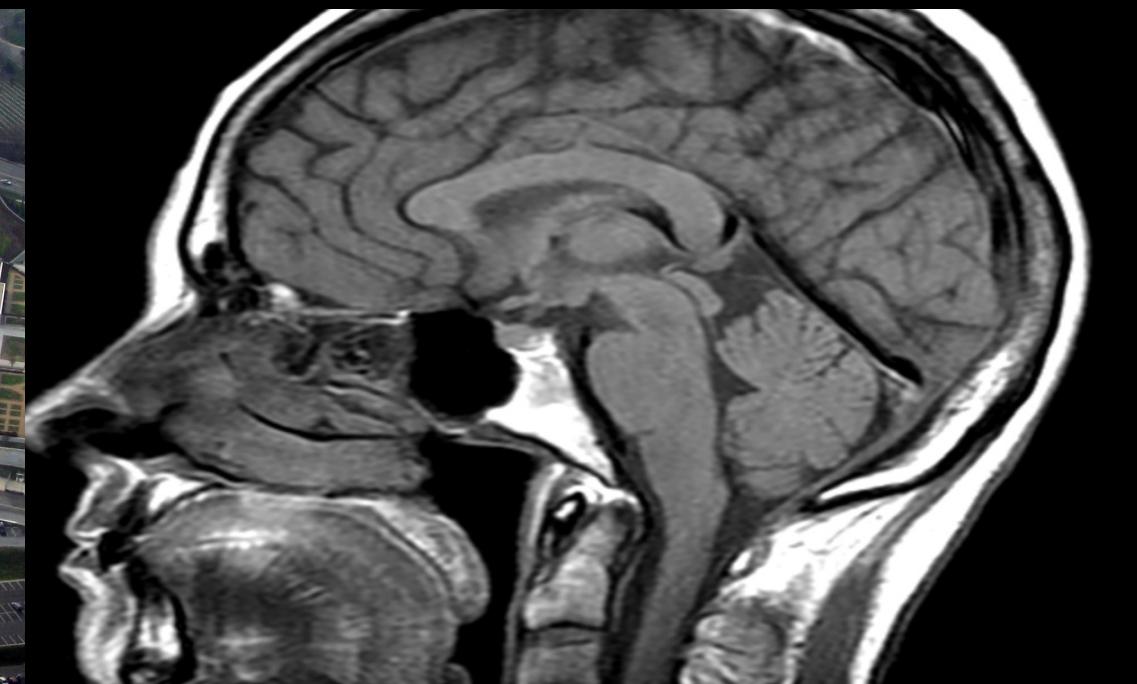
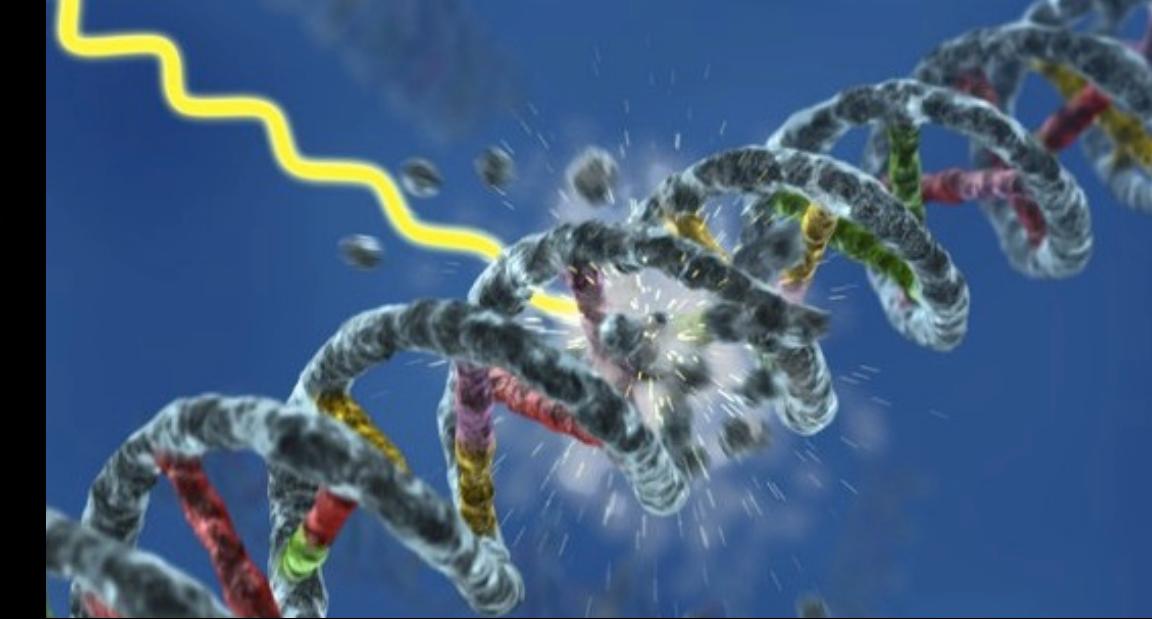
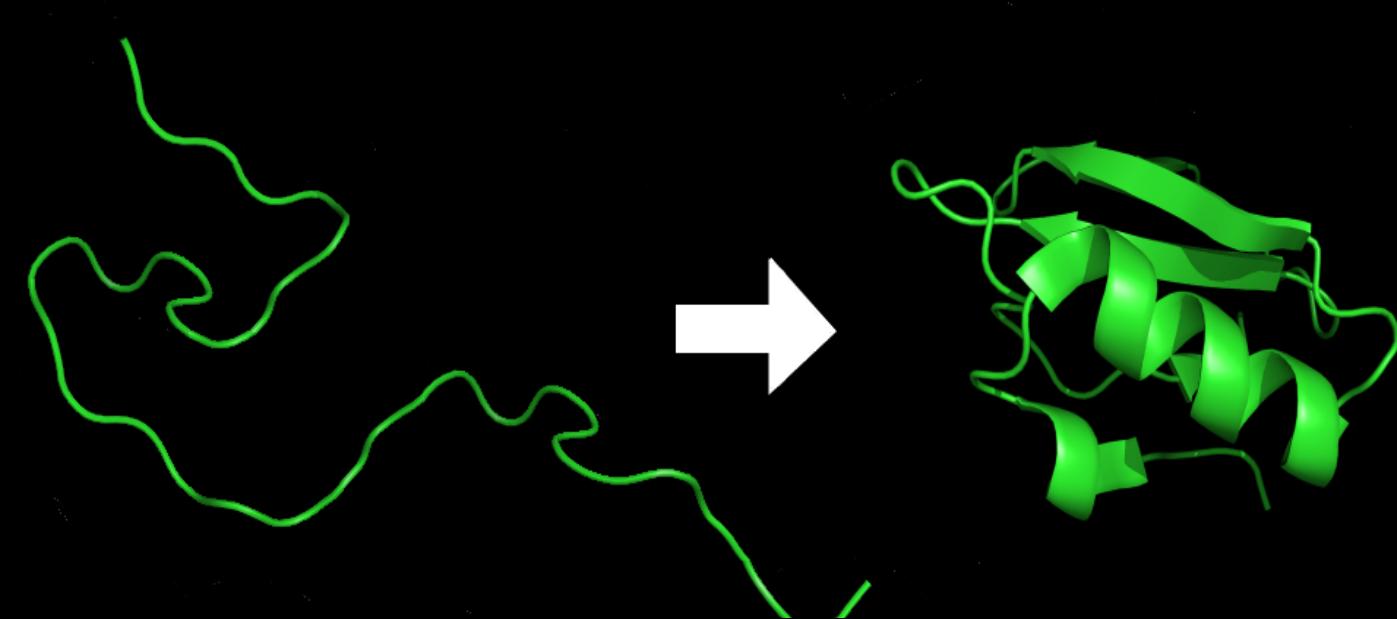
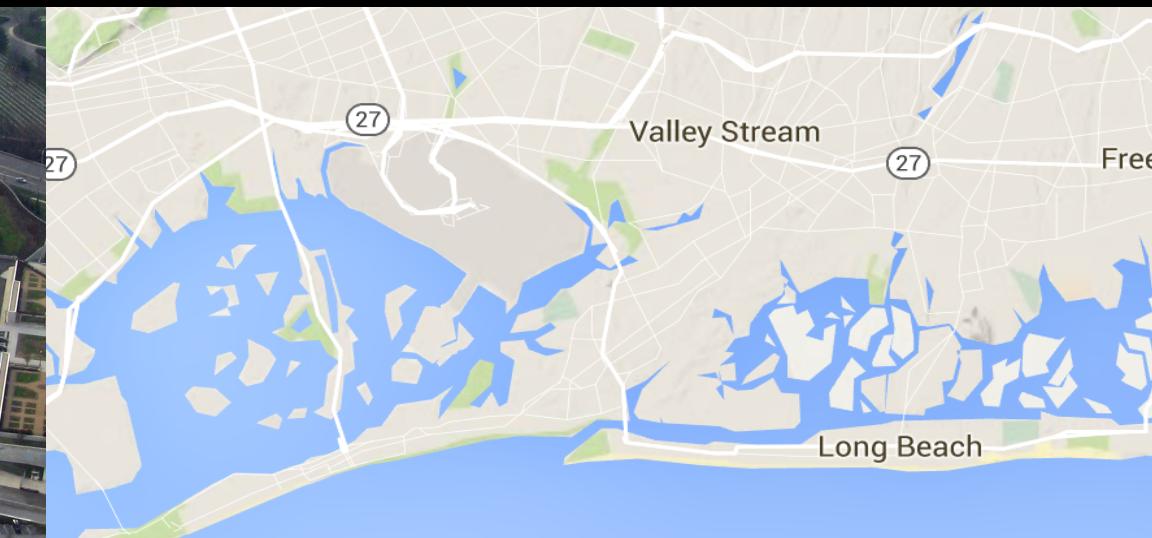


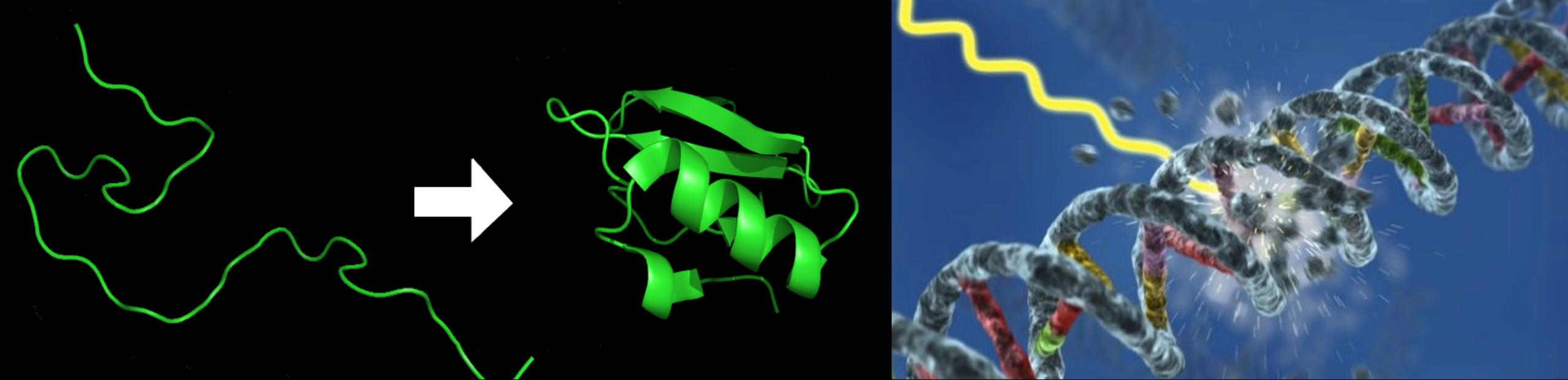
# Josh Friedman



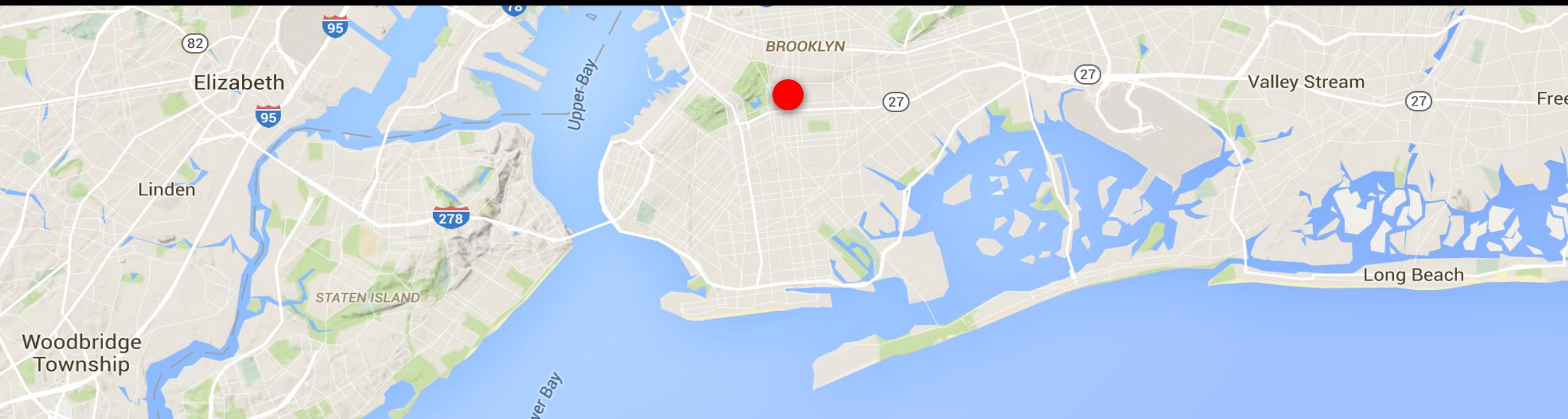


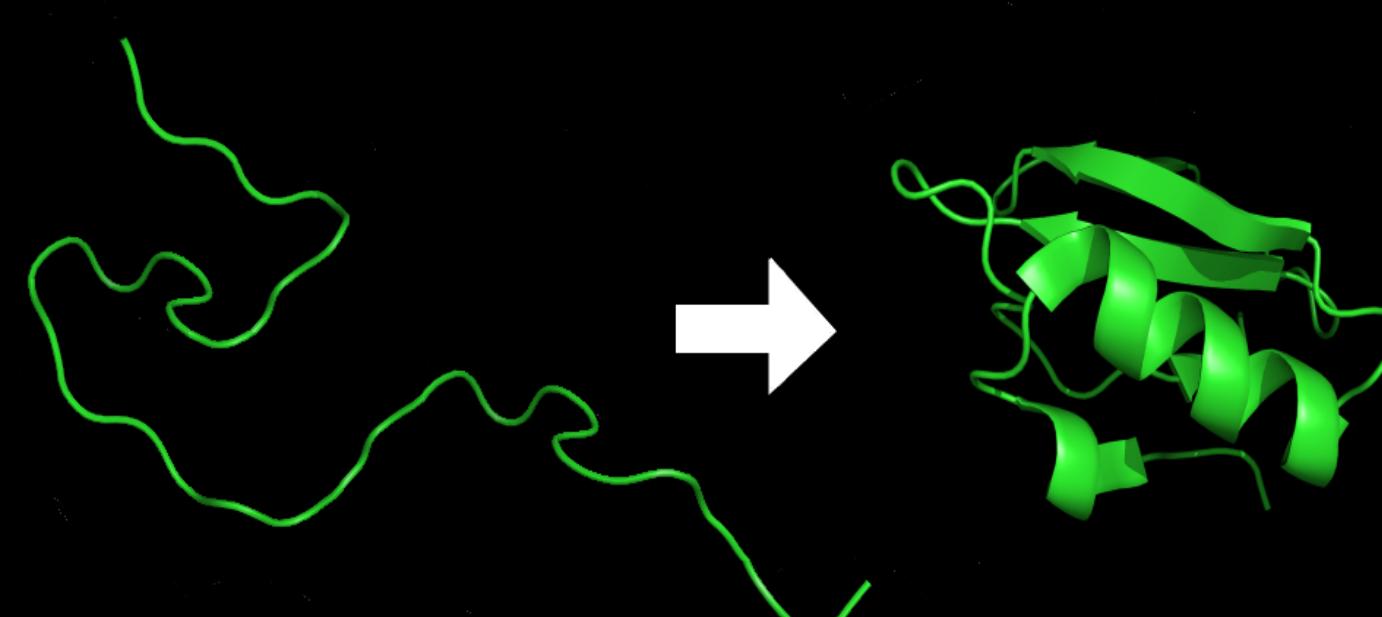
# Josh Friedman



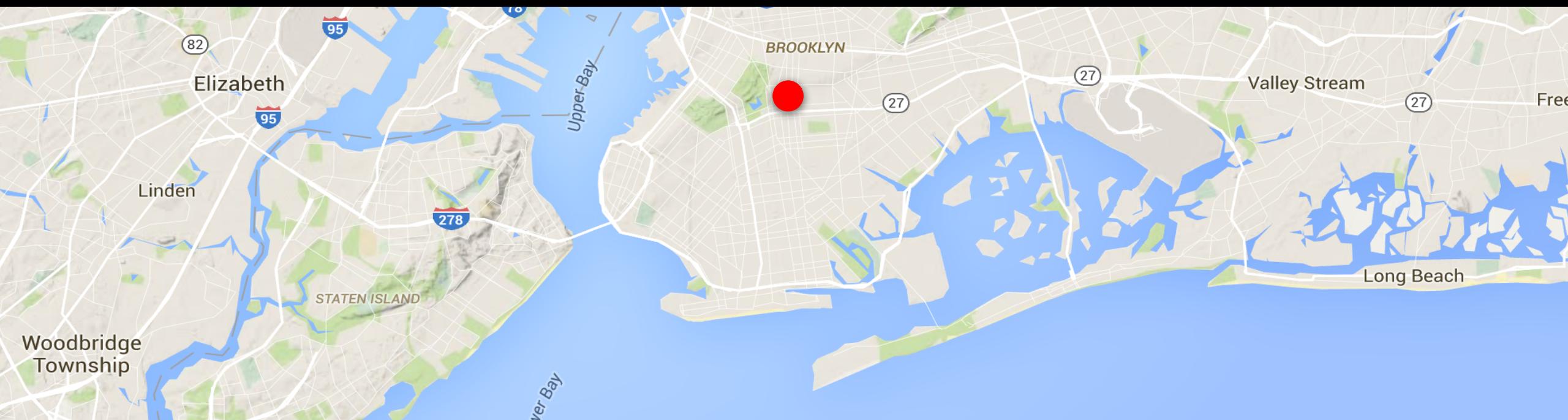


# Josh Friedman

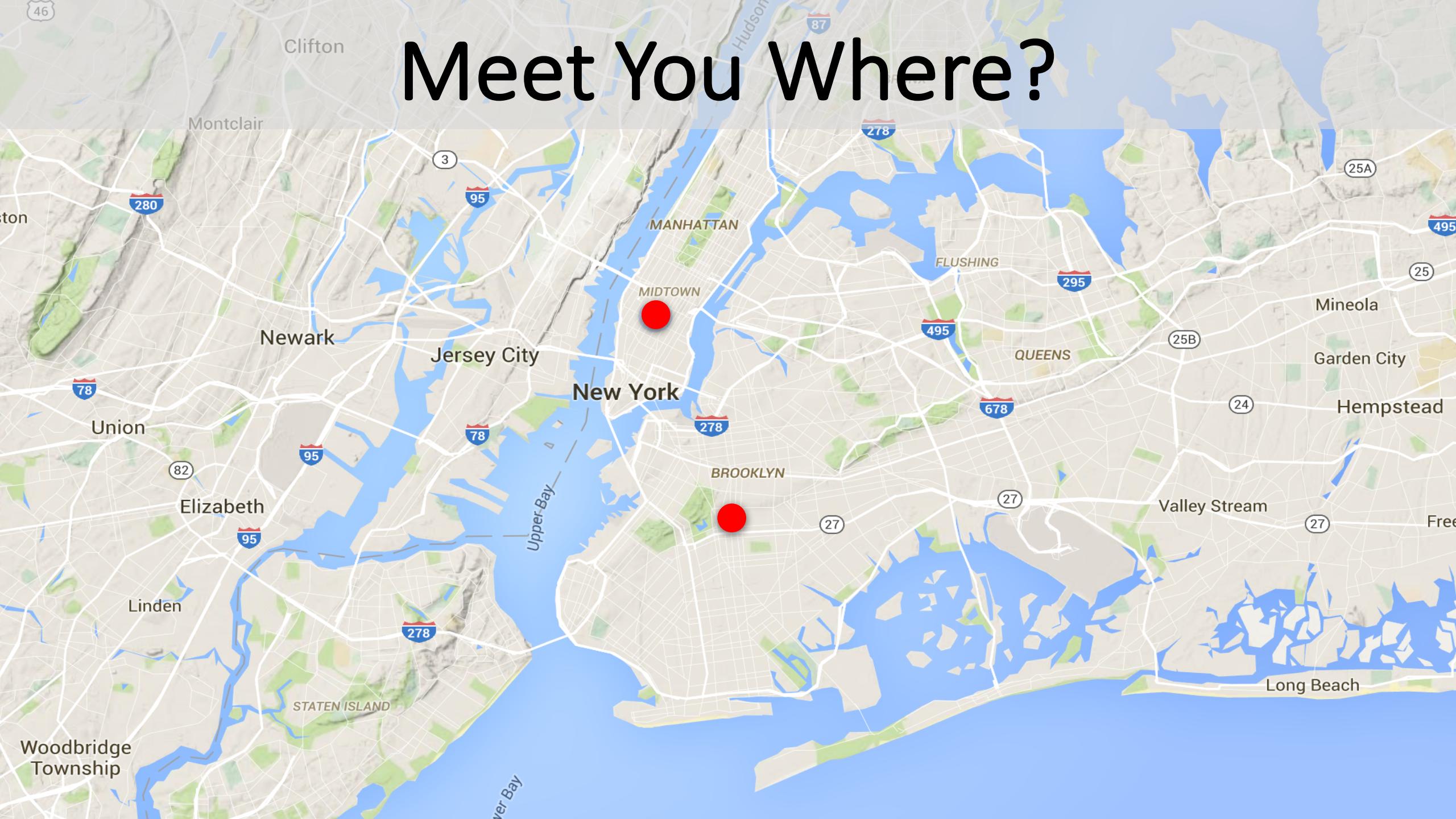




# Josh Friedman



# Meet You Where?

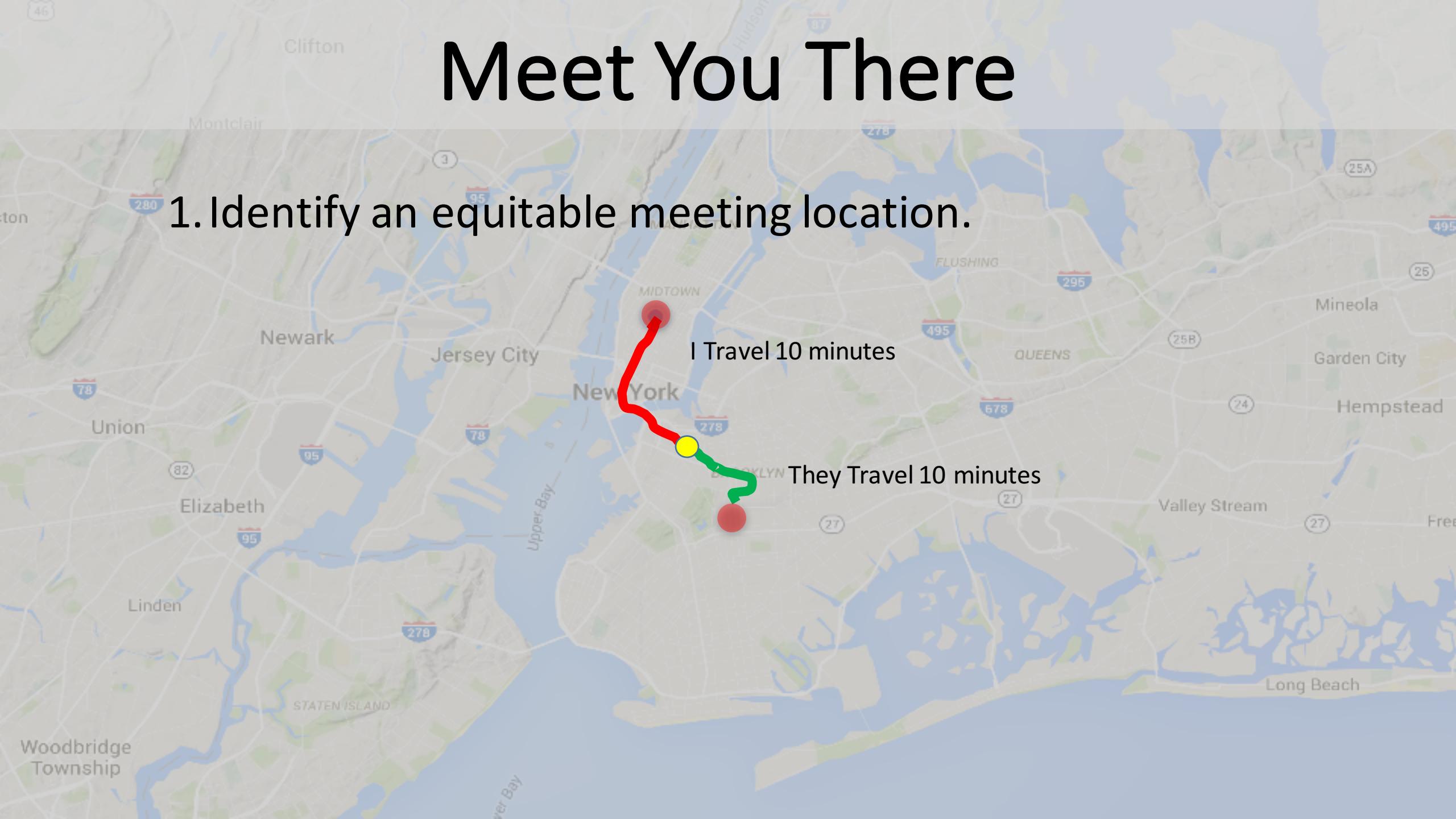


# Meet You Where?



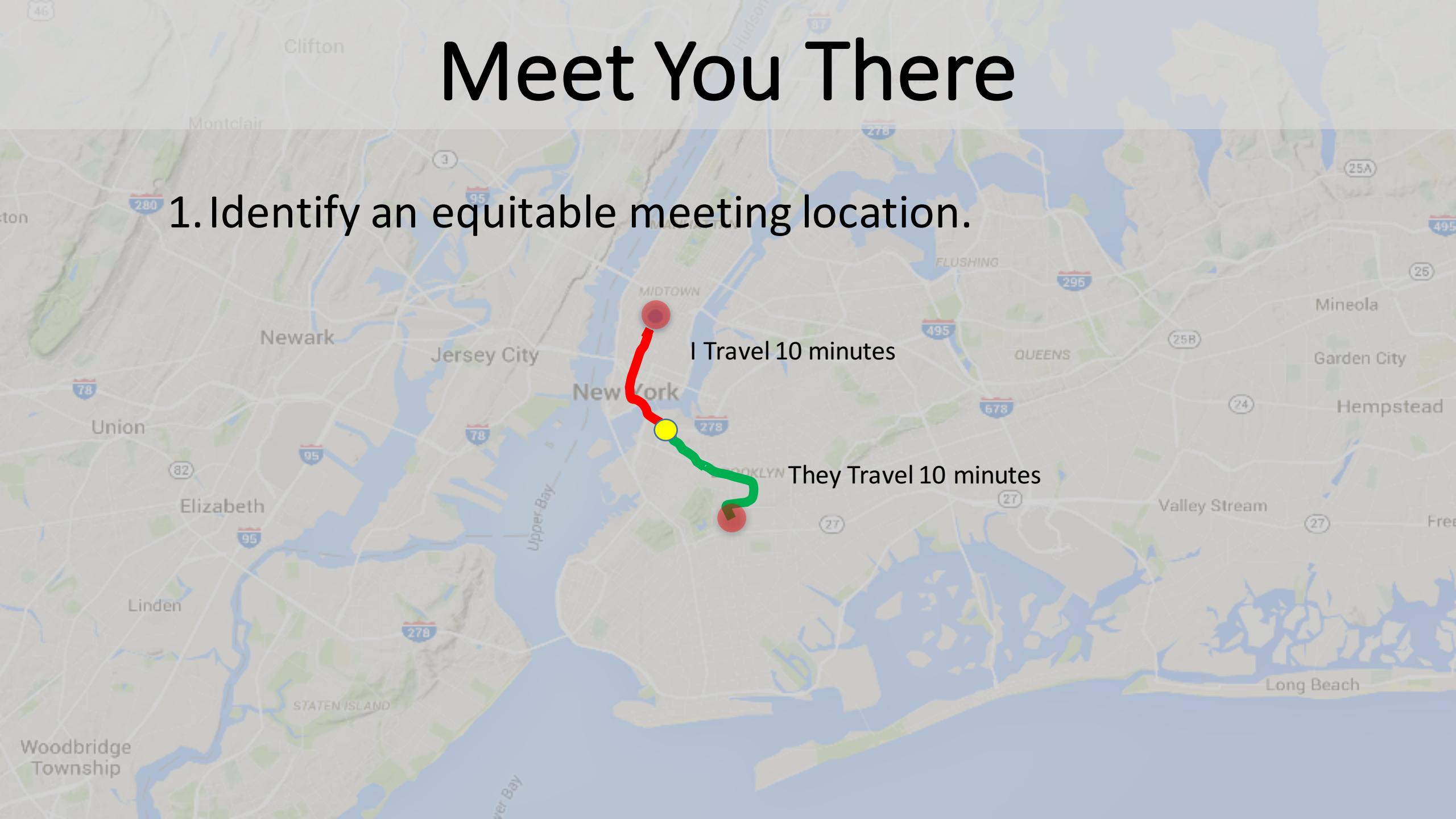
# Meet You There

1. Identify an equitable meeting location.



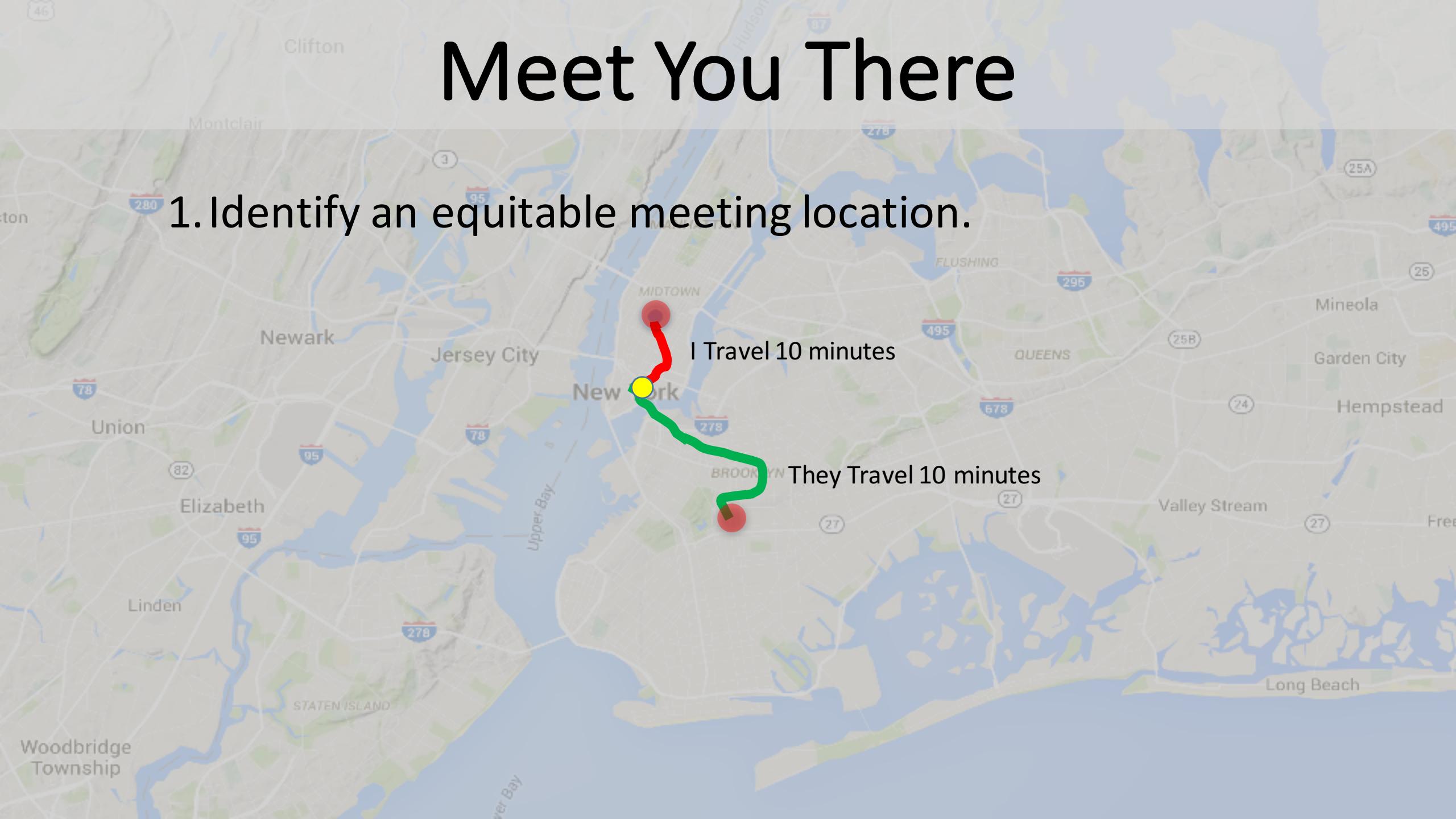
# Meet You There

1. Identify an equitable meeting location.



# Meet You There

1. Identify an equitable meeting location.



# Meet You There

1. Identify an equitable meeting location.

2. Finds restaurants near the ideal meeting place.



# Meet You There

1. Identify an equitable meeting location.

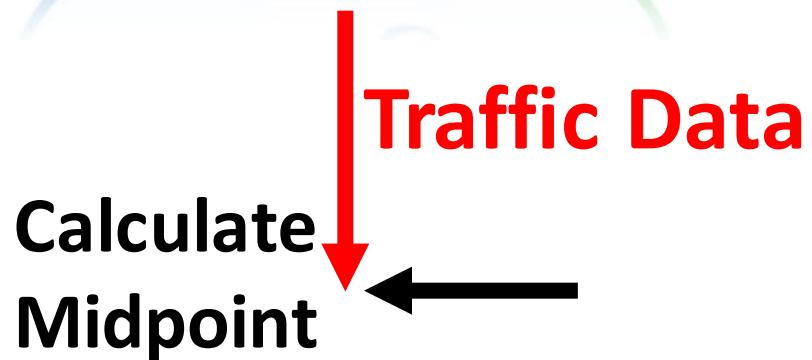
2. Finds restaurants near the ideal meeting place.

3. Identifies the best ones for you.

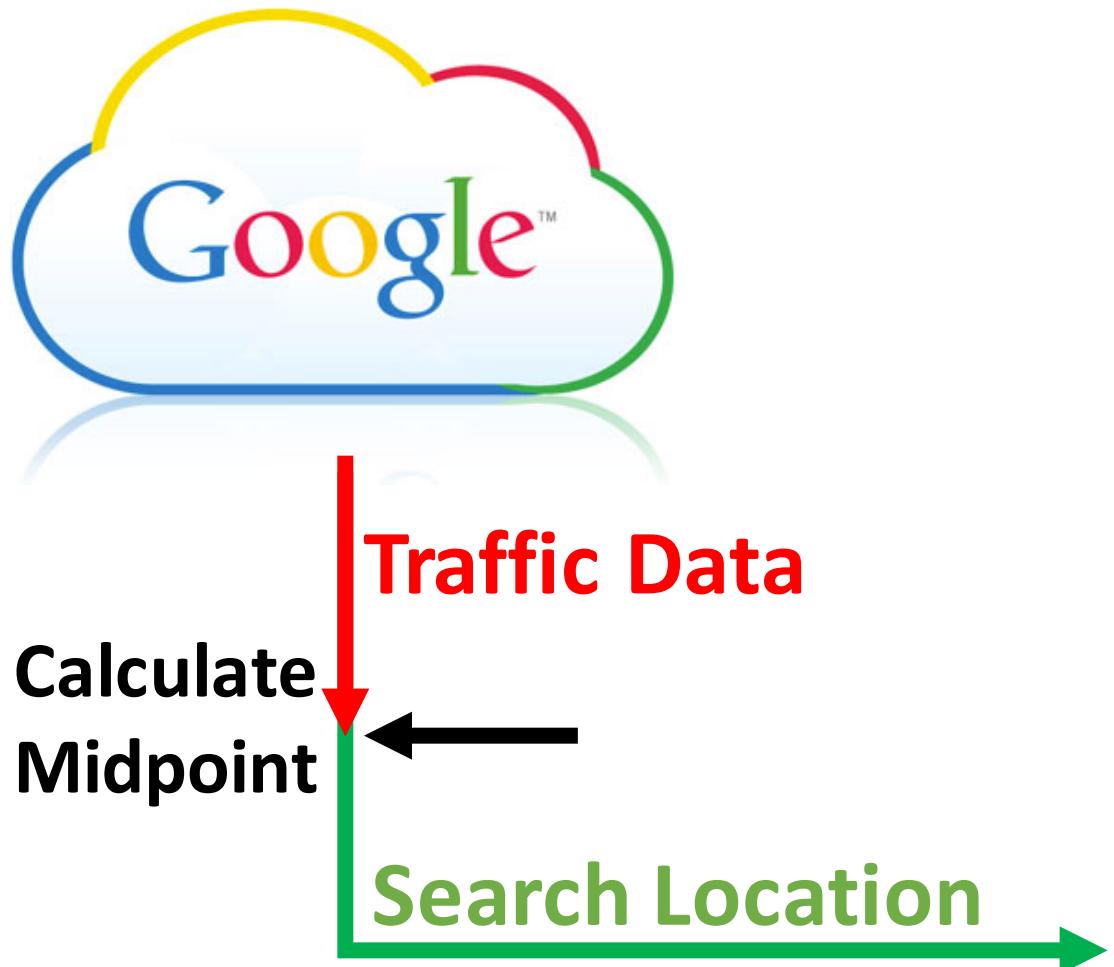
- Travel fairness ratio.
- Best reviewed.
- Lowest Price.



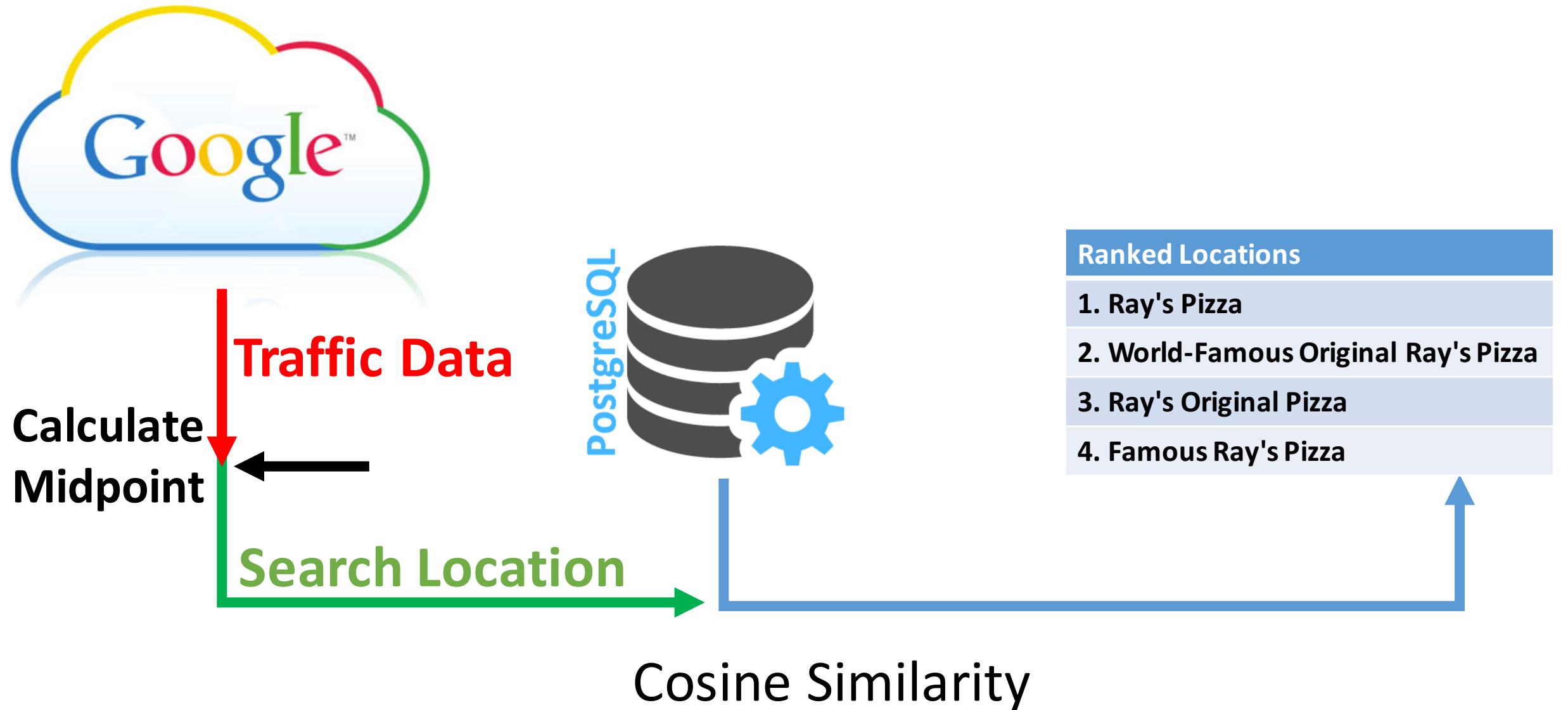
# Behind the scenes



# Behind the scenes



# Behind the scenes



Behind the scenes



Traffic Data  
Calculate Midpoint

Search Location

PostgreSQL



User Selection:

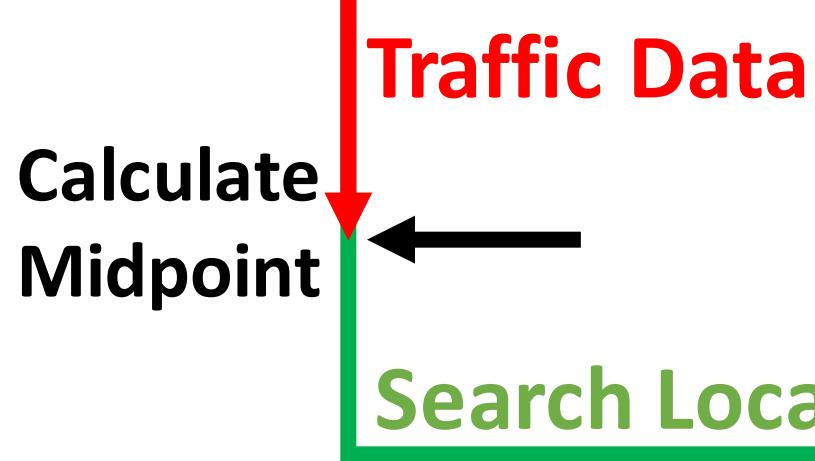
User,  
Review,  
Price,  
Travel Fairness Ratio

Ranked Locations

1. Ray's Pizza
2. World-Famous Original Ray's Pizza
3. Ray's Original Pizza
4. Famous Ray's Pizza

Cosine Similarity

Behind the scenes



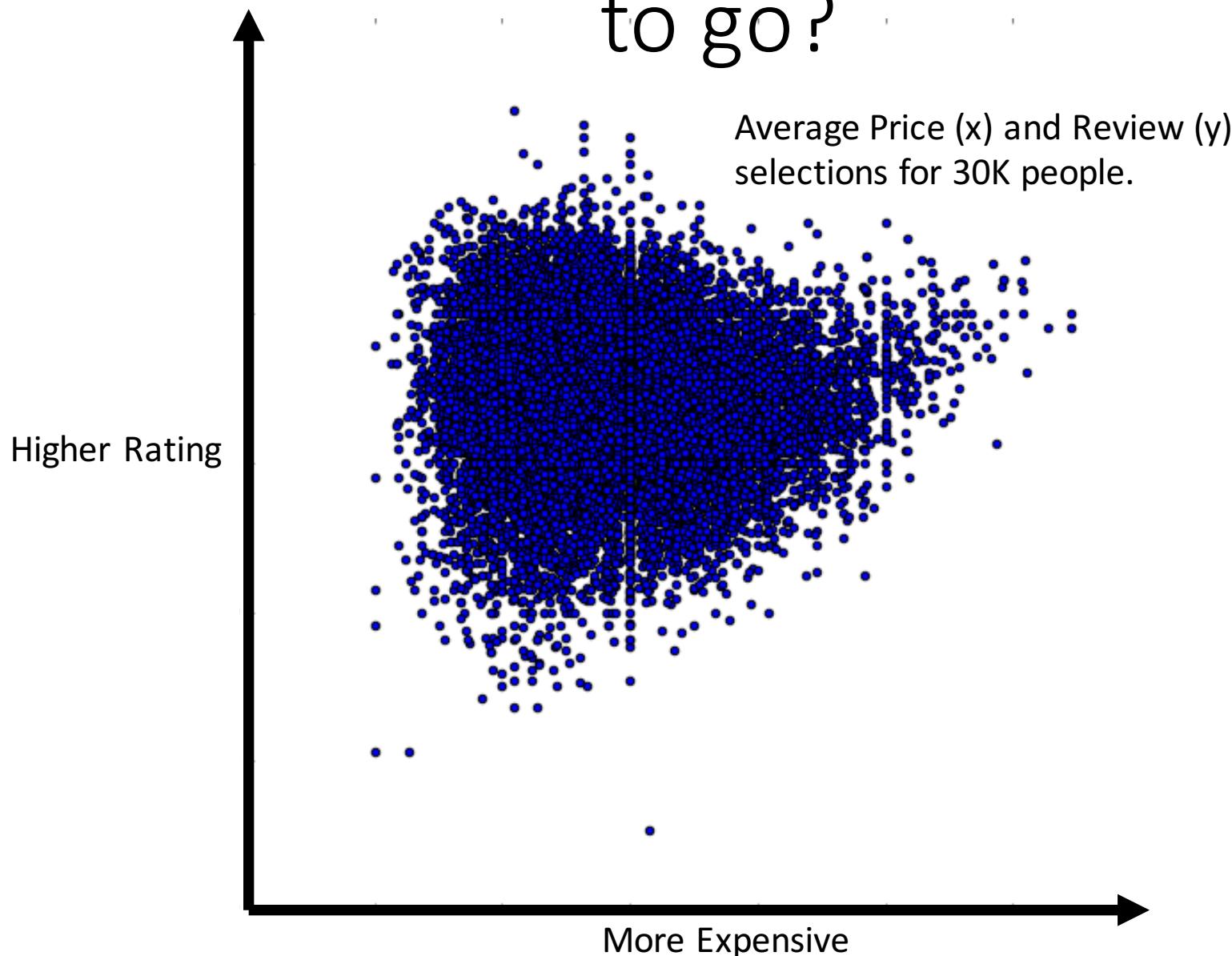
User Selection:

User,  
Review,  
Price,  
Travel Fairness Ratio

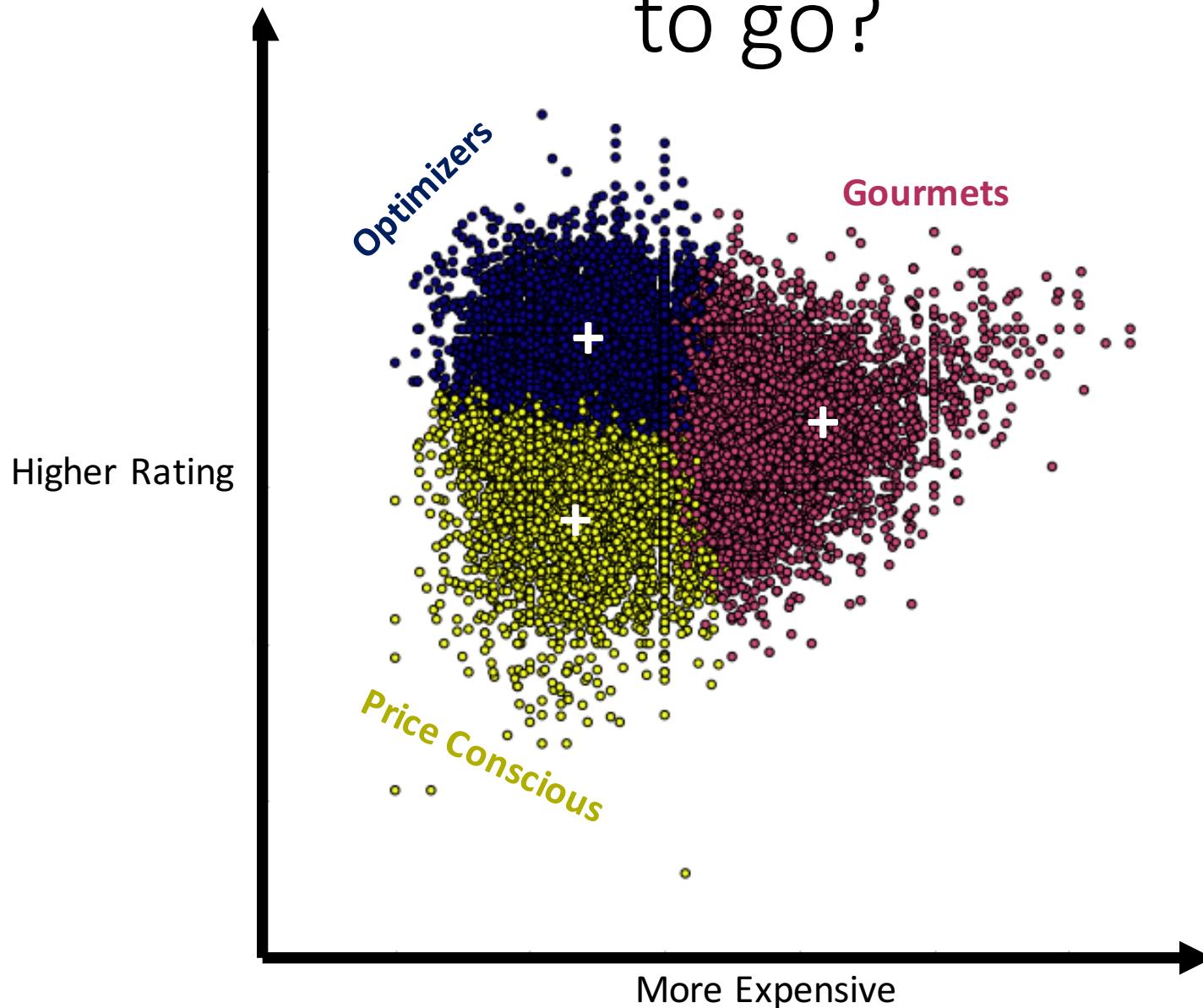


Cosine Similarity

# The Cold Start Problem – Where do people tend to go?

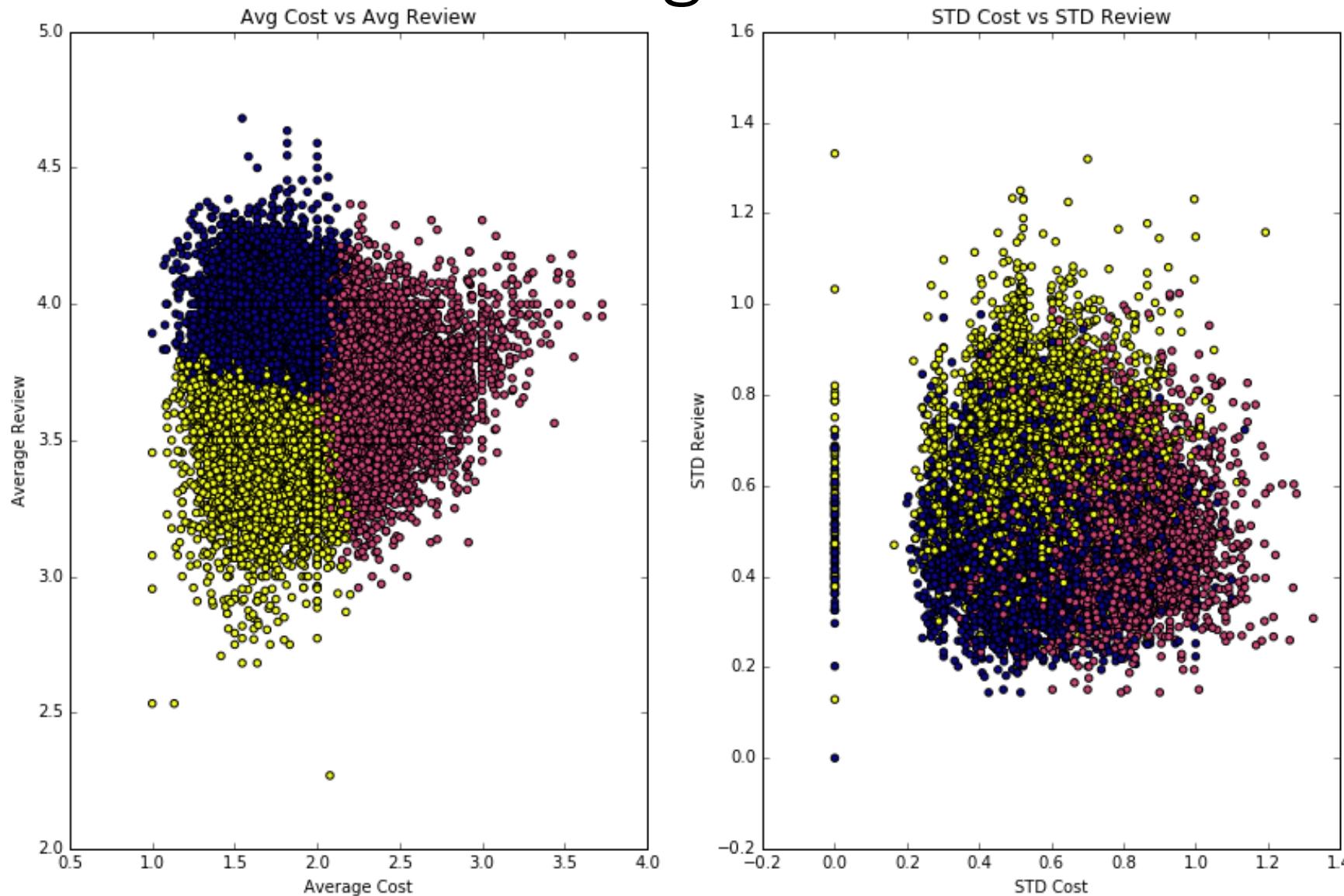


# The Cold Start Problem – Where do people tend to go?



[MeetMeWhere.xyz](https://MeetMeWhere.xyz)

# The Cold Start Problem – Where do people tend to go?



# Rank Locations by Cos Similarity to User Profile

Place Review (1-5)

Price Range (0-4)

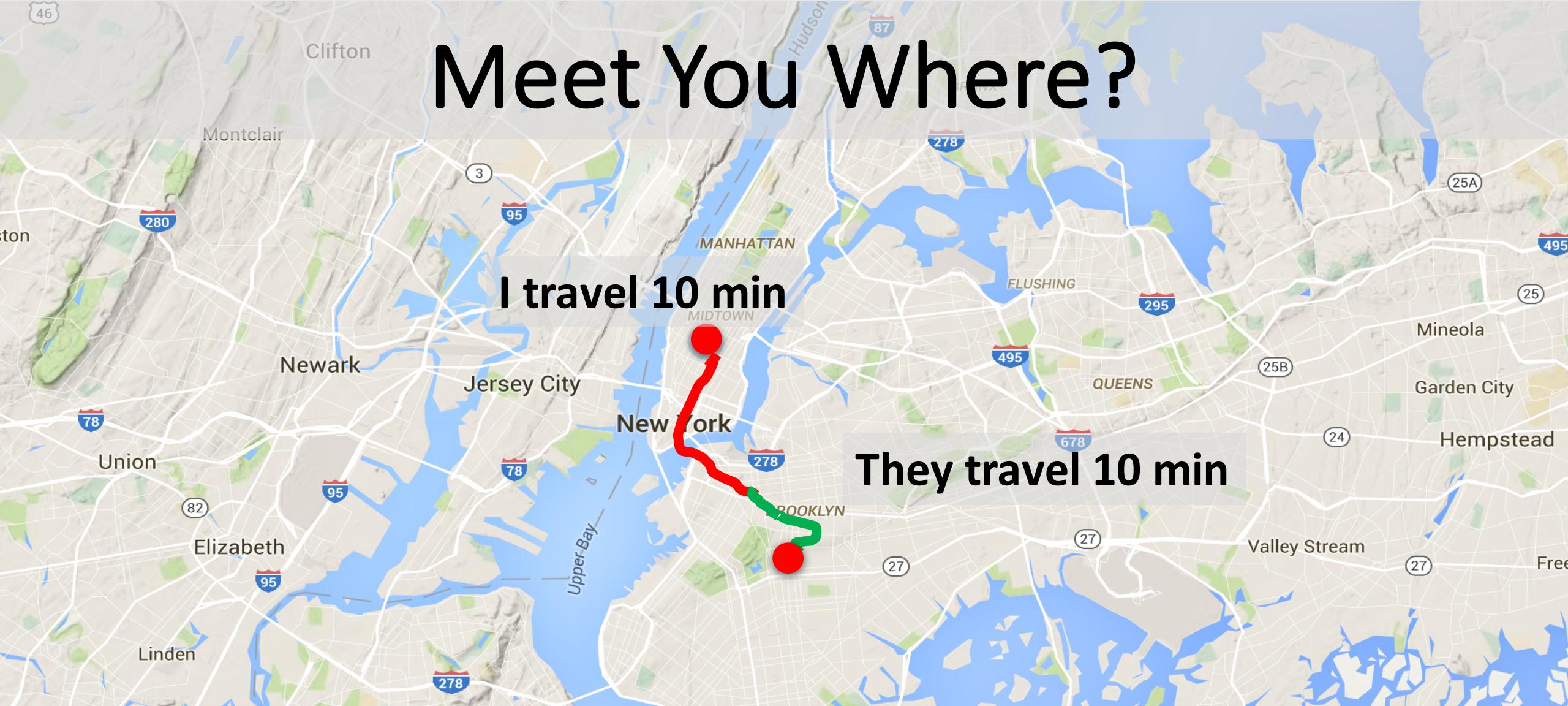
Travel Fairness (0-100%)

$$\nu_{local} = (R \cdot v_i) + (P \cdot v_j) + (T \cdot v_k)$$

$$\nu_{user} = (\omega \vec{R} \cdot v_i) + (\omega \vec{P} \cdot v_j) + (\omega \vec{T} \cdot v_k)$$

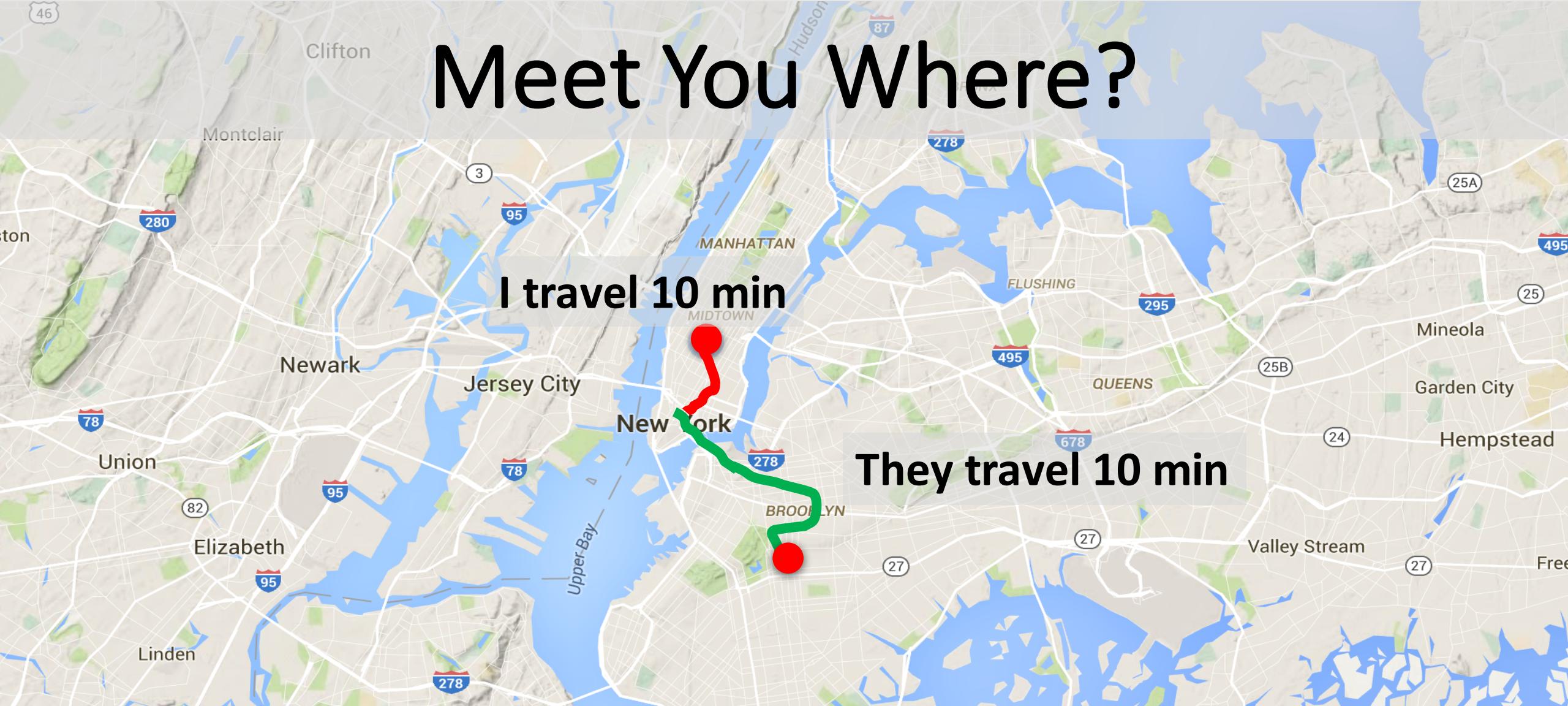
$$Rank = \nu_{local} \cdot \nu_{user}$$

# Meet You Where?



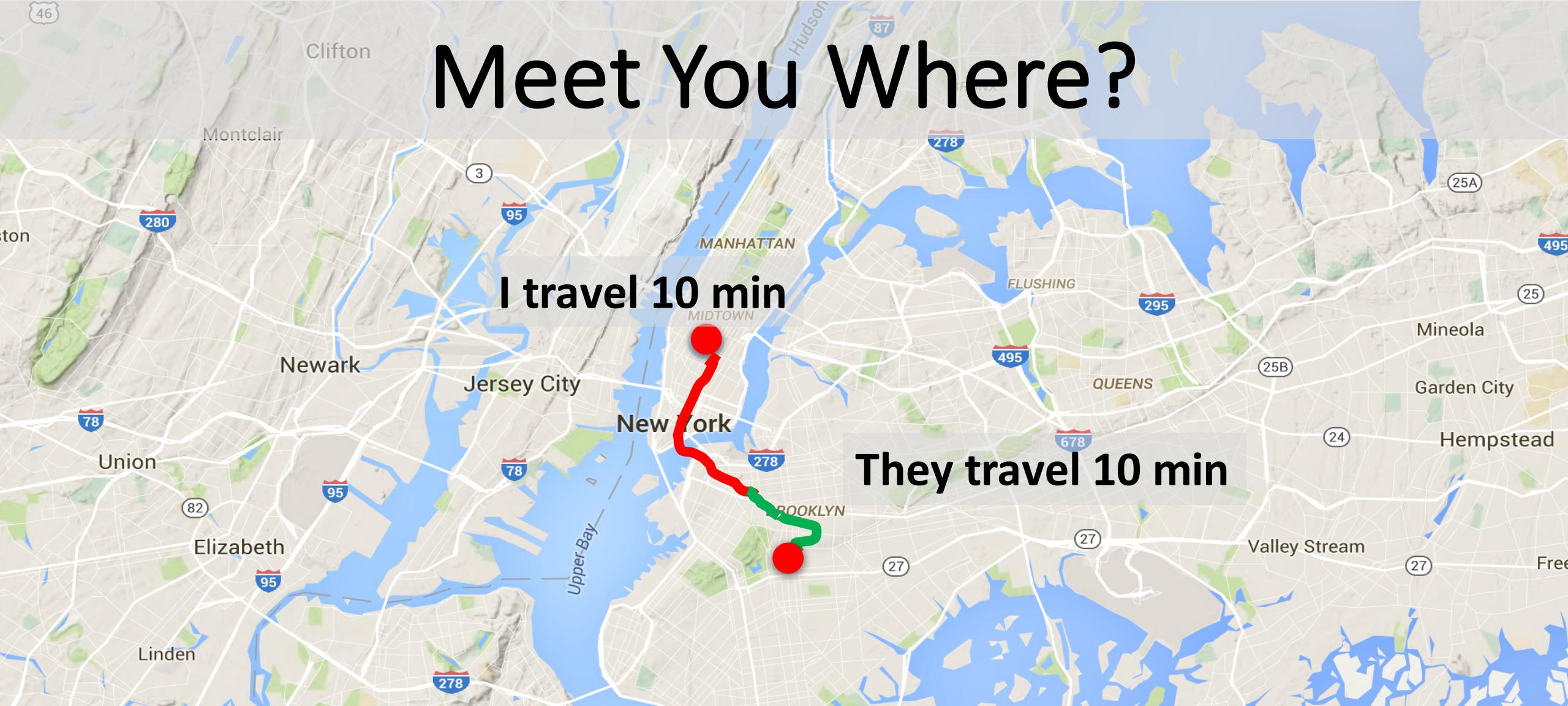
1. Calculate the mid-point based upon projected real time traffic data or public transit delays (google)

# Meet You Where?



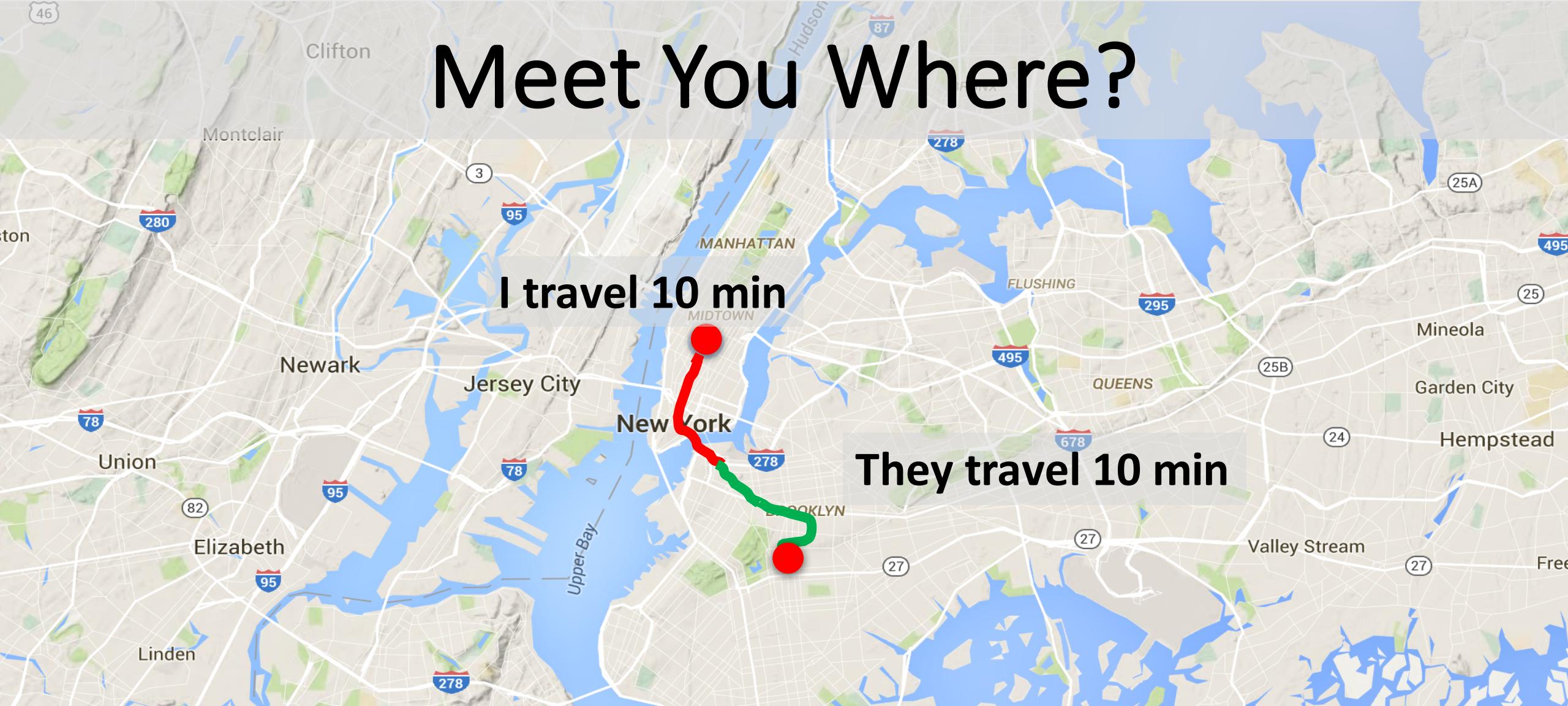
1. Calculate the mid-point based upon projected real time traffic data or public transit delays (google)

# Meet You Where?



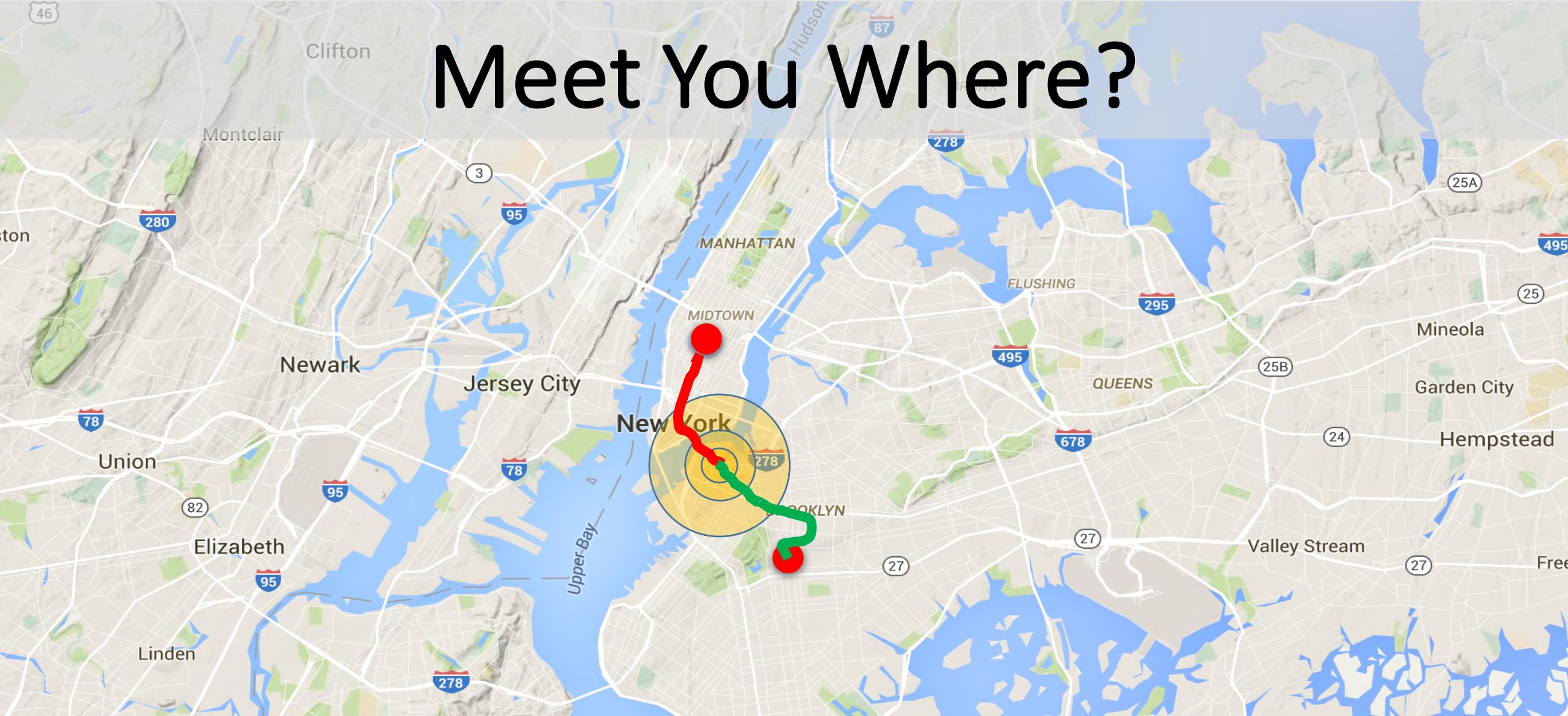
1. Calculate the mid-point based upon projected real time traffic data or public transit delays (google)

# Meet You Where?



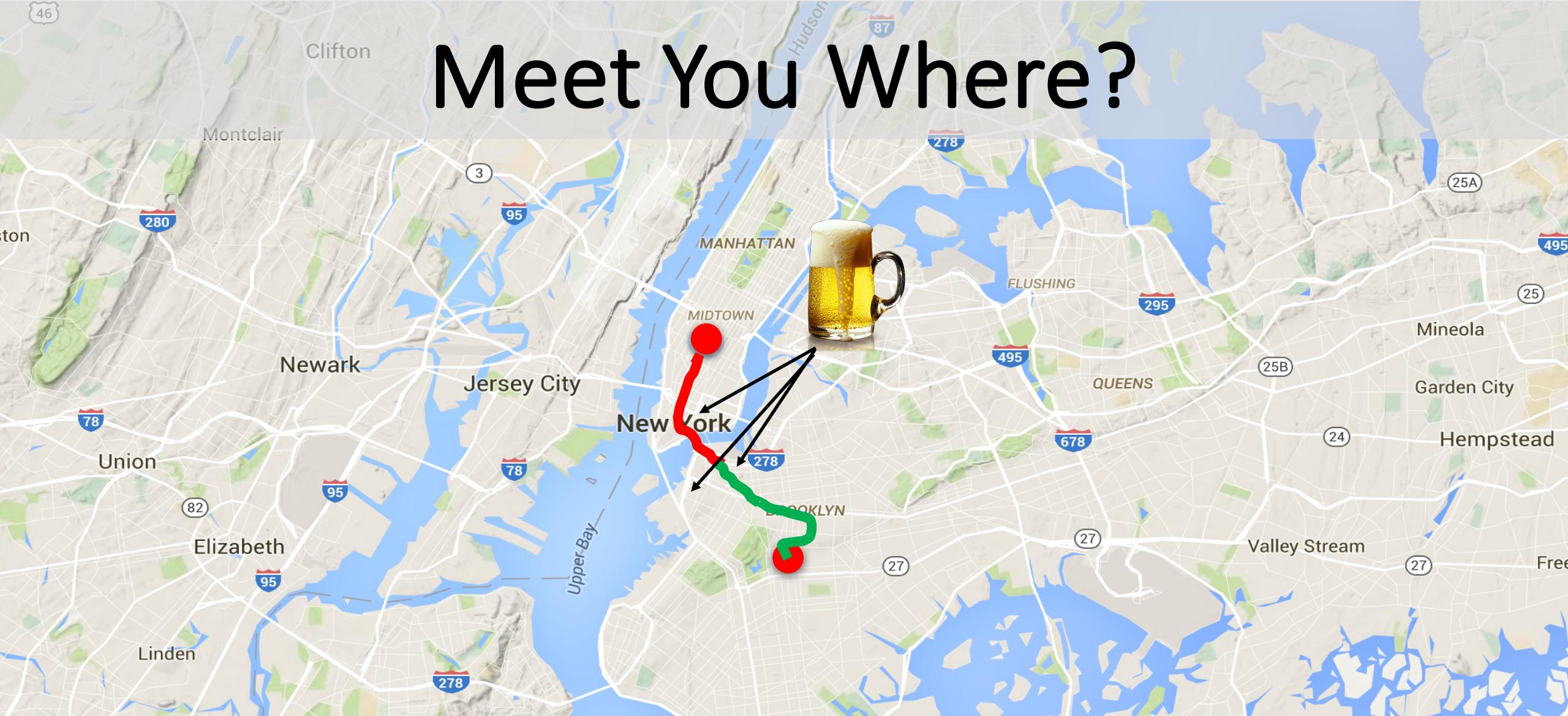
1. Calculate the mid-point based upon projected real time traffic data or public transit delays (google)

# Meet You Where?



1. Search the vicinity of the mid-point for establishments matching the user query (google places API)

# Meet You Where?



The Best One! ... But what's important? Review? Price?  
Distance? Some combination?

# Behind the scenes

