Data science project

Why we need API ?

Presented by **Morgan Gautherot**

Lack of Modularity & Maintainability

- Notebooks encourage writing code in a **monolithic** way, often mixing data exploration, preprocessing, model training, and evaluation in a single file.
- Production systems need **modular, reusable, and testable** code, typically structured into multiple files.

Hidden State Issues

- Code execution in notebooks is **non-linear**, meaning cells can be run out of order, leading to inconsistent states.
- This makes debugging difficult, as variables may depend on previous executions that aren't reflected in the notebook.

Poor Version Control

- Notebooks store code and output in **JSON format**, making them harder to track with **Git** compared to plain Python scripts.
- Merging changes in notebooks is difficult, making collaboration more error-prone.

Limited CI/CD Integration

- Most production workflows rely on **Continuous Integration/Continuous Deployment (CI/CD)** pipelines, which test and deploy code automatically.
- Notebooks aren’t easily integrated into these pipelines without extra tooling like **Papermill** or **nbconvert**.

Reproducibility Challenges

- Due to hidden states and non-deterministic execution, re-running a notebook may produce **different results**.
- Production environments require **consistent, reproducible** results.

Scalability Issues

- Notebooks often run in a **single interactive session**, making it difficult to scale across multiple machines.
- Production ML models often require **distributed computing**, batch processing, or integration with cloud environments.

Performance Overhead

- Notebooks are designed for **experimentation**, not optimized for performance.
- Production code is typically packaged as **Python scripts or APIs** to run efficiently in different environments.

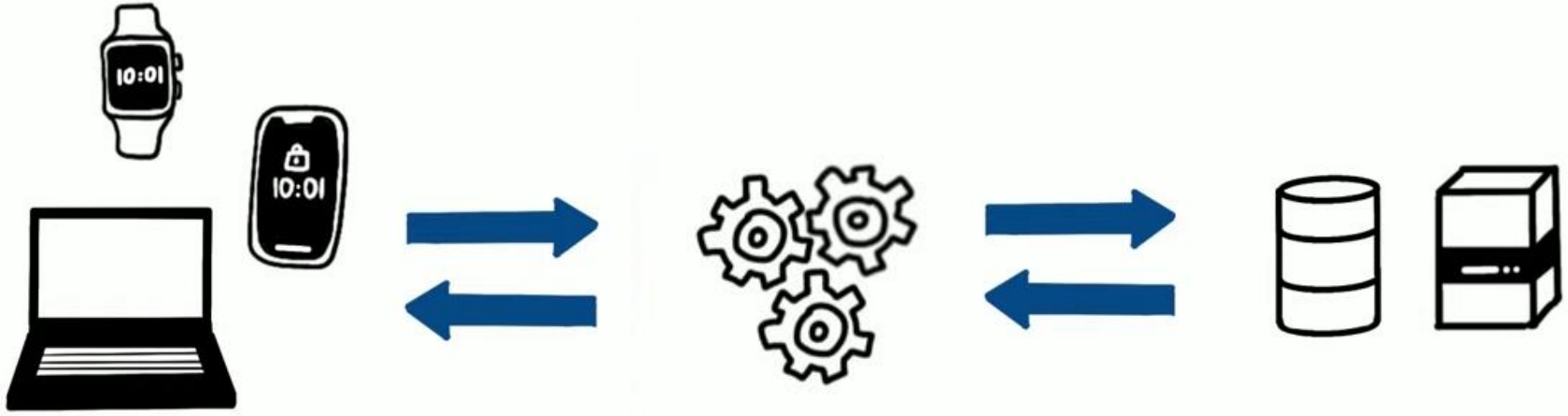
What’s the alternative ?

- Convert notebooks into **modular Python scripts** (.py files).
- Use tools like **MLflow** or **DVC** for reproducibility.
- Implement **unit tests** and CI/CD pipelines for robustness.
- Package ML models as **APIs** (e.g., using FastAPI, Flask) for real-world usage.

What is an API ?

- API (Application programming interface)
- Create a simple dialog between applications
- An application exposes the API for sending data
- Another application consumes the API to receive the data

create a simple dialog between applications



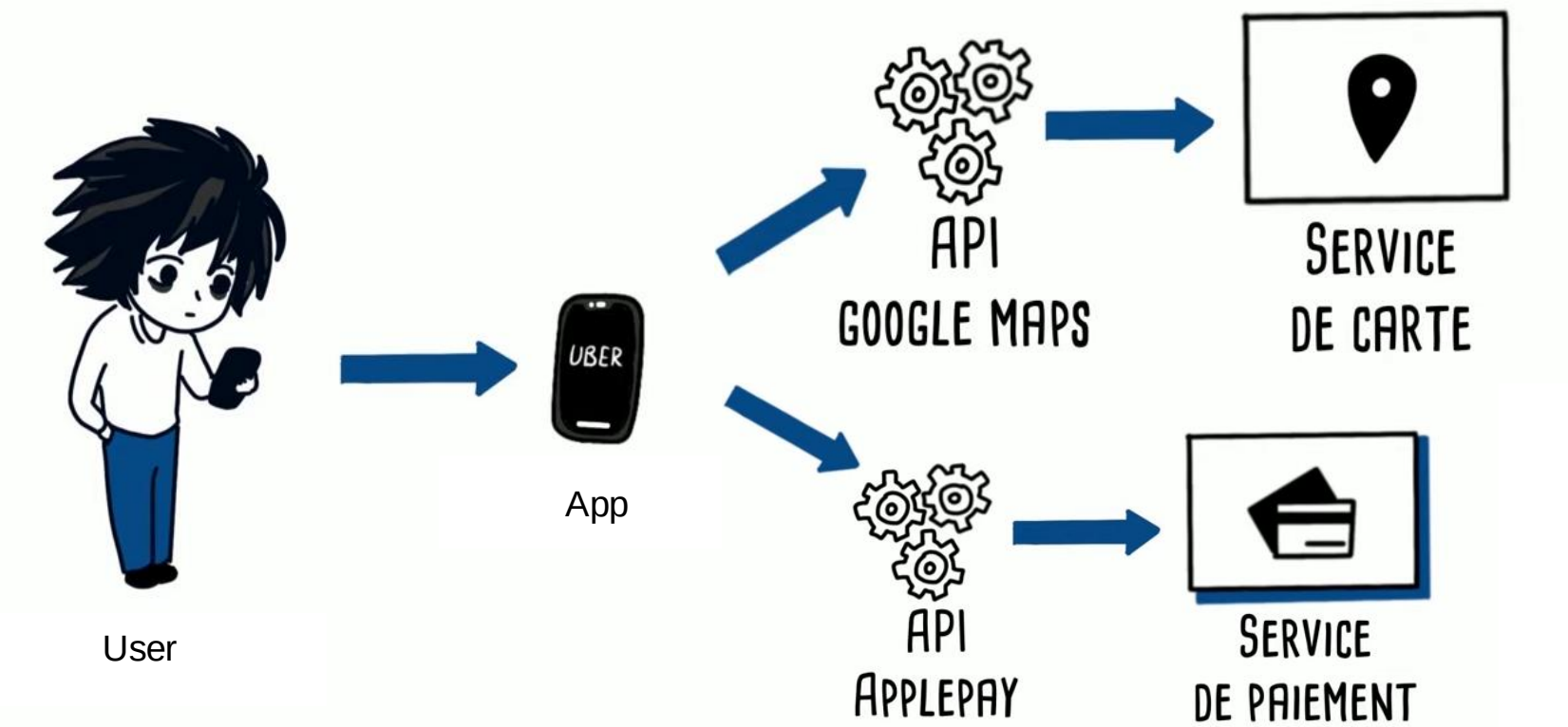
Clients

service consumer

Third-party system

service producer

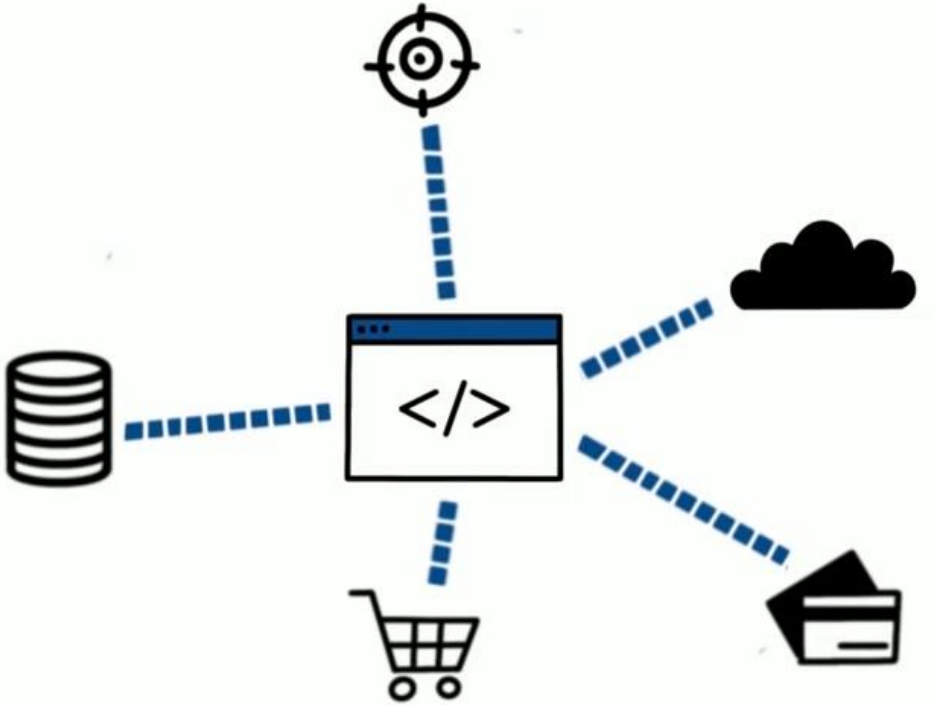
Exemple with Uber



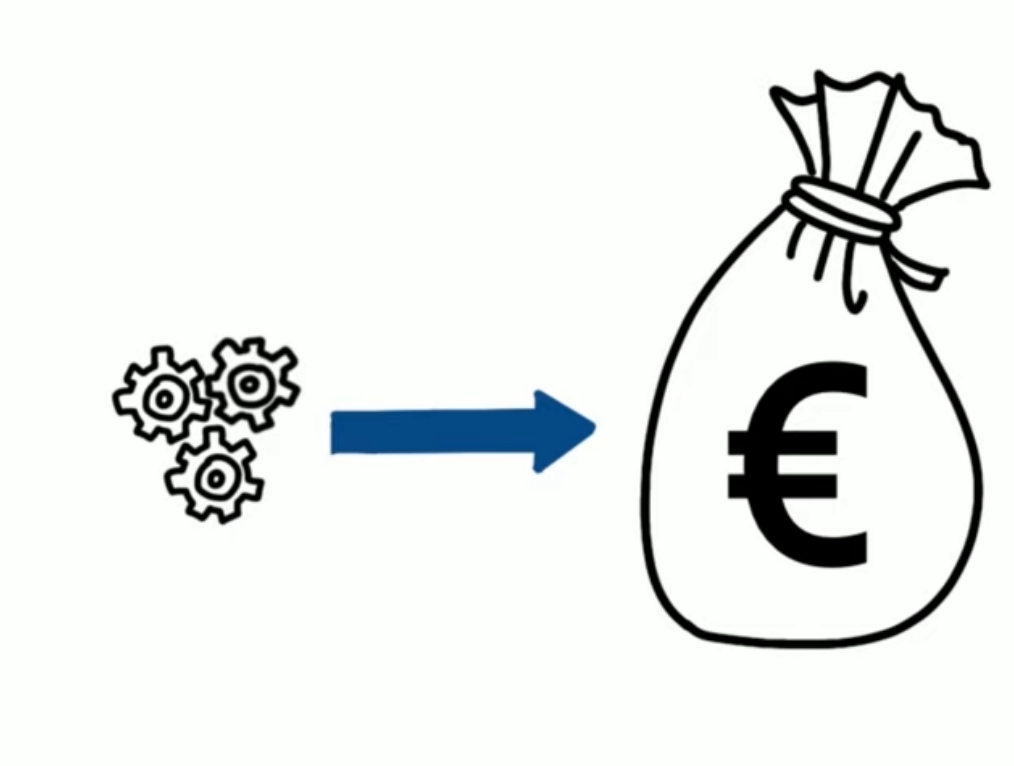
User

App

Simplification of application development



API is a product



REST API

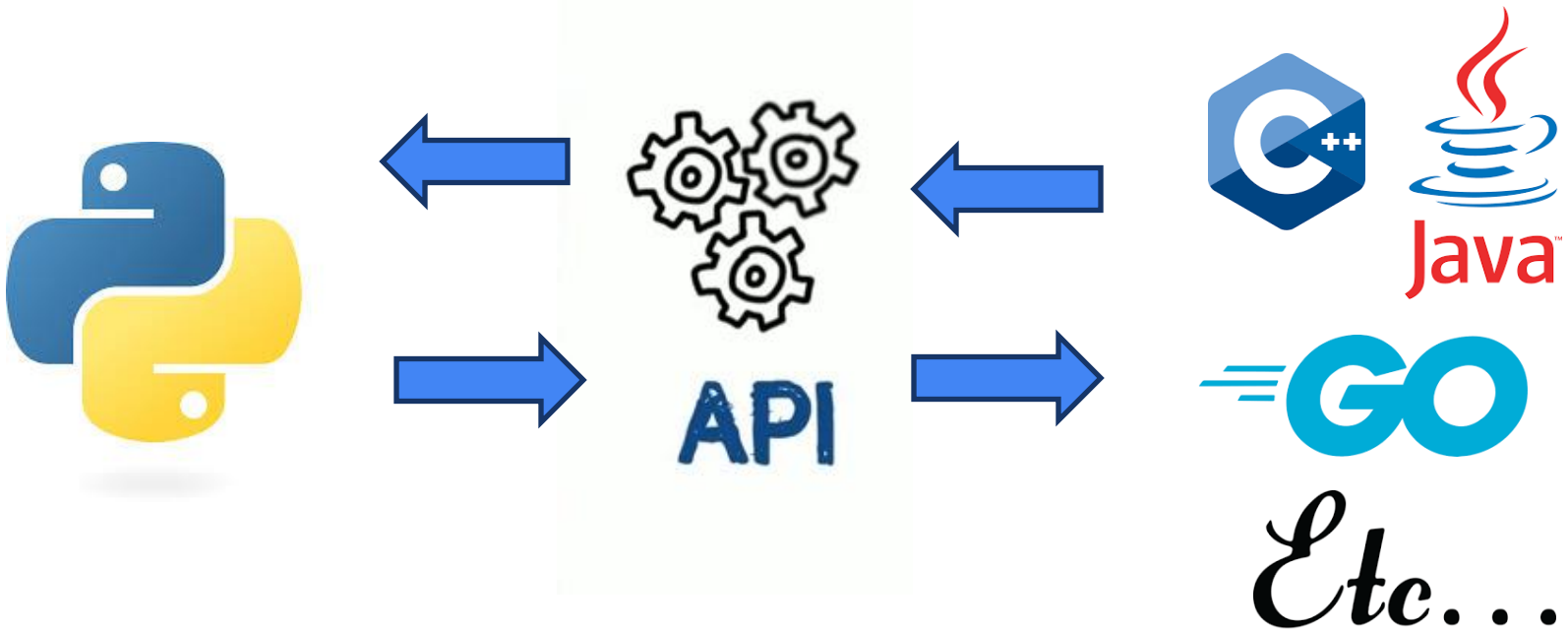
Compliance with
web standards

REST architecture based on HTTP

Better communicaiton from data science to software engineer

Data science team

Software engineer team



FastAPI

- Speed & Performance
- Automatic Data Validation & Serialization
- Built-in Interactive Documentation
- Asynchronous Support for High Concurrency
- Easy Deployment & Integration
- Standard in ML & Data Science Applications

Github

- Version Control with Git
- Collaboration & Teamwork
- Code Review & Quality Control
- CI/CD for Automation & Deployment
- Backup & Cloud Storage

Deploy your API

Render