Session 4

Exploring the knowledge and lessons from ETF project Big Data for LMI

Overview of the technical construction of the OJV data system: from landscaping of data sources to data visualisation

Mauro Pelucchi

November 2021

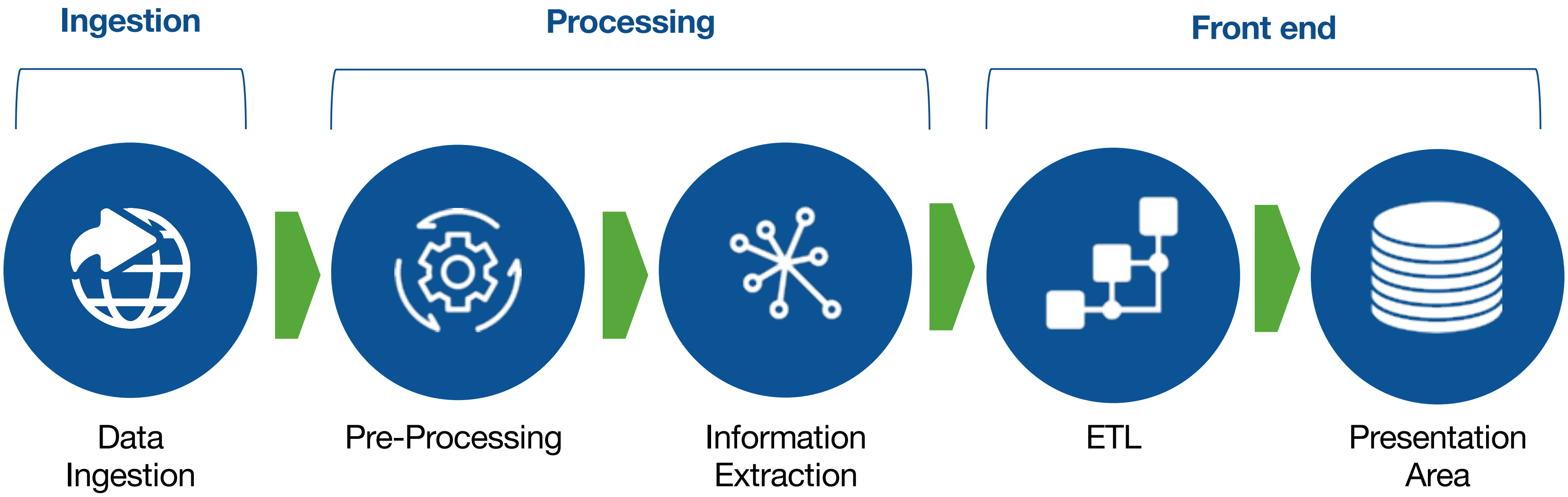
Topics

1. Overview & Recap
2. What is a pipeline?
3. Storage layer
4. Spark foundations
5. Lab sessions
    1. Find new job titles
    2. Find new occupations

Topics

1. **Overview & Recap**
2. What is a pipeline?
3. Storage layer
4. Spark foundations
5. Lab sessions
    1. Find new job titles
    2. Find new occupations

Overall Data Flow



Ingestion

Processing

Front end

Data Ingestion

Pre-Processing

Information Extraction

ETL

Presentation Area

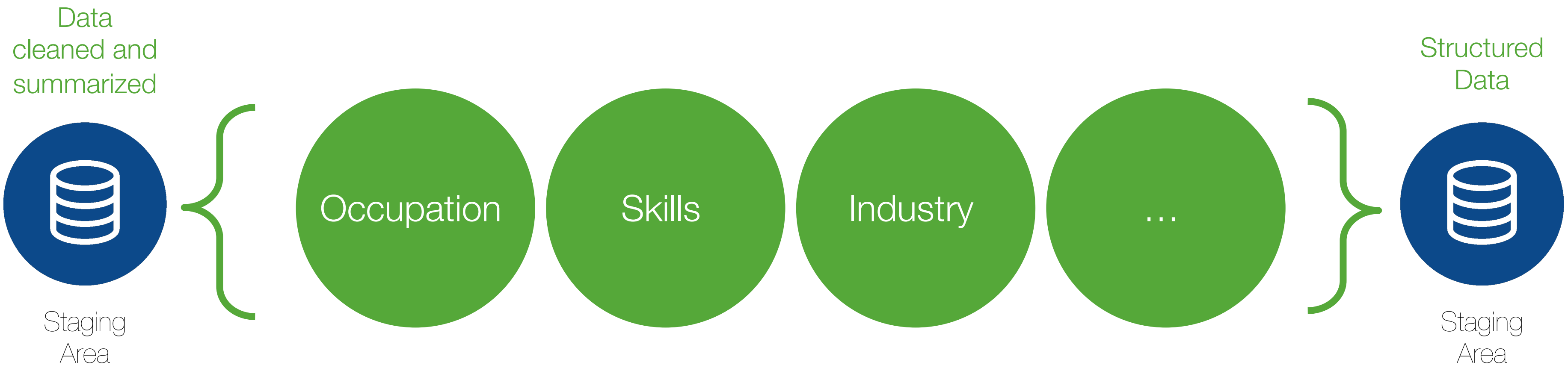
Information Extraction

- Goal:
  - Extract and structure information from data, to be provided to the presentation layer
- Challenges:
  - Handle massive amount of heterogeneous data written in different languages
- Approach:
  - Develop an adaptable framework, tailored on different information features. Some relevant challenges:
    - Occupation feature classification: combined methods such as Machine Learning, Topic Modeling and Unsupervised Learning
    - Skill feature classification: another different combined methods, such as Text Analysis with corpus based or Knowledge based similarity
- Features:
  - Guarantee Explainable information extraction, logging classification methods and relevant features.

Information Extraction and Classification
Real Time Labour Market Intelligence

Information Extraction is an area of natural language processing that deals with finding factual information in free text.

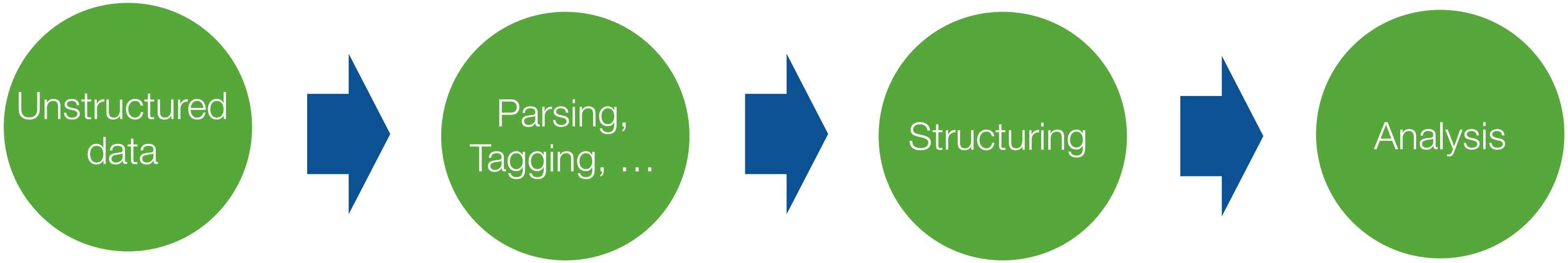
This task uses machine learning techniques (ontology based learning, supervised learning and unsupervised learning) to match job ads with standard classifications.

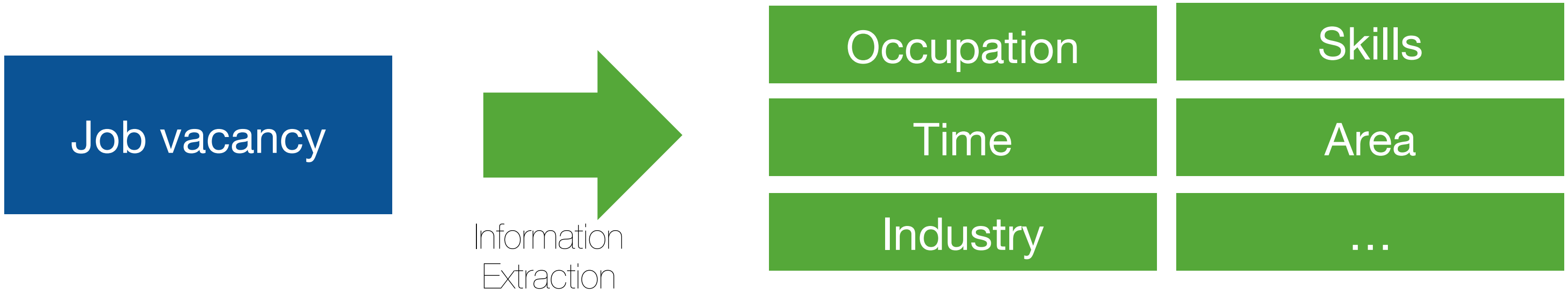


Machine Learning → Ontology based learning, supervised learning and unsupervised learning, etc.

Information Extraction

Information Extraction: analyse an unstructured document with the scope of extracting specific information.





Job vacancy

Information Extraction

Occupation

Skills

Time

Area

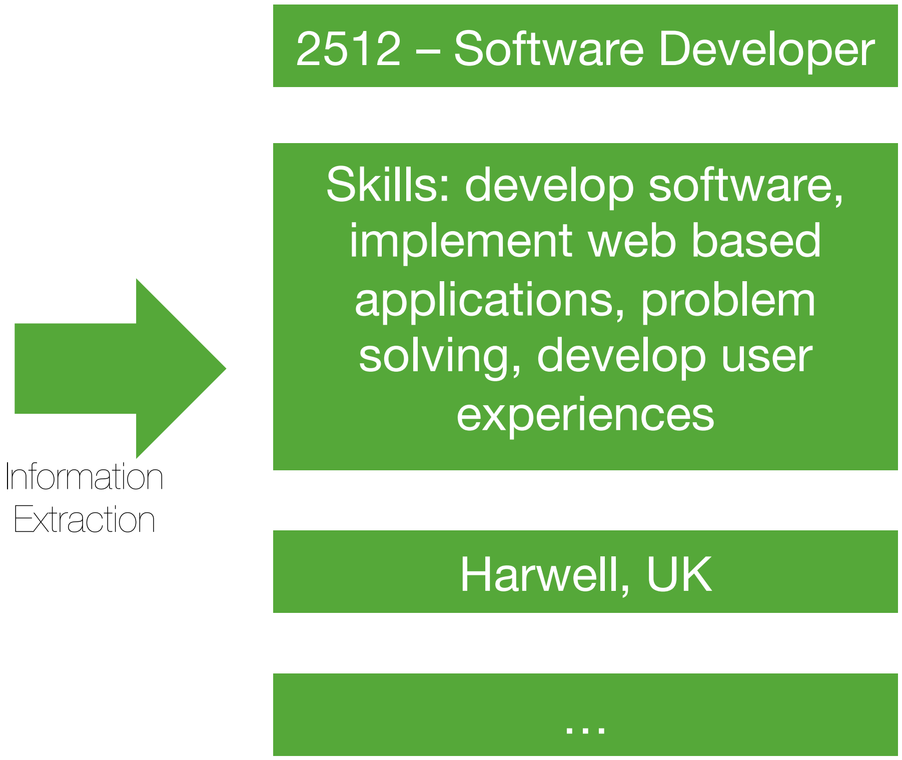
Industry

…

Junior Software Developer

As Junior Software Developer, you will develop excellent software for use in field mapping, data collection, sensor networks, street navigation, and more. You will collaborate with other programmers and developers to autonomously design and implement high-quality web-based applications, restful API's, and third party integration.

We're looking for a passionate, committed developer that is able to solve and articulate complex problems with application design, development and user experiences. The position is based in our offices in Harwell, United Kingdom.



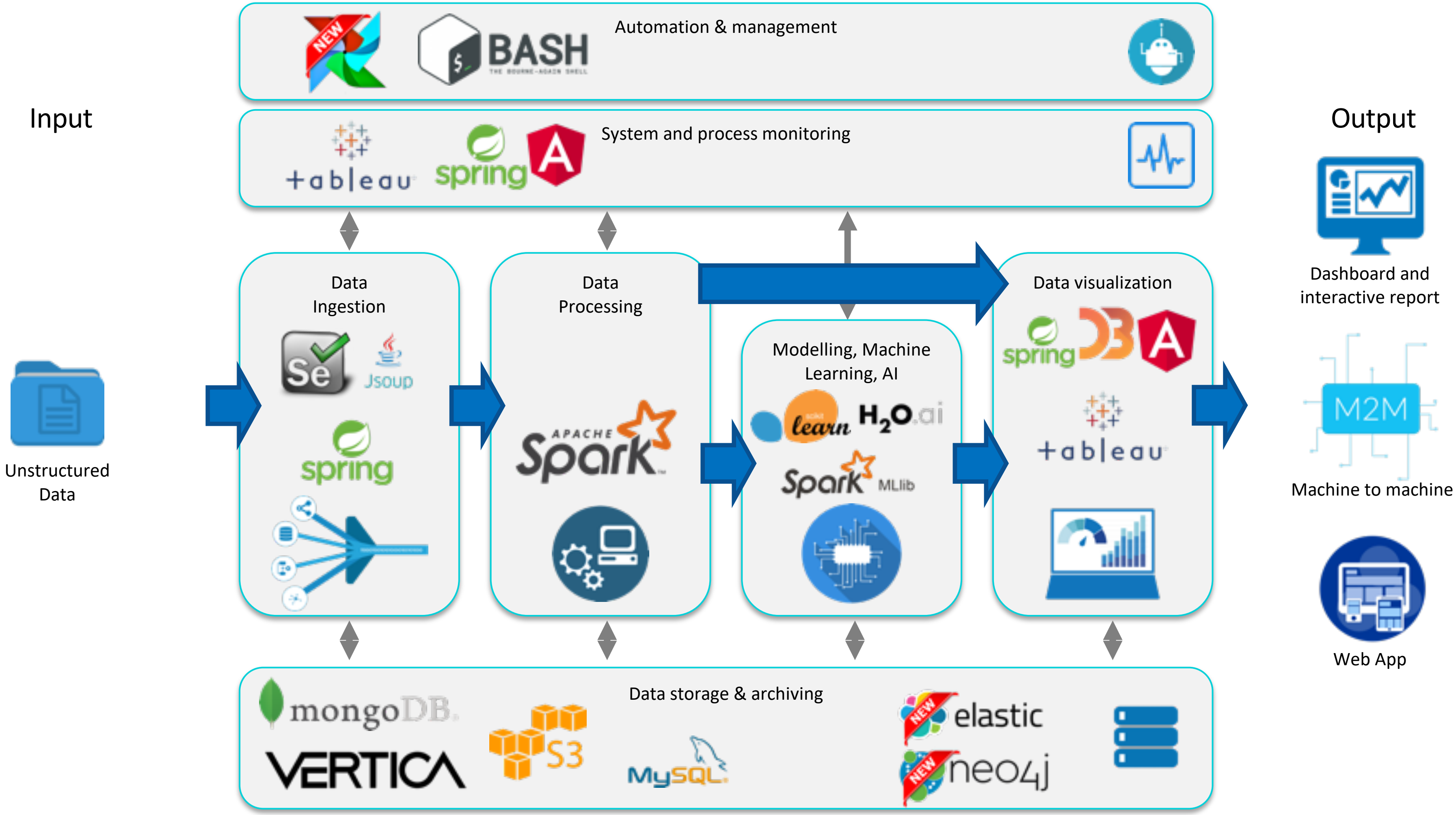
Information Extraction

2512 – Software Developer

Skills: develop software, implement web based applications, problem solving, develop user experiences

Harwell, UK

…



Input

Unstructured Data

Automation & management

System and process monitoring

Data Ingestion

Data Processing

Modelling, Machine Learning, AI

Data visualization

Data storage & archiving

Output

Dashboard and interactive report

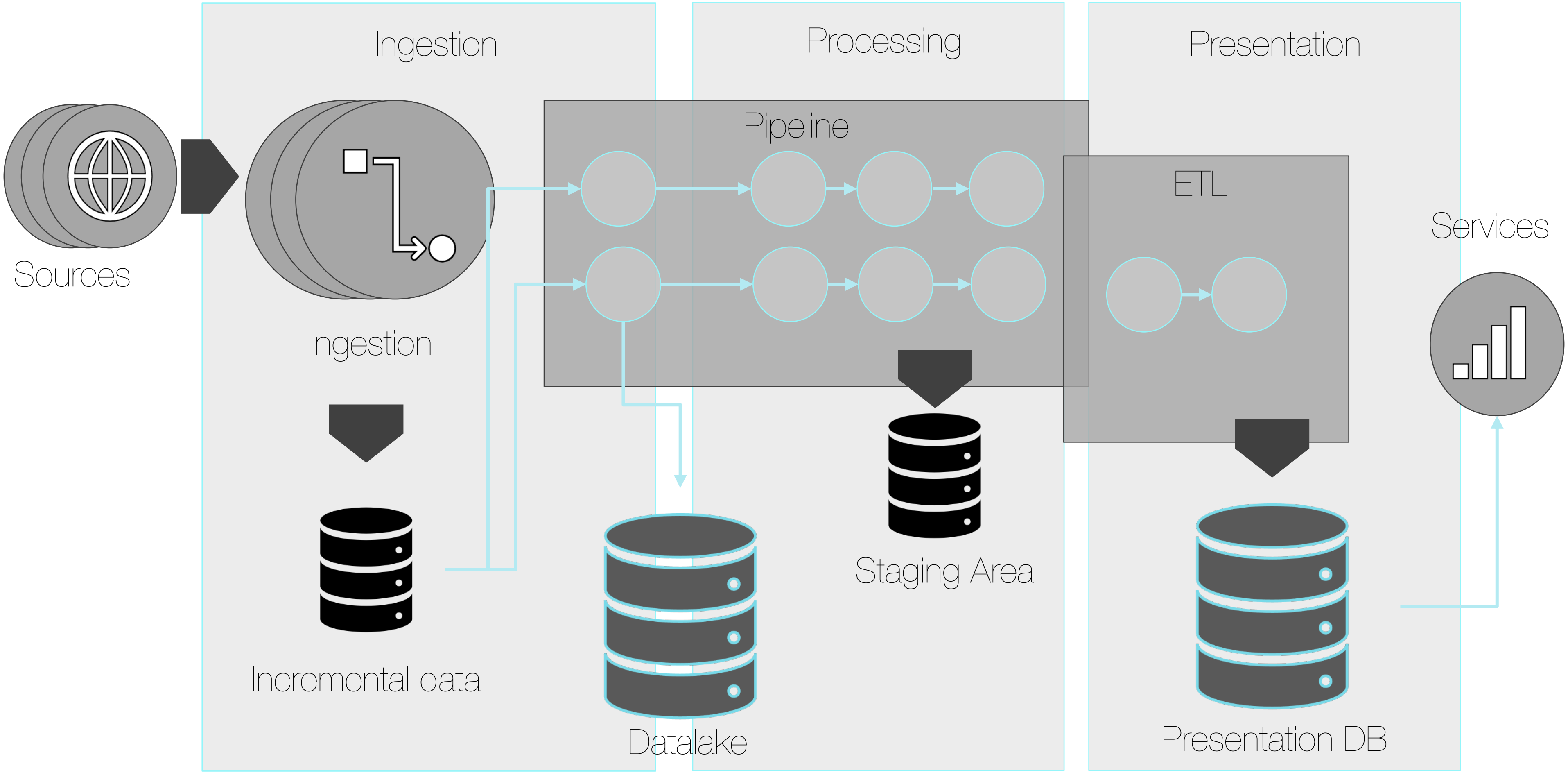
Machine to machine

Web App

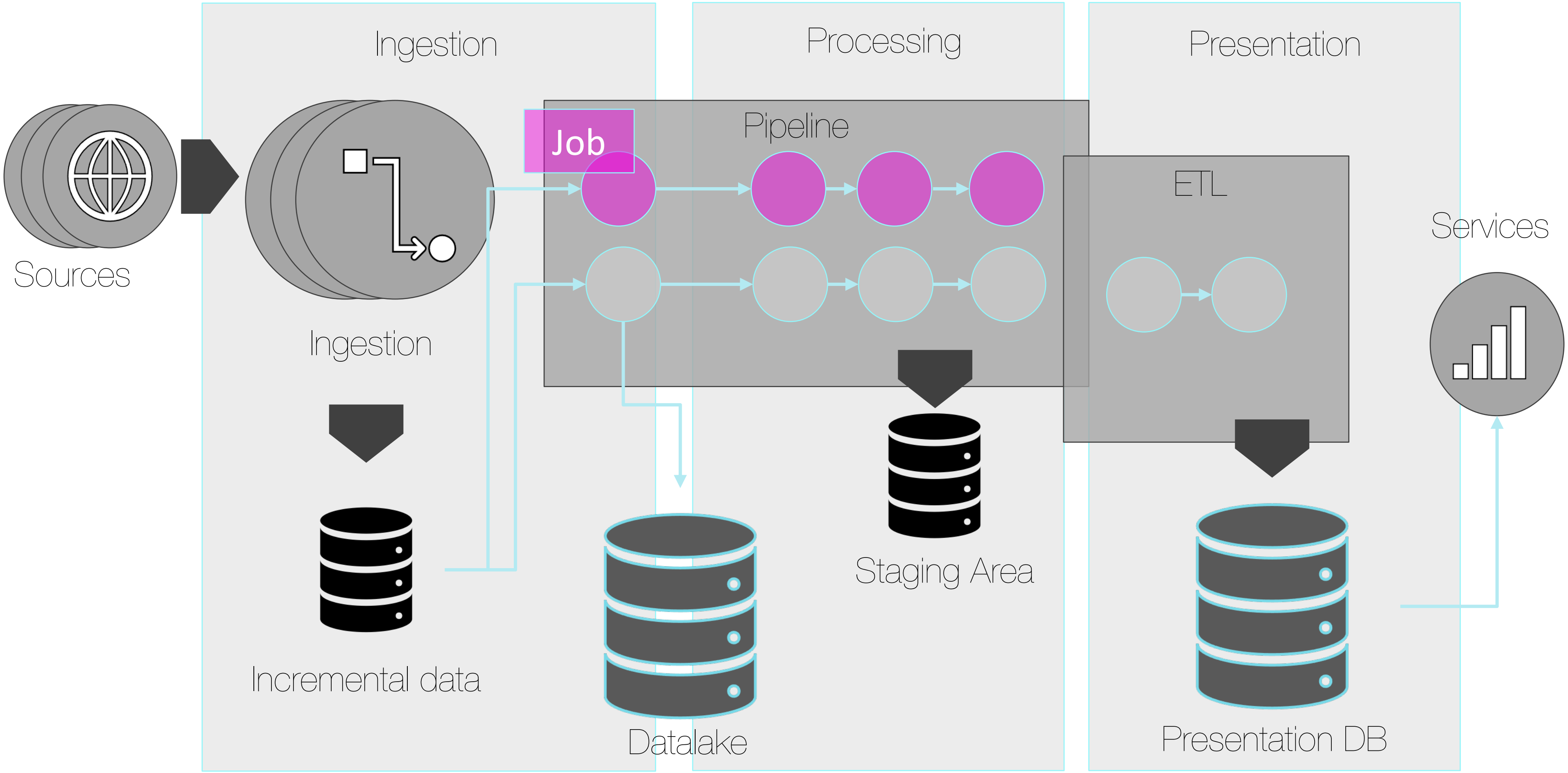
Topics

1. Overview & Recap
2. **What is a pipeline?**
3. Storage layer
4. Spark foundations
5. Lab sessions
    1. Find new job titles
    2. Find new occupations

Data product anatomy

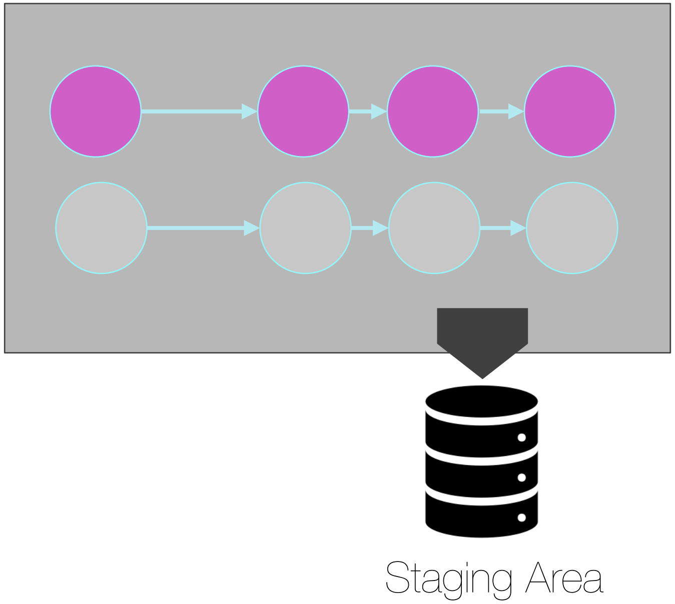


Data product anatomy



Data pipeline

Yet another computer program

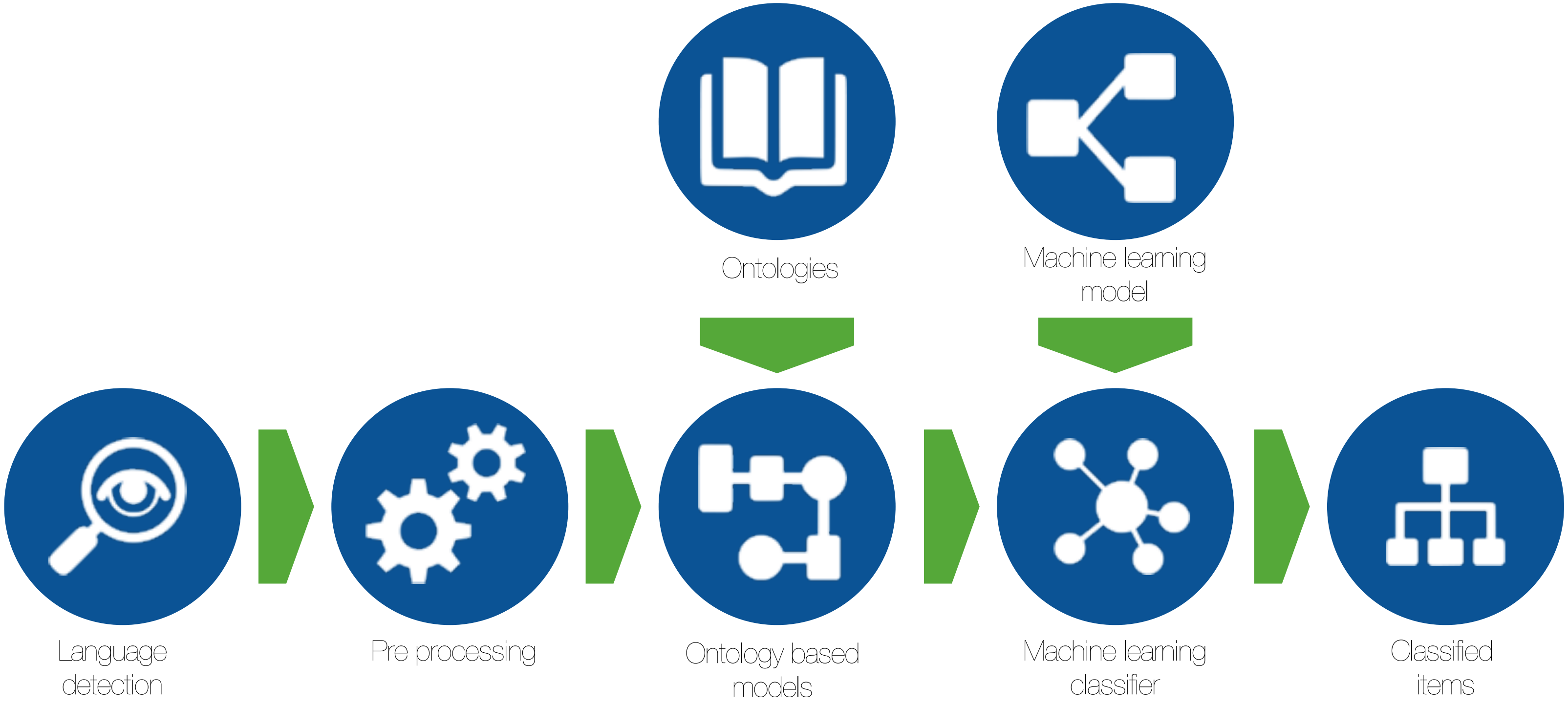


Batch job

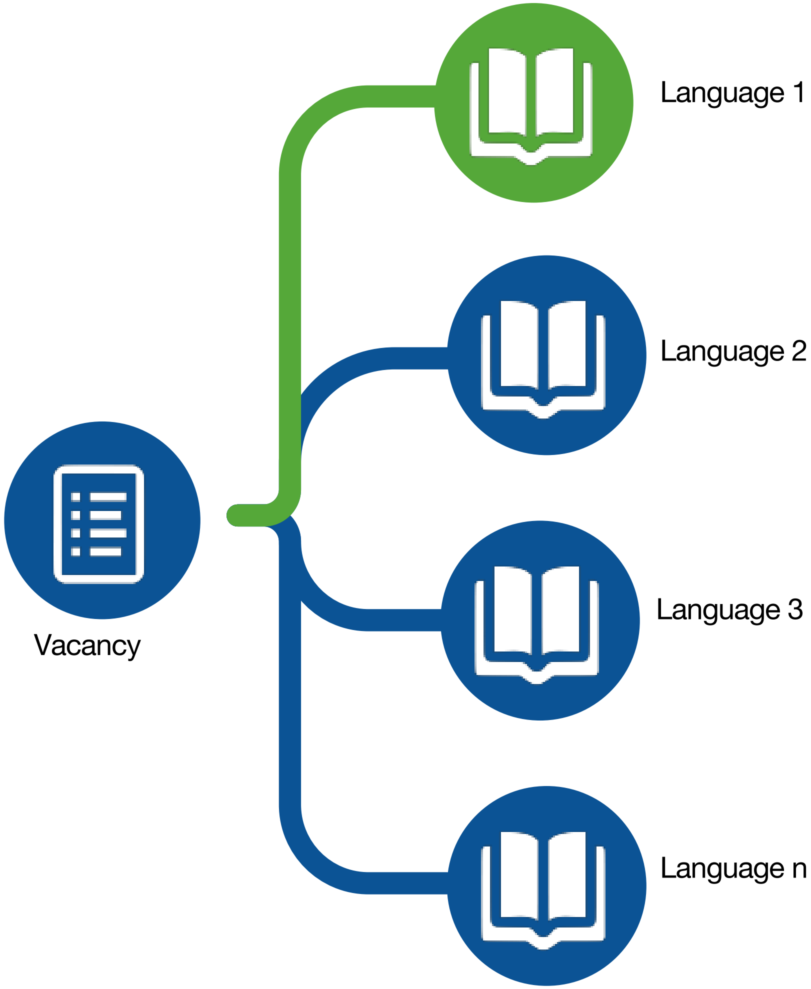
Job == function([input dataset]): [output dataset]

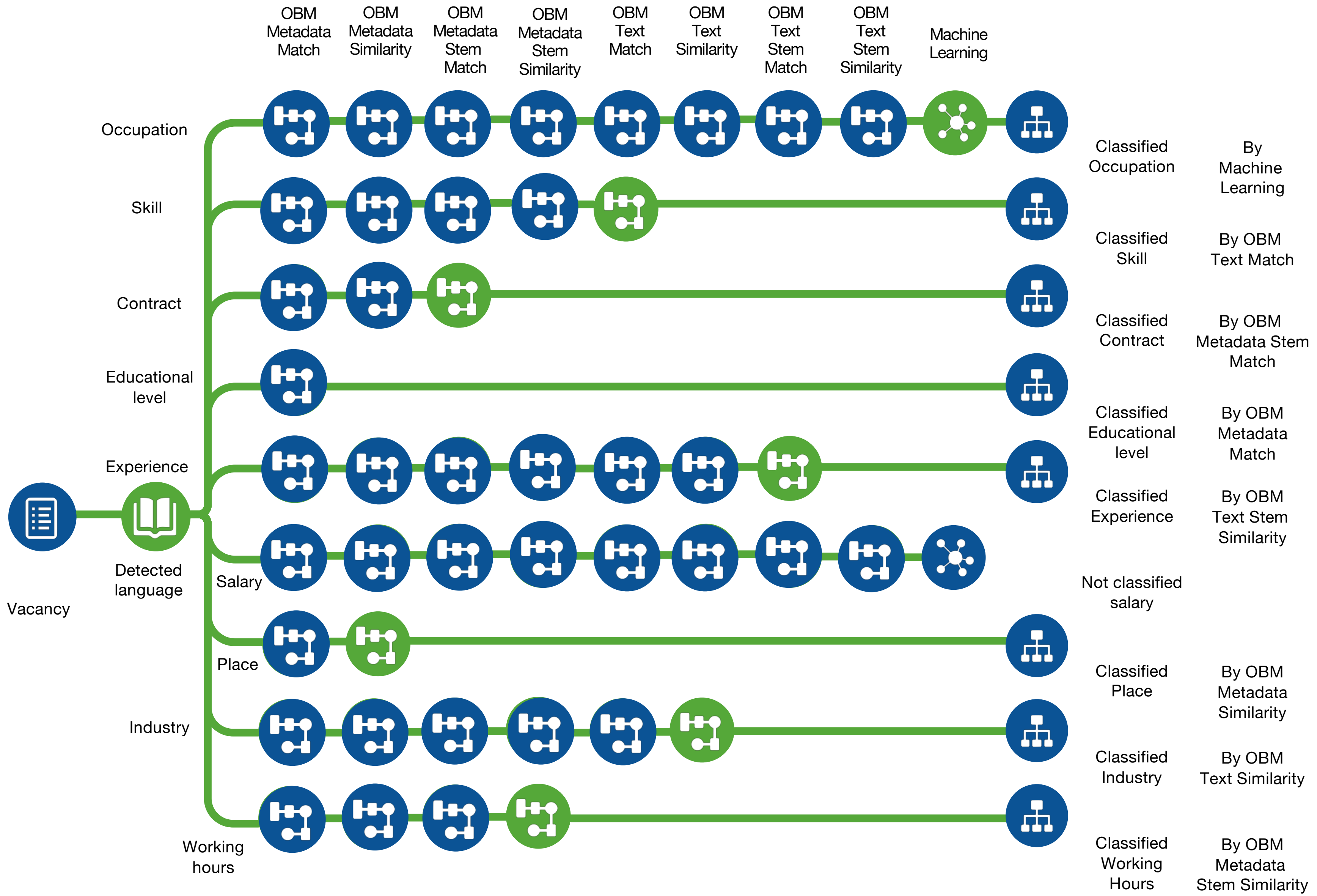
- Testable
- Atomic
- Deterministic
- Indempotent
- No other input factors

Pipeline details

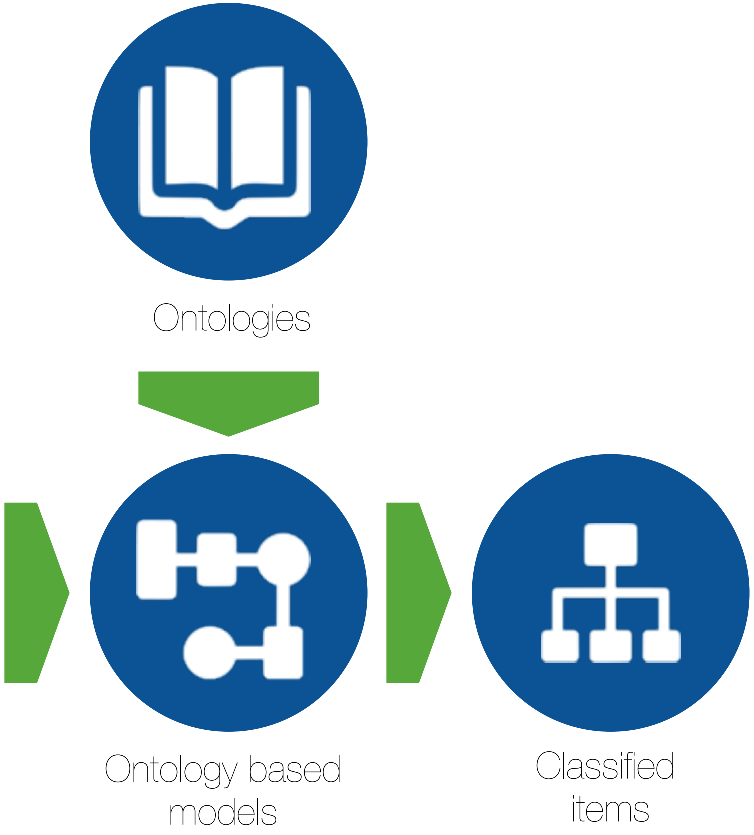


Pipeline and language detection



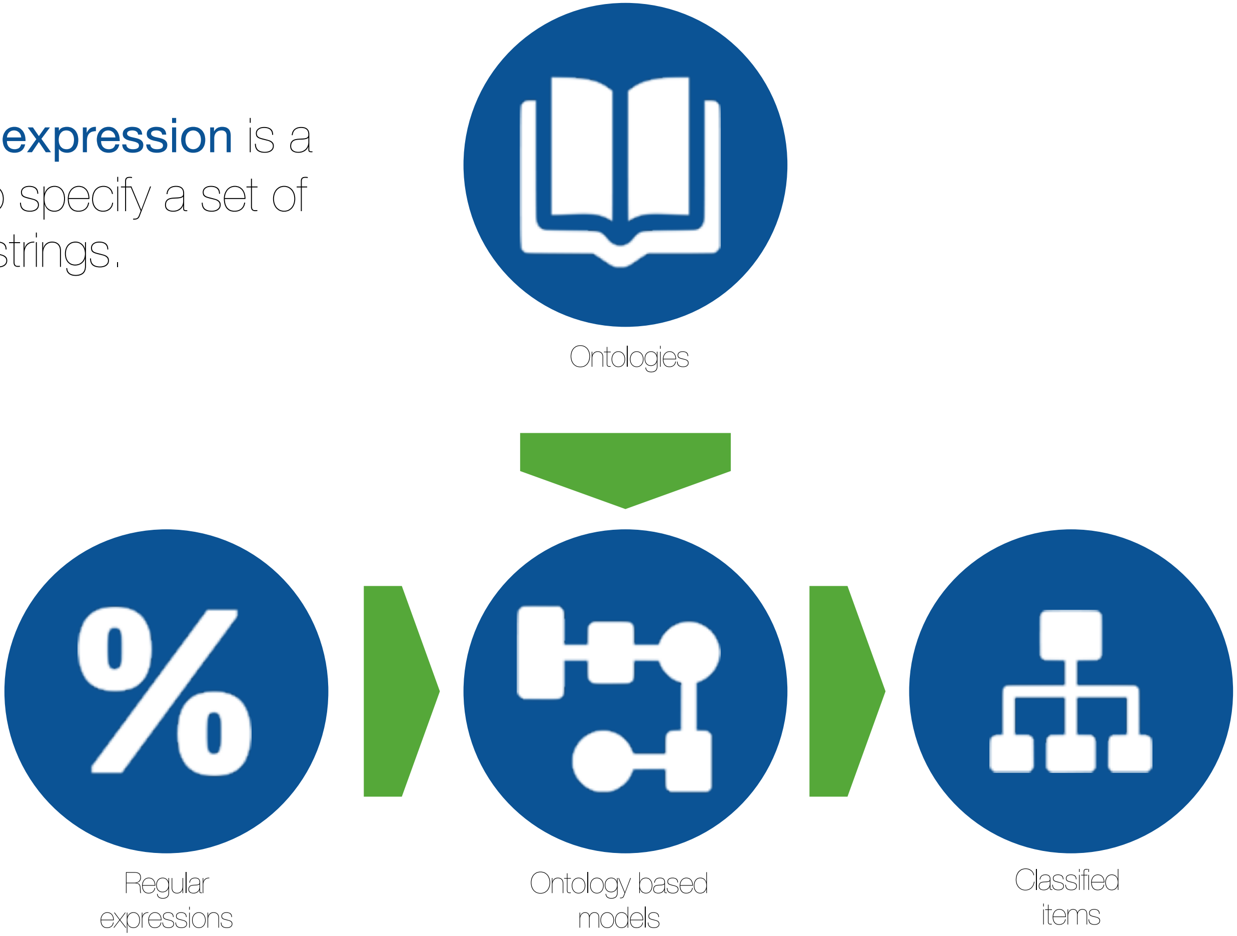

OBM Metadata Match
OBM Metadata Similarity
OBM Metadata Stem Match
OBM Metadata Stem Similarity
OBM Text Match
OBM Text Similarity
OBM Text Stem Match
OBM Text Stem Similarity
Machine Learning
Occupation
Skill
Contract
Educational level
Experience
Salary
Place
Industry
Working hours
Vacancy
Detected language
Classified Occupation
By Machine Learning
Classified Skill
By OBM Text Match
Classified Contract
By OBM Metadata Stem Match
Classified Educational level
By OBM Metadata Match
Classified Experience
By OBM Text Stem Similarity
Not classified salary
Classified Place
By OBM Metadata Similarity
Classified Industry
By OBM Text Similarity
Classified Working Hours
By OBM Metadata Stem Similarity

Ontology based components



Regular expressions

A **regular expression** is a notation to specify a set of strings.



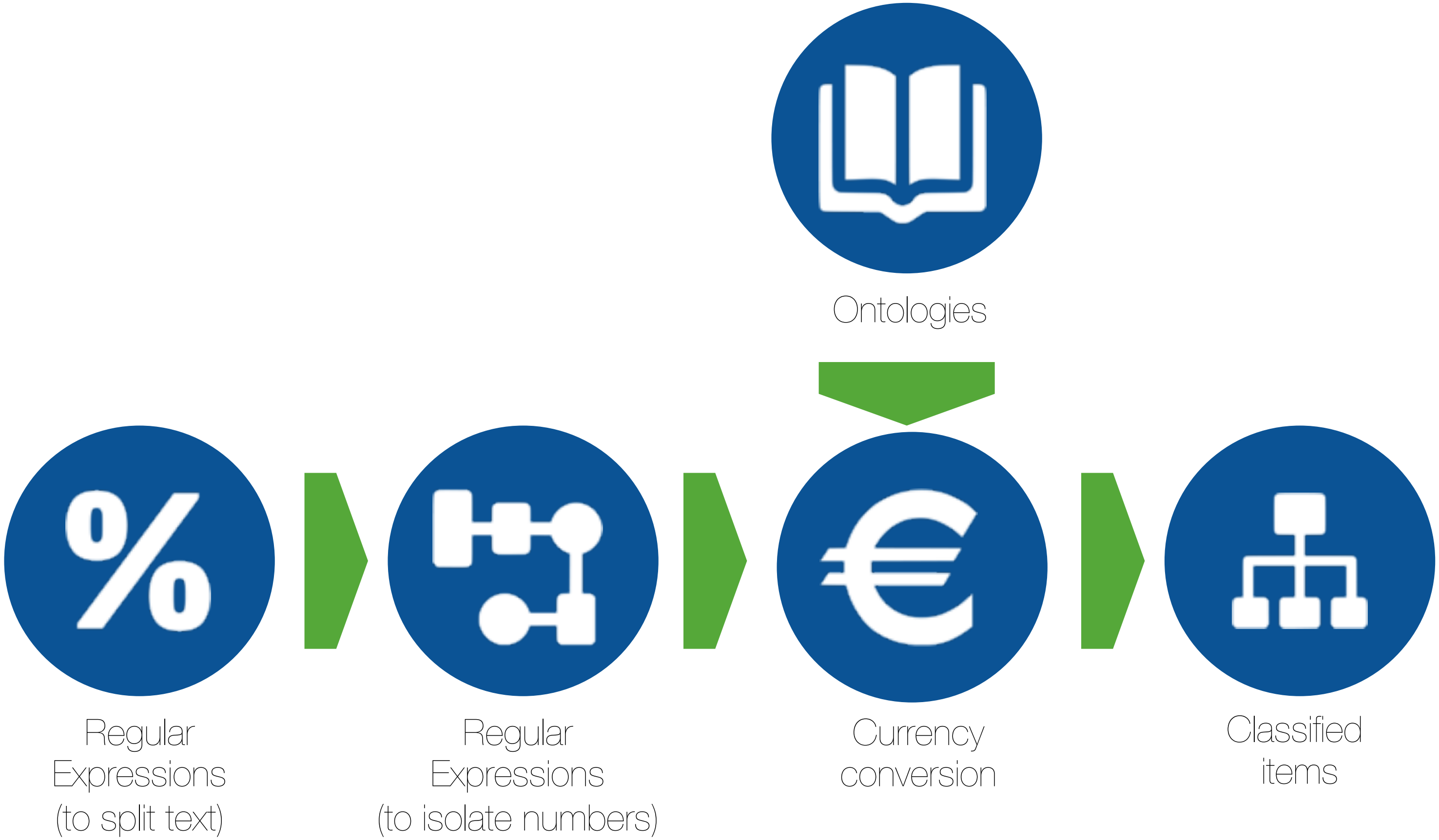
Ontologies

Regular expressions

Ontology based models

Classified items

Regular expression for salary detection



Ontologies

Regular Expressions (to split text)

Regular Expressions (to isolate numbers)

Currency conversion

Classified items

Testing (single job)

How test the pipeline?

- Test the single job / single component
- Standard dataset (gold dataset or mock dataset)
  - Generate input
  - Run in local / small cluster
  - Verify output

What's we need?
The toolkit

Statistical Methods

Tools

User Experience Research

Time series analysis

Missing data imputations

Data mining

Multilevel modeling

Classification and clustering

Pattern recognition

AB testing

Principal component and factor analysis

Machine learning

Forecasting

Network analysis

Regression techniques

What’s we need?
The toolkit

Statistical Methods | Tools | User Experience Research

Languages

Python
R
Scala
SQL

Libraries

Pandas
Sklearn
OpenNLP
Spacy
Fasttext
Word2Vec
H20.ai
. . .

Data Engineering

Hadoop
Spark
Profiling
ETL
Job notices
APIs
Optimized data pipelines
Optimized data storage/access

Cloud (AWS)
CI/CD

Visualization

D3.js
Gephi
R
Matplot
Shiny
Tableau

What’s we need?
The toolkit

Statistical Methods

Tools

User Experience Research

Iteractive Prototyping

Service blueprinting

User observation

Journey mapping

Topics

1. Overview & Recap
2. What is a pipeline?
3. **Storage layer**
4. Spark foundations
5. Lab sessions
    1. Find new job titles
    2. Find new occupations

Key concepts

- Columnar Data Formats
- Delta lake

Concepts
Columnar Data Formats

- Filters are not the only "predicate" that can be pushed down
- Column selection can also be pushed down
  - With a database like PostgreSQL, this is done with a SELECT statement
  - For files, we require a Columnar File Format
- Data is stored by column, not by row
    - Parquet and ORC
    - Delta lake format: Delta.io, Hudi, Iceberg
- Compared to Row-Based File formats that store data by row
  - CSV, TSV, JSON, and AVRO

Concepts
An Example: Columnar vs Row-Based

Row-Based

	name	color	city	age
Row 1	Tom	red	Chicago	32
Row 2	Sally	blue	Paris	87
Row 3	Mike	green	London	20
Row 4	Mary	yellow	Fresno	55

Reads Row #1

Columnar

	Row 1	Row 2	Row 3	Row 4
name	Tom	Sally	Mike	Mary
color	red	blue	green	yellow
city	Chicago	Paris	London	Fresno
age	32	87	20	55

Reads the “name” column

What is **Delta Lake**?

Technology designed to be used with Apache Spark to build robust data lakes

Open source project at delta.io

Databricks Delta Lake documentation

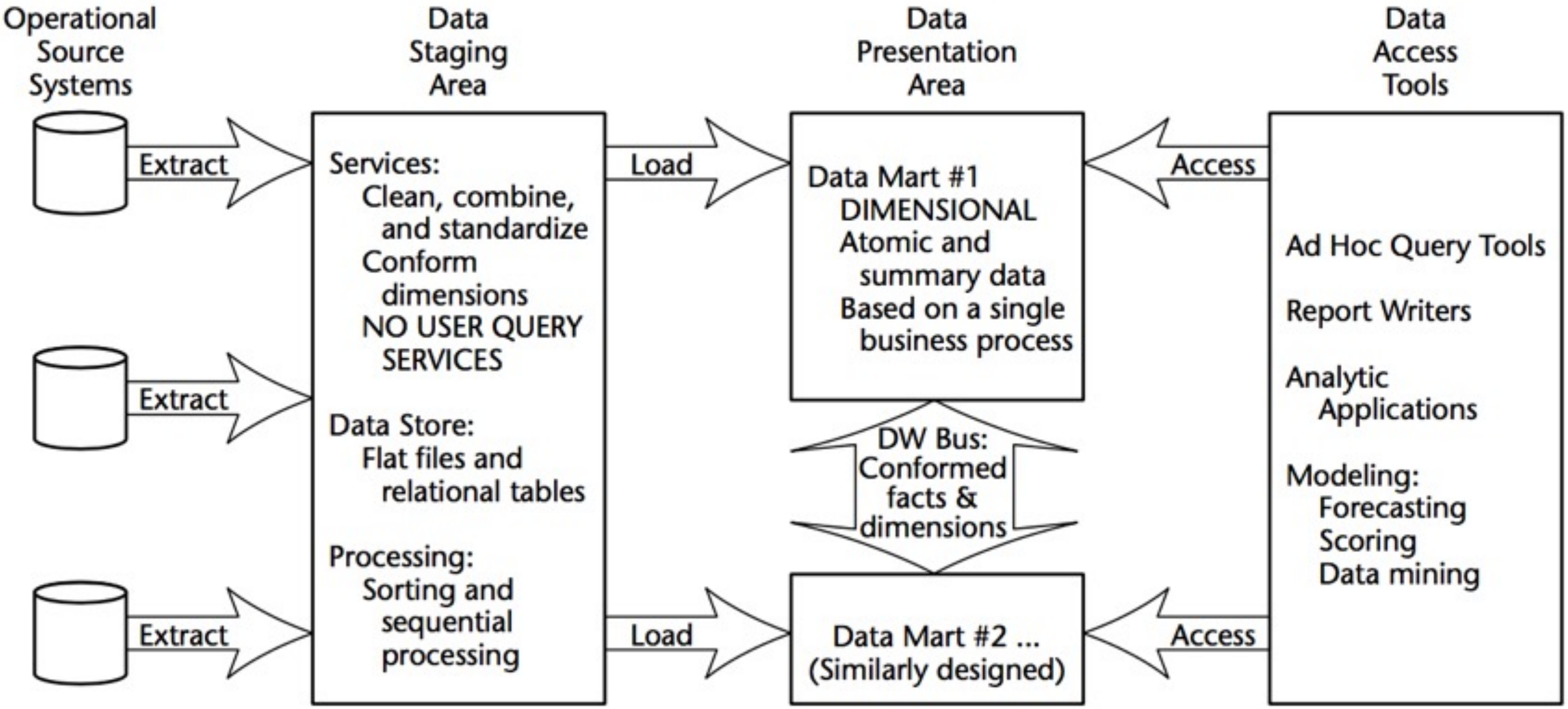
Delta Lake features

- ACID transactions on Spark
- Scalable metadata handling
- Streaming and batch unification
- Schema enforcement
- Time travel
- Upserts and deletes
- Fully configurable/optimizable
- Structured streaming support

Staging area

STAGING AREA = pipelines, ETL data and processes
is like a restaurant kitchen

- Data in the staging area must not be accessible to the end user: they are not ready to be consumed.
- "Dangerous" operations take place in the staging area: data cleaning, lookups and joins, creation of data marts, …
- Business users do not (and should not) care what happens during pipelines and ETLs.



Operational Source Systems

Data Staging Area

Data Presentation Area

Data Access Tools

Extract

Extract

Extract

Services:
Clean, combine, and standardize
Conform dimensions
NO USER QUERY SERVICES

Data Store:
Flat files and relational tables

Processing:
Sorting and sequential processing

Load

Load

Data Mart #1
DIMENSIONAL
Atomic and summary data
Based on a single business process

DW Bus:
Conformed facts & dimensions

Data Mart #2 ...
(Similarly designed)

Access

Access

Ad Hoc Query Tools

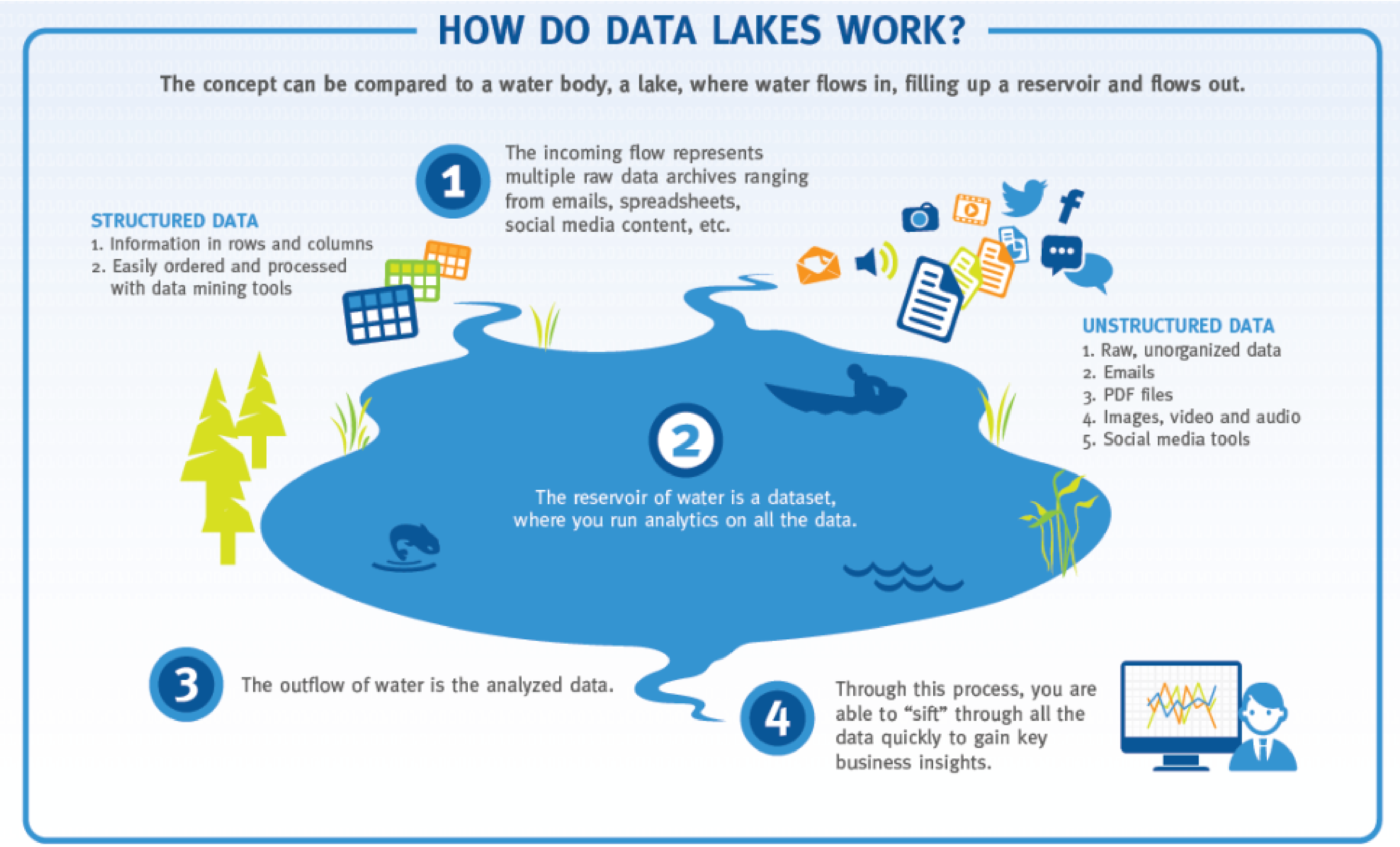
Report Writers

Analytic Applications

Modeling:
Forecasting
Scoring
Data mining

Kimball/Ross

Data lake



HOW DO DATA LAKES WORK?

The concept can be compared to a water body, a lake, where water flows in, filling up a reservoir and flows out.

1 The incoming flow represents multiple raw data archives ranging from emails, spreadsheets, social media content, etc.

STRUCTURED DATA

1. Information in rows and columns
2. Easily ordered and processed with data mining tools

UNSTRUCTURED DATA

1. Raw, unorganized data
2. Emails
3. PDF files
4. Images, video and audio
5. Social media tools

2 The reservoir of water is a dataset, where you run analytics on all the data.

3 The outflow of water is the analyzed data.

4 Through this process, you are able to "sift" through all the data quickly to gain key business insights.

Credits: EMC

https://40uu5c99f3a2ja7s7miveqgqu-wpengine.netdna-ssl.com/wp-content/uploads/2017/02/Understanding-data-lakes-EMC.pdf

The Data Lake Paradigm

Data Warehouse

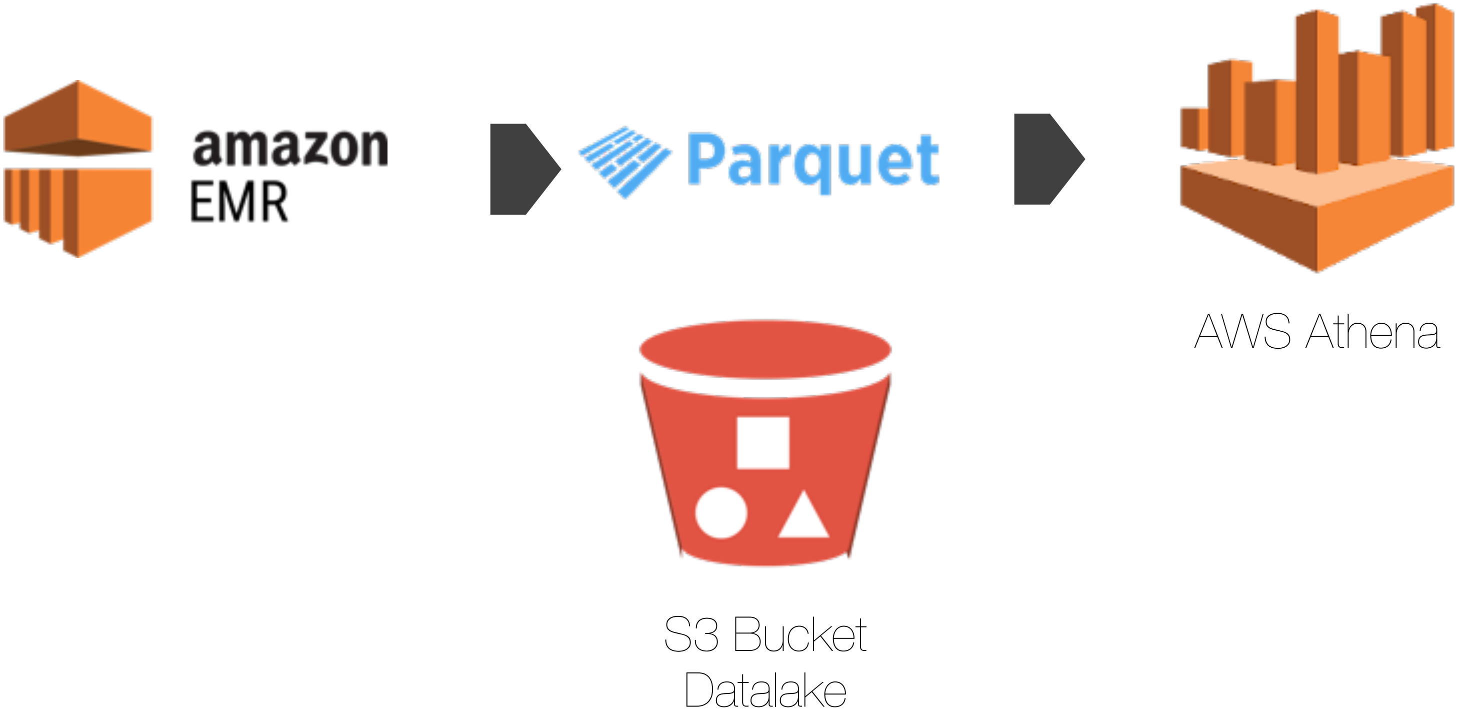
- Aggregated Subsets
- On-Demand Views
- Curated By Experts
- Structured - Tables, Views, Reports. Limited Context
- Data Quality Is Known And Tracked

Data Lake

- Store Everything As-is
- Let Business Decide What They Need
- Support Rapid Change
- Provide Data Lineage and History Tracking and Visualization
- Unstructured - Key-Word Search
- Data Is Available In Various States from Raw to Fully Conformed
- Quality Metrics Often Not Available

Modern Day Data Lake Architecture

- Schema-on-Read
- Descriptive Data Modeling
- New Data can start flowing any time and will appear retroactively
- Flexibility
- Scalability
- Rapid Data Ingestion
- Good for Exploration and Botton-Up Approach



AWS Athena

S3 Bucket
Datalake

Recap & Keywords

- Pipelines and jobs
    - Yet another computer programs
    - Batch job
- Different types of components
    - Machine Learning Based, ontology based, reg-ex, …
- Testing a pipeline
- Storage
    - Different format: json, parquet and delta.io
    - Diffeent scope: metadata, data lake, staging area

Questions?

Topics

1. Overview & Recap
2. What is a pipeline?
3. Storage layer
4. **Spark foundations**
5. Lab sessions
    1. Find new job titles
    2. Find new occupations

APACHE Spark™

De-facto standard unified analytics engine for big data processing

Largest open-source project in data processing

Key concepts & terms

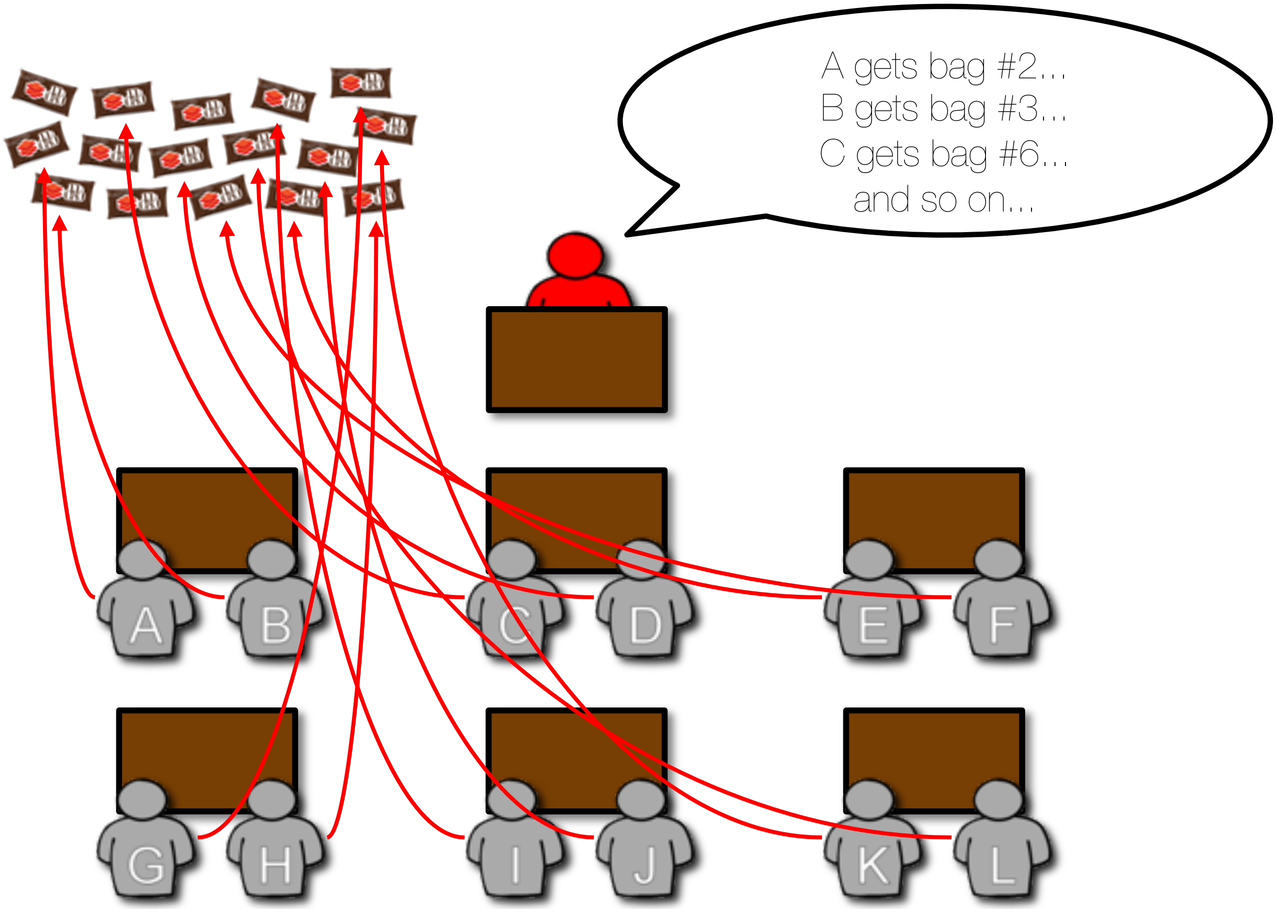
- Shared resources
- Parallelization
- Partitions
- Jobs, Stages, and Tasks
- Drivers
- Executors
- Cluster & Nodes
- Cores/Threads

Can you open the bag…

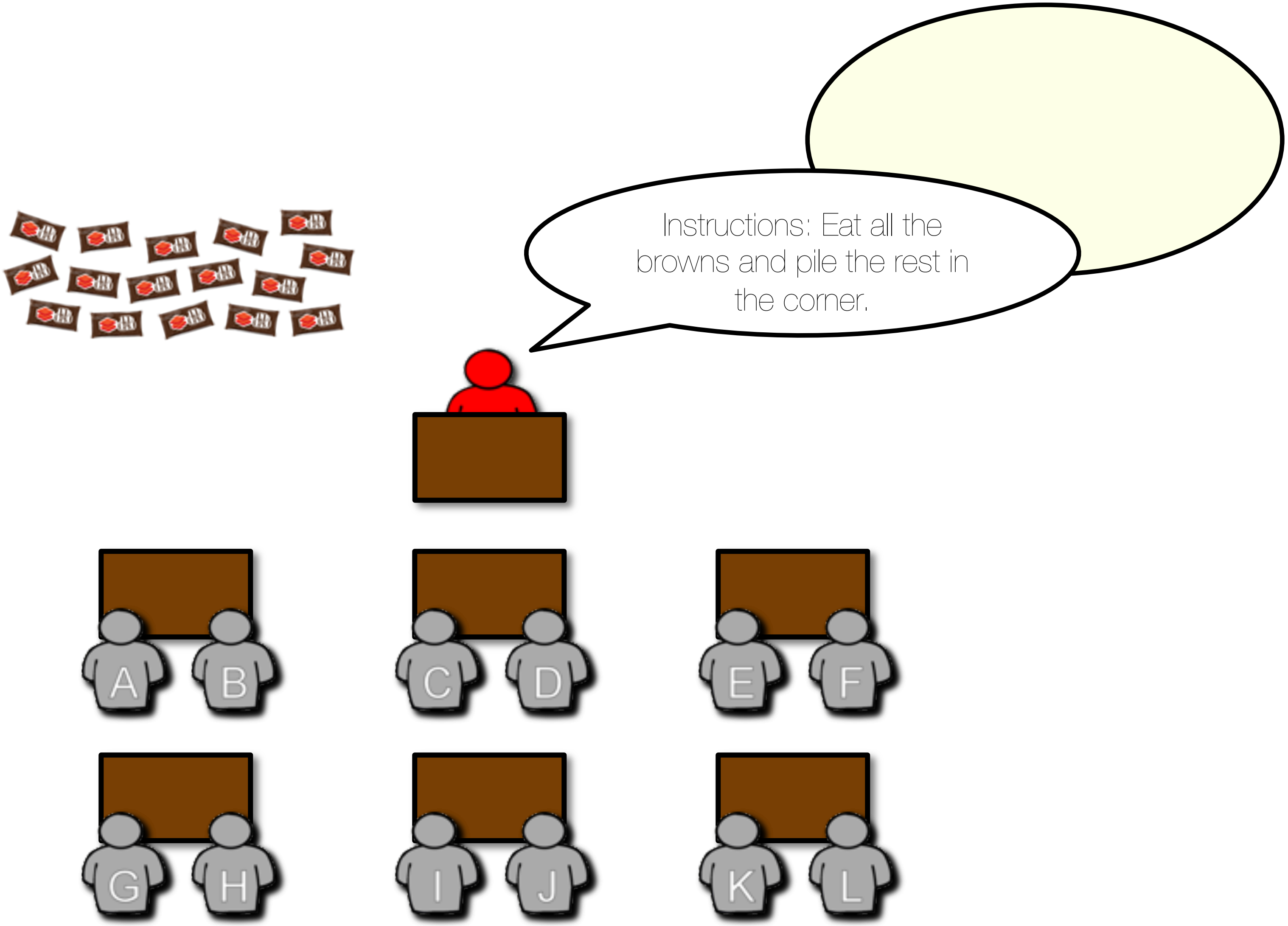
…and eat all the brown

…in 60 seconds?

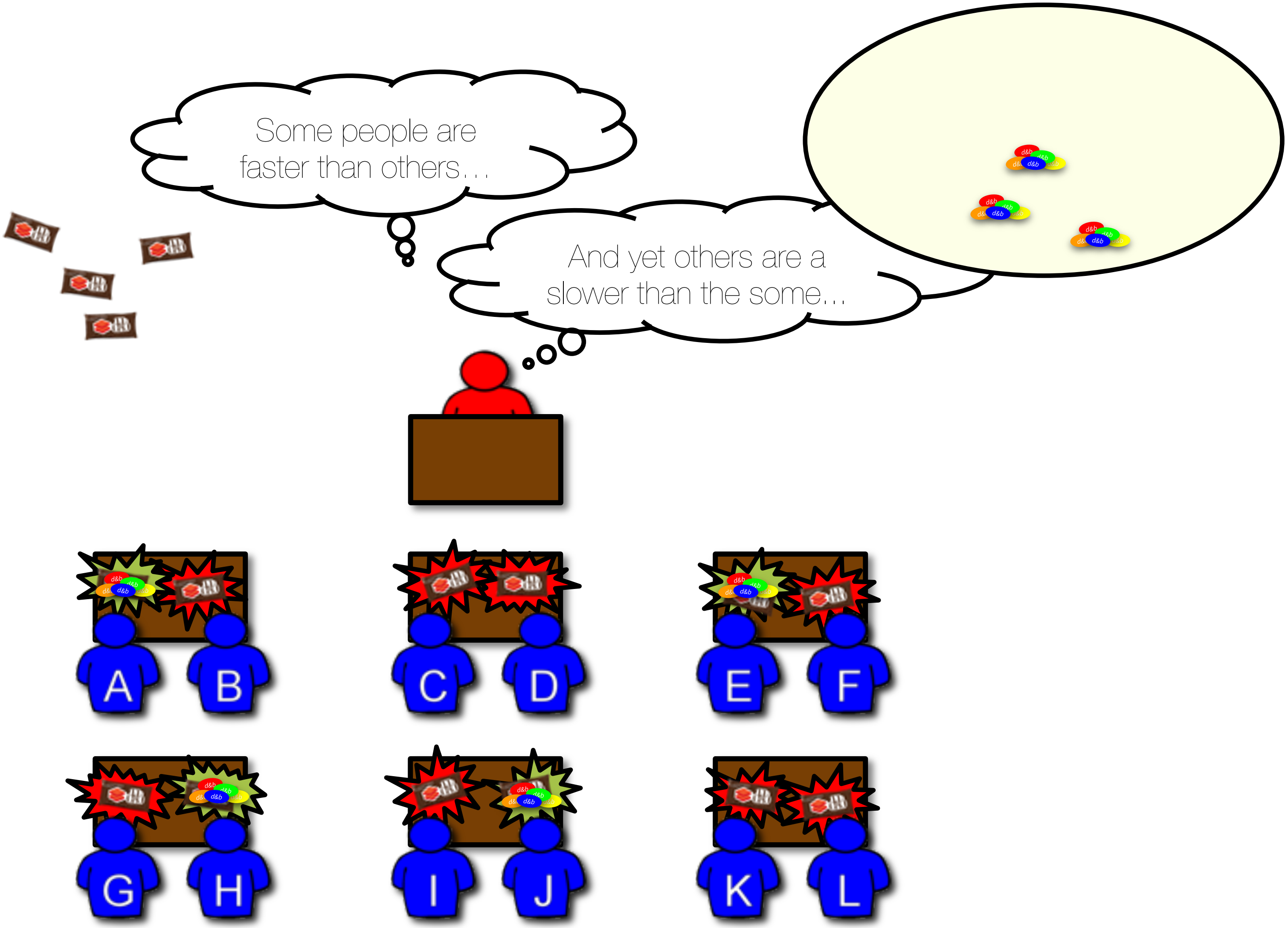
Now about 100 bags of M&Ms
Withing 60 secods?



A gets bag #2...
B gets bag #3...
C gets bag #6...
and so on...

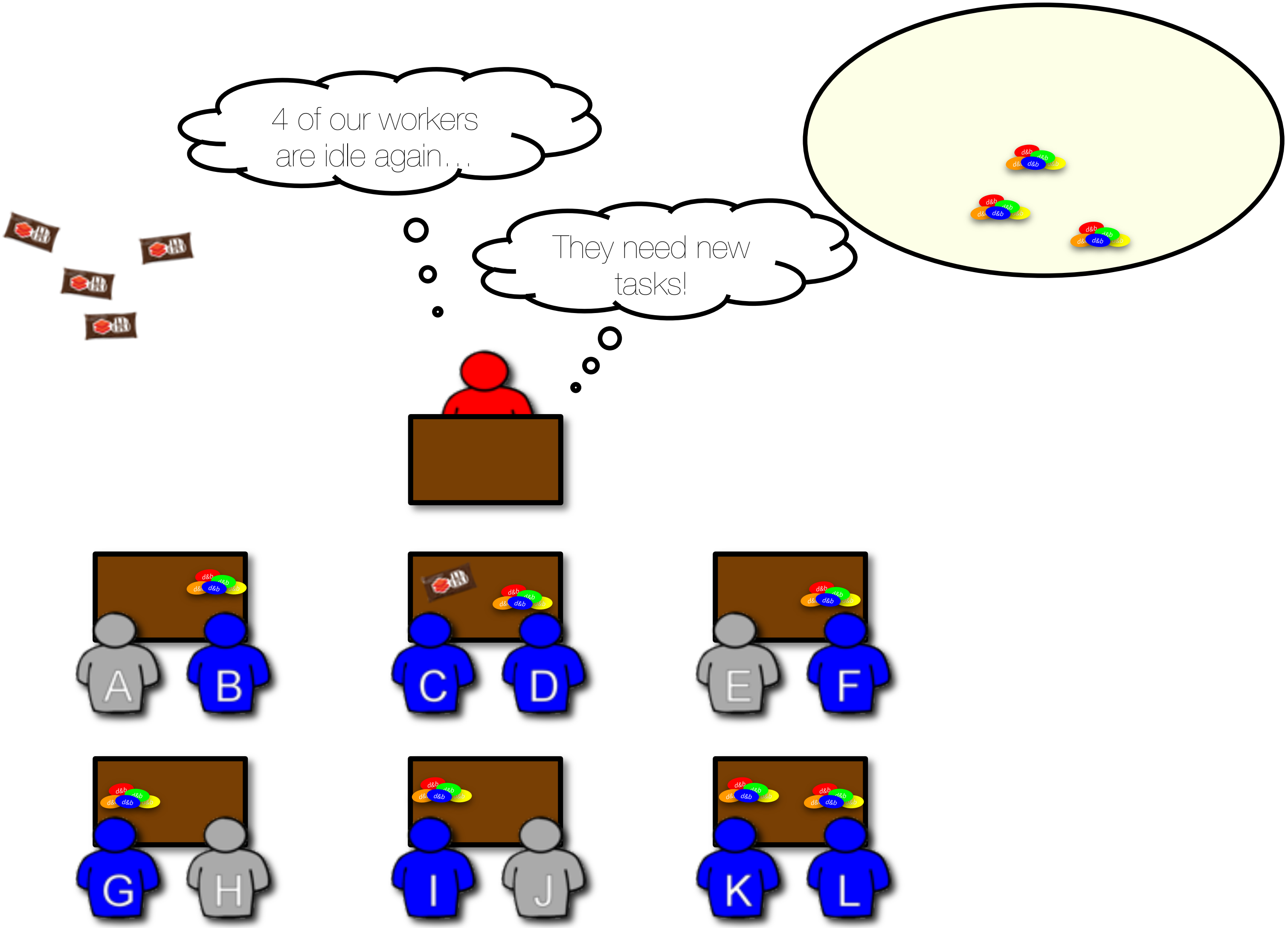


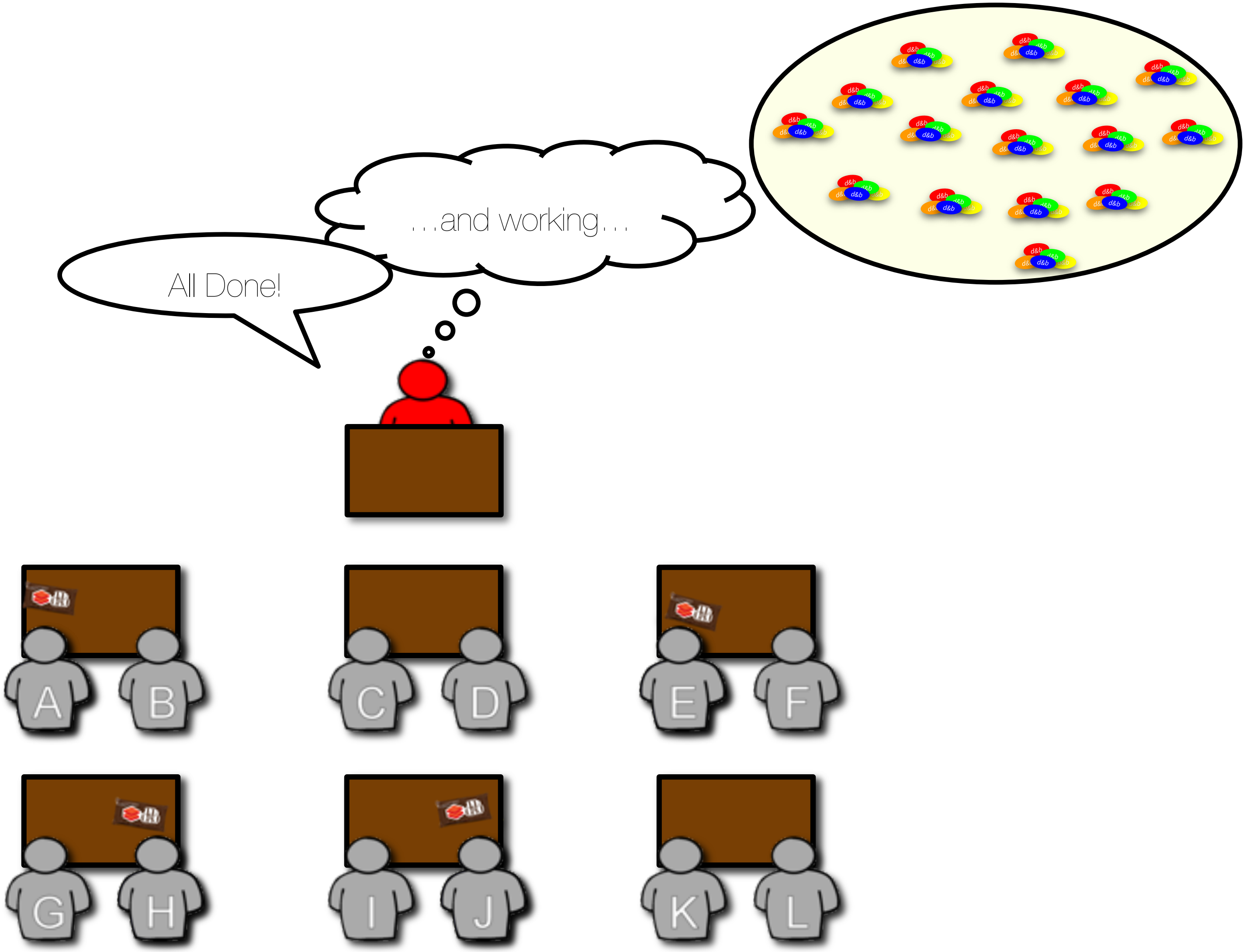
Instructions: Eat all the browns and pile the rest in the corner.

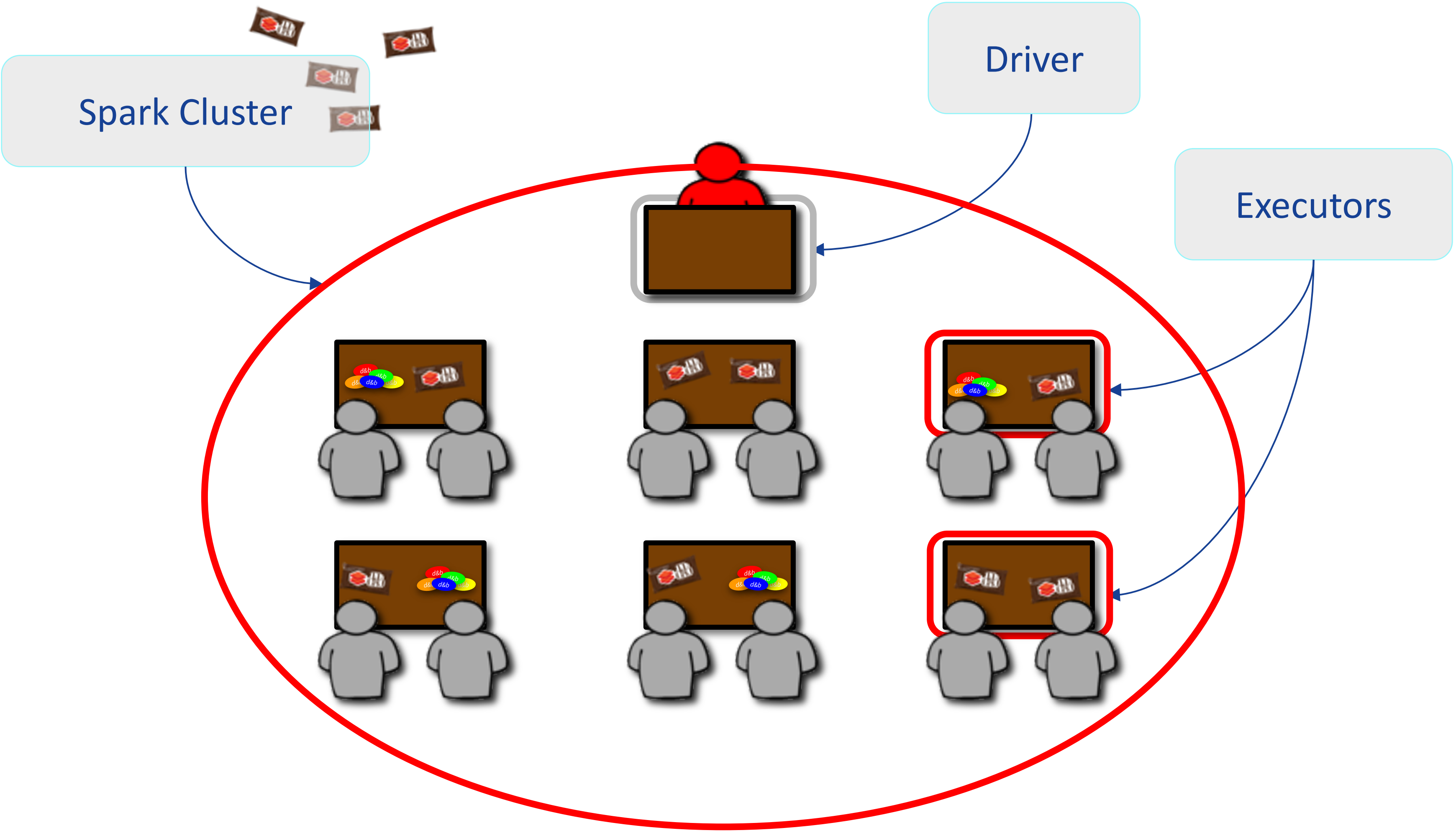


Some people are faster than others…

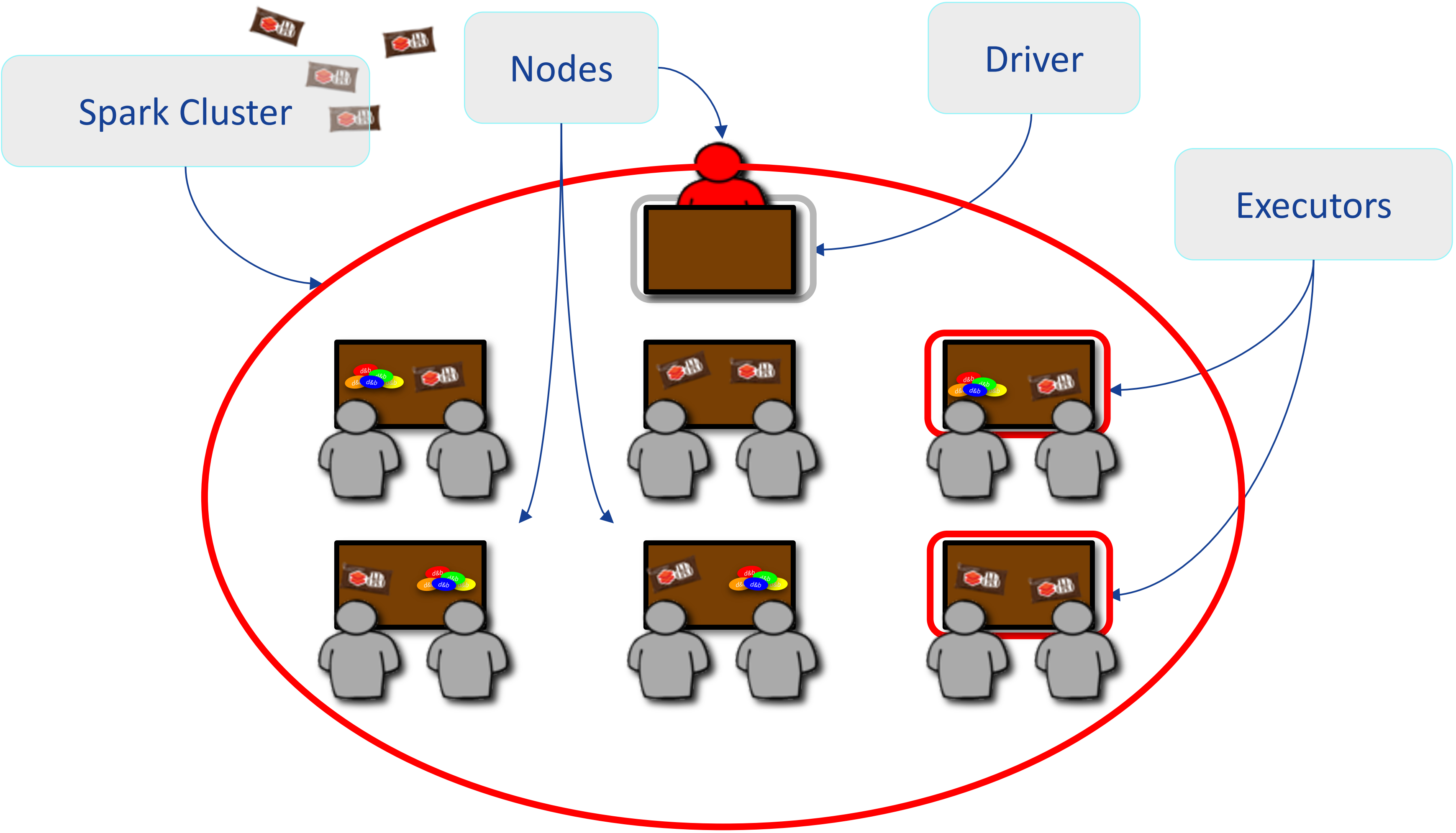
And yet others are a slower than the some…


4 of our workers are idle again…
They need new tasks!
A
B
C
D
E
F
G
H
I
J
K
L


All Done!
…and working…
A
B
C
D
E
F
G
H
I
J
K
L



Spark Cluster
Driver
Executors
d&b

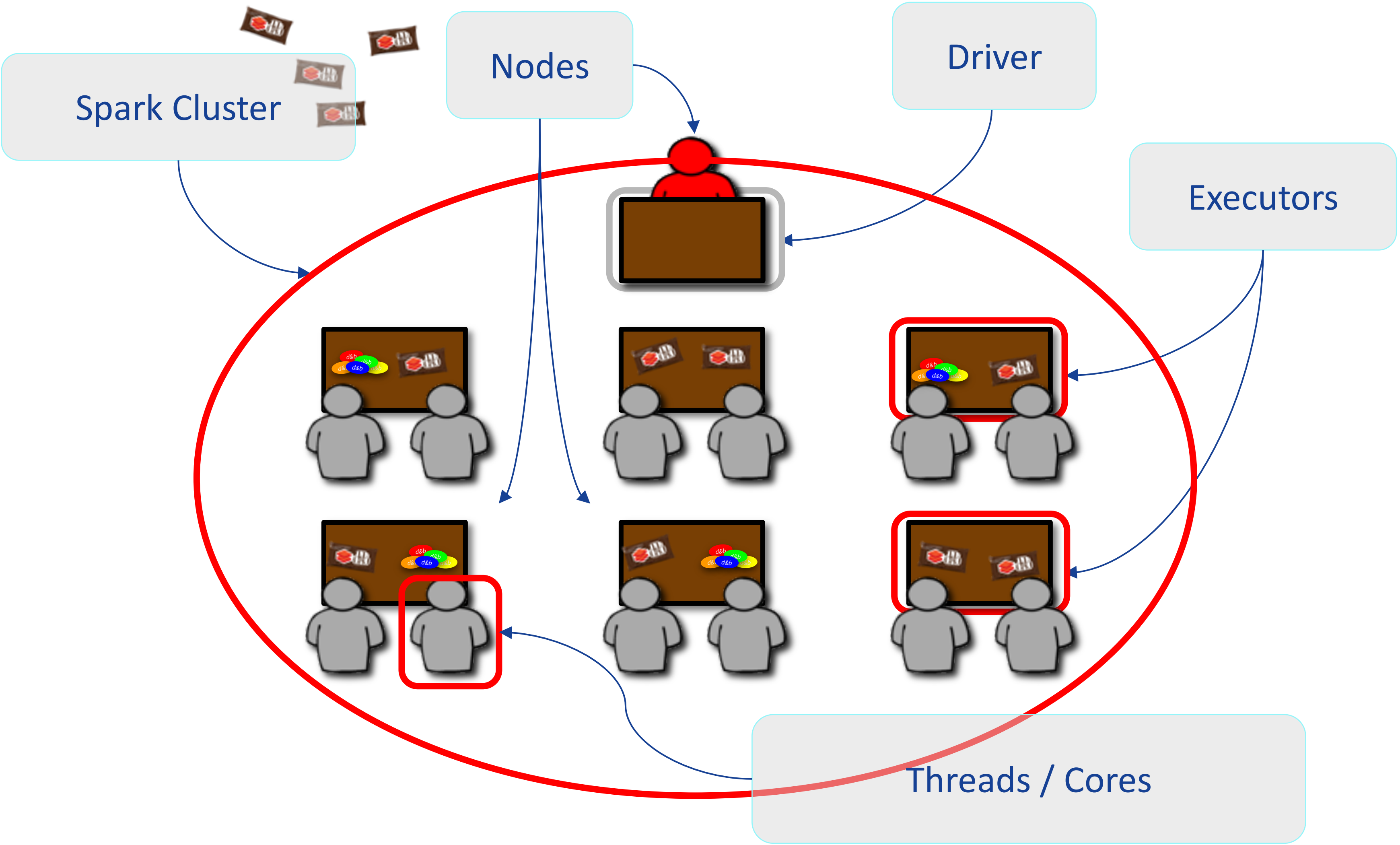


Spark Cluster

Nodes

Driver

Executors



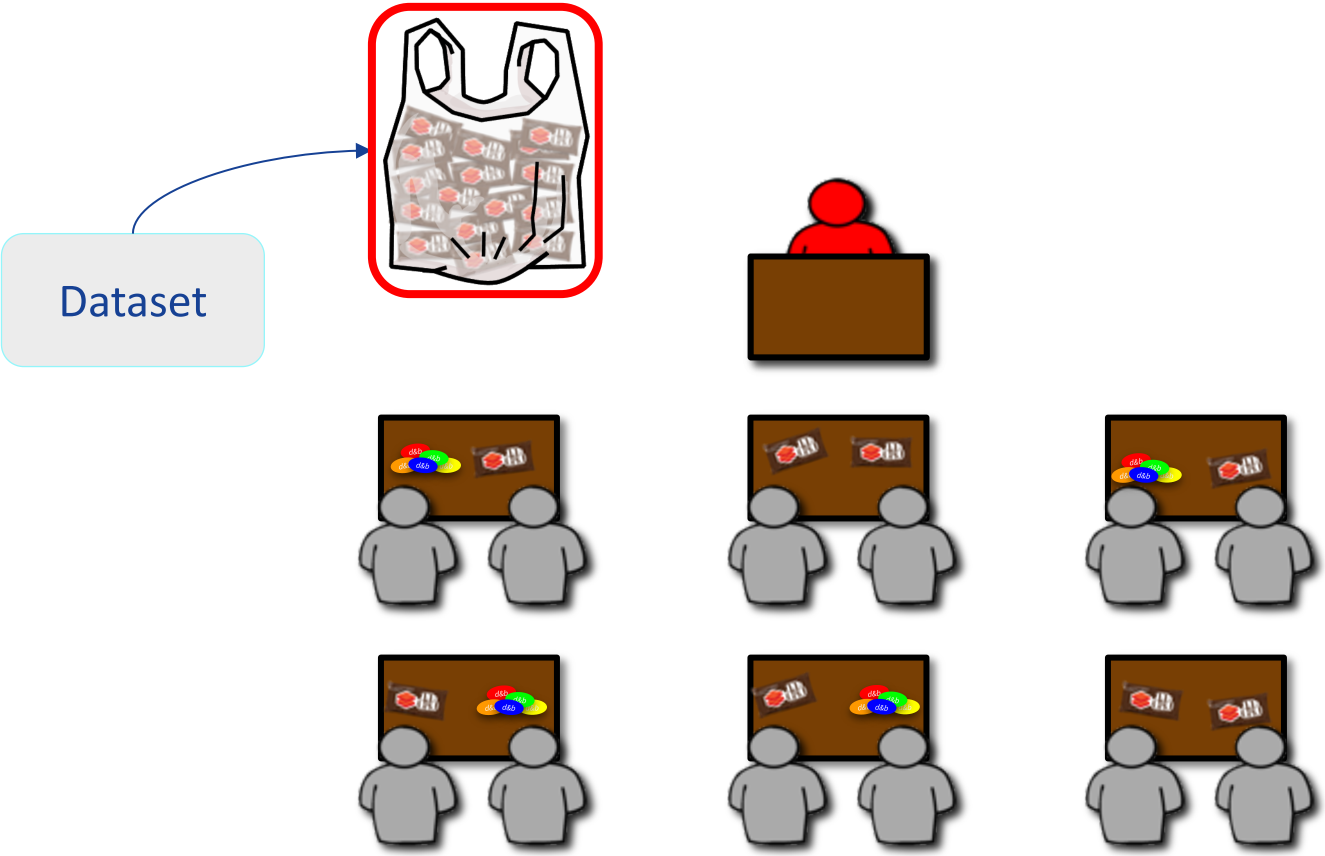
Spark Cluster

Nodes

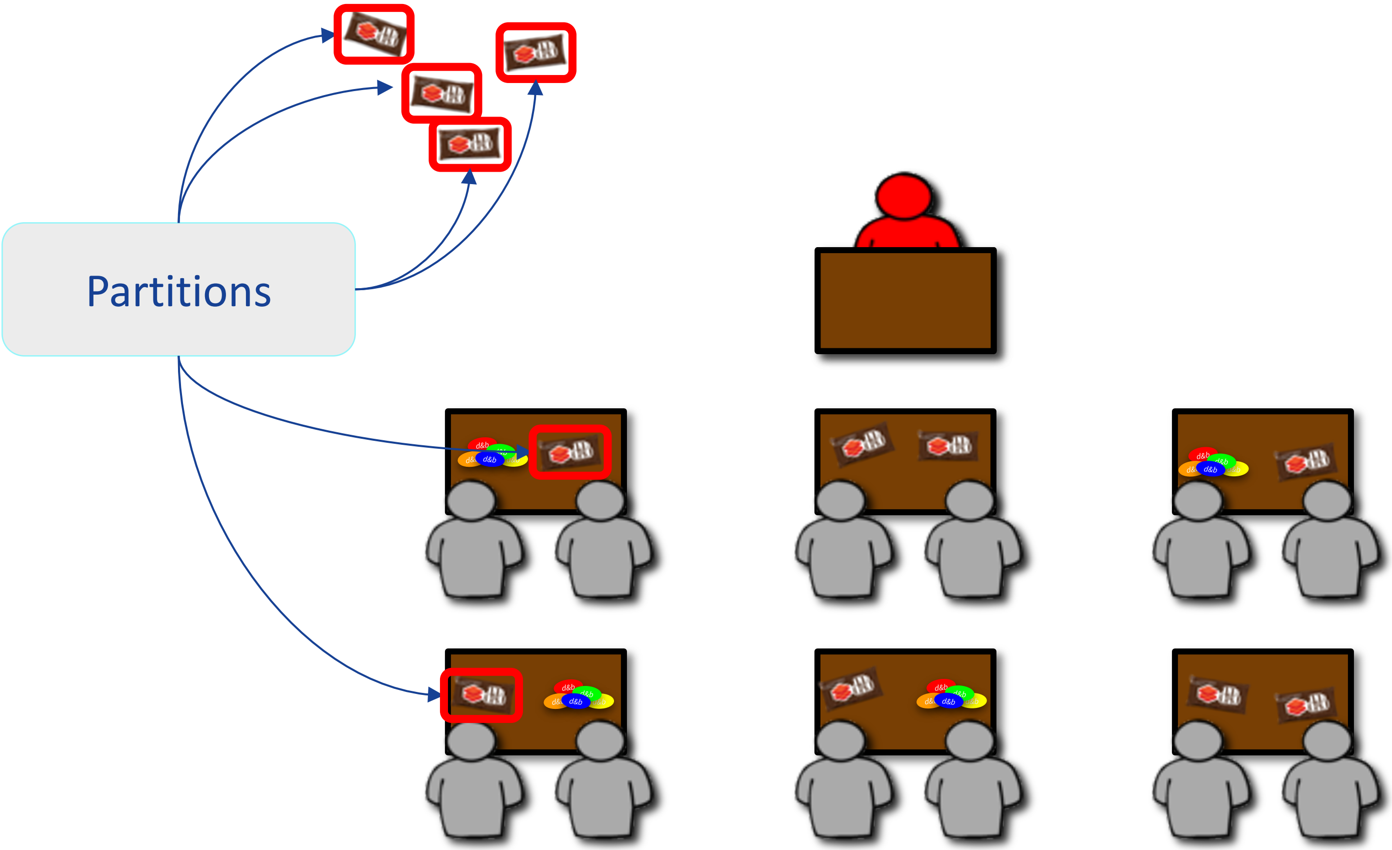
Driver

Executors

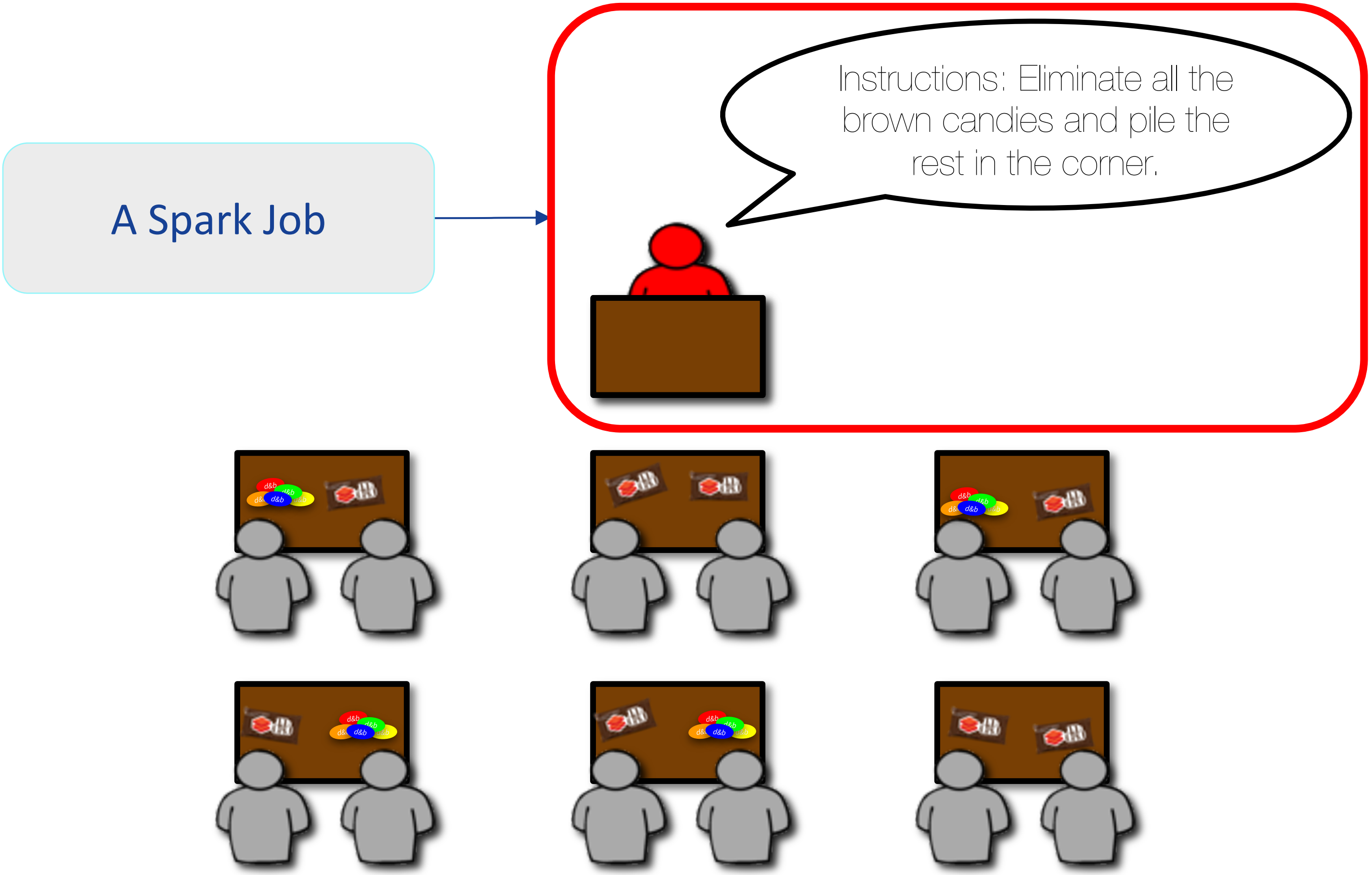
Threads / Cores



Dataset



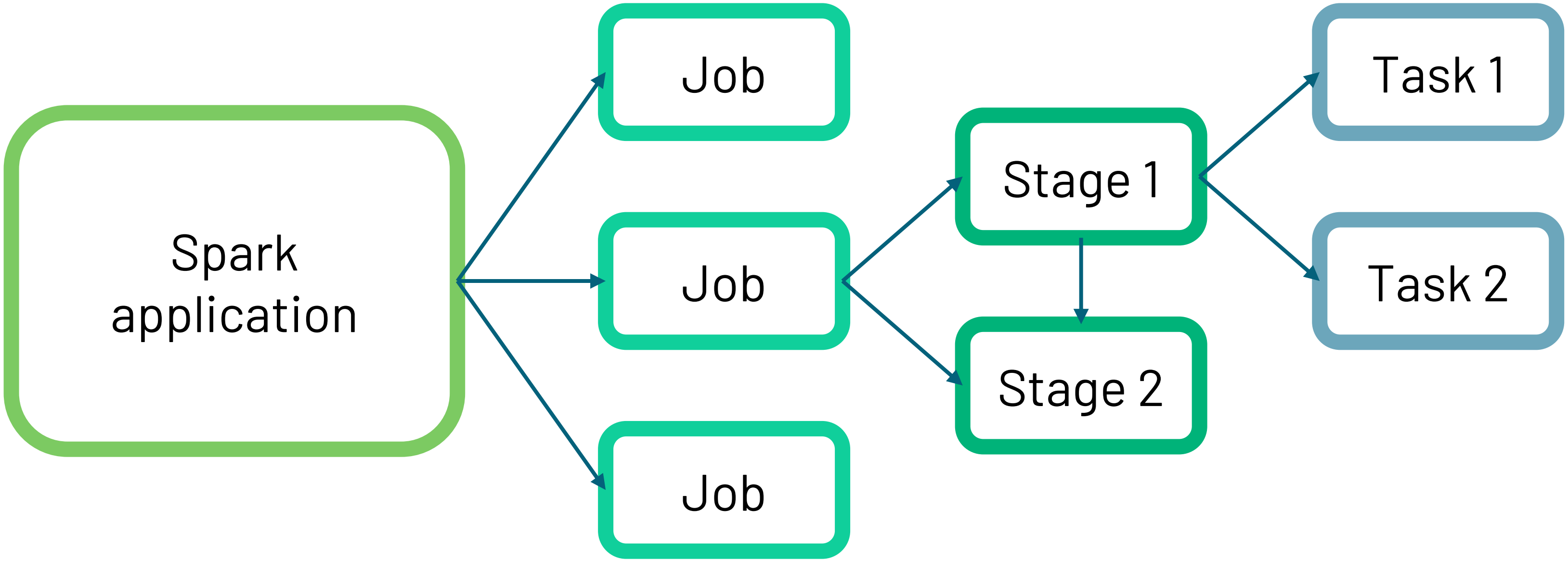
Partitions



A Spark Job

Instructions: Eliminate all the brown candies and pile the rest in the corner.

Spark Execution



Spark application

Job

Job

Job

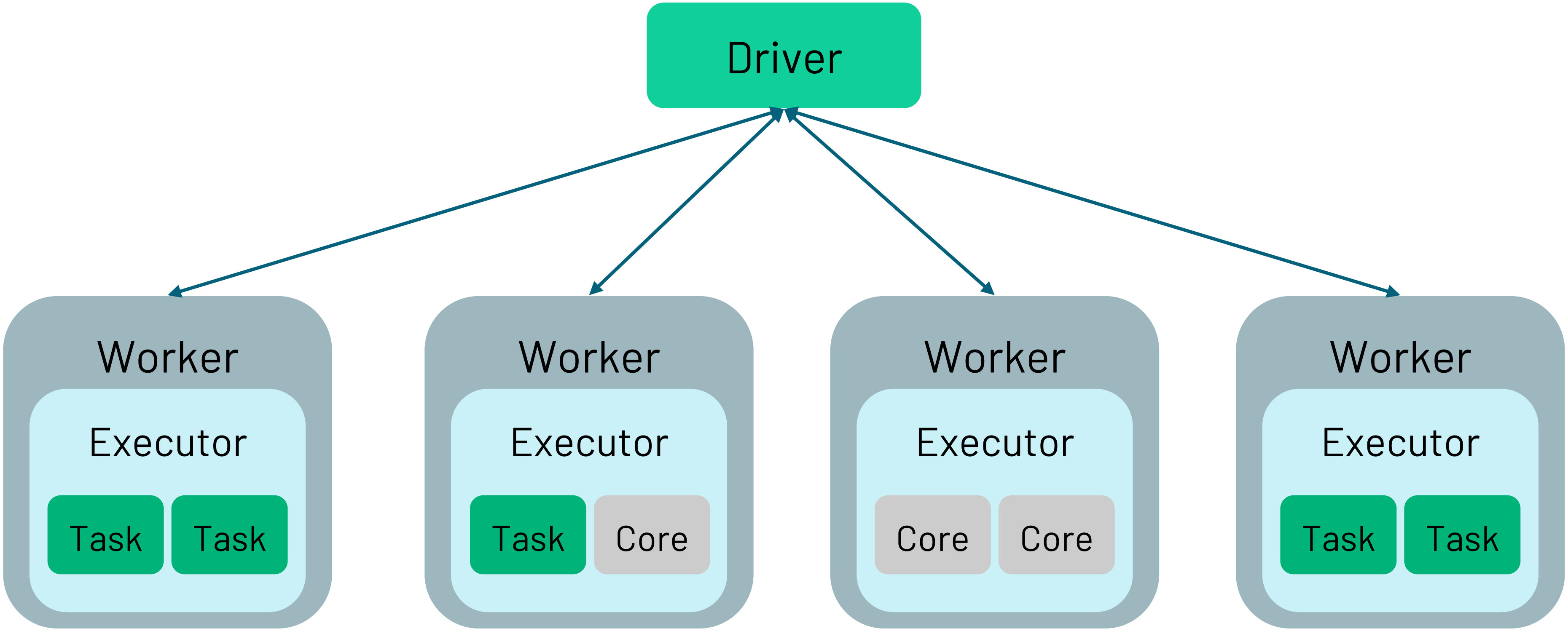
Stage 1

Stage 2

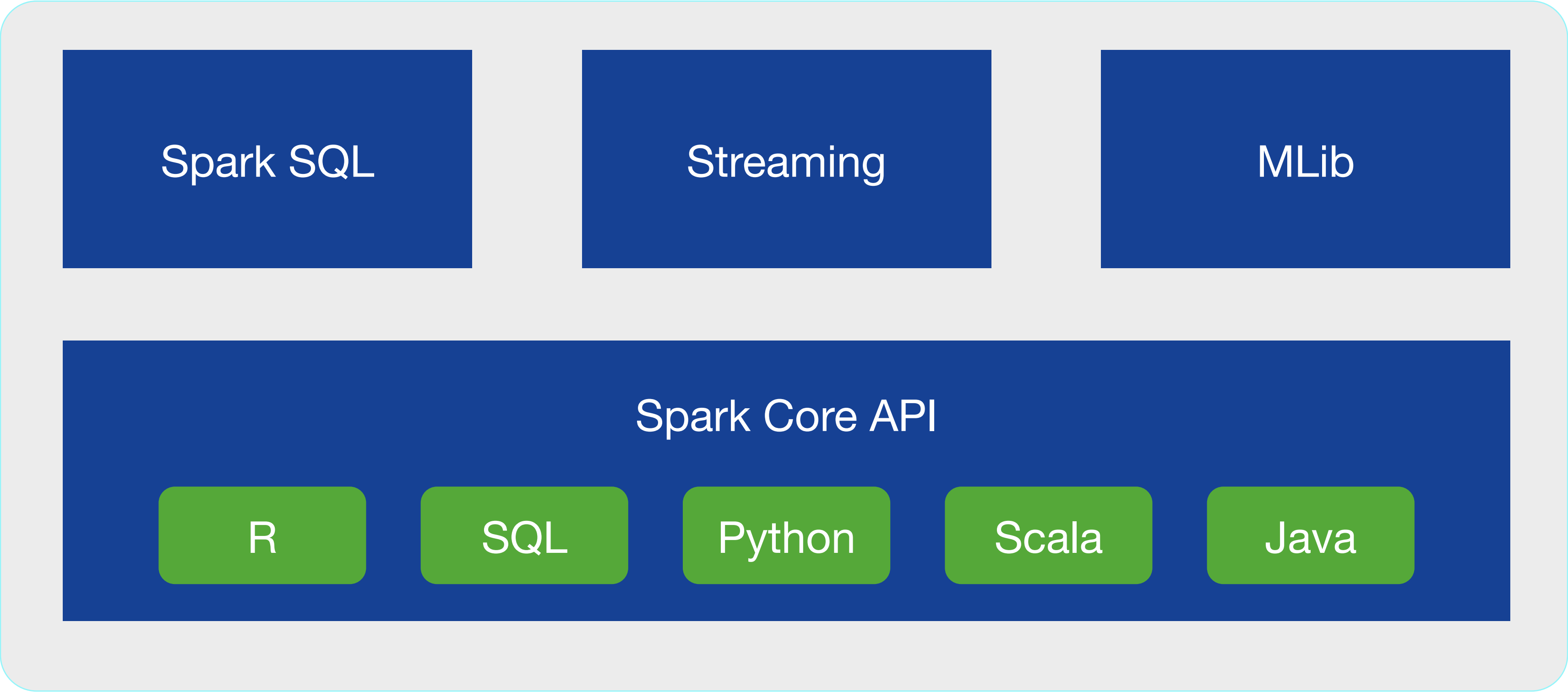
Task 1

Task 2

Spark Cluster



Spark API



Spark SQL
Streaming
MLib
Spark Core API
R
SQL
Python
Scala
Java

Recap & Keywords

- Spark
  - Standard de-facto big data processing
- Lazy evaluation
- Dataset partitions
- Orizontal scaling
- Transformations & Components

Questions?

Topics

1. Overview & Recap
2. What is a pipeline?
3. Storage layer
4. Spark foundations
5. **Lab sessions**
    1. Find new job titles
    2. Find new occupations

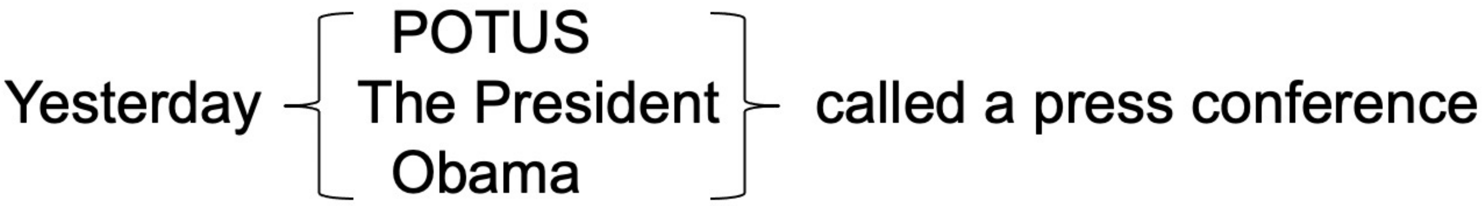
Goal

Train a Word2Vec model to improve our ontologies:

- Start from 1 occupation
- Create a corpus
- Pre-processing
- Train the model
- Use the model to extract new job titles

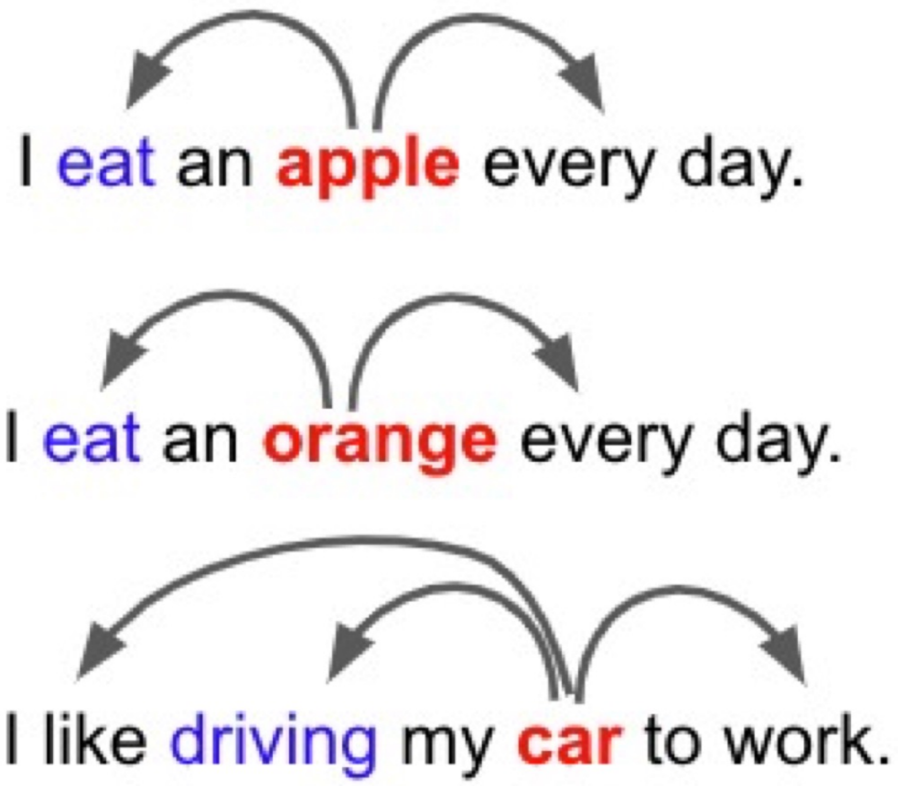
Word embeddings depend on a notion of word similarity.

A very useful definition is paradigmatic similarity:
Similar words occur in similar contexts. They are exchangable.



Yesterday POTUS / The President / Obama called a press conference

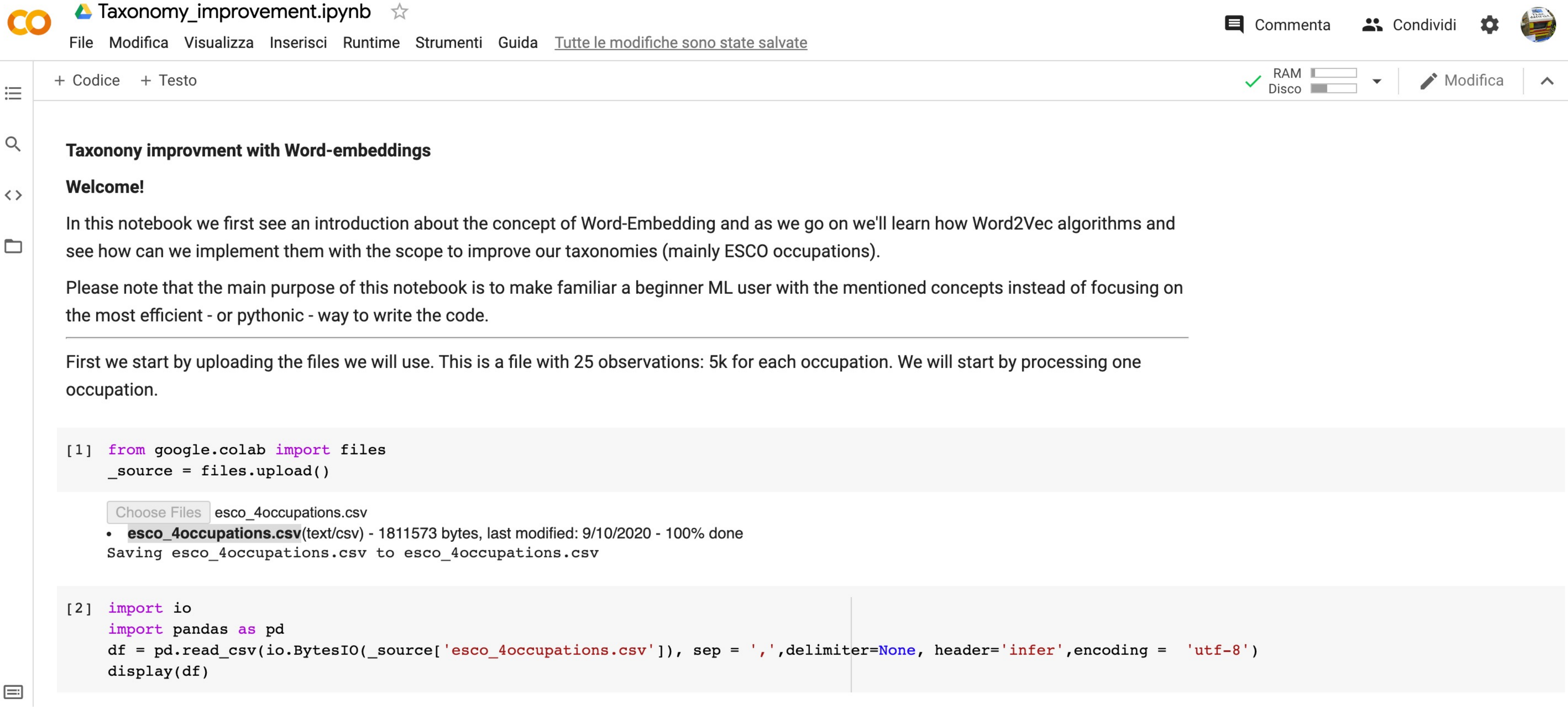
Intuition: Context also carries the meaning



I eat an apple every day.

I eat an orange every day.

I like driving my car to work.



Taxonony improvment with Word-embeddings

Welcome!

In this notebook we first see an introduction about the concept of Word-Embedding and as we go on we'll learn how Word2Vec algorithms and see how can we implement them with the scope to improve our taxonomies (mainly ESCO occupations).

Please note that the main purpose of this notebook is to make familiar a beginner ML user with the mentioned concepts instead of focusing on the most efficient - or pythonic - way to write the code.

First we start by uploading the files we will use. This is a file with 25 observations: 5k for each occupation. We will start by processing one occupation.

```
[1]  from google.colab import files
     _source = files.upload()
```

Choose Files esco_4occupations.csv

- **esco_4occupations.csv**(text/csv) - 1811573 bytes, last modified: 9/10/2020 - 100% done

```
Saving esco_4occupations.csv to esco_4occupations.csv
```

```
[2]  import io
     import pandas as pd
     df = pd.read_csv(io.BytesIO(_source['esco_4occupations.csv']), sep = ',',delimiter=None, header='infer',encoding =  'utf-8')
     display(df)
```

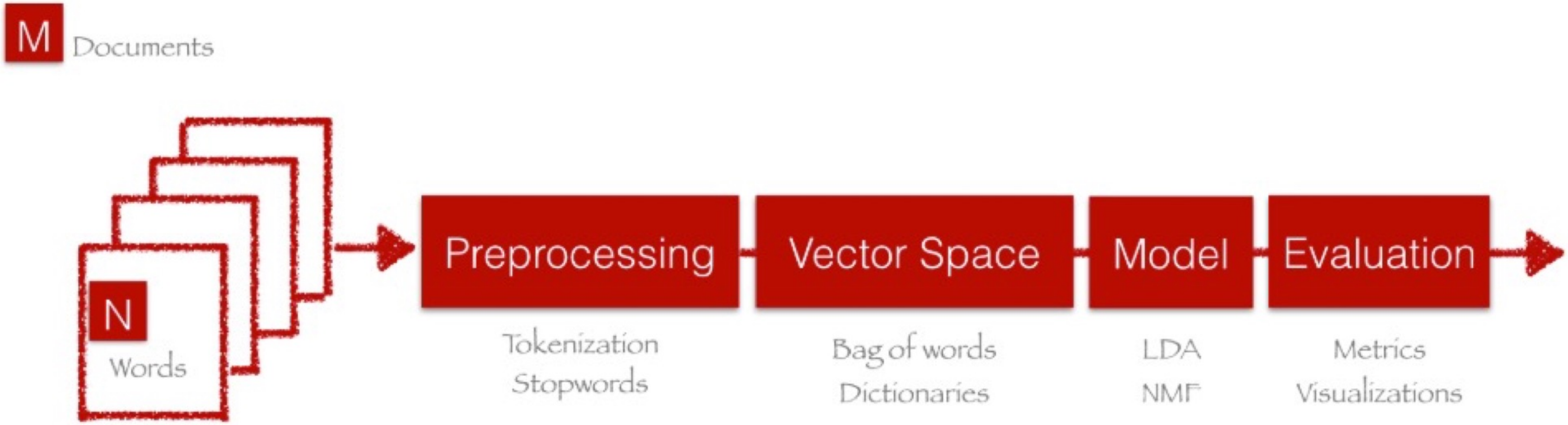
Lab session

Find new occupations

Goal

Use LDA to improve our ontologies and extract new insights:

- Start from a corpus of job vacancies
- Pre-processing
- Apply some topic modelling techniques
- Extract new occupations



M Documents

N Words

Preprocessing

Tokenization
Stopwords

Vector Space

Bag of words
Dictionaries

Model

LDA
NMF

Evaluation

Metrics
Visualizations

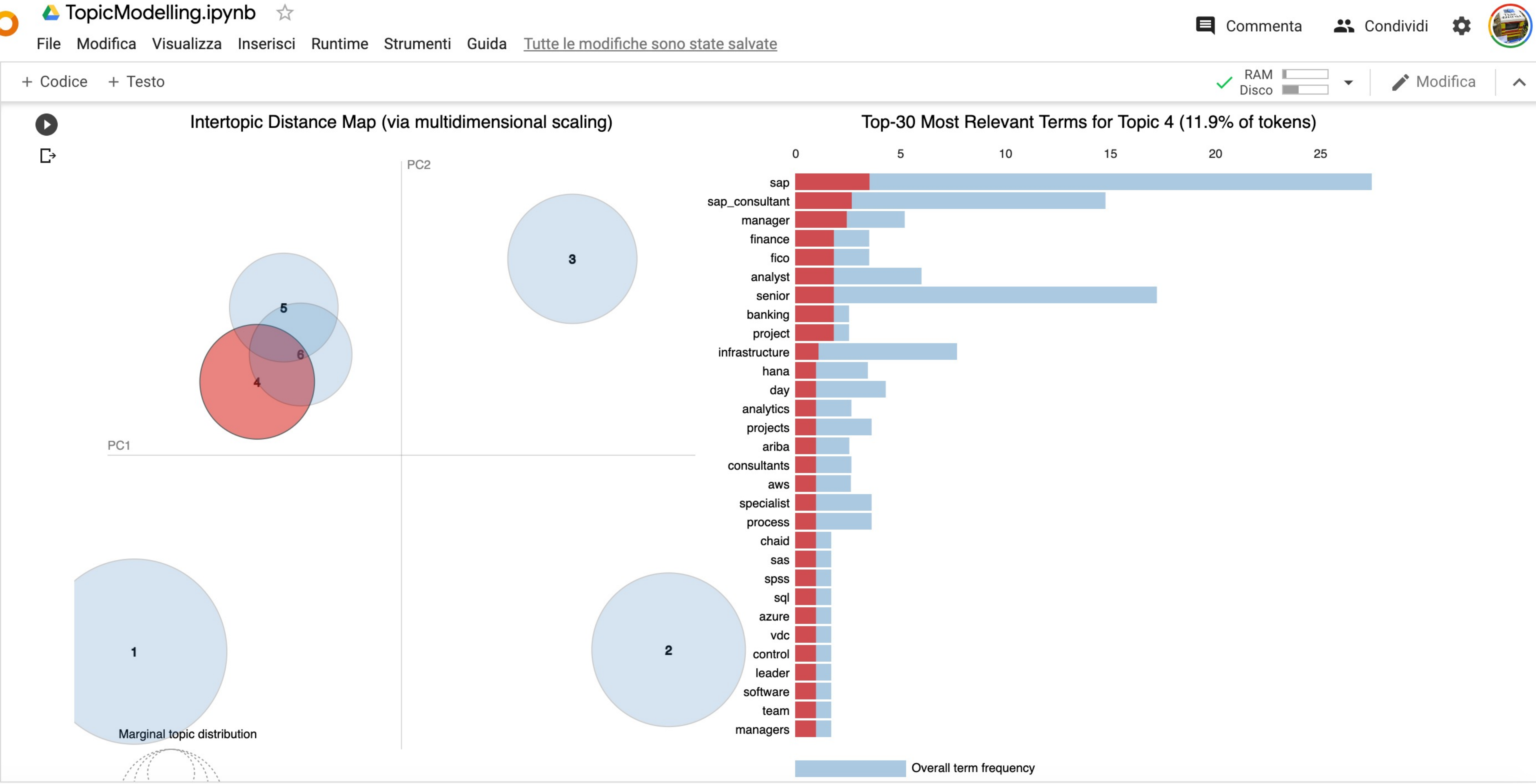
What «topic» means?

Observation

A group of words are likely to appear in the same **context**

A hidden (so, unknown) structure that helps determine what words are likely to appear in a corpus

A topic is a word-distribution over a fixed vocabulary


TopicModelling.ipynb
File
Modifica
Visualizza
Inserisci
Runtime
Strumenti
Guida
Tutte le modifiche sono state salvate
Commenta
Condividi
+ Codice
+ Testo
RAM
Disco
Modifica
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
6
Marginal topic distribution
Top-30 Most Relevant Terms for Topic 4 (11.9% of tokens)
0
5
10
15
20
25
sap
sap_consultant
manager
finance
fico
analyst
senior
banking
project
infrastructure
hana
day
analytics
projects
ariba
consultants
aws
specialist
process
chaid
sas
spss
sql
azure
vdc
control
leader
software
team
managers
Overall term frequency