Group 8 - Artemis

Concrete Architecture of Apollo

Video Link: https://www.youtube.com/watch?v=cydR_oTgCjo

Team Artemis

- **Josh Otten** - Use cases, Derivation Process, Lessons learned and limitations
- **Aleks Jugovic** - Localization (chosen subsystem), Reflexion analysis, Use cases, Conceptual architecture changes, Presentation
- **Muyun Yang** - Concrete architecture, Architecture graph creation
- **Chong Guan** - Abstract, Introduction, Conclusion, Reflexion analysis
- **Daniel Jang** - Localization (chosen subsystem), Presentation
- **Wooseok Lee** - Team Leader, New subsystems, Architecture graph creation

Agenda

1. Introduction
2. Derivation Process
3. Alternatives Considered
4. Conceptual Architecture Changes
5. Concrete Architecture Overview with Reflexion Analysis
6. Second-Level Subsystem (Localization) Overview with Reflexion Analysis
7. Use Case and Sequence Diagram
8. Lessons Learned and Limitations
9. Conclusion

Introduction

Apollo

- Building on Apollo's conceptual architecture
- Creating a concrete architecture
- Performing a reflexion analysis
- Explore a second-level subsystem, Localization

Derivation Process

Derivation Process - Apollo Concrete Architecture

1. Individual Understand Tool architecture generation

2. Consolidation of individual diagrams

3. Import Pub/Sub message flow, finalize architecture

4. Update conceptual architecture

Derivation Process - Localization Conceptual / Concrete Architecture

1. Individual research into research papers published by Apollo developers

2. Consolidation of research into conceptual architecture

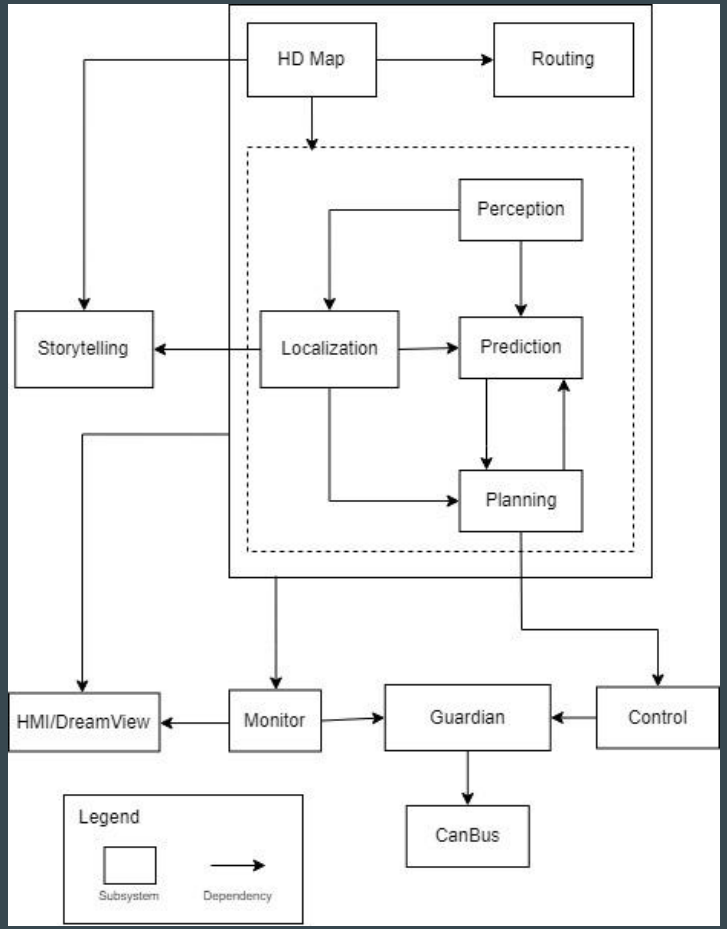
3. Code analysis to recover concrete architecture

4. Rationalization of discrepancies between architectures.

Alternatives Considered

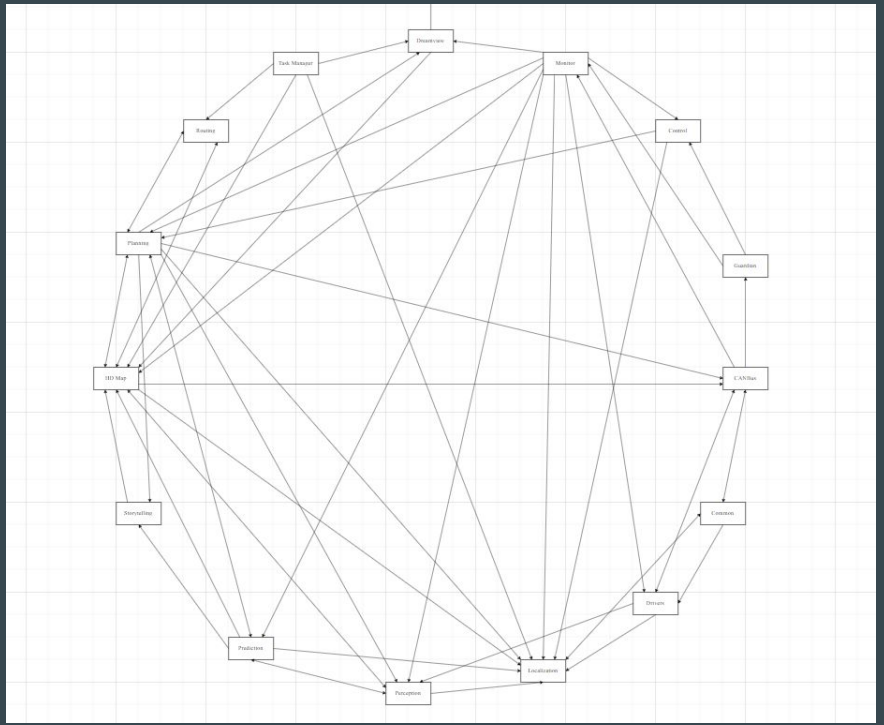
Alternatives - Conceptual Architecture

- No intention to modify architecture
- Issues appeared upon closer examination
- Changes made due to increased understanding



Alternatives - Concrete Architecture

- Analyzing Apollo with Understand
- Clearly visualize connections between subsystems
- Not including pub-sub connections



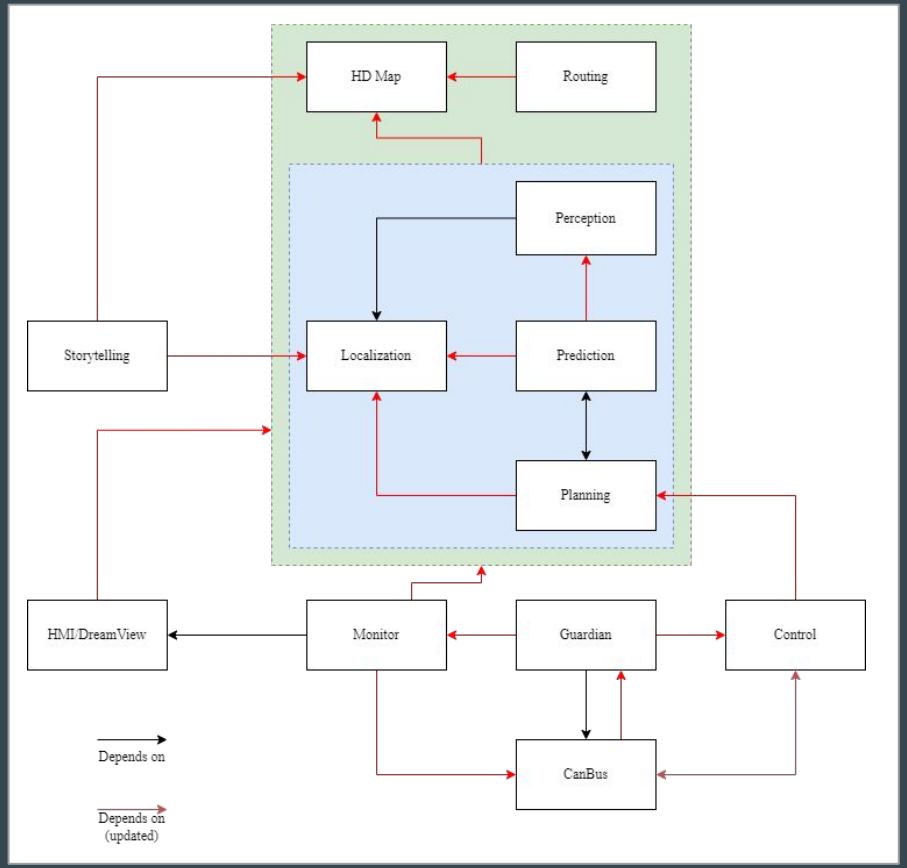
Conceptual Architecture Changes

Conceptual Architecture Changes

Two major changes:

1. Flipped arrows

2. Three new dependencies
   a. CanBus -> Guardian
   b. Control <-> CanBus
   c. Monitor -> CanBus



Concrete Architecture

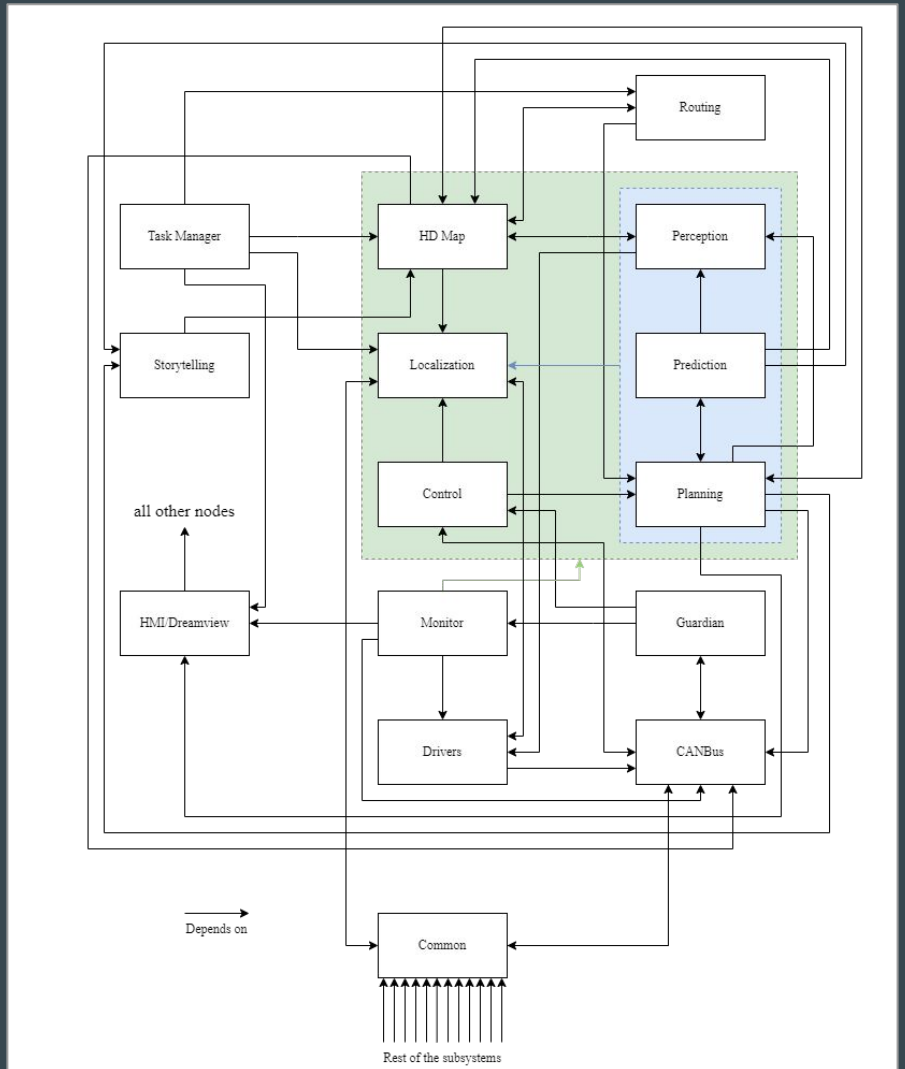
Concrete Architecture

Overview:

- Still Pub/Sub style

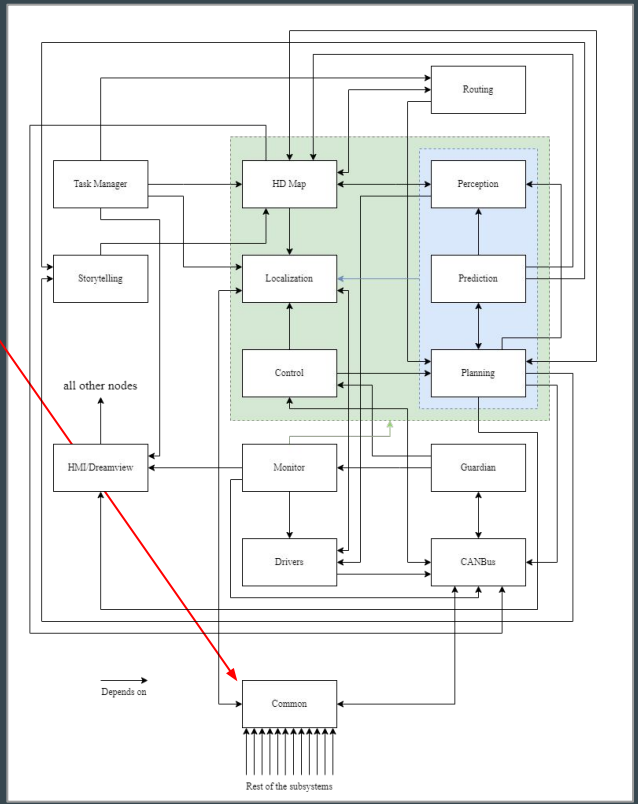
- Three new subsystems:
  - Drivers
  - Common
  - Task Manager

- Many new dependencies



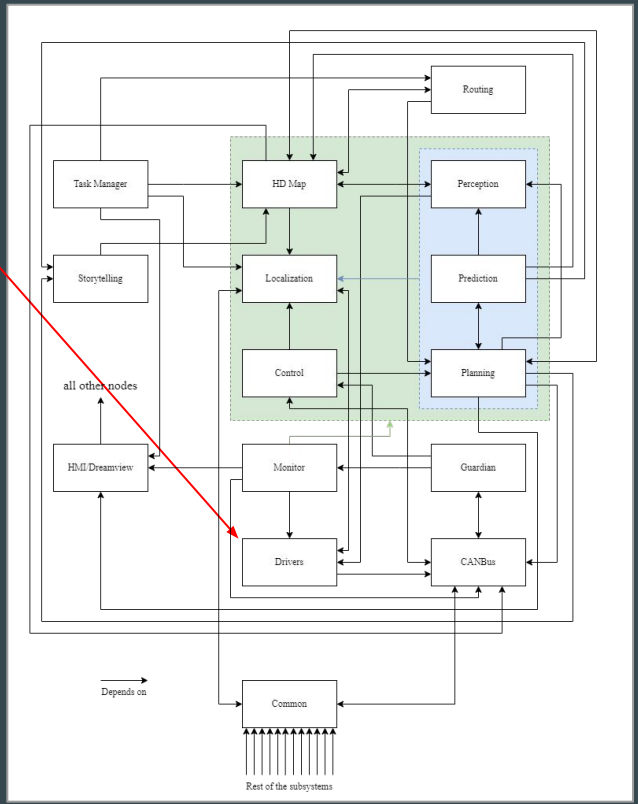
Concrete Architecture - Reflexion - New Subsystem - Common

- Code useful for function of Apollo but not specific to any subsystem
- Multiple different subfolders such as
  - Data
  - Filters
  - KVDB
- All subsystems depend on it
- It has a dependency with Localization and CanBus



Concrete Architecture - Reflexion - New Subsystem - Drivers

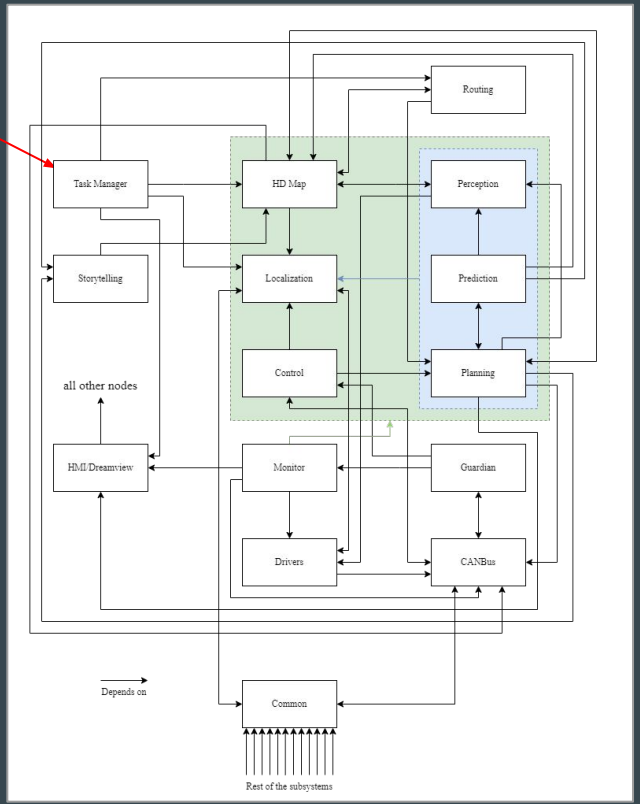
- Drivers to operate additional hardware such as cameras and radars
- Monitor, Localization, Perception, and CanBus have dependency with it
- Has dependency on CanBus and Localization



Concrete Architecture - Reflexion - New Subsystem - Task Manager

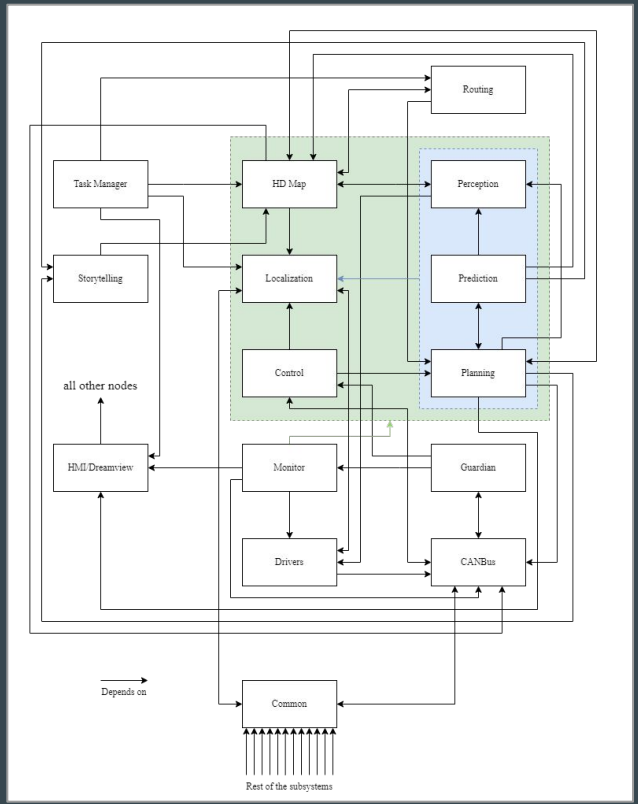
- Subscribes to the Localization and Routing subsystems

- Depends on HD Map, HMI Dreamview, and Routing



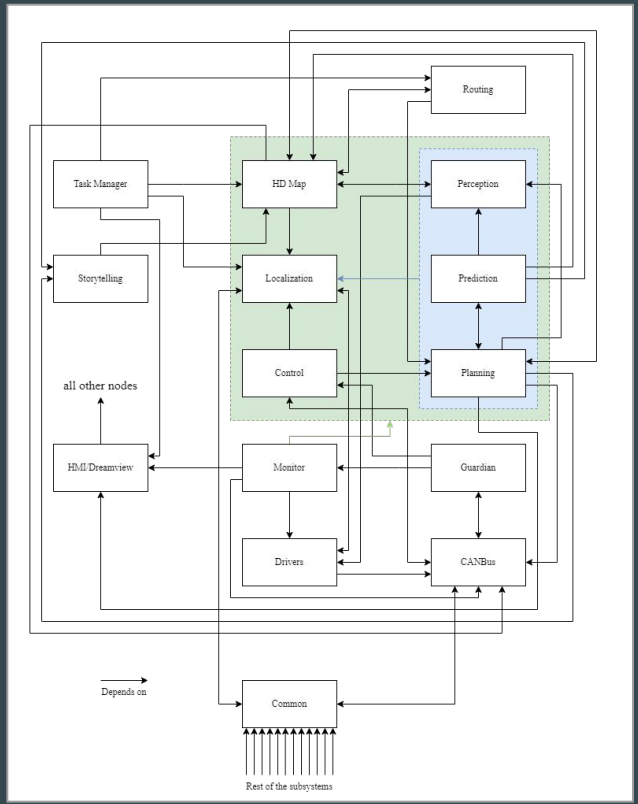
Concrete Architecture - Reflexion - New Dependencies

- Routing -> Planning
  - Debug information
- Prediction -> Storytelling
  - Improve prediction with stories
- Planning -> Storytelling
  - Improve planning with stories
- Planning -> Perception
  - Traffic light obstacles to improve the planning
- Planning -> CanBus
  - Monitor car chassis information



Concrete Architecture - Reflexion - New Dependencies

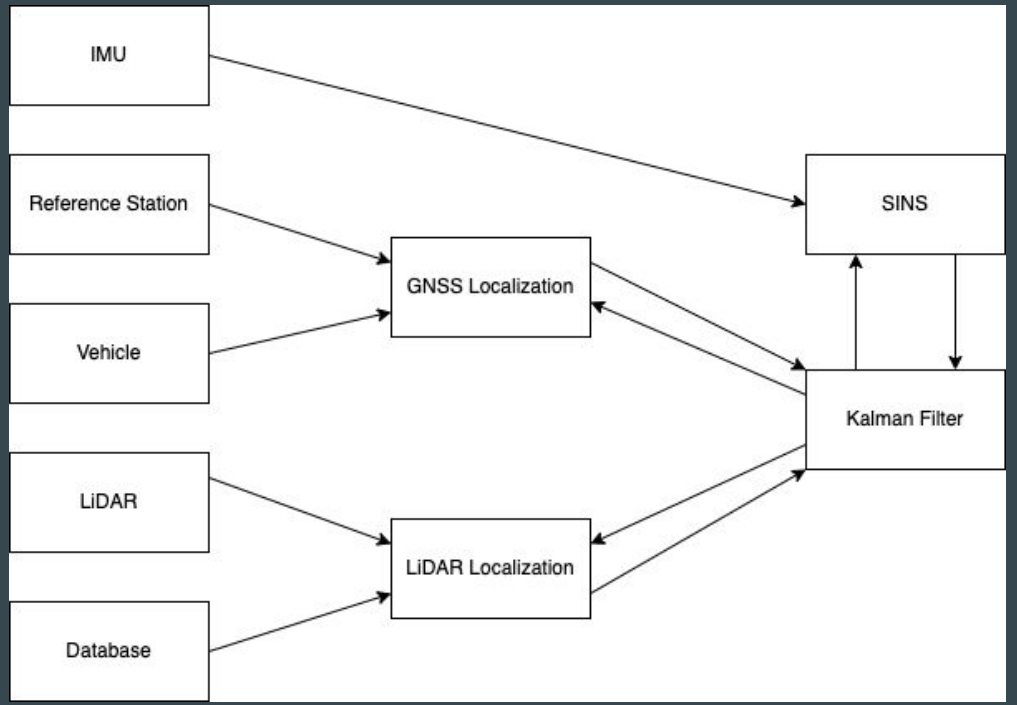
- HD Map -> Planning
  - Generate map for planned trajectory
- HD Map -> Perception
  - Generate map with perceived obstacles
- HD Map -> Routing
  - Generate map with current route
- HD Map -> CanBus
  - Generate relative map with chassis information
- HMI -> All other nodes
  - Monitor all subsystems for the user of car



Second-Level Subsystem (Localization)

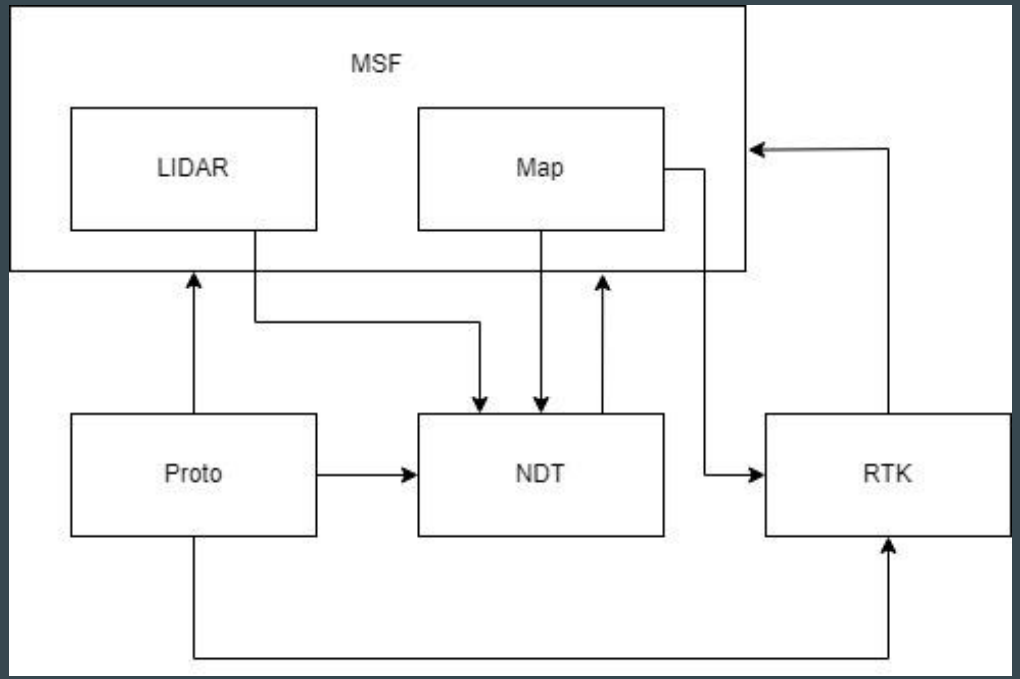
Localization Conceptual Architecture

- Estimation of the current location of the vehicle
- LiDAR, RTK, IMU
- Pipe and Flow architecture
- Fusion framework



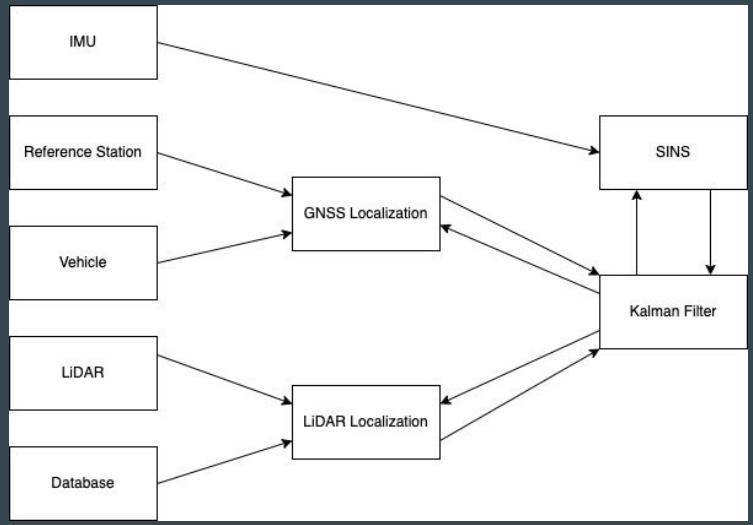
Localization Concrete Architecture

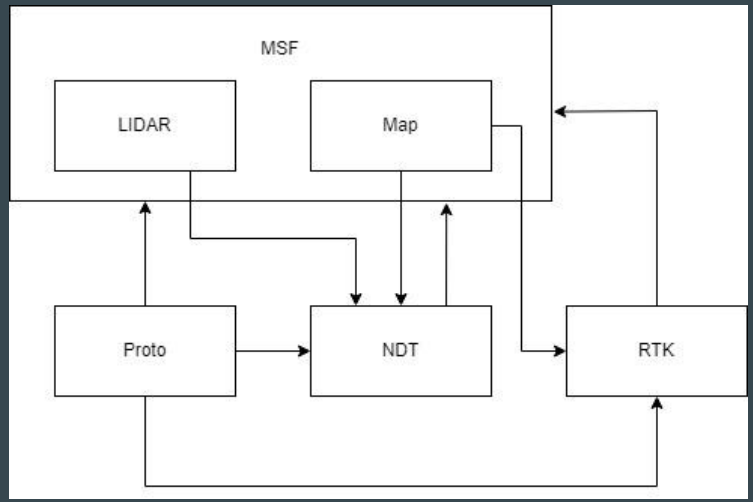
- RTK, NDT, MSF localization
- LiDAR, Proto input
- Map information processing



Localization Reflection Analysis

- Kalman + SINS -> MSF
- LiDAR -> LiDAR
- LiDAR localization -> NDT
- GNSS + Reference -> RTK
- Vehicle + IMU -> Proto

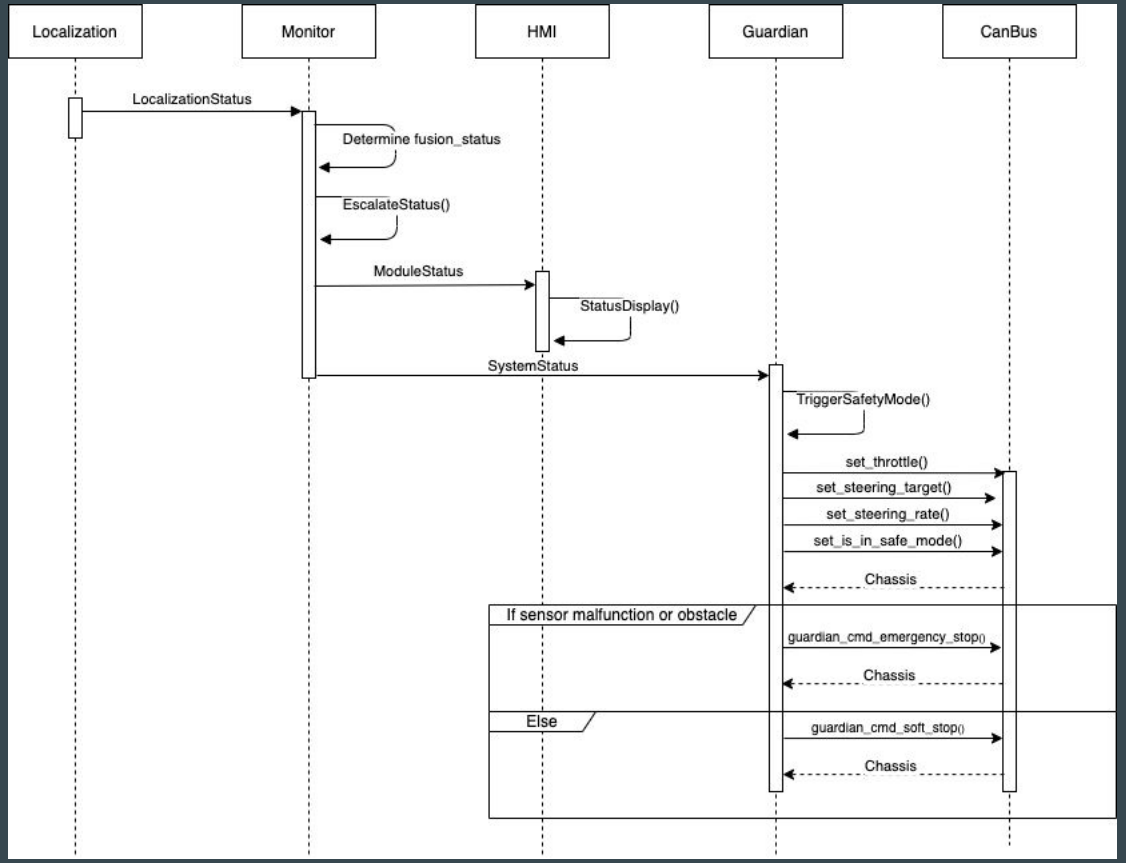




Use Case - Emergency Stop

Use Case: Emergency Stop

- Localization module crash
- Guardian steps in to execute emergency stop of the car
- Multiple stop scenarios



Lessons Learned and Limitations

Limitations and Lessons Learned

- Using the Understand Tool
- Vague commit messages from Apollo devs
- Documentation poor for some subsystems
- Unfeasible to consider all source code files

Conclusion

Conclusion

- Concrete architecture expanding on the conceptual architecture
- New subsystems added and reflexion analysis performed
- Localization follows a pipe and filter approach within publish and subscribe
- For the future