Architectural Enhancement for Apollo

Group7 ArchiTiger

Website: https://xuan1030.github.io/CISC322-326-ArchiTiger/

Video Url: https://youtu.be/GKdmRxw7fnA

Group Members Intro

- Leader:

  Poppy Li 20181706

- Presenter:

  Xuan Xiong 20147035

  Yuen Zhou 20186821

- Other Group members:

  Yingjie Gong 20144264

  Wang Zhimu 20190758

  Baisheng Zhang 20094496

Current Issue

- Not capable of acting properly under emergencies

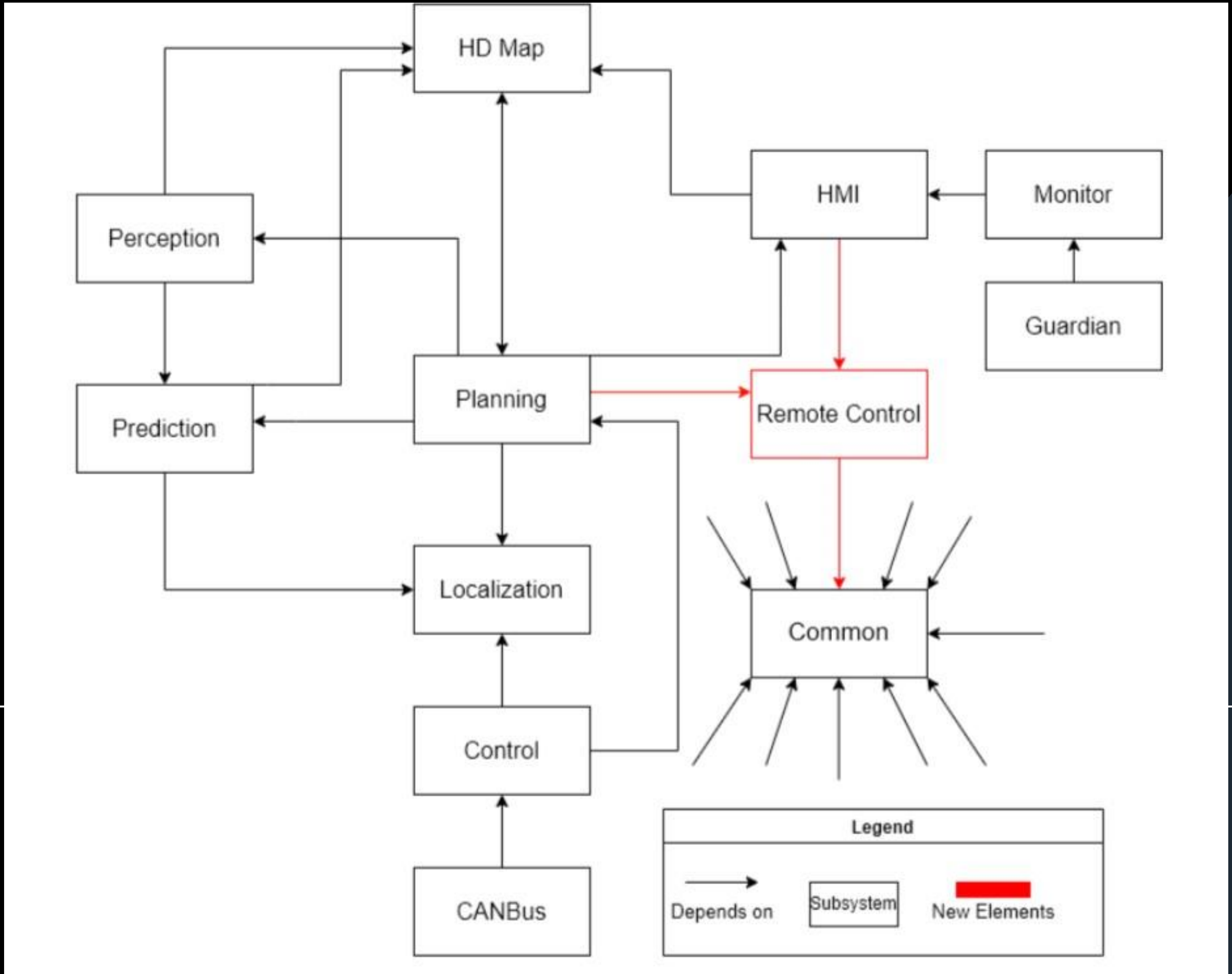
New Feature/Enhancement

- Adapbility to Traditional Traffic Control
  - Emergency vehicles are on a mission
  - Traffic light fails and the police need to guide the traffic

Two Approaches

- Introducing new module
- Adding extra functionalities under existing modules

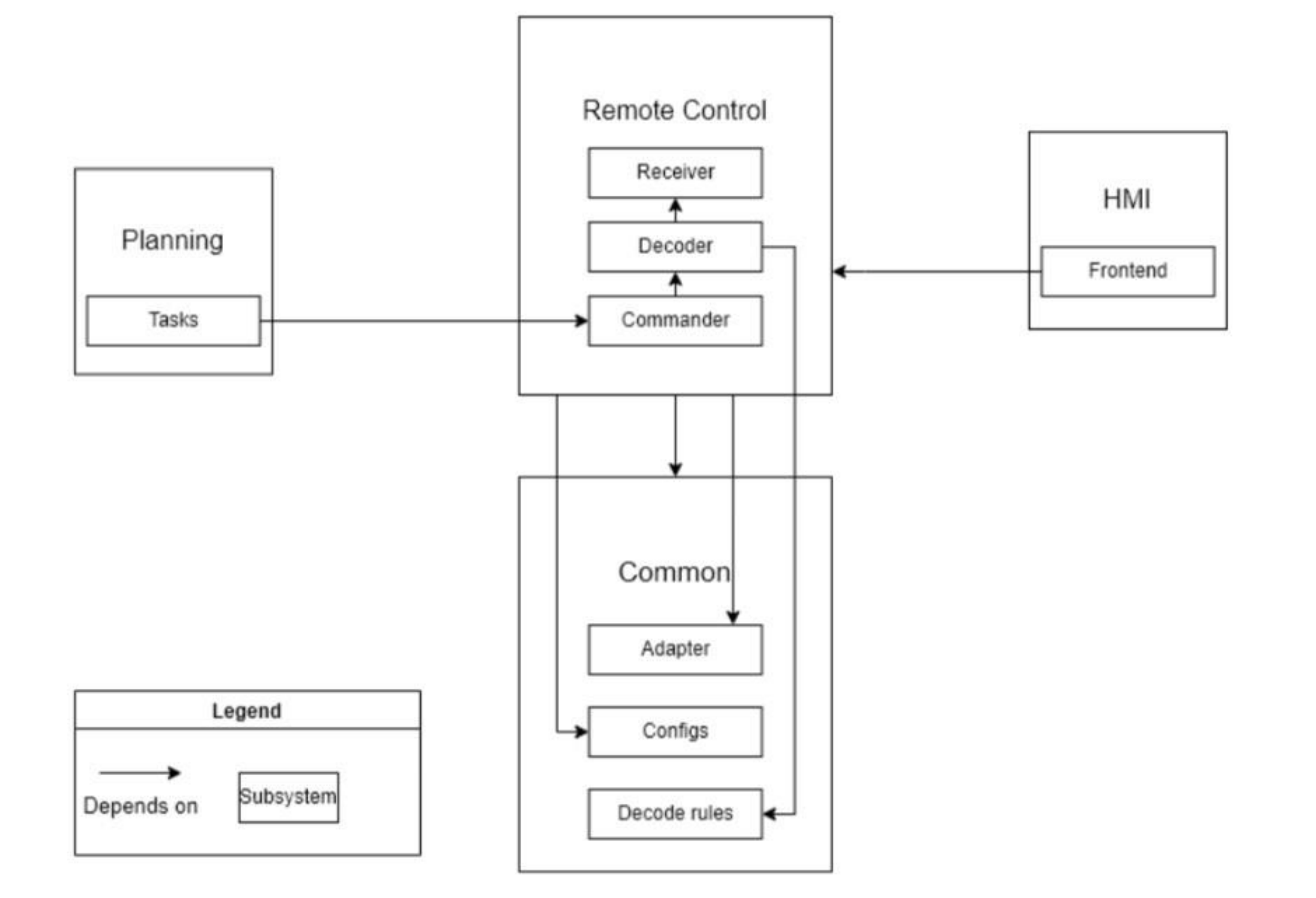
Approach 1: Introducing Remote Control



Approach 1: Introducing Remote Control

Changes to support:

- Remote Control
- Decode rule
- Adapter for Remote Control
- HMI Display



Approach 1: Introducing Remote Control

Impact on current directories/files

New directories/files:

- modules/remote_control/receiver
- modules/remote_control/decoder
- modules/remote_control/commander
- modules/common/math/decode_rules

Modified directories/files:

- modules/planning/tasks
- modules/dreamview/frontend

Modified directories/files:

- modules/planning/tasks
- modules/dreamview/frontend

Architectural Style: Pub-Sub Style

Approach 2: Adding Extra Functionalities

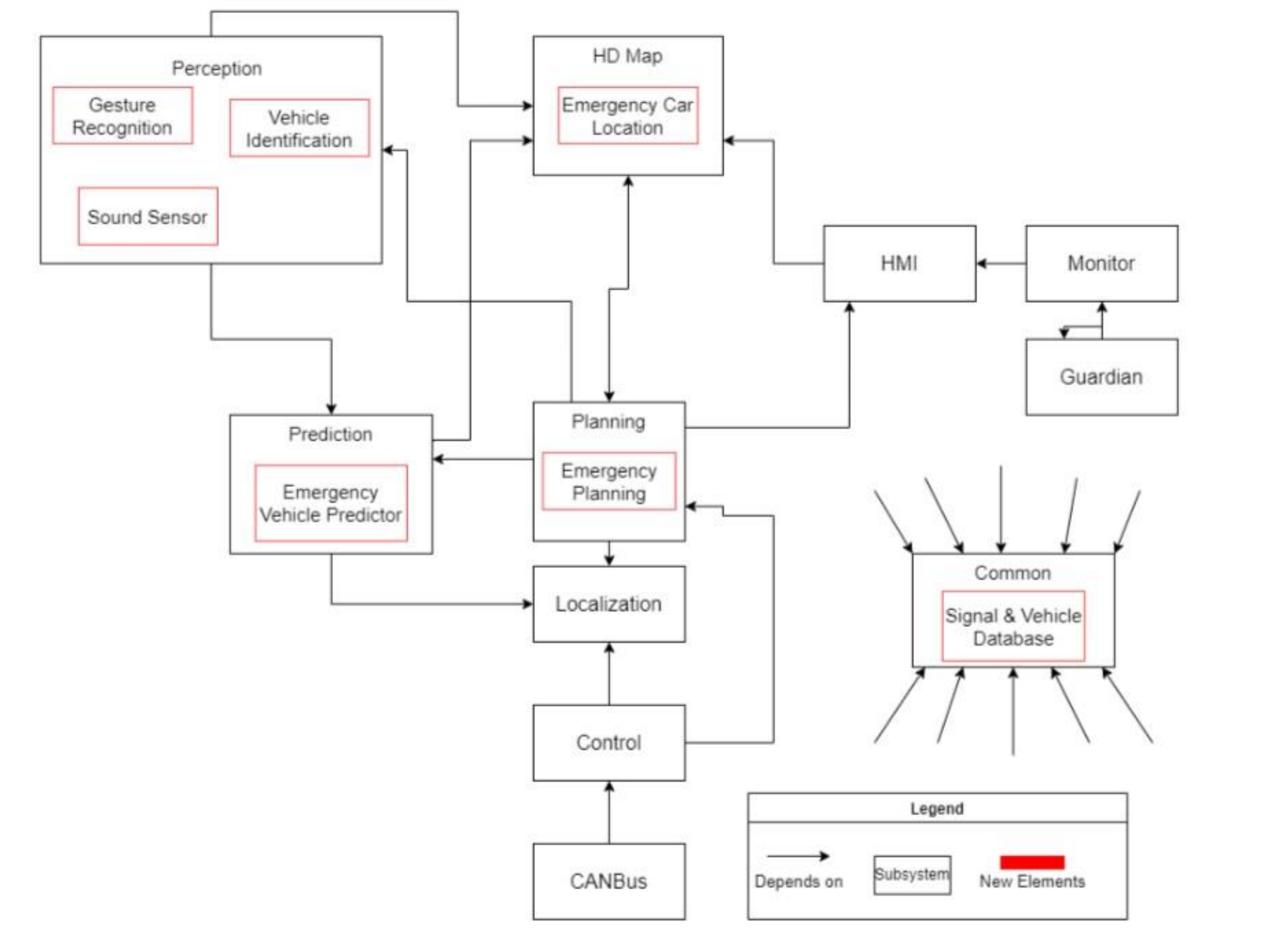


Figure 3. Updated Conceptual Architecture for Implementation 2

Approach 2: Adding Extra Functionalities

Changes to support:

- Gesture Recognition
- Vehicle Identification
- Sound Sensor
- Emergency Vehicle Predictor
- Emergency Planning
- Signals and Vehicles Database
- Emergency Car Location

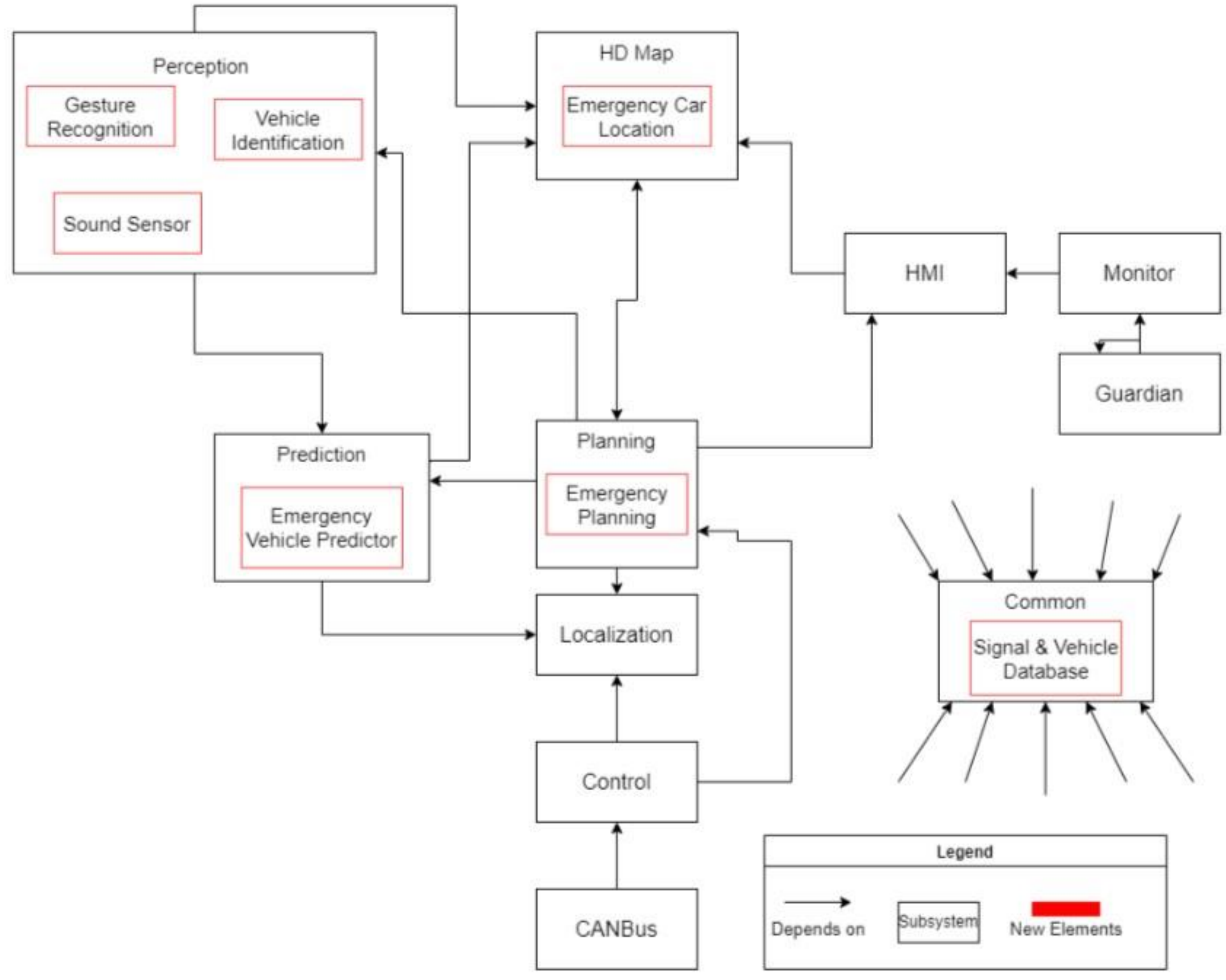


Figure 3. Updated Conceptual Architecture for Implementation 2

Approach 2: Adding Extra Functionalities

Impact on current directories/files

Modified directories/files:

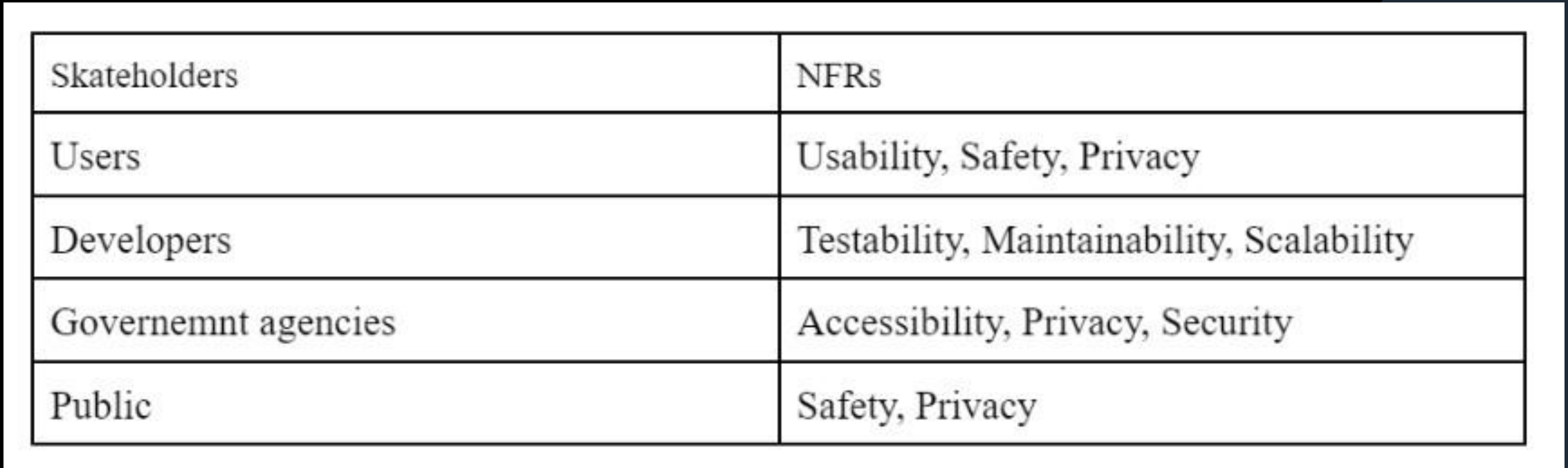
- modules/perception/gesture_recognition
- modules/perception/vehicle_identification
- modules/perception/sound_sensor
- modules/prediction/emergency_vehicle_predictor
- modules/hdmap/emergency_car_location
- modules/planning/emergency_planning
- modules/common/signal_vehicle_database

Updating adapter and configuration files:

- modules/common/adapters/adapter_gflags.h
- modules/common/adapters/adapter_gflags.cc
- modules/common/configs/config_gflags.h
- modules/common/configs/config_gflags.cc

Architectural Style: Pub-Sub Style

SAAM Analysis



Skateholders	NFRs
Users	Usability, Safety, Privacy
Developers	Testability, Maintainability, Scalability
Governemnt agencies	Accessibility, Privacy, Security
Public	Safety, Privacy

SAAM Analysis: Approach 1 Pros & Cons

Advantages

- Higher security
- More conducive to maintenance and repair
- Higher Control

Disadvantages:

- Uncertainty

SAAM Analysis: Approach 2 Pros & Cons

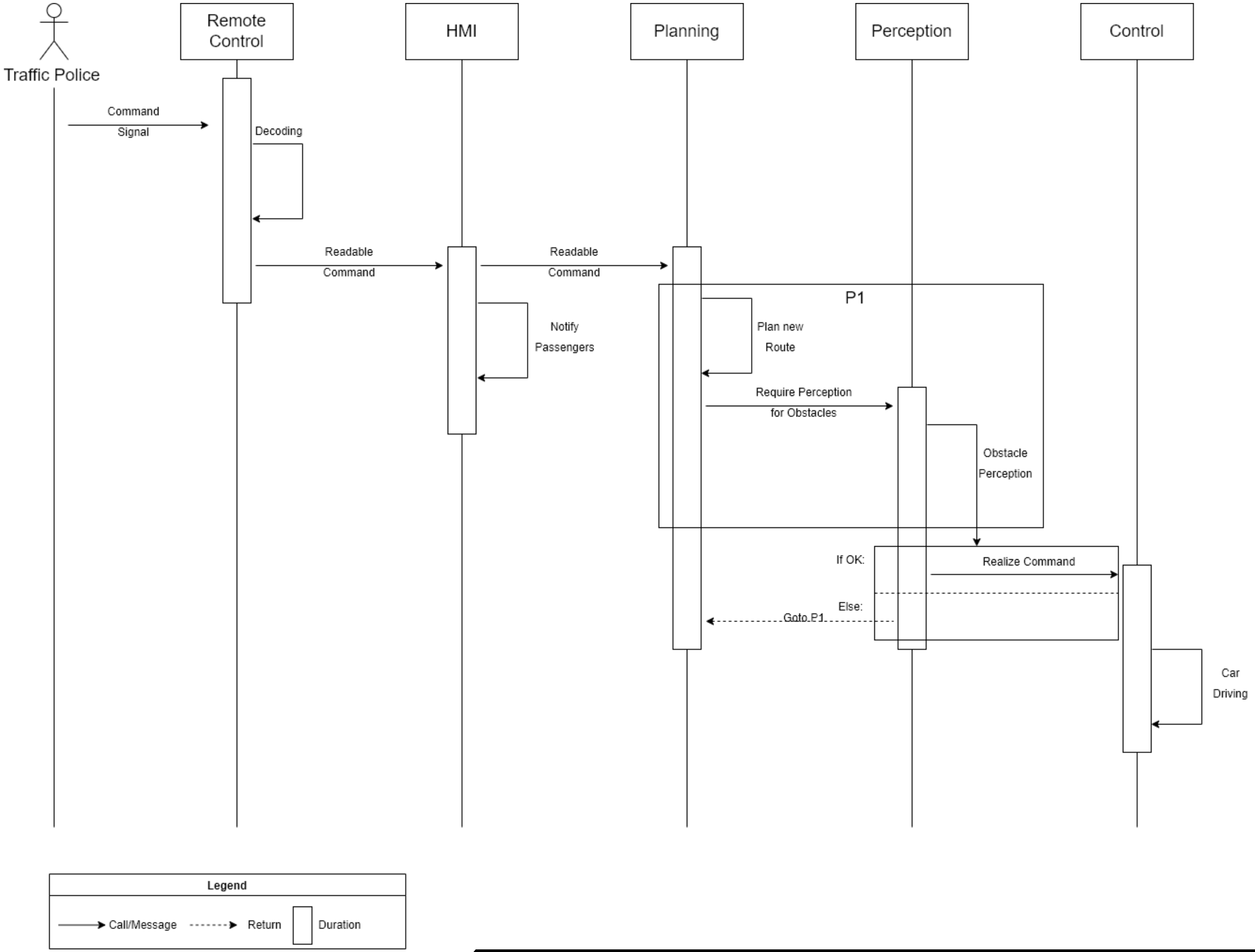
Advantages

- No need for introducing new module
- High degree of freedom

Disadvantages:

- Higher cost on sensor
- More subsystems are involved

Use Case



Sequence Diagram for Use Case: Remote Control By Traffic Police

Risks

- Command and dispatch may be problematic

- Slow reaction time

Limitations

1) Creating a new type of remote control module for the Apollo autopilot system

2) Updating the emergency program that comes with the Apollo autopilot system itself.

Test

- The first will test the car for usability, safety and privacy while the tester is in the remote control state. Under developer tuning tests;
- The second group will focus on the car's testability, maintainability, and scalability.

  - Test whether the software and hardware in the Apollo car is always available under different road conditions and speeds. Besides, whether there are faults that lead to dangerous situations；
  - After completing the test, check whether the car can automatically delete the sensitive information recorded before;
  - Detect the new hardware and software loaded on the original hardware, check the compatibility and test results;
  - Detect whether the maintenance time is faster than other car companies of the same level when there is a problem with a simulated module.

Concurrency

- Apollo has a particular scheduling system to meet real-time performance needs;

- Apollo's mode of action comes from multiple interacting processes with concurrency between processes. This high concurrency improves the performance of the perception module and allows Apollo to handle detection of obstacles well;

- Apollo's concurrency is composed of 5 nodes, each of which can be considered as a thread. These threads work for the perception module and there will be connections between the threads.

Team Issues

- Due to the size and complexity of Apollo's subsystems, some of which have similar functionality, it is difficult to analyze dependencies between modules;

- The idea of some enhancements could easily overlap with existing functionality;

- The functional ambiguity of some enhancements was difficult to express clearly.

Lessons Learned

- We were able to offer our insights into improving the whole system;

- We also learned the importance of privacy and security when perfecting a technique with user information;

- The SAAM analysis method is beneficial to assess the risks inherent in the architecture.

CONCLUSION

THANK YOU!