

Agentic AI Use Case: Driver Assignment Before Dispatch

Goal: Eliminate manual driver lookup and ensure optimal driver–load match to improve operational efficiency and compliance.

1. Problem Today

The traditional process involves significant manual intervention and risks:

- **Manual Effort:** Dispatcher manually checks extensive driver lists.
 - **Outdated Data:** Hours of Service (HOS) data is often outdated, leading to potential delays or violations.
 - **Reassignment Cost:** Reassignments are required when a driver rejects or is unavailable, causing delays.
 - **Mismatch:** Driver–load mismatch causes service delays and customer dissatisfaction.
 - **Compliance Risk:** Higher risk of Hours of Service (HOS) violations due to incorrect manual calculation.
-

2. Agentic AI Flow (LangGraph Pipeline)

This process is automated using a multi-agent system implemented via a LangGraph state machine.

Node	Agent/Function	Role
Driver Status Agent	Data Retrieval	Pulls real-time data: driver location, HOS remaining, and current job status.
Load Requirements Agent	Data Retrieval	Reads load details: pickup time, pickup city, weight,

		distance, trailer type (skill), and required skill.
Matching Engine	Scoring Logic	Calculates the Score for every eligible driver based on key criteria.
Assignment Agent	Decision Maker	Picks the top-scoring, eligible driver for the given load.
Notification Node	Action	Sends the final assignment notification to the selected Driver App.

3. Sample Dataset (for Development & Testing)

The following JSON data represents a small, valid dataset for initial development.

JSON

```
{
  "drivers": [
    {"driver_id": "D001", "name": "Arjun Kumar", "city": "Coimbatore", "hos_remaining": 6, "skill": "Reefer", "distance_km": 12, "on_time": 92, "status": "AVAILABLE"},
    {"driver_id": "D002", "name": "Ramesh P", "city": "Erode", "hos_remaining": 4, "skill": "Flatbed", "distance_km": 50, "on_time": 88, "status": "AVAILABLE"},
    {"driver_id": "D003", "name": "Manoj S", "city": "Salem", "hos_remaining": 2, "skill": "DryVan", "distance_km": 25, "on_time": 79, "status": "BREAK"},
    {"driver_id": "D004", "name": "Karthik R", "city": "Coimbatore", "hos_remaining": 8, "skill": "DryVan", "distance_km": 10, "on_time": 96, "status": "AVAILABLE"},
    {"driver_id": "D005", "name": "Siva R", "city": "Tiruppur", "hos_remaining": 7, "skill": "Reefer", "distance_km": 34, "on_time": 90, "status": "AVAILABLE"},
    {"driver_id": "D006", "name": "Prem K", "city": "Hosur", "hos_remaining": 3, "skill": "Hazmat", "distance_km": 120, "on_time": 87, "status": "AVAILABLE"}]
```

```

    {"driver_id": "D007", "name": "Vignesh", "city": "Cochin", "hos_remaining": 5, "skill": "DryVan",
    "distance_km": 160, "on_time": 95, "status": "AVAILABLE"},

    {"driver_id": "D008", "name": "Rahul", "city": "Pollachi", "hos_remaining": 6, "skill": "Flatbed",
    "distance_km": 22, "on_time": 82, "status": "AVAILABLE"}
],
"loads": [
    {"load_id": "L1001", "pickup_city": "Coimbatore", "pickup_time": "2025-11-21 09:00", "weight_kg": 1200,
    "required_skill": "DryVan", "delivery_city": "Bangalore", "distance_km": 360},
    {"load_id": "L1002", "pickup_city": "Tiruppur", "pickup_time": "2025-11-21 13:00", "weight_kg": 800,
    "required_skill": "Reefer", "delivery_city": "Chennai", "distance_km": 430}
]
}

```

A. Drivers Dataset

driver_id	name	current_city	hos_remainin g (hrs)	skill_ty pe	distanc e_to_pi ckup (km)	on_tim e_scor e	status
D001	Arjun Kumar	Coimba tote	6	Reefer	12	92	AVAILA BLE
D002	Ramesh P	Erode	4	Flatbed	50	88	AVAILA BLE
D003	Manoj S	Salem	2	DryVan	25	79	BREAK
D004	Karthik R	Coimb atore	8	DryVan	10	96	AVAILA BLE
D005	Siva R	Tiruppu r	7	Reefer	34	90	AVAILA BLE
D006	Prem K	Hosur	3	Hazmat	120	87	AVAILA BLE

D007	Vignesh	Cochin	5	DryVan	160	95	AVAILABLE
D008	Rahul	Pollachi	6	Flatbed	22	82	AVAILABLE

B. Loads Dataset

load_id	pickup_city	pickup_time	weight_kg	required_skill	delivery_city	delivery_distance_km
L1001	Coimbatore	2025-11-21 09:00	1200	DryVan	Bangalore	360
L1002	Tiruppur	2025-11-21 13:00	800	Reefer	Chennai	430

4. Sample Scoring Logic (for Matching Engine)

The *Matching Engine* calculates a composite score based on driver suitability.

Score = (Distance Bonus) + (HOS Bonus) + (Skill Match Bonus) + (Reliability Score)

Metric	Condition	Points Added
Distance Bonus	distance_to_pickup <= 20 km	+40
HOS Bonus	hos_remaining >= 6 hrs	+30
Skill Match Bonus	skill_type matches required_skill	+20

Reliability Score	$10 * (\text{on_time_score} / 10)$	+10.0 to +10.0 (e.g., \$96/10 \times 10 = 9.6\$)
--------------------------	--------------------------------------	--

I cannot directly create and download a Word document (.docx) for you. As an AI assistant, I can only provide text output.

However, I can format the entire use case, including the data and flow, using Markdown syntax, which is easily copied and pasted into a word processor (like Microsoft Word or Google Docs) for immediate conversion and download.

Here is the complete **Agentic AI Use Case for Driver Assignment Before Dispatch** ready to be copied:

Agentic AI Use Case: Driver Assignment Before Dispatch

Goal: Eliminate manual driver lookup and ensure optimal driver–load match to improve operational efficiency and compliance.

1. Problem Today

The traditional process involves significant manual intervention and risks:

- **Manual Effort:** Dispatcher manually checks extensive driver lists.
 - **Outdated Data:** Hours of Service (HOS) data is often outdated, leading to potential delays or violations.
 - **Reassignment Cost:** Reassignments are required when a driver rejects or is unavailable, causing delays.
 - **Mismatch:** Driver–load mismatch causes service delays and customer dissatisfaction.
 - **Compliance Risk:** Higher risk of Hours of Service (HOS) violations due to incorrect manual calculation.
-

2. Agentic AI Flow (LangGraph Pipeline)

This process is automated using a multi-agent system implemented via a LangGraph state machine.

Node	Agent/Function	Role
Driver Status Agent	Data Retrieval	Pulls real-time data: driver location, HOS remaining, and current job status.
Load Requirements Agent	Data Retrieval	Reads load details: pickup time, pickup city, weight, distance, trailer type (skill), and required skill.
Matching Engine	Scoring Logic	Calculates the Score for every eligible driver based on key criteria.
Assignment Agent	Decision Maker	Picks the top-scoring, eligible driver for the given load.
Notification Node	Action	Sends the final assignment notification to the selected Driver App.

3. Sample Dataset (for Development & Testing)

The following JSON data represents a small, valid dataset for initial development.

JSON

JSON

{

```
{"driver": "j
```

```
"Coimbatore", "hos_remaining": 6, "skill": "Reefer",  
"distance_km": 12, "on_time": 92, "status": "AVAILABLE"}
```

```
{"driver_id": "D002", "name": "Ramesh P", "city":
```

```

    "Erode", "hos_remaining": 4, "skill": "Flatbed",
    "distance_km": 50, "on_time": 88, "status": "AVAILABLE"},

        {"driver_id": "D003", "name": "Manoj S", "city": "Salem",
    "hos_remaining": 2, "skill": "DryVan", "distance_km": 25,
    "on_time": 79, "status": "BREAK"},

        {"driver_id": "D004", "name": "Karthik R", "city":
    "Coimbatore", "hos_remaining": 8, "skill": "DryVan",
    "distance_km": 10, "on_time": 96, "status": "AVAILABLE"},

        {"driver_id": "D005", "name": "Siva R", "city":
    "Tiruppur", "hos_remaining": 7, "skill": "Reefer",
    "distance_km": 34, "on_time": 90, "status": "AVAILABLE"},

        {"driver_id": "D006", "name": "Prem K", "city": "Hosur",
    "hos_remaining": 3, "skill": "Hazmat", "distance_km": 120,
    "on_time": 87, "status": "AVAILABLE"},

        {"driver_id": "D007", "name": "Vignesh", "city":
    "Cochin", "hos_remaining": 5, "skill": "DryVan",
    "distance_km": 160, "on_time": 95, "status": "AVAILABLE"},

        {"driver_id": "D008", "name": "Rahul", "city":
    "Pollachi", "hos_remaining": 6, "skill": "Flatbed",
    "distance_km": 22, "on_time": 82, "status": "AVAILABLE"}]

    ],
    "loads": [
        {"load_id": "L1001", "pickup_city": "Coimbatore",
    "pickup_time": "2025-11-21 09:00", "weight_kg": 1200,
    "required_skill": "DryVan", "delivery_city": "Bangalore",
    "distance_km": 360},

        {"load_id": "L1002", "pickup_city": "Tiruppur",
    "pickup_time": "2025-11-21 13:00", "weight_kg": 800,
    "required_skill": "Reefer", "delivery_city": "Chennai",
    "distance_km": 430}
    ]
}

```

A. Drivers Dataset

drive_r_id	name	current_cit_y	hos_remainin_g (hrs)	skill_type	distance_to_pickup (km)	on_time_score	status
D001	Arjun Kumar	Coimbatore	6	Reefer	12	92	AVAILABLE
D002	Ramesh P	Erode	4	Flatbed	50	88	AVAILABLE
D003	Manoj S	Salem	2	DryVan	25	79	BREAK
D004	Karthik R	Coimbatore	8	DryVan	10	96	AVAILABLE
D005	Siva R	Tiruppur	7	Reefer	34	90	AVAILABLE
D006	Prem K	Hosur	3	Hazmat	120	87	AVAILABLE
D007	Vignesh	Cochin	5	DryVan	160	95	AVAILABLE

D008	Rahul	Pollachi	6	Flatbed	22	82	AVAILABLE
------	-------	----------	---	---------	----	----	-----------

B. Loads Dataset

load_id	pickup_city	pickup_time	weight_kg	required_skill	delivery_city	delivery_distance_km
L1001	Coimbatore	2025-11-21 09:00	1200	DryVan	Bangalore	360
L1002	Tiruppur	2025-11-21 13:00	800	Reefer	Chennai	430

4. Sample Scoring Logic (for Matching Engine)

The *Matching Engine* calculates a composite score based on driver suitability.

Score =Distance Bonus + HOS Bonus + Skill Match Bonus + Reliability Score

Metric	Condition	Points Added
Distance Bonus	distance_to_pickup ≤ 20 km	+40
HOS Bonus	hos_remaining ≥ 6 hrs	+30
Skill Match Bonus	skill_type matches required_skill	+20
Reliability	\$10 * (on_time_score) / 100	+10.0 to +10.0 (e.g., \$96/100)

Score	10)\$	\times 10 = 9.6\$)
--------------	-------	--------------------

Example Calculation (Driver D004 for Load L1001 - DryVan):

- Distance 10 km -> +40
 - HOS 8 hrs -> +30
 - Skill match DryVan -> +20
 - On time \$(96/10) = 9.6 -> +9.6
 - **Total Score:** \$40 + 30 + 20 + 9.6 = 99.6
-

5. Expected Agentic AI Outcome

Load ID	Required Skill	Expected Best Driver	Rationale	Final Score
L1001	DryVan	D004 (Karthik R)	Closest (10 km), Highest HOS (8 hrs), Skill Match, Highest Reliability (96).	99.6
L1002	Reefer	D001 (Arjun Kumar)	Skill Match, Closest to pickup (12 km - triggers +40 bonus), High HOS (6 hrs).	99.2
