Cloud Sync.

1. Introduction

Home Edge will provide a way to store device readings and service information for all devices to be synchronized with an external cloud. This data can be later used for analytics.

2. What all data needs to be synchronized to the cloud?

- Device Readings – These are sensor readings that can be later used for running analytics.
- Service Info (Service DB) – This can be later used to understand of which device was running with what services that could have been used while offloading.
- Optional
  - Network Info (Network DB) – This can be offloaded in cases where the user needs to understand the P2P network state while services were running on a device. This info is dynamic and usually just used for score computation so it might not be required in most cases.
  - Configuration Info (Configuration DB) – This can be offloaded in cases where the user needs system configuration (Mainly Execution Type) during running analytics. This info also might not be necessary in all cases.
  - Resource Info (Resource DB) – This DB also can be offloaded only when user needs info like CPU count, CPU Frequency etc.

All this data is stored in the DB in form of key-value pairs and thus can be exported as a JSON to the cloud. Similarly all readings are given as JSON and thus they may also be stored as JSON in cloud.

3. How to store data to the cloud?

Home Edge uses EdgeX Foundry (EdgeX) for local storage of data. EdgeX also provides exporters (called application services) for a variety of cloud services and applications. These can be used by Home Edge to interact with the cloud. Most data is either stored as XML or JSON. As all the DBs are in JSON format, Home Edge can provide Value Descriptors for the DBs and All systems in the Home Edge Network can be registered as Service Devices which can then store data of their DBs using EdgeX using Value Descriptors as the schema for the data.

EdgeX has provided an Example where EdgeX data is shared to HiveMQ server using MQTT.

This can be seen at: https://docs.edgexfoundry.org/1.3/getting-started/quick-start/#exporting-data

4. When should Cloud Sync happen?

- Device related data – Meta data about device – Config, Network and Service data. All these are related to devices in system and hence the synchronization with cloud can happen when onboarding of the device happens. Home Edge DB has these values.
- Device Data – Sensor generated data. This needs to be synch as and when data is generated or in fixed frequencies. EdgeX (Coredata) has the data.

5. Use case

- Device Usage understanding: Complex ML models can be run at cloud on the device data and make inference/predictions;
- Scene creation: Turning on the ambient light as a user enters from outside. Changing the state of Device B based on Device A. For this device data is needed;
- Service Execution Analytics: When the service info is synch to cloud, this can help in understanding the services and type of platforms on which these services execute. If person identification runs effectively in Raspberry Pi (RPi) board rather than an Android device, it is good to offload to RPi in future during local execution.

Device data and service info are the major data to be synchronized with cloud.

6. EdgeX – Application Service

Application services are based on the idea of "Functon Pipeline". Trigger begins the function execution.
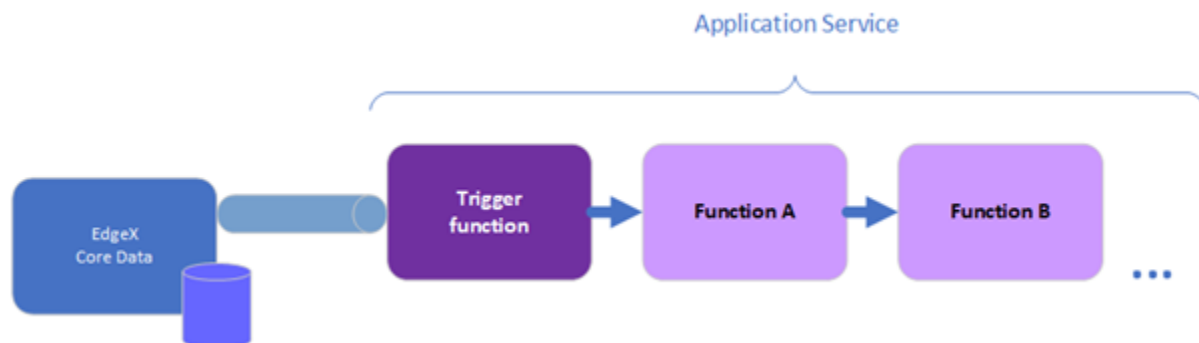
EdgeX provides App SDK to create Application Services. It is provided as Golang module. Four primary are included in the App SDK. The first is the DeviceNameFilter function which will remove events that do not match the specified IDs and will cease execution of the pipeline if no event matches. The second is the ValueDescriptorFilter which exhibits the same behaviour as DeviceNameFilter except filtering event readings on Value Descriptor instead of DeviceID. The third and fourth provided functions in the SDK transform the data received to either XML or JSON by calling XMLTransform or JSONTransform.

Application Service

```
[Trigger] → [Device Name Filter Function] → [Value Descriptor Filter Function] → [XML Transform Function]
                                                                              or
                                                                               → [JSON Transform Function]
```