IF3140 Database System

Data Integration and Interoperability

SEMESTER I 2024/2025

KNOWLEDGE & SOFTWARE ENGINEERING

Source

DAMA DMBOK: Data Management Body of Knowledge, 2nd Edition, Technics Publications, 2017

- Chapter 8 Data Integration & Interoperability

Outline

- Introduction
- Extract, Transform, Load (ETL)
- ETL Design Challenges
- ELT (Extract Load Transform)
- Mapping
- Latency
- Interaction Models
- DII Architecture Concepts
- Tools

KNOWLEDGE & SOFTWARE ENGINEERING

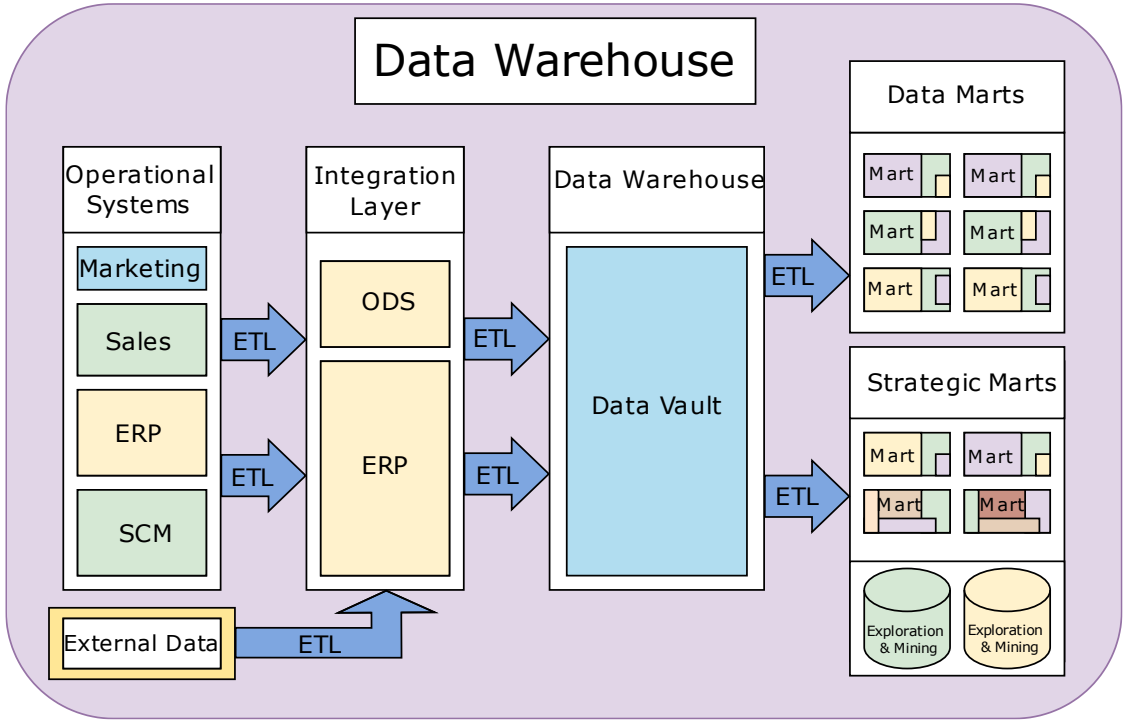
Introduction

- **Data Integration and Interoperability (DII)** describes processes related to the movement and consolidation of data within and between data stores, applications and organizations.
- Enables basic data management functions on which most organization depend, e.g.:
  - Data migration and conversion
  - Data sharing between applications and across organizations
  - Distributing data across data stores and data centers
  - …

KNOWLEDGE & SOFTWARE ENGINEERING

Introduction

- DII is critical for area of Datawarehousing & Business Intelligence
  - All of which focus on transforming and integrating data from source systems to consolidated data hubs

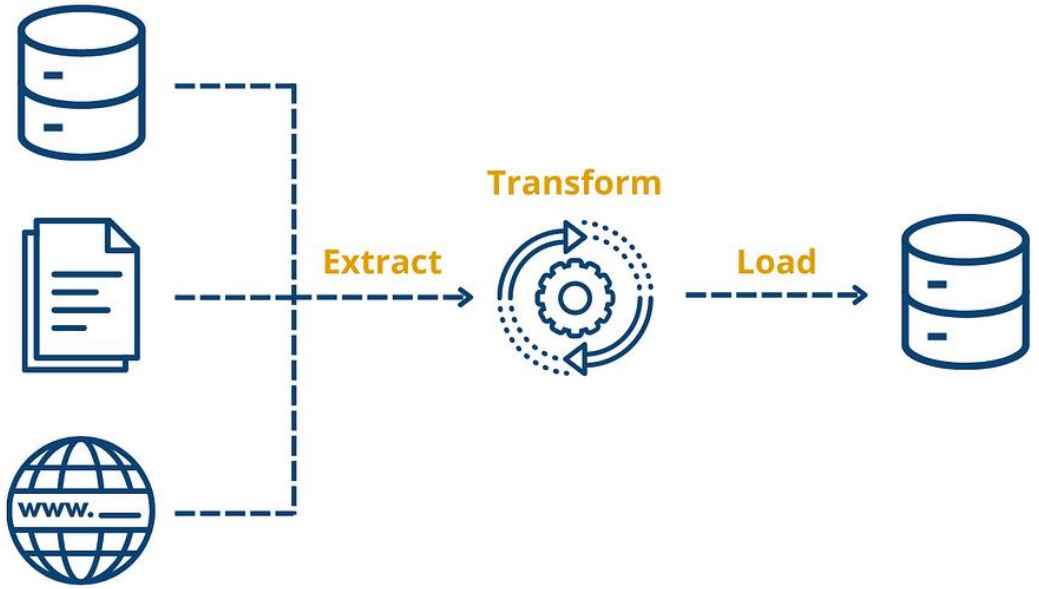


By MarsJson - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=145039784

KNOWLEDGE & SOFTWARE ENGINEERING

Extract Transform Load (ETL)

- The essential steps in moving data around and between applications and organizations
- Can be performed periodically (batch) or real-time (event-driven)



https://towardsdatascience.com/beginners-guide-extract-transform-load-etl-49104a8f9294

KNOWLEDGE & SOFTWARE ENGINEERING

Extract

- Selecting the required data and extracting it from its source
  - Extracted data is then staged, in a physical data store on disk or in memory
  - if this process executes on an operational system, it is designed to use as few resources as possible, in order to avoid negatively affecting the operational processes. Batch processing during off-peak hours is an option
  - Challenge: Combine data from different source systems
    - Each separate system may also use a different data organization and/or format.
    - Common data-source formats include relational databases, flat-file databases, NoSQL databases, XML, and JSON,

KNOWLEDGE & SOFTWARE ENGINEERING

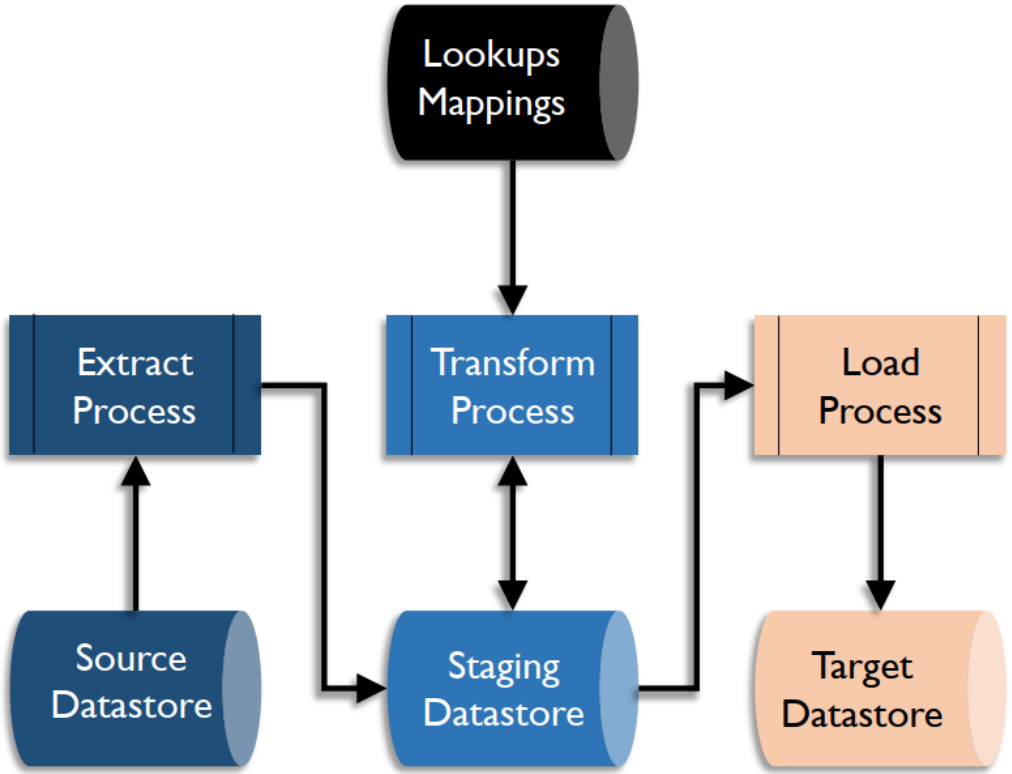
Transform

- Makes the selected data compatible with the structure of the target data store
- An important function of transformation is **data cleansing**, which aims to pass only "proper" data to the target.
- Challenge: when different systems interact is in the relevant systems' interfacing and communicating
  - E.g. Character sets that may be available in one system may not be in others.
- Examples of transformation:
  - **Format changes**: Conversion of the technical format of the data; for example, from EBCDIC to ASCII format
  - **Structure changes**: Changes to the structure of the data; for example, from denormalized to normalized records
  - **Semantic conversion**: Conversion of data values to maintain consistent semantic representation.
  - **De-duping**: Ensuring that if rules require unique key values or records, a means for scanning the target, and detecting and removing duplicate rows, is included
  - **Re-ordering**: Changing the order of the data elements or records to fit a defined pattern

KNOWLEDGE & SOFTWARE ENGINEERING

Load

- Physically storing or presenting the result of the transformations in the target system
- Depending on the transformations performed, the target system's purpose, and the intended use, the data may need further processing to be integrated with other data, or it may be in a final form, ready to present to consumers.



KNOWLEDGE & SOFTWARE ENGINEERING

ETL Design Challenges

- **Data variations**
  - The range of data values or data quality in an operational system may exceed the expectations of designers
  - Data warehouses are typically assembled from a variety of data sources with different formats and purposes.
  - Design analysis should establish the scalability of an ETL system across the lifetime of its usage – including understanding the volumes of data that must be processed within service level agreements.
  - **Data profiling** of a source during data analysis can identify the data conditions that must be managed by transform rules specifications

KNOWLEDGE & SOFTWARE ENGINEERING

ETL Design Challenges

- **Uniqueness of keys**
  - Unique keys play an important part in all relational databases, as they tie everything together.
  - As there is usually more than one data source getting loaded into the warehouse, the keys are an important concern to be addressed.
    - E.g. customers might be represented in several data sources, with their Social Security number as the primary key in one source, their phone number in another, and a surrogate key in the third. A data base target may require the consolidation of all the customer information into one dimension.

KNOWLEDGE & SOFTWARE ENGINEERING

ETL Design Challenges

- **Performance**
  - In real life, the slowest part of an ETL process usually occurs in the database load phase → may perform slowly because they have to take care of concurrency, integrity maintenance, and indices
  - Common methods to increase performance:
    - Disable integrity checking in the target database tables during load
    - Disable triggers in the target database tables during load
    - Drop indices before load, recreate them after load
    - Generate IDs in ETL layer (not in the database)
    - For insert, update, and delete processes find out which one to be done in ETL layer, which one to do via API
    - Try to keep partitions similar in size
    - Bulk load operations whenever possible, use parallel bulk load when possible

KNOWLEDGE & SOFTWARE ENGINEERING

ETL Design Challenges

- **Performance**
  - Other challenges:
    - Certain operations in the database or outside may involve a trade-off
    - A big number of dependencies among ETL job, e.g. B cannot be started before A finishes
    - When the data are spread among several databases, and processing is done in those databases sequentially.

KNOWLEDGE & SOFTWARE ENGINEERING

ETL Design Challenges

- **Parallel computing**
  - Parallel processing enables a number of methods to improve overall performance of ETL when dealing with large volumes of data.
  - Three types of parallelism
    - **Data**: By splitting a single sequential file into smaller data files to provide parallel access
    - **Pipeline**: allowing the simultaneous running of several components on the same data stream, e.g. looking up a value on record 1 at the same time as adding two fields on record 2
    - **Component**: The simultaneous running of multiple processes on different data streams in the same job, e.g. sorting one input file while removing duplicates on another file

KNOWLEDGE & SOFTWARE ENGINEERING

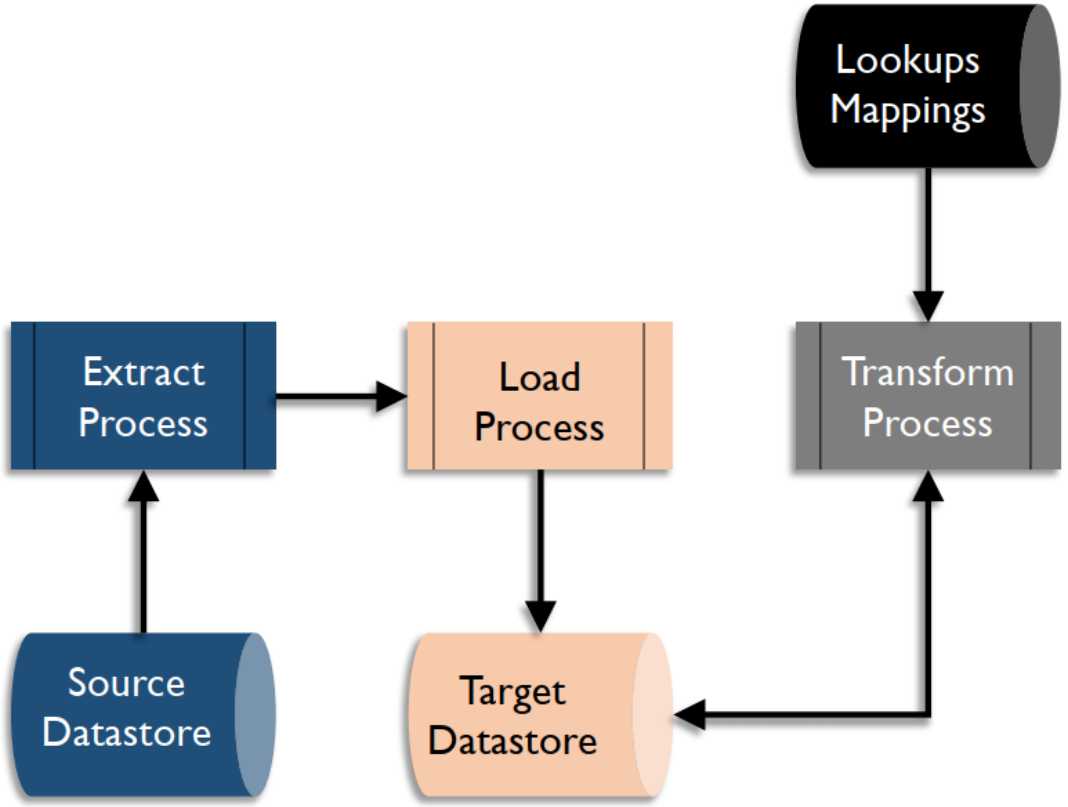
ETL Design Challenges

- **Failure recovery**
  - A big ETL process is usually subdivided into smaller pieces running sequentially or in parallel
  - Each data row is tagged with "row_id", each piece of process is tagged with "run_id" → in case of failure, these IDs will help roll back and rerun the failed piece
  - Other best practices: using checkpoints when certain phases of the process are completed.
    - In checkpoints: write everything to disk, clean out temporary files, log the states

KNOWLEDGE & SOFTWARE ENGINEERING

ELT (Extract Load Transform)

- If the target system has more transformation capability than either the source or an intermediary application system, the order of processes may be switched to ELT – Extract, Load, and Transform
  - Transformations are to occur after the load to the target system
  - Allows source data to be instantiated on the target system as raw data, which can be useful for other processes.



KNOWLEDGE & SOFTWARE ENGINEERING

Mapping

- A mapping is both the process of developing the lookup matrix from source to target structures and the result of that process
- Defines the sources to be extracted, the rules for identifying data for extraction, targets to be loaded, rules for identifying target rows for update (if any), and any transformation rules or calculations to be applied

Example of data mapping template

ETL Package Name	*Load_Dim_Product.dtsx*
Target Table	*ADDW.Dim.Product*
Source Tables	*Production.Product* *Production.ProductSubcategory*
Load Frequency	*Daily*
Target Type	*SCD Type 1*
Mapping Description	AD.Production.Product is the lowest level of source data. Join the source tables in the following manner: FROM AD.Production.Product INNER JOIN AD.Production.ProductSubcategory ON Product.SubcategoryID = ProductSubcategory.SubcategoryID INNER JOIN AD.Production.ProductCategory ON ProductSubcategory.ProductCategoryID = ProductCategory.ProductCategoryID
Error Handling	On an error occuring any inserted records or updates needs to roll back and leave the system in the previous correct state.

https://www.researchgate.net/figure/Sample-data-mapping-template_fig1_336851649

KNOWLEDGE & SOFTWARE ENGINEERING

Latency

- Latency is the time difference between when data is generated in the source system and when the data is available for use in the target system.
- Different approaches to data processing result in different degrees of data latency.
- Several concepts related to latency:
  - Batch
  - Near real time and event-driven
  - Change Data Capture
  - Asynchronous
  - Real-time, synchronous
  - Low latency or streaming

KNOWLEDGE & SOFTWARE ENGINEERING

Batch

- Most data moves between applications and organizations in clumps or files either on request by a human data consumer or automatically on a periodic schedule (batch)
- Data moving in batch mode will represent either the full set of data at a given point in time or data that has changed values since the last time the data was sent
- The set of changed data is called the **delta**, and the data from a point in time is called a **snapshot**
- There is often a significant delay between when data changes in the source and when it is updated in the target, resulting in **high latency**
- Batch processing is very useful for processing very high volumes of data in a short time window

KNOWLEDGE & SOFTWARE ENGINEERING

Near-Real-Time or Event-Driven

- Data is processed in smaller sets spread across the day in a defined schedule, or data is processed when an event happens, such as a data update.
- Near-real-time processing has a lower latency than batch processing and often a lower system load as the work is distributed over time, but it is usually slower than a synchronized data integration solution (see slide 23)
- State information and process dependencies must be monitored by the target application load process.
  - Data coming into the target may not be available in the exact order that the target needs to build the correct target data.

KNOWLEDGE & SOFTWARE ENGINEERING

Change Data Capture (CDC)

- Change Data Capture (CDC) is a method of reducing bandwidth by filtering to include only data that has been changed within a defined timeframe.
- CDC monitors a data set for changes (inserts, changes, deletes) and then passes those changes (the deltas) to other data sets, applications, and organizations that consume the data.
- Three techniques for data-based CDC
  - The source system populates specific data elements, such as timestamps within a range, or codes or flags, which serve as change indicators. The extract process uses rules to identify rows to extract.
  - The source system processes add to a simple list of objects and identifiers when changing data, which is then used to control selection of data for extraction.
  - The source system processes copy data that has changed into a separate object as part of the transaction, which is then used for extract processing. This object does not need to be within the database management system.

KNOWLEDGE & SOFTWARE ENGINEERING

Asynchronous

- In an asynchronous data flow, the system providing data does not wait for the receiving system to acknowledge update before continuing processing.
- Asynchronous data integration does not prevent the source application from continuing its processing, or cause the source application to be unavailable if any of the target applications are unavailable.
  - Since the data updates made to applications in an asynchronous configuration are not immediate, the integration is called near-real-time

KNOWLEDGE & SOFTWARE ENGINEERING

Real-Time Synchronous

- When data in one data set must be kept perfectly in synch with the data in another data set, then a real-time, synchronous solution must be used
- In a synchronous integration solution, an executing process waits to receive confirmation from other applications or processes prior to executing its next activity or transaction.
  - The solution can process fewer transactions because it has to spend time waiting for confirmation of data synchronization
- Real-time, synchronous solutions require less state management than asynchronous solutions because the order in which transactions are processed is clearly managed by the updating applications. However, they also may lead to blocking and delay other transactions.

KNOWLEDGE & SOFTWARE ENGINEERING

Low Latency or Streaming

- Low latency data integration solutions are designed to minimize the response time to events
  - They may include the use of hardware solutions like solid-state disk or software solutions like in-memory databases
  - Requires a large investment in hardware and software
  - Justified if an organization requires extremely fast data movement across large distances
  - Data streams capture events like the purchase of goods or financial securities, social media comments, and readouts from sensors monitoring location, temperature, usage, or other values.
  - Asynchronous solutions are usually used in low latency solutions
  - Massive multi-processing, or simultaneous processing, is also a common configuration in low latency solutions

KNOWLEDGE & SOFTWARE ENGINEERING

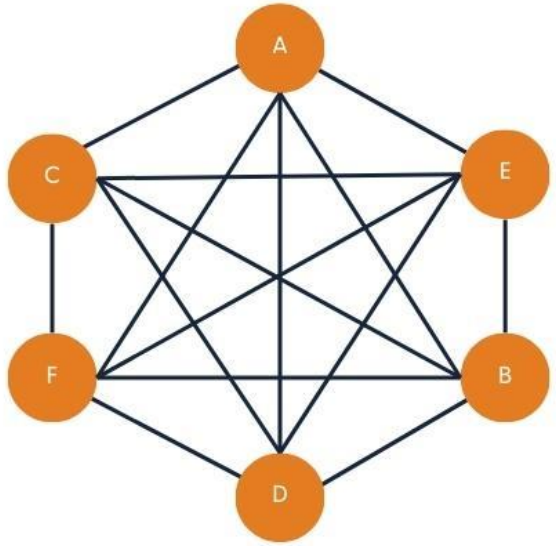
Interaction Models

- Interaction models describe ways to make connections between systems in order to transfer data.
- Several models:
  - Point-to-point
  - Hub-and-spoke
  - Publish-and-subscribe

KNOWLEDGE & SOFTWARE ENGINEERING

Point-to-Point

- In point-to-point interaction model, systems pass data directly to each other
  - Makes sense in the context of small set of systems
  - Become quickly inefficient and increases organizational risks when many systems require the same data from the same sources
    - **Impact to processing**: If source systems are operational, then the workload from supplying data could affect processing
    - **Managing interfaces**: The number of interfaces needed in a point-to-point interaction model approaches the number of systems squared (s2). he workload to manage and support interfaces between the systems can quickly become greater than supporting the systems themselves.
    - **Potential for inconsistency**: Design issues arise when multiple systems require different versions or formats of the data. The use of multiple interfaces to obtain data will lead to inconsistencies in the data sent to downstream systems.

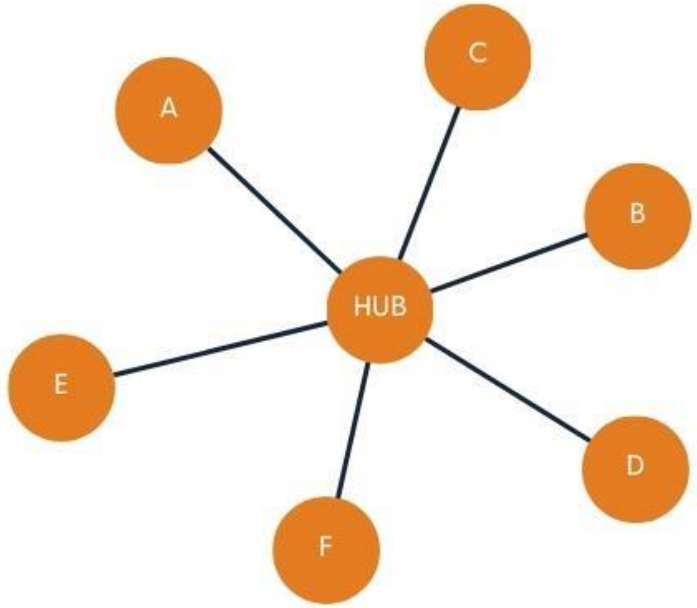


https://palletonline.co.uk/blog/hub-and-spoke-vs-point-to-point

KNOWLEDGE & SOFTWARE ENGINEERING

Hub-and-Spoke

- The hub-and-spoke model consolidates shared data (either physically or virtually) in a central data hub that many applications can use.
- The hubs provide consistent views of the data with limited performance impact on the source systems.
- Data hubs even minimize the number of systems and extracts that must access the data sources, thus minimizing the impact on the source system resources.
- Some hub-and-spoke model latency is unacceptable or performance is insufficient. The hub itself creates overhead in a hub-and-spoke architecture.

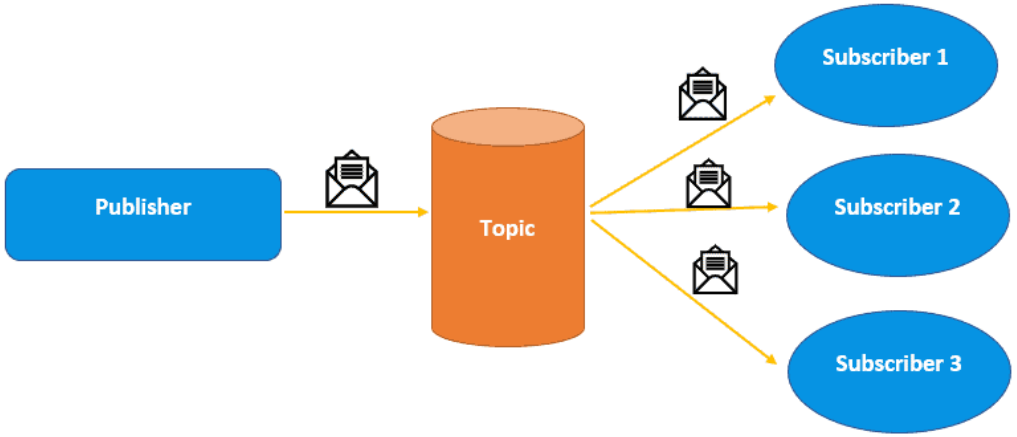


https://www.devopsschool.com/blog/what-is-the-hub-and-spoke-model/

KNOWLEDGE & SOFTWARE ENGINEERING

Publish-and-Subscribe

- A publish and subscribe model involves systems pushing data out (publish), and other systems pulling data in (subscribe).
- Systems providing data are listed in a catalog of data services, and systems looking to consume data subscribe to those services.
  - When data is published, the data is automatically sent to the subscribers.

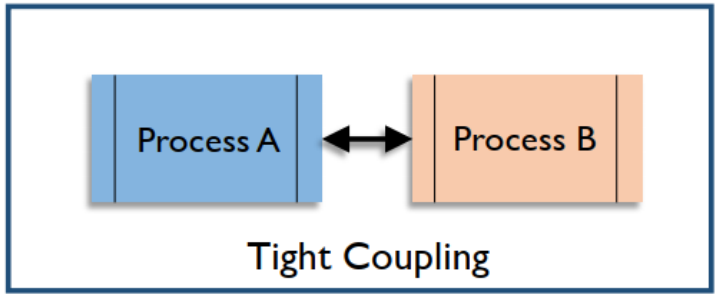


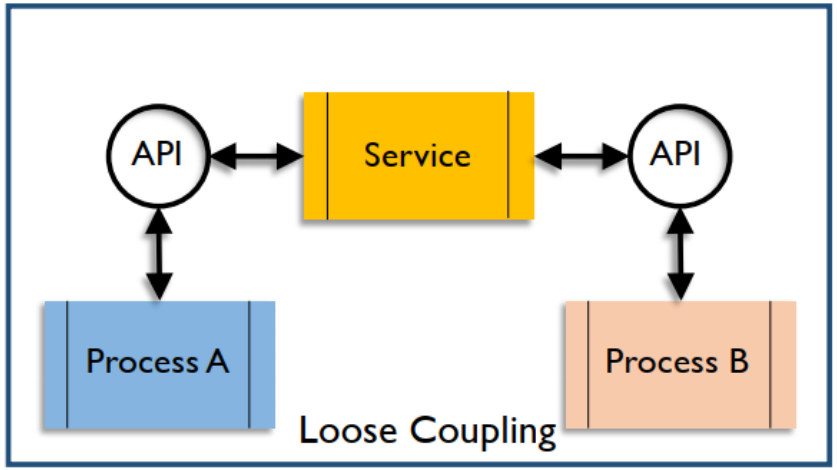
https://www.baeldung.com/cs/publisher-subscriber-model

KNOWLEDGE & SOFTWARE ENGINEERING

DII Architecture Concepts

- **Application Coupling**
  - Coupling describes the degree to which two systems are entwined.
  - Two systems that are tightly coupled usually have a synchronous interface, where one system waits for a response from the other.
    - Tight coupling represents a riskier operation: if one system is unavailable then they are both effectively unavailable, and the business continuity plan for both have to be the same.
  - In loose coupling, data is passed between systems without waiting for a response and one system may be unavailable without causing the other to be unavailable
    - E.g. Service Oriented Architecture using an Enterprise Service Bus





KNOWLEDGE & SOFTWARE ENGINEERING

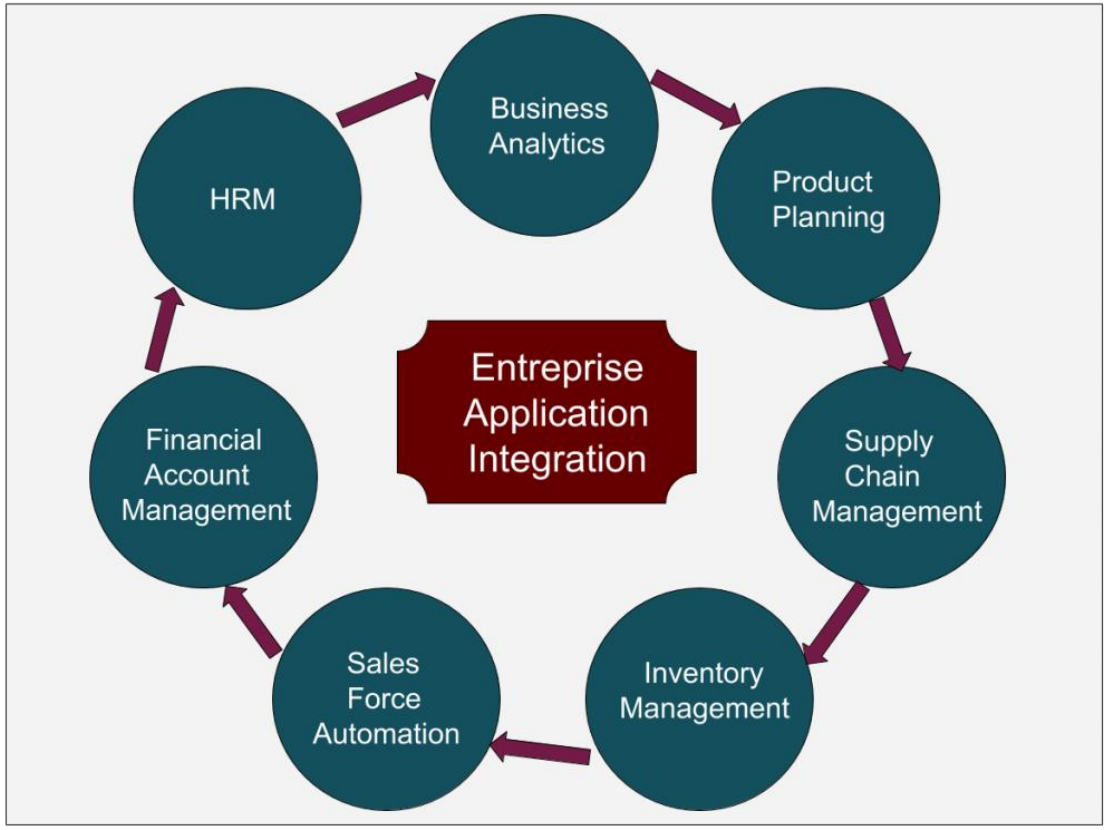
DII Architecture Concepts

- **Orchestration and Process Control**
  - **Orchestration** is the term used to describe how multiple processes are organized and executed in a system.
    - All systems handling messages or data packets must be able to manage the order of execution of those processes, in order to preserve consistency and continuity.
  - Process Controls are the components that ensure shipment, delivery, extraction, and loading of data is accurate and complete, including:
    - Database activity logs
    - Batch job logs
    - Alerts
    - Exception logs
    - Job dependence charts with remediation options, standard responses
    - Job clock information such as timing of dependent jobs, the expected length of the jobs, the computing window time

KNOWLEDGE & SOFTWARE ENGINEERING

DII Architecture Concepts

- **Enterprise Application Integration (EAI)**
  - In an enterprise application integration model (EAI), software modules interact with one another only through well-defined interface calls (application programming interfaces – APIs)
    - Data stores are updated only by their own software modules and other software cannot reach into the data in an application but only access through the defined APIs

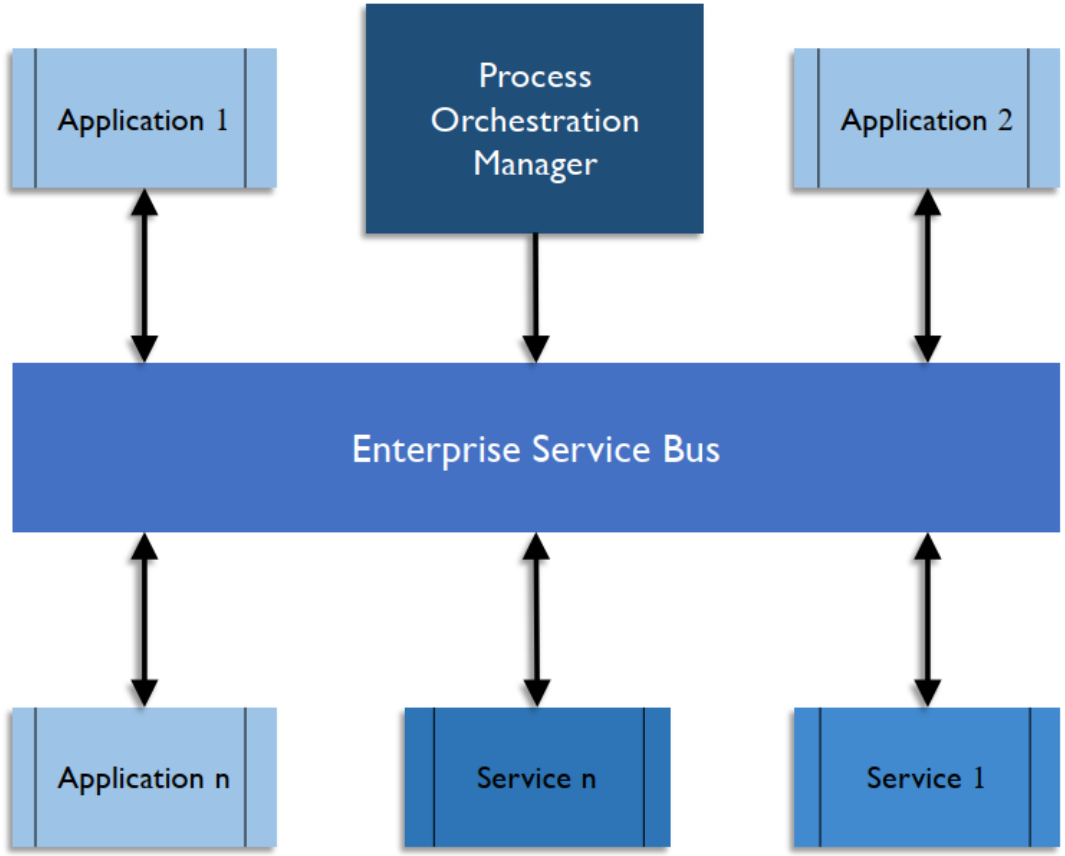


https://medium.com/@erpinformation/a-guide-to-enterprise-application-integration-7842eb399816

KNOWLEDGE & SOFTWARE ENGINEERING

DII Architecture Concepts

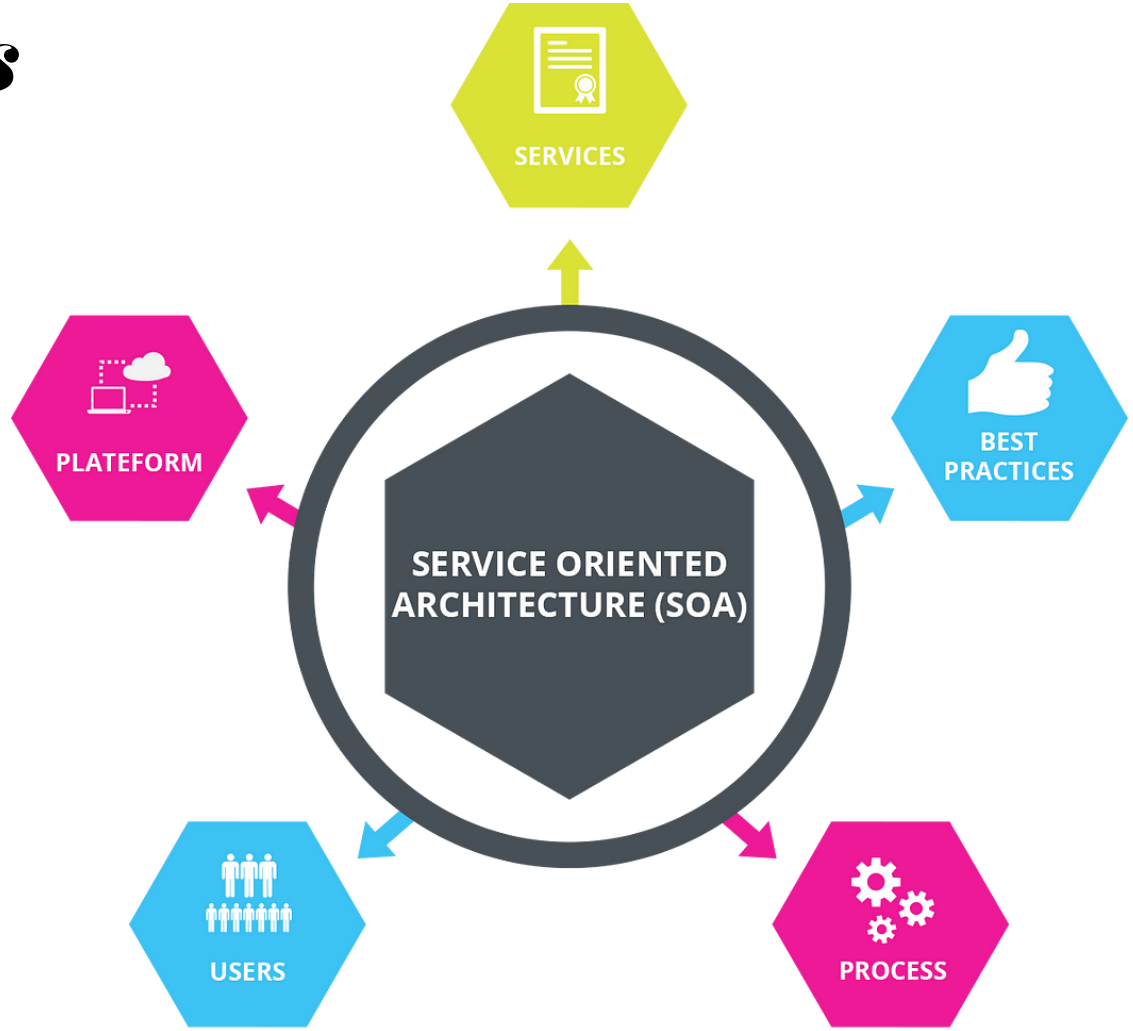
- **Enterprise Service Bus (ESB)**
  - An Enterprise Service Bus is a system that acts as an intermediary between systems, passing messages between them.
  - Applications can send and receive messages or files using the ESB, and are encapsulated from other processes existing on the ESB.



KNOWLEDGE & SOFTWARE ENGINEERING

DII Architecture Concepts

- **Service-Oriented Architecture (SOA)**
  - In Service-Oriented architecture (SOA), the functionality of providing data or updating data (or other data services) can be provided through well-defined service calls between applications
  - SOA enables application independence and the ability for an organization to replace systems without needing to make significant changes to the systems that interfaced with them



https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec

KNOWLEDGE & SOFTWARE ENGINEERING

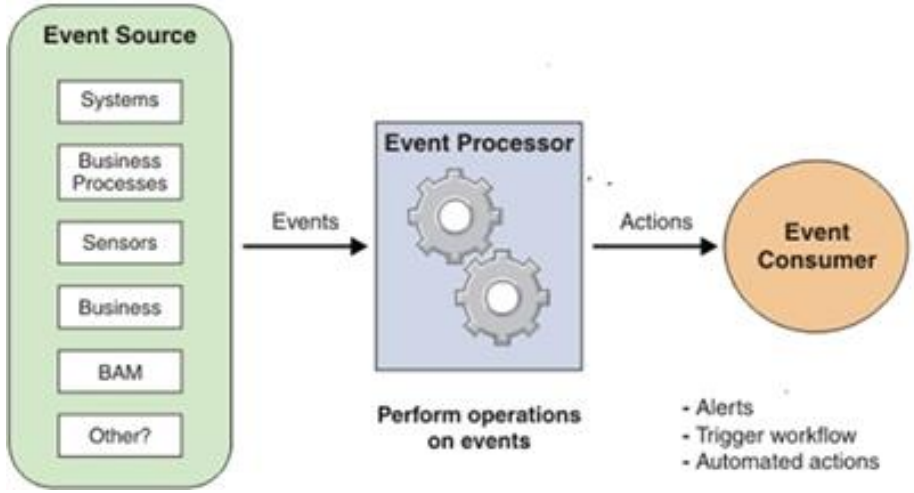
DII Architecture Concepts

- **Service-Oriented Architecture (SOA)**
  - The goal of service-oriented architecture is to have well-defined interaction between self-contained software modules.
  - The key concept is that SOA architecture provides independent services: the service has no fore knowledge of the calling application and the implementation of the service is a black box to the calling application.
  - A service-oriented architecture may be implemented with various technologies including web services, messaging, RESTful APIs, etc.
  - Data services, which may include the addition, deletion, update, and retrieval of data, are specified in a catalog of available services.

KNOWLEDGE & SOFTWARE ENGINEERING

DII Architecture Concepts

- **Complex Event Processing (CEP)**
  - Event processing is a method of tracking and analyzing (processing) streams of information (data) about things that happen (events) and deriving a conclusion from them.
  - Complex event processing (CEP) combines data from multiple sources to identify meaningful events (such as opportunities or threats) to predict behavior or activity and automatically trigger real-time response



https://www.databricks.com/glossary/complex-event-processing#:~:text=Complex%20event%20processing%20is%20an,insight%20into%20what%20is%20happening.

KNOWLEDGE & SOFTWARE ENGINEERING

DII Architecture Concepts

- **Complex Event Processing (CEP)**
  - Organizations can use complex event processing to predict behavior or activity and automatically trigger real-time response.
    - E.g. Events such as sales leads, web clicks, orders, or customer service calls; news items, text messages, social media posts, stock market feeds, traffic reports, weather reports, or other kinds of data.
  - CEP presents some data challenges:
    - In many cases, the rate at which events occur makes it impractical to retrieve the additional data necessary to interpret the event as it occurs.
    - Supporting complex event processing requires an environment that can integrate vast amounts of data of various types.

KNOWLEDGE & SOFTWARE ENGINEERING

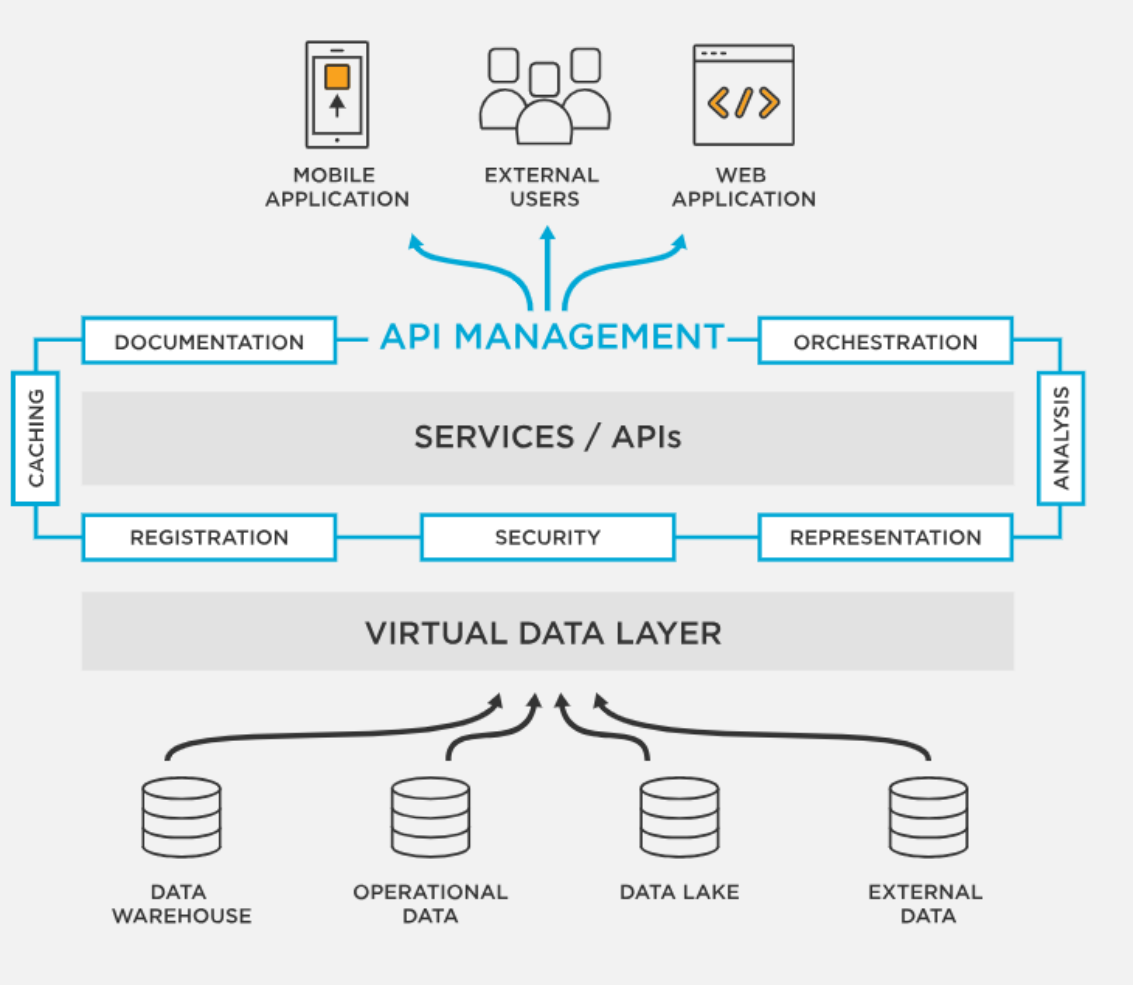
DII Architecture Concepts

- **Data Federation and Virtualization**
  - When data exists in disparate data stores, it can be brought together in ways other than physical integration.
  - Data Federation provides access to a combination of individual data stores, regardless of structure.
  - Data Virtualization enables distributed databases, as well as multiple heterogeneous data stores, to be accessed and viewed as a single database.
  - See again materials on Data Storage and Operations

KNOWLEDGE & SOFTWARE ENGINEERING

DII Architecture Concepts

- **Data-as-a-Service (DaaS)**
  - One definition of Data-as-a-Service (DaaS) is data licensed from a vendor and provided on demand, rather than stored and maintained in the data center of the licensing organization.
  - A common example includes information on the securities sold through a stock exchange and associated prices (current and historical)
  - The 'service' concept is also used within an organization to provide enterprise data or data services to various functions and operational systems.



https://www.tibco.com/glossary/what-is-data-as-a-service-daas

KNOWLEDGE & SOFTWARE ENGINEERING

DII Architecture Concepts

- **Cloud-based Integration**
  - Cloud-based integration (also known as integration platform-as-a-service or IPaaS) is a form of systems integration delivered as a cloud service that addresses data, process, service-oriented architecture (SOA), and application integration use cases.
  - The advent of SaaS (software-as-a-service) applications created a new kind of demand for integrating data located outside of an organization's data center, met through cloud-based integration.
  - Cloud-based integration solutions are usually run as SaaS applications in the data centers of the vendors and not the organizations that own the data being integrated.

KNOWLEDGE & SOFTWARE ENGINEERING

Data Exchange Standards

- Data Exchange Standards are formal rules for the structure of data elements.
- A data exchange specification is a common model used by an organization or data exchange group that standardizes the format in which data will be shared.
- Having an agreed upon exchange format or data layout between systems can significantly simplify data interoperability in an enterprise, lowering the cost of support and enabling better understanding of the data.
- Examples of data exchange standards:
  - **General Transit Feed Specification (GFTS)** is the worldwide de facto standard for publishing, accessing, sharing and using public transport information
  - **Health Level Seven (HL7)** is a range of global standards for the transfer of clinical and administrative health data between applications with the aim to improve patient outcomes and health system performance.
  - The **Financial Data Exchange (FDX)** is a nonprofit industry standards body that created the FDX API, a technical standard for sharing financial data.

KNOWLEDGE & SOFTWARE ENGINEERING

Tools

- Data transformation engine/ETL tool
- Data virtualization server
- Enterprise service bus
- Business rules engine
- Data and process modeling tools
- Data profiling tools
- Metadata repository

A list of the top data integration tools

https://deevita.com/top-data-integration-tools/

KNOWLEDGE & SOFTWARE ENGINEERING