data Warehouse Appliances
  - hohe Leistung / Skalierbarkeit
  - hohe Verfügbarkeit (durch eingebaute Fehlerbehandlungsmechanismen)
  - geringer Administrationsaufwand
  - schnelle DWH-Realisierung/Nutzung
- Pioniere mit Spezial-Hardware: Teradata, Netezza (jetzt IBM)
- allgemeine Optimierungen im Rahmen von „NewSQL“-Systemen (z.B. SAP HANA, EMC Greenplum, MS SQL Server PDW (Parallel Data Warehouse), HP Vertica, EXASOL …)
  - Nutzung riesiger Hauptspeicher-Datenbanken (In-Memory Data Warehouses) bzw. SSD (statt Magnetplatten)
  - Column-Store-Techniken mit Datenkompression
  - Parallelverarbeitung



Pionier SAP HANA

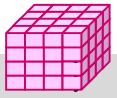
- dramatische Beschleunigung der DB-Verarbeitung
  - Vermeidung langsamer Plattenzugriffe
  - neue auf In-Memory-Verarbeitung zugeschnittene Datenstrukturen und Algorithmen
  - Vermeidung von Indexstrukturen, Cubes etc.
- gleichzeitige Unterstützung von OLTP + OLAP
  - Record Store und Column Store
  - hohe Datenaktualität
- Einschränkungen
  - ETL/Datenintegrationsaufgaben für Daten außerhalb der In-Memory-DB bleiben bestehen
  - geschlossene Umgebung
  - hohe Kosten



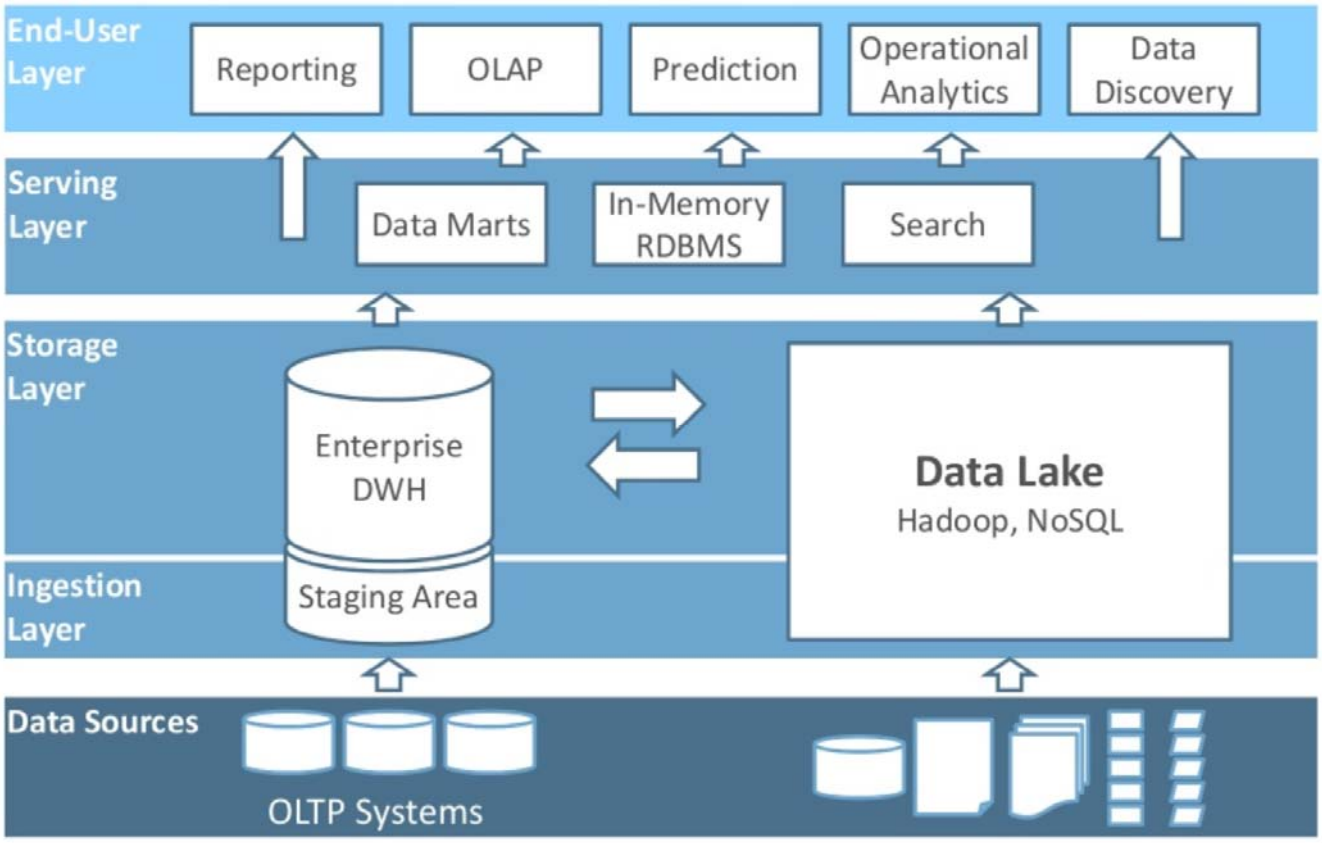
Data Warehouse vs Big Data

- Data Warehouse
  - Analyse strukturierter Daten auf Basis eines Datenbankschemas
  - Schema first / Schema on Write
- Big-Data-Technologien (z.B. Hadoop/Spark) komplementär nutzbar
  - Beschleunigung der ETL-Prozesse
  - Erfassung und Analyse unstrukturierter und hoch-dynamischer Daten (Texte, Bilder, Datenströme …)
  - Erfassung zunächst in schemalosem Data Lake
  - Datenaufbereitung bei Bedarf („Schema on Read“)
  - Analyse mit Frameworks wie Apache Spark/Flink, Textsuche etc.
  - Datenaustausch Data Lake – Data Warehouse möglich

www.pmone.com



BI-Architektur mit DWH und Data Lake



Quelle: J. Albrecht

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-Replikationsverfahren, Trigger, Log-Transfer
- abhängige vs. unabhängige Data Marts
- ODS (Online Data Store): Unterstützung operativer Anwendungen auf integrierten Daten
- zunehmender Einsatz voroptimierter DWH Appliances mit In-Memory-Technologie, Column Stores, …
- Ko-Existenz DWH und Big-Data-Technologien / Data Lakes
  - flexible Unterstützung unstrukturierter und hoch-dynamischer Daten
  - parallele ETL-Verarbeitung und Datenanalyse