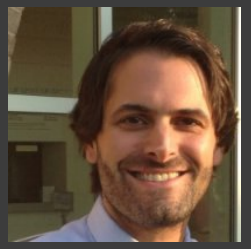

# Cities Unlocked

*Brightening the world for blind users with sound*

**Erik Schlegel - Microsoft**

@erikschlegel1

# A Little About Me



- Senior Engineer @ Microsoft New York City
- Open Source Enthusiast
- Backend Architect for the Cities Unlocked Project
- Early Contributor to the Windows React Native Platform

# What is Cities Unlocked

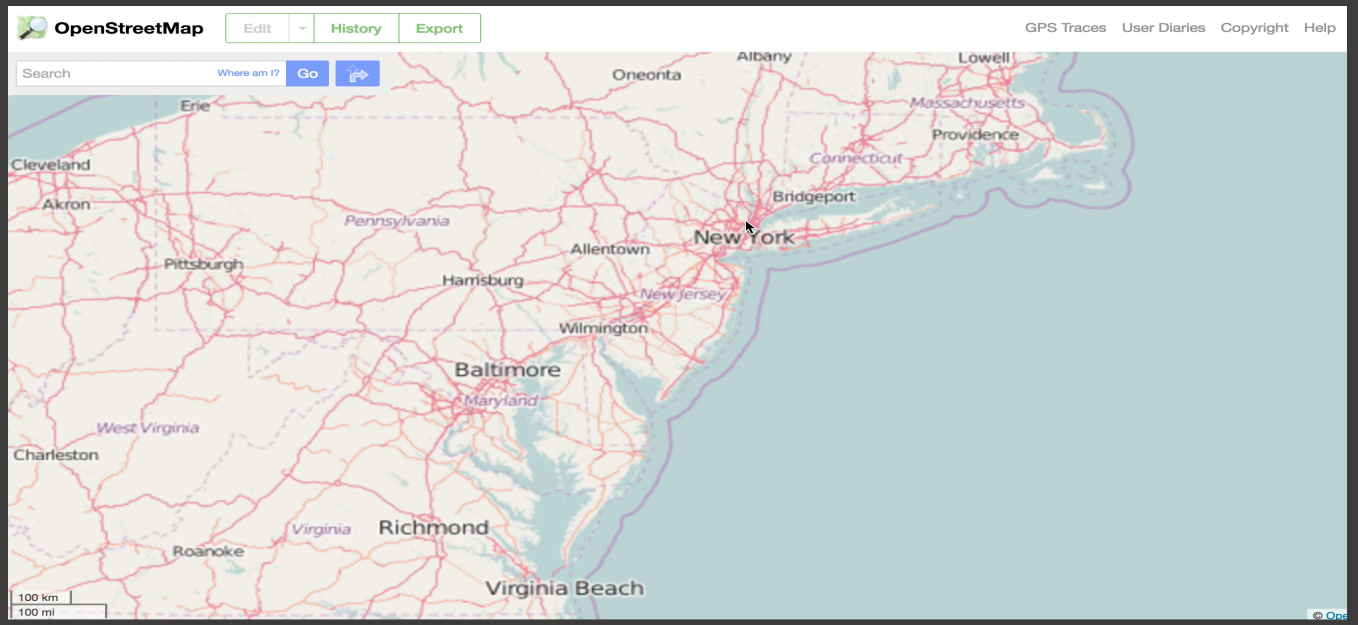
- Rendering a world with sound, powered by OpenStreetmaps.
- Empowering blind users to explore cities using 3D audio to improve independent mobility.

# Cities Unlocked Remote Device(*Unofficial*)



| Button | Experience |
|---|---|
| A | More Information \| **Toggle Navigation Sense (LP)** |
| B | Locate (where am I) |
| C | Orientate (micro) |
| D-PAD FRONT | Peek Front \| **Look Ahead (LP)** |
| D-PAD RIGHT | Peek Right |
| D-PAD BACK | Peek Back |
| D-PAD LEFT | Peek Left |
| D | Repeat |
| E | Hush |
| F | Listen |
| G | Vol - |
| H | Vol + |

# What is Openstreetmap?




- A PostgresSQL database often referred to as the Wikipedia of spatial data.
- Created at a time when maping data was expensive and restricted.
- Contributors use GPS-devices to record location data and publish to OSM.
- Initial dataset originated from Bing Aerial Imagery and Census Bureau sources.
- A free, editable, open map of the world.
- Crowdsourced dataset, made entirely by voluntary efforts of the online masses.
- Basically, unrestricted use and no data approval body.

# OSM powered Cities Unlocked Services

- Integrated Microsoft Cortana and Cognitive Service Suite.
- Automatic Callouts - Nearby points of interest, places and landmarks.
- Routing Engine - Turn-by-Turn Navigation.
- Forward / Reverse Geocoding(i.e. *Where Am I / Where is this*).
- Local Search – Users can be routed to any arbitrary place nearby.
- More information for nearby entities*(i.e. Central Park was built in…)*.
- Look Ahead – Calling out senses and intersections relative to heading.

MAPZEN

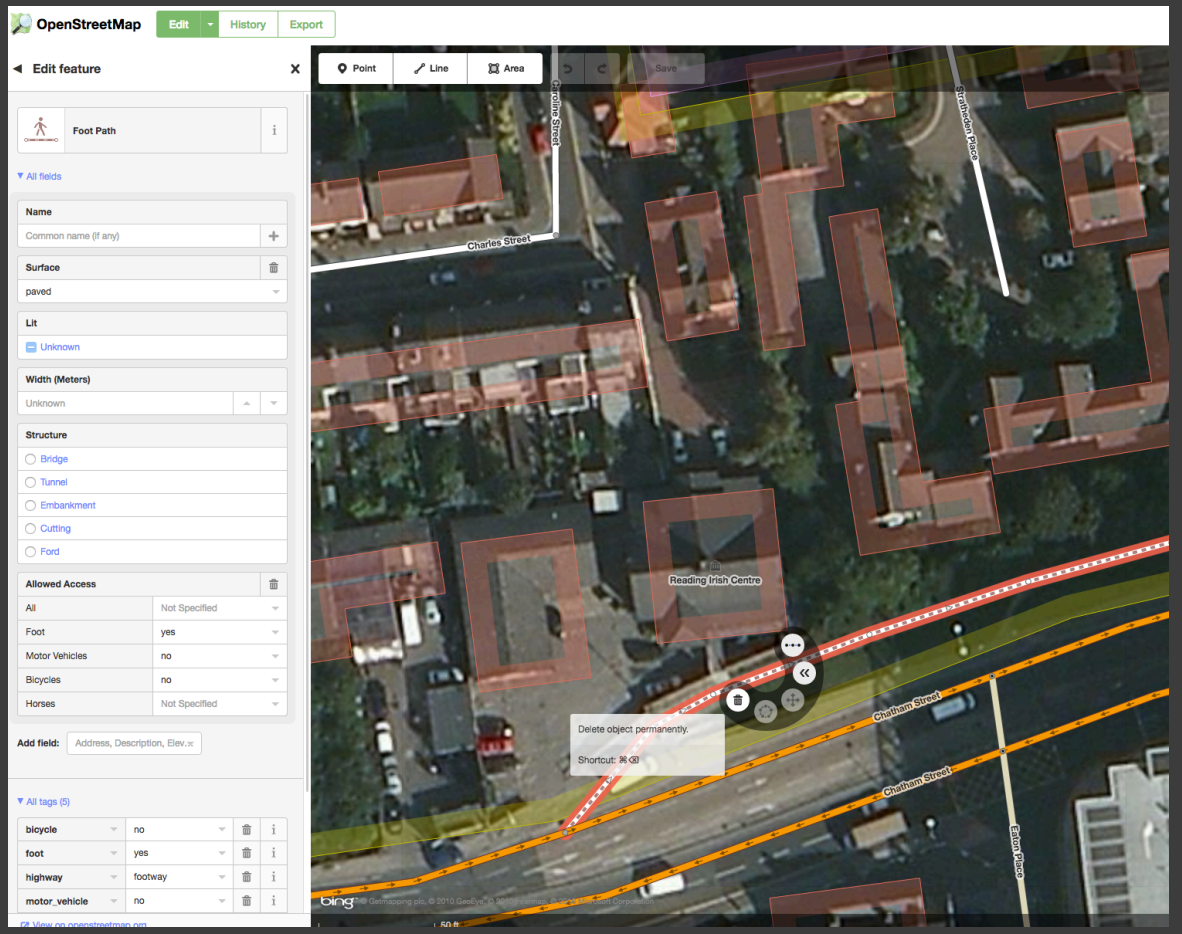
Cortana

OpenCage Data

# Cities Unlocked - Spatial Data Management

Explore – Publish location traces collected from your handheld device to OSM.


Editing– Web-based mapping feature editor(*Id*).



# OpenStreetmap Accessibility Features

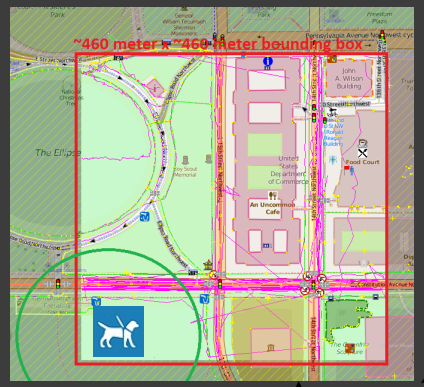
- OSM data accessibility for users that are deaf and/or blind.

---

**Accessibility Tags Cities Unlocked 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)

# Cities Unlocked– 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

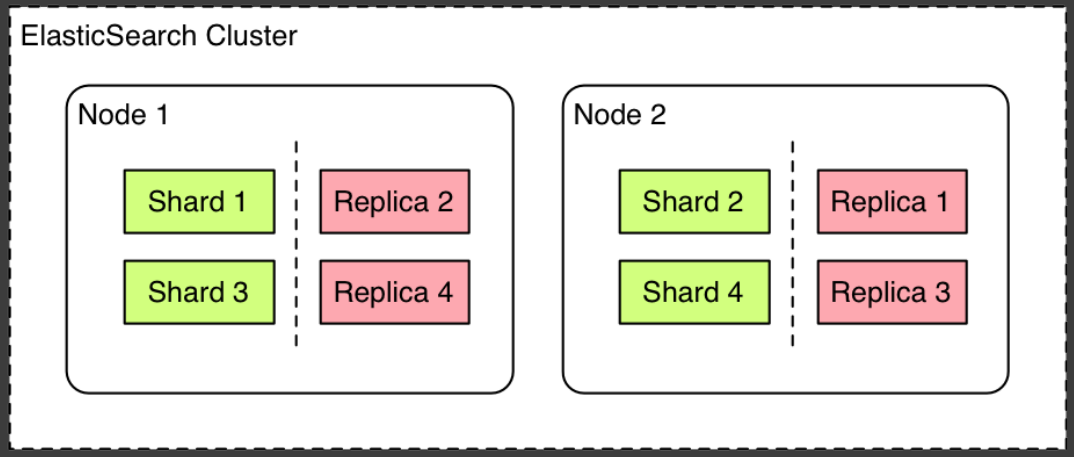
Wikipedia

OSM Global DataStore
PostgreSQL

# Near-Realtime Performance

- Content is re-rendered throughout user location / bearing changes.
- Responsiveness and availability of OSM data had to be near-realtime.
- Fault tolerant and load balanced.
- Efficiently supports complex geo shapes.
- Open sourcing a cache cluster PaaS of geo tiles using Docker / Elasticsearch.
- Powered by Azure.
- Tile data provided by OSM and Mapzen.
- Utilize OSM changeset info to only update tiles that were recently changed.

**Elastic Architecture Concept**



# Elasticsearch 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

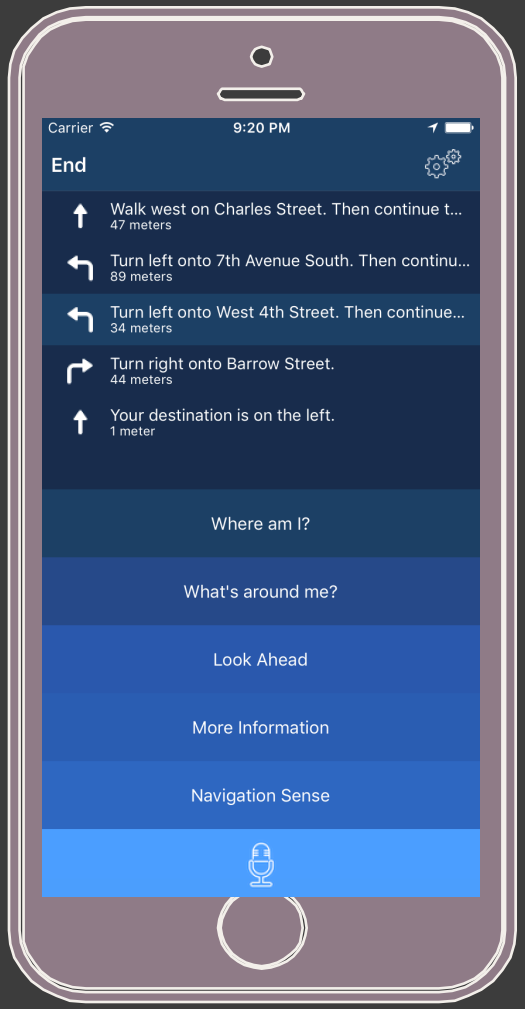
```
{
    "mappings": {
        "place": {
            "properties": {
                "id": {"type": "string"},
                "name": {"type": "string"},
                "type": {"type": "string"},
                "location": {
                    "type": "geo_point"
                },
                "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 walking paths.
- Steps, stairs and alleys are slightly avoided.

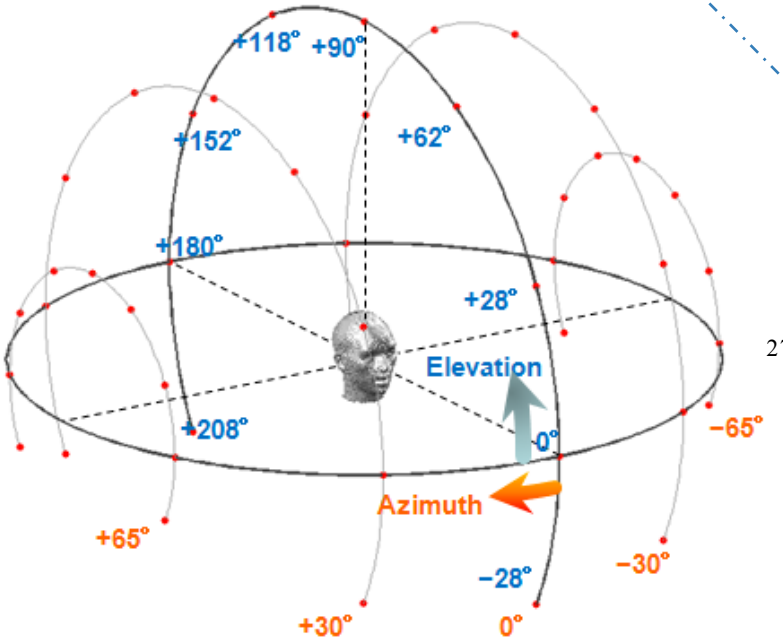


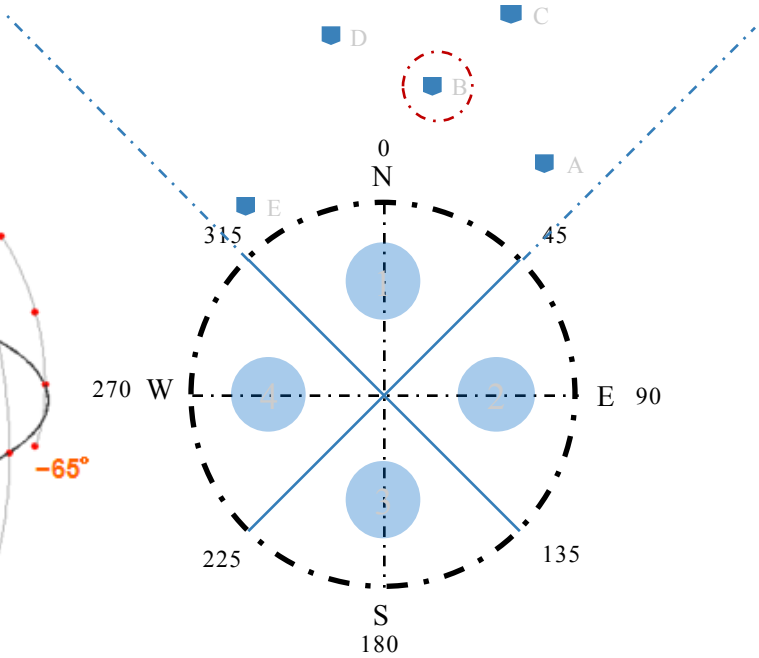

**RESPONSE**


```
    "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
```

# Spatial Audio

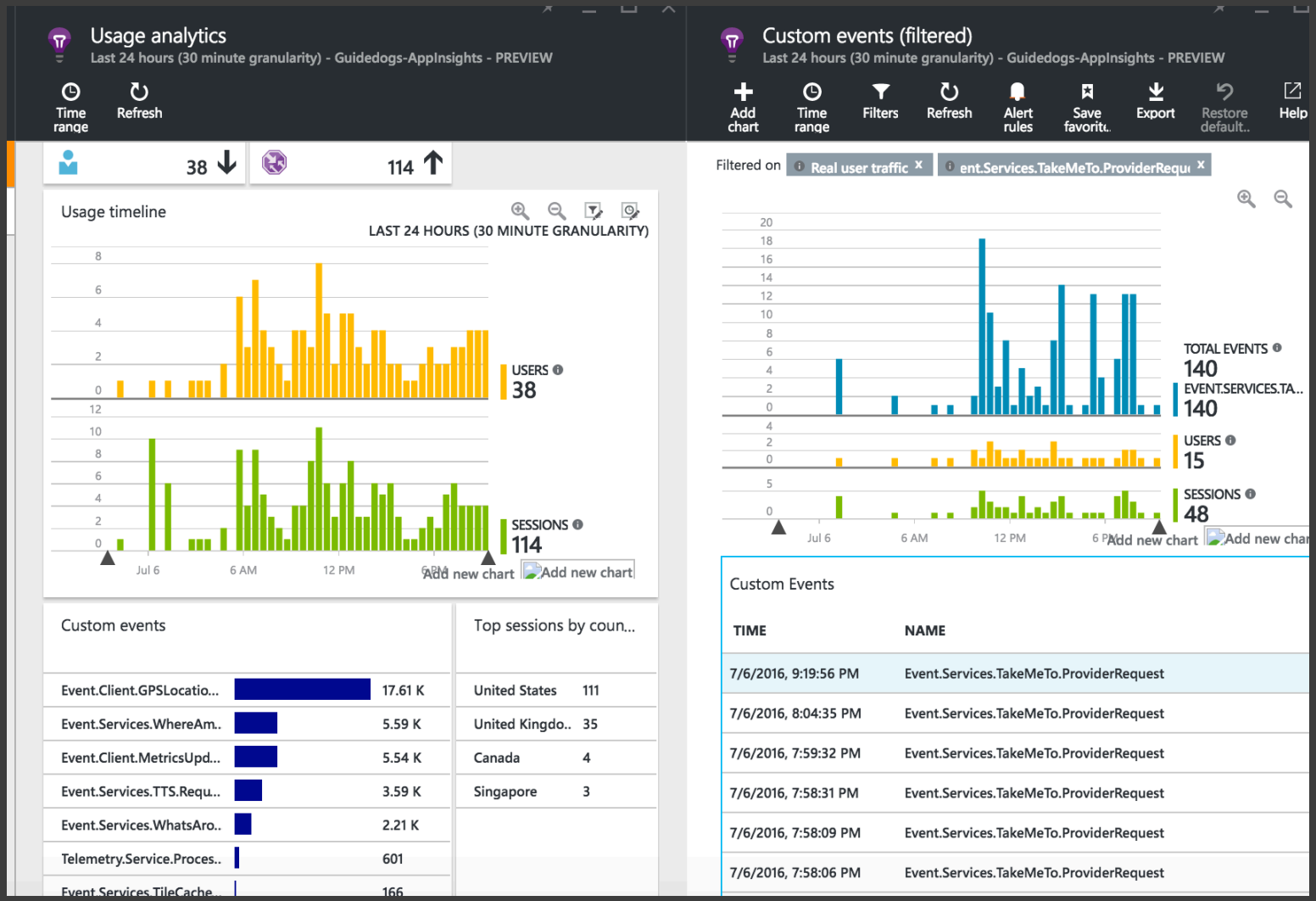
- Microsoft's HRTF 3-D audio engine gives OSM spatial data a voice, and enables users to build a mental map of the area.
- Spatial sound cues are used for navigation, points of interest and journey details alongside regular GPS instructions.
- Cortana services translate text results to an audio stream.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



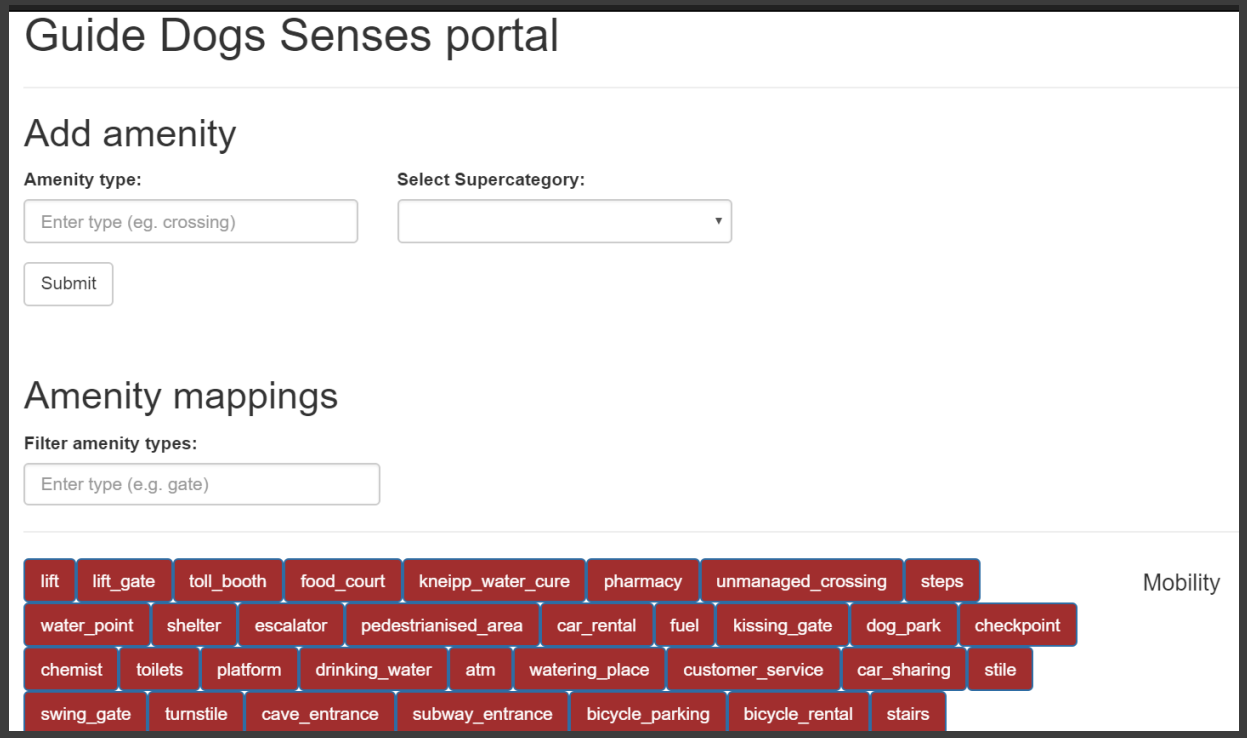
# Production Quality Oversight / Telemetry

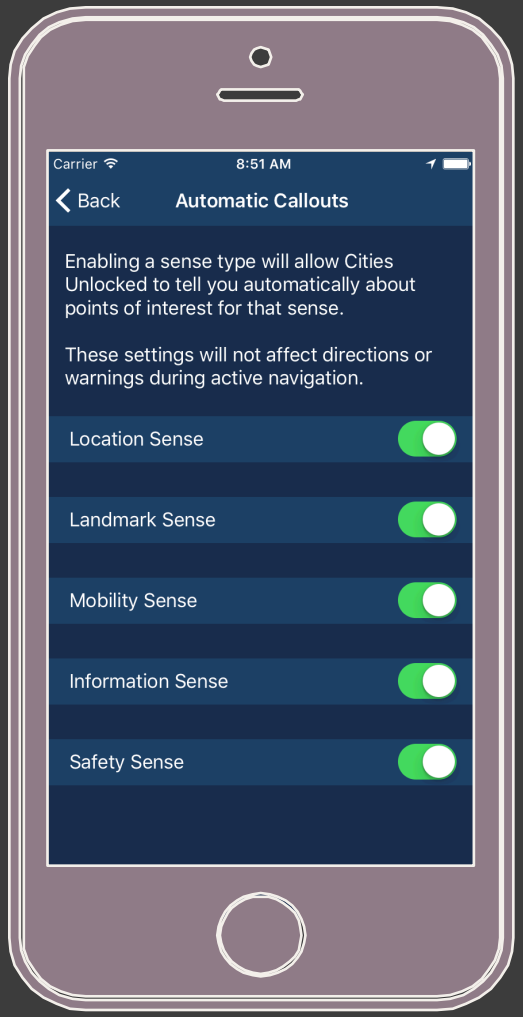
- Application Insights - Monitor / measure service performance and exceptions.
- Web Tests - Benchmark microservice(s) performance tests with alert escalation.



# Managing Automatic Callouts

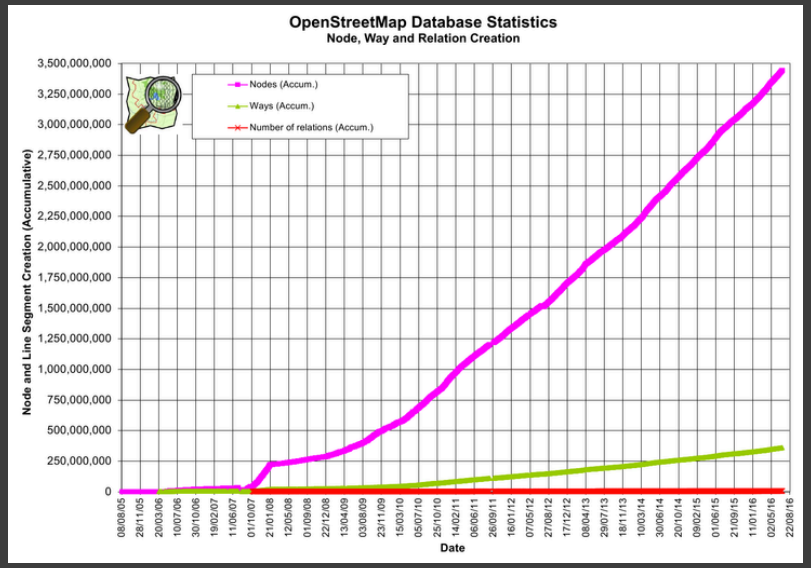
Points of Interest Classification: We built a web tool that enables UX designers to manage the classification of OpenStreetmap amenity types.

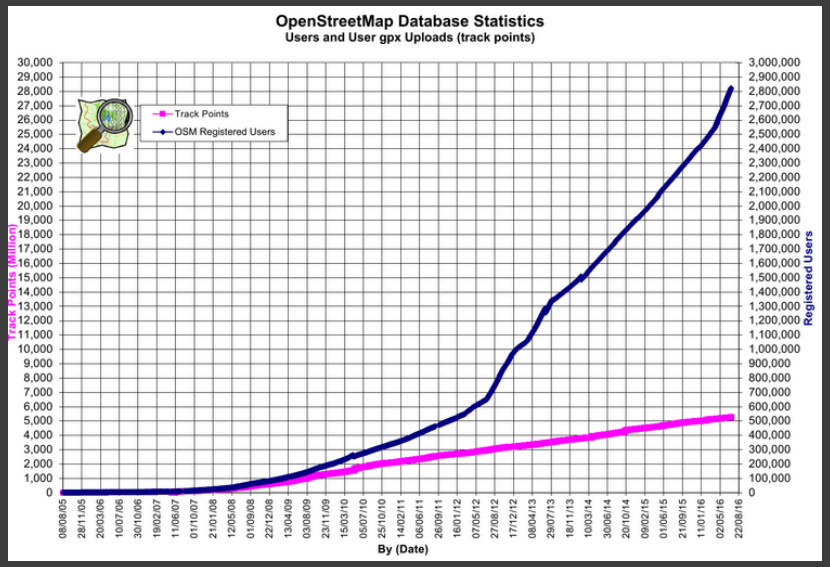


# Why OSM as opposed to other map providers

- ~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.
- Supports offline data download and navigation.
- Growing community that contribute mapping data to OSM daily.



# OSM Challenges

- No official data schema standardization.
- Mapped walking paths are sparse in OpenStreetmap.
- OpenStreetmap is subject to contributor biases.
- Retrieving OSM data requires more technical skill than retrieval.
- Detail and precision across OSM varies from region to region.
- No SLA with resolving systematic OSM data quality exception breaks.

# Thank You

ERIK SCHLEGEL


@erikschlegel1
https://github.com/erikschlegel