AGM: Business Case

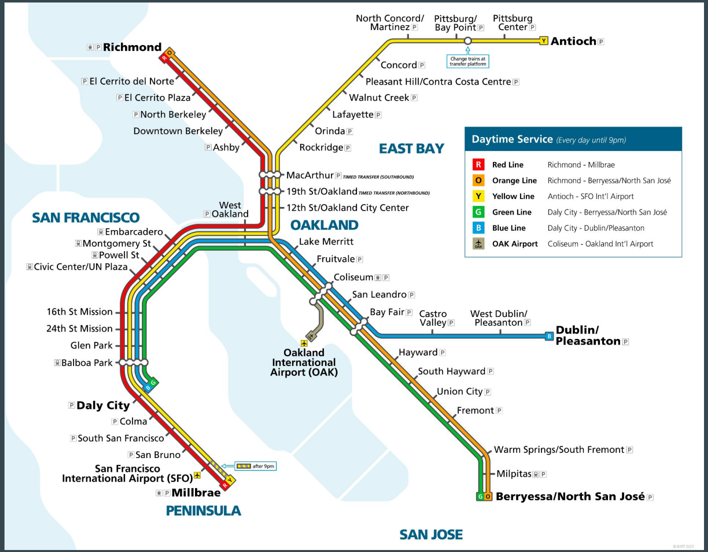
Ricky Pang, Stephanie Andrews, Sameer Karim, Michael Guldberg

Context

AGM's vision for the future revolves around:

- Revolutionizing its delivery operations through innovative technologies and strategic expansions.
- Using NoSQL databases, including Neo4j, MongoDB, and Redis, aligning with objectives of adding more pickup locations, utilizing public transportation, and integrating delivery drones and robots.
- Designing to optimize delivery routes, streamline inventory management, and enhance operational efficiency, positioning AGM at the forefront of the evolving retail landscape.

Using Public Transit to Optimize Delivery



Example Details:

- Use BART for bulk transport between central kitchen(s) and distribution points
- Use delivery drones for last-mile delivery
- Also leverage BART stations as a pick-up location for commuters
- AGM needs to be able to optimize and change routes efficiently given both historical and real-time BART data

Why Neo4j?

- Effective at visual modeling
- Can dynamically optimize routes for BART and delivery drones in real time
- Allows for better understanding of network intricacies, in turn creating opportunities for saving money and better service

Why not use a relational database?

- May struggle with the complexity and fluidity of AGM's needs
- Not able to produce a visual representation of the network like Neo4j would

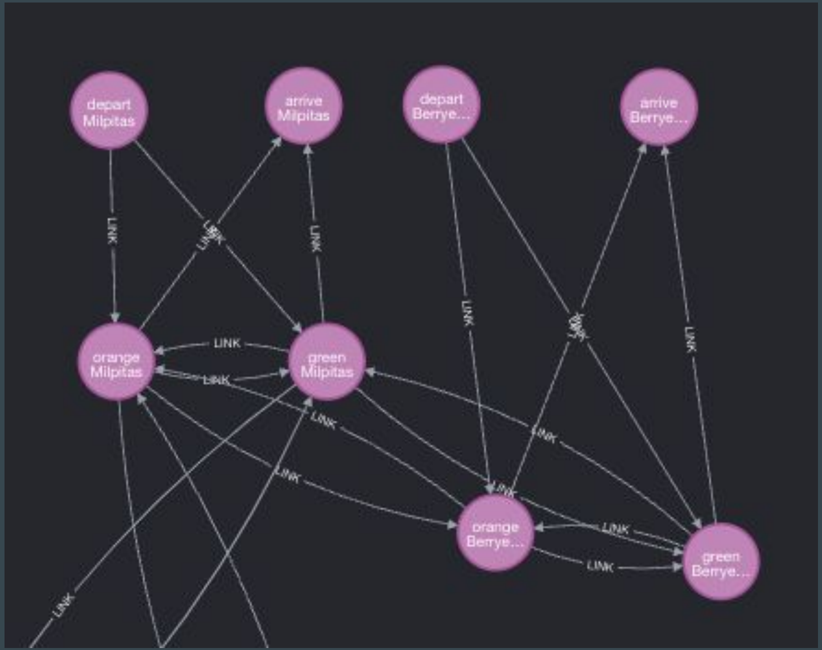
Selecting new AGM pickup locations

<u>Why Neo4j?</u>

- Effective at modeling complex relationships as nodes and edges, which can represent factors like geographic proximity, traffic patterns, customer demographics, purchasing patterns, and delivery routes
- Can leverage graph analysis and algorithms like centrality measures to identify potential pickup locations
- Integrates with spatial data such as coordinates of customers, pickup locations, and delivery routes

<u>Why not use a relational database?</u>

- Less effective at modeling complex relationships
- Less native support for analyzing spatial data
- Not as well-suited for real-time analysis of data



redis Operations that benefit from near-instant updates

General Business Examples:

- Weather station data
- Multiplayer game score tracking
- Streaming data, eg stocks with quotes

AGM Business Examples:

- Bart stalls and status updates
- Item stock and pricing
- Customer average wait time
- Drone status

Why Redis?

- In-memory key-value storage and instantaneous update
- Redis is in-memory and key-value database so if we query by key the result is extremely fast.
- Accessing and updating data in Redis is far faster than with a relational DB.

Why not use a relational database?

- Queries that are not by key are slow and exhaustive searches.

Adding pickup locations at BART stations

Why MongoDB?

Dynamic Pickup Location Management:

- Allows for storage of unstructured data
- Efficient management of pickup location information
- Accommodates changes and additions seamlessly

Real-time Order Tracking and Optimization:

- Suitable for tracking customer orders
- Optimizes delivery routes in real-time
- Integration with AGM's order management system and BART's scheduling data

Scalability for Future Growth:

- Horizontal scalability for seamless scaling of database clusters
- Accommodates increased data volume and transaction throughput
- Supports anticipated growth in demand without compromising performance or reliability

Why not use a relational database?

Schema Flexibility Issues:

- Requires a predefined schema
- Requires schema alterations to change data structure
- Does not accommodate heterogeneous data types as readily

Horizontal Scalability Issues:

- Not as suited for growing volume of data generated by an expanding network with complex and rapidly changing data models

Geospatial Querying Capabilities:

- Often lacks native support for geospatial queries
- Location-based operations, such as finding nearest pickup locations and optimizing delivery routes, may require additional tools or extensions

Harmonic Centrality Algorithm - Picking a Main Distribution Center

name	closeness
green West Oakland	0.160198
yellow West Oakland	0.159922
blue West Oakland	0.159903
red West Oakland	0.159674
green Embarcadero	0.154619
yellow Embarcadero	0.154551
blue Embarcadero	0.154371
red Embarcadero	0.154339
orange Lake Merritt	0.154278
green Lake Merritt	0.154140
orange 12th Street	0.154075
yellow 12th Street	0.153839
blue Lake Merritt	0.153784
red 12th Street	0.153544
yellow Montgomery Street	0.150553
green Montgomery Street	0.150458
red Montgomery Street	0.150370
blue Montgomery Street	0.150246
arrive West Oakland	0.147538
orange Fruitvale	0.147500

These are the stations that have the highest degree of closeness to all other stations on Bart

Based on this we recommend that AGM establishes a main distribution center within one of the top stations:

1. West Oakland
2. Embarcadero
3. Lake Merritt
4. 12th Street
5. Montgomery Street

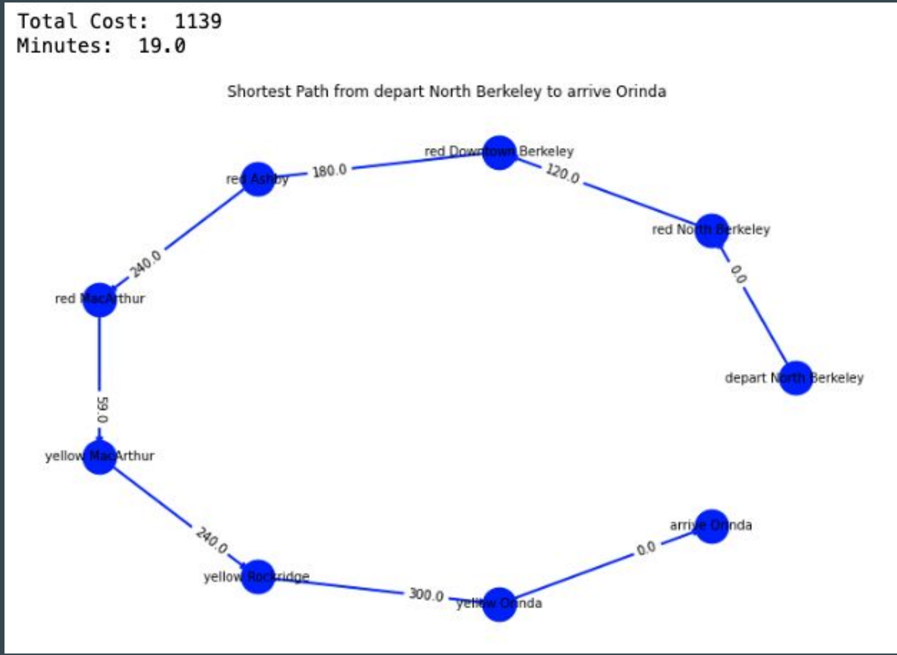
Creating a Distribution Network - Louvains Algorithm

	name	community	intermediate_community
0	arrive 16th Street Mission	7	[3, 7, 7]
12	arrive Civic Center	21	[21, 21, 21]
24	arrive Balboa Park	29	[13, 29, 29]
42	arrive Ashby	31	[11, 31, 31]
66	arrive Bay Fair	47	[15, 83, 47]
93	arrive Coliseum	51	[23, 51, 51]
112	arrive Embarcadero	59	[39, 59, 59]
130	arrive Antioch	63	[9, 71, 63]
160	arrive 12th Street	67	[1, 5, 67]
175	arrive Colma	85	[25, 89, 85]
194	arrive Berryessa	91	[17, 57, 91]

Using Louvains algorithm we determined that there are 11 communities within the BART Station mapping.

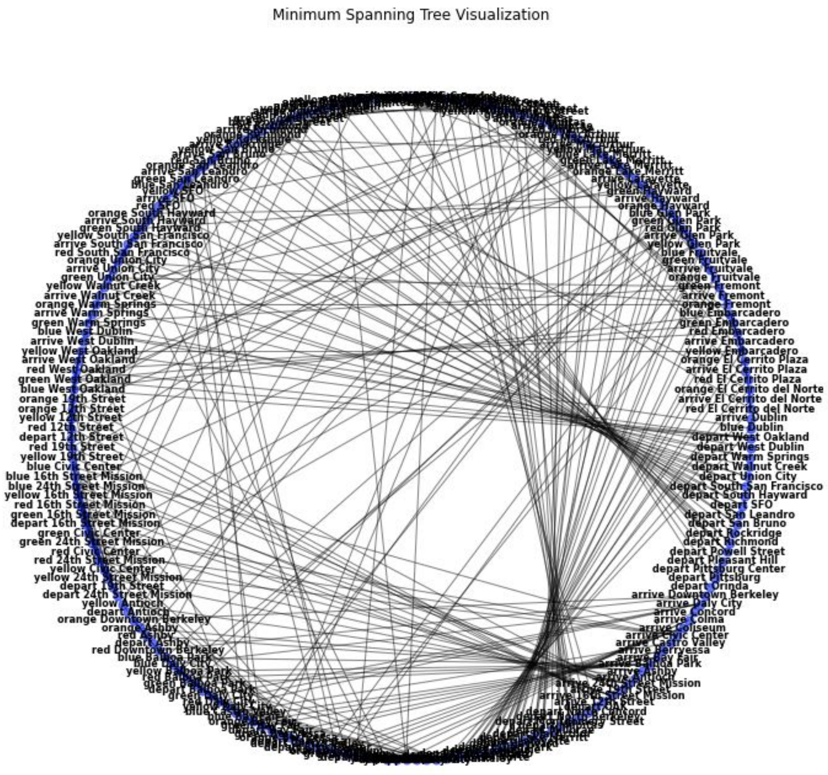
We suggest that AGM sets up a network of distribution end points within these 11 communities that receive deliveries from the main center.

Note: This Dataframe shows only one station per community for the sake of demonstration, each community has 2-6 stations within it.



Dijkstra’s Shortest Path Algorithm

Minimum Spanning Tree



Thank you!