Massachusetts
Institute of
Technology

STPA Applied to Autonomous Vehicles

Dr. John Thomas (MIT)
jthomas4@mit.edu

Jeff Stafford (Renesas)

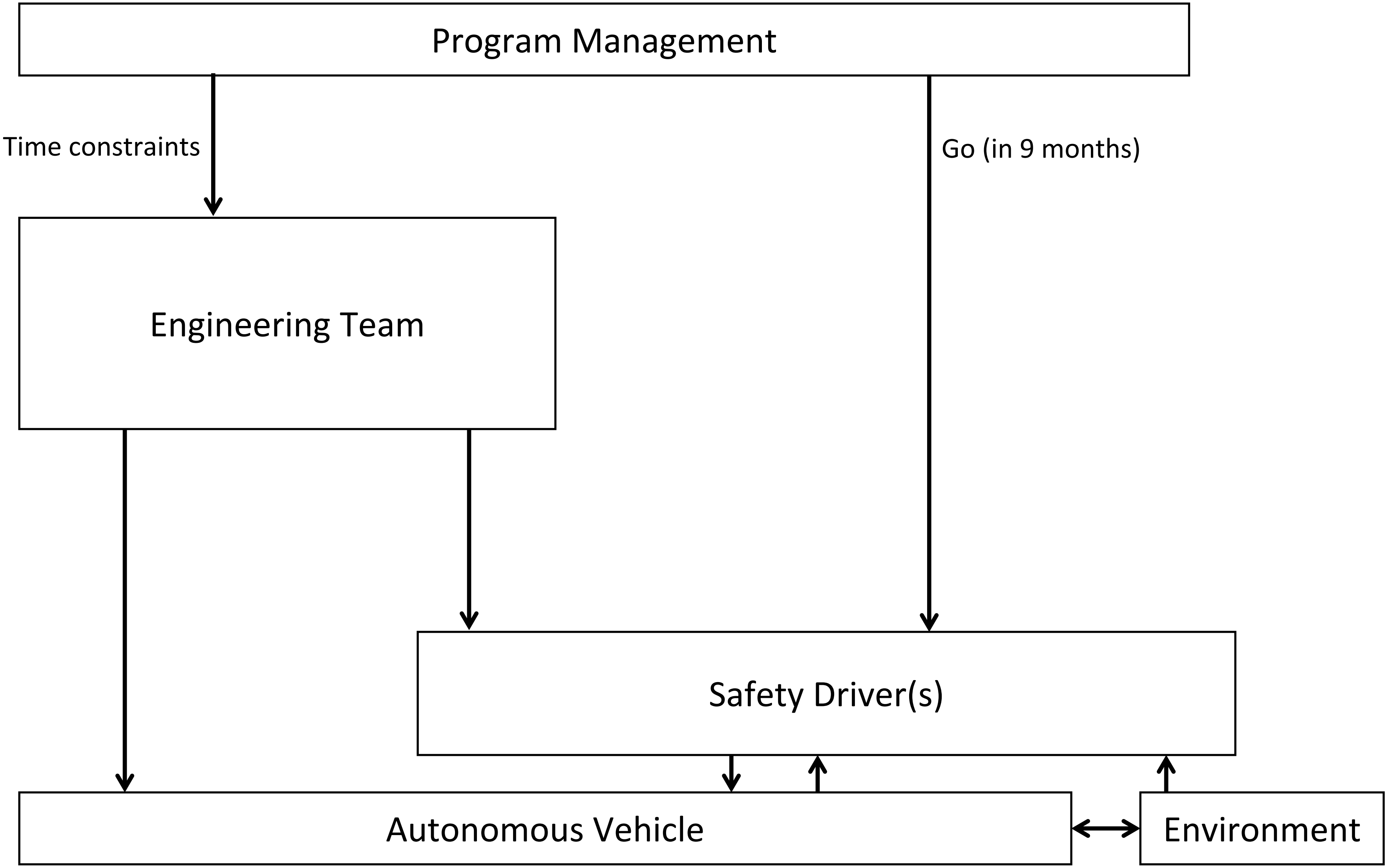
Shaun Mooney (Codethink)

Matthew Green (Codethink)

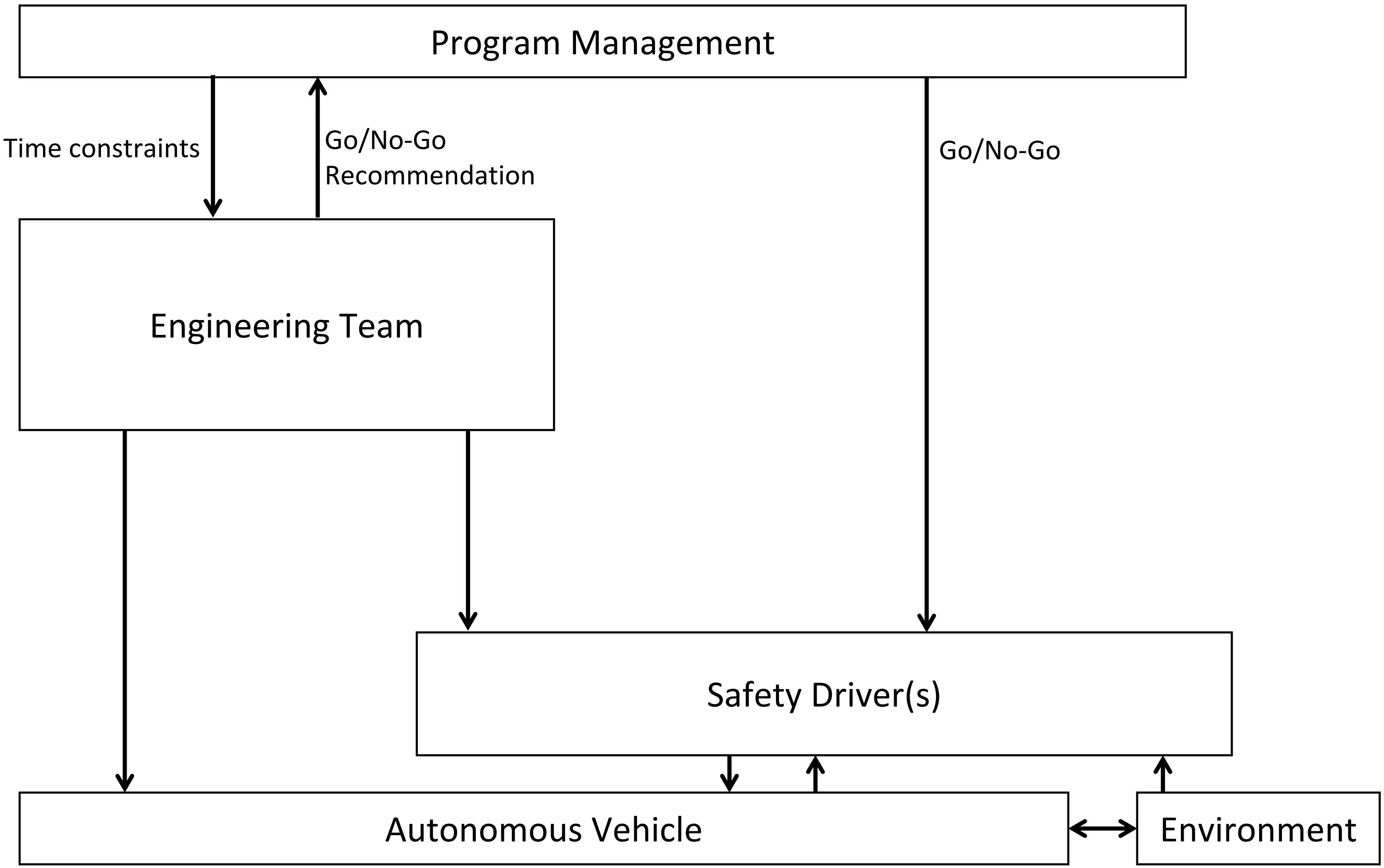
Original Company Plan

- Goal: Demonstrate self-driving car on public roads
- Use Baidu's Apollo software for self-driving functions
- Company is convinced that a systems approach to safety is required
- Decision: Use state-of-the-art STPA to demonstrate due diligence

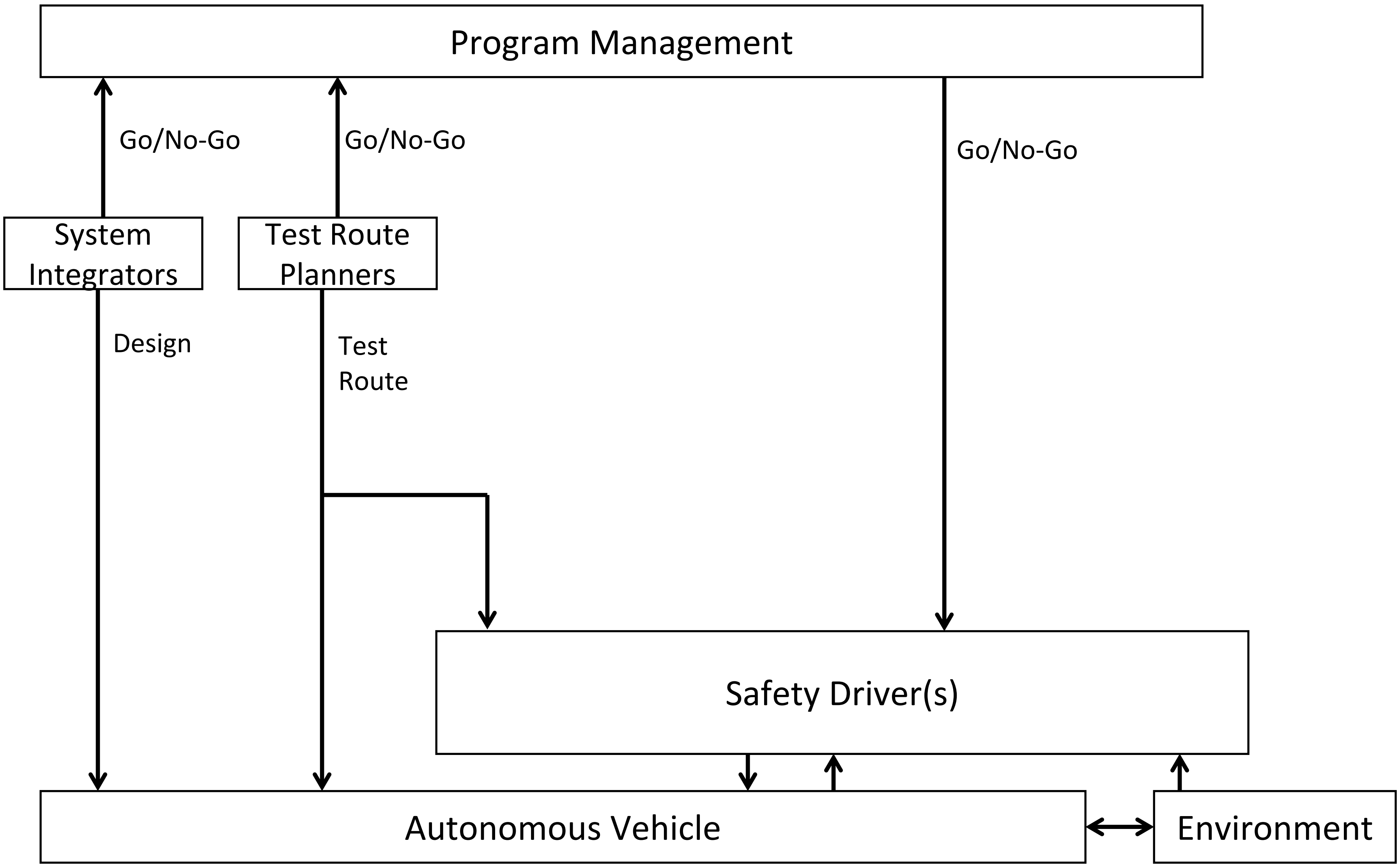
Control Structure: Level 1



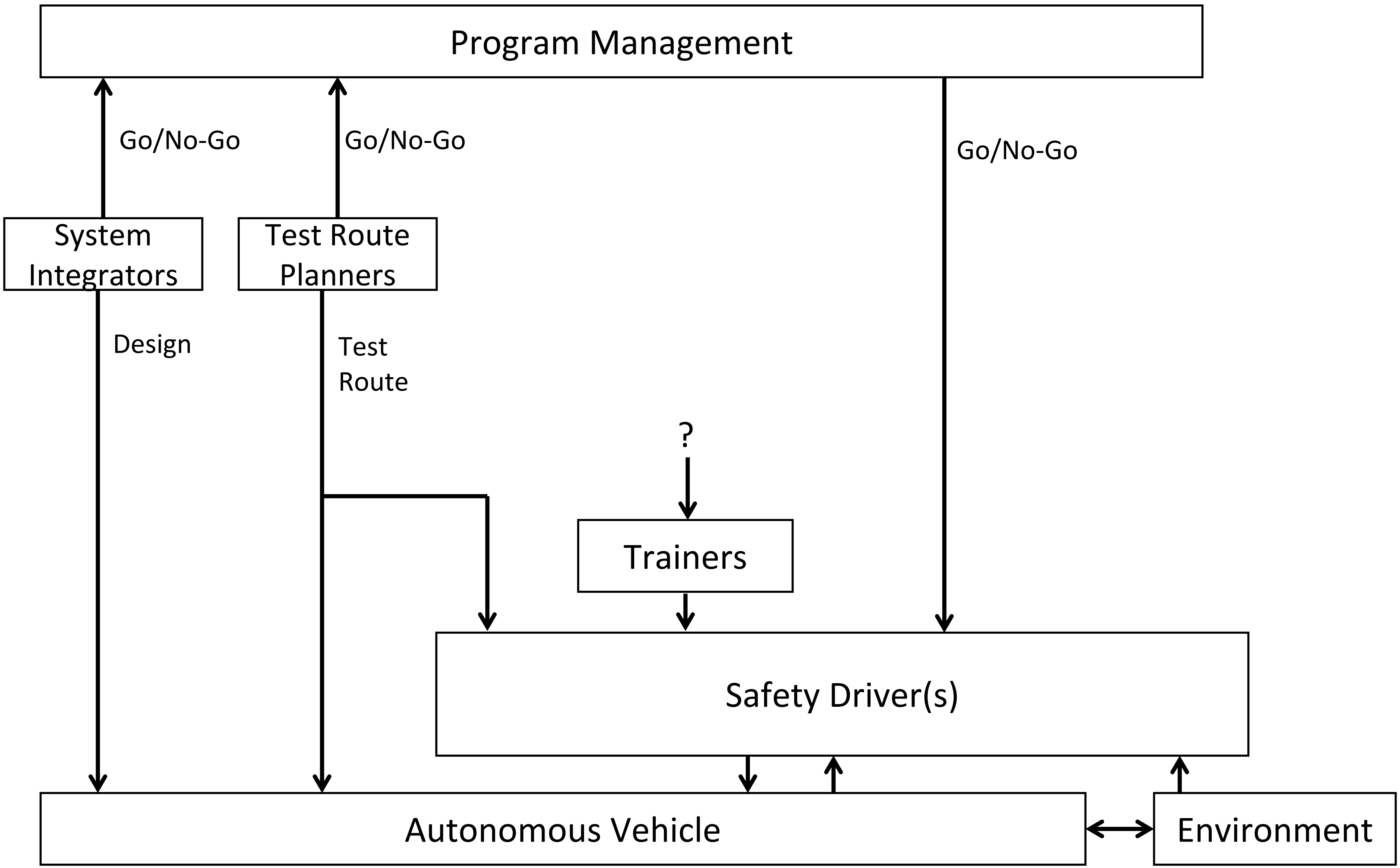
Control Structure: Level 1



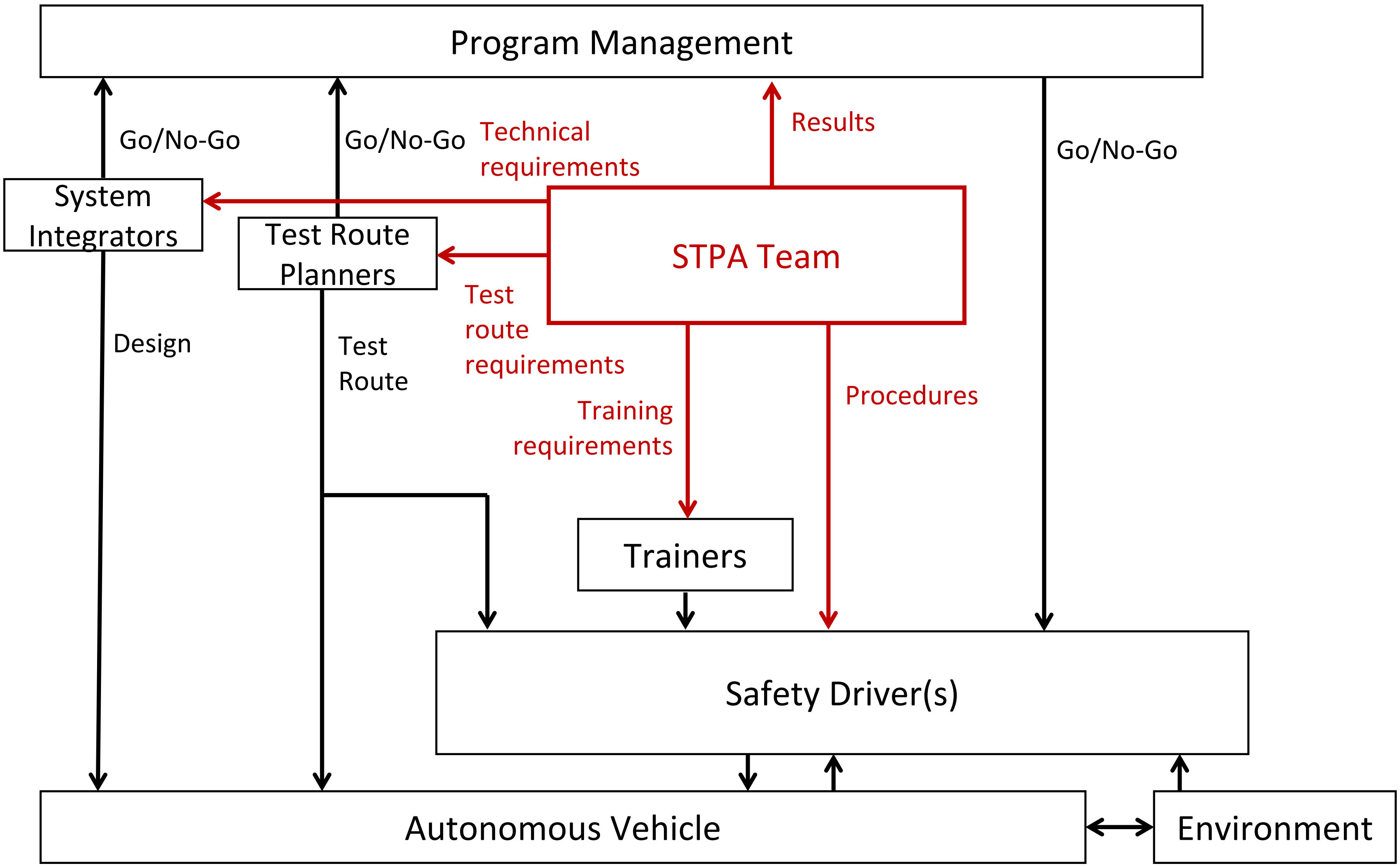
Control Structure: Level 1



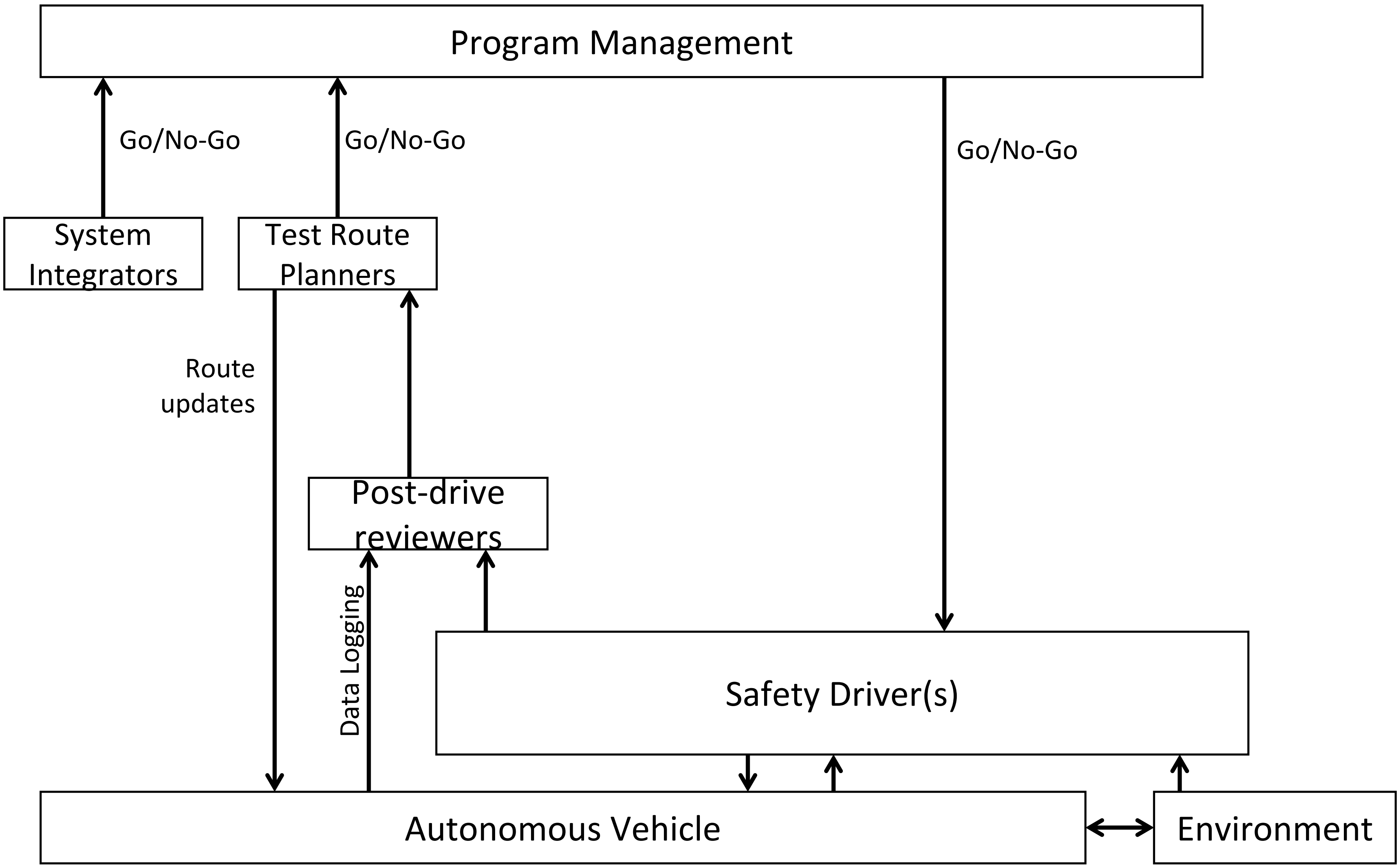
Control Structure: Level 1



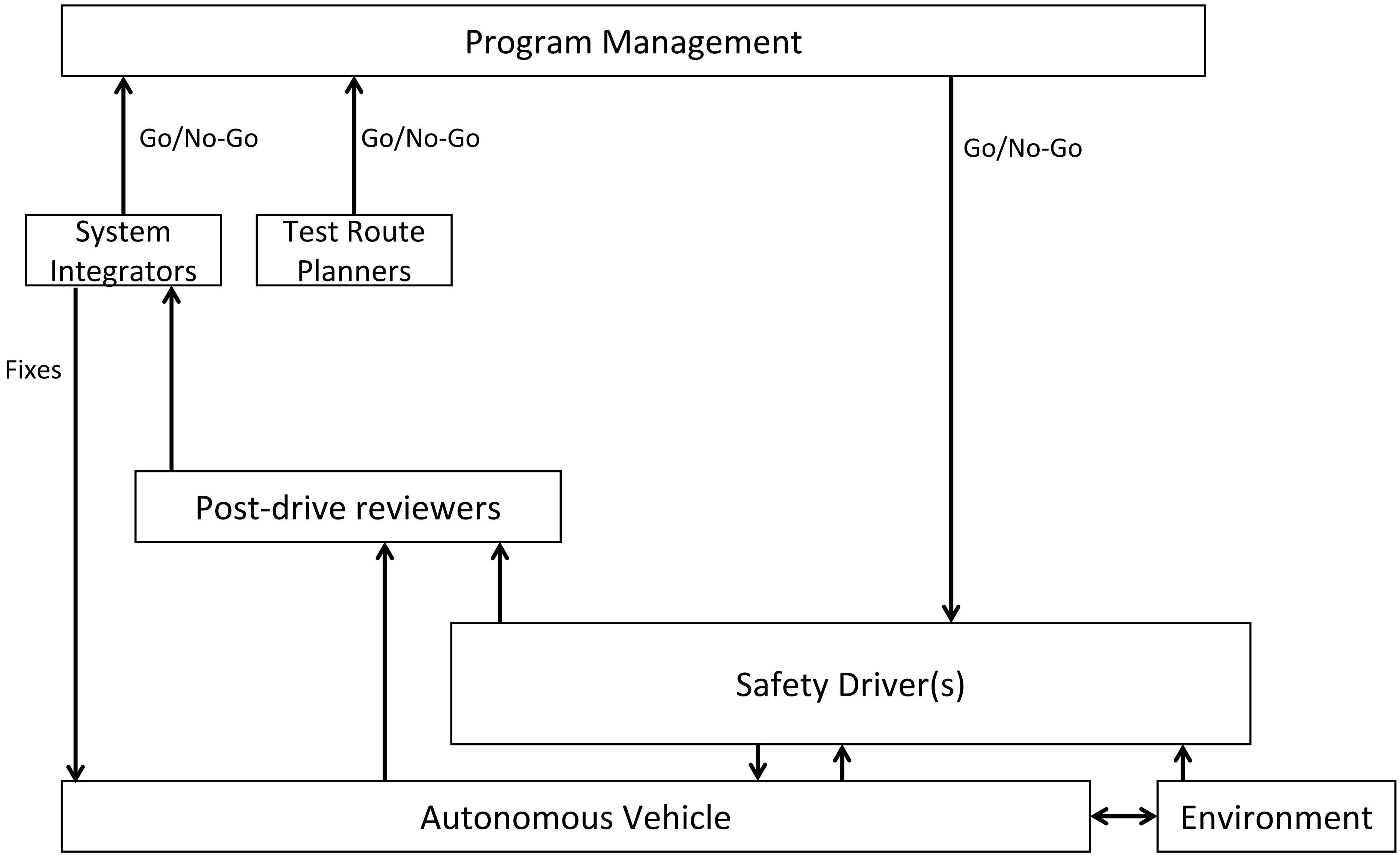
Control Structure: Level 1



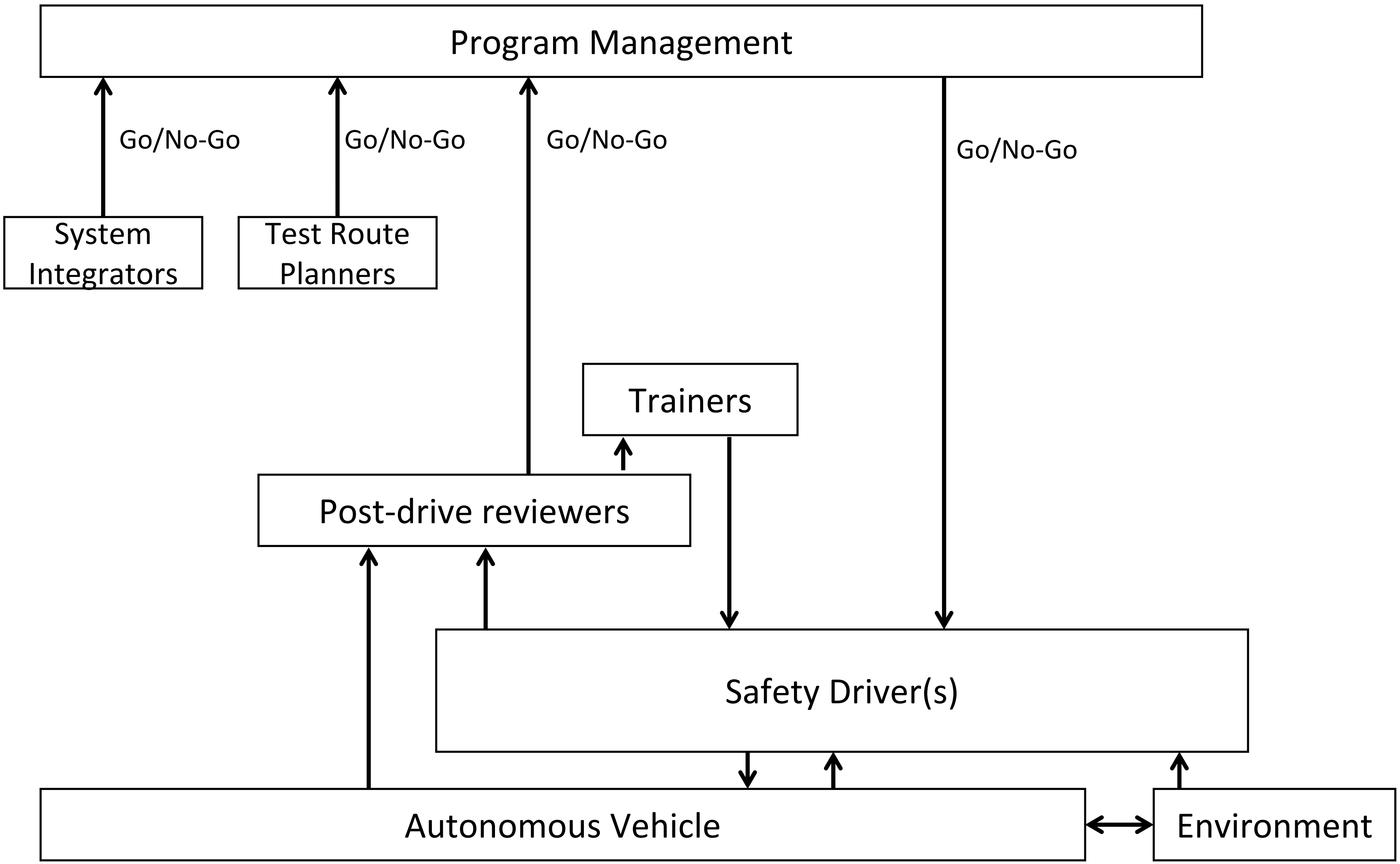
Control Structure: Level 1



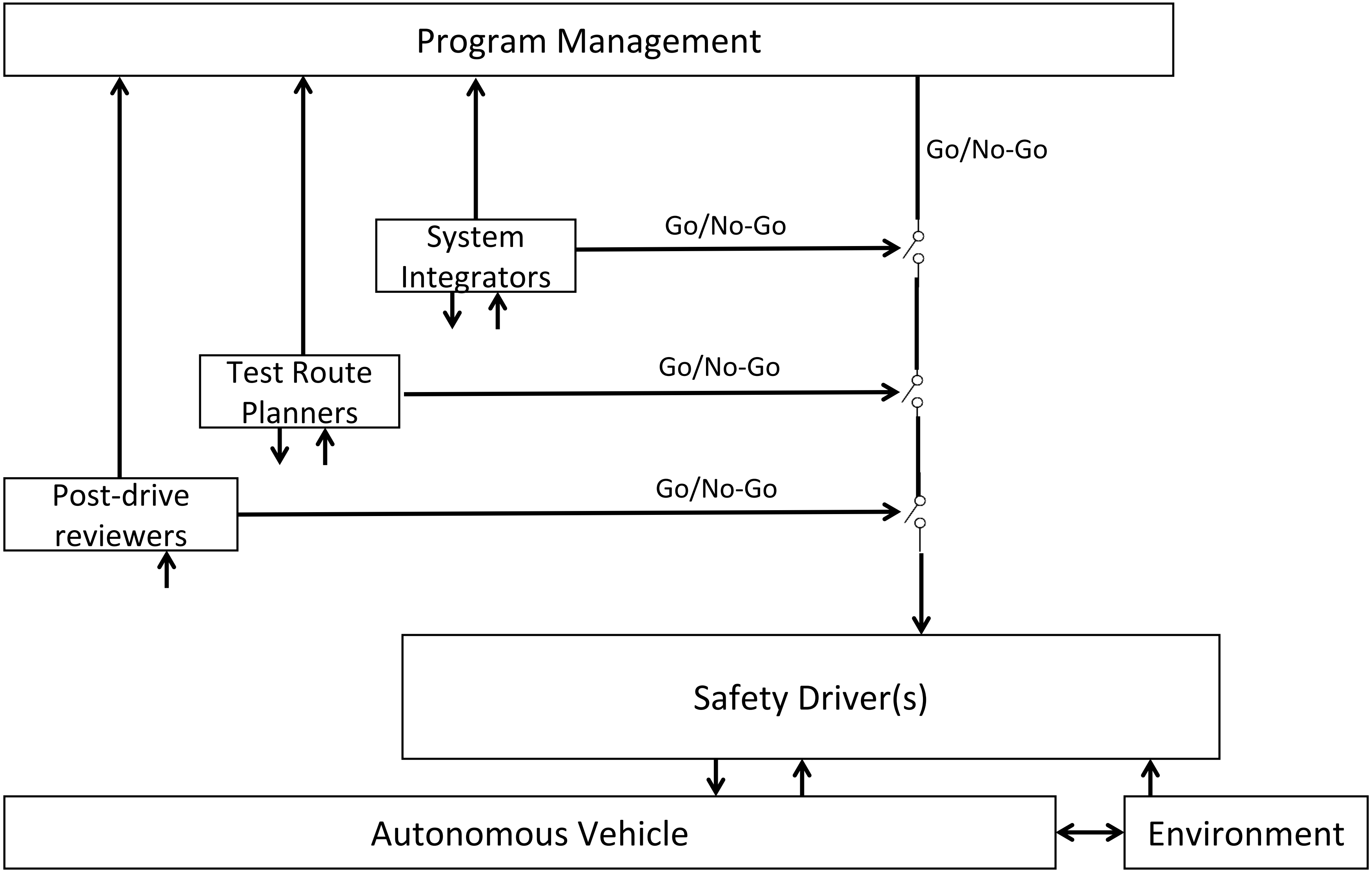
Control Structure: Level 1



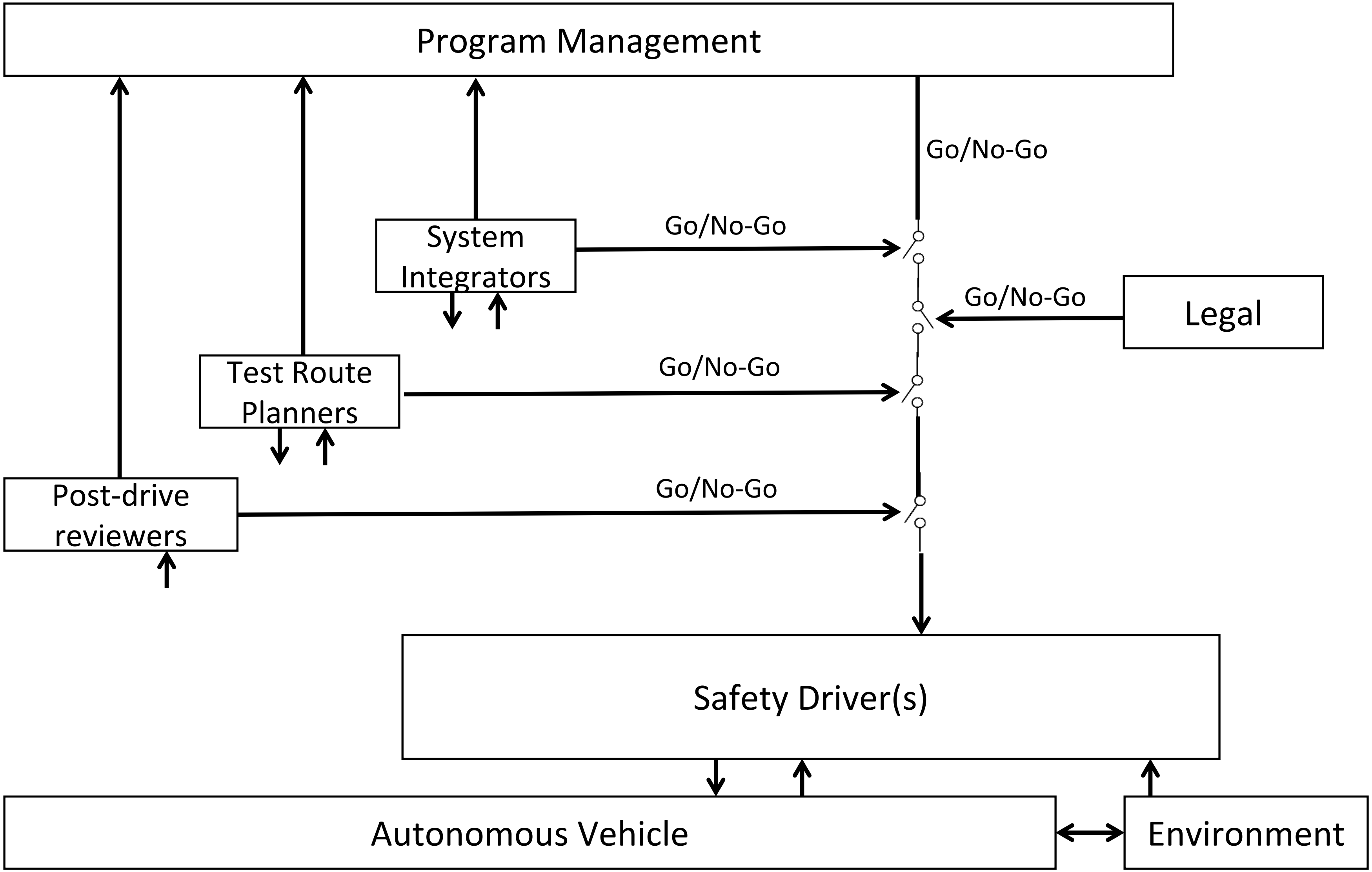
Control Structure: Level 1



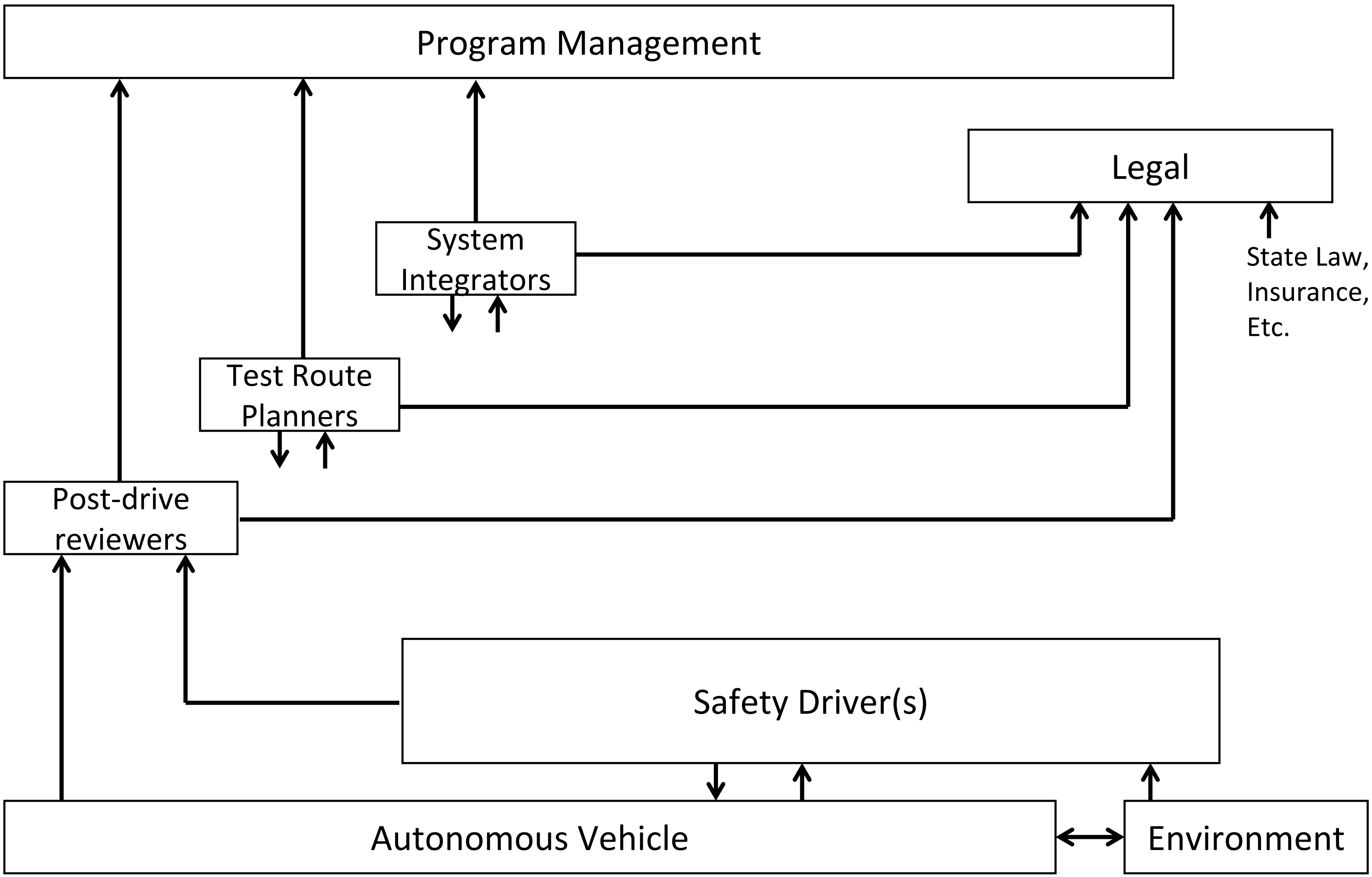
Control Structure: Level 1



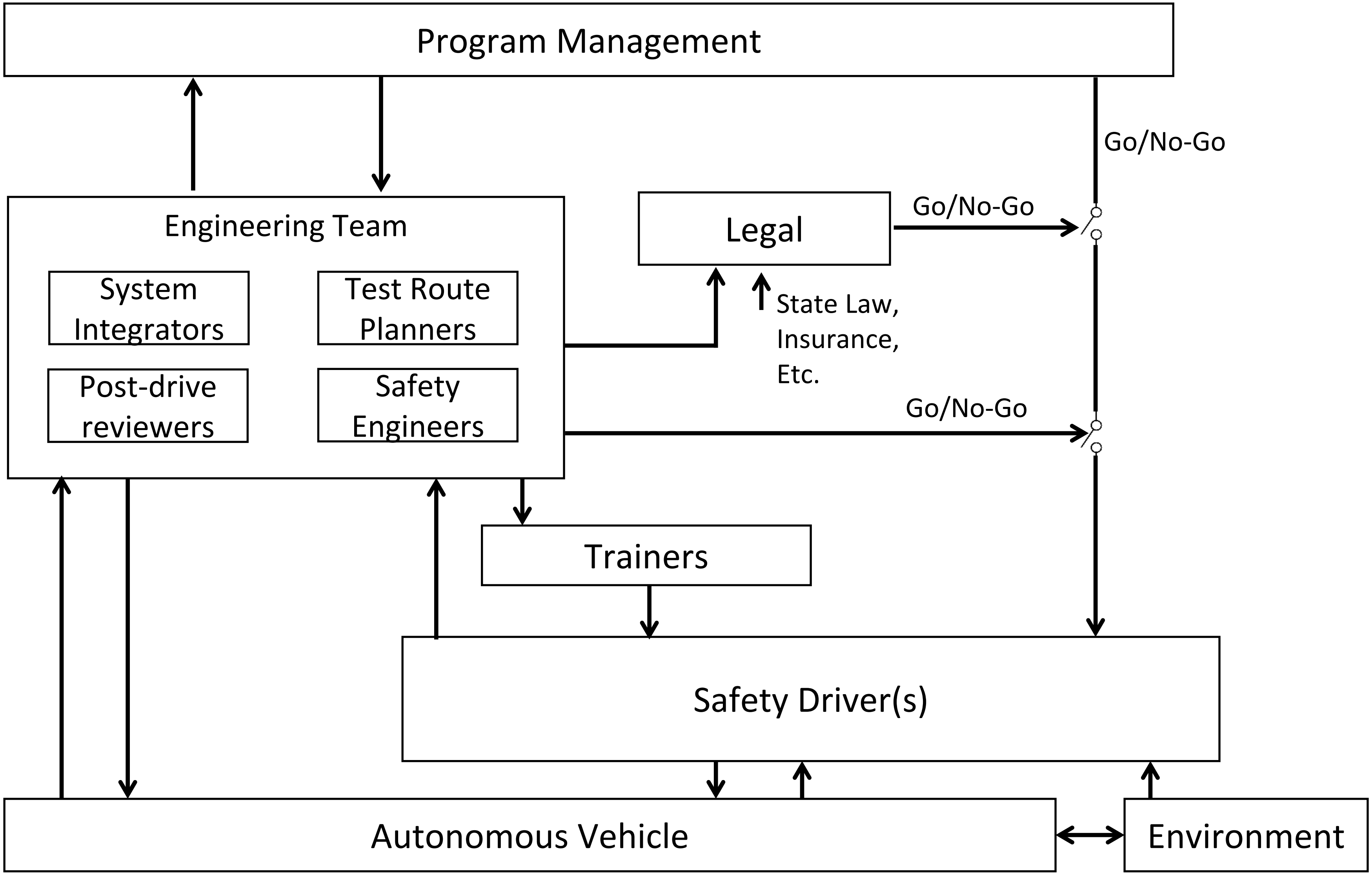
Control Structure: Level 1



Control Structure: Level 1



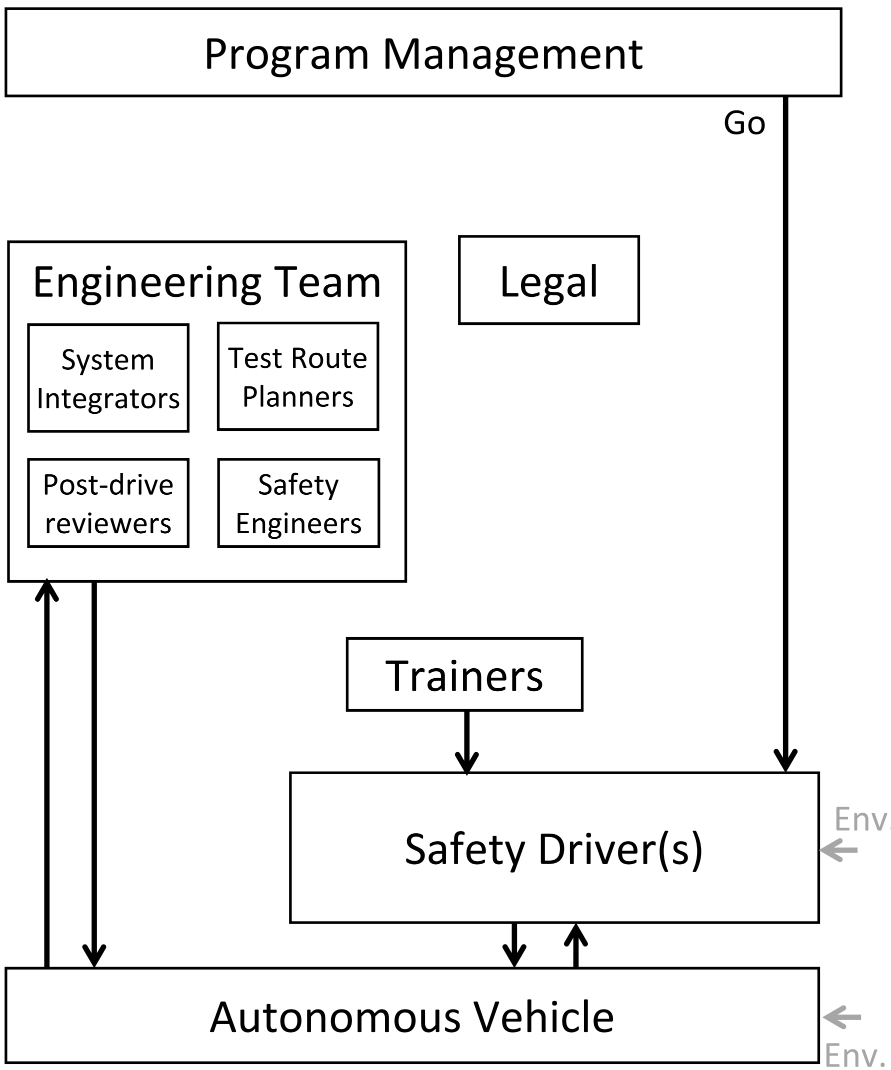
Control Structure: Level 1



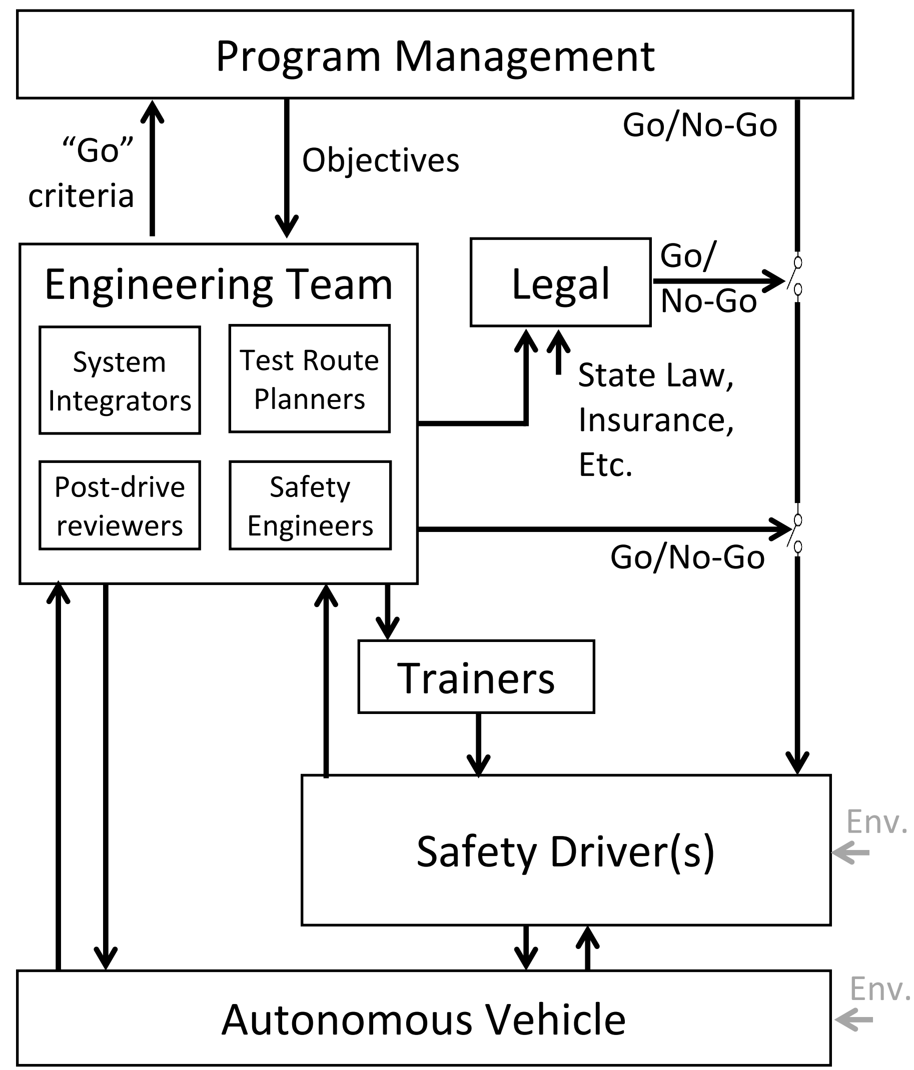
Massachusetts Institute of Technology

Control Structure

Before

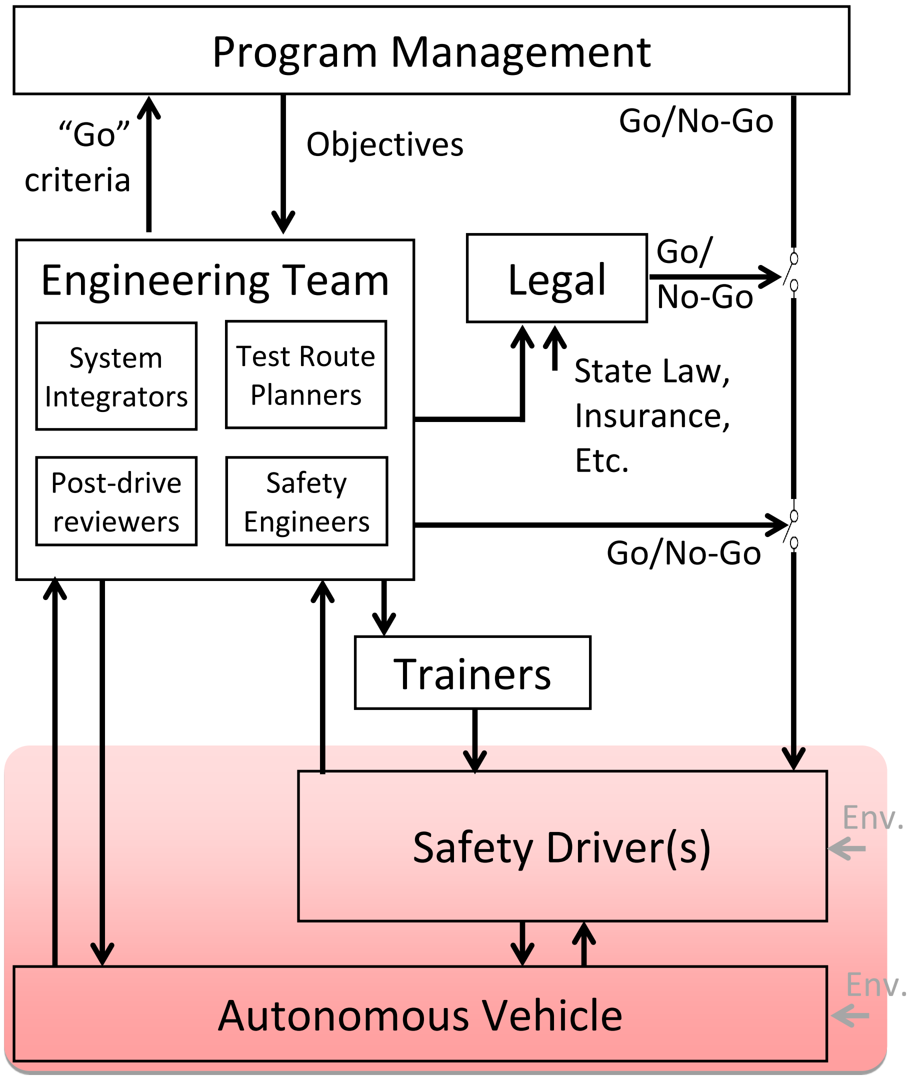


After

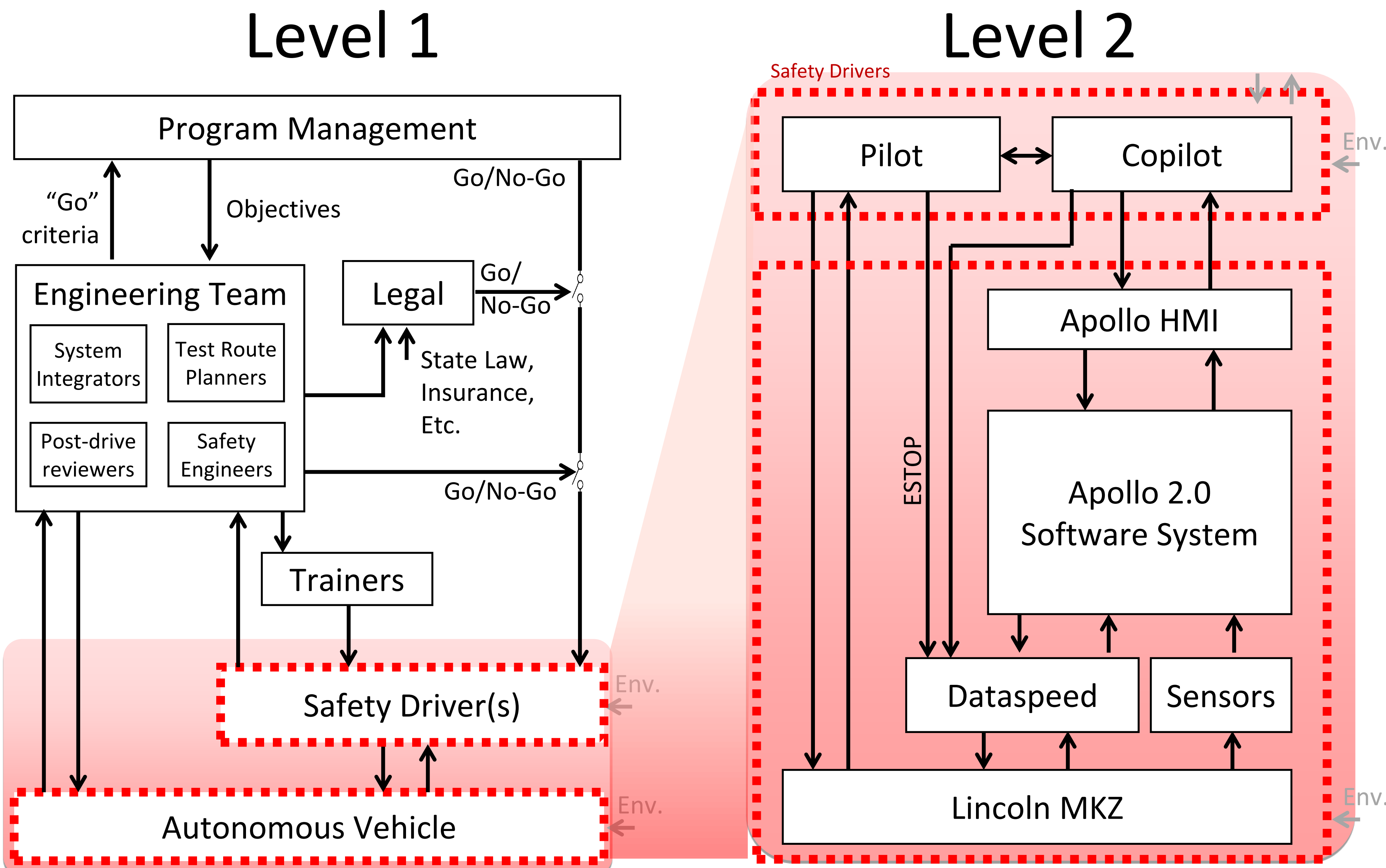


Control Structure Refinement

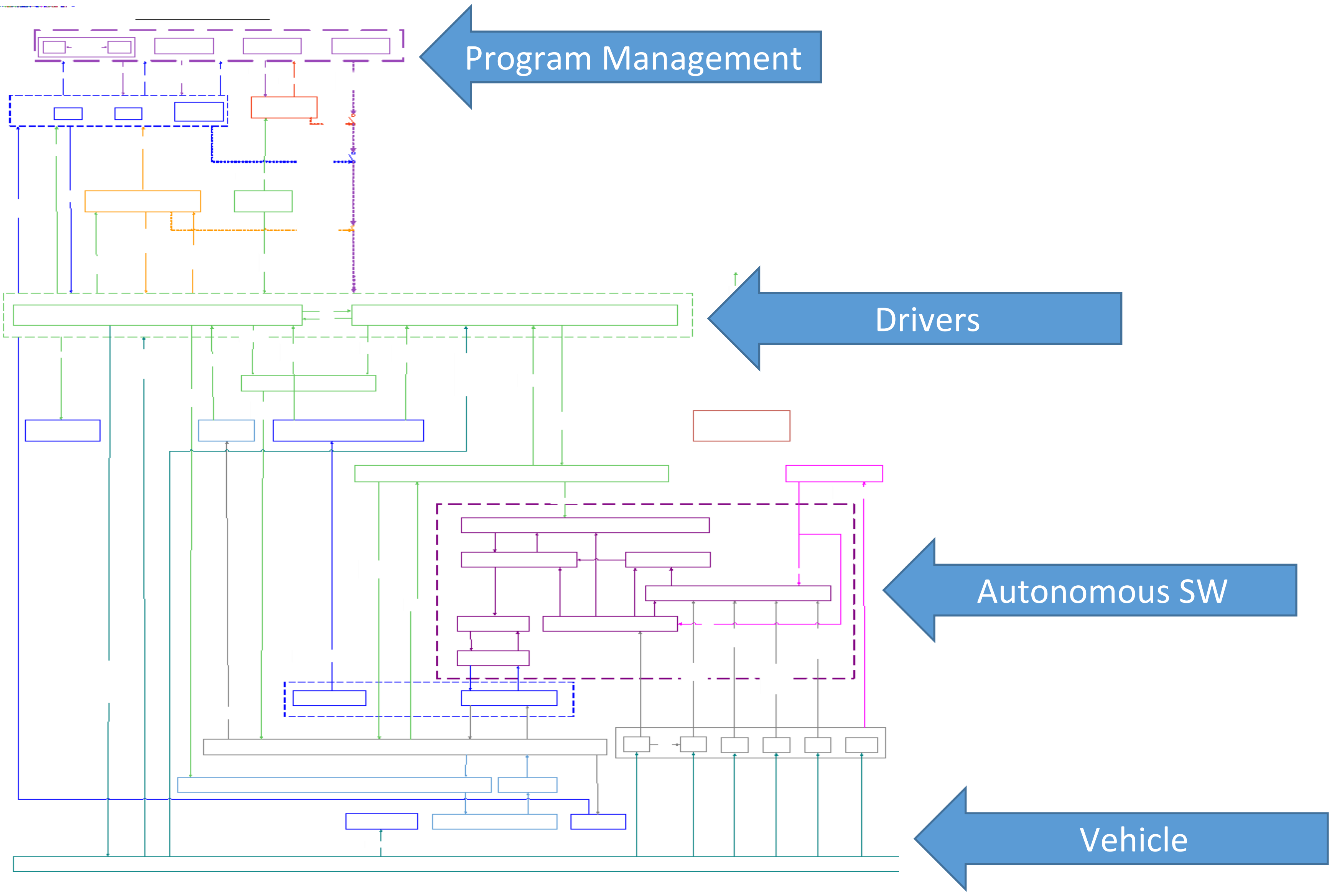
Level 1



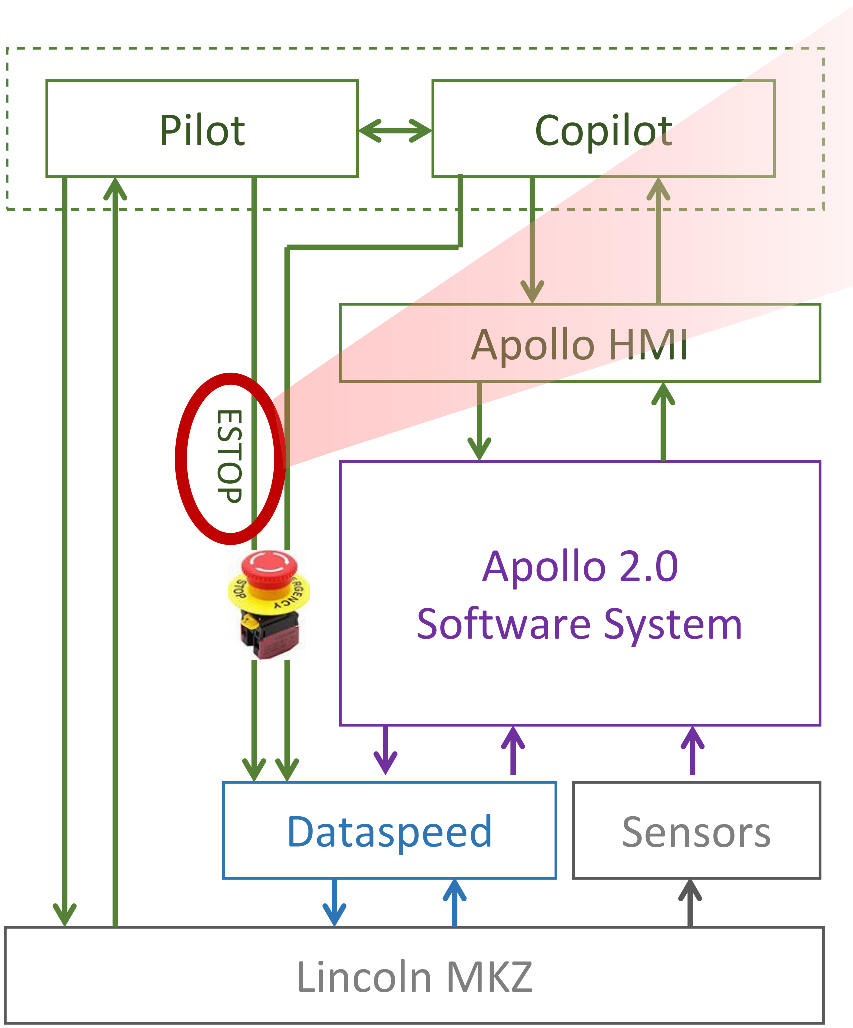
Control Structure Refinement



Complete Control Structure



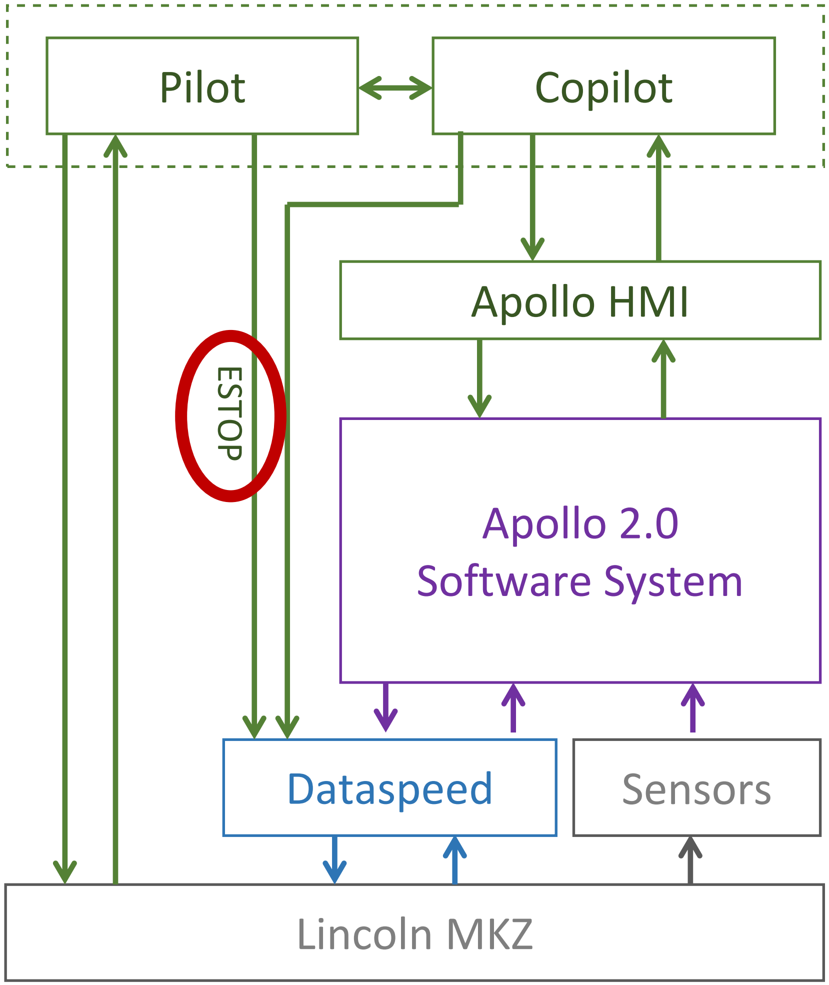
Unsafe Control Actions



UCA: Pilot does not provide ESTOP when autonomy is providing excessive throttle

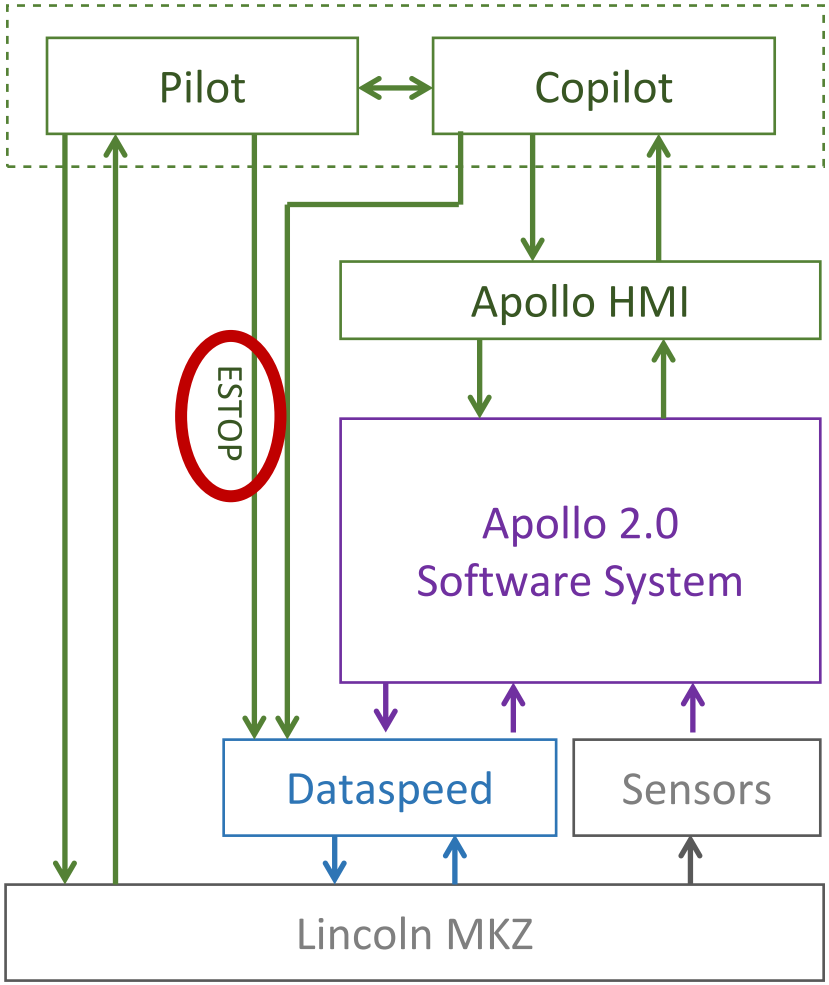
UCA :=	UCA:
<Source Controller> →	Pilot
<Type> →	does not provide
<Control Action> →	ESTOP
when	when
<Context> →	Autonomy providing excessive throttle

Human Interactions



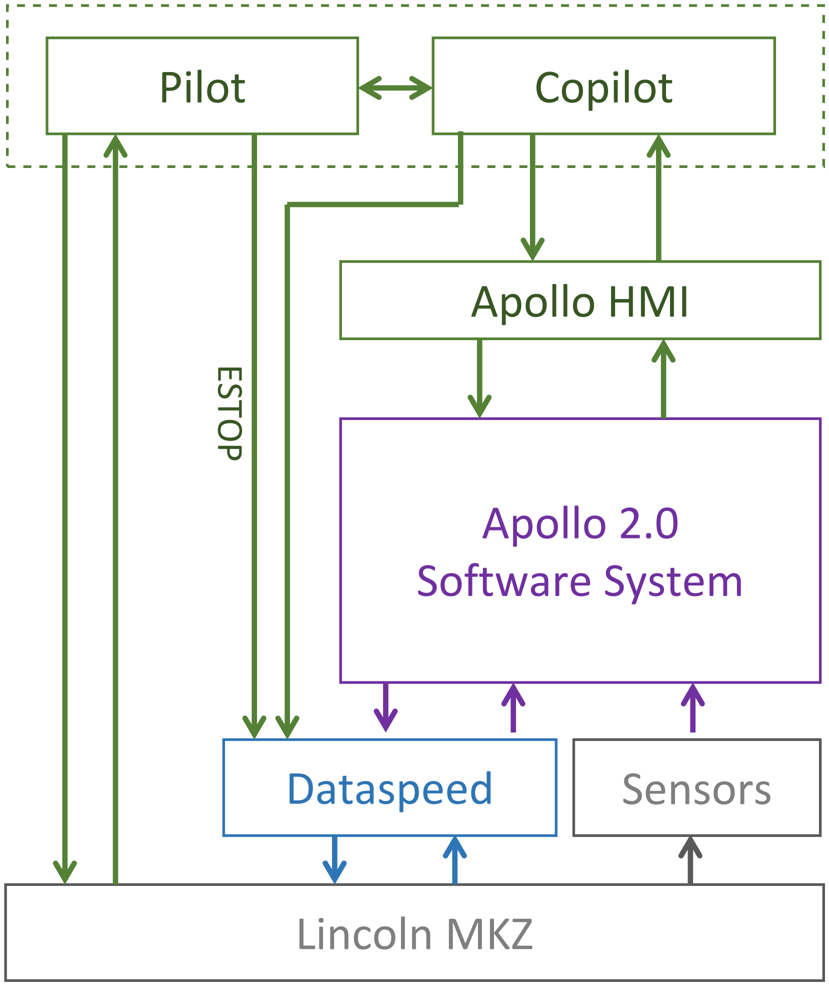
- UCA: Pilot does not provide ESTOP when autonomy is providing excessive throttle
  - PM: Pilot believes autonomy was disabled due to manual braking cmds…
- Generated Requirement
  - Dataspeed must override all Apollo cmds when driver applies brake
- Existing Design/Requirements can cause this!
  - <22% braking will not override
  - Braking override independent from steering override
  - Will ignore driver braking overrides if Apollo sends IGNORE/CLEAR

Human Interactions



- UCA: Pilot does not provide ESTOP when autonomy is providing excessive throttle
  - PM: Pilot believes autonomy was disabled due to manual braking cmds…
- Generated Requirement
  - Dataspeed must override all Apollo cmds when driver applies brake
- Existing Design/Requirements can cause this!
  - <22% braking will not override
  - Braking override independent from steering override
  - Will ignore driver braking overrides if Apollo sends IGNORE/CLEAR

Results: Requirements

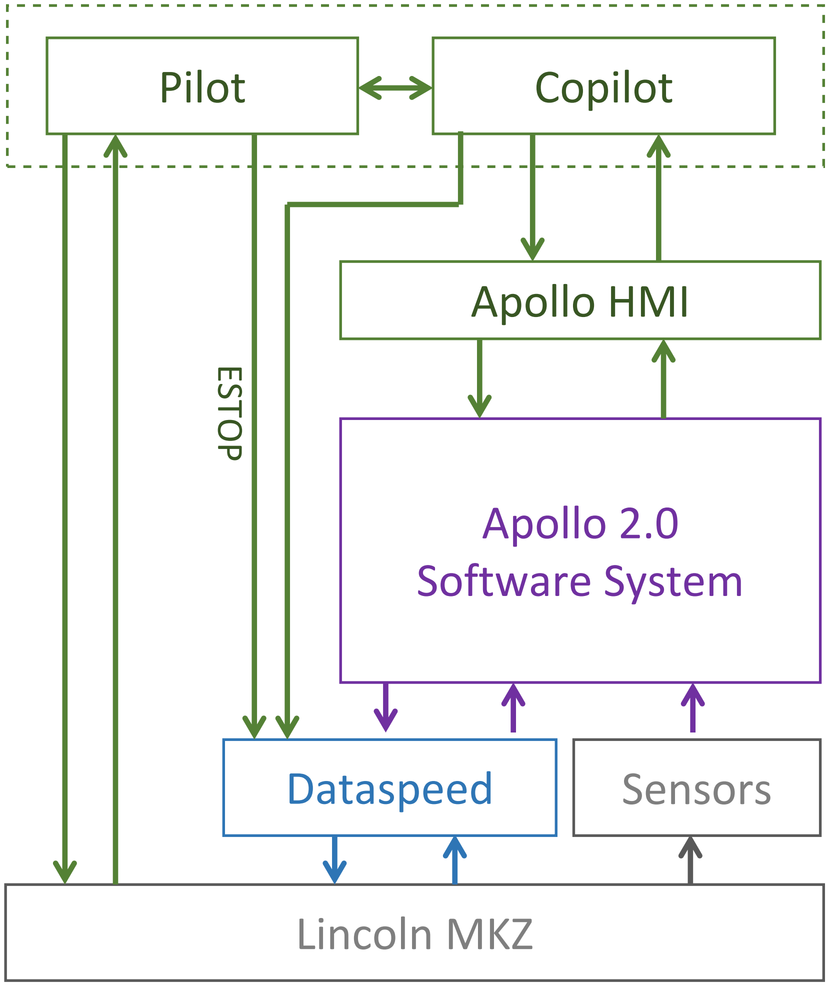


Generated Possible Requirements

- Dataspeed must override Apollo (all channels) when driver applies brake
- Apollo must not override driver (must not provide throttle, IGNORE/CLEAR, etc.
- Pilot/Copilot must be notified when manual commands do not result in automation override
- Pilot/Copilot test track training must include cases in which manual commands do not result in automation override (e.g. <22%)
- Post-drive review must identify any cases in which manual commands do not result in automation override
- Public road testing approval must stop if operation encounters manual commands that do not result in automation override (assumption violated)
- …

Massachusetts Institute of Technology

Results: Requirements



Generated Possible Requirements

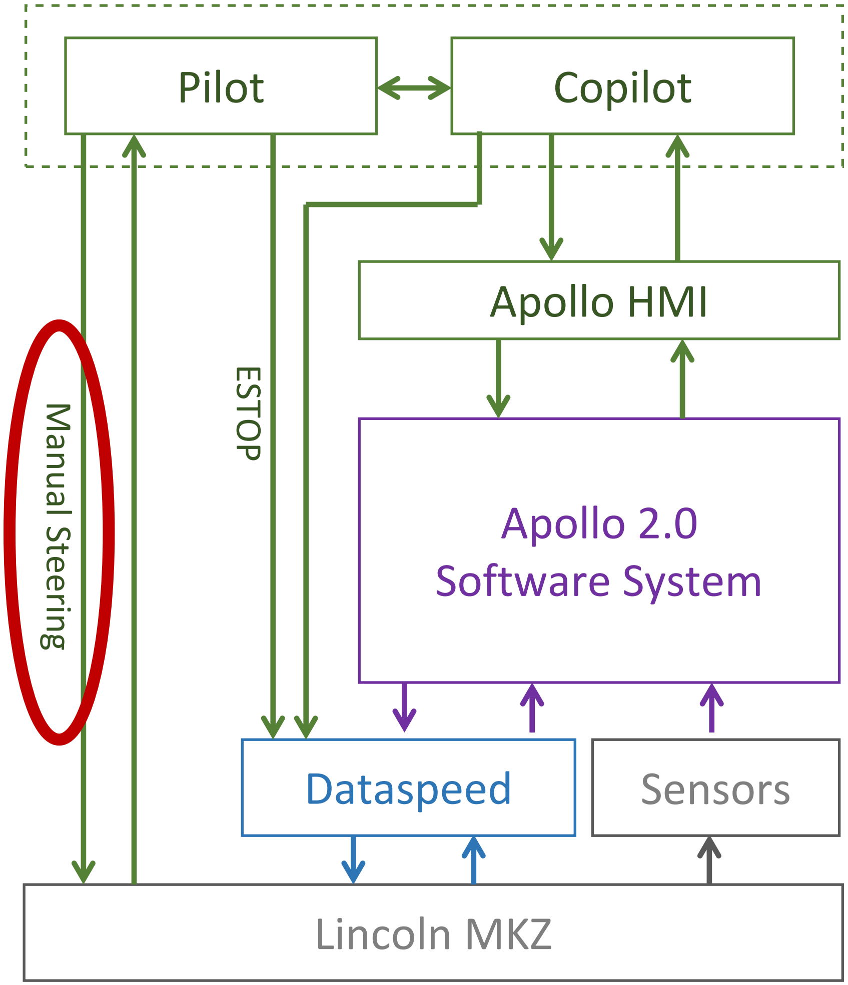
- Dataspeed must override Apollo (all channels) when driver applies brake
- Apollo must not override driver (must not provide throttle, IGNORE/CLEAR, etc.

How?

- Pilot/Copilot must be notified when manual commands do not result in automation override
- Pilot/Copilot test track training must include cases in which manual commands do not result in automation override (e.g. <22%)
- Post-drive review must identify any cases in which manual commands do not result in automation override
- Public road testing approval must stop if operation encounters manual commands that do not result in automation override (assumption violated)
- …

Massachusetts Institute of Technology

Human Interactions



UCA: Pilot provides **manual steering** too late after autonomous mode exits

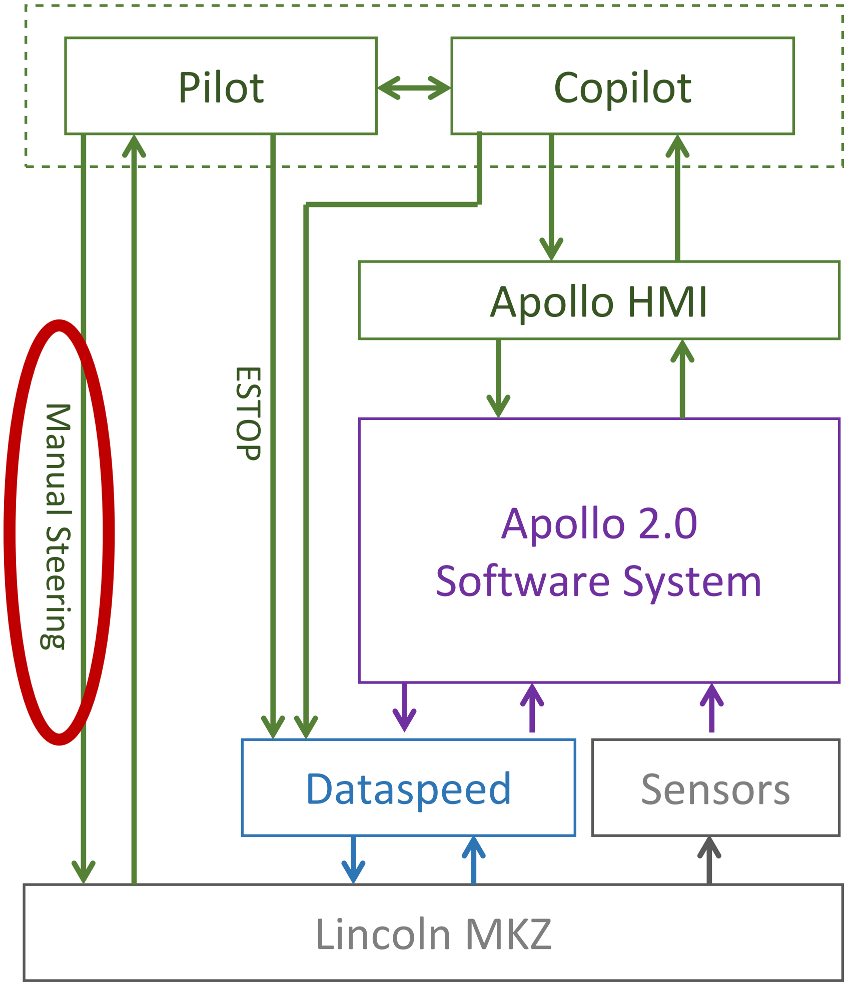
- Pilot believes vehicle is in autonomous mode
- Vehicle exits autonomous mode unexpectedly (e.g. fault occurs, ESTOP applied)
  - ESTOP applied by copilot during turn
  - …

Enforced?

Generated potential requirements

- ESTOP must not cause sudden steering angle changes
- Pilot/copilot must have advance indication before autonomous mode ends
- …

Human Interactions



New question: Does existing system enforce this?

Answer: ESTOP activation results in immediate "return to position" torque.

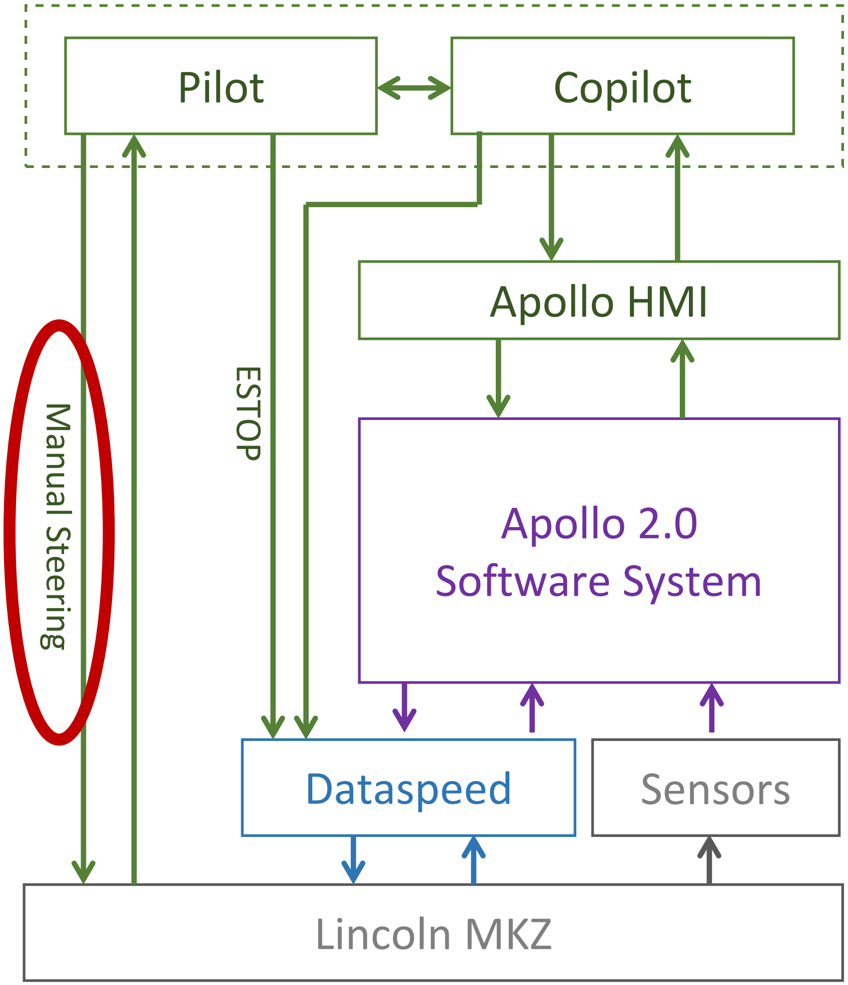
- Can't change.

Enforced?

Generated potential requirements

- ESTOP must not cause sudden steering angle changes
- Pilot/copilot must have advance indication before autonomous mode ends
- ...

Human Interactions



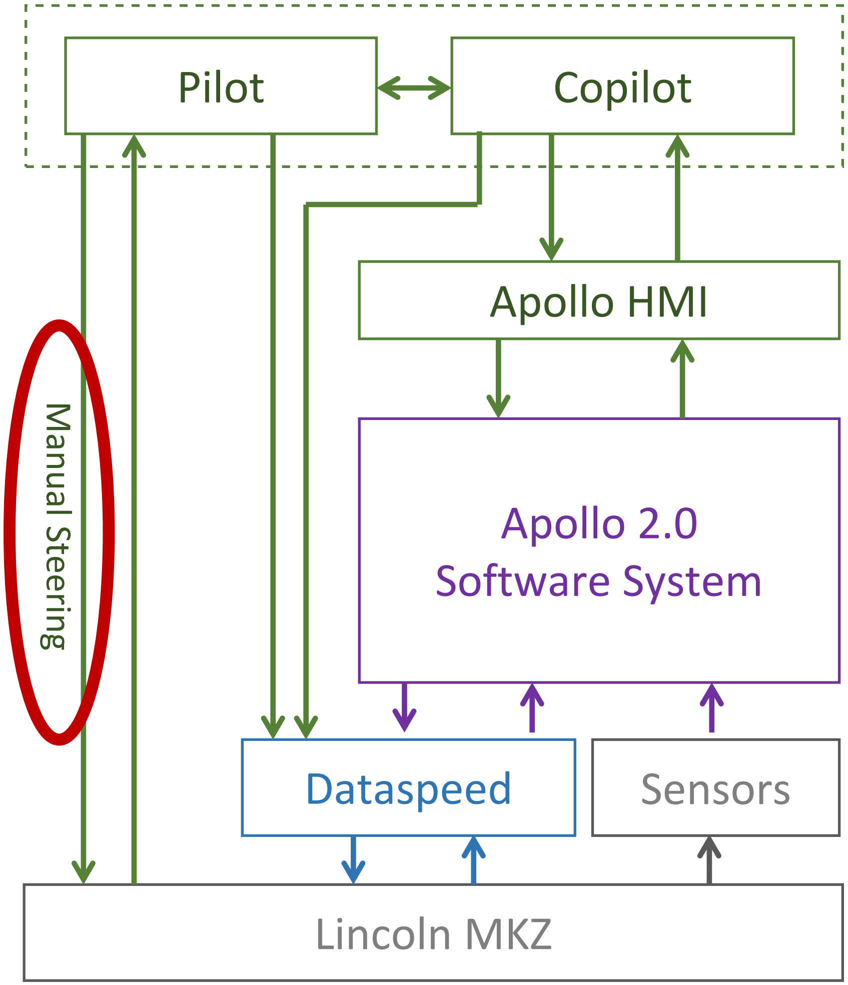
UCA: Pilot provides **manual steering** too late after autonomous mode exits

- Pilot believes vehicle is in autonomous mode
- Vehicle exits autonomous mode unexpectedly (e.g. fault occurs, ESTOP applied)
  - ESTOP applied by copilot during turn
  - …

Existing Design/Requirements will cause this!

- ESTOP designed to instantly remove power: "pull the plug"
- Results in immediate "return to position" steering wheel torque.
- Not configurable, can't change.

Results: Requirements



New question: Does existing system enforce this?

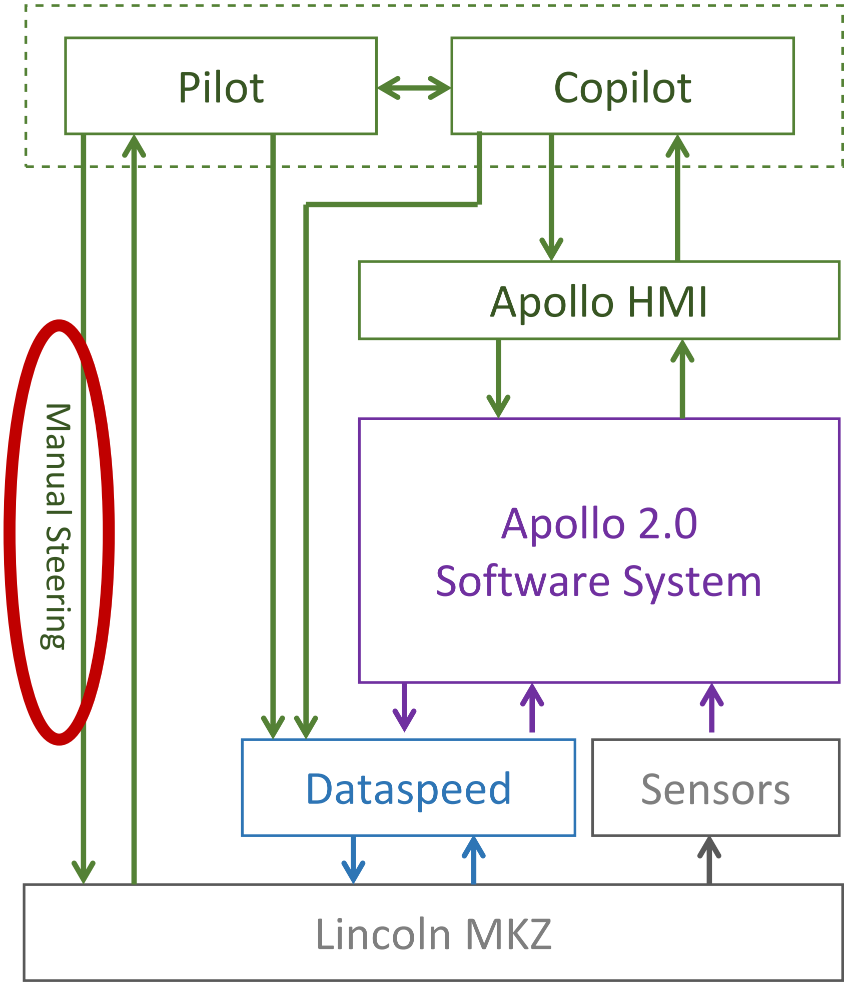
Answer: ESTOP activation results in immediate "return to position" behavior.

- Can't change

Resulting potential requirements

- **ESTOP must not cause sudden steering angle changes**
- **Pilot/copilot must have advance indication before autonomous mode ends**
- Copilot must confirm Pilot hands on wheel before providing ESTOP.
- Test track training must include copilot activation of ESTOP
- Test track training must include ESTOP activation during turns
- ...

Results: Requirements



New question: Does existing system enforce this?

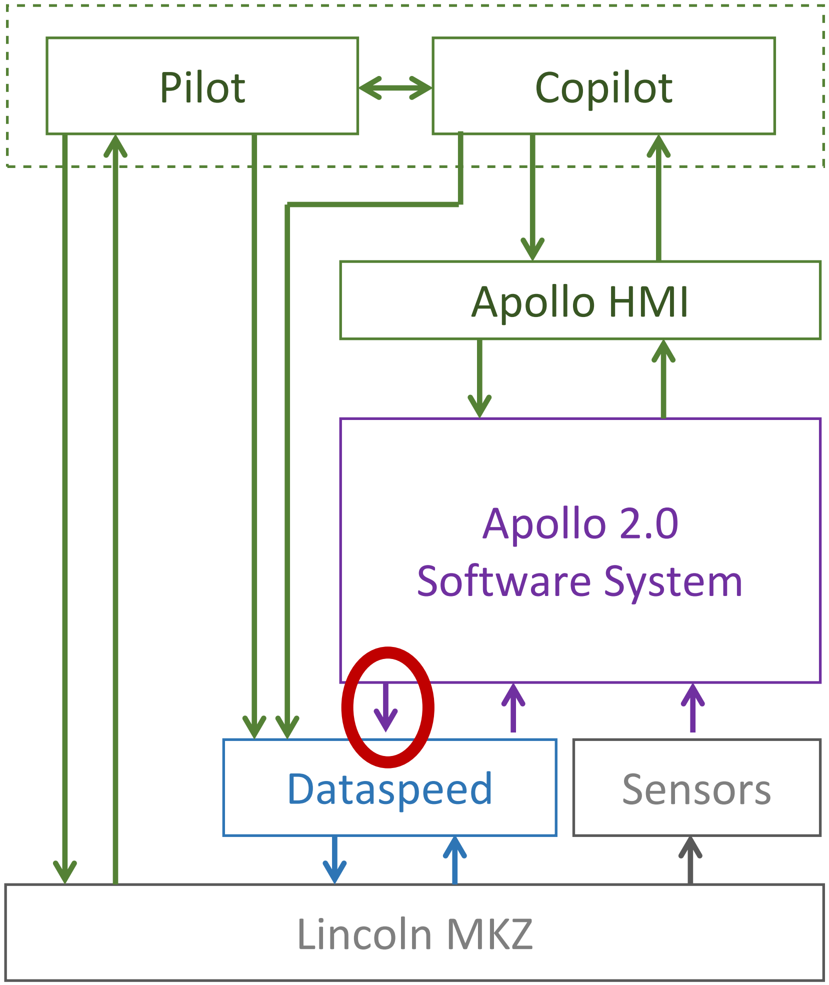
Answer: ESTOP activation results in immediate “return to position” behavior.

- Can’t change

Resulting potential requirements

- **ESTOP must not cause sudden steering angle changes**
- **Pilot/copilot must have advance indication before autonomous mode ends**
- **Copilot must confirm Pilot hands on wheel before providing ESTOP.**
- **Test track training must include copilot activation of ESTOP**
- **Test track training must include ESTOP activation during turns**
- …

Software Interactions



UCA: Apollo provides throttle cmd when forward collision is imminent

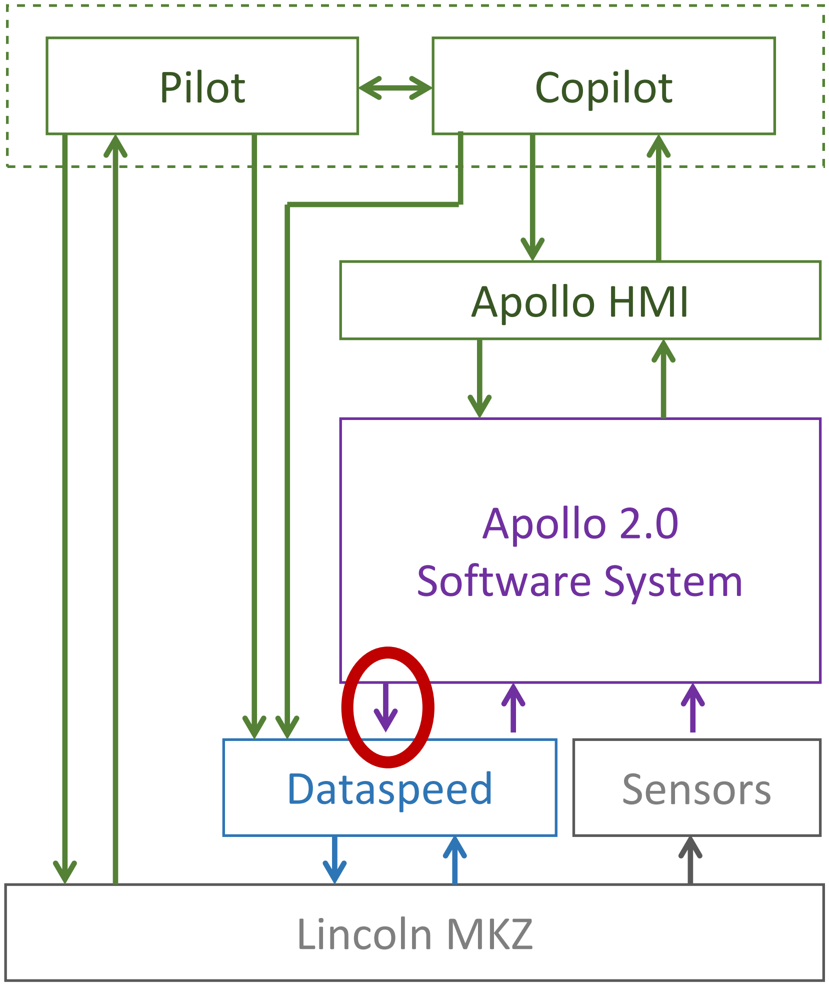
- PM: Apollo incorrectly believes forward collision is not imminent
- Feedback: LIDAR, Camera, Braking status, AEB (automatic emergency braking)
  - Feedback inadequate, missing, etc.

Generated potential requirements

- Apollo must not provide throttle cmd when manual braking is applied
- Apollo must not provide throttle cmd when AEB engages
- ..

Enforced?

Software scenarios



New question: Does Apollo respond to AEB feedback?

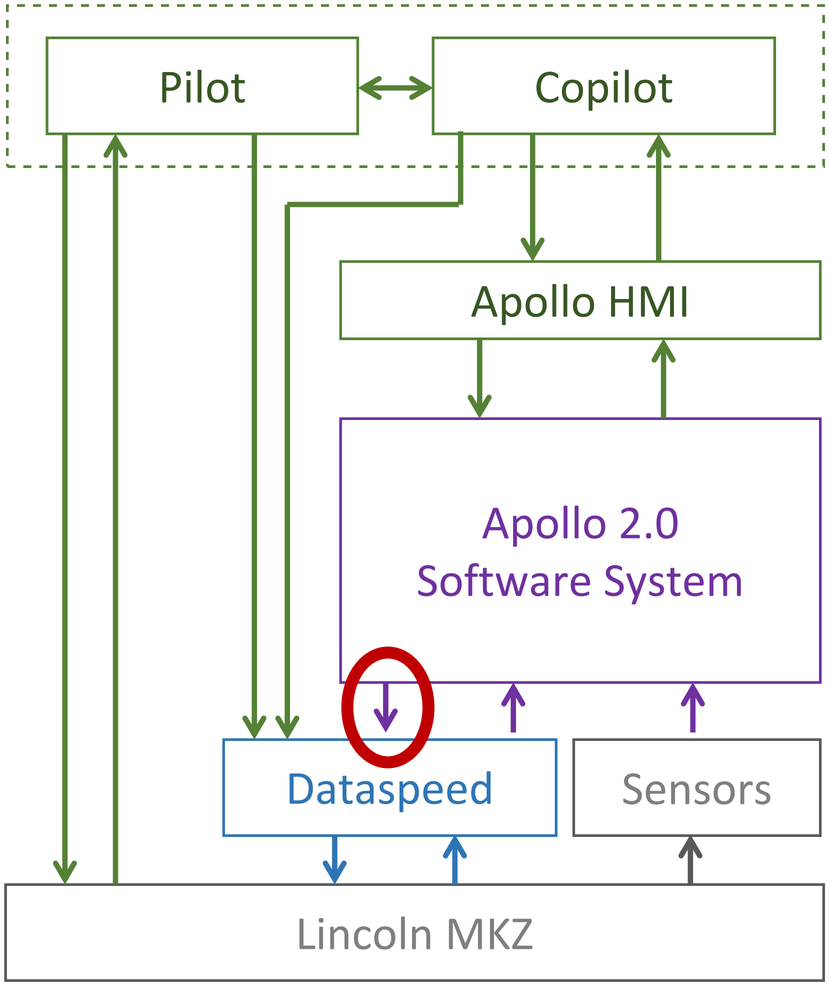
Answer: Apollo ignores AEB status. Operates independent of AEB.

Generated potential requirements

- Apollo must not provide throttle cmd when braking is applied
- Apollo must not provide throttle cmd when AEB engages
- ..

How?

Software Interactions



UCA: Apollo provides throttle cmd when forward collision is imminent

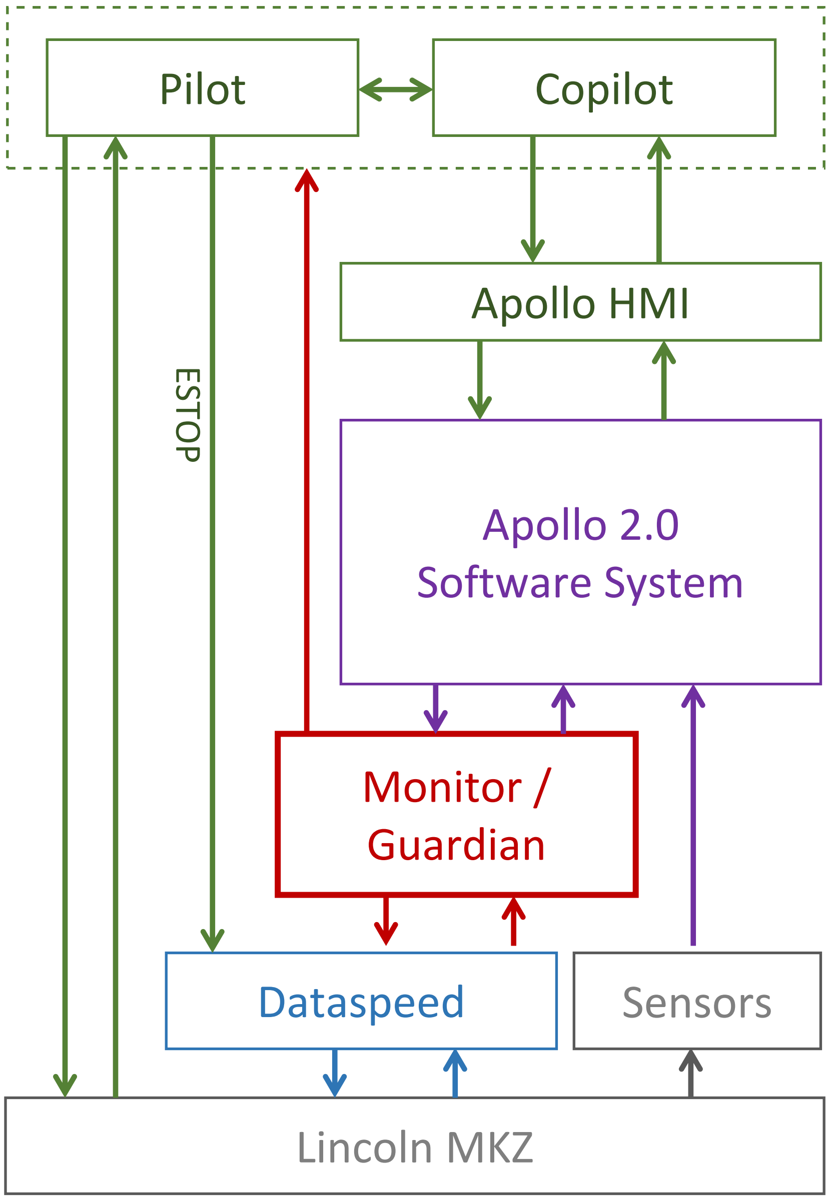
- PM: Apollo incorrectly believes forward collision is not imminent
- Feedback: LIDAR, Camera, Braking status, AEB (Automatic Emergency Braking)
- …

Existing Design/Requirements will cause this!

- Apollo designed to ignore AEB and operate independently
- Apollo relies on AEB as an independent backup
- Apollo throttle commands are designed to spoof driver commands
- AEB is designed to never override driver commands
- Apollo is disabling AEB any time it sends a positive throttle command! (>50% of driving)

Safety features inadvertently defeated by design choices!

Software Monitor/Guardian



- What needs to be monitored?
- Should it ever intervene? If so, when?

- Software team initially proposed 4 requirements
  1. Each Apollo SW module is running
  2. A single instance of each Apollo SW module is running
  3. Each Apollo SW module is sending messages at the correct rate
  4. Each Apollo SW module self-reports no internal faults

- Is this everything? How do you know?
- Decision: use STPA to check

Generated Requirements

Shall detect and warn copilot when:

- Apollo provides throttle commands while AEB is active
- Apollo provides throttle commands while driver applies brake
- Apollo provides throttle commands while parking brake engaged
- Apollo provides IGNORE/CLEAR cmd at any time
- …

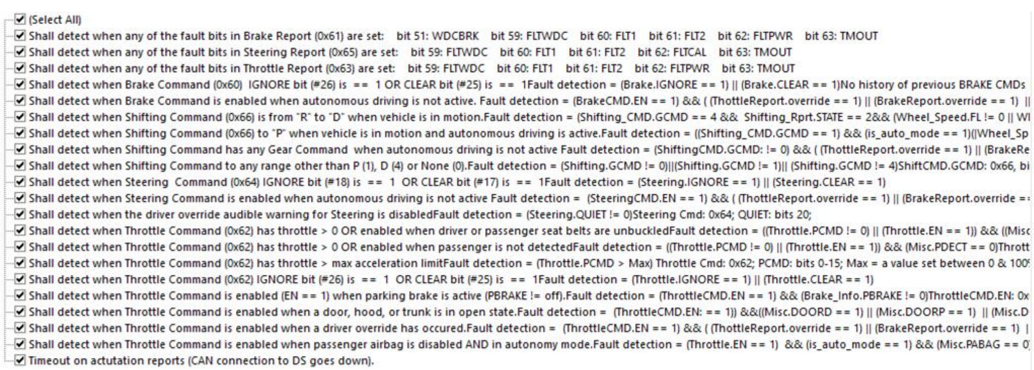
Shall block Apollo commands and report to copilot when:

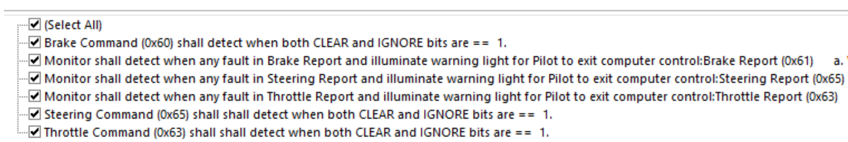
- Apollo steering command specifies excessive steering rate (>TBD) that can destabilize vehicle
- Apollo positive throttle command when vehicle speed exceeds maximum velocity limit for planned test (>TBD)
- Apollo throttle command when not in autonomous mode
- Vehicle is in R when Apollo enters autonomous mode
- Vehicle door is open when Apollo enters autonomous mode
- …

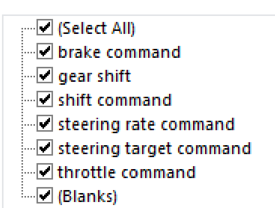
Company engineering decision:
Do not implement at this time

Requirements Allocation

- 84 requirements identified
- Initially allocated to:
  - Safety Actuator Monitor (SAM)
  - Additional SW-based monitor tracking ROS (Robot Operating System) topics
- Team agreed to 20 SW requirements for Safety MCU
  - Highly dependent on a tight development schedule
  - Warning light used to inform Pilot/Co-pilot







Existing Public Road Testing: Examples of Disengagements

	January 2017
Total Autonomous Miles	26
Qualifying Disengagements	3
Disengagement Information	
Date	Causal Factors
01/11/2017	Localization error caused drift
01/13/2017	Perception discrepancy for an object caused braking with traffic behind
01/27/2017	Planning discrepancy caused steering maneuver

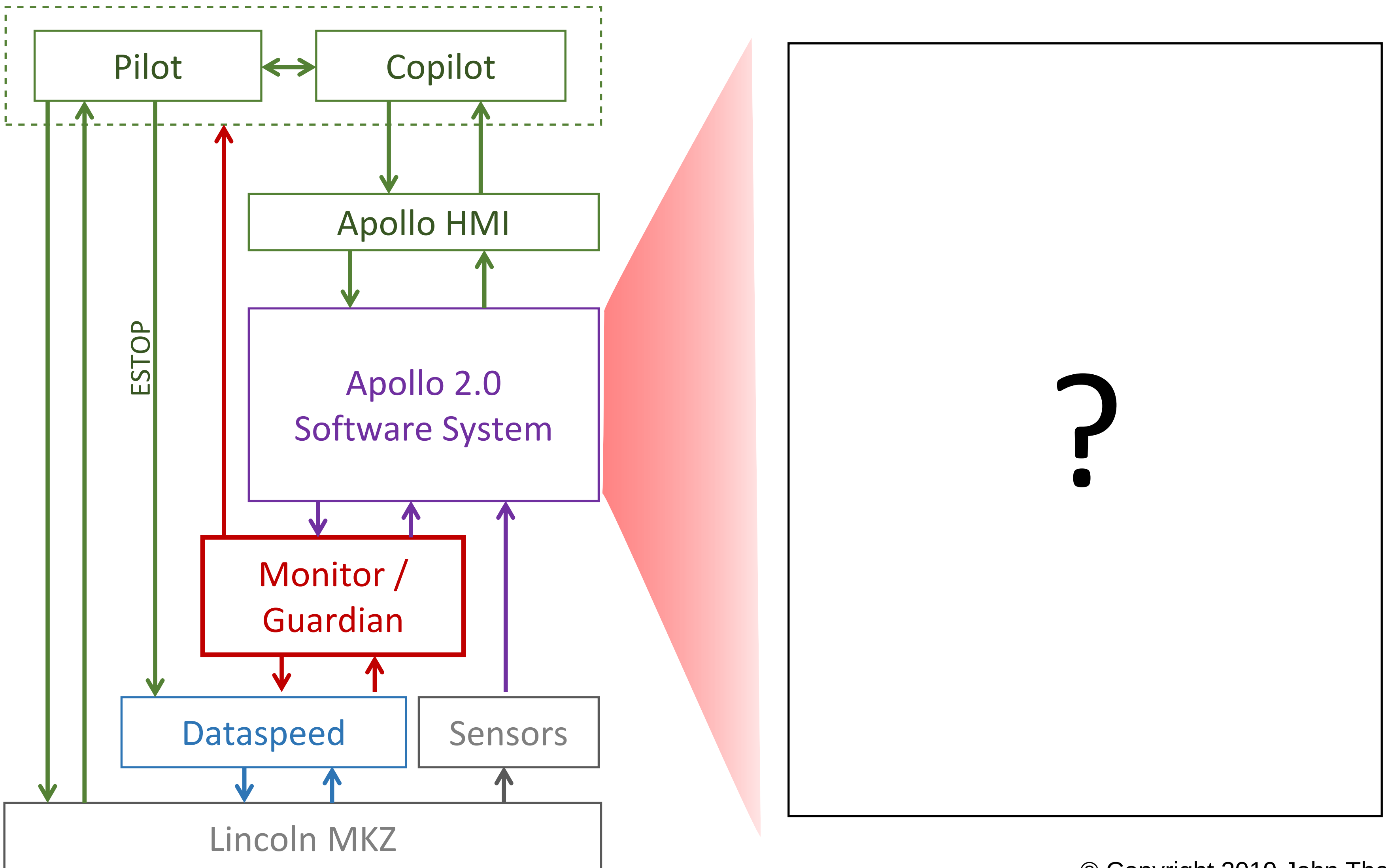
	February 2017
Total Autonomous Miles	5.1
Qualifying Disengagements	2
Disengagement Information	
Date	Causal Factors
02/12/2017	System Fault
02/22/2017	Planning discrepancy caused brake jabs

	March 2017
Total Autonomous Miles	132.63
Qualifying Disengagements	7
Disengagement Information	
Date	Causal Factors
03/03/2017	Misclassification of traffic light detection
03/13/2017	Braking upon engaging system
03/14/2017	Perception discrepancy caused no yield for cross traffic
03/14/2017	Planning discrepancy caused delayed braking for car that cut in and slowed quickly
03/15/2017	Perception discrepancy caused delayed yield at intersection
03/15/2017	Perception discrepancy caused proceeding during right on red with cross traffic
03/30/2017	Perception discrepancy for a pedestrian in crosswalk caused braking with traffic behind

Control Structure Refinement

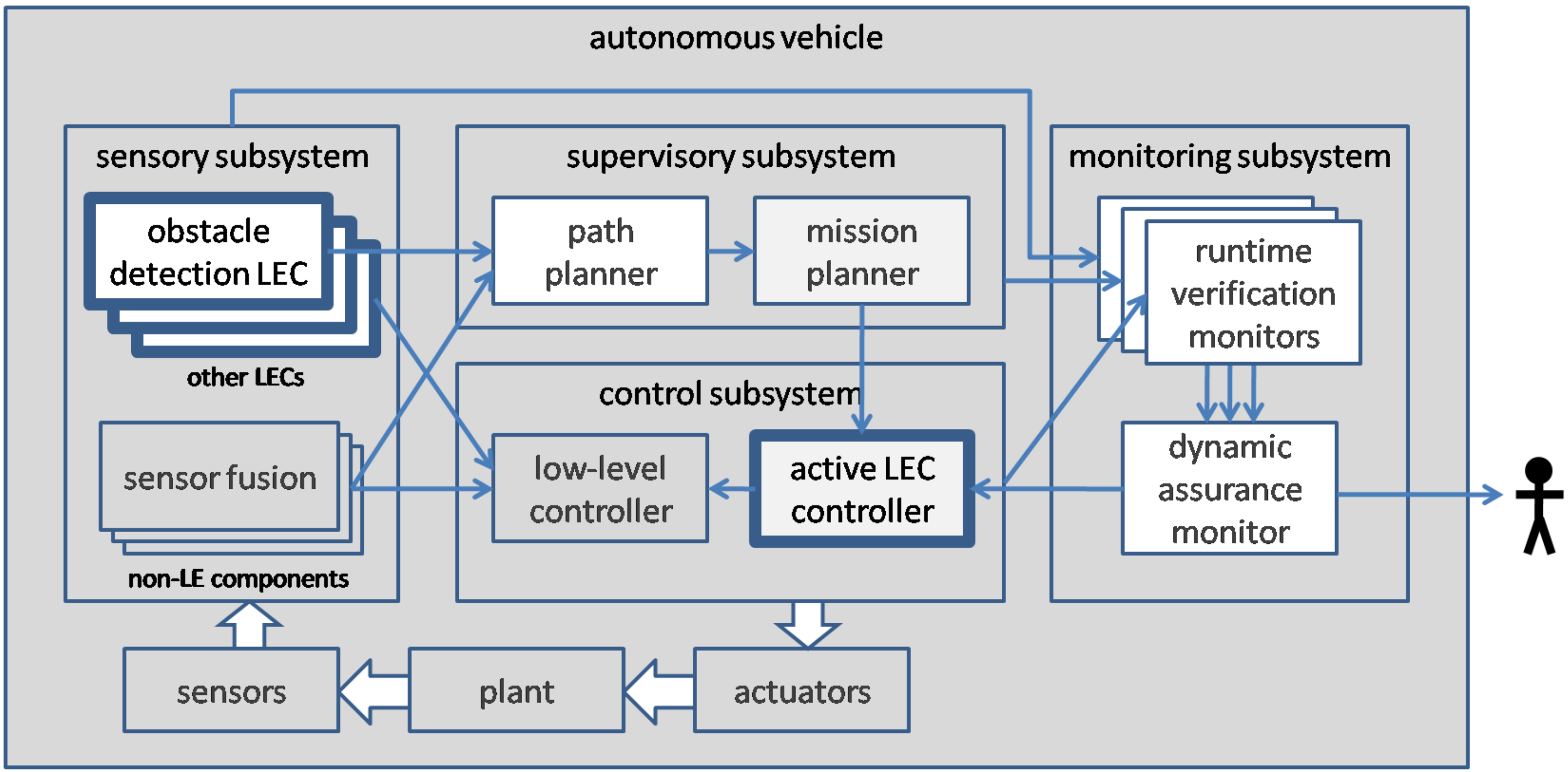
Level 2

Level 3



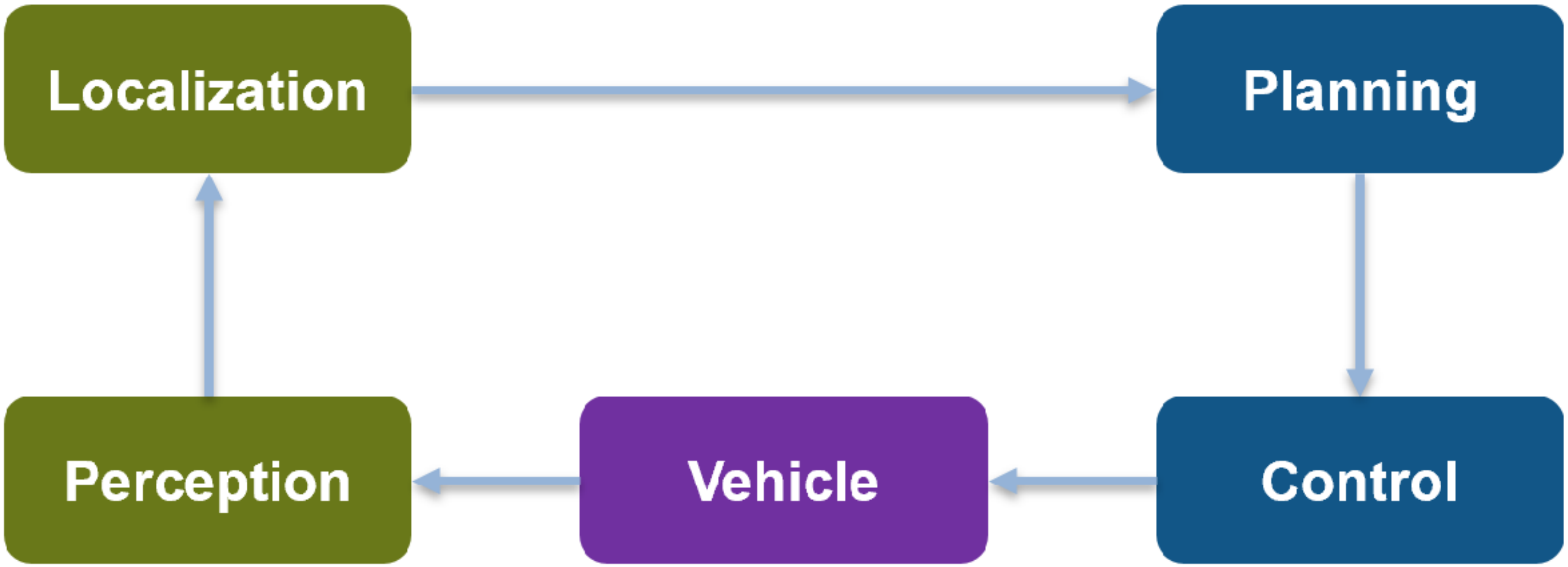
Control Structure
vs.
Data Processing Pipeline

Example data pipeline



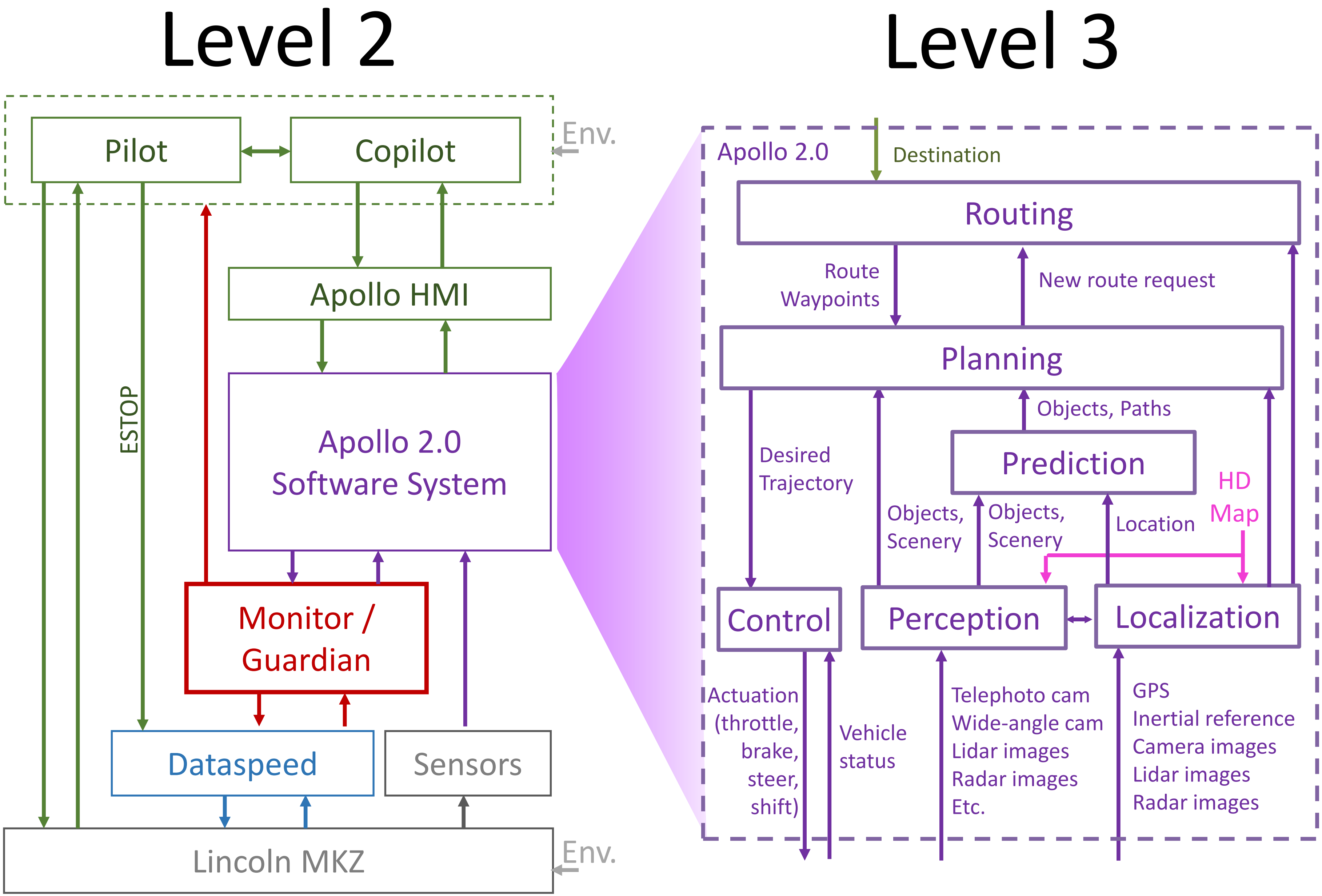
Integrated Static and Dynamic Approaches to High-Assurance for Learning-Enabled Cyber-Physical Systems
https://rtg.cis.upenn.edu/assured-autonomy/

Another model – A control loop



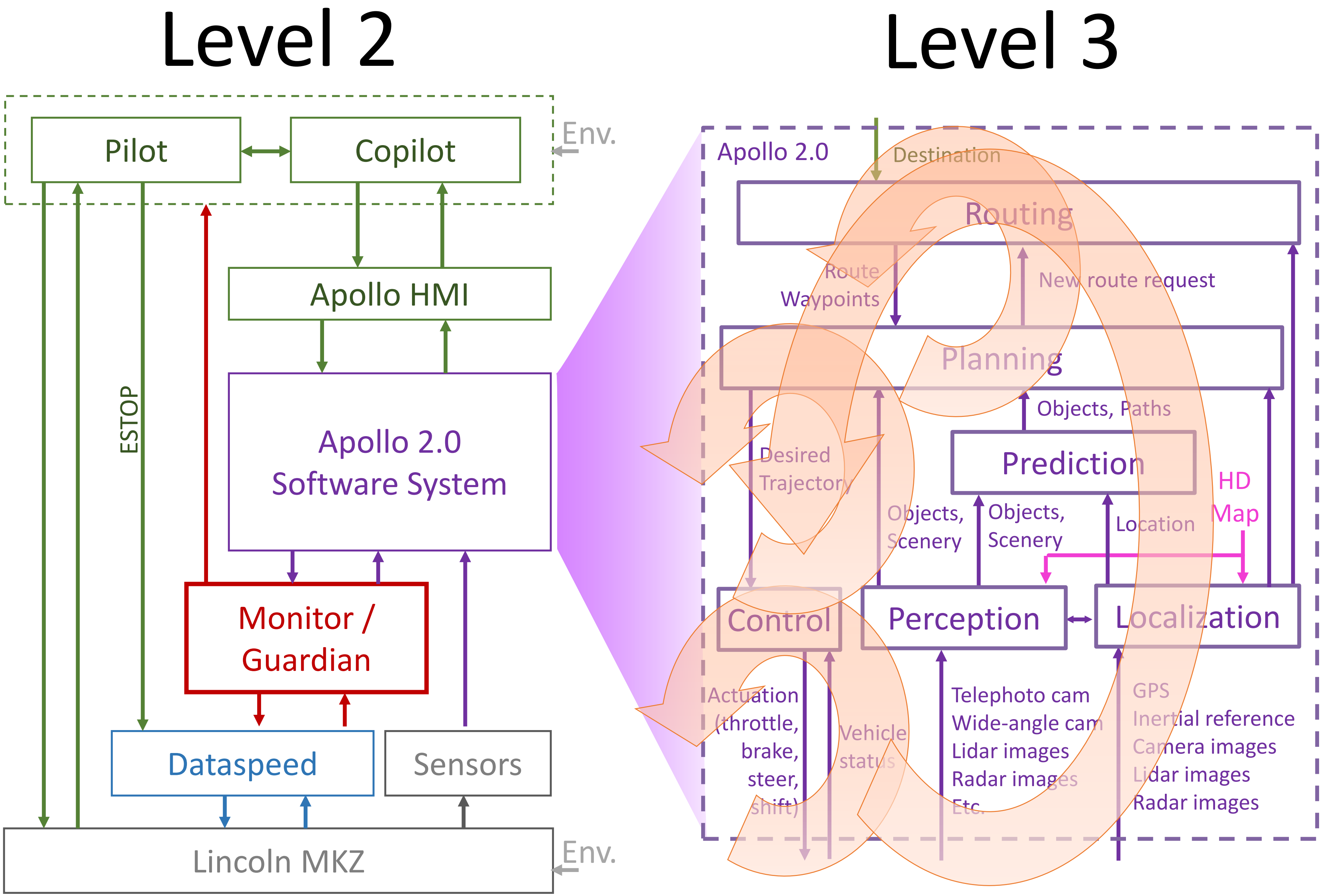
Automated Parking Valet with ROS 2 in Simulink
https://ww2.mathworks.cn/help/ros/ug/automated-valet-using-ros2-simulink.html

Control Structure Refinement



Thomas, 2019

Control Structure Refinement



Thomas, 2019

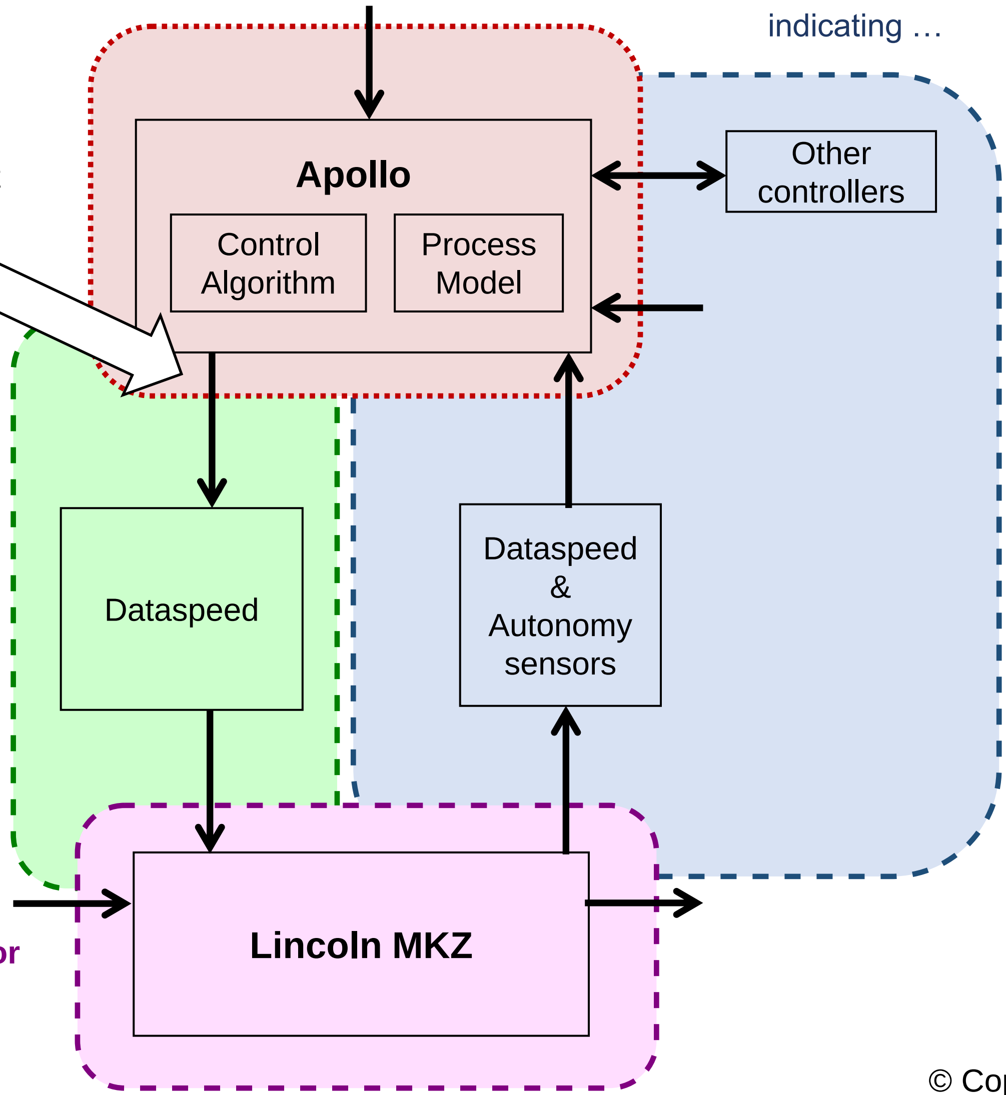
Massachusetts Institute of Technology

New Scenario Approach using Basic Scenarios

1) Inadequate Controller Behavior
- Feedback indicates …
- Apollo does not …

2) Inadequate feedback/information
- Vehicle is …
- Controller receives inadequate feedback indicating …

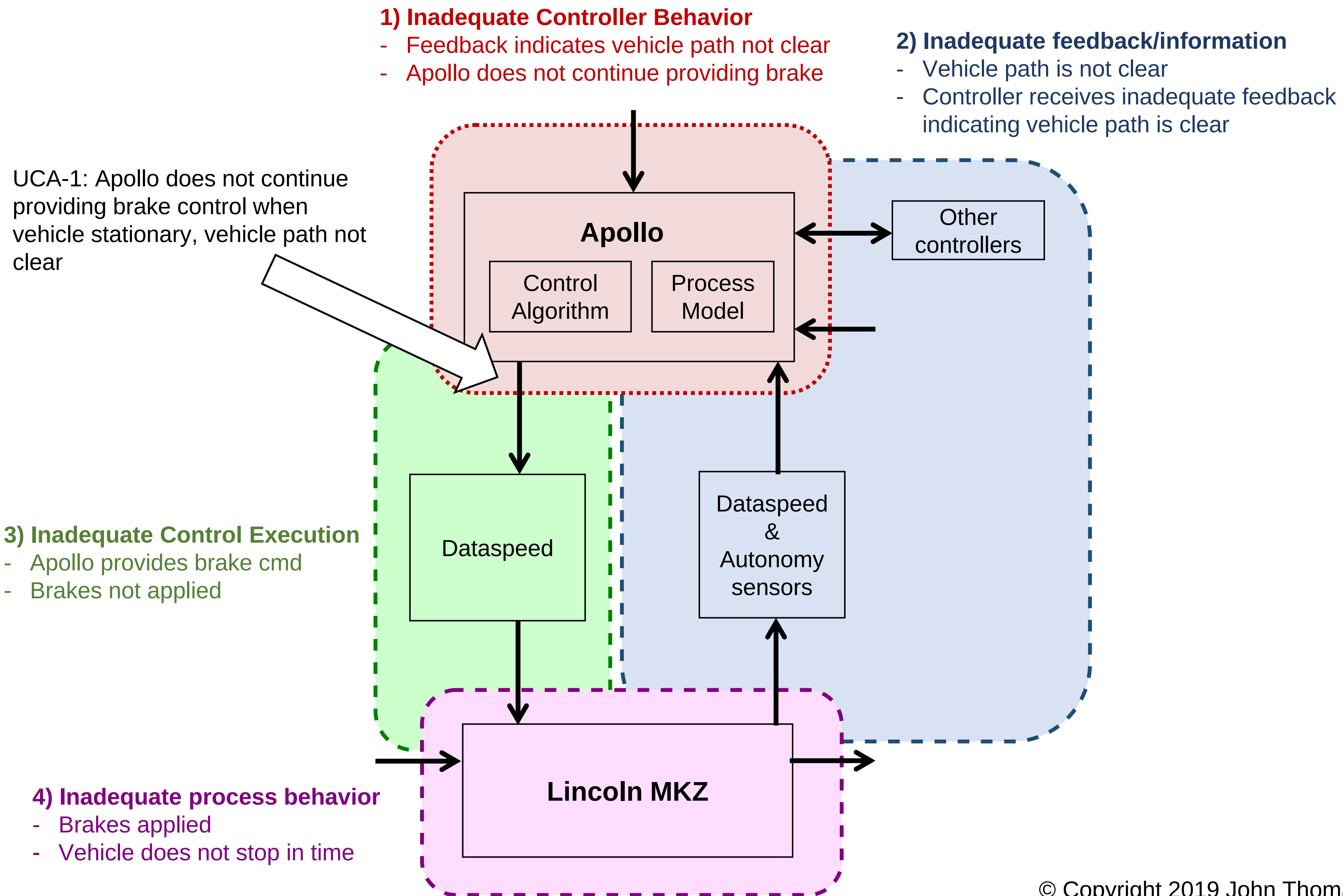
UCA-1: Apollo does not continue providing brake control when vehicle stationary, vehicle path not clear



3) Inadequate Control Execution
- Apollo provides …
- <> not applied

4) Inadequate process behavior
- <> applied
- Vehicle is …

Basic Scenarios



Massachusetts Institute of Technology

Basic Scenario Generation

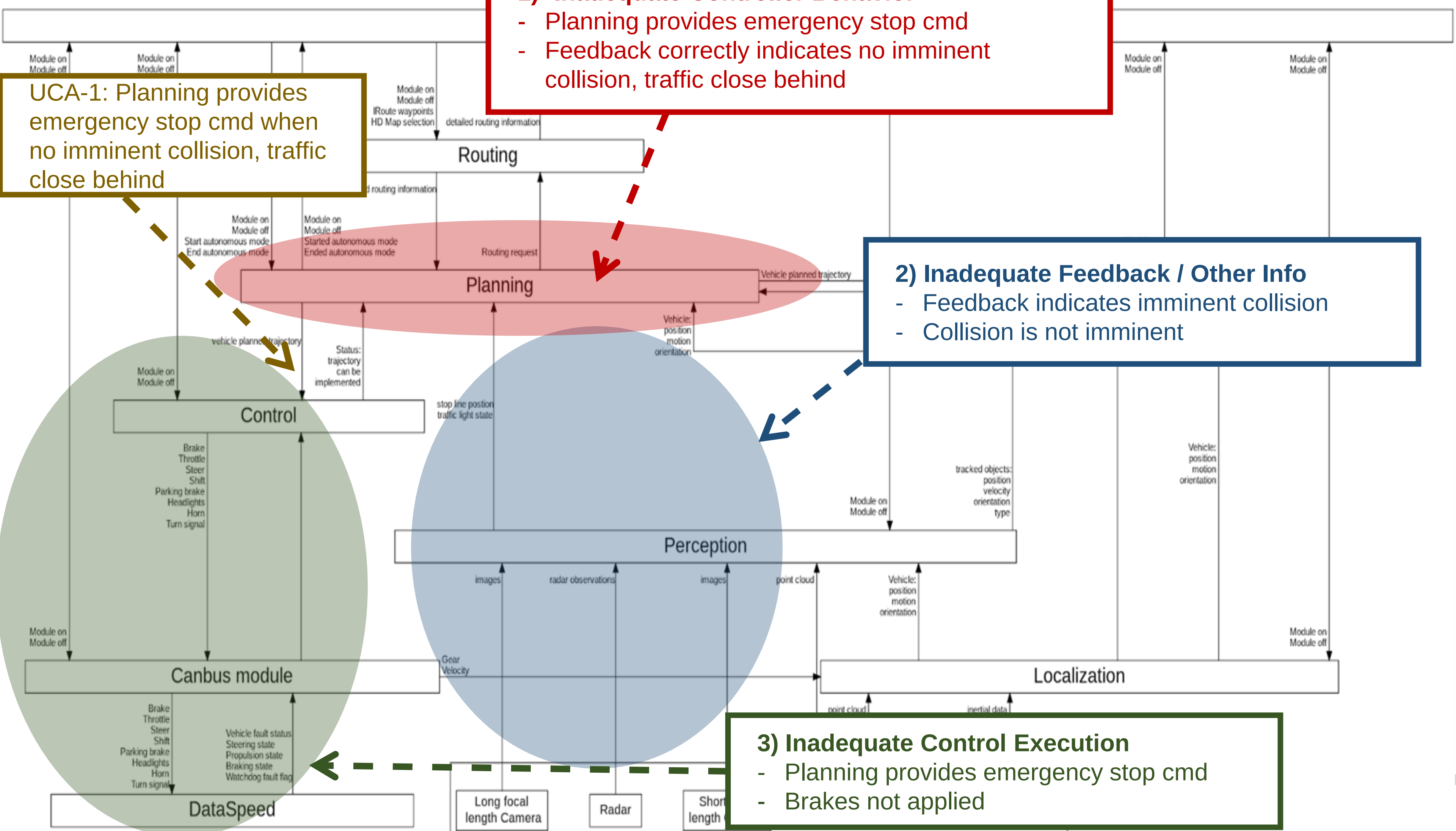
	Not providing causes hazard	Providing causes hazard	Too early, too late, Order	Stopped Too Soon / Applied too long
Control Action	UCA-1	UCA-2	UCA-3	UCA-4

Basic Scenario Table:

	UCA type 1: not providing causes hazard (UCA-#)	UCA type 2: providing causes hazard (UCA-#)	UCA type 3: too early, too late, out of order causes hazard (UCA-#)	UCA type 4: stopped too soon, applied too long causes hazard (UCA-#)
Scenario Type 1: Unsafe Controller Behavior	1) controller doesn't provide <cmd> 2) controller received feedback (or other inputs) that indicated <context>	1) controller provides <cmd> 2) controller received feedback (or other inputs) that indicated <context>	1) controller provides <cmd> too late/early/out of order 2) controller received feedback (or other inputs) that indicated <context> on time / in order	1) controller stops providing <cmd> too soon 2) controller received feedback (or other inputs) that indicated <context> on time
Scenario Type 2: Unsafe Feedback Path	1) feedback received by controller does not indicate <context> 2) <context> is reflected in information from controlled process	1) feedback received by controller does not indicate <context> 2) <context> is reflected in information from controlled process	1) feedback received by controller does not indicate <context> on time / in order 2) <context> is reflected in information from controlled process on time / in order	1) feedback received by controller does not indicate <context> 2) <context> is reflected in information from controlled process
Scenario Type 3: Unsafe Control Path	1) controller does provide <cmd> 2) <cmd> is not received by controlled process	1) controller does not provide <cmd> 2) <cmd> is received by controlled process	1) controller provides <cmd> on time / in order 2) <cmd> is received by controlled process too late/early/out of order	1) controller provides <cmd> with appropriate duration 2) <cmd> is received by controlled process with in appropriate duration
Scenario Type 4: Unsafe Controlled Process Behavior	1) <cmd> is received by controlled process 2) controlled process does not respond by <…>	1) <cmd> is not received by controlled process 2) controlled process does not respond by <…>	1) <cmd> is received by controlled process on time / in order 2) controlled process does not respond by <…>	1) <cmd> is received by controlled process with appropriate duration 2) controlled process does not respond by <…>

(Thomas, 2016), (Thomas, 2017)

Results: Basic Scenarios



UCA-1: Planning provides emergency stop cmd when no imminent collision, traffic close behind

1) Inadequate Controller Behavior
- Planning provides emergency stop cmd
- Feedback correctly indicates no imminent collision, traffic close behind

2) Inadequate Feedback / Other Info
- Feedback indicates imminent collision
- Collision is not imminent

3) Inadequate Control Execution
- Planning provides emergency stop cmd
- Brakes not applied

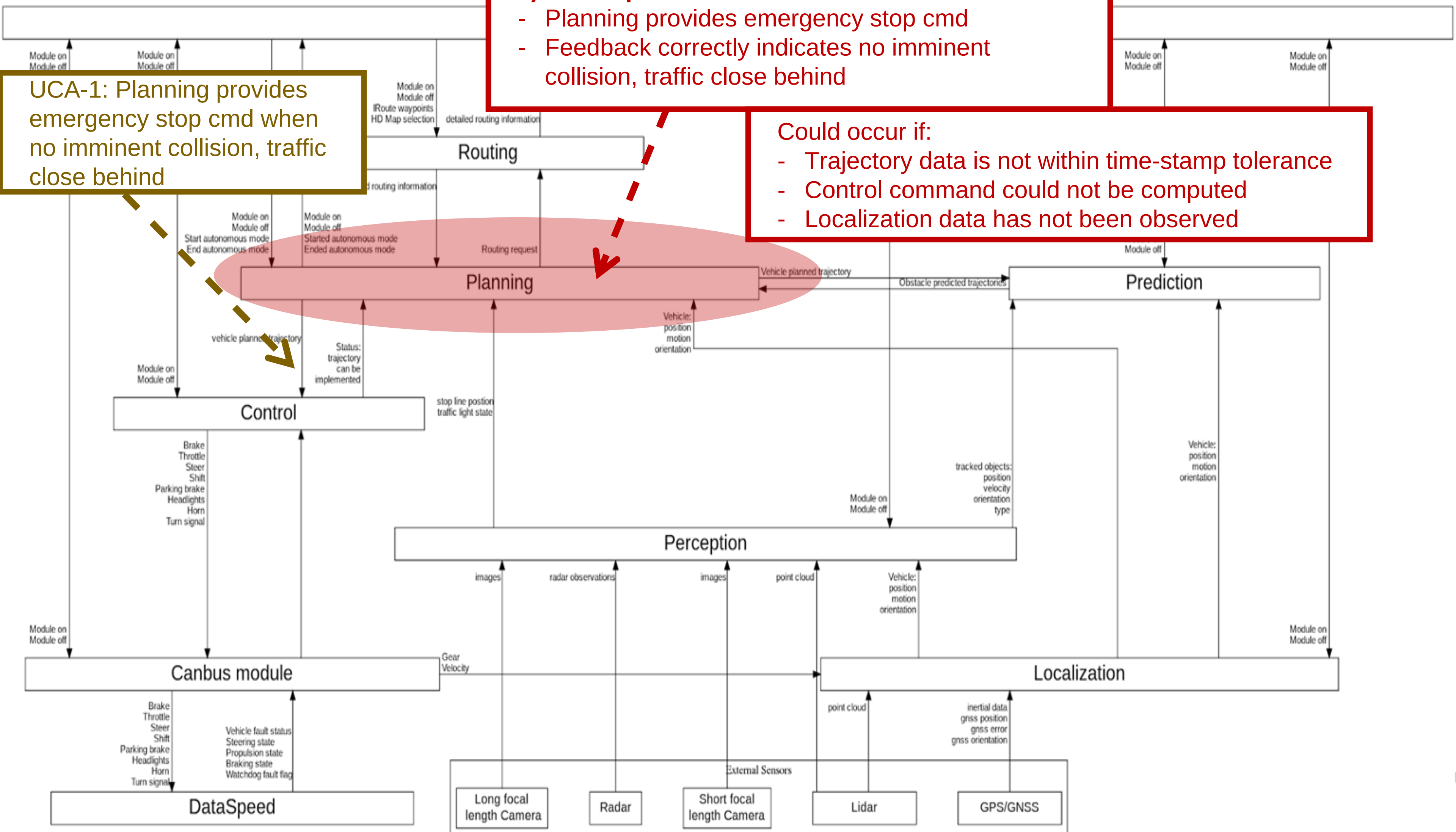
Massachusetts Institute of Technology

Scenario Refinement

1) Inadequate Controller Behavior
- Planning provides emergency stop cmd
- Feedback correctly indicates no imminent collision, traffic close behind

UCA-1: Planning provides emergency stop cmd when no imminent collision, traffic close behind

Could occur if:
- Trajectory data is not within time-stamp tolerance
- Control command could not be computed
- Localization data has not been observed



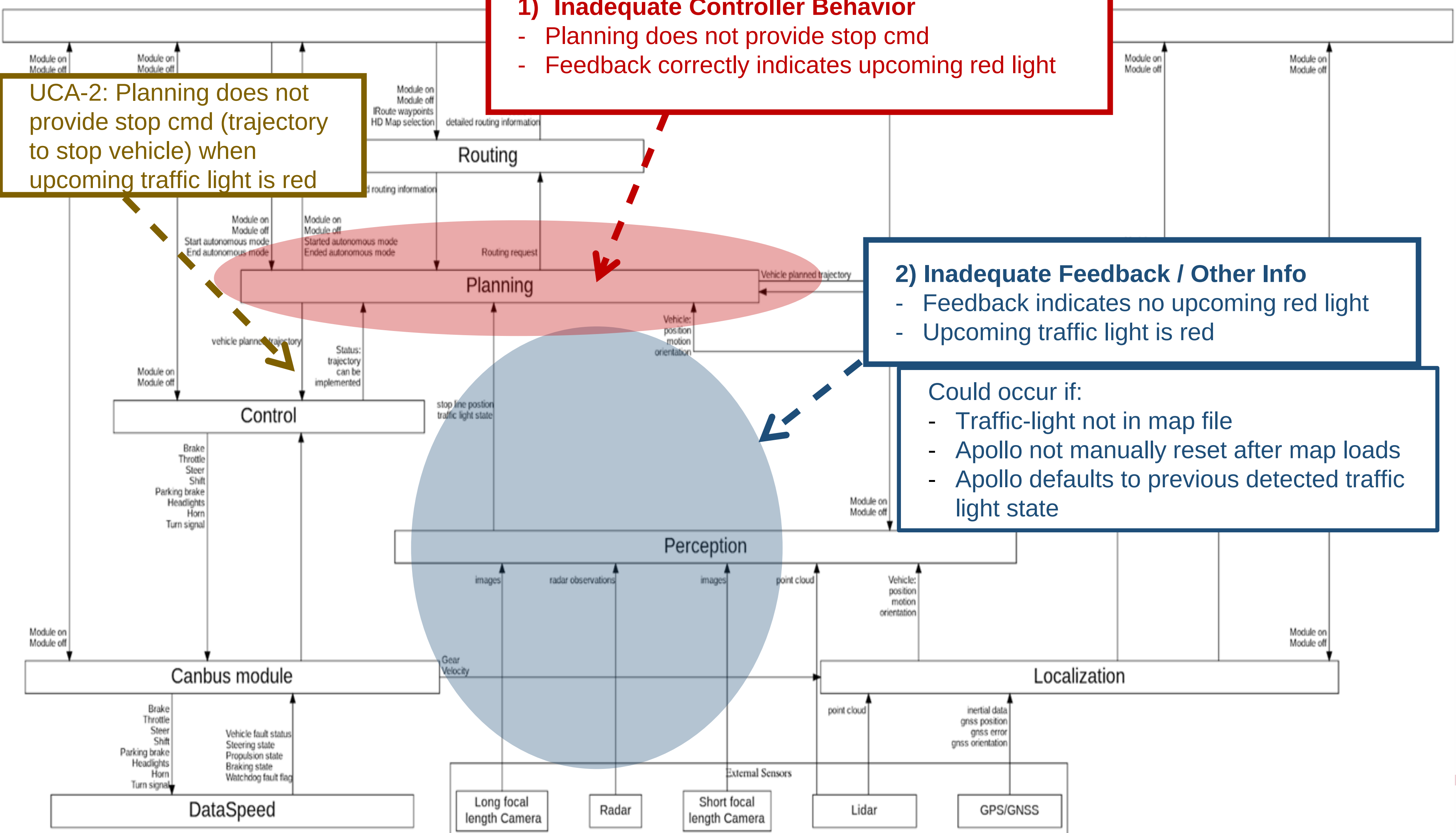
Scenario Refinement

1) Inadequate Controller Behavior
- Planning does not provide stop cmd
- Feedback correctly indicates upcoming red light

UCA-2: Planning does not provide stop cmd (trajectory to stop vehicle) when upcoming traffic light is red

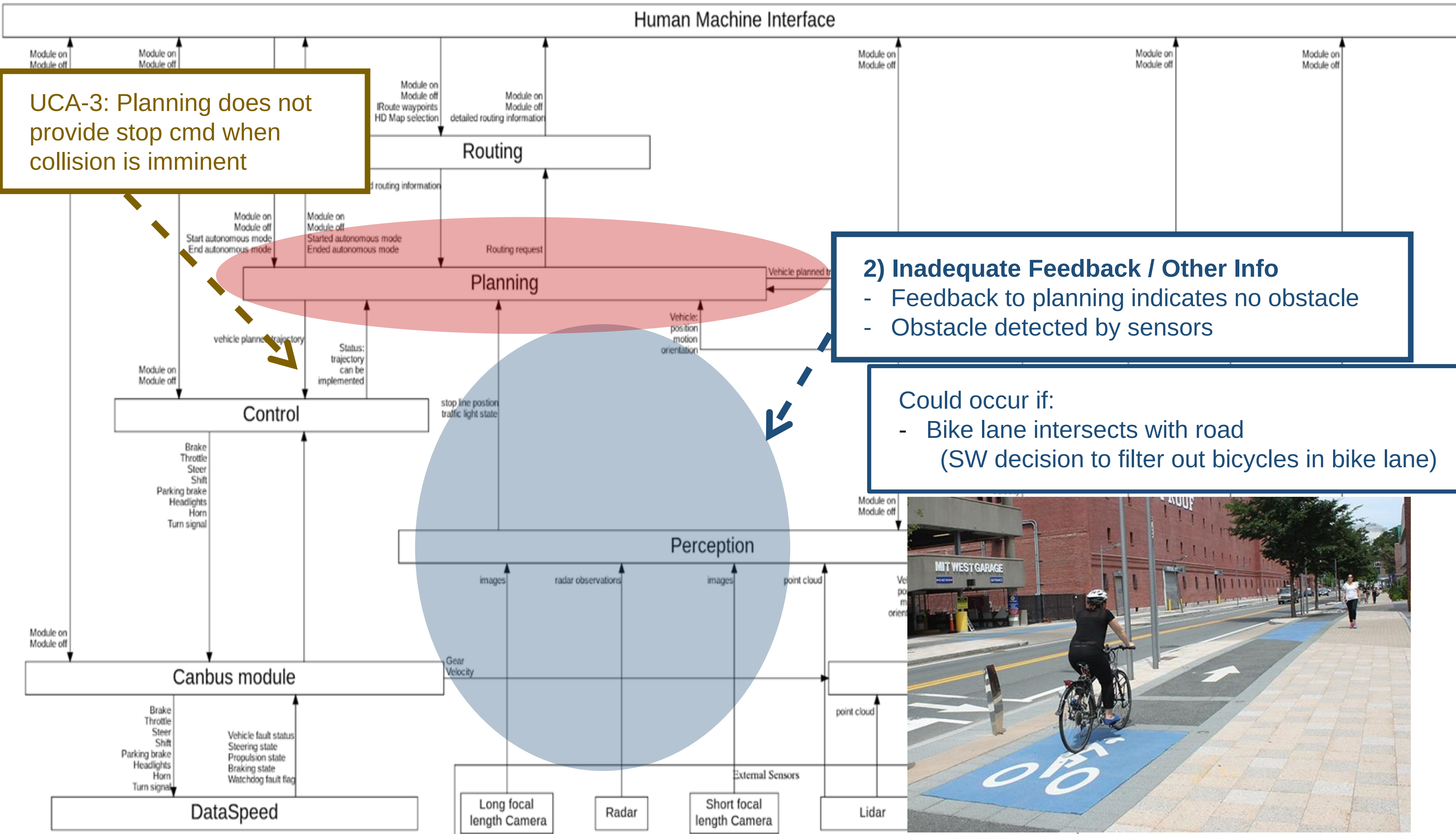
2) Inadequate Feedback / Other Info
- Feedback indicates no upcoming red light
- Upcoming traffic light is red

Could occur if:
- Traffic-light not in map file
- Apollo not manually reset after map loads
- Apollo defaults to previous detected traffic light state



Massachusetts Institute of Technology

Scenario Refinement

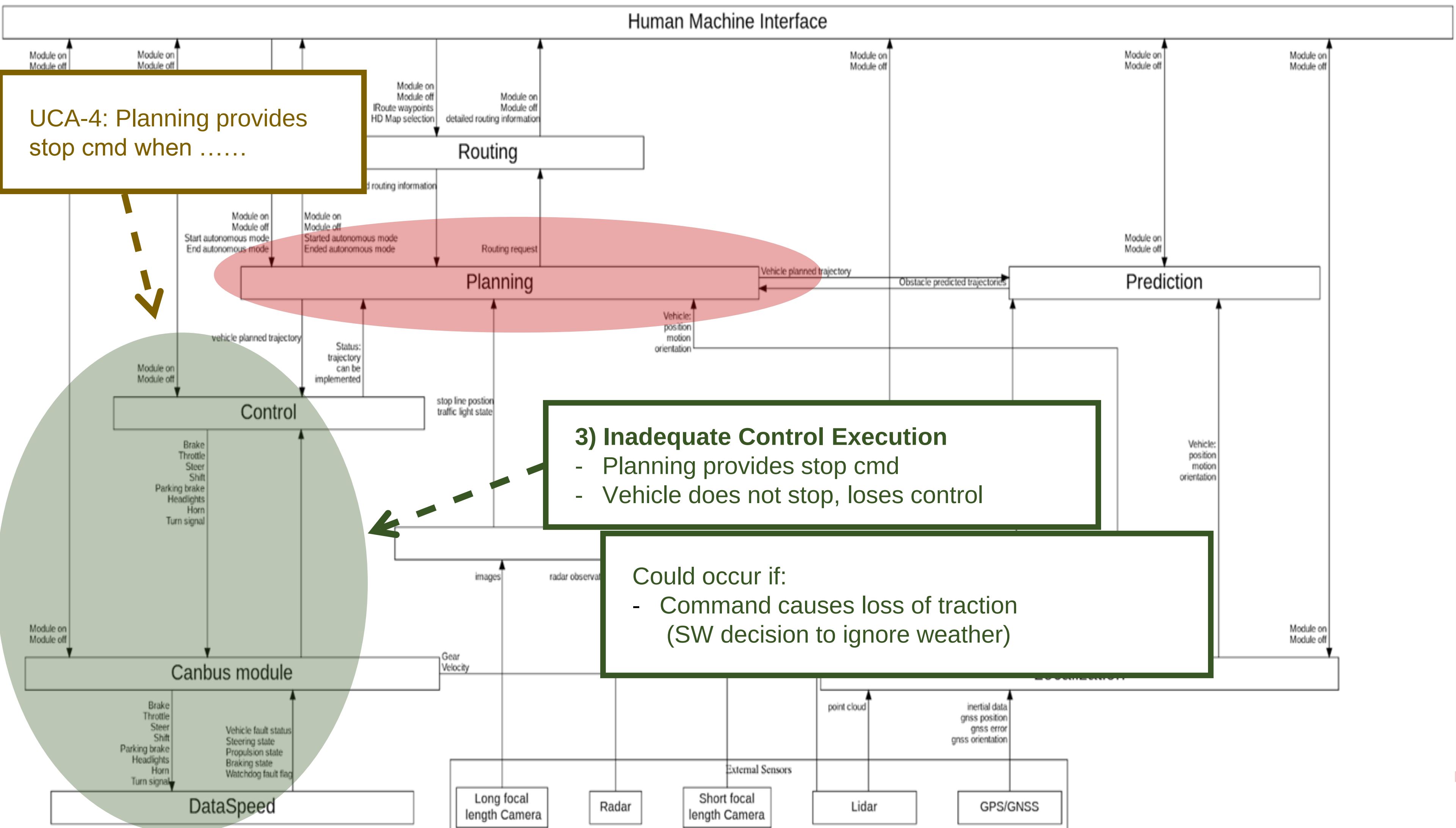


UCA-3: Planning does not provide stop cmd when collision is imminent

2) Inadequate Feedback / Other Info
- Feedback to planning indicates no obstacle
- Obstacle detected by sensors

Could occur if:
- Bike lane intersects with road
  (SW decision to filter out bicycles in bike lane)

Scenario Refinement



UCA-4: Planning provides stop cmd when ……

3) Inadequate Control Execution

- Planning provides stop cmd
- Vehicle does not stop, loses control

Could occur if:

- Command causes loss of traction
  (SW decision to ignore weather)

Level 2 Analysis (ad-hoc)

3/24/2019 apollo-example/STPA/level-1/Level1Type1and2Scenarios.md · sm-step-4-option-1 · trustable / av-stpa · GitLab

Level1Type1and2Scenarios.md 58.4 KB

Scenario Analysis for Unsafe Control Actions: Brake Control

Unsafe by Not Providing

UCA-6.1: Apollo does not provide the brake control action when relative velocity and distance to an obstacle mean that a collision is imminent.

True statement from UCA context: The vehicle is approaching an obstacle with a velocity and acceleration vector that indicate a collision

Belief:

- Apollo incorrectly believes that the relative velocity is lower than in reality so that there is no need to brake

Type 2 scenario:

- Controller receives incorrect feedback / information:
  - Information received: The feedback received is insufficient to accurately determine the relative velocity
  - How this could happen given the true statement above:
    - The radar sensor (doppler) / lidar sensor (point cloud) is compromised
    - A data error on the vehicle CAN bus prevents accurate, up-to-date data being received
    - The process model receives stale relative speed (e.g. doppler information) and fails to correlate the changing point-position of the object with a dangerous relative velocity

Type 1 scenario:

- Controller receives correct feedback but interprets it incorrectly or ignores it:
  - Information received: At least one sensor presents an accurate distance measurement, but it is overridden by the process model
  - How this could occur given the true statement above
    - A malfunctioning sensor yielding an incorrect value leads to the true value being overwritten, overridden, or distorted

Belief:

- Apollo incorrectly believes that the relative distance is higher than in reality

Type 2 scenario:

- Controller receives incorrect feedback / information:
  - Information received: The feedback received is insufficient to accurately determine the relative distance
  - How this could occur given the true statement above:
    - The rangefinding or object tracking sensors are compromised, [HOW-1]
      - Roof mounted optics are vulnerable to collision with a variety of unexpected obstacles, and could therefore have their optical alignment and focus compromised
      - Environmental or load-shed debris such as leaves or plastic bags could block or distort the images / beams
      - The relevant sensor suffers an internal failure

Type 1 scenario:

- Controller receives correct feedback but interprets it incorrectly or ignores it:

https://gitlab.com/trustable/av-stpa/blob/sm-step-4-option-1/apollo-example/STPA/level-1/Level1Type1and2Scenarios.md

1/20

Massachusetts Institute of Technology

83 UCAs
~20 scenarios per UCA

Is there a better approach?

<50% of 22 UCAs

Level 2 Analysis (new method)

phase-1.md 15 KB

UCA-6.1: Apollo does not provide brake control when relative velocity and distance to an obstacle mean that a collision is imminent

Type 1.1:

- Apollo does not provide brake control when relative velocity and distance to an obstacle mean that a collision is imminent
- The feedback does indicate that an obstacle is in the vehicle's path

Type 2.1:

- The feedback does not indicate that an obstacle is in the vehicle's path
- An obstacle is in the vehicle's path

Feedback info:

- Inadequate vehicle speed
- Inadequate other vehicle / object velocity
- Inadequate other vehicle / object distance
- Inadequate object detection

UCA-6.2: Apollo does not provide brake control when in autonomous mode and vehicle speed exceeds limits (limits for controllability, stability, upcoming manoeuvre, speed limit, traffic flow limit, planned test limit, etc.)

Type 1.2:

- Apollo does not provide brake control when in autonomous mode and vehicle speed exceeds limits (limits for controllability, stability, upcoming manoeuvre, speed limit, traffic flow limit, planned test limit, etc.)
- The feedback does indicate that the vehicle exceeds limits

Type 2.2:

- The feedback does not indicate that the vehicle exceeds limits
- The vehicle does exceed limits

Feedback info:

- Inadequate vehicle speed
- Inadequate speed limits

UCA-6.3: Apollo does not provide brake control when in autonomous mode, the vehicle is stationary, and vehicle path is not clear

Type 1.3:

- Apollo does not provide brake control when in autonomous mode, the vehicle is stationary, and vehicle path is not clear
- The feedback does indicate that the vehicle is in autonomous mode, the vehicle is stationary and the vehicle path is not clear

Type 2.3:

- The feedback does not indicate that the vehicle is in autonomous mode, the vehicle is stationary and the vehicle path is not clear
- The vehicle is in autonomous mode, the vehicle is stationary and the vehicle path is not clear

Feedback info:

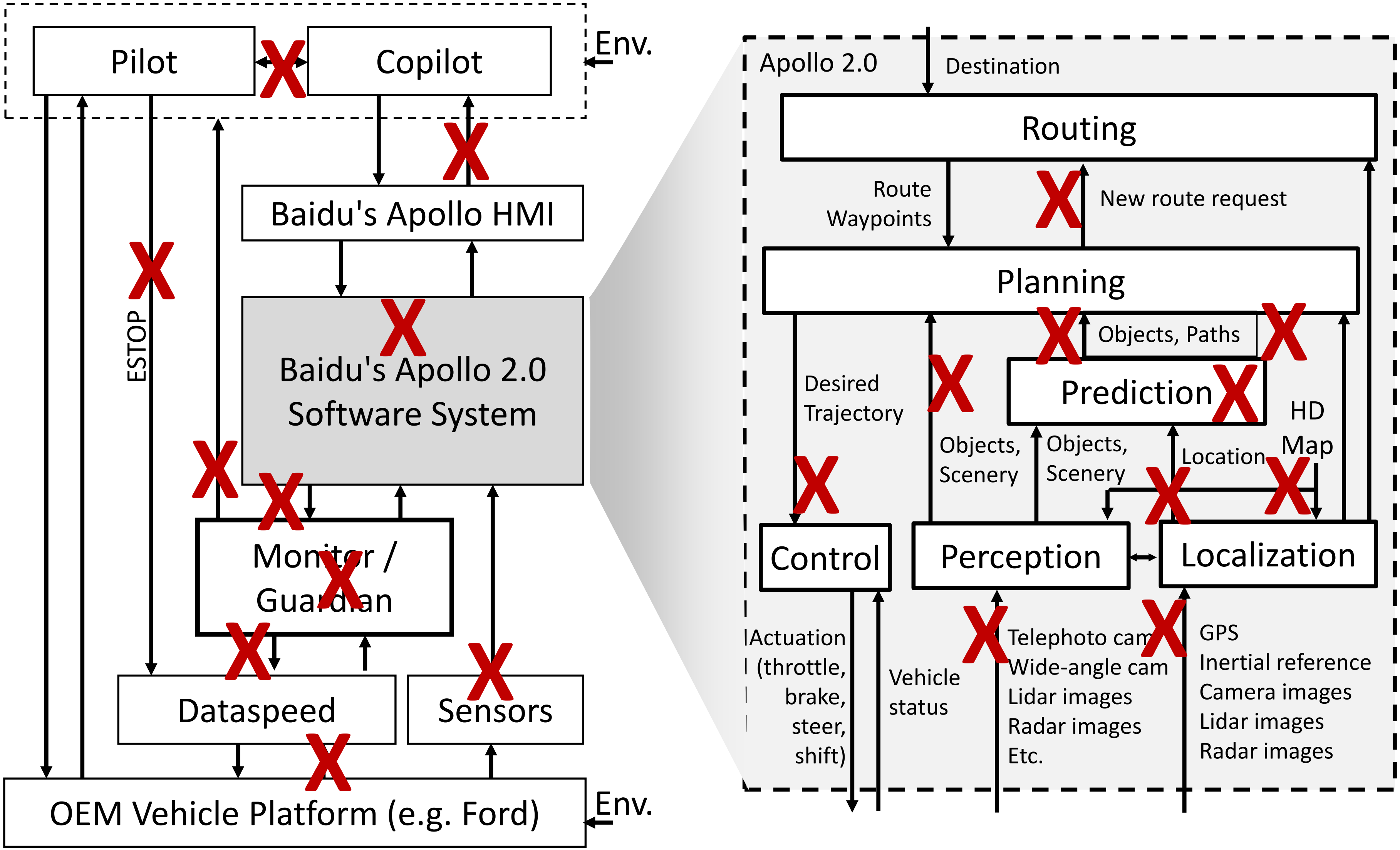
- Inadequate autonomous mode detection

https://gitlab.com/trustable/av-stpa/blob/sm-step-4-option-1/apollo-example/STPA/level-1/new-method/phase-1.md

1/8

~100% of 22 UCAs (~5x reduction)

STPA: Autonomous Vehicle Software



STPA identified many vulnerabilities and unintended, designed behaviors in the product.
STPA results were used to fix the system and improve the design while product in operation.

Massachusetts Institute of Technology

Examples of STPA Impact

- ✓ Unanimous Go/No-Go decision path (incremental acceptance increased over time):
  - Program management, System Integrators, Legal, Mechanical
- ✓ STPA scenarios -> Closed test tracks, test routes, technical req's
- ✓ Test route criteria, proposed routes reviewed against UCAs
- ✓ Clear test start/end procedure
- ✓ ESTOP usage clearly defined, irrespective of who is in the rear seat (safety > marketing)
- ✓ Safety Actuator Monitor
- ✓ Identified incorrect autonomy SW behavior assumptions
  - ✓ E.g. impact between v2.0 and v3.x SW
- ✓ Identified many actions not previously identified, such as throttle commanded with EPB activated
- ✓ Generated requirements: Driver training, procedures, test track, autonomy, etc.

Reflections from Codethink

- Open Source Safety
- https://gitlab.com/trustable/av-stpa
- Manage complexity
- Safety led software architecture

STPA Impact on Program

- ✓ STPA provided key feedback to Program Management to recognize risk, enable informed Go/No-Go decision
- ✓ STPA provided key feedback about market gap, triggered new products

Impact Discussion