Concrete Architecture of Apollo System

Group7 ArchiTiger

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

Video url:https://youtu.be/7JXOic5UkZM

Group Members Intro

- Leader:

  Poppy Li 20181706

- Presenter:

  Xuan Xiong 20147035

  Yuen Zhou 20186821

- Other Group members:

  Yingjie Gong 20144264

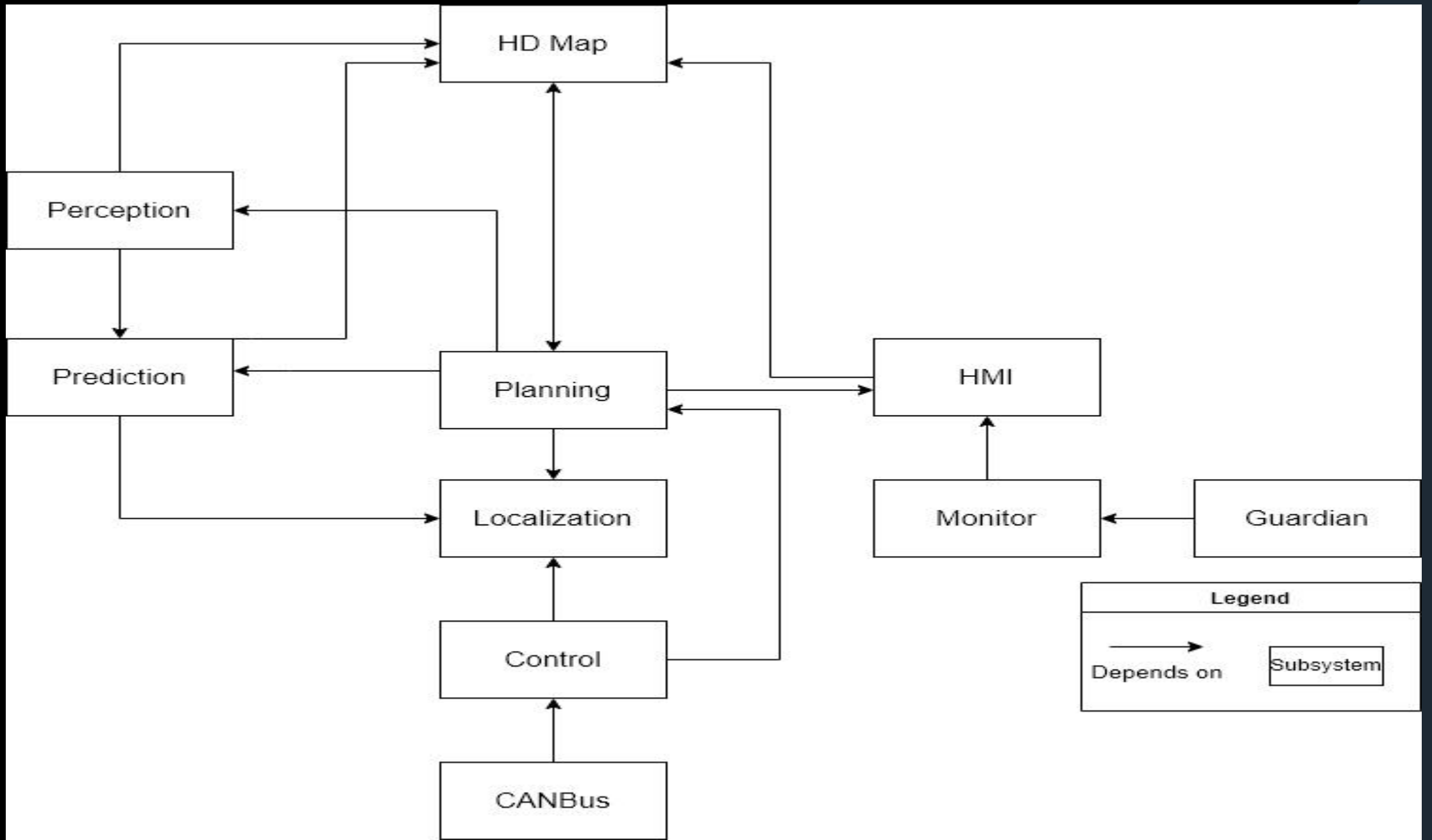
  Wang Zhimu 20190758

  Baisheng Zhang 20094496

Derivation Process

- Update Conceptual Architecture
- Use Understand

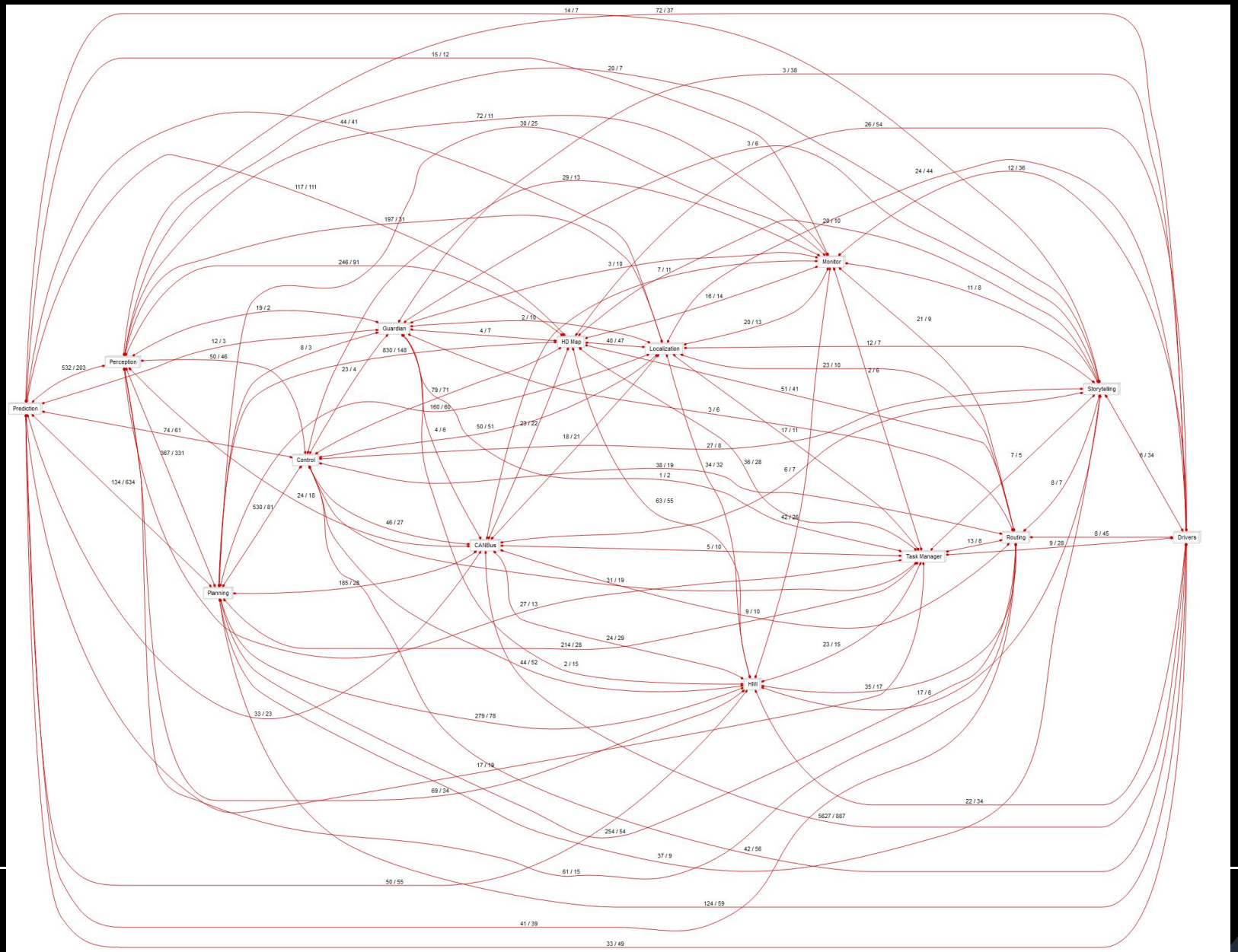
Updated Conceptual Architecture



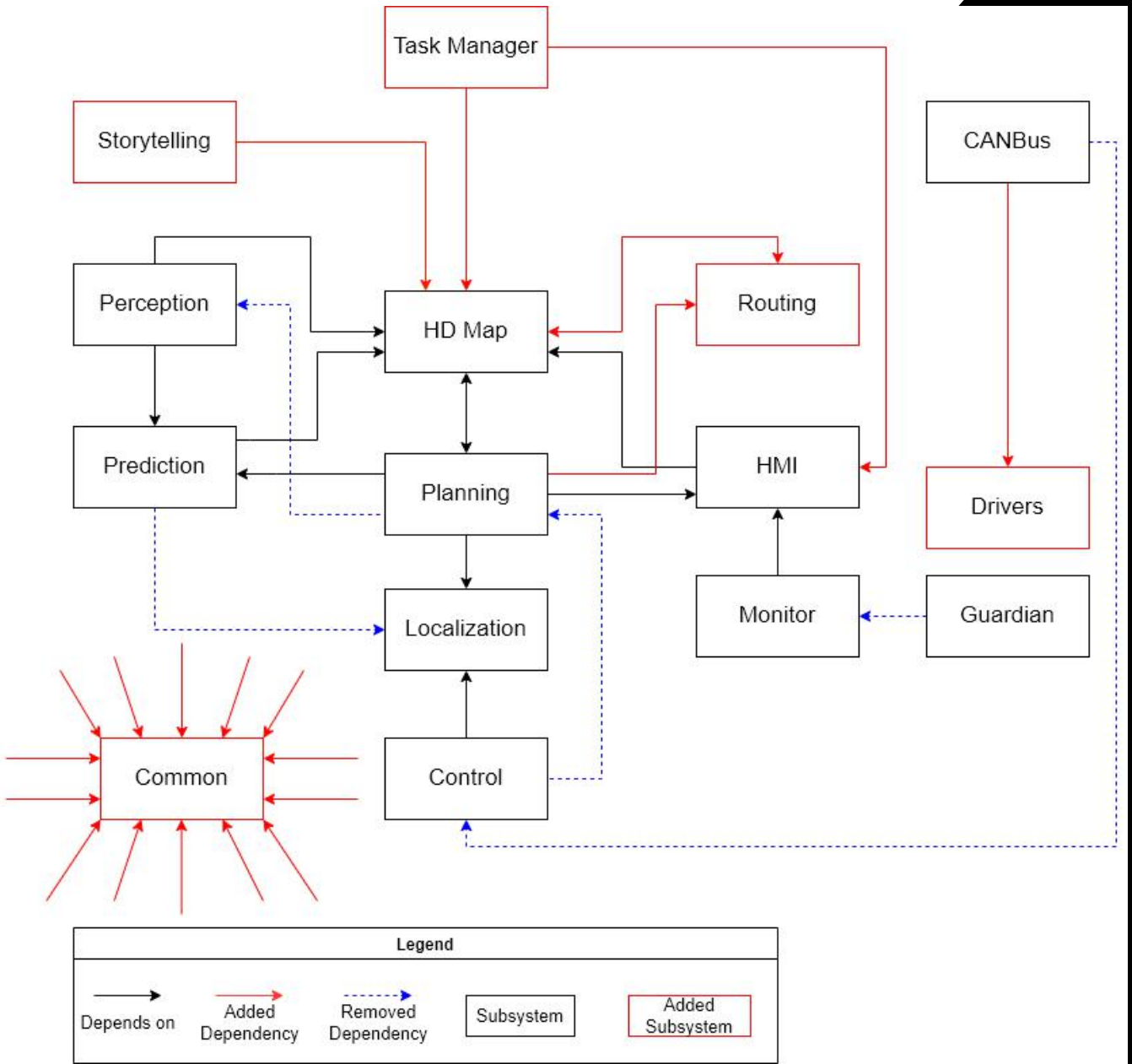
Derivation Process

- Update Conceptual Architecture
- Use Understand

Considered Alternatives



Concrete Architecture



Divergences at High Level

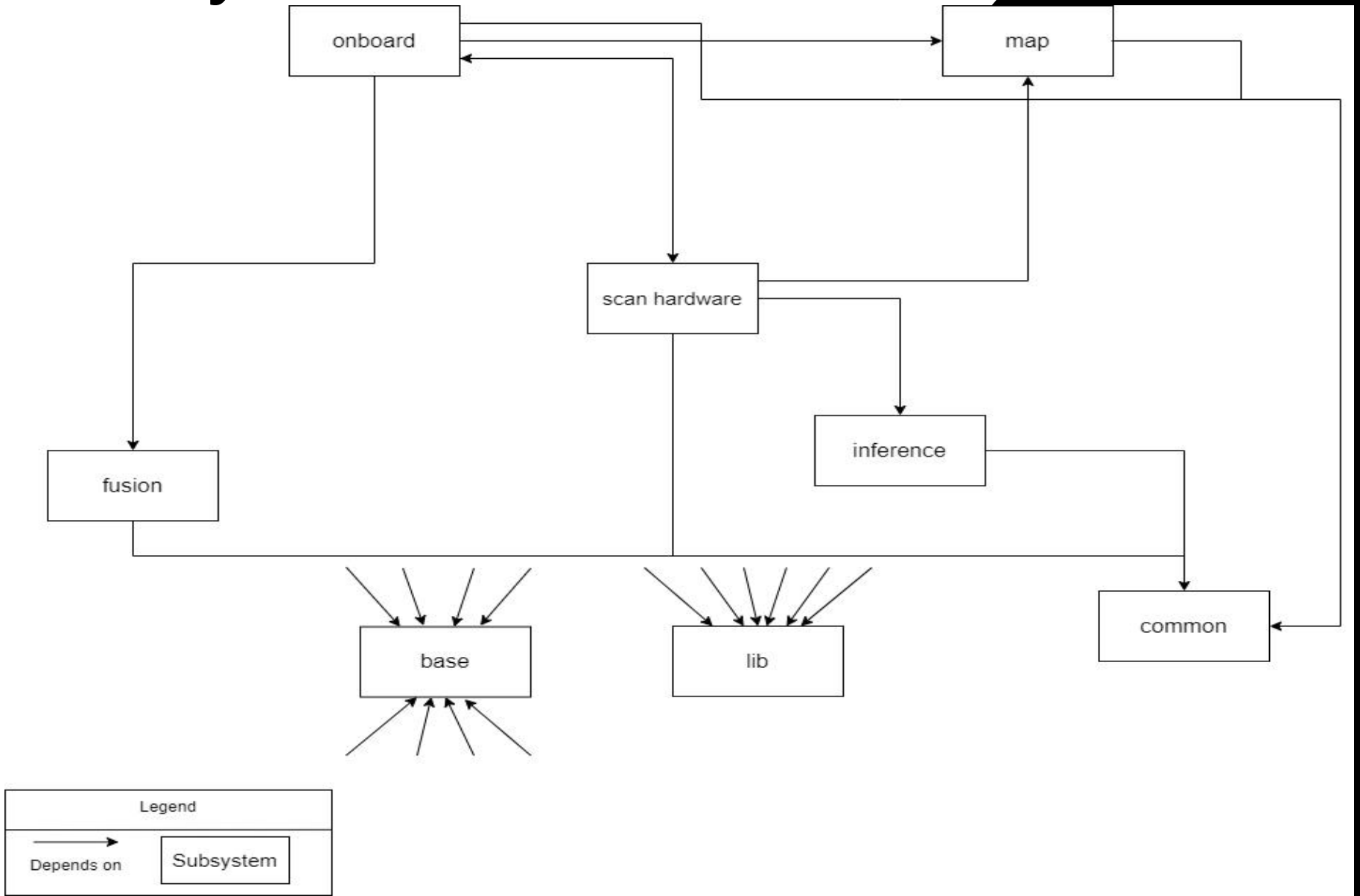
Added Dependencies:

- CANBus => Drivers
- TaskManager => HMI
- TaskManager => HD Map
- Everything else => Common
- Storytelling => HD Map
- HD Map <=> Routing
- Planning => Routing

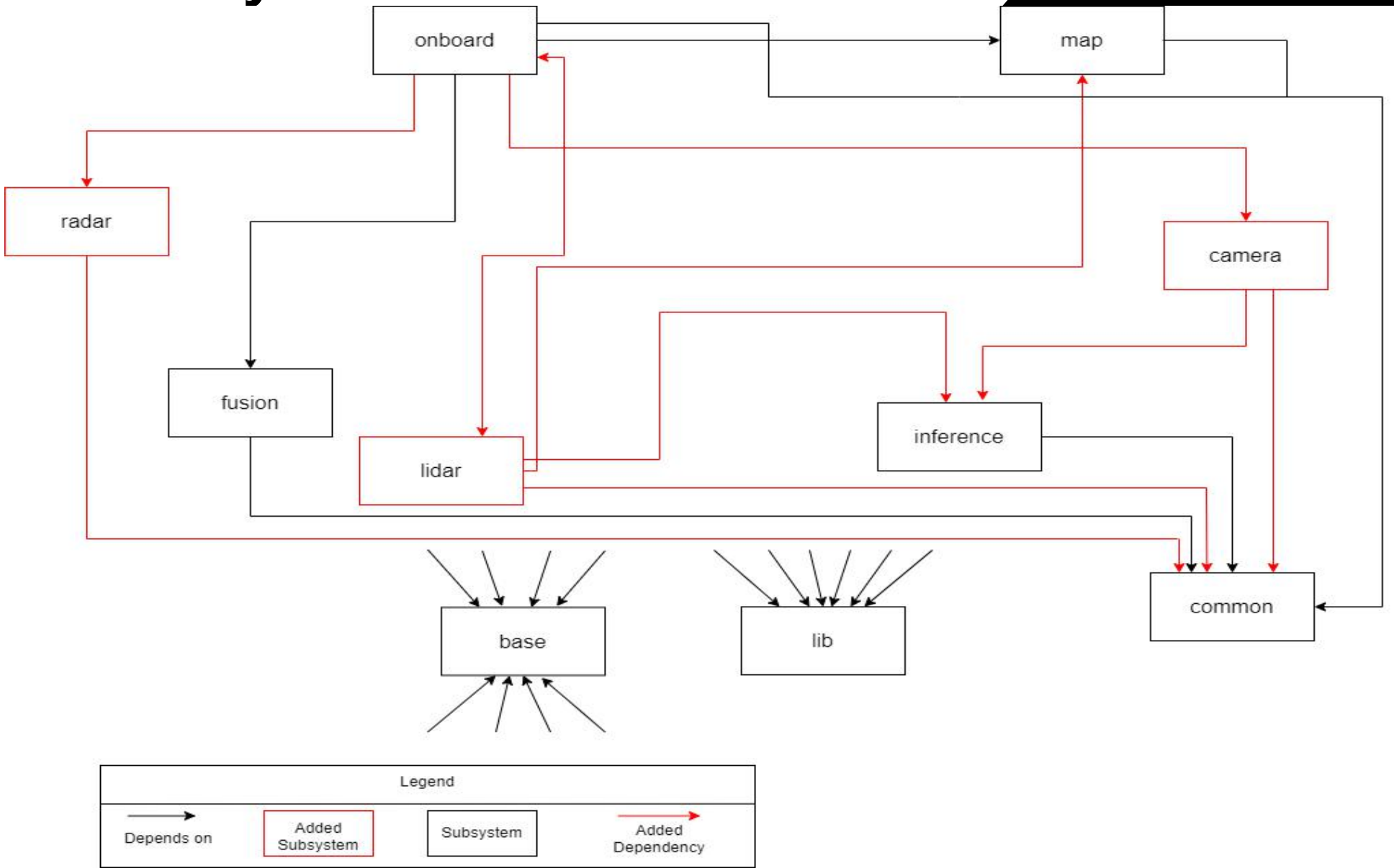
Removed Dependencies:

- Planning=>Perception
- Prediction=>Localization
- Control=>Planning
- Guardian=>Monitor
- CANBus=>Control

Conceptual Architecture of 2nd level subsystem



Concrete Architecture of 2nd level subsystem



Divergences at 2nd Level

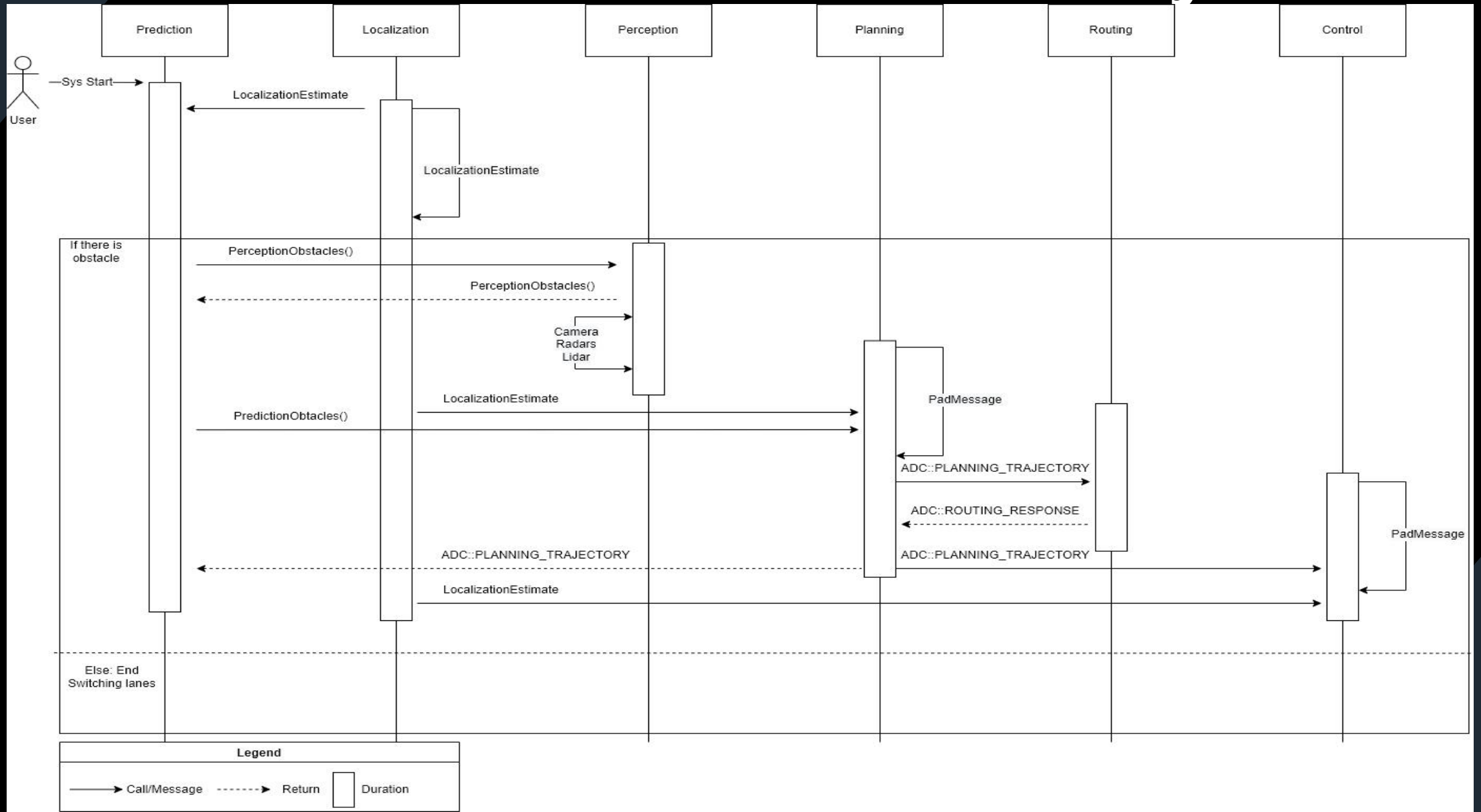
Added Dependencies:

Onboard => Rader
Rader => Common
Onboard => Lidar
Lidar => Onboard
Lidar => Common
Lidar => Map
Lidar => Inference
Onboard => Camera
Camera => Inference
Camera => Common

Removed Dependencies:

Scan Hardware => Onboard
Onboard => Scan Hardware
Scan Hardware => Map
Scan Hardware => Inference
Scan Hardware => Common

Use Case& Functionality



Concurrency

- Apollo has a particular scheduling system to meet real-time performance needs;
- Apollo's action mode comes from multiple interacting processes;
- Concurrency occurs in the Apollo perception module - responsible for checking obstacles around the car, consisting of 5 nodes (LiDAR, Radar, Fusion, Traffic Light preprocessing and Traffic Light process);
- Each node can be regarded as a thread;

Team Issues

- Apollo's subsystems are very numerous and complex;
- The content of the ReadMe document of some modules is not straightforward.
- Later maintenance will be challenging.

Lessons Learned

- We understood why conceptual architecture and concrete architecture do not match;
- The most important thing is that we learned how to use Understand to analyze the dependencies between modules;
- We also learned a deep understanding of the importance of teams.

Limitations of Reported Findings

- Firstly, SciTools Understand is complex software. Using it to analyze the dependencies between modules always takes us much time;

- Secondly, when we analyzed the concrete architecture of Apollo, we discovered that the module ' Task Manager' was not completed. Since Task Manager is a new system of Apollo 7.0, it is still in the development stage, so many functions are not finalized.

CONCLUSION

THANK YOU!