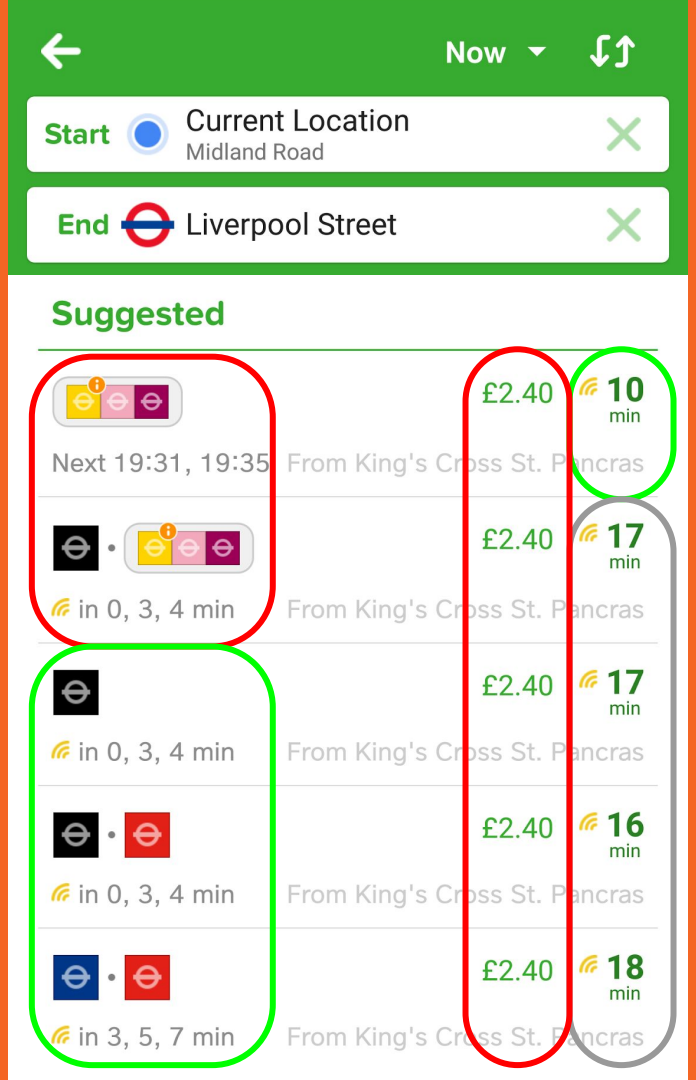
Haggle

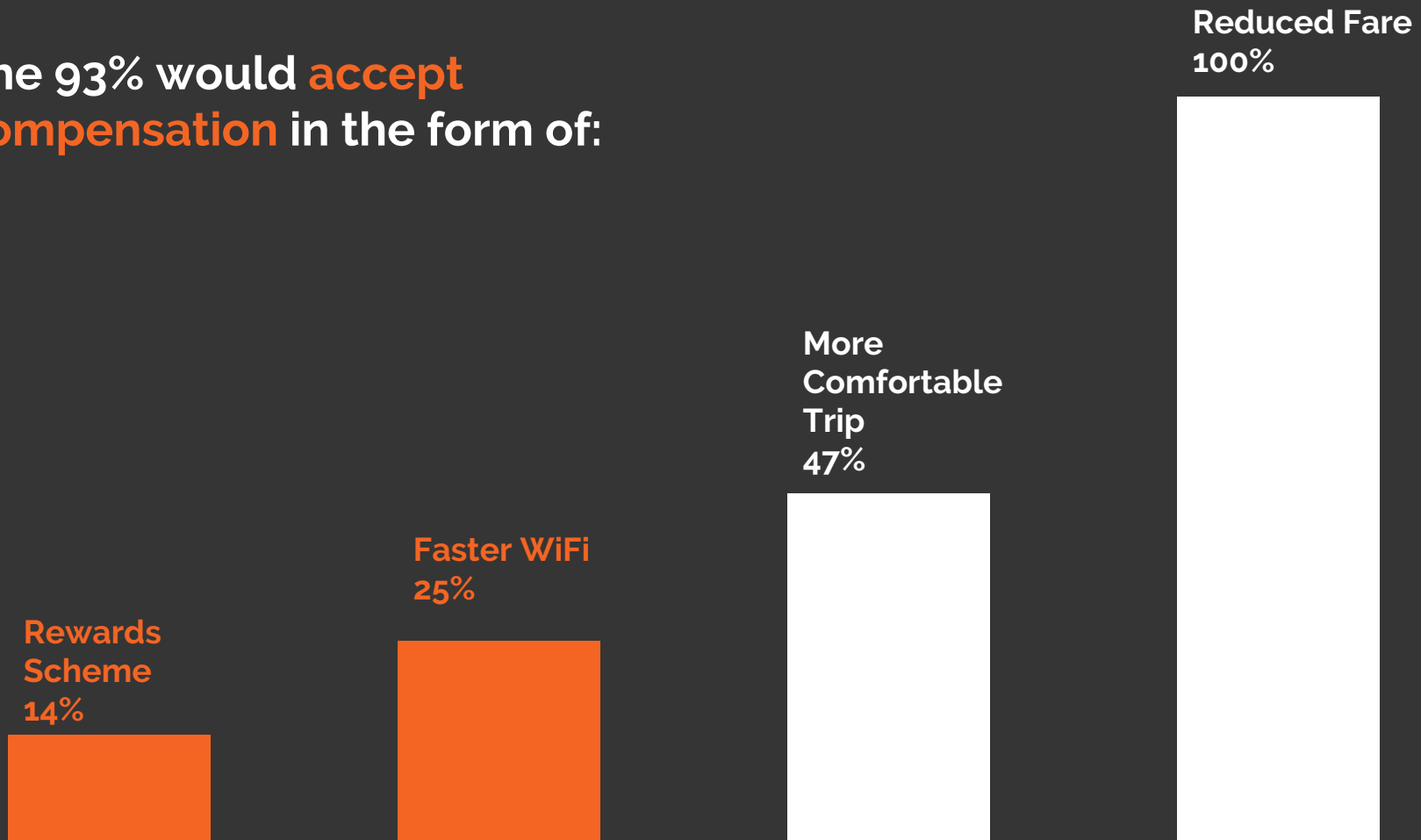

Now
Start
Current Location
Midland Road
End
Liverpool Street
Suggested
£2.40
10 min
Next 19:31, 19:35
From King's Cross St. Pancras
£2.40
17 min
in 0, 3, 4 min
From King's Cross St. Pancras
£2.40
17 min
in 0, 3, 4 min
From King's Cross St. Pancras
£2.40
16 min
in 0, 3, 4 min
From King's Cross St. Pancras
£2.40
18 min
in 3, 5, 7 min
From King's Cross St. Pancras

Solution:

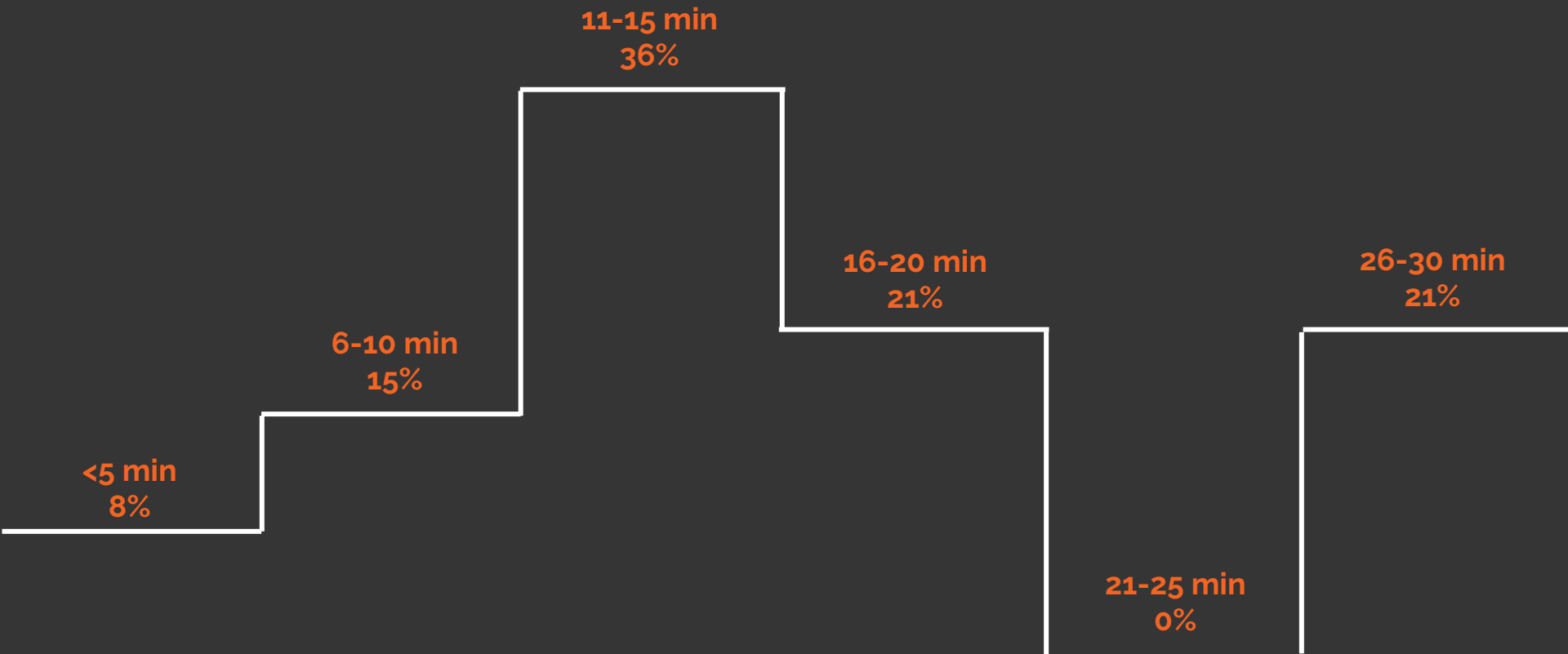
Ease congestion by appealing to passenger behavioural preferences

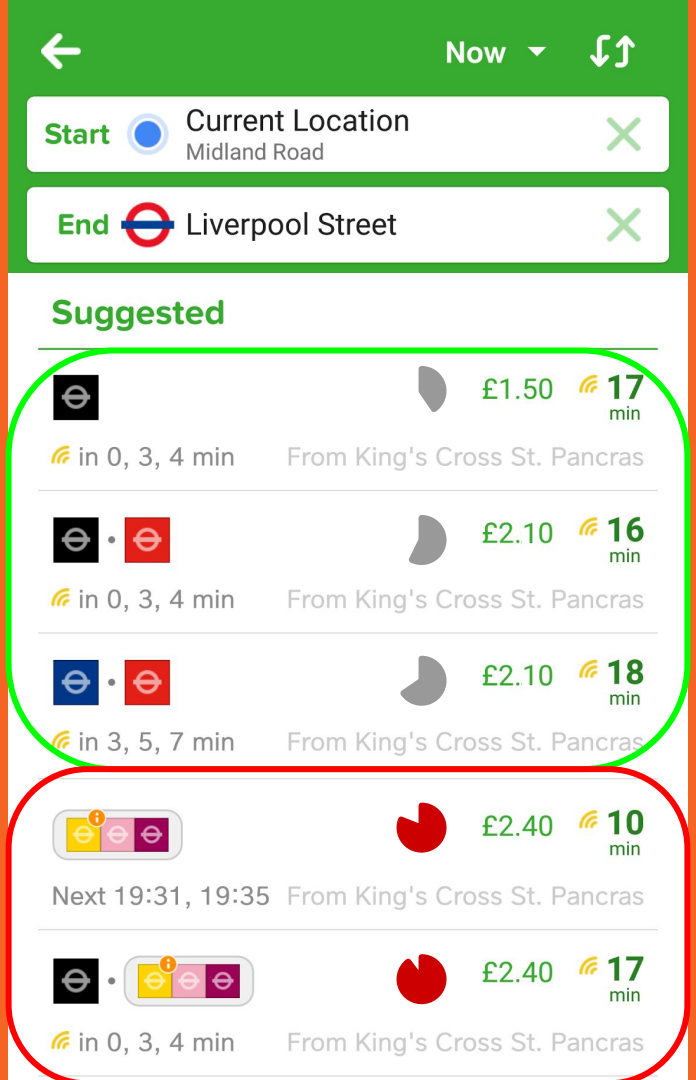
Would you be willing to deviate from the fastest route for a potentially less congested route and some form of compensation?

The 93% would accept compensation in the form of:



The 93% would accept travel duration increases of:



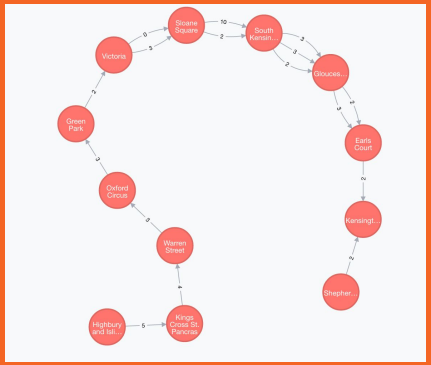

Now
Start
Current Location
Midland Road
End
Liverpool Street
Suggested
£1.50
17 min
in 0, 3, 4 min
From King's Cross St. Pancras
£2.10
16 min
in 0, 3, 4 min
From King's Cross St. Pancras
£2.10
18 min
in 3, 5, 7 min
From King's Cross St. Pancras
£2.40
10 min
Next 19:31, 19:35
From King's Cross St. Pancras
£2.40
17 min
in 0, 3, 4 min
From King's Cross St. Pancras

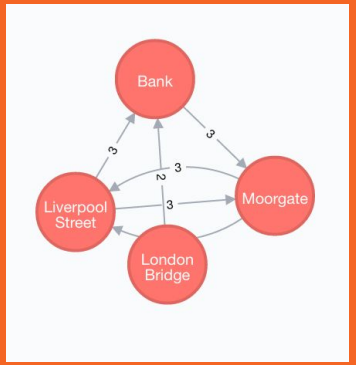
Haggle is an API that will negotiate a compensation scaling factor based on **Train Loading** and **Journey Duration**

$$H = \alpha L + \beta D$$

e.g. Final Price = H × (Base Price)

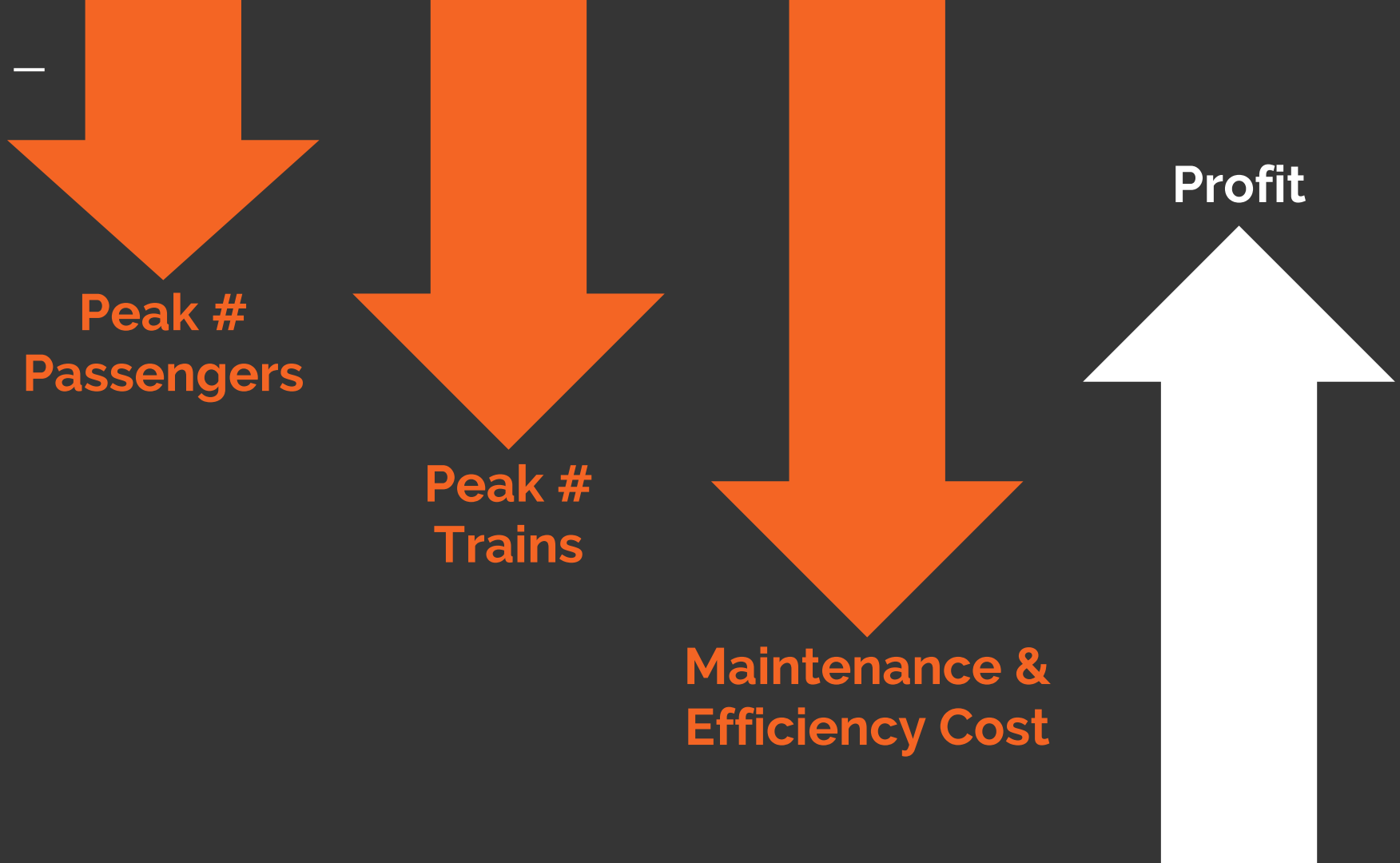
$$y = 0.0002x^6 - 0.0335x^5 + 1.2724x^4 - 11.48x^3 - 172.03x^2 + 2847.2x - 1953.8$$





Demo


```
{
  "from": "Goodge Street",
  "to": "Baker Street",
  "stops": [
    {
      "name": "Warren Street",
      "line": "Northern",
      "loading": 1
    },
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...}
  ],
  "totalLoading": 41
}
```


Peak #
Passengers
Peak #
Trains
Maintenance &
Efficiency Cost
Profit

Haggle with us

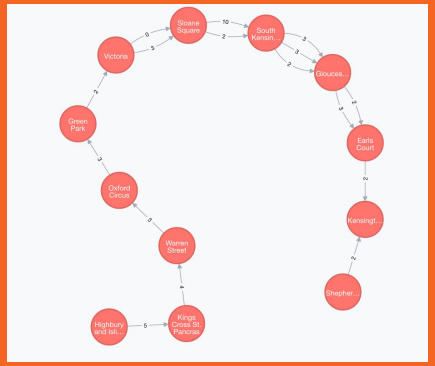
Matthias Günther

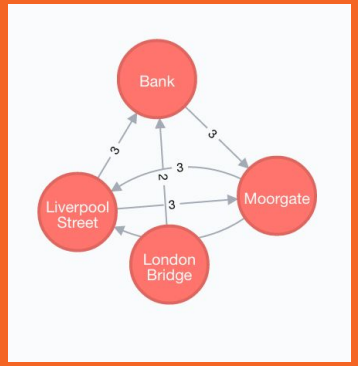
Ben Trew

Kratesh Ramrakhyani

Alan Jamieson

$$y = 0.0002x^6 - 0.0335x^5 + 1.2724x^4 - 11.48x^3 - 172.03x^2 + 2847.2x - 1953.8$$





How?

- Used TFL Tram data to create a loading distribution function - apply to all lines
- Imported into a Neo4J Graph Database to simulate train loadings in real-time
- For a given journey, outputs a JSON including Loading and Duration
- Input into the Haggle function to calculate the haggle factor - multiply by price


```
{
  "from": "Goodge Street",
  "to": "Baker Street",
  "stops": [
    {
      "name": "Warren Street",
      "line": "Northern",
      "loading": 1
    },
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...}
  ],
  "totalLoading": 41
}
```