Offline Navigation & Maps Engine from scratch

Data, Processing, Fetching, Rendering, Search and Navigation

Evgeny Karpov - Software Engineering @ kMaps LLC
@eukarpov kmaps.co

When is offline needed?

Traveling
Limited data plans
Poor network coverage
Overloaded networks
Self-driven cars
Connected cars
Privacy
Disasters

and many other reasons
and domains

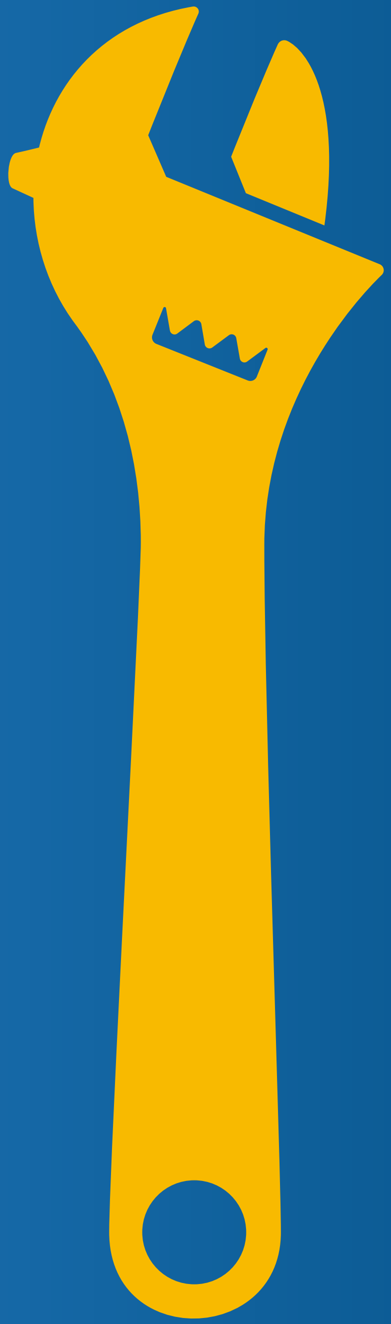
What does “from scratch” really mean?

- No SQL, NON-SQL and GEO oriented Databases
- No frameworks, APIs and 3rd party components
- No graphical frameworks
- No multi-languages libraries

```
// OfflineNavigationApp.cpp

int main(int argc, const char * argv[]) {

    // Type your code here

     return 0;
}
```

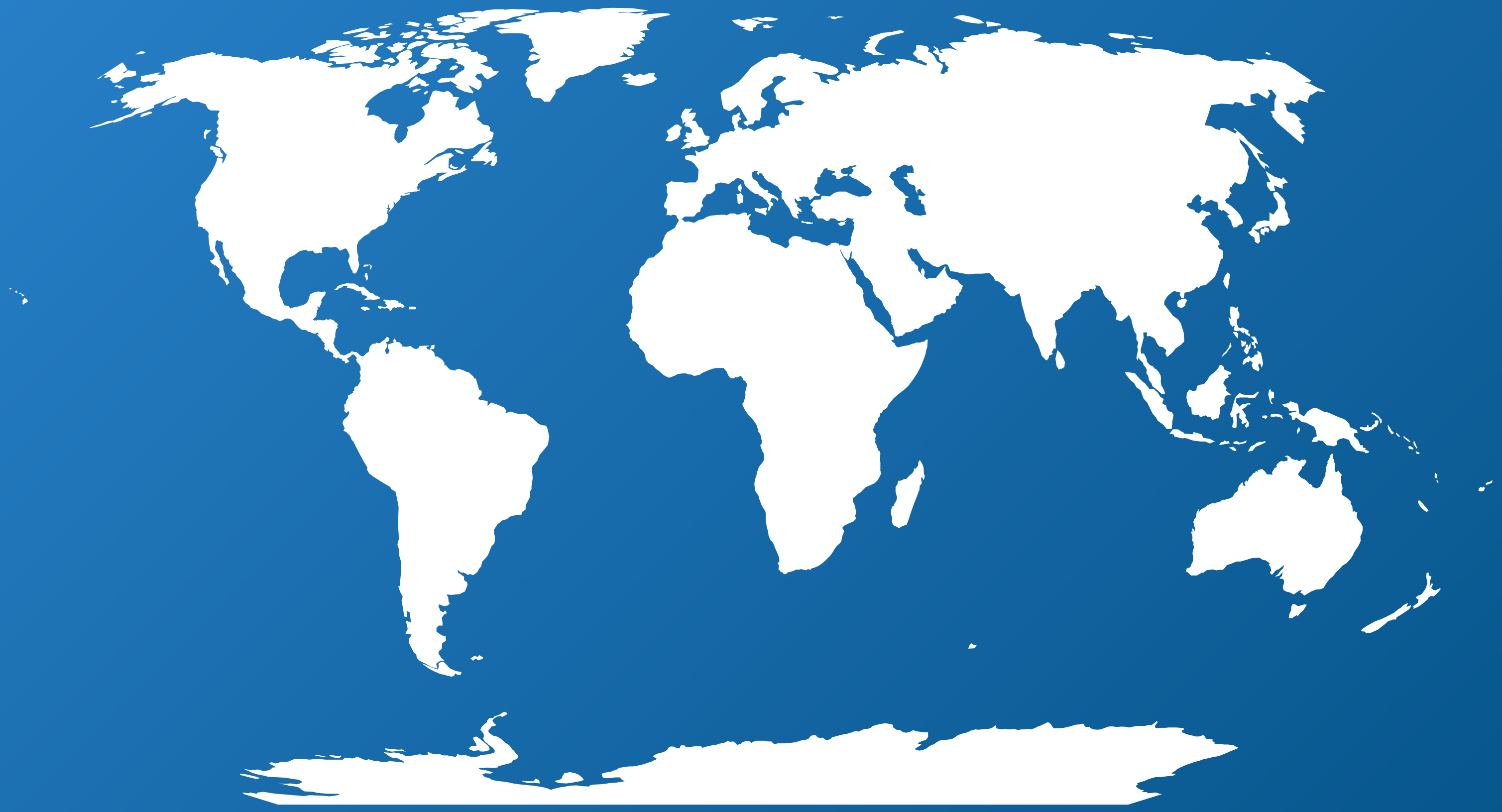
What do we need to implement?

- Getting and Processing Data
- Building Vector Tile Storage
- Hosting, Fetching and Caching tiles
- Fetch function and Rendering tiles
- Indexing tiles and search
- Building routing tiles
- Routing from point to point
- Turn by turn navigation



Where can I get data of the entire world?

- Open Street Map
- The entire world
- 400M+ Points of Interest
- Navigation metadata



How to process data without any database?

- Converting to binary representation
- Points
- Geometry
- Relations
- Metadata

1100101001010011100101
0111000001

How to build vector tile storage?

- Understanding algorithms
- kdTree, quick sort, partitioning, balancing tree
- Broken polygons should be fixed
- Triangulation can be done on server side
- Cutting tiles
- Building tile pyramid
- Compressing tiles

kdTree

Vector Tiles

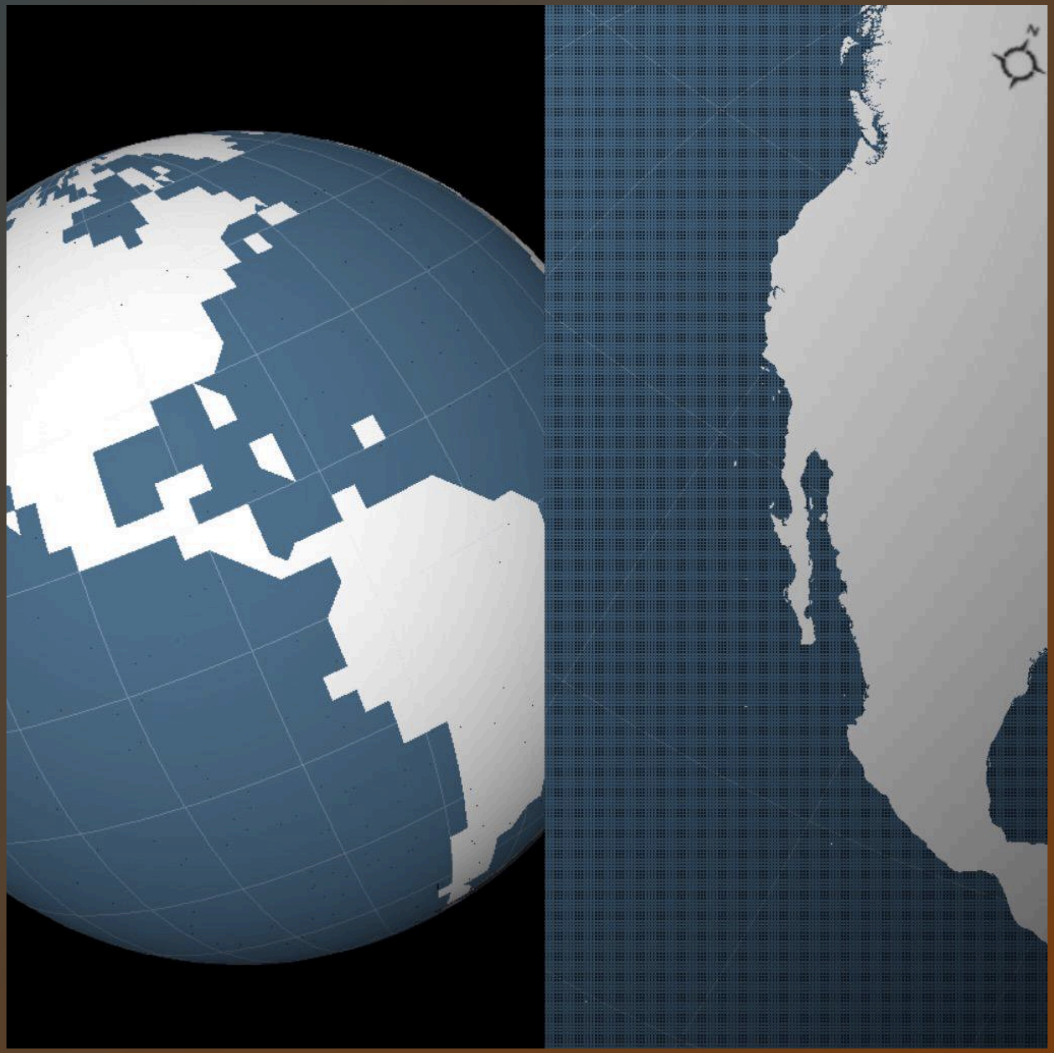
How to host the tile storage and fetch data on the client side?

- Simple, cheap, fast and friendly SSD hosting (for example Digital Ocean)
- Simple web server (NGINX)
- No backend
- Privacy
- Caching tile storage kdTree on client side
- Caching tiles by request with HTTP range request

Hosting
Tile storage file

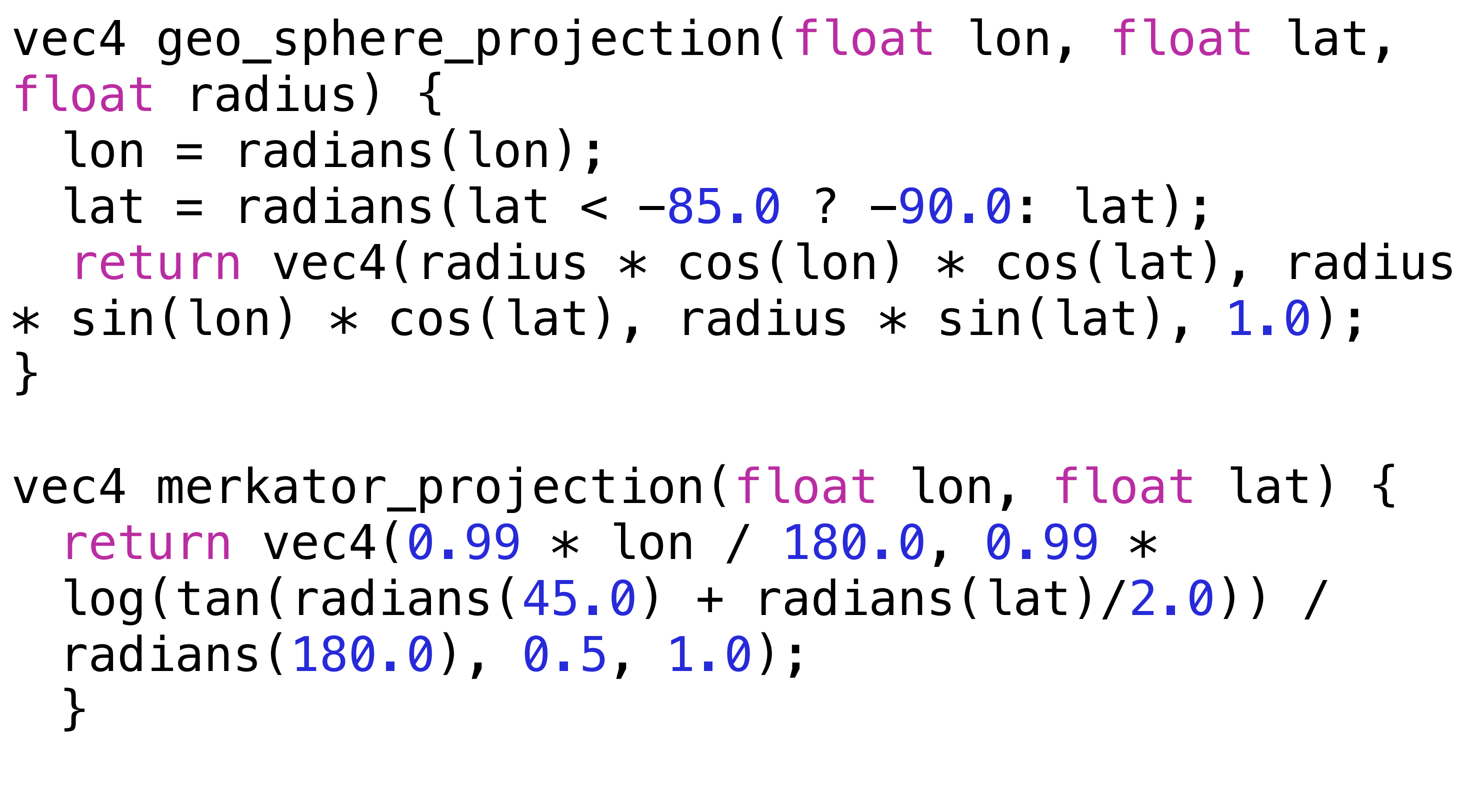
OK. I have data on the client side. But how to render?

- OpenGL ES >= 2.0
- Power of shaders
- Implementing Rendering Loop
- Fetching required tiles
- Converting vector tiles to GPU tiles
- Triangulation
- Managing tiles in GPU



I have heard of shaders before. But what does it look like and how do I render POIs?

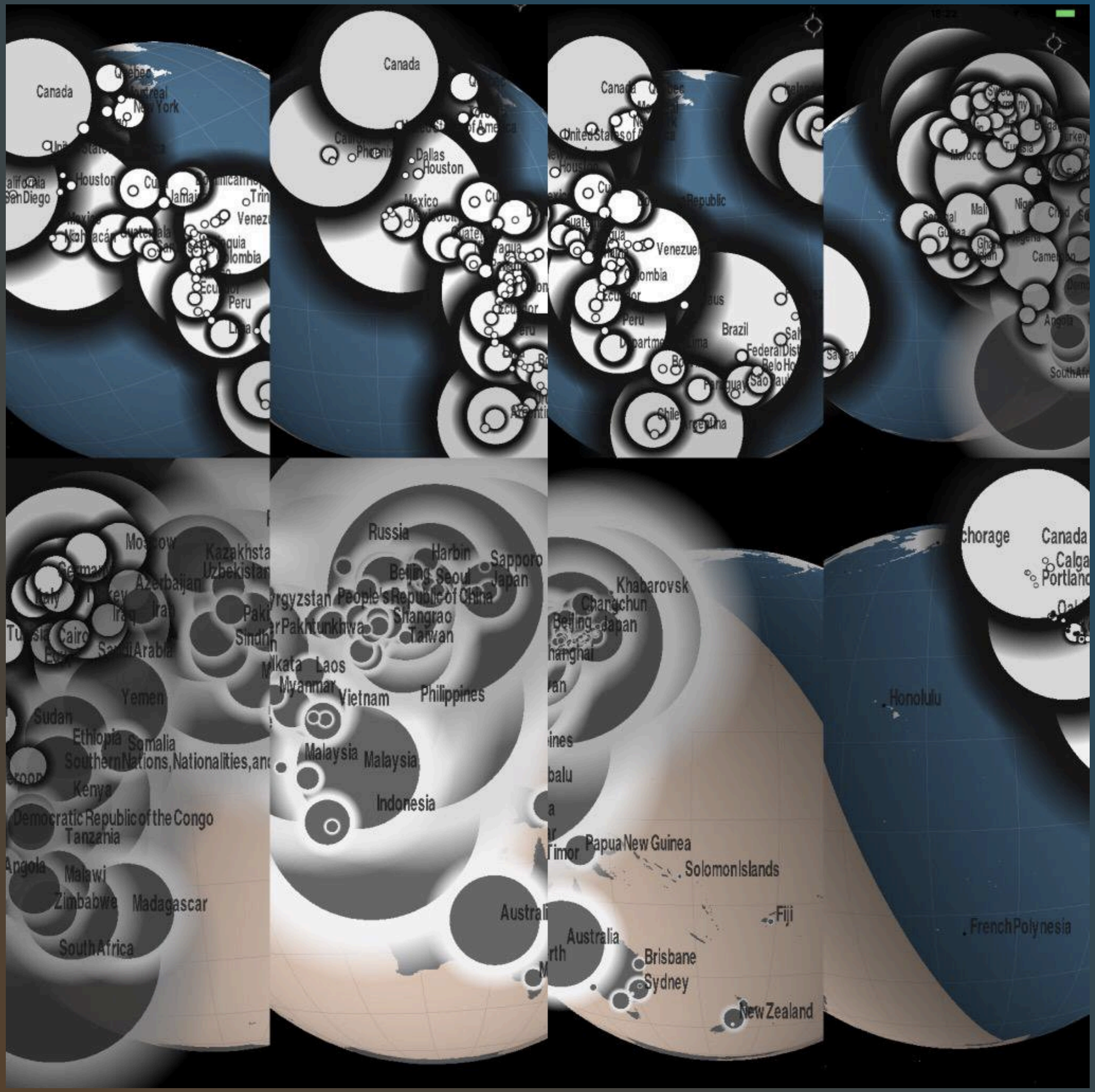
- Projection shaders
- Writing shaders for different type of objects
- Separate task for fetching POIs
- Triangulating fonts
- Rendering POIs



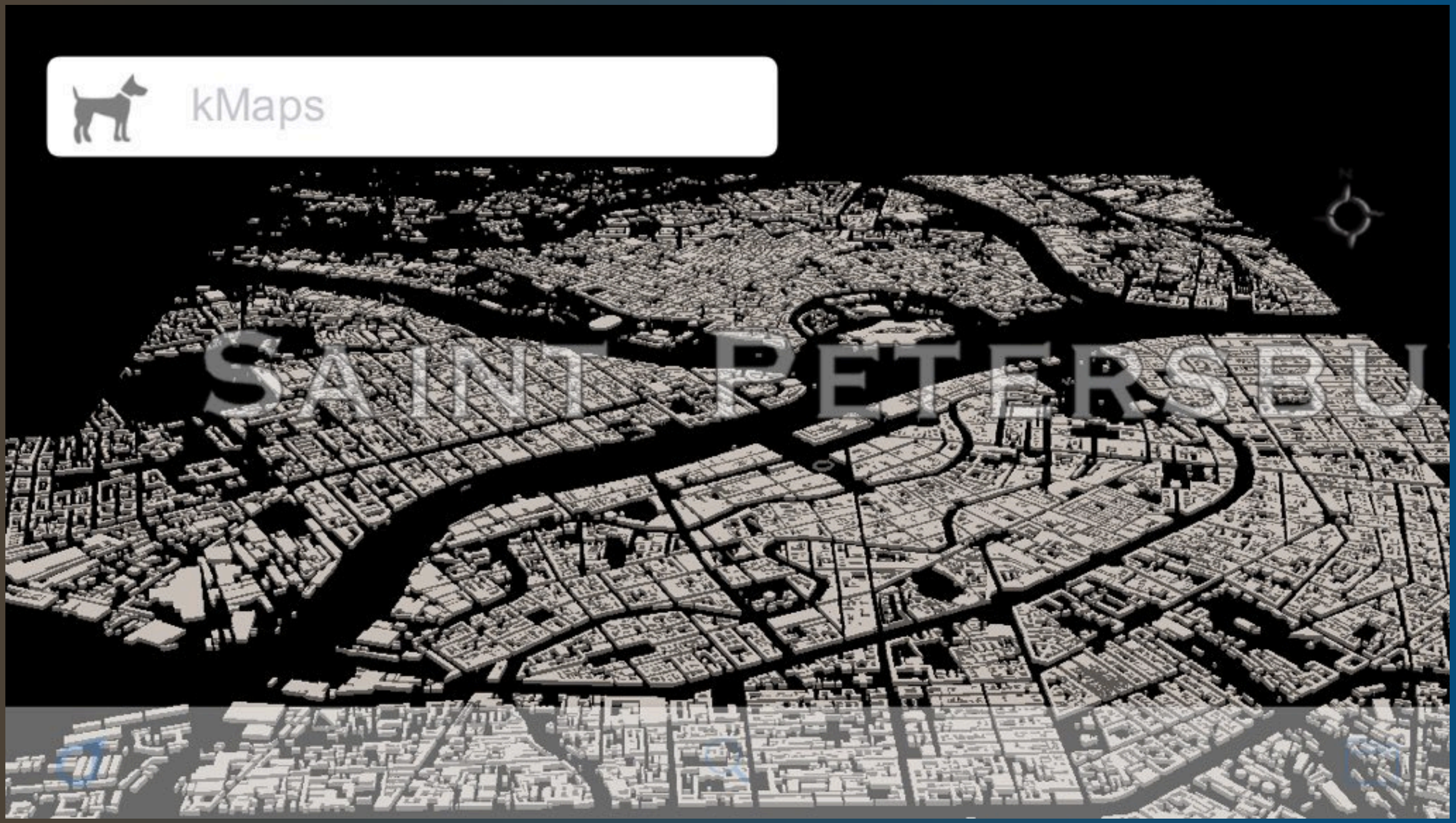
```
vec4 geo_sphere_projection(float lon, float lat,
float radius) {
  lon = radians(lon);
  lat = radians(lat < -85.0 ? -90.0: lat);
   return vec4(radius * cos(lon) * cos(lat), radius
* sin(lon) * cos(lat), radius * sin(lat), 1.0);
}

vec4 merkator_projection(float lon, float lat) {
  return vec4(0.99 * lon / 180.0, 0.99 *
  log(tan(radians(45.0) + radians(lat)/2.0)) /
  radians(180.0), 0.5, 1.0);
  }
```

Is there any best practice to render with OpenGL?

- Limitation of vertex on different platforms
- Using Vertex Array Objects, Vertex Buffer Objects, Index Array Objects
- Exclude branches and loops in shaders
- Measuring Vertex and Fragment performance



How is the user interface integrated with rendering?



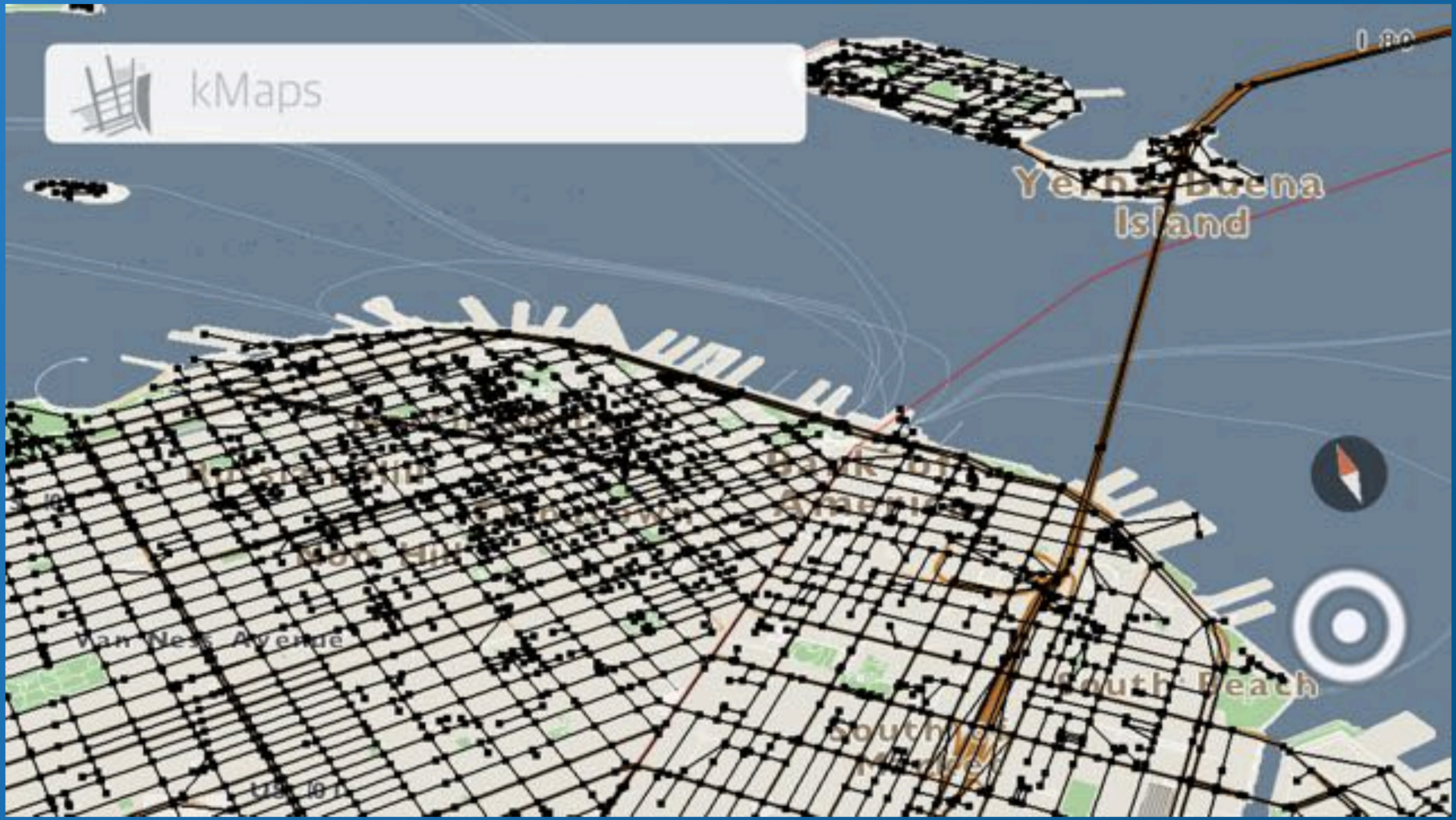
- Moving the map
- Rotating the map
- Zooming In/OUT
- Changing angle
- Accessing a POI by user interaction (click on the map)
- Standard map gestures

Maps without search are almost useless. How to search?

- Using UTF8 to support multiple languages
- Unicode normalization and lowercase transformation
- Indexing tile metadata on the fly
- Merging Inverted Indexes
- Boolean search

Building routing tiles

- Building routing tiles on the fly base on vector tiles
- Graph simplification
- Routing metadata
- Graph serialization



Routing tiles are done. How to implement simple routing?

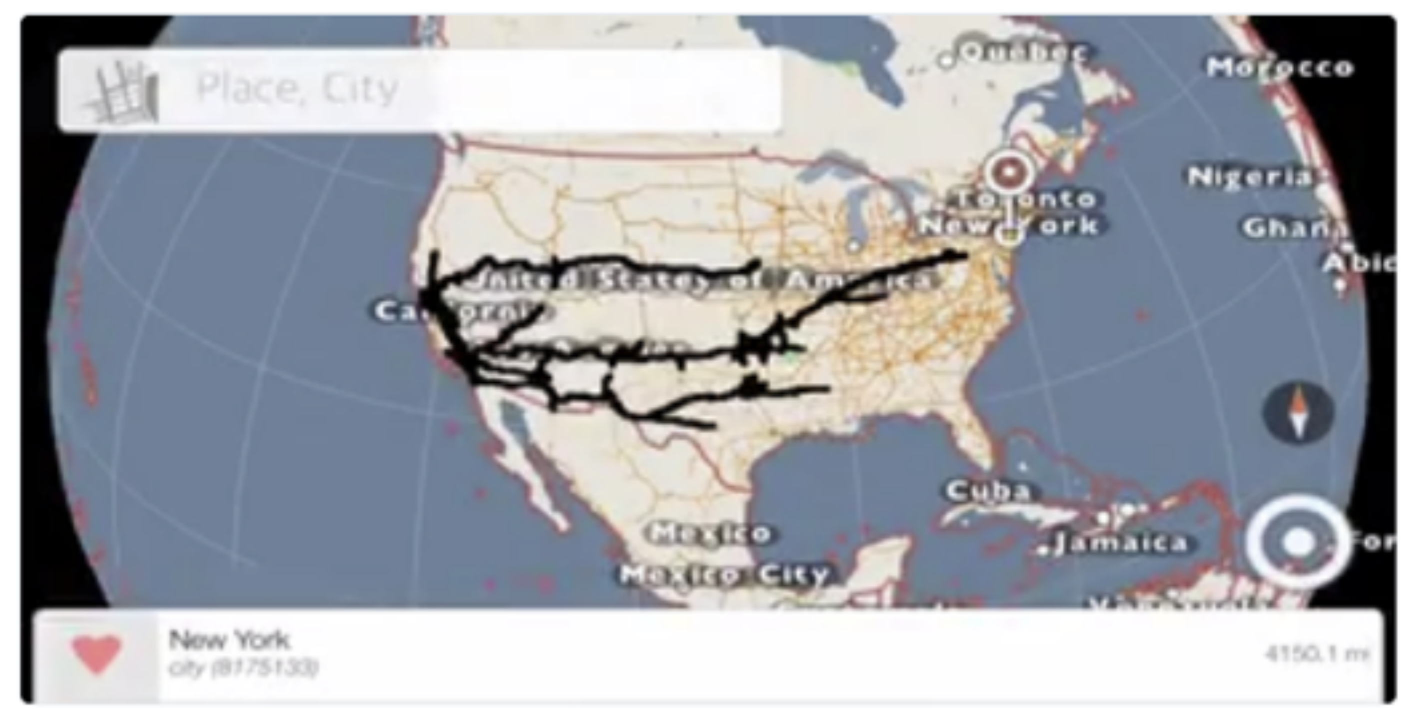
- Finding start and end nodes for routing

- Using classical A* algorithm

- Art of writing heuristic functions

- Performance optimization for fetching routing tiles

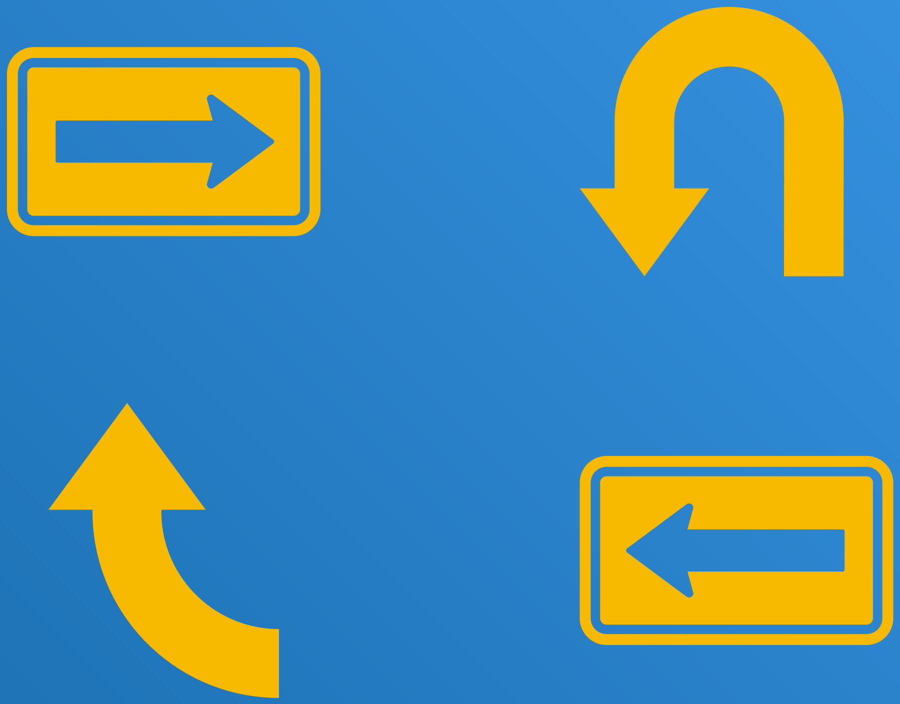
- Reconstruction of real path



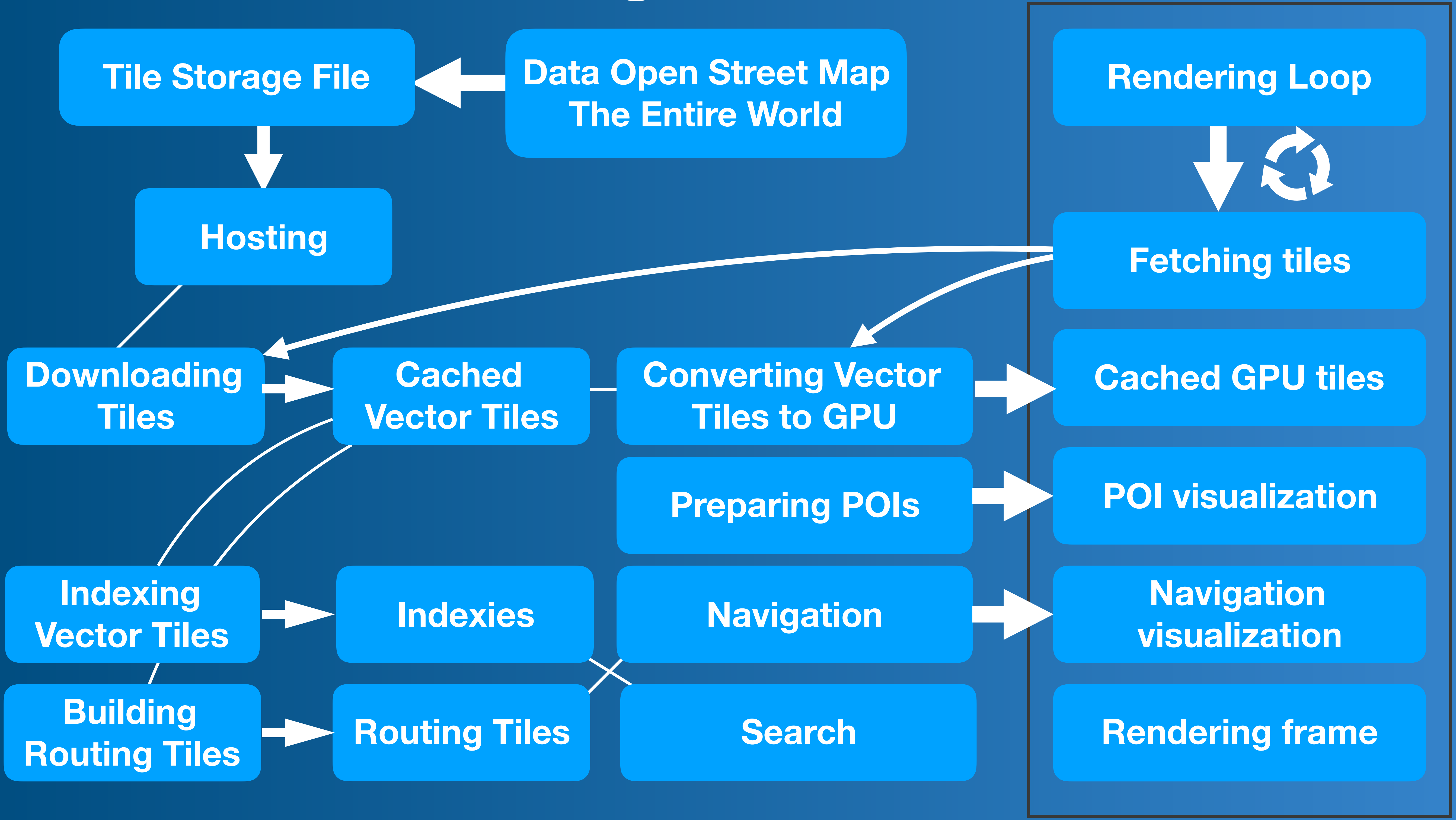
New York
city (8175133)
4150.1 mi

Now we need to navigate user to destination

- Calculating all turns and branches on the path
- Informing user in advance
- Using location to detect when the route should be recalculated



Got it! How does it look all together?



Tile Storage File

Data Open Street Map
The Entire World

Rendering Loop

Hosting

Fetching tiles

Downloading Tiles

Cached Vector Tiles

Converting Vector Tiles to GPU

Cached GPU tiles

Preparing POIs

POI visualization

Indexing Vector Tiles

Indexies

Navigation

Navigation visualization

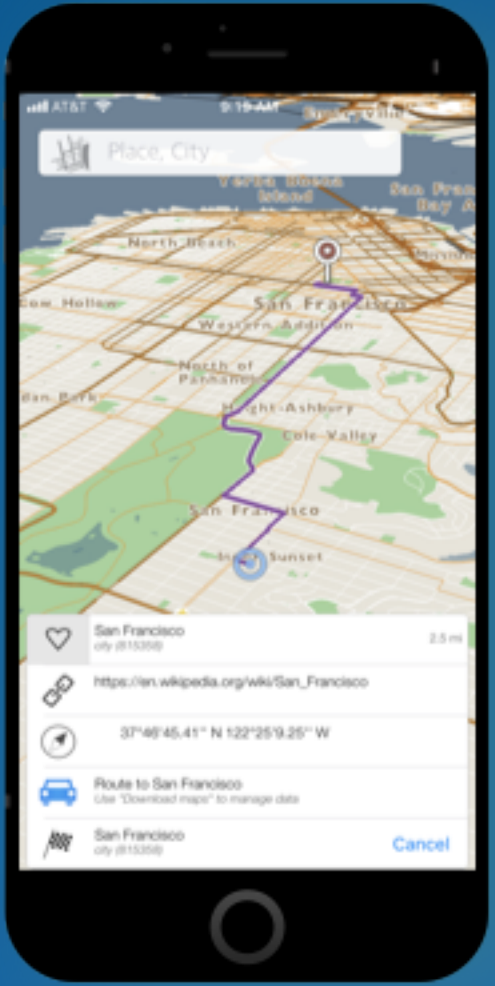
Building Routing Tiles

Routing Tiles

Search

Rendering frame

We did something similar to this pipeline. You can as well!



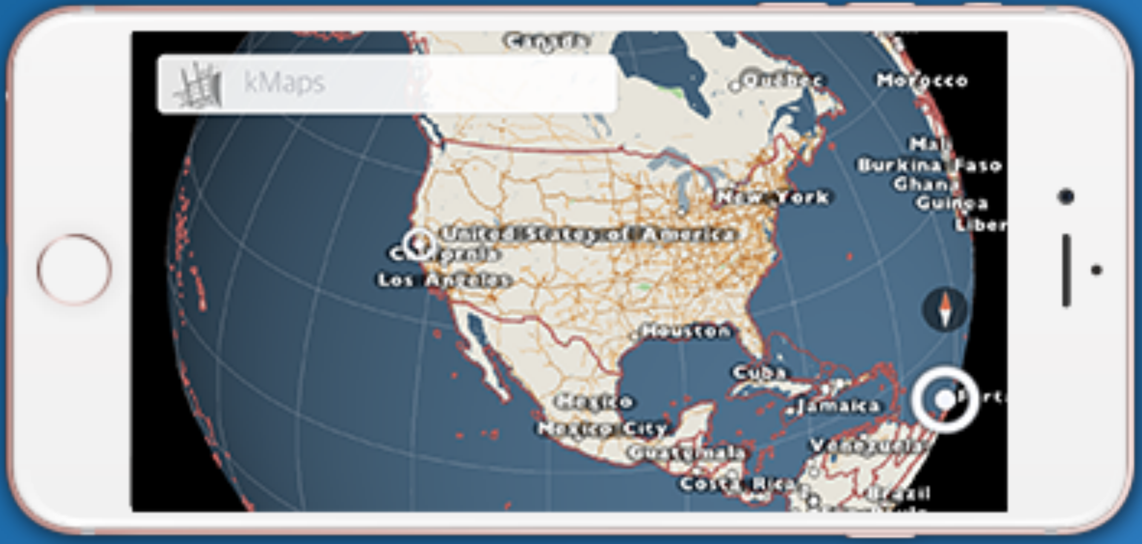
San Francisco
city (815358)
2.5 mi

https://en.wikipedia.org/wiki/San_Francisco

37°46'45.41'' N 122°25'9.25'' W

Route to San Francisco
Use "Download maps" to manage data

San Francisco
city (815358)
Cancel



Q & A
TIME

Offline Navigation & Maps Engine
from scratch
@eukarpov
kmaps.co