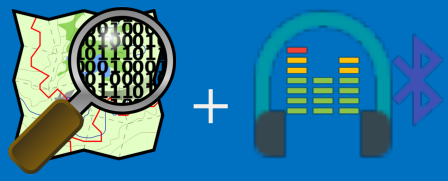
Open Street Maps

Brightening the world for the blind using spatial audio

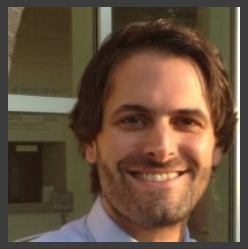


ERIK SCHLEGEL

@erikschlegel1
https://github.com/erikschlegel

Microsoft

A little about me



Open Source Enthusiast and Data Geek.
Mapping Hobbyist.
Senior Engineer @ Microsoft in New York City.
Backend Architect for the Guide Dogs Project.
Early Contributor to the UWP React Native Platform.

What is Openstreet Map, really?

A PostgresSQL database often referred to as the wikipedia of spatial data.

A free, editable, open map of the world.

Tile caching server.

Crowdsourced dataset, made entirely by voluntary efforts of the online masses.

Basically, unrestricted use and no data approval body.

Built by the community in the open.

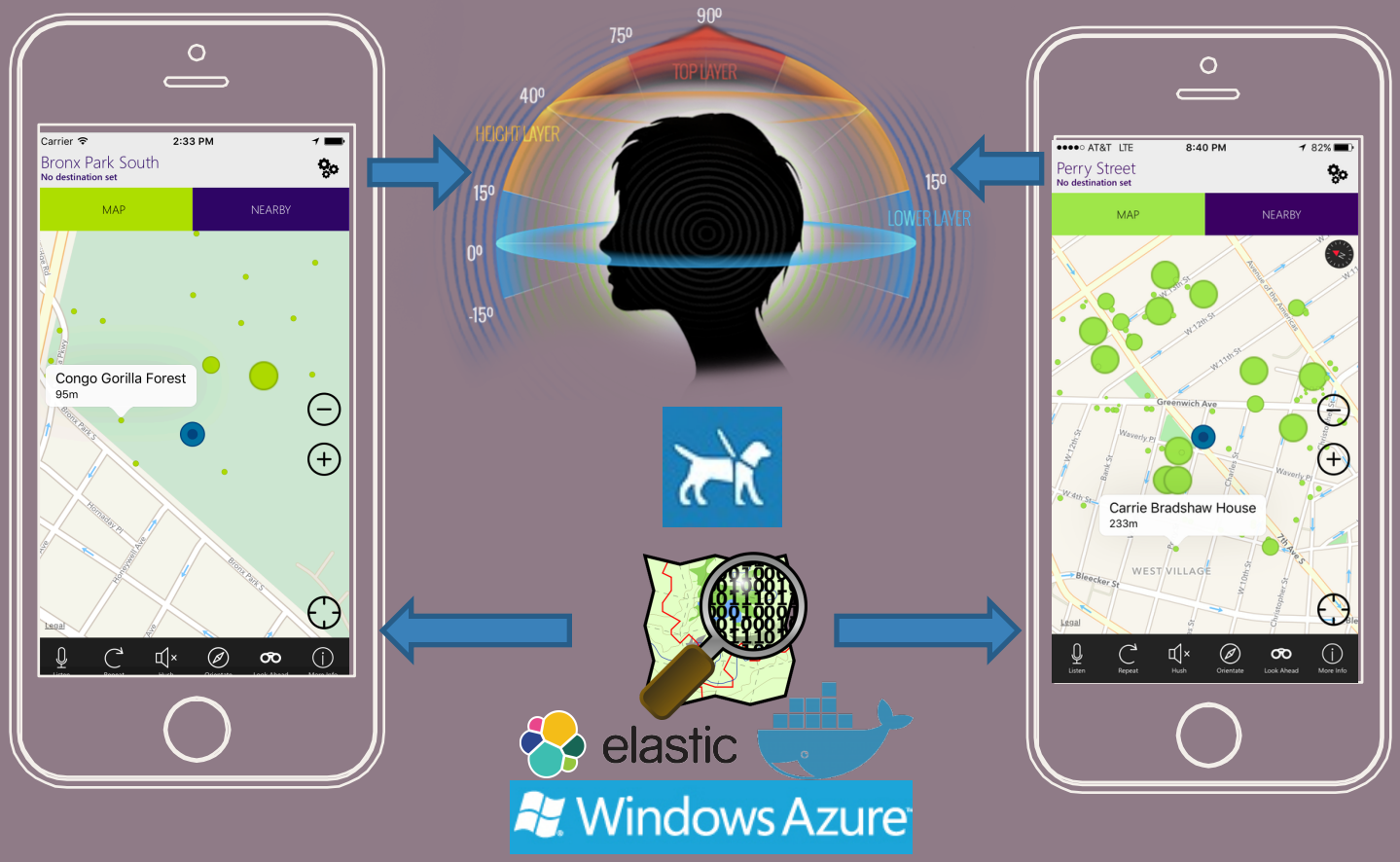
What is Openstreetmap

An OSM map targeted for the Melbourne Hipster community.

hipstermelbourne.org

What Is Guide Dogs

- Rendering a world with sound, powered by Open Street Maps.
- Empowering blind users to explore a city using 3D audio to improve independent mobility.



What can I use OSM for?

Determining nearby points of interest
Rendering custom map tiles(*i.e. heatmaps*)
Geocoding(i.e. *Where Am I / Where is this*)
Turn-by-Turn Navigation
Advanced Mapping Websites
Offline mapping and smartphone access
Realtime mass transit updates
Local Search
Pretty much anything

Who Uses OSM?

Foursquare
Apple
Github
Road Trippers
Mapbox
Mapquest
FLICKR
and many more.

Show me the data

Flexible Data Schema – Arbitrary Attributes. 3 main primitives

```
<node id="4045616048" visible="true" version="
"40.7364859" lon="-74.0022051">
 <tag k="amenity" v="restaurant"/>
 <tag k="cuisine" v="american"/>
 <tag k="name" v="Beatrice inn"/>
</node>
<way id="46610734" visible="true" version="2"
 <nd ref="42433278"/>
 <nd ref="42433275"/>
 <nd ref="42433272"/>
 <tag k="highway" v="unclassified"/>
 <tag k="name" v="Waverly Place"/>
 <tag k="oneway" v="yes"/>
 <tag k="tiger:cfcc" v="A41"/>
 <tag k="tiger:county" v="New York, NY"/>
 <tag k="tiger:name_base" v="Waverly"/>
 <tag k="tiger:name_type" v="Pl"/>
 <tag k="tiger:reviewed" v="no"/>
 <tag k="tiger:separated" v="no"/>
 <tag k="tiger:source" v="tiger_import_dch_v0.
</way>
<way id="250474195" visible="true" version="1"
 <nd ref="2569066398"/>
 <nd ref="2569066464"/>
 <nd ref="2569066466"/>
 <nd ref="2569066404"/>
 <nd ref="2569066398"/>
 <tag k="addr:housenumber" v="220"/>
 <tag k="addr:postcode" v="10014"/>
 <tag k="addr:street" v="West 11th Street"/>
 <tag k="building" v="yes"/>
 <tag k="height" v="8.5"/>
 <tag k="nycdoitt:bin" v="1080205"/>
</way>
```

1) Nodes*(single point)* - Points Of Interest

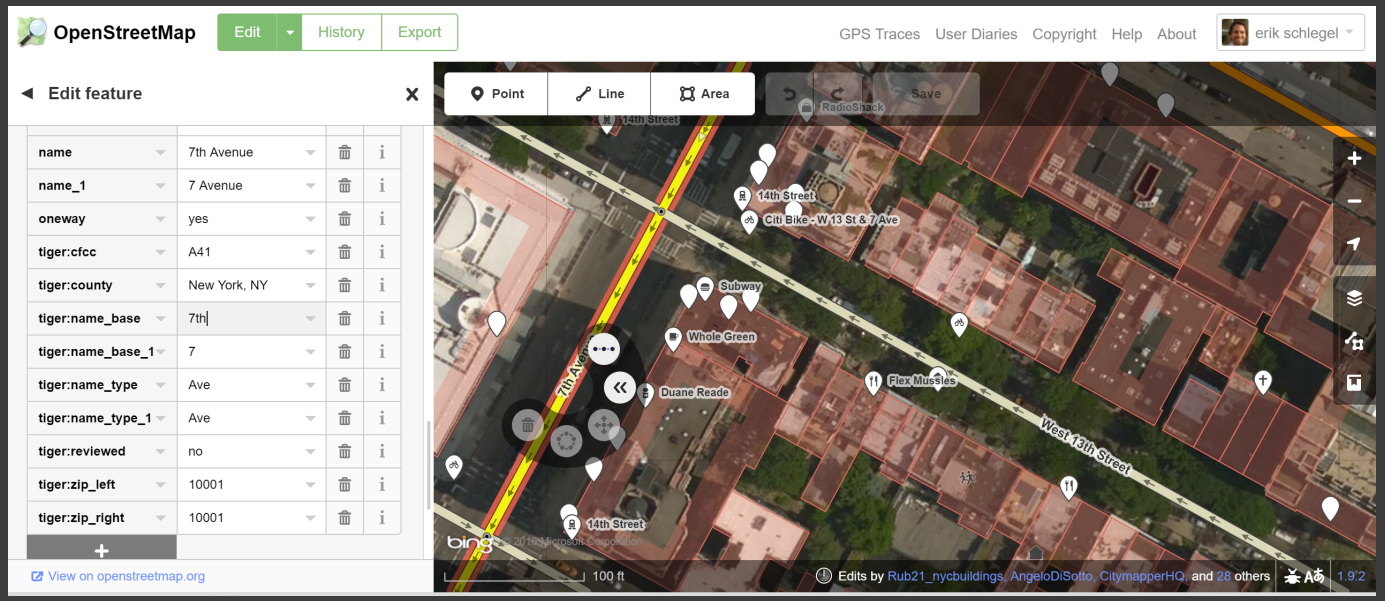
2) Open Way*(lines)*- Roads / Highways / Rivers / Walkways

3) Closed Ways*(polygons)*- Buildings / Parks / Universities

Map a Trail or Road

Explore - Create traces on your handheld GPS device and import to OSM.

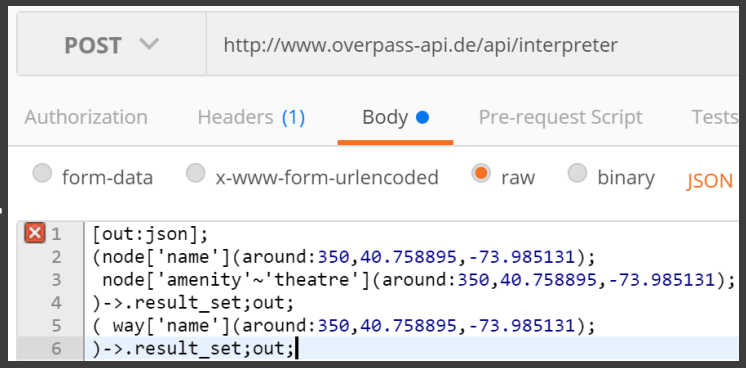
ID – Web-Based Openstreet Map Editor.



Data Extraction

Overpass API – 3rd party REST based service provider for retrieving OSM data. Offers a powerful query language off OSM.

REQUEST->



RESPONSE->

Planet Files – All OSM data in a single file.

Osmosis – A CLI that imports / exports planet files to / from a DB. Also compares planet files and creates changesets

```
//import all speed_cameras in the world into a Postgres DB
osmosis --read-pbf theworld.osm.pbf --node-key-value keyValueList="highway.speed_camera" --log-progress --write-pgsql database=pgsnapshot
```

Demo: Custom Tiles

Custom tile rendering frameworks TileMill and Mapzens Tangram WebGL.

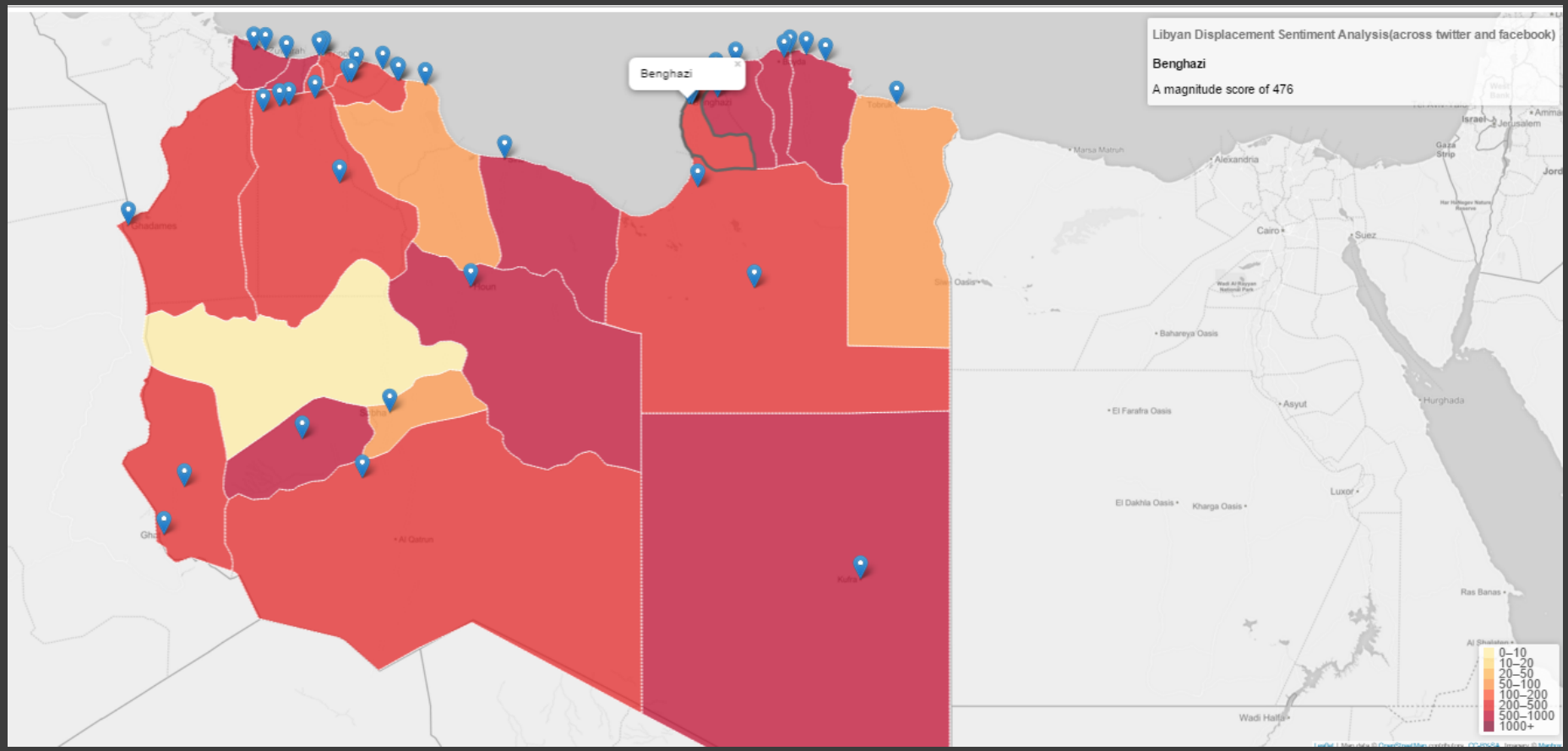
1. **Gotham Tangram Scene**

2. **Tron Tangram Scene**

Demo: HeatMaps using Leaflet and OSM

Demo: A heatmap representing sentiment magnitude levels of displaced Libyan citizens over the last month, by region.

Boundaries are sourced from OSM, filtered on bounding view via TurfJS.



Demo: Geospatial Analysis Tools

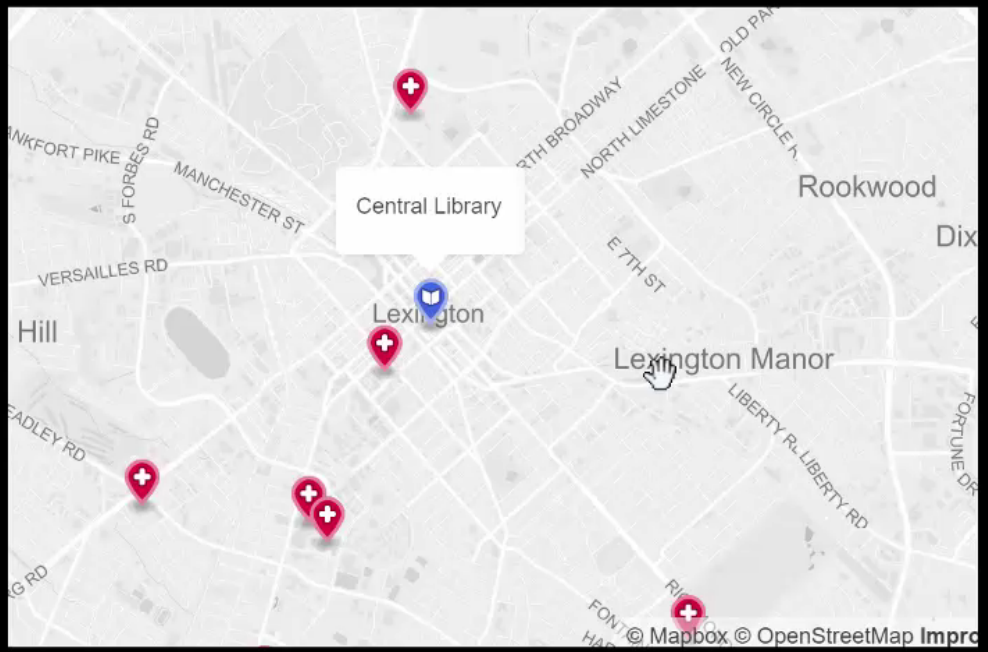
Enter TurfJS – Geospatial-based JS framework that speaks GEOJSON.

```
var libraries = {
  "type": "FeatureCollection",
  "features": [
    { "type": "Feature", "properties": { "Name": "Village Branch", "Address": "2185 Versailles Rd" }, "
    { "type": "Feature", "properties": { "Name": "Northside Branch", "ADDRESS": "1733 Russell Cave Rd"
    { "type": "Feature", "properties": { "Name": "Central Library", "ADDRESS": "140 E Main St" }, "geome
    { "type": "Feature", "properties": { "Name": "Beaumont Branch", "Address": "3080 Fieldstone Way" },
    { "type": "Feature", "properties": { "Name": "Tates Creek Branch", "Address": "3628 Walden Dr" }, "
    { "type": "Feature", "properties": { "Name": "Eagle Creek Branch", "Address": "101 N Eagle Creek Dr"
  ]
};

var map = L.mapbox.map('map', 'mapbox.light')
  .setView([38.05, -84.5], 12);
map.scrollWheelZoom.disable();

var hospitalLayer = L.mapbox.featureLayer(hospitals)
  .addTo(map);
var libraryLayer = L.mapbox.featureLayer(libraries)
  .addTo(map);

    // When a library is clicked, do the following
libraryLayer.on('click', function (e) {
  // Reset any and all marker sizes to small
  reset();
  // Get the GeoJSON from libraryFeatures and hospita
  var libraryFeatures = libraryLayer.getGeoJSON();
  var hospitalFeatures = hospitalLayer.getGeoJSON();
  // Using Turf, find the nearest hospital to library clicked
  var nearestHospital = turf.nearest(e.layer.feature, hospitalFeatures);
```



Data Accuracy Oversight

TOOLS OSMLint, Osmose, Keep Right, Tile Reduce, Map Roulette, JOSM Validator and OSM-QA-Tiles

---

Growing number of OSM error detection tools to flag mapping errors.

Mapbox has an operations team that monitor / resolve OSM breaks.

Automated comparison models are reconciling OSM against Google Maps, TIGR, Bing aerial imagery and many other sources.

Version Control

All OSM changes are time machined by user. The State of the Map can be re-played from any given time*(“Attic Data”)*.

overpass.osm.rambler.ru/cgi/interpreterdata=[bbox][date:"2012-09-12"];node[amenity=post_box];out;&bbox=7.0,50.6,7.3,5

OSM changes can be exported by date range for a given area.

One can compare OSM spatial changes / trends over time.

overpass.osm.rambler.ru/cgi/interpreterdata=[bbox][diff:"2012-09-14","2012-09-21"];node[amenity=post_box];out;&bbox=7.0,50.6,7.3,50.8

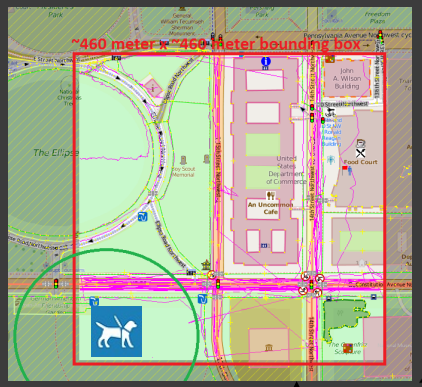
OSM Accessibility

One of the key OSM selling points for the Guide Dogs project was how accessible the data is for users that are deaf and/or blind.

Accessibility Tags Guide Dogs uses for Navigation

entrance=yes (the entrance of a way / park)
blind:description:en=(callout text for blind users) Warning: Dangerous automatic door turning anticlockwise.
tactile_paving=incorrect
wheelchair=limited
kerb= lowered or raised(predefined curb type). Kerb:heaight is also tracked
highway=steps (for escalators) or footway (for moving walkways)
conveying=incline or decline for escalator direction
footway=sidewalk
surface=* (defines paving).
access= (legal access restrictions) yes or no or private
crossing=*(defines the restrictions of the pedestrian crossing)
lit=*(indicates the presence of lighting)

Guide Dogs – a tile based system



Key Legend

- Azure Services
- 3rd Party Service
- OSM Service

REST

OSM ElasticSearch Cluster

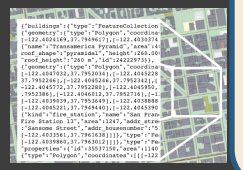
- Master Node
- Master Node
- Search Load Balancer
- Data Node
- Data Node
- Data Node

WEB API Server

- Spatial Data
- Location Sense
- More Information
- TTS Services

Updated minutely from OSM

MAPZEN Tile Server Cluster



Azure Cache Web Job

Windows Azure Marketplace

Wikipedia

OSM Global DataStore
PostgreSQL

Demo: Near-Realtime Performance

Content rendering is based on user movement and heading changes.

Responsiveness and availability of OSM data had to be near realtime.

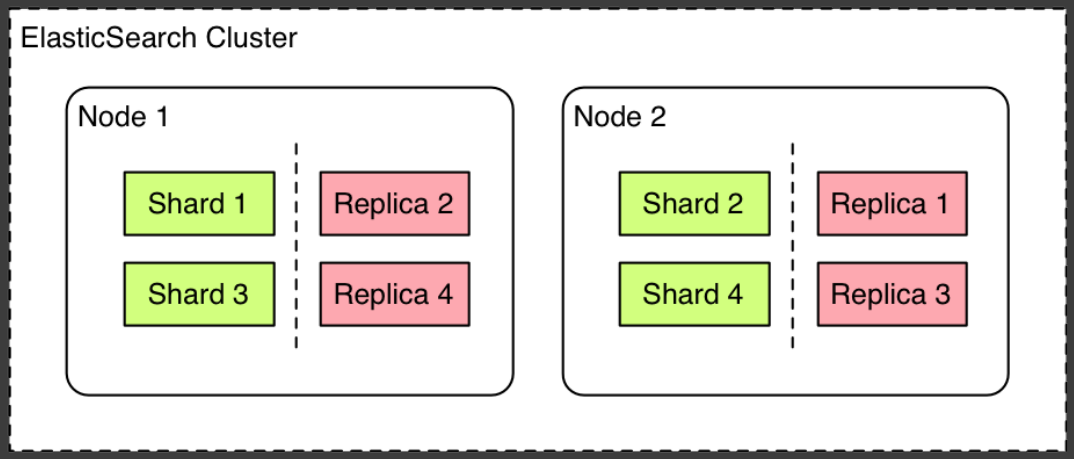
Support for handling complex geo shapes that's fault tolerant and load balanced.

Powered by Azure and Elasticsearch NEST .NET Client.

Tile data provided by OSM and Mapzen.

Utilize OSM changeset info to only update tiles that were recently changed.

Elastic Architecture Concept



Elastic Search NEST Client

Bulk Indexing Documents

```
var descriptor = new BulkDescriptor();
var coord1 = new double[][][] { new double[][] { new double[] { 40.73659338, -74.0008906 }, new double[] { 40.73659339
descriptor.Index<Place>(op => op.Document(new Place()
{
    amenity = "restaurant",
    cuisine = "american",
    name = "Bluestone",
    Id = "232323223",
    type = "node",
    coordinates = new Coordinates() { type = "multilinestring", coordinates = coord1 }
})
.Index("places"));

descriptor.Index<Place>(op => op.Document(new Place()
{
    amenity = "restaurant",
    cuisine = "italian",
    name = "Starbucks",
    Id = "232323221",
    type = "node",
    location = new Location() { lat = 40.73659338, lon = -74.0008906 }
})
.Index("places"));

var test = await ElasticManager.Instance.BulkOperationAsync(descriptor);
```

Elastic POI Index

```
{
    "mappings": {
        "place": {
            "properties": {
                "id": {"type": "string"},
                "name": {"type": "string"},
                "type": {"type": "string"},
                "location": {
                    "type": "geo_point"
                },
                "coordinates": {
                    "type": "geo_shape"
                },
                "amenity": {"type": "string"},
```

Geo Shape Intersection Search

```
var bbox = new BoundingBox.Builder().Build(location, meterDistance, DistanceType.Meters);

return await Connection.SearchAsync<Place>(search => search.Size(2000)
          .Query(f => f.Bool(b =>
          {
              var filters = new List<Func<QueryContainerDescriptor<Place>, QueryContainer>>();

              filters.Add(filter => filter
                     .GeoBoundingBox(bb => bb.Field(bf => bf.location)
                                             .BoundingBox(bbox.maxMapPoint.Latitude, bbox.minMapPoint.Longitude,
                                                          bbox.minMapPoint.Latitude, bbox.maxMapPoint.Longitude)));

              filters.Add(filter => filter.GeoShapeEnvelope(box => box
                         .Field(p => p.coordinates)
                         .Coordinates(new List<Nest.GeoCoordinate>() { new Nest.GeoCoordinate(bbox.minMapPoint.Longitude,
                                                                       bbox.minMapPoint.Latitude),

                              new Nest.GeoCoordinate(bbox.maxMapPoint.Longitude, bbox.maxMapPoint.Latitude) })));

              return b.Should(filters);
          })));
```

Turn-By-Turn Navigation

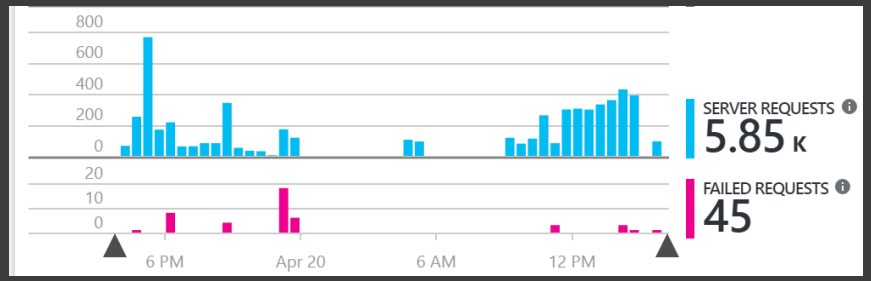
Mapzen has an open source turn-by-turn routing engine working off OSM.
Pedestrian route costing model that increases weighting on routes with foot paths, sidewalks, pedestrian access and entrances. Steps, stairs and alleys are slightly avoided.

RESPONSE

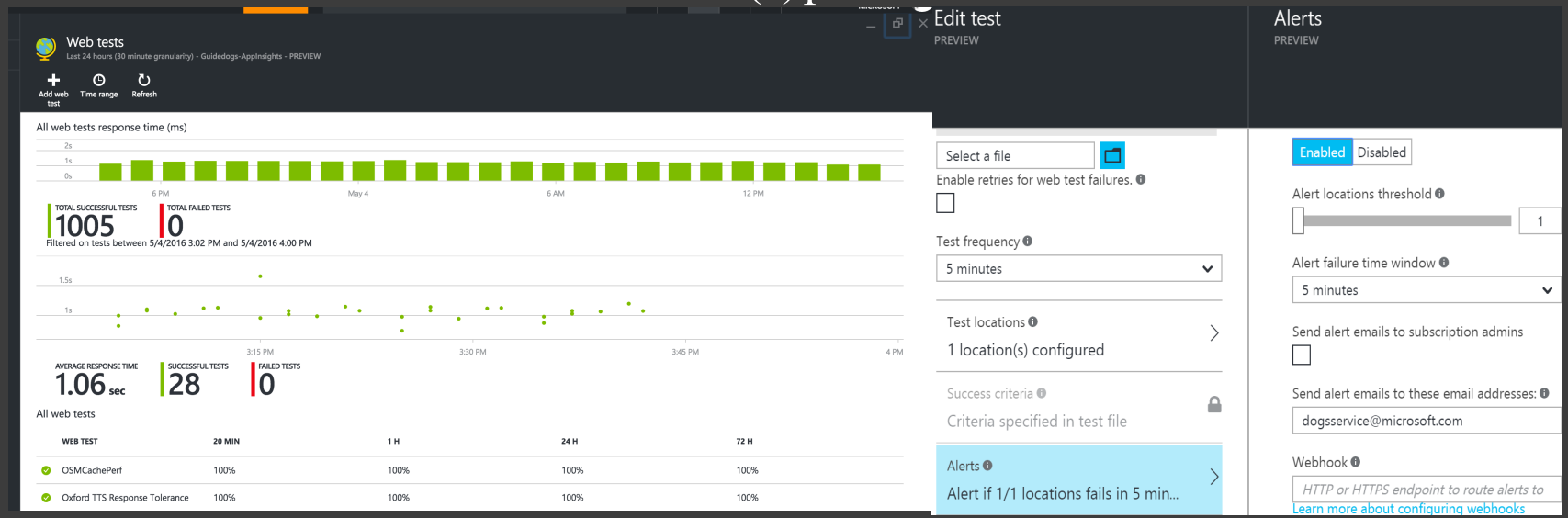

```
    "endPathIndices": [
      4
    ],
    "startPathIndices": [
      2
    ],
    "maneuverType": "Left",
    "names": [
      "Greenwich Avenue"
    ],
    "roadType": null,
    "rough": false
  }
],
"instruction": {
  "formattedText": null,
  "text": "Turn left onto Greenwich Avenue.",
  "verbal_pre_transition_instruction": "Turn left onto Greenwich Avenue.",
  "verbal_post_transition_instruction": "Continue for 90 meters.",
  "verbal_transition_alert_instruction": "Turn left onto Greenwich Avenue."
},
"sideOfStreet": right,
"travelDistance": 0.0860000029206276,
"travelDuration": 63
```

Production Quality Oversight

Application Insights - monitor and measure service performance and exceptions.



Web Tests - benchmark microservice(s) performance tests with alert escalation.



Why OSM as opposed to other map providers

~20 minute update schedule for propagating data changes.
A flexible data model where new tags / fields are well supported.
Availability of accessibility data in OSM.
Data is free, accurate and open source.
Flexible license.
Open Source tooling around collecting and analyzing OSM data.
Custom tiles.
Supports offline data download and navigation.
Growing community that contribute mapping data to OSM daily.

OSM Challenges

No official data schema standardization.

OSM is subject to contributor biases.

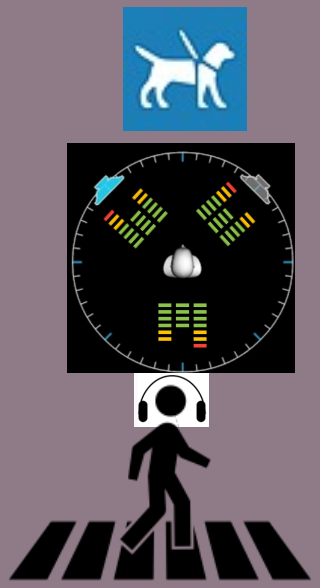
Retrieving OSM data requires more technical skill than getting data in.

Detail and precision across OSM varies from region to region.

No SLA with resolving systematic OSM data quality exception breaks.

3D Audio / Demo

Microsofts HRTF 3-D audio engine gives OSM spatial data a voice, and enables users to build a mental map of the area.
Spatial sound cues are used for navigation, points of interest and journey details alongside regular GPS instructions.
Cortana TTS services are used to translate text results to an audio stream.
Guide Dogs uses sound event notifications to help drive the user experience.



Thank You

ERIK SCHLEGEL


@erikschlegel1
https://github.com/erikschlegel