Machine Learning for IoT

LAB1: Intro & Getting Started

Valentino Peluso
valentino.peluso@polito.it

What you will learn

- Develop ML pipelines for IoT applications
  - Collect data with sensors
  - Store & process data
  - Train & optimize ML models for IoT data
  - Deploy ML models at the IoT edge
  - Implement edge-to-cloud/cloud-to-edge communication
- Programming language:
  - Python
- Tools/IDEs:
  - Visual Studio Code (local development)
  - Deepnote (cloud development)
  - Redis (cloud storage)

LAB Schedule

Part I

- LAB1: Getting started
  - 9 Oct, 14:30-17:30 – Room 9I
- LAB2: Pre-processing
  - 16 Oct, 14:30-17:30 – Room 9I
- Homework #1:
  - 23 Oct, 14:30-17:30 – Room 9I
  - 30 Oct, 14:30-17:30 – Room 9I
  - **Submission deadline: 26 Nov**

Part II

- LAB3: Training & Deployment
  - 6 Nov, 14:30-17:30 – Room 9I
  - 13 Nov, 14:30-17:30 – Room 9I
- LAB4: Optimization
  - 20 Nov, 14:30-17:30 – Room 9I
- Homework #2
  - 27 Nov, 14:30-17:30 – Room 9I
  - 5 Dec, 8:30-10:00 – Room 1I
  - **Submission deadline: 7 Jan**

Part III

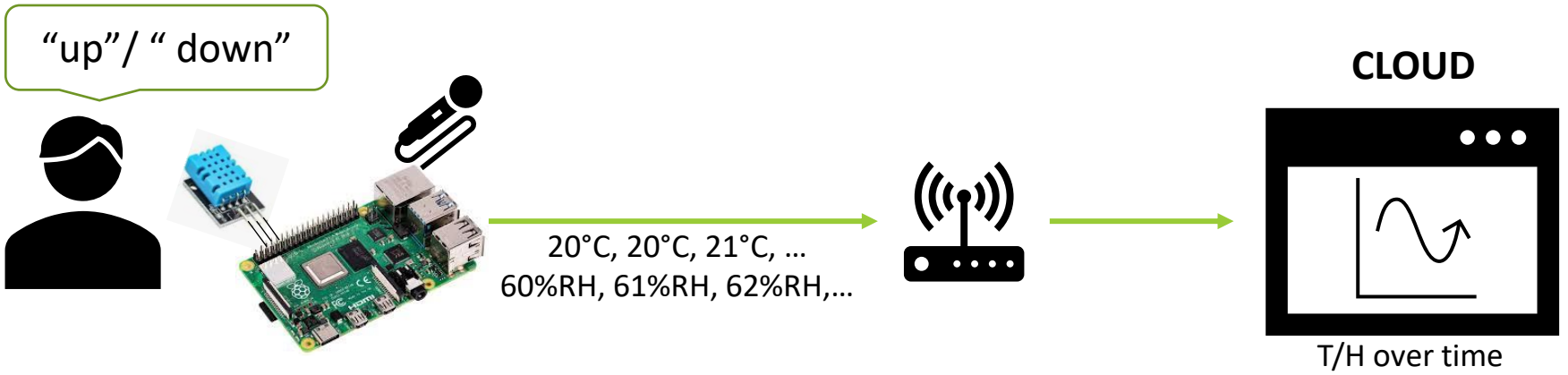
- LAB5: Communication
  - 11 Dec, 14:30-17:30 – Room 9I
- Homework #3
  - 18 Dec, 14:30-17:30 – Room 9I
  - 9 Jan, 8:30-10:00 – Room 1I
  - **Submission deadline: 31 Jan**

LAB Evaluation

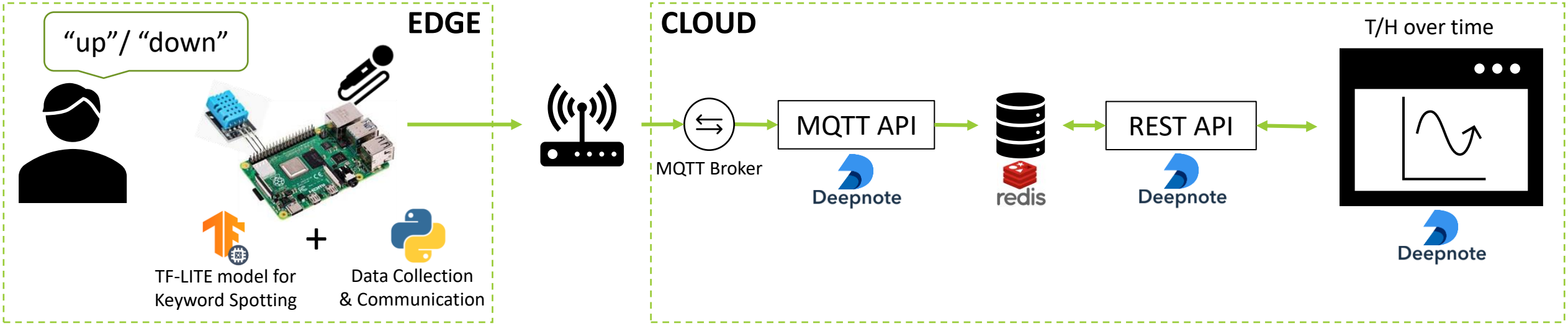
- Homework submissions are mandatory
- Deliverables:
  - Code (Python scripts/notebooks)
  - 1-page PDF report
- Evaluation:
  - 6 points (max.) for each homework
  - 18 (over 32) points total

Use Case: Smart Hygrometer

- Remote monitoring of Temperature/Humidity (T/H)
- The User can start/stop the monitoring using voice commands
- The User can visualize the T/H over time from a Web Application



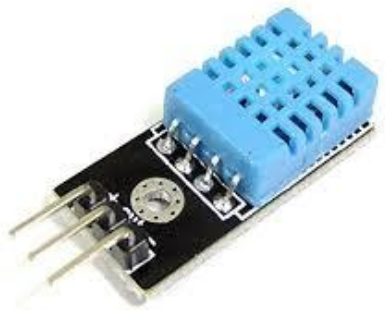
System Architecture



Local Environment: RPI Kit

- Sensing:
  - USB microphone
  - DHT-11 Temperature & Humidity sensor
- Processing: Raspberry Pi 4 Model B 2GB
  - Visual Studio Code
  - Python 3.11
    - TensorFlow 2.13.1

USB Mic.



DHT-11

Raspberry Pi

SD Card

Ethernet Cable

RPI Power Supply

Cloud Environment

- Processing: Deepnote
  - Collaborative Jupyter-like platform for Data Science teams
- Storage: Redis Cloud
  - Key-Value Database
- Communication: Eclipse MQTT Broker
  - Message broker for MQTT protocol

Redis

- NoSQL Database
  - Every record is a key-value pair
    - Key: a string that identifies the record
    - Value: Any kind of data
      - Example:
        - Add a new value

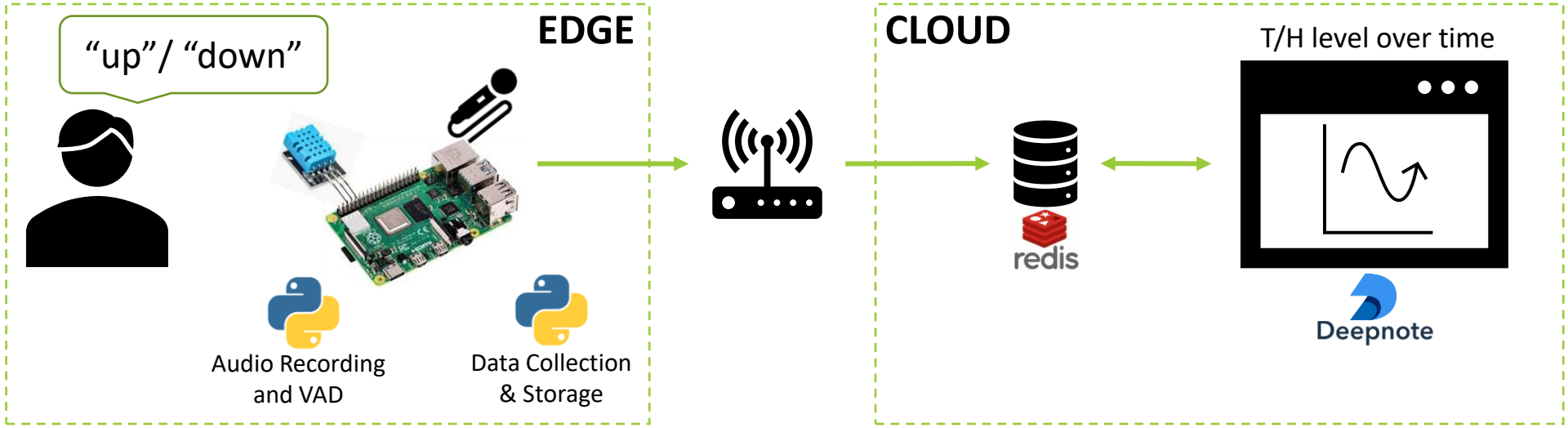
          ```
          SET "message" "Hello World!"
          ```
        - Read a value

          ```
          GET "message"
          ```
- In-Memory: data resides in RAM (instead of disk)
  - Ultra-fast read/write (< 1 ms)
    - good fit for real-time IoT Apps
- Integrates a dedicated data structure for timeseries
  - Redis TimeSeries

Redis TimeSeries

- A Redis TimeSeries is a sequence of records
- Each TimeSeries is associated to a name
  - Example:
    TS.CREATE “temperature”
- Each TimeSeries record is a key-value pair
  - Key: Unix timestamp in milliseconds
    - Unix timestamp = time difference from the midnight of the 1st of January 1970
  - Value: a floating-point number
    - Example:
      TS.ADD “temperature” 1652824475581 25.3
- In-database data processing operations:
  - Compression
  - Filtering
  - Retention
  - Aggregation

LAB1-2: Simplified Architecture



Preparation Status

1. Create a group (deadline Oct. 9th): link
2. Sign up for Deepnote: accept the invitation received in your student inbox
3. Sign up for Redis Cloud: https://redis.com/try-free
4. Download VSCode: https://code.visualstudio.com/download

- Please reply to the poll at this link
- Next Steps:
  - Follow the instructions in **LAB1-Exercises.pdf** (Material → Labs → lab1-getting-started)