

# Fog computing

- Jonas Arnhold (381360)
- Niklas Teschner (398509)

We created an application to receive and handle environmental data from sensor nodes. To demo this functionality we decided to create two virtual sensors. One mocks the temperature of the edge device and the other the usage. The Sensors are coded in Python due to simplicity and minimalism. After building the final docker container of each sensor is roughly 63 MB in size.

After “measurement” the sensors send the data to the edge server. The communication is based on ZeroMQ and the Publisher/Subscriber paradigm. The collected data is stored in a buffer. After storing the edge service reads from the buffer and try to send the data to the cloud service. If the sending fails the buffer keeps the data and the edge service tries to send the data again. This creates fault tolerance as requested. To collect and send at the same time we decided to multithread the tasks and as a result allow a better scaling.

The communication between edge and cloud server is handled by request and response paradigm. After processing the batch data at the cloud server the server sends the aggregated data back. Edge and cloud server are coded in java with a maven build. We choose Java to handle this tasks because in big environments OOP helps a lot building a clean and good usable servers. In Addition Java is very well tested in real environments.

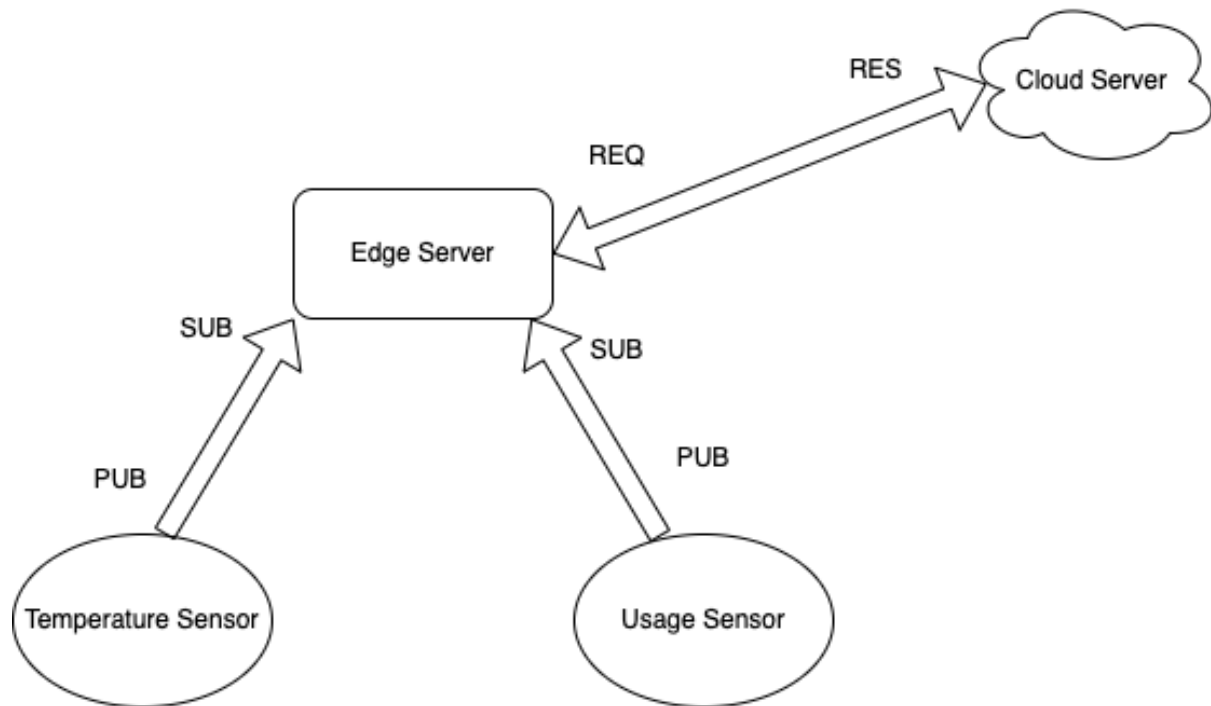
To make the application platform independent we decided to create a docker container for each component. To run the components only a docker distribution is required. All other dependencies and requirements are handled inside the container. This makes the whole system easy to use.

To build the application the first time you have to run `docker compose up - --build`. This command creates all four images and starts four fresh containers. The components can also be built on their own by using the names specified in the docker compose file. After building the first time `docker compose up` is just enough to start the containers.

Due to the modalized structure it's really easy to deploy the components to different environments like a google cloud. But for demo purposes all components are defined in the docker compose and run on one machine.

The adresses of all components are passed to the edge server to connect to cloud and the sensors. The sensors are publishing their data to port 5555 and 5556. While the cloud and egde server communicate over port 8080. In the local environment the ip address is not important. If deployed to a real environment the ip addresses or DNS have to be passed. The docker compose file allows to set the variables for each component and passes them as env Variables into the container.

Architecture overview:



Further information is given in the readme of the repository here:

<https://git.tu-berlin.de/nikkite99/fog-computing/-/tree/main>