CMPE 296A – IoT
Project 2: Smart Door

By Group 4

Agenda

- Introduction
- Market and Business Plan
- High Level Design Architecture
- High Level Specification
- Technology Stack
- Demo

Introduction

- **Smart Door – 'See who's there, from anywhere'.**

- The door to today's home isn't just about keeping the bad guys out. It's also about letting the right people in—your family, friends, and even favorite service providers. With the Smart Door, you are always in control of your front door, no matter where you are, right from your smartphone or computer.

- **How it works** – When the doorbell is pressed, the camera module (can be mounted on peephole or doorbell) gets activated and sends the picture to the registered user android mobile app, from where the user can either unlock the door or deny the permission. Also, when one of the registered user is at the front door, it checks on with the App server and unlocks the door. So, the product also works as a keyless-lock for the registered user.

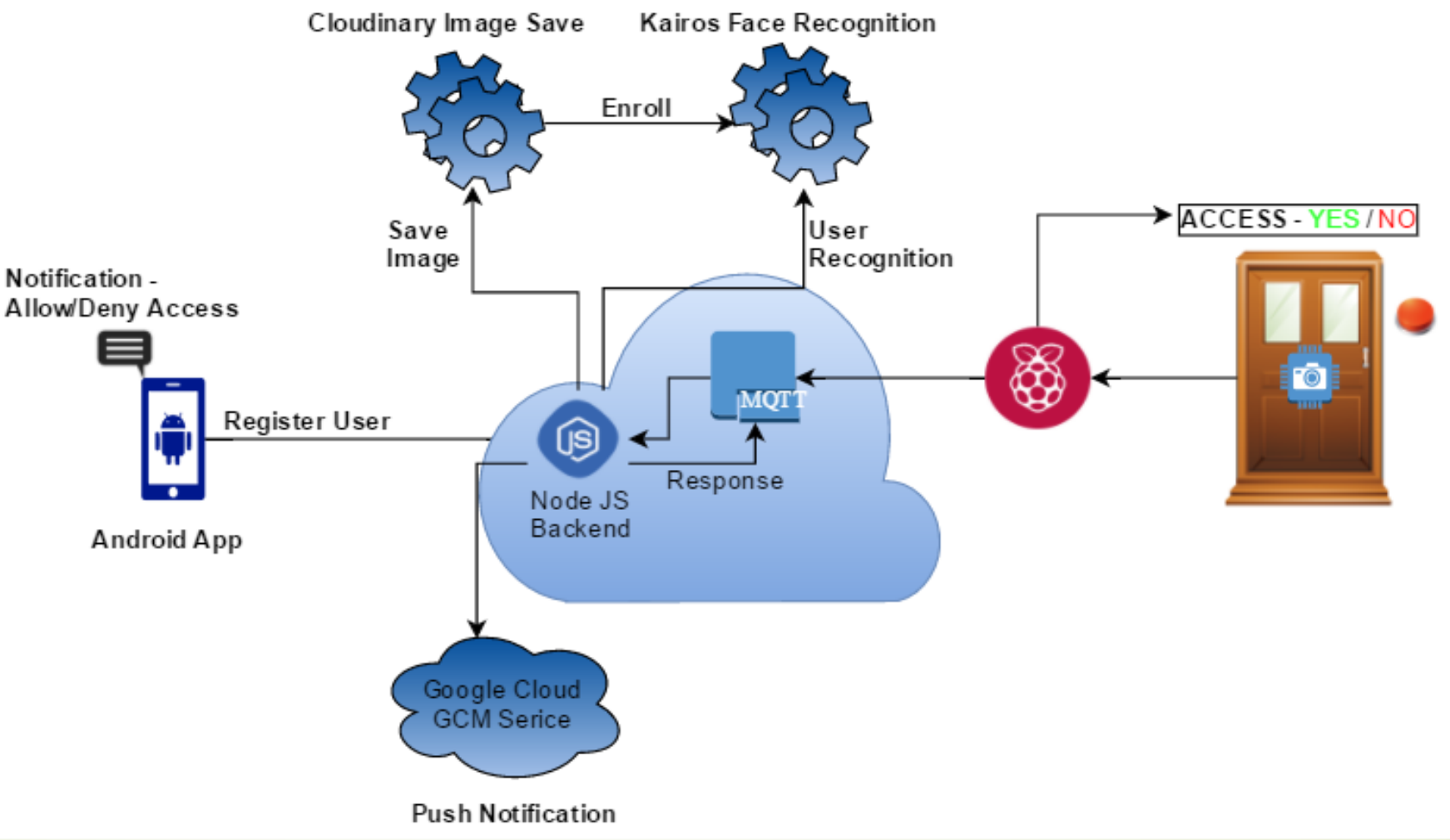
Market and Business Plan

- **Market** – Home Automation.

- **Market Size and Growth rate** - Our product competes in more than one market. It can be a useful feature for a smart door. It can also provide a system integration application for home security systems.

- **Target Market** - Our target customer could be any homeowner or apartment owner who wants to remotely view who's at the main door and be able to remotely control the door.

High Level Design Architecture



High Level Specification

- **Android App**
  - User Enrollment - The user who owns the home needs to enroll first via a mobile app.
  - Notifications - Notifications showing the image of who's at the door and permission to accept or deny it.
- **Device/Raspberry Pi**
  - Captures the image whenever the doorbell is pressed.
  - It can be mounted on peephole or doorbell itself.
- **Cloud**
  - **Mosquitto - MQTT Broker:** The broker acts as a bridge between the NodeJS recognition service and mosquito.
  - **NodeJS Server:** Support for the User Enrollment and User Recognition service.
  - **3rd party Services:** Cloudinary Service API to store the image on cloud and Kairos Service API to enroll and recognize the images.

Technology Stack

- **Hardware -**
  - Raspberry Pi
  - Raspberry Pi Camera Module
  - Push button
  - Arduino with LCD Display (with RFs)
- **Software/Technology -**
  - Mosquitto Broker
  - Python – Pi Code
  - NodeJS – App Server, with following API Services -
    - Cloudinary – Cloud Image Service
    - Kairos – Face Recognition and Analysis
    - GCM – to send notifications to Android App
  - Android Application

Demo