Apollo Conceptual Architecture

Video URL: https://youtu.be/mbfVes7JVNc

Apollogizers

Group 25

Lavi Loans - Group Leader, Use Case

Xinyu Chen - Intro, Control and Data Flow

John Scott - Overview, Evolution

Baorong Wei - Conclusion

Zhihan Hu - Implications of responsibility division

Anthony Galassi - Abstract

ABSTRACT

- Baidu’s Apollo Auto has undergone extensive changes throughout its development of it’s open-source autonomous driving project
- We found tasks in control flow are executed linearly, hence a process-oriented architecture at the control level
- Three modules did not have concurrency, which were the perception, prediction, and planning modules
- The three notable use cases for Apollo’s software are: RoboTaxi lane change, Minibus bus stop, and valet parking

Image source: Apollo.Auto

Introduction

- What is Apollo Auto?
- What is conceptual architecture?
- Why are we documenting?

Overview

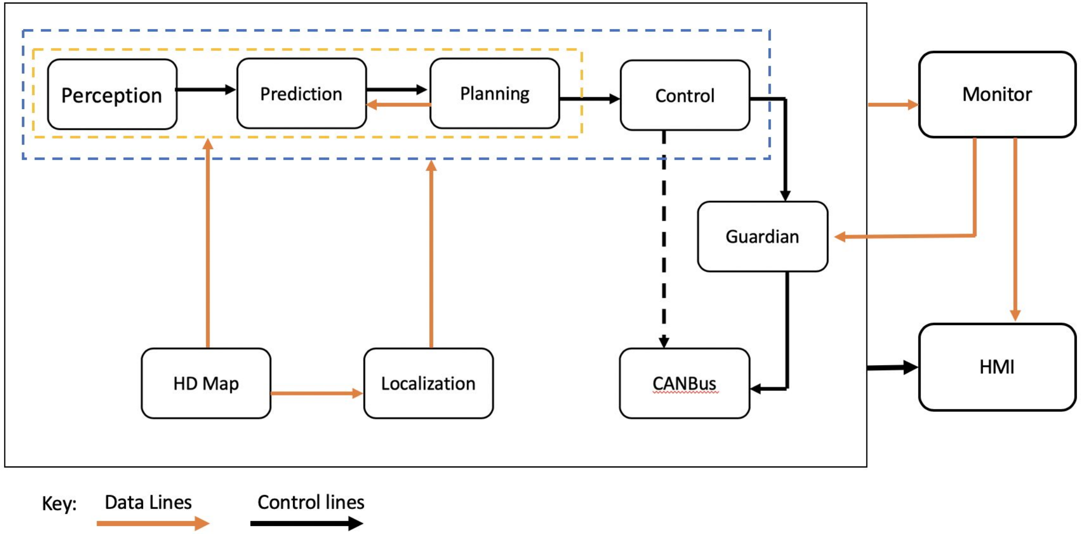
- Component parts and interaction
- Evolution
- Control and Data Flow
- Concurrency
- Developer responsibility allocation
- Sequence Diagrams

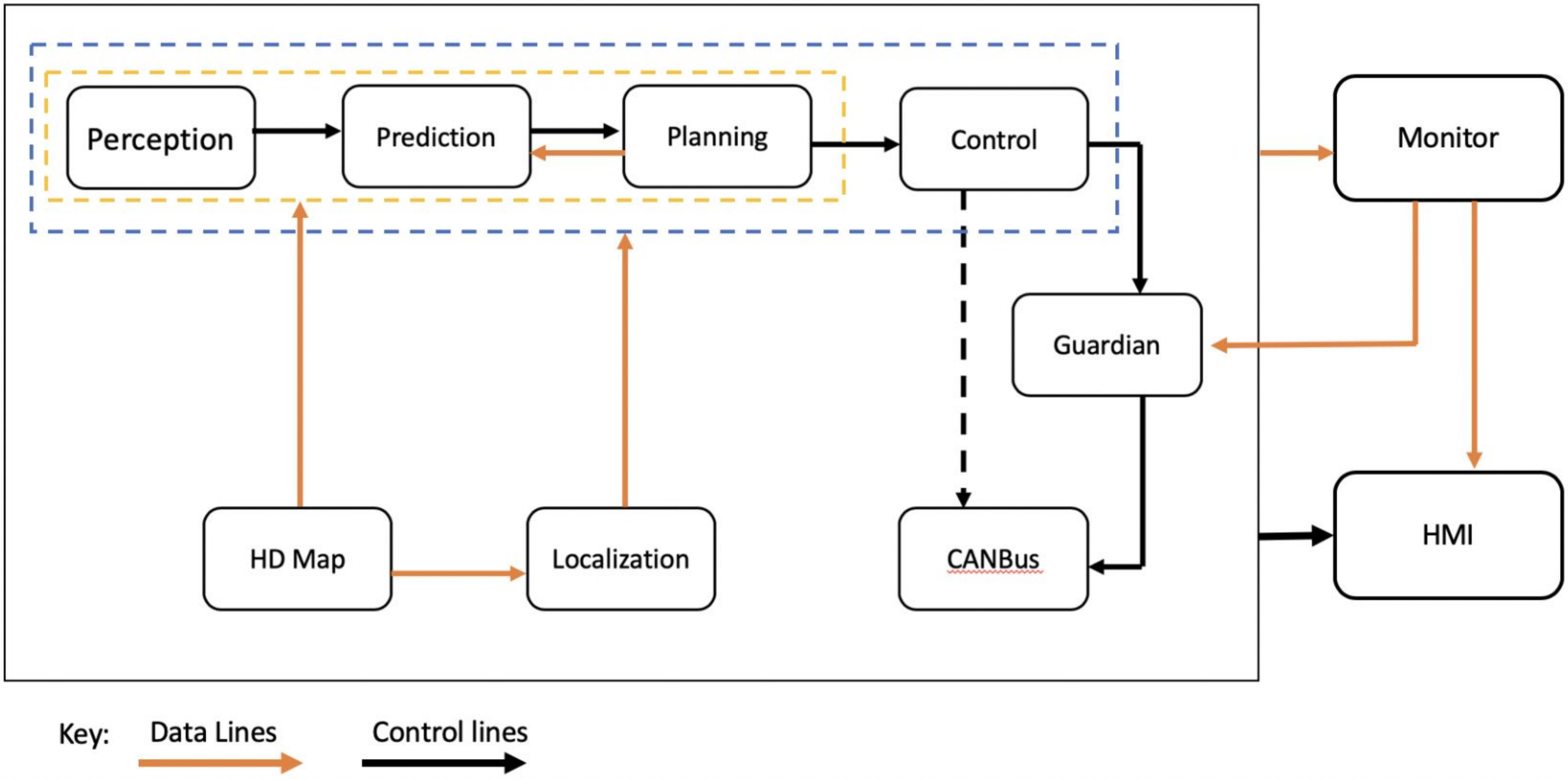
Components Parts and Interaction

- Perception module uses LiDAR and RADAR data to detect obstacles and predicts their properties from a neural net
- Prediction module takes images from the perception module and labels things from the deep neural net, such as traffic lights, lane/flow of traffic and objects in the real surroundings
- Guardian module is a safety module that is the action centre for the vehicle and intervenes if necessary
- Localization module provides localization services via the RTK (Real Time Kinematic) based method or the multi-sensor fusion method
- Control module is intelligent in that it can handle different road conditions, speeds, vehicle types and canbus protocols with a high control accuracy

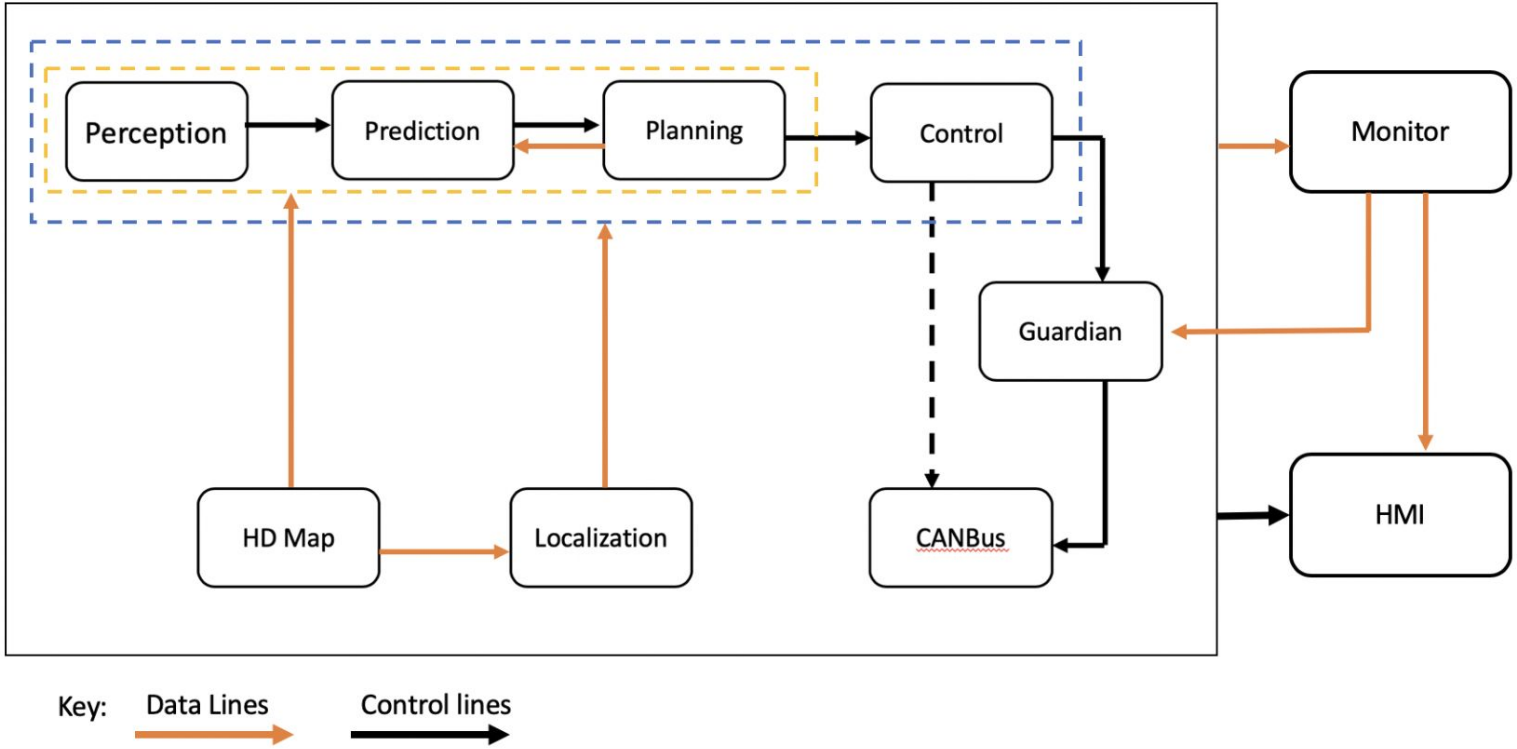
Components Parts and Interaction

- Perception module and HD-map module feed into the prediction module
- Prediction and planning modules interchange data and control lines to detect and react to various things
- All data is interpreted, then passed to the control module. It outputs to the guardian module, which will protect the vehicle and passenger in case of system failures

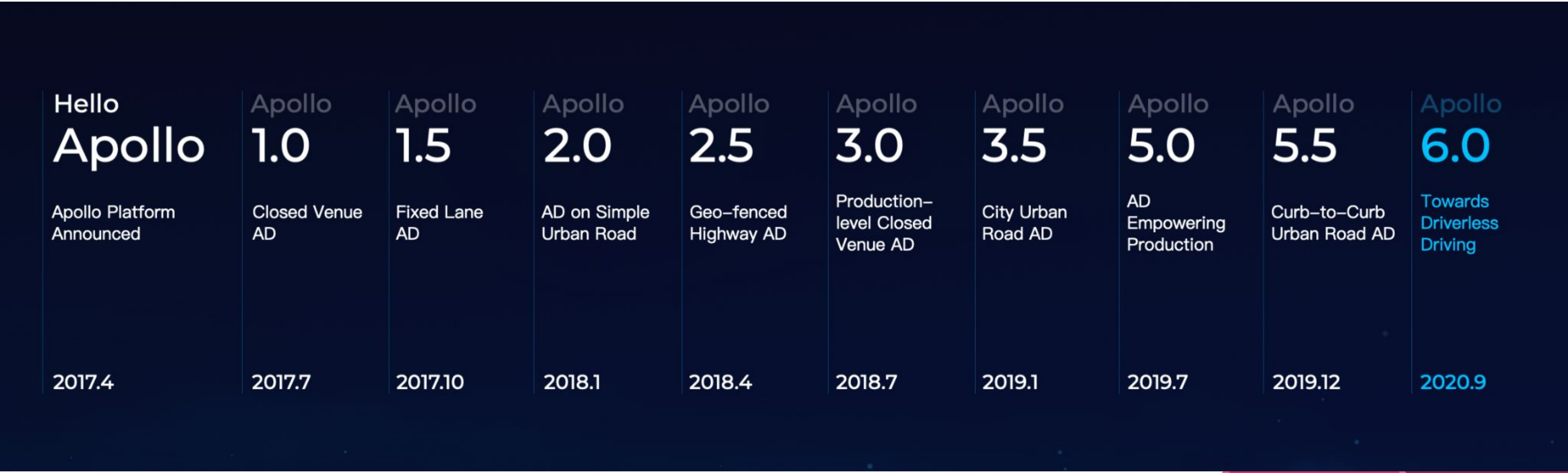



Perception
Prediction
Planning
Control
Monitor
Guardian
HD Map
Localization
CANBus
HMI
Key:
Data Lines
Control lines

Concurrency of System Architecture



Evolution



Hello Apollo	Apollo Platform Announced	2017.4
Apollo 1.0	Closed Venue AD	2017.7
Apollo 1.5	Fixed Lane AD	2017.10
Apollo 2.0	AD on Simple Urban Road	2018.1
Apollo 2.5	Geo–fenced Highway AD	2018.4
Apollo 3.0	Production–level Closed Venue AD	2018.7
Apollo 3.5	City Urban Road AD	2019.1
Apollo 5.0	AD Empowering Production	2019.7
Apollo 5.5	Curb–to–Curb Urban Road AD	2019.12
Apollo 6.0	Towards Driverless Driving	2020.9

Implications of Responsibility Division of Developers

- Based on the Apollo 7.0 module, there are four main sections:
  - Cloud Service Platform
  - Open Software Platform
  - Hardware Dev Platform
  - Open Vehicle Certification Platform
- Apollo Studio **(NEW)**
- A variety of development components
- Responsibility could be divided based on the general categories

Implications of Responsibility Division of Developers

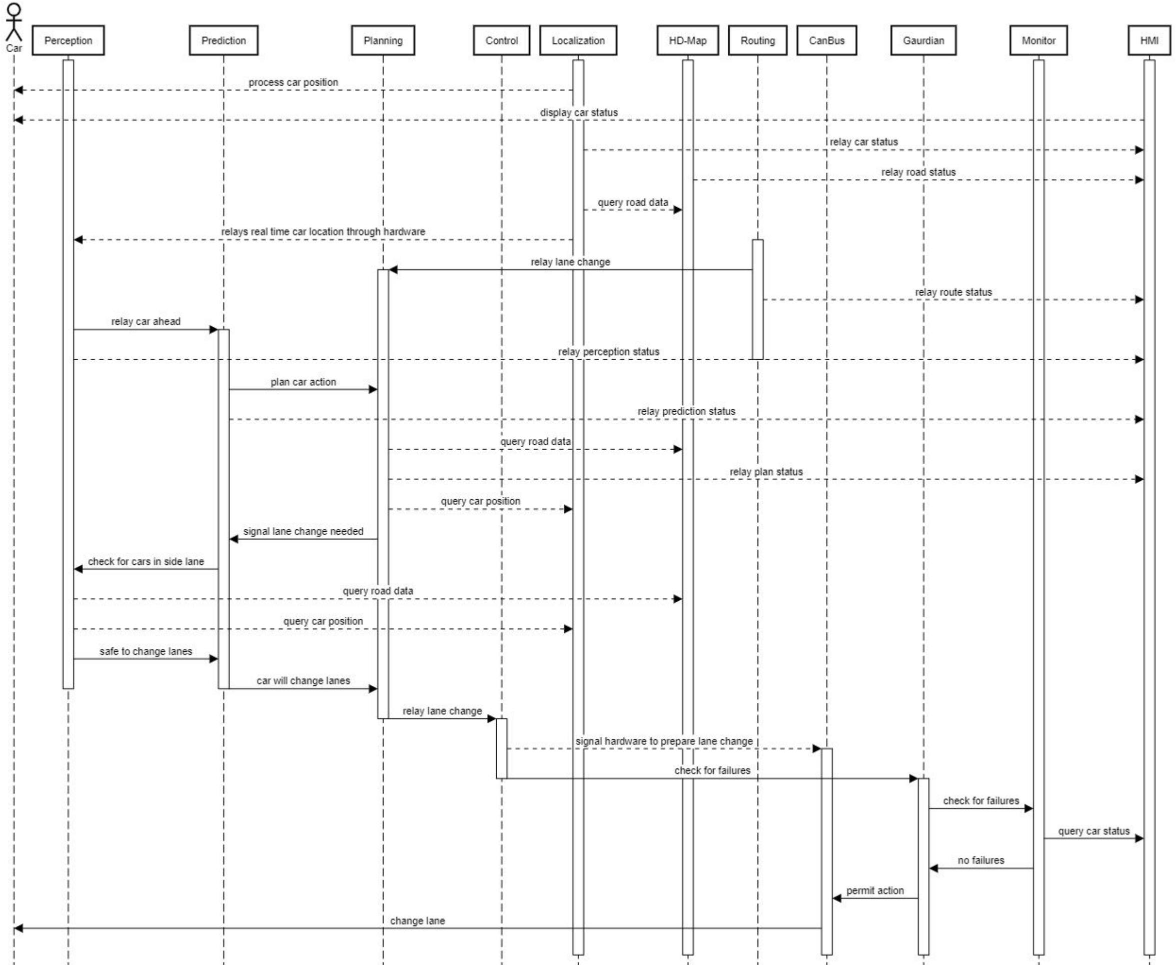
- Cloud Computing
  - analyzing the basic infrastructure of a business model, and researching how to transfer different functionalities
  - include Cloud Solution Architect, Cloud Developer, Cloud DevOps Engineer, Cloud Data Engineer, and Cloud Operations Engineer
- Automated Driving Development
  - Developing and testing algorithms for sensor perception
  - Behavioral planning, motion planning, and motion control
- Front-end
  - Provide Apollo users with an intuitive and efficient machine interface design
  - Responsible for different aspects of HMI development, most notably including hardware, software, and infrastructure.

Implications of Responsibility Division of Developers

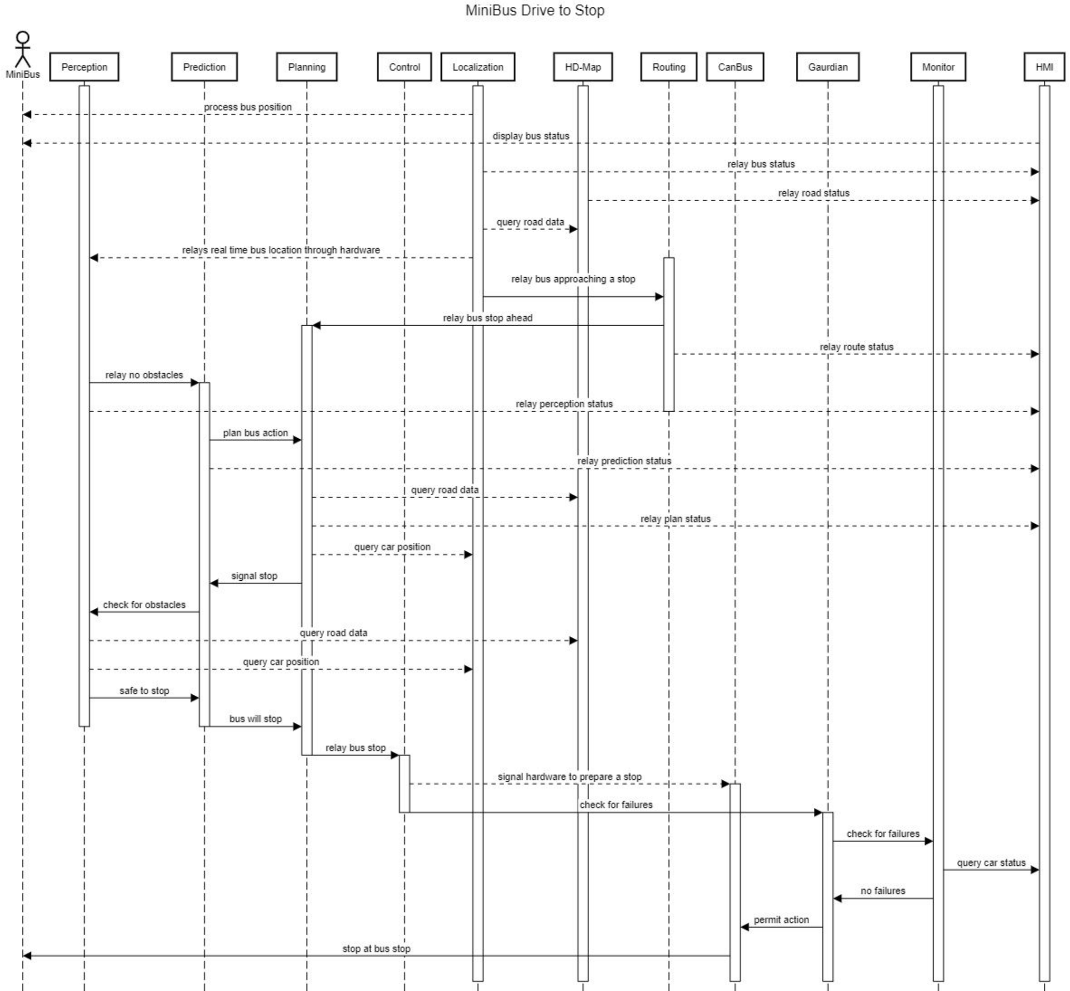
- Data Scientist
  - Ensure that every commit can follow the license, testing, and coding style guidelines
- Quality Assurance
  - Ensure the safety of the systems and verify, validate, and test for automated driving
- Cyber Security
- Hardware
  - ensure the safety of environment sensor configuration, the computing platform, sensors and actuators.

Notable Use Case 1: RoboTaxi

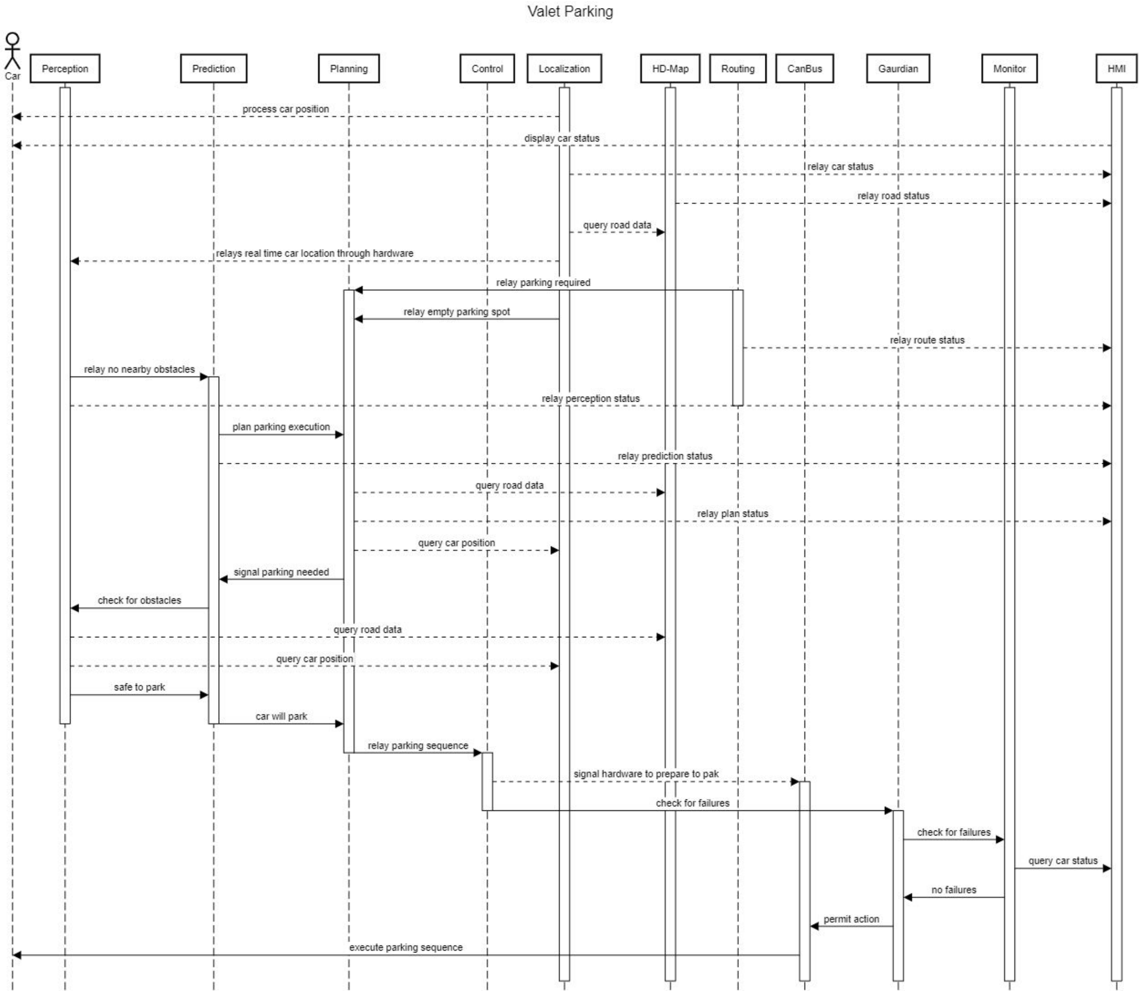
RoboTaxi Lane Change



Notable Use Case 2: MiniBus


MiniBus Drive to Stop

Notable Use Case 3: Valet



Valet Parking

Car

Perception

Prediction

Planning

Control

Localization

HD-Map

Routing

CanBus

Gaurdian

Monitor

HMI

process car position

display car status

relay car status

relay road status

query road data

relays real time car location through hardware

relay parking required

relay empty parking spot

relay route status

relay no nearby obstacles

relay perception status

plan parking execution

relay prediction status

query road data

relay plan status

query car position

signal parking needed

check for obstacles

query road data

query car position

safe to park

car will park

relay parking sequence

signal hardware to prepare to pak

check for failures

check for failures

query car status

no failures

permit action

execute parking sequence

Conclusion

Through our analysis in previous sections, we can get a rough idea of the organization of Apollo system. This organization includes all components, how they interact with each other, the environment in which they operate, and the principles used to design the software. We can also see the evolution of the software. We can tell the enormous effort the project team has put into this system, as well as the different roles within this team.