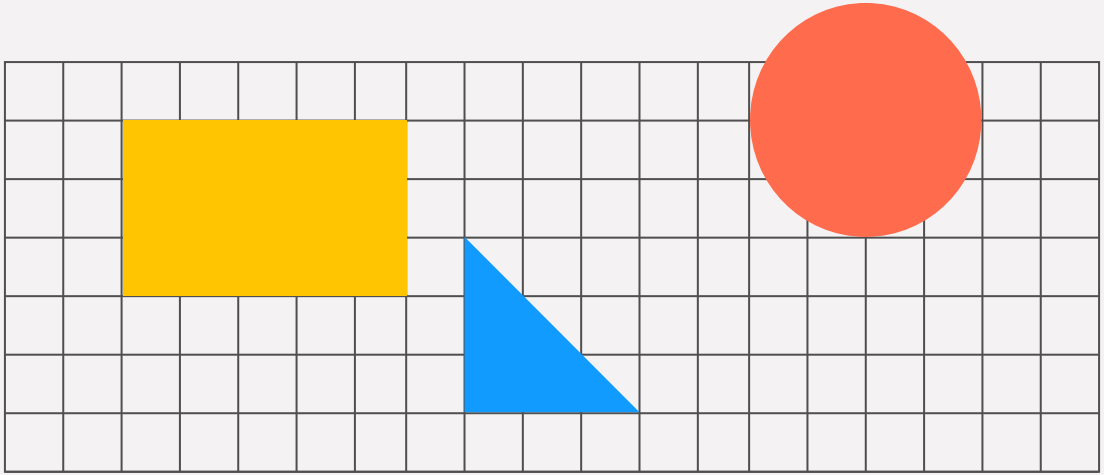
Exam Presentation

IoT-To-Cloud Solution Showcase: From Sensor to Service



Voluntary Choice of One of These Topics (or Sub-Topics) or Home Work Presentation

- **Cloud Fundamentals & Core Concepts**
- **Virtualization, Containers & Serverless**
- **Cloud Storage & Databases**
- **Security, Privacy & Ethics**
- **The IoT Landscape**
- **Connecting IoT to the Cloud**
- **DevOps & Continuous Integration**
- **Processing & Analyzing IoT Data with GCP**

Otherwise...

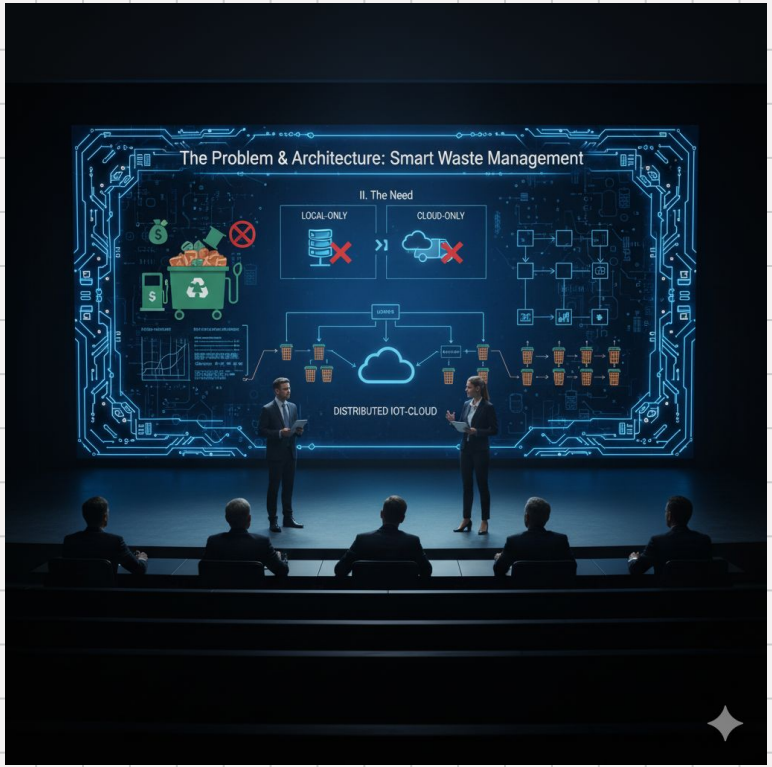
Presentation: Topic and Requirements

- The core assignment is to present a **complete end-to-end solution** for an IoT problem, using the concepts and tools taught in the course:
  - **Topic: Design, Implement, and Present a Cloud-Enabled IoT Solution**
  - Duration: 15 minutes presentation + 5 minutes Q&A.
  - Group Size: Individual or Small Group up to 3 people

Component	Focus Area	Competence Demonstrated
IoT Layer	Data acquisition (simulated or real sensors), Edge/Local processing.	Professional Competence (Technical problem-solving)
Cloud Layer	**"Everything-as-a-Service"** concept, virtualization, cloud infrastructure setup, storage.	Professional Competence (Cloud infrastructure, tools)
Software	Programming logic for data handling, analysis, and application.	Professional Competence (Independent coding)
Presentation	Clear structure, justification of architectural choices, explanation of programming models.	Methodological & Personal Competence (Structured work, oral communication)
Discussion	Defending and justifying the solution approach.	Social Competence (Communication, justification of choices)

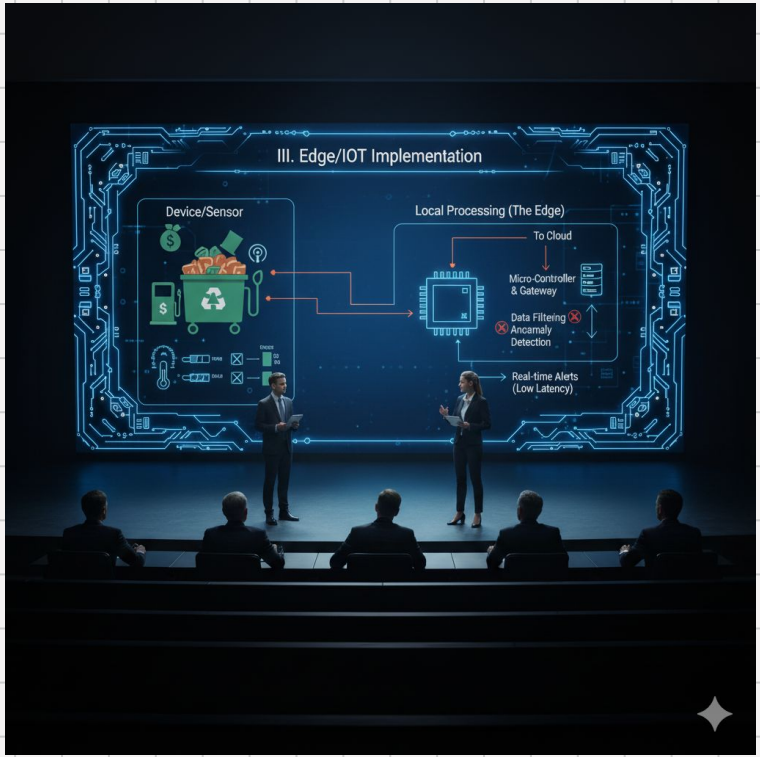
Structure: Problem Definition

- **The Problem**: Clearly define the real-world problem the solution addresses (e.g., smart waste management, predictive maintenance, energy monitoring, etc.)
- **The Need**: Justify why a **distributed IoT and Cloud** approach is necessary (i.e., why local-only or cloud-only won't suffice)
- **Architectural Overview**: Present a high-level diagram of the full solution architecture



Structure: The Edge/IoT

- **Device/Sensor**: Describe the data being collected (the "Thing")
- **Local Processing**: Explain what processing/filtering happens **at the edge** (addressing the need for **low latency** and **bandwidth optimization**)
- **Technology Used**: Mention specific local hardware/programming environment

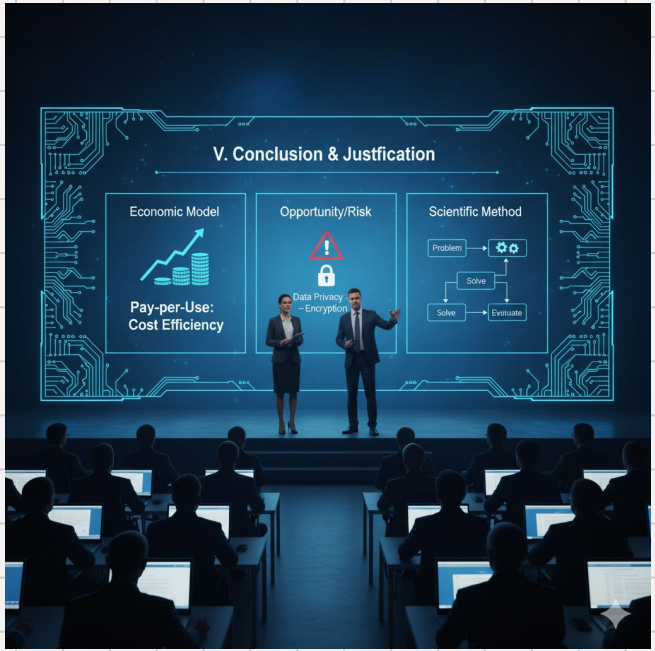


Structure: The Cloud

- **Service Model**: Identify and justify the chosen Cloud Service Model (IaaS, PaaS, or SaaS) or "**X-as-a-Service**" concepts used
- **Cloud Infrastructure**: Demonstrate the use of virtualization or specific cloud tools/services (e.g., an MQTT broker, a serverless function, a containerized application)
- **Programming Model**: Detail the software logic for handling incoming IoT data (e.g., stream processing, event-driven architecture)

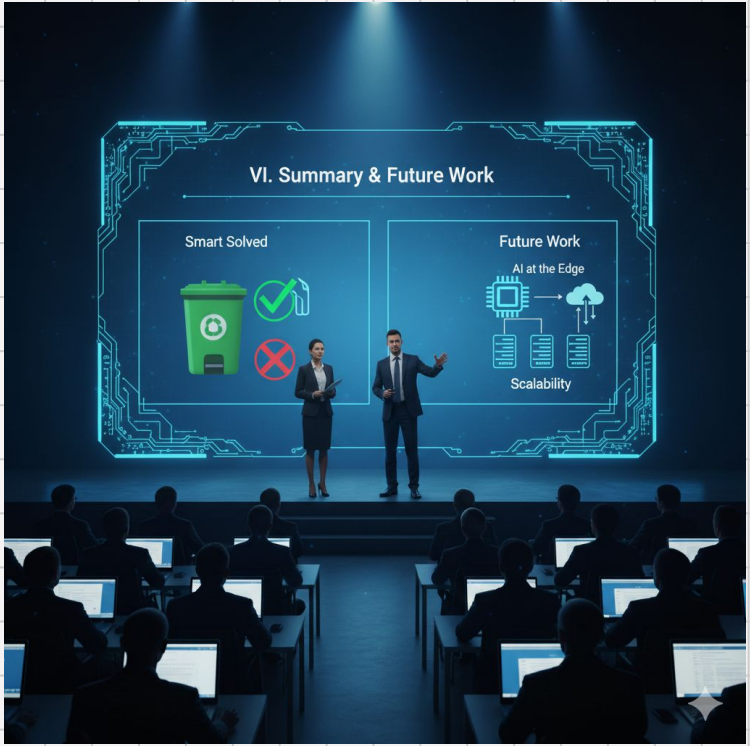
Structure: Economic, Risks, and Scientific Method

- **Economic Consideration**: Briefly analyze the **pay-per-use** model for the utilized cloud services (justifying the economic choice)
- **Opportunity/Risk**: Identify one key security/privacy risk of the solution and how it was mitigated
- **Scientific Working Method**: Briefly explain the steps taken to recognize, formulate, and solve a specific technical challenge during the implementation



Structure: The Future

- **Summary of Achievement**: Recap how the solution solves the initial problem
- **Future Work**: Suggest a practical next step or optimization (e.g., adding machine learning inference at the edge, scaling the infrastructure)



Structure: Time Schedule

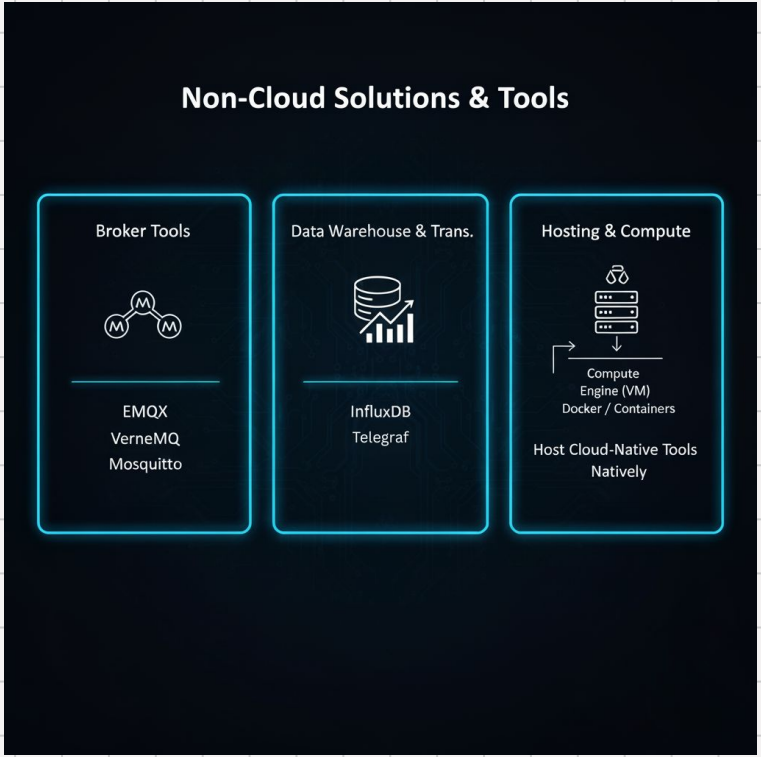
- **Introduction and Problem Definition** (2 minutes)
- **The Edge/IoT Implementation** (4 minutes)
- **The Cloud Implementation** (5 minutes)
- **Economic, Risks, and Scientific Method** (2 minutes)
- **Conclusion and Future Work** (2 minutes)

Recommandations

- **The Goal is a Working Solution**: The project must demonstrate a clear, end-to-end operational flow from your IoT device to the application layer
- **Simplify the Architecture**: Choose the simplest architecture that effectively solves your problem. Avoid adding unnecessary cloud services solely for the sake of complexity or "ticking boxes."
- **Quality of execution** is more important than the quantity of services used
- **Have an eye on your cloud budget**: GCP provides only 300 EUR **free** credits

Tips: Non-Cloud Solutions

- **Broker Tools**: EMQX, VerneMQ, Mosquitto
- **Data Warehouse Tools**: InfluxDB, and Telegraf for data transmission
- **Hosting:** You can spawn a Compute Engine (VM) in the cloud and host cloud-native tools natively or inside containers via Docker/Docker-Compose



Tips: IoT Devices

- **IoT/MQTT Simulator**: Simulator for publishing JSON objects to a broker, simulating sensors and devices (no physical device needed)
- **Zigbee Gateway Device**: USB Dongle to collect data from devices
- **Zigbee IoT Devices:** Different types of sensors, like temperature or humidity

Questions?