AdTrans project, 2023-1-PL01-KA220-HED-000158917

ADTRANS
ADVANCED DATA ANALYSIS FOR TRANSPORTATION AND MOBILITY ACADEMIC TEACHERS

# Session 3

# Autonomous (Self-Driving) Cars

*Paraskevi Fasouli*

Co-funded by the European Union

# Overview

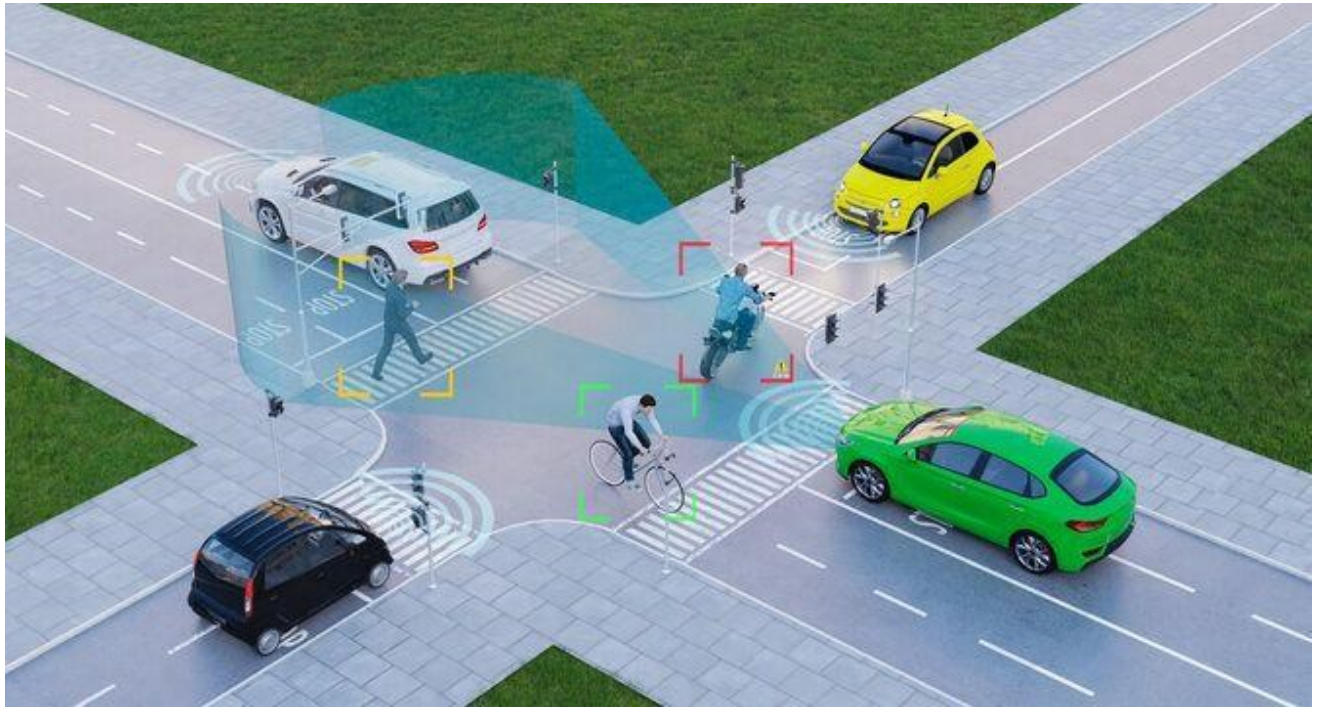
**Self-Driving Cars & SAE Levels of Automation**

**Software Architectures of Self-Driving Cars**

**Modular Stack Overview & Explanation**

**End-to-end Architecture Overview**

# What defines a self-driving car?



Google has experimented with a fleet of driverless cars since 2009, driving...

**more than 700,000 miles without a crash.**

# Self-Driving Ca

## But what are the challenges?

While they may reduce crashes, saving money and lives, introducing self-driving cars will likely present its own set of challenges.

**Create new safety risks such as...**
system or technological failures, poor safety when vehicles encounter situations that aren't included in their program, and/or encourage users to engage in risky behaviors.

**Greater connectivity can cause privacy concerns such as...**
vulnerability to information abuse, unwanted GPS tracking and data sharing.

**The convenience of driverless cars could lead to more travel...**
which could cause increased pollution and raise the costs of parking.

## Are computers better drivers than humans?

- Self-driving cars could eliminate or drastically reduce the risks of accidents caused by driver error.
- Computers have greater awareness and faster reaction times than humans, while not being subject to **distractions**, **fatigue** or **alcohol consumption**.
- Drinking and driving is the leading cause of car crashes.

**More than 90%** of crashes are caused by **human error**.

**81% can be avoided** by efficient use of vehicular communications systems.

In 2013, car accidents caused...
- **35,000 deaths**
- **3.8 million injuries**

Self-driving cars would reduce car accidents by 90%...
- **preventing 30,000 deaths**
- **2 million less injuries**
- **$400 billion in savings by 2020**

# SAE Levels of Driving Automation

**Level 0**

**No Driving Automation**

A human is completely driving the vehicle.

**Level 1**

**Driver Assistance**

A human is driving, but the vehicle offers adaptive cruise control.

**Level 2**

**Partial Automation**

Adaptive control with both lateral & longitudinal aspects. The driver is still in control and monitors the rest of the driving tasks.

# SAE Levels of Driving Automation

**Level 3**

**Conditional Automation**

The car can operate adaptive cruise control & collision avoidance. The driver must be prepared to intervene.

**Level 4**

**High Automation**

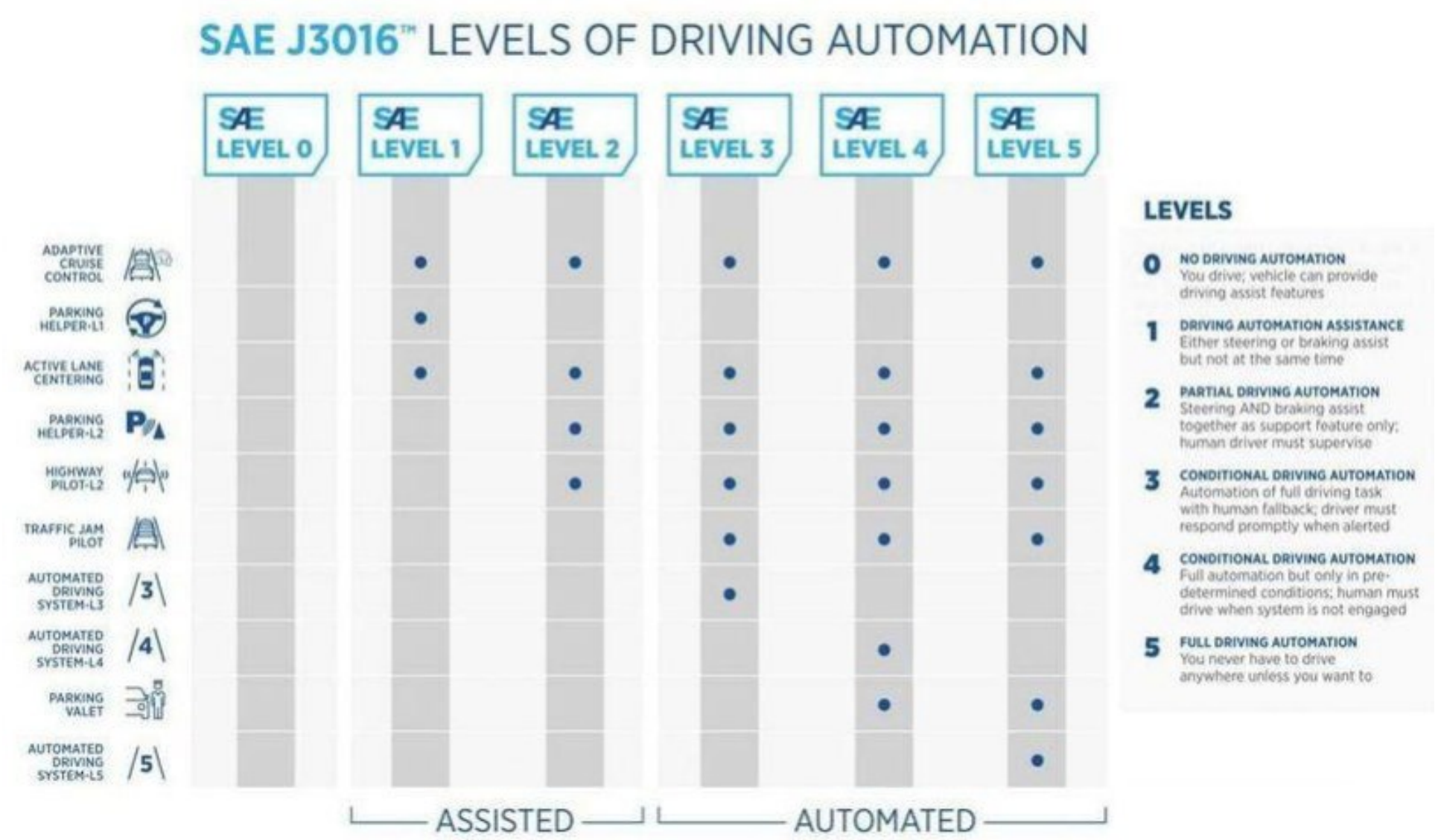
All driving tasks are automated under specific circumstances. The driver can control at any time.

**Level 5**

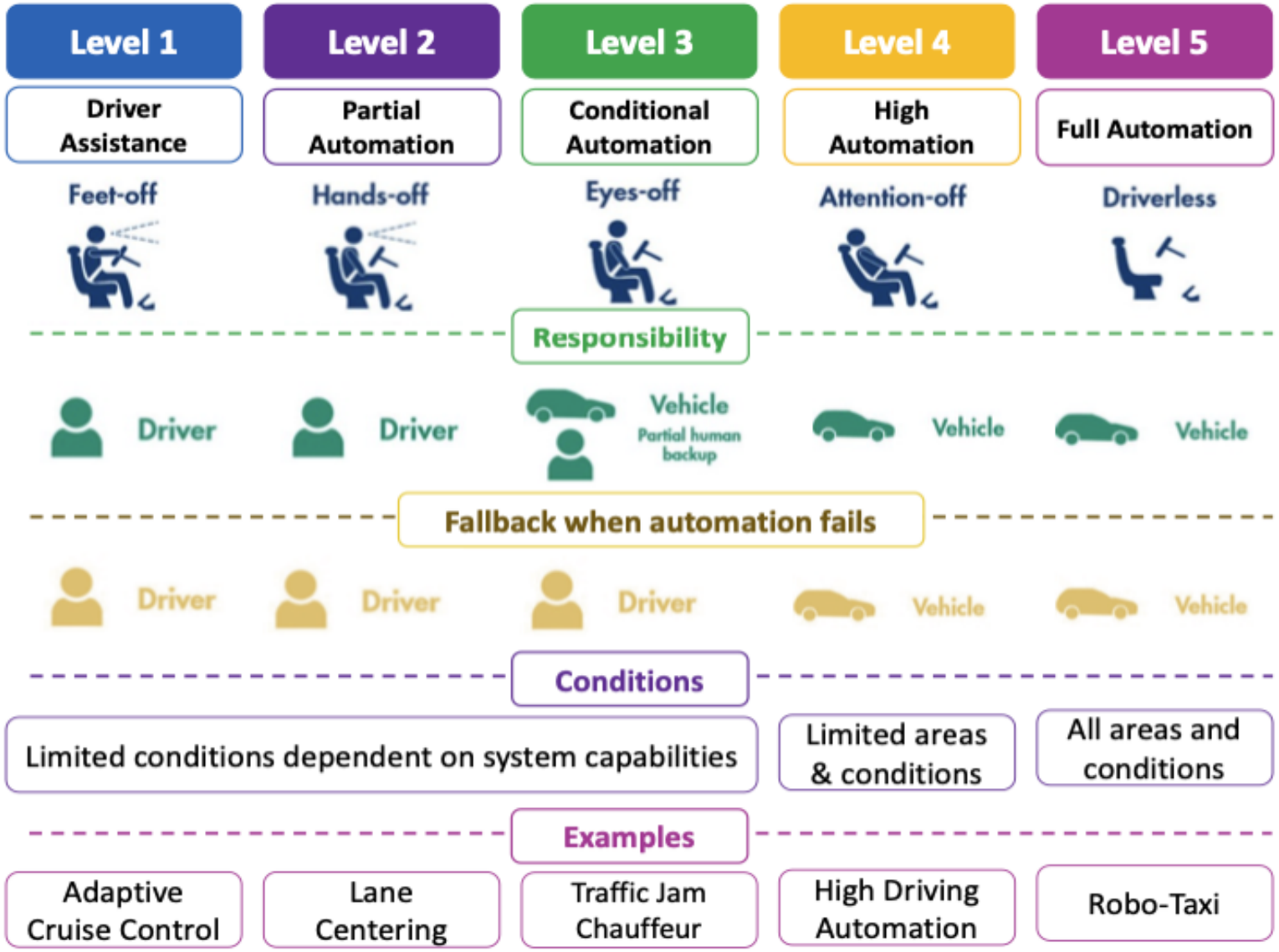
**Full Automation**

All driving tasks are automated. Humans' interaction is not needed.

***SAE**: Society of Automotive Engineers*



# SAE Levels of Driving Automation



# Key Features of Self-Driving Cars

| | |
|---|---|
| Sensors: | • Self-driving cars are equipped with an array of sensors that enable them to perceive their surroundings. These sensors include **cameras**, which provide visual information about the environment, **radar**, which detects objects and their velocities, **lidar**, which measures distances using laser pulses, and **ultrasonic sensors**, which help with close-range detection. |
| Control Systems: | • Advanced control systems process the data from the sensors in real-time, allowing the vehicle to **make decisions** and adjust its movements accordingly. These control systems are often powered by **artificial intelligence and machine learning algorithms** that learn from data and improve over time. |
| Mapping and Localization: | • Self-driving cars rely on **high-definition maps and precise localization techniques** to understand their position and navigate routes accurately. These maps include detailed information about roads, lanes, traffic signs, and other infrastructure elements. |
| Decision-Making Algorithms: | • Self-driving cars use decision-making algorithms to interpret sensor data, detect objects, predict their movements, and plan safe trajectories. These algorithms consider factors such as **traffic rules, road conditions, and potential hazards** to make informed decisions in real-time. |
| Redundancy and Fail-Safe Mechanisms: | • Self-driving cars incorporate redundancy and fail-safe mechanisms to **ensure safety and reliability**. This includes backup systems for critical components, redundant sensors, and protocols for handling unexpected situations, such as sensor failures or adverse weather conditions. |

# Software Architectures of Self-Driving Cars

## Modular Stack Approach

Refers to the layered architecture of software and hardware components that work together to enable autonomous driving capabilities. Each layer of the stack performs specific functions.

## End-To-End Approach

Aims to learn a single integrated model that directly maps sensor inputs to driving actions. The entire driving task is learned directly from raw sensor data without explicit decomposition into intermediate tasks.

# Modular Stack: Idea

**Ultimate goal of a self-driving car is to go from A to B**

How can we fulfil the goal? We should follow these steps:

1. Decompose the driving problem into multiple subproblems
2. Individual modules solves the subproblems and pass the result forward
3. Orchestrate a pipeline: Sensor inputs -> Control signals (Trajectory)

Our pipeline should be Interpretable to a certain degree

Usually has a mixture of traditional methods and deep learning

# Modular Stack: Requirements

**Ultimate goal of a self-driving car is to go from A to B**

While driving from A to B, a set of **behaviours** must be planned:

- Behaviour: What is required step to satisfy the abstract goal
- Consider the road rules and regulations
- Infer the future configurations of the dynamic objects (object detection, tracking and motion prediction)
- Know where the static objects are
- Produce the appropriate next behaviour

Now given a **planned behaviour**, generate a set of feasible **local trajectories**

- Trajectories are contained by the kinematic model of the vehicle
- For a selected trajectory, generate lateral and longitudinal velocity control commands

# Modular Stack: Tasks

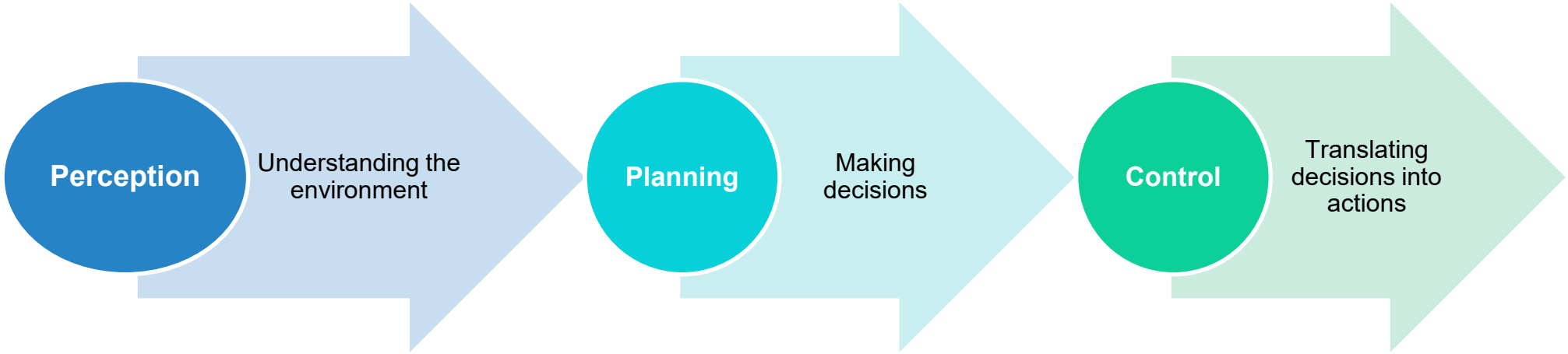
**Ultimate goal of a self-driving car is to go from A to B**

So in order to generate a proper local trajectory to follow, we need to:

- **Perceive and map the environment:**
  - Recognise static objects & Road signs (Object detection & classification)
  - Recognise dynamic objects & Predict future paths (Object detection, classification, tracking & motion prediction)
  - Recognise the driving surface, lane & boundaries (Segmentation, key point recognition or polynomial fitting)
- **Localise the vehicle:**
  - Determine the vehicle odometry using LIDAR/Camera and/or IMU (Inertial
  - measurement unit)
  - Fuse odometry measurements with GNSS to localise the ego vehicle on a map
- **Detect the drivable surface:**
  - Determine where the vehicle can drive using detected static objects
  - Constrain the drivable surface using predicted motion of the dynamic objects and applied regulations

# Modular Stack: Overview

# 3 Categories



# Modular Stack: Overview

# 8 Modules

# Modular Stack: Overview

**1. Perception:**

- The perception layer consists of sensors such as cameras, radar, lidar, and ultrasonic sensors that capture data about the vehicle's surroundings. This data includes information about other vehicles, pedestrians, road signs, lane markings, and obstacles. Perception algorithms process this sensor data to detect and classify objects in the environment.

**2. Localization:**

- The localization layer determines the vehicle's precise position and orientation within its environment. This is achieved using GPS, inertial measurement units (IMUs), and odometry data from wheel encoders. Localization algorithms fuse these sensor inputs with high-definition maps to accurately localize the vehicle’s position relative to its surroundings.

**3. Mapping:**

- The mapping layer generates and maintains high-definition maps of the vehicle's operational domain. These maps contain detailed information about road geometry, lane markings, traffic signs, traffic signals, and other relevant infrastructure features. Mapping algorithms combine data from surveying vehicles, satellite imagery, and crowd-sourced sources to create and update these maps.

**4. Prediction:**

- The prediction layer anticipates the future behavior of objects in the environment, including other vehicles, pedestrians, and cyclists. Prediction algorithms analyze historical motion patterns, current trajectories, and contextual information to forecast the future positions and intentions of these objects. This information is essential for planning safe and efficient driving maneuvers.

# Modular Stack: Overview

### 5. Planning:

- The planning layer generates a sequence of driving commands that specify the vehicle's trajectory over a short time horizon. Planning algorithms consider the vehicle's current state, perception data, localization information, and predicted trajectories of surrounding objects to generate collision-free and socially acceptable driving paths. These paths are optimized to achieve driving objectives while adhering to traffic rules and safety constraints.

### 6. Control:

- The control layer executes the driving commands generated by the planning layer, controlling the vehicle's acceleration, braking, and steering actuators. Control algorithms ensure smooth and precise execution of driving maneuvers, taking into account the vehicle's dynamics, actuators' limitations, and environmental conditions.

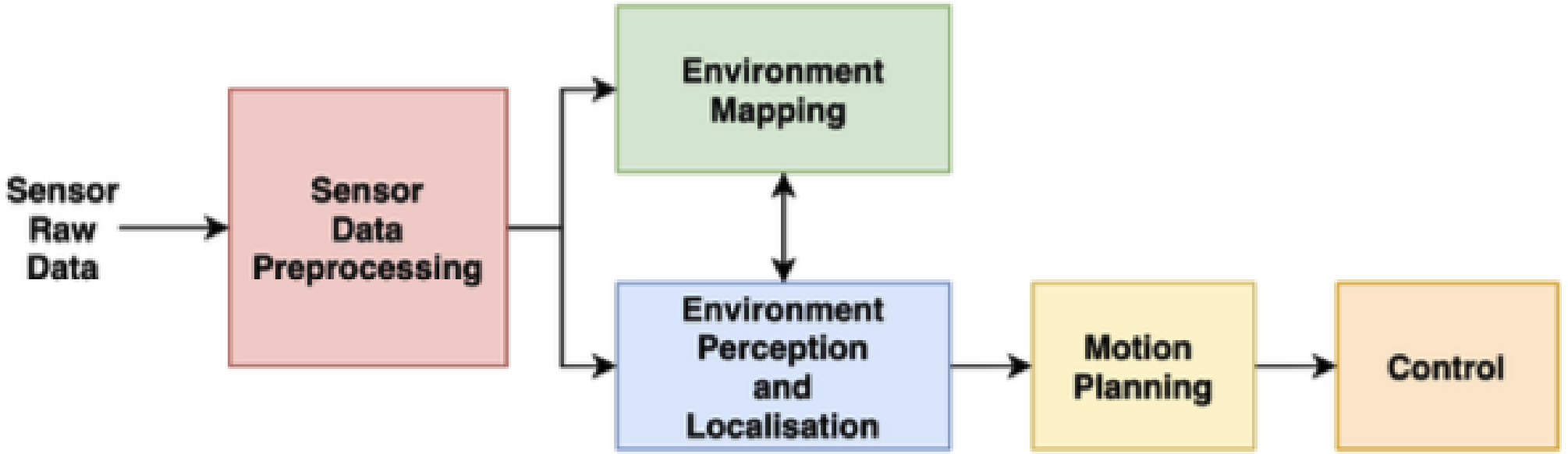
### 7. System Integration:

- The system integration layer orchestrates the interactions between different modules of the self-driving system, ensuring seamless communication and coordination. This layer manages data fusion, sensor calibration, software updates, diagnostics, and fault detection and recovery mechanisms.

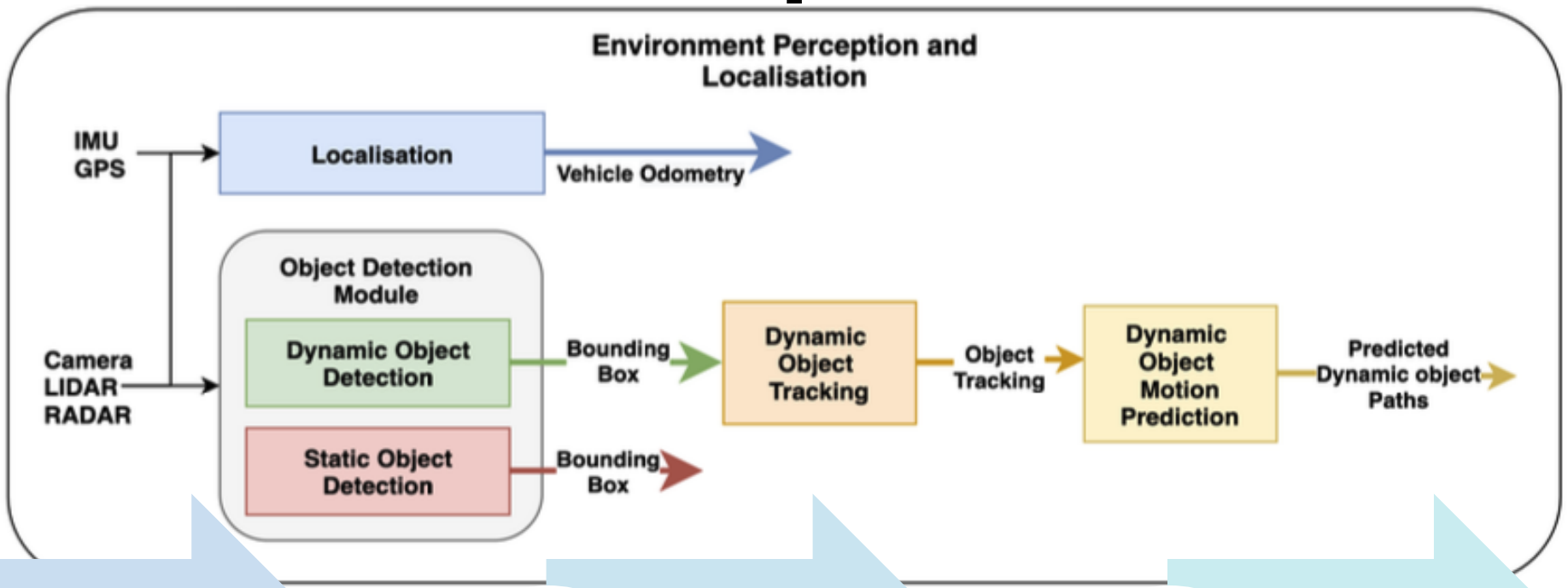
### 8. User Interface:

- The user interface layer provides a means for human interaction with the self-driving system. This includes graphical interfaces, voice commands, and haptic feedback mechanisms that allow passengers to monitor the vehicle's status, provide input, and intervene if necessary.

# Modular Stack: Overview



# Modular Stack: Perception




**1. Perception:** Get to know the dynamic and static aspects of the surrounding

**2. Localisation:** Find location and orientation of the vehicle

**3. Mapping:** Get to know the surrounding

# Modular Stack: Perception

**1. Perception:**

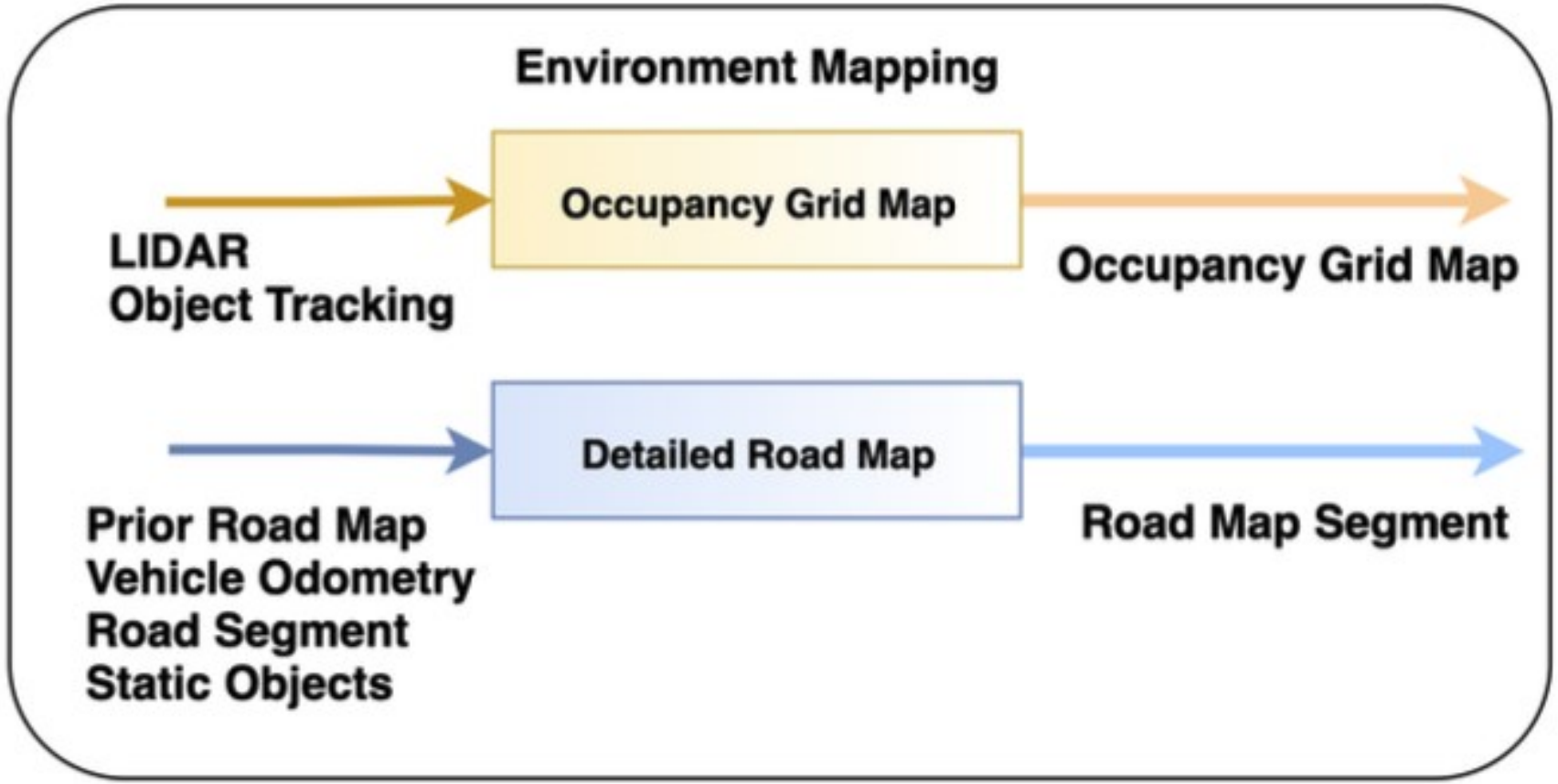
- **Inputs:** Camera, LIDAR, RADAR
- **Internal Process:** 3D/2D Object detection, Object tracking and motion prediction
- **Output:** 3D/2D bounding boxes, Segmentation instances, Motion trajectories

**2. Localization:**

- **Inputs:** IMU, LIDAR, Camera, GPS
- **Internal Process:** Vehicles Odometry Estimation, Sensor Fusion
- **Outputs:** Estimated vehicle odometry (Position and Orientation)

# Modular Stack: Perception - Mapping

**Job: Generate the maps containing the information about the surrounding of the ego vehicle**



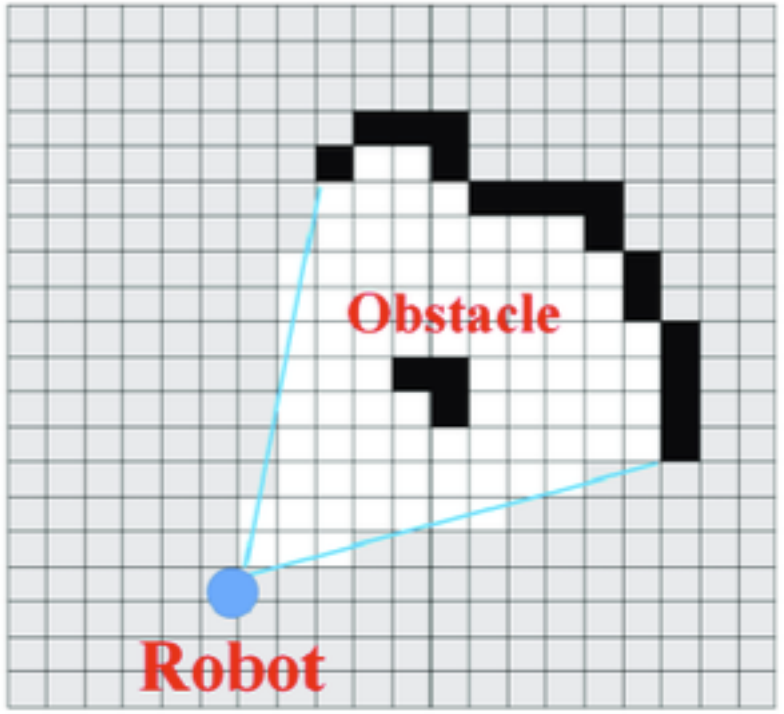
# Modular Stack: Perception - Mapping

### 3. Occupancy Grid Mapping:

- **Inputs:** LIDAR, Tracked Dynamic Objects
- **Internal Process:** Remove dynamic objects , Compute occupancy per grid using LIDAR
- **Output:** Occupancy Grid with static objects

### 3. Detailed Road Mapping:

- **Inputs:** Prior Road Map, Vehicle Odometry, Static objects, Road Segment
- **Internal Process:**
  - Build a complete understanding about the environment w.r.t the ego vehicle's configuration
  - Recognise the road signs and lane markings for mission and behavioural planning
- **Output:** Road map segment

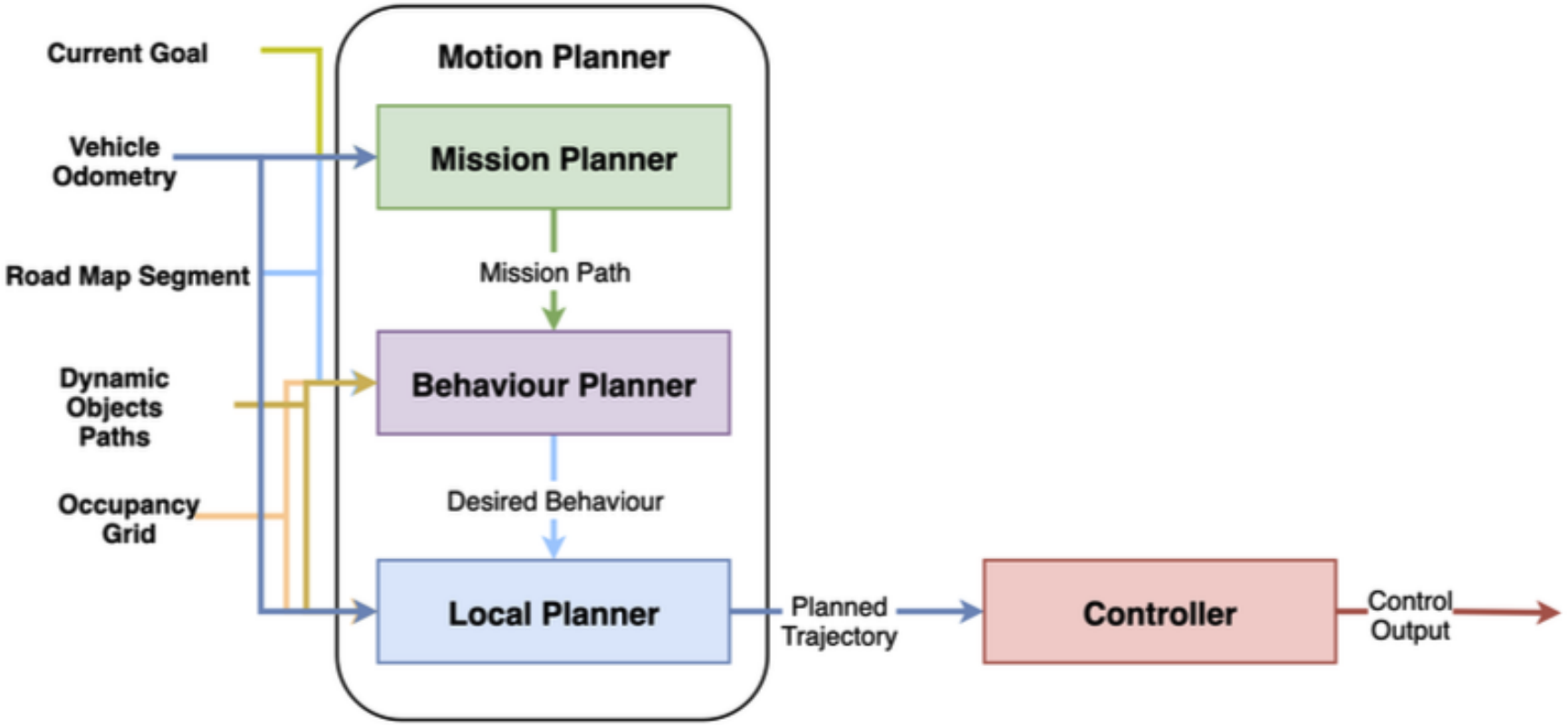



**Occupancy Grid Map**

*White pixels*: Robot can move
*Black pixels*: Obstacles
*Grey pixels*: Unmapped area

# Modular Stack: Planning – 4. Prediction & 5. Motion Planner

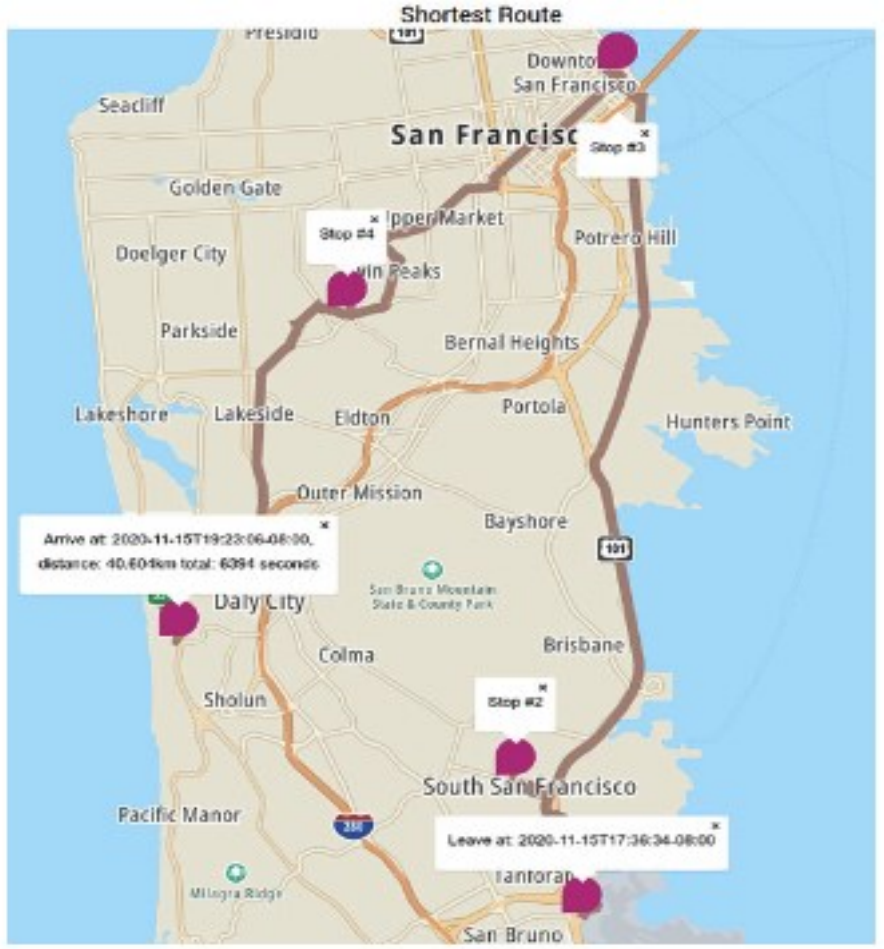
**Job: Generate the most suitable trajectory given the information about the surrounding and the destination**



# Modular Stack: Planning

## 5.1 Mission Planner:

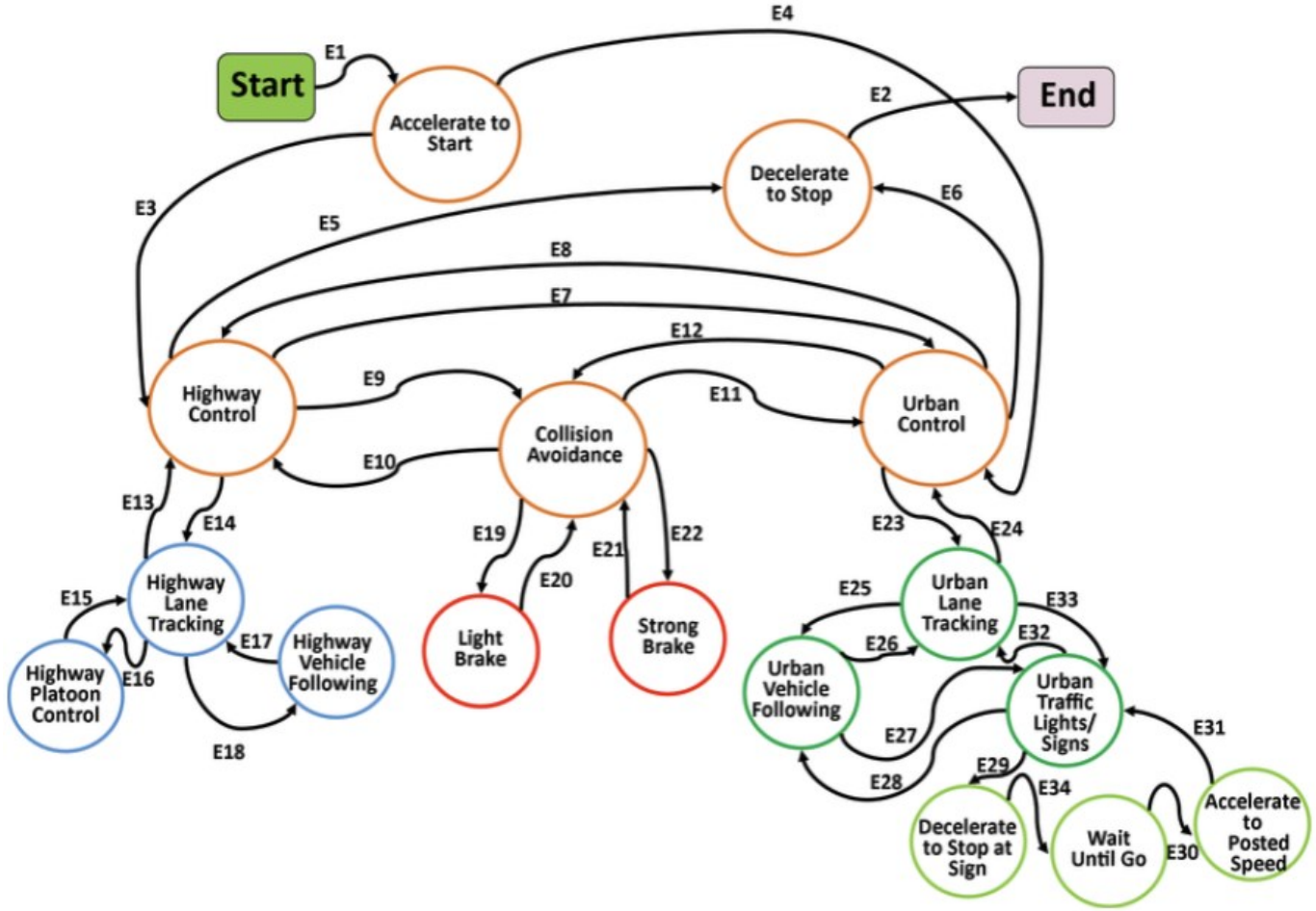
- Plan the mission from A —> B
- Solve long term planning problems (Including rerouting)
- **Inputs:** Current Goal, Vehicle odometry, Road Map Segment
- **Internal process:** Find the shortest path from A —> B given the vehicle position and environment information
- **Outputs:** Mission Path



# Modular Stack: Planning

## 5.2 Behaviour Planner:

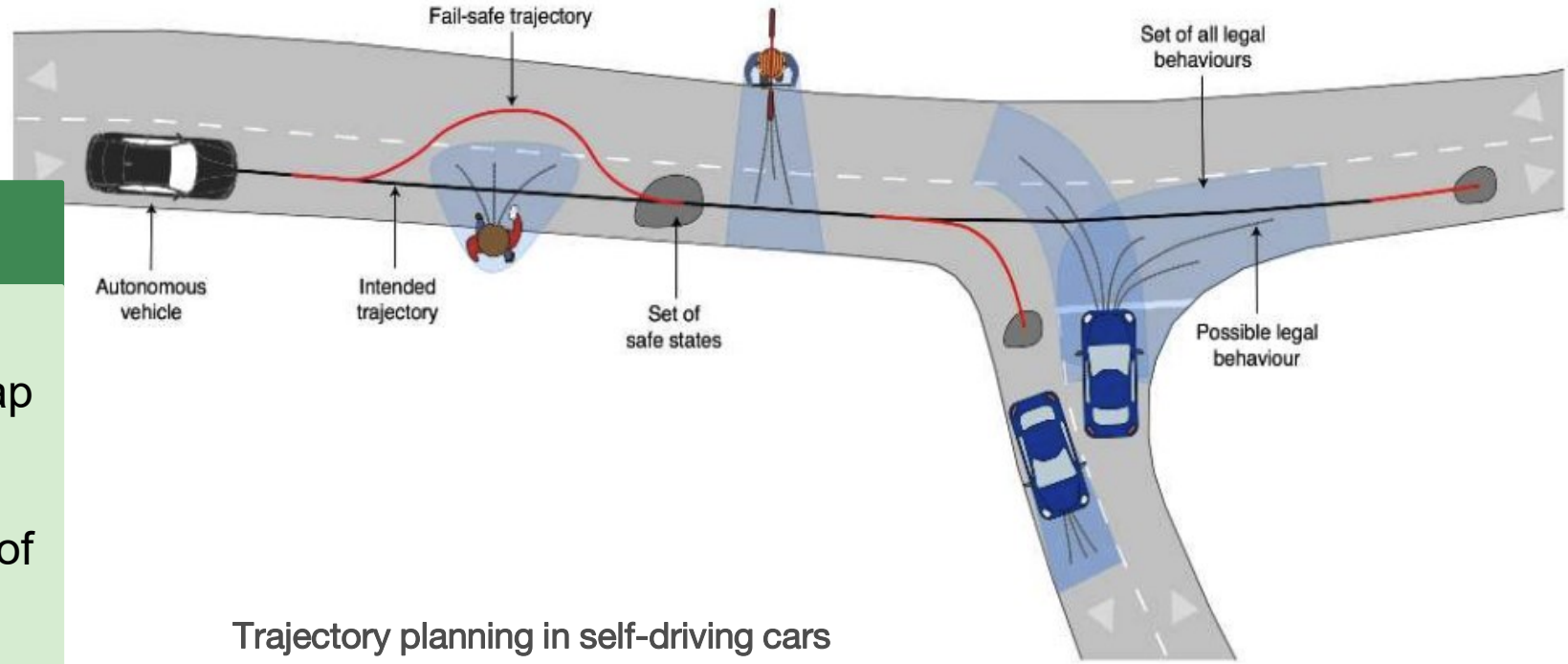
- Performs shot term planning
- **Inputs:** Vehicle odometry, Road map segment, dynamic object paths, occupancy grid
- **Internal process:** Determine a set of safe behaviours given the environment
- **Output:** Best manoeuvre with the behavioural constraints



# Modular Stack: Planning

## 5.3 Local Planner:

- Performs shot term planning
- **Inputs:** Vehicle odometry, Road map segment, dynamic object paths, occupancy grid
- **Internal process:** Determine a set of safe behaviours given the environment
- **Output:** Best manoeuvre with the behavioural constraints

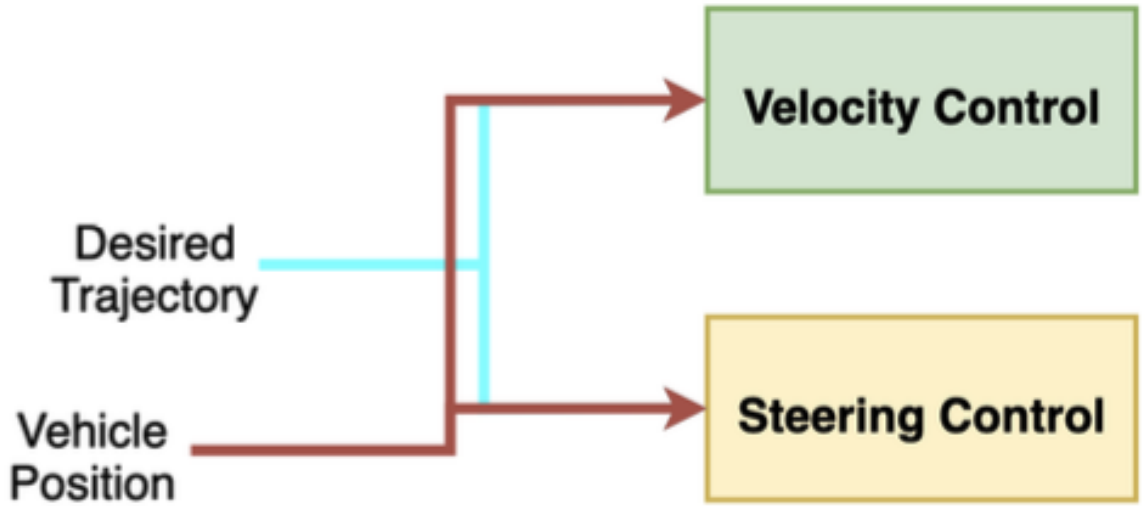



Trajectory planning in self-driving cars

# Modular Stack: Control

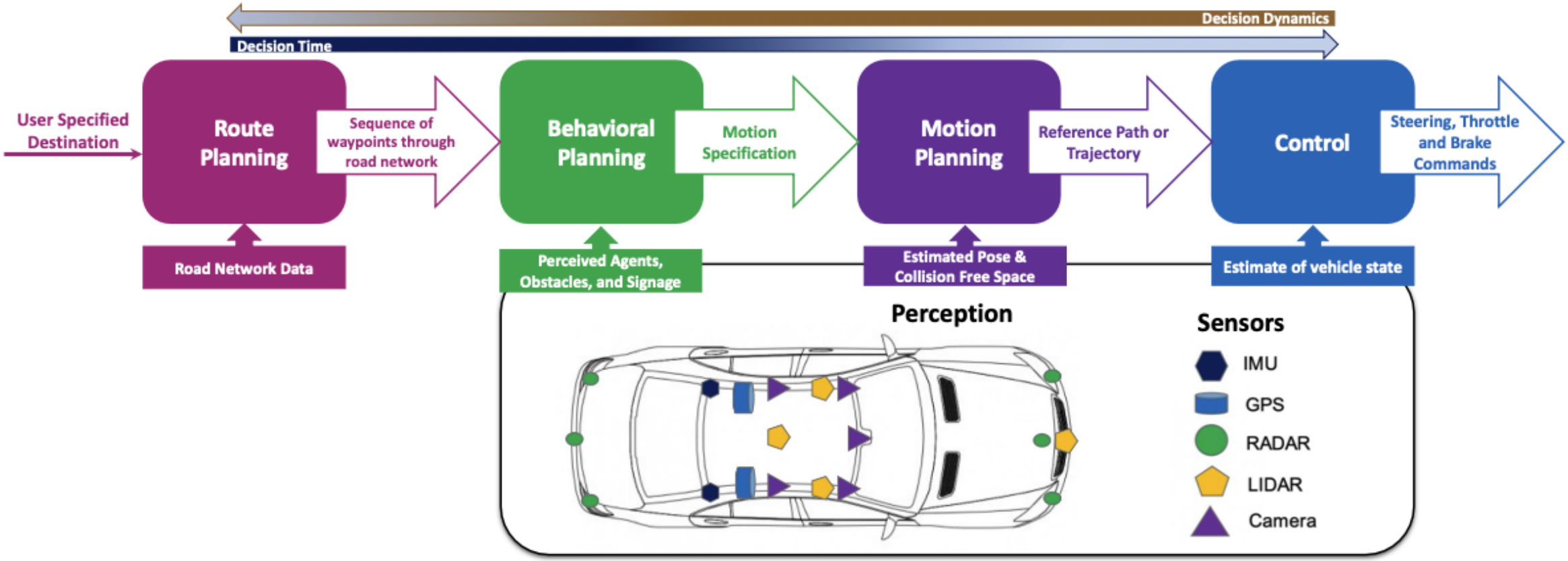
**6. Controller:**

- **Inputs:** Desired trajectory
- **Internal process:** Separate the task to longitudinal and lateral control
- **Output:**
  - Lateral Control: Steering angles
  - Longitudinal Control: Throttle, Break

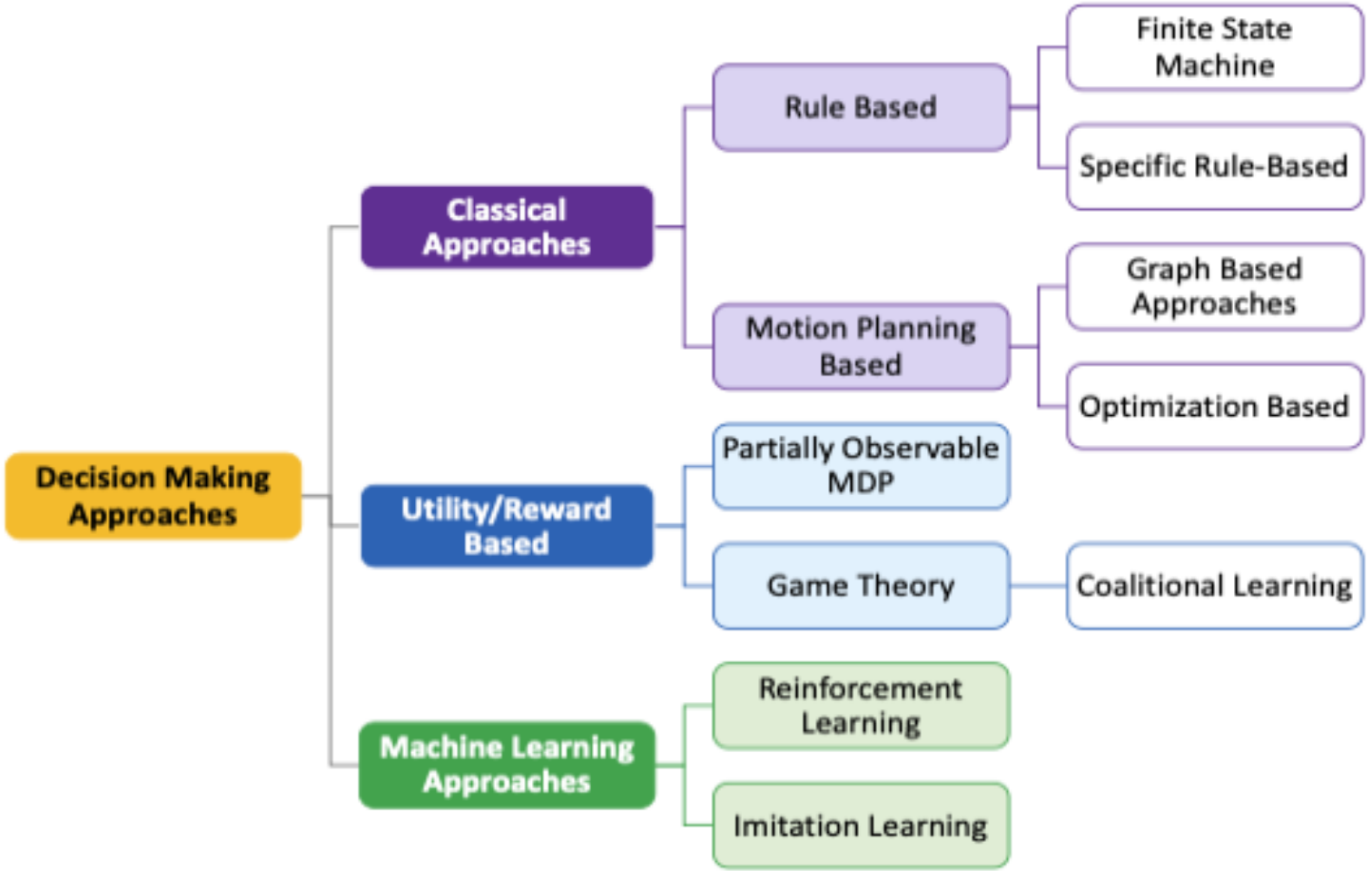



Trajectory planning in self-driving cars

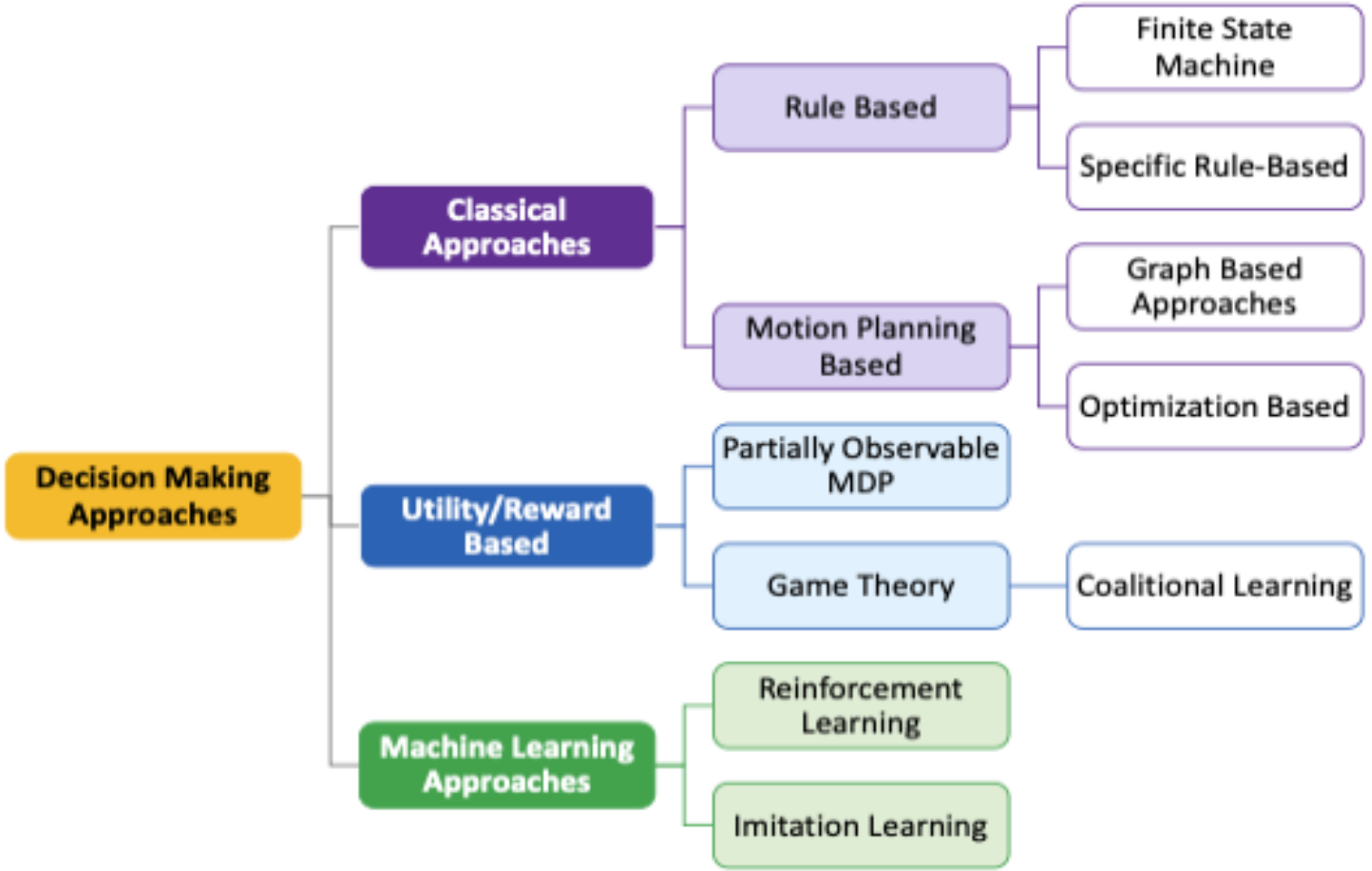
# Modular Stack: Recap



# Categorization of Decision-Making Approaches for Autonomous Vehicles

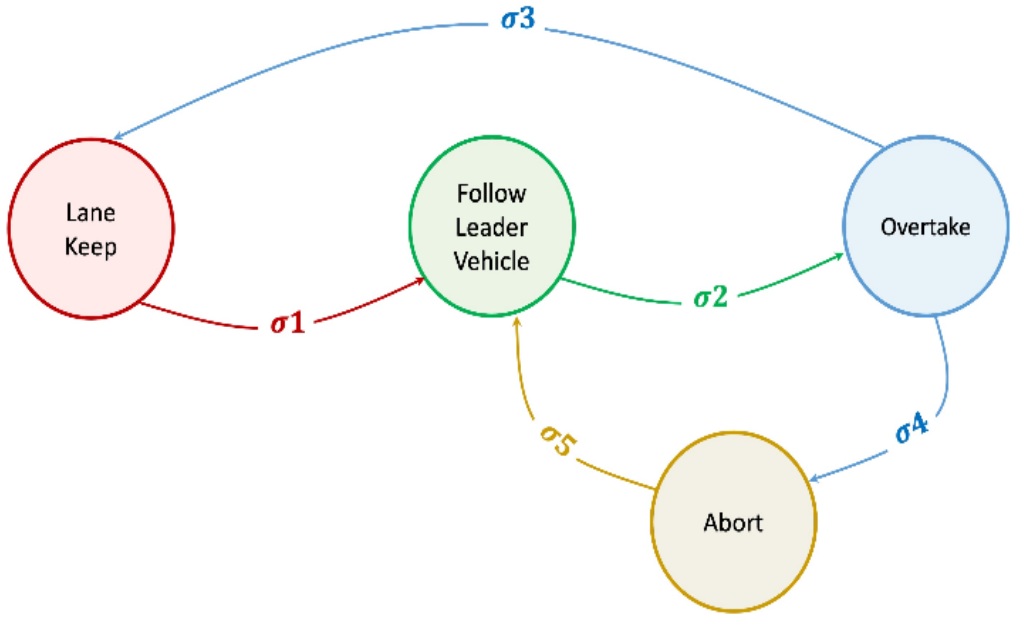


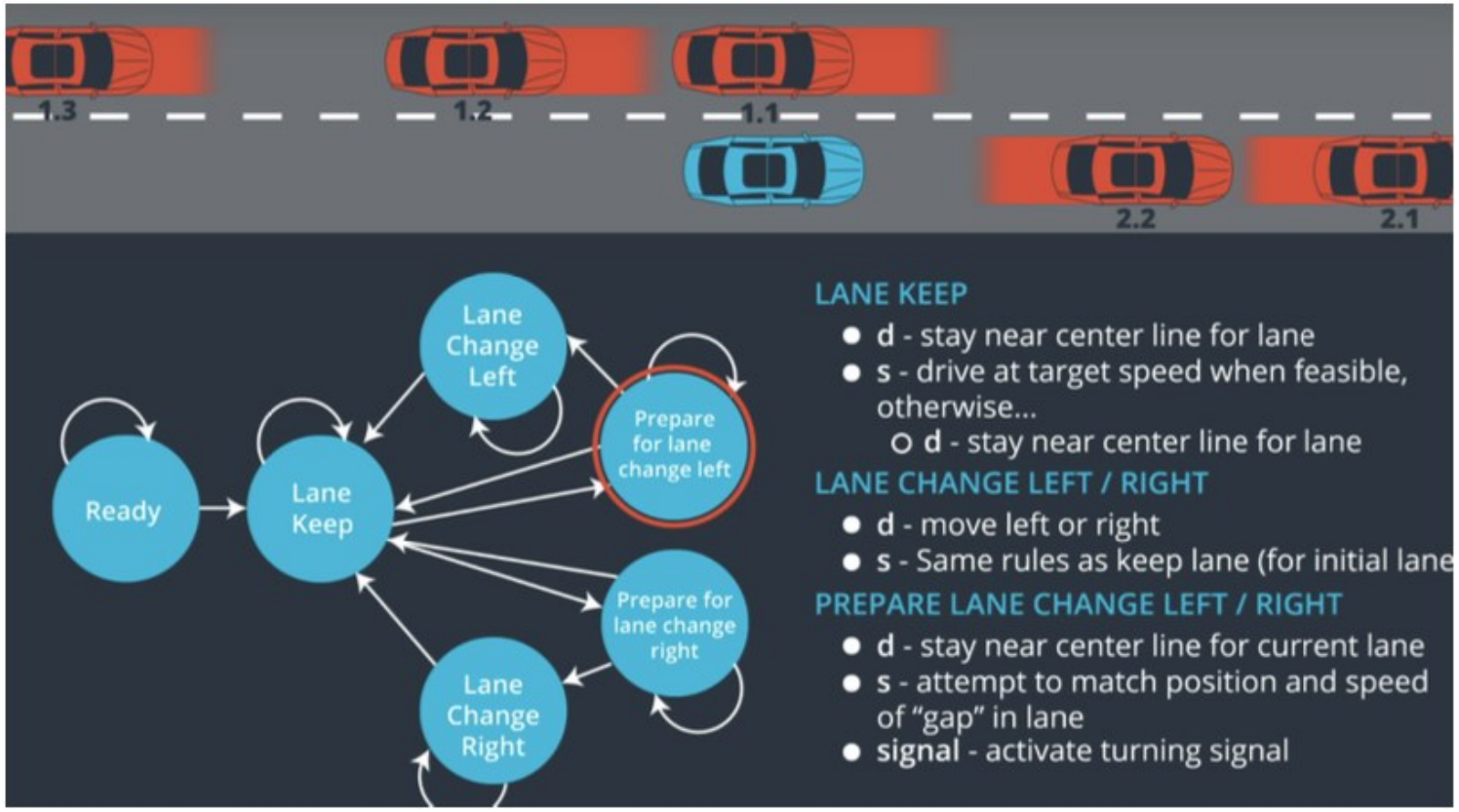
# Categorization of Decision-Making Approaches for Autonomous Vehicles



# Decision Making: Classical Approach

**Rule-based -> Finite State Machine**

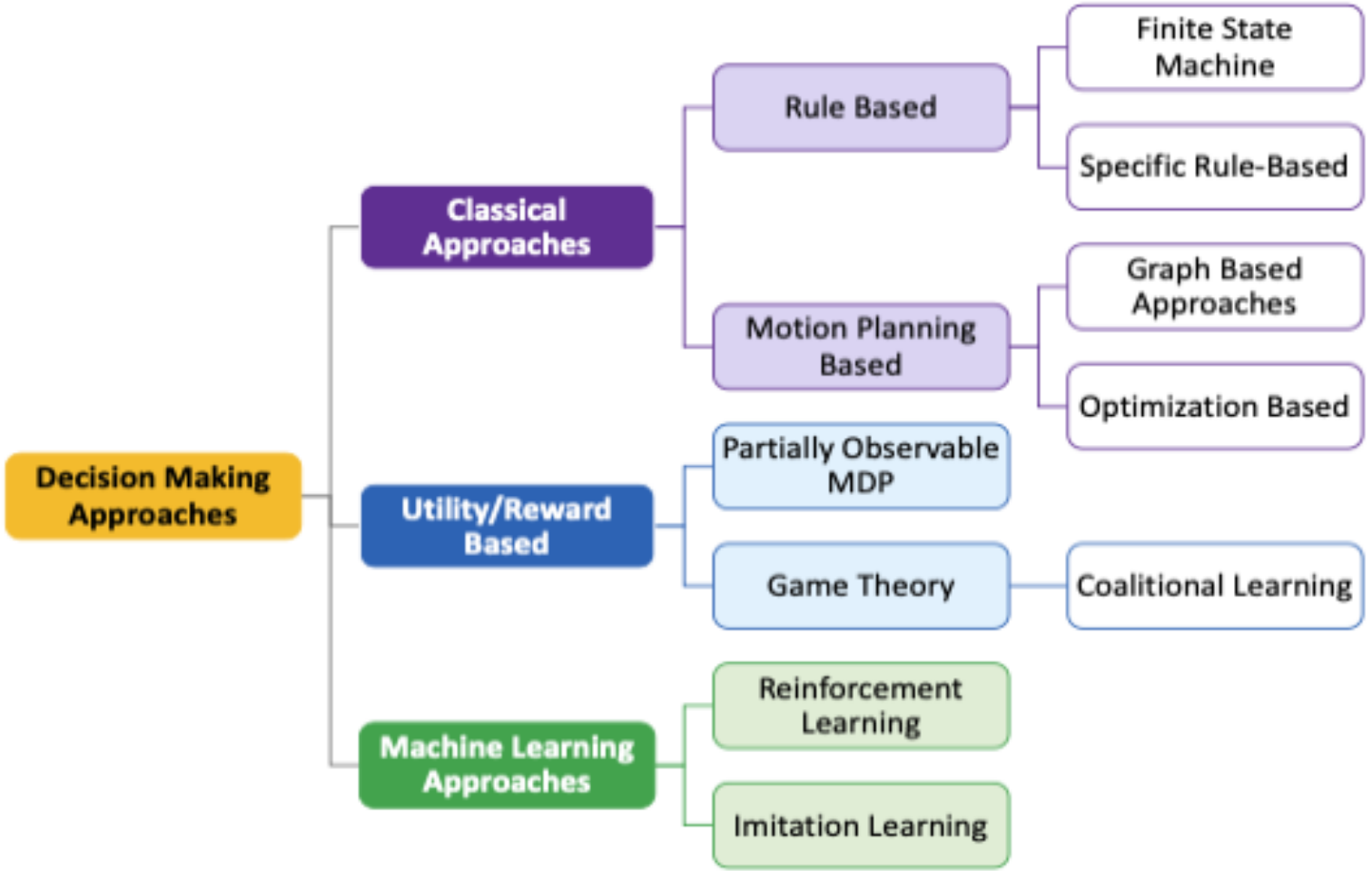



Finite State Machine Lane keeping example

Original example at: https://medium.com/@david010/autonomous-vehicle-path-planning-f8027a5f83f8

# Categorization of Decision-Making Approaches for Autonomous Vehicles

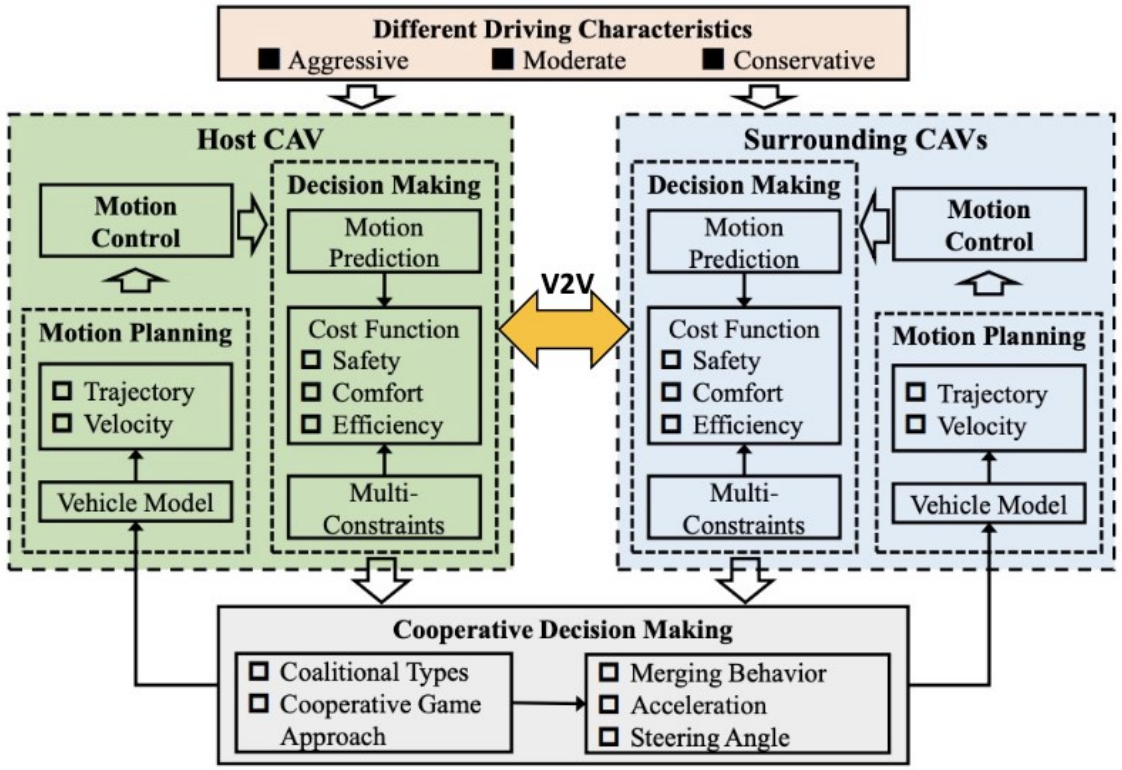


# Decision Making: Reward/Utility-based

**Game Theory -> Coalitional Learning**

Coalitional GT for autonomous driving provides very fitting characteristics for realizing the solutions of complex and urban driving, specifically for **short-term, highly dynamic, and L4/L5 platooning.**

The motion prediction module forecasts the motion states of the vehicles. The cost function for decision-making was created considering **certain constraints** including safety, comfort, and traffic efficiency based on the estimated motion states of the ego and surrounding vehicles.




Coalitional game theory based decision-making framework

# Decision Making: Reward/Utility-based

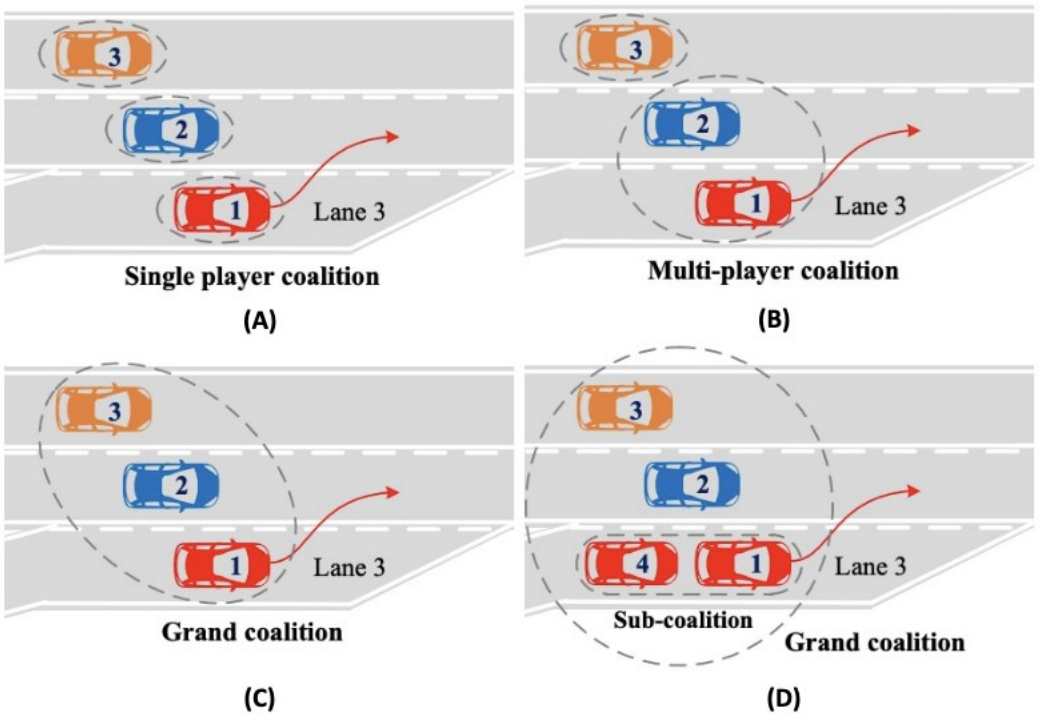
## Game Theory -> Coalitional Learning

***In highway settings, one of the causes of traffic congestion is vehicle merging.***

Several factors, including traffic dynamics, driver preferences, and travel aims, make it difficult for vehicles to perform merging moves.
Coalitional Learning deals with forming cooperative groups referred to as **coalitions**, allowing the cooperating players to strengthen their position in a game. The coalitional game is made up of two components:

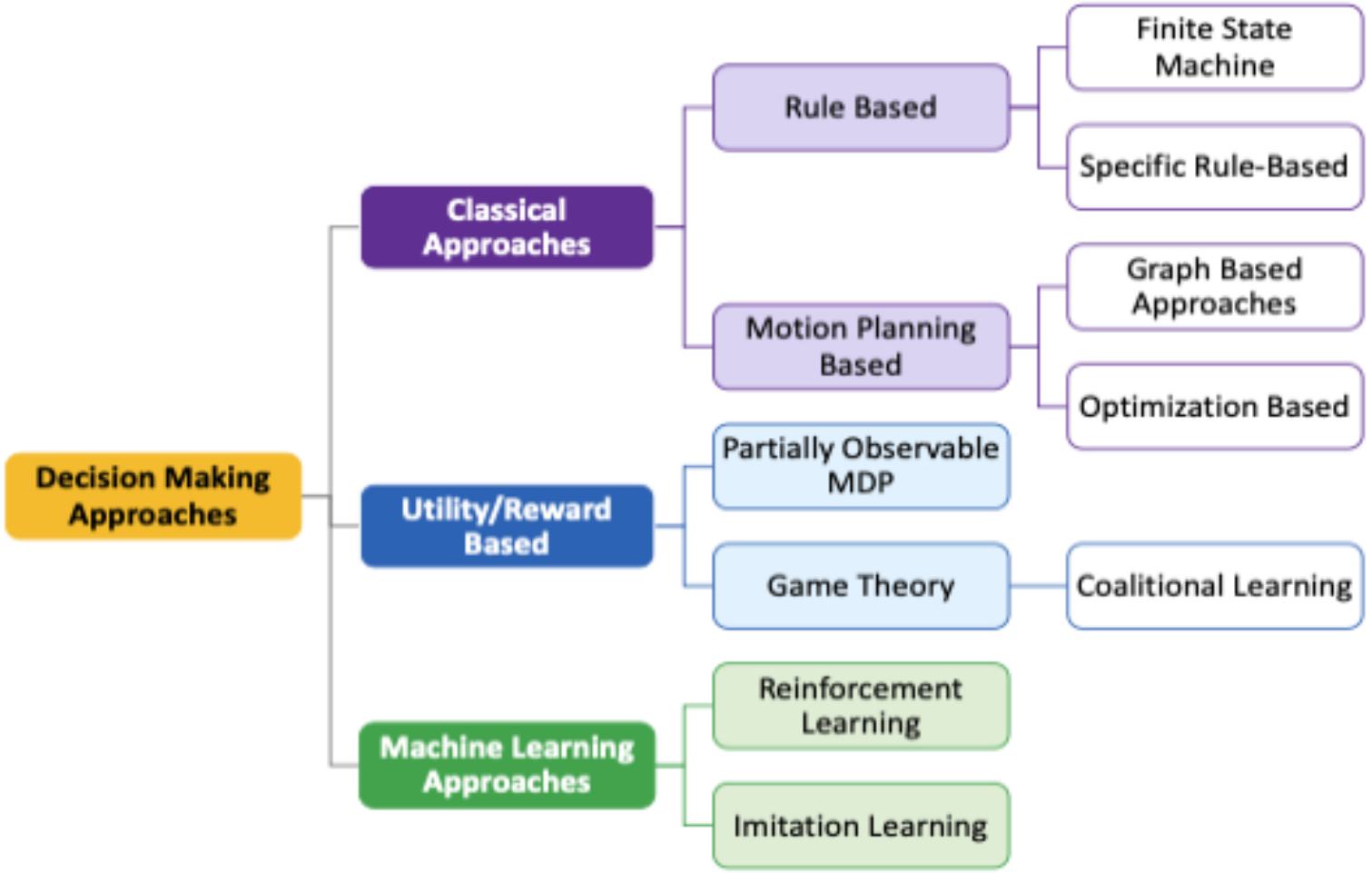
(i) The **set of players** which interact with each other to form cooperative coalitions and make agreements among them to act as a single entity in the given game

(ii) The **coalition value** which denotes the utility of the coalition in the game




Four types of coalitions
(A) The single player coalition (B) The multi-player coalition
(C) The grand coalition (D) The grand coalition with a sub-coalition

# Categorization of Decision-Making Approaches for Autonomous Vehicles



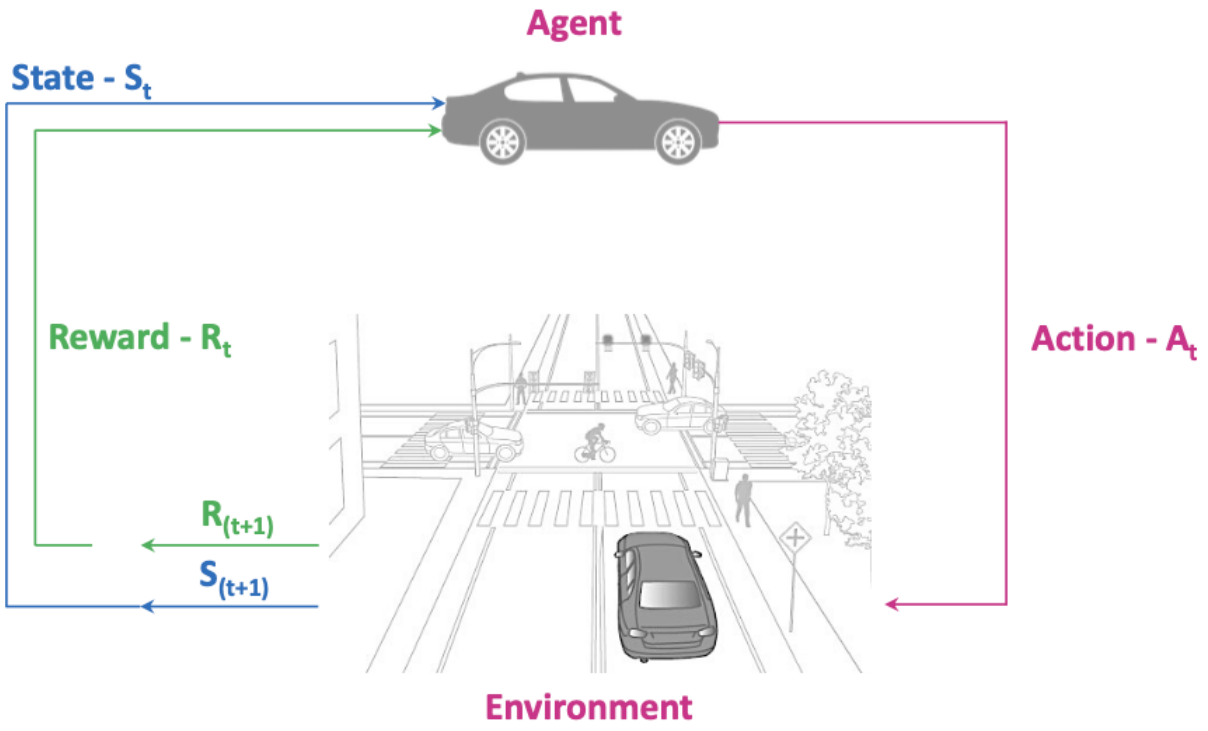
# Decision Making: Machine Learning

## Reinforcement Learning

Risk-aware driving decision approach which enables AVs to acquire an ideal driving strategy through **constant interaction with the environment.**

Some difficulties that come with RL:

- The policy is learned through trial-and-error, which relies on the experience gained through interactions with environments
- Poor stability and over-fitting in DRL methods
- Performance greatly depends on how the reward function is designed
- It is challenging to design the reward function for complex tasks



# End-to-End: Overview

---

# 5

# Components

# End-to-End Components

### 1. Data Collection:

- End-to-end self-driving systems require **large amounts of labeled sensor data** for training. This data includes raw sensor inputs such as images, lidar point clouds, and radar measurements, along with corresponding driving actions such as steering angles, throttle, and brake commands.

### 2. Neural Network Architecture:

- The **core component** of the end-to-end approach is a deep neural network that learns to map raw sensor data to driving actions. This network typically consists of convolutional layers for processing image data, recurrent layers for temporal dependencies, and fully connected layers for decision-making. Some architectures may also incorporate attention mechanisms or memory modules to improve performance.

### 3. Training Process:

- During training, the neural network learns to **predict driving actions directly from raw sensor data**. The network is trained using **supervised learning** techniques, where it minimizes a loss function that measures the discrepancy between predicted and ground-truth driving actions. Training is typically performed on large-scale datasets collected in diverse driving conditions to ensure robustness and generalization.

### 4. End-to-End Driving Policy:

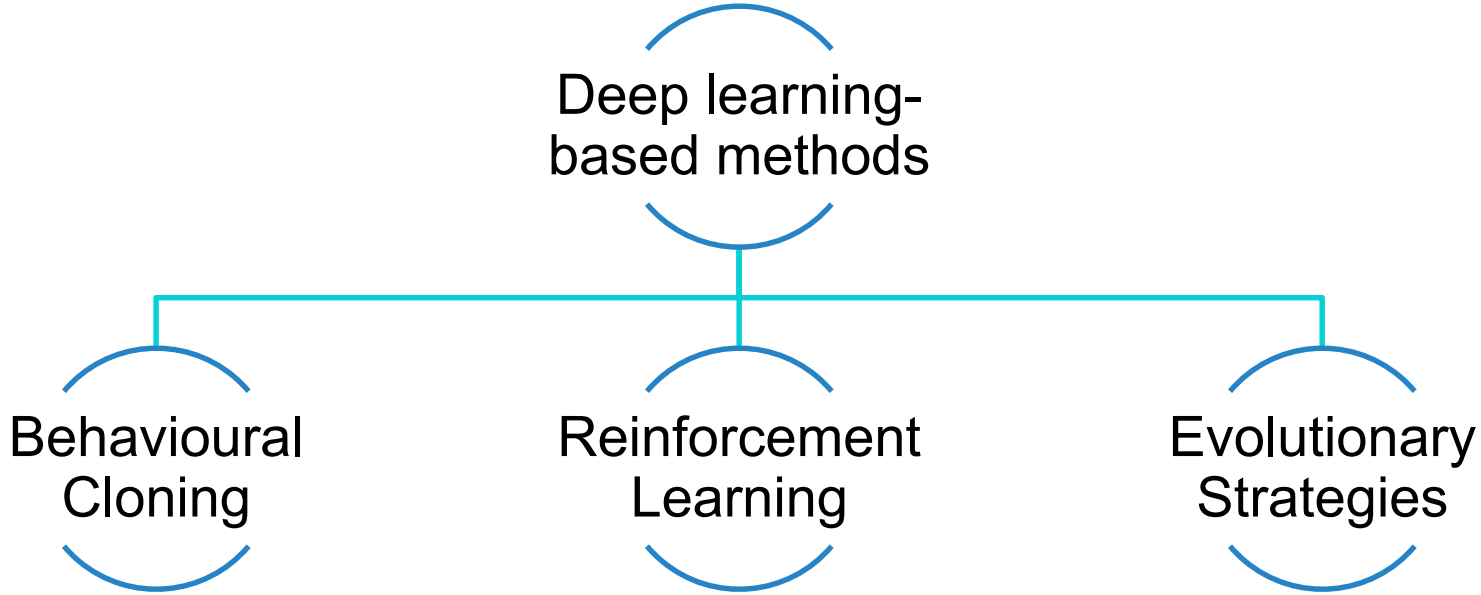
- The trained neural network serves as an end-to-end driving policy that takes raw sensor inputs as input and **outputs driving actions in real-time**. This policy **encapsulates the entire decision-making process**, from perception and understanding of the environment to planning and execution of driving maneuvers.

### 5. Deployment and Testing:

- Once trained, the end-to-end driving policy can be deployed on a self-driving car for real-world testing and validation. **Extensive testing** is conducted to evaluate the system's performance across a **wide range of driving scenarios**, including variations in weather, lighting, traffic, and road conditions.

# End-to-End: Neural Networks

In contrast to the modular stack approach, which decomposes the driving task into separate modules, the end-to-end approach aims to **learn a single integrated model that directly maps sensor inputs to driving actions.**



# End of Session 3