

ARCHITECTURE FOR AN EXTENSIBLE INTERNET OF THINGS(IOT) FRAMEWORK

OVERVIEW

A service provider(microservices) for IOT is a modular structure which provides plug-n-play architecture where new support and features can be added on the fly without changing the framework.

Microservices architecture enables both the edge and the cloud to deploy a service where appropriate and to migrate a service from one to the other as the system involves.

The client can query any of the registered service providers and get reliable information as well as monitor different devices.

TECHNICAL SPECIFICATIONS

1.Redis for creating service registry

The service registry is a key feature. It helps in managing all the services.

They help in:

- ☐ Registration and Deregistration of services on the fly.
- ☐ Monitoring the health of microservices
- ☐ Load balancing

2.REST API

REST API's are used for all sorts of communications.

- ❑ **Service to Service Communication** : Service might need to connect to other services these communication happens through REST API's
- ❑ **Service to Service Registry Communication** : Service Registry handles requests from the client and redirect it to the specific microservice through REST API'S
- ❑ **Virtual Device** : For development purposes The device is also mimicked using a REST API.

3. Microservices using FLASK framework

Microservices will be built on The flask framework,docker will be used to containerize the services, devices can register themselves to these services and they will be controlled and monitored using them.For implementation purposes we will be using two services namely:

- ❑ **Get inventory microservice-** Individual IOT device has a unique identification number (ID) with other device specifications like model number, batch number, etc. these information can be fetched using this service.
- ❑ **Get monitoring microservice-** The health of the device like the current state, Packer Error Rate(PER), Sensitivity, last updated location is fetched using this microservice,these information can be used to monitor and also to troubleshoot the errors.

4. Docker

- ❑ Docker is used to containerize each microservice. To deploy each microservices to their own VM instances.
- ❑ With Docker, it's possible to reduce performance overhead and deploy many microservices on the same server since Docker containers require a lot fewer computing resources than virtual machines.

5.Docker Swarm

Docker swarm is used to orchestrate the microservices, It make sures that the services never go down by using multiple hosts to deploy docker containers.docker swarm can help in bringing up,bringing down and monitoring the required number of the microservice instances.

