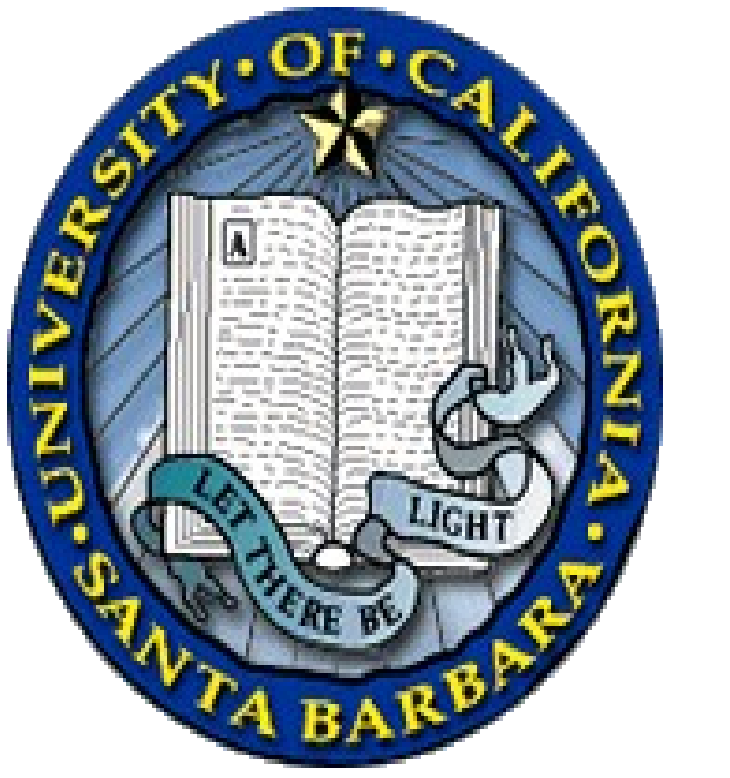


EmerGence: A Delay Tolerant App for Disaster Relief

Udit Paul, Michael Nekrasov, Elizabeth Belding

University of California, Santa Barbara

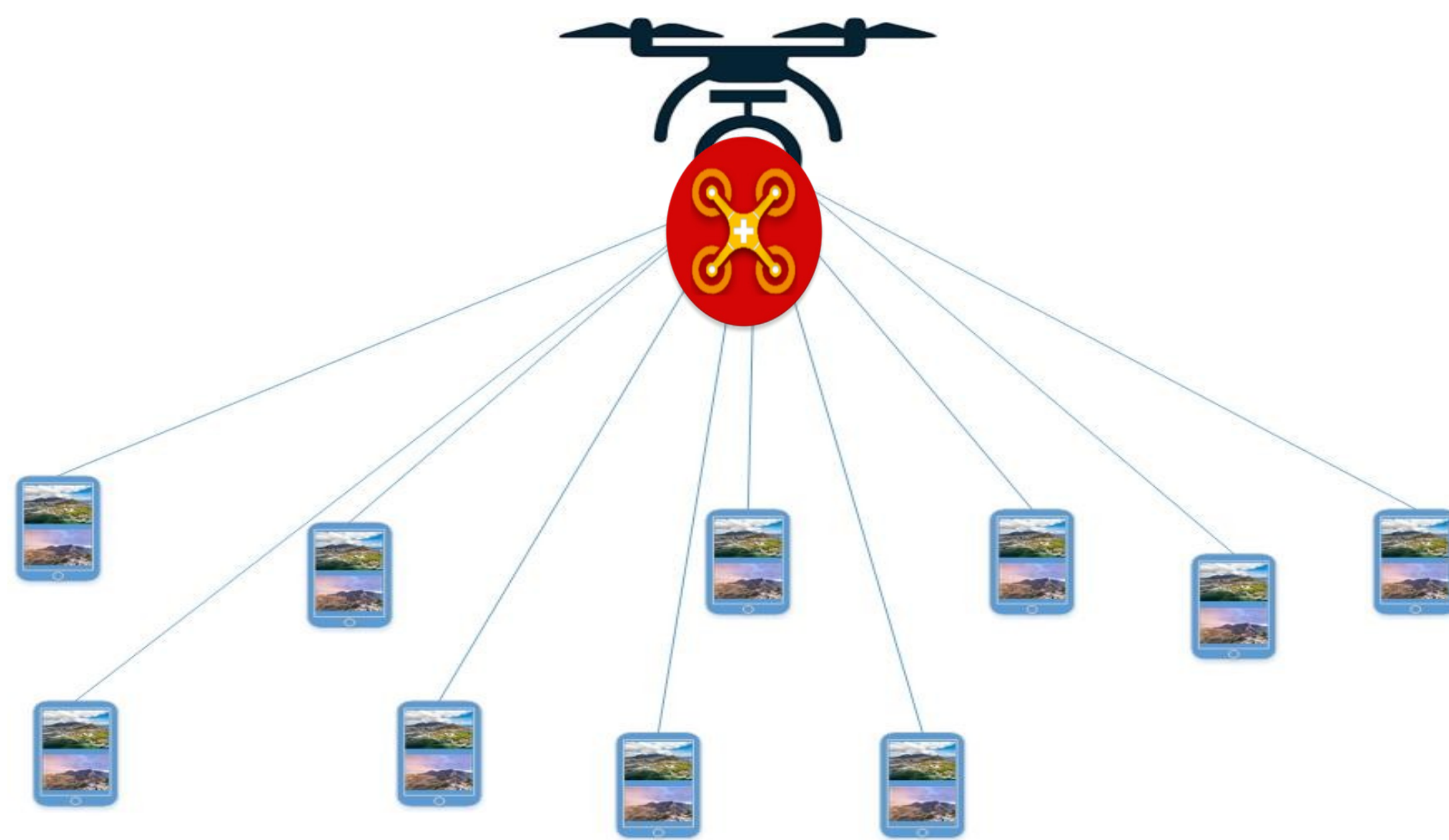


Motivation

When natural disasters occur, they often partially or completely destroy communication infrastructure. This results in either significant delays or complete loss of communication capabilities for the people trapped in the affected areas. Our work seeks to design a user application that can leverage opportunistic network connectivity when it exists, post disaster.

Project Goals:

1. Design a user application, EmerGence, that can function with intermittent network connectivity.
2. Use Unmanned Aircraft Systems (UAS) to bring edge network connectivity and computing to users in disaster-struck areas.

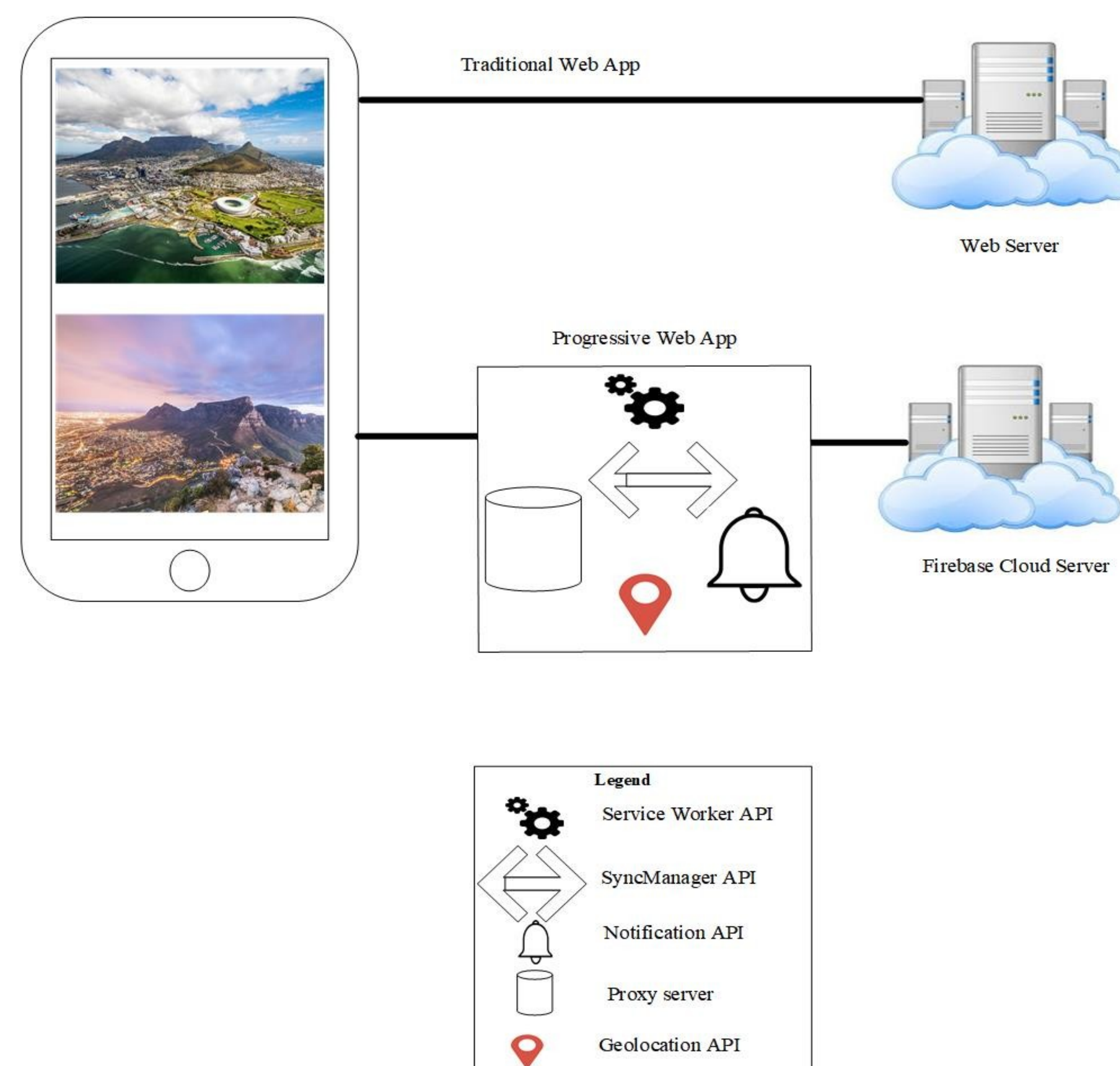


Overview

EmerGence's main features:

- UAS acts as a *data mule* when no end-end Internet connectivity is available
- UAS serves a web app built using a new web development concept known as the Progressive Web Application (PWA)
 - Unlike web pages, it appears as a native user application using capabilities of a JavaScript file called a ServiceWorker
 - The ServiceWorker provides features such as background processing and pre-caching of data, even when UAS is out of range
 - While behaving like a native platform app, EmerGence does not need to be pre-installed to a user device
- It leverages *opportunistic connectivity*
 - Without network connection, it renders locally stored contents to the users.
 - Data created by users is buffered locally and pushed once connection to Internet or a nearby UAS is established without user intervention.
 - *Local data sync* between users in range of drone
 - It uses a captive portal to allow discoverability by user without prior knowledge of the app.
 - Every post is geotagged to help first responders locate people with ease.

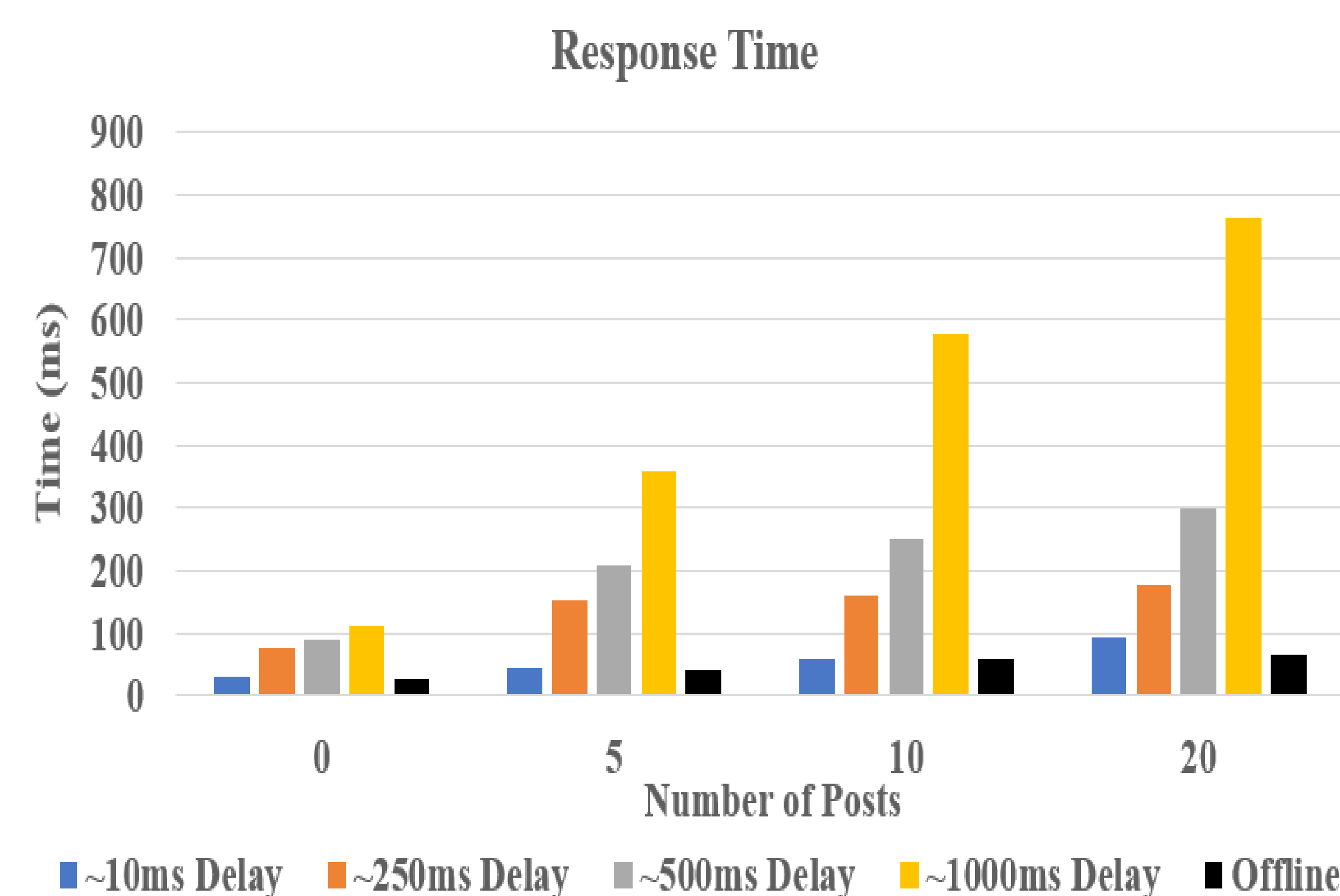
To implement EmerGence, various APIs were employed. The figure below shows the difference between a traditional web app and a PWA when accessed from a user device.



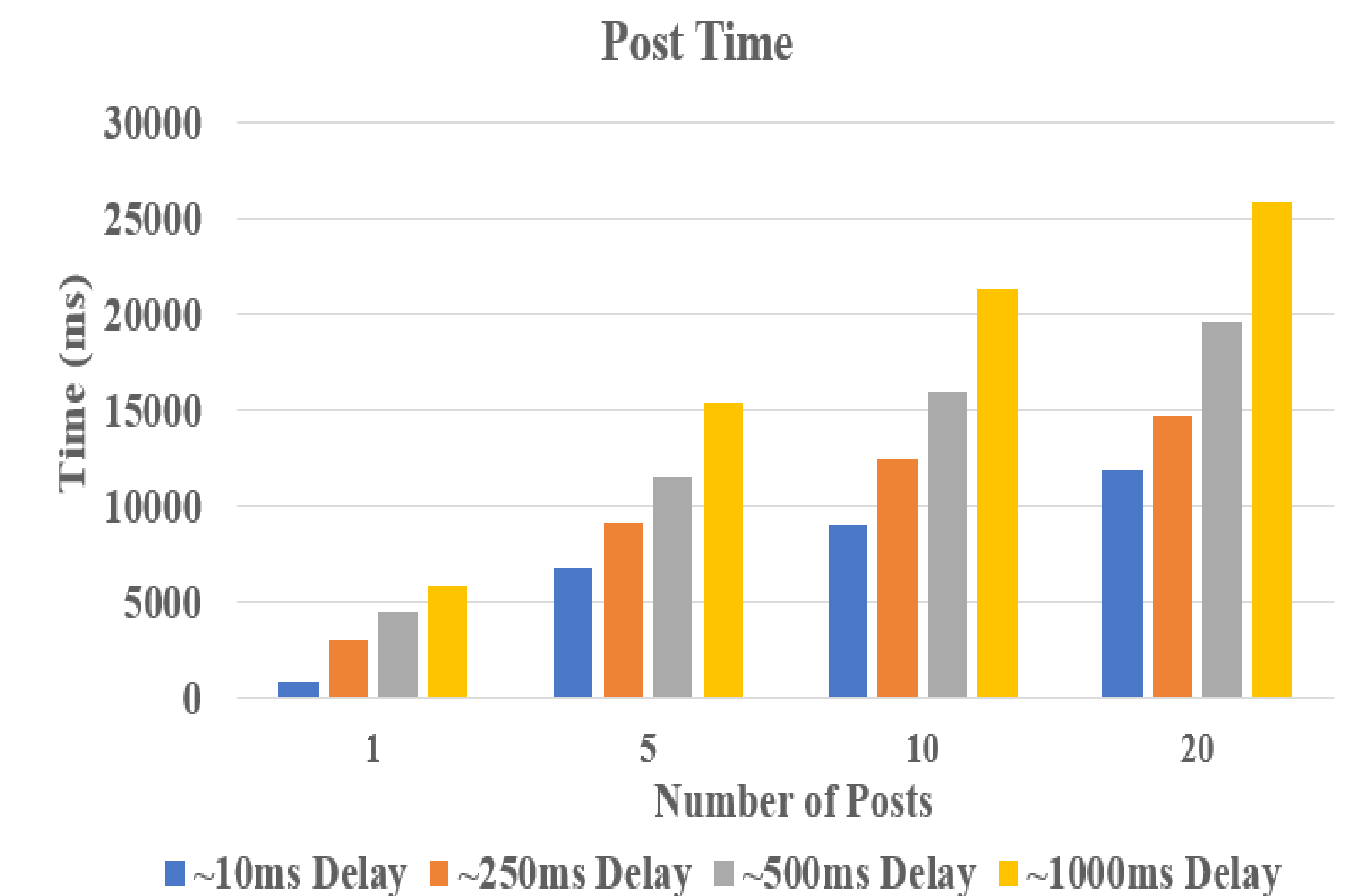
Preliminary Evaluation

Some results obtained so far include:

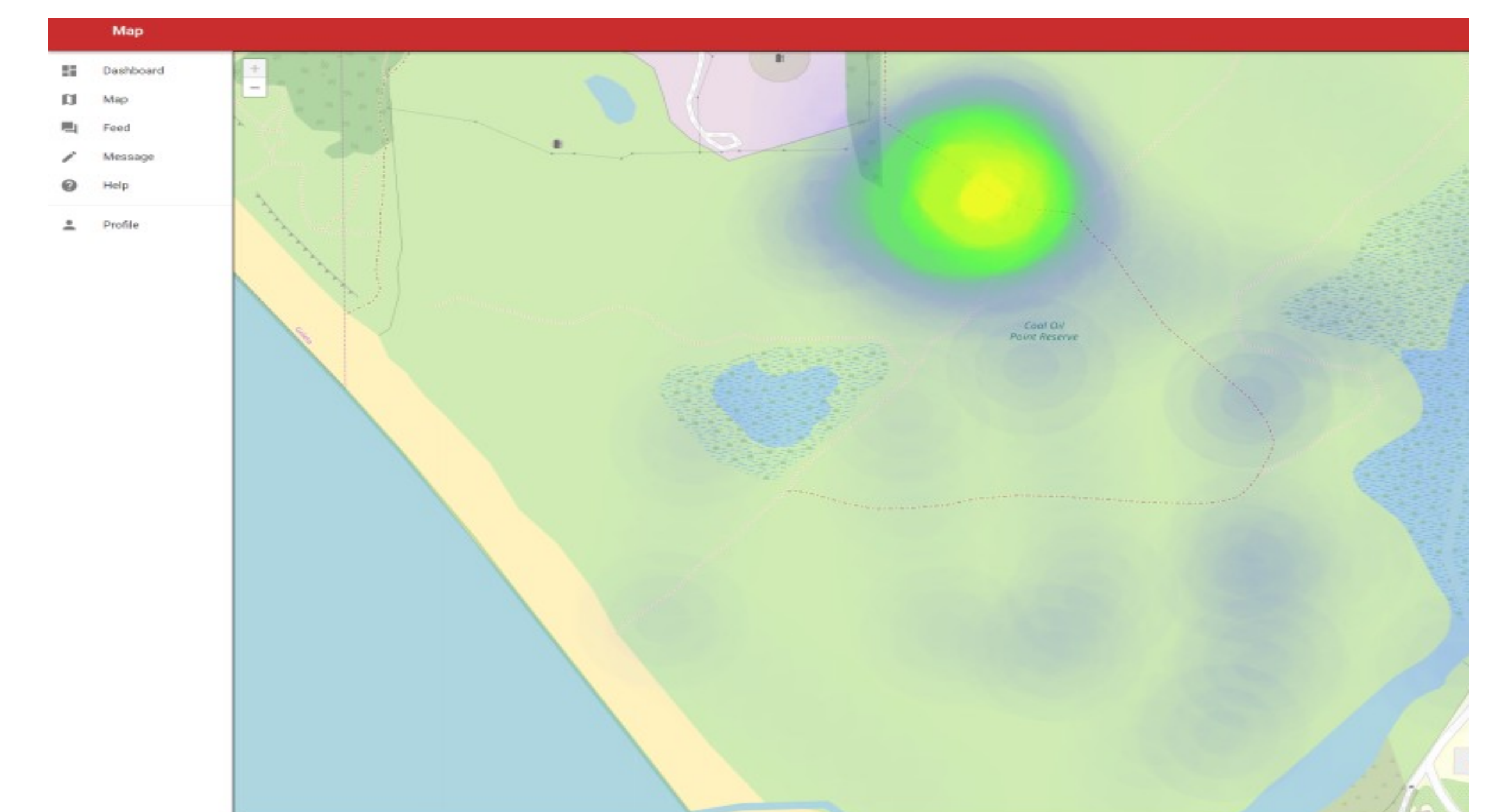
- **Response time:** The app was loaded under various amount of network delay and the response time increased as the number of posts rendered, and network delay increased.
- **Offline rendering:** The app was able to render locally cached posts almost immediately in the absence of network connectivity.



- **Post Time:** The app is able to buffer a number of posts when made during lack of end-to-end connectivity.
 - As the network delay increased, the delivery time for the posts also increased.



- **Cache Storage:** Periodic cache trimming strategy was used to prevent using large amount of browser storage space.
 - As more posts were buffered, more cache storage was used.
 - **Device Storage:** As EmerGence does not need to be preinstalled to a user device, it occupies only 203KB of device storage space.
- **Passive Localization:** : To locate users incapable of requesting for help, we:
 - have utilized an 802.11 adapter on the UAS to passively and accurately locate mobile devices on the ground.



Future Work

Using EmerGence as the data delivery platform, we are researching ways of using wireless sensing and on-the-drone processing for improving disaster situational awareness and response capabilities.



We are targeting these tools for first responders and other on the ground personnel to extend the use of UASs beyond purely visual applications.