Mahindra™ University
Global Thinkers. Engaged Leaders.

Reinforcement Learning and Autonomous Systems (CS4122)

Lecture 1 (12/08/2023)

Instructor: Gourav Saha

Lecture Content

- **What is Reinforcement Learning (RL)?**
- Overview of the modules of this course (**through examples**).
- Course logistics.
- Miscellaneous Topics.

Motivation

- ➢ Reinforcement learning in motivated by how animals (humans are also animals 😂) learn various skills.

- ➢ Example 1: Learning to ride a bicycle.
  - If we fall down we get hurt and avoid doing something that lead to falling.
  - If we are about to fall we get scared because of the final outcome and immediately take countermeasures.
  - If we can ride the bike straight for a certain distance, we feel good about yourself and try to redo something similar.

Motivation

- Reinforcement learning in motivated by how animals (humans are also animals 😂) learn various skills.

- Example 2: Teaching dogs a trick/skill.
  - If a dog does the trick, it gets a treat.
  - If it does not then no treat (not even just a goooodddd boy!).

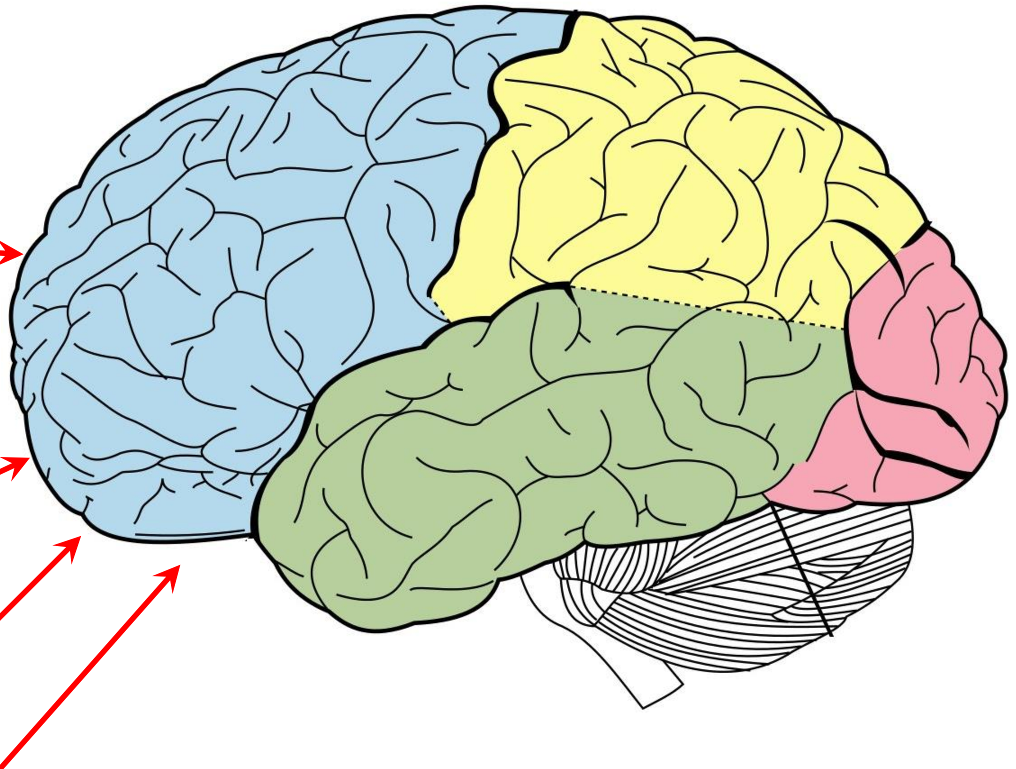
Motivation

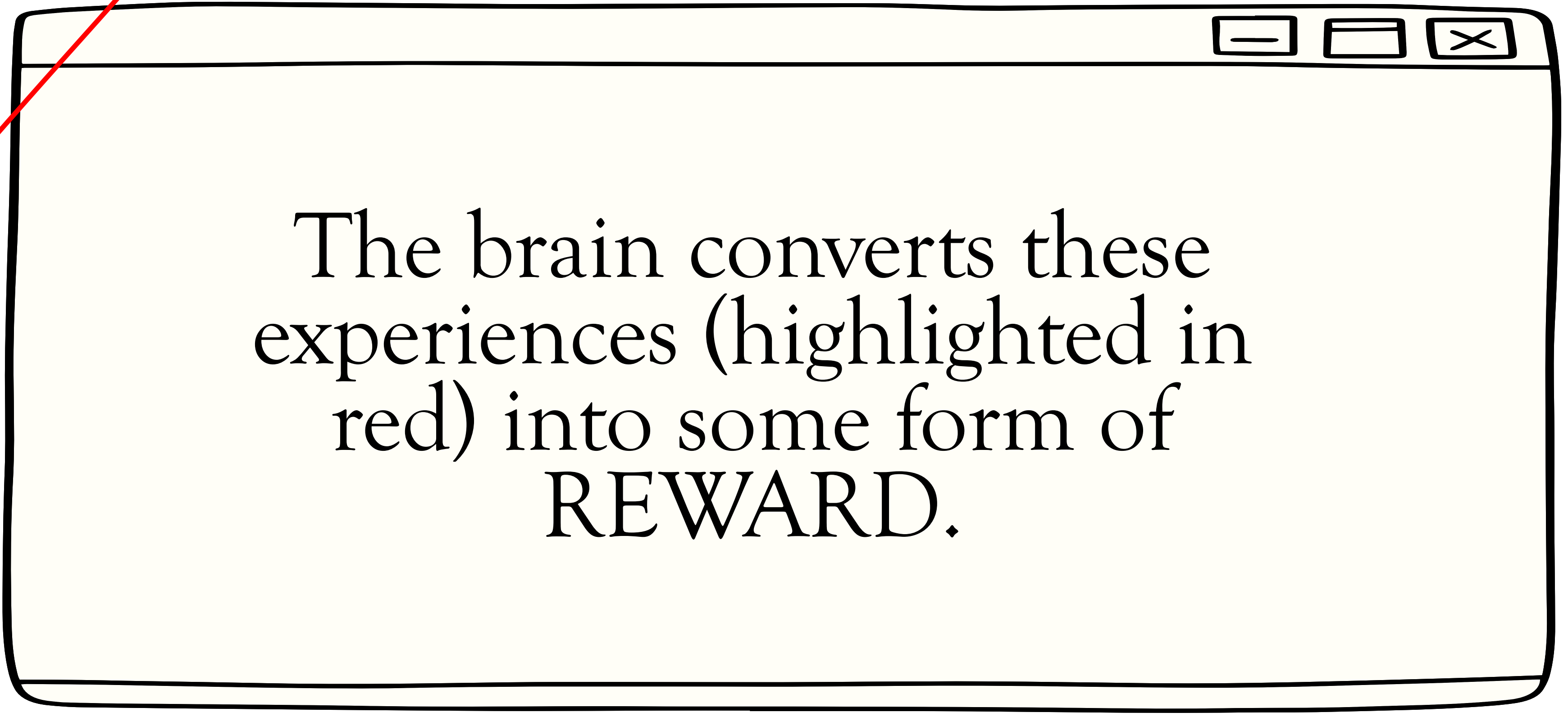
Example 1: Learning to ride a bicycle.

- If we fall down we get **hurt** and avoid doing something that lead to falling.
- If we are about to fall we get **scared** because of the final outcome and immediately take countermeasures.
- If we can ride the bike straight for a certain distance, we **feel good** about ourself and try to redo something similar.

Example 2: Teaching dogs a trick/skill.

- If a dog does the trick, it gets a **treat**.
- If it does not then **no treat** (not even just a gooooddddd boy!).



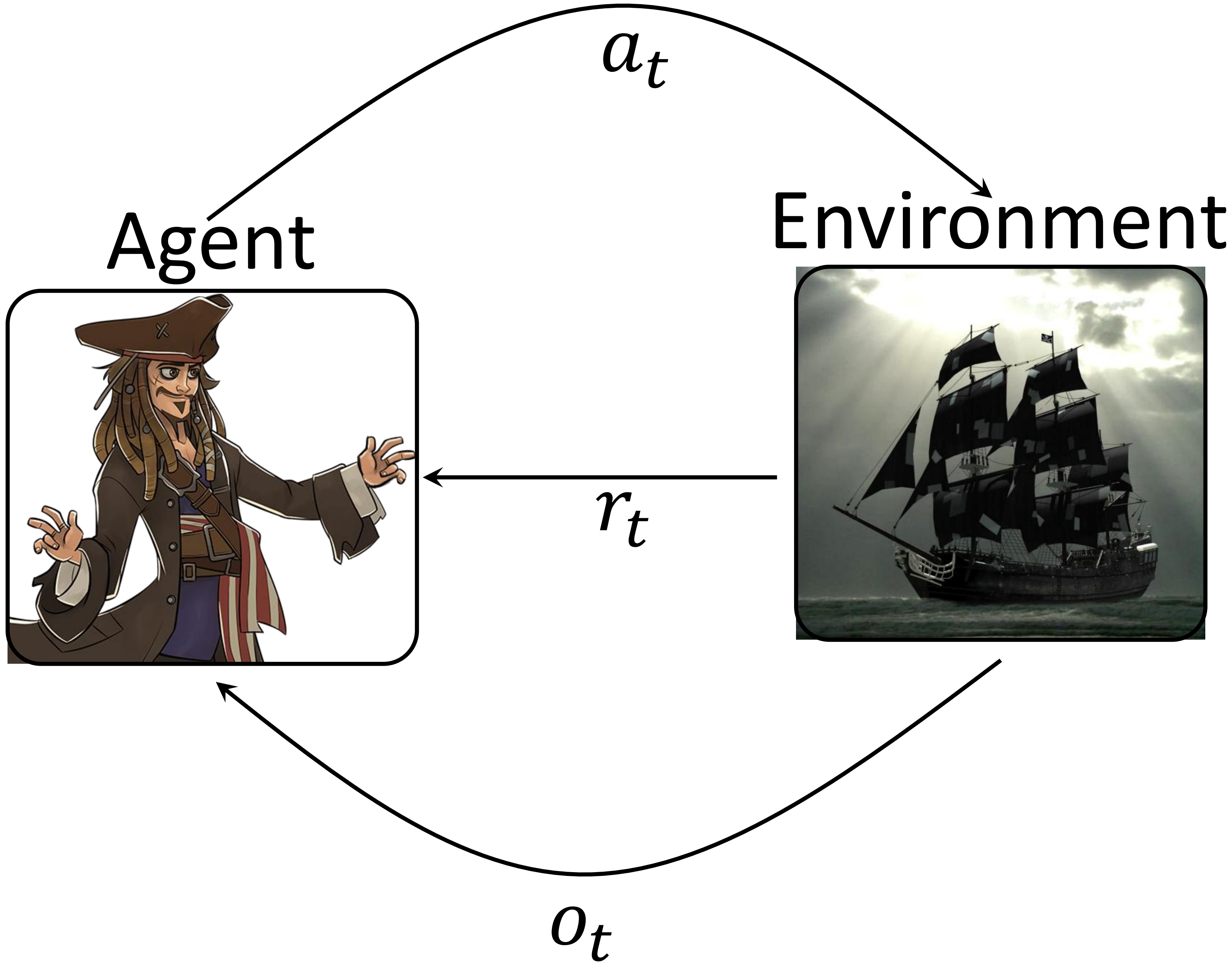


The brain converts these experiences (highlighted in red) into some form of REWARD.

What is Reinforcement Learning?

- Reinforcement learning (RL) deals with:
  - Sequential decision making. “Sequential” means over time. This is the **planning** step.
  - Learning from experience in order to maximize reward (minimize cost). This is the **learning** step.
- In RL, time is often discretized into time slots. This is a default assumption throughout the course.
  - RL can also be applied for continuous time but that is an advanced topic.

What is Reinforcement Learning?



- RL setup is often visualized as an **interaction** between an **agent** and an **environment**.

- In time slot t:
  1. The agent makes an observation o_t about the environment.
  2. Based on this observation, it takes an action a_t.
  3. Based on the action a_t:
     a) The agent will get a reward r_t.
     b) The **environment will change** in time slot $t + 1$. Hence, the requirement of the **planning step**.
  4. Based on the reward, the agent will update it's strategy to take actions. **The learning step**.

What is Reinforcement Learning?

VERY IMPORTANT

This leads to one of the two fundamental aspects of RL:

If we take an action to maximize the current reward, then the environment may change such that the future rewards are low.

Example: In chess, if we are greedy to take out opponents pieces, we can put yourself in a bad situation.

- RL setup is often visualized as an **interaction** between an **agent** and an **environment**.
- In time slot t:
  1. The agent makes an observation o_t about the environment.
  2. Based on this observation, it takes an action a_t.
  3. Based on the action a_t:
     a) The agent will get a reward r_t.
     b) The **environment will change** in time slot $t + 1$. Hence, the requirement of the **planning step**.
  4. Based on the reward, the agent will update it's strategy to take actions. **The learning step**.

Examples of Reinforcement Learning

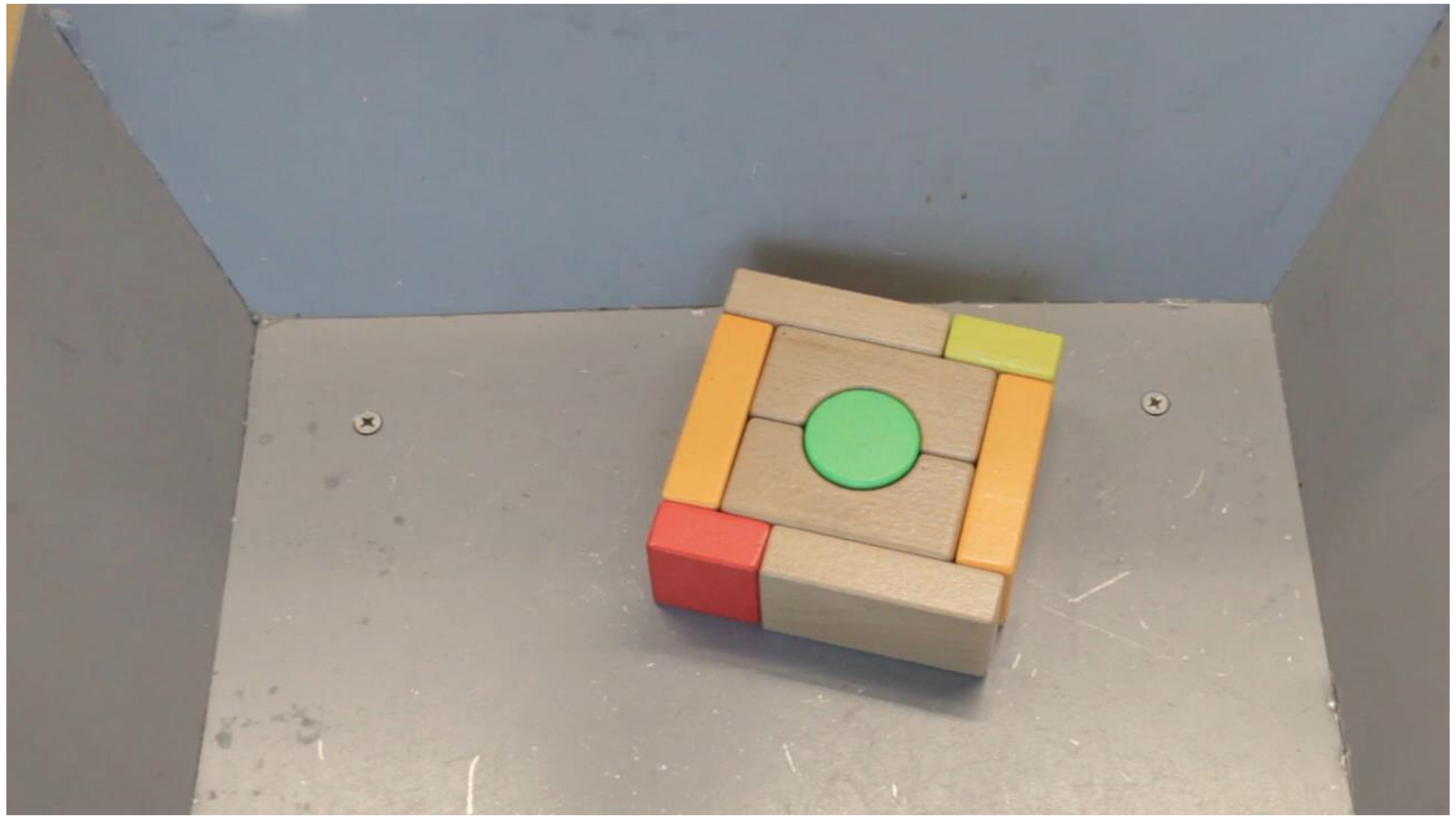
Example 1: DeepMind Atari

- *Observations*: The RGB image.
- *Actions*: To move the tray left or right.
- *Rewards*: The net score.

0:50
1:11

[1] Youtube channel: “Two Minute Papers” [Link]

Examples of Reinforcement Learning



Example 2: Robotic Grippers

- *Observations:*
  - The RGB image of the objects to grip.
  - The configuration of the robotic arm plus gripper arrangement.
- *Actions:* The servo motor speed.
- *Rewards:* Whether it could grip an object or not.

[2] Dmitry Kalashnikov et al, "Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation", PMLR. [Video link]

Examples of Reinforcement Learning

Example 3: Millimeter Wave Communication

- ➢ The millimeter wave antenna is transmitting to the women in picture.
- ➢ Millimeter waves are highly susceptible to blockages by humans, trees, buildings. The millimeter wave antenna doesn’t know about the surrounding of these women that may or may not have blockages. **It has to learn it**.
- ➢ The objective is to minimize a weighted sum of transmission delay + transmission power cost.

Examples of Reinforcement Learning

Example 3: Millimeter Wave Communication

- *Observations*: The ACK/NACK (acknowledge signal in wireless communication) signals.
- *Actions*: The number of packets to transmit.
- *Cost (negative reward)*: Transmission Delay + Transmission cost.

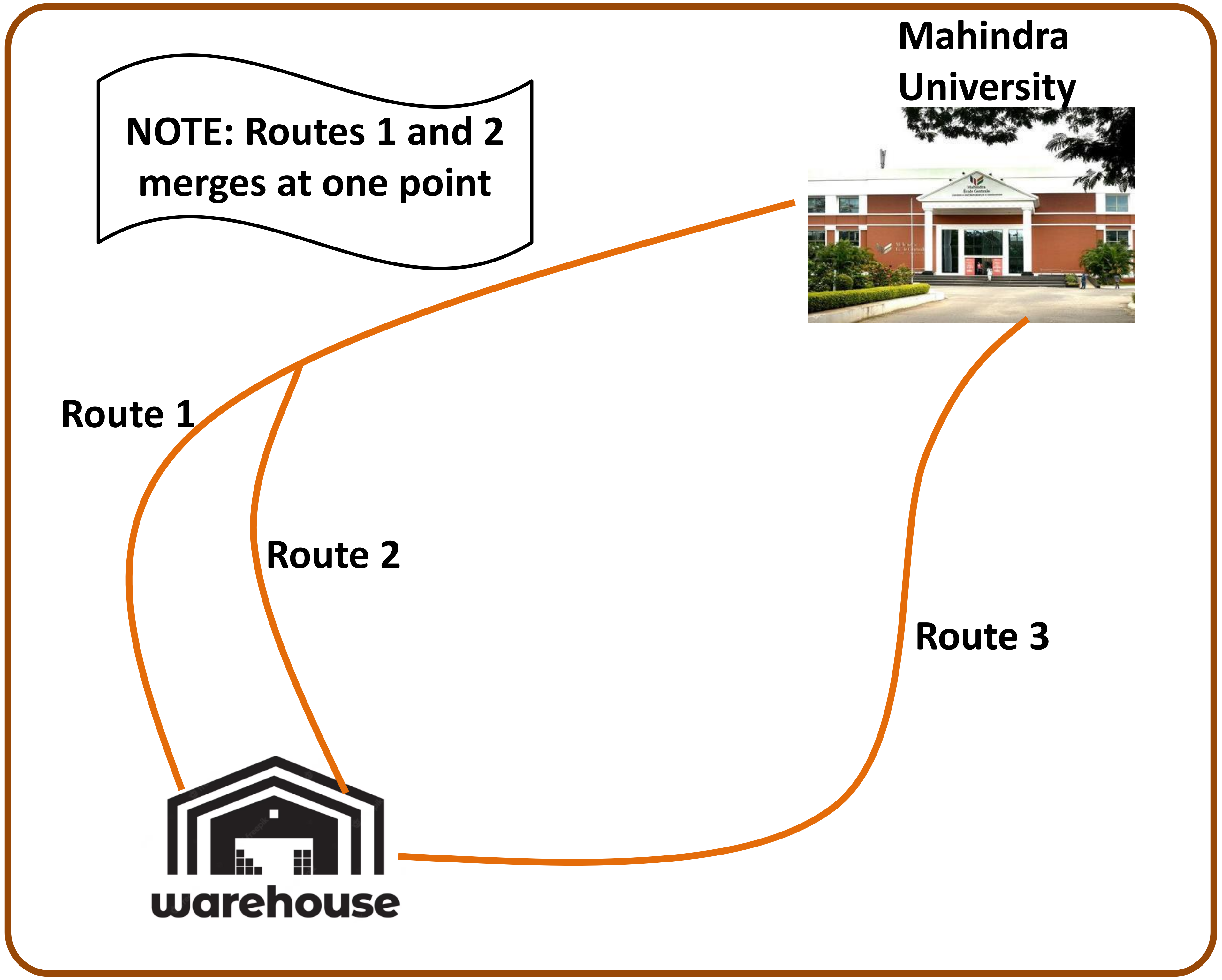
Lecture Content

- What is Reinforcement Learning (RL)?
- Overview of the modules of this course (through examples).
- Course logistics.
- Miscellaneous Topics.

Overview of the modules of this course

- Let us get an high level overview of Modules 1 – 4 of this course through a set of similar examples.
- These examples are in one way or the other related to "autonomous system".
- Module 5 does not deal with Reinforcement Learning (though some of the topics covered in Modules 1 – 4 helps in understanding Module 5) and hence we will discuss it directly towards the end of the course.

Overview of the modules of this course



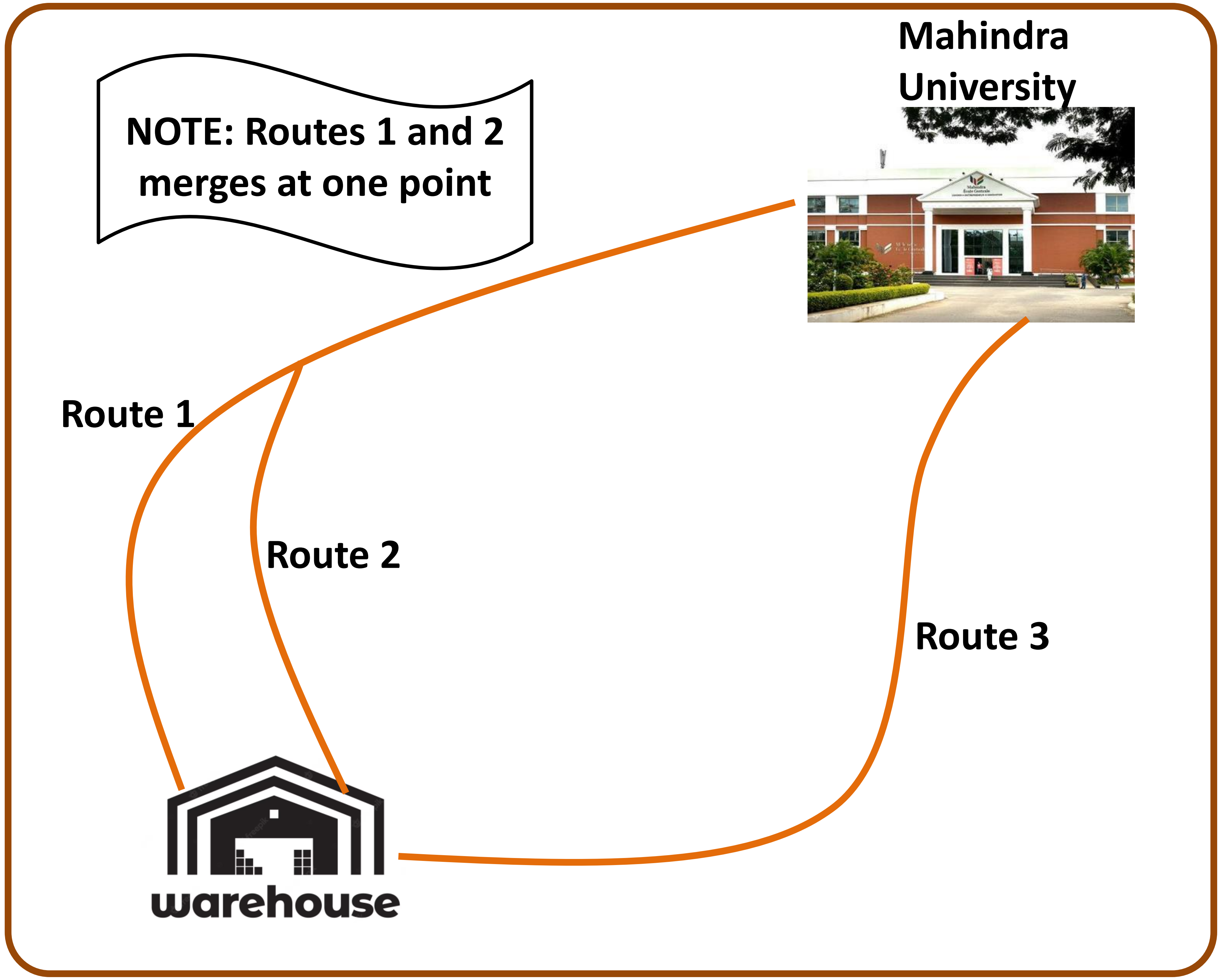
NOTE: Routes 1 and 2 merges at one point

Module 1 (Multi-Armed Bandit)

Example 1:

- ➢ The warehouse (Amazon) has to dispatch a lot of autonomous vehicles to Mahindra University.
- ➢ There are three routes that the vehicle and take. The route is decided before a vehicle leaves the warehouse.
- ➢ Off course, conditions of the routes are uncertain. How should the warehouse decide which route to take in order to:
  1. Minimize fuel cost.
  2. Minimize travel time.

Overview of the modules of this course



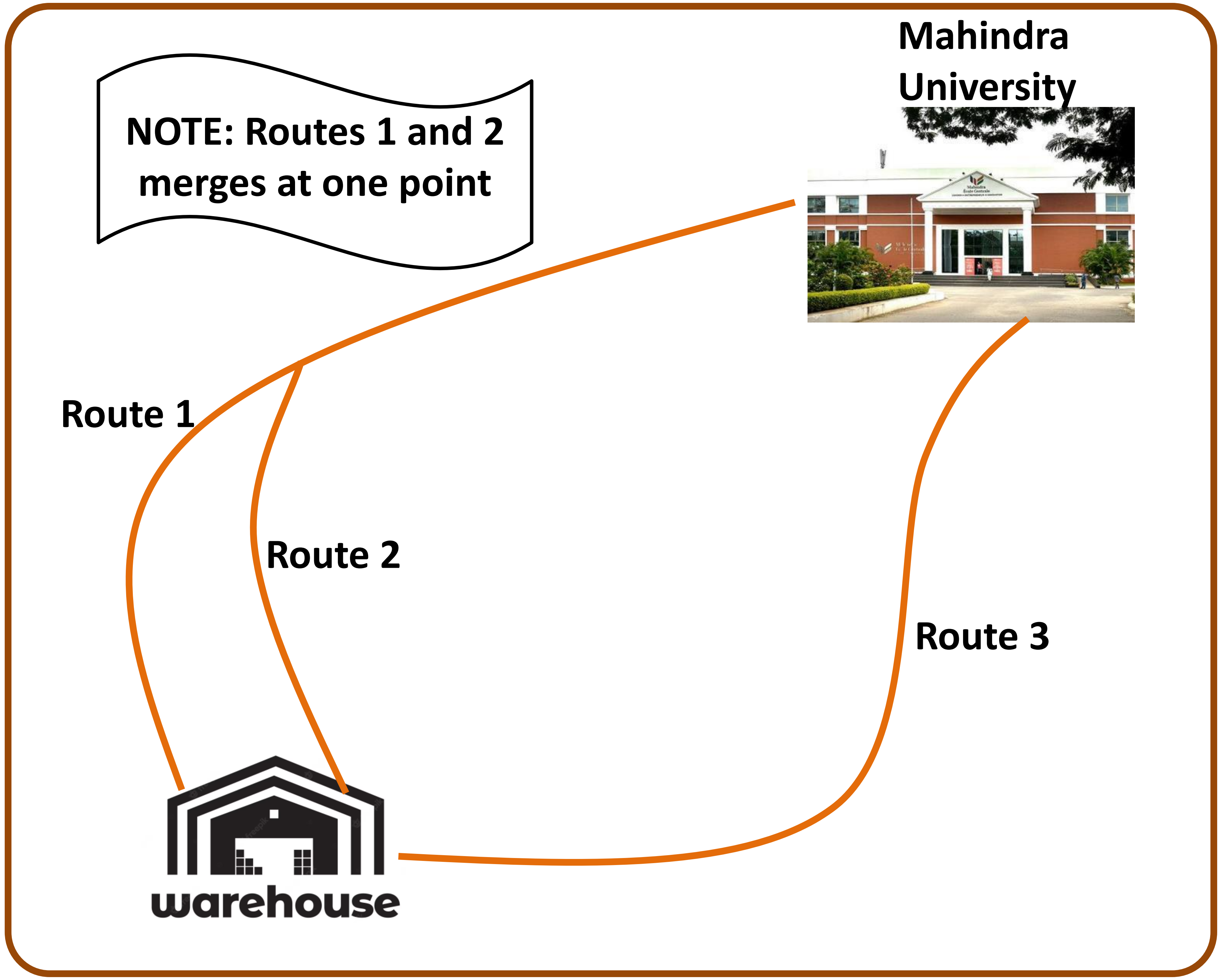
NOTE: Routes 1 and 2 merges at one point

Module 1 (Multi-Armed Bandit)

Example 1:

- The warehouse (Amazon) has to dispatch a lot of autonomous vehicles to Mahindra University.
- There are three routes that the vehicle and take. The route is decided before a vehicle leaves the warehouse.
- IMPORTANT: The warehouse **does not know the probabilistic model of the uncertainty** that decides the fuel cost/travel time.

Overview of the modules of this course

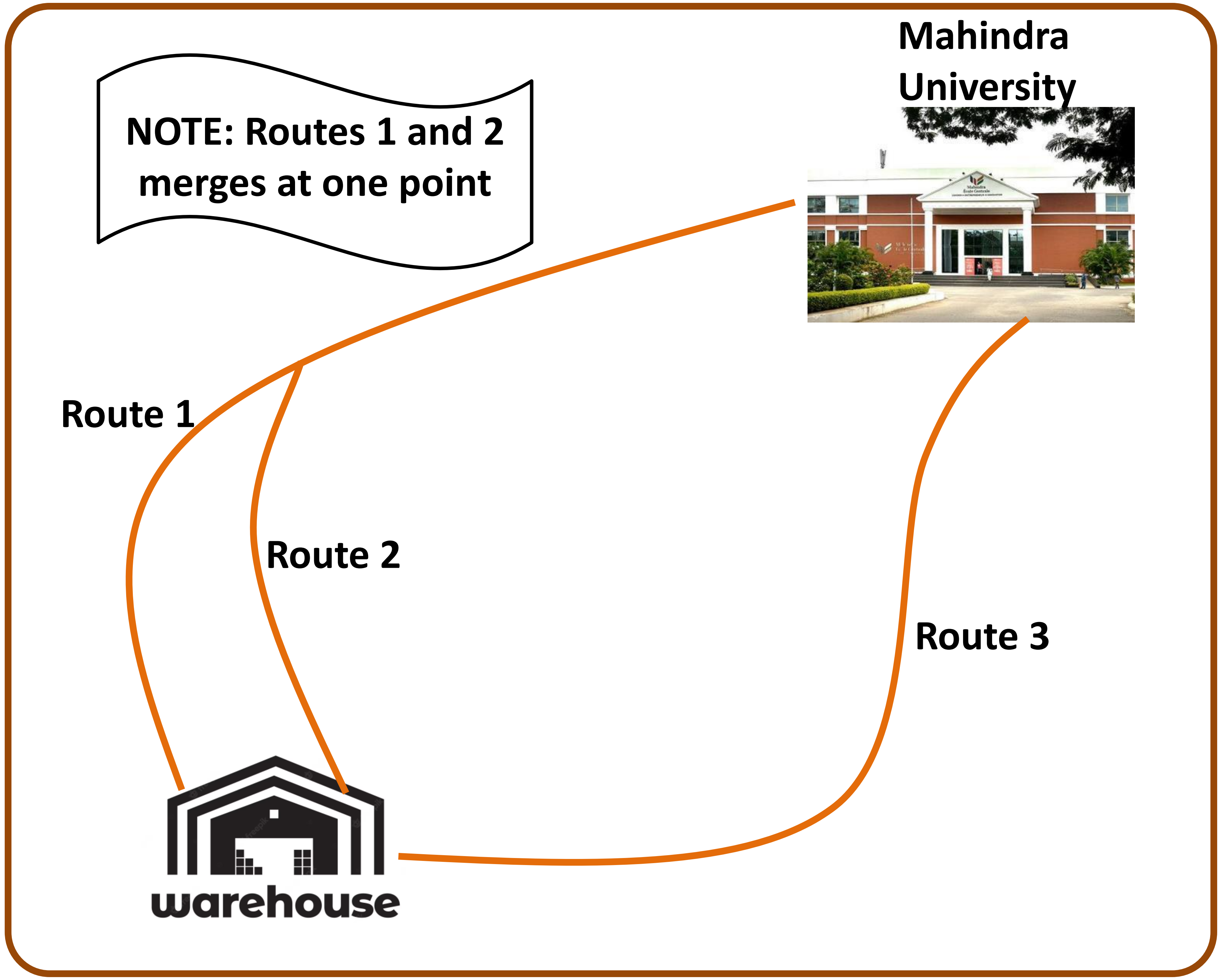


Module 1 (Multi-Armed Bandit)

Broad aspects of the solution to Example 1:

- The warehouse can try different routes and learn from experience.
- Eventually, it will get an idea of which of the three route has the best average fuel cost or travel time.
- IMPORTANT (online data collection): The warehouse can learn the travel time of a route only if it takes the route. We cannot learn about a route that we are not travelling.

Overview of the modules of this course



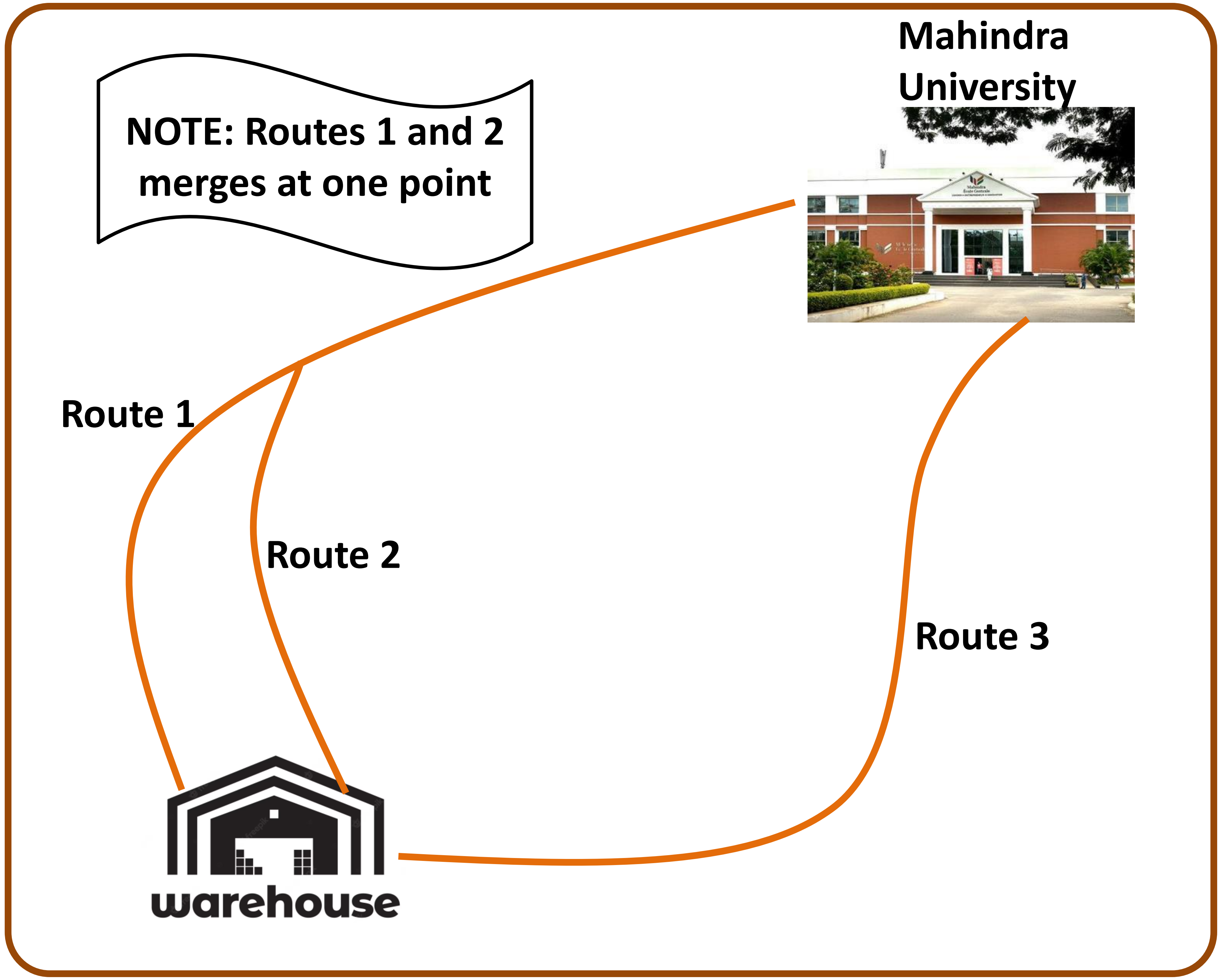
Module 1 (Contextual Bandits)

Example 2:

➢ Same as Example 1 with the following modification:

- The warehouse has two additional information to decide which route to take:
  1. Average travel time of vehicles dispatched in the previous hour.
  2. Average travel time of vehicles dispatched in the current hour of the previous day.

Overview of the modules of this course


NOTE: Routes 1 and 2 merges at one point

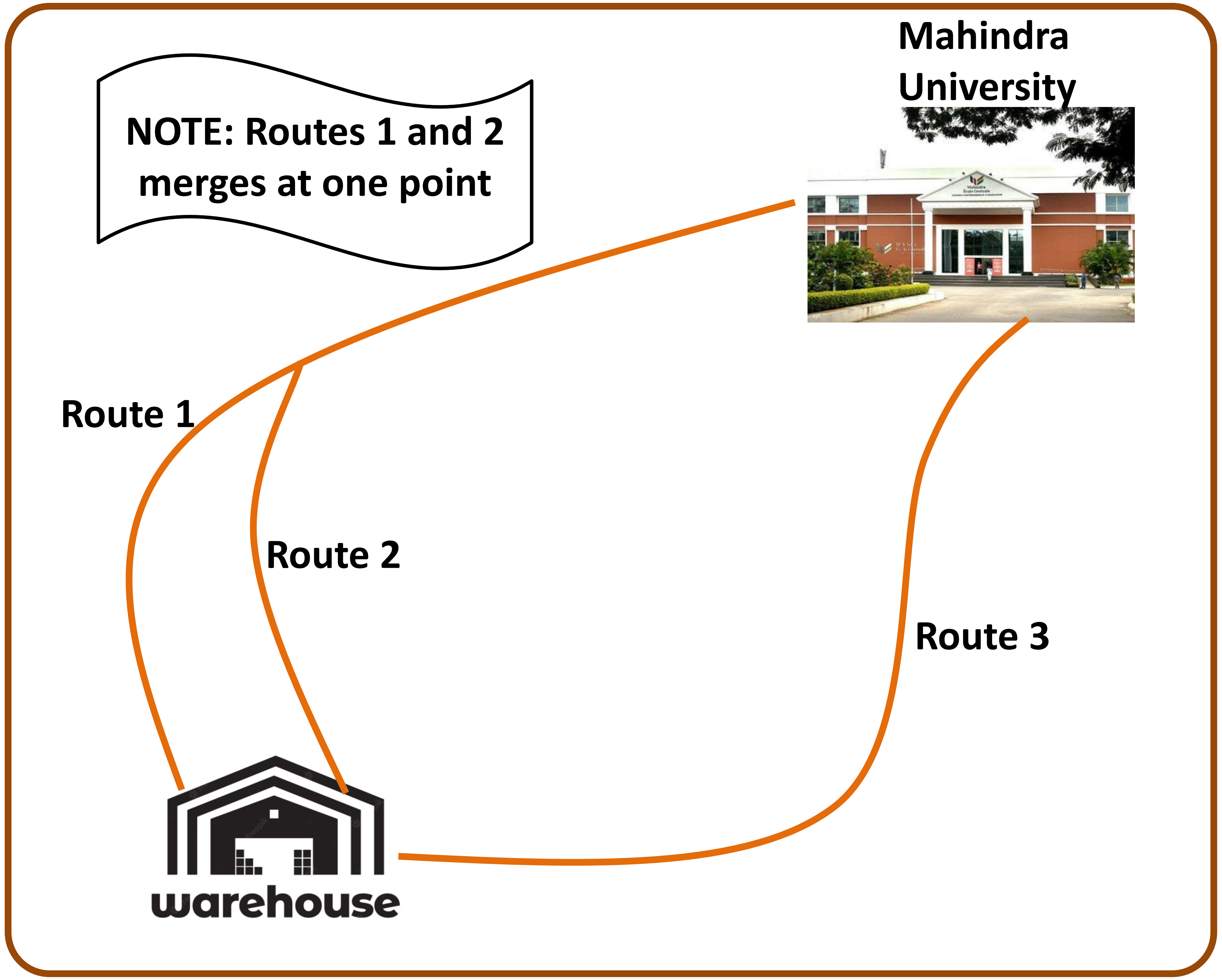
Module 1 (Contextual Bandits)

Broad aspects of the solution to Example 2:

➢ The two additional information, the context, should intuitively help the warehouse to make better routing decisions.

➢ However, uncertainty of the three routes still exists in this problem as in Example 1. So, the warehouse still has to learn from experience.

Overview of the modules of this course



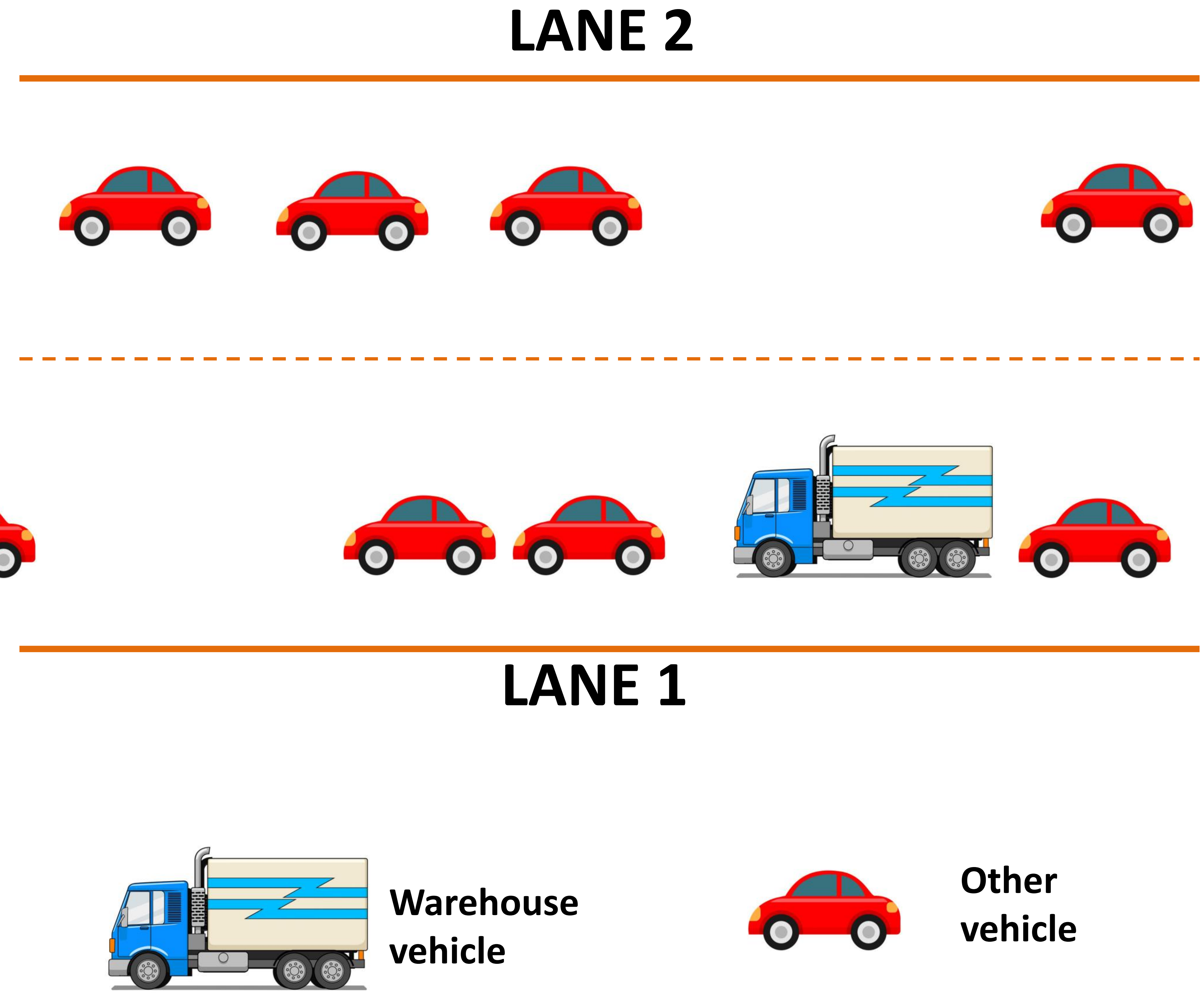
NOTE: Routes 1 and 2 merges at one point

Module 1 (Contextual Bandits)

Hey! This is just a forecasting problem. What is so new about that?

- Yes. But to train a forecasting model, we need data. Here, we have to collect data by traveling a route (hence online data collection).
- Similar to Example 1, we can collect data about a route only by travelling it.

Overview of the modules of this course

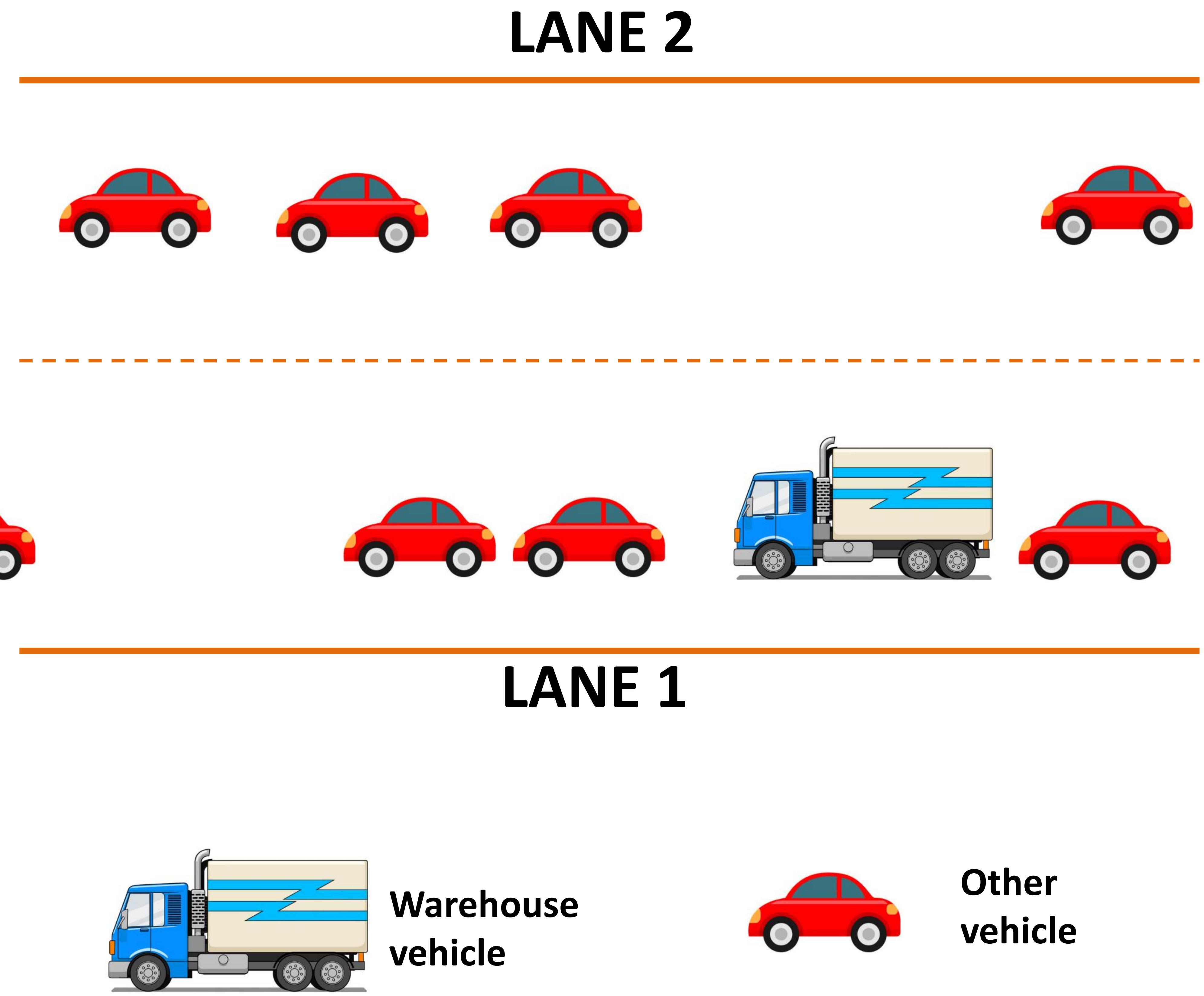


Module 2 (Markov Decision Process (MDP))

Example 3:

- Suppose the warehouse's vehicle has been dispatched in a given route.
- The job of the autonomous system of the warehouse vehicle is to decide whether to change lane or not.
  - Other aspects of driving like controlling the brakes and accelerator are done by humans.
- Objective is same as Example 1. Either minimize fuel cost or travel time.

Overview of the modules of this course



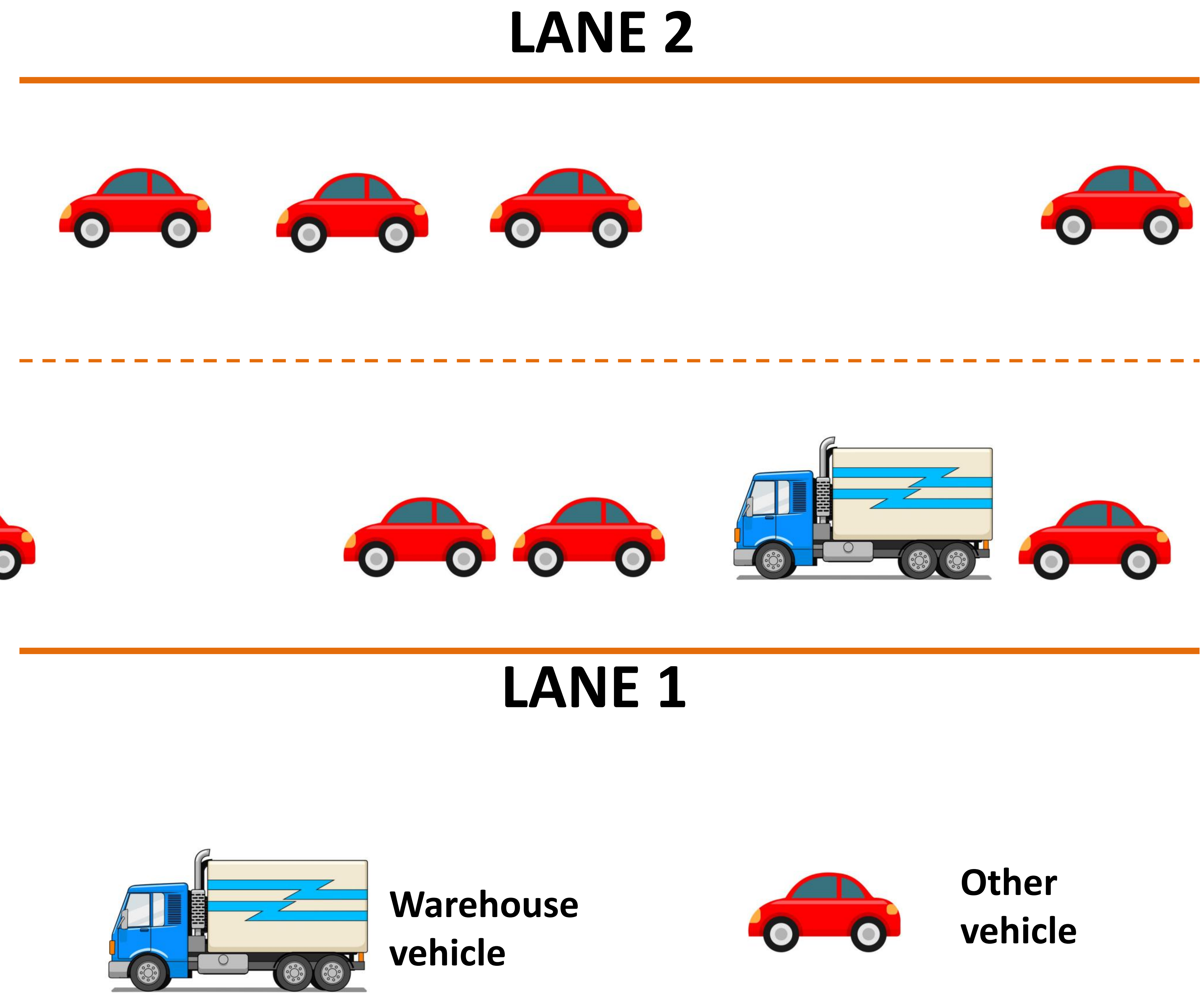
Module 2 (Markov Decision Process (MDP))

Example 3:

- Uncertainties associated with the system: How the other vehicles will react to lane change. Also, how the human driver in the warehouse vehicle will react by changing its speed.

- IMPORTANT: For MDP, the **autonomous system knows the probabilistic model of the uncertainty.** Hence, a **planning problem** NOT a learning problem.

Overview of the modules of this course



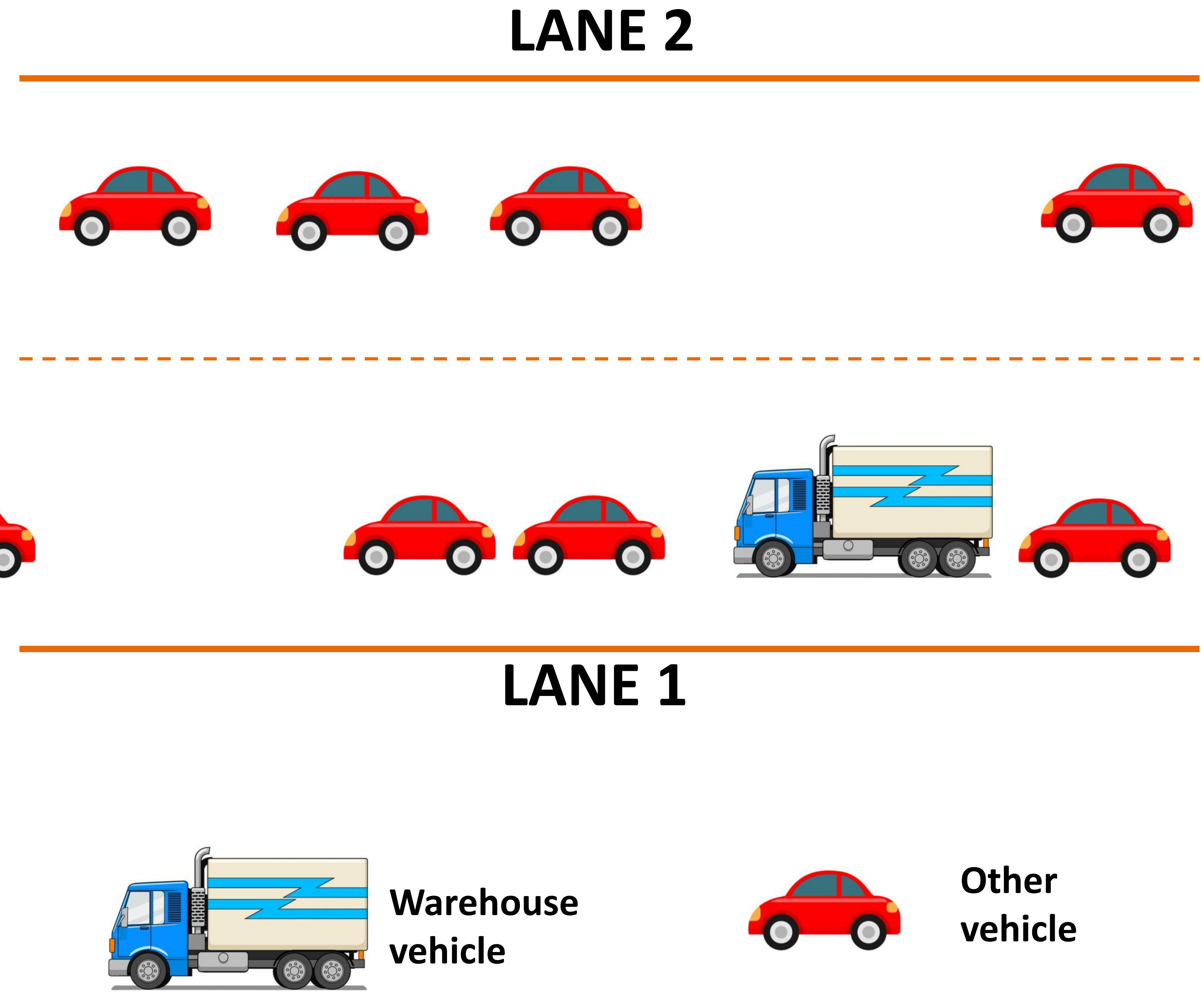
Module 2 (Markov Decision Process (MDP))

Broad aspects of the solution to Example 3:

- The state* of the “environment” are:
  - x - coordinate of all the vehicles.
  - The lane of all the vehicles.
  - The velocity of all the vehicles.
- We assume that these states are available to the warehouse vehicle through GPS and VANET (vehicular adhoc network).

* We will learn more about **states** in the next lecture but for now we can think of states as a set of parameters that sufficiently captures how the environment evolves over time.

Overview of the modules of this course



Module 2 (Markov Decision Process (MDP))

Broad aspects of the solution to Example 3:

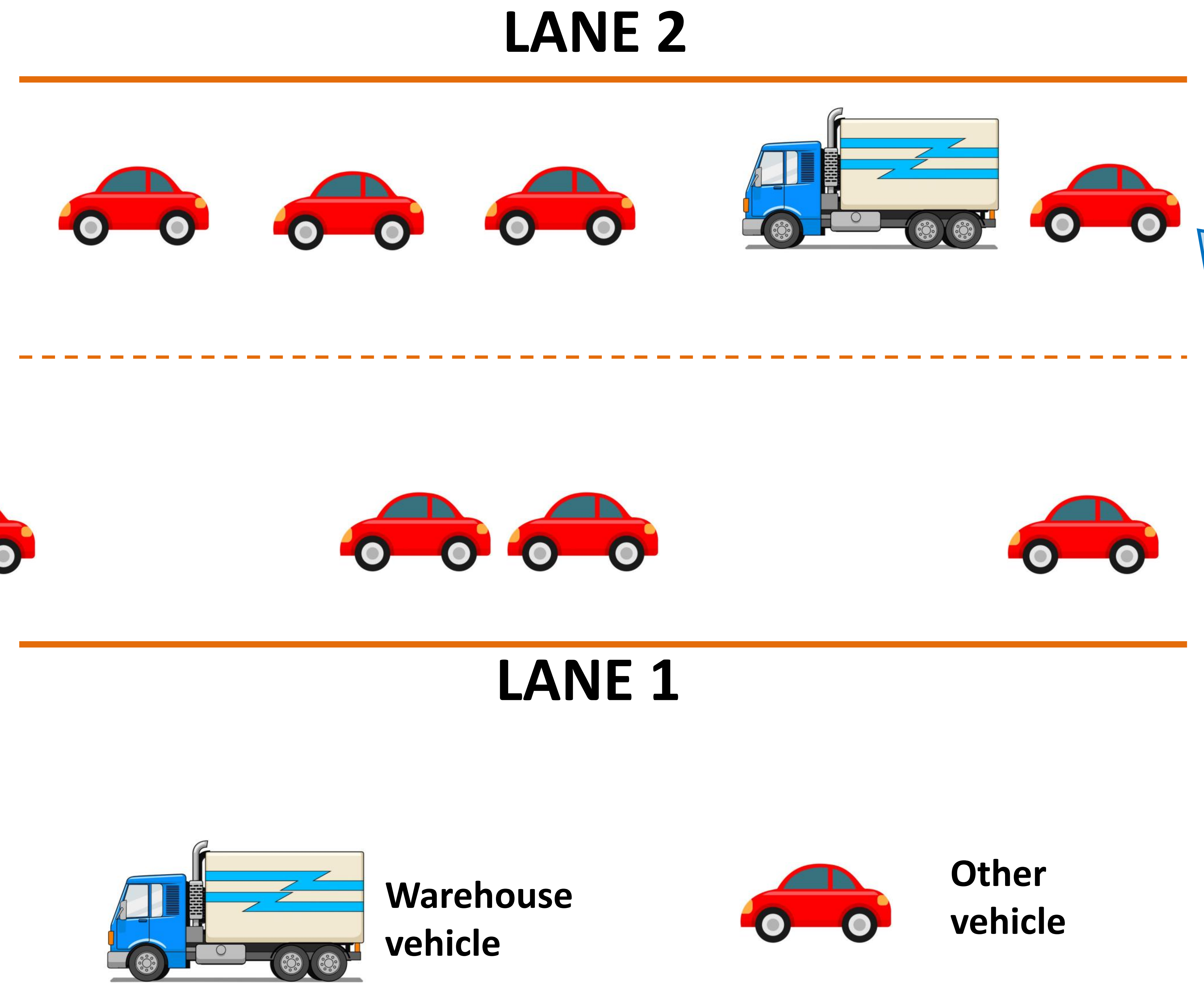
- ➢ Just because the:

  1. probabilistic model of uncertainty, and
  2. all the states are available

  it does not mean the deciding the optimal action is an easy task. This is because.....

(PTO)

Overview of the modules of this course

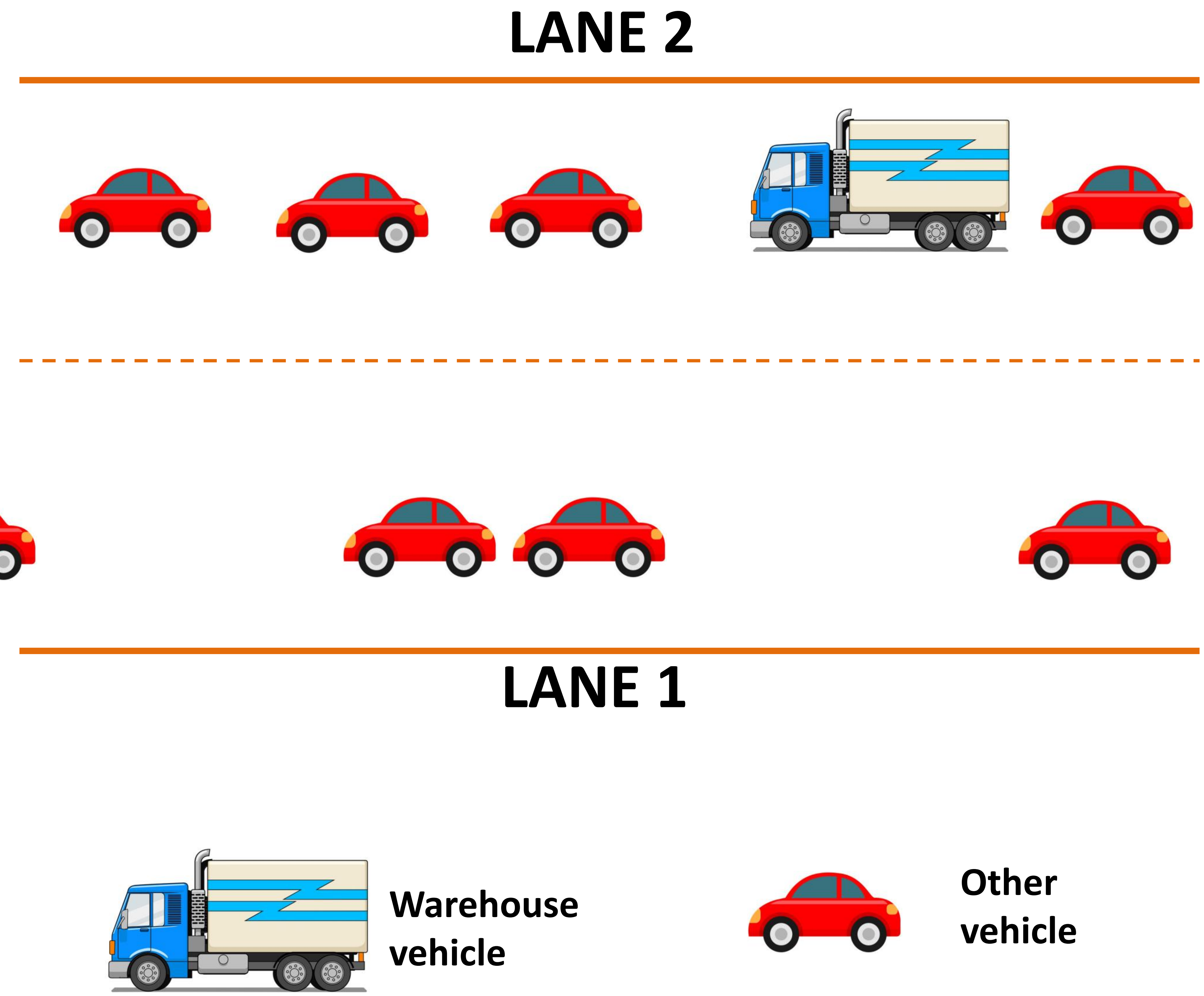


Module 2 (Markov Decision Process (MDP))

Broad aspects of the solution to Example 3:

- IMPORTANT: Depending on the action of the warehouse vehicle (change lane or not), the states will change. E.x. If the vehicle changes to lane 2:
  - The velocity of the warehouse vehicle may decrease (because a vehicle has to reduce its speed a little during lane change).
  - The speed of this vehicle may also reduce.

Overview of the modules of this course



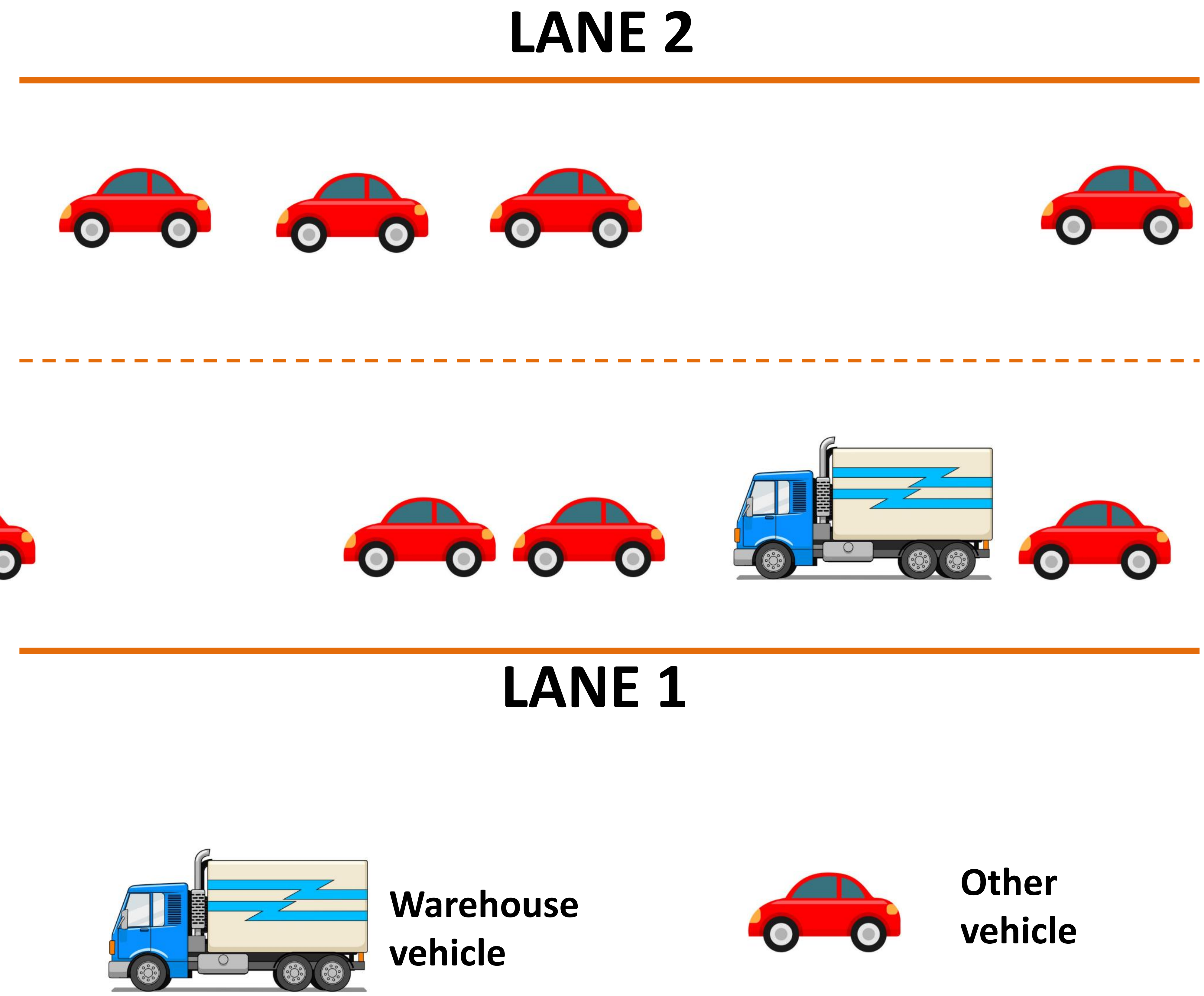
Module 2 (Markov Decision Process (MDP))

Broad aspects of the solution to Example 3:

- IMPORTANT: Depending on the action of the warehouse vehicle (change lane or not), the states will change. Ex. if the vehicle changes to lane 2:
  - The velocity of the warehouse vehicle may decrease (because a vehicle has to reduce its speed a little during lane change).
  - The speed of this vehicle may also reduce.

This is unique to Example 3 that we didn't see in Examples 1 and 2, i.e. in Examples 1 and 2, the action of the agent didn't change the environment.

Overview of the modules of this course

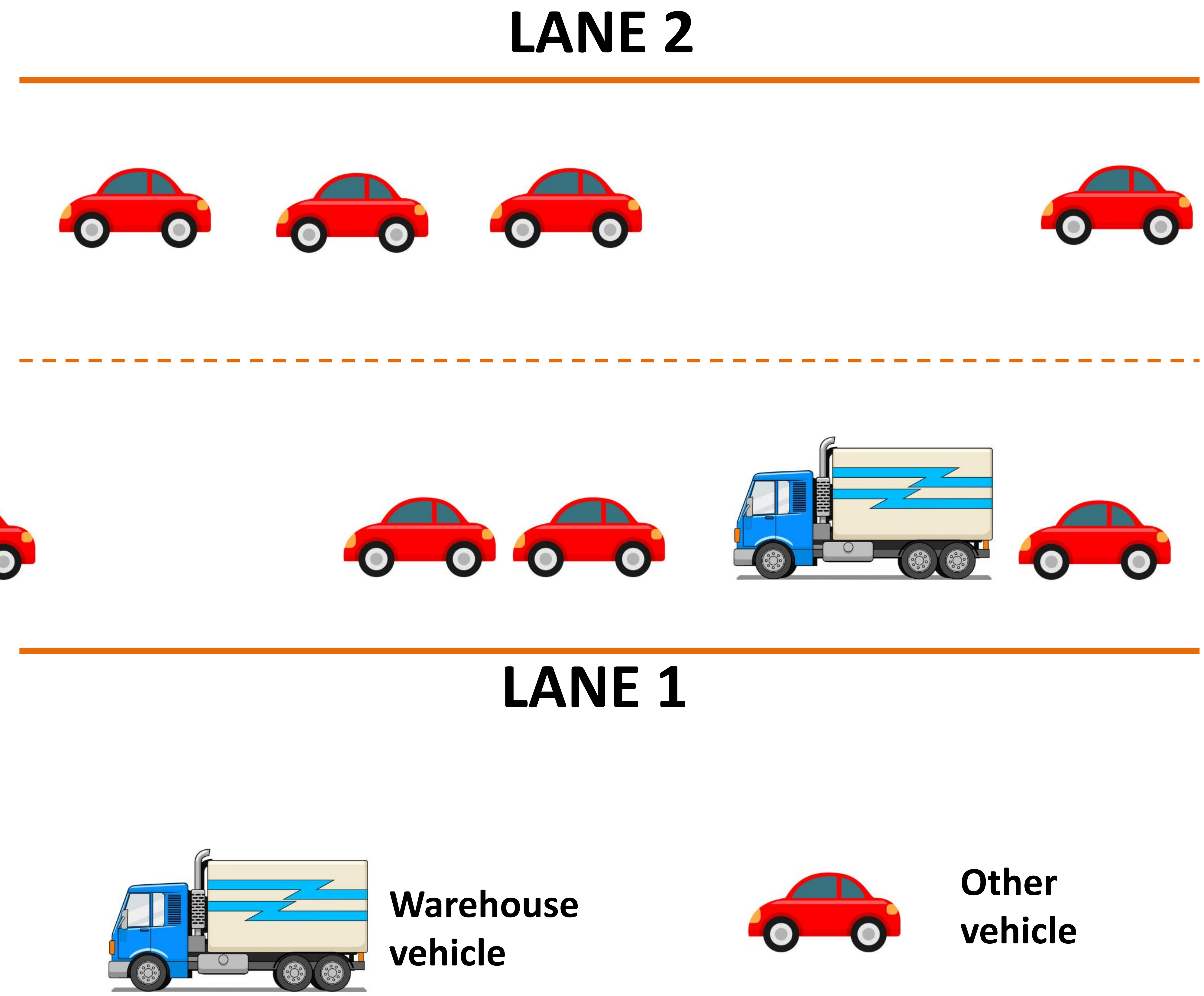


Module 2 (Markov Decision Process (MDP))

Broad aspects of the solution to Example 3:

- If the environment changes the future rewards (that depends on the environment) also changes.
- Hence, while making a decision, the autonomous system SHOULD NOT employ a "greedy strategy". E.x. in this case even through lane 2 seems less jammed from where the warehouse vehicle is, it is in fact more jammed compared to lane 1 in the long run.

Overview of the modules of this course



Module 3 (Reinforcement Learning)

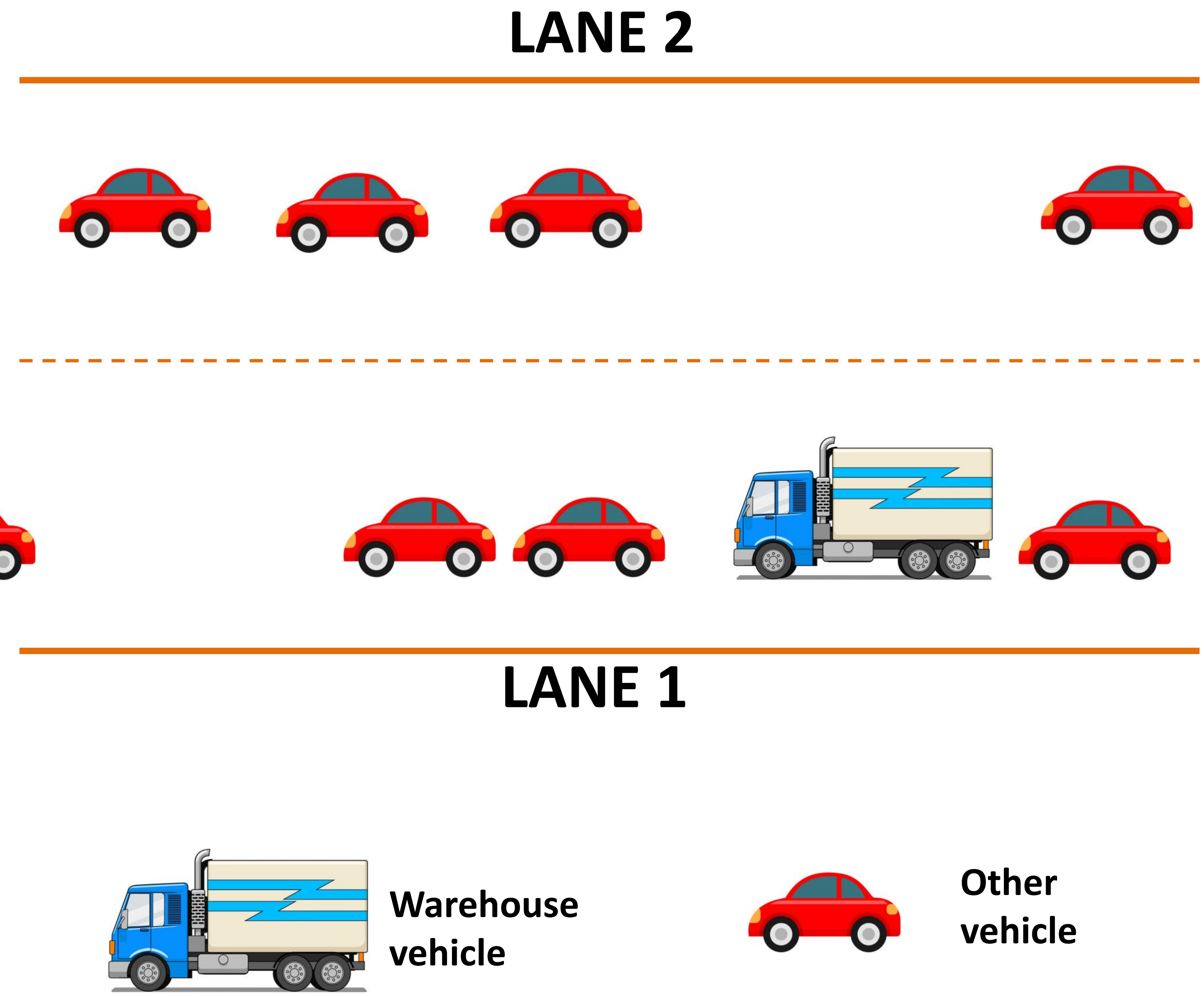
Example 4:

- Same as Example 3 with the following modification:
  - The autonomous system **DOES NOT know the probabilistic model of the uncertainty**. Hence, it is both a planning and a learning problem.

Broad aspects of the solution to Example 4:

- The broad aspects for this example is same as that of Example 3 with the additional component that the autonomous system has to learn the optimal strategy.

Overview of the modules of this course

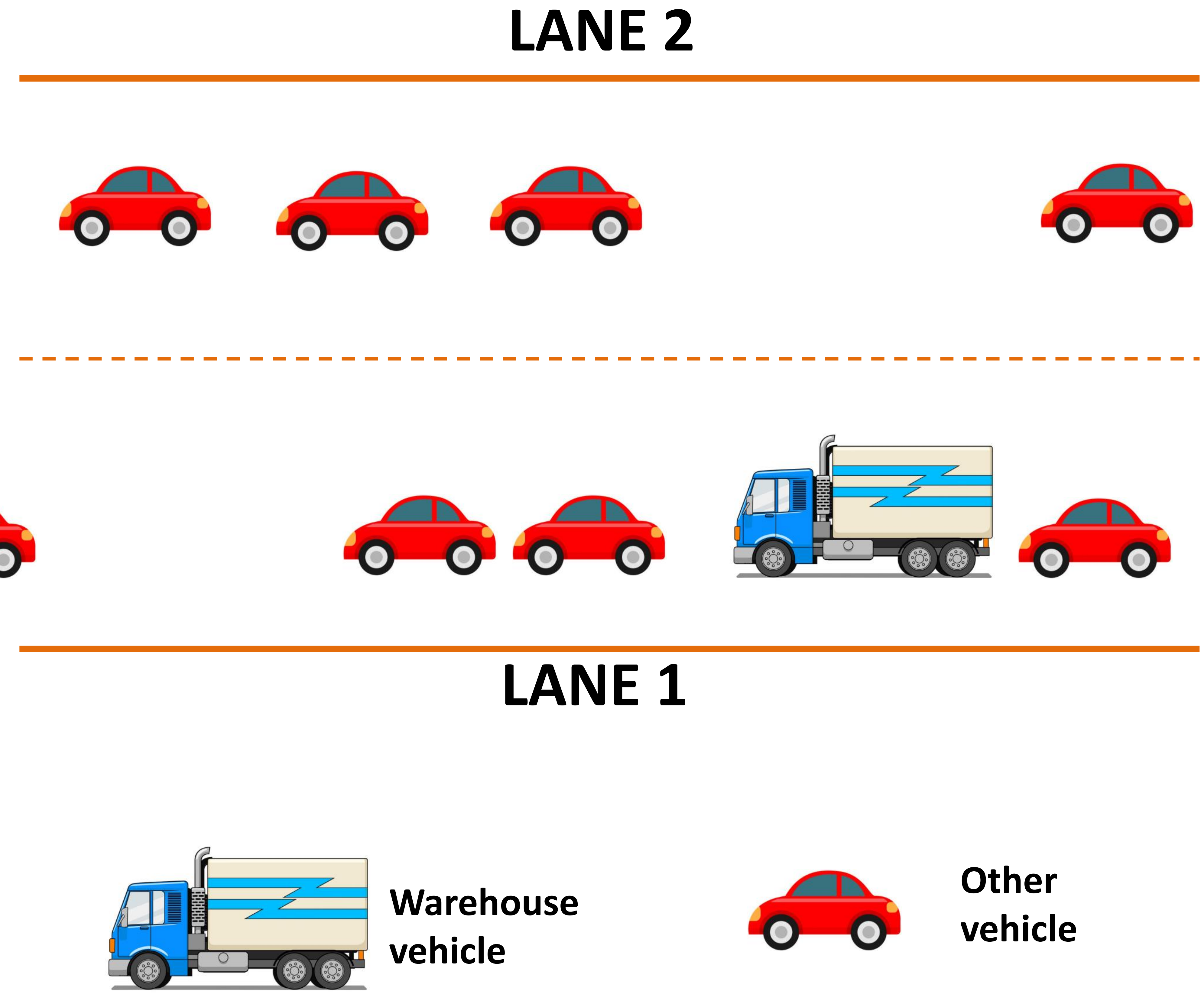


Module 4 (Deep Reinforcement Learning)

Example 4:

- ➢ NOTE: We are dealing with Example 4 (same as before and without any change).
- ➢ As mentioned while discussing Example 3, the warehouse vehicle knows the states of all the cars in the road through GPA and VANET.

Overview of the modules of this course



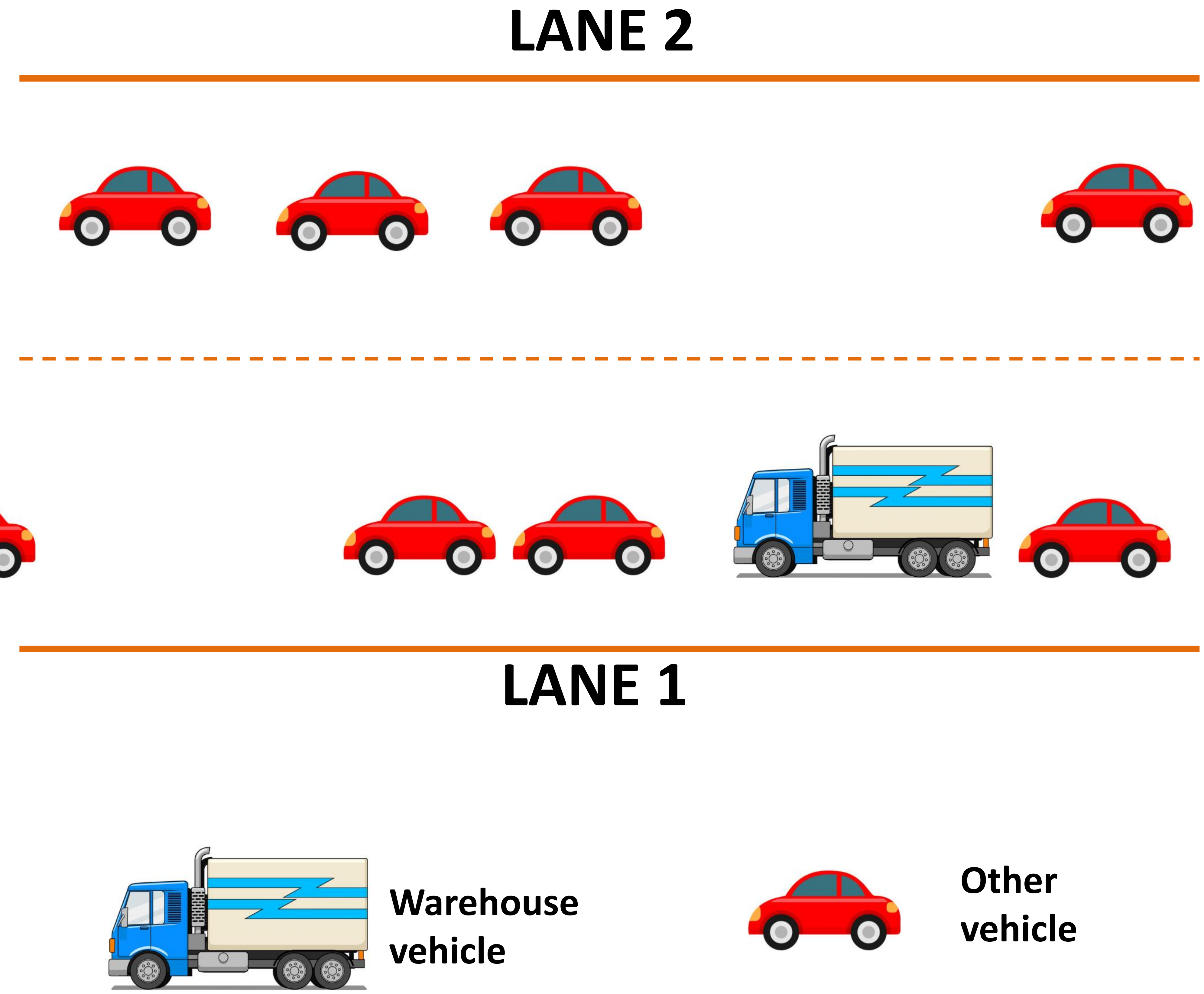
Module 4 (Deep Reinforcement Learning)

Example 4:

- **Question: How many car's states does the warehouse vehicle have to know in order to make a "sufficiently good" decision?**

- This question is important because while the state of a car that is 5 kms ahead may improve the decision a very little bit, it will definitely increase the computational complexity by a significant amount if the warehouse vehicle has to consider the state of all the cars in 5 kms radius.

Overview of the modules of this course



Module 4 (Deep Reinforcement Learning)

Example 4:

- ➢ Deep Reinforcement Learning (DRL) helps in **reducing computational complexity when state space is large**. But how?

- ➢ It is because neural networks are good in finding patterns, i.e. **if neural network can find optimal action for one state, it can predict the optimal action for another state**. This "generalization" capability reduces the learning time.

Overview of the modules of this course

Module 4 (Deep Reinforcement Learning)

Example 5:

- In this example the observation of the warehouse vehicle is an image (rather than GPS data through VANET in Example 4).

- The need for DRL becomes even more relevant here because if an image is 1028*1028 with 256-bit color, then the state space* is $256^{1028 \cdot 1028}$ which is humongous!

- CNNs are the de-facto neural network model for images.

Lecture Content

- What is Reinforcement Learning (RL)?
- Overview of the modules of this course (**through examples**).
- **Course logistics.**
  - **Prerequisites**
  - **References**
  - **Marks Distribution**
  - **Office hours**
  - **Practice problems**
  - **MU Email Id**
- Miscellaneous Topics.

Prerequisites

- Probability (strong knowledge).

- Linear algebra (not much).
  - Should at least know how to do matrix multiplication.
  - Matrix and vector norms.

- Machine Learning/Deep Learning.
  - Required for **(i)** around 25% of Module 1, and **(ii)** all of Module 4.
  - **I have already emailed Deep Learning tutorial to all those students who are there in the current student list.**

References

- Books:

  1. Reinforcement Learning: An Introduction, Sutton and Barto, 2nd Edition.

  Available online for free

  Well written book (adequate amount of math). For modules 1 to 3, you MUST follow read it.

- Video lectures:

  1. NPTEL, Indian Institute of Technology Madras, Prof. Balaraman Ravindran.
  2. CS 234, Stanford University, Prof. Emma Brunskill.
  3. CS 285, University of California Berkeley, Prof. Sergey Levine.
  4. CS 885, University of Waterloo, Prof. Pascal Poupart.

- My lecture notes will be available in Dropbox.
  - Already shared the Dropbox link with all the students in the current student list.

Marks Distribution

- Programming assignments (50%). **In groups of 3-4. Will send an email about this.**
  - There will be 5 assignments each having weightage of 10% of the total grade.
  - Programming assignments 1 to 4 will be from modules 1 to 4 respectively.
  - Programming assignment 5 will be either from module 4 or module 5.
  - **Academic dishonesty will be heavily penalized. Ask your seniors!**
- Minor 2 (15%).
- Final Exam (35%).

Office Hours

- No weekly office hours! But, I will have office hours before:
  - Exams
  - A week before programming assignment deadline.

(Based on my previous experience, students don’t come for office hours unless it is exam time.)

- You can also come to my cubicle by booking **prior appointments** mentioning the **topic of discussion in brief**. (so that I have some time to prepare).

Practice Problems

- I will be giving at most one practice problem set for modules 1, 3, 4, and 5.

- Module 2 is likely to have two practice problem set.

- I have already uploaded previous years exam questions which should act like practice problem set.

MU Email Id

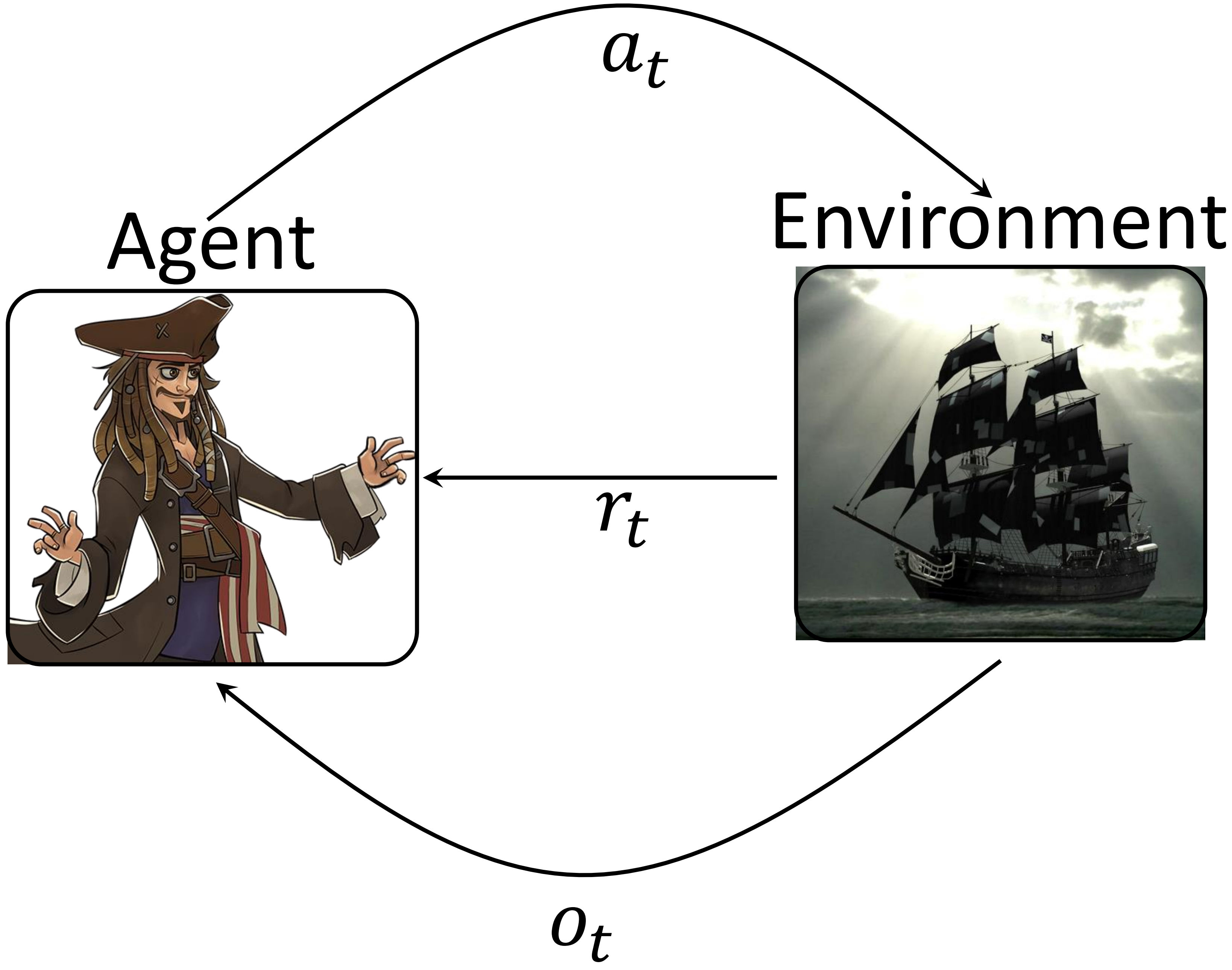
- I will be creating a **group email**. If you did not receive any email from me about this course yesterday (especially **Masters/PhD students**) please email me your **Roll Number** and **MU Email Id**.

My email: **gourav.saha@mahindrauniversity.edu.in**

Lecture Content

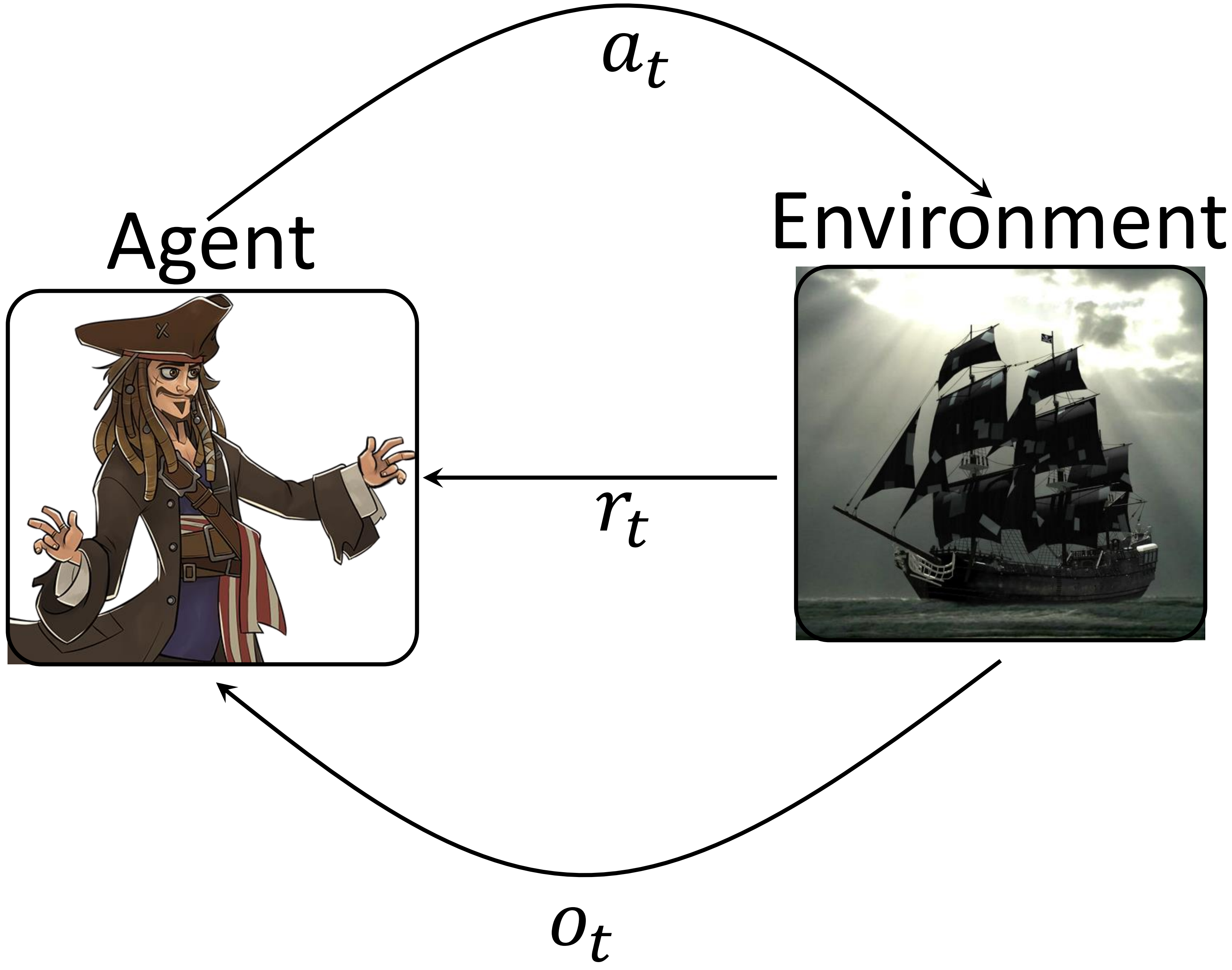
- What is Reinforcement Learning (RL)?
- Overview of the modules of this course (**through examples**).
- How is RL different from other AI topics?
- Course logistics.
- **Miscellaneous Topics.**
  - **"I want to know the application of RL".**
  - **Why learn the working of RL algorithms?**

"I want to know the application of RL"



- I am in the process of recording **lecture 0** which is **three-part video lecture series** designed to give you a **big picture of RL** so that you can **start applying RL for some projects**.
  - Completely **optional** to watch it.
  - Two of these three lectures are going to be **coding lectures**.
  - These are **not going to be graded** projects.
  - You have to find the projects yourself; I can give you minimum guidance given my timetable.

“I want to know the application of RL”

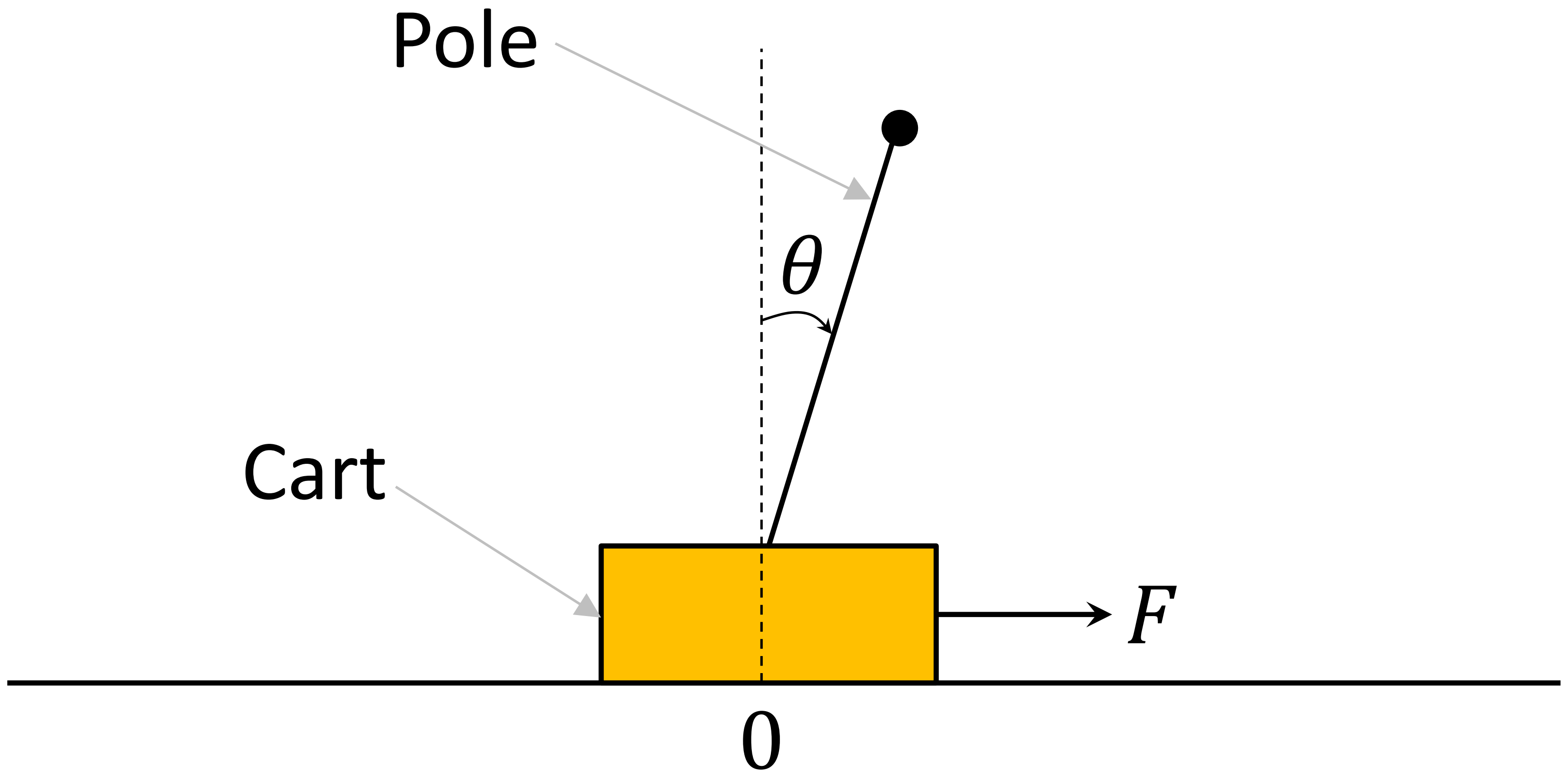


- Done recording one of the three-lectures. Will post it latest by Wednesday evening.

- If lecture 0 is enough to get started with RL, what is the **importance of the remaining lectures?** Answer in the next few slides…

Why learn the working of RL algorithms?

Original Cart-Pole System



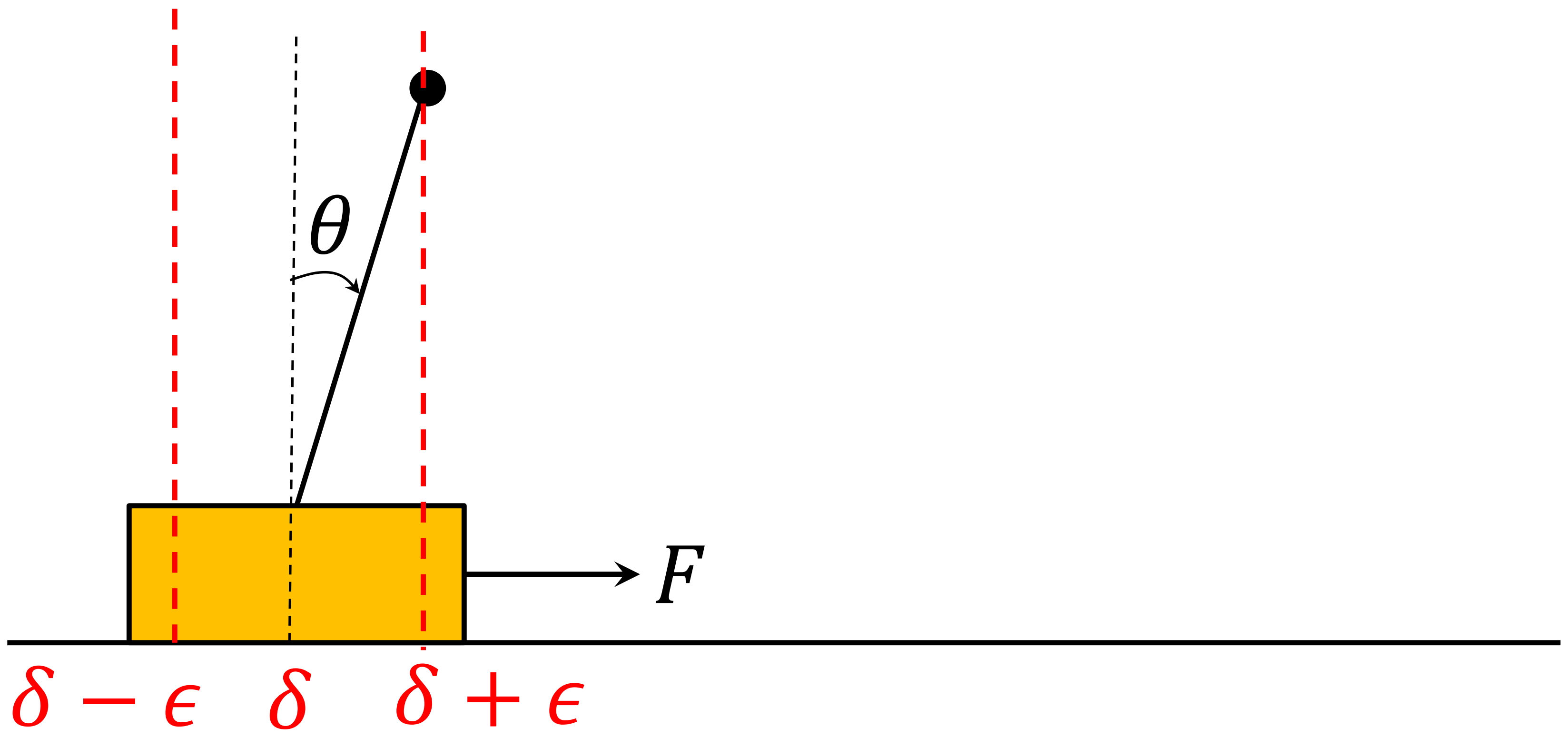
- Let's consider an example to answer the above question.

- The figure shows the cart-pole environment. The objective is to design and RL agent to control the force F (can be in both directions) on the cart to hold the pole in the upright position (θ should be between $\pm 10°$).

- Designing an RL agent for this system is quite straightforward.

Why learn the working of RL algorithms?

Modified Cart-Pole System



- No let's consider a modification of the original cart-pole system.

- The objective is to design and RL agent to control the force F (can be in both directions) on the cart to hold the pole in the upright position (θ should be between ±10°) and **to also bring the center of the cart within a certain region shown in red.**

- **I could not solve this problem yet!** I gave this problem as a small part of programming assignment to the previous batch and still unsolved.
  - We can design a "conventional"controller using tools from control systems course.

Why learn the working of RL algorithms?

- Just last week, I found a paper in **American Control Conference** titled **“Robust nonlinear set-point control with reinforcement learning”** which **might** have addressed this problem.

“One observed drawback of standard RL methods is that they take a **very long time to learn a policy**, and may even fail on simple set-point control problems. As discussed in the paper, one problem is that because of **the training on multiple set-points the importance of efficient exploration is increased**. If common stochastic Gaussian policies are used for this purpose, the **exploration of different amplitude domains is far too unlikely**, and therefore the training process takes much longer time than necessary. The first contribution of the paper therefore **applies an untuned proportional-controller (P-controller) as a prior controller to aid the exploration**. After the modification, the policy learns how to track different setpoints much more efficiently.”

Why learn the working of RL algorithms?

- Just last week, I found a paper in **American Control Conference** titled **"Robust nonlinear set-point control with reinforcement learning"** which **might** have addressed this problem.

- The overall idea is to use a "not so good controller" to train the RL agent first (rather than RL agent training from scratch) and then the RL agent keeps learning and optimizing itself.
    - To implement this change we need to tweek existing RL algorithms by a little bit.

- **Finally the answer to the question in the heading**: It is important to learn the working of RL algorithm so that you have an **intuition** which in turn will help you in **modifying existing algorithms** (like the modified cart-pole system) and to **invent new RL algorithms** when needed.

Why learn the working of RL algorithms?

Few other takeaways from this example:

- As I said, a "conventional controller" can be used for the modified cart-pole system. So RL is not a "one-stop solution". **Don't get carried away by the hype!**
- RL, though a pretty old field, has a long way to go. The existing libraries for RL is not enough to train many of the real-world problems. Hence, increases the need to know the working of RL algorithms.

Why learn the working of RL algorithms?

Few other takeaways from this example:

- As I said, a “conventional controller” can be used for the modified cart-pole system. So RL is not a “one-stop solution”. **Don’t get carried away by the hype!**

- RL, though a pretty old field, has a long way to go. The existing libraries for RL is not enough to train many of the real-world problems. Hence, increases the need to know the working of RL algorithms.

Not claiming that after taking this course you will be able to apply RL to real-world problems. But, it will definitely give you a foundation.

Thank you