Mobile Health

Mobile Health Ecological Momentary Assessment (EMA) for Real-Time Behavioral Measures

Micah Jenkins
Kily Nhan
Kim Nguyen
Edwin Zheng
Team lead: Erin Mullen

The Problem

Create a system that allows researchers to collect data from study participants in real time

- Data collected from study participants using surveys and will consist of survey results, time started, and time completed
- Researchers must be able to download data as CSV file
- Participants will be able to receive messages and take surveys on a mobile app

User Roles

- Researchers:
  - Uses website
  - Administrators
    - Can delete data, add clinicians, remove/promote clinicians, remove/demote administrators
  - Clinicians
    - Create & manage studies

- Study participants
  - Uses mobile app
  - Take surveys & receive messages

Components

- Mobile app:
  - Cross platform
  - Participants can:
    - take surveys
    - receive messages

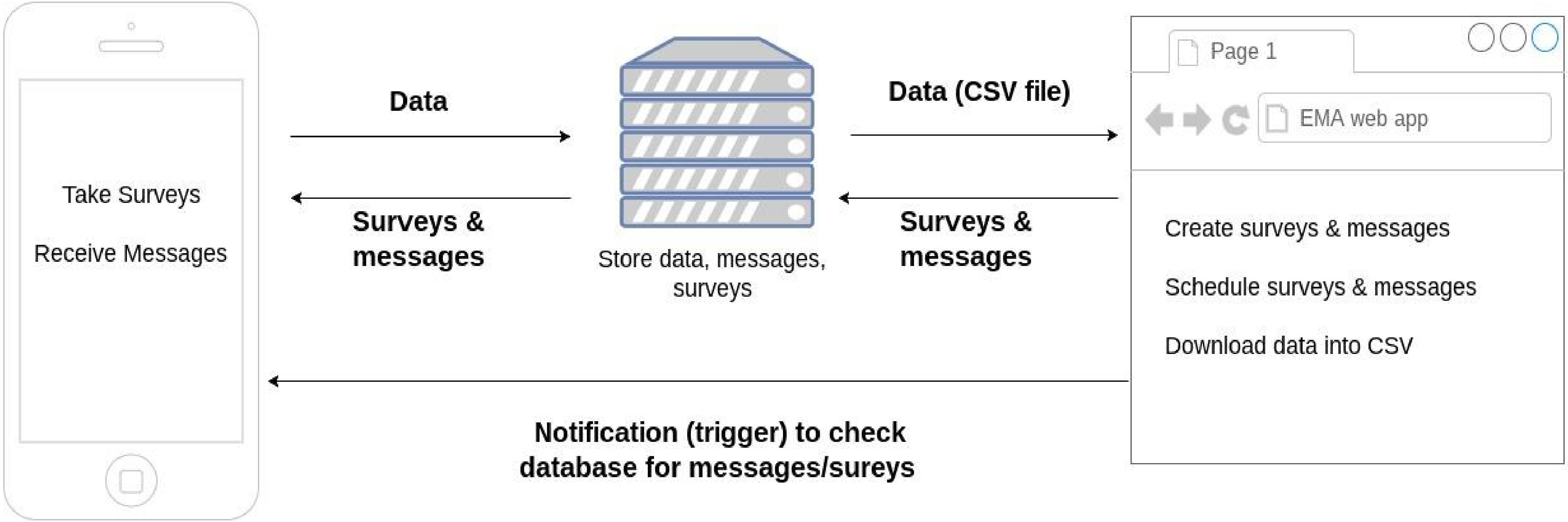
Take Surveys
Receive Messages

- Website:
  - Only accessible via login system
  - Study administrators/clinicians can:
    - Create and schedule surveys & messages
    - Download data as a CSV file
    - Manage study settings & participants

Page 1
EMA web app
Create surveys & messages
Schedule surveys & messages
Download data into CSV

- Database
  - Stores data and scheduled surveys and messages
  - Data consists of survey results, time taken to complete, start & end times

Components as a System



Take Surveys

Receive Messages

Data

Surveys & messages

Store data, messages, surveys

Data (CSV file)

Surveys & messages

Create surveys & messages

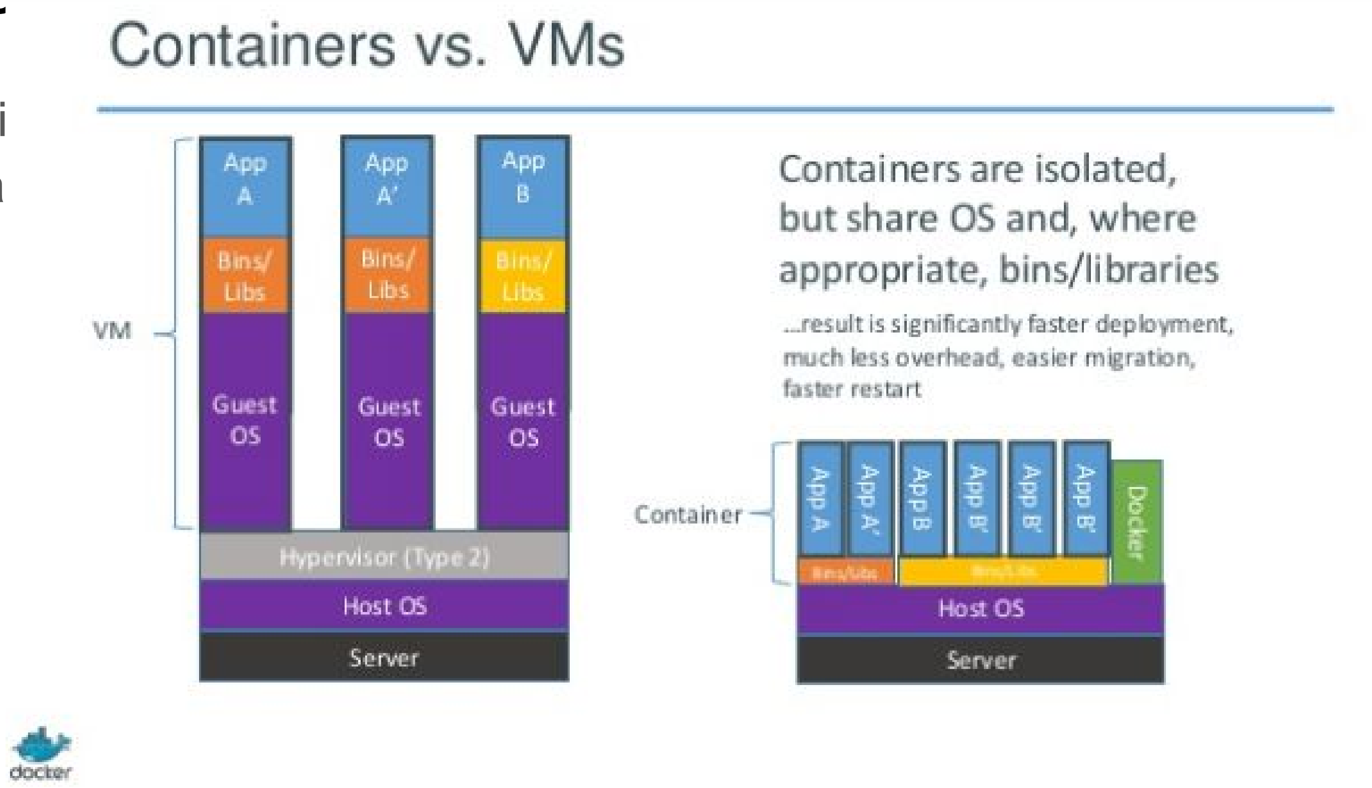
Schedule surveys & messages

Download data into CSV

Notification (trigger) to check database for messages/sureys

Docker

- Elimi
- Isola



Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart

Image from:Docker Introduction Slides, https://www.slideshare.net/dotCloud/docker-intro-november

Backend

- Database: PostgreSQL
- API: Python
  - Flask - web framework, few dependencies
  - Werkzeug - dependency for flask, WSGI utility library
    - Encryption library for passwords etc.
  - Marshmallow - validate, deserialize, serialize data
  - SQLAlchemy - database toolkit

Frontend

- WEB: HTML and AngularJS
  - Bootstrap
  - NGINX - web server

Mobile Side

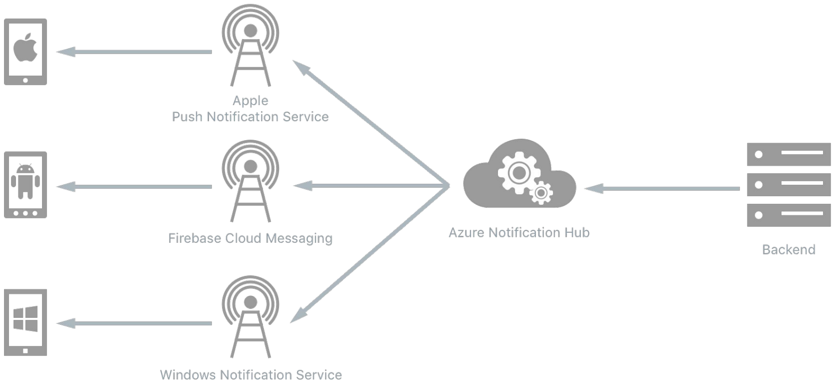
- Receive notification from Firebase Cloud
- Retrieve information from database (in Docker)
- Respond to survey question
- Send back the answer to database

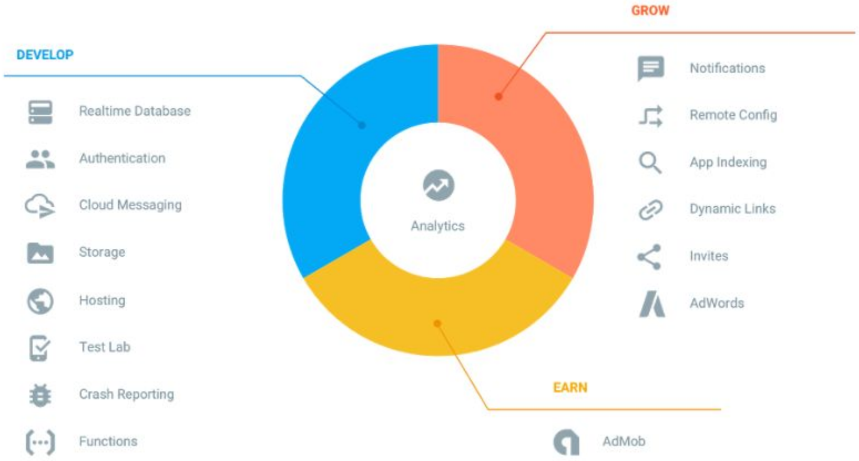
Remote Push Notification

Notification:

-Apple Push Notification service (APNS)

-Azure Notification Hub





Firebase

Image credit:https://firebase.google.com/ Image credit: https://developer.xamarin.com/guides/xamarin-forms/cloud-services/push-notifications/azure/

How Firebase works:

- Firebase contains App ID and device ID
- Features to schedule when to push notification
- Receive request from backend

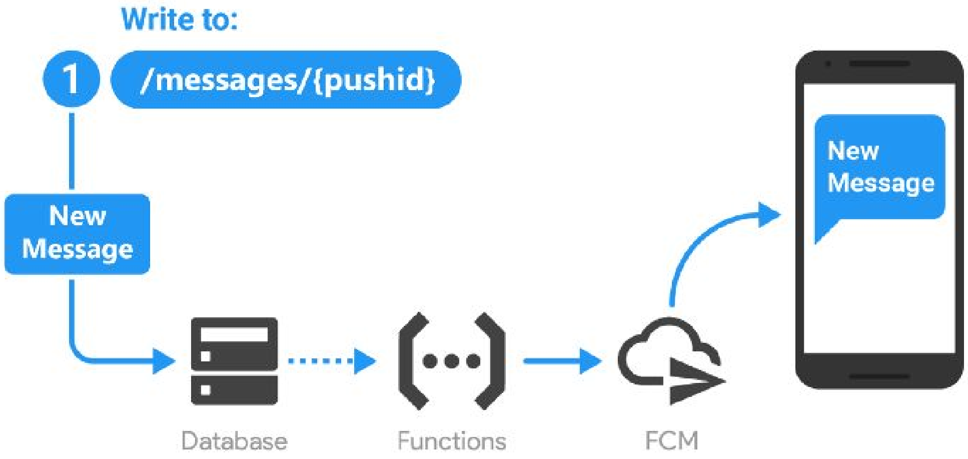


Image credit: https://blog.chizobaogbonna.me/device-to-device-push-notification-using-cloud-functions-for-firebase-5e607d74d7a2

Network Reachability

- Before sending any data to the database, the app will have to detect the reachability of the network
- Connection to the server will be check every time the app need to make a web request
  - Reachable: The host is reachable
  - On traffic: Reachable, but a connection must be initiated. The connection will be initiated on any traffic to the target detected.
  - IsWWAN: Reachable over the cellular connection
    - (GPRS, EDGE or 3G)

Web Request

- The mobile app has the ability to create post requests with the URL of the server
    - URL can start with www, http, https, or an IP address...
- Since our API is written in Python, the data that the app sends to the server must be in json format.

Web Response

- Send the post request to the server
- We can also get the stream containing response data sent by the server
- If the connection is reachable, the data that the app sends to the server will be saved to the database.

Next semester

- Focus on refining and completing the user interface
- Create full messaging functionality
  - Send images
- Set up the system on a permanent external server
- Have the remote push notification send from Docker