

A decorative background featuring a cluster of grey circles of various sizes on the left side, with a larger, irregular grey splash shape extending from them towards the center. The rest of the background is plain white.

**Map**nificent



# Mapumental



Public transport arriving at 9:00

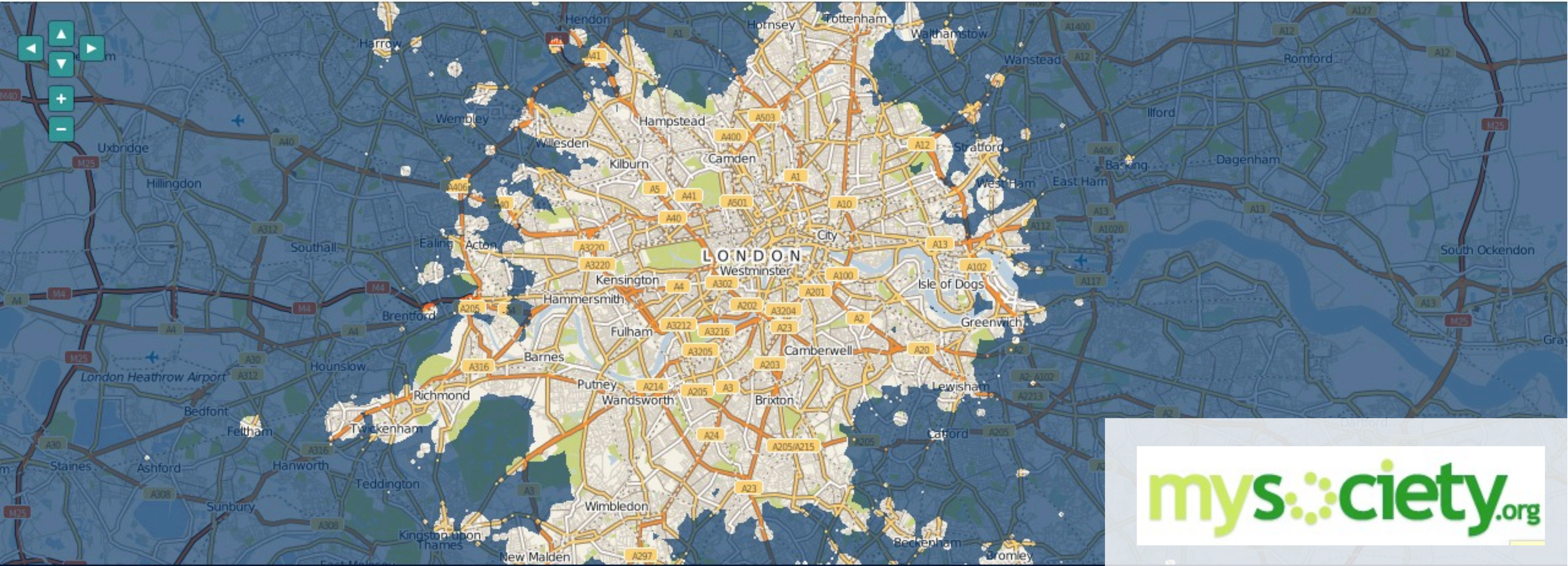
Departing at 8:20

House price (average)

Any price

Scenicness

Score: 1



mysociety.org

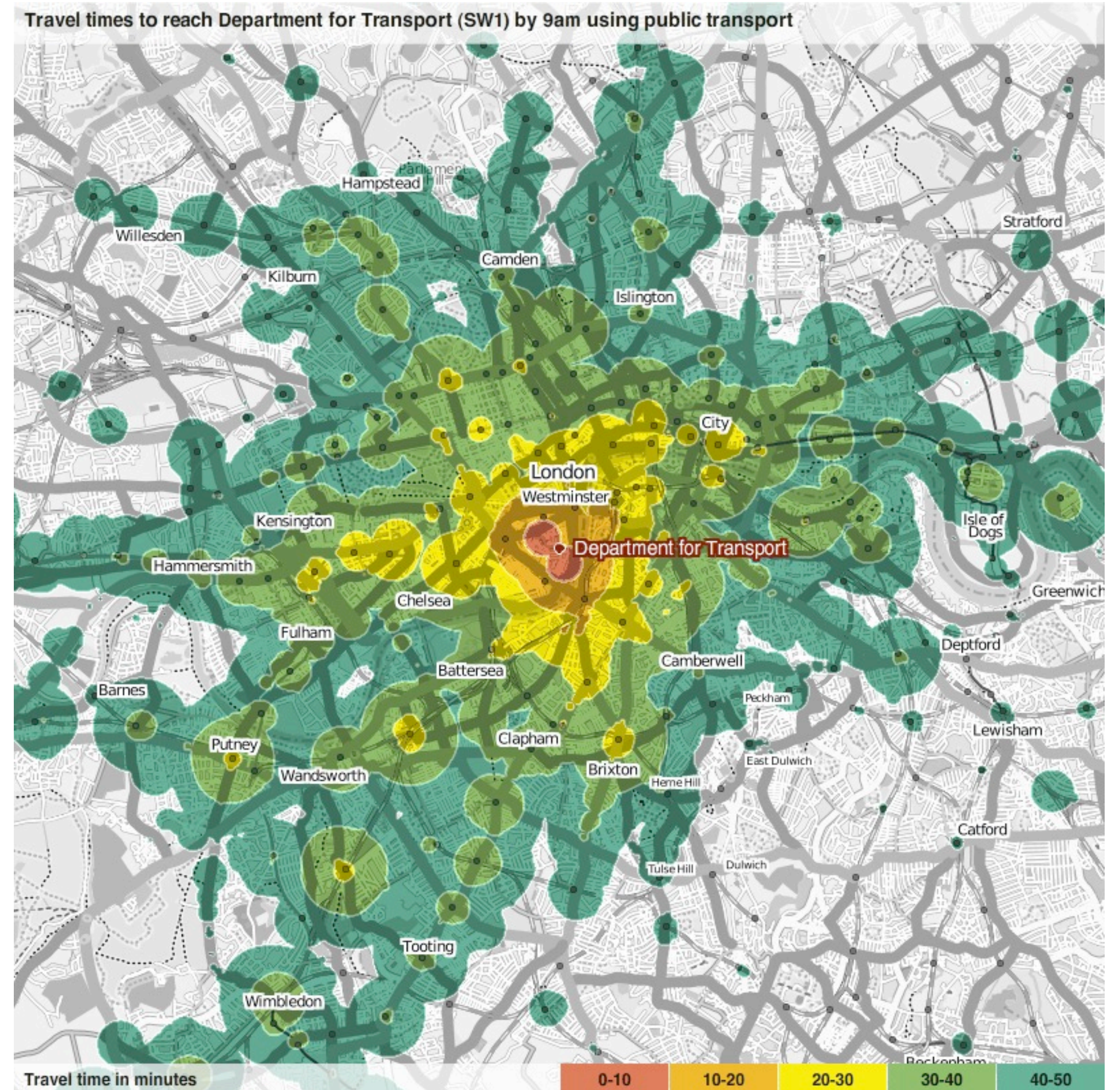
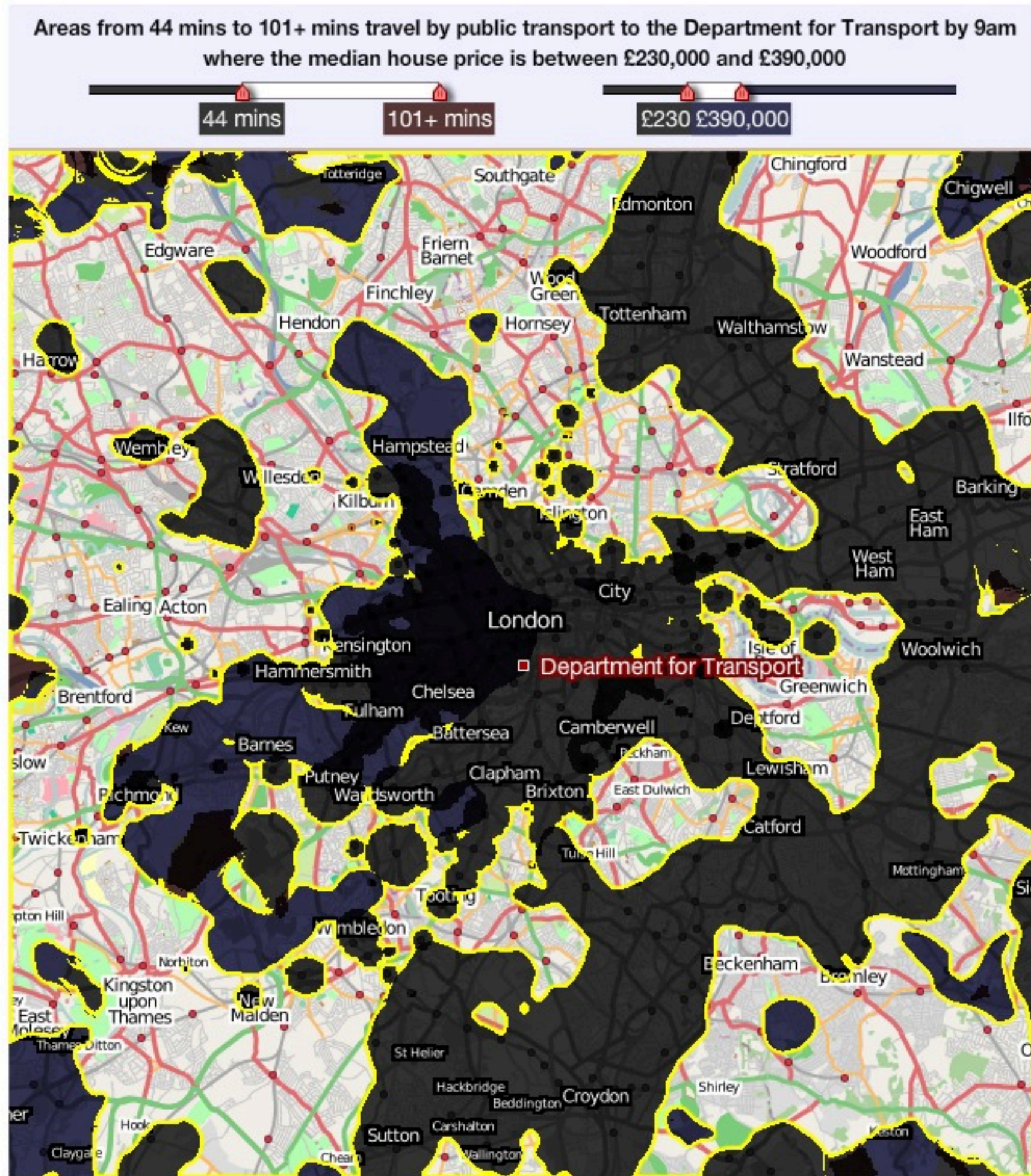
New map? GB postcode:  Go ( SE14 6NW  Go )

You have no invites | [Help](#) | [Stamen](#) | [mysociety.org](#)

<http://mapumental.channel4.com>



# MySociety Travel Maps



<http://www.mysociety.org/2007/more-travel-maps/>



# Two Challenges

1. How to draw on the map?

2. What to draw on the map?

# How to draw on the map?

*Google Maps Drawing API (e.g. GPolygon)?*

Doesn't scale. Lots of small canvas DOM objects.

*Try some SVG?*

May scale better, but doesn't offer some capabilities.

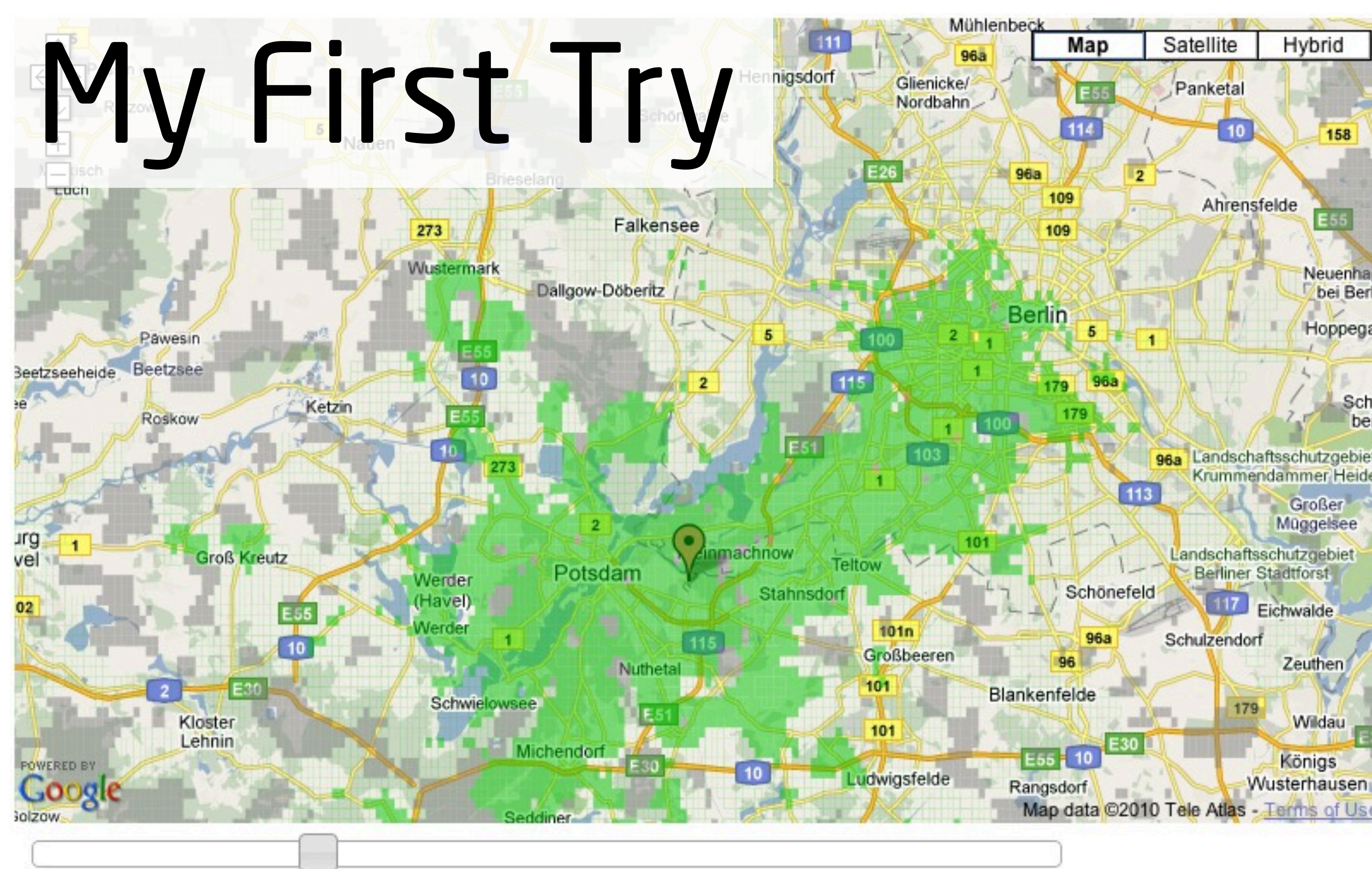
*Flash to the rescue?*

No, thank you. There must be another way.

HTML5 Canvas it is.



# My First Try



Grid of 121 x 322  
= 38962 requests  
(took 24 hours).

Aus dem grünen Bereich ist das HPI in **55** Minuten erreichbar.

<http://myhpi.de/~stefan.wehrmeyer/hpidistance/>

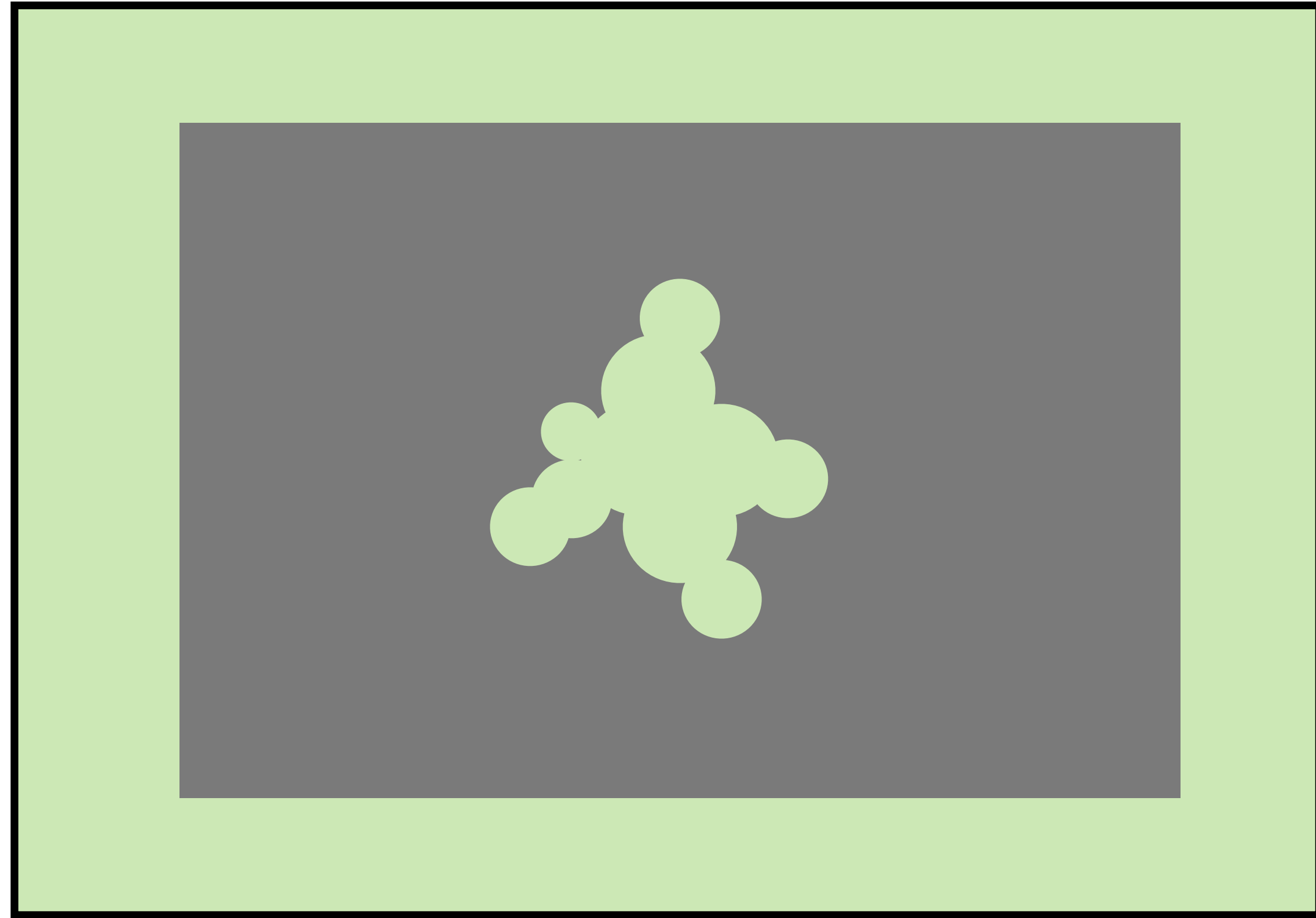
Fixed Destination: Hasso Plattner Institute, Potsdam.

Canvas as overlay inside Google Maps (thanks to ELabel).

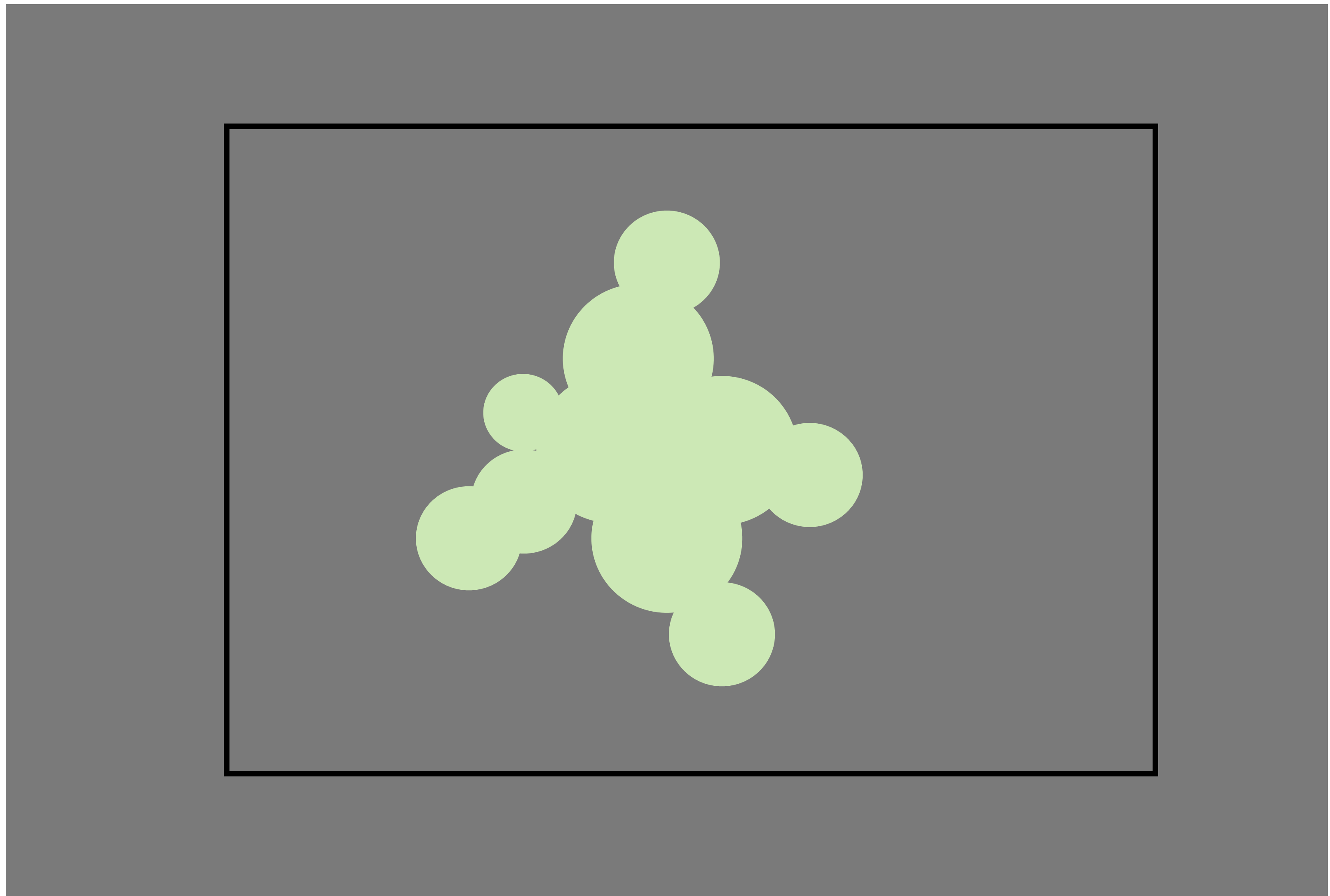
<http://econym.org.uk/gmap/label.htm>



# Zoom: naive approach.

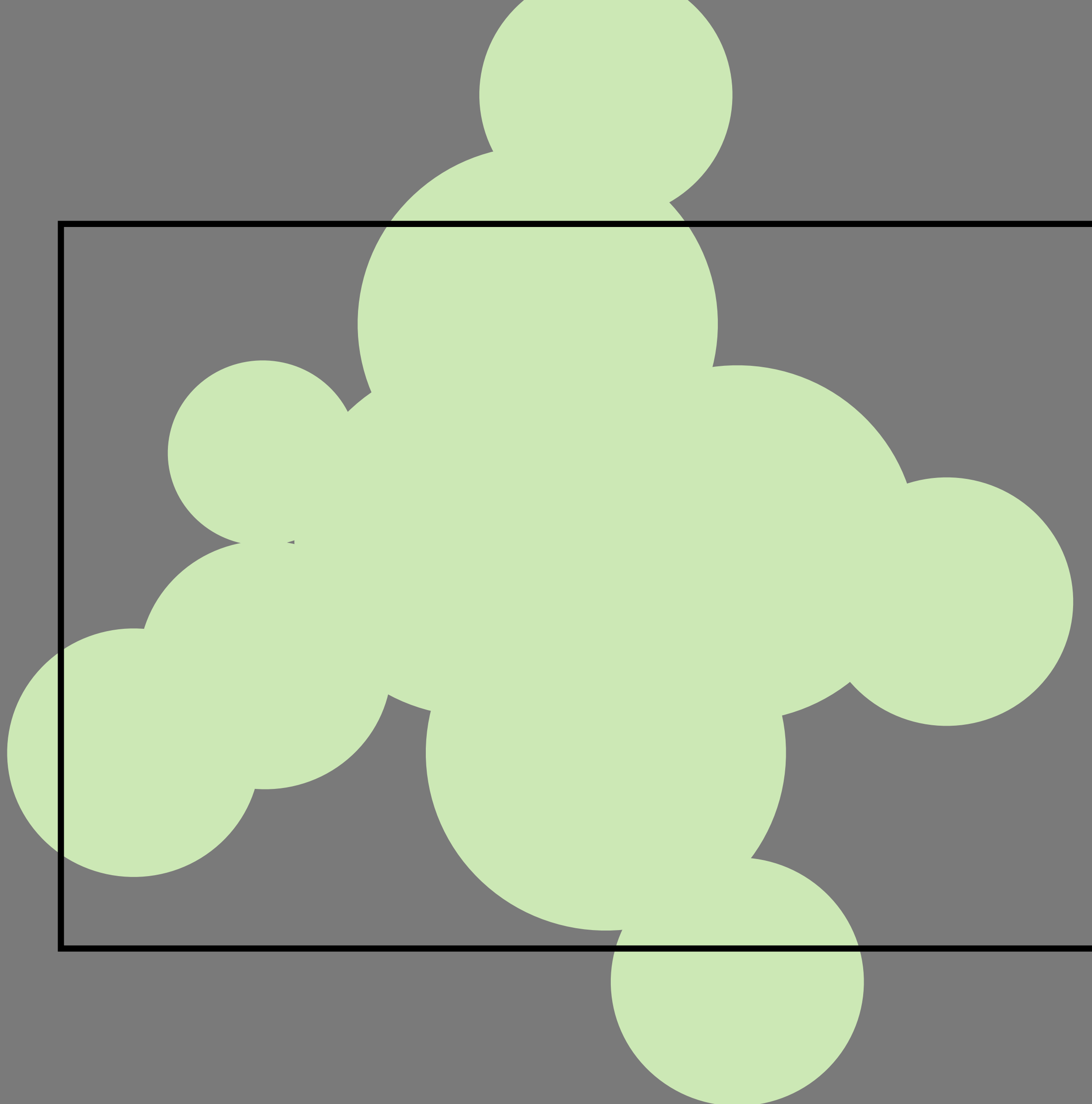


Zoom-Level 10



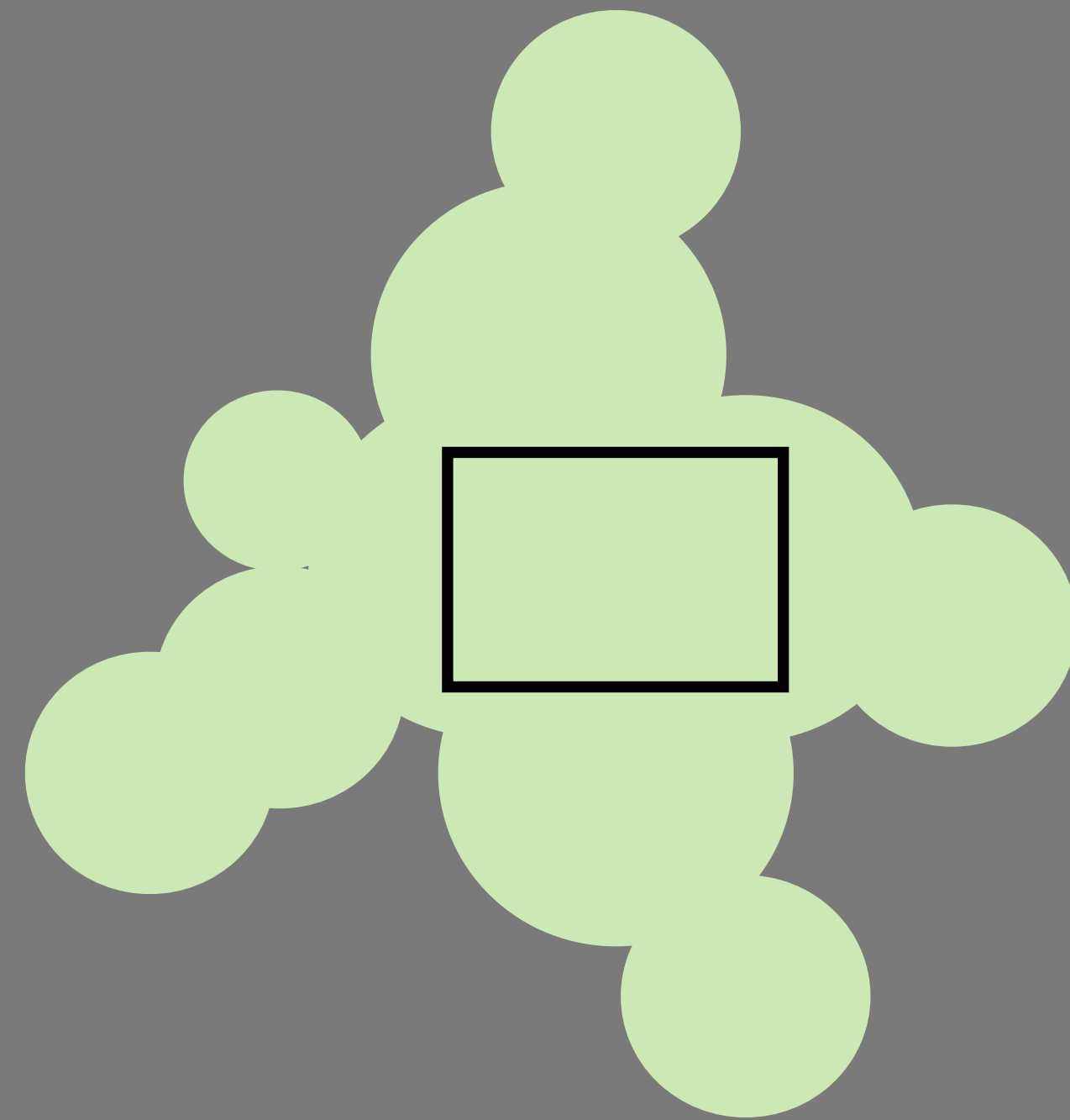
Zoom-Level 11





Zoom-Level 12





Zoom-Level 13





Zoom-Level 14?

Canvas doesn't want to draw  
37,399 x 22,984 (=859,578,616) pixels.

Canvas can't grow beyond certain dimensions.



# Yes, we canvas!

Resize canvas only to max.  
(3 x map width) x (3 x map height)



- allows zooming
- allows panning
- reposition when needed
- user doesn't notice
- good performance



# What to draw on the map?

Mapumental approach  
(optimised server side routing and caching)?

Pre-calculated maps for every post code in Berlin?

Lots of calls to a Travel Planner Service?

Screw it, I do the routing myself.  
In the Browser.
































































(How hard can it really be?)



# Getting the data

Querying route from  
first station  
to  
last station  
of ~200 lines.  
Pretty cheap.

Bonus: latitude  
and longitude of  
station hidden in  
this link.

| Karte   | Adresse/Haltestelle  | Zeit                   | Bemerkungen   |
|---|--|------------------------|---|
|     |  S+U Rathaus Steglitz (Berlin)           | ab: 09:35              |  <br>Blindenleitstreifen am Bahnsteig U-Bahn  |
|   |  |                        |  U9<br>Richtung: U Osloer Str. (Berlin) [1] [2]  |
|    |  U Schloßstr. (Berlin)                  | an: 09:36<br>ab: 09:36 |    |
|    |  U Walther-Schreiber-Platz (Berlin)     | an: 09:37<br>ab: 09:37 |     |
|    |  U Friedrich-Wilhelm-Platz (Berlin)     | an: 09:38<br>ab: 09:38 |    |
|    |  S+U Bundesplatz (Berlin)               | an: 09:40<br>ab: 09:40 |     |
|    |  U Berliner Str. (Berlin)               | an: 09:41<br>ab: 09:41 |          |
|    |  U Güntzelstr. (Berlin)                 | an: 09:43<br>ab: 09:43 |    |
|    |  U Spichernstr. (Berlin)                | an: 09:44<br>ab: 09:44 |   |
|   |  U Kurfürstendamm (Berlin)             | an: 09:45<br>ab: 09:45 |   |
|  |  S+U Zoologischer Garten Bhf (Berlin) | an: 09:46<br>ab: 09:46 |     |
|  |  U Hansaplatz (Berlin)                | an: 09:48<br>ab: 09:48 |    |
|  |  U Turmstr. (Berlin)                  | an: 09:50<br>ab: 09:50 |    |
|  |  U Birkenstr. (Berlin)                | an: 09:51<br>ab: 09:51 |   |
|  |  S+U Westhafen (Berlin)               | an: 09:52<br>ab: 09:52 |     |
|  |  U Amrumer Str. (Berlin)              | an: 09:54<br>ab: 09:54 |    |
|  |  U Leopoldplatz (Berlin)              | an: 09:55<br>ab: 09:55 |    |
|  |  U Nauener Platz (Berlin)             | an: 09:57<br>ab: 09:57 |    |
|  |  U Osloer Str. (Berlin)               | an: 09:58              |  <br>Blindenleitstreifen am Bahnsteig U8  |



# Public Transport Graph in JSON

```
[{"9196508": {
  "stationId": "9196508"
  , "reachableStations": [
    {"stationId": "9196003", "minutes": 1, "line": "263", "stay": 0}
    , {"stationId": "9196507", "minutes": 1, "line": "263", "stay": 0}
  ]
  , "lines": ["263"]
  , "name": "Joachimstr. (Berlin)"
  , "pos": {
    "lat": 52.403066
    , "lng": 13.576713}
  }
  , ...
]
```

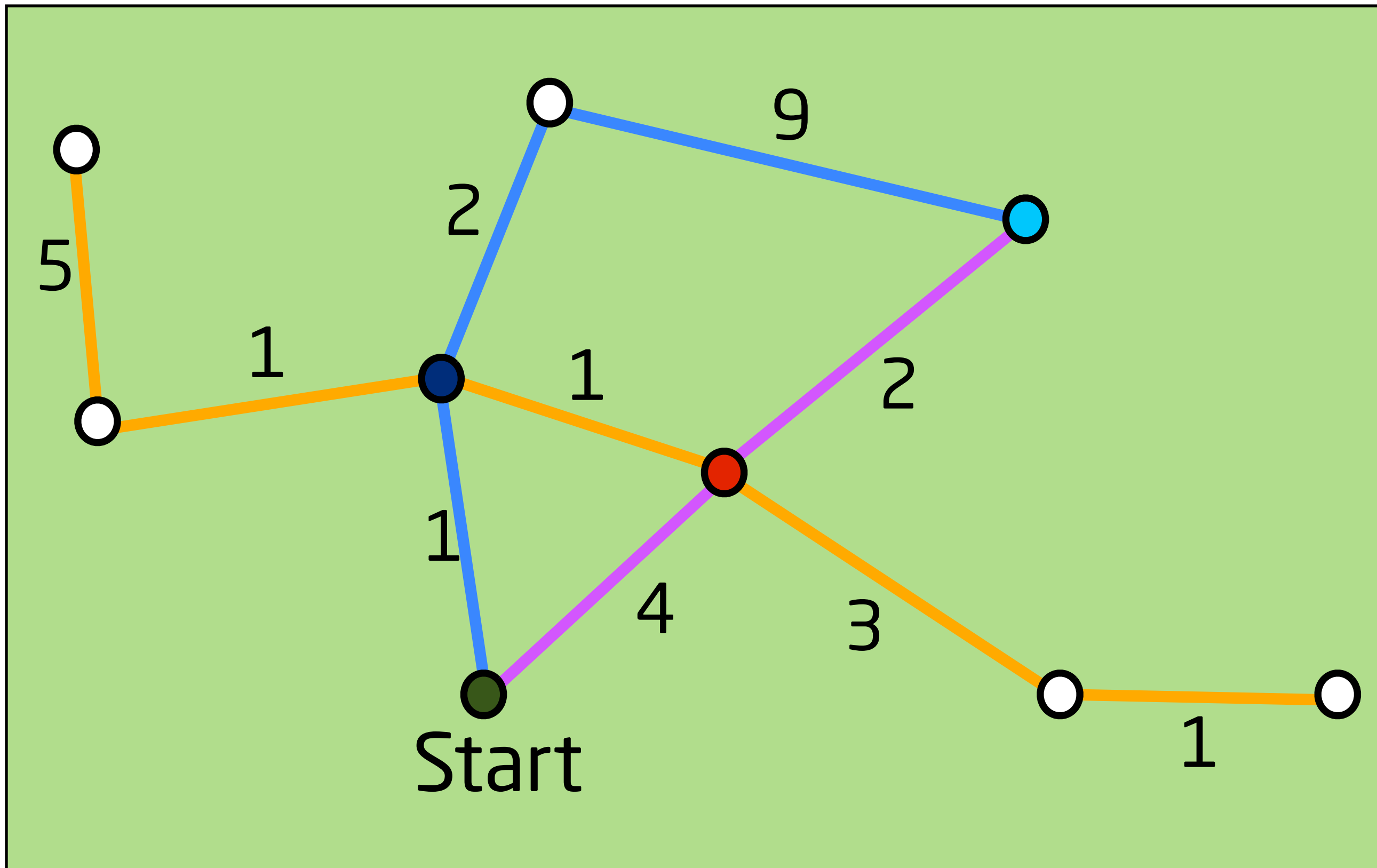
```
{
  "218": {"interval": 25}
  , "133": {"interval": 20}
  , ...
}
```

size: ~800 kb



# Mapnificent Routing

It's Dijkstra with some subtleties.



- best *time* to *every* station (instead of best route to one)
- line switch time
- no-switch-stay time
- same line look-ahead



I felt like I had to show some code, so here we go:

# The routing algorithm

```
var calculateTimes = function(index, stationId, minutes, line, stay){-
  var station = stations[stationId];-
  if (line != null && typeof(stationMap[index][stationId]) !== "undefined" && -
    stationMap[index][stationId].minutes <= minutes){-
    /* Same line look-ahead:-
      I got here faster before, but maybe switching lines caused a delay for-
      the next station on this line, so I'll be faster at the next station even-
      though it took me longer to get to the current one. Let's check it out!-
    */-
    for(var i=0;i<station.reachableStations.length;i++){-
      if(station.reachableStations[i].line == line){-
        // a station on the same line-
        var nextMinutes = minutes + station.reachableStations[i].minutes + stay;-
        if (typeof(stationMap[index][station.reachableStations[i].stationId]) === "undefined" ||-
          stationMap[index][station.reachableStations[i].stationId].minutes > nextMinutes){-
          // Yeah, I can get to the next station on this line faster than before, let's go there!-
          calculateTimes(index, station.reachableStations[i].stationId, nextMinutes, -
            station.reachableStations[i].line, station.reachableStations[i]["stay"]);-
        }-
      }-
    }-
    return;-
  }-
  stationMap[index][stationId] = {"minutes": minutes};-
  for(var i=0;i<station.reachableStations.length;i++){-
    if (line == null){-
      // My first station! I don't have to wait!-
      var nextMinutes = minutes + station.reachableStations[i].minutes;-
    } else if(station.reachableStations[i].line == line){-
      // Same line! The current transport may pause here for some time-
      var nextMinutes = minutes + station.reachableStations[i].minutes + stay;-
    } else {-
      // Switch line! Guess the wait time for the next line-
      var nextMinutes = minutes + getWaitTime(stationId, line, station.reachableStations[i].stationId, -
        station.reachableStations[i].line) + station.reachableStations[i].minutes;-
    }-
    calculateTimes(index, station.reachableStations[i].stationId, nextMinutes, -
      station.reachableStations[i].line, station.reachableStations[i]["stay"]);-
  }-
  return true;-
};-
```

 This is slow, needs improvement.



Result: map of minute values



Hard work done: now just draw circles



# How fast is it?

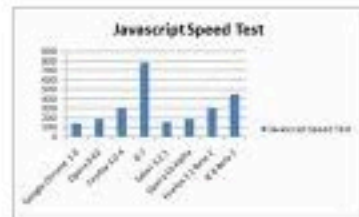
javascript benchmark

Suche

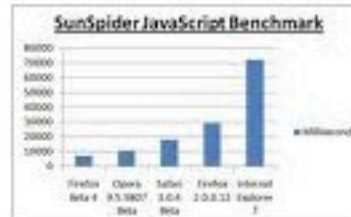
SafeSearch - Mittel ▼

Ungefähr 393.000 Ergebnisse (0,19 Sekunden)

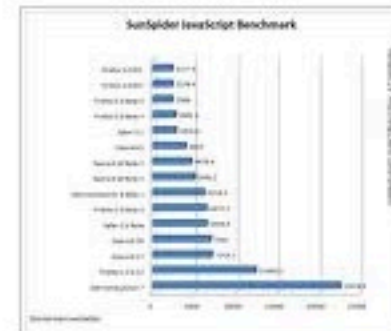
Erweiterte Suche



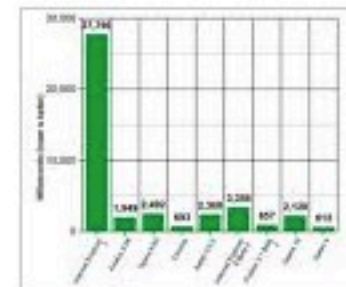
**JavaScript Benchmark**  
479 × 287 - 26 KB - jpg  
yofenet.wordpress.com  
[Ähnliche Bilder suchen](#)



**SunSpider JavaScript Benchmark**  
455 × 282 - 30 KB - jpg  
tech-buzz.net  
[Ähnliche Bilder suchen](#)



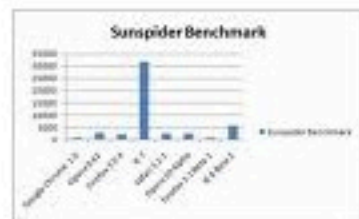
**JavaScript**  
711 × 607 - 25 KB - png  
zdnet.com  
[Ähnliche Bilder suchen](#)



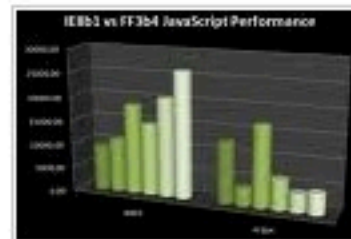
**JavaScript**  
415 × 342 - 13 KB - png  
maximumpc.com  
[Ähnliche Bilder suchen](#)



**JavaScript**  
1280 × 800 - 441 KB - png  
favbrowser.com  
[Ähnliche Bilder suchen](#)



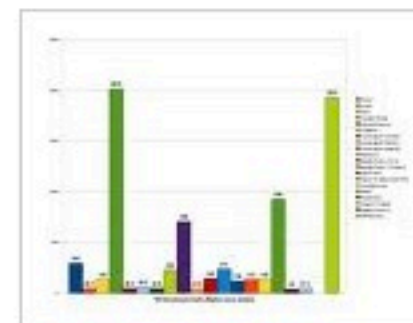
**Kita lihat hasil**  
479 × 286 - 25 KB - jpg  
yofenet.wordpress.com  
[Ähnliche Bilder suchen](#)



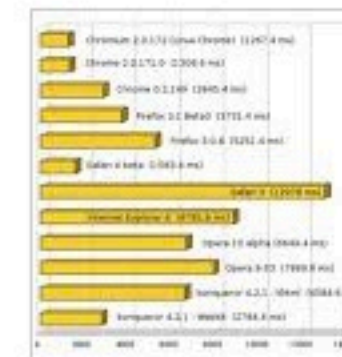
**JavaScript**  
541 × 352 - 57 KB - png  
wictorwilen.se  
[Ähnliche Bilder suchen](#)



**JavaScript**  
1024 × 768 - 686 KB - png  
ejohn.org



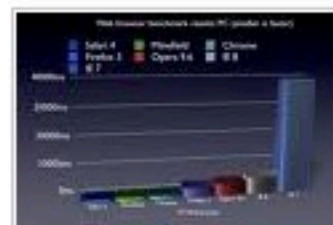
**SunSpider JavaScript**  
1056 × 816 - 13 KB - png  
tmsnetwork.org  
[Ähnliche Bilder suchen](#)



**SunSpider JavaScript**  
460 × 485 - 47 KB - jpg  
hyperray.net  
[Ähnliche Bilder suchen](#)



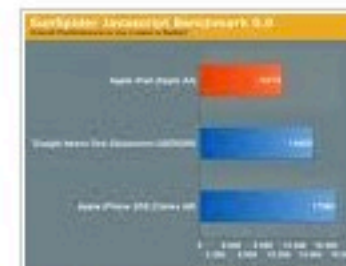
**as it can run**  
550 × 526 - 72 KB - jpg  
legitreviews.com  
[Ähnliche Bilder suchen](#)



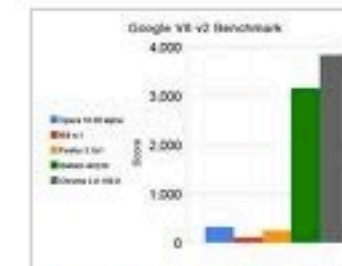
**Browser-Benchmark**  
336 × 225 - 11 KB - jpg  
zdn.net



**37 JavaScript**  
500 × 500 - 117 KB - jpg  
pathf.com  
[Ähnliche Bilder suchen](#)



**Auch im**  
400 × 305 - 62 KB - png  
benm.at



**JavaScript**  
440 × 330 - 21 KB - jpg  
zdnet.com.au  
[Ähnliche Bilder suchen](#)

Current online version:  
works well enough

Current dev version:  
Chromium: 2.247s  
Opera: 5.852s  
Firefox: 20.938s



Warning: Unresponsive script

A script on this page may be busy, or it may have stopped responding. You can stop the script now, open the script in the debugger, or let the script continue.

Script: <http://localhost/mapnificent/mapnificent/media/layers/urbandistance.js:329>

☐ Don't ask me again

Debug script

Continue

Stop script

# How accurate is it?

Feeling: good enough (you need a travel planner anyways).

Greater distance leads to greater error.

Missing transport options: regional trains.

Going to make comparison with VBB command line tool.



# Mapumental

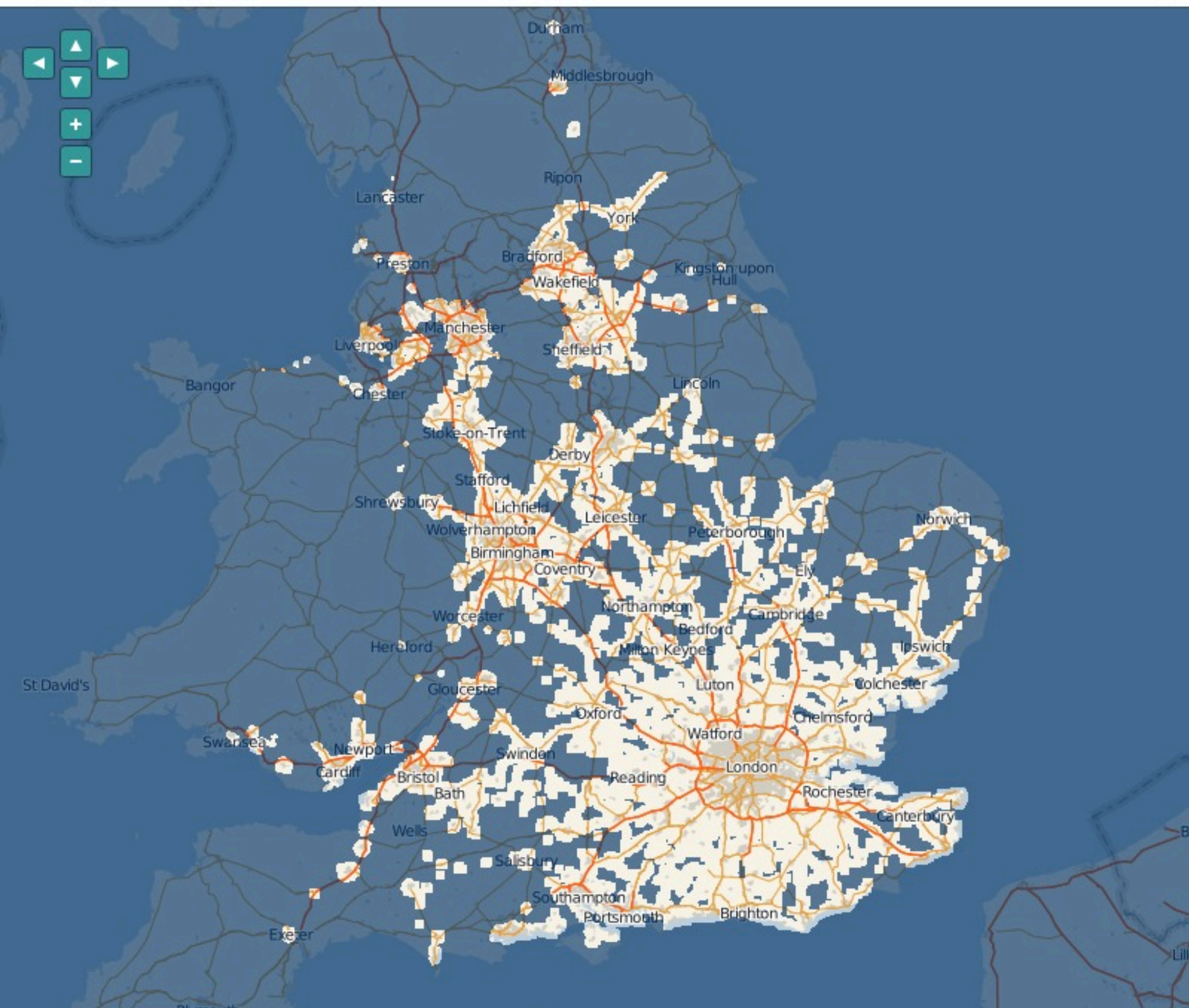
- Works for whole UK
- C++ Route Finder
- Post code level only
- Aggressive Caching
- Tile Server
- Flash Interface

Public transport arriving at 9:00

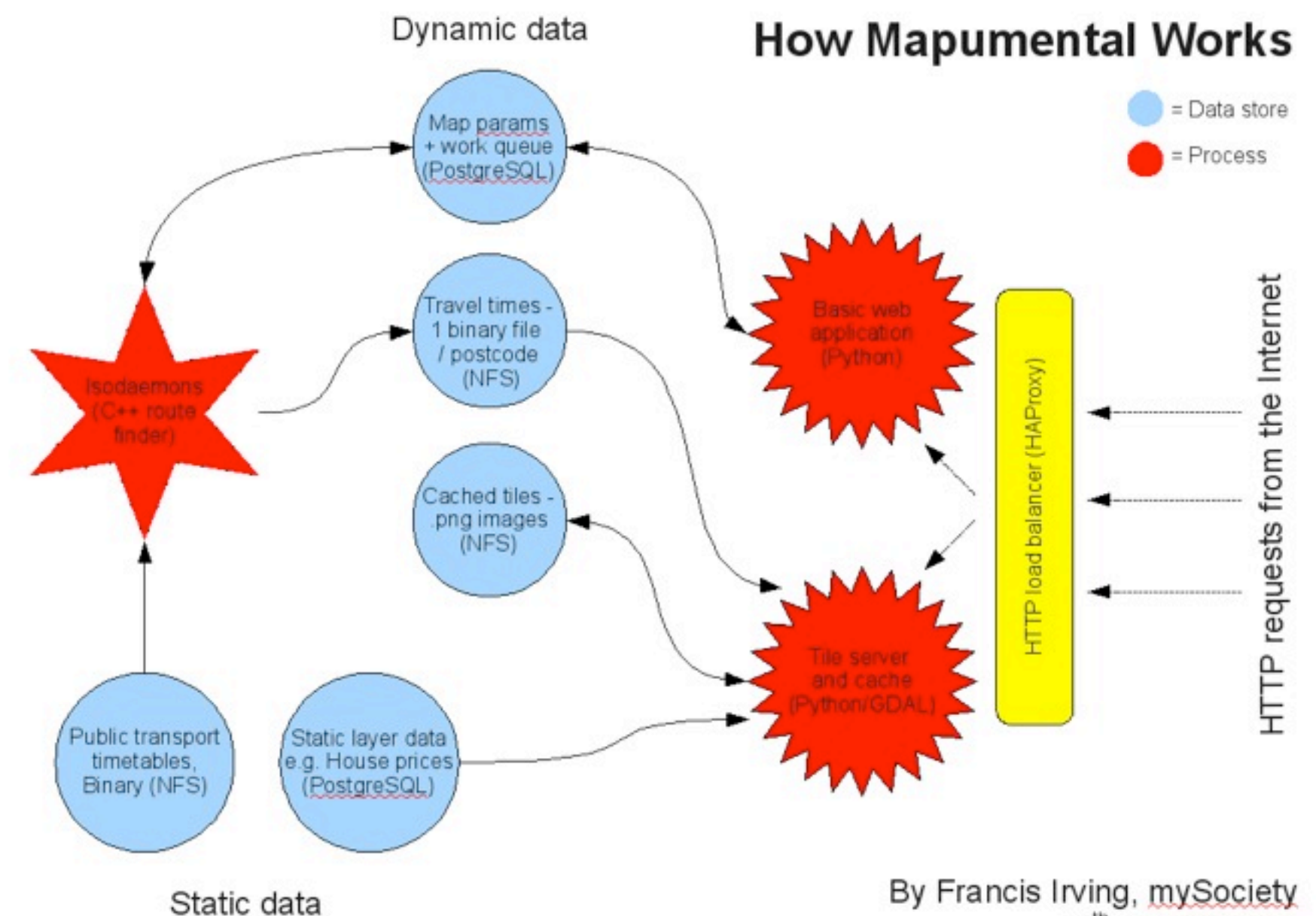
House price (average)

Departing at 4:57

Any price



New map? GB postcode:  (  (   )



By Francis Irving, mySociety  
17<sup>th</sup> August 2009

<http://www.mysociety.org/2009/08/18/how-mapumental-works/>



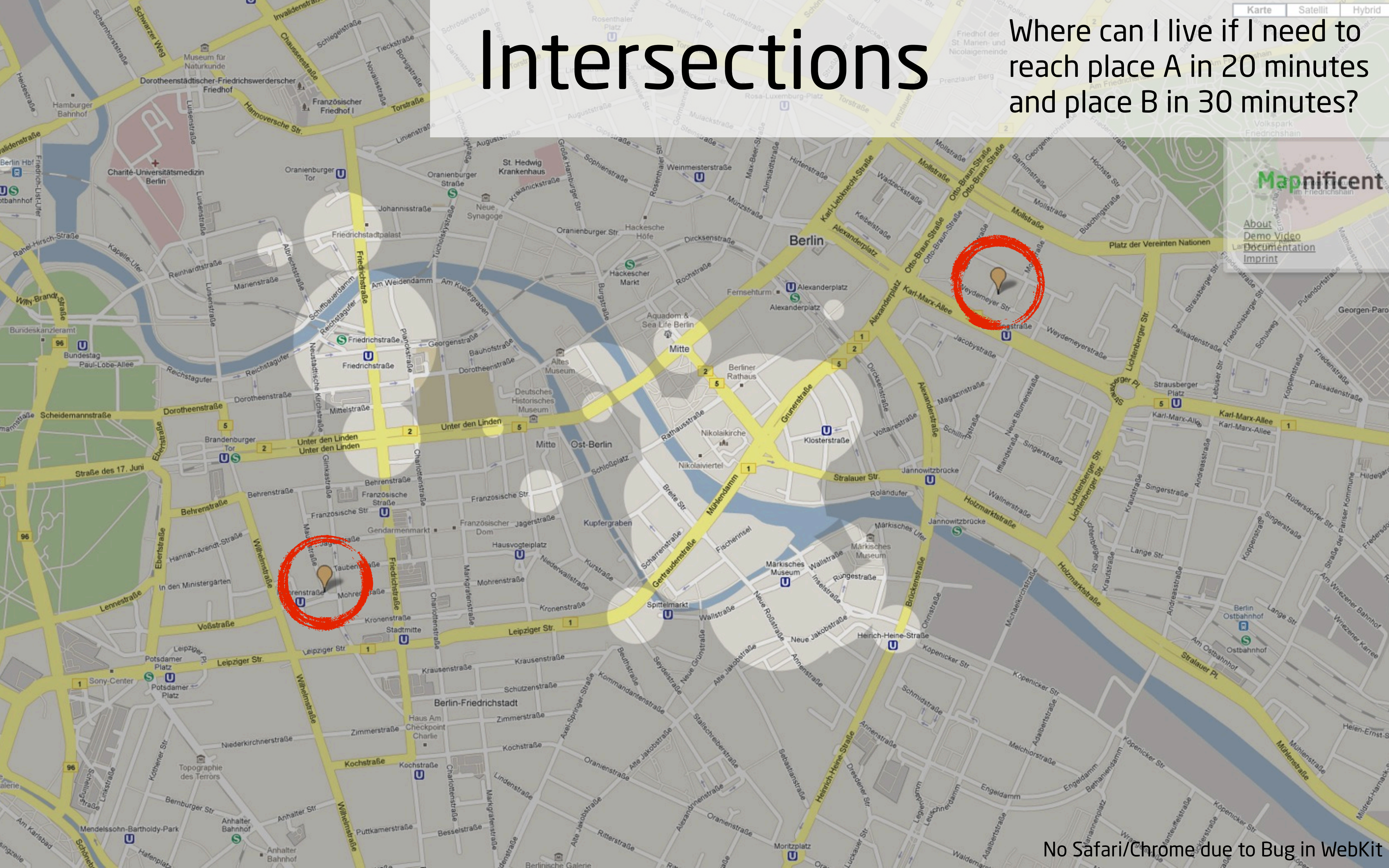
The screenshot displays the Mapnificent web application interface. At the top right, there are tabs for 'Map', 'Satellite', and 'Hybrid'. A 'Mapnificent' logo is visible in the top right corner, with links for 'About', 'Demo Video', 'Documentation', and 'Imprint'. The main map area shows a detailed view of Berlin, with a green location pin placed in the city center. Below the map, there is a Google search bar and a 'Search' button. A row of checkboxes allows users to toggle features: 'Public Transport' (checked), 'Recovery Package' (unchecked), 'Location' (checked), 'Web Map Service' (unchecked), and 'Load your own!' (unchecked). At the bottom, the 'Urban Distance' section is active, showing 'Area reachable in max. 27 minutes (no guarantee)' and a green link 'Click in the grey area to set a new position.' A slider bar is present for adjusting the time. To the right, there are radio buttons for 'with Public Transport' (selected) and 'with Public Transport and bike', and a text input for 'Max. time to walk/ride from/to stations: 10 minutes'.

Works from any point in, but only in Berlin area.  
Calculates approximate time live, no caching, no requests.  
Canvas rendered in browser, no Flash, no tiles

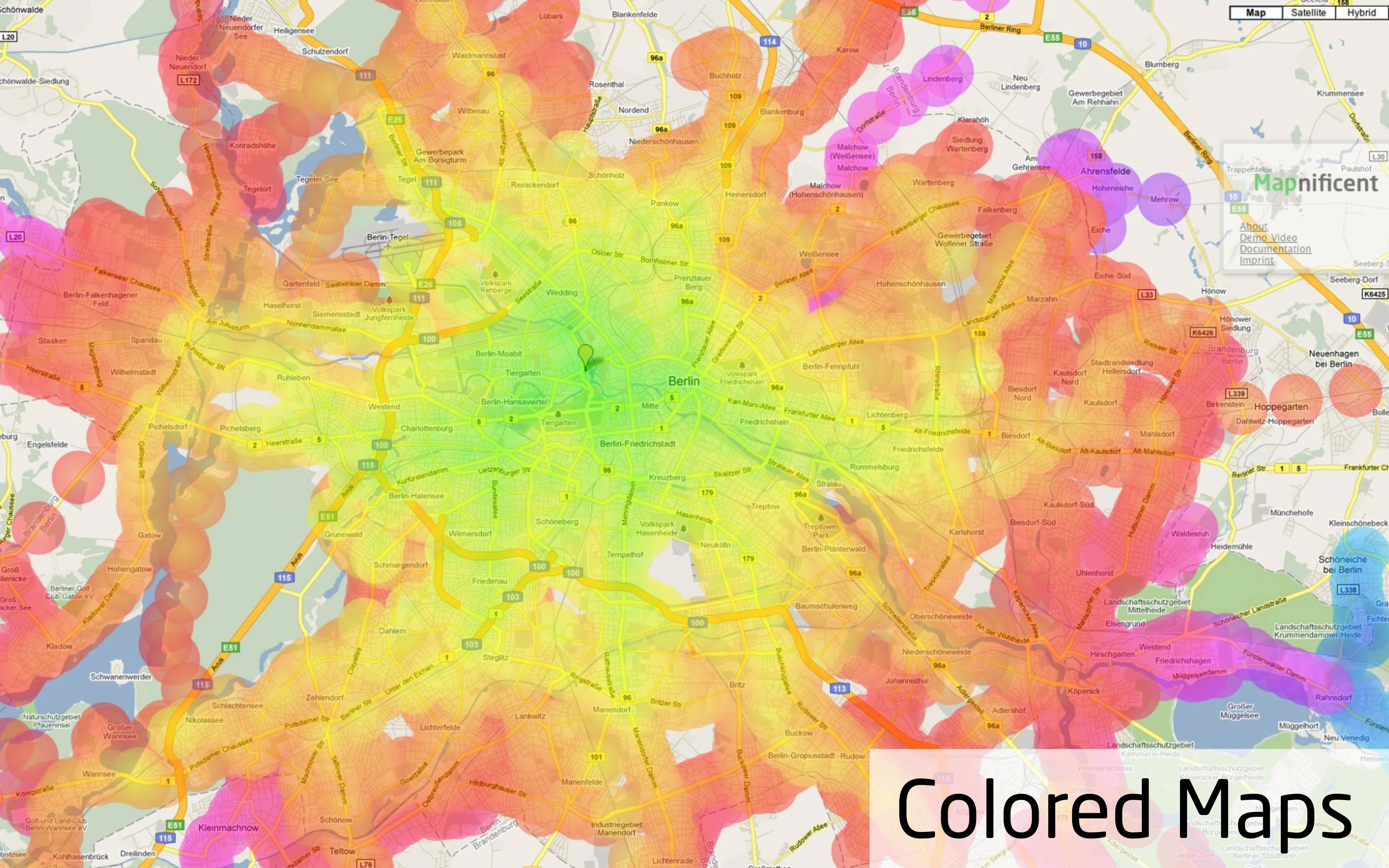


# Intersections

Where can I live if I need to reach place A in 20 minutes and place B in 30 minutes?







Colored Maps





# Thank you!

<http://www.mapnificent.de>

<http://github.com/stefanw/Mapnificent>

<http://stefanwehrmeyer.com>

<http://twitter.com/stefanwehrmeyer>

Map Satellite Hybrid

Mapnificent

[About](#)  
[Demo Video](#)  
[Documentation](#)  
[Imprint](#)