

HashCode 2020

Team Name: Baby Yodas

University: PES University, RR Campus



Problem Statement and Solution

Problem Statement Number – PS4 Geofence

Problem –

We aim to create a boundary within which we can monitor any device. Once this boundary is set up, the product/app can setup triggers, which can send a notification when an entity enters (or exits) the fence. Geofencing enables the creation of unique user experiences that are specific to the context of the location.

Proposed Solution –

For a centralized solution, we intend to offer the master node with selection of an origin (in terms of latitude, longitude) and radius of the fence. Once this is created, any device entering the area within the fence will be shown in the central dashboard and can be monitored. All the known devices will be periodically checked for their position to monitor if they are in/out of the fence.

For decentralized solution, we consider the convex hull of all the distributed masters.



Feasibility

1. The target audience can be – a) **Proximity Based marketing** – When a potential customer is near a particular store, special offers can be sent to them to win them.
b) **Asset Tracking**
2. Deployment will only take a few hours of effort if executed efficiently and having knowledge of Cloud is a plus point.
3. The target audience will find a valuable customer experience if implemented successfully. The retail, fleet trackers and industries can be prime candidates for geofencing. Seeing better customer experience at not an heavy expense will lead to adoption of this solution by majority of the prime candidates.



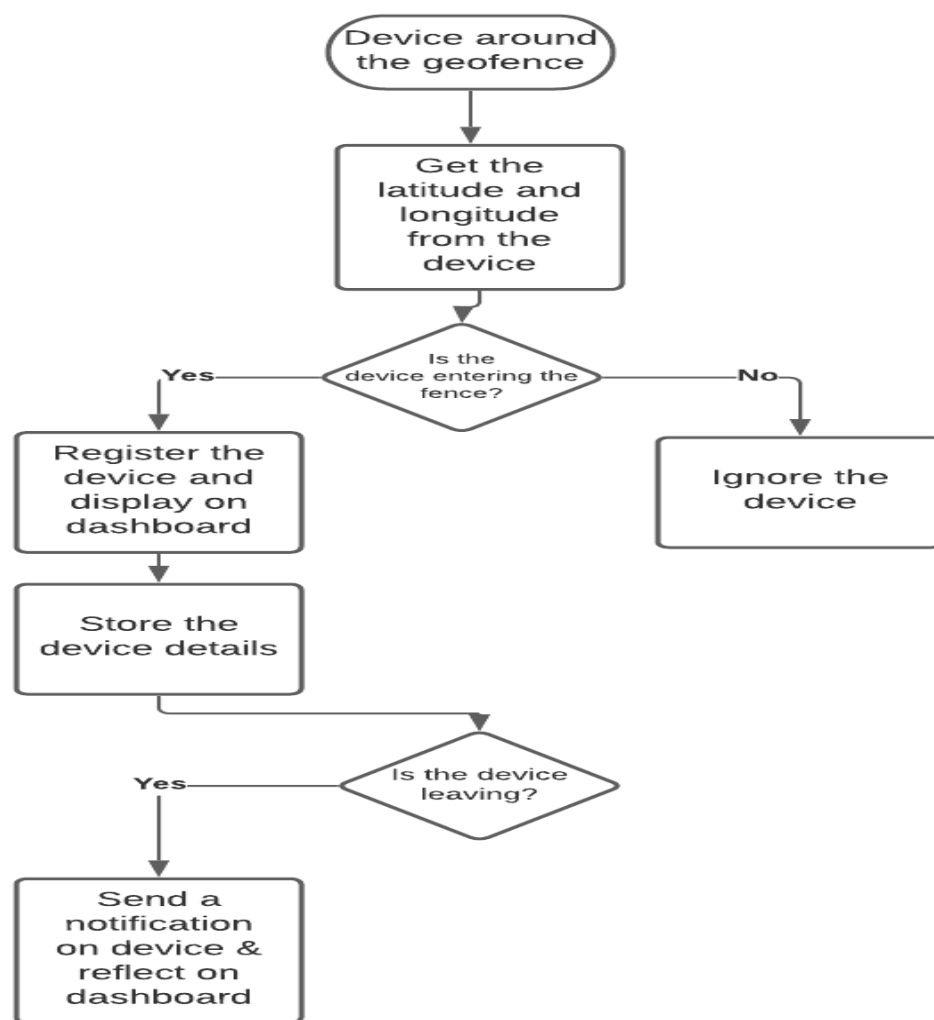
Working Principle

Logic & Solution -

The geofence will be set around a particular master node at a pre fixed radius. The devices around the fence will expose their positions at any instant (latitude, longitude). If the device falls within the defined perimeter, we will be able to monitor the device.

We can extend the solution to form a series of masters. Using similar logic as above, if a device enters the convex hull of the decentralized masters, we will be able to monitor it.

Activity Diagram





Requirements

Hardware Requirements – Nothing in specific because we intend to use Cloud based machines and not actual machines for ease of demonstration

Software Requirements – Python, REST APIs using Python Frameworks, A basic database for storage of device details, Features provided by Cloud Service Providers, cloud service provider like Azure, Google Cloud, AWS and we might use Containers like Docker to simulate devices.



References

AWS - <https://aws.amazon.com/>

Rest APIs - <https://restfulapi.net/>

Docker - <https://www.docker.com/>