Betriebliche Informationssysteme

L10 – Systemintegration, Datenintegration, Data Lineage

L10 – Systemintegration, Datenintegration, Data Lineage

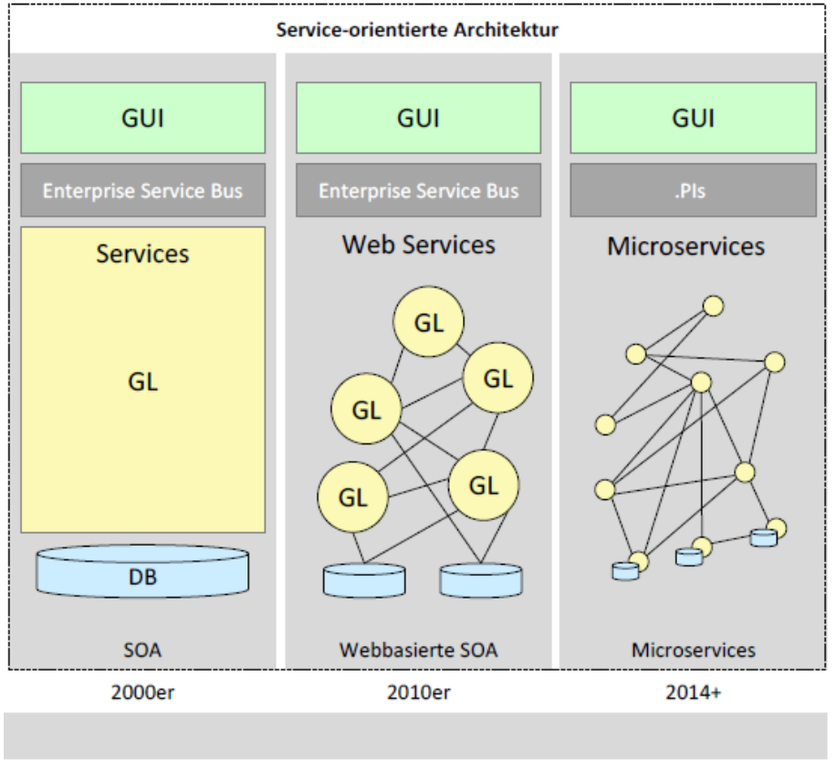
**„Wie werden Systeme miteinander verbunden (integriert)?
Welchen Zusammenhang hat dies mit der Datenintegration?“**

- **Verstehen, warum Systemintegration angewendet wird.**
- **Grundverständnis der Datenintegration. Zusammenhang mit Datenqualität & Datenbankmanagementsystemen erläutern können.**
- **Den Begriff „Data Lineage“ kennen.**

Repetition: Systemarchitekturen

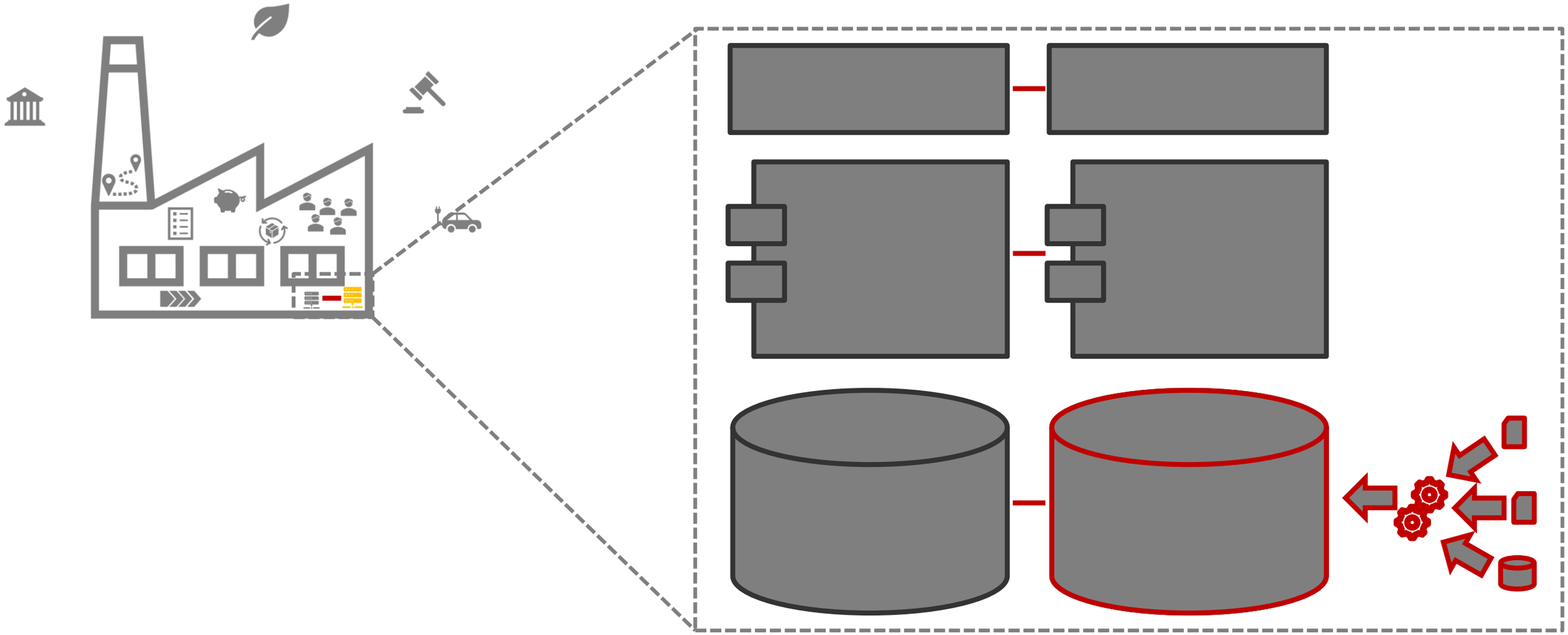
Die Aufteilung von Komponenten im SOA Paradigma zog den Bedarf einer Kommunikation zwischen Diensten mit sich – einem „Vermittler“.

Dafür gibt es Architekturstile. Im Beispiel hier der Enterprise Service Bus (ESB) oder Published Interfaces.



Quelle: Schubert, Winkelmann (2023). Betriebswirtschaftliche Anwendungssysteme

Big Picture



Herausforderungen in der Systemintegration
Architekturstile

Begriffe für die Beispiele auf den Folgefolien

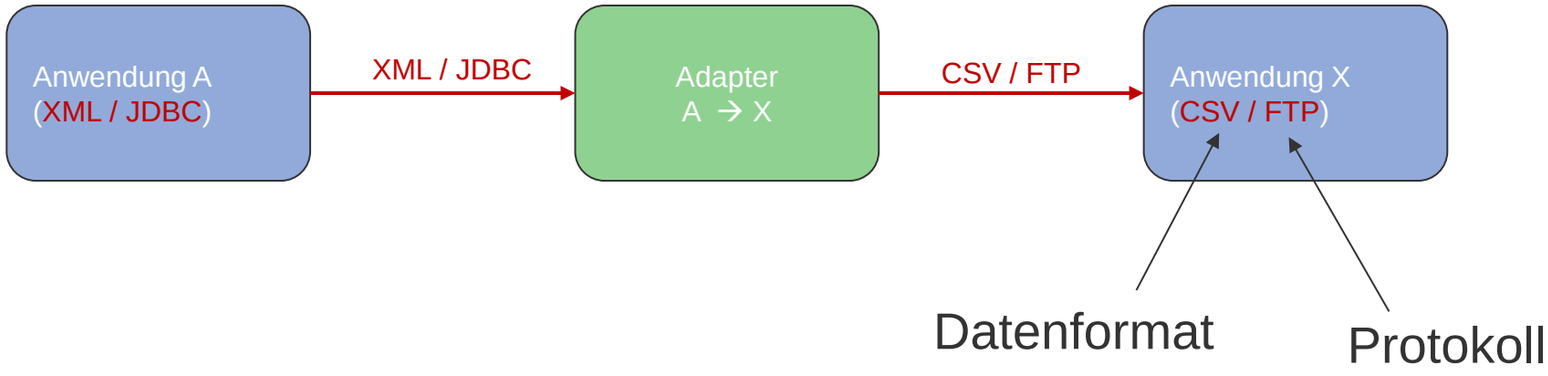
Datenformat	Erläuterung
CSV (Comma-Separated Values)	ist ein einfaches, textbasiertes Dateiformat zur Speicherung von tabellarischen Daten, bei dem einzelne Werte durch Kommas getrennt werden.
XML (Extensible Markup Language)	ist eine Auszeichnungsprache (Markup Language), die zur Definition und Strukturierung von Daten dient (mittels Tags/Markups wie <...>) und durch ihre flexible, hierarchische Struktur besticht.
JSON (JavaScript Object Notation)	ist ein leichtgewichtiges, textbasiertes Datenformat, das zur strukturierten Darstellung von Informationen dient und sowohl für Menschen als auch Maschinen gut lesbar ist.

Protokoll	Erläuterung
FTP (File Transfer Protocol) bzw. SFTP oder FTPS	ist ein Netzwerkprotokoll zum Übertragen von Dateien zwischen Rechnern über TCP/IP-Verbindungen, das keine standardmässige Verschlüsselung bietet.
HTTP(S) (Hypertext Transfer Protocol)	ist ein Anwendungsprotokoll, das den Austausch von Informationen im World Wide Web ermöglicht, indem es die Kommunikation zwischen Client und Server regelt.
JDBC (Java Database Connectivity)	ist eine Programmierschnittstelle, die Java-Anwendungen den Zugriff und die Interaktion mit relationalen Datenbanken ermöglicht.

Erläuterung Systemintegration (I)

Anwendung A hat Daten, die Anwendung X zur Verfügung gestellt werden müssen. Weder die Betreuer von Anwendung A noch die von Anwendung X haben ein Interesse daran, ihre Anwendung anzupassen.

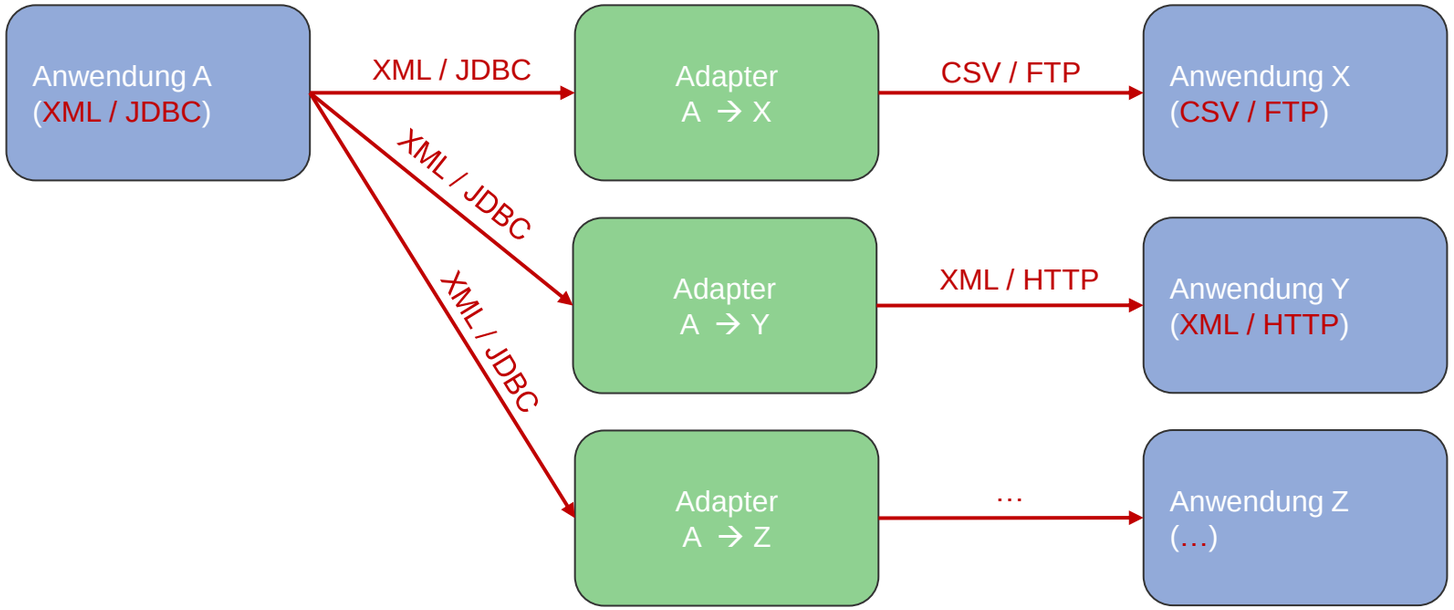
Dieses Problem lässt sich zunächst nach einem einfachen Muster lösen - einer „Point-to-Point“ Kommunikation.



Quelle: Basierend auf https://www.codecentric.de/wissens-hub/blog/serie-integrationsarchitektur-teil-1

Erläuterung Systemintegration (II)

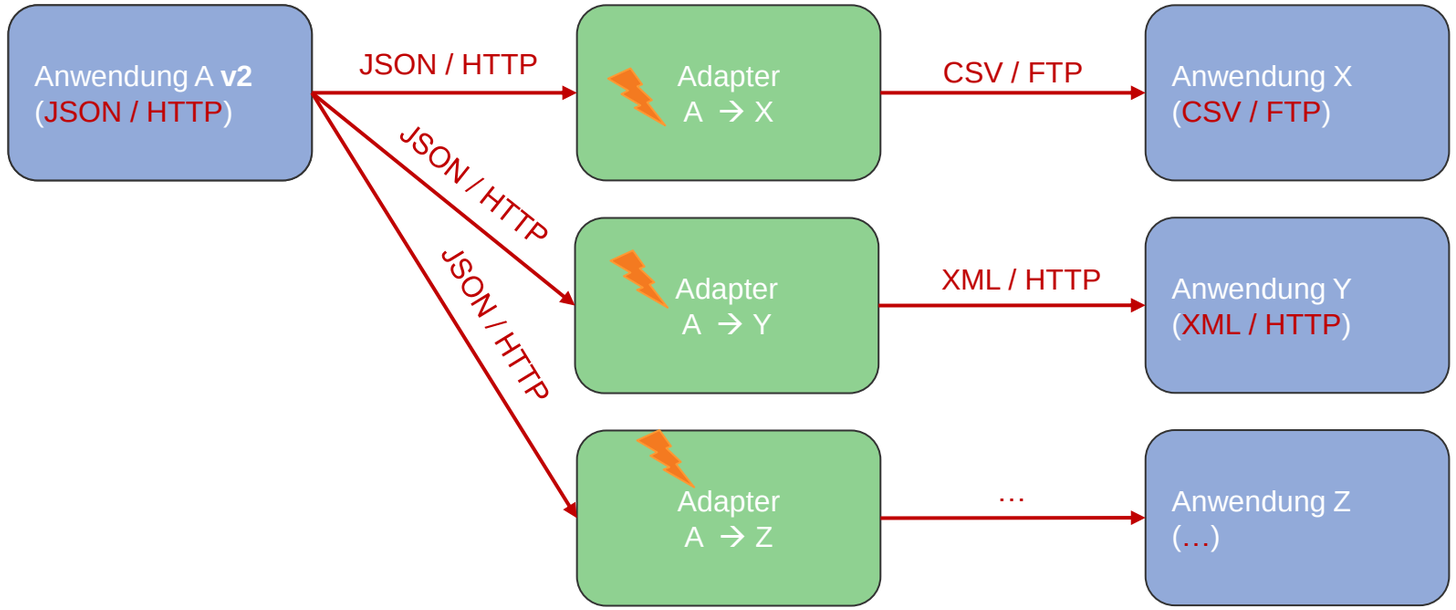
Angenommen, die Daten von Anwendung A werden von zwei weiteren Anwendungen Y und Z benötigt, werden entsprechend weitere Adapter hinzugefügt:



Quelle: Basierend auf https://www.codecentric.de/wissens-hub/blog/serie-integrationsarchitektur-teil-1

Erläuterung Systemintegration (III)

Im Verlaufe der Zeit wird Anwendung A jetzt durch ein neues System (v2) abgelöst. Statt XML/JDBC werden die Daten jetzt als JSON-Objekt über HTTP bereitgestellt.

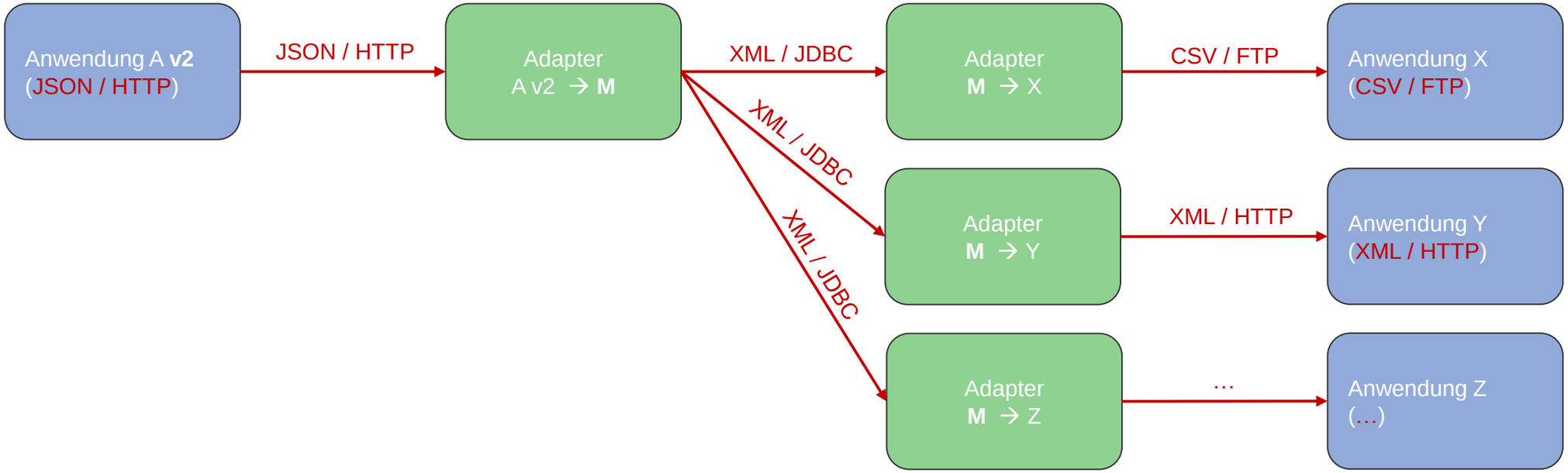


Quelle: Basierend auf https://www.codecentric.de/wissens-hub/blog/serie-integrationsarchitektur-teil-1

Erläuterung Systemintegration (IV)

Nun könnten alle drei Schnittstellen angepasst werden, oder es wird eine gemeinsame Schnittstelle eingeführt. Hier M benannt.

Dies greift in das Prinzip der *losen Kopplung*.



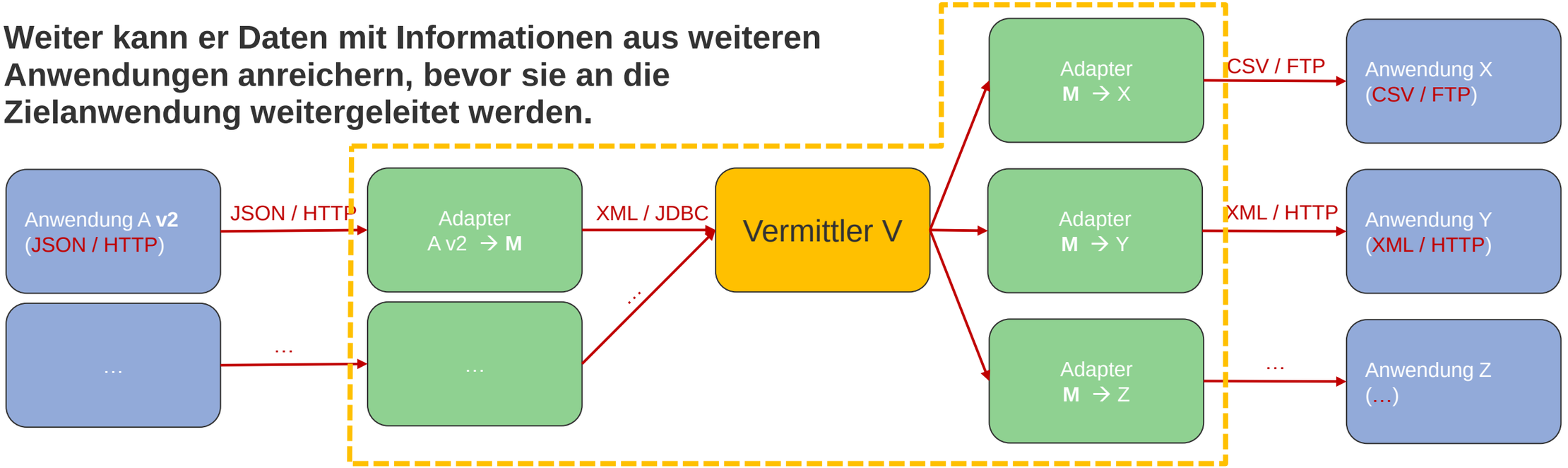
Quelle: Basierend auf https://www.codecentric.de/wissens-hub/blog/serie-integrationsarchitektur-teil-1

Erläuterung Systemintegration (V)

In vielen Fällen ist es nicht erforderlich, dass die Quell-Anwendung darüber entscheidet, an welche Zielanwendung die Daten weitergeleitet werden. In diesen Fällen kann die Einführung eines Vermittlers von Vorteil sein.

Der Vermittler übernimmt somit das Routing und i.d.R. auch gleich die Transformation.

Weiter kann er Daten mit Informationen aus weiteren Anwendungen anreichern, bevor sie an die Zielanwendung weitergeleitet werden.

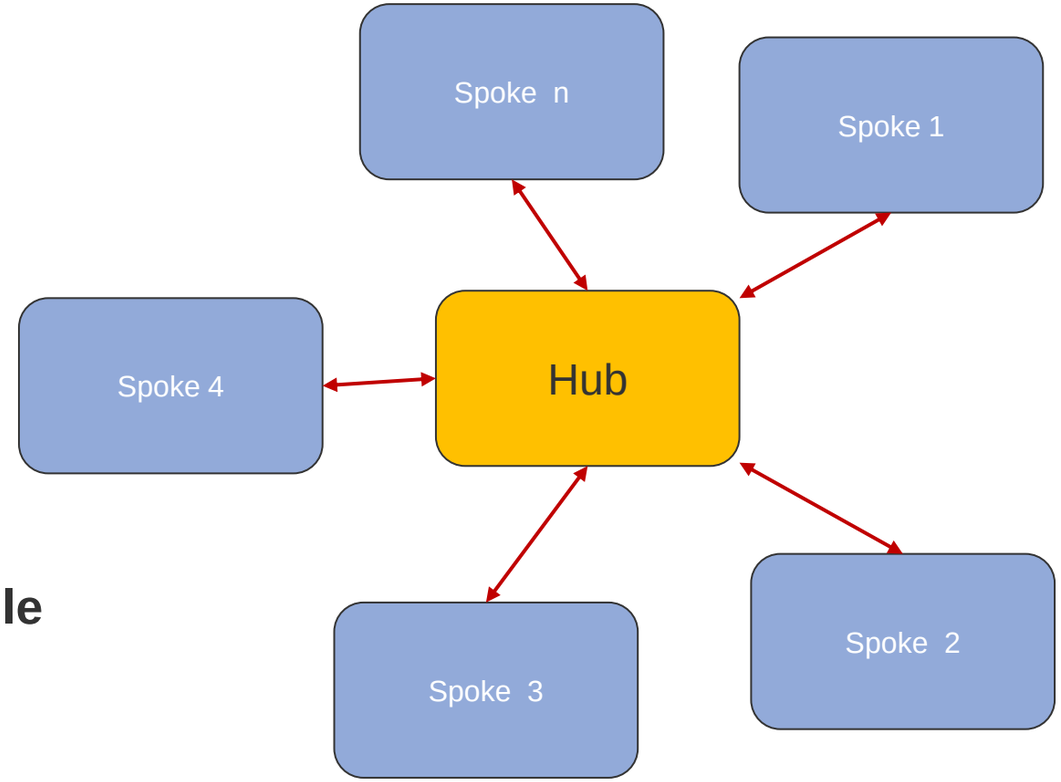


Architekturstil: Hub-Spoke-Architektur

Für Vermittler gibt es in unterschiedliche Architekturstile (neben der Point-to-Point)

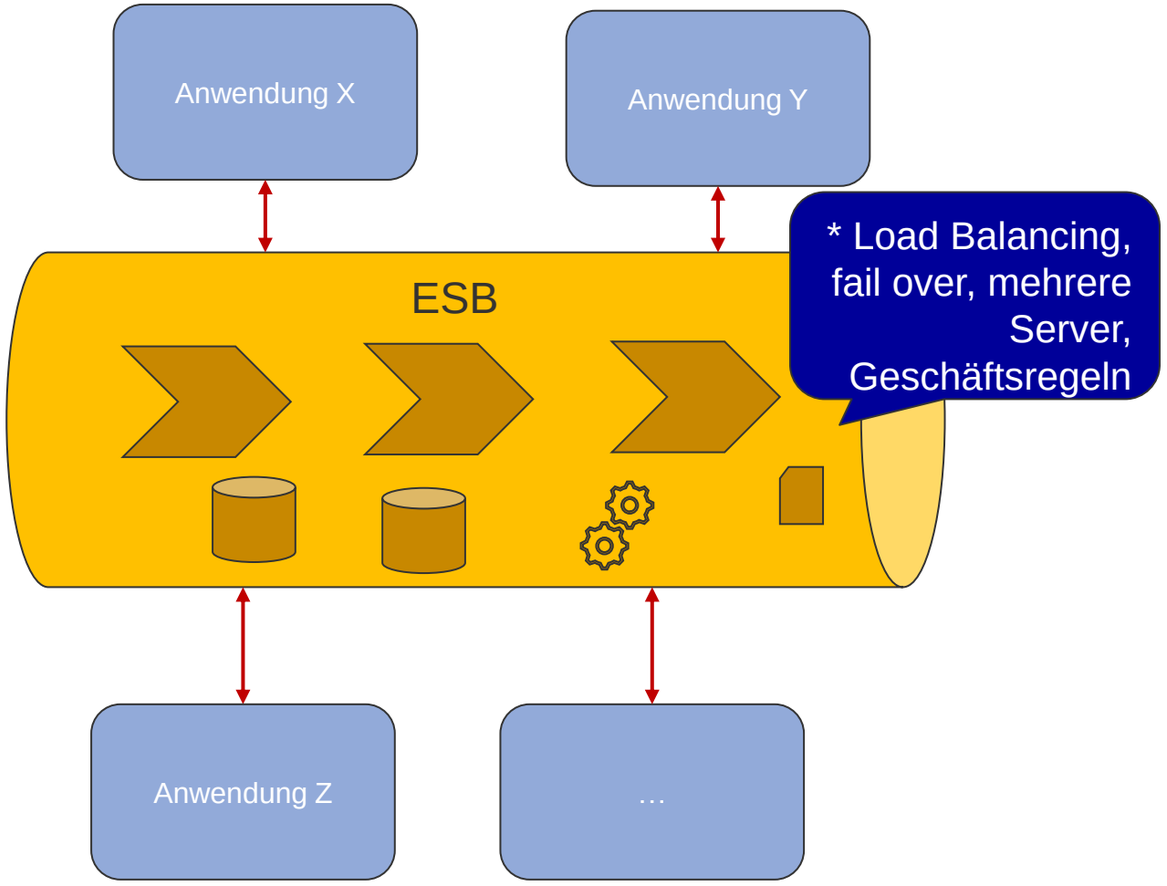
Hub-Spoke-Architektur:

- **Zentraler Message Broker (Hub) verbindet alle integrierten Systeme (Spokes).**
- **Der Hub übernimmt in der Regel auch die Transformation der Datenformate**
- **Wichtigster Nachteil: Begrenzte Skalierbarkeit & Single point of failure.**
- **1990er und frühen 2000er Jahren**



Architekturstil: Enterprise Service Bus (ESB)

- **ESB dienen ebenso als «zentrale Drehscheibe», haben jedoch einige Vorteile.**
- **Sie sind von Grund aus dafür ausgerichtet, eine breite Palette von Nachrichtenformaten und Protokollen zu unterstützen,**
- **erlauben eine dezentrale Verarbeitung, sind leichter erweiterbar und sind mit mehr Konfigurations- und Anpassungsmöglichkeiten ausgestattet*.**
- **Wichtigste Nachteil: Komplex und schwer zu verwalten. Performance-Probleme.**
- **Mitte 2000er**

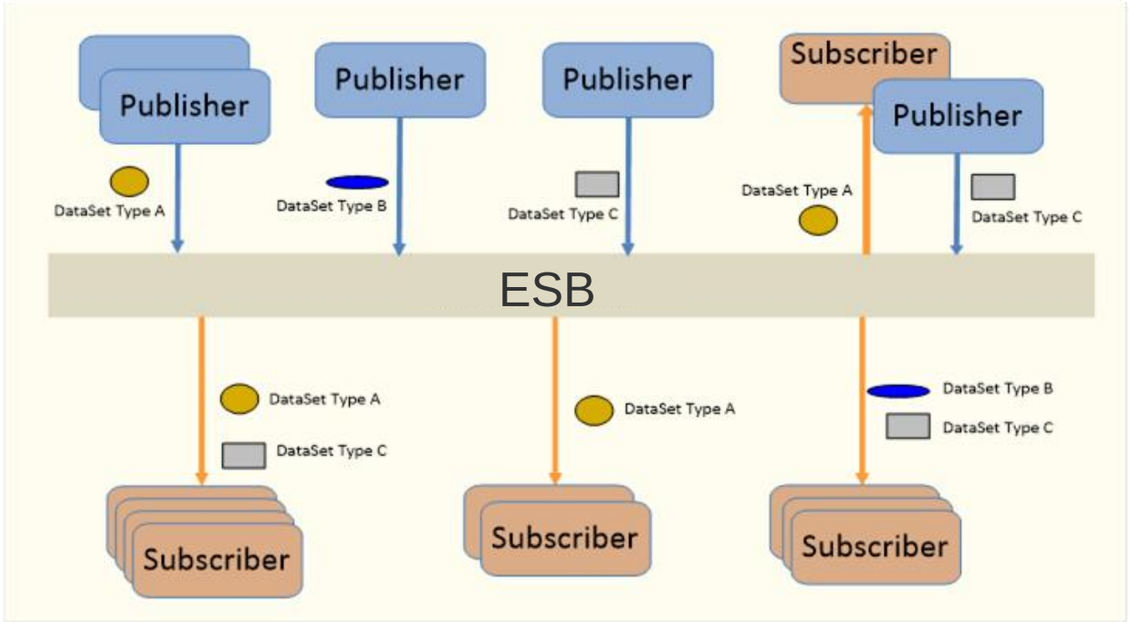


Architekturstil: Enterprise Service Bus (ESB)

Lose Koppelung

- **Unabhängigkeit**: Jeder Service kennt nur die definierten Schnittstellen (APIs) oder Nachrichtenformate, aber nicht die internen Details anderen Services.
- **Flexibilität**: So kann ein Service geändert oder aktualisiert werden, ohne dass dies zwangsläufig Auswirkungen auf andere Services hat.
- **Vermittlung über den Bus**: Der ESB übernimmt Funktionen wie Routing (Nachrichten an den richtigen Empfänger weiterzuleiten), Transformation (Umwandlung von Datenformaten oder Datenstrukturen) und Protokollanpassung (unterschiedliche Kommunikationsprotokolle miteinander zu verbinden).

Publish / Subscribe



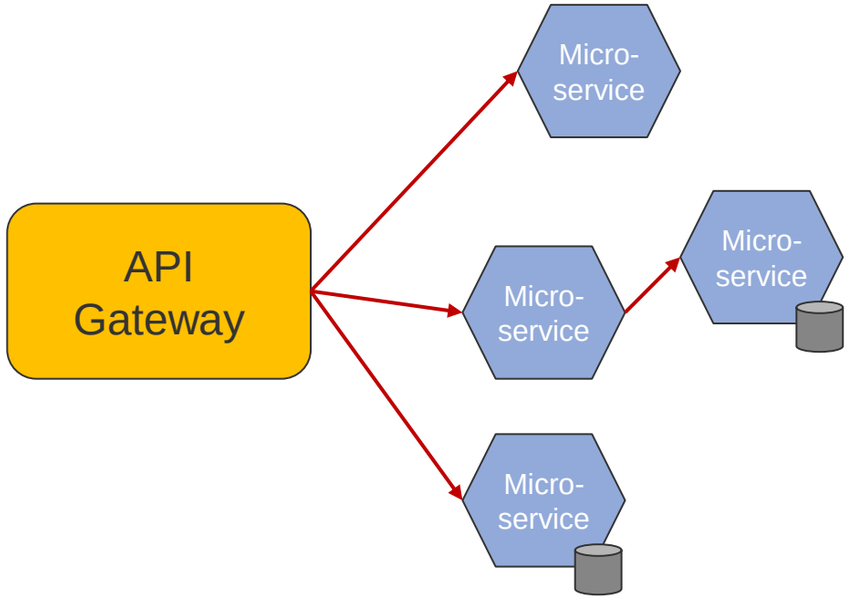
Quelle: https://reference.opcfoundation.org/Core/Part14/v104/docs/B.3

Architekturstil:
API-basierte Integration und Microservices-Architekturen

- **Mit der zunehmenden Verbreitung von *Web- und mobilen Anwendungen sowie Cloud-Services* wurde die Integration über APIs (Application Programming Interfaces) zunehmend wichtig.**

- **API-Gateways fungieren als zentrale Schnittstellen, die den Zugang zu Microservices steuern, absichern und überwachen.**

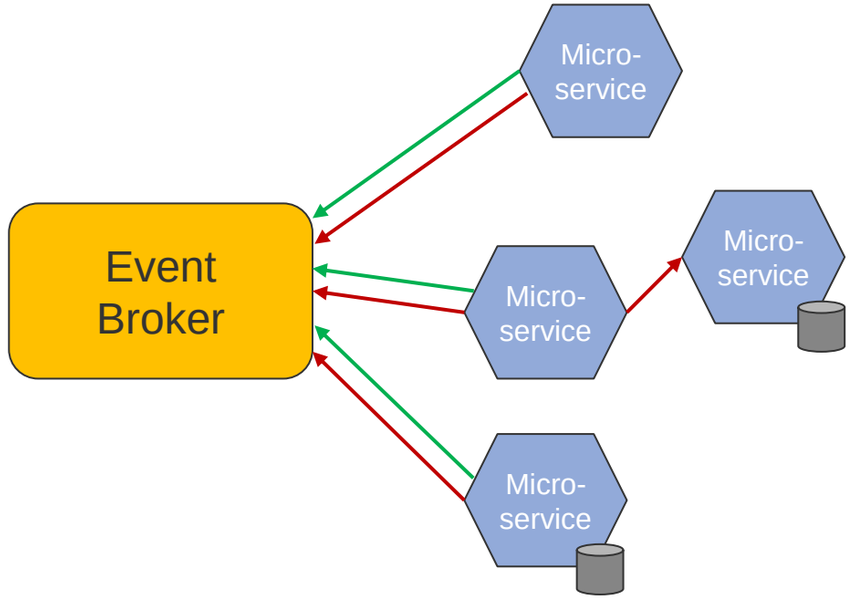
- **Ab den 2010er Jahren**



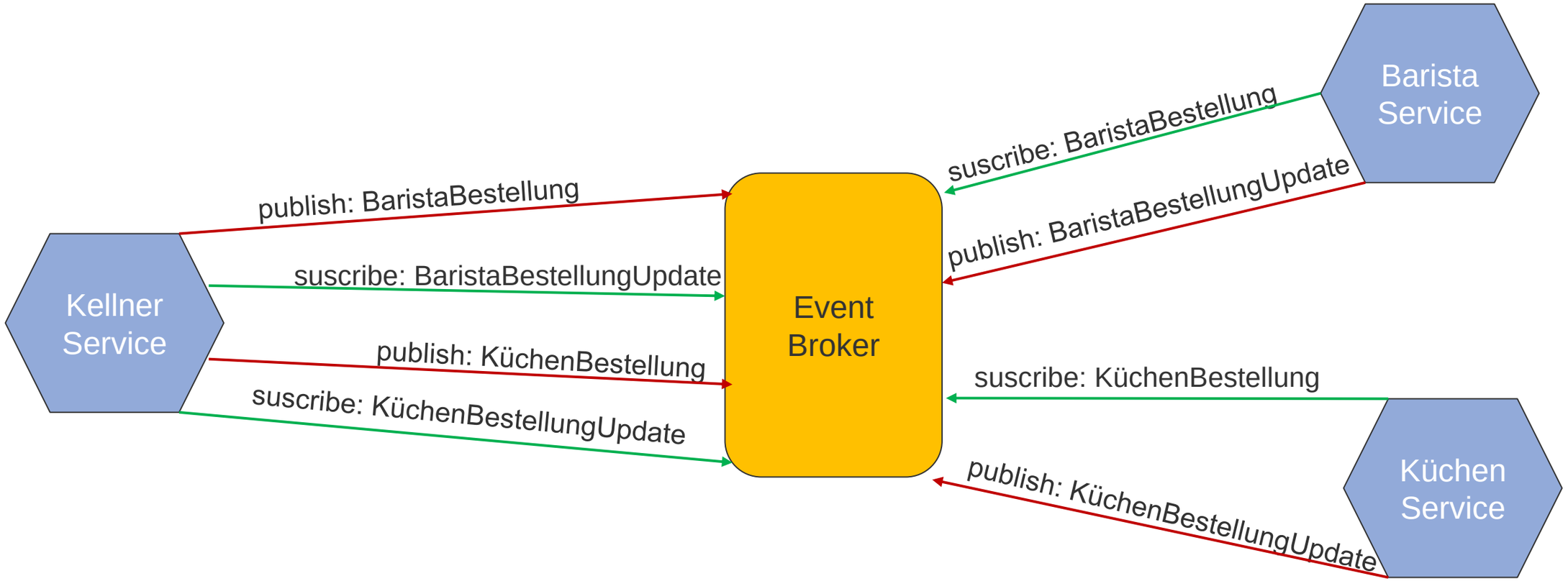
Event-Driven Architecture (EDA) und Event Streaming

EDA setzt darauf, dass Systeme auf Ereignisse (Events) reagieren und diese verarbeiten. Dabei kommen oft asynchrone Kommunikationsmuster wie Publish/Subscribe zum Einsatz (Beispiel: Apache Kafka).

- In der EDA erzeugen die Microservices Ereignisse als Reaktion auf Zustandsänderungen oder Aktionen.
- Andere Dienste konsumieren diese Ereignisse und ergreifen bei Bedarf Massnahmen.
- Der Producer und der Consumer sind lose gekoppelt und wissen nichts voneinander.



Beispiel EDA – Restaurant



Quelle: Basierend auf https://github.com/thangchung/go-coffeeshop. Further reading: https://talent500.com/blog/understanding-event-driven-architectures-in-devops/

Zusammenfassung Systemintegration

Folglich kann die Systemintegration wie folgt zusammengefasst werden

- In Unternehmen existiert eine Vielzahl unterschiedlicher Systeme, die unabhängig voneinander operieren und mit diversen Programmiersprachen, Datenformaten, & Protokollen entwickelt wurden.

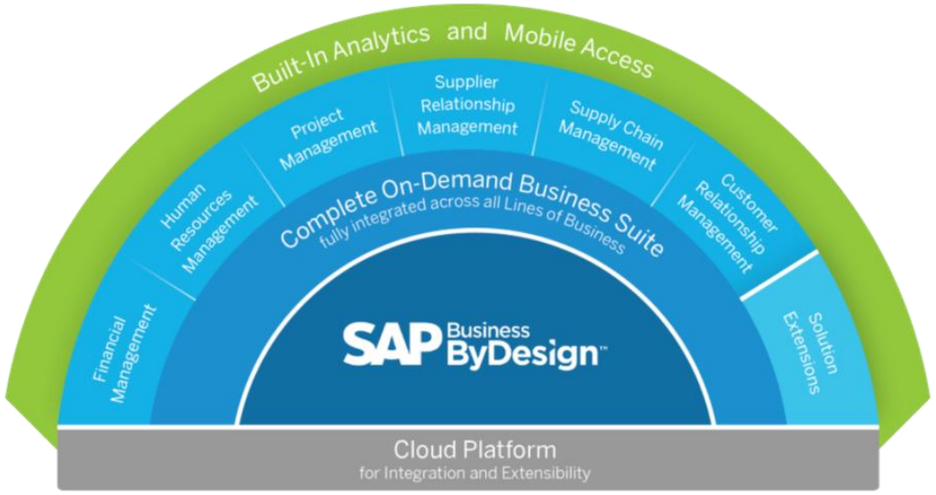
- Die Systemintegration ermöglicht einen reibungslosen Datenaustausch und die koordinierte Abwicklung von Geschäftsprozessen.

- Die Systemintegration fungiert daher als Dolmetscher /Vermittler zwischen den Elementen, welche sie verbindet.

Quellen: https://www.arcmedia.ch/de/blog/vorteile-systemintegration, https://blog.bismart.com/en/what-is-system-integration-problems-benefits, https://www.kaseya.com/blog/system-integration/

Repetition

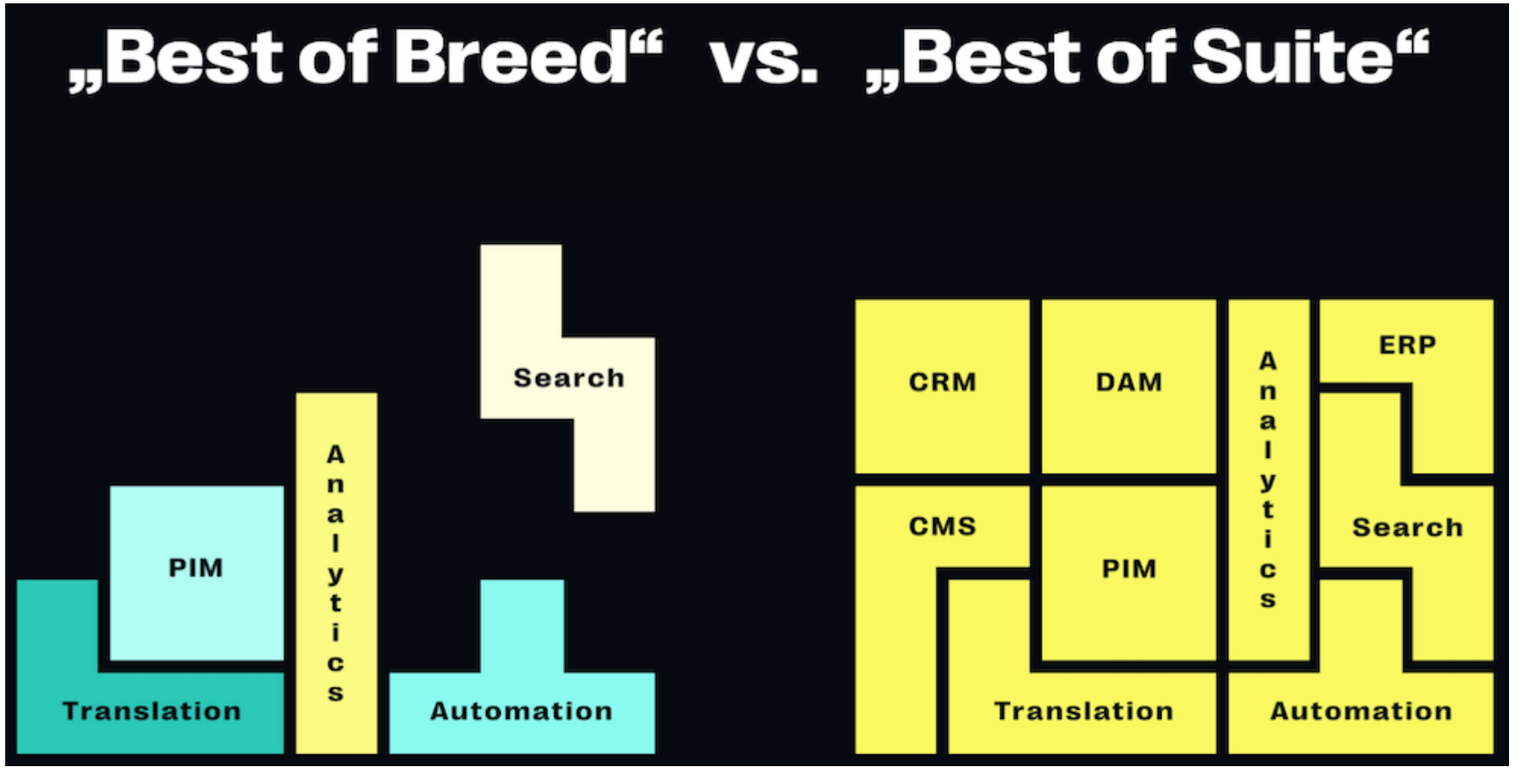
Der Begriff „**Best of Breed**“ bedeutet per Definition, dass ein Unternehmen, anstatt auf eine Komplettlösung von einem Anbieter zu setzen (**Best of Suite**), die Strategie verfolgt, für jeden Unternehmensbereich die optimale Software-Lösung zu integrieren.



Quelle: https://neteris.com/en/business-applications/sap-business-by-design/

Quelle: https://mindsquare.de/knowhow/sap-ecc/

Unterschiedlicher Integrationsaufwand der beiden Ansätze ist offensichtlich



Quelle: https://www.bitgrip.com/blog/best-of-breed-vs-best-of-suite-der-grosse-praxisguide

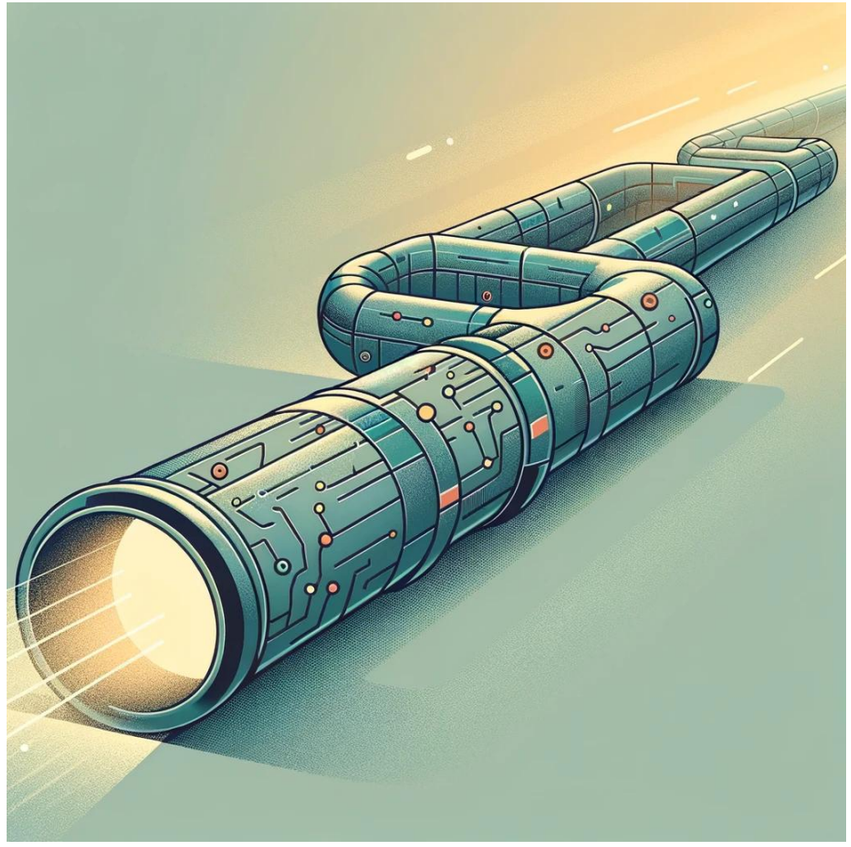
Beispiele Systemintegration mit einem Onlineshop

Integration des Onlineshops mit CRM und Marketing-Systemen: Über Systemintegration kann der Onlineshop mit CRM- und Marketing-Systemen verbunden werden. So können Kundendaten, Bestellhistorien und Marketingaktionen synchronisiert werden, um personalisierte Angebote und Empfehlungen im Shop anzuzeigen.

Integration eines Onlineshops mit einem ERP-System: Wenn ein Kunde im Onlineshop eine Bestellung aufgibt, wird diese Bestellung über eine Schnittstelle in Echtzeit an das ERP-System des Unternehmens übermittelt. So können die Bestände aktualisiert, die Rechnungsstellung angestossen und die Logistikprozesse für den Versand initiiert werden - alles ohne manuelle Eingriffe.

Quellen: https://blog.seeburger.com/de/b2b-onlineshop-integration-wie-bringen-sie-ihren-b2b-webshop-auf-das-naechste-level/, https://www.lobster-world.com/de/wiki/application-integration/

Datenintegration



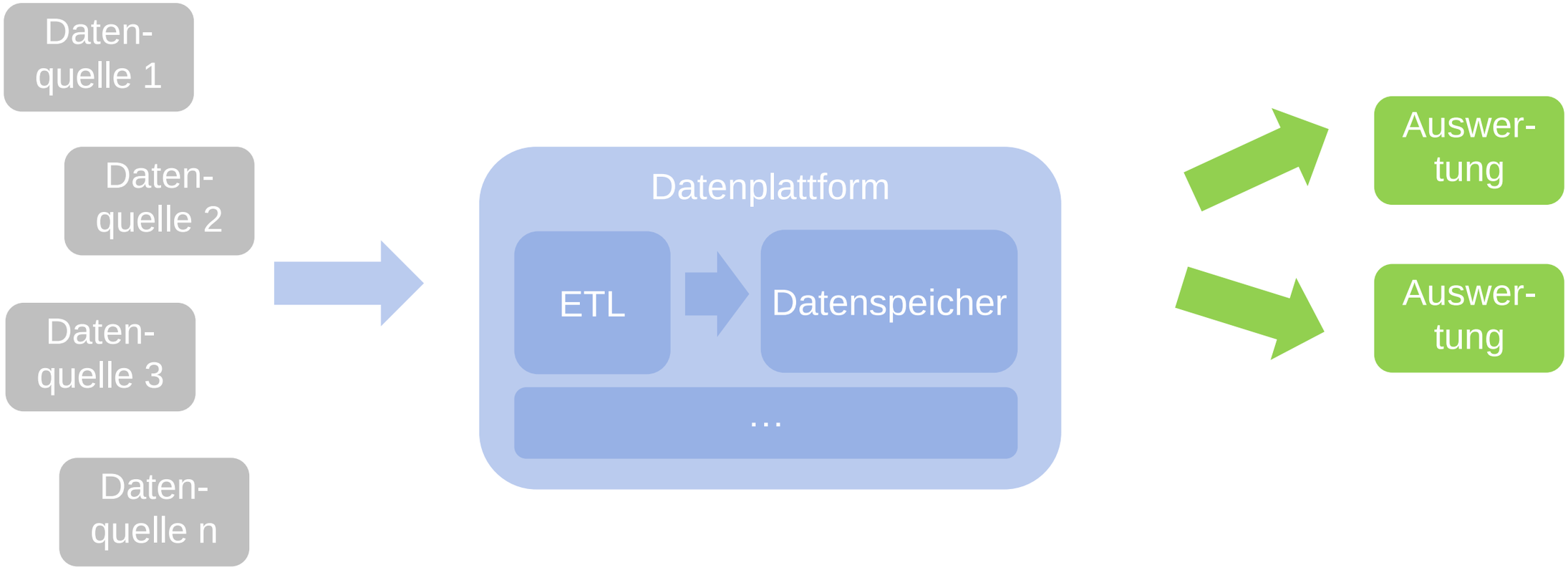
Quelle: OpenAI. (2024). ChatGPT (Version 4), Chatbot-Output vom 26.03.2024 (Prompt: «a pipeline that integrates data. singel data object going through the pipe»)

Definition

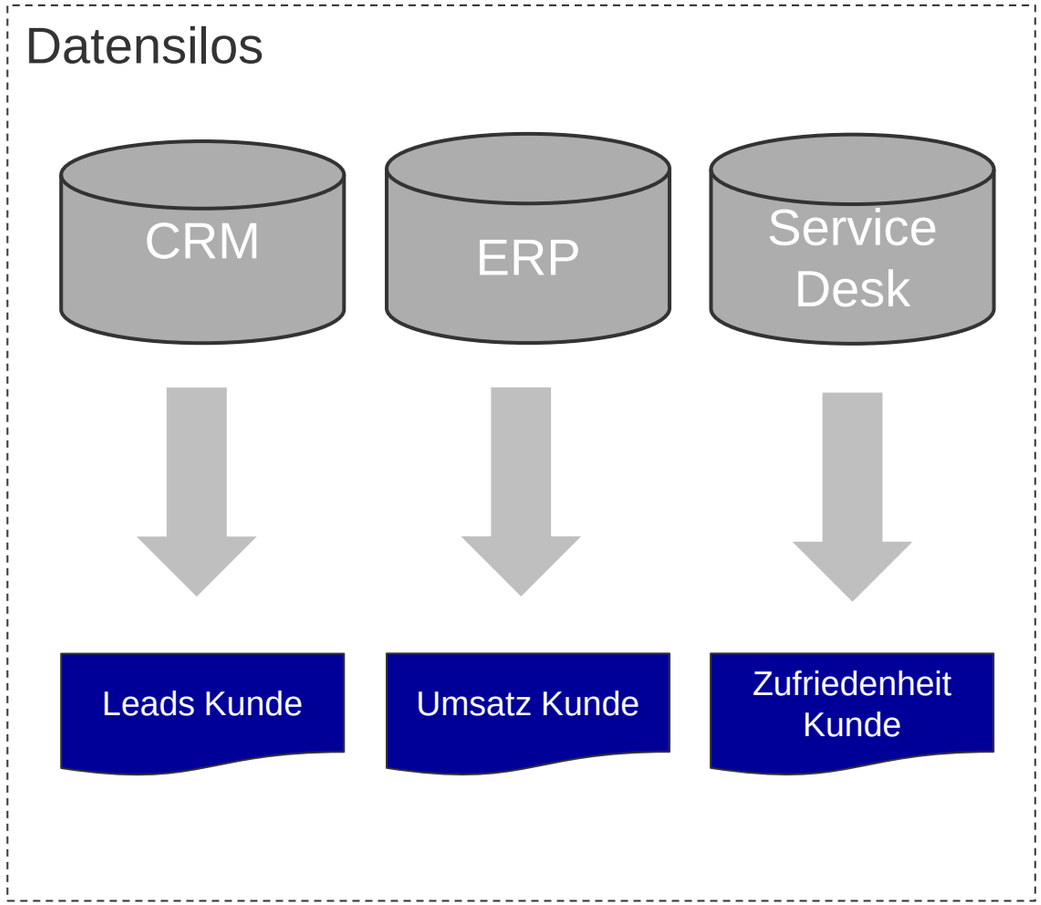
„Bei der **Datenintegration** wird für alle Applikationen eine (oder mehrere) zentrale Datenbank mit einem einheitlichen Datenmodell bereitgestellt. Ziel ist es, **Datenredundanzen durch das zentrale Datenkonzept** zu vermeiden und gleichzeitig allen beteiligten Applikationen **zur selben Zeit den aktuellen Datenstatus** zur Verfügung zu stellen.“

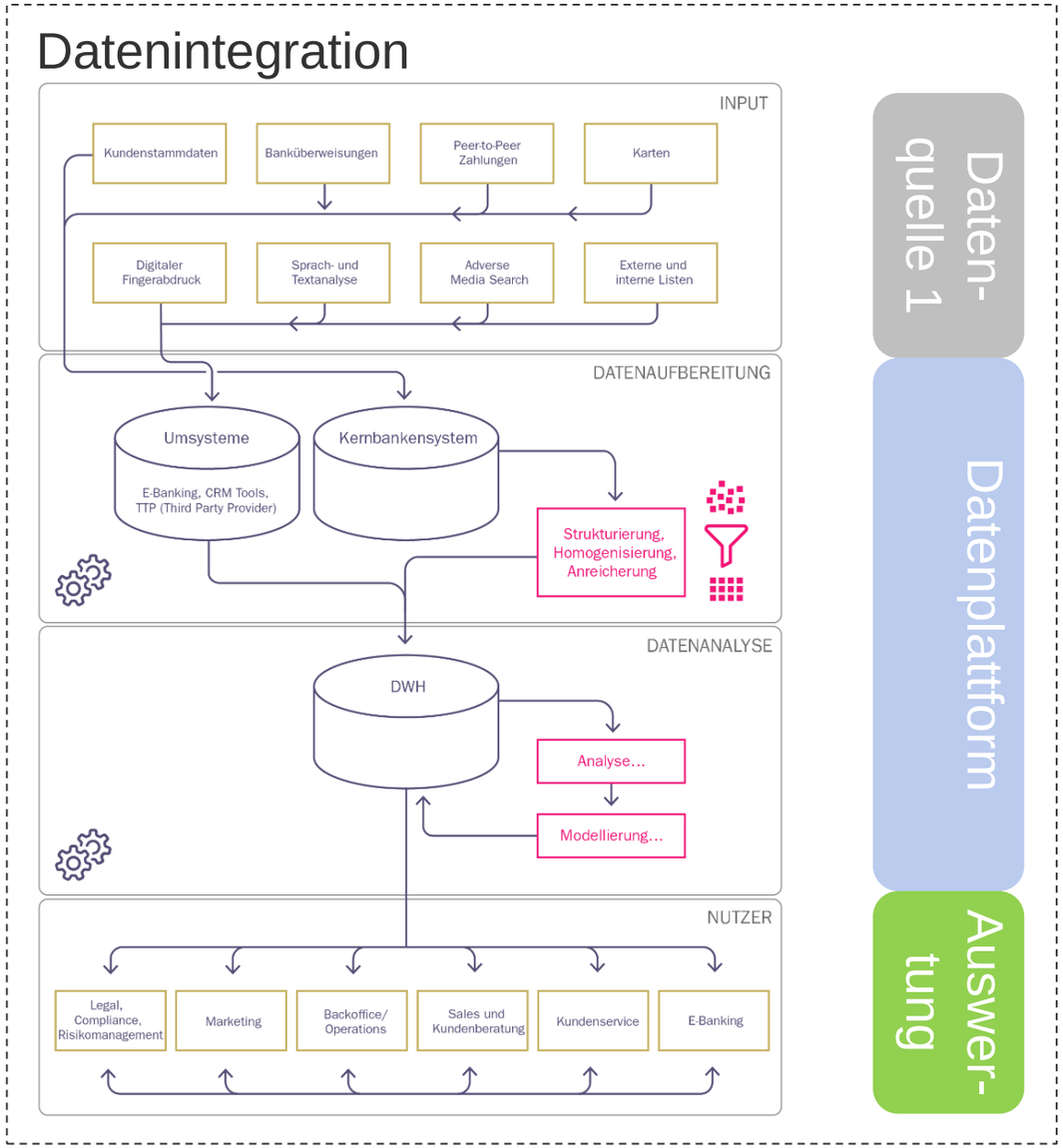
Quelle: Schubert, Winkelmann (2023). Betriebswirtschaftliche Anwendungssysteme, Seite 14

Datenintegration high level



Ziel:
Vermeidung von Datensilos





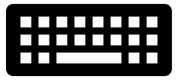
Quelle: https://www.netzwoche.ch/news/2022-11-14/data-driven-banking-das-brachliegende-potenzial-bei-banken

Repetition: Datenqualität

Datenqualitätsprobleme

1. Datenduplikate
2. Unvollständigkeit
3. Fehlende Werte
4. Falsche Formatierung
5. Gültigkeit Wertebereich
6. Widerspruch in den Daten

Ursachen mangelnder Datenqualität



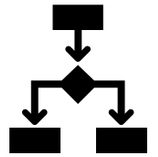
Fehler bei der Erfassung

Mangelhafte Datenpflege und -verwaltung

Mangelnde Datenstandards

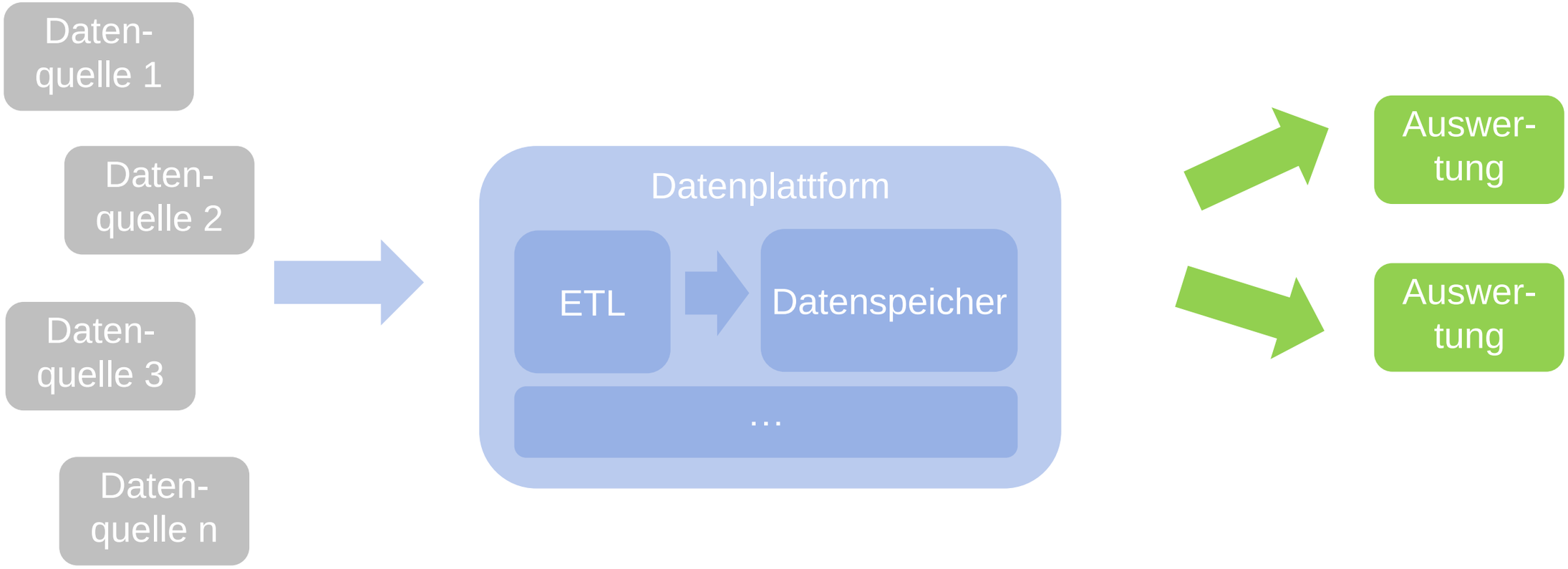
Zeitdruck

Veraltete Daten



Integration ohne Bereinigung / Konsolidierung

Wo muss Datenqualität in der Integrationskette berücksichtigt werden?



Beispiele Datenintegration

Produktdatenintegration aus verschiedenen Quellen: Die Produktdaten (Beschreibungen, Bilder, Preise etc.) für den Onlineshop stammen oft aus unterschiedlichen Systemen wie PIM (Product Information Management), ERP, Datenbanken etc. Durch Datenintegration können diese Daten in einem zentralen Produktkatalog zusammengeführt und für den Onlineshop bereitgestellt werden. So erhält der Kunde stets aktuelle und konsistente Produktinformationen.

Ein **Data Warehouse/Data Lake** wird aufgebaut, um Analysen des Kundenverhaltens und Personalisierung zu ermöglichen, was zu verbesserten Entscheidungsfindungen & effizienteren Marketingstrategien führt.

Quelle: https://blog.seeburger.com/de/b2b-onlineshop-integration-wie-bringen-sie-ihren-b2b-webshop-auf-das-naechste-level/,

Systemintegration

- Konzentriert sich auf die Integration von Geschäftsanwendungen und Prozessen über Anwendungsgrenzen hinweg.
- Ermöglicht den *Austausch von Daten* und Funktionen zwischen verschiedenen Anwendungen in Echtzeit.
- Zielt auf die Verknüpfung der Geschäftslogik und Prozesse über mehrere Anwendungen ab.
- Ermöglicht eine einheitliche Benutzererfahrung über verschiedene Anwendungen hinweg.

Datenintegration

- Konzentriert sich auf die *Zusammenführung und Bereinigung* von Daten aus verschiedenen Quellen.
- *Aggregiert und transformiert Daten* aus Datenbanken, Dateien, Streams etc. in einem zentralen Datenspeicher.
- Zielt auf eine *einheitliche, konsistente* Sicht auf die *Unternehmensdaten* ab.
- *Beseitigt Redundanzen* und ermöglicht effizientes Datenmanagement.

System- & Datenintegration ist für „mein“ BIS relevant, weil

- **Daten oft von anderen Systemen integriert werden müssen oder Daten anderen Systemen zur Verfügung gestellt werden müssen.**

- **Bekannt sein muss, aus welchen Systemen, in welchem Format, über welche Schnittstelle, etc. die Daten, welche mein BIS benötigt, bezogen werden können (dazu wird bspw. ein Technischer Data Owner & ein Data Engineer benötigt)**

Data Lineage

Einführung Data Lineage

- [Data Lineage - einfach erklärt. Definition und Praxisanwendung.]

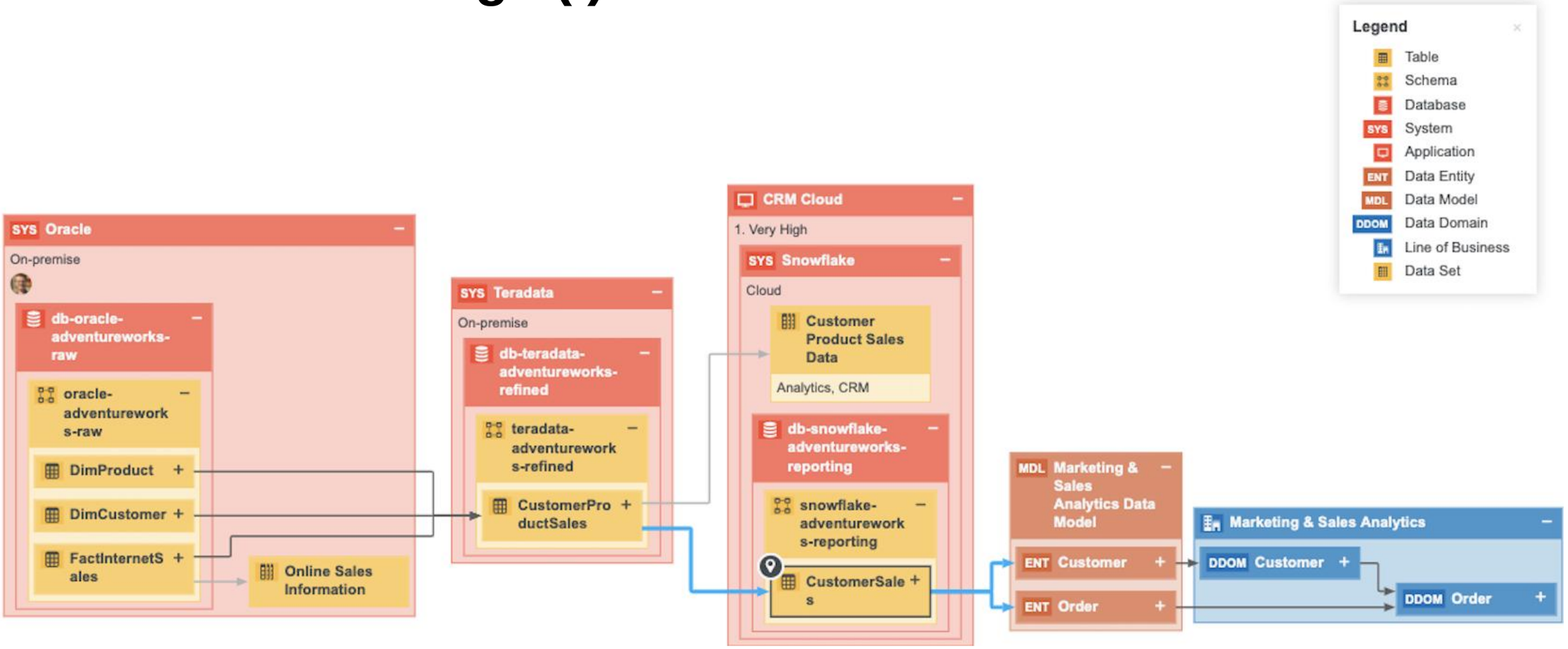
→ Notiert euch die Stichworte – wir setzen diese nachher in Verbindung

Woher stammen die verwendeten Daten?

Data Lineage zeigt auf, woher Daten, bspw. in einem Data Warehouse, ursprünglich stammen. Es wird auch ersichtlich über welche Wege sie sich im Unternehmen bewegen. Dadurch kann der Kontext der verwendeten Daten verstanden werden.

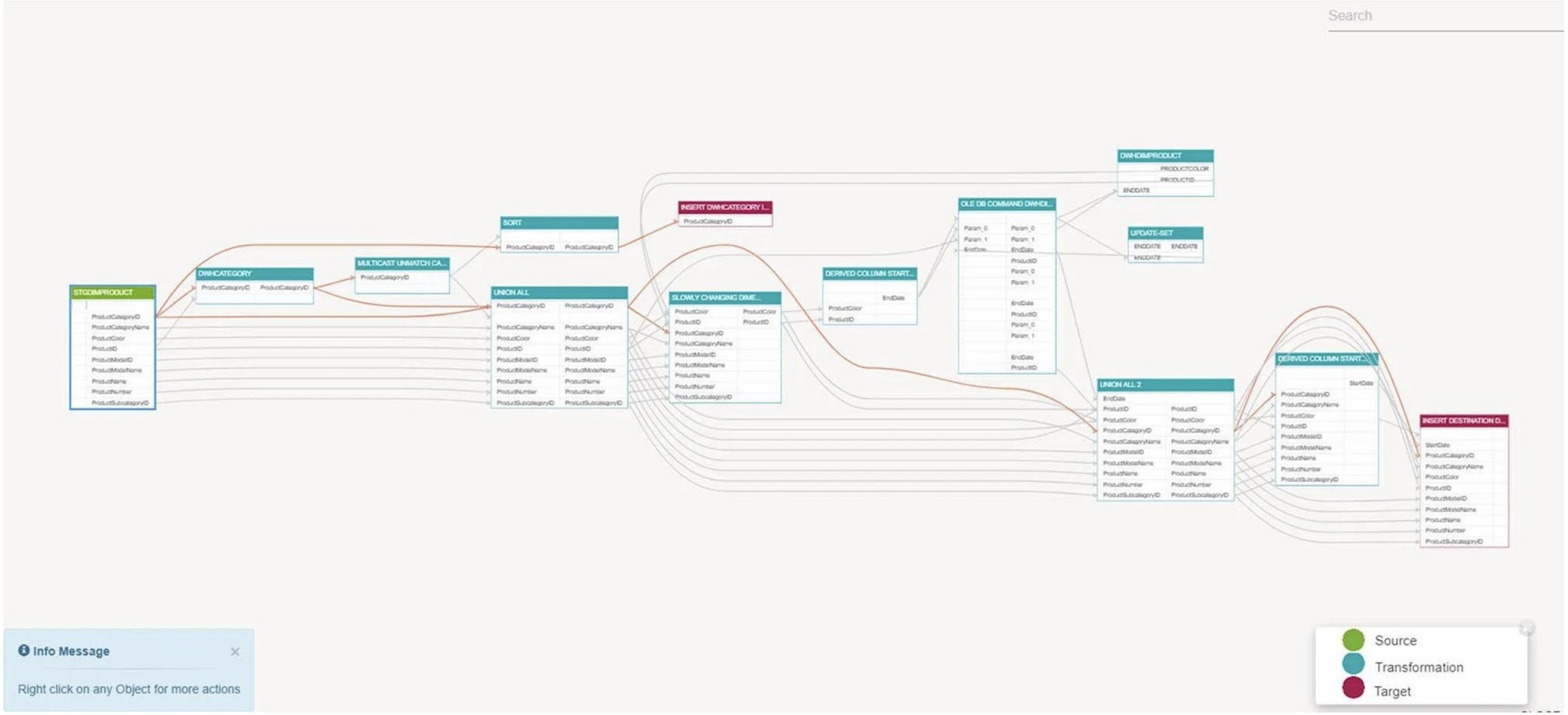
In gewisser Weise lässt sich ein Data-Lineage-System als "Daten-GPS" bezeichnen, das "Abbiegehinweise sowie einen visuellen Überblick über die vollständig kartierte Route" ***Quelle: https://www.computerwoche.de/a/was-ist-data-lineage,3551218***

Tools für Data Lineage (I)



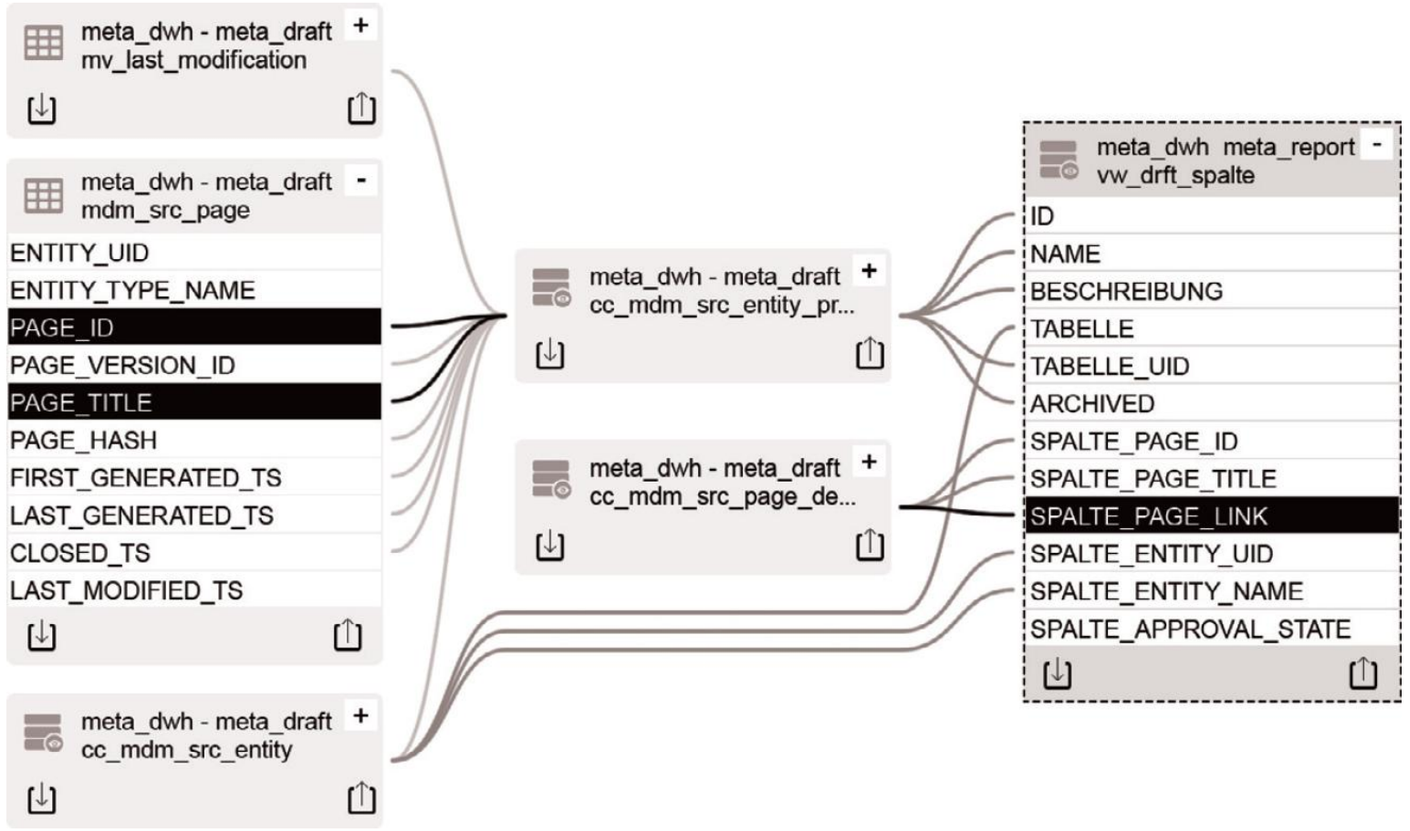
Quelle: https://www.collibra.com/us/en/blog/data-lineage-diagrams

Tools für Data Lineage (II)



Quelle: https://www.keboola.com/blog/data-lineage-tools

Tools für Data Lineage (III)



Quelle: Screenshot aus Collibra, April 2024

Nutzen von Data Lineage

- **Fehlersuche bei gebrochenen ETL-Prozessen**
- **Rückverfolgung von Änderungen verbessert die Datenqualität**
- **Einhaltung gesetzlicher Vorschriften (bspw. DSG „Recht auf Vergessen“, Datenflüsse personenbezogener Daten)**
- **Unterstützung bei Migrationen durch Identifizierung ungenutzter Daten**
- **Analyse der Auswirkungen von Prozessänderungen (Anpassung Datenfluss) Datenänderungen**

rückwirkend

kontinu-
ierlich

voraus-
schauend

Quelle: Malcolm Chisholm, Ph.D., The Essential Guide to Data Lineage

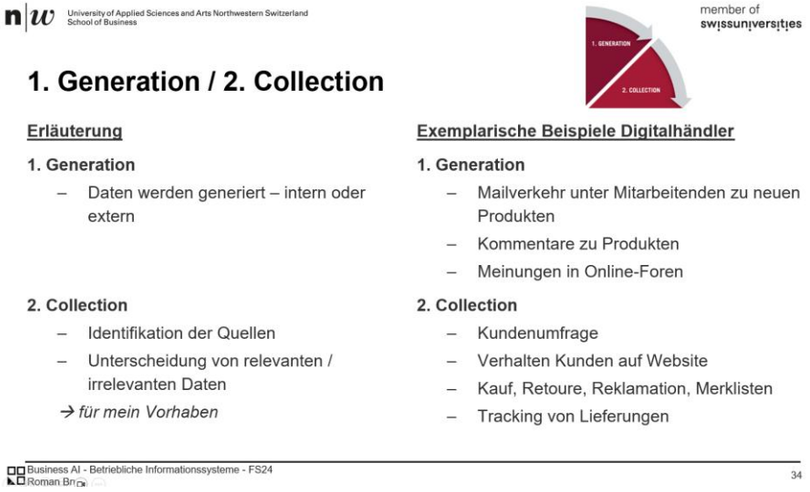
Gruppenarbeit

Auftrag 8 & 9: Klassenfeedback

Zyklus

Generation: Daten werden generiert – intern oder extern. Auch Daten, welche für mein Vorhaben irrelevant sind.

Collection: Identifikation der relevanten Quellen. Unterscheidung von relevanten / irrelevanten Dater



Bezug zur gesamten Arbeit herstellen

Generell und für diesen Auftrag: «Berücksichtigt auch die verwendeten Geschäftsobjekte bzw. Datenobjekte aus dem ArchiMate® Modell eurer Gruppenarbeit.»

2. Collection:

- Scan von Barcodes bei Wareneingang
- Einlesen von Lieferscheinen
- Erfassung von Kundendaten
- Komissionierung/Auftragseingang

Gruppenarbeit: Klassenfeedback

Erinnerung: Berücksichtigt die Hinweise zu Verwendung von Künstlicher Intelligenz (KI) für Leistungsnachweise

- **Anhang 1: Eigenständigkeitserklärung**
- **Anhang 2: Hilfsmittelverzeichnis**

Auftrag 8 & 9: Feedback pro Gruppe

Vormittag
Gruppe 6
Gruppe 7
Gruppe 4
Gruppe 2

Nachmittag
Gruppe 5
Gruppe 2
Gruppe 1
Gruppe 8
Gruppe 12
Gruppe 7
Gruppe 10