THE DATA SCIENCE LAB
- Introduction -

COM 490 – Module 1a

Week 1

Week 1 - Agenda

- Introduction to the course
  - Overview of big data concepts
  - Objectives
  - Organization: course structure, labs project and evaluation
- Lab environment set up

Meet the team

Sofiane Sarni
SDSC
Module 4

Pamela Delgado
SDSC
Module 3

Eric Bouillet
SDSC
Module 1
Module 2

Hantao Zhang
Doctoral Assistant

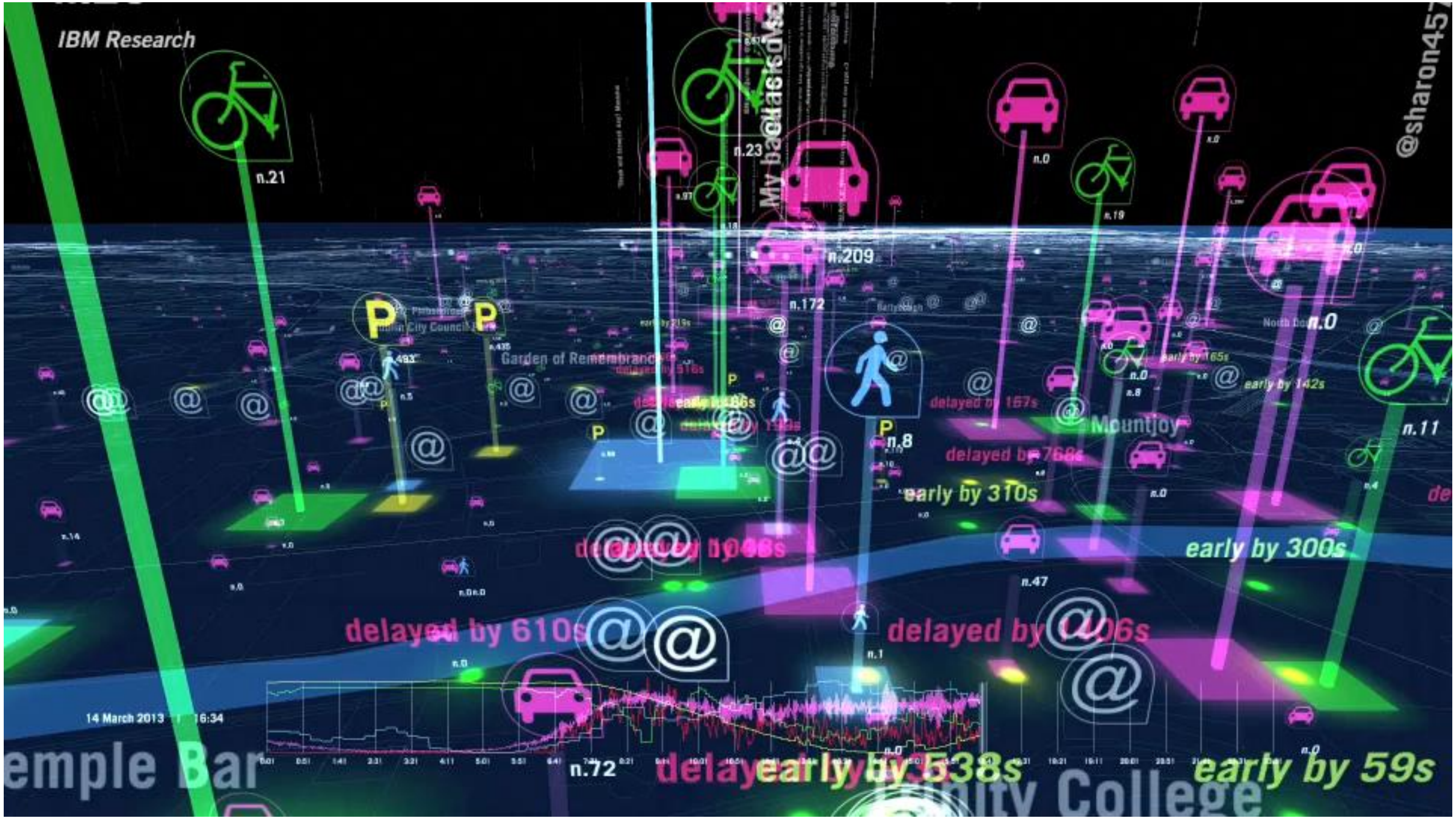
Hao Zhao
Doctoral Assistant

Junyu Liu
Doctoral Assistant

A Journey In Data Science

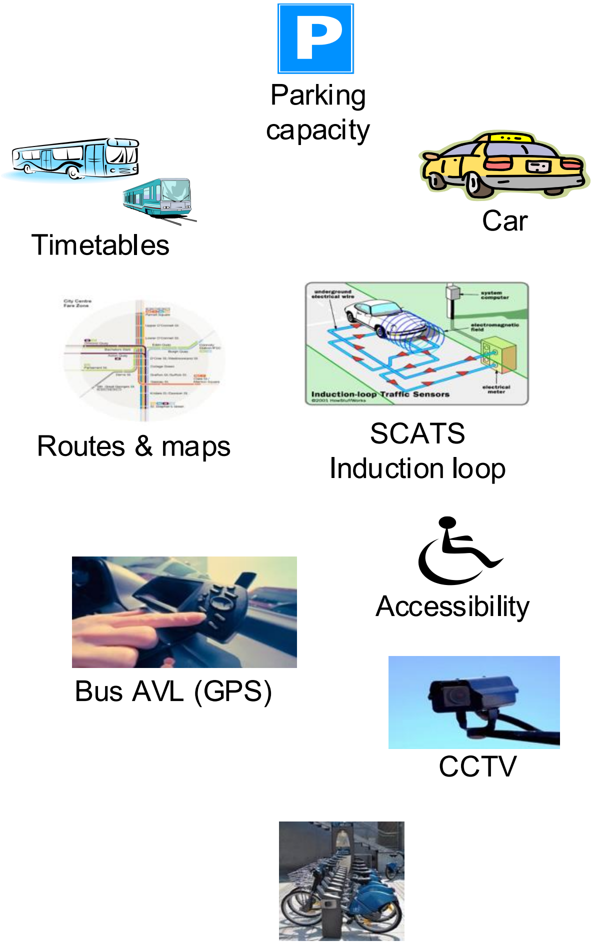
O'Connel Bridge / D'ollier St. Dublin City CCTV
8 Apr 2013 18:31:50 GMT Daylight Time

A Journey In Data Science



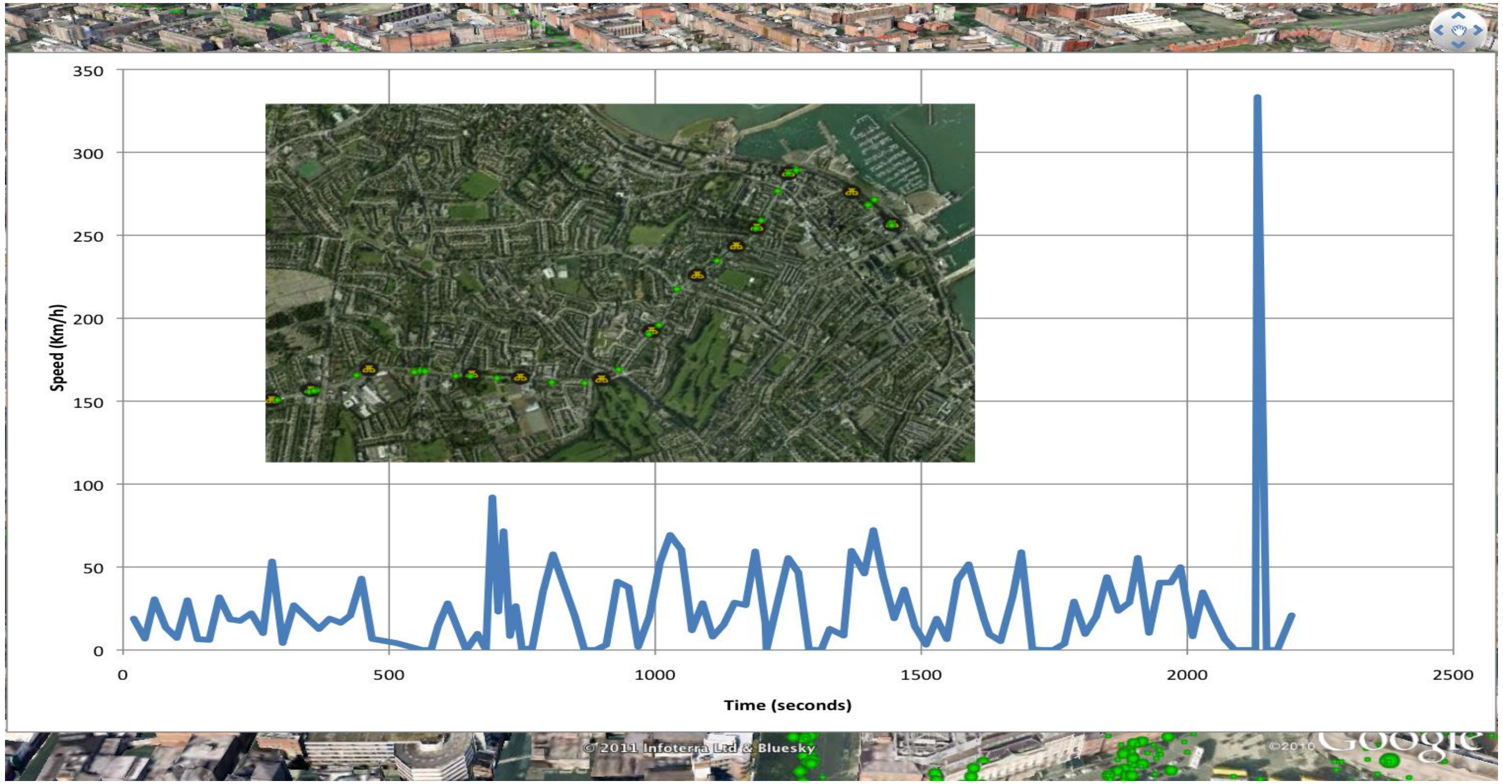
From row data to information

- Complex system & analytics challenges
  - Data diversity, heterogeneity



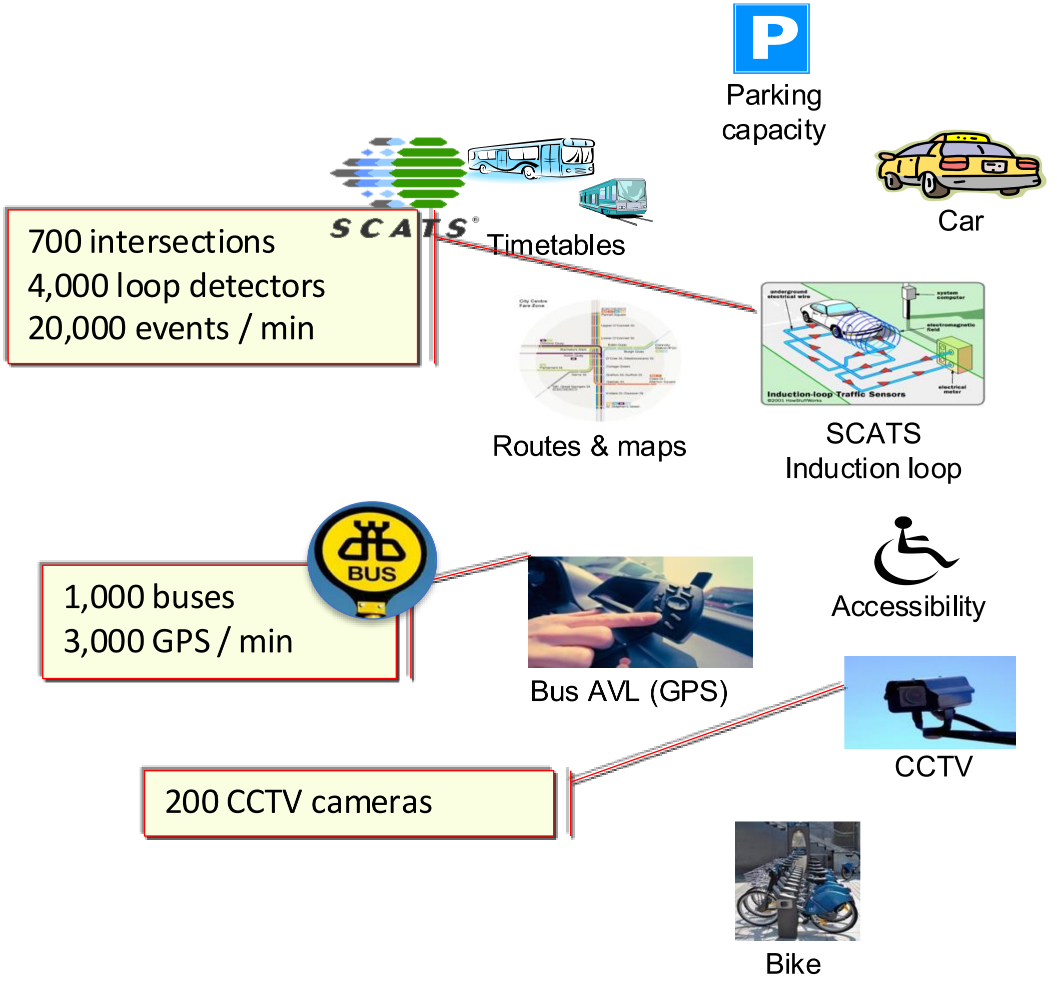
From row data to information

- Complex system & analytics challenges
  - Data diversity, heterogeneity
  - Data accuracy, sparsity



From row data to information

- Complex system & analytics challenges
    - Data diversity, heterogeneity
    - Data accuracy, sparsity
    - Data volume

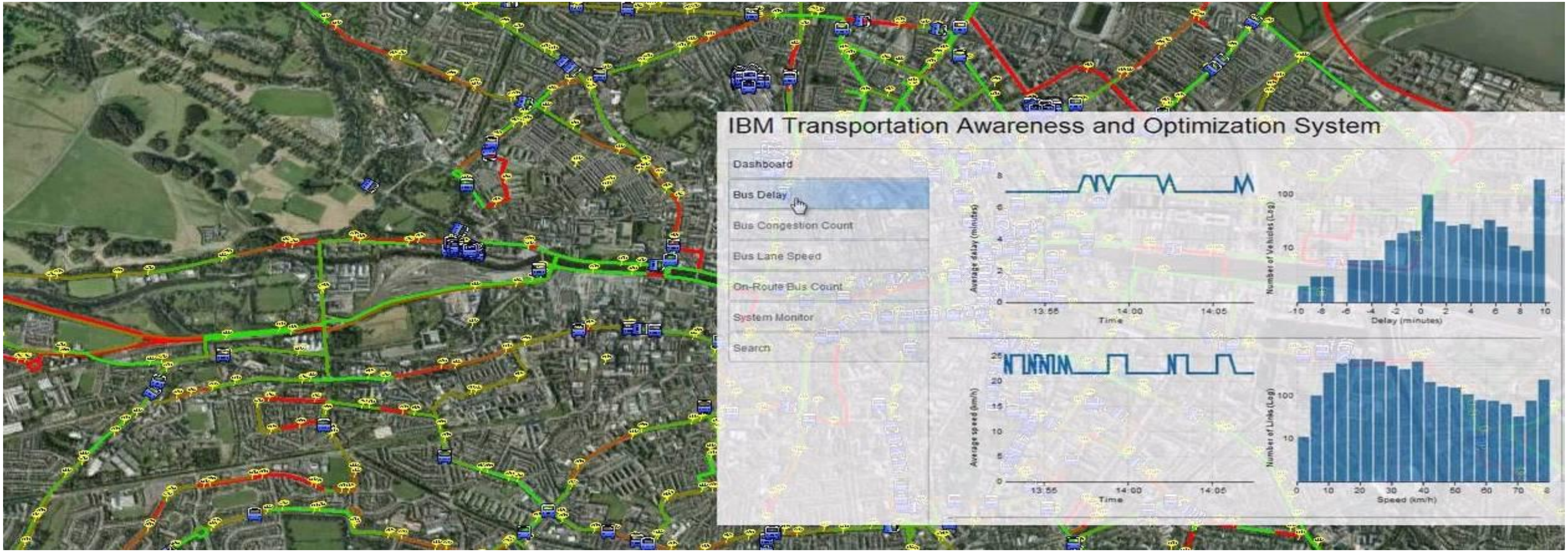


700 intersections
4,000 loop detectors
20,000 events / min

1,000 buses
3,000 GPS / min

200 CCTV cameras

From row data to information



"System and analytics for continuously assessing transport systems from sparse and noisy observations: Case study in Dublin"
L. Gasparini, E. Bouillet, F. Calabrese, O. Verscheure, IEEE Conference on Intelligent Transport Systems, 2011.

Data is the new oil (circa 2017)

The Economist

MAY 6TH–12TH 2017

Crunch time in France

Ten years on: banking after the crisis

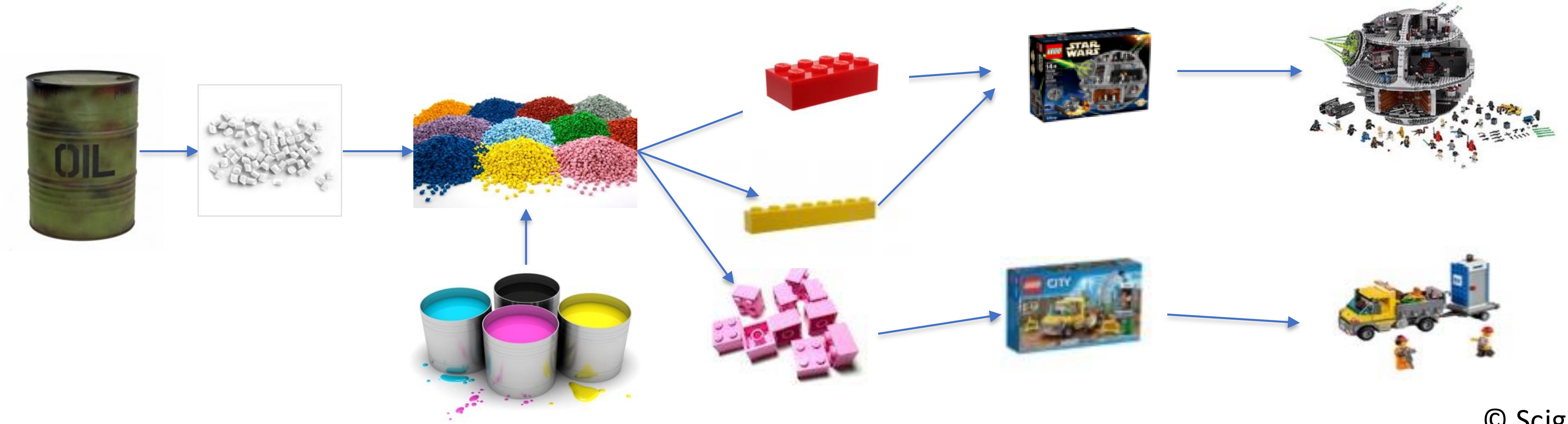
South Korea's unfinished revolution

Biology, but without the cells

The world's most valuable resource

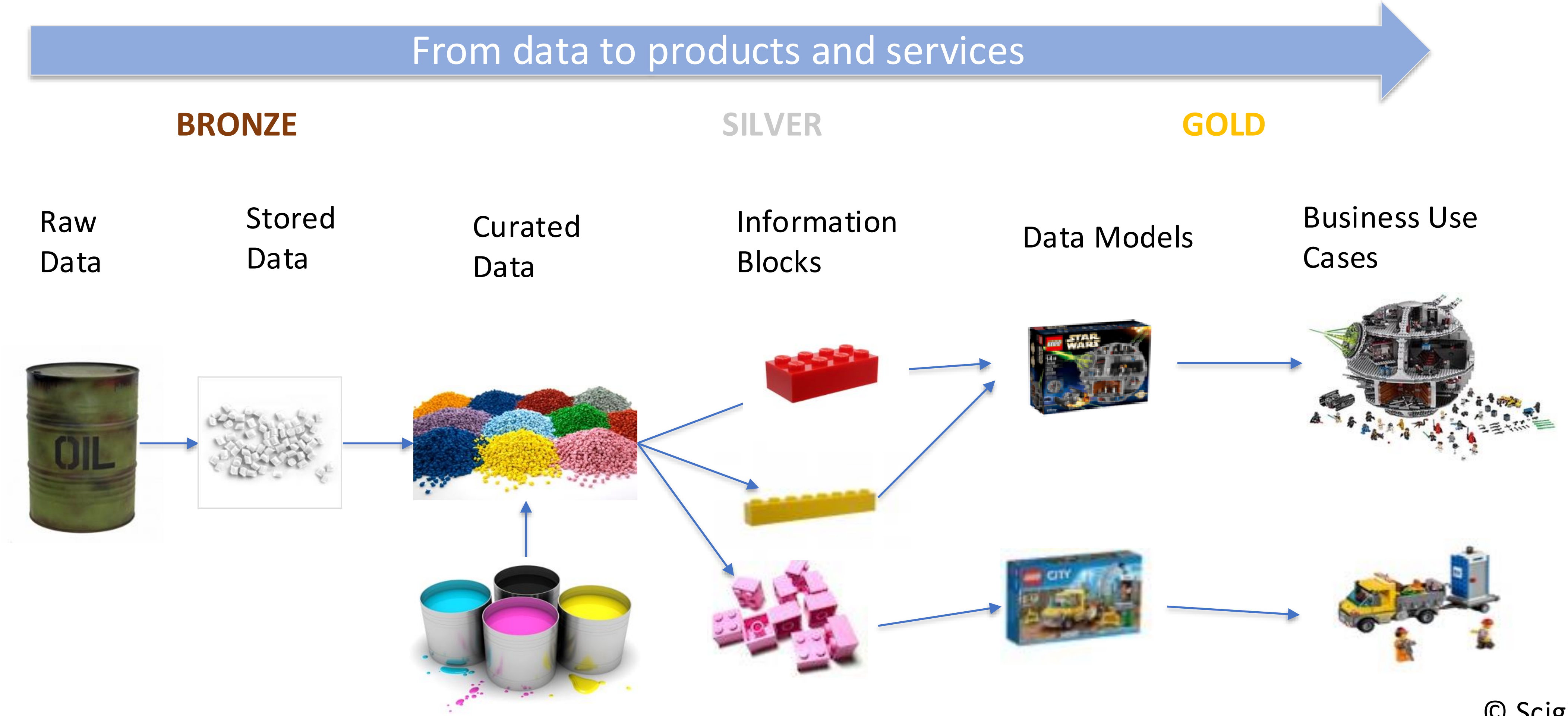
Data and the new rules of competition

Data vs. Traditional Assets



© Scigility AG

Data vs. Traditional Assets



From data to products and services

BRONZE

SILVER

GOLD

Raw Data

Stored Data

Curated Data

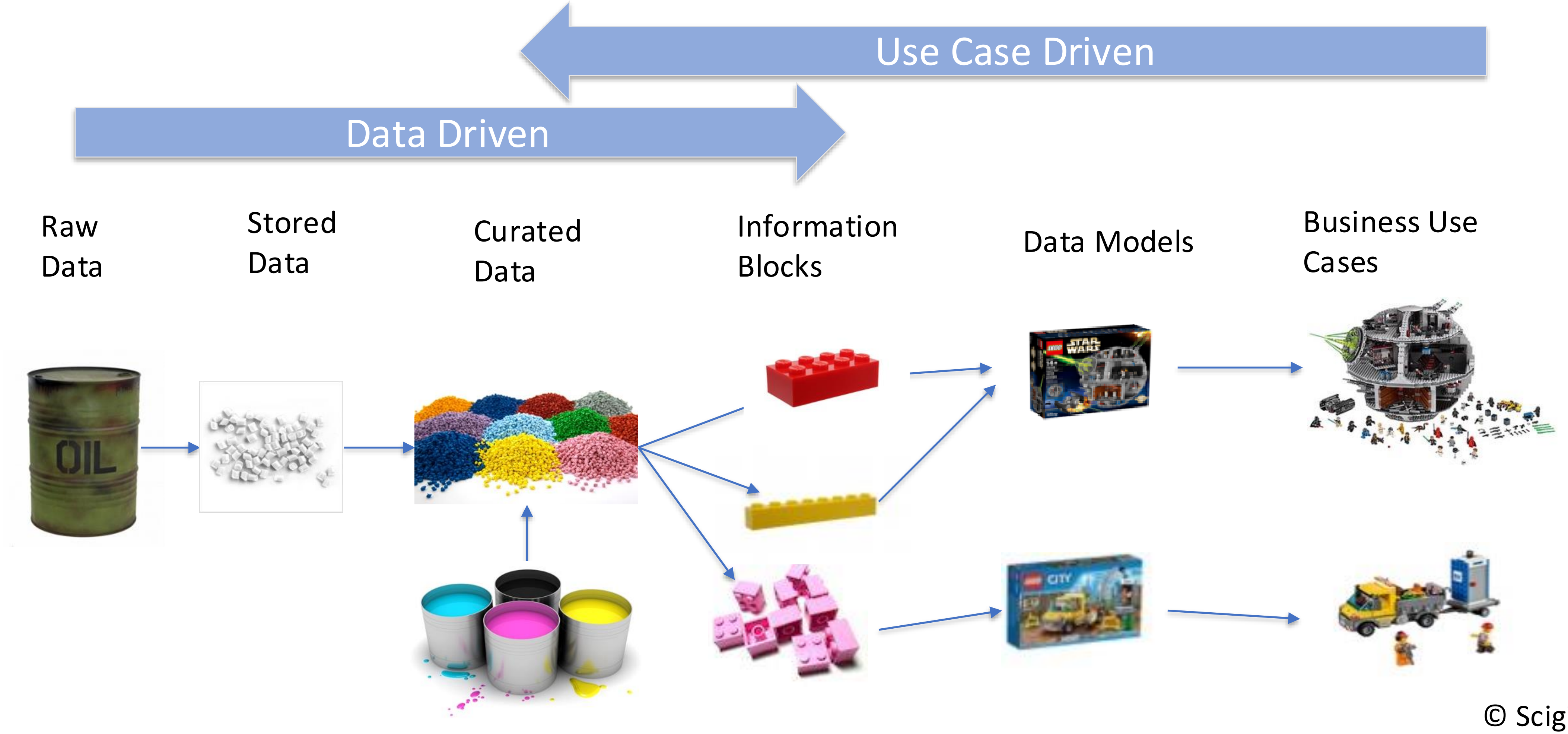
Information Blocks

Data Models

Business Use Cases

© Scigility AG

Use Case or Data Driven

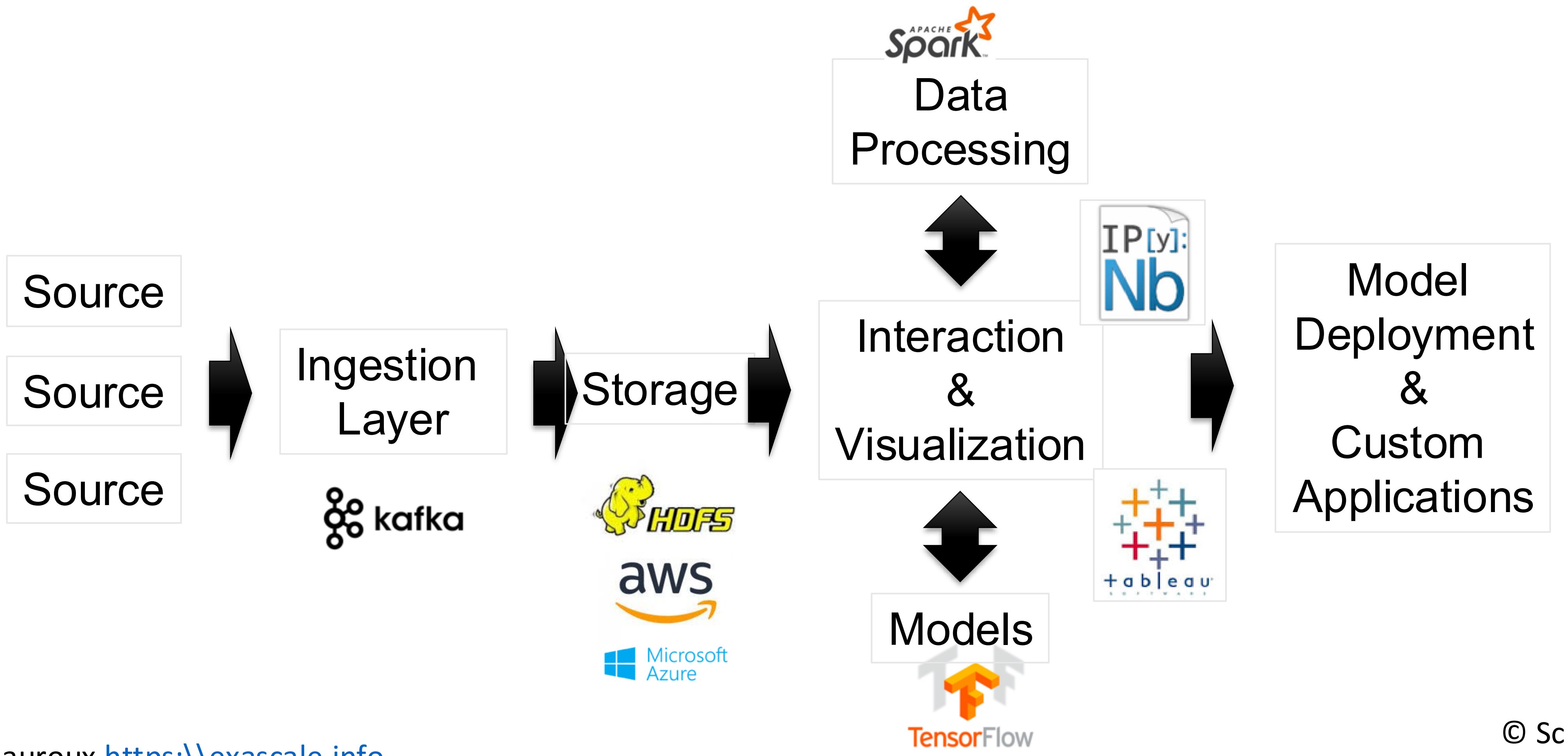


© Scigility AG

Enterprise Data Platform

- The "plumbing and storage" behind how companies collect, store, manage and use data for decisions and innovation
- 4 Core functions:
  - **Data ingestion**: getting the data in from ERP (DB), sensors, web apps, etc.
  - **Data storage**: where <u>raw</u> and <u>refined</u> data lives
  - **Data processing & querying**: making sense of the data
  - **Data serving & insight**: putting it to use in dashboards, reports, machine learning

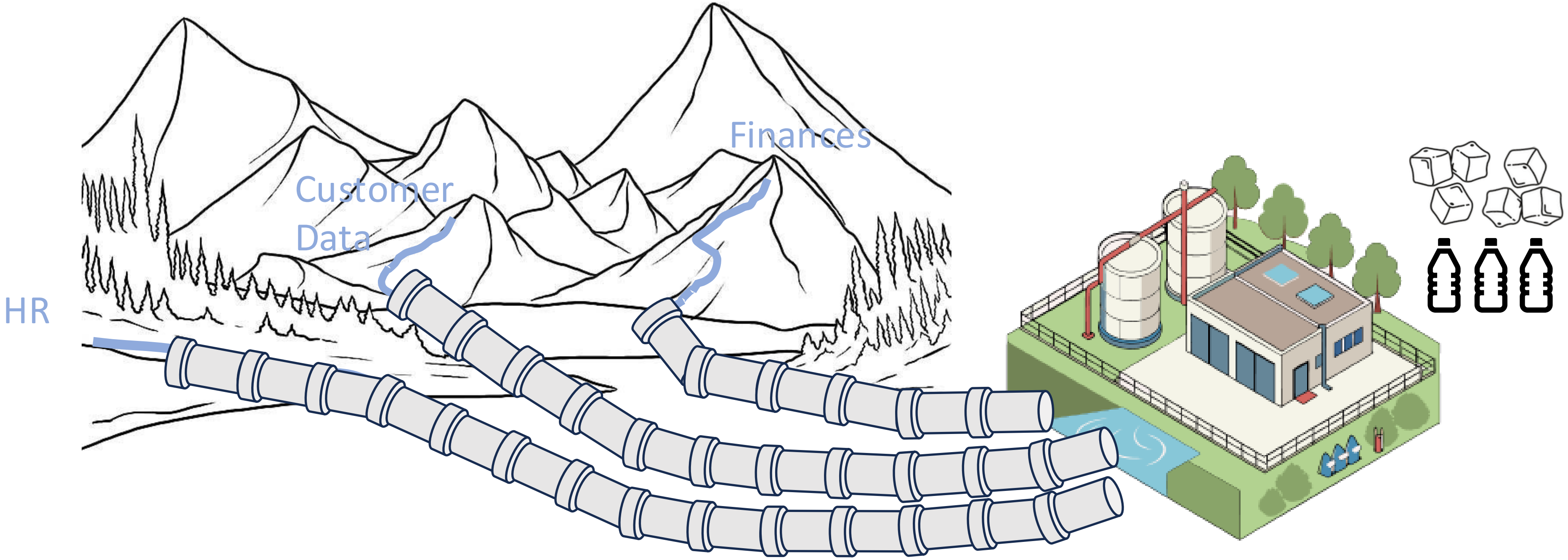
Typical Architecture



© P. Cudre-Mauroux https:\\exascale.info

© Scigility AG

Data warehouse – From sources to refined data storage

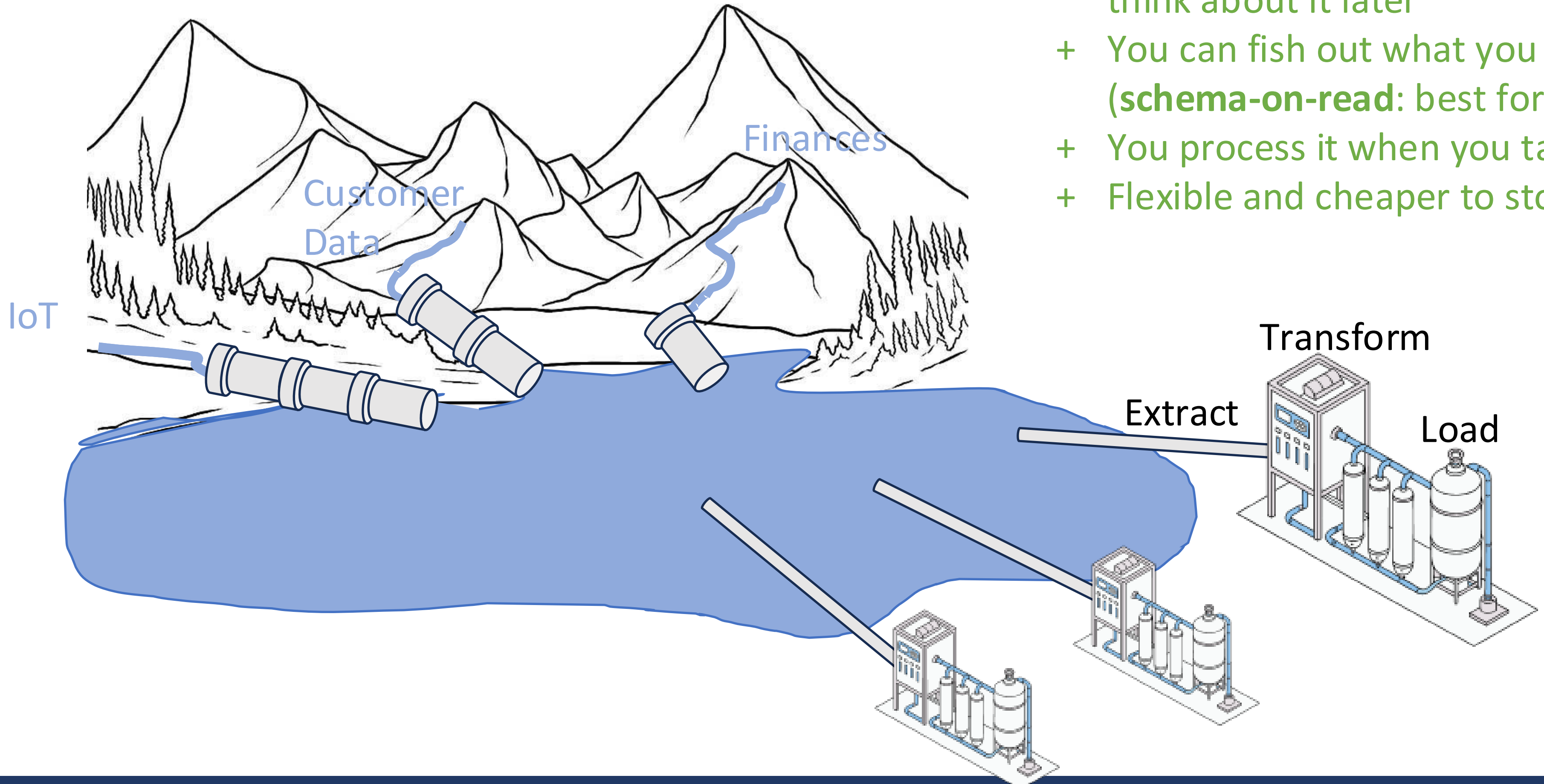


+ **Clean**, **structured** data optimized for reports or export to specific tools, highly scalable

- **Rigid: schema on write (best for <u>use case driven</u>),** high cost, slow update.

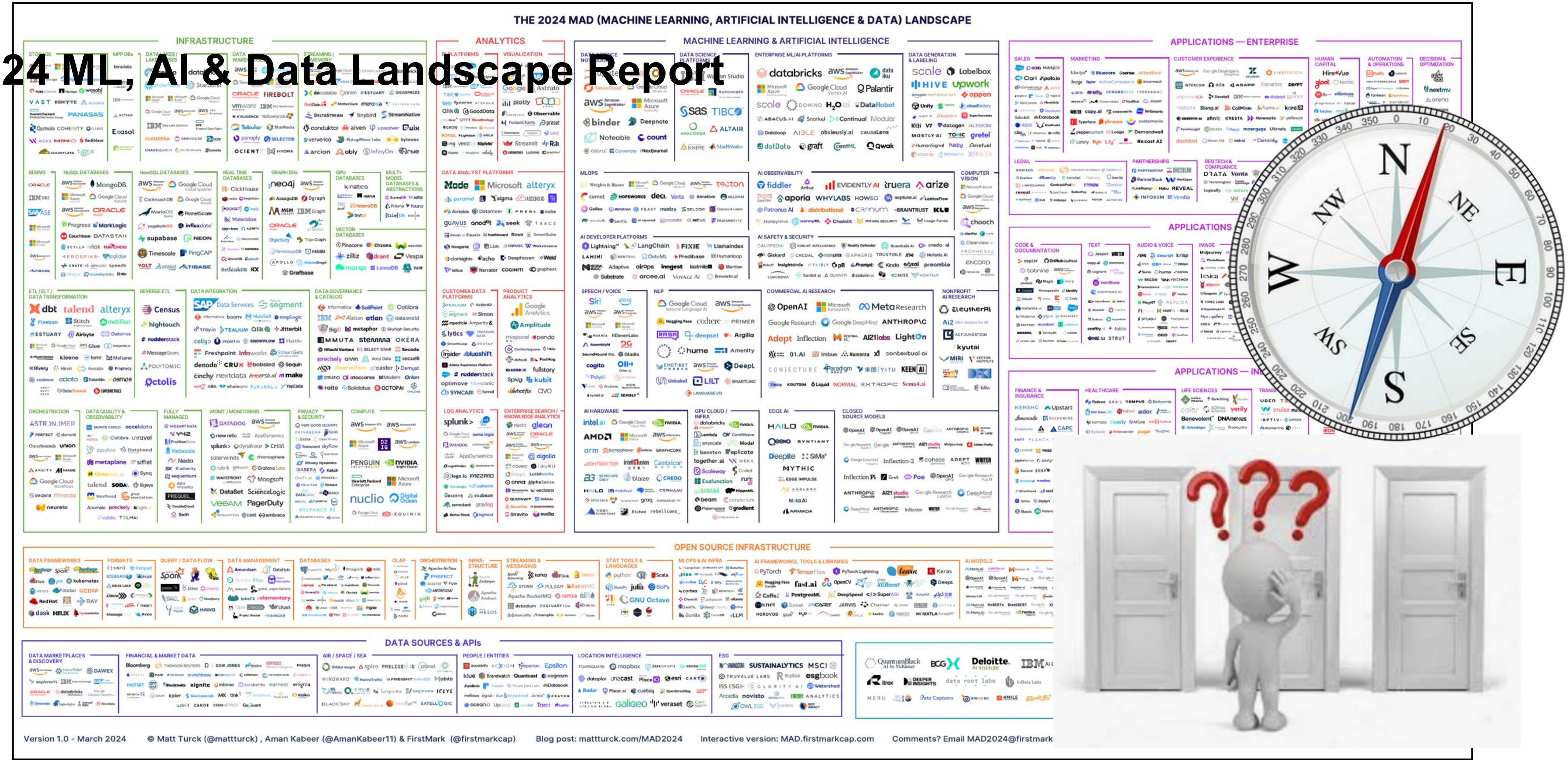
Data Lake – Large open reservoir of raw data

- + You can dump any data in (raw logs, images, etc.) and think about it later
- + You can fish out what you need, when you need it (**schema-on-read**: best for **data-driven** approach)
- + You process it when you take it out (curate, transform)
- + Flexible and cheaper to store



Focus of this course

What this lab is about?



2024 ML, AI & Data Landscape Report

THE 2024 MAD (MACHINE LEARNING, ARTIFICIAL INTELLIGENCE & DATA) LANDSCAPE
INFRASTRUCTURE
ANALYTICS
MACHINE LEARNING & ARTIFICIAL INTELLIGENCE
APPLICATIONS — ENTERPRISE
APPLICATIONS
APPLICATIONS — IN
OPEN SOURCE INFRASTRUCTURE
DATA SOURCES & APIs
Version 1.0 - March 2024
© Matt Turck (@mattturck) , Aman Kabeer (@AmanKabeer11) & FirstMark (@firstmarkcap)
Blog post: mattturck.com/MAD2024
Interactive version: MAD.firstmarkcap.com
Comments? Email MAD2024@firstmark
N
NE
E
SE
S
SW
W
NW
???

https://mattturck.com/landscape/mad2024.pdf

Lab Overview

- Big Data Foundations:
  - Introduction to data engineering workflows
  - Building and managing data lakes on distributed storage
  - Query engines with Trino and Spark
  - Real time event processing with Kafka
- A journey through a real-world data science project
- Hand-on and pragmatic

- **4 Modules**
  - Module 1 – Data Science with Python
  - Module 2 – Building a data lakes, and data wrangling with Trino
  - Module 3 – Big data processing & Machine Learning with Apache Spark
  - Module 4 – Real time data processing (critical applications)

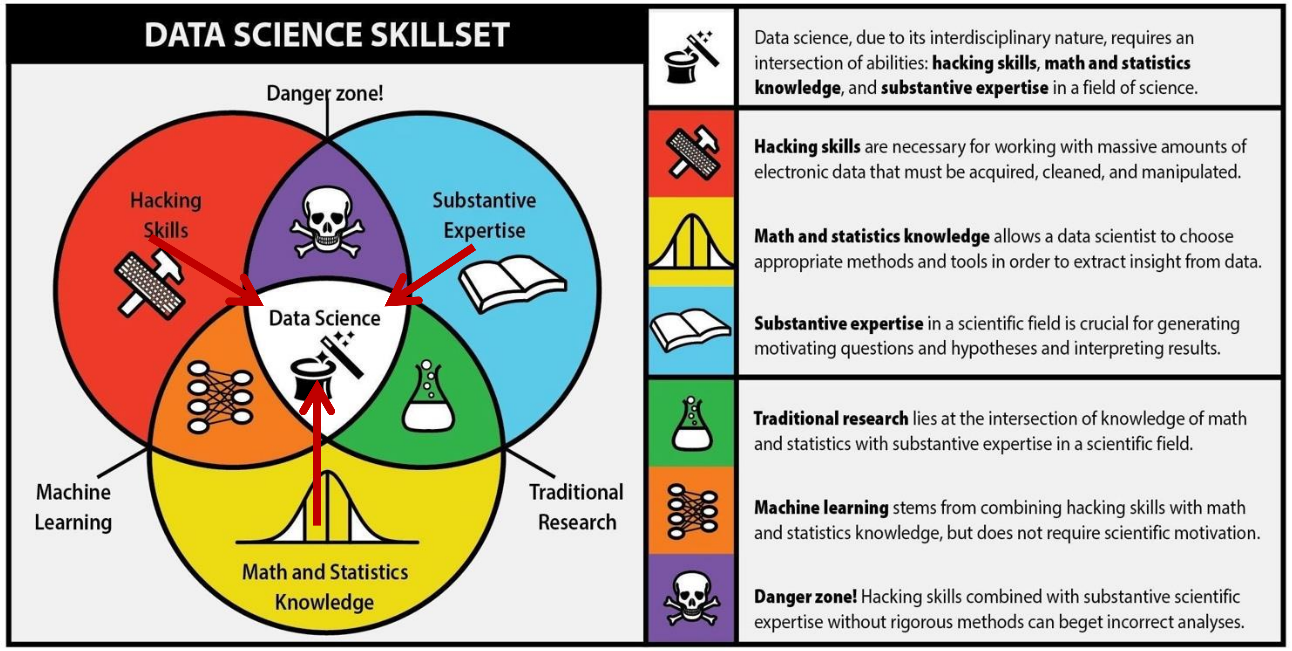
Agenda 2025 - Module 1a

- 1a Introduction to Data Science with Python
- 1b (Bigger) Data Science with Python
- 2a Introduction to Big Data Technologies
- 2b Big Data Wrangling with Hadoop
- 2c Advanced Big Data Queries
- 3a Introduction to Spark
- 3b Spark Data Frames
- 3c Advanced Spark
- 4a Introduction to Stream Processing
- 4b Stream Processing with Kafka
- 4c Stream Processing with Kafka and Spark
- Proj Final Project, Q&A
- Proj Final Project Due (short video and code)
- Proj Oral Sessions

Lab Overview

- 50% (Big) Data/Feature Engineering
- 30% (Big) Data Science
- 20% Build foundations for ML-Ops

Drew Conway’s Venn Diagram



Data science, due to its interdisciplinary nature, requires an intersection of abilities: **hacking skills**, **math and statistics knowledge**, and **substantive expertise** in a field of science.

Hacking skills are necessary for working with massive amounts of electronic data that must be acquired, cleaned, and manipulated.

Math and statistics knowledge allows a data scientist to choose appropriate methods and tools in order to extract insight from data.

Substantive expertise in a scientific field is crucial for generating motivating questions and hypotheses and interpreting results.

Traditional research lies at the intersection of knowledge of math and statistics with substantive expertise in a scientific field.

Machine learning stems from combining hacking skills with math and statistics knowledge, but does not require scientific motivation.

Danger zone! Hacking skills combined with substantive scientific expertise without rigorous methods can beget incorrect analyses.

Class Format

- **Labs on Wednesday – 13h10 to 16h00**
  - Theory and general introduction to exercises
  - Exercise sessions of 30min to 40min each, and 10min recap between sessions
  - Classes are recorded (Zoom*), and videos are made available after the class

- **Office hours**
  - Interactive communication via Ed forum(*)
  - Outside class hours on demand - time to be adapted according to students' schedule

*Details on Moodle

Communication

- **Moodle**
  - https://moodle.epfl.ch/course/view.php?id=15635
  - Class materials (slides), form groups, oral schedule, and other useful links

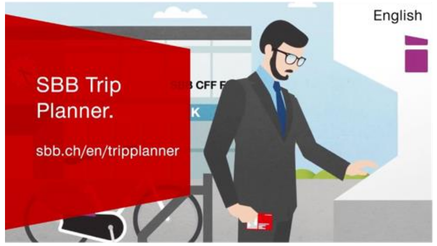
- **Ed (*)**
  - For real-time intra/inter group communication, and to reach us outside class hours
  - Channels:

General	For our general announcements or to forward EPFL guidelines
Labs	Discussions related to the lectures and labs
Assignments	Channel for each assignment (A1, ...), and one for the final
Social	Looking for a team, or a team-mate ?

  - Etiquette:
    - **DO** Respond to comments, answer questions in the same thread (do not start a new thread)
    - **DO** Help each other with technical issues etc.,
    - **DO NOT** provide solutions to assignment

*Details on Moodle

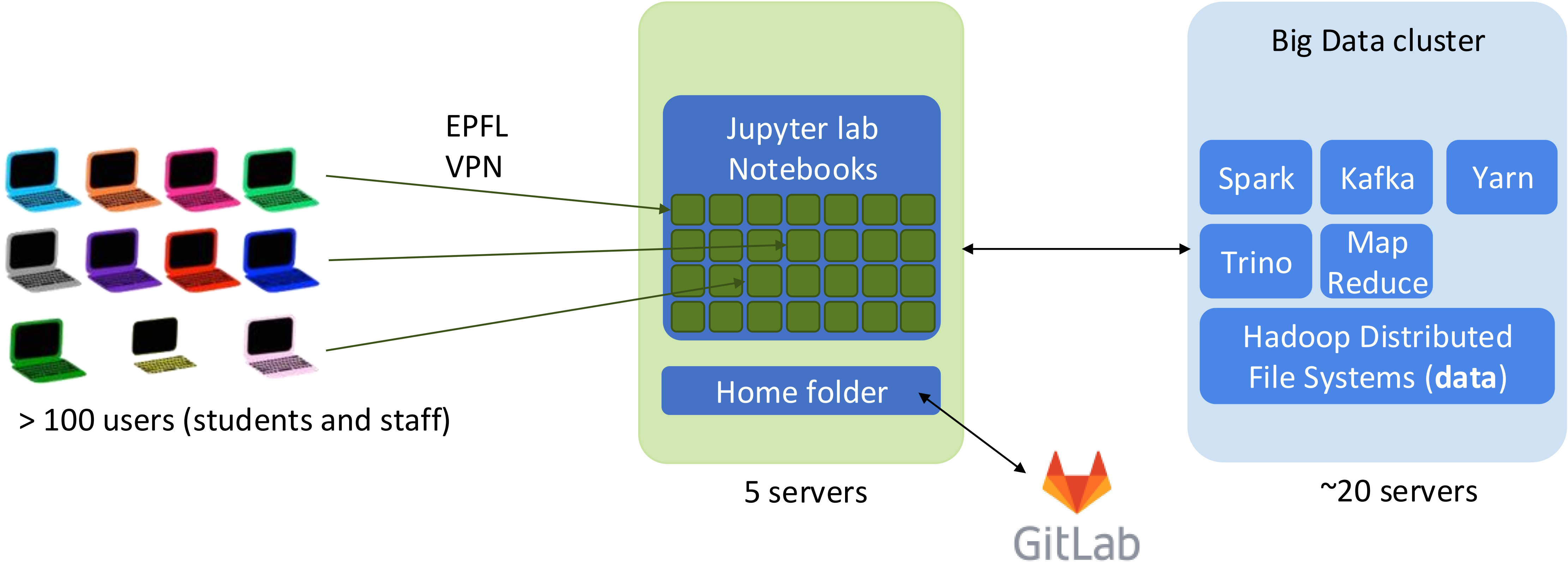
Lab Assessment



- 40% Final project
  - Collaborative project, in teams of 4, max 5
  - Due before final week of semester
    - 6-7min video presentation and code
  - 15min group discussion (oral) during the final week
  - Top 3 will be invited to present to tl - Transport public lausannois

- 60% Continuous assessment
  - One take-home assignment per module 2 to 4
  - To complete in groups, within 3 weeks each
  - Assignments are related to the final project

Programming Environment



1 **BYOL**: Students work remotely using their laptops. Nothing to install – only web browser is needed.

2 Students work in teams, write and share code and environment in jupyter notebooks and gitlab

3 All data stored, and compute intensive processing executed on the distributed Big Data cluster.

Programming Environment

- **Programming Languages**
  - Mainly Python
    - Numpy, Pandas, Scikit-Learn, Matplotlib, PySpark, ...
  - Also: SQL-like, Linux Shell command lines

- **Development Environment**
  - Jupyter notebooks
  - Git (gitlab)
  - Hadoop big data cluster (HDFS, Spark, Kafka, Trino)

Today's check list – key objectives

- **You have access to EPFL network (VPN)**
  - Otherwise: → https://vpn.epfl.ch
- **You have registered for the class on IS-Academia**
  - Otherwise: → http://is-academia.epfl.ch
- **You have access to our Moodle page and have bookmarked it**
  - https://moodle.epfl.ch/course/view.php?id=15635
  - Contact us to add you to the list
- **You have access to our programming environment (JupyterHub)**
  - You can login to your assigned jupyter notebook with your usual EPFL (gaspar) username and password
- **You have access to the exercises of module 1a**
  - You can login and access https://dslabgit.datascience.ch/course/2026/module-1a
- **You master the ABCs of building and validating a predictive model with Scikit-learn**

Start your engines

Bootstrapping into Jupyter notebooks

Jupyter Hub – Login

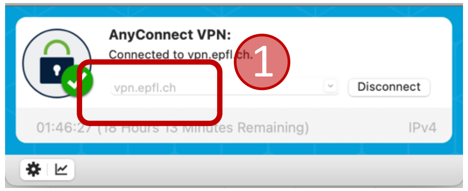
1. Must be on EPFL network (use VPN if required)
2. Sign in https://groups.epfl.ch/ and in “My groups” search for **com-490** to find your assigned Jupyter hub server URL

   You should see **ic-spark-com-490-**... If not, come to us

3. Based on the above, in a browser (Firefox, Safari, Chrome), sign in with your EPFL (gaspar) username and password at the URL assigned to your group

Group	URL
ic-spark-com-490-1:	iccluster094.iccluster.epfl.ch
ic-spark-com-490-2:	iccluster095.iccluster.epfl.ch
ic-spark-com-490-3:	iccluster096.iccluster.epfl.ch
ic-spark-com-490-4:	iccluster097.iccluster.epfl.ch
ic-spark-com-490-5:	iccluster098.iccluster.epfl.ch

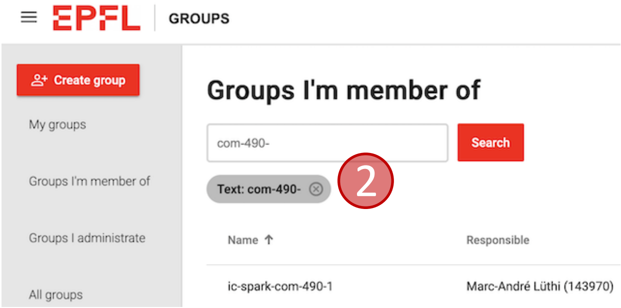
4. Click: **Start My Server**.



Connected to vpn.epfl.ch.

vpn.epfl.ch

01:46:27 (18 Hours 13 Minutes Remaining)

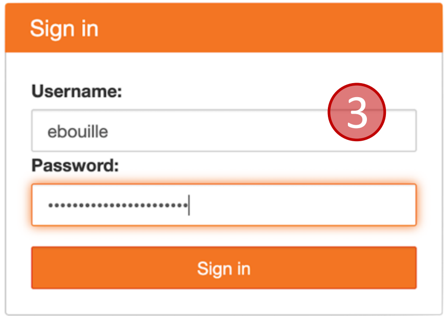


Groups I'm member of

com-490-

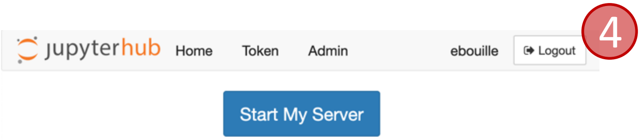
Text: com-490-

Name ↑	Responsible
ic-spark-com-490-1	Marc-André Lüthi (143970)



Username: ebouille

Password: ••••••••••••••••••••••

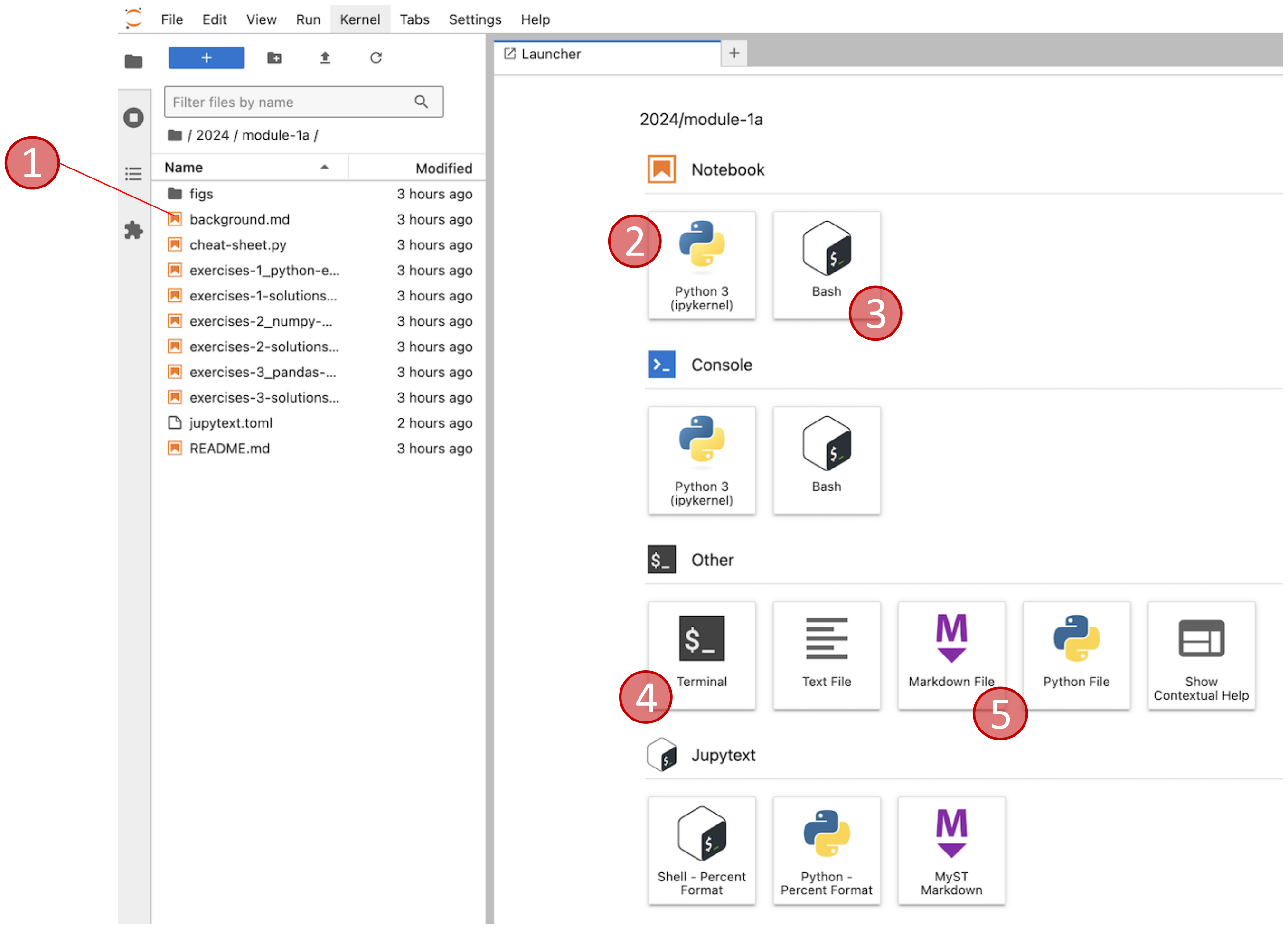


Jupyter Lab – Interactive sessions

1. Folders and files of weekly lab

   E.g. module-1a, module-1b, ...

2. New python notebooks
3. New shell script (bash) notebooks
4. New terminal (bash/linux)
5. Markdown .md files (README, doc)



Name	Modified
figs	3 hours ago
background.md	3 hours ago
cheat-sheet.py	3 hours ago
exercises-1_python-e...	3 hours ago
exercises-1-solutions...	3 hours ago
exercises-2_numpy-...	3 hours ago
exercises-2-solutions...	3 hours ago
exercises-3_pandas-...	3 hours ago
exercises-3-solutions...	3 hours ago
jupytext.toml	2 hours ago
README.md	3 hours ago

Jupyter Lab – Exercises module 1a

1. Start a new JupyterHub terminal session (4. "New Terminal" in previous page)
2. Open a terminal and in the terminal, type:

```
git clone git@dslabgit.datascience.ch:course/2026/module-1a.git
```

3. Press enter
4. You should have a new folder

```
./module-1a
```

5. If git clone does not work for you, download the file module-1a.zip from moodle in the same terminal

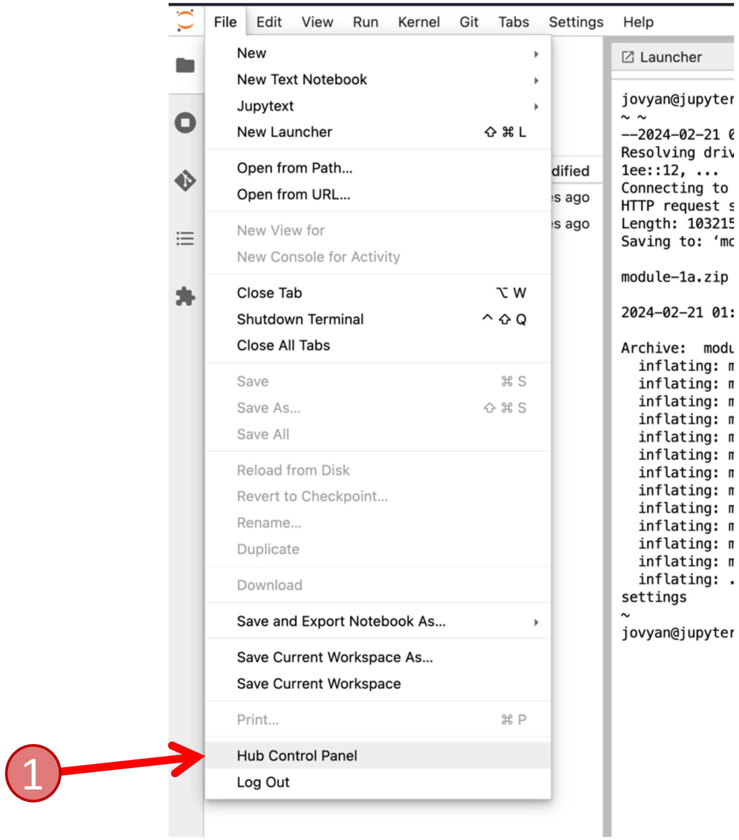
```
wget -O module-1a.zip https://drive.switch.ch/index.php/s/wNzW6ntm1zbbXfa/download
unzip module-1a.zip
```

You should a new folder

```
./module-1a-main
```

Jupyter Lab – Exercises module 1a

- If you need to restart your jupyter lab server

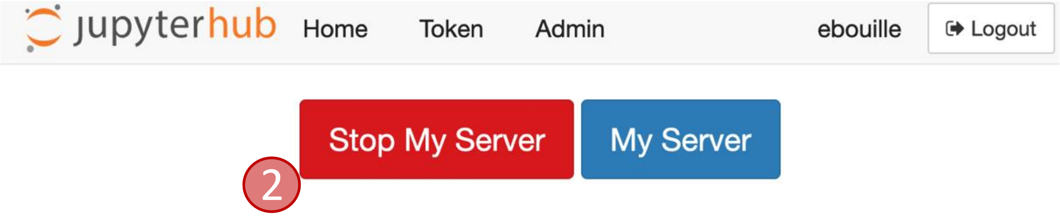


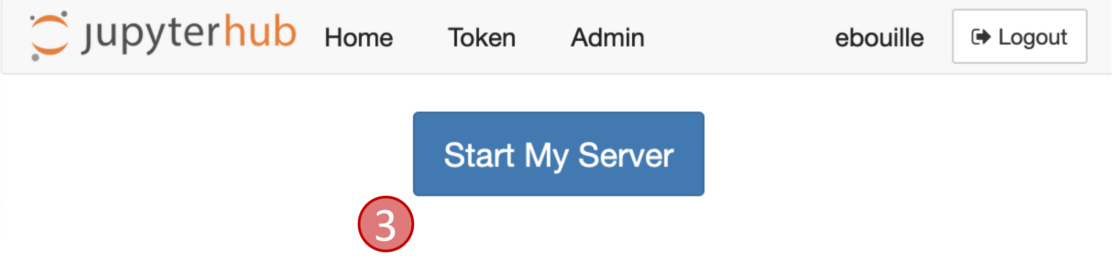
```
jovyan@jupyter
~ ~
--2024-02-21 0
Resolving driv
1ee::12, ...
Connecting to
HTTP request s
Length: 103215
Saving to: ‘mo

module-1a.zip

2024-02-21 01:

Archive:  modu
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: m
  inflating: .
settings
~
jovyan@jupyter
```

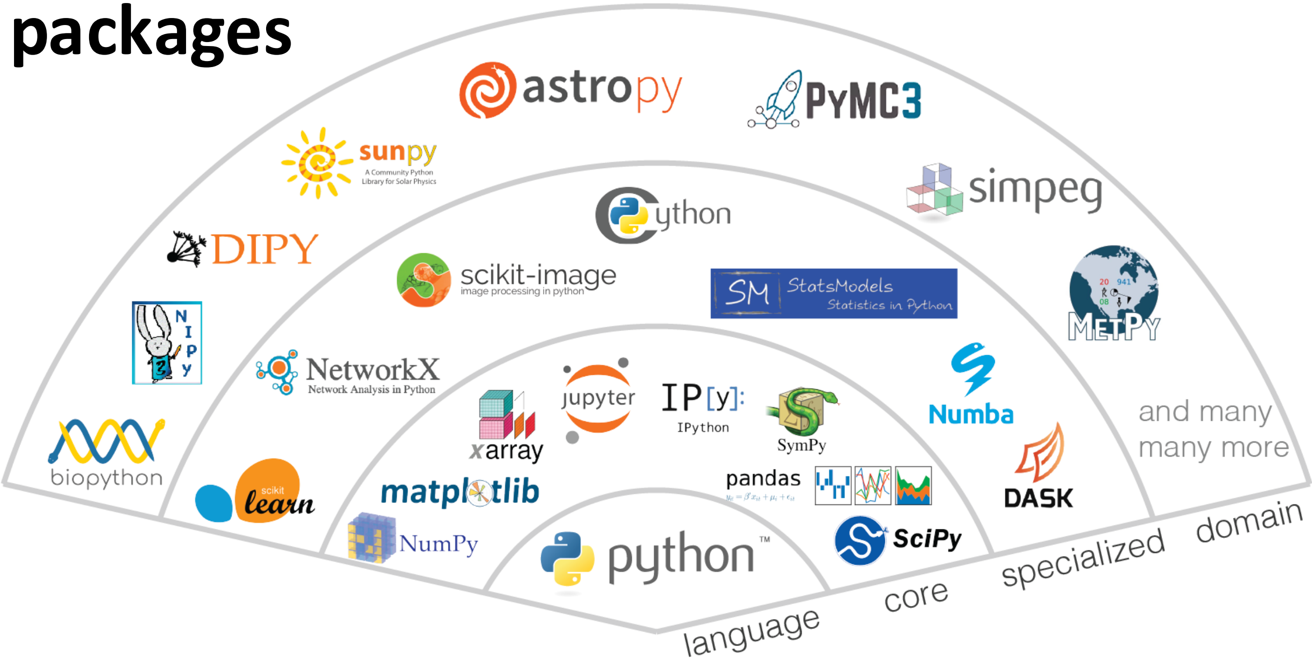




Gentle Introduction to Data Science With Python

Python Data Science Ecosystem

- **Python**
  - Core programming language used in the class
- **Python Math & Data Science packages**
  - Numpy
  - Pandas
  - Scikit-Learn
  - …



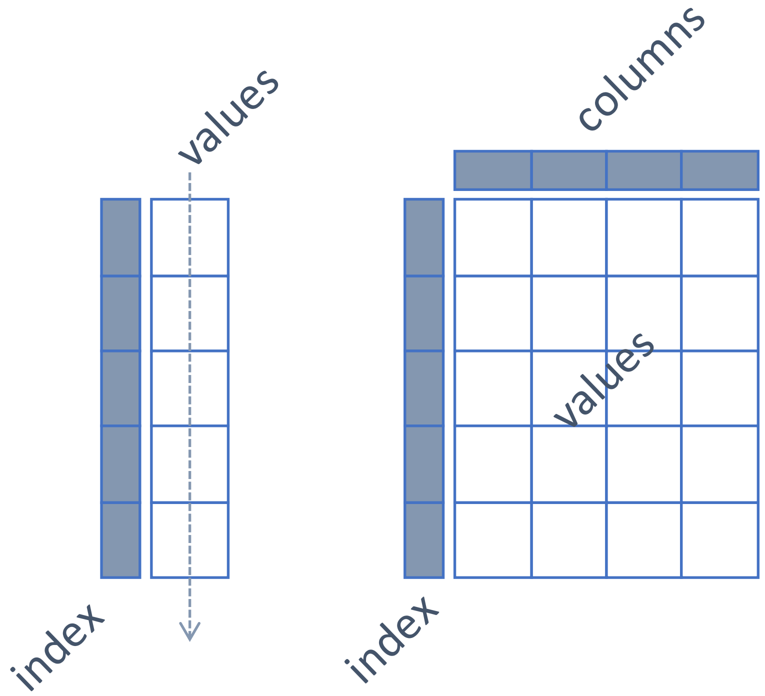
and many more …

Python Data Science Ecosystem

- **Numpy**
  - Core library for scientific computing in Python
  - Provides a high-performance multidimensional array object, <N>-D
  - Large collection of high-level mathematical functions to operate on arrays objects
  - Optimized for size and performance
- **SciPy**
  - Built on NumPy
  - Mathematical library for Scientific and Technical Computing
    - Integration, linear optimization, spatial, stats, FFT, ...

Python Data Science Ecosystem

- **Pandas**
  - 1D or 2D structures
  - Built on top of NumPy
    - NumPy stores your data in arrays
    - Pandas takes the arrays, ...
      ... and gives you labelled index to it
    - Basically dictionary based NumPy *ndarray*
  - Powerful & flexible data munging library
  - Recommended reading: [pandas documentation]()



1-D
(Series)

2-D
(DataFrame)

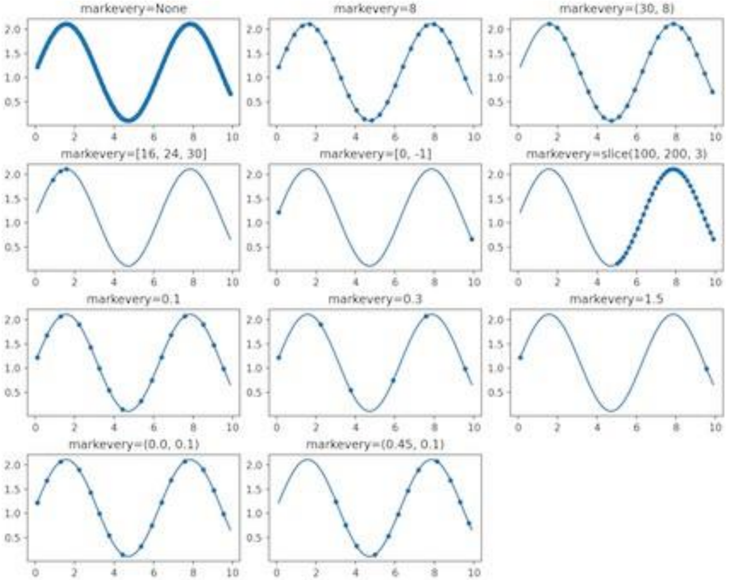
Python Data Science Ecosystem

- **Scikit-learn - Machine Learning in Python**
  - Model algorithms (Classification, Regression, Clustering, NN, ...)
  - Performance metrics
  - Model hyper-paremeter tunings
  - Model Training, Validation
  - Feature selection
  - Data Processing, Pipelines
  - ...
- **PyTorch, TensorFlow**
  - AI, Deep Learning
  - GPU-based optimization
  - ...

Python Data Science Ecosystem

- **Matplotlib**
  - The library for creating visualizations in Python
  - Pandas’ default visualization engine

    `pandas.DataFrame.plot()`
  - Powerful, but low level programming interface
  - Best for quick and basic data exploration
- **Alternatives**
  - Plotly
  - Seaborn, folium, bokeh, osmnx, vispy, pygal, cufflinks, …


markevery=None
markevery=8
markevery=(30, 8)
markevery=[16, 24, 30]
markevery=[0, -1]
markevery=slice(100, 200, 3)
markevery=0.1
markevery=0.3
markevery=1.5
markevery=(0.0, 0.1)
markevery=(0.45, 0.1)