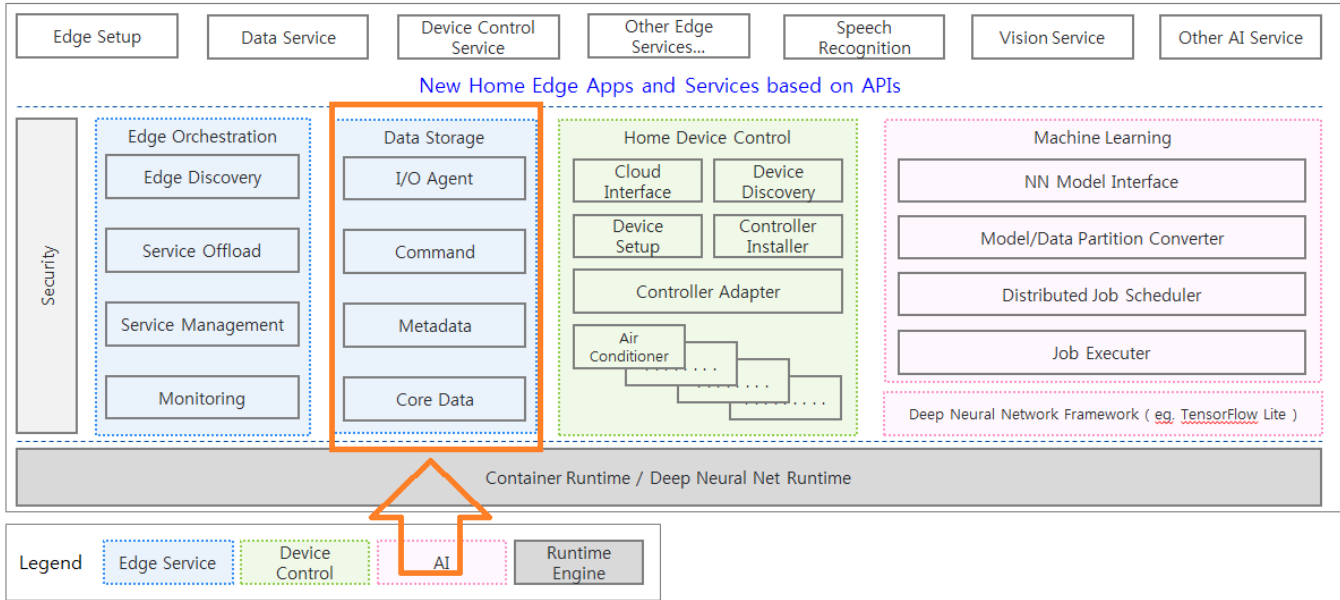


# Data Storage

## Introduction

This module to be developed would be responsible for storing the data (can be sensor data or any reading or image/video data, etc.) from different devices in the home environment. Any device registered to this module can also request for data to different device. The databases used to store the data is Redis DB. At regular intervals this data would also be synced up with the cloud to remove the data from the device and free the device space for new incoming data.



## System description

**Centralized data storage architecture:** A centralized database is stored at a single location such as high computing power device in the home. It is maintained and modified from that location only and usually accessed using REST APIs by the devices connected by LAN/WAN. The centralized database collects the data can be used to train the model and also provide data to the different devices when needed.



The laptop acts as the centralized storage. All the devices are connected to each other using Wi-Fi. Camera sends the data to the database using the REST API. Similarly other devices also send their data to the central storage. Devices can also ask for data of other devices like camera ask for sensor data. For this the devices needs to call the API using the camera device id. At regular intervals the data stored are also synced with the cloud infrastructure and thereby deleted from the data controller.

The devices would be categorized into categories like the devices which send the data frequently for example CCTV camera and devices which send data less frequently for example thermostat devices. The time interval at which the data will be send to the data controller will be decided upon the frequency nature of the device to send the readings. For example, in the devices which send data frequently, the interval can be in seconds say 20s, while for the other category it can be set to 1 minute. Also the device capacity to store the data in the database locally is also considered while deciding this interval.

Similarly while sending the data if the data controller is offline or not available, then the data is stored locally at the device and it would try again after every 1 min for 10 minutes to check if the data controller has come online. Once the locally stored data is synced with the data controller, device deletes the data and frees its local space.

After syncing to the cloud the data is deleted from the data controller. In the cases where a device queries for data which is not available in the data controller, in such scenario the data controller will respond with 404 error or with the URL of the cloud where the data is synced to the cloud.

## Limitations and assumptions

The current development assumes the data storage module to have one single Data Storage in a home, Centralized Storage Architecture. All the devices will register to the Data Storage's metadata and provide their info. Then the devices can send the data to data controller or also query the data controller for a data specific to a device or time interval.

The following architecture assumes that data stored at data controller would be uploaded to the cloud at some scheduled intervals to free up the space at the device level. Hence the Data Storage device will be dependent on the cloud connectivity at those intervals.

## Code availability

Data Storage is available as a new feature of Coconut release (Oct. 2020) from <https://github.com/lf-edge/edge-home-orchestration-go/>.