



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# IT4142E
# Introduction to Data Science

## Chapter 3: Data cleaning and integration

Lecturer:

Muriel VISANI: murielv@soict.hust.edu.vn

Acknowledgements:

Khoat Than
Viet-Trung Tran

Department of Information Systems
School of Information and Communication Technology - HUST

# Contents of the course

- Chapter 1: Overview
- Chapter 2: Data scraping
- Chapter 3: Data cleaning, pre-processing and integration
- Chapter 4: Introduction to Exploratory Data Analysis
- Chapter 5: Introduction to Data visualization
- Chapter 6: Introduction to Machine Learning
  - Performance evaluation
- Chapter 7: Introduction to Big Data Analysis
- Chapter 8: Applications to Image and Video Analysis

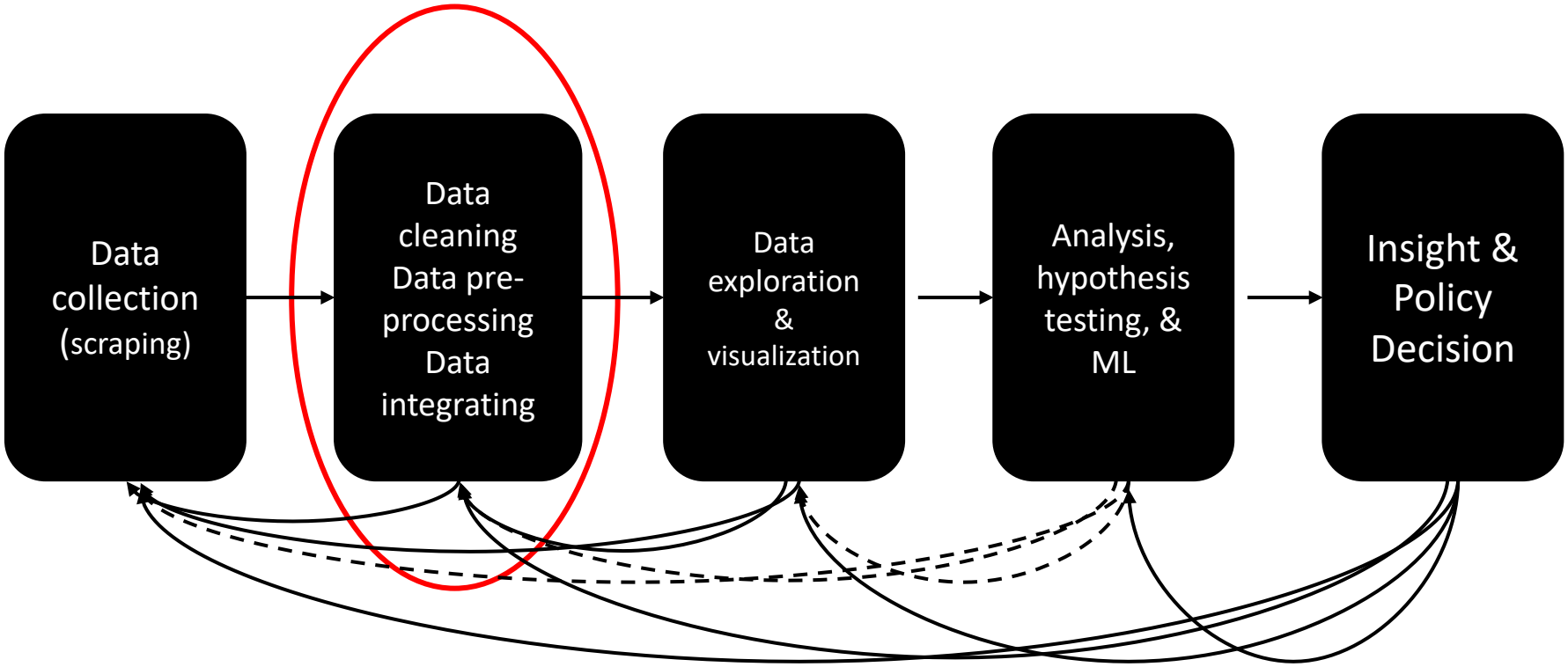
# Outline

- **Chapter 3: Data cleaning, processing and integration**
  - **Part1: data integration**
    - What is data integration?
    - Decision Support Systems (DSS)
    - Different possible architectures of a DSS
    - Technical solutions for data integration
    - Focus on some solutions
      - ETL for real architecture using SQL Server
      - Integration for virtual / remote architecture using Apache NIFI
  - Summary
  - Homework

# Goals of this chapter

| Goal | Description of the goal |
|---|---|
| **M1** | **Understand and be able to design and manage the systems which are based on Data Science (DS)** |
| M1.1 | Identify and understand the components of the systems based on DS |
| M1.2 | Identify, compare, and categorize the data types and systems in practice |
| M1.3 | Be able to design systems based on DS in their future organizations |

# Recall: insight-driven DS methodology

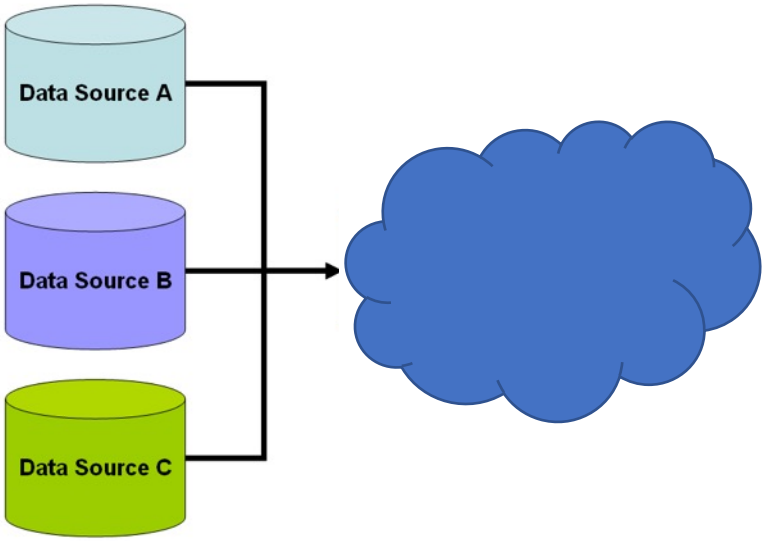


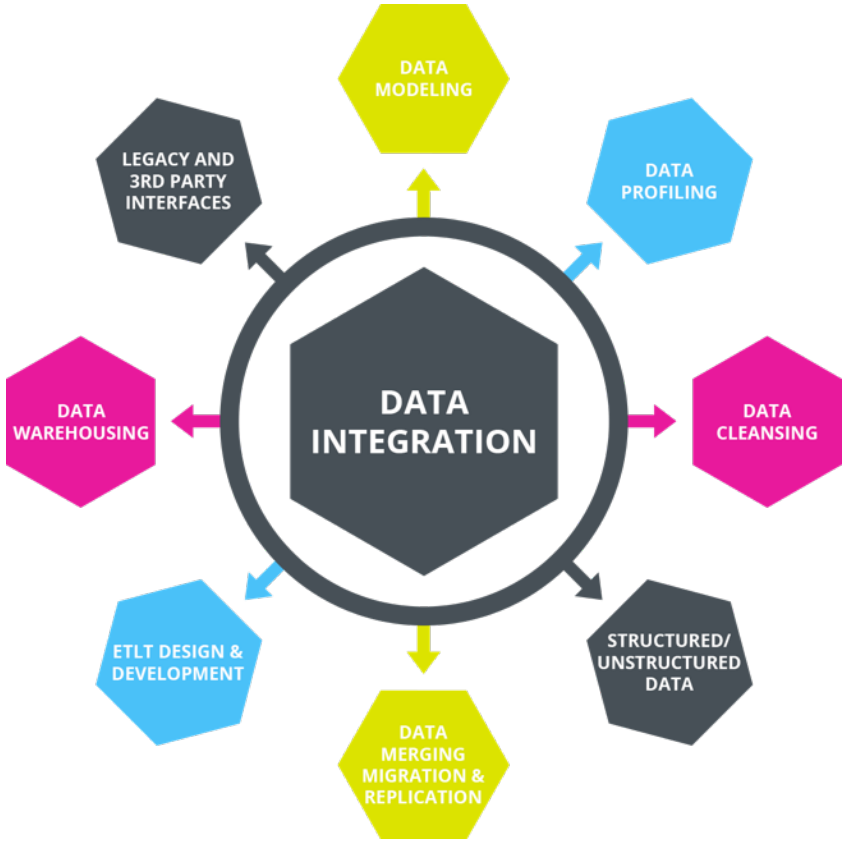
# What is data integration?

# Data integration

- Provide uniform access to data available in multiple, autonomous, heterogeneous and distributed data sources

Basic illustration of data integration



# Why data integration?

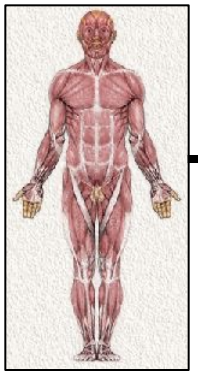
- To facilitate information access and re-use through **a single information access point**
- Data from different complementing information systems is **to be combined to gain a more comprehensive basis** to satisfy the final user's need
  - Improve decision making
  - Improve customer experience
  - Increase competitiveness, streamline operations
  - Increase productivity
  - Predict the future

# Data integration challenges

- Physical systems
  - Various hardwares, standards
  - Distributed deployment
  - Various data format
- Logical structures
  - Different data models
  - Different data schemas
- Business organization
  - Data security and privacy
  - Business rules and requirements
  - Different administrative zones in the business organization

# Decision Support Systems (DSS)

# Human decision process



Internal Stimuli

decision

External stimuli

Memory

- Example: fever
  - I am cold (internal stimuli)
  - Is the temperature low? (external stimuli)
    - If yes => I am looking for a pull-over (memory)
    - If no => I am going to the doctor
      - I may have fever (memory)

# Decision Support System

- An information only has value if it is positioned in its **context** and by taking into account the **history**
- Problem: when it is about business, the human brain is not enough to remember the huge masses of data
- Nowadays, about 25% of the working time of a decision-maker is dedicated to information searching, accessing and formatting
  - ⇒ Need to use a *decision support system (decision information system)* in order to reduce that loss of time

# Computer-supported decision process

Enterprise departments

Operational data

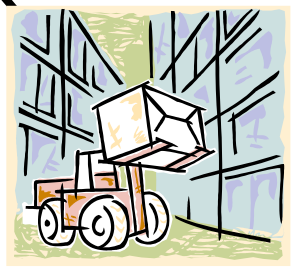
*Day-to-day internal data*

Decision support system

Decision-maker

*Day-to-day external data*

Market trends, demographical, geographical, sociological information…



Historical data

*Internal data*

(from the enterprise departments)

*External data*

# Decision Support System: definition

- A Decision Support System (DSS) serves to analyze data...
  - Extracted from the **operational information system** (internal data)
    - From the database, ERP, etc.
  - Enriched with **external data**
  - But the DSS is in general **distinct from the operational information system**, for two main reasons
    - Performance
      - To avoid monopolizing the operational system for queries which are in general very time-consuming
    - Readability and ease of use for the user

# Decision Support System: definition

- The success of Decision Support Systems is made possible thanks to the technological progresses:
  - Important increase in the storage capacity
    - Possible to store huge masses of data (historical data)
  - Introduction of parallel techniques
    - Possible to process and query these huge masses of data

# Decision Support System: definition

- **Decision Support System** =
  - Set of data +
  - Set of **tools**…
    - To collect the data – clean it and integrate it (make it uniform)
    - To organize the data according to the final user's needs
    - To analyse the data (data mining tools)
    - To communicate and present the data to the user (restitution tools)

# Different possible architectures of a DSS

# Types of architectures for a Decision Support System

- The DSS may be conceived following three types of architectures
- Most of the earliest implementations used the **real architecture**…
- ... However, other architectures are now possible: **virtual** and **remote** architectures
- So, today, we will discuss the 3 architectures:
  - Real
  - Virtual
  - Remote

# Real architecture

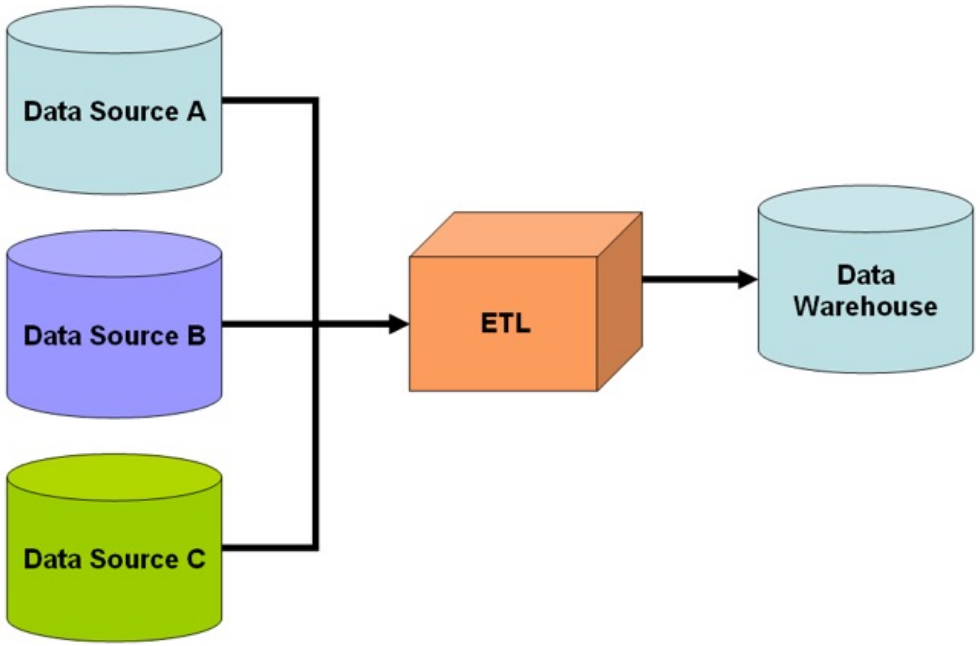
- Factually, the organization of the data from the operational database is not adapted to a decision objective

⇒ The DSS implements another database (distinct from the operational database): the Data Warehouse (DW)

  - The DW is fed within regular intervals of time
  - Data is pre-processed (extracted and transformed) before being loaded in the data warehouse
    - ETL procedure: Extract, Transform, Load

# Decision Support System: real architecture (using DW)

- Simplified view of the real architecture



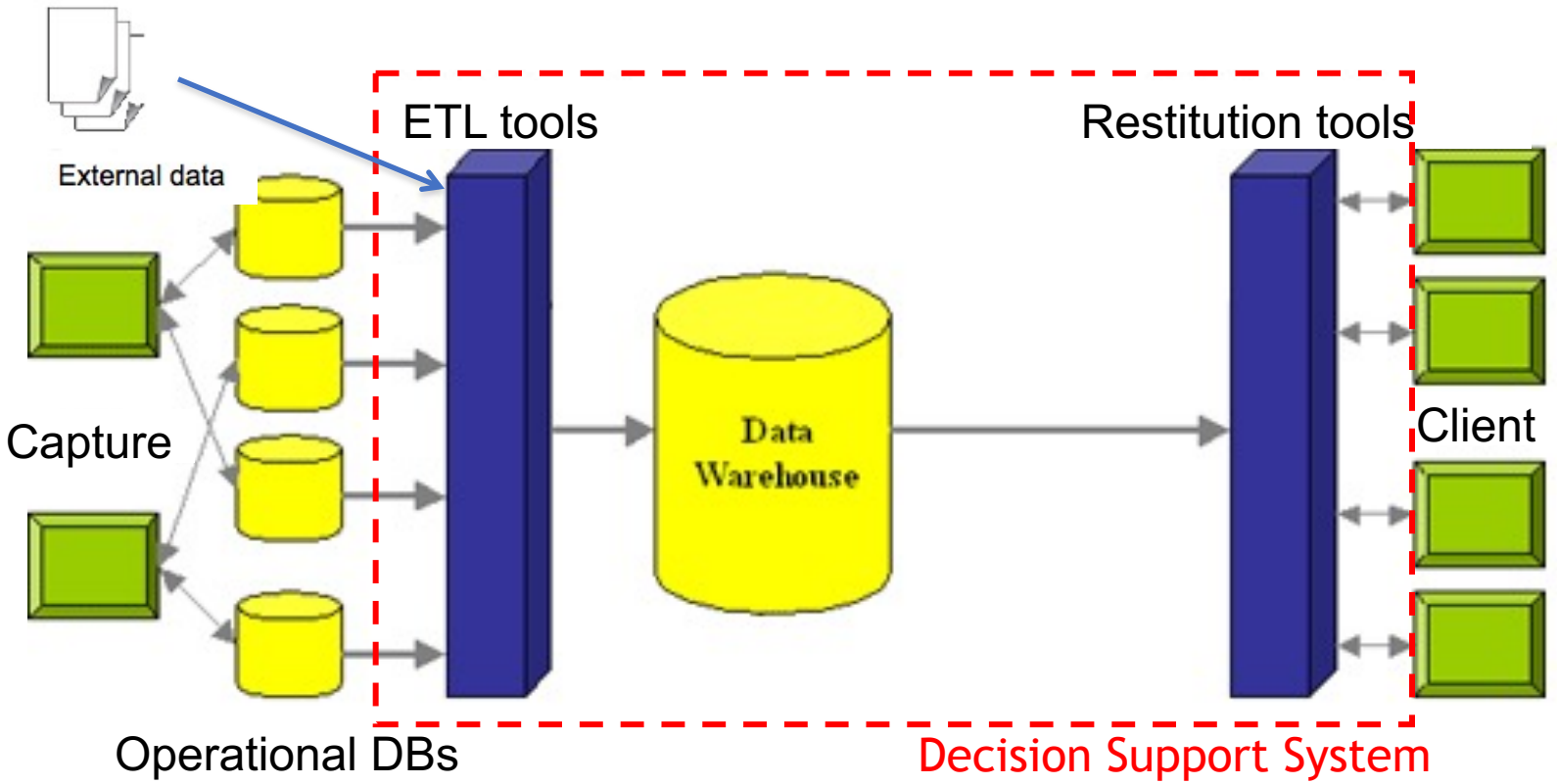
# ETL process

- In the real architecture, ETL=data integration
- With real architectures, 70-80% of BI (DI or DW) project relies on the ETL process
- ETL = Extract – Transform – Load
- Extract
  - Get the data from source system as efficiently as possible
- Transform
  - Perform calculations on data
- Load
  - Load the data in the target storage

# Why is ETL so important?

- Adds **value** to data
  - Removes mistakes and corrects data
  - Documented measures of confidence in data
  - Captures the flow of transactional data
  - Adjusts data from multiple sources to be used together (conforming)
  - Structures data to be usable by BI tools
  - Enables subsequent business / analytical data processing

# Decision Support System:
# real architecture (using DW)



# Operational DB *vs.* Data Warehouse

- Technical differences
  - Operational Databases
    - Operational DBs generally **store** the data using **OLTP** engines
      - OLTP = Online Transaction Processing
      - Under strict normalization rules (3rd normal form…)
    - Operational DBs are generally **queried**
      - Either ad-hoc using **SQL**, or using **pre-defined procedures**
  - Data warehouses
    - DWs can either **store** the data using
      - **OLAP** engines (cubes)
        - OLAP = Online Analytical Processing
      - OLTP engines, but with relaxed normalization rules (redundancy is OK if it speeds up the frequent queries)
    - DWs are generally **queried**
      - Either by using specific, **user-friendly** *ad-hoc* querying tools, or pre-defined **reports** (pdf, HTML…)

# Operational DB *vs.* Data Warehouse

- In short…
  - The users of the **operational DB** make the enterprise work day by day
  - The users of the **data warehouse** observe how the enterprise works in order to analyse and/or to make the right decisions

| | Operational DB | Data warehouse |
|---|---|---|
| Typical operation | Inserting, updating | Complex querying |
| Type of access | Read write | Read |
| Level of analysis | Elementary | Global |
| Quantity of information per query | Small | May be huge |
| Size of the database | Small (max a few GO) | Important (until a few TO) |
| Age of the data | Recent | Historical |

# Getting the data into the DW

- How to load data into the DW?
  - Scripts in linux shell, perl, python, …
  - sqlldr + SQL
  - Hardcoded in Java, C#, C
  - In-house built ETL tool
  - Off-the shelf ETL tool
- Aspects to be kept in mind
  - Manageability
  - Maintainability
  - Transparency
  - Scalability
  - Flexibility
  - Complexity
  - Auditing
  - Job re-startability
  - Testing

# Real architecture

☺ Is easy and intuitive to manipulate

☺ Provides data history and traceability

☺ Is efficient despite the huge masses of data (OLAP)

☺ Is GLOBALLY and logically coherent

☹ Additional storage cost

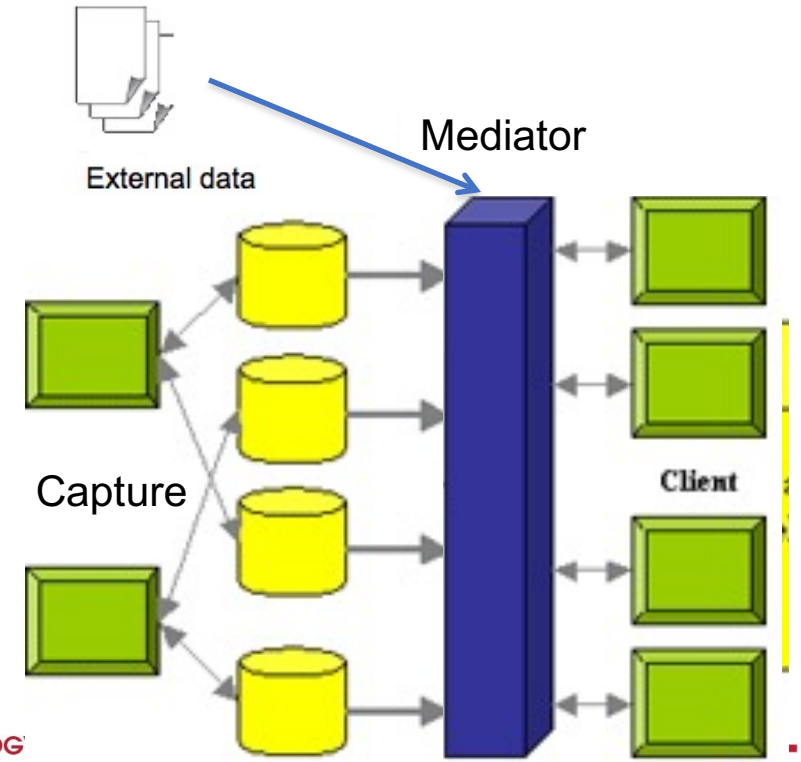
☹ Need for a « Extract, Transform, Load » (ETL) procedure (**data integration)**

- Also called feeding procedure…
- Role: take the data from operational and external sources and process it to make it fit the DW data model
- Performed at regular intervals of time (every day/week…)
- ☹ Difficult and complex step
- ☹ Difficult to set its parameters (update frequency…)
- ☹ Lack of flexibility towards changes in the sources
- ☹ Potential source of errors
- ☹ Lack of real-time access to the data which have just been inserted in the operational system

# Virtual architecture

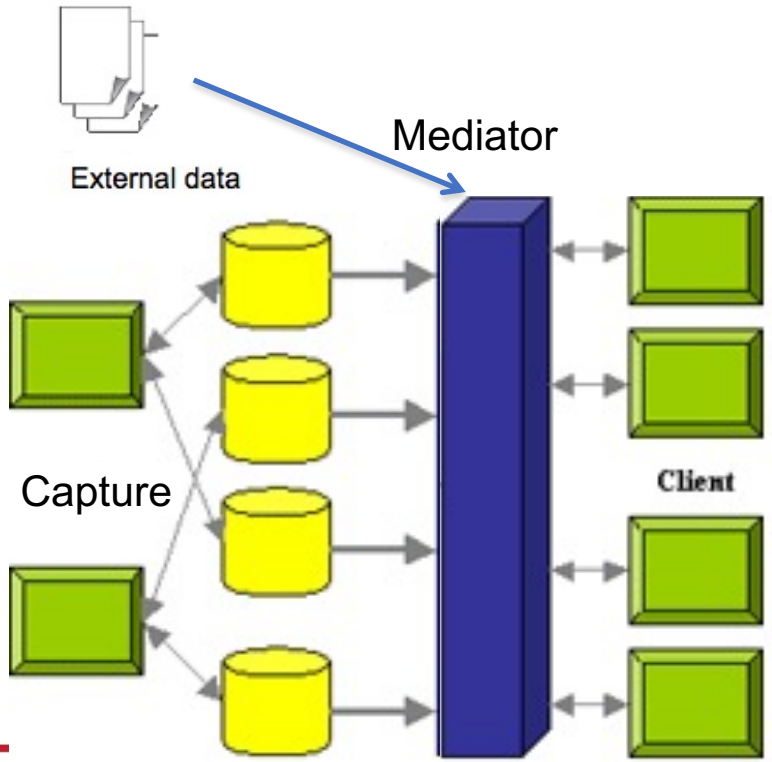
- **Virtual** architecture (*a.k.a.* federated architecture)
  - Data remains in the source databases of the operational DBs
  - No decisional database -> **no DW**
  - Data can be made visible by **data virtualization** tools

  (for instance a “mediator “ or an “Enterprise Information Integration” (EII) tool)



# Virtual architecture

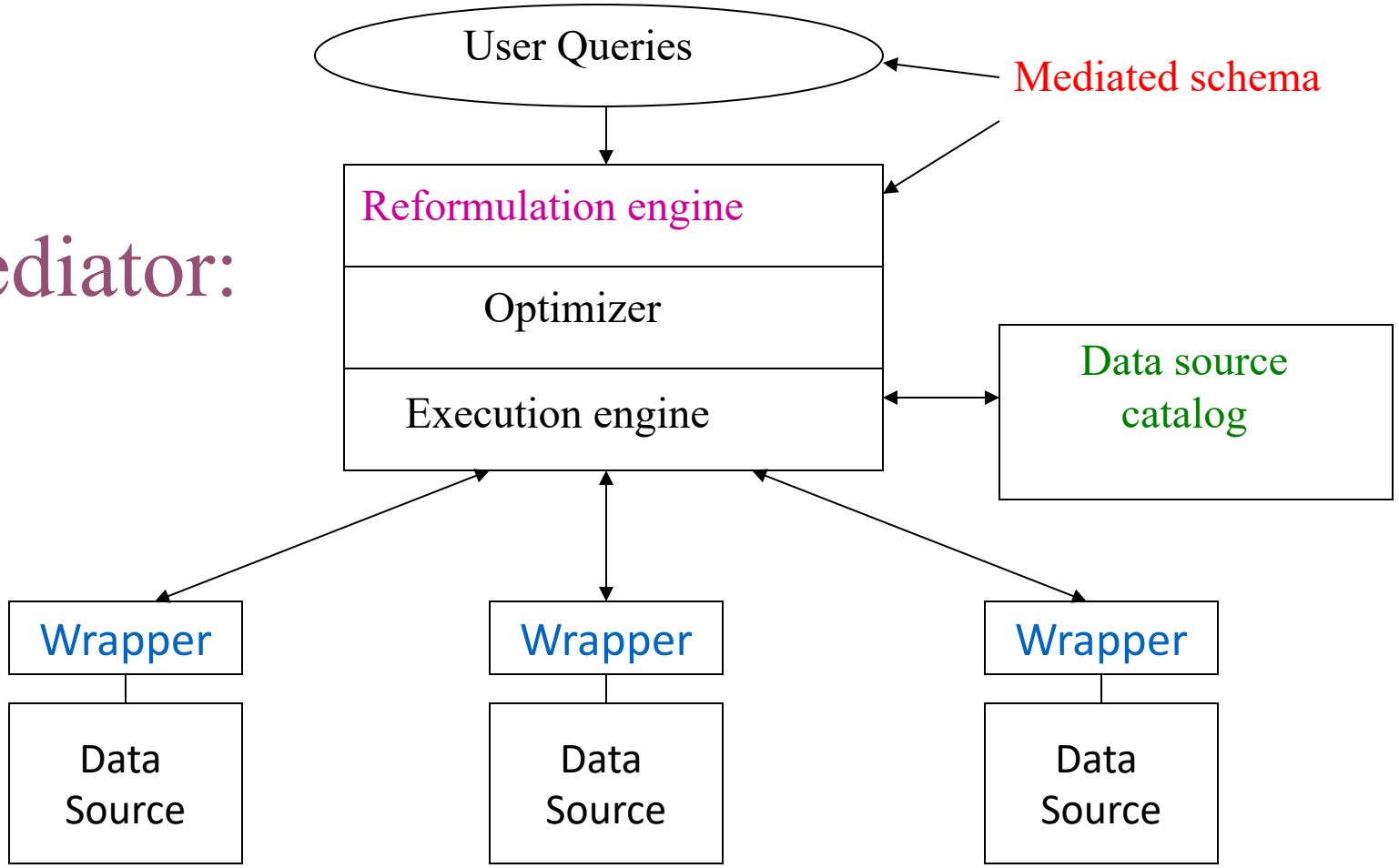
- **Mediator-based Virtual architecture**
- A mediator is a software system that offers a common query interface to a set of heterogeneous information sources
- We have:
  - A collection of data sources that are independent and autonomous
  - A virtual “database” is created which is accessed via the mediator
  - A **mediator** that creates the illusion on users of being interacting with a real database
  - Queries are posed and answered via the mediator



# Virtual architecture

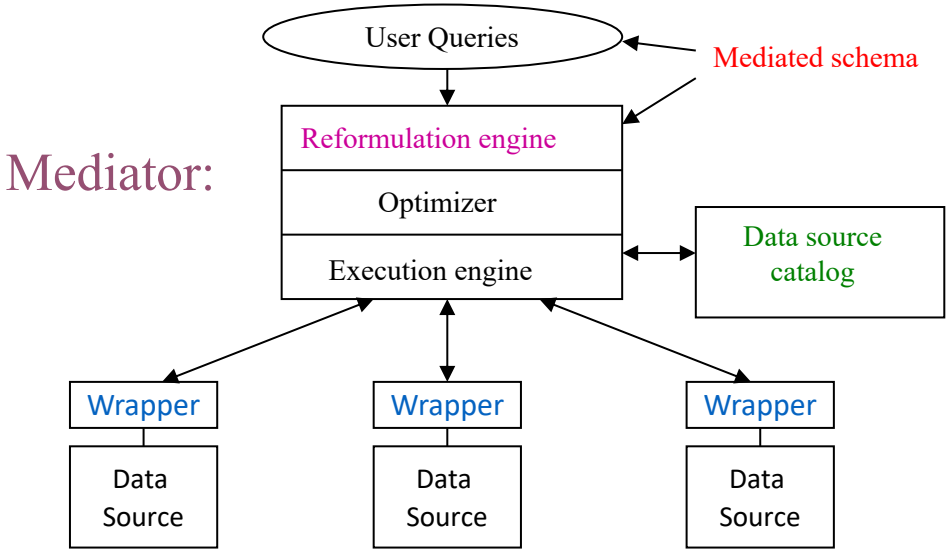
- Example of mediator

Mediator:



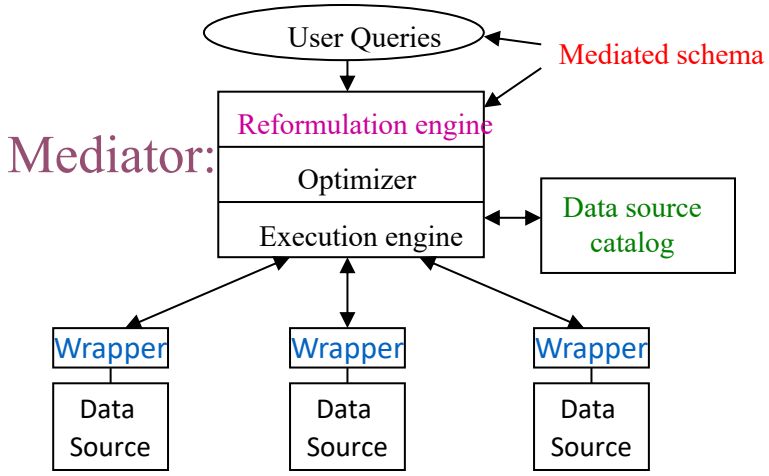
# Virtual architecture

- Leave the data in the data sources
- For every query over the mediated schema
  - Find the data sources(s) that have the data (probably more than one)
  - Query the data source(s)
  - Combine results from different sources if necessary



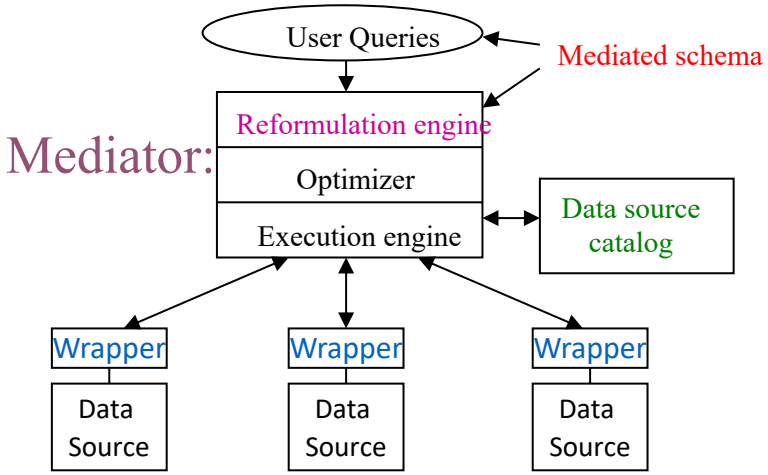
# Mediated schema

- **Mediated schema** (*a.k.a.* **global schema)** is the schema used to present and export the data
  - For example, if it is a relational schema, then it is a set of names for relations (tables), their attributes, etc.
    - like a regular relational DB schema
  - The DB "instance" corresponding to the mediated schema is **virtual**
    - Data is not stored in "tables" of the global schema, but in the sources
- User poses queries in terms of the relations in the mediated schema
  - The mediator will reformulate the query so as to get the information from the data sources
    - -> need for a data source catalog



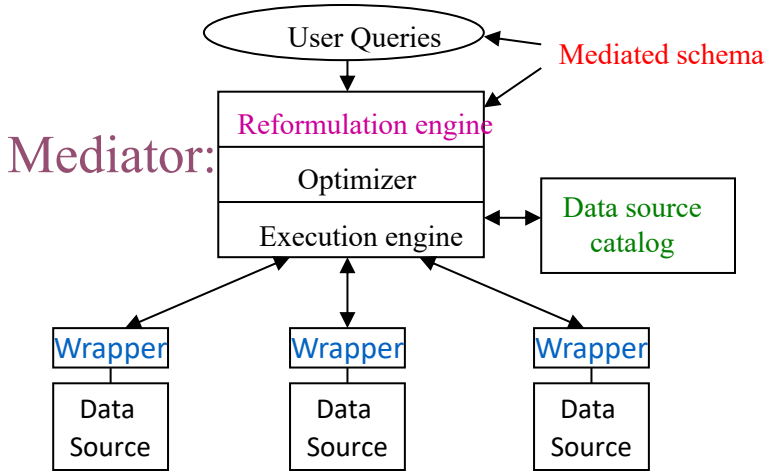
# Data Source Catalog

- Contains meta-information about sources
  - Logical source contents (books, new cars)
  - Source capabilities (can answer SQL queries?)
  - Source completeness (has all books?)
  - Physical properties of source and network
  - Statistics about the data (like in an RDBMS)
  - Source reliability
  - Mirror sources
  - Update frequency



# Wrappers

- Sources export data in different formats
- Wrappers are custom-built programs that transform data from the source native format to something acceptable to the mediator
- Example: from HTML to XML




HTML

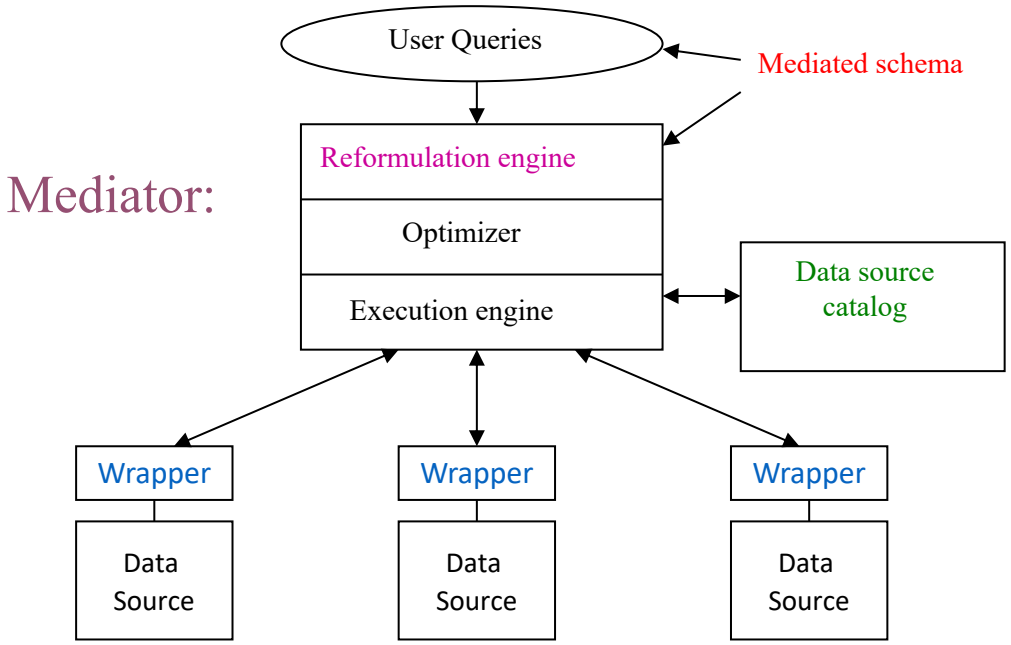
```
<b> Introduction to DB </b>
<i> Phil Bernstein </i>
<i> Eric Newcomer </i>
 Addison Wesley, 1999
```

XML

```
<book>
<title> Introduction to DB </title>
<author> Phil Bernstein </author>
<author> Eric Newcomer </author>
<publisher> Addison Wesley </publisher>
<year> 1999 </year>
</book>
```

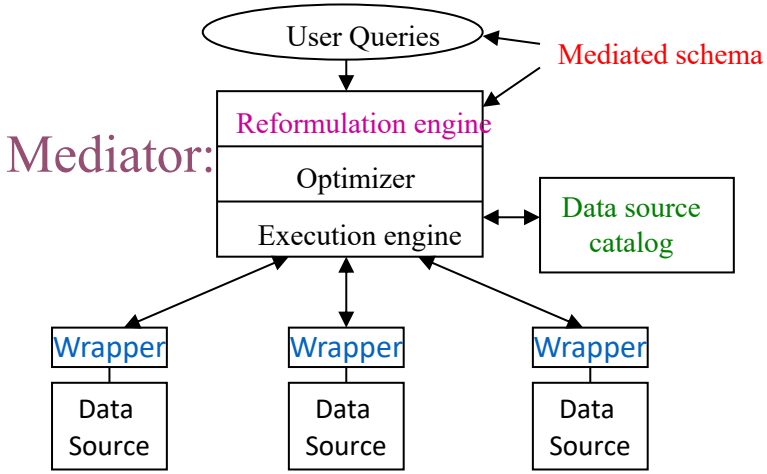
# Wrappers

- Can be placed either at the source or at the mediator
- Maintenance problems
  - Wrappers have to change if source interface changes



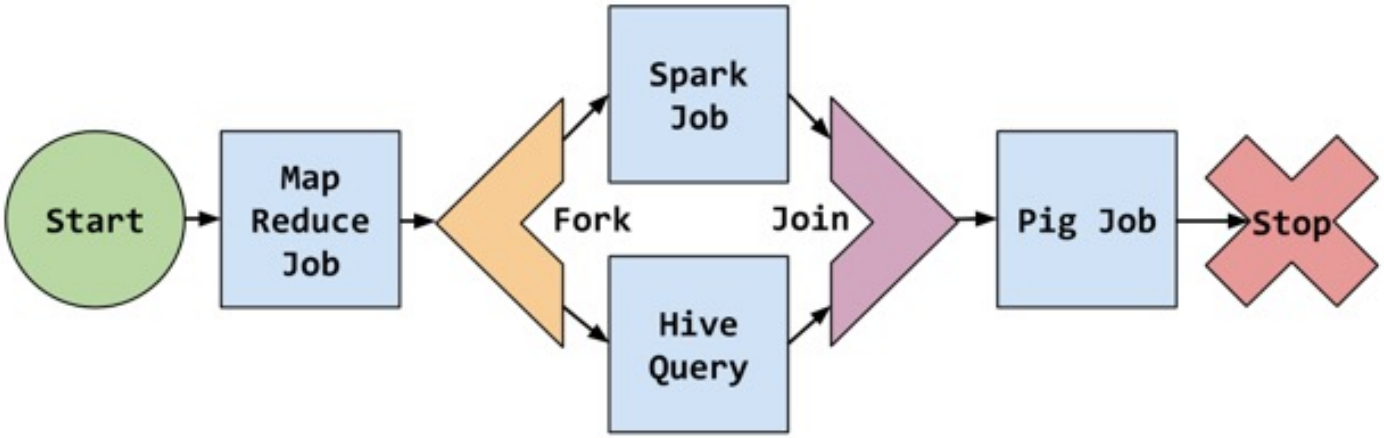
# Reformulation engines

- Reformulation: Queries over the mediated schema have to be rewritten as queries over the source schemas
- Different possible strategies to perform re-formulation
  - 1- Workflow management approach
  - 2- Web service approach



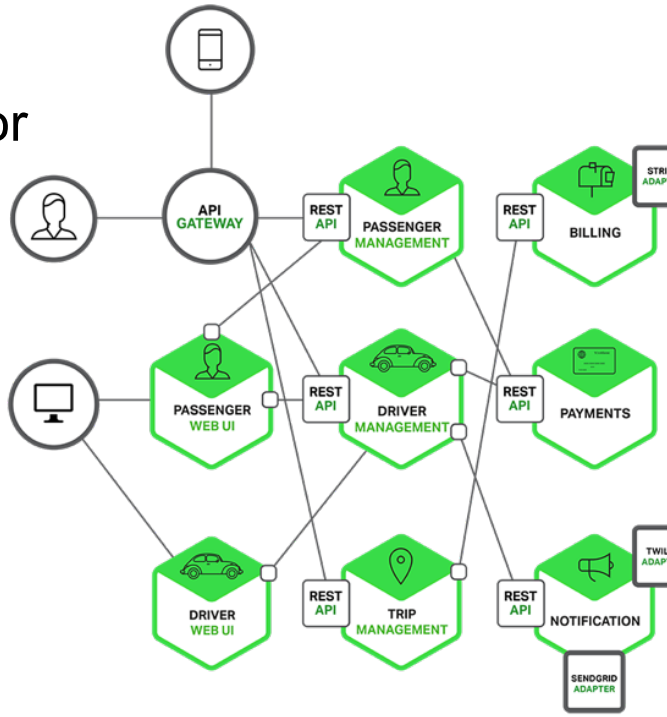
# 1- Workflow management approach

- Represent an integration-by-application approach
- Example, using Hadoop and Spark
  - See chapter 7



# 2- Web service approach

- Performs integration through software components (web services)
  - These web services support machine-to-machine interaction by XML-based messages
- Depending on the solution, it might offer
  - a uniform data access approach, or
  - a common data access for later access or
  - application-based integration



# Virtual architecture

- **Virtual** architecture

☺ No additional storage cost (compared to operational database)

☺ Less maintenance, no additive cost related to system hardware, software (except the EII)…

☺ Real-time access to the data

☺ No need for an ETL procedure

☹ Data is not cleaned nor pre-processed

☹ Data is not historical (in general)

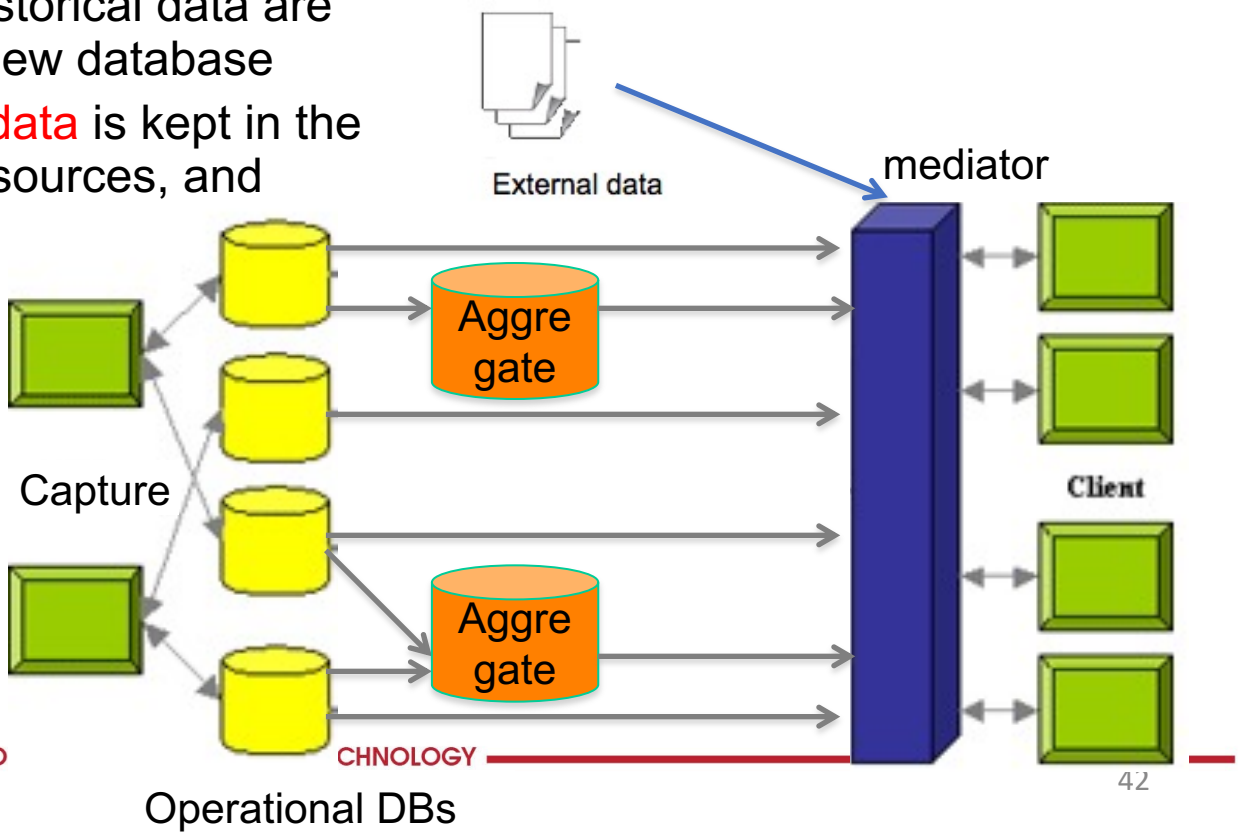
☹ Complex queries may disrupt operational systems

# Virtual architecture

- Data virtualization is a **subset** of data integration
  - Commonly used for business intelligence and decision support systems…
  - …. but also for service-oriented architecture data services, cloud computing, enterprise search, master data management...
  - Google Book about virtualization:
    - « Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses »
      - Rick F. van der Lans, Elsevier, 2012 - 275 pages
  - Another on-line course on the topic:
    - http://people.scs.carleton.ca/~bertossi/talks/datIntegr07.pdf

# Remote architecture

- **Remote** architecture
  - Trade-off (**compromise**) between real and virtual architectures:
    - Only aggregates (summarized data) and historical data are stored in a new database
    - All detailed data is kept in the operational sources, and virtualized



# Remote architecture

- **Remote** architecture
  - ☺ Is easy and intuitive to manipulate
  - ☺ Provides data history and traceability
  - ☺ Is efficient despite the huge masses of data (OLAP)
  - ☺ Only a small additional storage cost
  - ☺ Less maintenance than with real architecture
  - ☺ Real-time access to the data
  - ☺ Detailed data is not cleaned nor pre-processed
  - ☹ Complex queries on data sources may disrupt operational systems
  - ☹ Is not “physically” globally and logically coherent
  - ☹ Can be affected by changes in the operational sources

# Technical solutions for data integration

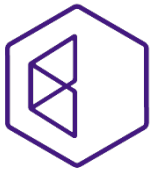
# ETL market (real architecture)

# Free tools for ETL (real architecture)

- For **ETL**
  - Pentaho Data Integration (Kettle), Talend Open Studio, ….



- For creating **reports**
  - BIRT, JasperReports, Pentaho Report Designer…




- For data **visualization**
  - Mondrian, JPivot, Palo and Jpalo, Tableau Public….
- For data **mining**
  - R, Weka, Orange, pandas/scikit-learn in Python…

# Free web-service mediators (virtual & remote architectures)

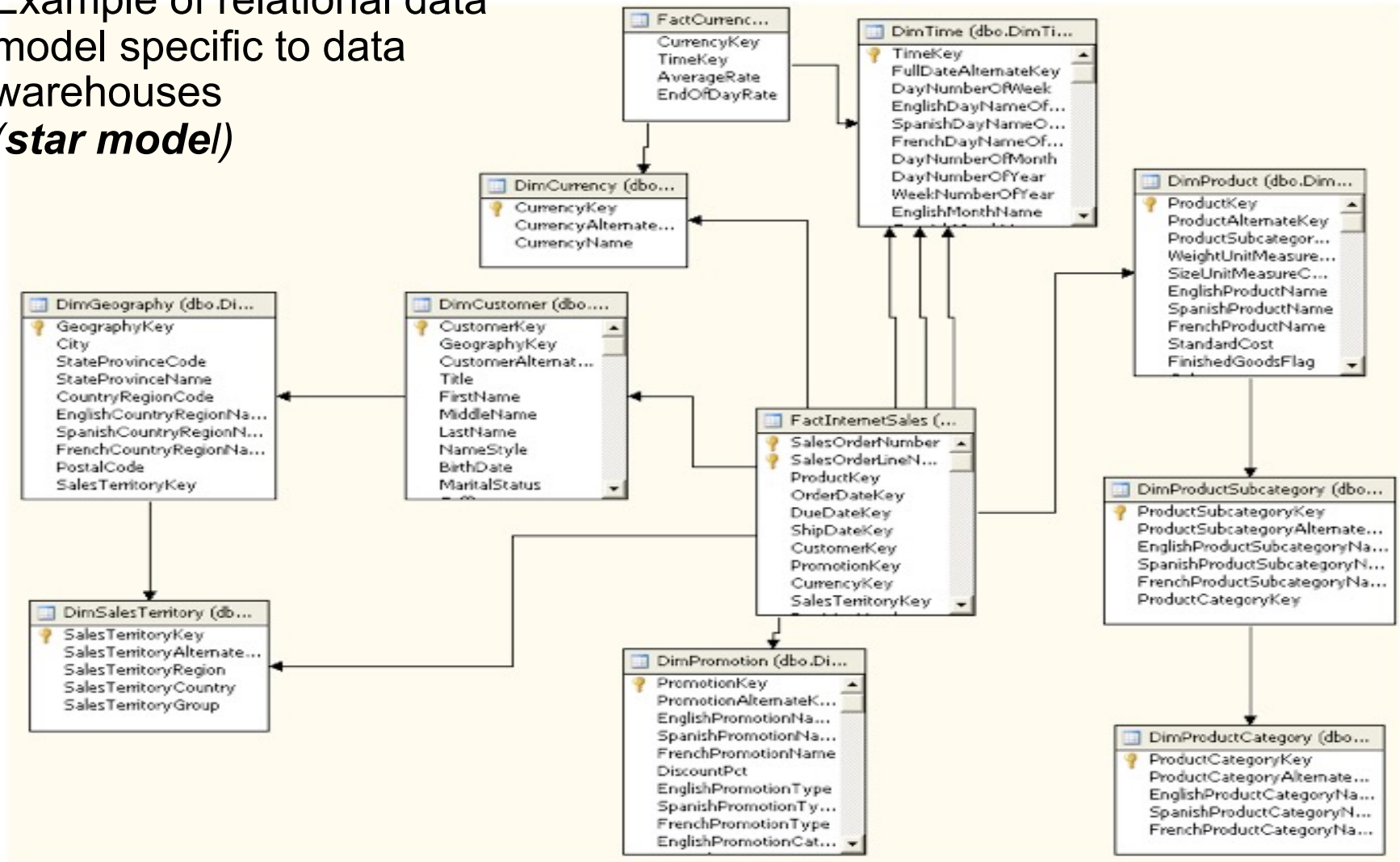
- New mediators constantly appear:
  - https://analyticsindiamag.com/top-9-etl-tools-for-data-integration-in-2020/
- One ohe most used web-services for data integration is Apache NIFI

# Focus on some solutions

ETL for real architecture using Microsoft SQL Server

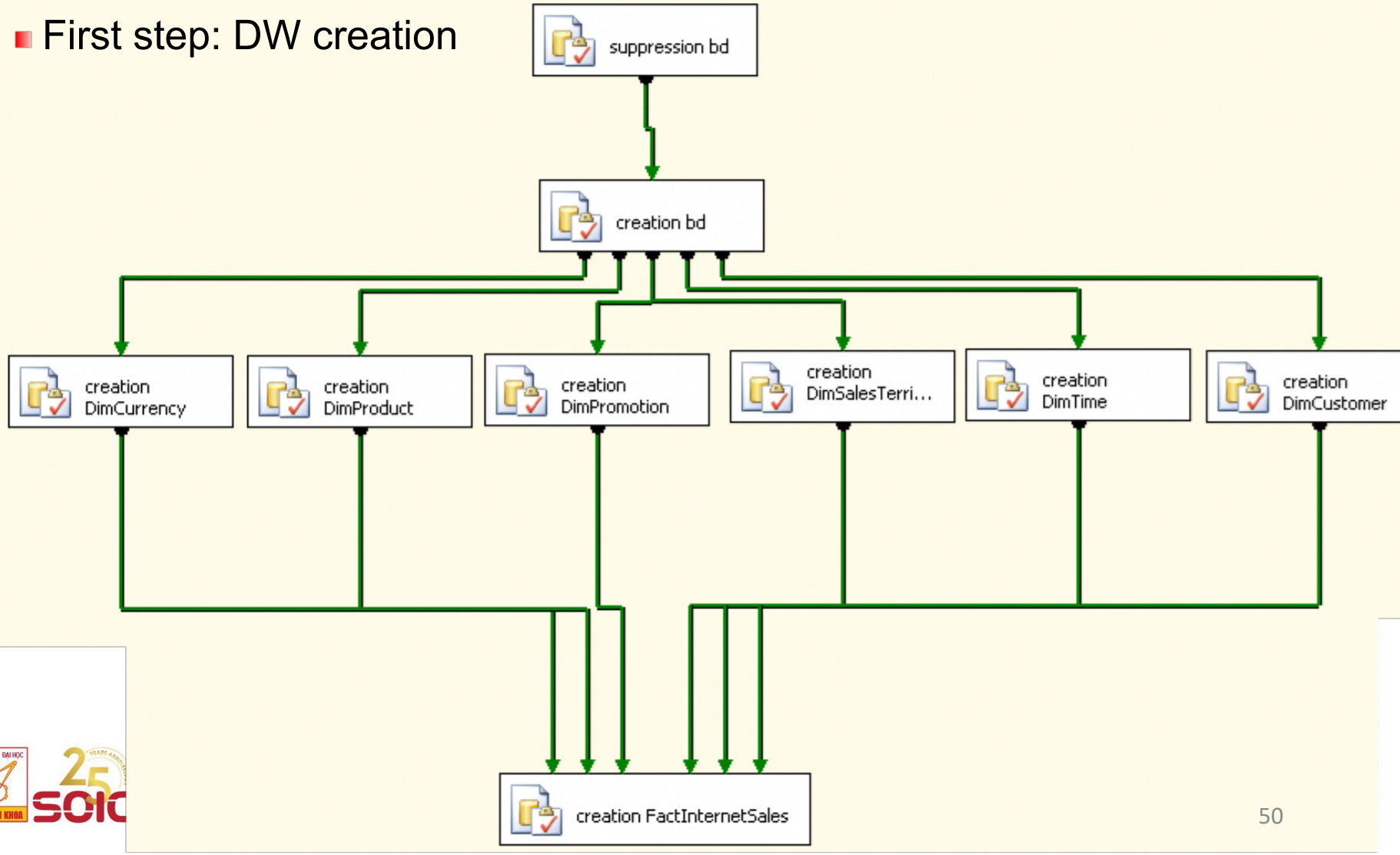
# Microsoft SQL Server Screenshots

- Example of relational data model specific to data warehouses
  ***(star model)***



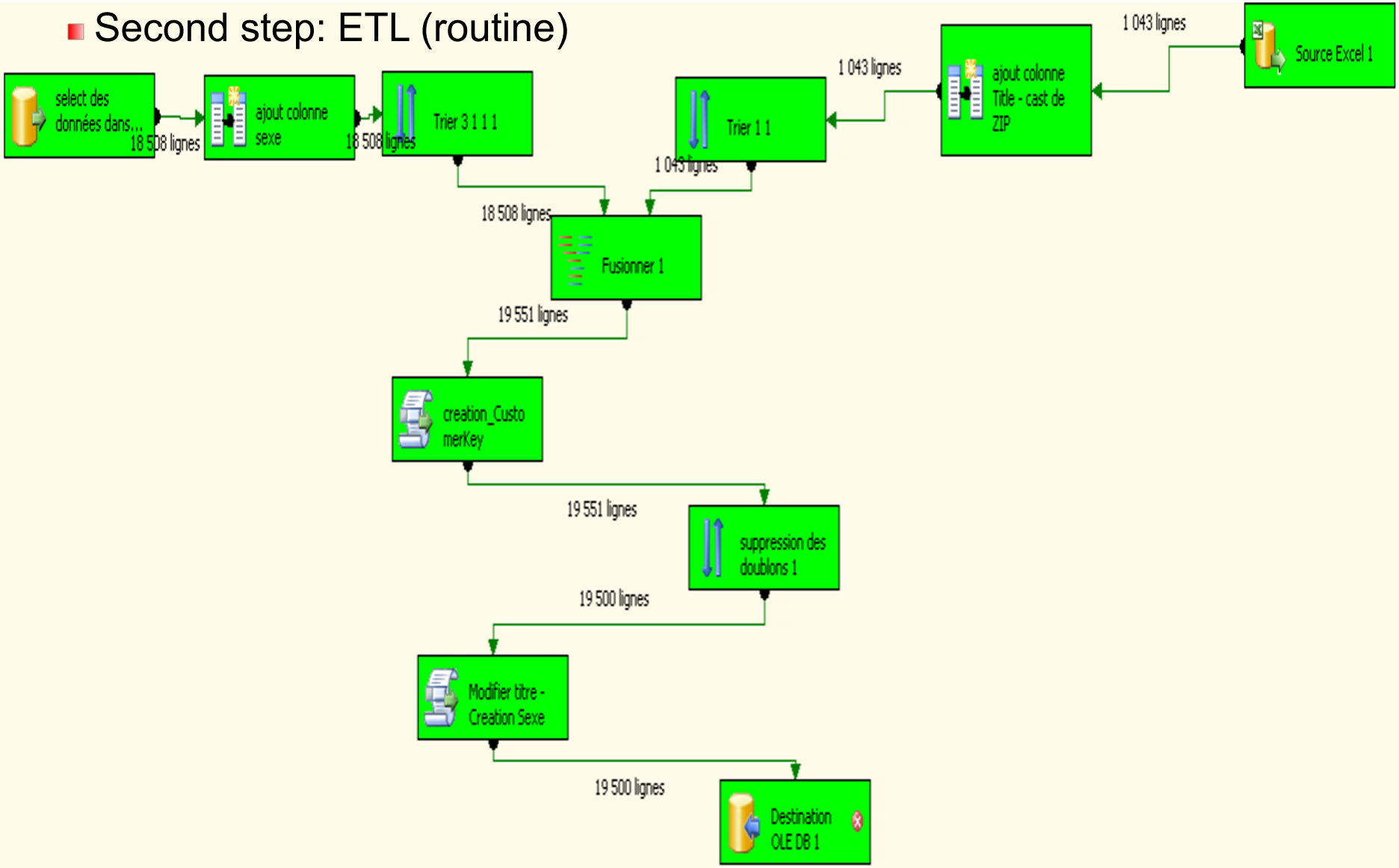
# Microsoft SQL Server Screenshots

- First step: DW creation



# Microsoft SQL Server Screenshots

- Second step: ETL (routine)



# Focus on some solutions

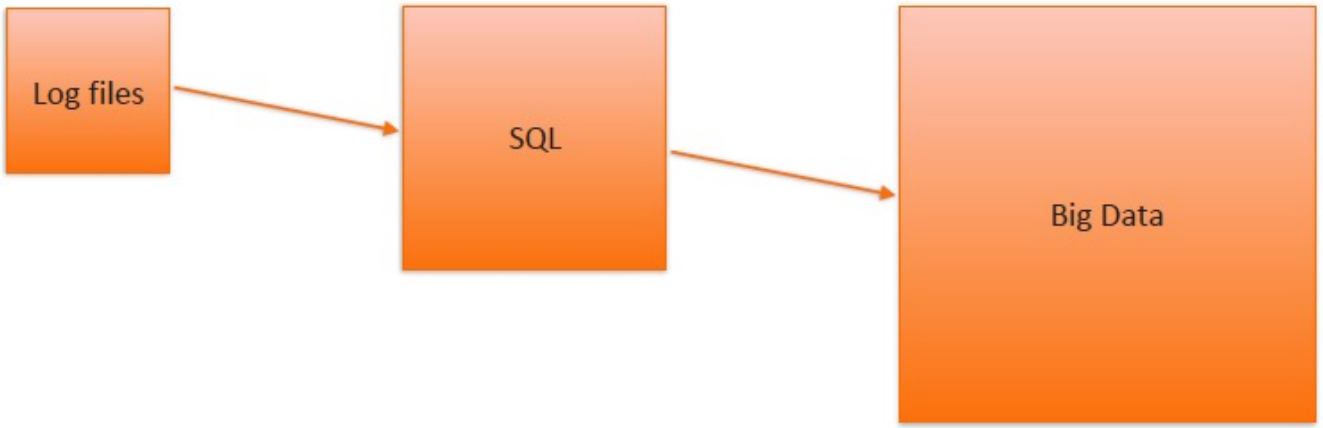
Integration for virtual / remote architecture using Apache NIFI

# What is dataflow?

- Moving some content from A to B
- The source content could be
  - Logs
  - HTTP
  - XML
  - CSV
  - Images
  - Video

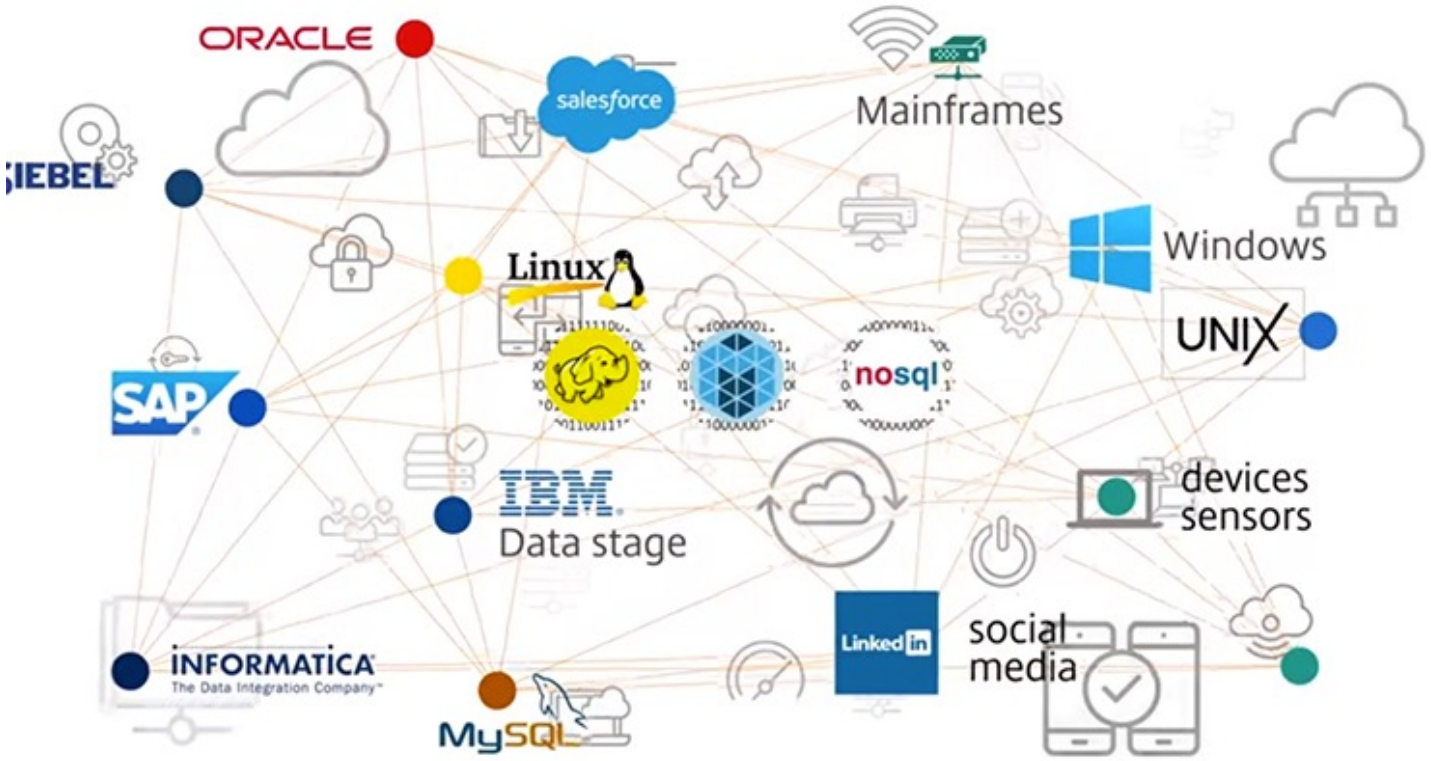
# Connecting data points is easy…

- Simple enough to write a process
  - Bash/Ruby/Python
  - SQL proc
  - etc.

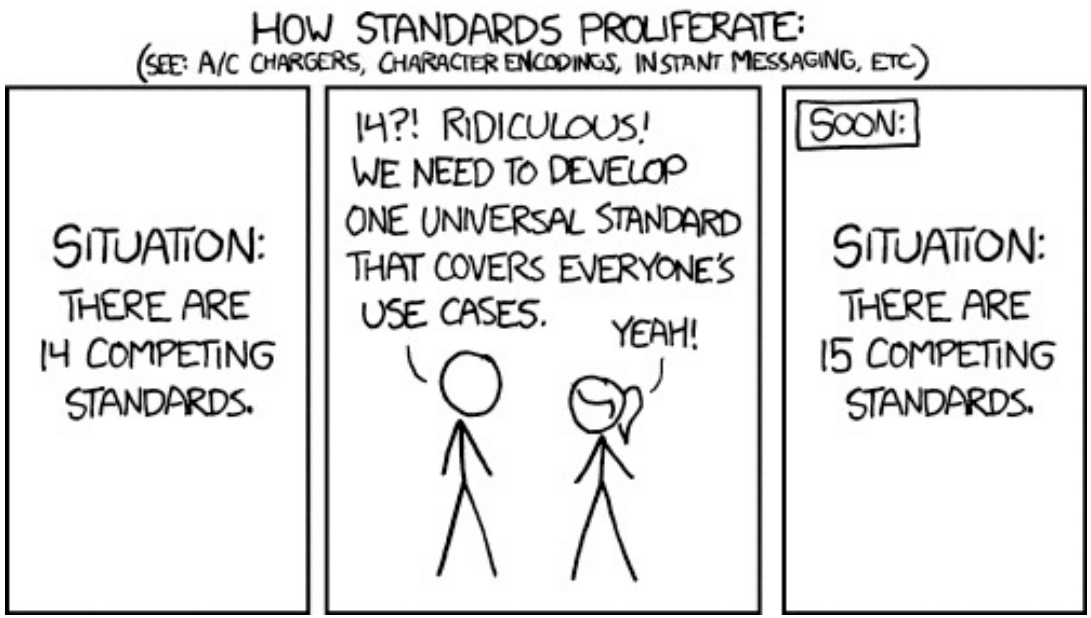


# … but, this approach doesn't scale

https://www.slideshare.net/LevBrailovskiy/data-ingestion-and-distribution-with-apache-nifi



# Moving data effectively is hard




Standards: http://xkcd.com/927/

# Dataflow challenges in 3 categories

- Data
  - Standards
  - Formats
  - Protocols
  - Veracity
  - Validity
  - Schemas
  - Partitioning/Bundling
- Infrastructure
  - “Exactly Once” Delivery
  - Ensuring Security
  - Overcoming Security
  - Credential Management
  - Network
- People
  - Compliance
  - “That [person|team|group]”
  - Consumers Change
  - Requirements Change
  - “Exactly Once” Delivery

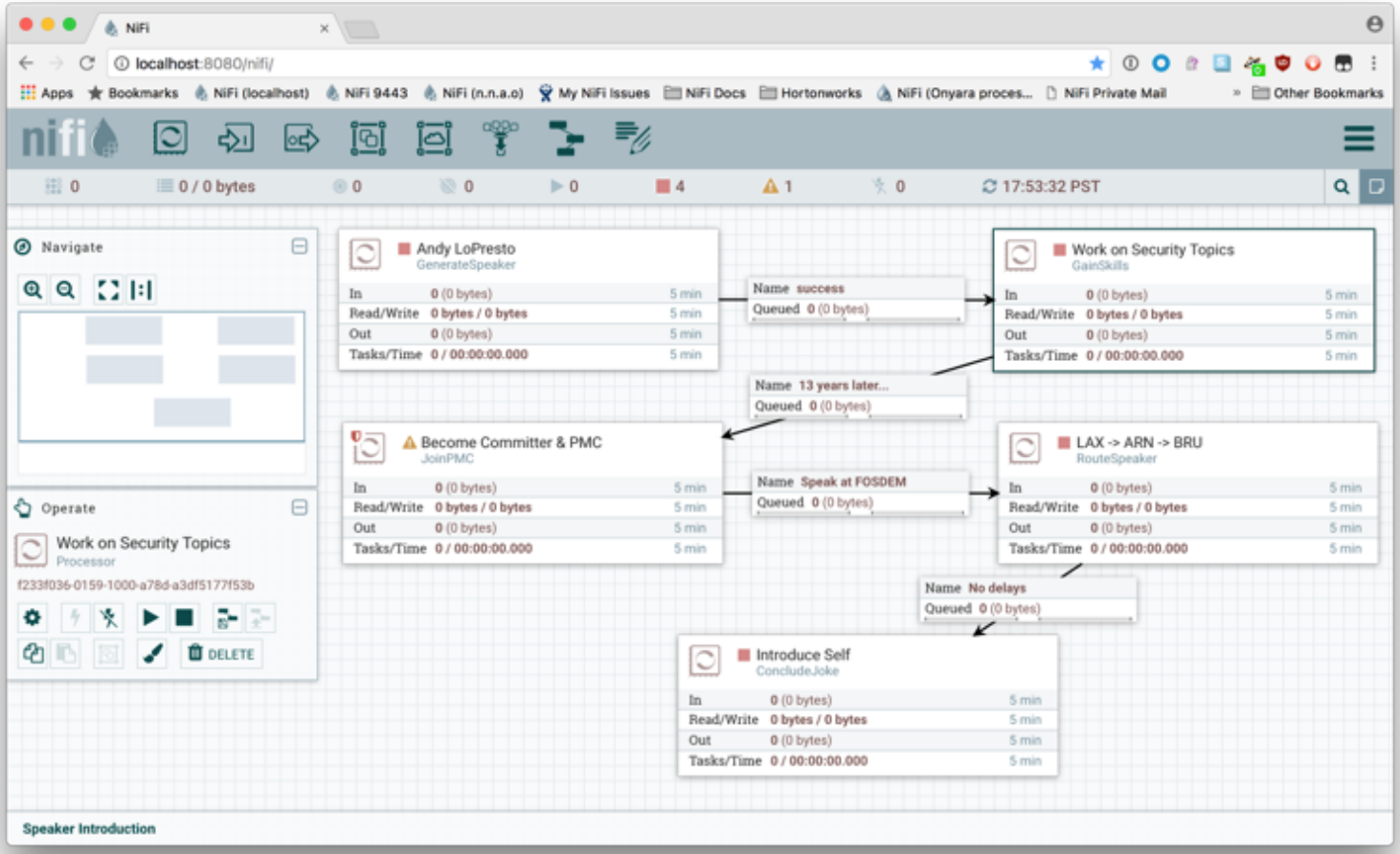
# NiFi is based on Flow Based Programming (FBP)

| FBP term | Niffi Term | Description |
|---|---|---|
| Information Packet | FlowFile | Each object moving through the system. |
| Black Box | FlowFile Processor | Performs the work, doing some combination of data routing, transformation, or mediation between systems. |
| Bounded Buffer | Connection | The linkage between processors, acting as queues and allowing various processes to interact at differing rates. |
| Scheduler | Flow Controller | Maintains the knowledge of how processes are connected, and manages the threads and allocations thereof which all processes use. |
| Subnet | Process Group | A set of processes and their connections, which can receive and send data via ports. A process group allows creation of entirely new component simply by composition of its components. |

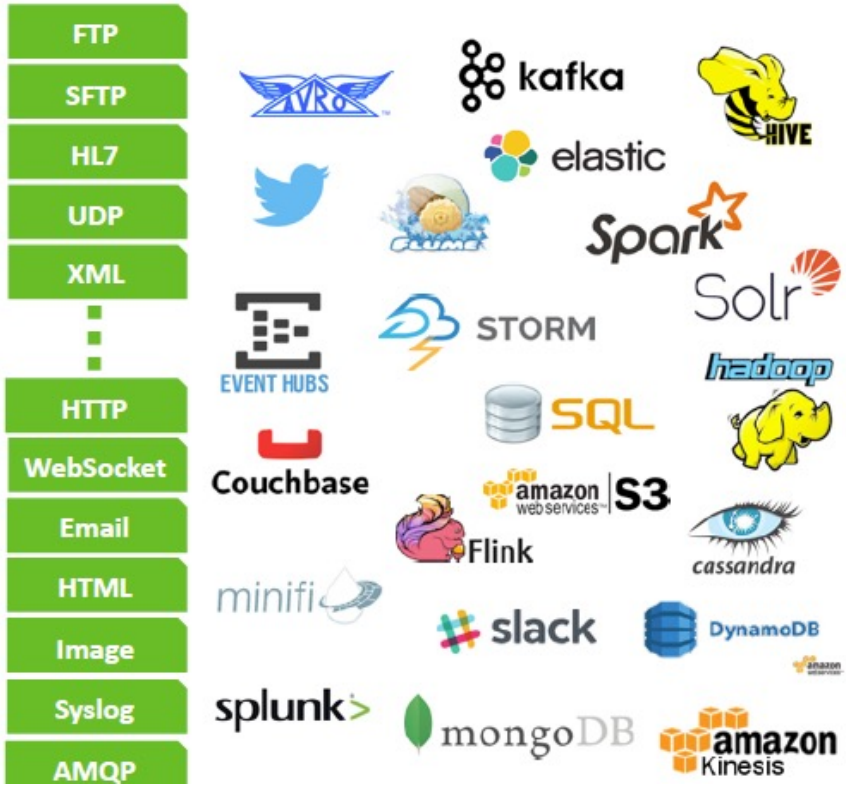
# Main characteristics of NIFI

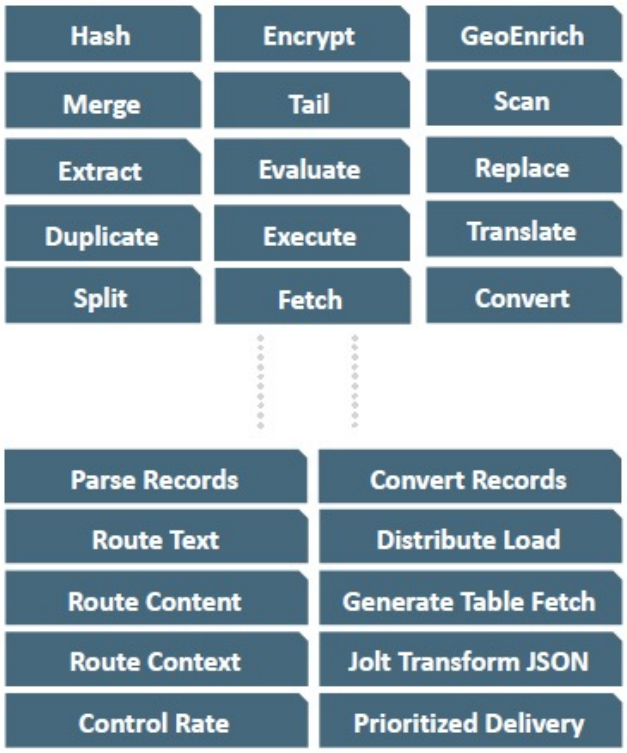
- Flow specific Quality of service (QoS)
  - Latency vs. Throughput
  - Loss tolerance
- Supports most data source formats
- Supports push and pull models
- Recovery/recording a rolling log of fine-grained history
- Visual command and control
- Flow templates
- Pluggable
- Designed for extension
- Clustering of Nifi instances

# User Interface



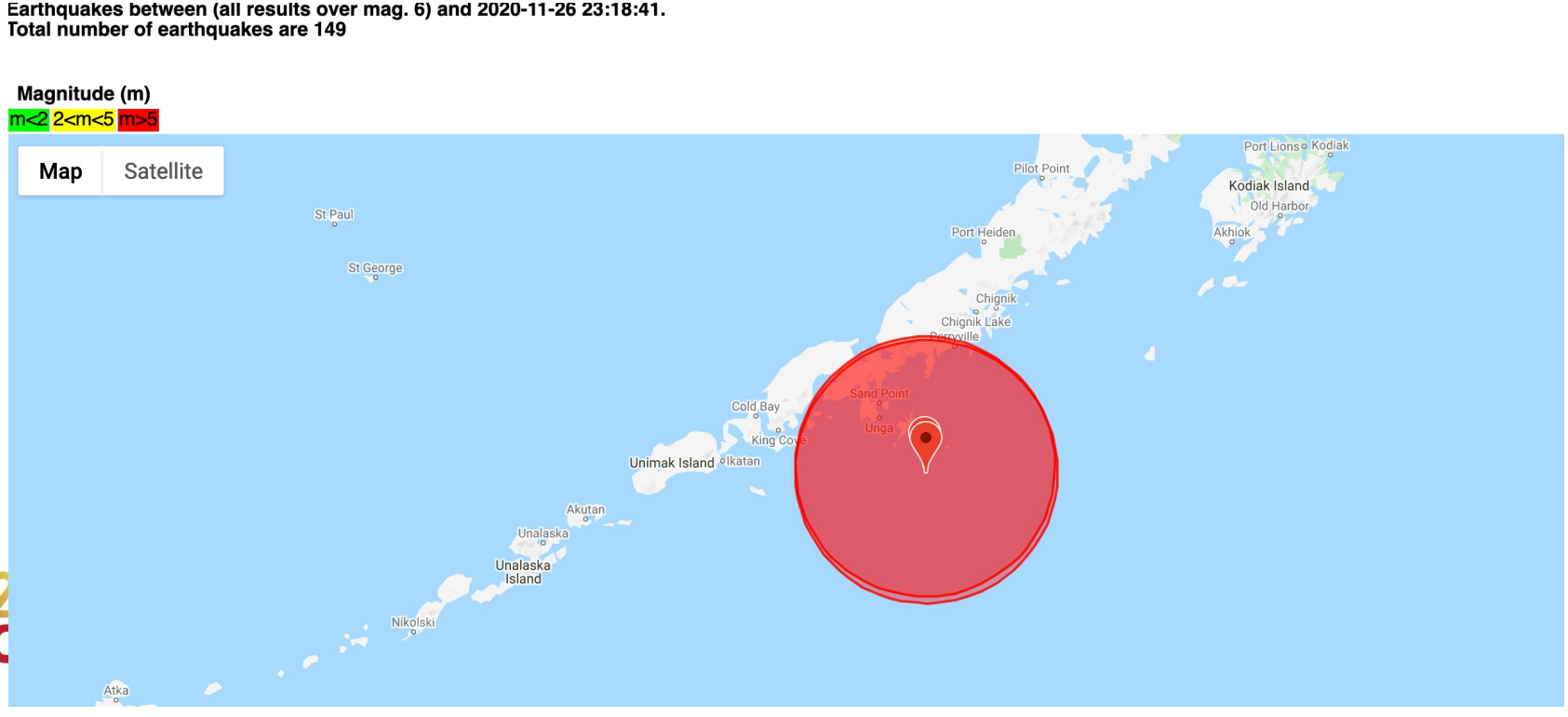
# Ecosystem Integration: 260+ Processors, 48 Controller Services



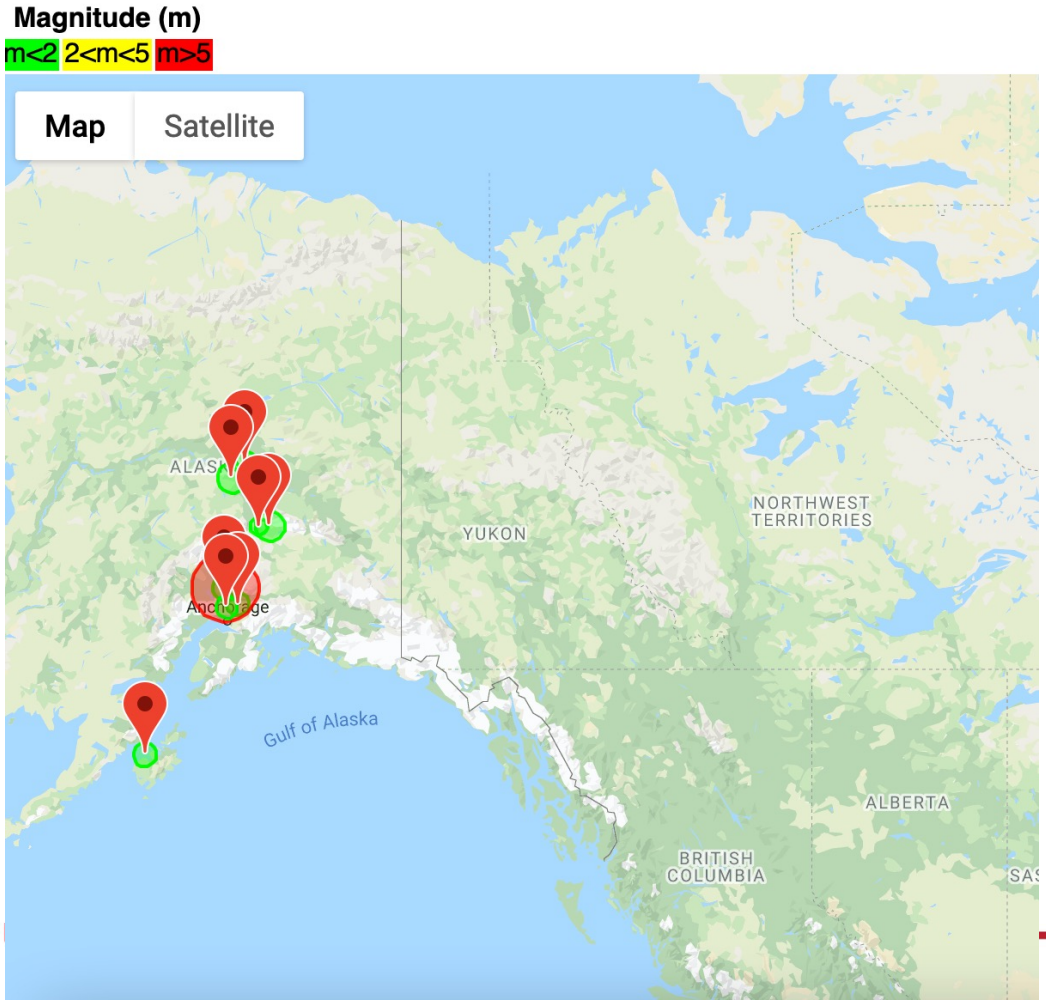
# Apache NIFI demo

- Demonstration
- How to access the demonstration website yourself and more information about the NIFI flows:
  - https://youtu.be/MARazprrNYA



# Apache NIFI demo

**Earthquakes between 2020-11-25T23:28:50 and 2020-11-26T23:28:50.**
**Total number of earthquakes are 504**



# Summary

# Summary

- You have learnt about
  - Decision Support Systems and data warehouses
  - Data integration
- You have had demos / tutorials on
  - Microsoft SQL Server
  - Apache Nifi
- During the newt lecture, you will learn about data cleaning and pre-processing

# Homework

# Homework

- Data integration / pre-processing
  - Access the demonstration website yourself and learn more about the NIFI flows:
  - https://youtu.be/MARazprrNYA

# Questions

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# Thank you for your attention!!!