Project Revere

Atrocity Watch Hackathon - June 6, 2014

Cloudera Offices - Palo Alto California

Project Members

Jason Almeida

Middle-end engineer

jasonkalmeida@gmail.com

Sam Joseph

Front-end engineer

samaj94@gmail.com

Sergey Serebryakov

Backend engineer

hellomind@gmail.com

Aaron DeVera

Middleware engineer

aaronsdevera@gmail.com

Project Description

Developing nations lack the infrastructure for Amber alert notifications, but there is also no homogeneity to devices or services being used.

We aim to provide a cross-device and cross-service platform that allows people, not only governments, to provide alerts on a localized and specific scale.

Technology Used

Twilio API and Twitter API

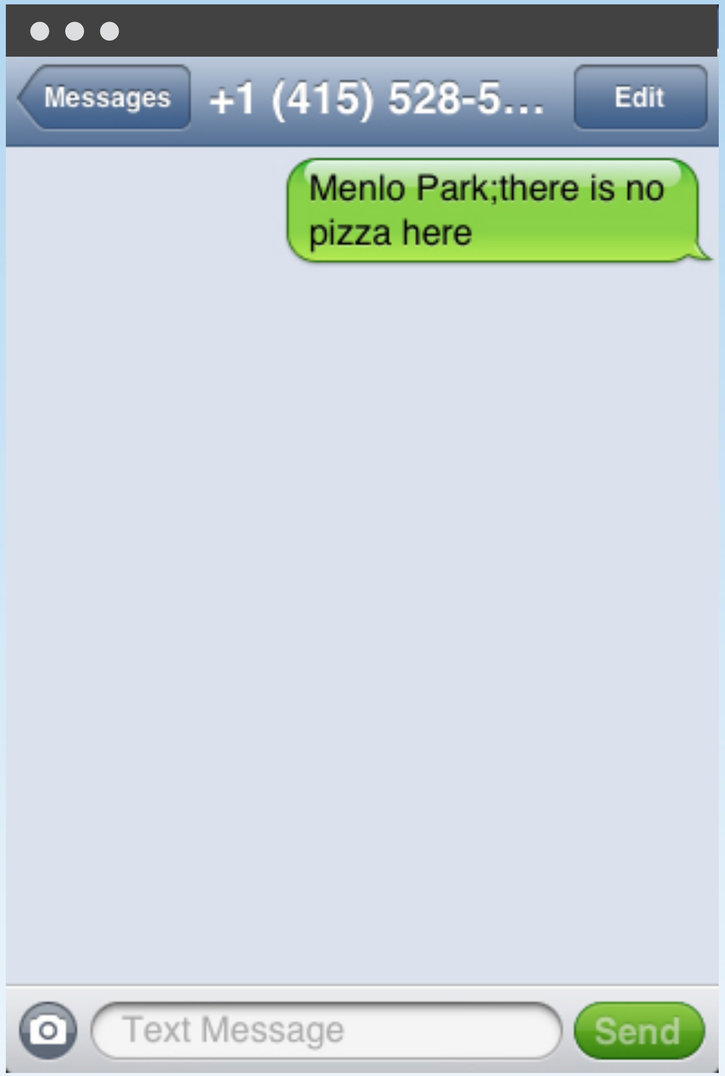
Javascript

Web server on DigitalOcean

Request routing server with Node.js

Web framework for visualization: HTML 5 + CSS + jQuery

Demo



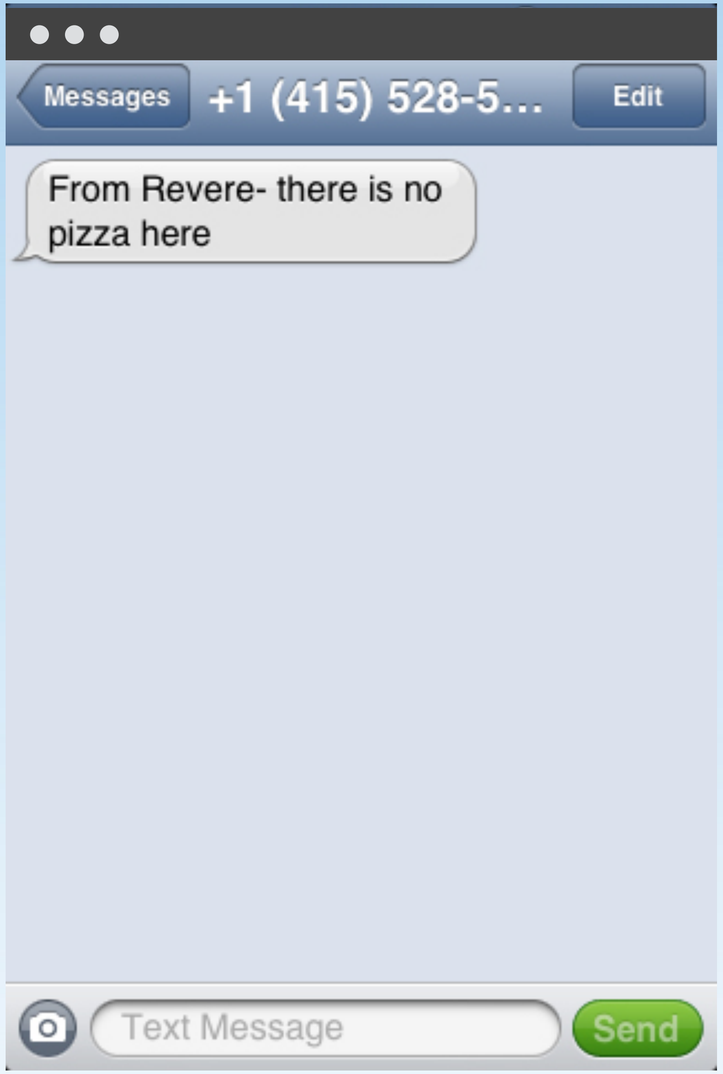
Field operator texts the presupposed number established by the Revere server.

Use of syntax:

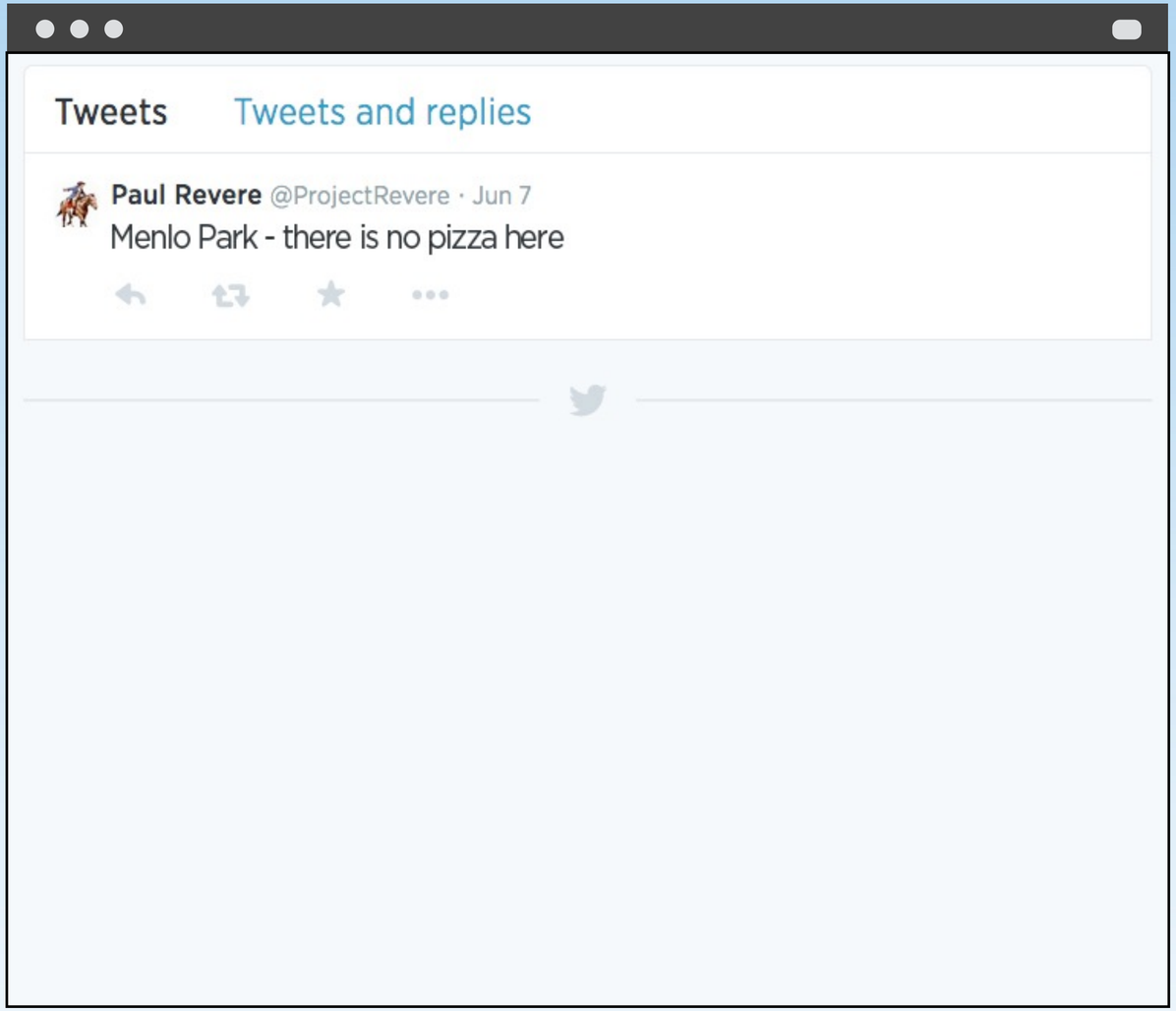
[location];[message to send to people in that location]

Demo

The message is distributed to all field operators tagged by the server to be in the specified area within seconds.



Demo



Messages are pushed to a specified Twitter account for internet communication.

Technologies Used

```
exports.parseMessage = function(message) {
    // splits message into an array, with semicolon as
separator
    var message_tokens = message.split(";");
    var message_location = message_tokens[0].trim();
    var message_content = message_tokens[1].trim();

    console.log("location = " + message_location);
    console.log("content = " + message_content);
```

parsemessage.js

JavaScript - 505 bytes

```
var querystring = require("querystring");

exports.receiveMessage = function(req, res, parseCallback)
{
    if (req.method == "POST") {
        console.log("[200] " + req.method + " to " +
req.url);

        req.on("data", function(chunk) { // hopefully we
```

receivemessage.js

JavaScript - 974 bytes

```
var Twit = require("twit");

exports.tweetMessage = function(message) {
    // Aaron's account
    var T = new Twit({
```

tweetmessage.js

JavaScript - 1 KB

```
var http = require("http"),
    path = require("path"),
    fs = require("fs"),
    send = require("./sendmessage"),
    receive = require("./receivemessage"),
    parse = require("./parsemessage"),
    tweet = require("./tweetmessage");

// our virtual twilio number
```

server.js

JavaScript - 3 KB

```
var twilio = require("twilio");

exports.sendMessage = function(from, to, body) {
    // Create a new REST API client to make authenticated
requests against the
    // twilio back end
```

sendmessage.js

JavaScript - 1 KB

Revere is modular and lightweight; new modules can be added for more features, and can be built upon by future developers.

Messages are not stored, thus making Revere efficient, fast, and secure.The entire platform requires less than 45kb.

Future Steps

- Implement threshold and curation functionality to prevent spam
- Incorporate natural language processing to increase usability
- Logging and location triangulation for visualization
- Utilize other social networks and inputs
- Collaboration with cellular carriers for increased verification and scale