Hello and welcome! Today we'll be presenting our preliminary findings for AGM's delivery expansion initiative.

Our target audience is AGM executives.

BACKGROUND

- **Investigation Scope:** Berkeley store, the San Francisco Bay Area, and BART
- **Goals:**
  - Explore futuristic delivery and pickup options
  - Expand customer base
  - Increase customer purchase frequency
  - Increase revenue
- **Technologies used:** NoSQL and SQL relational databases

Just to give you some background and context..

As you requested, we limited the scope of our investigation to the Berkeley store, the San Francisco Bay Area, and BART.

Our goals are to:

- Explore futuristic delivery and pickup options
- Expand our customer base
- Increase customer purchase frequency
- and Increase revenue

We used NoSQL and traditional SQL relational databases to store the BART map.

We also leveraged technology and algorithms that were relevant to our goals and appropriate for our data so that the interpretation is as accurate as possible.

HOW CAN WE FUTURE PROOF OUR BUSINESS?

a) Delivery Drones

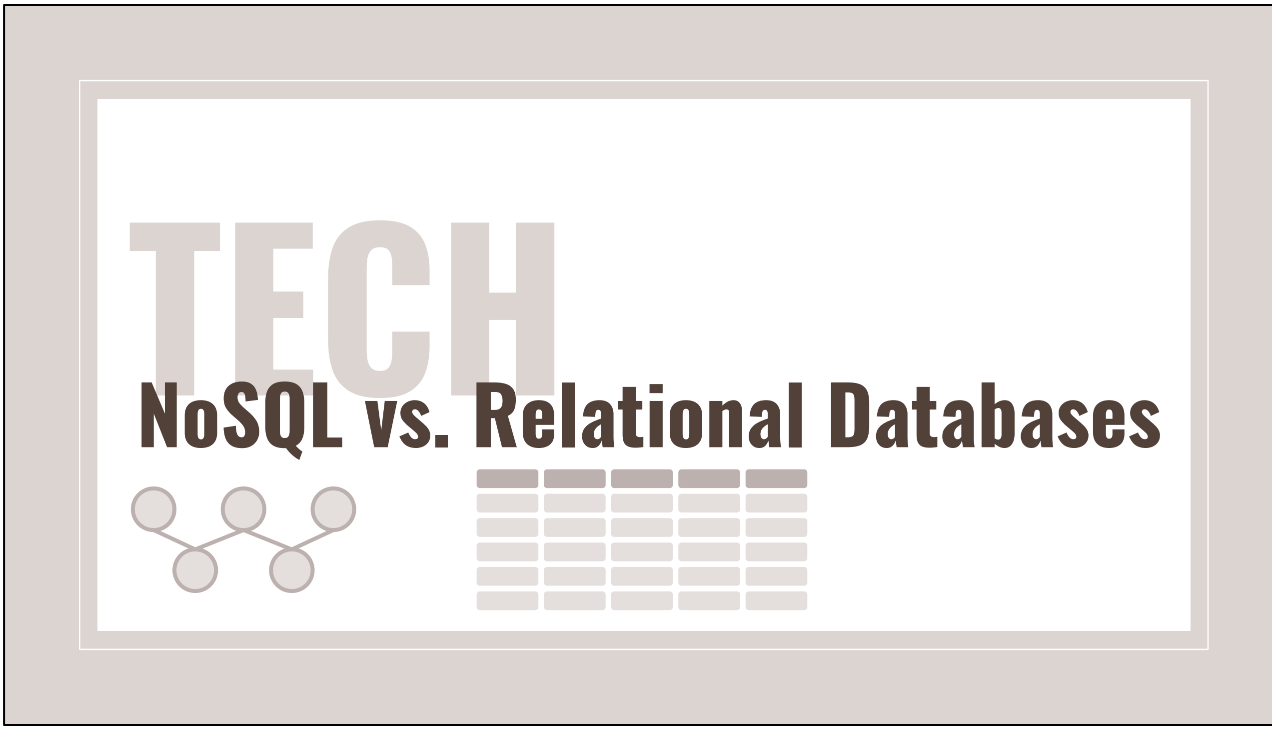
b) Delivery Robots

c) Add pickup locations

d) Delivery via BART ☆

e) Hybrid Approaches

There are several ways to future-proof our business. These are the main options we focused on in our investigation.



Let’s go over the technology we used.

NoSQL Graph Databases

Difference between graph and relational databases?

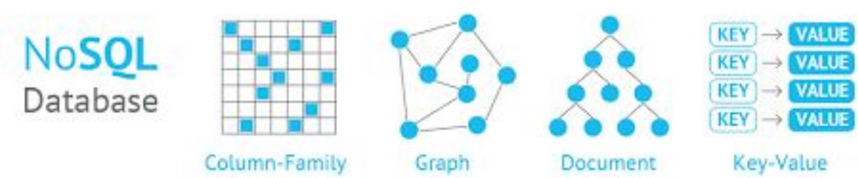
- They store data differently.
- Relational database
  - Tables with columns and rows (structured)
- Graph database
  - Nodes and relationships (more flexible structure).
  - Can run graph algorithms to understand node connections

One of our main technologies was NoSQL graph databases. The fundamental difference between relational and graph databases is in how the data is stored. A relational database stores data in a structured manner using tables with columns and rows. Whereas a graph database stores nodes and relationships, allowing us to store the data without restricting it to a predefined model. This gives us flexibility to run graph algorithms and calculations to understand the connections between nodes.

NoSQL Graph Databases

Why graph databases?

- It’s appropriate for our goal.
- Analyzing delivery routes is naturally represented as a graph.
- Better performance for running graph algorithms (avoid expensive joins in relational DBs)



So why graph databases?
Well, we are interested in the data problem of analyzing delivery routes to reach our goals, which is naturally represented as a graph. We specifically used Neo4j for analyzing the BART stations since the primary focus was on the connections between stations.

With graph databases, we were able to avoid expensive join operations and cross-lookups that we would otherwise encounter with a relational database. Therefore graph databases have better performance when it comes to traversing through data quickly.

How we can use MongoDB

Ways we can use a NoSQL document store like MongoDB:

- Pre-compute optimal delivery routes and store in MongoDB to use as heuristics for A*
- Store traffic data
- Store different point of views like:
  - Customer POV
  - Pick-up location POV
  - Store POV

mongoDB

We can also use a NoSQL document store like MongoDB. Here are some ways we can use it:

- We can use the graph algorithms on the graph database to pre-compute optimal delivery routes, which we can store in MongoDB to use as heuristics for the A star algorithm for finding shortest paths between nodes.
- We can also use MongoDB to store traffic data.
- Or store different point of views like:
  - Customer POV
  - Pick-up location POV
  - Store POV

How we can use Redis

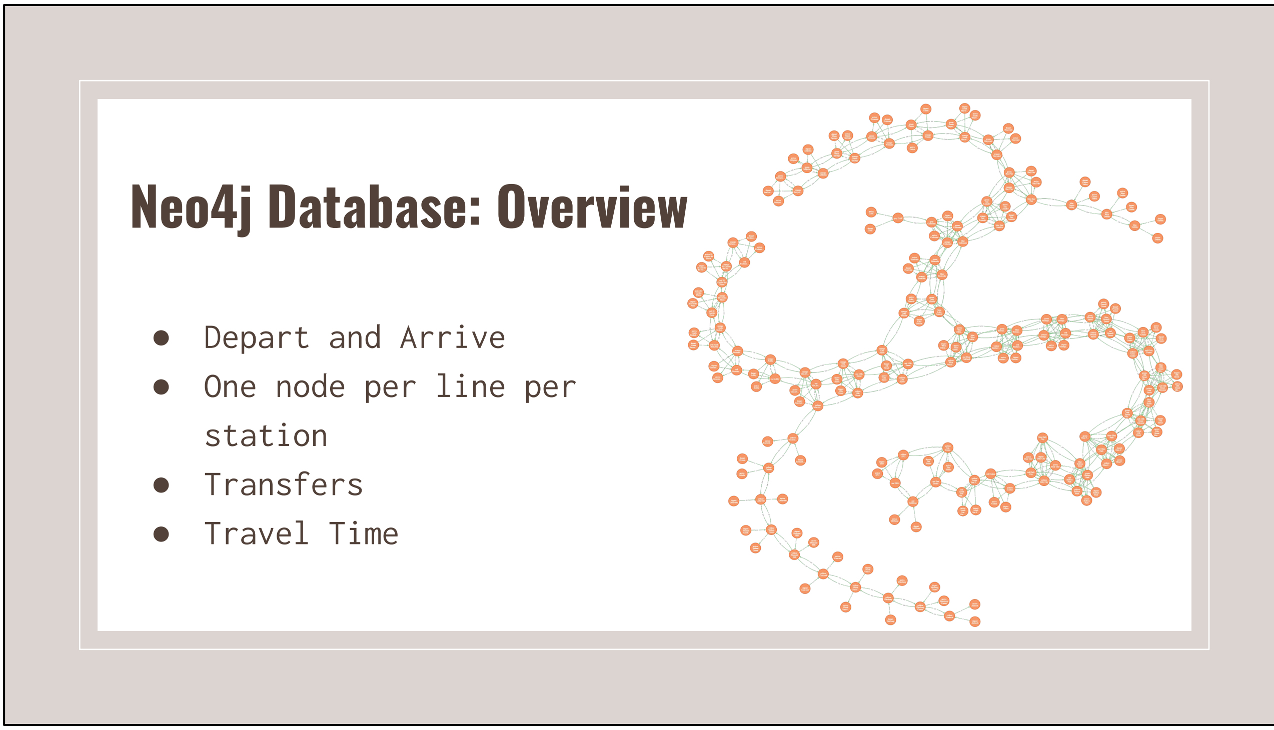
Ways we can use a NoSQL in-memory key-value store like Redis:

- Store delivery locations
  - Key can be delivery vehicle, courier, robot, or drone
  - Log the GPS location and update every 1 minute

Lastly, we can also use a NoSQL in-memory key-value store like Redis. Here are some ways we can use it:

- We can store delivery locations
  - Where the key can be either a delivery vehicle, courier, robot, or drone
  - And we would then log the GPS location and update every 1 minute

01 Methodology

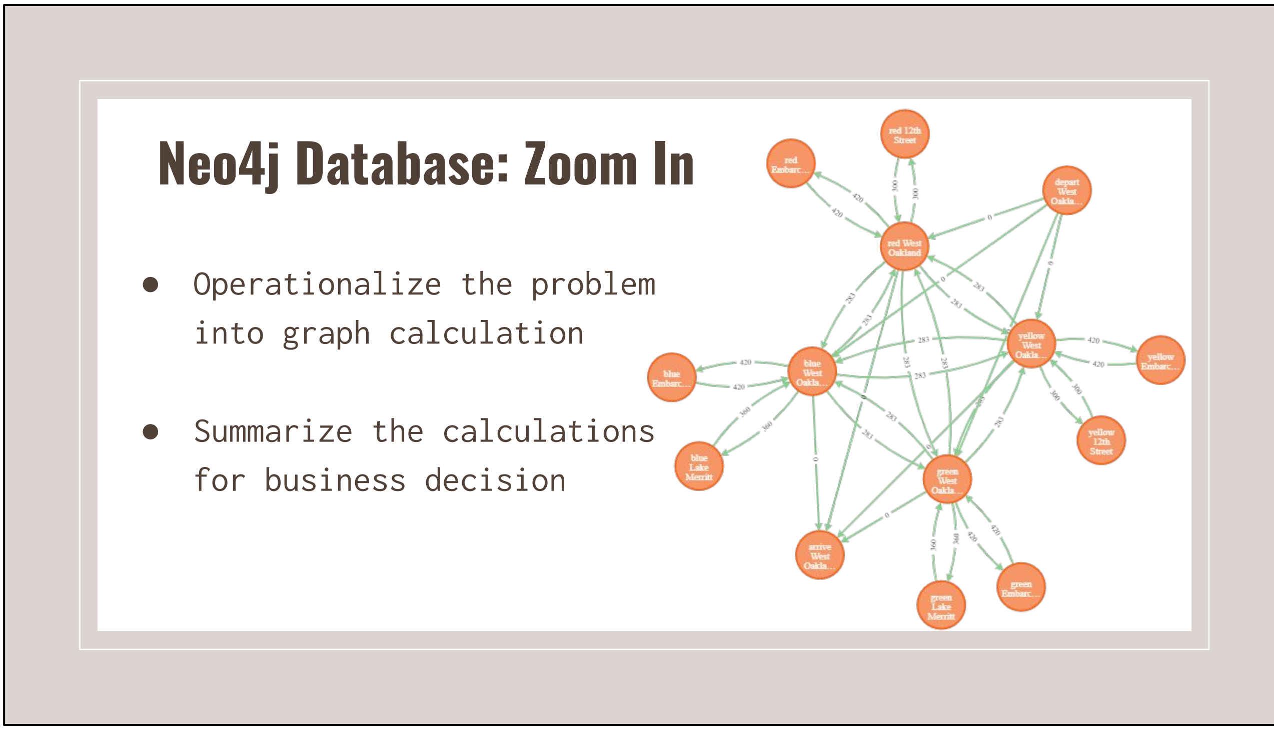


This is what the database looks like on a high level.

We constructed a node for each station with depart and arrive, to enable getting into or out of a station.

We also create additional node per line and station, to reflect the ability of transferring between different lines, and then link all stations based on transfers times.

Finally we link different line/stations based on the travel time.

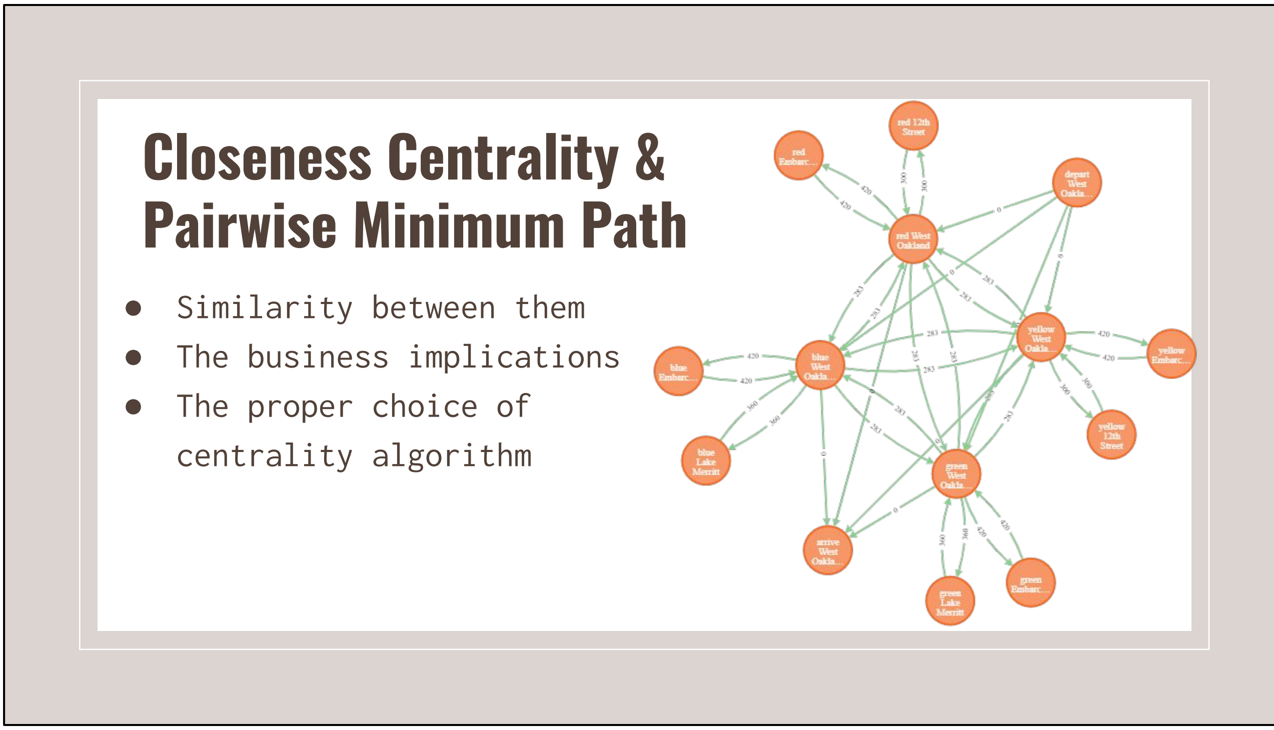


Looking at a small part of the graph around station: West Oakland, this describes exactly what you can do in this station.

But for some of the calculation results, we also need to bring the problem back to the physical concept of stations to draft our actual recommendations.

For example, West Oakland has 6 nodes in the center of the graph, and each node has its own centrality measurement. Therefore, we add up centrality at each station level, since high centrality in one of the nodes doesn't mean high centrality on the station level - after all, they collectively represent the same station!

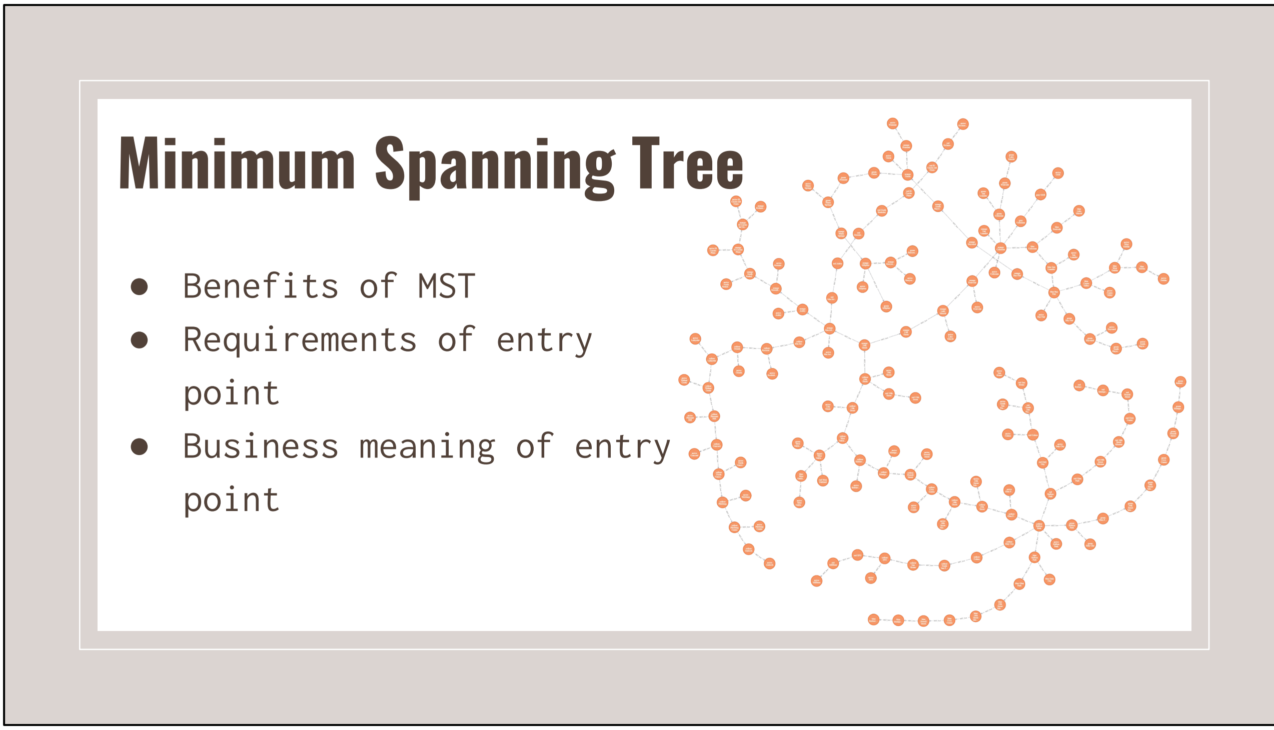
Origin of the graph query: match(n1)-[r:LINK]-(n2) where n1.name ends with ('West Oakland') return n1,r,n2



We use both closeness centrality and pairwise minimize path in our analysis: They both describe the average of shortest path distances between a node and all other nodes.

Since we have a connected map, we sticked with the basic algorithm of closeness. We had to avoid using some of the other alternatives involving reverses calculations (like Wasserman and Faust, Harmonic algorithm), to make sure we can sum the results.

In some cases, pairwise minimum path is more handy: We used it to calculate the closest N stations near a specific station, sum the population up, to measure how many population is reachable by that particular station.



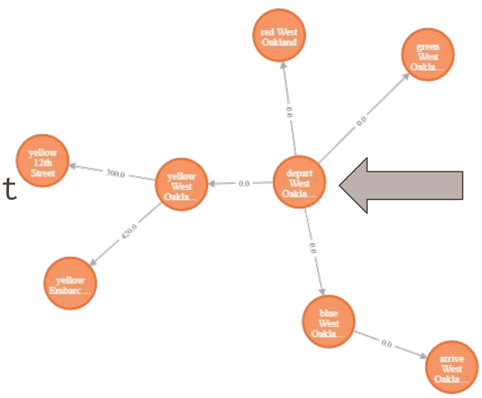
MST is a tree structure path that will visit all nodes in the graph with smallest cost.

MST will simply our graph as you can see, so we essentially only need to deal with one station at a time, with one relationship arriving and another one departing: It is very similar to how most ppl would use BART: imagine you will use a google map navigation to find best route recommendations.

MST calculation requires a starting point. We believe the entry point indicates that if we transit food via BART, where the food is leaving from.

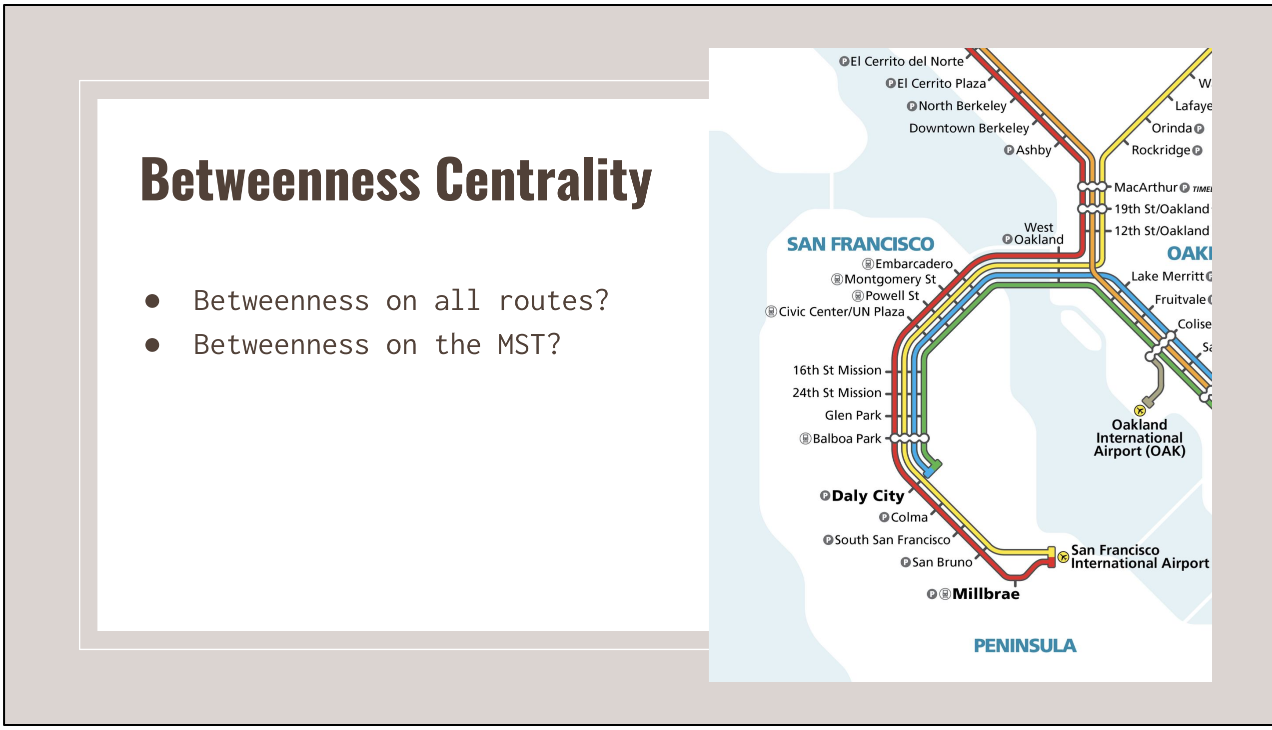
Different Entry Points

- closest to existing store (using Geodesic distance)
- best tree: least overall cost
- most populated station



As a result, we considered three candidates for our entry points into a MST, and thus built three trees respectively.

1. We considered the closest station to the existing Berkeley store
2. We tried to find a best tree: The station that has the minimum aggregated cost to any other station in the tree - same idea as the "station with highest closeness centrality"
3. Most populated station. Which combines two key factors that we will mention later.



Betweenness Centrality:

- It describes for each station, how many existing paths pass through the node.

- Calculating high betweenness based on the all routes will simply gives us stations in the middle with four lines over them; Not too different from closeness centrality;

- Since MST has an entry point, and the stations with high betweenness will always indicate the bridging stations that is outside of the entry point.

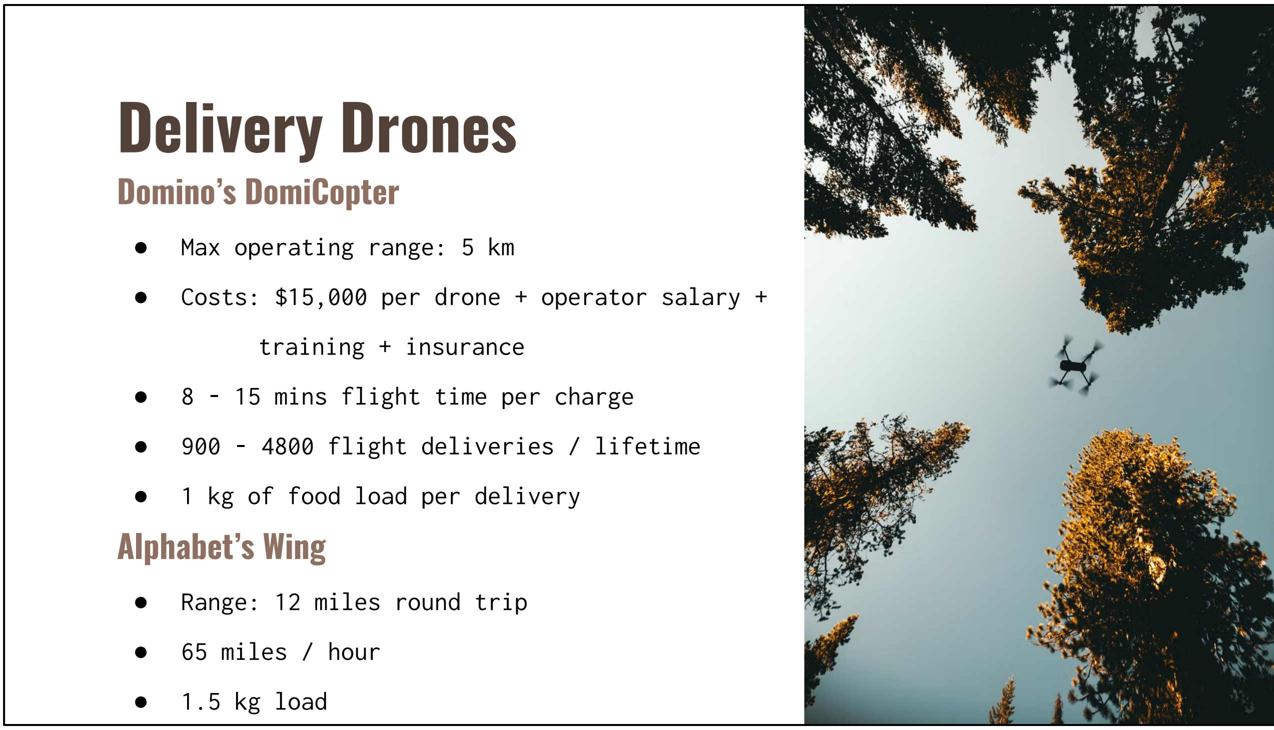
  We interpret that as pivot points that our customers go through a lot when travelling away from the entry point - kind of like “their pivot station - on their way home” kind of thing.

Overall, Closeness vs Betweenness are two separate topics and would tell different stories, and the business should decide which model makes more sense with more business inputs.

02

DELIVERY DRONES/ ROBOTS

- Looked into delivery robots options to future proof our business



- Two main drones we are focusing on are DomiCopter and Wing:
  - Cost: upwards of $15,000 per drone
  - 8-15 mins flight time per charge
  - Able to go 65 miles / hr
- Last mile delivery – how we would use the delivery drones / robots to deliver from the BART station to our customers

Delivery Robots

Starship Technologies

- “Last mile” delivery currently available for university campuses
- 1700 robot fleet making over 10k deliveries per day
- Able to carry 10 kg load / delivery
- Charges $2 / delivery
- Cost: $2500 - 5000 per robot + operator training + salary + insurance
- 2 hours of usage (6 km range) per charge
- Power consumption: 250 W

Starship Technologies’ robots are operational in a few universities with a range of 6km per charge

- charges $2 / delivery
- Cost upwards of 2.5 thousand dollars

03 HYBRID APPROACHES

- Here we will be exploring hybrid approaches to maximize the delivery experience

Futuristic Delivery Comparison

	Advantages	Disadvantages
Drone	▪ **Fast delivery** (no traffic jam) ▪ **Safer food handling** (fewer human contacts)	▪ **Very costly** with current technology ($15k+), poor range and battery ▪ **Weather prone** ▪ Subject to changing policies ▪ Only carry 1 kg load ▪ Privacy concerns
Robot	▪ **Drone-convenience but far cheaper ($3.6k) and carry more load (10kg)** ▪ Potentially safer roads, faster flow, and smaller footprint than cars if have mature algorithms	▪ **Privacy and policy concerns** ▪ **Slower than drones** (15 mins for 1 mile delivery) ▪ Can only travel on sidewalks ▪ 4-6 deliveries / charge ▪ Subject to changing policies
Food Vendor Hub	▪ **Bigger inventory**: can sell in bulk and maybe promotional pricing ▪ **Refrigeration options** ▪ Stable policy ▪ No weather limitations	▪ **Bigger initial capital**: food vendor permits: $330 + $200,000 for average convenience store (proxy for physical store in BART station) + insurance + rent (loc. dependent: ~$5000+ /month) ▪ **More employees needed**

This is a comparison of the different delivery options

1) Food vendor hub - is a storage and / or selling avenue in the BART stations. This is a “baseline” comparison as such entities are established infrastructure.
2) Read out the bolded bullet points

Hybrid Approach #1

BART + Electric Vehicles

- Delivery person (with wagons of meals) takes BART to the customer-nearest station
- 2 approaches:
  1) Notify customers to pick up at the BART station
  2) Meals are picked up by vehicles for delivery

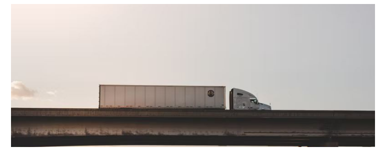
The first approach involves BART + Electric Vehicles

- Read bullet points

If have time:

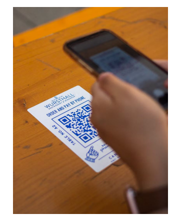
- Electric vehicles for the rising gas prices and smaller carbon footprint favourable for a more sustainable business model

Hybrid Approach #2



BART + Truck / Electric Vehicle + Robot

- Robots / humans take BART to the customer-nearest BART station
- Wagons of meals loaded to trucks with robots
- Truck drives to an epicentre of deliveries away from the station
  - Focus on collaborating downtown office building
- Robot delivers food to the customers' office building floor number
- Robot alerts the customers to pick up the food by the elevator
- Customers "unlocks" the robot via QR code scan of their meal order



Our second approach involves BART, Truck and Robot

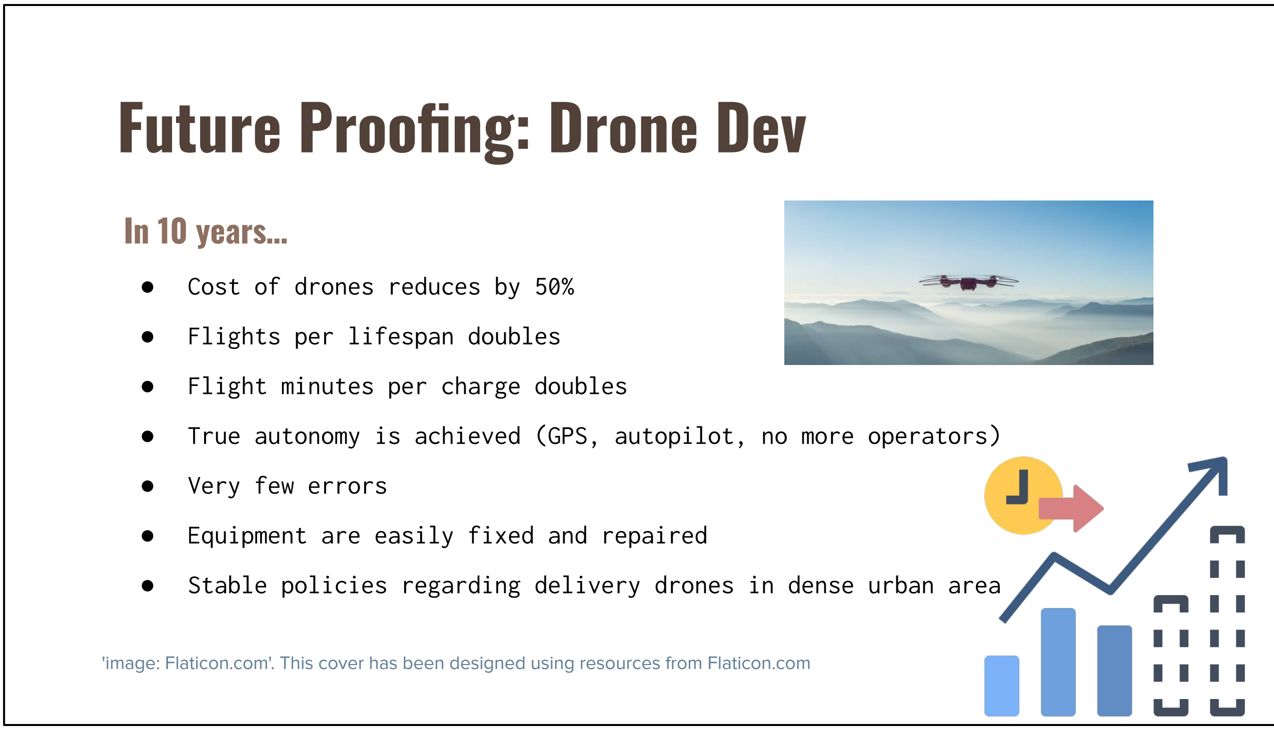
- Robot carry the meals via BART to the trucks waiting at the customer-nearest station
- Trucks are driven to office buildings far from BART stations
- Robots deliver the food to the corresponding building-floor number and alert the customer to pick up their meals via QR code scanning

If more time:

Truck if there are a massive amount of orders - so don't need to drive back and forth for replenishment

Electric vehicles - smaller carbon footprint

Which is more economically advantageous depends on the load they are carrying and how many trips needs to be made



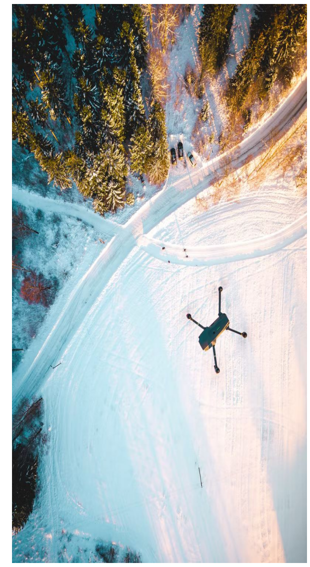
Drone icons created by Pixel perfect - Flaticon

At present drone technologies are not mature enough with cost, battery and range issues. There are also many regulations such as FAA rules that prevents drones from flying over large groups of people. Drones will be included in our delivery strategies when drones *fly over these obstacles*.

Hybrid Approach #3: Future

BART + Electric Truck + Robot + Drone

- Robot to deliver wagons of meals to the nearest BART station
- Wagons of meals loaded to electric trucks with robots / drones
- Truck drives to an epicentre of deliveries away from the station
- Robot / drone delivers to the customers’ designated pick-up location
- Robot / drone alerts the customers to pick up the food
- Customers “unlocks” the robot / drone via QR code of their meal order



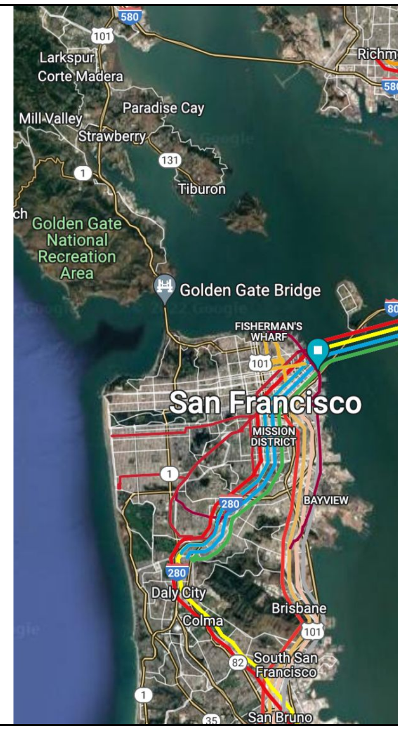
With future proofing in mind, this approach is a spin-off of the previous idea, but includes drone-delivery when the technology matures and enables deliveries beyond downtown office buildings or elevator entrances

So keeping the future approaches in mind and our data-driven methodology what is the best recommendation for pick up or delivery location?

Considerations

Optimization problem with multiple variables

- Types of locations
- BART travel times
- Population
- Current store and customers
- Future of transportation



We've considered multiple variables and approached this as an optimization problem with different factors.

We looked into different types of locations, delivery vs pickup.
We agreed in the beginning that BART will be the basis of our recommendation, so the graph network of BART stations and travel times was our primary source of data.

Population was another key factor, driven by the goal to acquire new customers
And we set it against the context of our current store, initial cost and the research on future of transportation

Pick Up Location

Have a location in a highly populated area that will attract new customers

- High population
- Away from existing store
- High betweenness
- Transfer station

Best Location:

- Balboa Park

When choosing a pick up location, meaning that customers come to us for their orders, we were looking for high population areas that will bring in new customers.

That's also driving the need to set up a location away from the current store which at the Ashby station.
We also want a location with high foot traffic which we translated as high betweenness, meaning that in order to get to other stations in the BART network people need to hit this station more frequently.

Balboa park showed high betweenness, another contributing factor could be that it is a transfer station. It was also at the top of most populous stations and is surrounded by stations that are highest in population across the whole area.

Delivery Using BART

Location that will be most efficient for deliveries to all stations

- Highest centrality
- Lowest cost MST
- Close to existing store
- Close to populous area

Best Location:

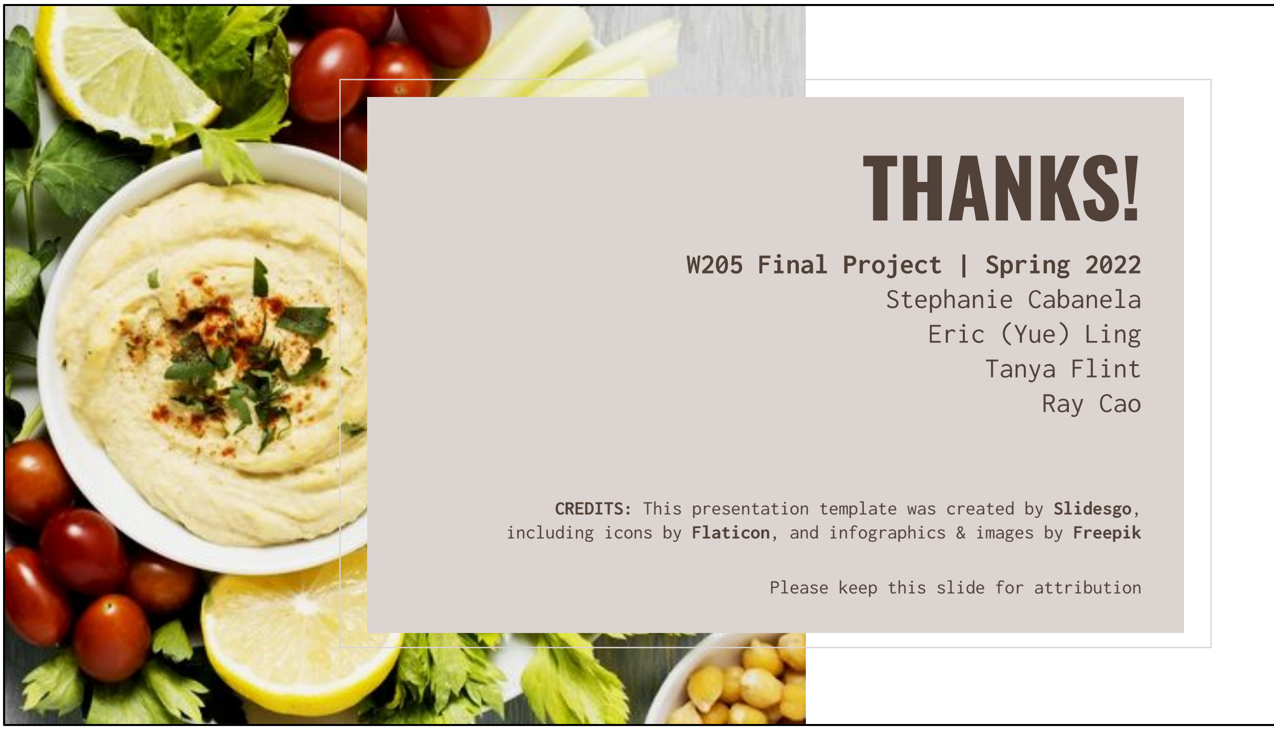
- West Oakland

And finally let's look at it from another angle of choosing the delivery hub, meaning that this will be a departure point for deliveries. Here we're looking for the station that will be most efficient at reaching the whole BART network.

And we're optimizing the time it would take us to get to any other station from this hub. We started with the centrality measure, for any given hub what is the average distance to all the other stations. And to double check the results we also ran the Minimum Spanning Tree algorithm, which calculates the time it would take to visit every possible station from this hub. We're thinking speed of delivery will be a key factor and optimizing on time is more important for this location.

West Oakland scored very high on both measures. It may also be beneficial for restocking or warehousing as this station is closer to our existing store, compared to Balboa Park. And West Oakland being outside of the San Francisco city limits should have a lower real estate cost.

We think West Oakland location would be the most optimal solution for both delivery and pick up, it is close enough to most populous area, and set us up for the future of transportation that may include using drones, robots or self-driving vehicles.



Thank you for listening! We hope you found this informative and interesting. Let us know if you have any questions.