Data Integration

Past, Present and Future

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

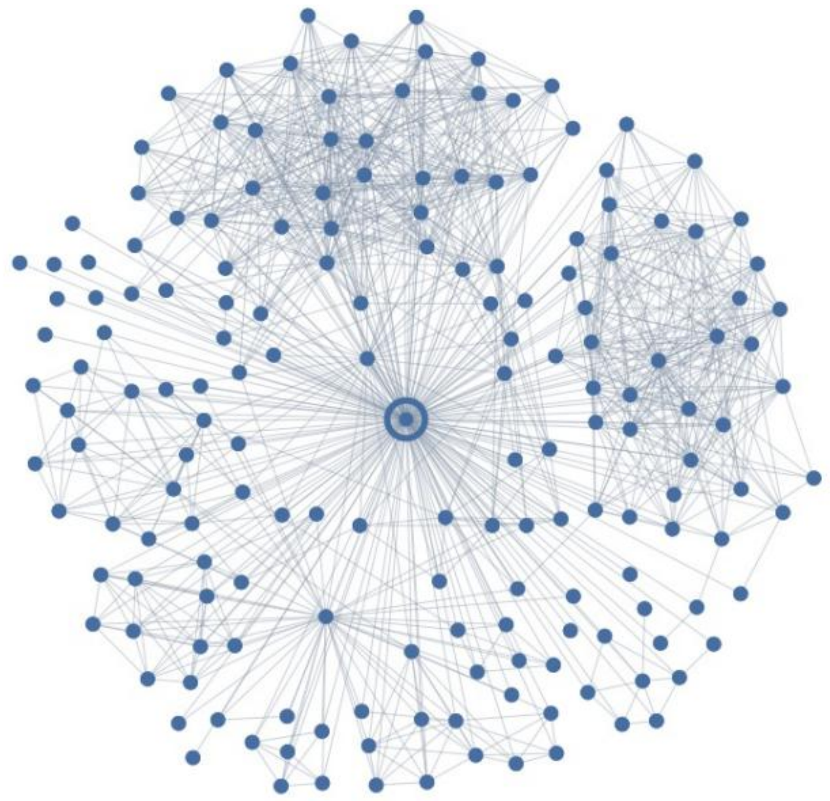
DTIM www.essi.upc.edu/dtim

“WITHOUT DATA,
YOU’RE JUST
ANOTHER PERSON
WITH AN OPINION”

W. Edwards Deming, American Statistician

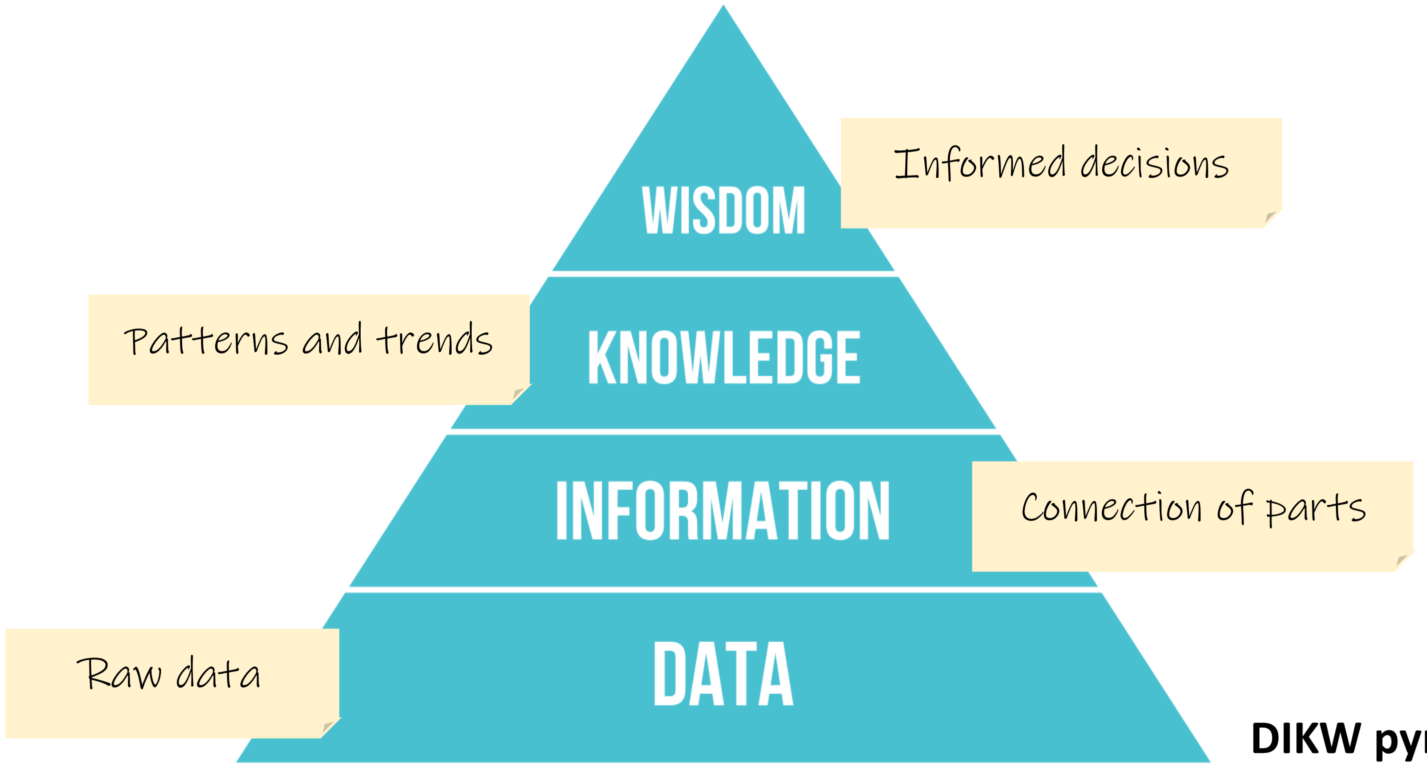
An Example: Will They Split Up?

- Crossing data from social networks it is possible to identify a graph like the one that follows:
  - In the centre there is a specific person *P*
  - The rest are *P* connections and connections among them
- Using sociology techniques…
  - We can identify *P social foci*:
    - Dense clusters of friends corresponding to long periods of interaction
    - Typically, college friends, coworkers, relatives, etc.
  - The *significant other* can be identified by a high *dispersion* rate
    - Highly connected with *P* connections,
    - But with a high dispersion degree wrt *P* social foci
- **Hypothesis**: when the node with higher dispersion degree Identified is not the partner, this couple is likely to split up in a period of 60 days
- L. Backstrom, J. Kleinberg. Romantic Partnerships and the Dispersion of Social Ties: A Network Analysis of Relationship Status on Facebook https://arxiv.org/pdf/1310.6753v1.pdf



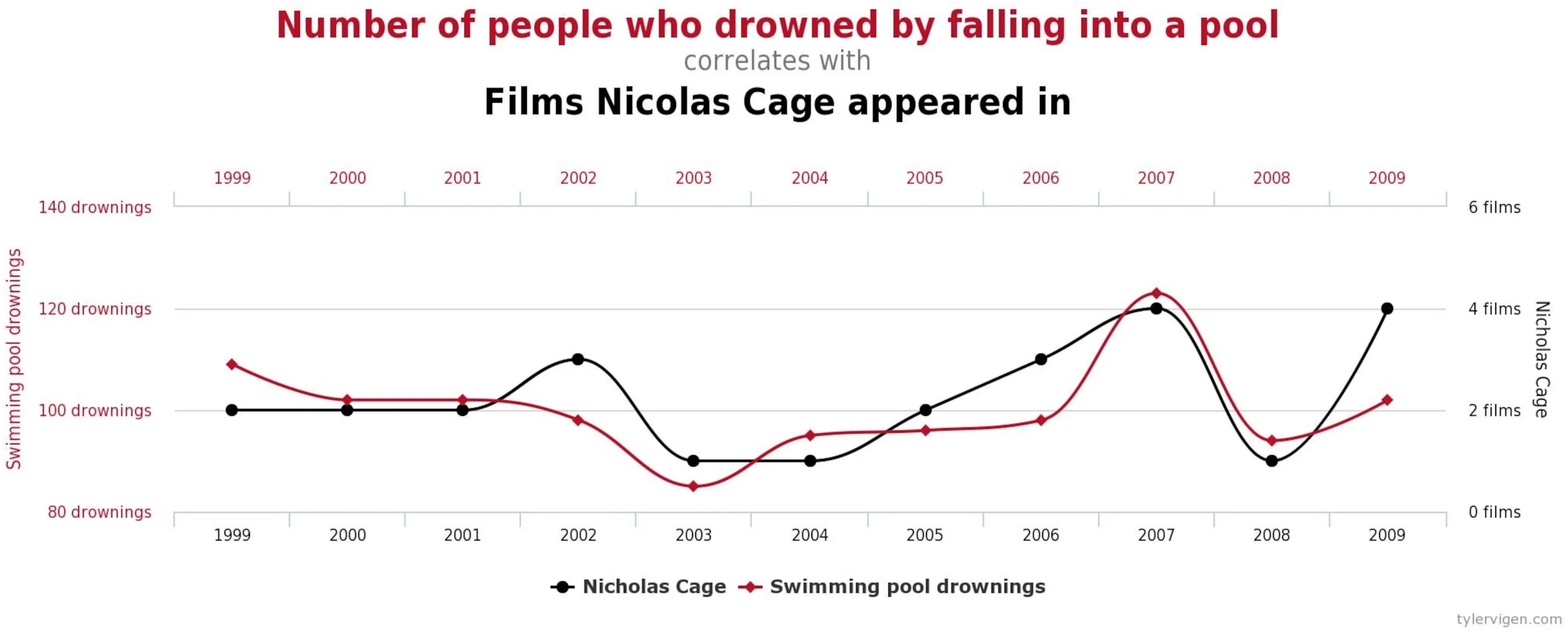
Decision Support Systems

- IT systems aimed at exploiting data and transform it into information and knowledge

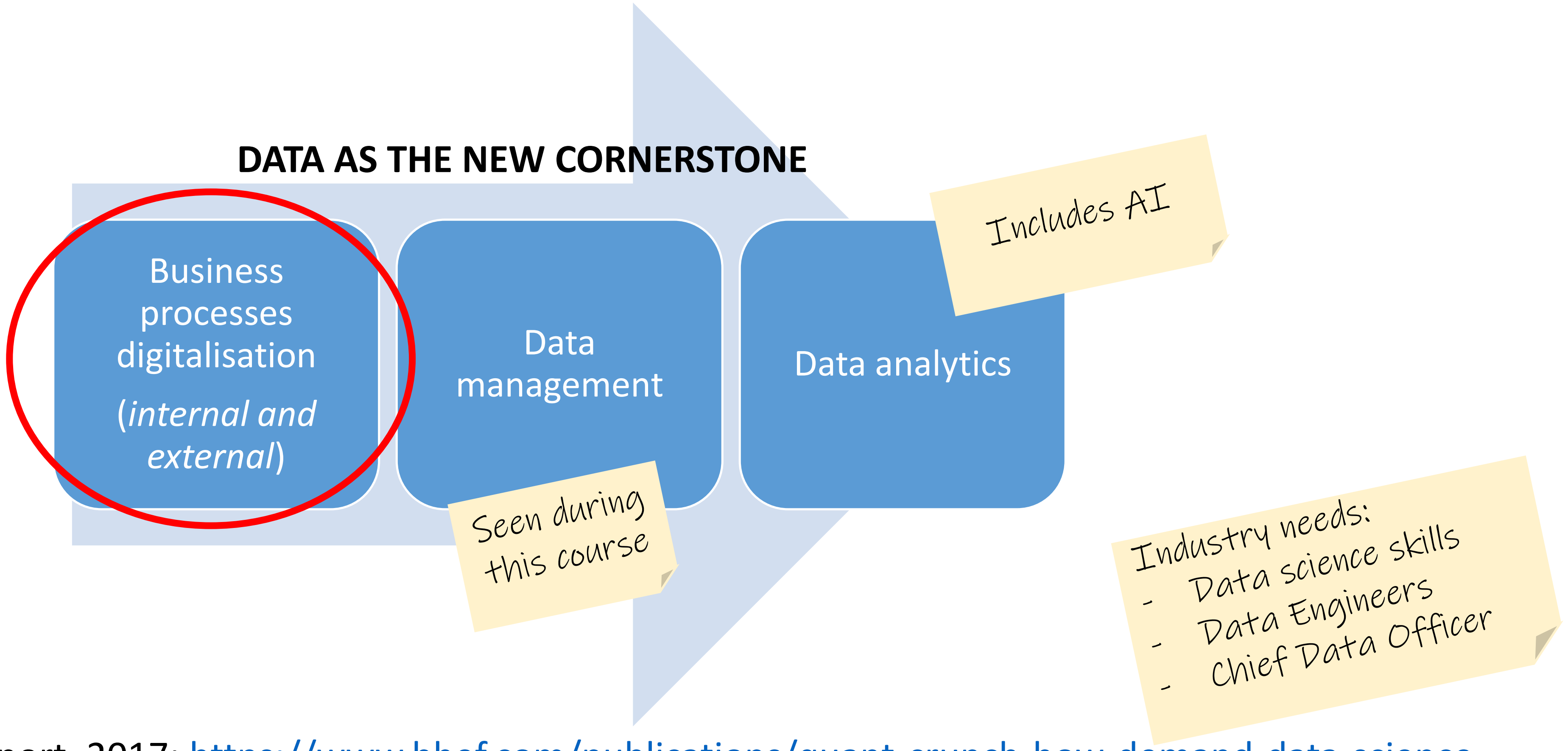


DIKW pyramid

From Information to Knowledge: a Counterexample

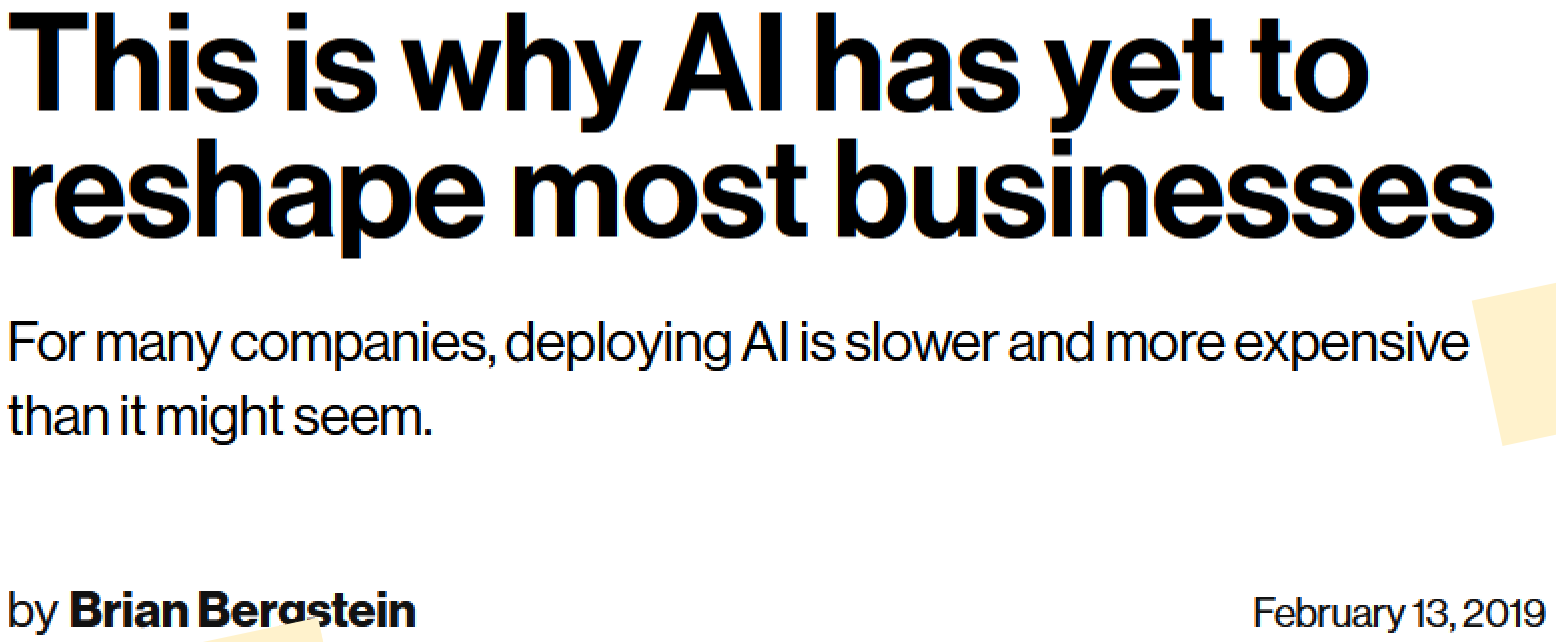


Decision Support Systems



The IBM Quant Crunch report, 2017: https://www.bhef.com/publications/quant-crunch-how-demand-data-science-skills-disrupting-job-market

Linking Data, Analysis and Business



This is why AI has yet to reshape most businesses

For many companies, deploying AI is slower and more expensive than it might seem.

by Brian Bergstein

February 13, 2019

10% AI
90% Data

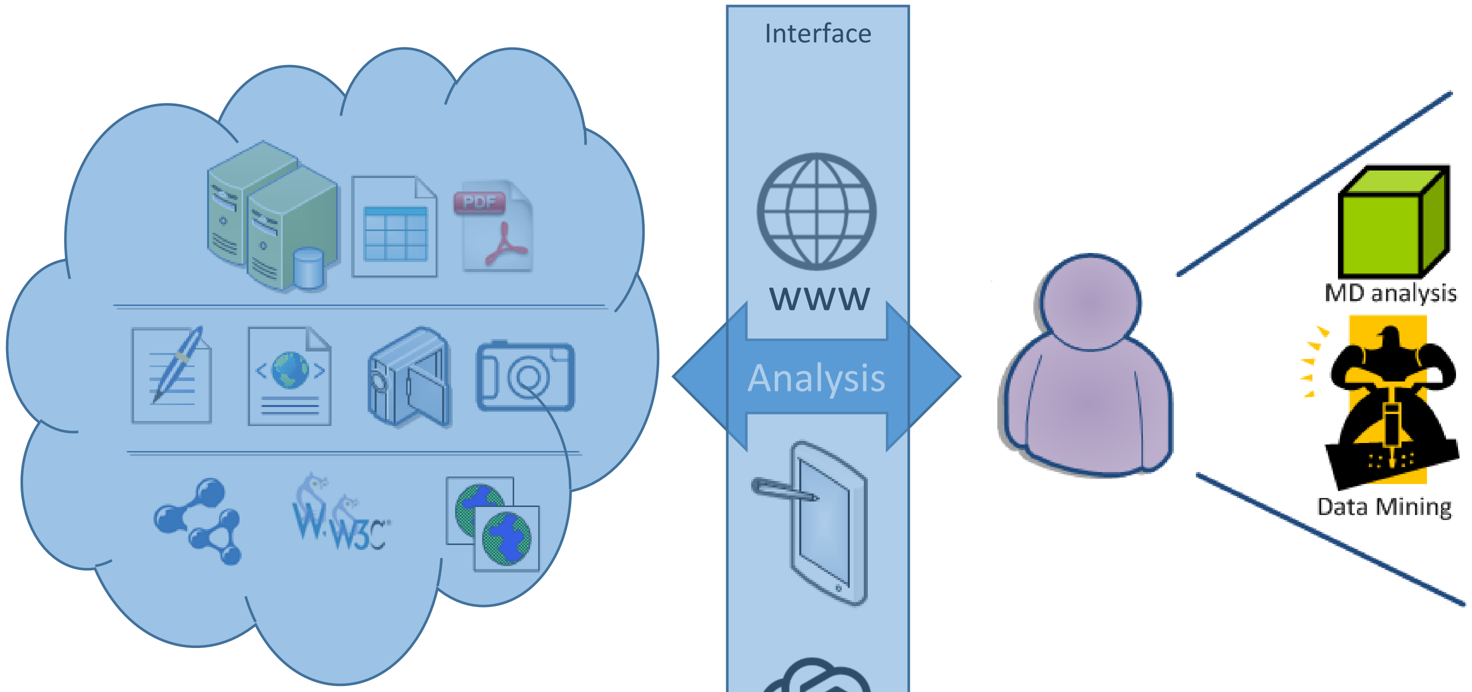
Biggest obstacles:
- Get and prepare all required data
- Lack of qualified professionals
- Reluctance of end users (trust)

Solutions:
- Automation of data-related tasks
- Data literacy (citizen data scientist)

nk: https://www.technologyreview.com/2019/02/13/137047/this-is-why-ai-has-yet-to-reshape-most-businesses/

Data Analysis Democratisation

- From the point of view of the end-user:



Recent AutoML tools go in this direction

Examples

- *A company wants to enrich their DW data (products, customers, shops) with external data coming from Twitter (opinion about their products), logs generated by their web and feedback received by phone, web or the company app. The constraint is that external data must be loosely coupled but be completely integrated from an analysis point of view*
- *A journalist wants to analyze the evolution of asylum demands in Europe in the last 10 years. What data are available in the EU open data portals? How data should be fetched and crossed in order to answer questions?*
- *A company with a large dataset of information from their activities wants to deploy a flexible Machine Learning pipeline. They want to avoid data analysts spend 80% of their time pre-processing data. There are many reasons: (i) data analysts should not repeat the same transformations once and again, (ii) they should collaboratively share their knowledge in pre-processing data and (iii) they want to avoid lock-in knowledge (i.e., analysts keep their knowledge in personal scripts not shared with the company)*

Exercise 1: Looking for clusters visually

From the course *Transition to Data Science*. Buy the entire course for just $10 for many more exercises and helpful video lectures.

You are given an array `points` of size 300x2, where each row gives the (x, y) co-ordinates of a point on a map. Make a scatter plot of these points, and use the scatter plot to guess how many clusters there are.

Step 1: Load the dataset *(written for you)*.

```
In [1]: import pandas as pd

        df = pd.read_csv('../datasets/ch1ex1.csv')
        points = df.values
```

Read data from temporal files

Step 2: Import PyPlot

```
In [2]: import matplotlib.pyplot as plt
```

Use off-the-shelf ML libraries

Step 3: Create an array called `xs` that contains the values of `points[:,0]` - that is, column `0` of `points`.

```
In [3]: xs = points[:,0]
```

Step 3: Create an array called `ys` that contains the values of `points[:,1]` - that is, column `1` of `points`.

```
In [4]: ys = points[:,1]
```

Pre-process the data to fit the needs of the algorithm input parameters

Step 4: Make a scatter plot by passing `xs` and `ys` to the `plt.scatter()` function.

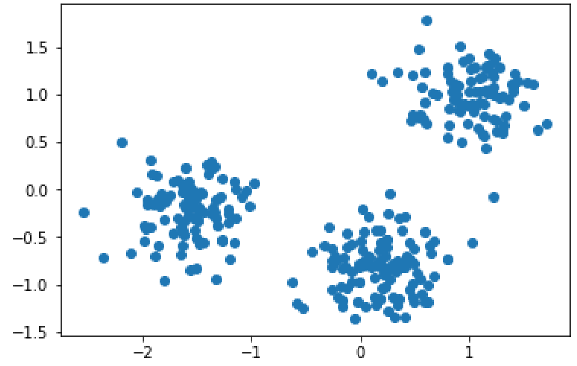
```
In [5]: plt.scatter(xs, ys)

Out[5]: <matplotlib.collections.PathCollection at 0x110fc45c0>
```

Run the algorithm

Step 5: Call the `plt.show()` function to show your plot.

```
In [6]: plt.show()
```

Visualise the result to allow its interpretation



Extracted from: https://github.com/benjaminwilson

Challenges in Data Management

From the IT point of view

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

DTIM www.essi.upc.edu/dtim

What is Big Data?

VOLUME

Veracity

Velocity

Value

vArIaBiLiTy

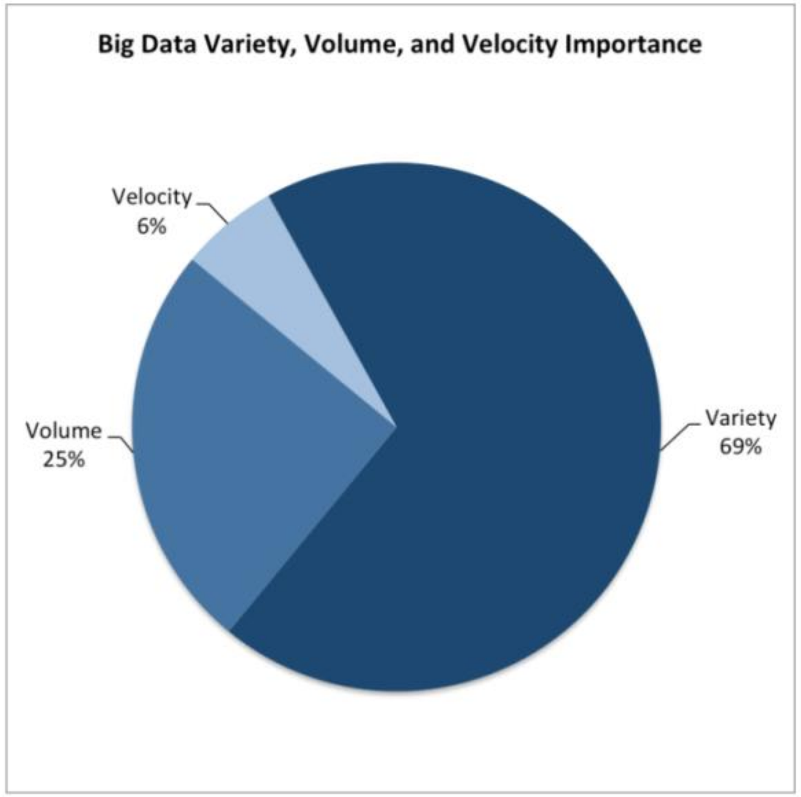
Variety

Today, the Focus is on Variety

That Big Data is synonymous with large volumes of data is a **myth**

*"Rather, it is the ability to **integrate** more sources of data than ever before — new data, old data, big data, small data, structured data, unstructured data, social media data, behavioral data, and legacy data"*

The Variety Challenge

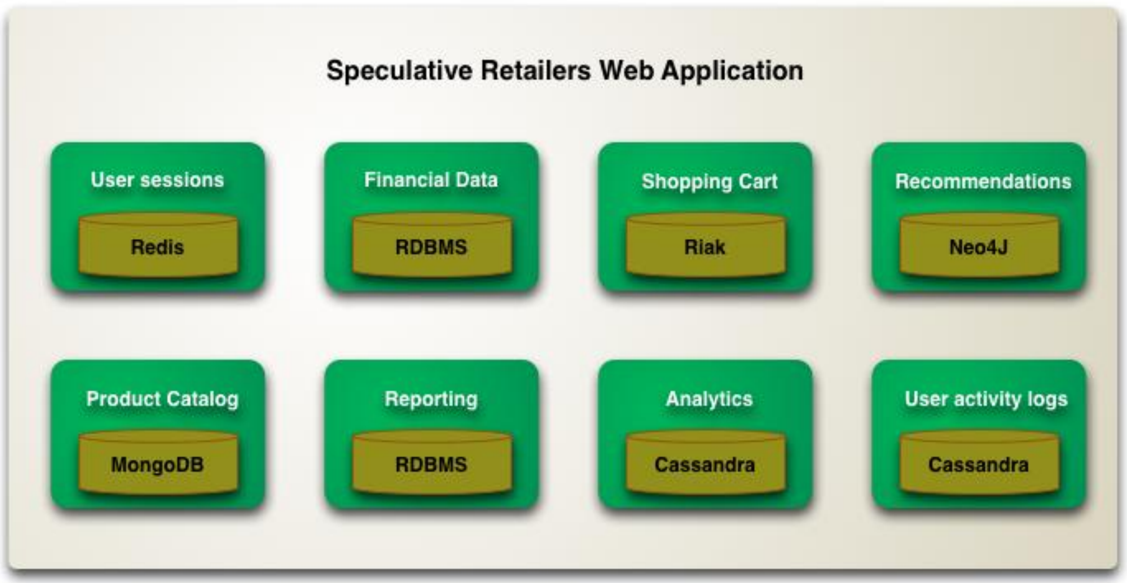


MIT Sloan Management Review (2016): http://sloanreview.mit.edu/article/variety-not-volume-is-driving-big-data-initiatives/

The Problem: Motivation

Currently, most organizations use different databases for different applications

- Example:



Polyglot persistence, Martin Fowler

The Problem: Motivation

When performing data analytics (e.g., reporting, recommendations, machine learning) we may need to extract / query data from different databases.

For example, to perform recommendations we may need to cross information from the product catalog, user sessions and user activity logs:

- Collaborative recommendations: find *similar* users and suggest to them what their similars bought
- Similarity is built based on their session and activity information, and also based on the characteristics of the product bought

The Solution

***Data integration** involves combining data residing in different sources and providing users with a unified view of them*

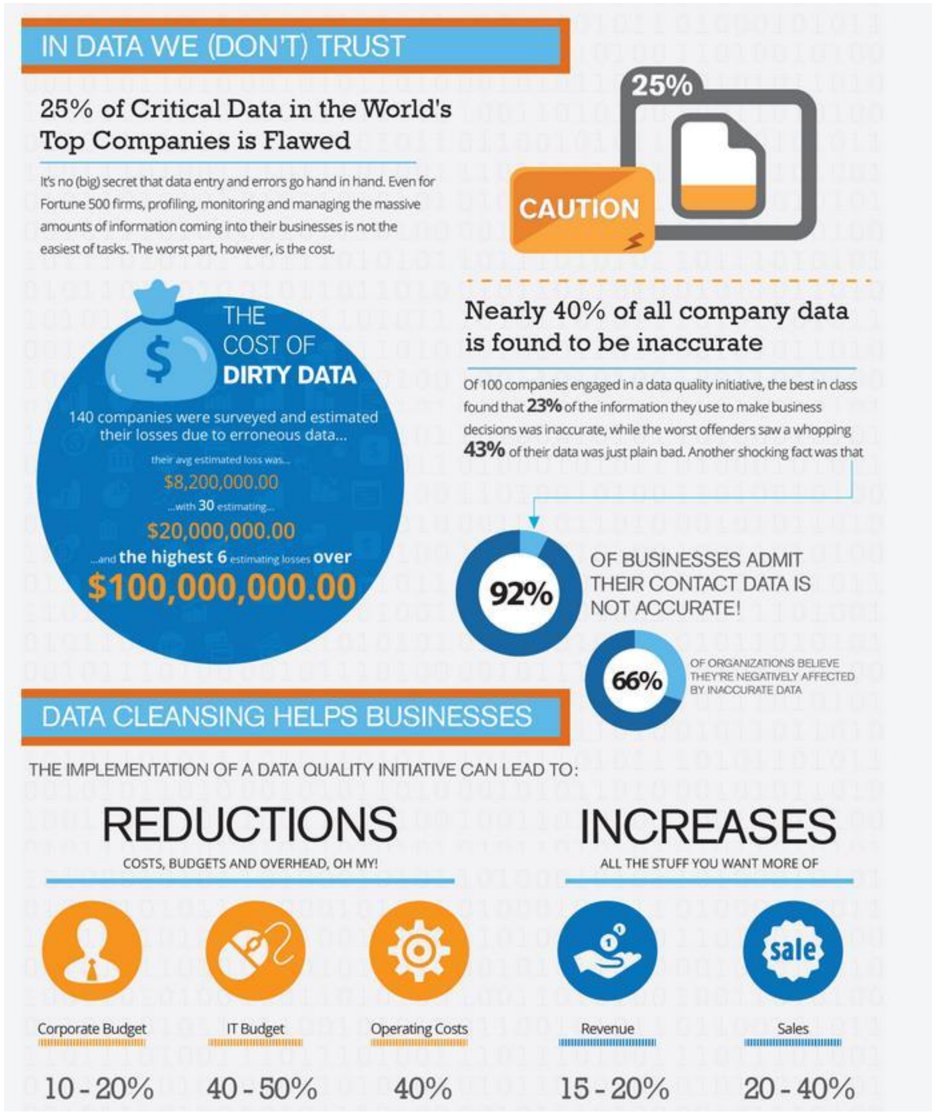
Maurizio Lenzerini (2002). "Data Integration: A Theoretical Perspective". PODS 2002. pp. 233–246.

Today: Data sources can be **within** or **outside** the organization

Data Quality

The Relevance of Data Cleaning in Data Integration

Motivation



IN DATA WE (DON'T) TRUST

25% of Critical Data in the World's Top Companies is Flawed

It's no (big) secret that data entry and errors go hand in hand. Even for Fortune 500 firms, profiling, monitoring and managing the massive amounts of information coming into their businesses is not the easiest of tasks. The worst part, however, is the cost.

THE COST OF **DIRTY DATA**

140 companies were surveyed and estimated their losses due to erroneous data...

their avg estimated loss was...

$8,200,000.00

...with **30** estimating...

$20,000,000.00

...and **the highest 6** estimating losses **over**

$100,000,000.00

Nearly 40% of all company data is found to be inaccurate

Of 100 companies engaged in a data quality initiative, the best in class found that **23%** of the information they use to make business decisions was inaccurate, while the worst offenders saw a whopping **43%** of their data was just plain bad. Another shocking fact was that

92% OF BUSINESSES ADMIT THEIR CONTACT DATA IS NOT ACCURATE!

66% OF ORGANIZATIONS BELIEVE THEY'RE NEGATIVELY AFFECTED BY INACCURATE DATA

DATA CLEANSING HELPS BUSINESSES

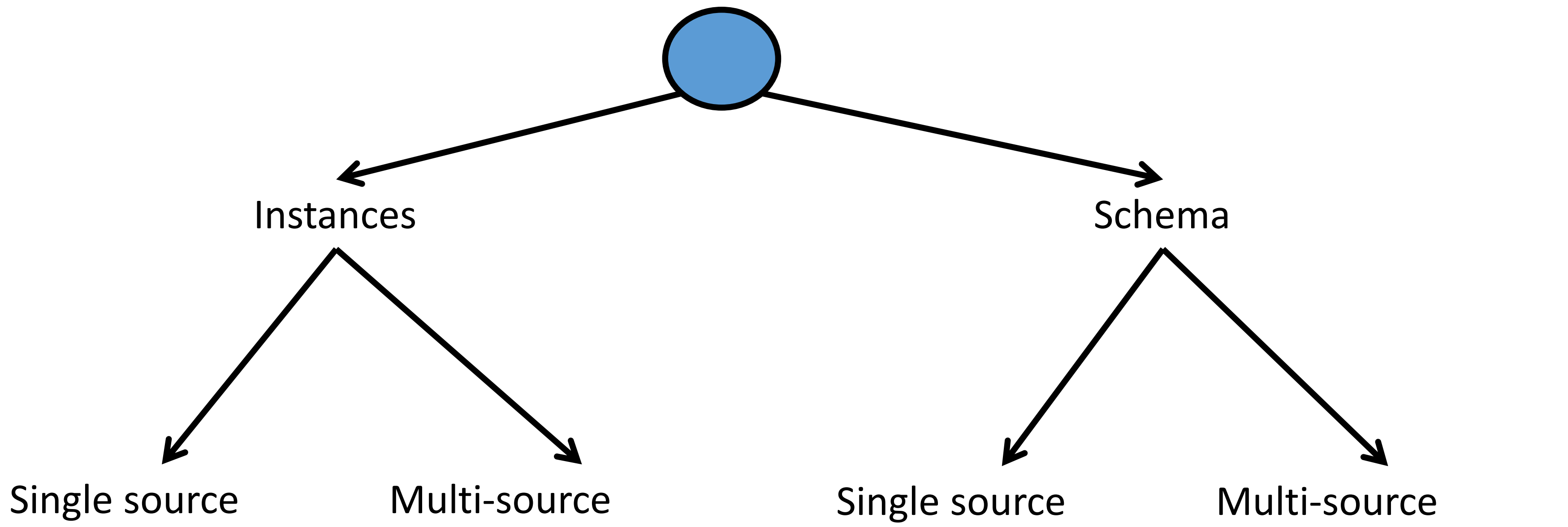
THE IMPLEMENTATION OF A DATA QUALITY INITIATIVE CAN LEAD TO:

REDUCTIONS COSTS, BUDGETS AND OVERHEAD, OH MY!			INCREASES ALL THE STUFF YOU WANT MORE OF	
Corporate Budget	IT Budget	Operating Costs	Revenue	Sales
10 - 20%	40 - 50%	40%	15 - 20%	20 - 40%

Can you think of some examples of data quality problems?

Halo Business Intelligence

Data Conflicts



Instances – Single source

(Data entry errors)

- Misspelling
- Redundancy
- Contradictions
- …

Instances – Multi-source

(Overlapping, contradicting and inconsistent data)

- Duplicates
- Inconsistent aggregation
- Inconsistent timing
- …

Schema – Single source

(Lack of integrity constraints, poor schema design)

- Uniqueness
- Referential integrity
- …

Schema – Multi-source

(Heterogeneous data models and schema designs)

- Naming conflicts
- Structural conflicts
- …

They may appear at every possible stage of a data pipeline!

Typical Quality Measures

- Completeness
- Accuracy
- Consistency
- Timeliness (Freshness)

Completeness

"The degree to which a given collection of data describes the corresponding set of real-world objects"

- Issues:
  - Missing values (instance level)
  - Missing entities (instance level)
  - Missing attributes (schema level)

Not all NULL values have the same meaning or importance

- Solution:
  - Imputation of values
  - Additional data sources

Accuracy

“The extent to which data are correct, reliable and certified error free”

- Issues:
  - Typing errors
  - Inappropriate precision
- Solution:
  - Using dictionaries or ontologies
  - Removing inexact values
  - Applying data cleaning techniques

Distance between the data and the value in the real world

Consistency

"The degree of violation of semantic rules defined over a set of data items"

- Issues:
  - Integrity constraint violations
    - Entity, Domain, Referential, User-defined
  - Inconsistency between copies
    - Temporal or permanent
- Solution:
  - Removing or replacing inconsistent data

Rules can be specified by the user, or discovered from the data

Timeliness (Freshness)

"How old the stored value of an attribute is with regard to the current value in the real world"

- Issues:
  - Outdated data
    - Depends on the update frequency
    - ...And of the goals of the user
- Solution:
  - Additional data sources

The Future of Data Integration

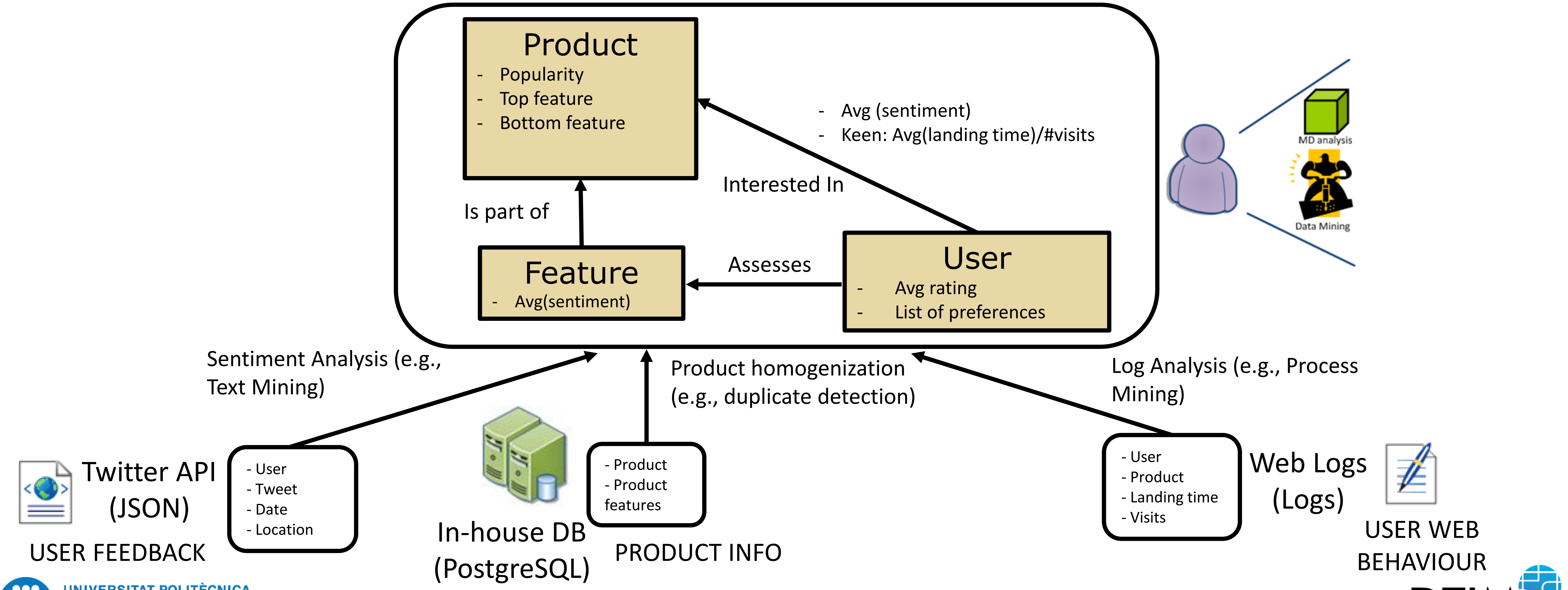
Data Lakes and DataOps

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

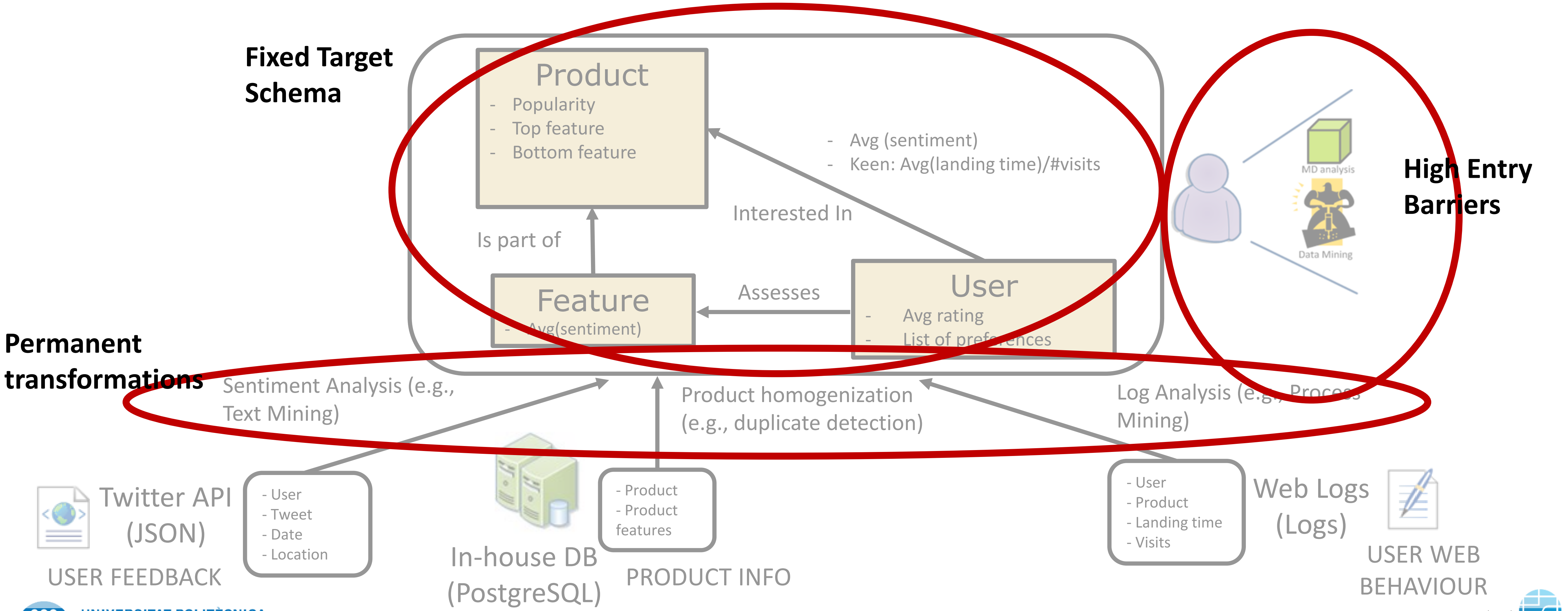
DTIM www.essi.upc.edu/dtim

Traditionally: Model-First (Load-Later)

Typical of Data Warehousing (ETL)



Drawbacks

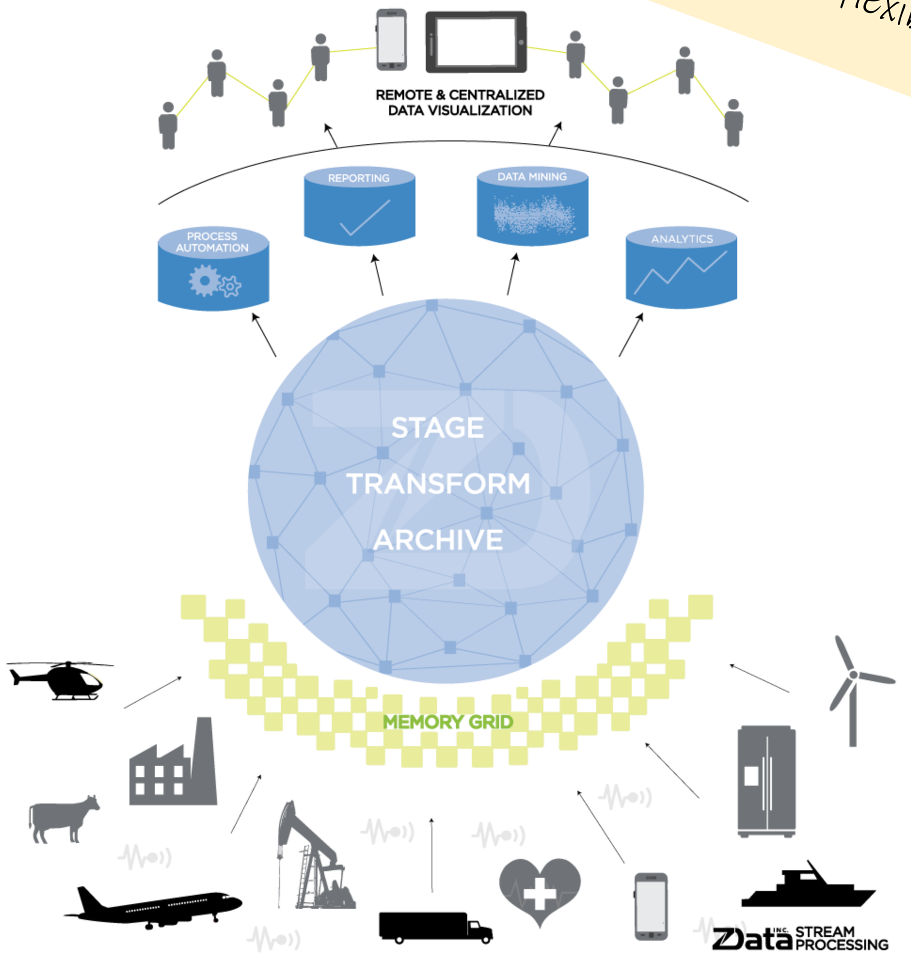

Fixed Target Schema
Product
- Popularity
- Top feature
- Bottom feature
- Avg (sentiment)
- Keen: Avg(landing time)/#visits
Interested In
Is part of
Feature
- Avg(sentiment)
Assesses
User
- Avg rating
- List of preferences
MD analysis
Data Mining
High Entry Barriers
Permanent transformations
Sentiment Analysis (e.g., Text Mining)
Product homogenization (e.g., duplicate detection)
Log Analysis (e.g., Process Mining)
Twitter API (JSON)
- User
- Tweet
- Date
- Location
USER FEEDBACK
In-house DB (PostgreSQL)
- Product
- Product features
PRODUCT INFO
- User
- Product
- Landing time
- Visits
Web Logs (Logs)
USER WEB BEHAVIOUR

The Data Lake

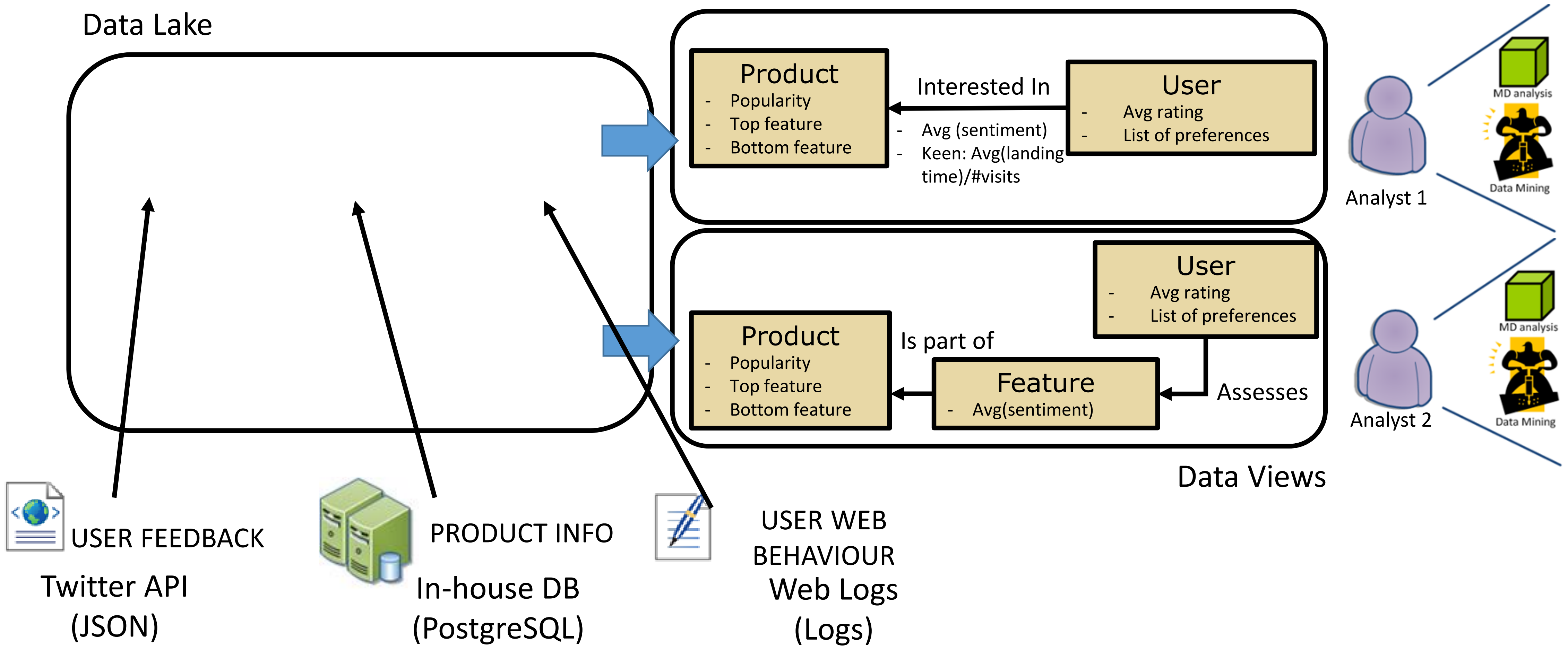
- **IDEA: Load-first, Model-Later**
- Modeling at load time restricts the potential analysis that can be done later (Big Analytics)
- Store raw data and create on-demand views to handle with precise analysis needs

The goal is flexibility

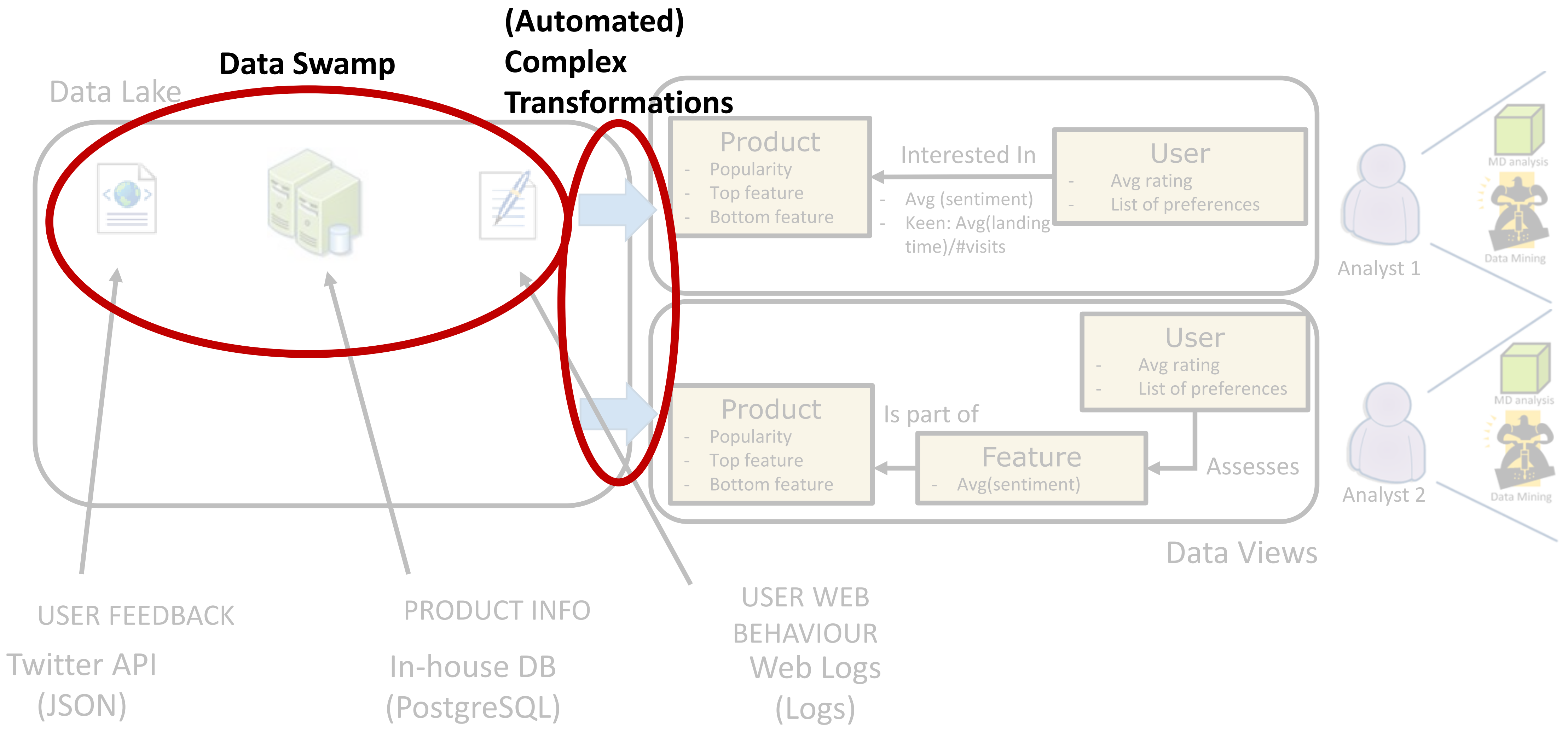
It is not a technology or architecture, it is a philosophy



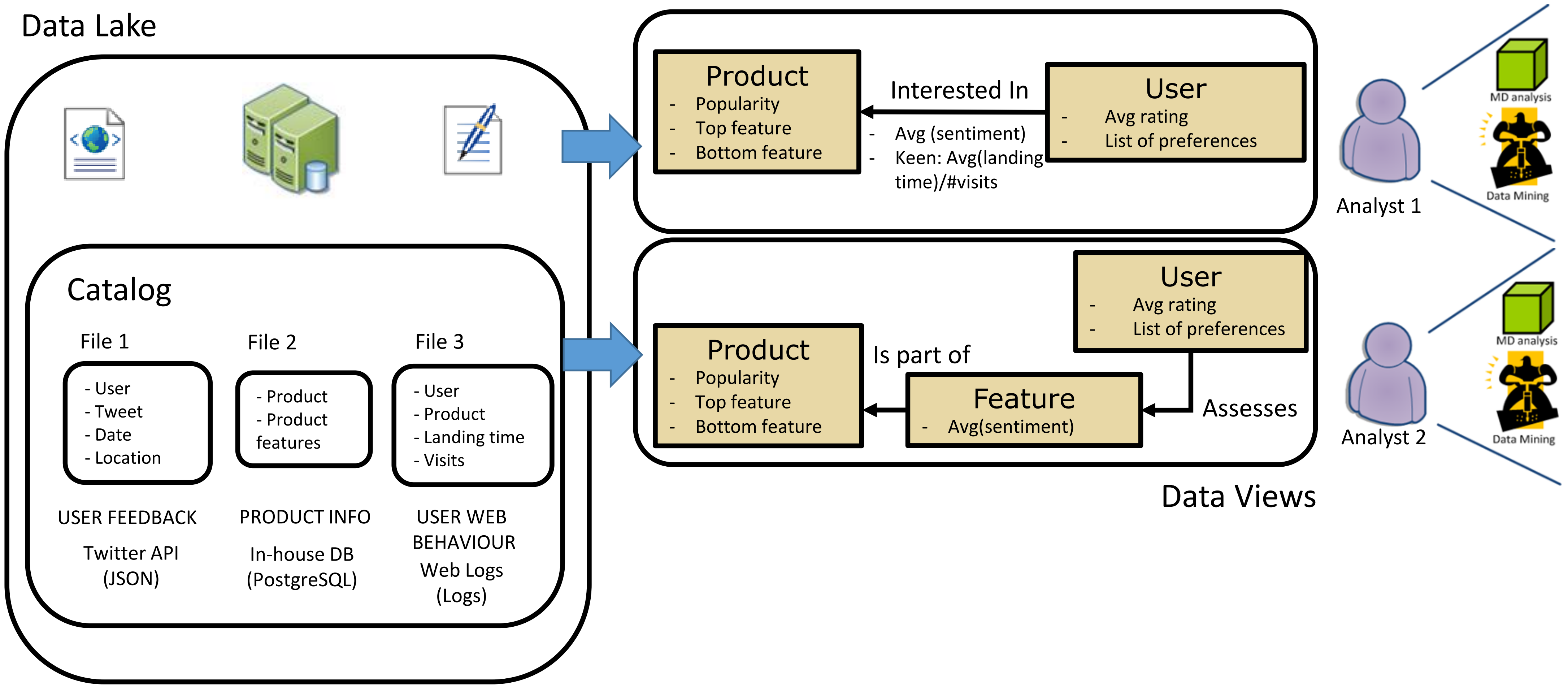
Load-First Model-Later



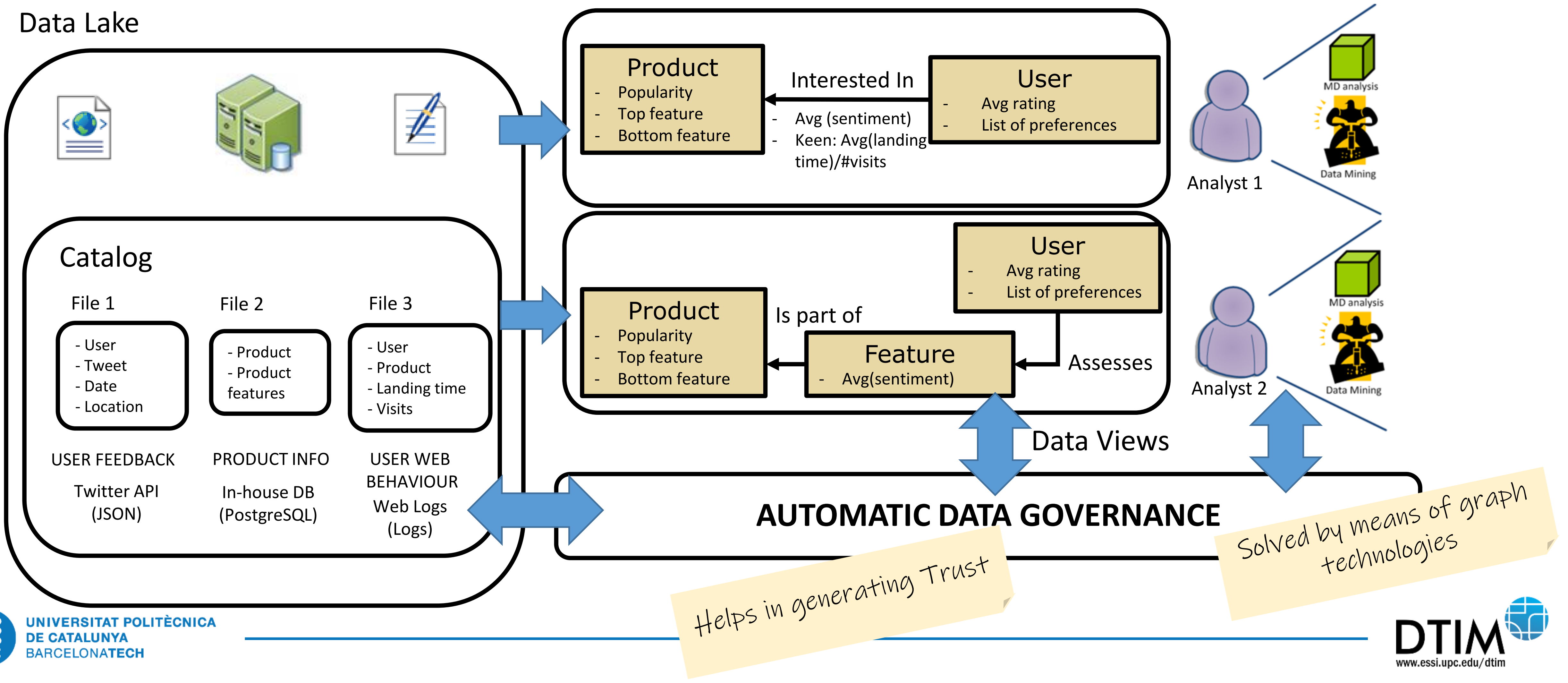
Drawbacks


(Automated) Complex Transformations
Data Swamp
Data Lake
Product
- Popularity
- Top feature
- Bottom feature
Interested In
- Avg (sentiment)
- Keen: Avg(landing time)/#visits
User
- Avg rating
- List of preferences
Analyst 1
MD analysis
Data Mining
User
- Avg rating
- List of preferences
Product
- Popularity
- Top feature
- Bottom feature
Is part of
Feature
- Avg(sentiment)
Assesses
Analyst 2
MD analysis
Data Mining
Data Views
USER FEEDBACK
Twitter API (JSON)
PRODUCT INFO
In-house DB (PostgreSQL)
USER WEB BEHAVIOUR
Web Logs (Logs)

From Data Swamps to Semantic Data Lakes



From IT-Centered to User-Centered

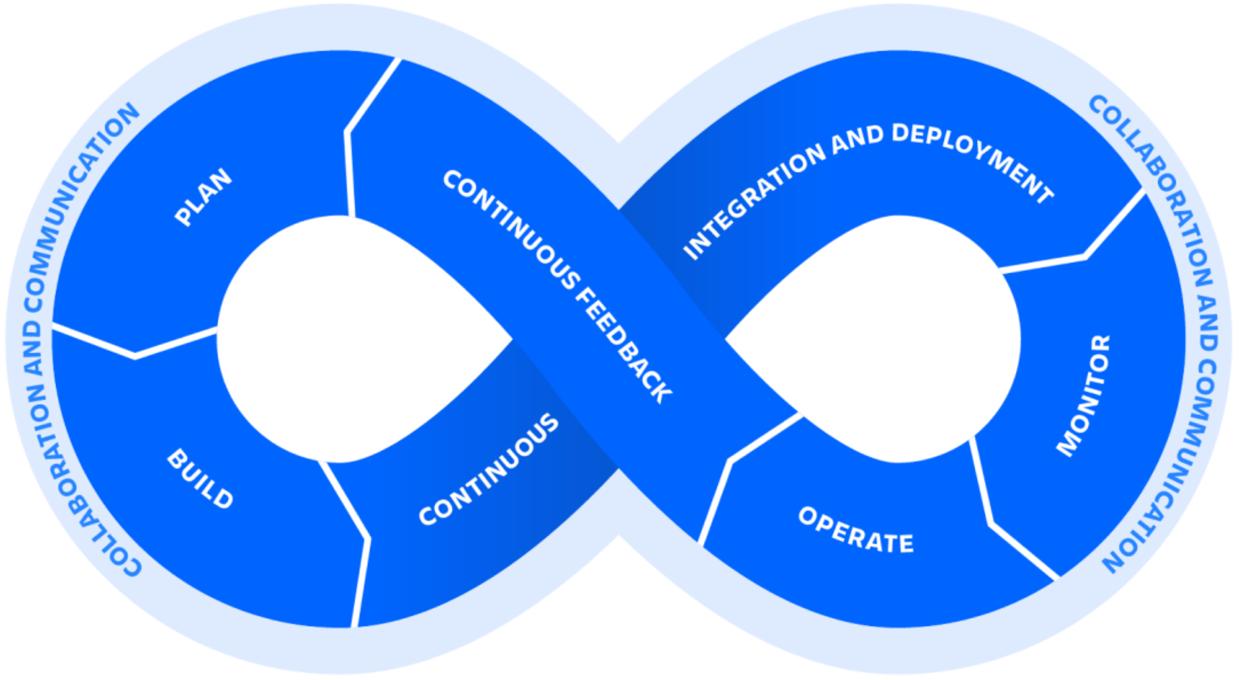


DevOps

DevOps is about fast, flexible development and provisioning processes.

It integrates the two worlds of development and operation, using automated development, deployment, and infrastructure monitoring. Its an organizational shift in which, instead of distributed isolated groups performing functions separately, cross-functional teams work on continuous operational feature deliveries.

Ebert, C., Gallardo, G., Hernantes, J., Serrano, N. (2016). DevOps. *IEEE Software 33*(3): 94-100.


COLLABORATION AND COMMUNICATION
PLAN
BUILD
CONTINUOUS
CONTINUOUS FEEDBACK
INTEGRATION AND DEPLOYMENT
MONITOR
OPERATE
COLLABORATION AND COMMUNICATION

From DevOps to DataOps / MLOps

DevOps ignores a key aspect in Data Science: data and its life cycle

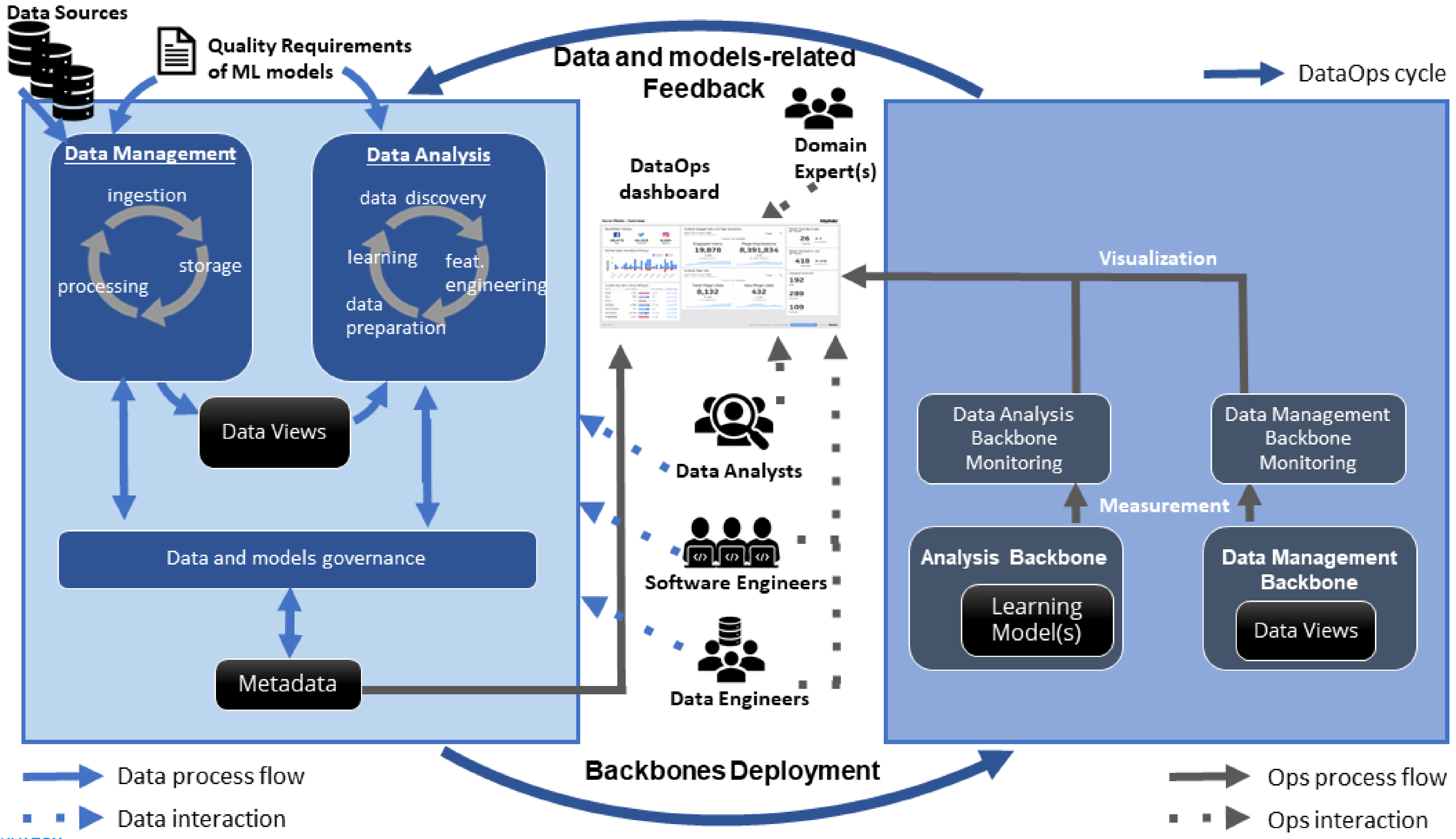
DataOps was introduced to cover this gap by *combining an integrated and process-oriented perspective on data with automation and methods from agile software engineering, like DevOps, to improve quality, speed, and collaboration and promote a culture of continuous improvement.*

Ereth, J. (2018). DataOps - Towards a Definition. LWDA 2018: 104-112.

In short, the DataOps life cycle is needed to manage the complexity of the data life cycle in data science projects (data engineering)

***Disclaimer**: you may read about MLOps too. However, in essence, DataOps / MLOps talk about the same problem. Simply, the latter focuses more on the ML part while the former covers the whole data lifecycle.*

DataOps in a Nutshell



DataOps in a Nutshell

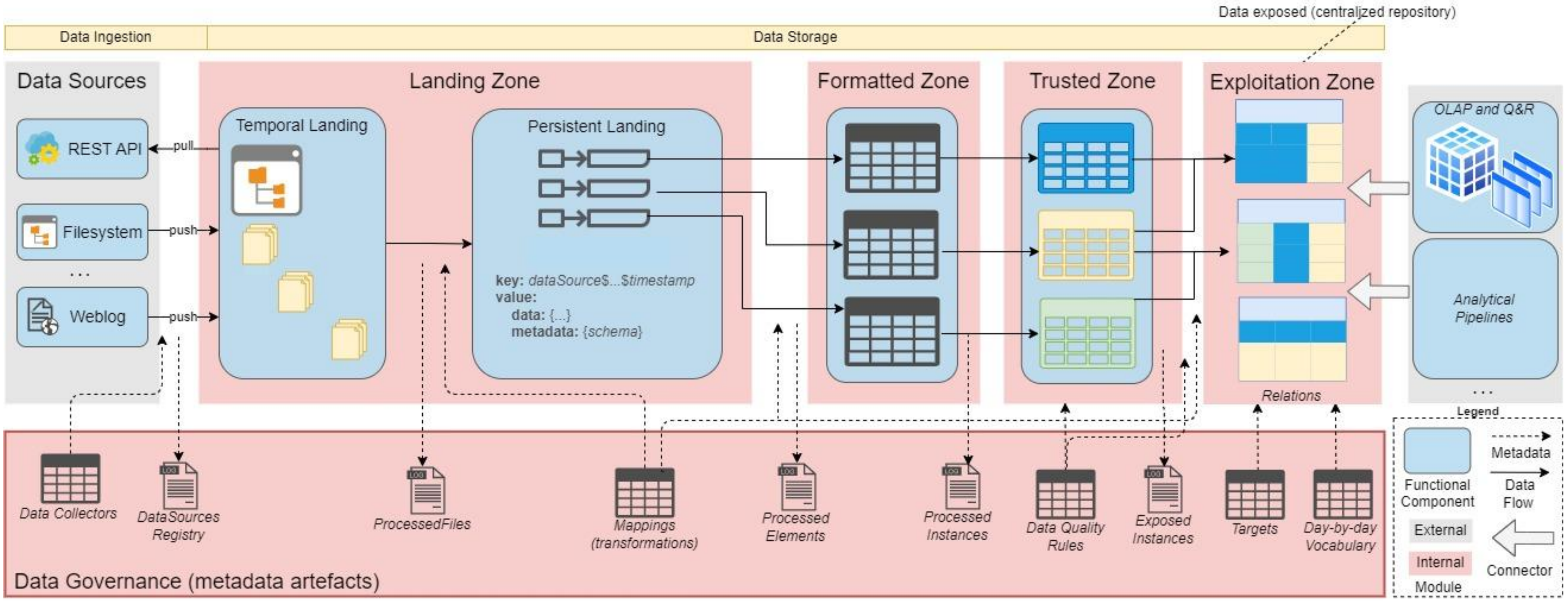
The data management backbone (common for the whole organisation)

- Ingest and store external data into the system
  - Data Integration (standardization and data crossing)
    - Syntactic homogenization of data
    - Semantic homogenization of data
  - Clean data / eliminate duplicates / …
- Expose a cleaned and centralized repository
  - Data profiling

The data analysis backbone (repeats for every analytical pipeline)

- Extract a data view from the centralized repository
- Feature engineering
- Specific pre-processing for the given analytical task at hand
  - Labeling
  - Data preparation specific for the algorithm chosen
- Create test and validation datasets
- Learn models (either descriptive stastitical analysis or advanced predictive models)
- Validate and interprete the model

The Data Management Backbone



The Data Management Backbone

- Data is ingested in the landing zone as it is produced (raw data)
  - The temporal landing stores temporary the files, until they are processed
  - The persistent landing tracks ingested files by data source and timestamp (i.e., versions)
- Data is then homogenized, according to a canonical data model in the formatted zone (syntactic homogenization)
- The trusted zone is where data cleaning happens.
  - Data quality processes: per dataset
- The exploitation zone exposes data ready to be consumed / analysed either by advanced analytical pipelines or external tools. Two main kind of tasks are conducted to generate this zone (semantic homogenization):
  - Data integration: new data views are generated by combining the instances from the trusted zone. A view may serve several data analysis.
  - Data quality processes: across datasets

Governing the Data Management Backbone

The data governance layer must fulfill the following tasks:

- Monitor all the data flow executions between zones, as well as the required metadata to allow their incremental execution (logs)
- Metadata artifacts capturing the schema of the data views created in the exploitation zone, but also in any of the intermediate zones
- Metadata required to automate processes: templates to process different types of data, database connections, etc.
- Mappings between zones: transformations conducted between elements of two consecutive zones to enable data lineage / traceability
- Profiling information of every zone: for maintenance purposes for the landing, trusted and formatted zones and analytical purpose in the exploitation zone

Tools for DataOps

Tools for the Data Management Backbone

- Ingestion: it can be performed by means of off-the-shelf tools or ad-hoc scripting. Further, it can be pull or push
  - Scripting: Java, Python, bash scripts, etc.
  - Tools: Kafka, Spark, etc.
- Storage: when facing Big Data, distributed storage is a must. Anyway, specially in the exploitation zone, a vast range of tools might be used to store a given data view
  - Distributed storage: distributed file Systems (e.g., HDFS) and distributed databases (e.g., HBase, SimpleDB, Cassandra).
  - Centralized storage: relational databases (e.g., PostgreSQL, DuckDB), etc.
- Data quality and profiling: in the past, they were tightly related to the analytical tools (e.g., SAP, IBM, etc.). But more and more there is an extensive offer of stand-alone tools that fit better the required *openness* view of data science: Trifacta, HoloClean, Attacama, Informatica, Dataprep.ai, etc.
- Transformations between zones: Spark and Flink the most used ones. There is the option of going to a lower level of abstraction and use frameworks that allow to tailor the execution to your needs and gain efficiency: e.g., Akka.

Tools for Governance

Knowledge graphs are nowadays the standard the facto to govern Data Science projects. We differentiate repositories from generic frameworks supporting all data governance steps

- Repositories: GraphDB, Stardog, Amazon Neptune, etc.
- Governance frameworks: typically embedded as part of the whole data management backbone (specially in Cloud services). However, there are quite a few stand-alone solutions:
  - Cloud services: Delta Lake (Databricks), Azure Purview (Microsoft), IBM Data Governance, Data Catalog (Google Cloud), etc.
  - Standalone: Apache Atlas, Collibra, Semantic Web Company, Salesforce Einstein, Alteryx, DataRobot, H2O Driverless AI, etc.

Main Programming Languages

- Development environment
  - An exploratory phase where data scientists aim to flexibly explore the data sources at hand, select the most relevant variables or evaluate different models.
  - Desired features
    - Dynamically typed languages
    - Interpreted code (no compilation required)
    - Unnecessary verbose code
  - Exemplary languages: Python, R, Perl
- Operations environment
  - A consolidation phase where the goal is to deploy a data infrastructure with performance guarantees and robust to external changes.
  - Desired features
    - Statically typed languages
    - Compiled code (low-level optimization and memory management)
    - Built-in parallelism for concurrent applications
    - Modularity
    - Advanced error and exception management
  - Exemplary languages: Java, Scala, C++

Summary

- Variety in Big Data requires Data Integration to obtain valuable insights
- Data Quality must be guaranteed
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
- DataOps to operationalize data management and analysis

Bibliography

- M. Golfarelli and S. Rizzi. *Data Warehouse Design*. McGrau-Hill, 2009
- R. Kimball, J. Caserta. The Data Warehouse ETL Toolkit. Wiley Publishing, 2004
- J. Han and M. Kamber. "*Data Mining: Concepts and Techniques*". Morgan Kauffman Publishers, 2000
- Carlo Batini, et al. *Methodologies for data quality assessment and improvement*. ACM Compututing Surveys, 41(3), 2009
- Arkady Maydanchik. *Data Quality Assessment*. Technics Publications, 2007
- H. Garcia-Molina et al. *Database Systems*. Prentice Hall, 2009
- J. Bleiholder and F. Naumann. Data Fusion. ACM Computing Surveys, 41(1), 2008