MODULE 5: MAKING DOORBELL "SMART" WITH ARTIFICIAL INTELLIGENCE

RING & RUN STEM EVENT.

ORION
OPEN ARCHITECTURE RESILIENT IOT
FOR OPERATIONAL NETWORKS

Making the Doorbell "Smart"...

- How?
  - Using Advanced Artificial Intelligence (AI) Models
    - GPT (Generative pre-trained Transformer) and LLM (Large Language Models) are two advanced and related models for natural language processing, created by OpenAI.
    - Videos:
      - What is an LLM?: https://www.youtube.com/watch?v=zKndCikg3R0
      - Inspirational : https://www.youtube.com/watch?v=Zq710AKC1gg
      - How do they work? : https://www.youtube.com/watch?v=5sLYAQS9sWQ
      - https://www.youtube.com/watch?v=VwN49pC2h_I

  - GPT-4o ("o" for "omni") is a step towards much more natural human-computer interaction—it accepts as input any combination of text, audio, image, and video and generates any combination of text, audio, and image outputs.
    - https://openai.com/index/hello-gpt-4o/

  - You can test the older ChatGPT3.5 (without logging in for free), but visiting this link
    - https://chatgpt.com/

LLM versus GPT …

- A large language model and a general pre-trained transformer both refer to advanced machine learning models based on the transformer architecture. However, they have some differences in their focus and application.

  - **Large Language Model: A large language model, like OpenAI's GPT (Generative Pre-trained Transformer) series, is specifically designed and trained for natural language processing tasks**. These models are trained on vast amounts of text data and are capable of generating human-like text, understanding context, and answering questions. They can be fine-tuned for specific tasks like translation, summarization, or sentiment analysis. Examples of large language models include GPT-3, GPT-4, BERT, and RoBERTa.

  - **General Pre-trained Transformer: A general pre-trained transformer is a more broad term for models based on the transformer architecture. While these models can also be used for natural language processing tasks, they can be applied to a wider range of problems, including computer vision, speech recognition, and reinforcement learning**. These models are pre-trained on large datasets and can be fine-tuned for specific tasks. Examples of general pre-trained transformers include ViT (Vision Transformer) for computer vision tasks and Conformer models for speech recognition tasks. [1]

Source: https://www.brinwilson.com/whats-the-difference-between-a-large-language-model-llm-and-a-general-pre-trained-transformer-gpt/ [1]

OpenAI Overview

- Introduction: https://openai.com/index/chatgpt/

- Common Uses:
  - https://openai.com/
  - https://openai.com/index/introducing-chatgpt-edu/

  - The way we do life in future will be different:
    - https://copilot.microsoft.com/
      - https://www.cnet.com/tech/services-and-software/microsoft-copilot-embraces-the-power-of-openais-new-gpt-4-o/

    Challenge Question:
    - Can anyone think ways that AI will impact our lives?
   - Several of the image slides in this presentation are AI generated. Can you identify them?

OpenAI Overview cont/

- Platform Overview
  - https://openai.com/api/
- OpenAI Models availability:
  - https://platform.openai.com/docs/models
- It's not free, but is cost effective
  - https://openai.com/api/pricing/
- Fine-tuning models
  - https://www.spaceo.ai/blog/how-to-train-openai-gpt-models/
- Custom models
  - https://openai.com/form/custom-models/

Interacting with ChatGPT?

- Two primary products:
  - Both use separate logins:
    1. ChatGPT Login
    2. API Login

  - **Chatbot (Virtual assistant)** experience

  - Programmatically, through what is referred to as **an "application" programming interface (API)**
    - In our case the *"application"* is the Doorbell project
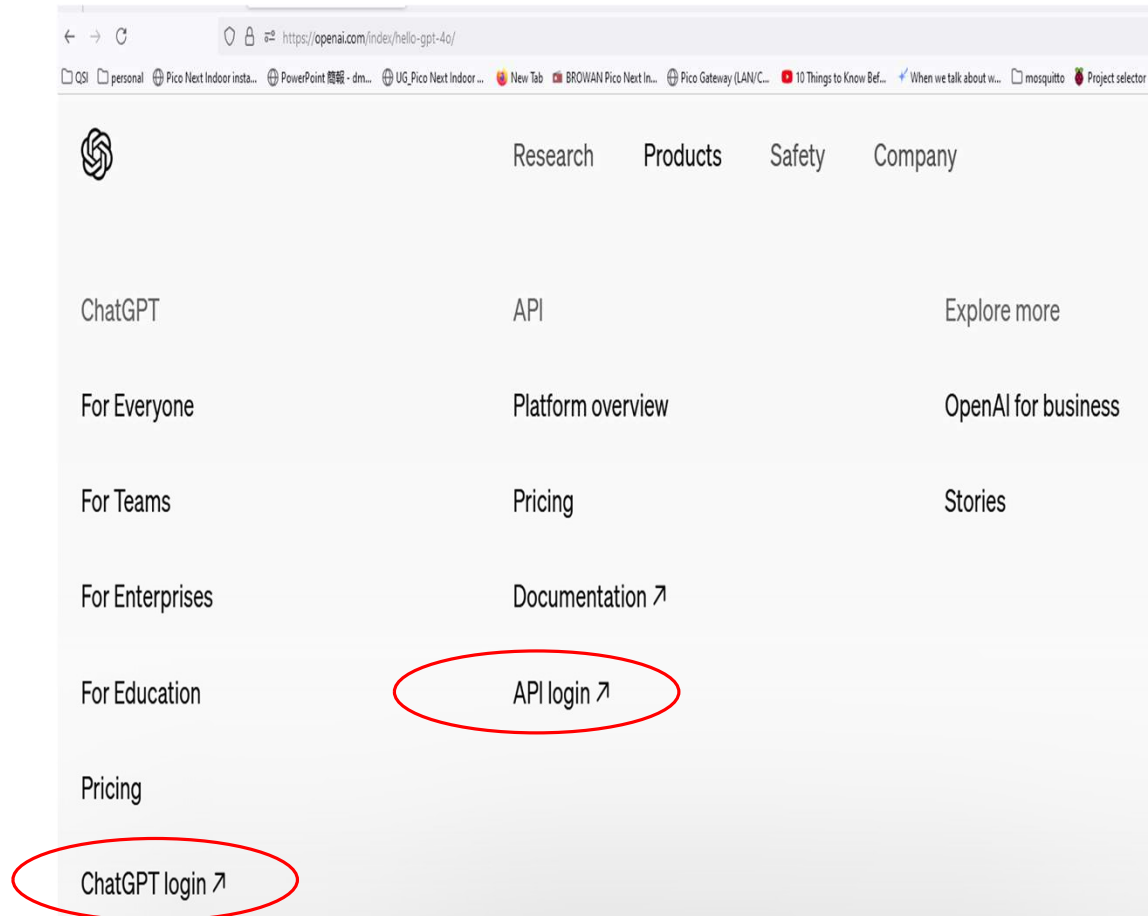
How did you say I integrate OpenAI models into the Doorbell Design?

- Answer:
  - Really easily, using the API approach!

  - Step 1: Click the "API Login", create and account and sign in

  - Step 2: Once you're signed in you can create an organization, and new projects and add users to projects within different roles
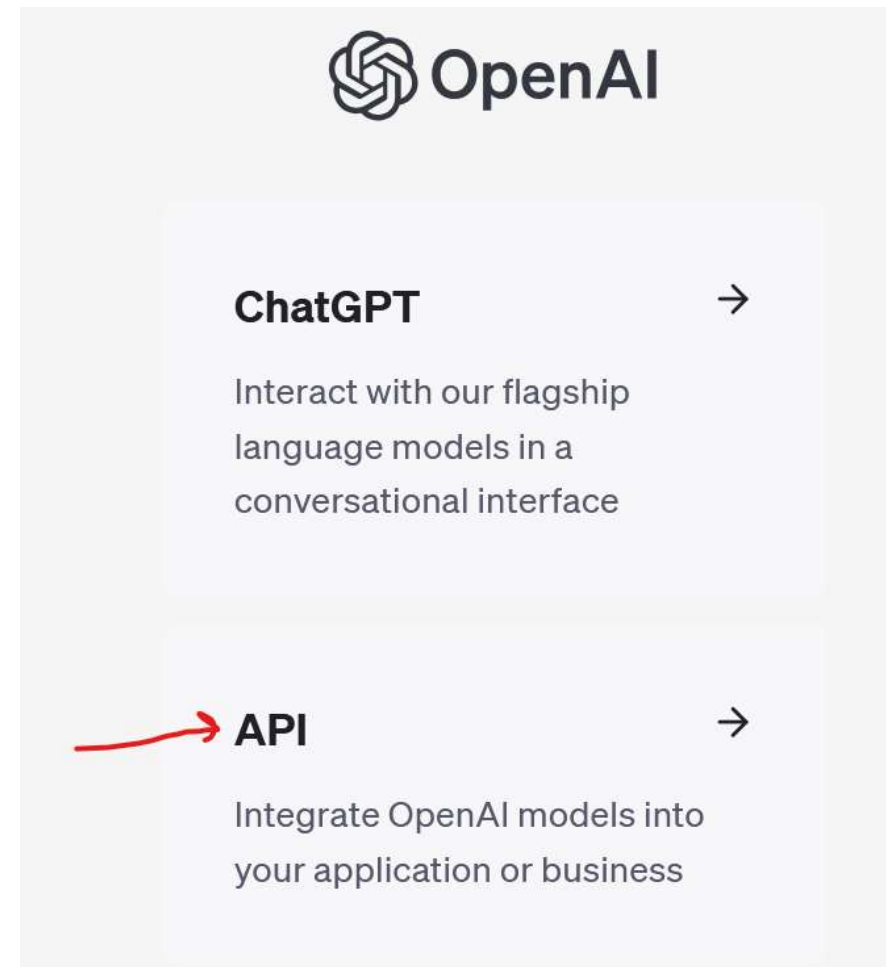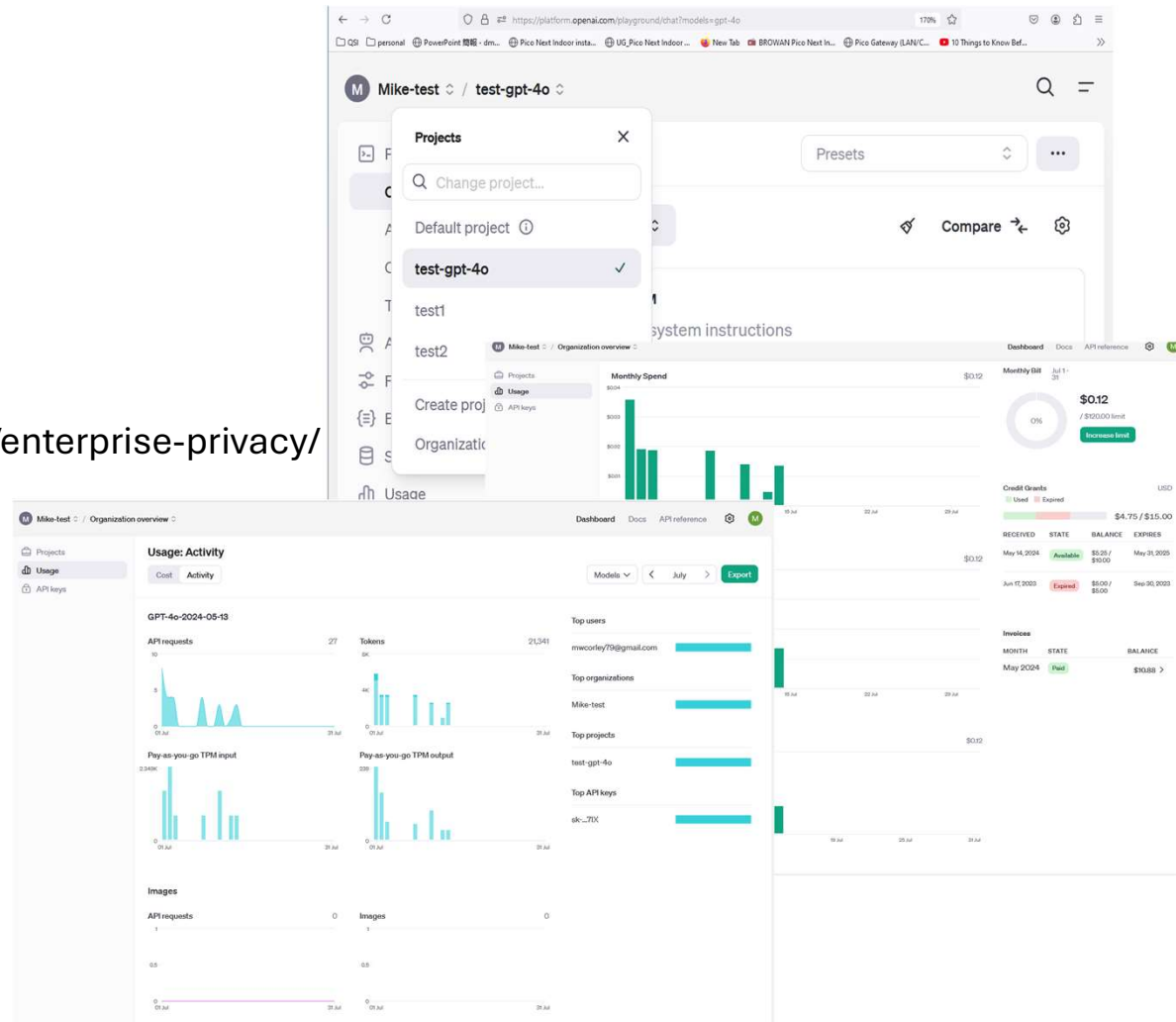
How did you say I integrate OpenAI models into the Doorbell Design?

- Projects enable you to organize, select, and fine-tune models for consumption, and create API keys for secure programmatic access to services
  - *Note: this how we will integrate the "Omni" model for the Doorbell*

  https://openai.com/enterprise-privacy/

- At the organization level
  - Monitor/control model usage (API requests (for inputs audio, images, text),
  - Define/add projects, assign users etc.

- Manage Costs and billing
  - API uses Token based approach for mapping resource consumption to billing
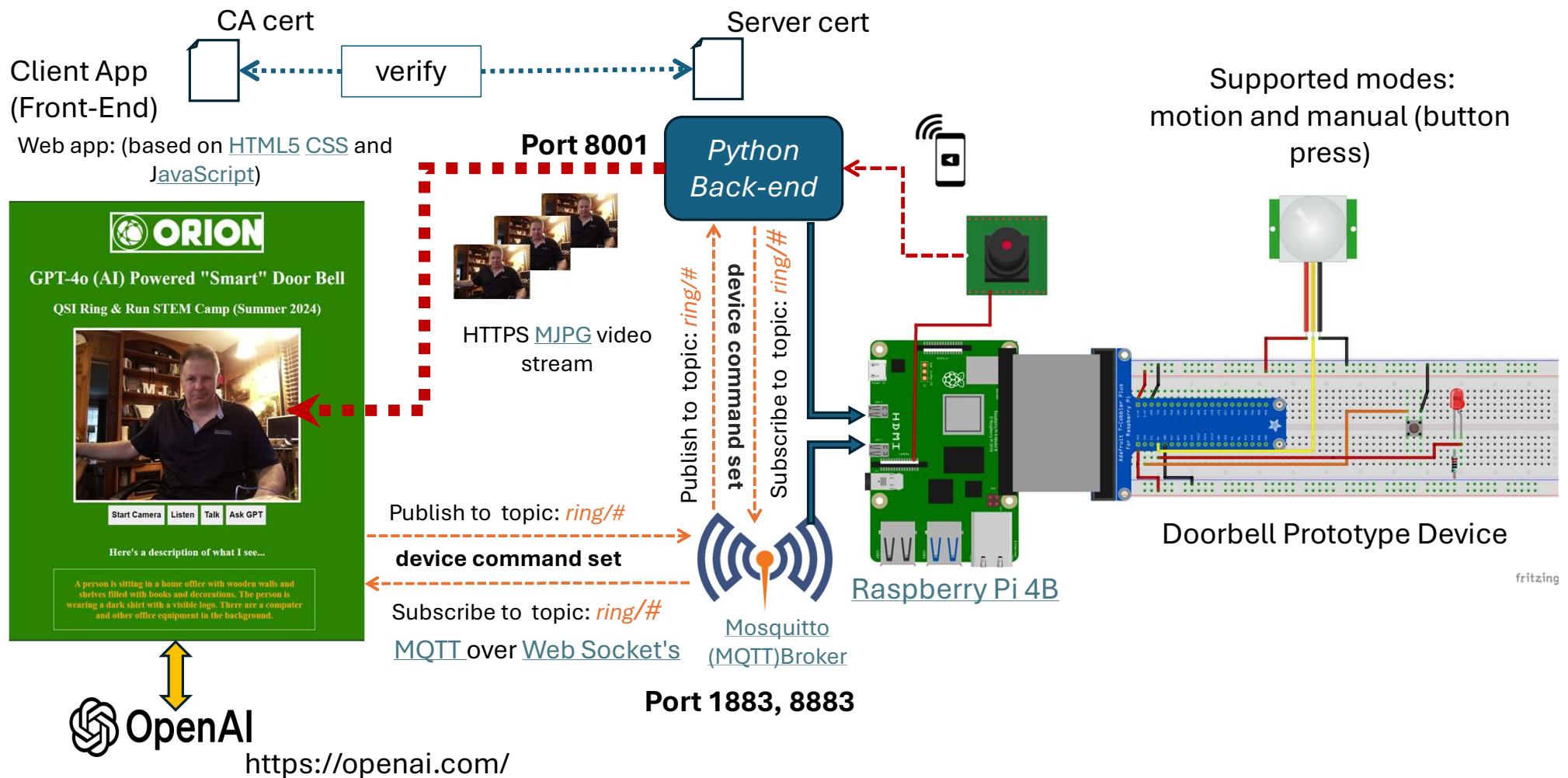
Risks? What?

- Privacy
  - Your data…
    - https://openai.com/enterprise-privacy/
    - https://openai.com/policies/usage-policies
  - Zero Data Retention
    - https://community.openai.com/t/zero-data-retention-information/702540
- Getting Wrong Answers…
  - https://www.linkedin.com/pulse/does-chatgpt-give-correct-answers-explained-jon-rishworth-q8qze
- Safety and Ethics
  - https://www.kaspersky.com/resource-center/preemptive-safety/is-chatgpt-safe
  - https://medium.com/@gargg/guiding-the-future-of-ai-an-inside-look-at-openais-ethical-policies-3503f17b6b0b

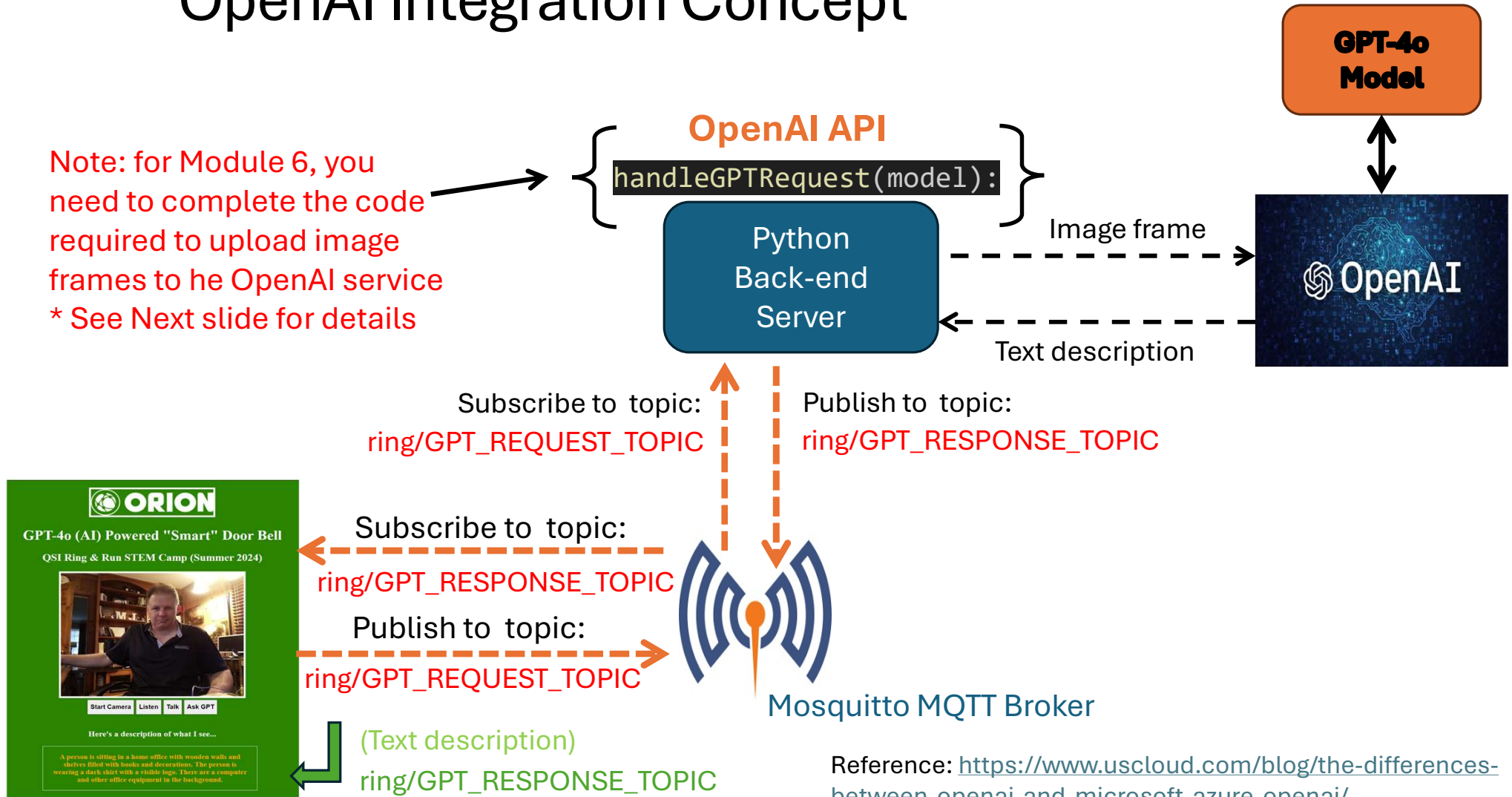# Integrating OpenAI models into the Doorbell Project: Synopsis

- The (back-end) server uses the "picamera" package to stream camera imagery (defaulted 24 frames per second) to the font-end (web application).
  - Note: you can view *StreamingServer* Code in *ring_server.py (*lines 28-142)

- When someone presses the doorbell button (when set in "manual" mode), and when motion is detected (in "motion" mode), the camera will start recording and streaming captured camera frames to the front-end app.
  - The user of the client (web) app can stop the streaming by pressing "Stop camera" and resume steaming by pressing "Start camera" to manually initiate streaming.
    - Note: this works because the client app connects and dispatches corresponding MQTT control message/topics to the Broker, which in turn forwards message (using pub/sub semantics) back-end (subscribed) server for processing

- When the user clicks the "Ask GPT" button (with the camera streaming), an MQTT message is dispatched to signal the server, which subsequently receives the request, captures image frame and dispatches the image to OpenAI for to be processed by the model specified in the API invocation.

- The OpenAI model (GPT-4o) will process the input (image frames) and return intelligent (human like) description of what it sees in the image, thus making the Doorbell "Smart"

IoT enabled "Smart" Doorbell Concept

CA cert

Server cert

verify

Client App (Front-End)

Web app: (based on HTML5 CSS and JavaScript)

Port 8001

Python Back-end

Supported modes: motion and manual (button press)

HTTPS MJPG video stream

Publish to topic: *ring/#*

device command set

Subscribe to topic: *ring/#*

ORION

GPT-4o (AI) Powered "Smart" Door Bell

QSI Ring & Run STEM Camp (Summer 2024)

Start Camera | Listen | Talk | Ask GPT

Here's a description of what I see...

A person is sitting in a home office with wooden walls and shelves filled with books and decorations. The person is wearing a dark shirt with a visible logo. There are a computer and other office equipment in the background.

Publish to topic: *ring/#*

device command set

Subscribe to topic: *ring/#*

MQTT over Web Socket's

Mosquitto (MQTT)Broker

Port 1883, 8883

Raspberry Pi 4B

Doorbell Prototype Device

OpenAI

https://openai.com/

fritzing

OpenAI Integration Concept

Integrate OpenAI (flagship) model - GPT-4o ("omni") into the Doorbell Project

1. Open a terminal in the Remote VSCode session
   - Install the officiate OpenAI python package: *pip install openai*

2. **Challenge Exercise**:
   - Study the *ring_server.py* server code function: "***handleGPTRequest()***" beginning on line 225
   - Study the function "***Open_AI_Tell_Me_Who_Is_There(base64_image)***" beginning on line 141
   - Study the "**Getting Started, vision**" guide given in the references below: "***Uploading base64 encoded images***"
     - Study these resources very carefully, and try to complete the function:
       "***Open_AI_Tell_Me_Who_Is_There(base64_image)***"
       - *Note:* **completing this function correctly is all that is required to successfully integrate the OpenAI service and thus , make the Doorbell "Smart"**
         - ***You will need an API KEY to access the GPT-4o service (this will be provided). You <u>do not</u> need to create an account***
   - You **will need to export the API KEY environment variable in VSCode shell terminal** using the following command: *export DOORBELL_KEY="[api-key]"*
     - **On line 144 uncomment API key variable:** `api_key  = os.getenv('DOORBELL_KEY')`

- References:
  - **Getting started guide, vision (study this for the Doorbell project):**
    https://platform.openai.com/docs/guides/vision
    - API reference page: https://platform.openai.com/docs/api-reference/introduction

Completed "Smart" Doorbell
with "Smarter" answers!!