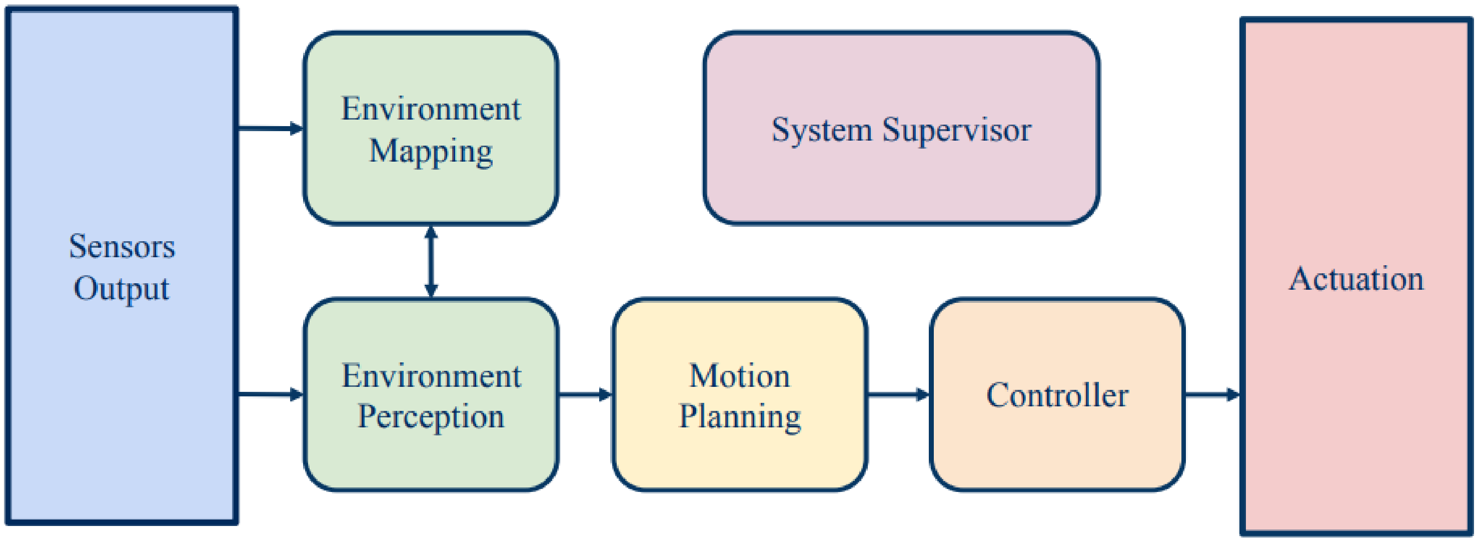
Lesson 3

Software Architecture

Software Architecture - High-level:



The car **observes the environment around it**, **using a variety of sensors**. The **raw sensor measurements** are **passed** into **two sets of modules** dedicated to understanding the environment around the car.
Those are,

1. **Environment Perception**
2. Environment Mapping

The **environment perception modules** have **two key responsibilities**,

1. **Identifying the current location** of the autonomous vehicle in space.
2. **Classifying and locating important elements of the environment for the driving task**.

Examples of these elements include **other cars, bikes, pedestrians, the road, road markings, and road signs, anything that directly affects the act of driving.**

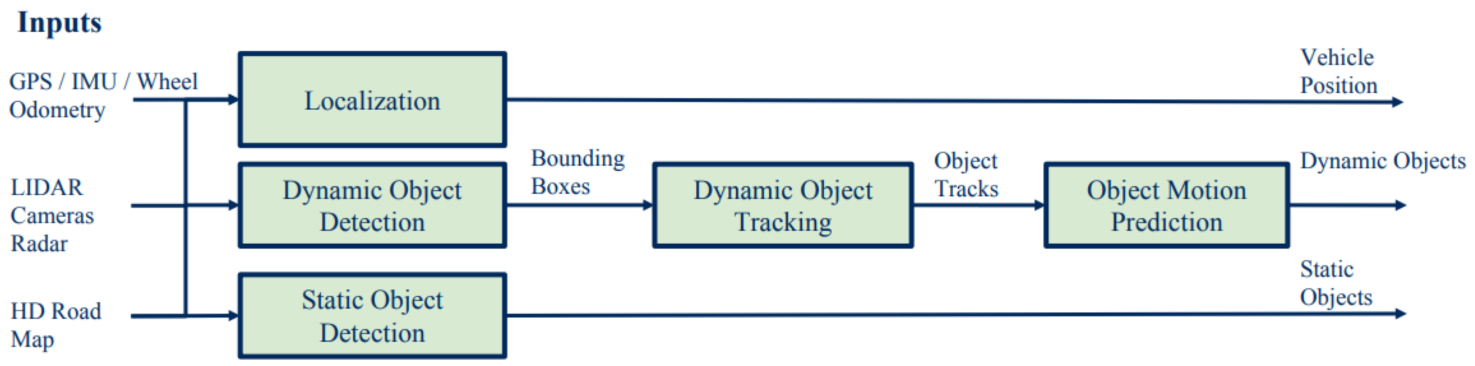
The **environment mapping** module **creates a set of maps which locate objects in the environment** around the autonomous vehicle for a range of different uses, From **collision avoidance to egomotion tracking and motion planning.**

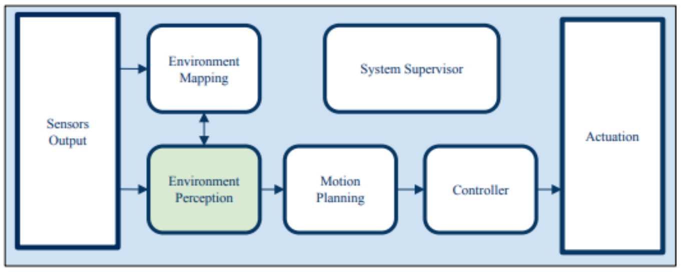
The **motion planning** module **makes all the decisions about what actions to take and where to drive based on all of the information provided** by **the perception** and **mapping modules.**
The motion planning module's main **output is a safe, efficient** and **comfortable planned path** that **moves the vehicle towards its goal.**

The **planned path is then executed** by **the controller**. **The controller module takes the path** and **decides on the best steering angle**, **throttle position**, **brake pedal position**, and **gear settings** to precisely follow the planned path.

The **system supervisor monitors** all parts of the software stack, as well as the **hardware output**, to make sure that **all systems are working as intended. The system supervisor** is also responsible for informing the **safety driver of any problems found in the system.**

Software Architecture - Environment Perception:





There are **two important parts of the perception stack**, **localizing the ego-vehicle in space** and **classifying and locating the important elements of the environment.**

Localization:
The **localization module** takes in **multiple streams of information**, such as the current **GPS location, IMU measurements** and **wheel odometry. It then combines them to output an accurate vehicle location.**

For greater accuracy, some **localization modules** also **incorporate LIDAR and camera data.**

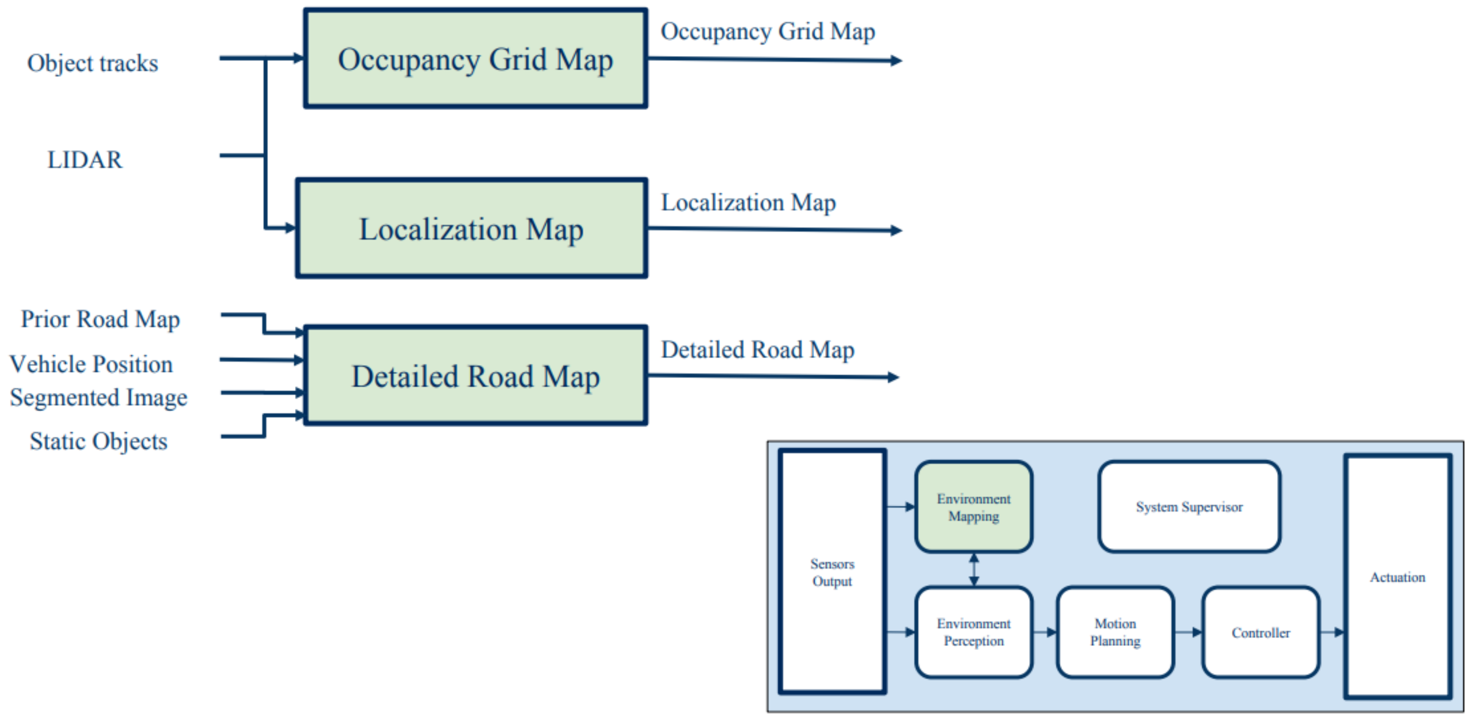
Typically, **The problem of classification** and **localization of the environmental elements is divided into two segments**.
1. Detecting dynamic objects in the environment
2. Detecting the static objects in the environment.

Dynamic Object Detection:
The **dynamic object detection module** uses a **set of camera inputs** as well as **LIDAR point clouds** to **create 3D bounding boxes around dynamic objects in the scene.** The **3D bounding boxes encode the class** or **type of object** as well as the **exact position, orientation and size of the object**. Once the object is detected, **the dynamic objects are tracked over time by a tracking module.**

The tracker module provides not only the **current position of the dynamic objects** but also **the history of its path through the environment. The history of the path is used along with the road maps In order to predict the future path of all dynamic objects.** This is usually handled by a prediction module, which combines all information regarding the dynamic object and the current environment to predict the path of all dynamic objects.

Static Object Detection:
The **static object detection** module also relies on a **combination of camera input and LIDAR data** to identify **significant static objects in the scene**. Such **important data include the current lane** in which the self-driving vehicle is found, and **the location of regulatory elements such as signs** and **traffic lights.**

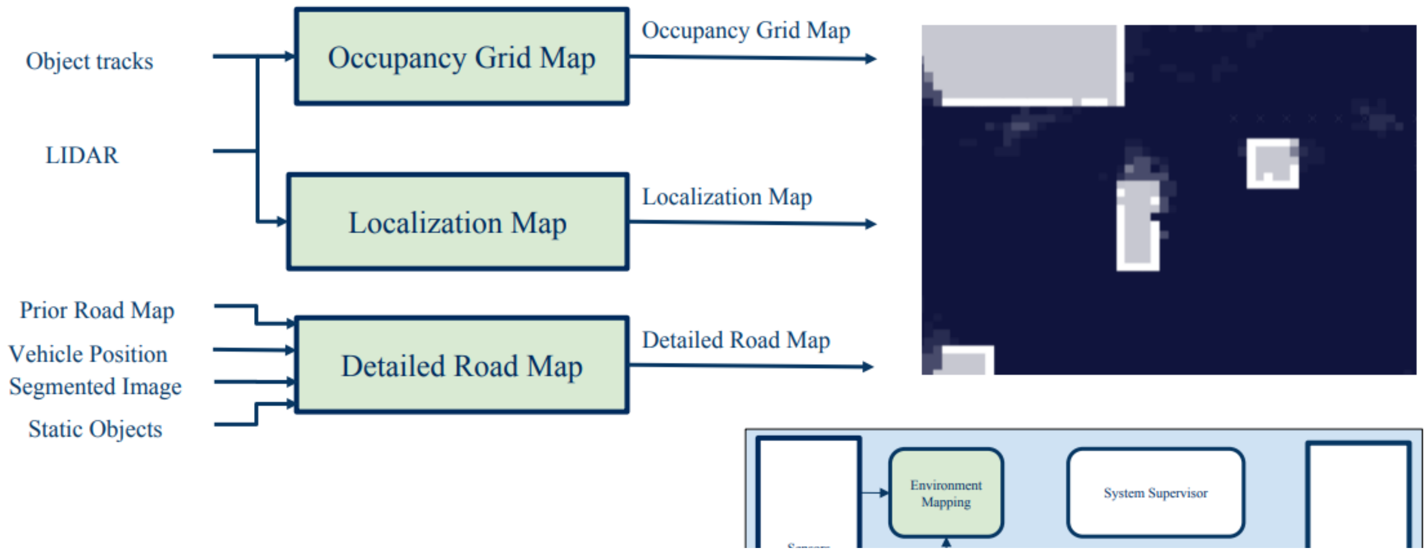
Software Architecture - Environmental Maps:

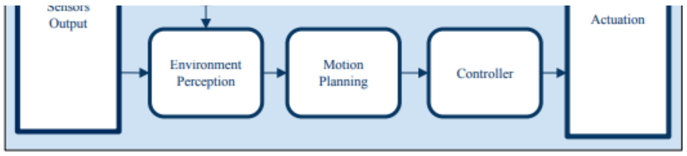


Environment maps create several different types of representation of the current environment around the autonomous car. There are three types of maps,

1. Occupancy grid map
2. Localization map
3. Detailed road map

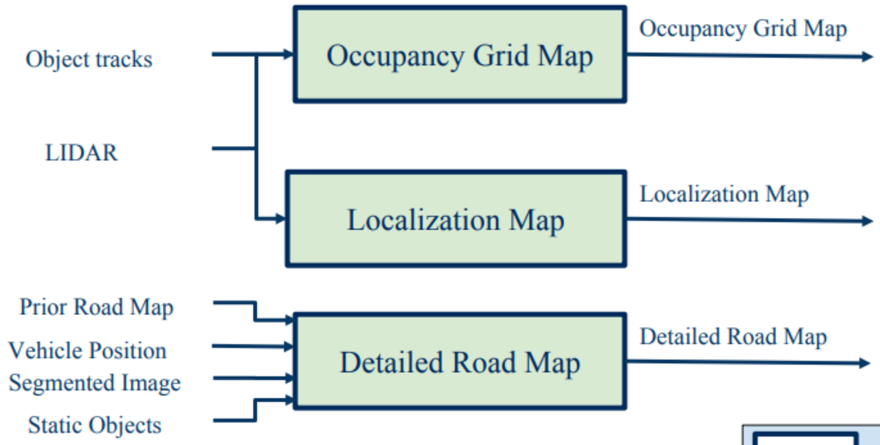
Occupancy Grid Map:

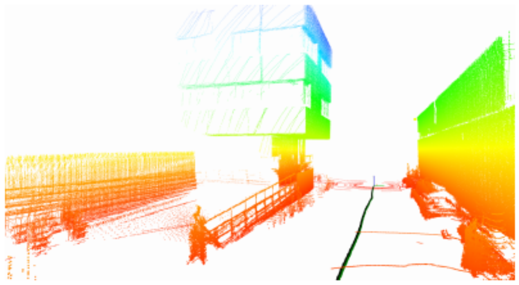


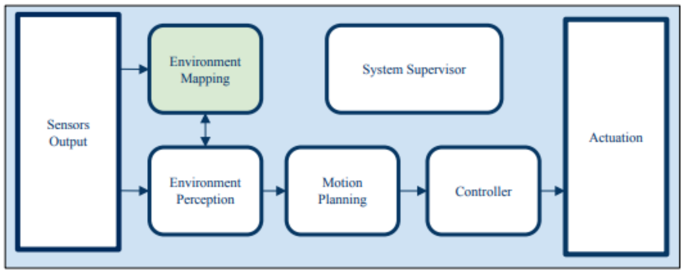


The occupancy grid is a map of **all static objects in the environment surrounding the vehicle. LIDAR** is predominantly used to **construct the occupancy grid map.**
A set of **filters are first applied to the LIDAR data** to make it usable by the occupancy grid. For example, the drivable surface points and dynamic object points are removed. The occupancy grid map represents the environment as a set of grid cells and associates a probability that each cell is occupied. This allows us to handle uncertainty in the measurement data and improve the map over time.

Localization Map:

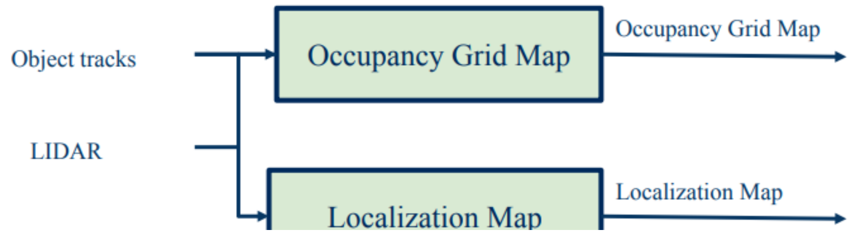


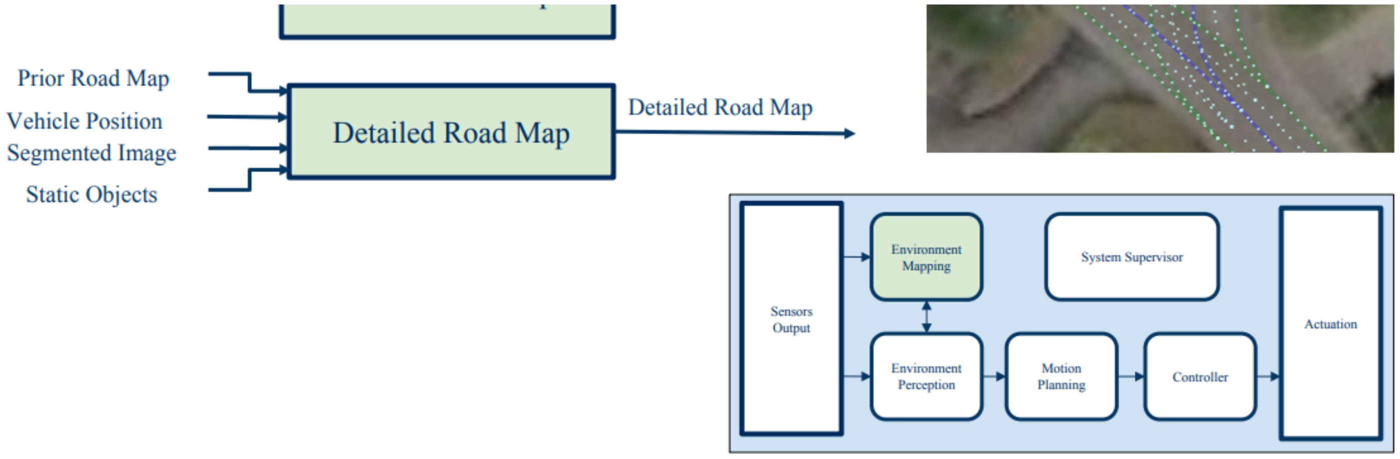




Which is **constructed from LIDAR**, or **Camera data** is used by the **localization module** in order to **improve Eco state estimation.** Sensor data is compared to this map while driving to determine the motion of the car relative to the localization map. **This motion is then combined with other proprioceptor sensor information** to accurately localize the Eco vehicle.

Detailed Road Map:



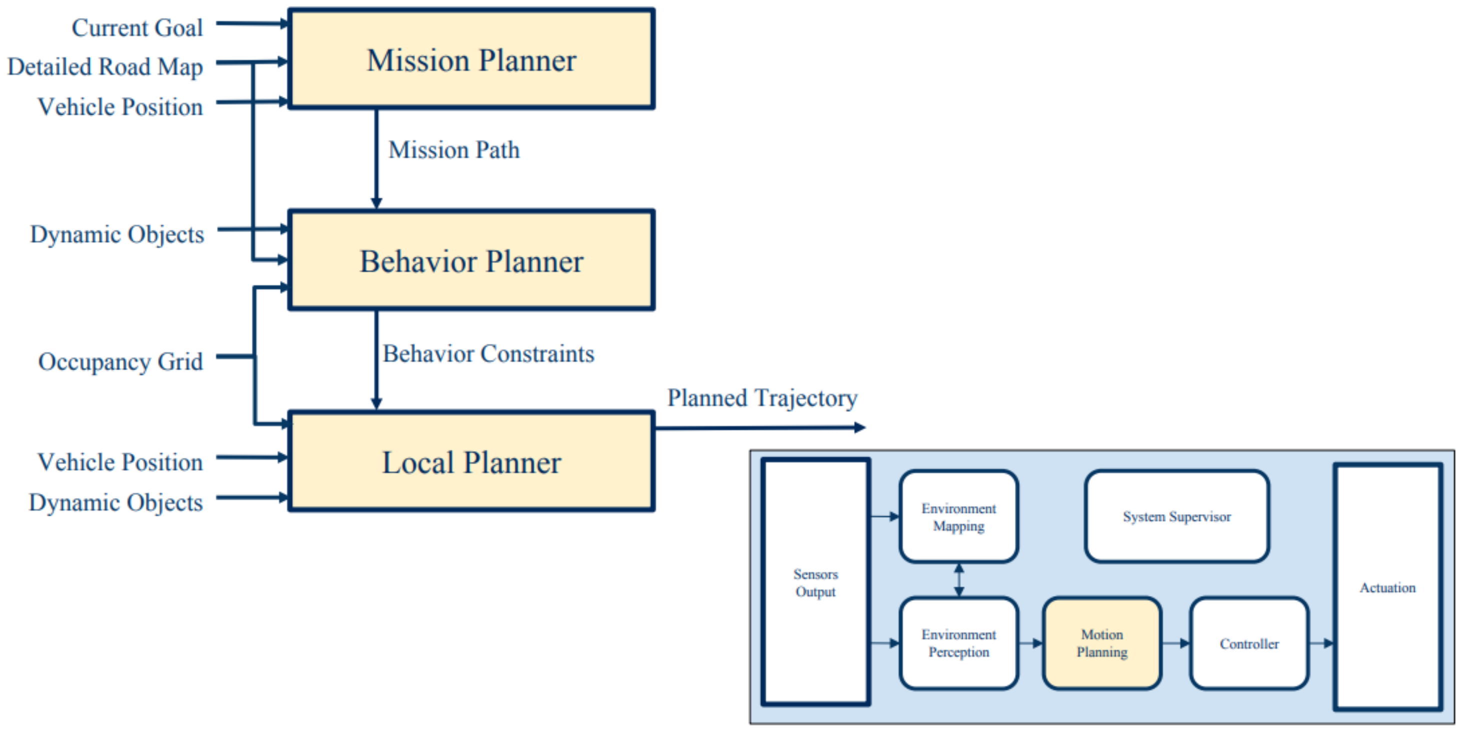


Detailed road map is provides a **map of road segments which represent the driving environment** that the autonomous vehicle is currently driving through. It captures **signs and lane markings** in a manner **that can be used for motion planning.** This map is traditionally a combination of prerecorded data as well as incoming information from the **current static environment gathered by the perception stack.**

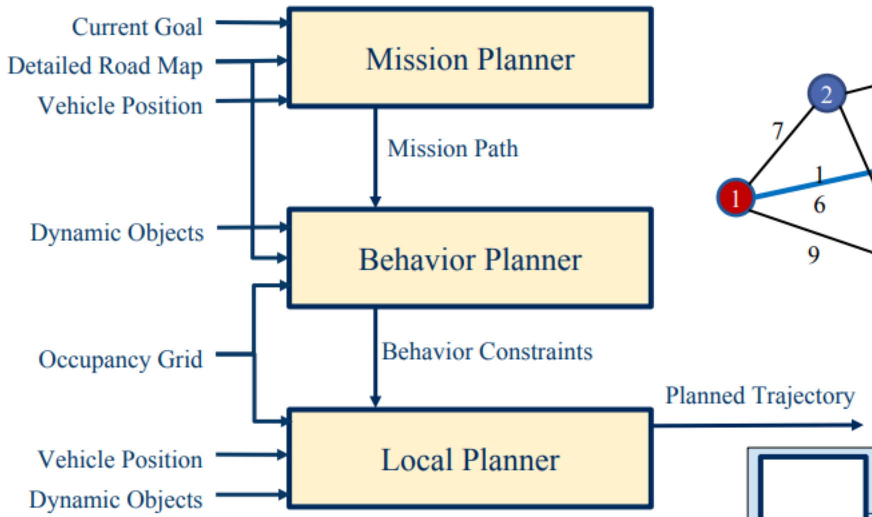
The **environment mapping** and **perception modules interact significantly to improve the performance of both modules.** For example, **The perception module provides the static environment information** needed to update the **detailed road map,** which is then **used by the prediction module to create more accurate dynamic object predictions.**

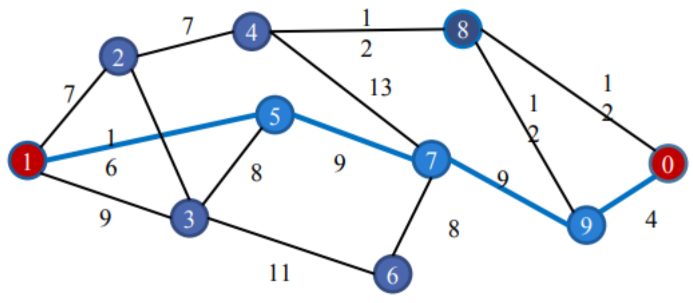
The output from both the **environment maps** and **the perception modules** are combined and used by the **motion planning module to create a path through the environment.**

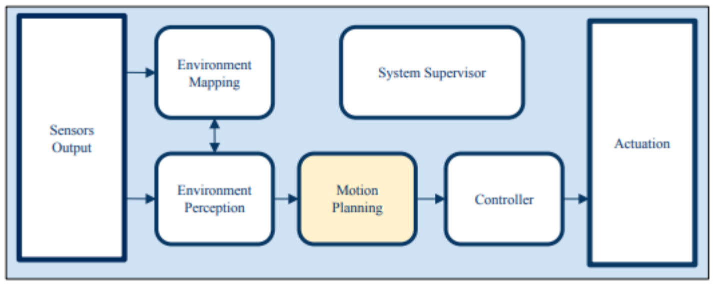
Software Architecture - Motion Planning:



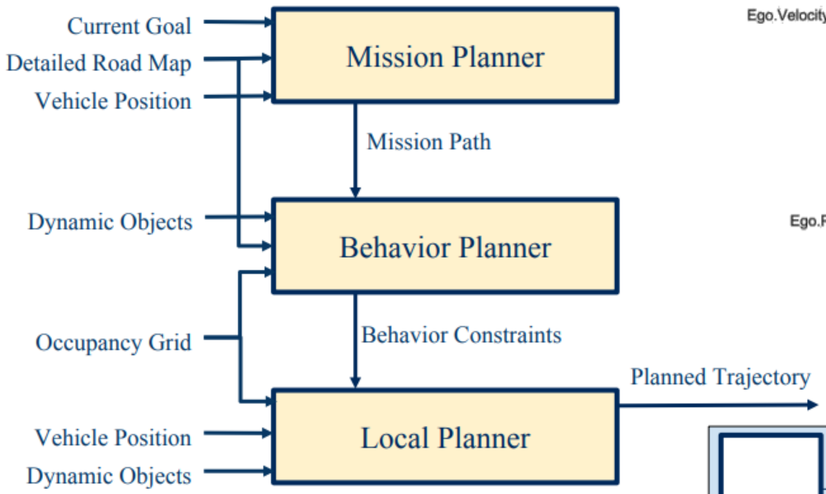
Most self-driving cars today use a **decomposition that divides the problem into several layers of abstraction,**

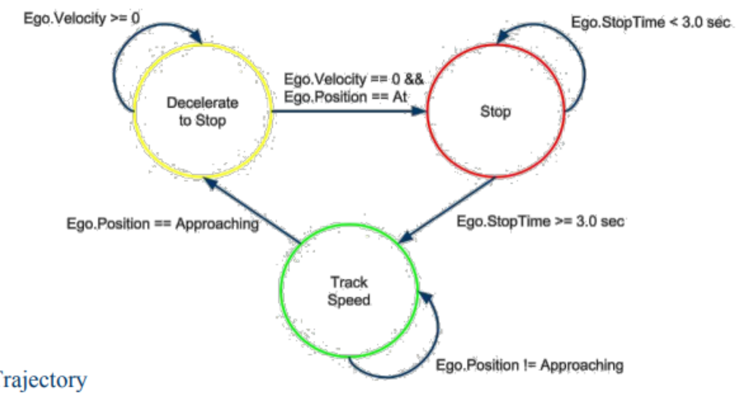


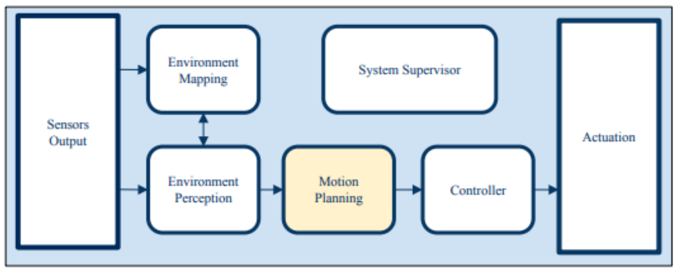




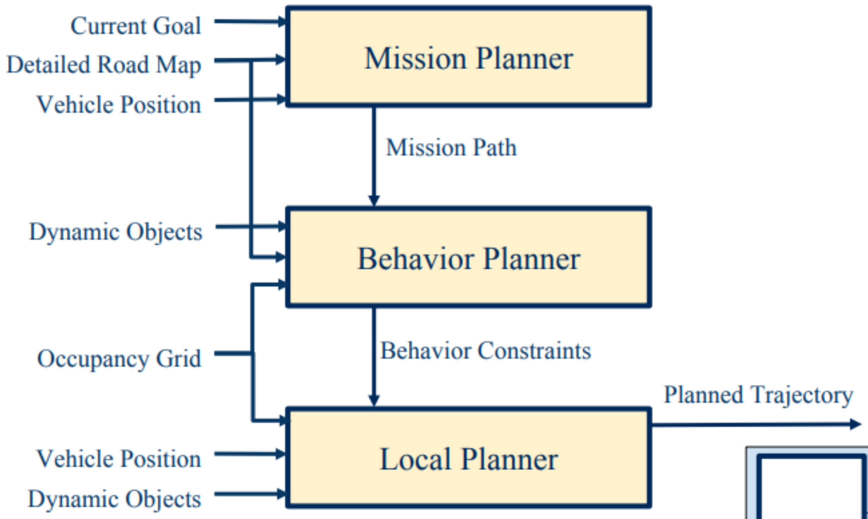
At the top level, the **mission planner handles long term planning** and **defines the mission over the entire horizon of the driving task**, from the current location, through the road network to its final destination. To find the complete mission, **the mission planner determines the optimal sequence of road segments that connect your origin and destination**, and then **passes this to the next layer.**

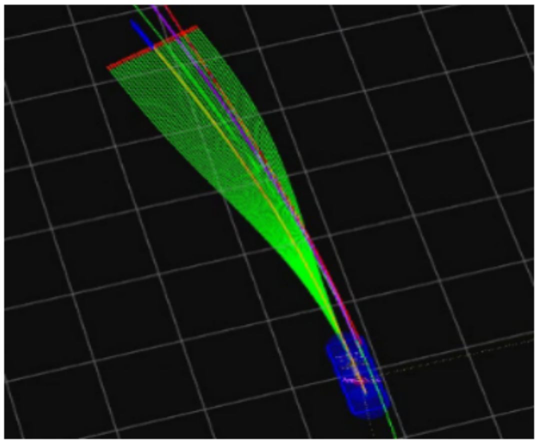


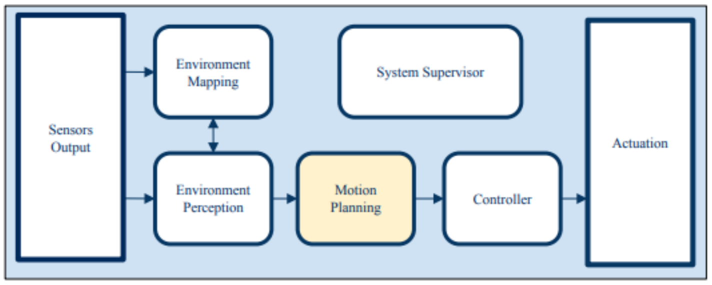




The **behavior planner** is the next level of abstraction, **solving short term planning problems.** The behaviour planner is **responsible for establishing a set of safe actions or maneuvers to be executed while travelling along the mission path**. An example of the behaviour planner decisions would be **whether the vehicle should merge into an adjacent lane given the desired speed and the predicted behaviors of nearby vehicles**. Along with the maneuver of decisions, the behavior planner also provides a set of constrains to execute with each action, such as how long to remain in the current lane before switching.

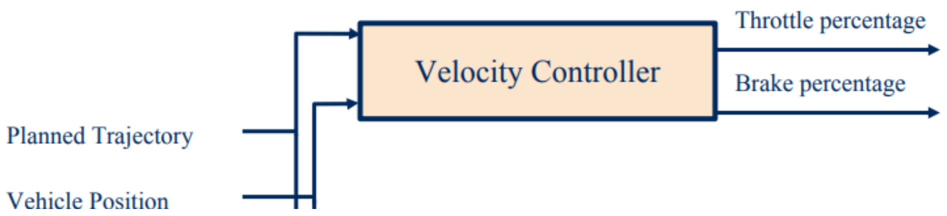




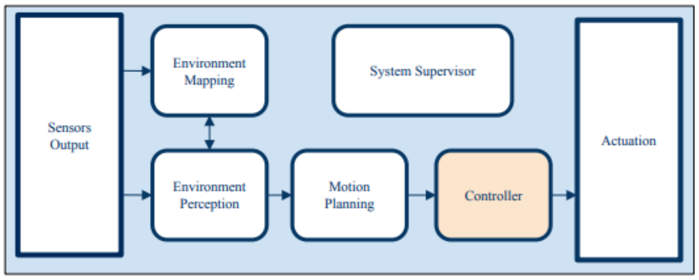


The **local planner** performs immediate or reactive planning, and is responsible for **defining a specific path and velocity profile to drive.** The local plan must be smooth, safe, and efficient given all the current constraints imposed by the environment and maneuver. In order to create such a plan, **the local planner combines information provided by the behavior planner, occupancy grid, the vehicle operating limits, and other dynamic objects in the environment.** The output of the local planner is a planned trajectory which is a combined **desired path and velocity profile for a short period of time into the future.**

Vehicle controller:



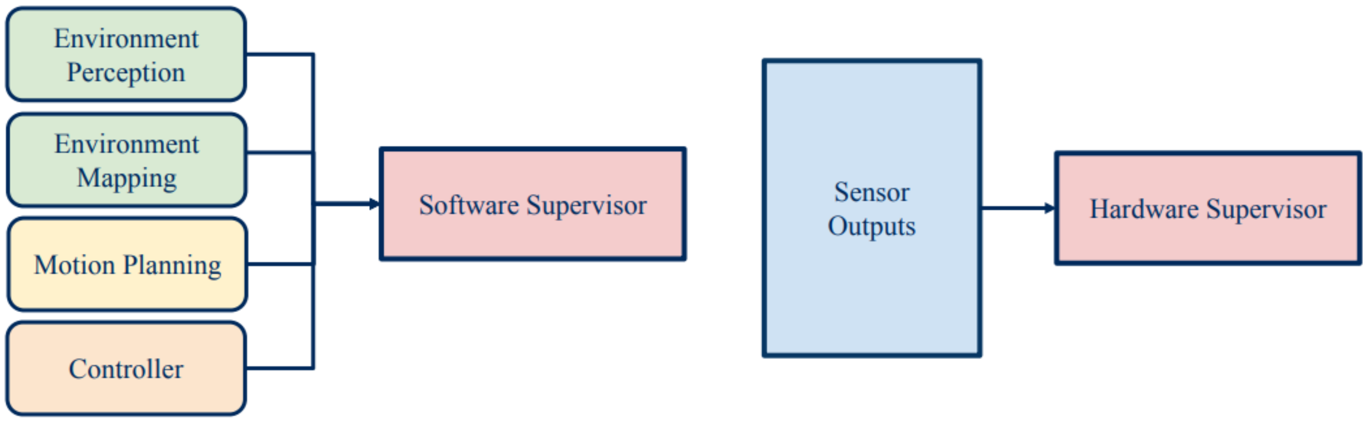
Steering Controller
Steering Angle



Vehicle controller takes the given trajectory and turns it into a set of precise actuation commands for the vehicle to apply. A typical **controller separates the control problem into a longitudinal control and a lateral control.**

The **lateral controller outputs** the steering angle required to maintain the planned trajectory, whereas the **longitudinal controller** regulates the **throttle, gears and braking system to achieve the correct velocity.** Both controllers calculate **current errors and tracking performance of the local plan, and adjust the current actuation commands to minimize the errors going forward.**

System Supervisor:


Environment Perception
Environment Mapping
Motion Planning
Controller
Software Supervisor
Sensor Outputs
Hardware Supervisor