

Video Link:<https://youtu.be/iom81qD1C7U>

CISC 322 Assignment 2

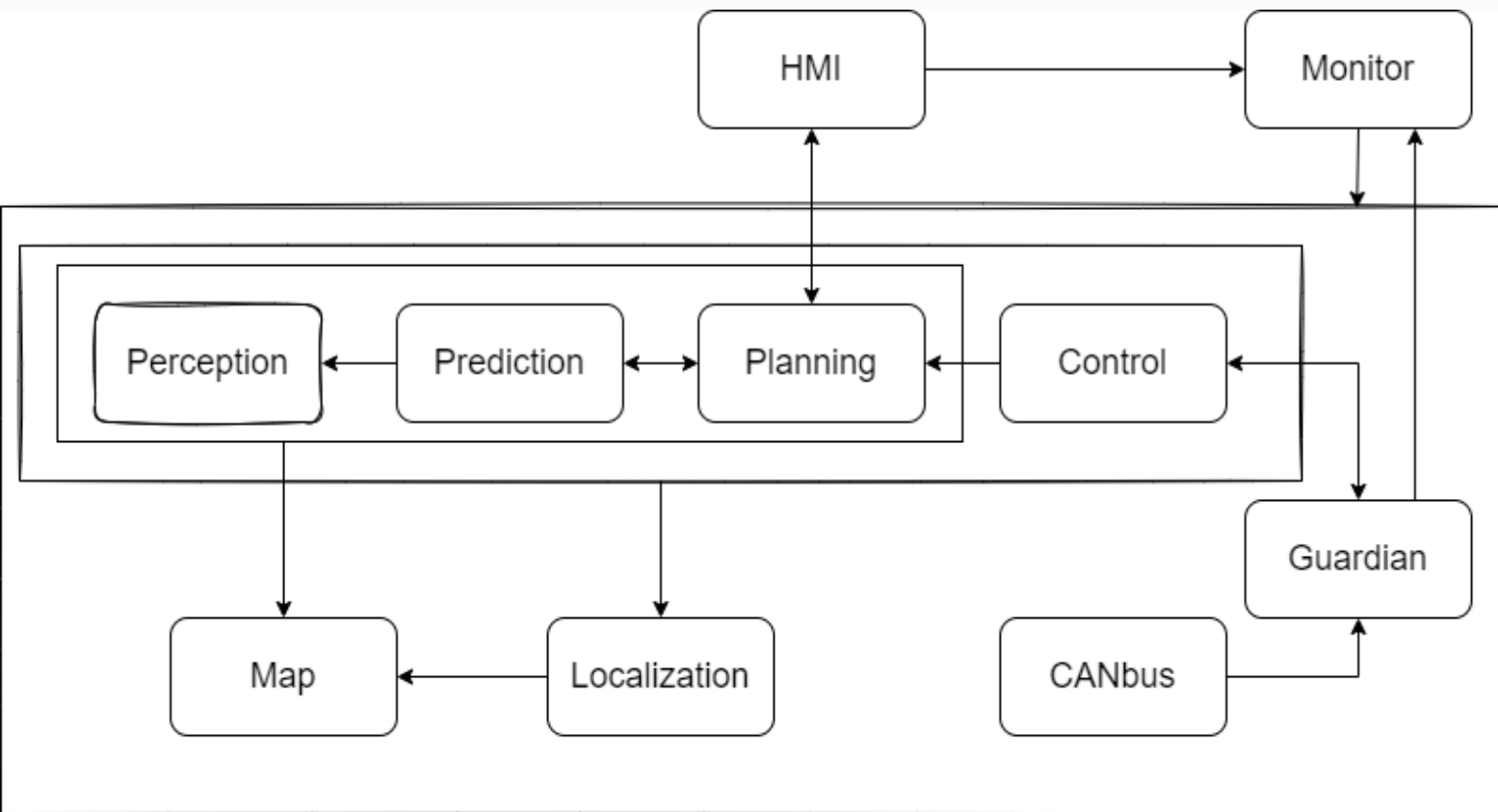
Concrete Architecture of Apollo

Group 19

Presented by: Runze Lin, Wonton Zhou, Haoyun Yang



Conceptual Architecture Review

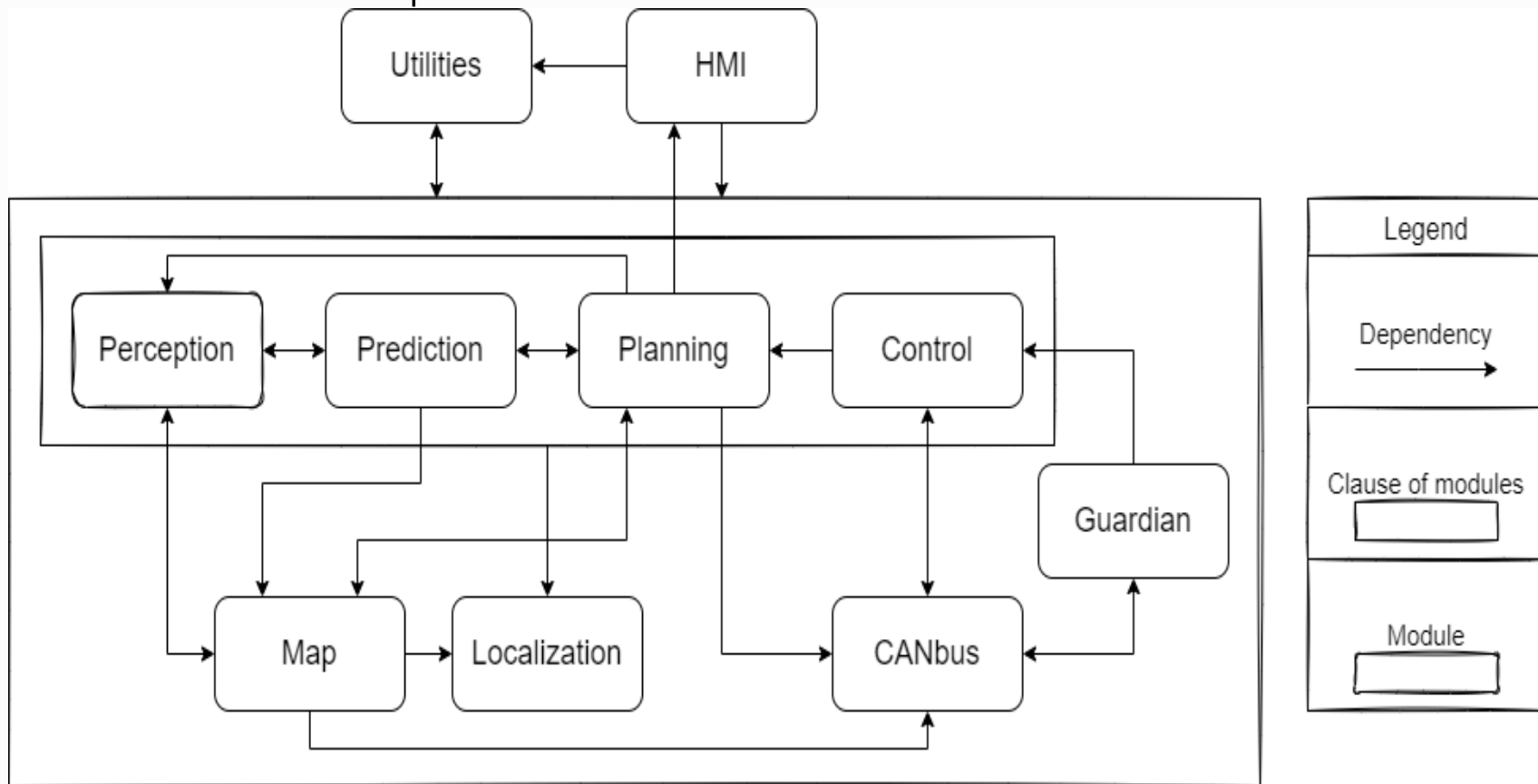




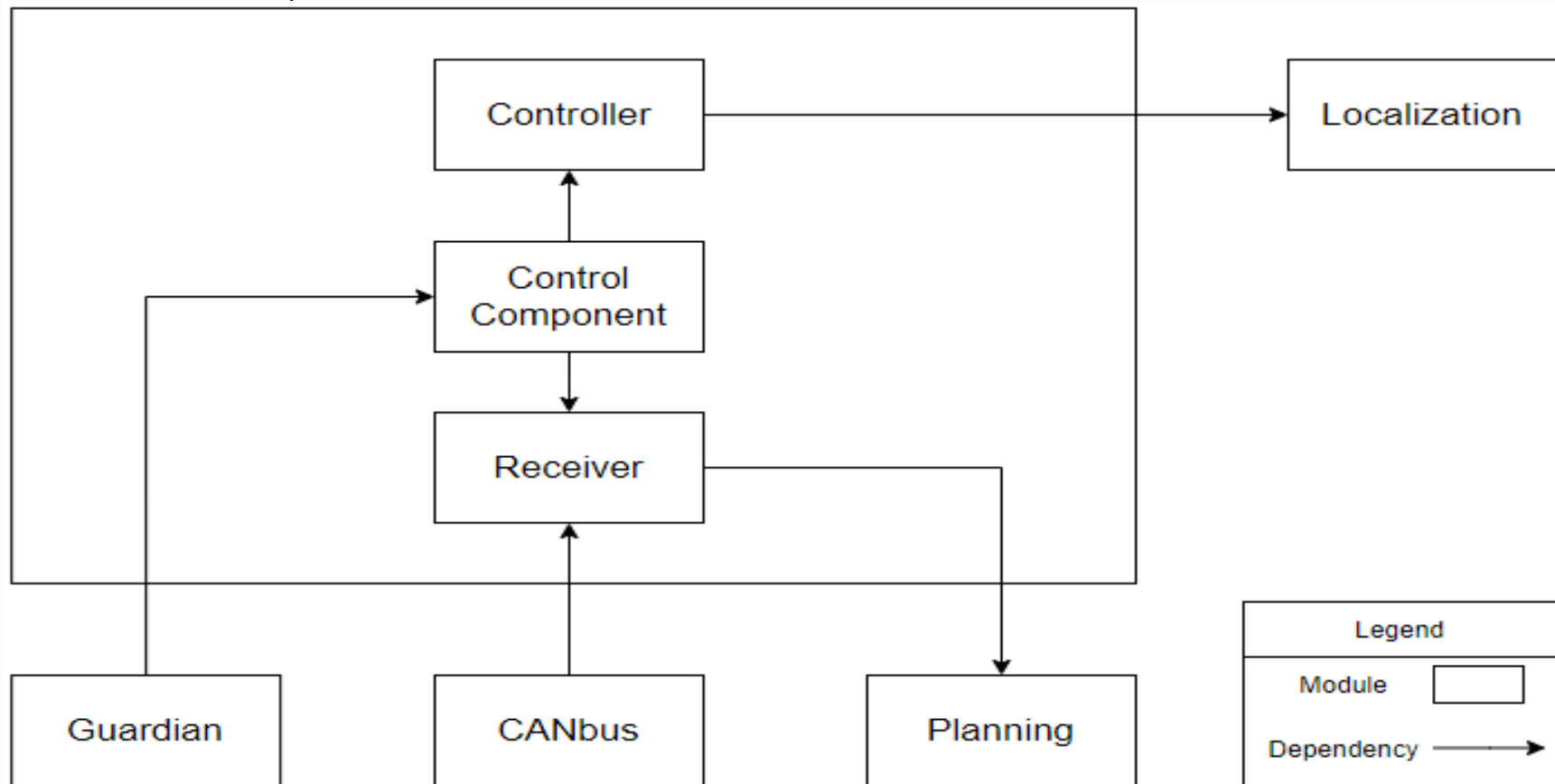
Derivation Process

- Scitools Understand
- Pub-sub architecture
- reflexion

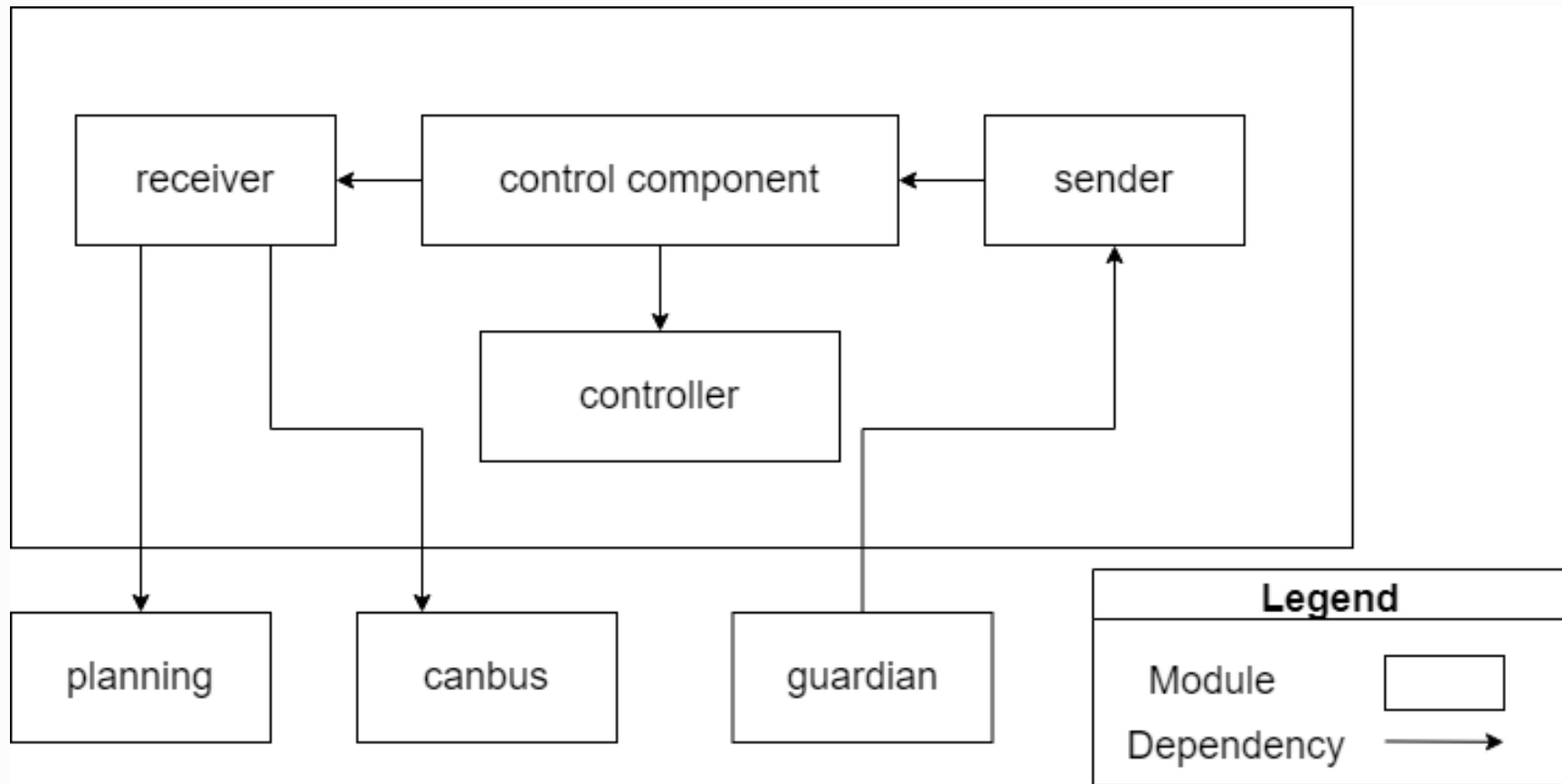
Top-level Concrete Architecture



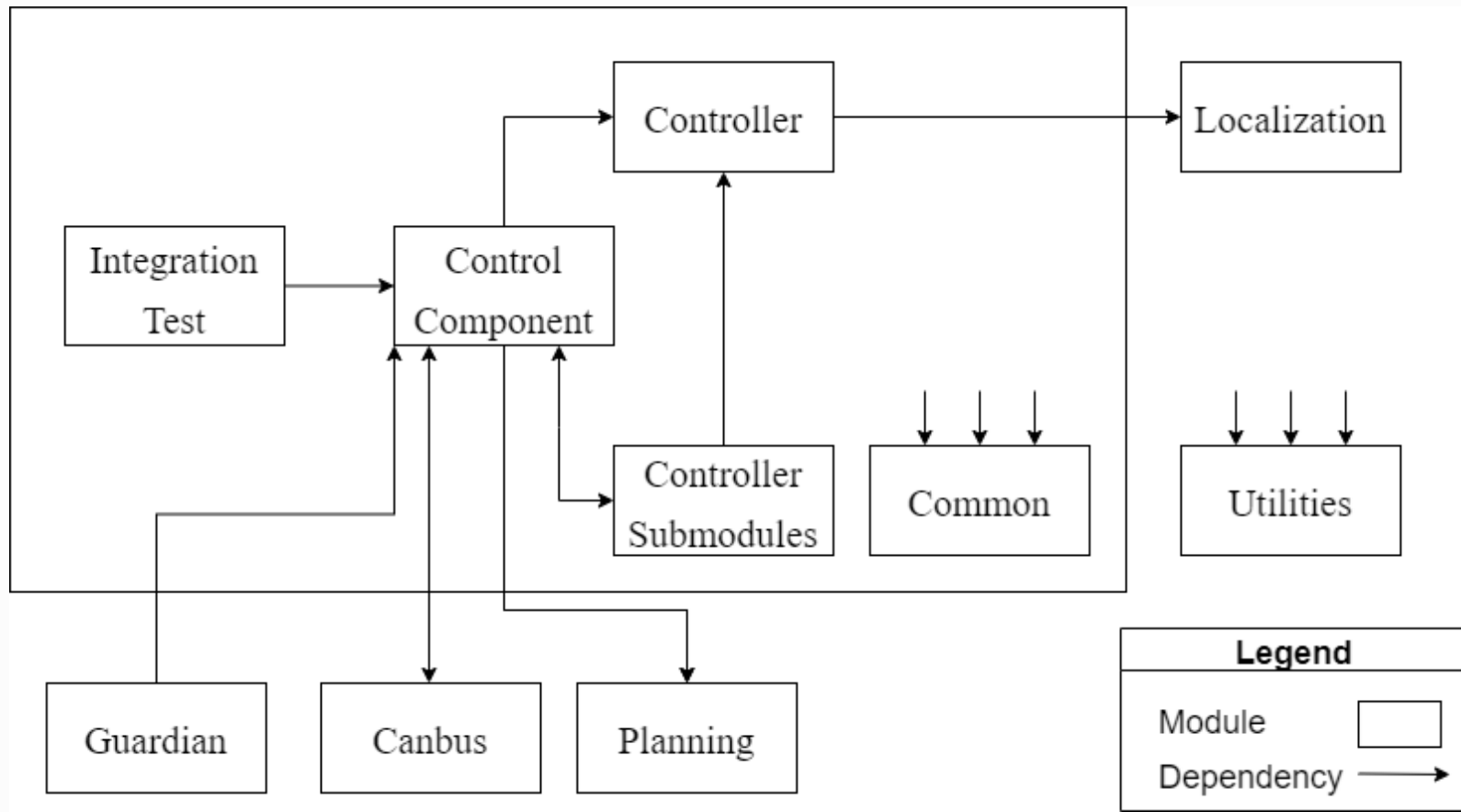
Conceptual Architecture of Control module



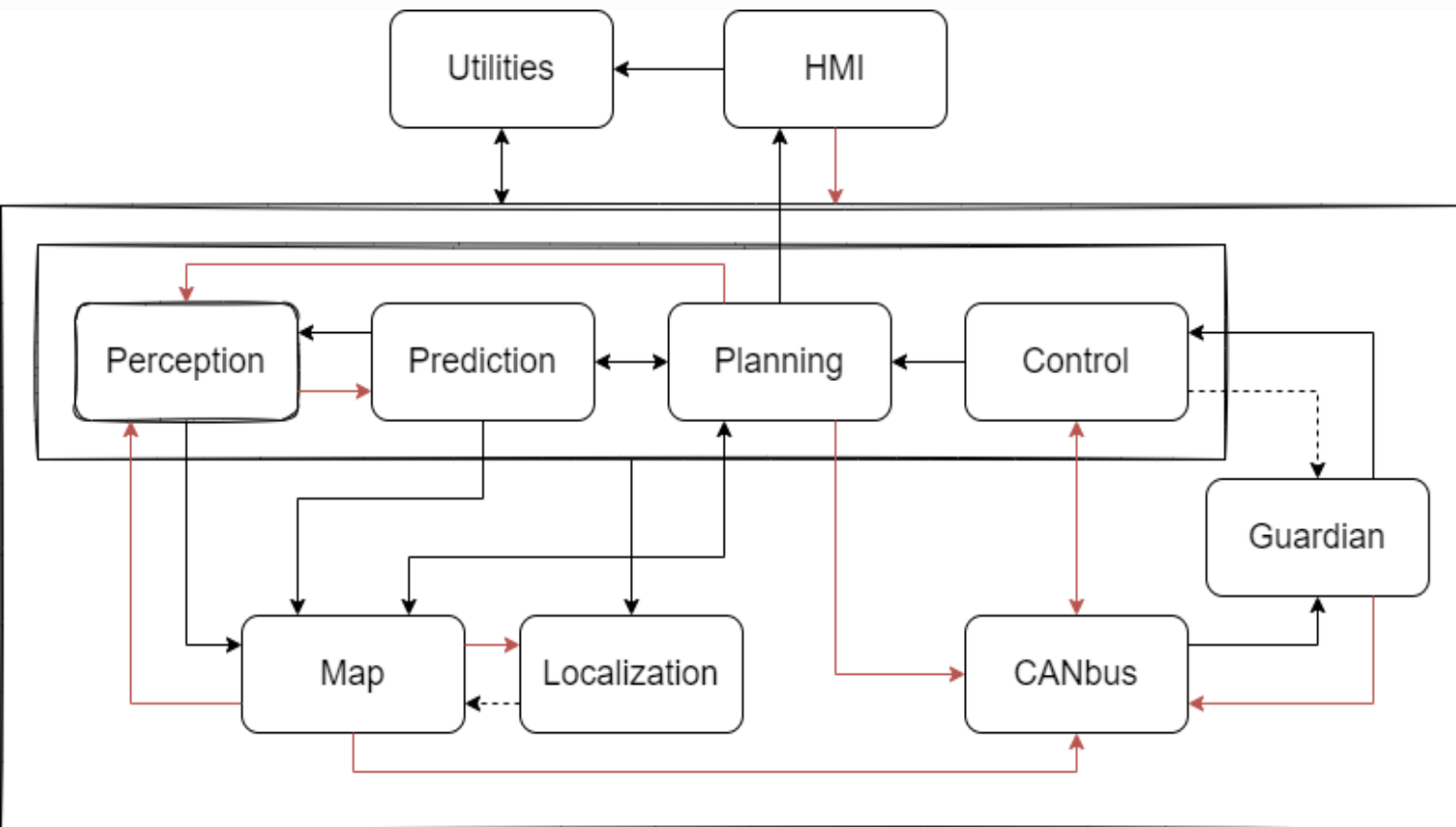
Alternative Conceptual Architecture of Control module



Concrete Architecture of Control module



Reflexion analysis of top-level architecture



Reflexion analysis of top-level architecture

Unexpected dependencies:

- Perception \rightarrow Prediction
- Planning \rightarrow Perception
- Planning \rightarrow CANbus
- Control \rightarrow CANbus
- Map \rightarrow Perception
- Map \rightarrow Localization
- Map \rightarrow CANbus
- Map \rightarrow Planning
- CANbus \rightarrow Control
- Guardian \rightarrow Control

Absent dependencies:

- Control \rightarrow Guardian
- Localization \rightarrow Map

Reflexion analysis of top-level architecture

Unexpected dependencies

Perception → Prediction: This dependency is for adding obstacle priority to the obstacle history map. This process needs an evaluator manager to achieve while the evaluator manager is a part of the prediction module. We found this connection in the file [apollo/modules/prediction/evaluator/evaluator_manager.cc](#).

CANbus → Control: CANbus module receives messages named ControlCommand from Control module. This message is sent by the control component ([apollo/modules/control/control_component.cc](#)). After it receives the message, the command is converted into ASCII then passed on to functions in the Control module.

Guardian → Control: Guardian receives the commands from the control module, and checks the malfunctions of throttle, steering target, steering rate, and safe mode status.

We found this connection in the file “/modules/guardian/guardian_component.cc”

Absent dependencies

Control → Guardian: The control module is supposed to depend on the Guardian module based on the functions they have, the Guardian should check the signals the Control module has. We investigated further into the source codes and found out that there is no such dependency which is out of our expectation.

Localization → Map: Localization provides the RTK based method which incorporates GPS LIDAR and IMU information, there should be a dependency from Localization to Map since it needs the GPS information. However, looking deep into the source codes we didn't find the dependency or relation we expected to be.

Reflexion analysis of second-level architecture (control)

Unexpected modules:

- Integration test
- Controller submodules

Absent modules:

- receiver

Unexpected dependencies:

- Integration Test → Control Component
- Control Component → Controller Submodules
- Controller Submodules → Controller
- Controller Submodules → CANbus
- Controller Submodules → Planning
- CANbus → Control Component

Reflexion analysis of second-level architecture (control)

Unexpected modules

Integration Test: The integration test is composed of 4 different tests: hybrid_transceiver test, intra transceiver test, rtps transceiver test and shm transceiver test. It is responsible for testing the validity and integrity of other submodules through an integrated test before operation.

Controller Submodules: The second unexpected composite module is the controller submodules which contain smaller 3rd-level modules. It has a pre-process submodule that receives LocalView written by the control component module and processes it, then two controller submodules will process the information sent by pre-process to generate the controlcommand.

Unexpected dependencies

Controller Component → Planning: Because we merge the receiver module to the controller component, so the controller component will directly get the ADJTrajectory from the planning module.

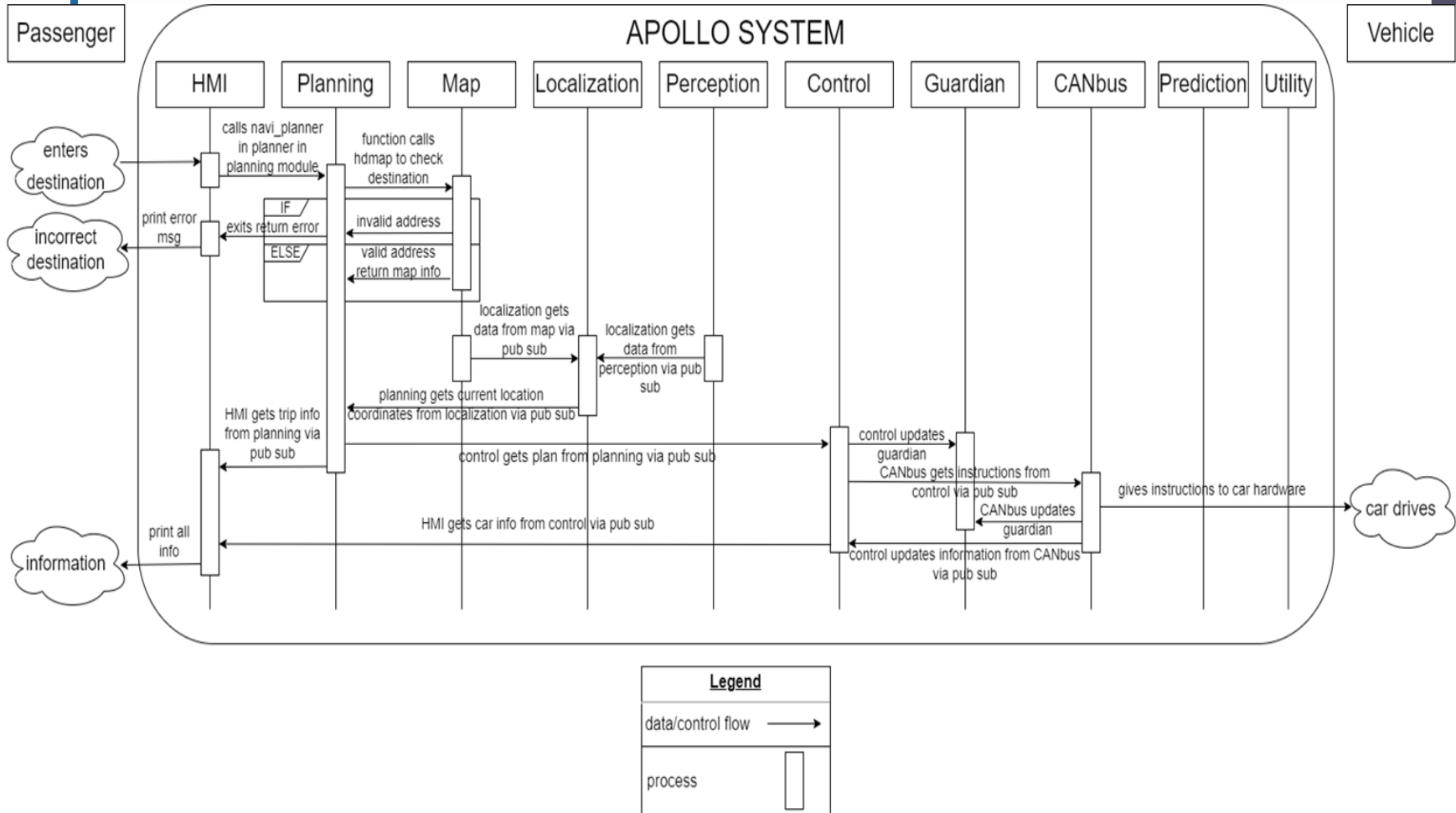
CANbus → Control Component: At the end of the control component process, it will send the ControlCommand to the cyberRT, and the CANbus will get this control command in order to control the car.

Absent modules

Receiver: The receiver module is missing from our concrete architecture since its functions are integrated into the 'controller submodules' module.

Use Case: Apollo plans route and starts trip according to address provided by passenger.

12



Concurrency & Team Issues & Lesson Learned

Limitations & Conclusion



Thank you