Fog Compting prototype project

Local Setup

- Have a recent python version install, best python3.11
- Install poetry `curl -sSL https://install.python-poetry.org | python3 -`
- Install dependencies `poetry install`
- To run the cloud component locally execute `make run-cloud`
- For sensor device one run `make run-server-one ip = <CLOUD_IP>` where `<CLOUD_IP>` is the IP of the cloud component or `localhost`
- Same goes for `make run-server-two ip = <CLOUD_IP>`
- To build the docker image for the cloud run `make build-local`

Architecture

This system consists of two virtual sensor IOT components that generate temperature data between 20 and 40 degrees and send it to a cloud component. Additionally the IOT components generate humidity data. The data is sent in the format `{temperature: x, humidity: y}` The cloud component computes the average temperature of the two sensors and sends an action to the IOT components. The action is `red` if the average temperature is above 30 degrees and `green` if it is below 30 degrees. The IOT components then change their LED color to the received action. The action format is `{action: "red"}` or `{action: "green"}`. The IOT components and the cloud component communicate using the ZeroMQ library.

Cloud Server

The cloud server communicates with the IOT devices as a ZeroMQ Router. It calculates the actions every 15 seconds using an extra thread. The actions are sent to two predefined IOT devices `sensor-1` and `sensor-2`. If a sensor cannot be reached the messages are stored in a queue which gets flushed once the sensor is reachable again.

Sensor Device

The sensor devices communicate with the cloud component as ZeroMQ Dealers. They generate data every 5 seconds in an extra thread. If they can't reach the cloud component the data gets stored in a queue which gets flushed once the cloud component is reachable again. The sensor devices also have a LED which changes color depending on the action received from the cloud component.