Introduction to Practice Lessons for Deep Learning Course

Claudiu D. Hromei, Federico Borazio, Roberto Basili

Deep Learning a.a. 2023 - 2024

Who am I?

- Claudiu Daniel Hromei: you can call me Claudiu or Claudio, but not Daniel (please)
- Bachelor's and Master's at Tor Vergata, completed in 2020
- Currently PhD at Tor Vergata *Artificial Intelligence, XXXVII cycle, course on Health and life sciences, organized by the Università Campus Bio-Medico di Roma*
- I am interested in (and very pleased to talk about):
  - Artificial Intelligence
  - Machine Learning
  - Natural Language Processing
  - Computational Linguistics and Grammar (people call me nazi-grammar)
  - Food (pasta, pizza, hamburgers, burritos, …) & Beer
  - Video games (Civilization, TFT, Baldur's Gate, …)

https://scholar.google.it/citations?user=YQRKKFoAAAAJ&hl=it

hromei@ing.uniroma2.it

Who am I?

- Federico Borazio: you can call me Federico
- Bachelor's and Master's at Tor Vergata, completed in 2023
- Currently PhD at Tor Vergata *in Data Science course, Università degli Studi di Roma Tor Vergata, XXXIX cycle Department of Enterprise Engineering*
- I am interested in (and very pleased to talk about):
  - Artificial Intelligence
  - Machine Learning
  - (Clinical) Natural Language Processing
  - Food (pasta, pizza, hamburgers, kebab, …) & Beer
  - Sport (football, gym, running, …)

borazio@ing.uniroma2.it

Organization of the Labs

- After a major (theoretical) topic we will have a (practical) Lab, where we will solve a problem using the algorithms/architectures
- Usually:
  - we will introduce the problem, the objective and the data used
  - we'll start with a prepared Notebook (Jupyter/Colab)
  - we'll give you some exercises (like playing with the hyper-parameters and optimize the solution)
  - we will ask you to send us via email your solutions (this could grant you some extra points in the final exam)
- E.g. after HMMs we will have a Lab (20° of March?) and after Transformers (29° of April?)

A challenge for you

The Course Challenge: Information Extraction

- The challenge will be hosted on Kaggle.com, I highly recommed you to create an account there if you haven't already
- It will run until June/July 2024
- There will be a description of the task to be solved and we ask you to give more and more solutions, based on what you learn during the course
- You will be able to download the data from the website
- You will upload your solution on Kaggle and it will be evaluated automatically
- Participation is optional, but is required to build a team (2 or 3 people)
- Finally, we will give you more details next time!

AI course competition

This competition is intended for the AI students at Tor Vergata. The objective is to detect the sentiment in a given sentence.

Overview

This competition is designed to provide a practical platform for AI students at the University of Rome Tor Vergata to apply and experiment with the various methodologies and algorithms they will acquire during their coursework.

The specific challenge at hand involves performing Sentiment Analysis (SA) on Twitter data, brief texts called tweets. Over the years, a dataset of tweets has been curated and labeled with one of three polarities:

- **Positive**: for tweets that elicit joyful sentiments.
- **Negative**: for tweets that evoke sad sentiments.
- **Neutral**: for tweets that do not lean towards either a positive or negative sentiment.

We encourage students to access the training dataset, located in the "Data" tab, and delve into the tweets in order to devise solutions. These solutions may be based on specific keyword searching or may leverage Machine Learning techniques. As the course progresses, students are encouraged to submit multiple solutions as they gain exposure to new methods and approaches.

Objective: To predict the sentiment polarity of a given tweet, categorizing it as positive, negative, or neutral.

Start
Sep 29, 2023

Close
Feb 28, 2024

Competition Host

Claudiu Daniel Hromei

Prizes & Awards

Kudos
Does not award Points or Medals

Participation

7 Competitors
3 Teams
4 Entries

Tags

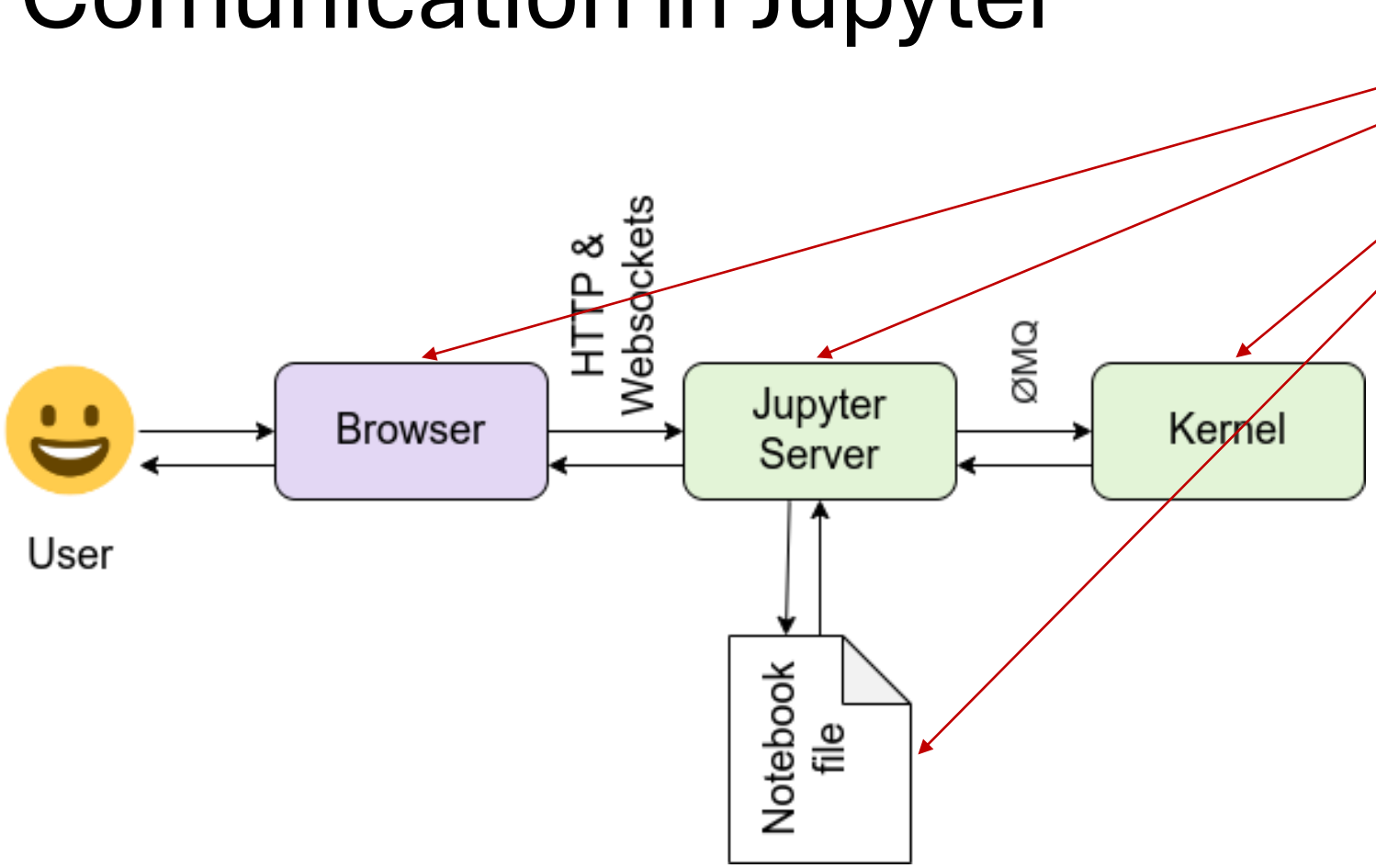
Add Tags

Table of Contents

+ Add Section

Introduction to Jupyter/Colab Notebooks

Comunication in Jupyter

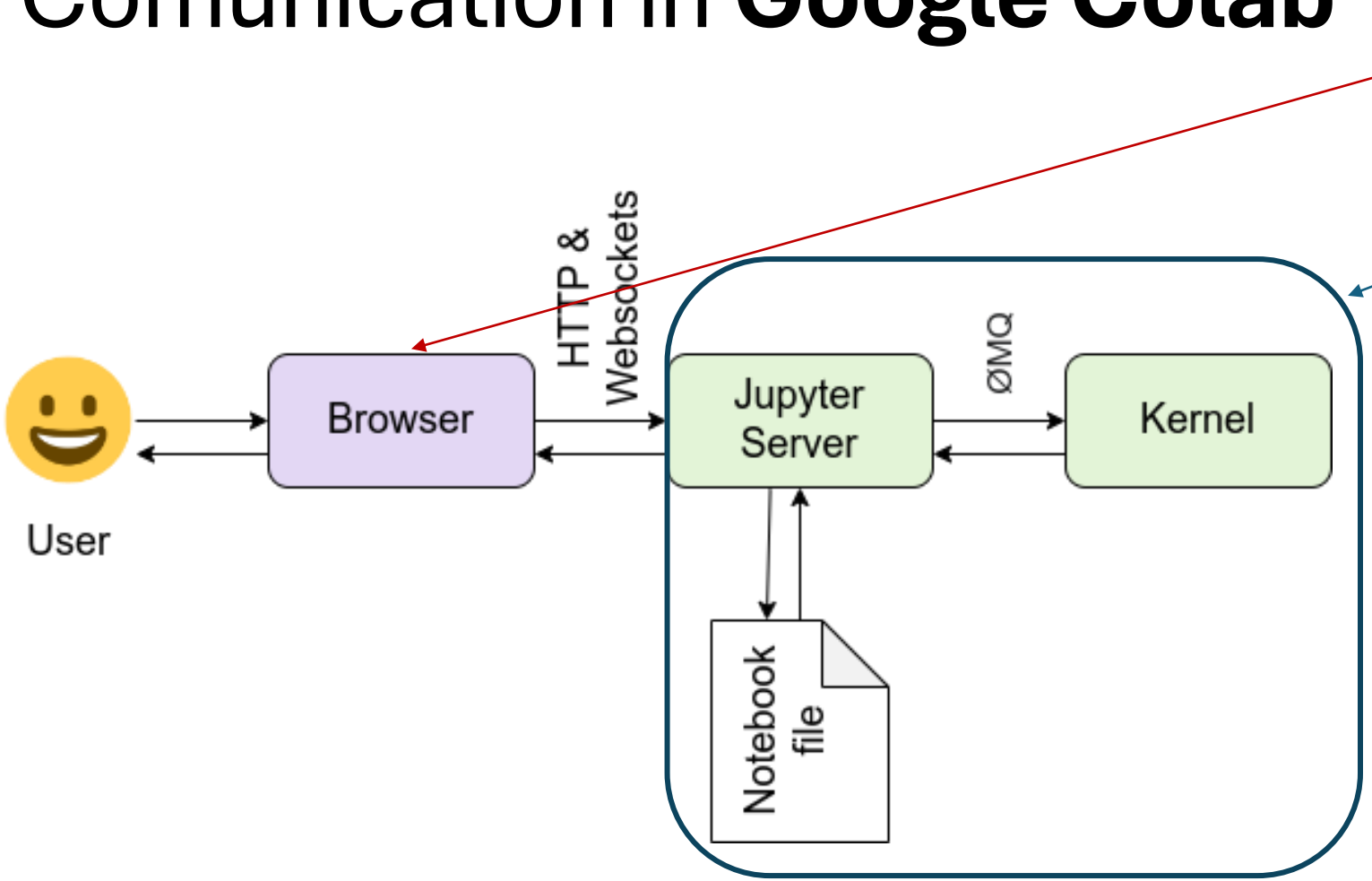


Local

- You need to install every package on your machine
- You rely on your machine computational resources

Taken from docs on jupyter.com

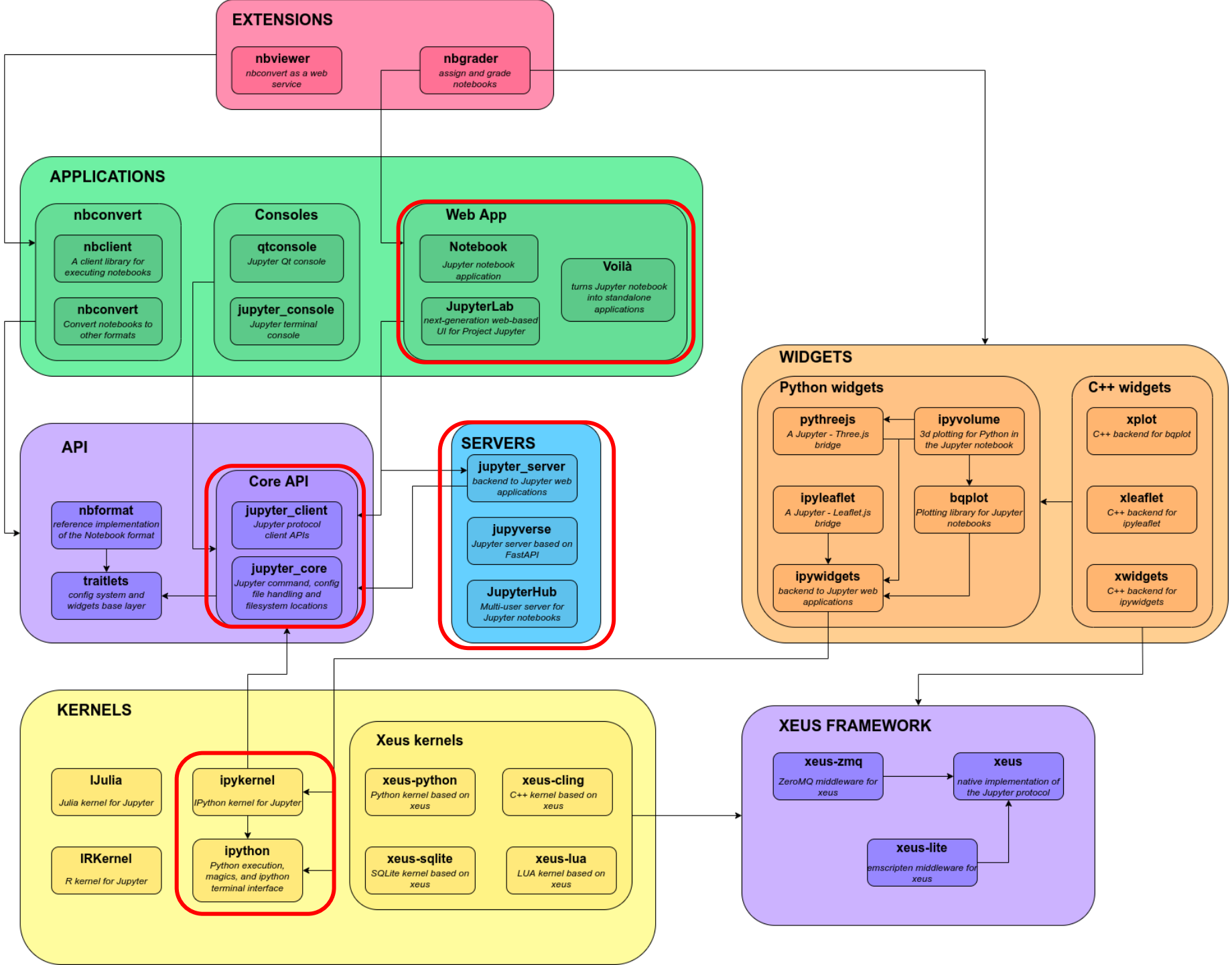
Comunication in **Google Colab**



- It will handle the packages (you still need to install them)
- It will give for free some computational resources

Readjusted from docs on jupyter.com

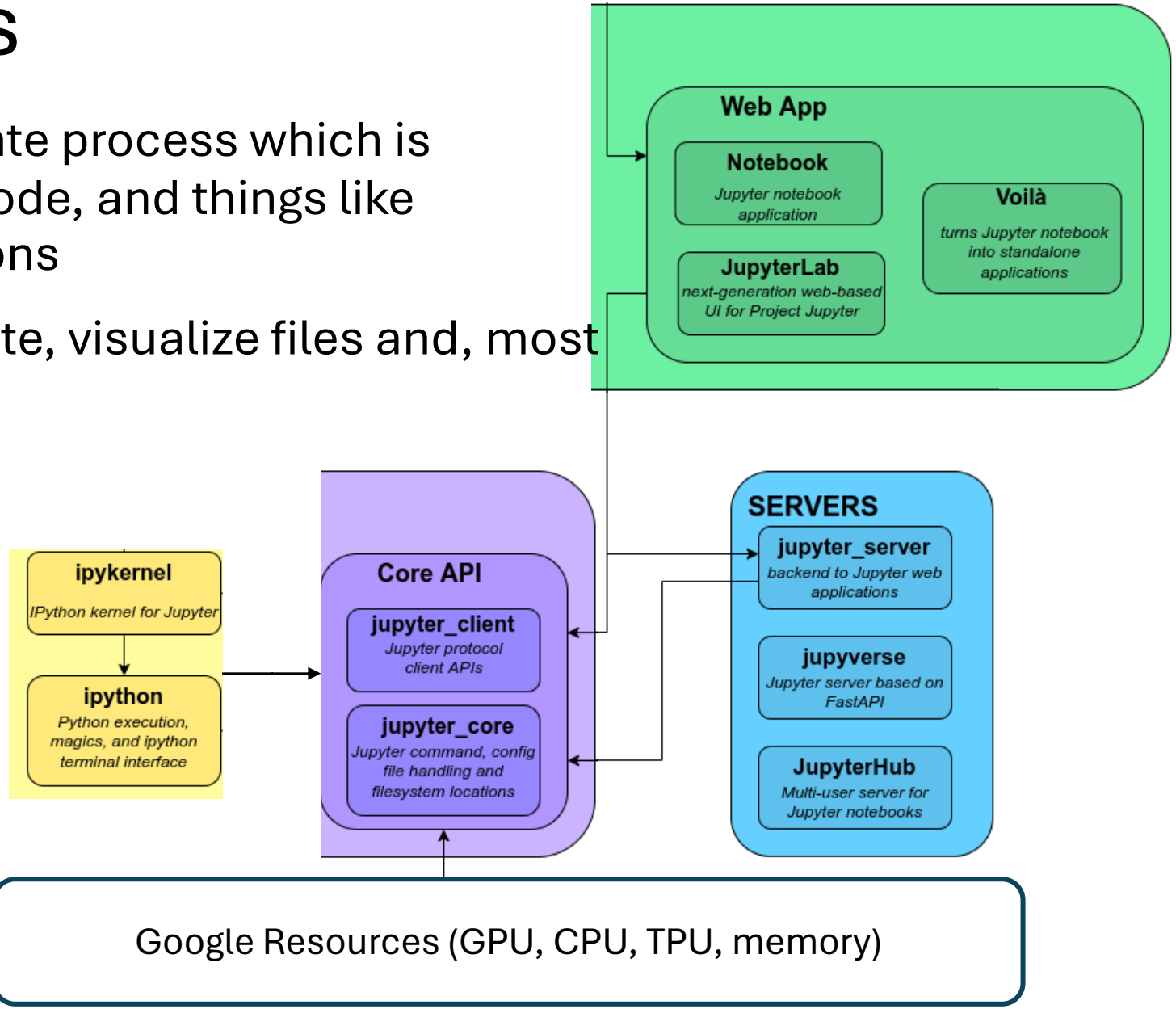
Jupyter Architecture



Taken from docs on jupyter.com

Important modules

- *Kernels*: *IPykernel* is a separate process which is responsible for running user code, and things like computing possible completions
- *Web App*: Interface to navigate, visualize files and, most importantly, to code!
- *Servers*: Local (or online) server to store files and variables (memory)
- *Core API*: File system, APIs and Protocols Manager



Introduction to Pytorch

PyTorch

PyTorch is an optimized Deep Learning tensor library based on Python and Torch and is mainly used for applications using GPUs and CPUs.

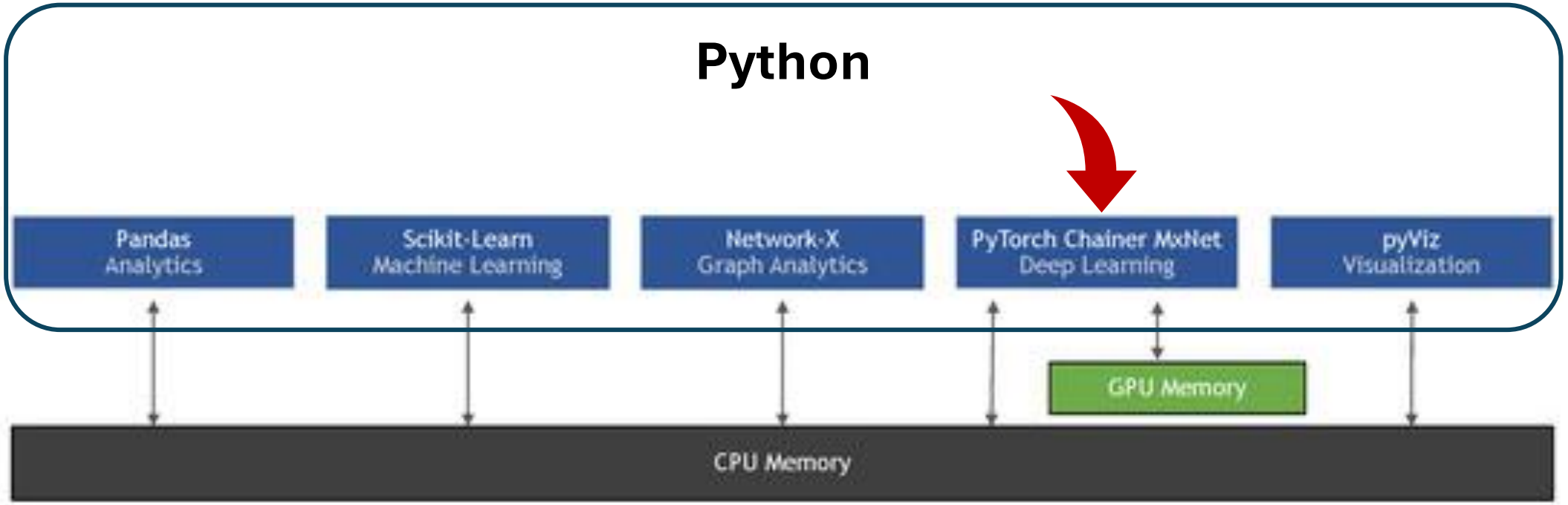
The two main features of PyTorch are:

- **Tensor Computation** (similar to NumPy) with strong GPU (Graphical Processing Unit) acceleration support
- **Automatic Differentiation** for creating and training deep neural networks: it dynamically creates the optimized flow, based on your code, to execute *at run time*!

We usually import into our code the `Optimizer` and the `NeuralNetwork` (`NN`) module

I highly recommend you to take a look at the Documentation

Where is PyTorch?



Taken from: Raschka, S.; Patterson, J.; Nolet, C. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information* **2020**, *11*, 193. https://doi.org/10.3390/info11040193

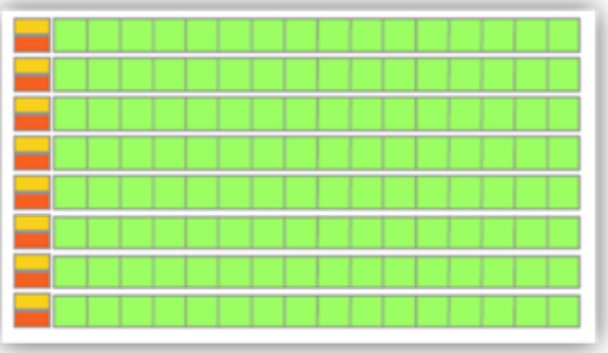
Why PyTorch and GPUs?

More details at https://blogs.nvidia.com/blog/why-gpus-are-great-for-ai/

CPU

* Low compute density
* Complex control logic
* Large caches (L1$/L2$, etc.)
* Optimized for serial operations
  - Fewer execution units (ALUs)
  - Higher clock speeds
* Shallow pipelines (<30 stages)
* Low Latency Tolerance
* Newer CPUs have more parallelism

GPU



* High compute density
* High Computations per Memory Access
* Built for parallel operations
  - Many parallel execution units (ALUs)
  - Graphics is the best known case of parallelism
* Deep pipelines (hundreds of stages)
* High Throughput
* High Latency Tolerance
* Newer GPUs:
  - Better flow control logic (becoming more CPU-like)
  - Scatter/Gather Memory Access
  - Don't have one-way pipelines anymore

Image taken from https://hypertec.com/blog/gpus-taking-over-cpus/