

#FABCONSQLCON2026

FABCON

Microsoft Fabric
COMMUNITY CONFERENCE

SQLCON

Microsoft SQL
COMMUNITY CONFERENCE

ATLANTA MARCH 16 - 20, 2026

Smart Routing: Real-Time Transport Optimization with Fabric

Solving the Vehicle Routing Problem with PyVRP on Microsoft Fabric

Agenda

1. Meet the speakers
2. The Problem
3. Tech Stack
4. Solutioning & Demo

Meet the Speakers



Emmanuel Huygens

Cloud & Data Engineer

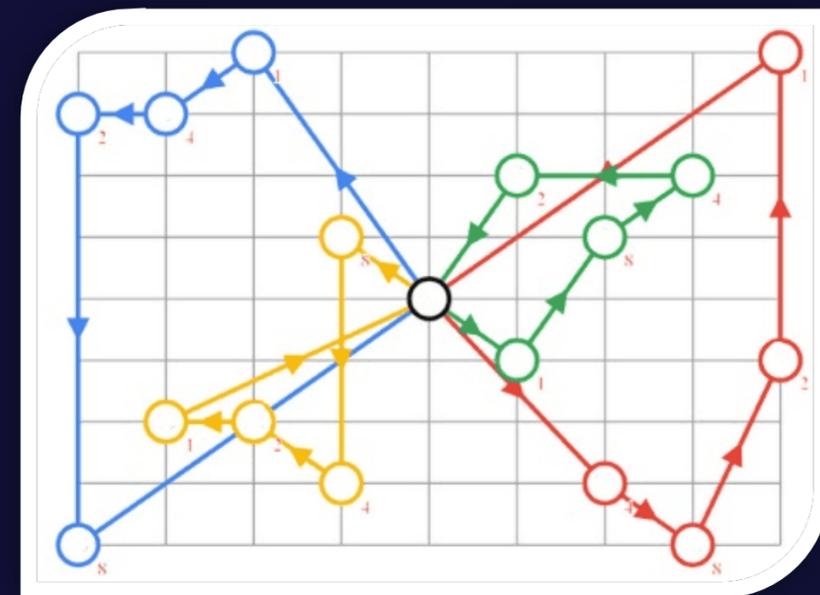
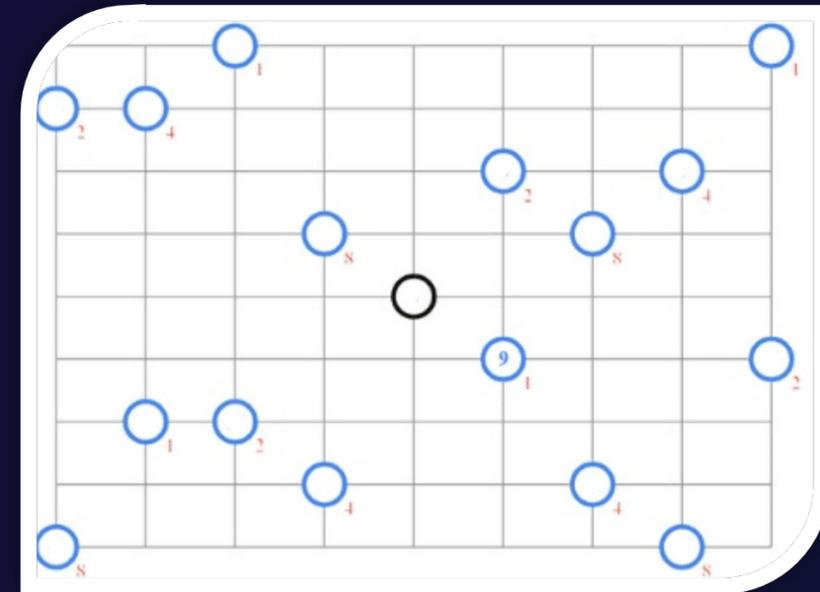


Robert Leal

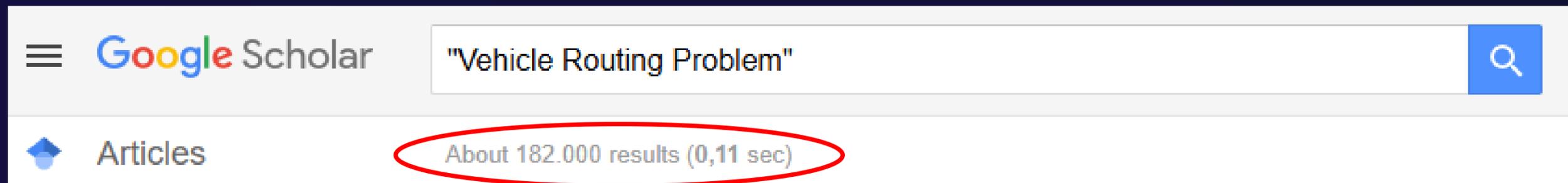
Data & AI Engineer

The Vehicle Routing Problem (VRP)

- Truck dispatching problem, travelling salesman problem ...
 - "What is the optimal set of routes for a fleet of vehicles to traverse to deliver to a given set of customers?"
- **NP-hard problem** (*Non-deterministic Polynomial-time hard*)
 - Gets exponentially more challenging with each new variable
e.g. more vehicles, more stops, more constraints
 - Requires heuristic approach
- Common constraints VRP must solve for:
 - *Vehicle capacity, delivery time windows, load balancing, pickup and delivery pairing, Q depots ...*



VRP Infamy



ORION *On-Road Integrated Optimization and Navigation* (2016)

- Estimated cost of 250m USD
- Provides optimized sequence for package deliveries.
- Approx. 55k drivers rely on it
- Projected annual savings exceed 300m USD



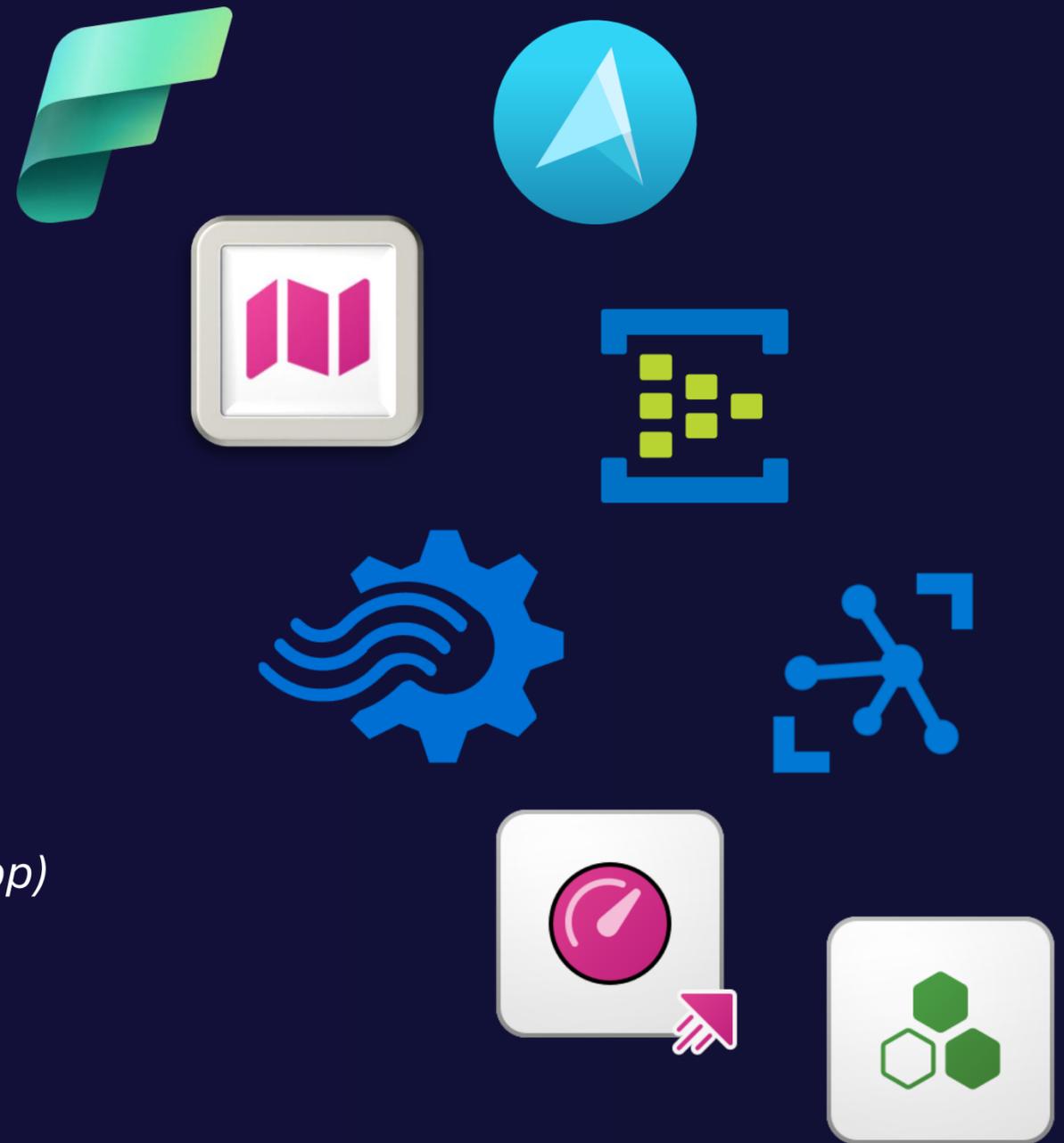
Last Mile Routing Research Challenge (2021)

- Competition to train models on delivery routes
- Bridge algorithmic optimization with real-world driver knowledge
- 220 Participating academic teams
- Drivers frequently deviate from computed routes due to tacit knowledge of traffic patterns, parking, etc..

To help solve this problem, we need data..

With data, we can use:

- **Microsoft Fabric** *as the unified data platform*
- **Fabric Maps** *as the visual layer*
- **Azure Maps API and PyVRP** *for route optimization*
- **Eventhub and Eventstreams** *for real-time data ingestion*
- **IoT Datasources** *for fleet telemetry (simulated using FunctionApp)*
- **Real-Time Dashboards and Live Maps**
- **Fabric Data Agents** *for interactive fleet feedback*
- *Bonus: Model Planetary Computer in Fabric Maps*



Microsoft Fabric : Unified Fleet & Logistics Platform

Real Time Intelligence : Fleet Telemetry



Eventstreams

Vehicle telemetry ingest



Eventhouses

Fleet KQL store



RTI Dashboards

Live fleet health, driver alerts & route performance

Medallion Architecture : Delivery Orders



Bronze

Raw order data



Silver

Validated orders



Gold

Order dashboards



OneLake : Unified Storage

Centralised delivery & order data lake



Power BI : Order Dashboards

Order & Delivery Analytics

CourierIQ : Fabric Data Agent



Conversational AI for Ops & Dispatch

"Which routes are delayed today?"

"Show me courier performance this week."



Microsoft Copilot Integration

Powered by Azure OpenAI & Fabric semantic models

Azure Maps : Visual & Live Feed Layer



Real-Time Vehicle Positions

Live GPS tracking & route overlays on interactive map



Order Depth Map Visualisations

Hexbins, heatmaps, planned routes, delivery locations & Microsoft Planetary Computer

Shared Fabric Capabilities



Security & Governance



Unified Compute



Power BI Integration



Copilot / AI Assist



Cost Management



Data Factory Pipelines

Case Study: Beltway Couriers

- Logistics firm
- Handles parcel deliveries to residential areas
- Greater Houston Area
- 3 delivery trucks: AC-01, AC-02, AC-03
- Approx. 50 deliveries daily
- Morning & Afternoon shifts



Beltway Couriers: Logistics Manager



Daily orders



ETL Medallion



Order insights

Beltway Couriers: Logistics Manager (Demo)

Order book dashboard

Refreshes daily



Order filter

All

Priority

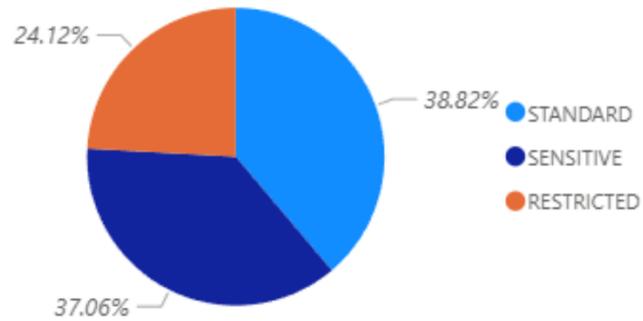
Product name

False

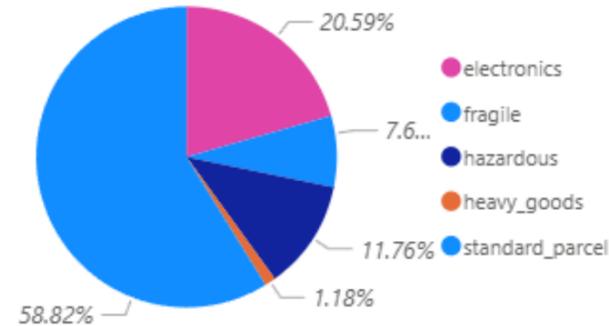
True

All

Shipping clasifications



Product categories

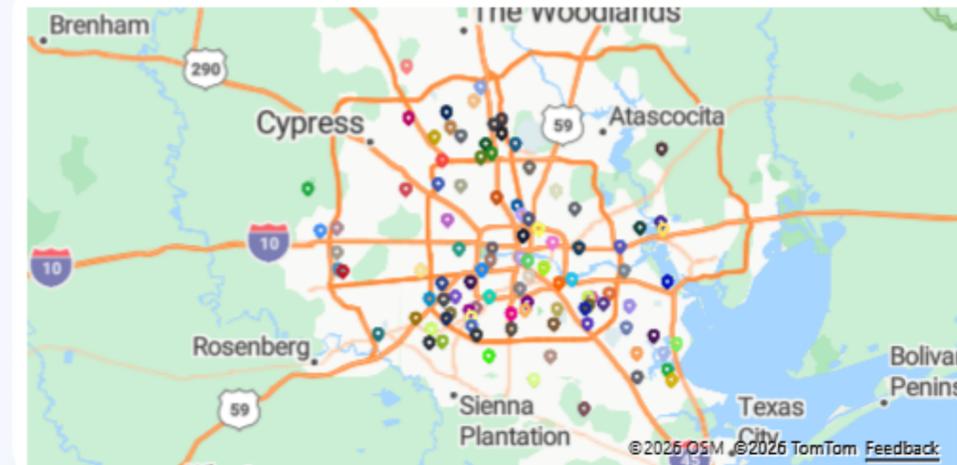


Order details

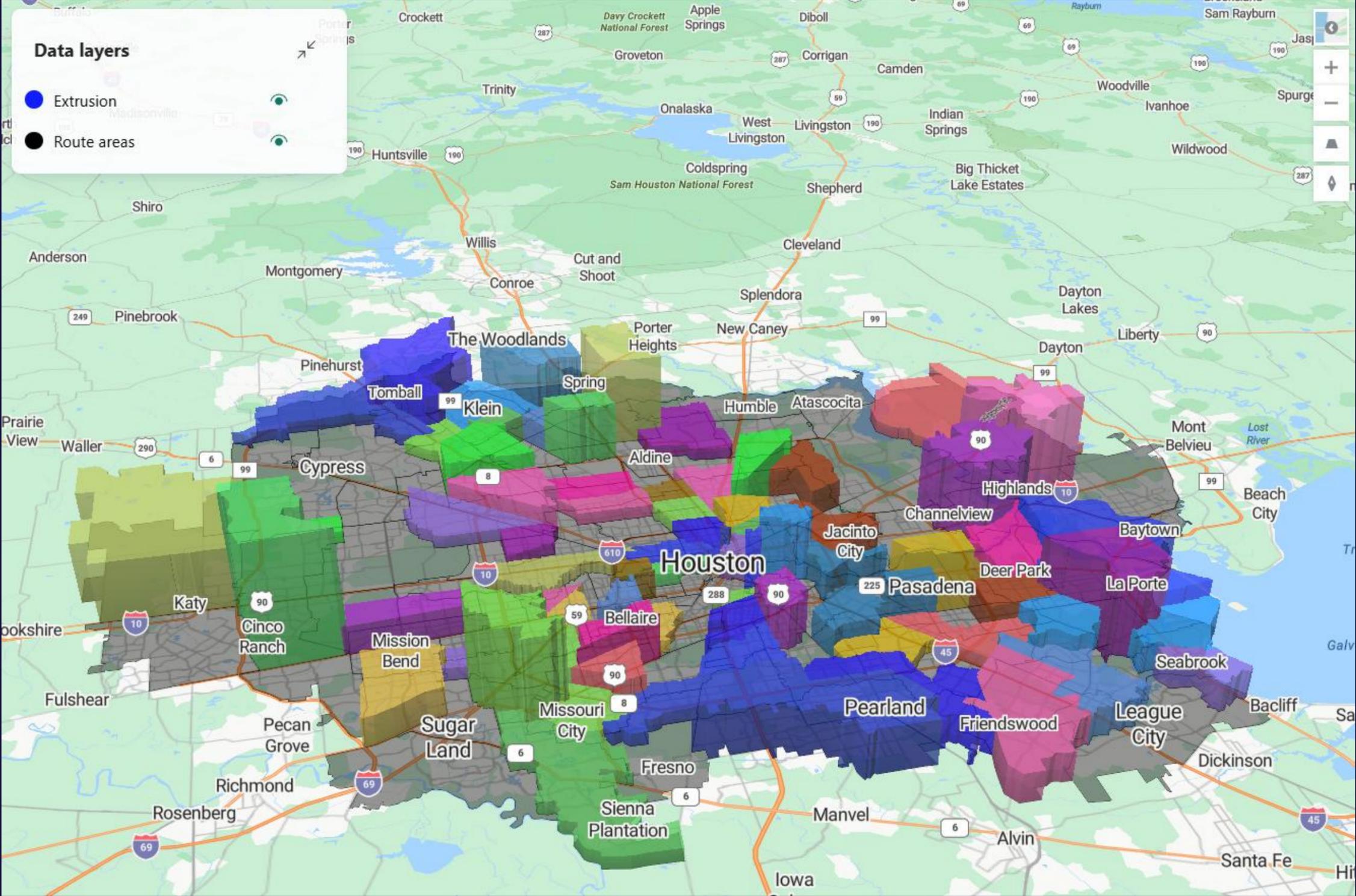
Product name	Quantity	Value (USD)	Value (Pounds)
Air Purifier	1	436.16	28.02
Antifreeze (5L)	1	120.59	26.26
Backup UPS	1	254.40	22.40
Battery Charger	1	144.32	5.42
Bolt Cutters	1	31.32	4.63
Caulking Gun	1	27.34	3.53
Cordless Drill (Bare Tool)	1	111.32	1.76

Order table

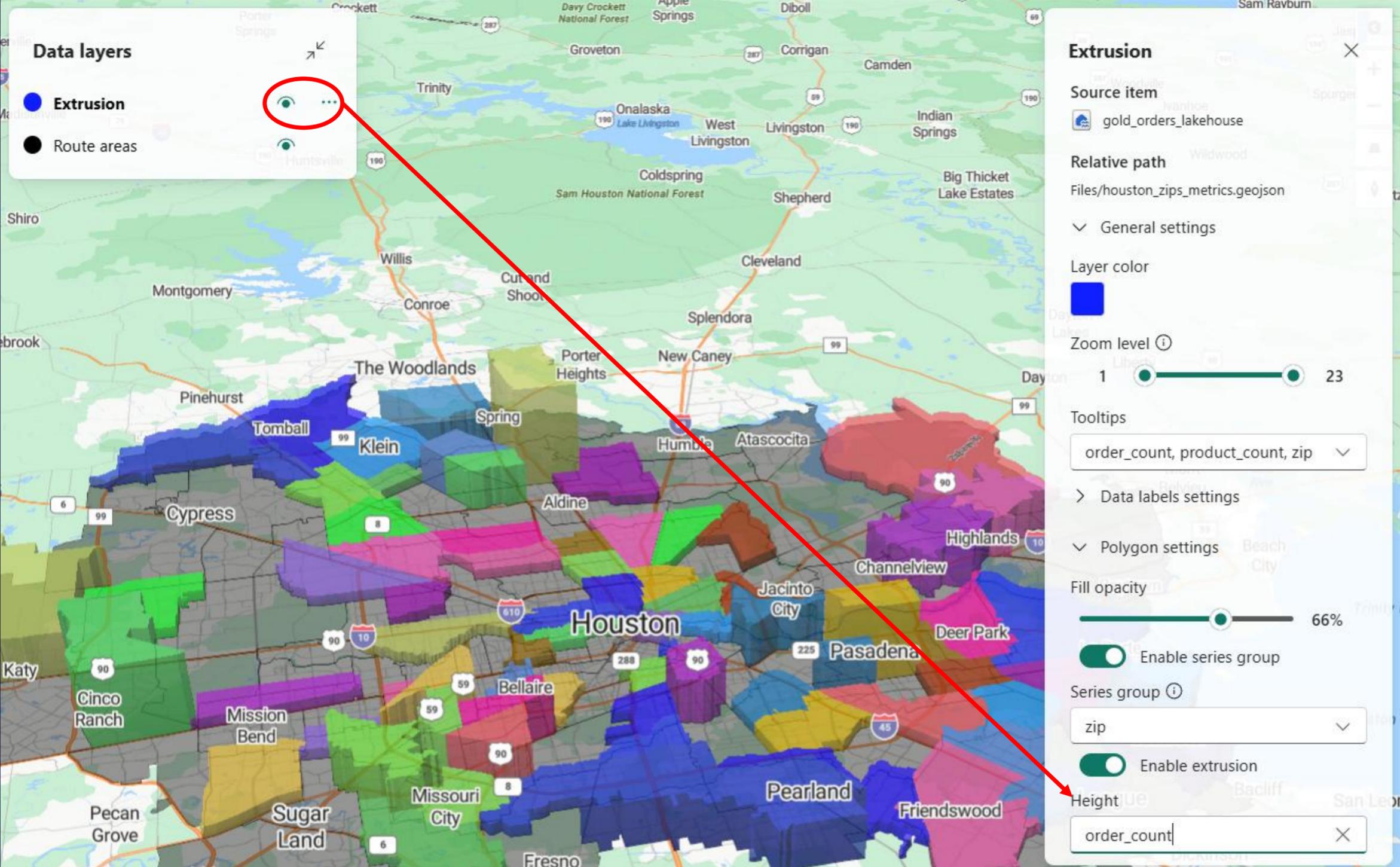
Order ID	Value (USD)	Weight (Pounds)	ZIP	Street name	House number
ORD-0001	257.20	30.14	77027	Drexel Drive	3810
ORD-0002	30.45	6.31	77587	Avenue I	701
ORD-0003	48.08	1.65	77503	Pineswept Drive	2906
ORD-0004	53.40	10.12	77033	Bellfort Street	5051
ORD-0005	14.89	4.56	77502	Shaver Street	2208
ORD-0006	254.40	22.40	77074	Rintliff Drive	8915
Total	39,263.59	6,663.98			



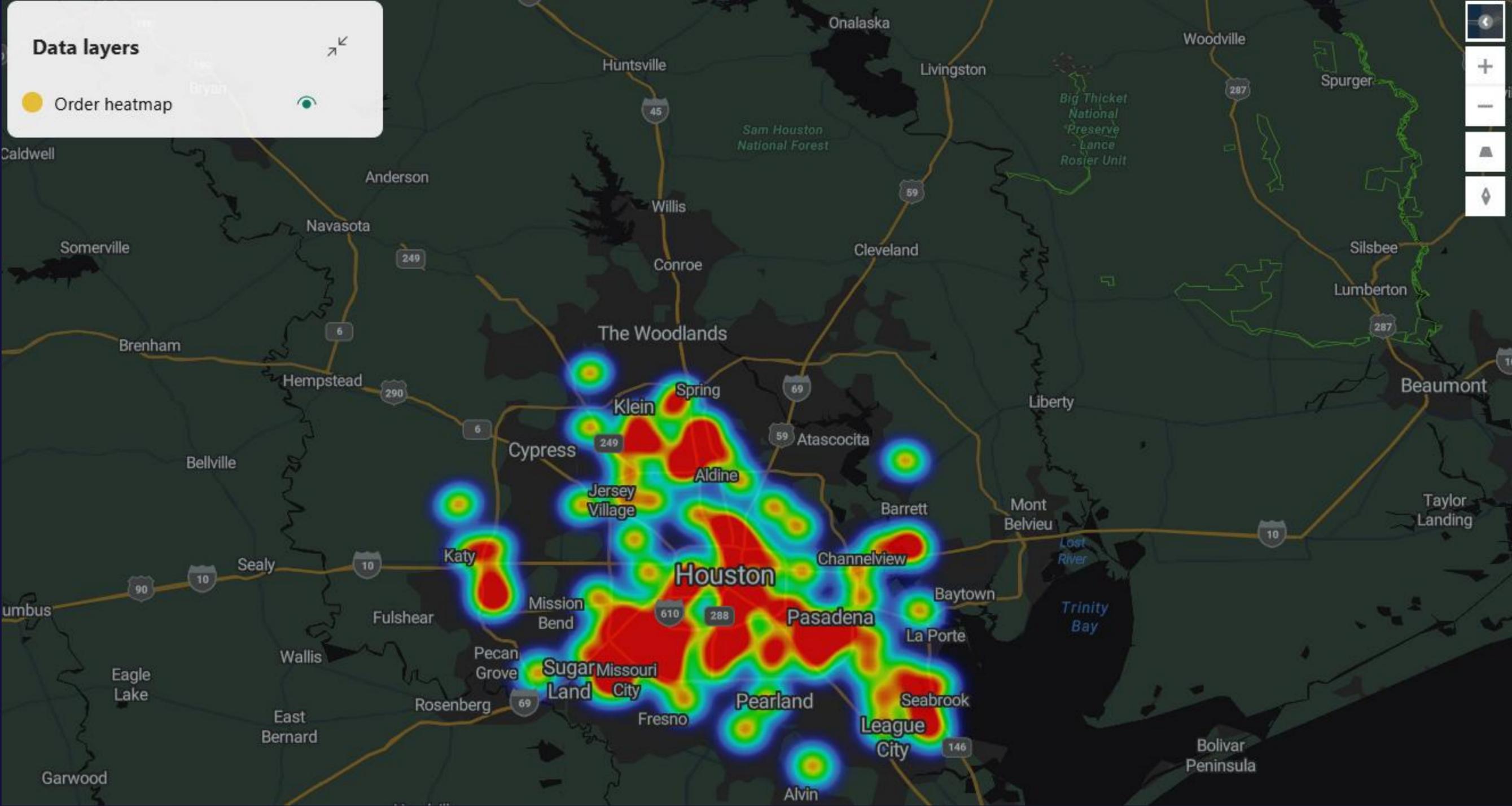
Beltway Couriers: Logistics Manager (Demo)



Beltway Couriers: Logistics Manager (Demo)



Beltway Couriers: Logistics Manager (Demo)



Beltway Couriers: Logistics Manager (Demo)

```
1 {  
2   "type": "FeatureCollection",  
3   "features": [  
4     {  
5       "type": "Feature",  
6       "properties": {  
7         "zip": 77003,  
8         "order_count": 1,  
9         "extrusion_value": 40,  
10        "centroid_lat": 29.749754,  
11        "centroid_lon": -95.345064  
12      },  
13      "geometry": {  
14        "type": "Polygon",  
15        "coordinates": [  
16          [  
17            [-95.365885,  
18             29.746091  
19          ],  
20          ],  
21          [  
22            [-95.366261,  
23             29.745624  
24          ],  
25          ]  
26        }  
27      }  
28   ]  
29 }
```

GeoJSON:

- Fabric Notebooks
- Azure Maps API
- Python libraries

API	Method	Endpoint	What It Returns / Does	Fabric & Routing Fit
Route Matrix	POST	/route/matrix	N x M grid of travel time & distance for all stop pairs in one call	Core API — distance/time input that drives the routing optimizer
Reverse Geocode	GET	/search/address/reverse	lat/lon to street address, city, ZIP, country & optional speed limit	Enrich Fabric Gold orders with readable addresses before dispatch
Fuzzy Search	GET	/search/fuzzy	Combined POI + address search; returns geometry IDs for polygons	Geocode depot names to get geometry ID for boundary fetch
POI Search	GET	/search/poi	Businesses near a point: name, coords, phone, opening hours, brand	Find fuel stops, rest areas or depots along driver routes
Get Polygon	GET	/search/polygon	GeoJSON boundary shape for ZIP codes, cities, counties or states	Render delivery zones as overlays in Fabric Maps / Power BI
Isochrone	GET	/route/range	Reachable area polygon from a depot in X minutes by car or truck	Pre-filter out-of-range orders; visualize driver coverage zones
Traffic Flow	GET	/traffic/flow/segment	Live road speed vs free-flow speed and congestion level per segment	Inject real-time traffic into Matrix API to improve route ETAs
Traffic Incidents	GET	/traffic/incident/detail	Accidents, closures, hazards filtered by severity, type, bounding box	Alert drivers; trigger re-optimization when incidents hit active routes
Elevation	GET	/elevation/point	Terrain height in meters; supports batching up to 100 points per call	Add slope cost to optimizer for accurate truck fuel and time estimates
Weather	GET	/weather/currentConditions	Current conditions, hourly/daily forecast and severe weather alert flags	Block unsafe routes pre-dispatch; shift windows in bad weather

How PyVRP Finds Solutions



Soft Constraints

Time windows and capacity treated as soft constraints with auto-adjusted penalties during local search

Target Feasibility

~43% of local search runs produce feasible solutions — balances exploration vs. exploitation

Our FleetPulse runs: 30,000+ iterations in ~30 seconds, converging on near-optimal solutions for 44 stops across 3 vehicles.

The Constraints

HARD CONSTRAINTS

Capacity

No truck exceeds 1,000 kg – ever.

Time Windows

Trucks may wait if early. Arriving late is forbidden.

Depot Return

Every truck departs from and returns to depot.

Overtime Cap

Maximum 30 minutes beyond shift end.

SOFT OBJECTIVES

Distance

Minimize total travel – direct fuel cost proxy.

Overtime Penalty

Penalized at **5× normal rate** – forces business tradeoff.

Load Balance

Duration cost of 1/sec across all vehicles prevents overloading one driver.

Morning Shift – 20 Orders Solved

PLANNING NOTEBOOK

20

Orders Routed

195.2 kg total load

72,378

VRP Cost

40,588 iterations in 30 sec

0

Overtime

All drivers back before 12:00

Driver	Stops	Load	Distance	Return	Status
AC-01 (Marcus)	5	69.0 kg	112.3 km	10:18	✓ On time
AC-02 (Sarah)	5	74.3 kg	148.9 km	10:40	✓ On time
AC-03 (David)	10	51.8 kg	151.4 km	11:39	✓ On time

Afternoon Shift – 24 Orders Solved

PLANNING NOTEBOOK

24

Orders Routed

1 unroutable dropped · 407.8 kg total

74,461

VRP Cost

27,230 iterations in 30 sec

826.5km

Fleet Total

44 stops · zero overtime

Driver	Stops	Load	Distance	Return	Status
AC-01 (Marcus)	7	129.4 kg	152.0 km	16:14	✓ On time
AC-02 (Sarah)	10	109.9 kg	87.3 km	15:45	✓ On time
AC-03 (David)	7	161.9 kg	174.6 km	16:11	✓ On time

When Reality Hits – Failure Processing

5 delivery failures detected in `fact_deliveries`. The system classifies each automatically:

Order	Failure Reason	Retry?	Outcome
ORD-0009	Customer not home	Yes	 Rescheduled → Afternoon (4.5 kg)
ORD-0047	Customer not available	No	 Next day
ORD-0005	Refused delivery	No	 Next day
ORD-0006	Afternoon refused	Yes	 Next day (no later shift)
ORD-0087	Access denied	No	 Next day

Result: 1 order rescheduled into the afternoon pool. 4 deferred to next day. Re-optimization triggered automatically.

The Solver Adapts in 30 Seconds

RE-OPTIMIZATION

Updated Stop Pool

21 original + 1 rescheduled (ORD-0009) = **22 total**

Fresh 23×23 Azure Maps distance matrix computed on-demand.

New VRP Cost: 77,189

32,773 iterations · 30 seconds

AC-02 (Sarah Chen) – Absorbed the Change

Before

9 stops · 95.0 kg

After

10 stops · 99.5 kg

AC-01 and AC-03 routes **unchanged**. All drivers back before 17:00.
Zero overtime.

What “Optimal” Actually Means

Single Cost Function

Distance + duration + overtime penalties + load balance — combined into one score. Lower = better.



Provably Better

Before/after cost comparison proves the solver beats naive approaches like "append to nearest truck."



Complete Operational Plan

Ordered sequences, arrival/departure times, overtime warnings, road-following polylines.



Dispatch Integration

routes.json → dispatch system → drivers see updated sequences instantly.



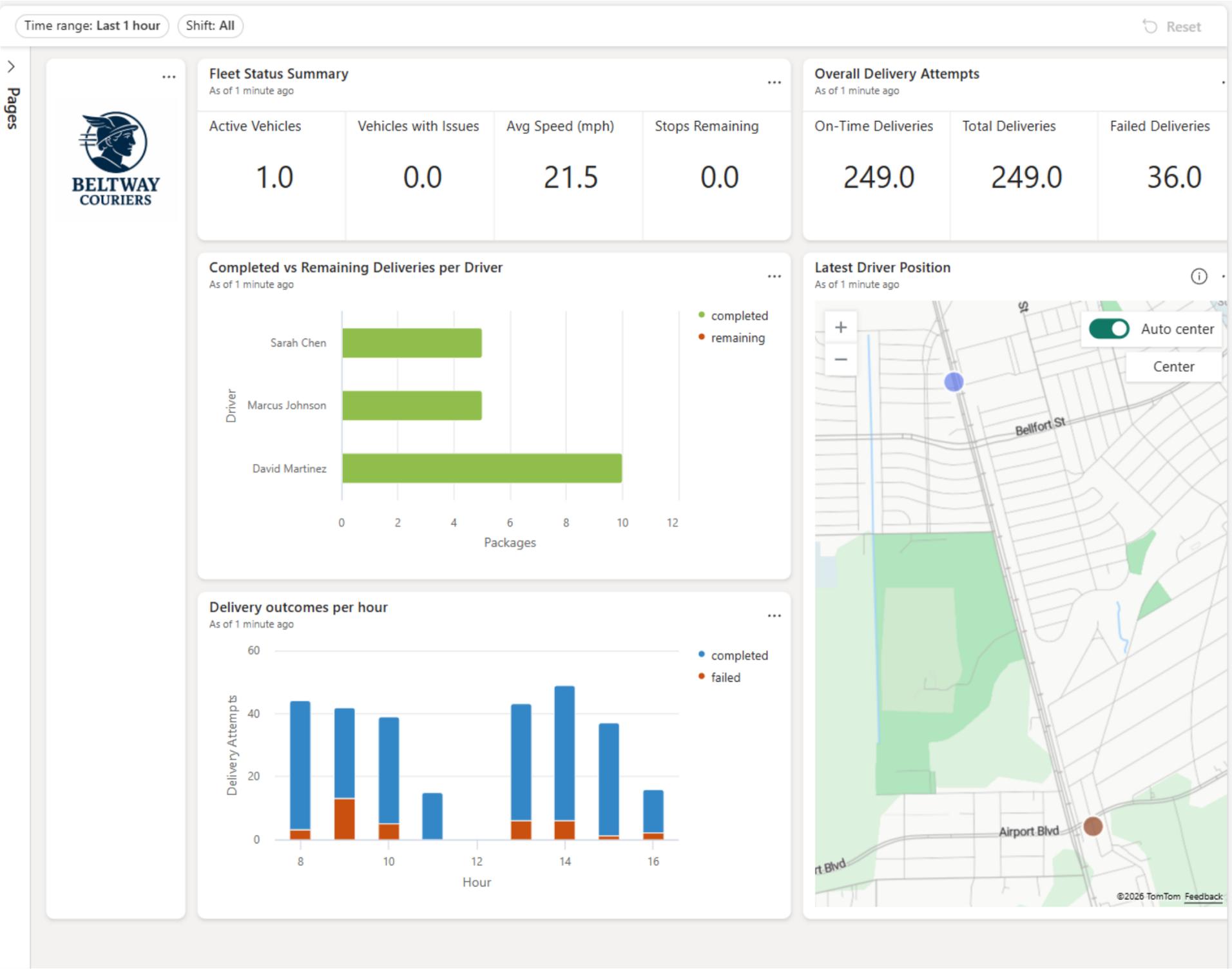
Why This Matters on Fabric

Every component of the pipeline lives in **one workspace, one security boundary, one bill**. The data never leaves the platform.



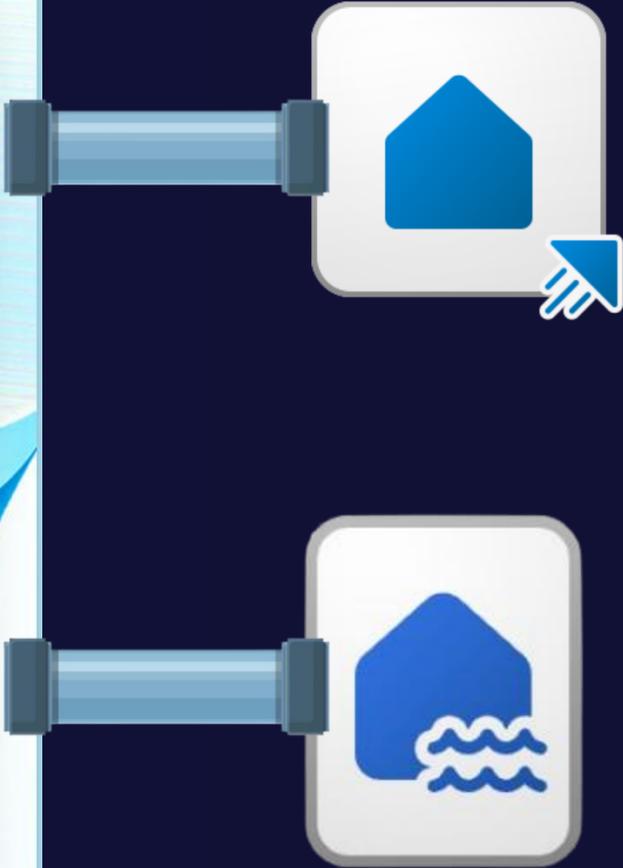
Traditional approach: 5+ separate services to provision, secure, monitor, and pay for. **Fabric:** one platform.

Beltway Couriers: Logistics Manager (Demo)



Beltway Couriers: Logistics Manager (Demo)

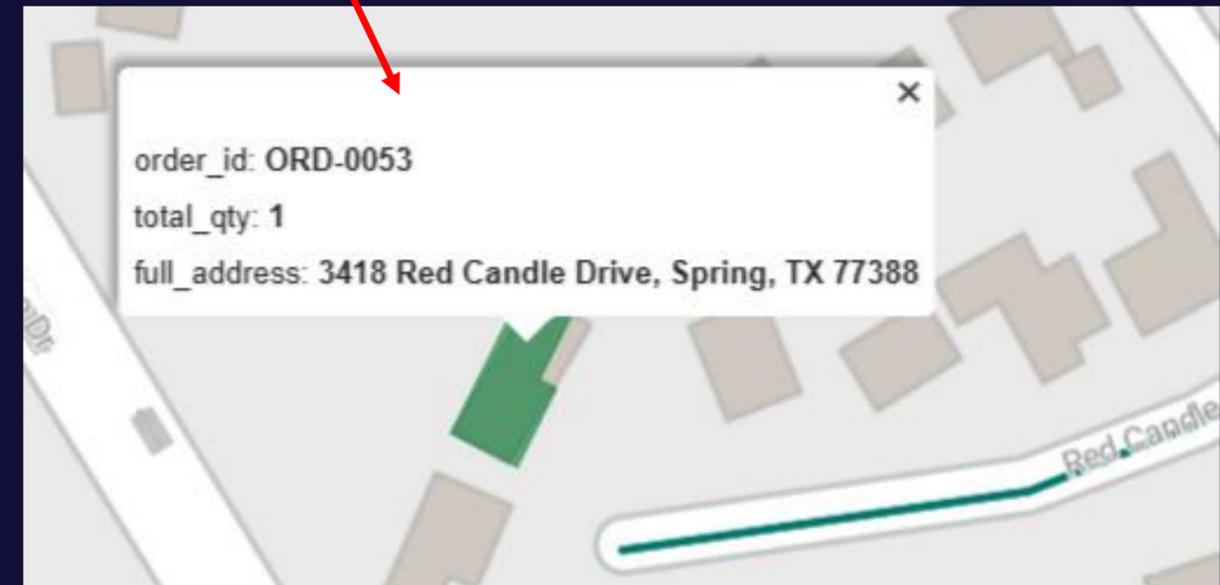
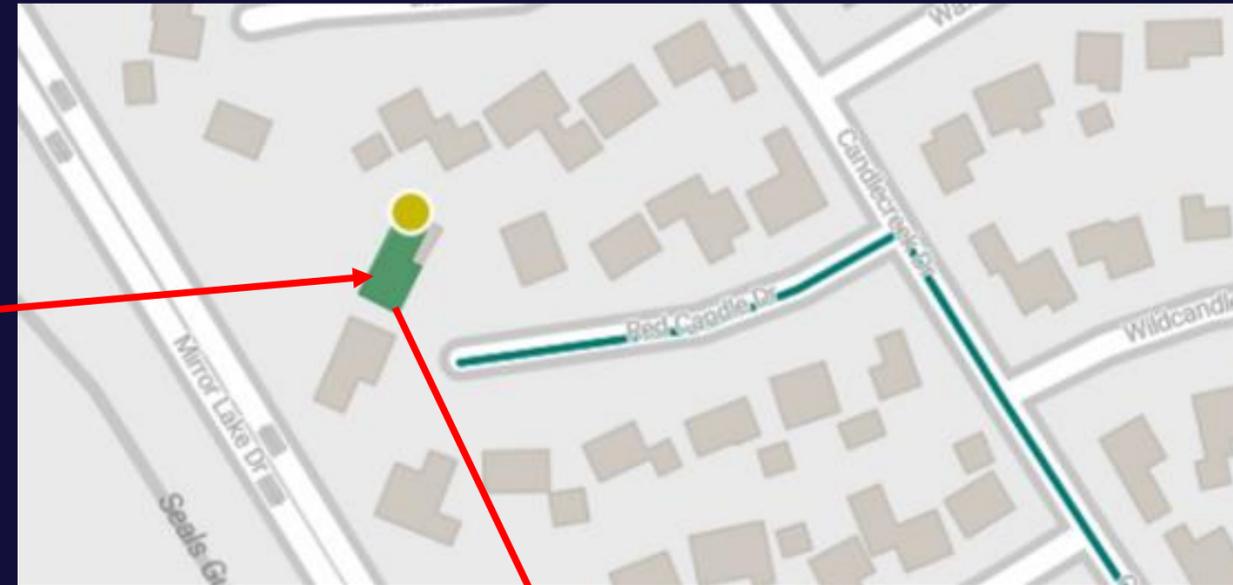
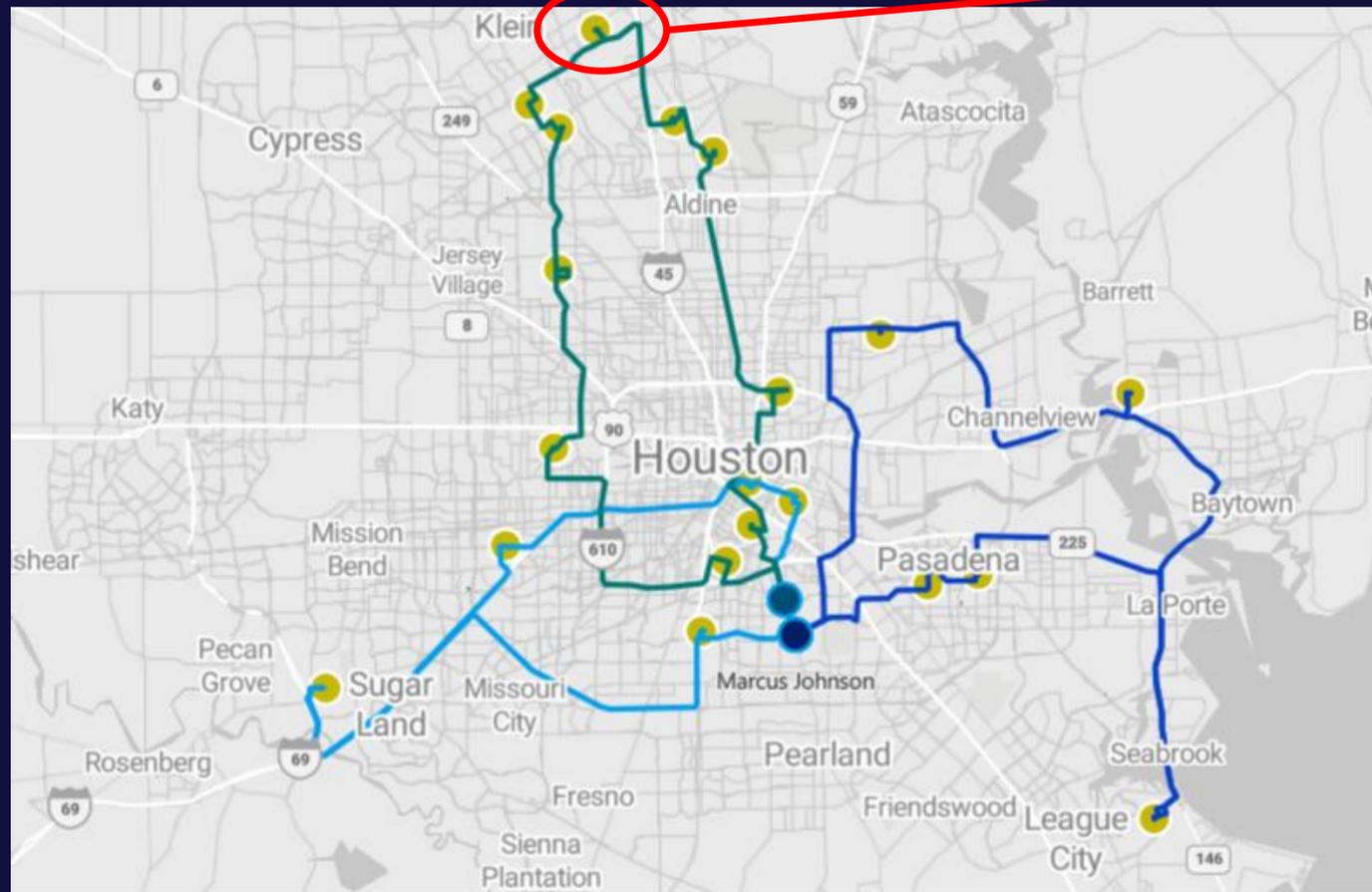
The screenshot displays the 'Beltway Couriers Analytics' dashboard. On the left, a navigation sidebar includes 'Fleet Analytics Overview', 'CourierIQ', 'Live Operations', and 'Order Analytics'. The main content area features the 'Copilot Studio' header and a large greeting: 'Hi! I'm the Beltway Couriers fleet assistant.' Below this, a text block explains the testing space and provides a 'Learn more' link. Four suggested prompts are listed in rounded boxes: 'What's the current status of our fleet?', 'How many deliveries were completed today?', 'Which vehicles are idle right now?', and 'Show me today's delivery failure reasons'. A chat window titled 'CourierIQ' is overlaid on the right, containing a welcome message: 'Hi! I'm the Beltway Couriers fleet assistant, CourierIQ. I can help you check real-time vehicle positions, delivery performance, route optimization results, and sustainability metrics — all from your Fabric data. Just ask me a question to get started.' The chat window also includes thumbs up/down icons and a 'Type your message' input field with a send button.



Microsoft Planetary Computer



Microsoft Building Footprints
Machine learning detected polygons



Microsoft Planetary Computer



Biodiversity intactness

Global terrestrial biodiversity intactness



Biomass carbon density

HGB: Harmonized Global Biomass



Digital Elevation Models (30m)

ALOS: Advanced Land Observing Satellite



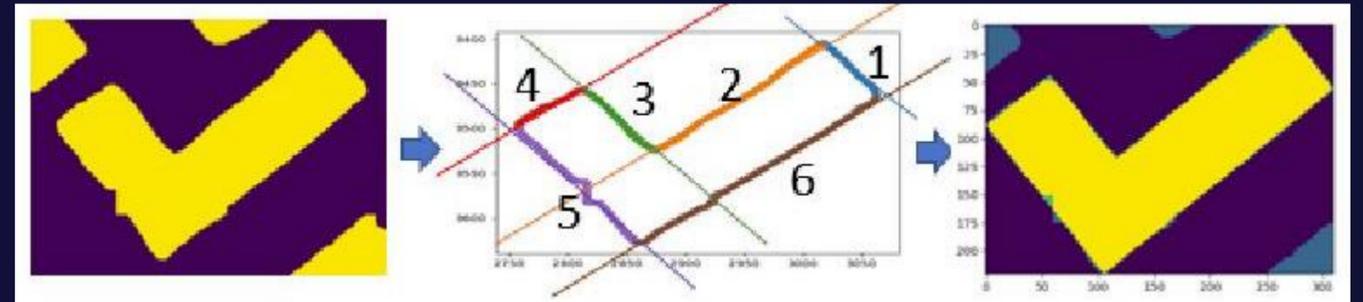
Microsoft Building Footprints

Machine learning detected polygons



Stage 1: Semantic Segmentation

*Recognizing building pixels using **deep neural networks***



Stage 2: Polygonization

Converting building pixel detections into polygons

How was the session?



Complete Session Surveys in
Whova for your chance to WIN
PRIZES!

