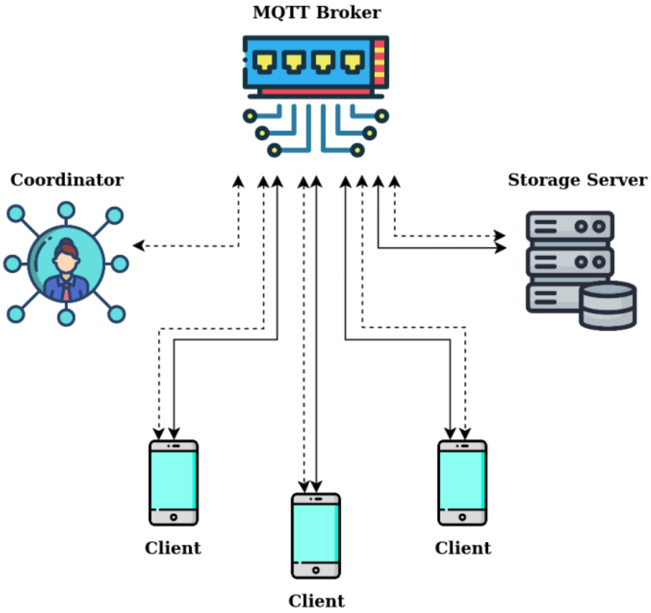
Decentralized Federated Learning over MQTT: Project Description

Introduction

Our DFL over MQTT comprises the following nodes:

- ❑ Broker
- ❑ Client
- ❑ Coordinator
- ❑ Storage Server



Note: All the nodes above inherit the MQTTFC_Client class for operation supervision through MQTT Fleet Control Dashboard.

Broker

Broker here can be any of the open source MQTT broker, or any online available broker. We use Mosquitto by default.

Client

Client node has three jobs: 1) Data collection and Training 2) Aggregation of received model parameters 3) Propagate model parameters to other nodes.

Classes

- Trainer
  - Houses the training loop plus the pre and post processing procedures
  - Houses a copy of the model lended from FL_Client (Ephemeral)
- Aggregator
  - Houses Masking and Unmasking code for secure aggregation
  - Houses Aggregation logic code
  - Houses a copy of the model lended from Trainer (Ephemeral)
- FL_Client::MQTTClient

    - Houses a list of topics for control/ and model propagation
    - Houses MQTT-visible Function to receive model and lend it to Application Logic
    - Houses MQTT-visible Function to receive model updates and lend it to Aggregator
    - Houses MQTT-visible Function to send Model Updates lended from Trainer
    - Houses MQTT-visible Function to send Model lended from Aggregator
- Application Logic
    - Houses functions to use the model Lended from FL_Client
    - Houses functions to lend data from pripheral sources to Trainer

Coordinator

Coordinator is a task scheduler, and task manager. It schedules the model propagation for clients, the initiation of training, and propagation of model updates for units that have finished training.

Coordinator also makes the deicision on which nodes will be the aggregators.

Coordinator keeps track of nodes that have finished training, and those are in the middle of training.

Coordinator Sends commands to Client nodes to perform their next task in terms of aggregation, training, model update, and model parameter propagation.

Coordinator keeps track of the oracle model on the Parameter Server, and sends periodic command to update the model on the P.S as well.

Paramter Server

Parameter Server has two tasks: 1) Broadcast a model to clients 2) Update the oracle model upon receiving updated model parameters.

Note: Once the model is updated on the chosen aggregator and propagated to the clients, the session is terminated on the coordinator, and further training and aggregation will commence through a new session.