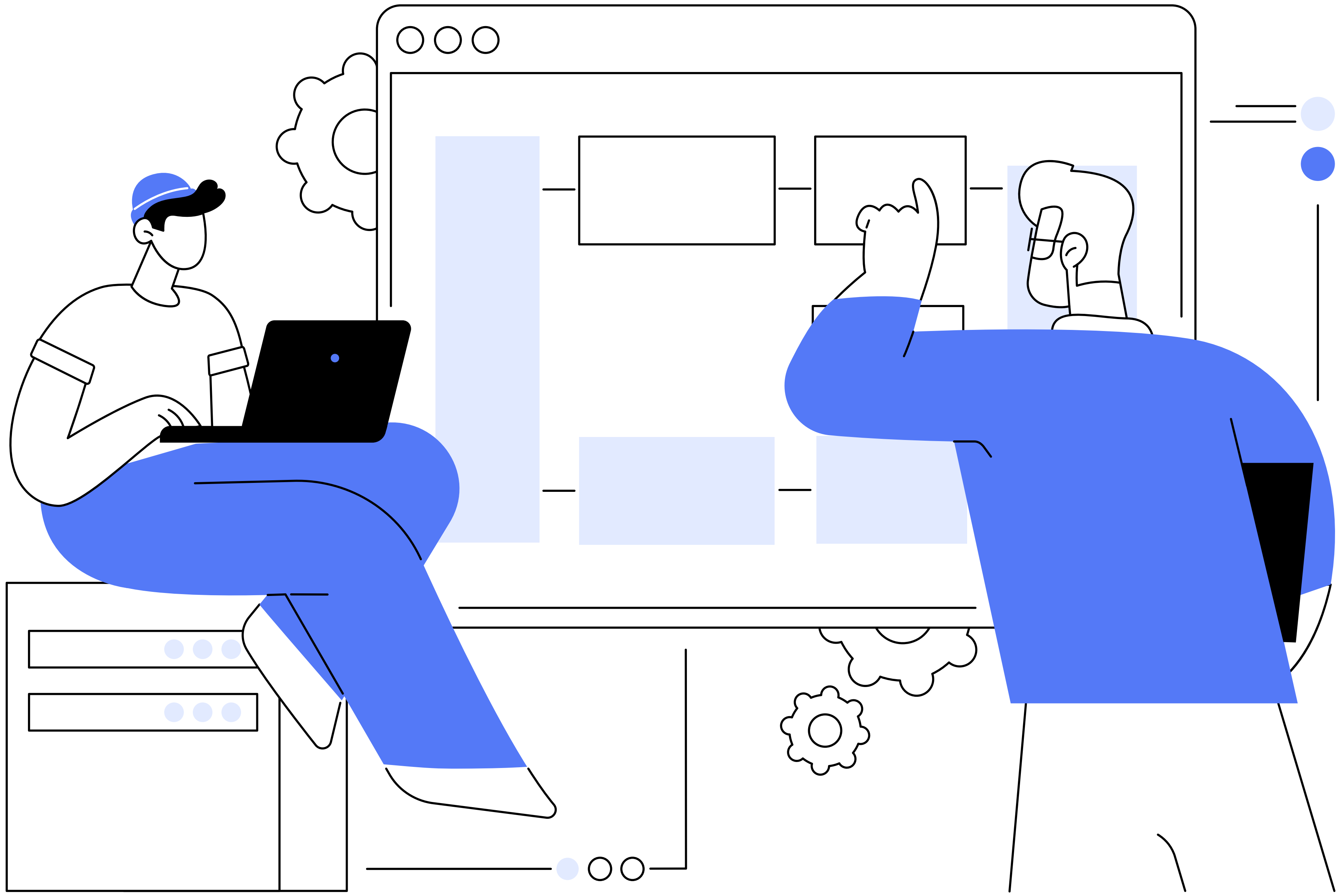
HOW TO BUILD A CHATBOT

Hands-On Workshop

WELCOME

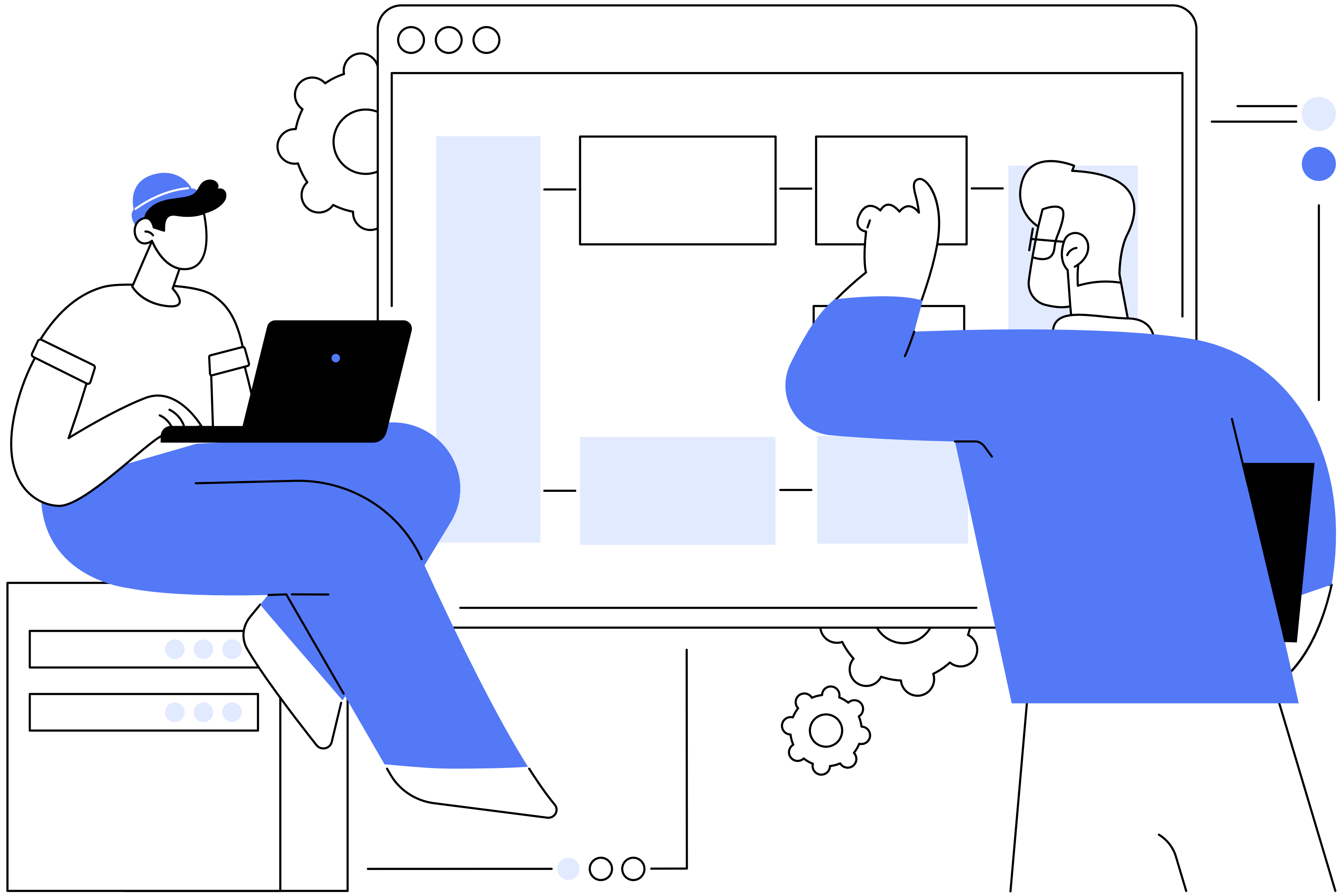
Martin Kovacs (M.Sc.)

- AI Research Engineer @ Festo
- Lecturer Machine Learning @ HS Esslingen
- Research Field:
  - Generative AI
  - LLM Agents
  - LLM Multi Agent Systems



INTRODUCTION

- Overview of the day’s agenda and workshop goals
- Introduction to workshop hardware NVIDIA Jetson Orin Nano
- Setting up the development environment



WORKSHOP AGENDA

Session 1

Theory (20 min):

Introduction to Large Language Models (LLMs)

Practise (40 min):

Deploy and use LLMs

Session 2

Theory (20 min):

Introduction to LangChain

Practise (40 min):

Use LangChain with LLMs

Session 3

Theory (20 min):

Introduction to Retrieval-Augmented Generation

Practise (40 min):

Deploy vector database, data integration & search

Session 4

Theory (20 min):

Introduction to RAG Chains in LangChain

Practise (40 min):

Implement a Q/A-RAG Chain

Session 5

Theory (20 min):

How to build a RAG-Chatbot

Practise (40 min):

Implement a RAG-Chatbot App

-> STEP BY STEP TO YOUR OWN CHATBOT

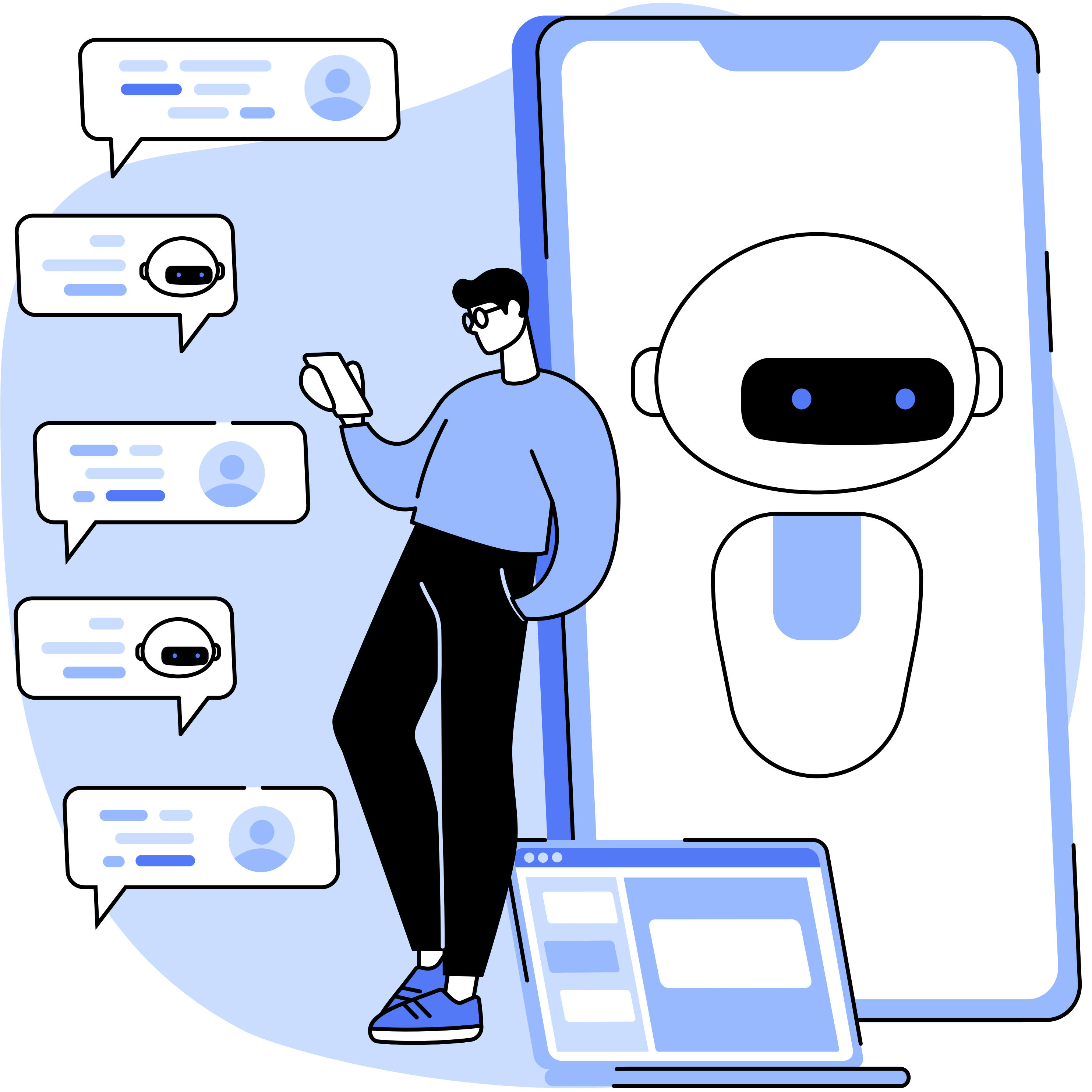
WORKSHOP GOAL

Personalized Learning Assistant:

- Create a chatbot that acts as a learning tutor.

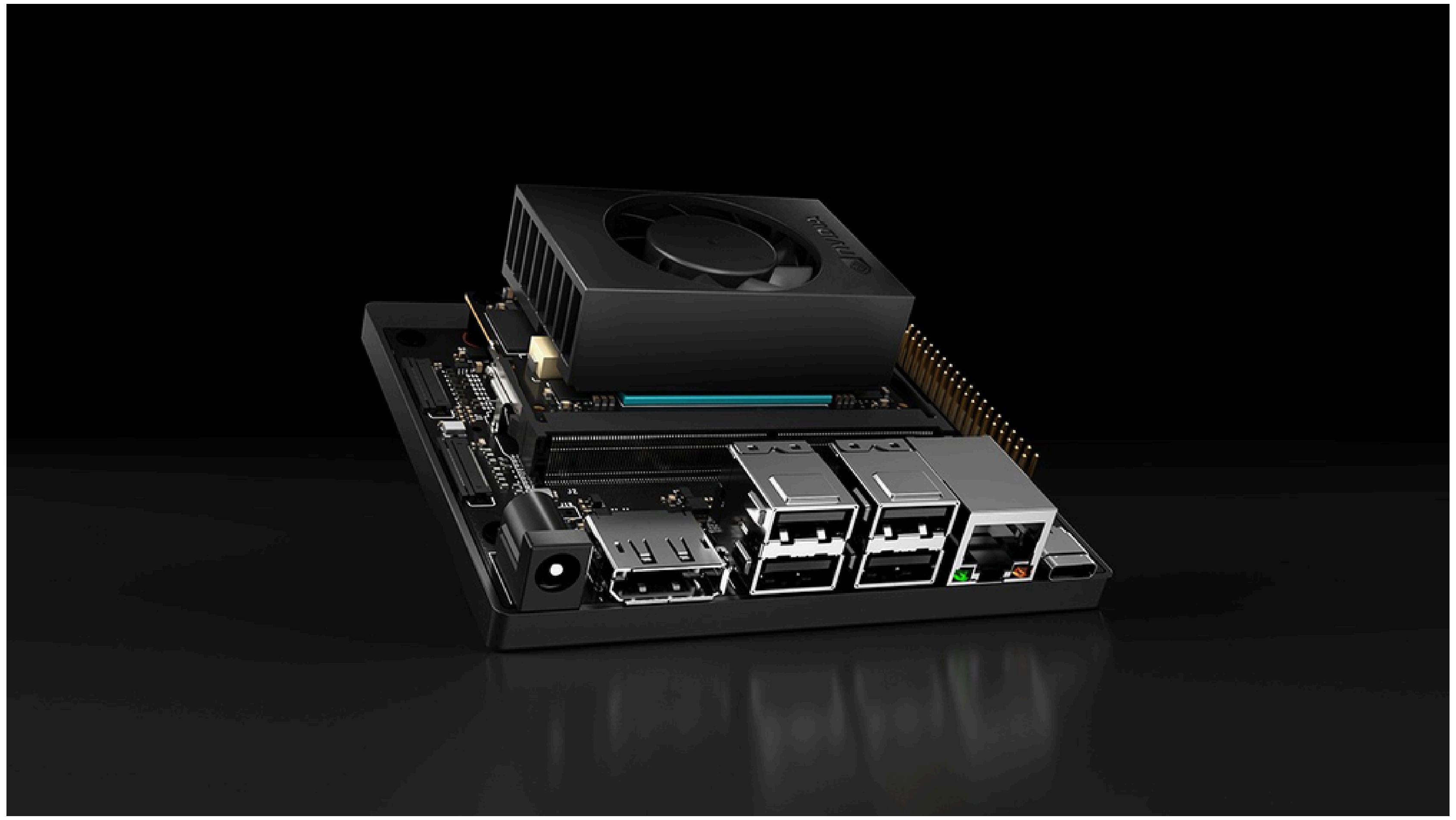
Interactive Study Tool:

- Use your own lecture script
- Ask questions about the content
- (Generate exam-related questions.)



NVIDIA JETSON ORIN NANO

- Edge AI platform
- ARM-based CPU with NVIDIA Ampere GPU
- Supports NVIDIA JetPack SDK and AI frameworks
- Ideal for on-device AI applications and models



DEVELOPMENT ENV

Hardware Layer:

- ARM CPU and NVIDIA Ampere GPU handle computing.

Operating System Layer:

- Ubuntu OS provides the base environment.

Development Tools Layer:

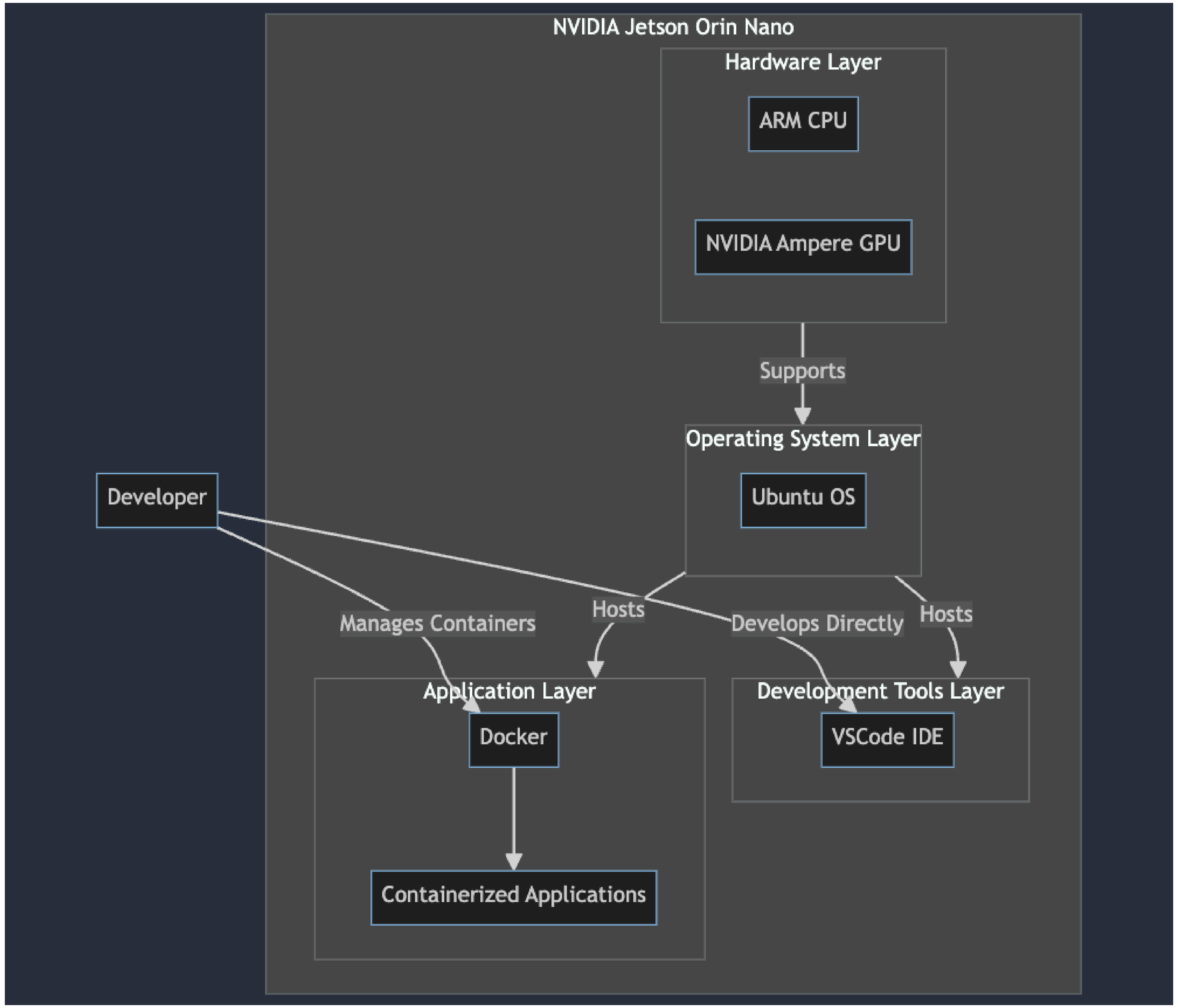
- VSCode IDE is used for direct development on the device.

Application Layer:

- Docker runs containerized AI applications.

Developer Interaction:

- Developers code and manage containers directly on the Orin Nano.



GOAL ARCHITECTURE

Frontend:

- Web app built with Gradio, accessible via browser.

Backend:

- Python-based with FastAPI and LangChain.

LLM Serving:

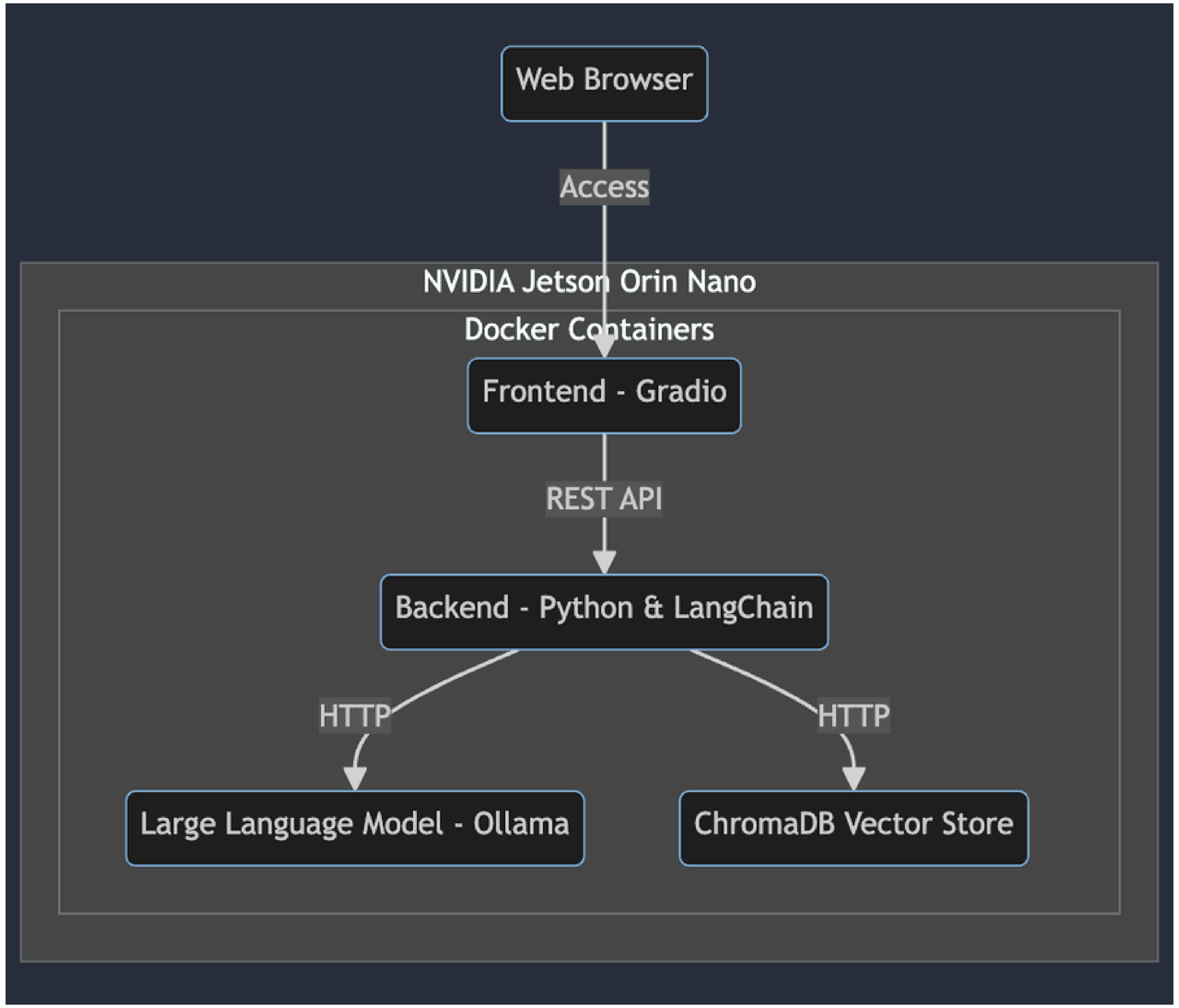
- Ollama for managing large language models.

Knowledge Storage:

- Vector database for knowledge management.

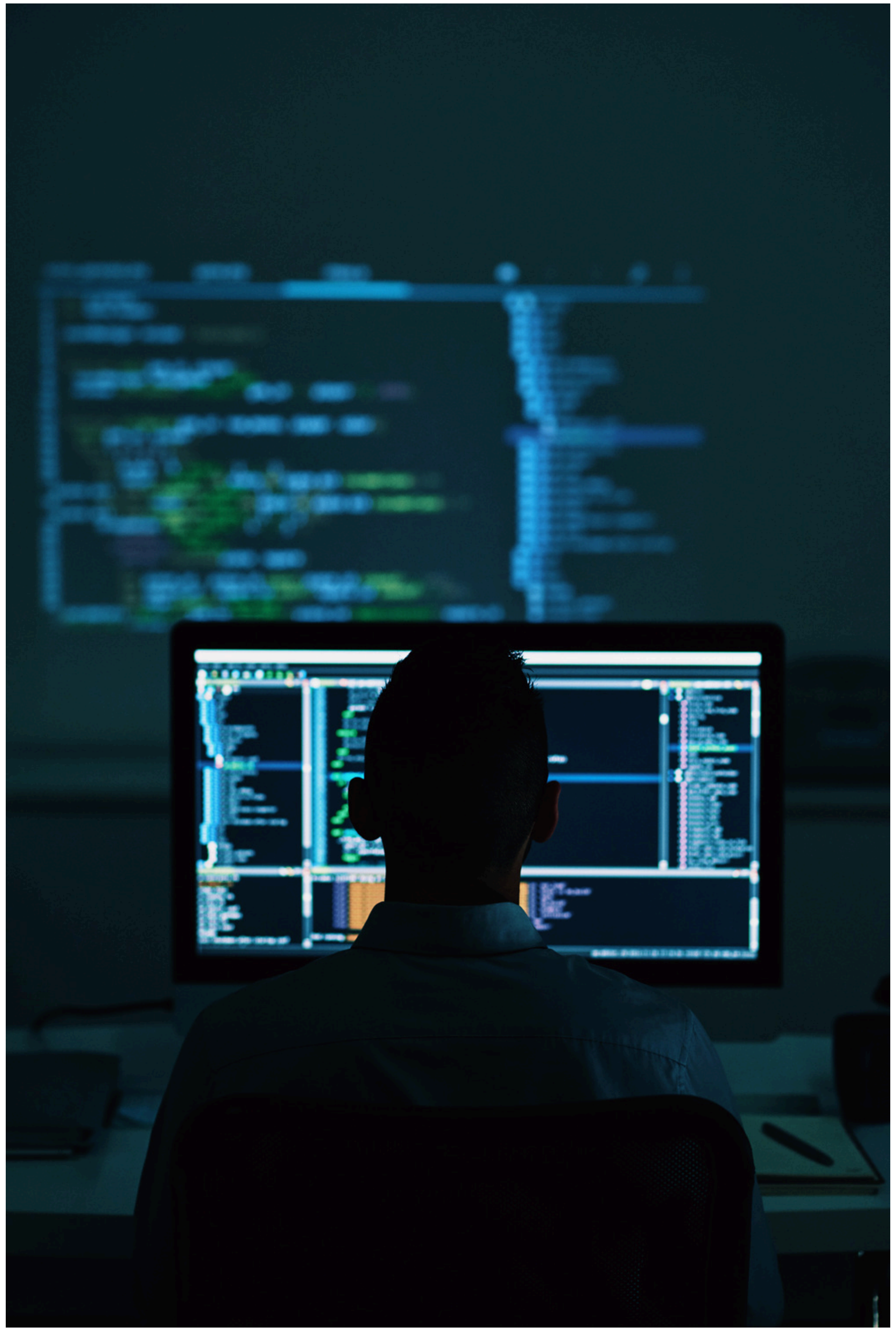
Deployment:

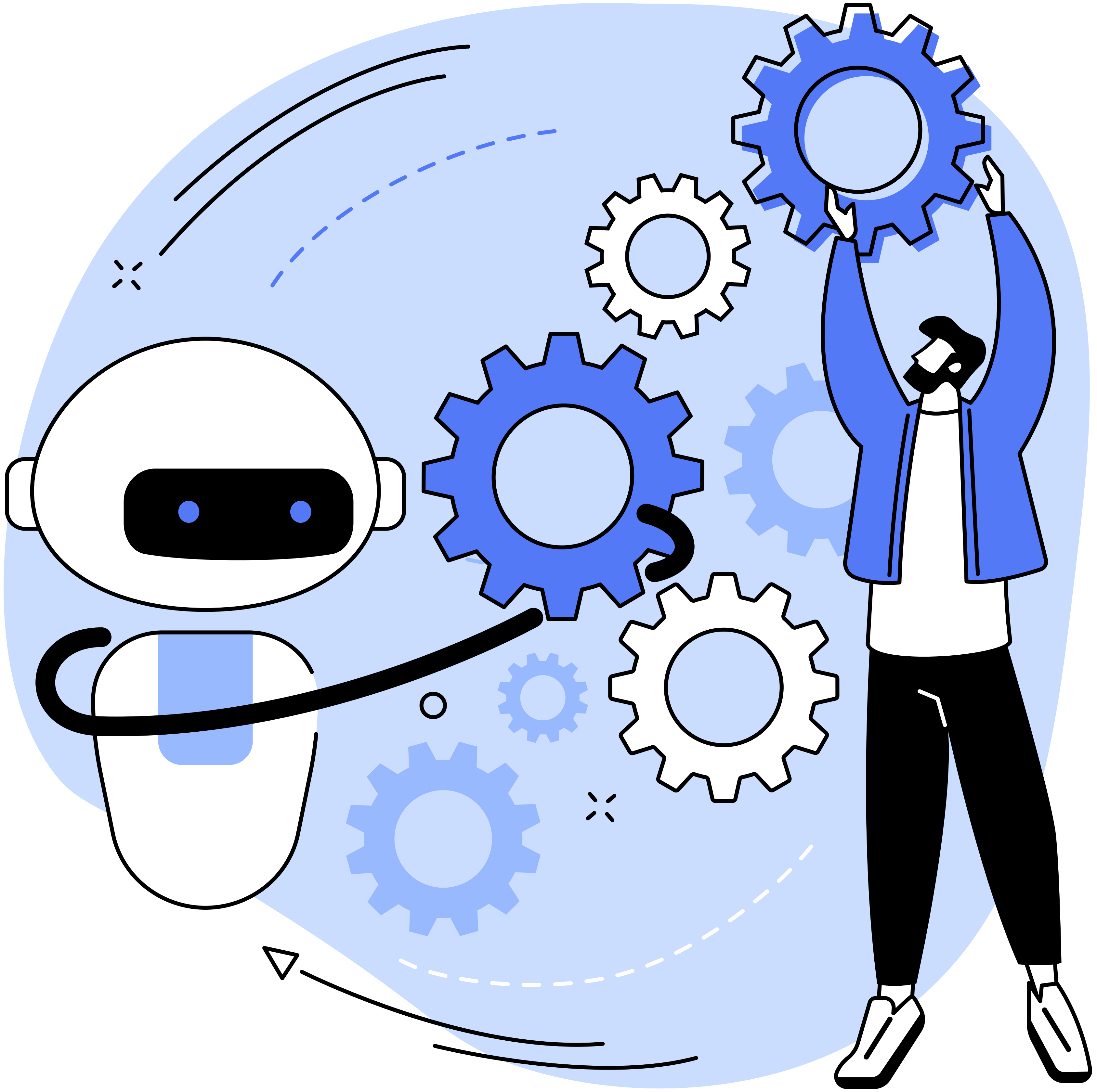
- Docker containers for application deployment.



STARTUP DEV ENV

- **Power On:**
  - Start NVIDIA Jetson Orin device.
- **Login:**
  - Authenticate with user credentials.
- **Launch VSCode:**
  - Open the development environment.
- **Open Repository:**
  - Access template project.
- **Verify Docker:**
  - Ensure Docker is running
- **Follow instructions** > "startup_dev_env.md"





IT’S YOUR TURN