



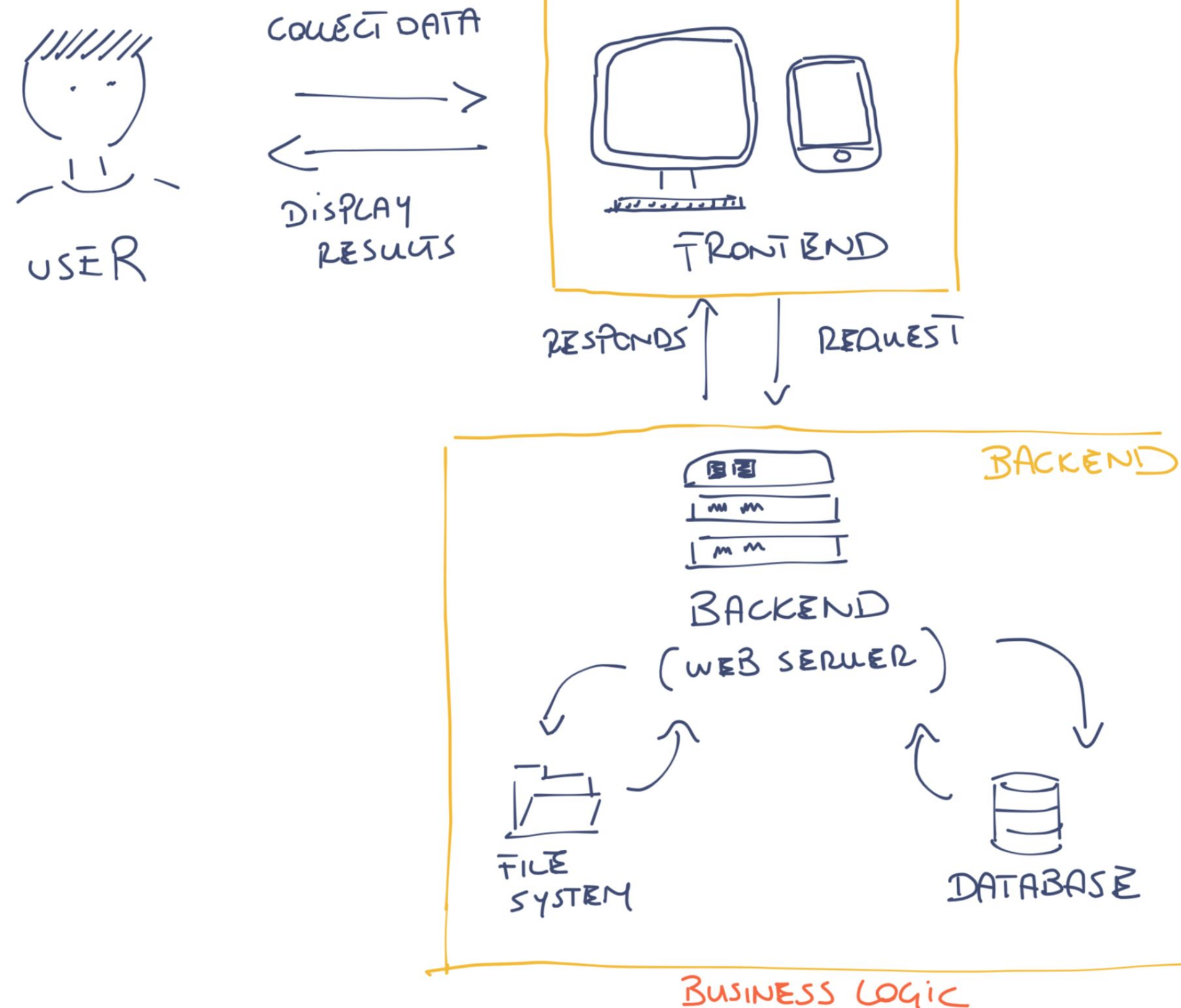
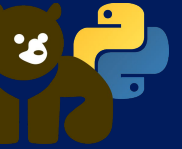
Python Biella Group

# Architettura Web

## Come si compone?

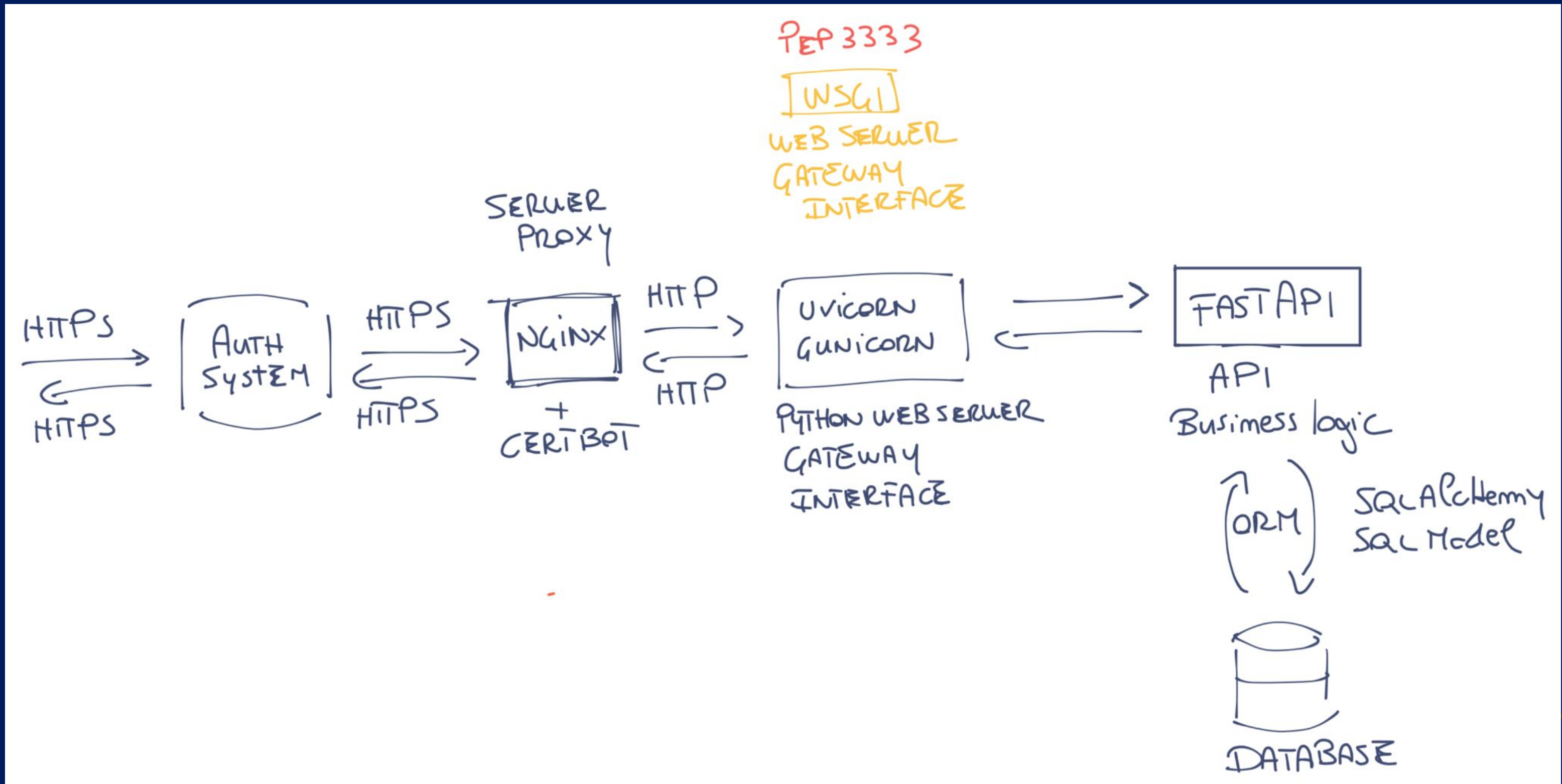
a cura di: **Andrea e Mario**

# Generic Architecture

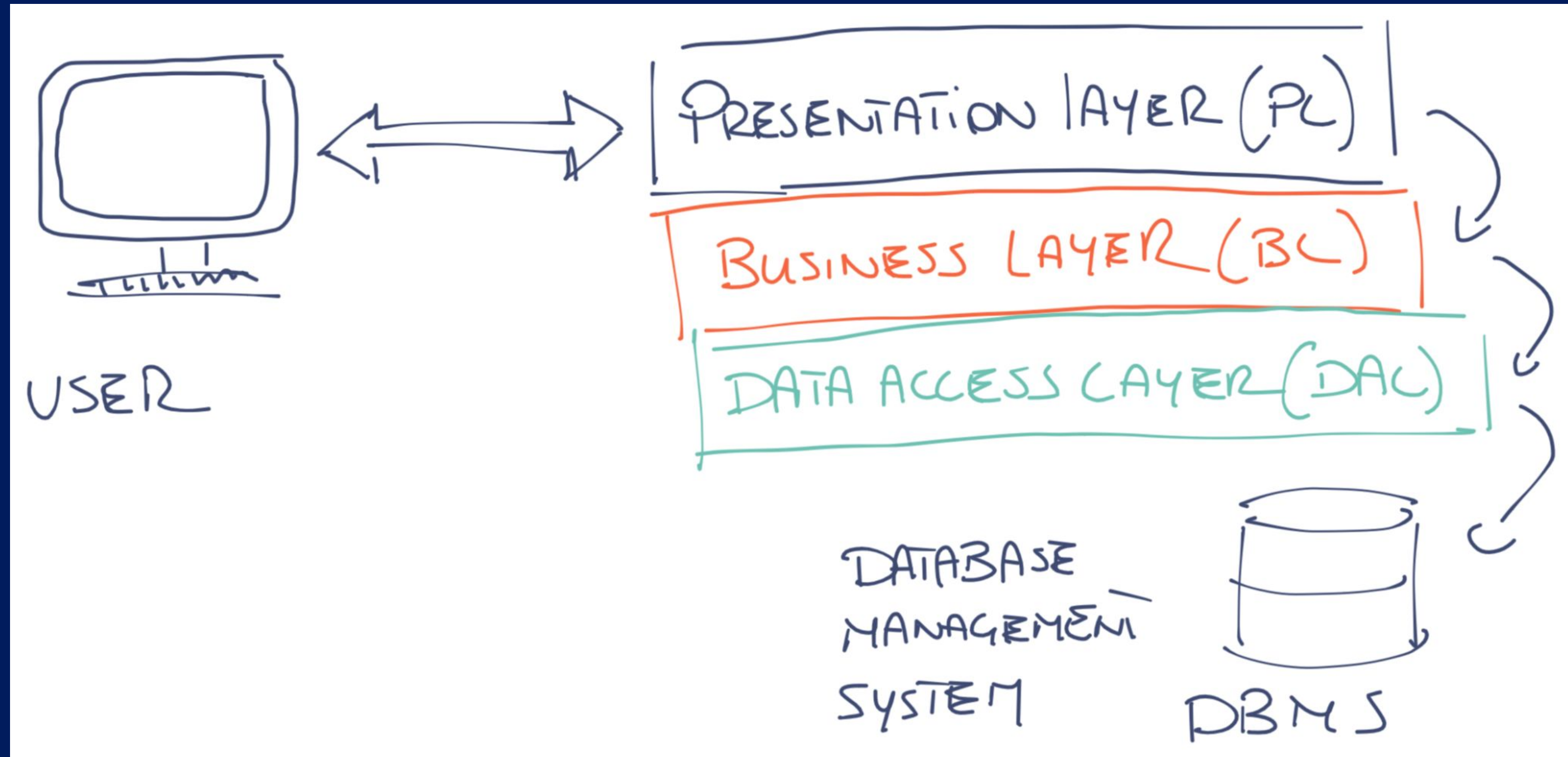
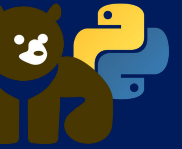




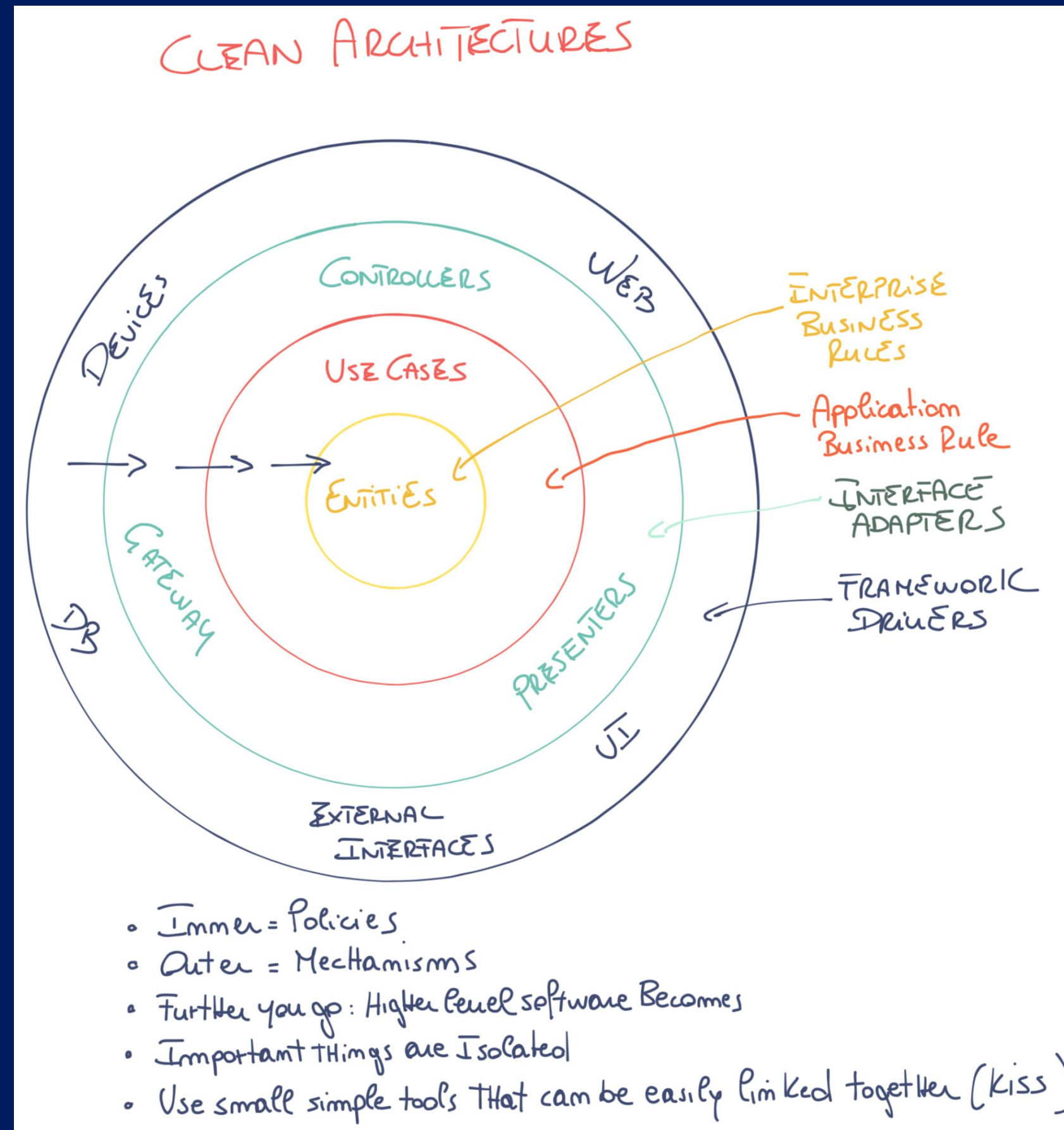
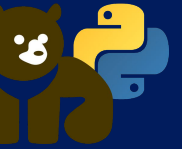
# Backend structure



# Layer Architecture

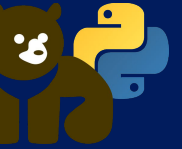


# Clean Architecture

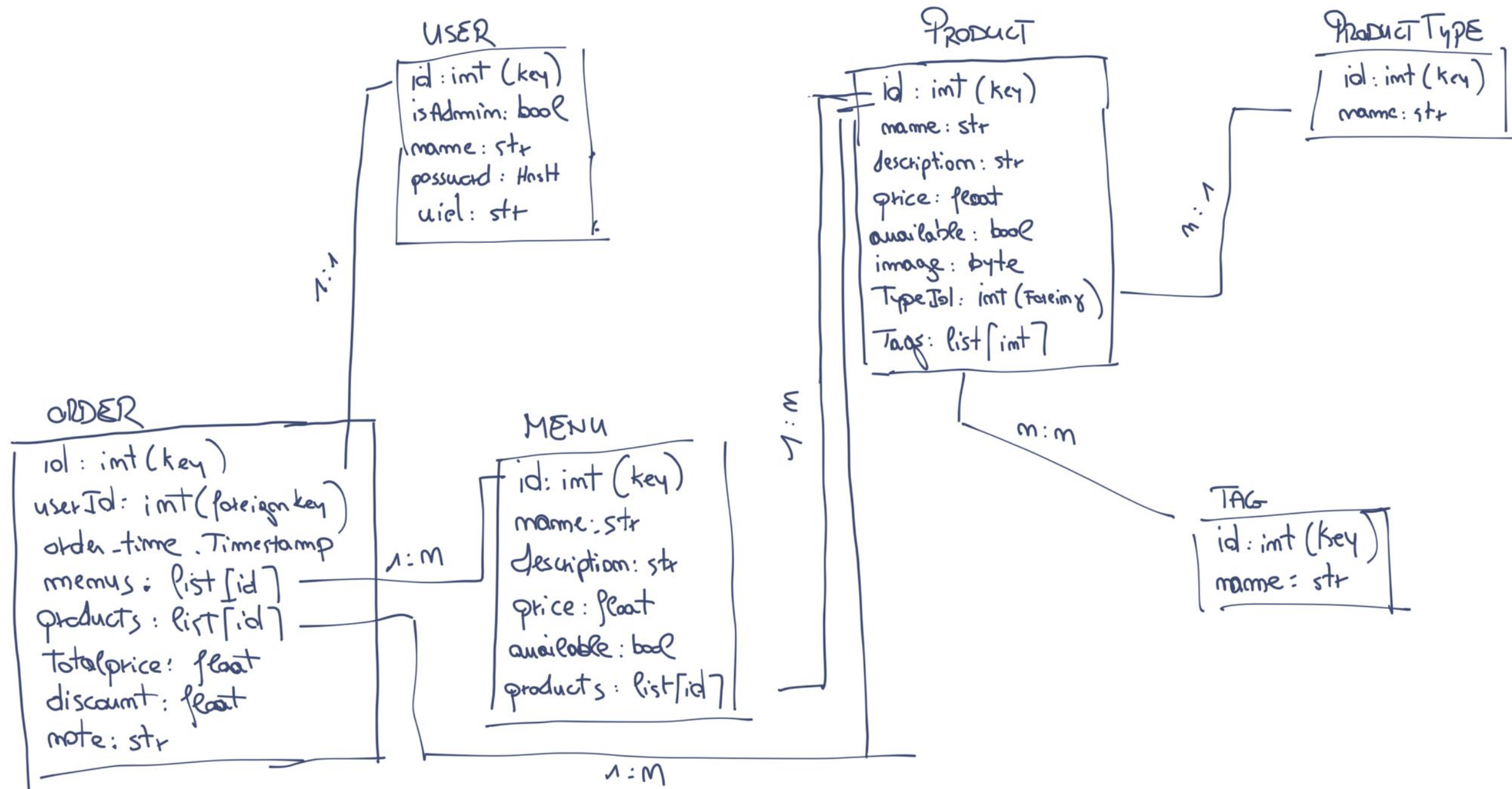




# Che cosa costruiremo?



## DATA MODEL



## API

### Product

- GET Product [id]
- GET Products
- POST Product } -> only admin
- DELETE Product [id]
- PUT (POST) Modify Product [id]
- ⊕ come filtrare l'immagine?

### ORDER

- GET ORDER [id]
- GET ORDERS -> By user
- POST order
- DELETE Order [id]
- PUT (POST) Modify Order [id] } -> By user
- GET All Orders -> By Admin

### USER

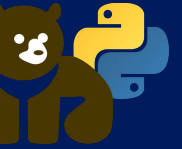
- GET user [id]
- POST user } -> By Admin
- DELETE User
- GET users -> By Admin

### Product Type

- GET Product type [id]
- GET Product types
- POST Product type
- DELETE Product type } -> By Admin
- PUT (POST) Modify Product type

### TAG

- GET Tag [id]
- GET Tags
- POST Tag
- DELETE Tag } -> By Admin
- PUT (POST) Modify tag



# Adesso parliamo di:

- Come funzionano le API
- Come configurare il proprio IDE
- Come creare il proprio virtualenv (con poetry)
- Concetti:
  - Async
  - Pydantic (dataclasses)
  - type hints
- Un piccolo esempio per iniziare