Lecture

Control and Perception in Networked and Autonomous Vehicles

Patrick Scheffe, M. Sc. | Dr.-Ing. Bassam Alrifaee

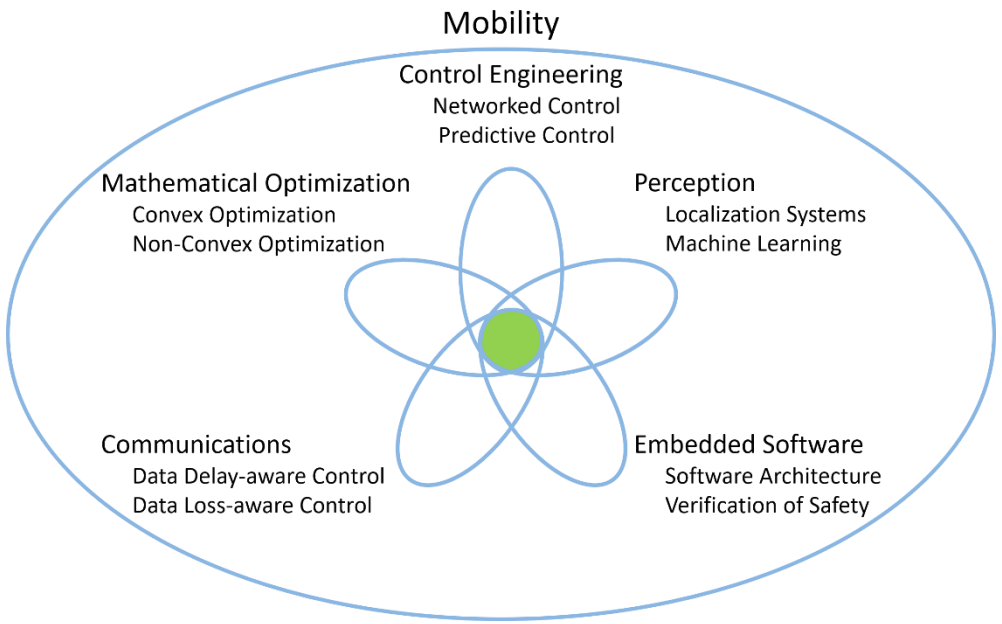
Winter Semester 2021/2022

Part 6

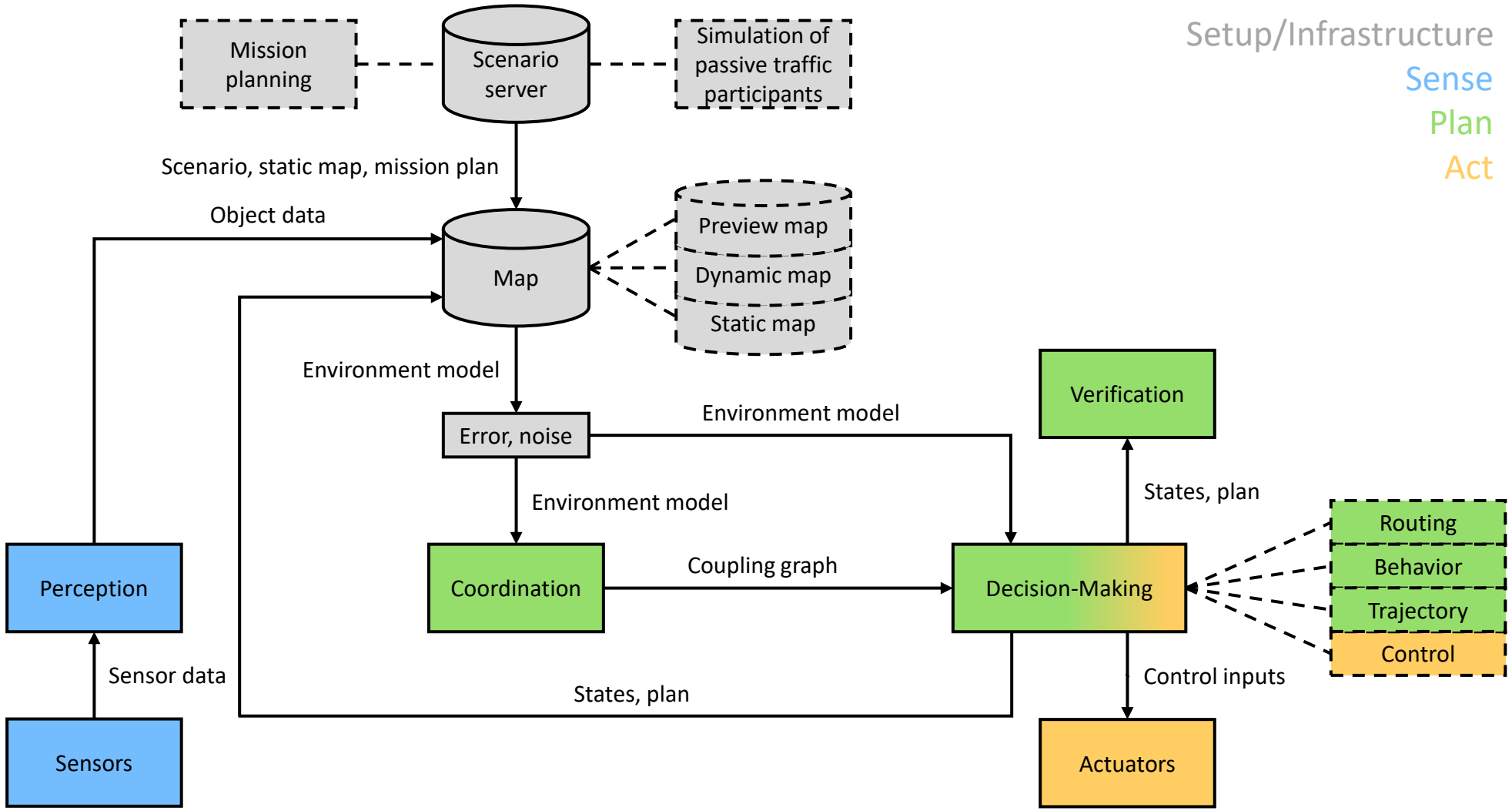
Software Architectures and Testing Concepts

Course contents

- Dynamic vehicle models
- Control and optimization
- Network and distribution
- Machine perception
- Software architectures and testing concepts
- Case study



CPM Lab architecture



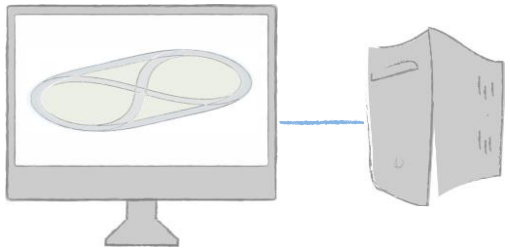
Further literature (1)

- B. Alrifaee. Networked Model Predictive Control for Vehicle Collision Avoidance. PhD thesis, RWTH Aachen University, 2017.
- K. Beck. Extreme Programming Explained. Addison-Wesley, 1999.

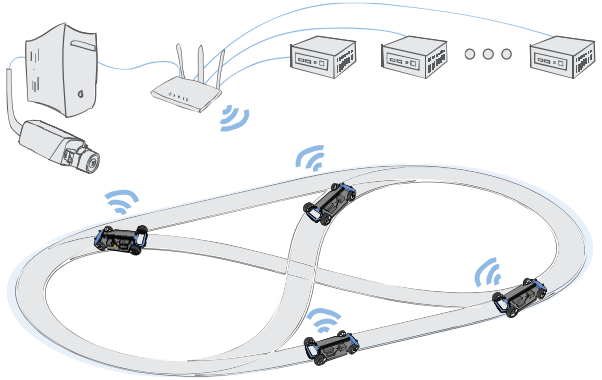
Further literature (2)

- M. Kloock, J. Maczijewski, P. Scheffe, A. Kampmann, A. Mokhtarian, S. Kowalewski, and B. Alrifaee. Cyber-Physical Mobility Lab – An Open-Source Platform for Networked and Autonomous Vehicles, 2020.
- P. Scheffe, J Maczijewski, M. Kloock, A. Kampmann, A. Derks, S. Kowalewski, and B. Alrifaee. Networked and Autonomous Model-scale Vehicles for Experiments in Research and Education, 2020.
- M. Kloock, P. Scheffe, I. Tülleners, J. Maczijewski, S. Kowalewski, and B. Alrifaee. Vision-Based Real-Time Indoor Positioning System for Multiple Vehicles, 2020.

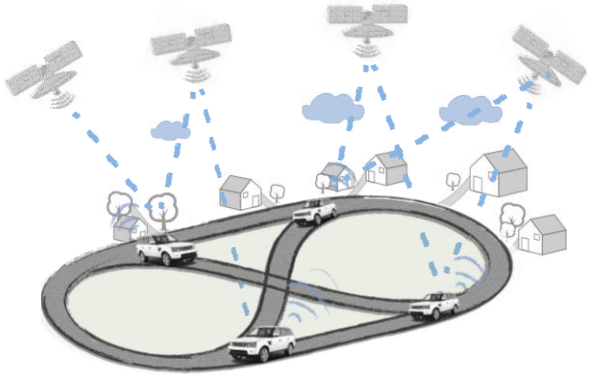
CPM Lab motivation



Simulation:
abstracts from
real-world behavior



Lab:
rapid functional
prototyping, cost
and time efficient

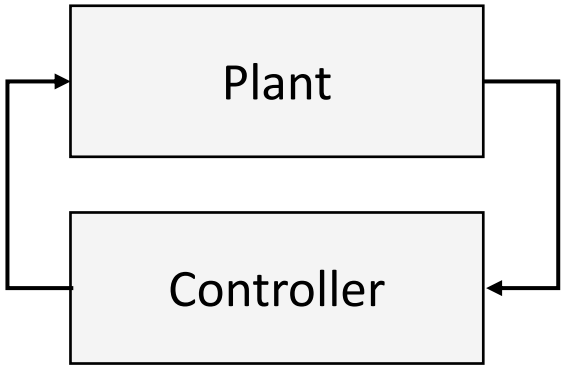


Real-world:
expensive and
time-consuming

CPM Lab motivation

- Dynamics
- Network
  - Delays
  - Packet drop
- Hardware / software
  - Sensors
  - Actuators
- Isolate effects – control uncertainties
- Goal
  - Determinism and reproducibility
  - In-the-loop testing for networked systems

In-the-loop testing (XIL)



Software

Model

C

Generated Code

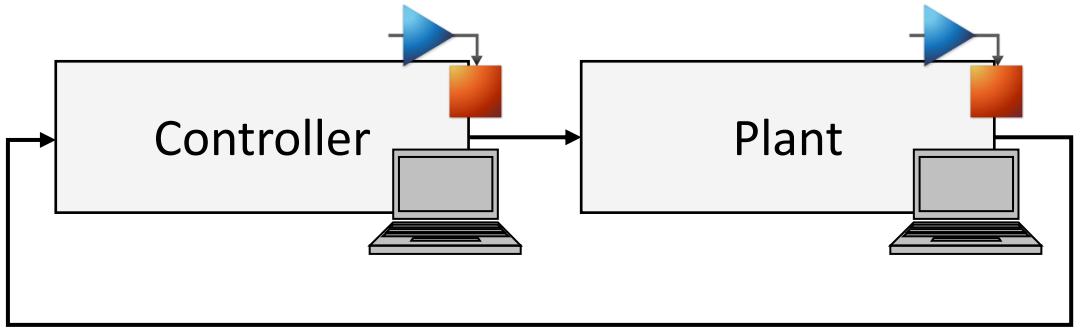
Hardware

Desktop PC

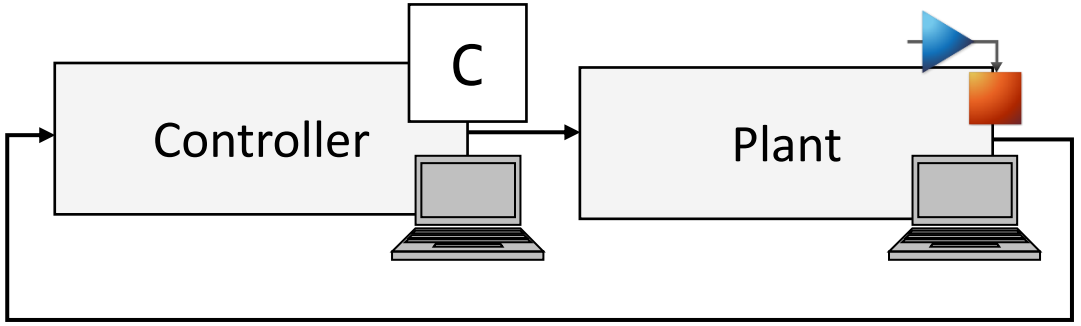
HIL simulator platform
(Real-time capable)

Target HW
(e.g., Microcontroller)

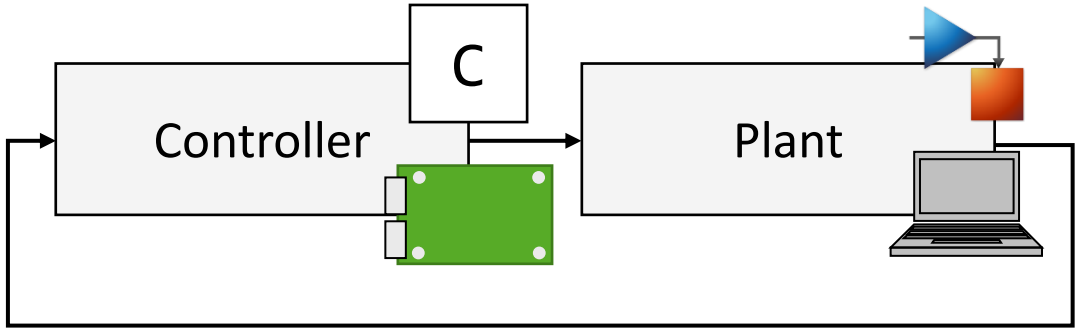
In-the-loop testing (XIL)



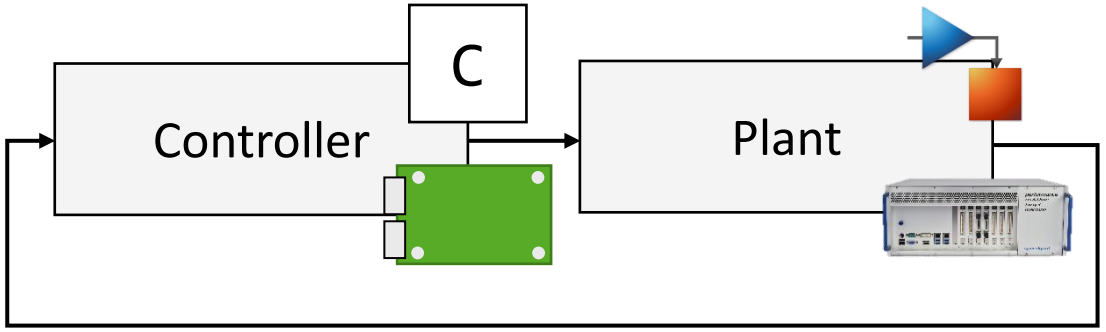
Model in the loop (MIL)



Software in the loop (SIL)

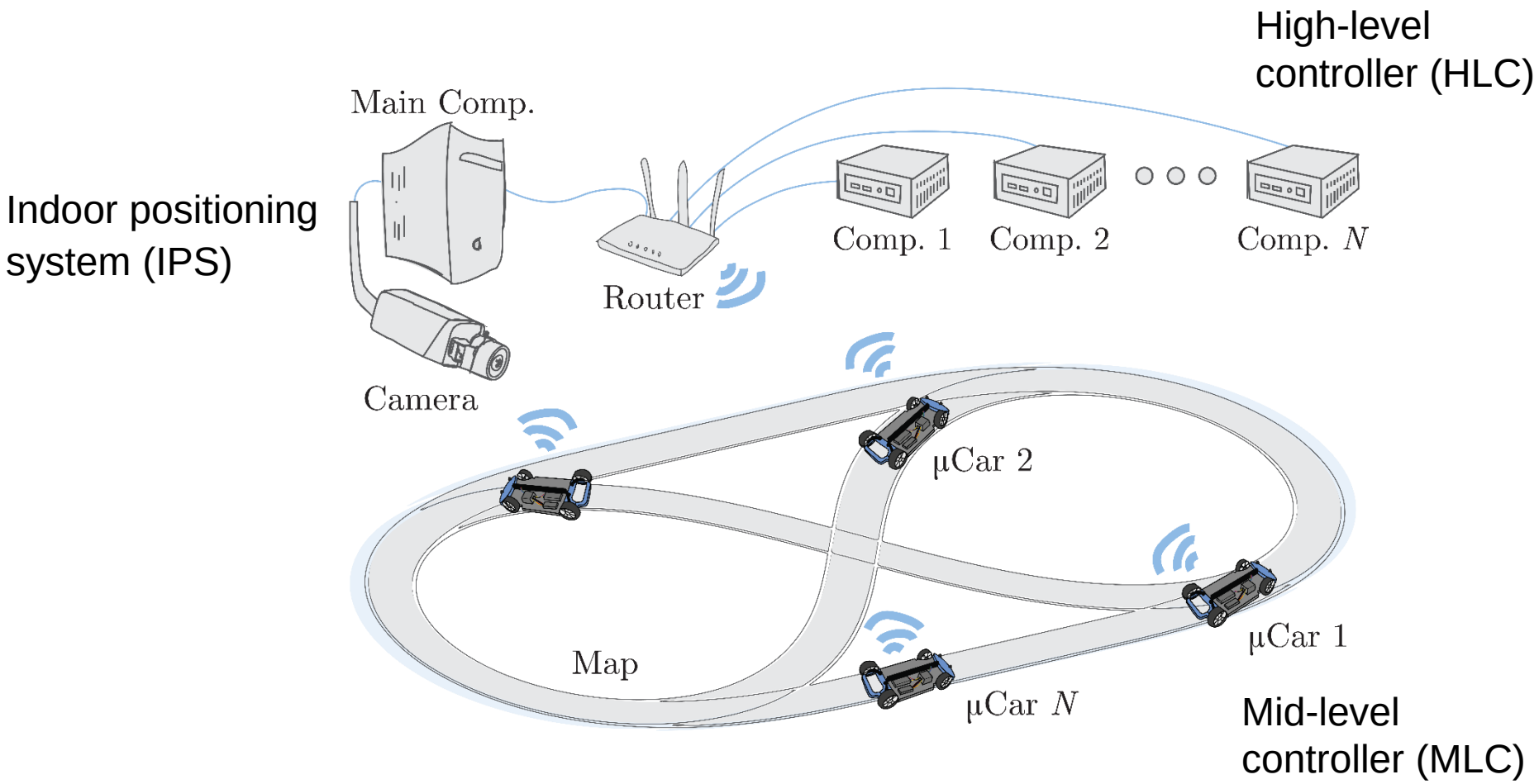


Processor in the loop (PIL)

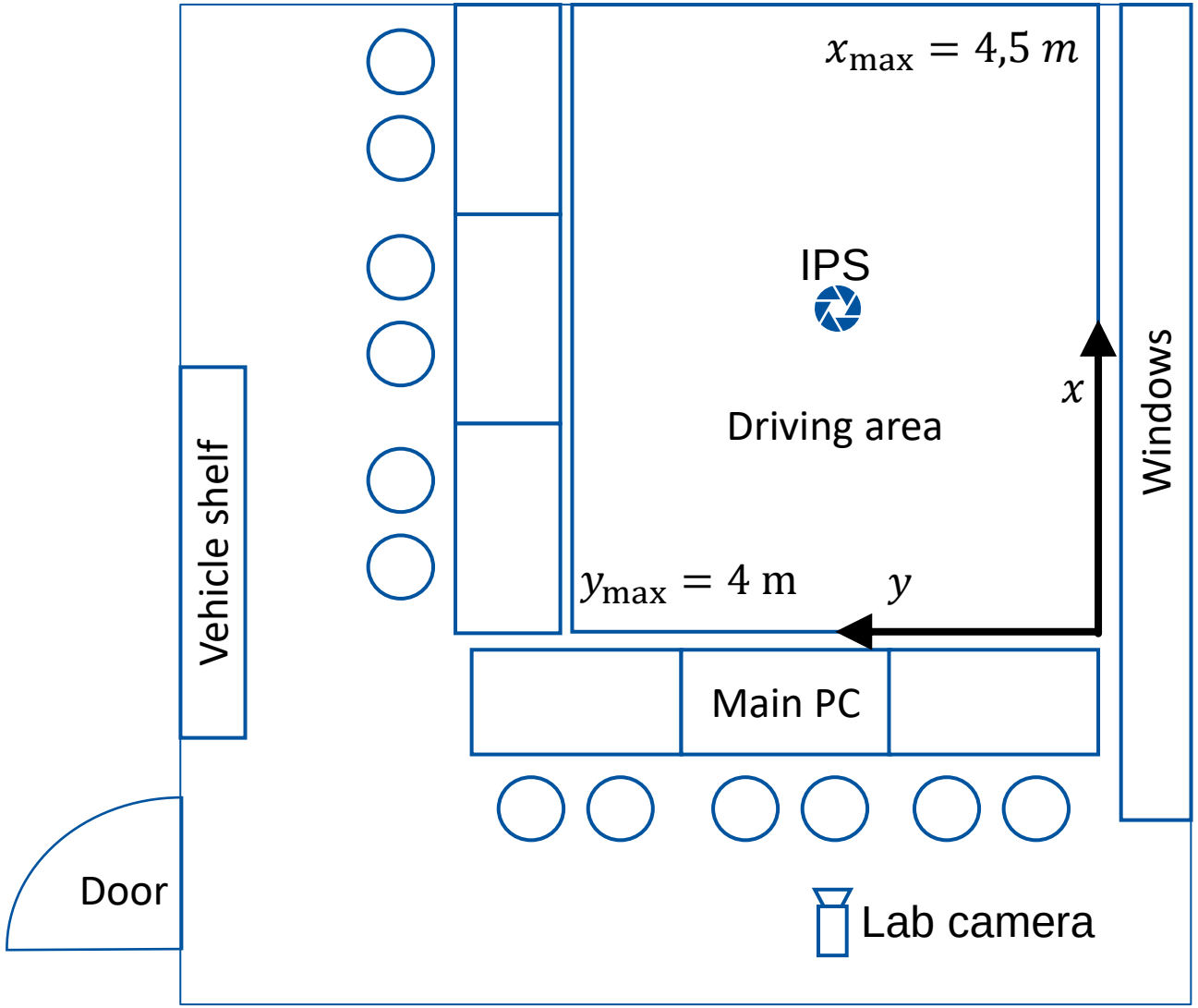


Hardware in the loop (HIL)

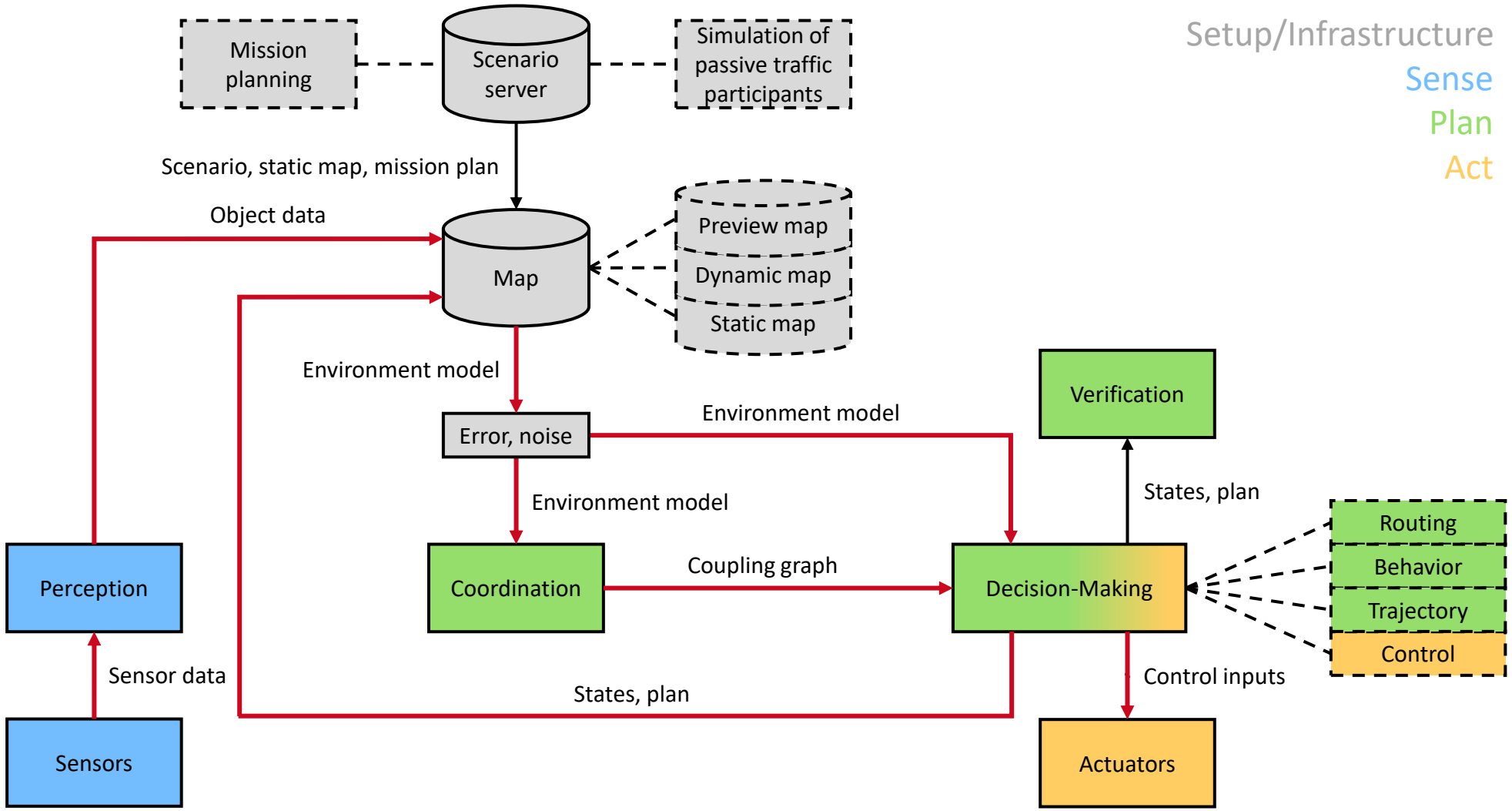
CPM Lab components



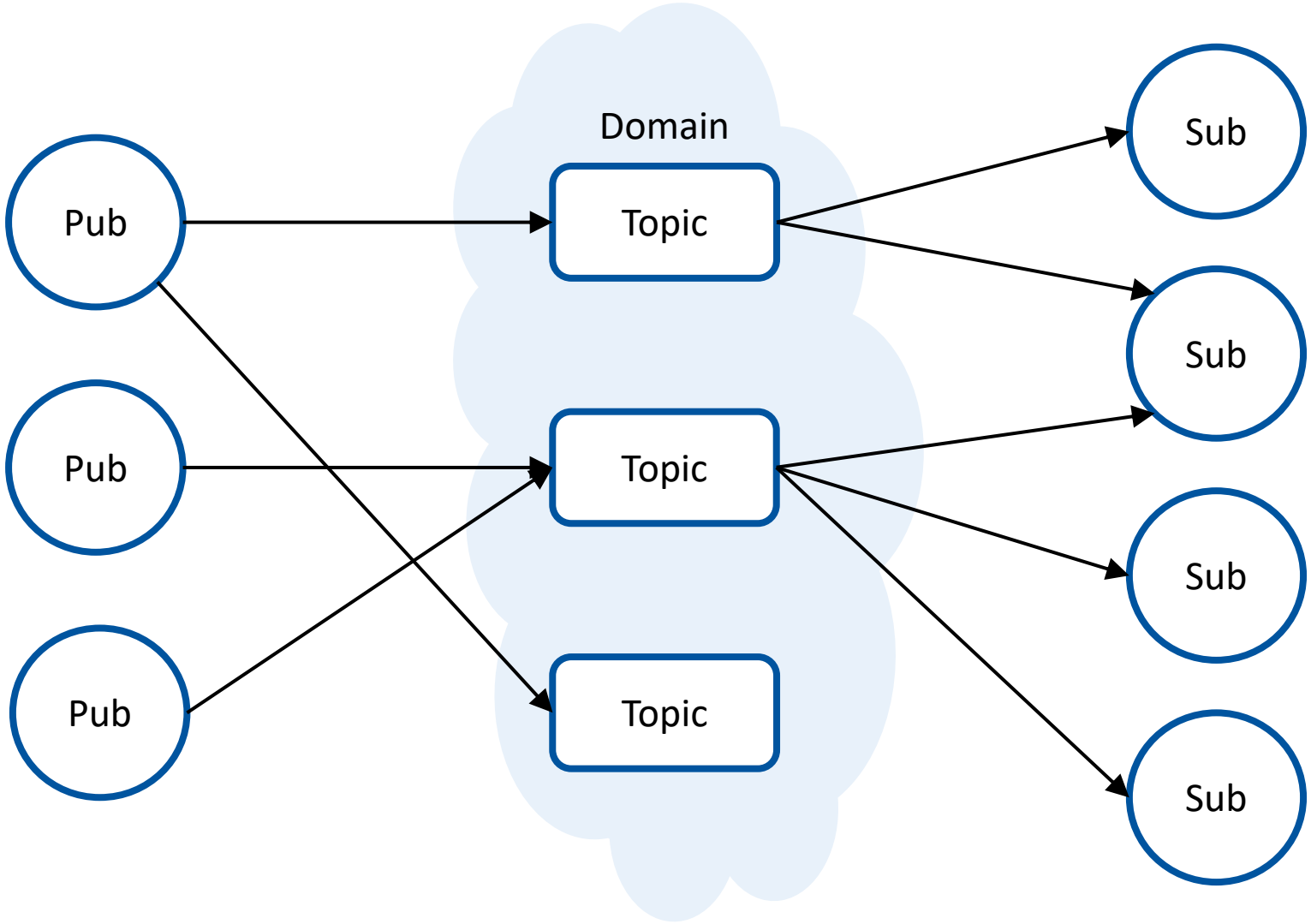
CPM Lab room


$x_{\max} = 4{,}5\ m$
IPS
Driving area
x
Windows
Vehicle shelf
$y_{\max} = 4\ \mathrm{m}$
y
Main PC
Door
Lab camera

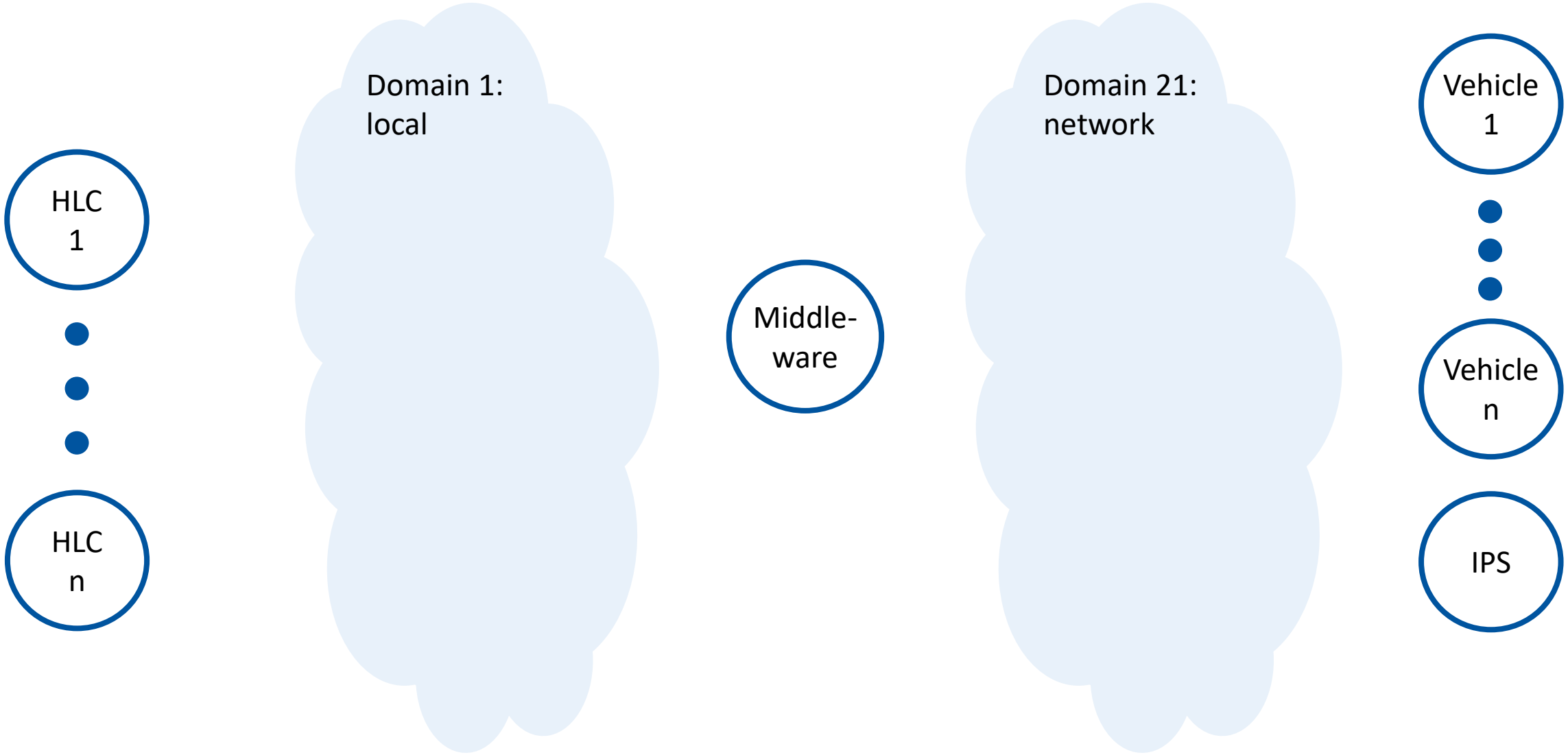
CPM Lab architecture: communication



Data Distribution Service



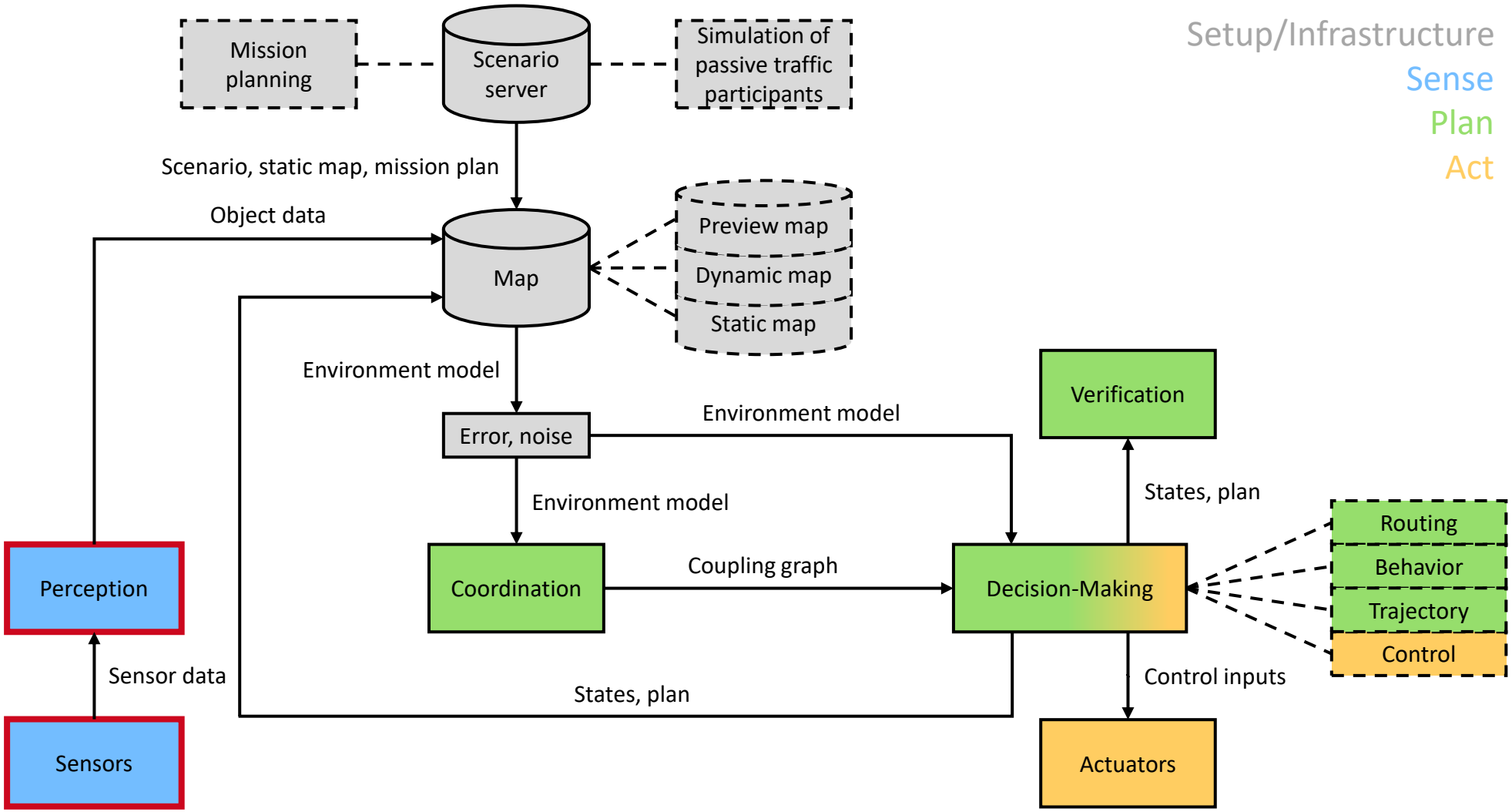
Data Distribution Service – CPM Lab



Data Distribution Service – message format

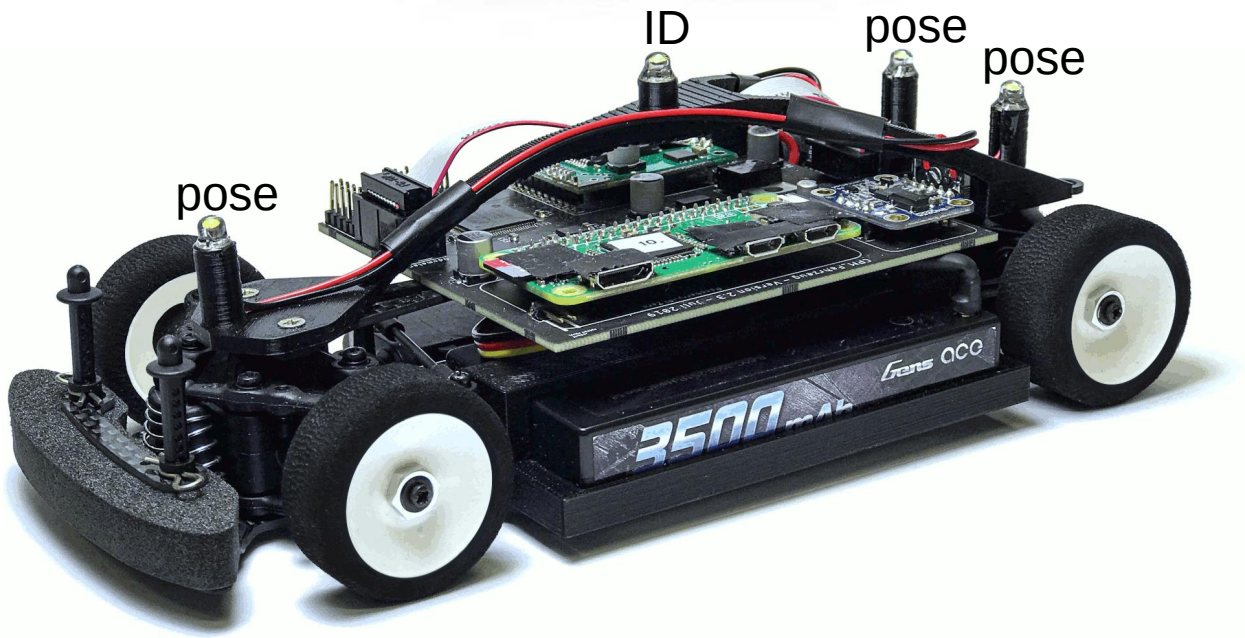
- Message datatype on topic defined with Interactive Data Language (IDL)
- Examples
  - Pose ID, Header, $x(t), y(t), \psi(t)$
  - State ID, Header, $x(t), y(t), \psi(t), s(t), v(t), a(t), \ldots$
  - DirectControl ID, Header, motor_throttle, steering_servo
- Header
  - create_stamp
  - valid_after_stamp

CPM Lab architecture: indoor positioning system



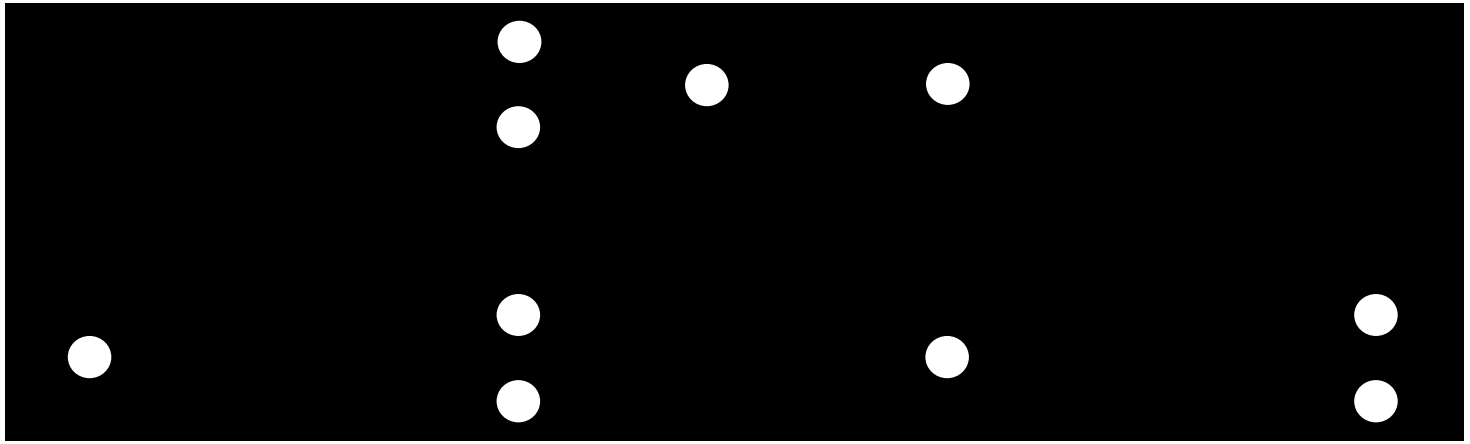
Indoor positioning system

- Camera mounted on the ceiling
  - Top-down view on the field
  - Streams images with 50 Hz
- Vehicles are equipped with LEDs
  - 3 LEDs to encode pose
    - Non-equilateral triangle
  - 1 LED to encode ID
    - Frequency

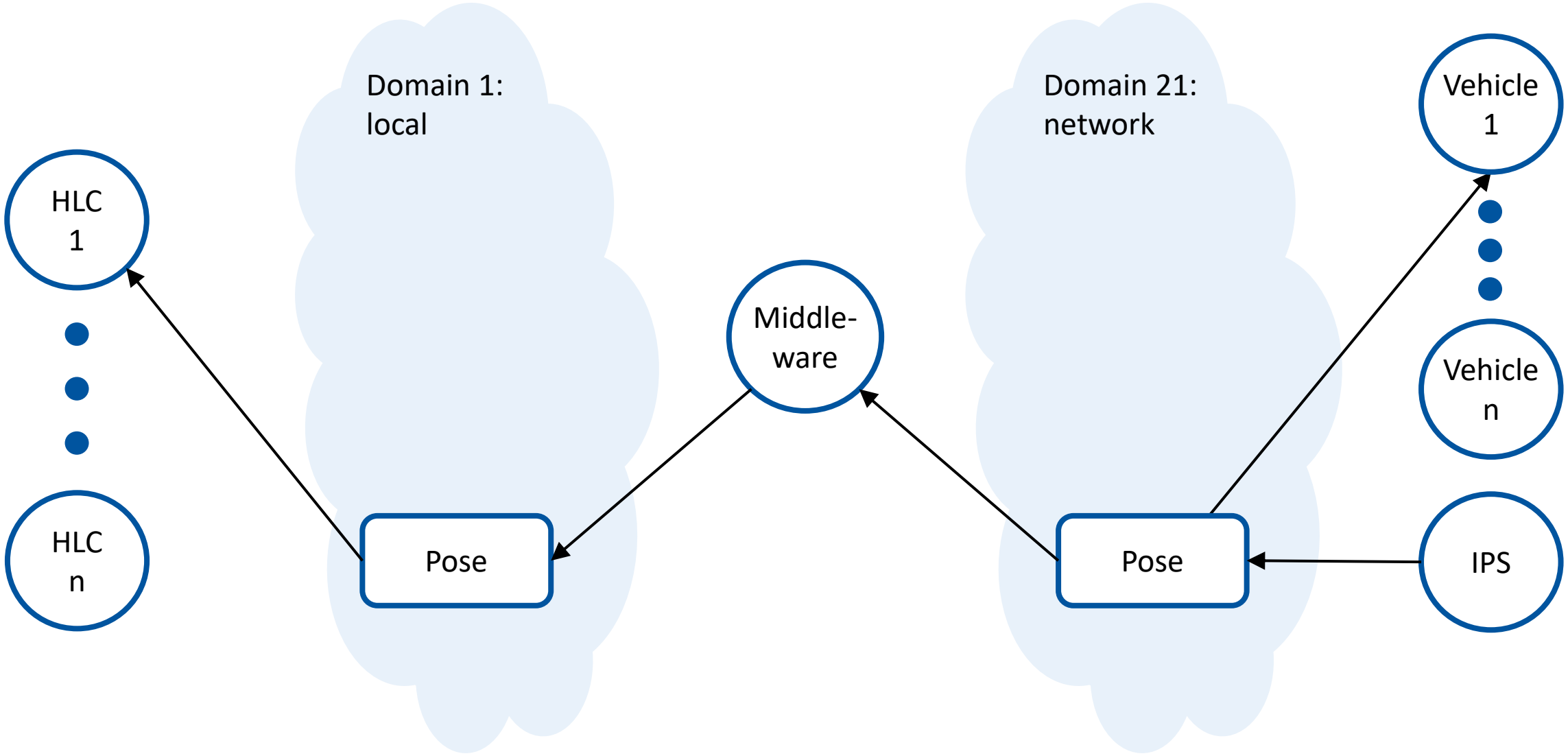


Indoor positioning system

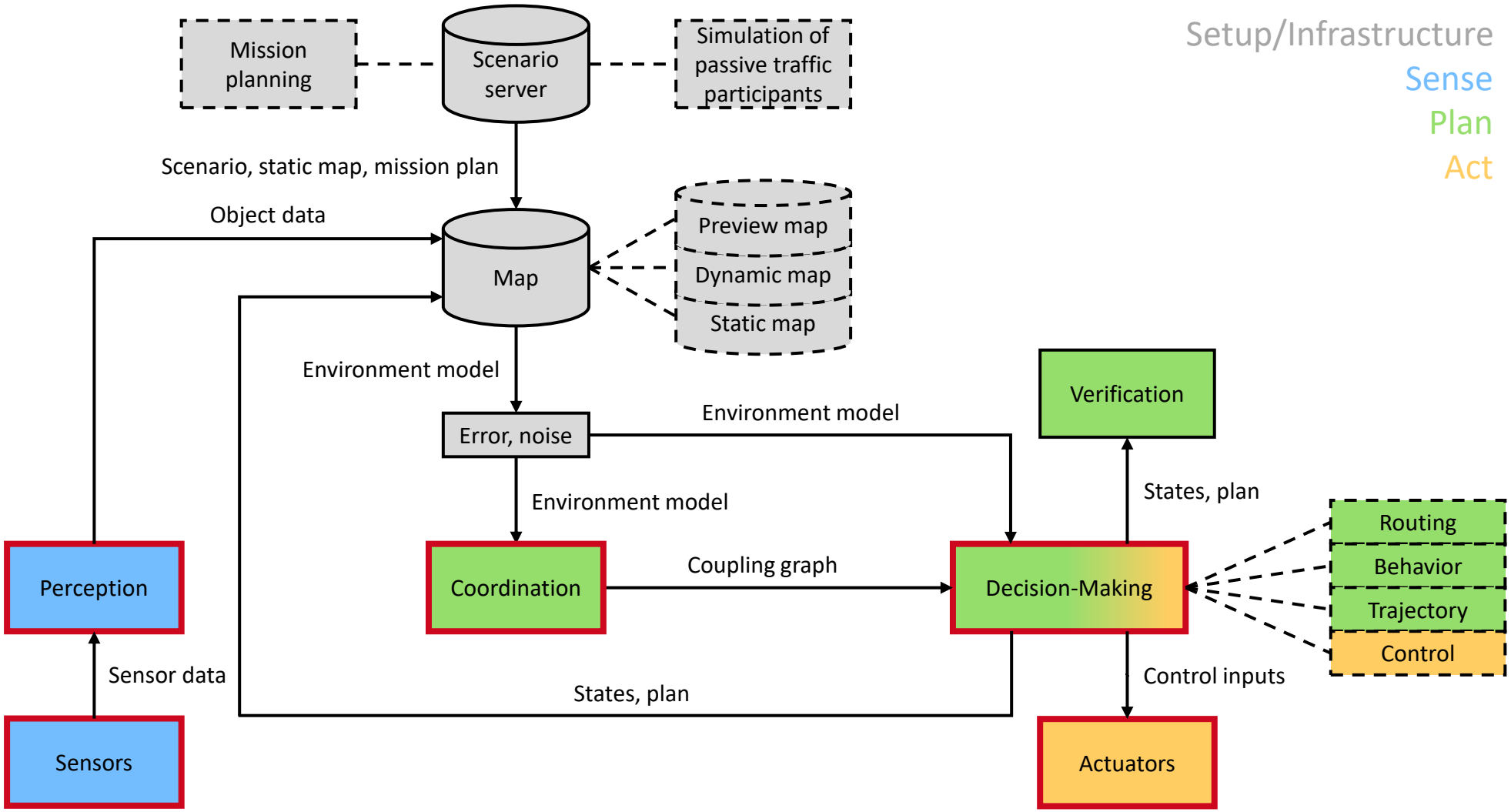
- Low exposure time leads to high contrast of LEDs to background
- IPS pipeline:
  1. Extract LED points of camera image
  2. Transform image coordinates to world coordinates
  3. Find vehicles
  4. Map found vehicles to past vehicles
  5. Extract position, orientation, and ID



Data Distribution Service – CPM Lab

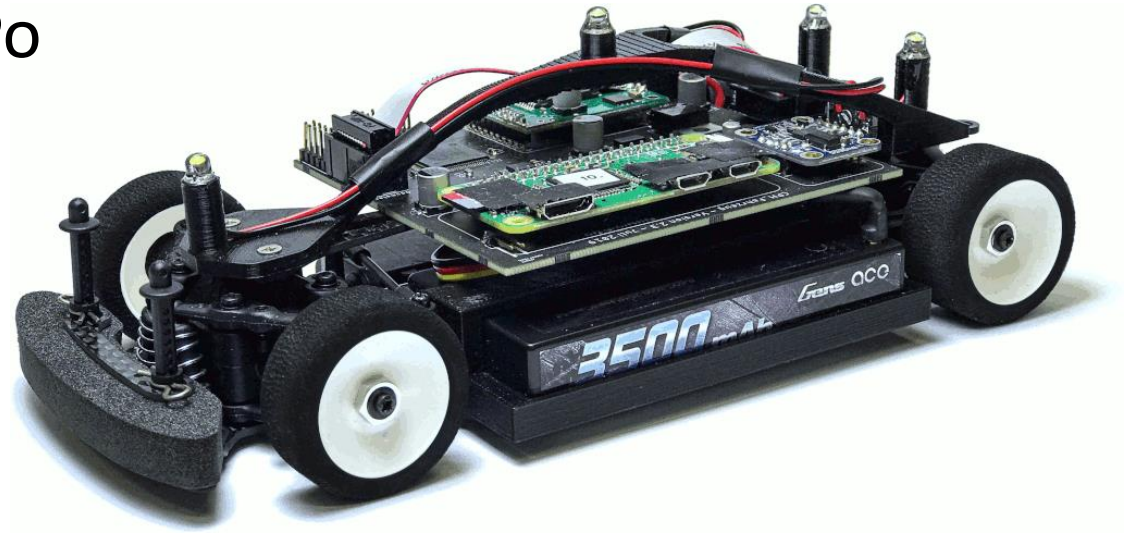


CPM Lab architecture: high level controller and vehicle

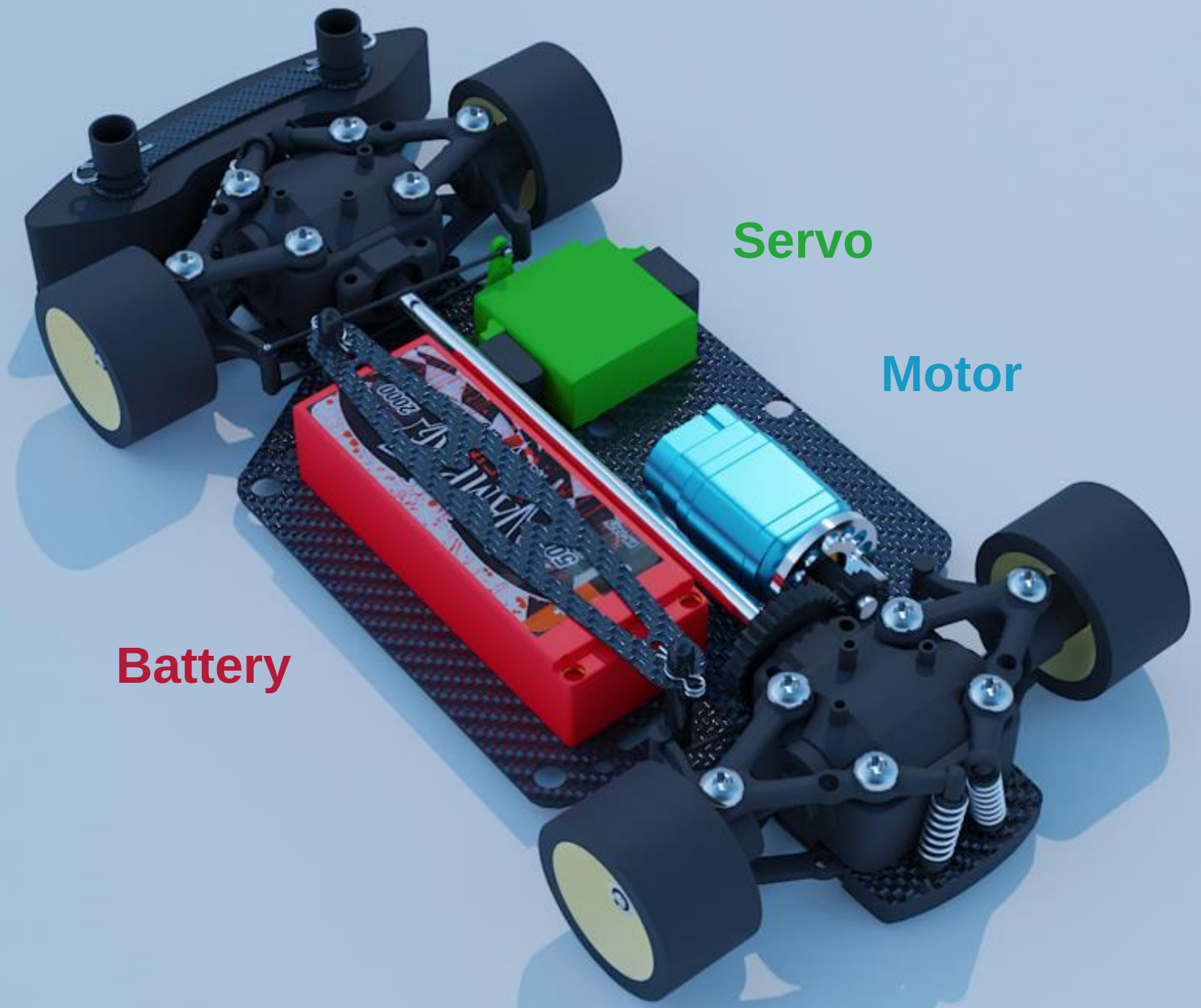


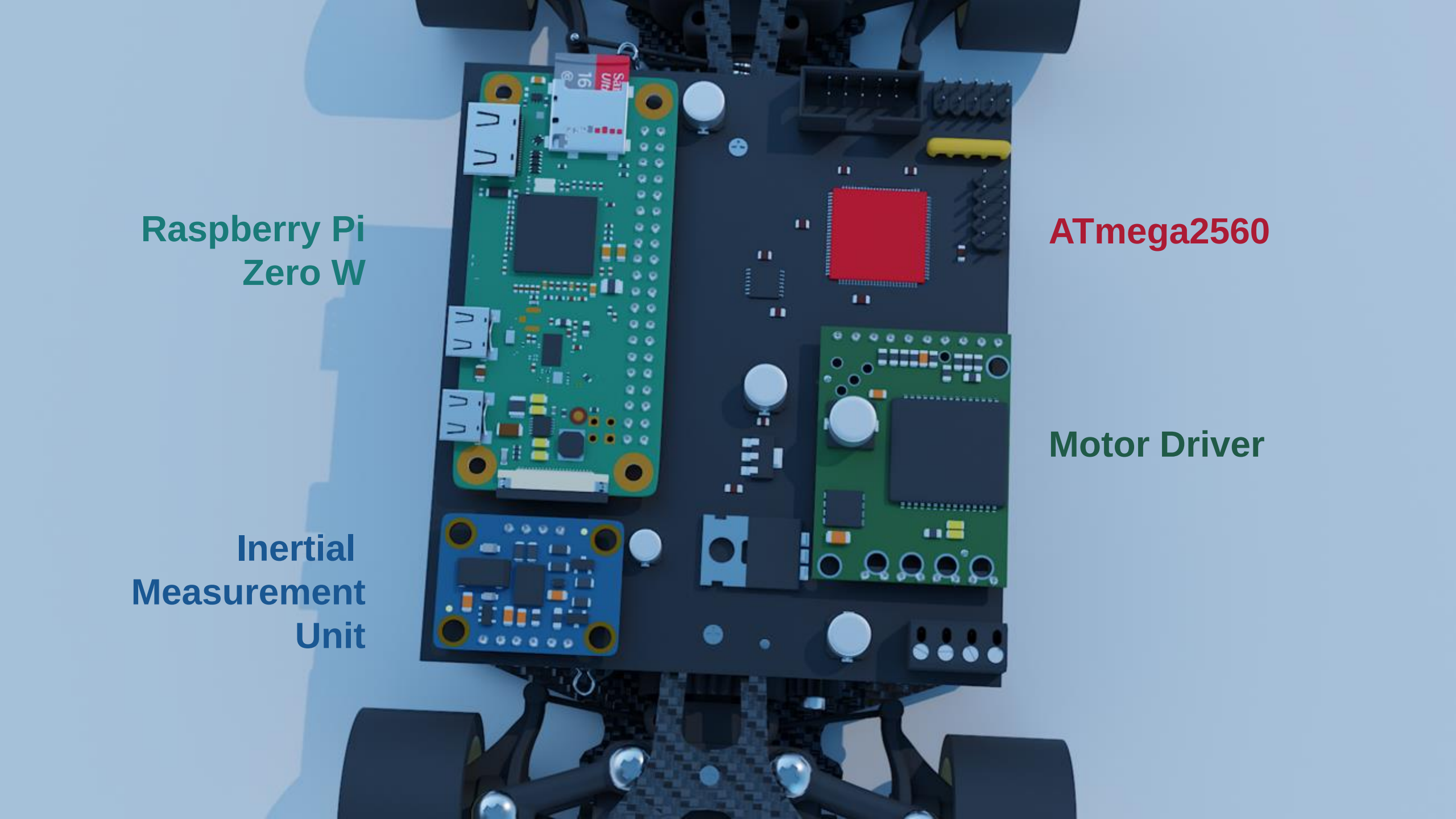
Vehicle – µCar

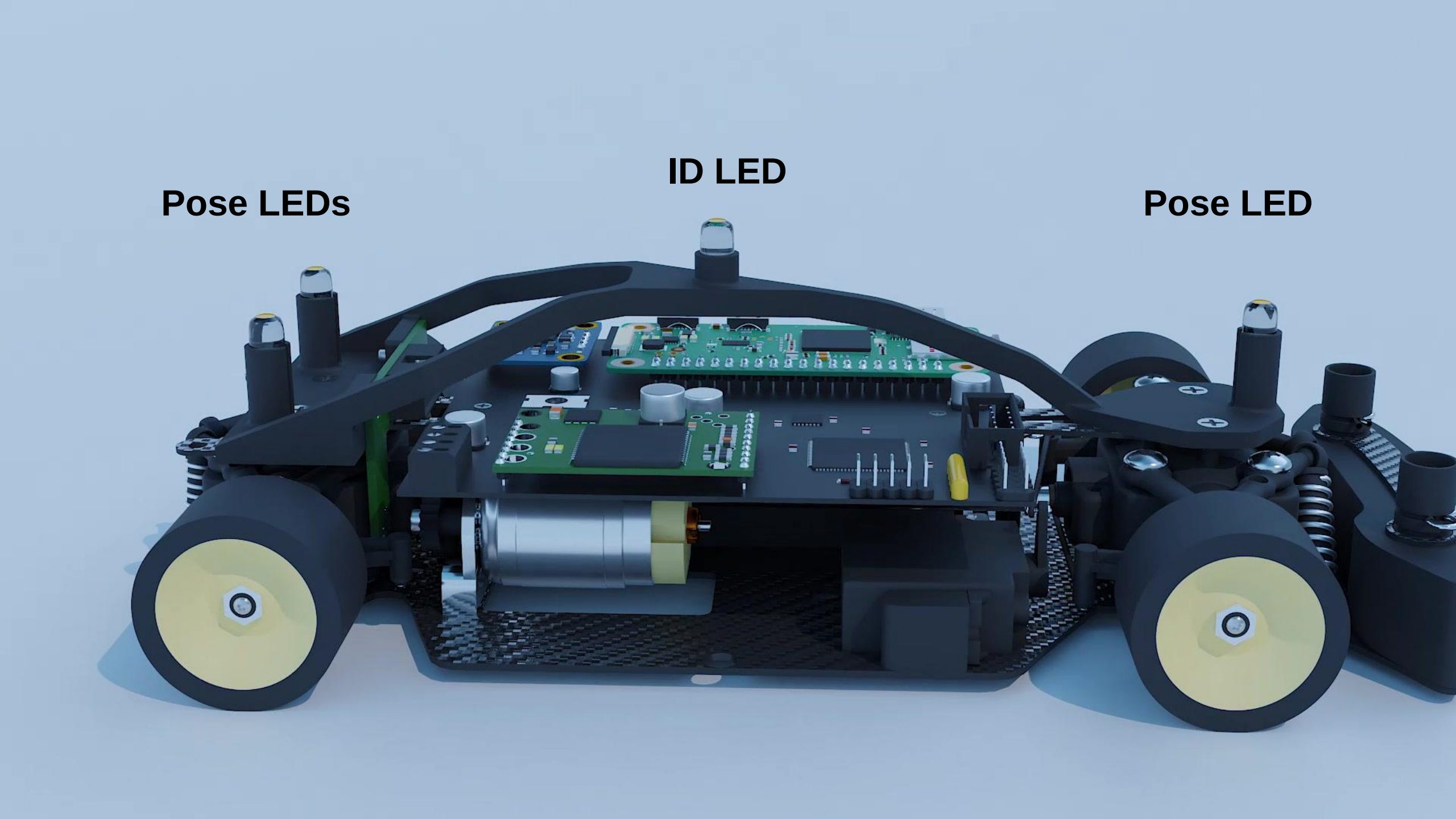
- Mechanical platform XRAY M18 PRO LiPo



- Sensors
  - IMU (DeboSens BNO055)
  - Odometer (3 Hall-effect sensors + magnet)
- Actuators
  - Servo motor (Hitec D89MW) for steering
  - Brushless DC Motor (Pololu VNH5019) for propulsion
- Computation
  - Mid-level controller (Raspberry Pi Zero W) for WLAN, clock synchronization, sensor fusion, trajectory following control and path tracking
  - Low-level controller (ATmega2560) for reading sensor data and actuation

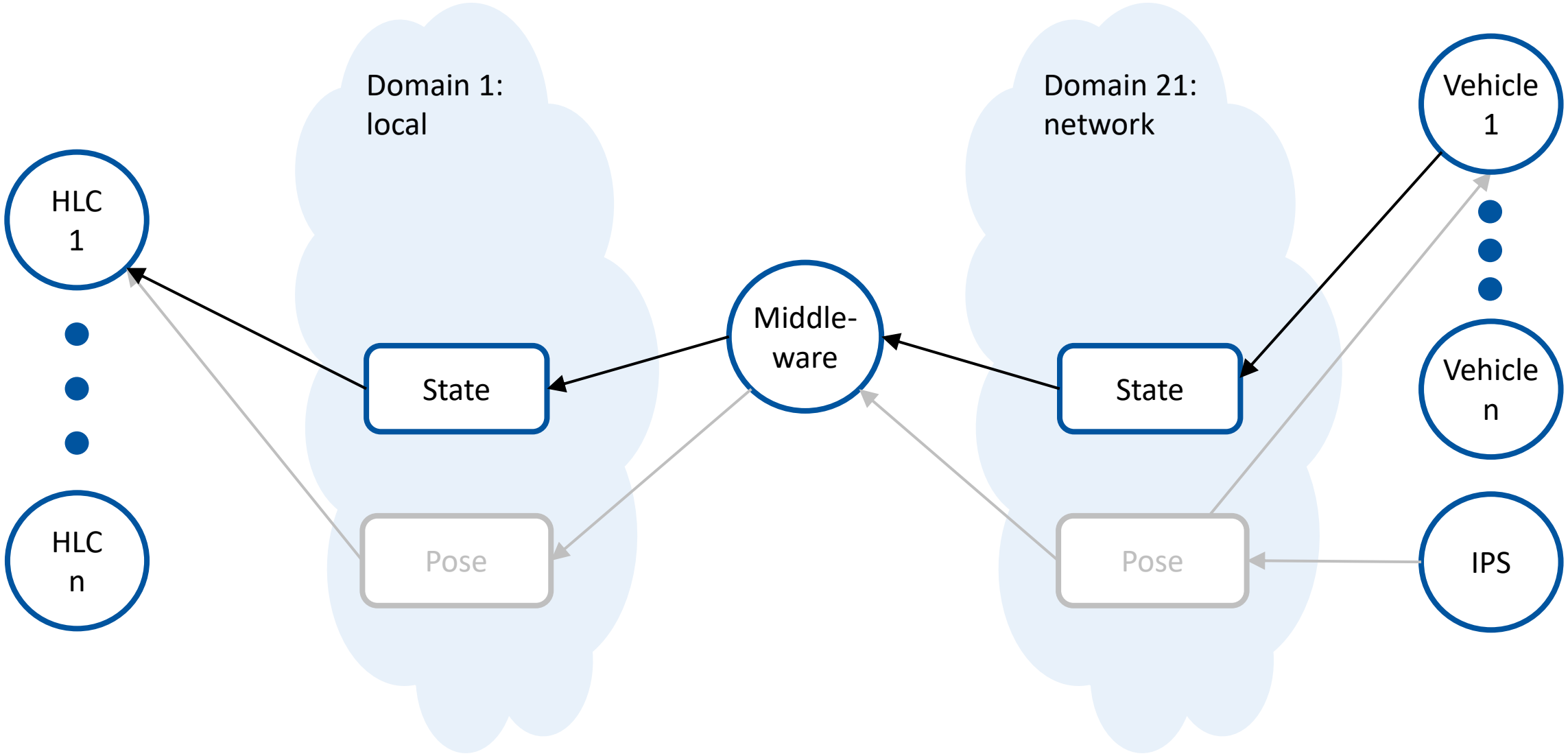

Servo
Motor
Battery


Raspberry Pi
Zero W
ATmega2560
Motor Driver
Inertial
Measurement
Unit


ID LED
Pose LEDs
Pose LED

Hall Effect Sensors and Magnet

Data Distribution Service – CPM Lab

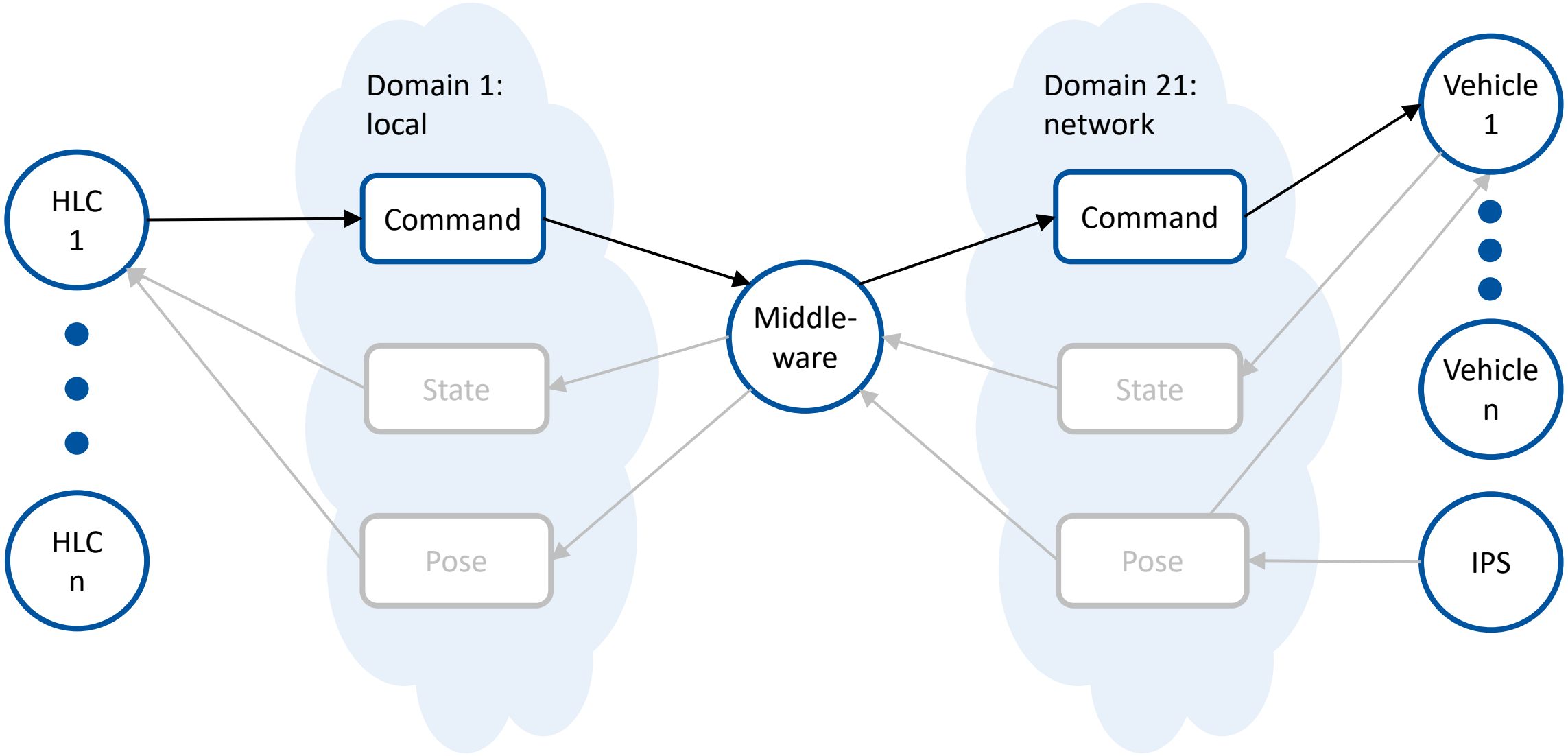


High-level controller

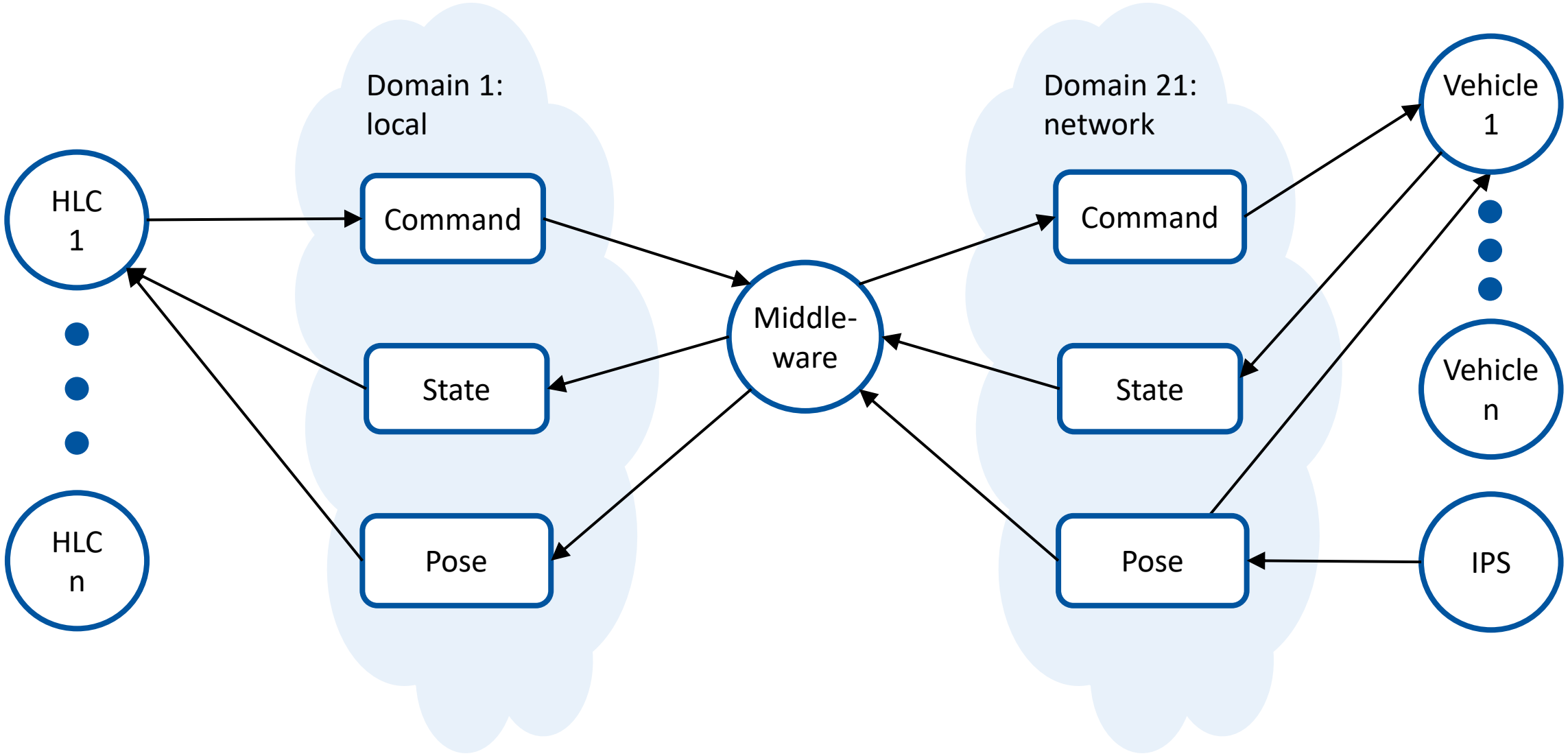
- Direct control
  - Motor as input
  - Steering servo as input
- Path tracking
  - Path and speed as inputs
  - Stanley controller [1]
- Trajectory following
  - Trajectory as input
  - Model predictive controller on Raspberry Pi Zero W using a simplified bicycle model [2]
- HLC in any programming language, only interface to DDS necessary

[1] Hoffmann, Gabriel M., et al. "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing." 2007 American control conference
[2] Scheffe, Patrick, et al. „Networked and Autonomous Model-scale Vehicles for Experiments in Research and Education“ 2020 IFAC World Congress

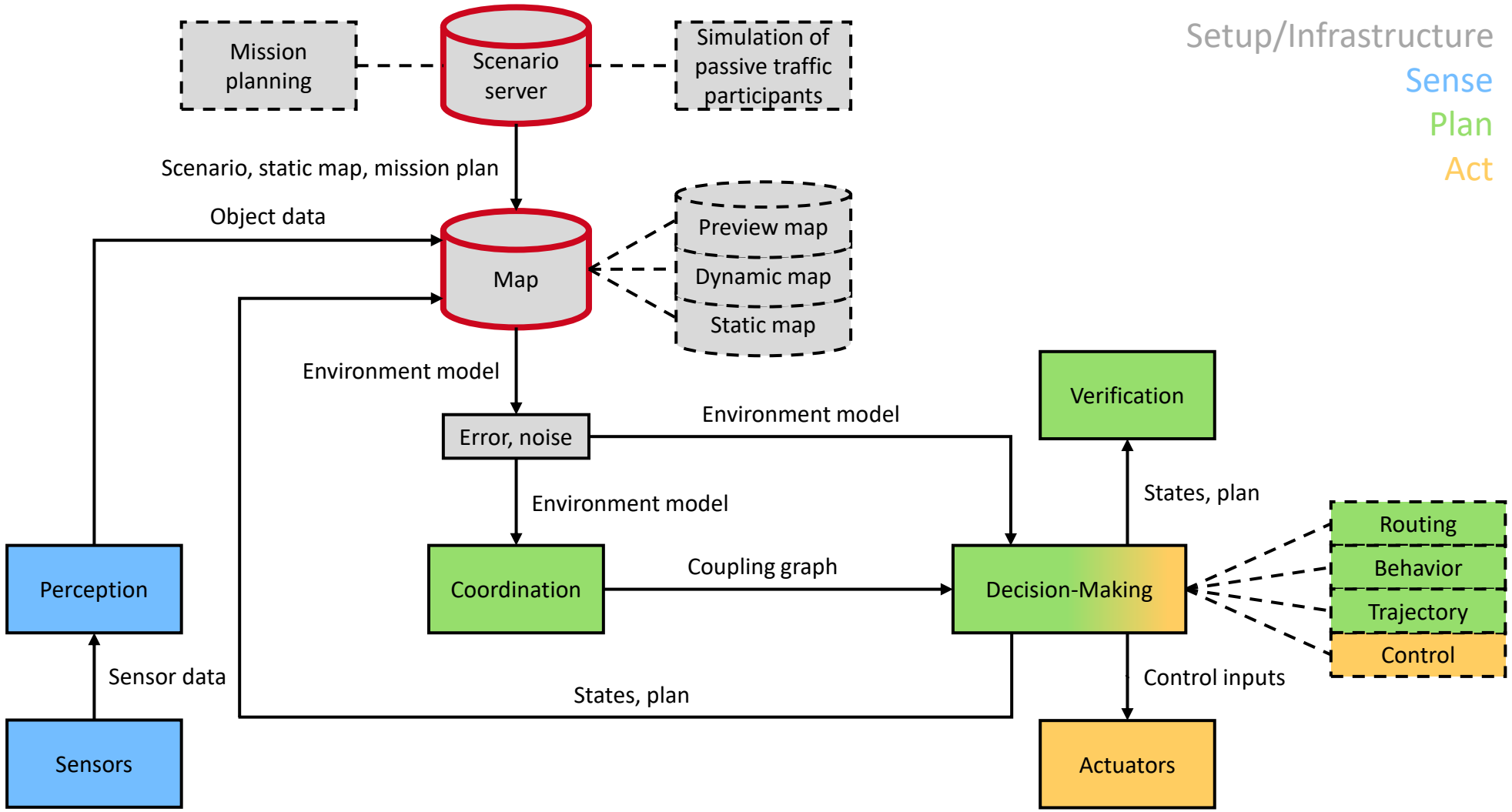
Data Distribution Service – CPM Lab



Data Distribution Service – CPM Lab

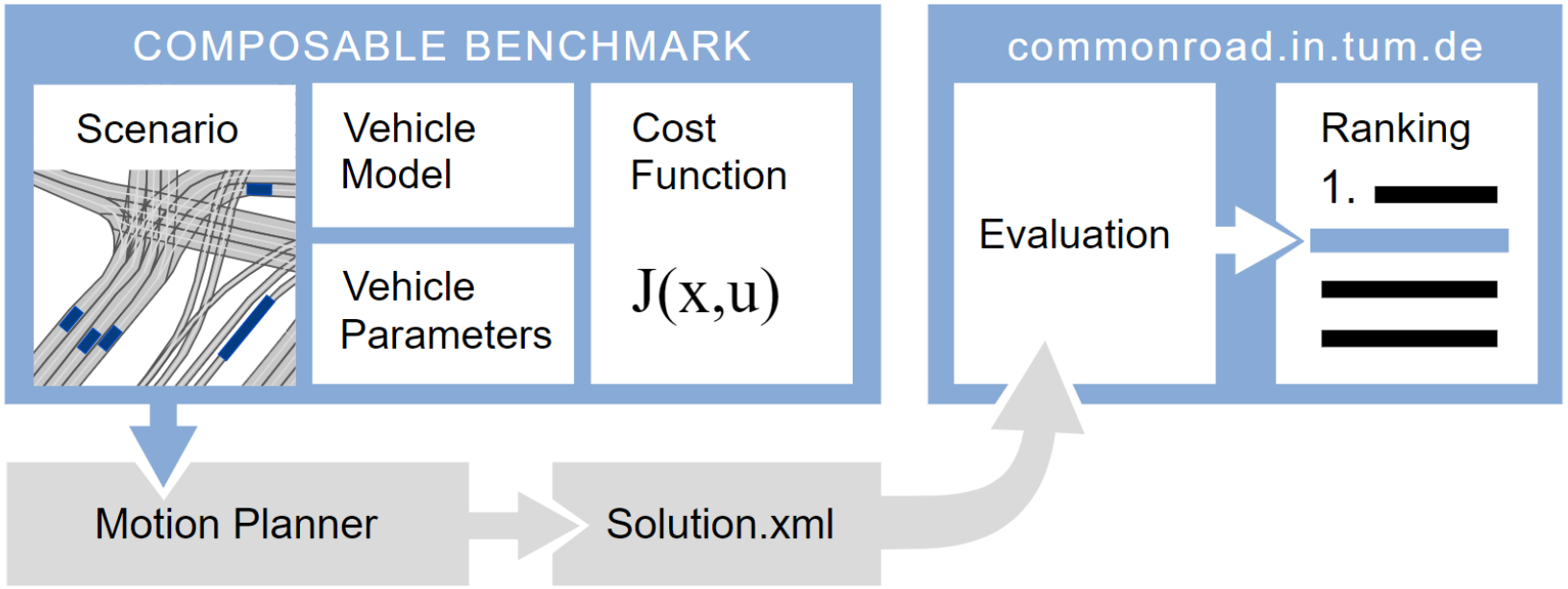


CPM Lab architecture: Scenario definition



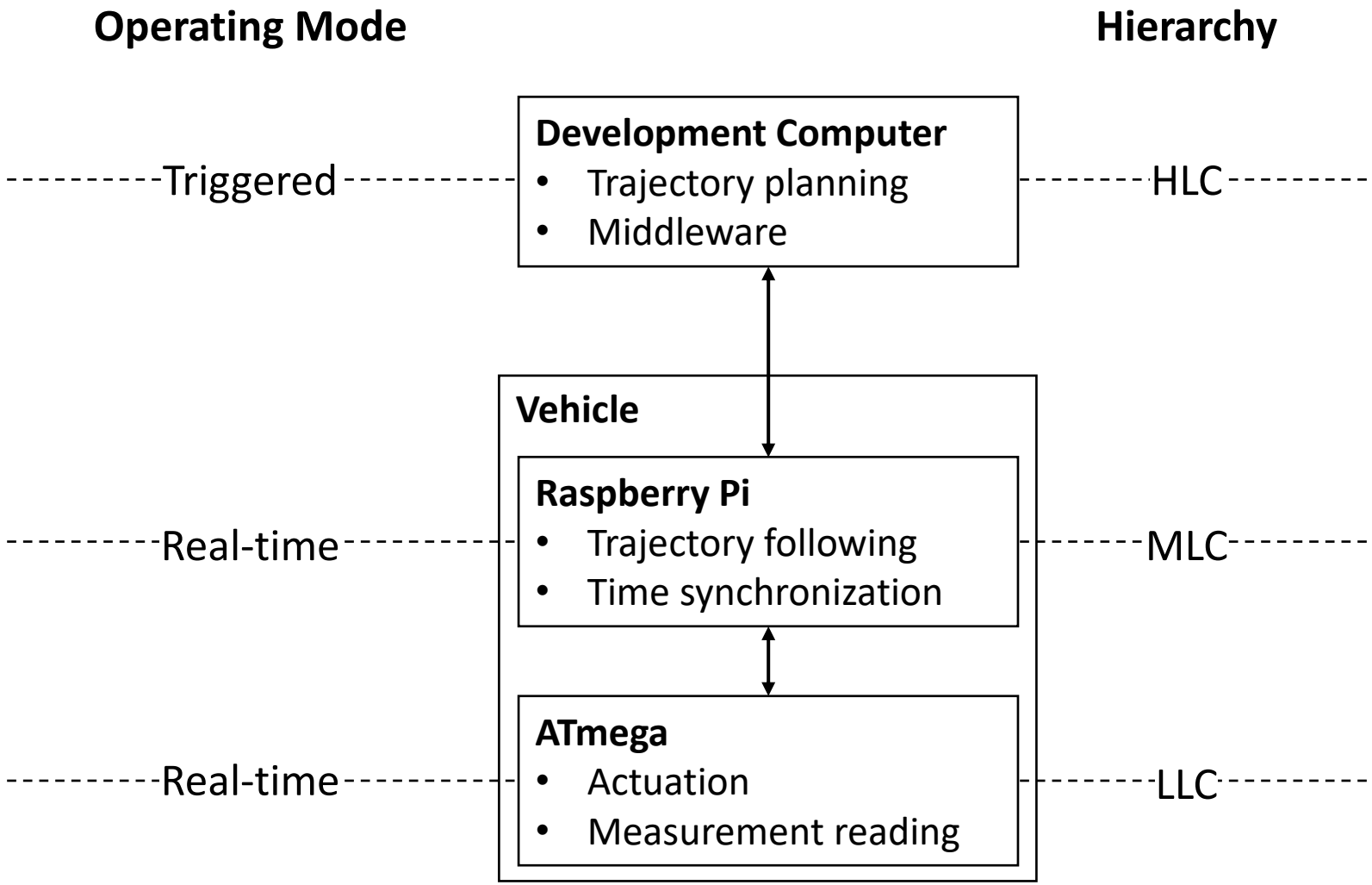
Scenario definition: CommonRoad [1]

- Lanelets
  - Left and right bounds
  - Connections
- Static obstacles
  - Dimensions
- Dynamic obstacles
  - Dimensions
  - Trajectory
- Planning problem
  - Initial state
  - Goal state

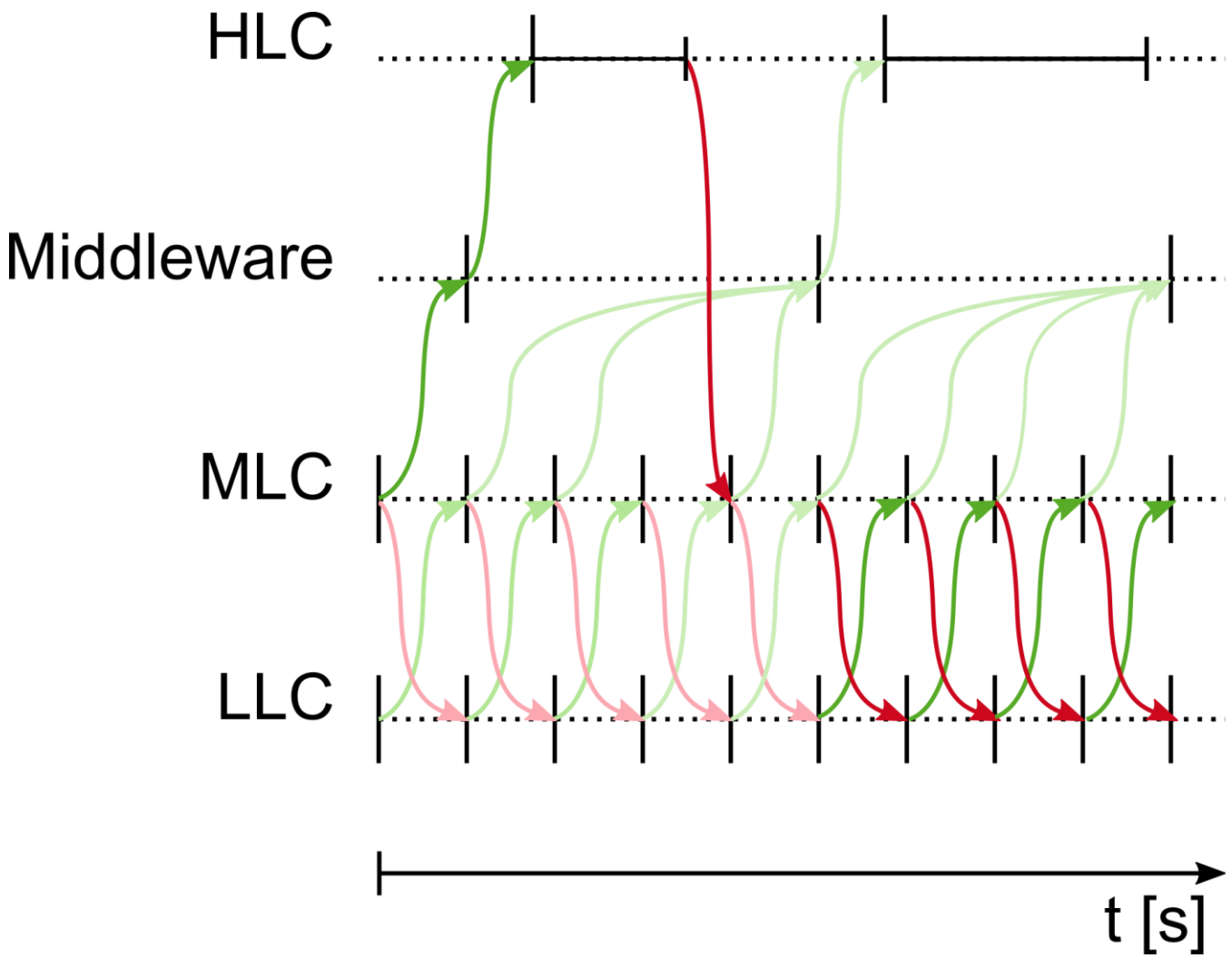


[1] Althoff, Matthias, Markus Koschi, and Stefanie Manzinger. "CommonRoad: Composable benchmarks for motion planning on roads." 2017 IEEE Intelligent Vehicles Symposium (IV)

Experimental concept



Experimental concept



Process model – eXtreme Programming

- Evolutionary development in small increments
- Most important features first
- Test-first, automated testing
- Have working code
- Pair programming

- No explicit design, documentation, review → suitable for small projects

Git commands

1. **git clone**
   To copy a git repository from remote source
2. **git status**
   To check the status of files you've changed in your working directory
3. **git add**
   Adds changes to stage/index in your working directory
4. **git commit**
   Commits your changes and sets it to new commit object for your remote
5. **git push**
   Push your changes to remote
6. **git pull**
   Pull changes from the remote

You can find a more elaborate introduction to git in the moodle room.

cpm ubuntu 64 - VMware Workstation 16 Player (Non-commercial use only)

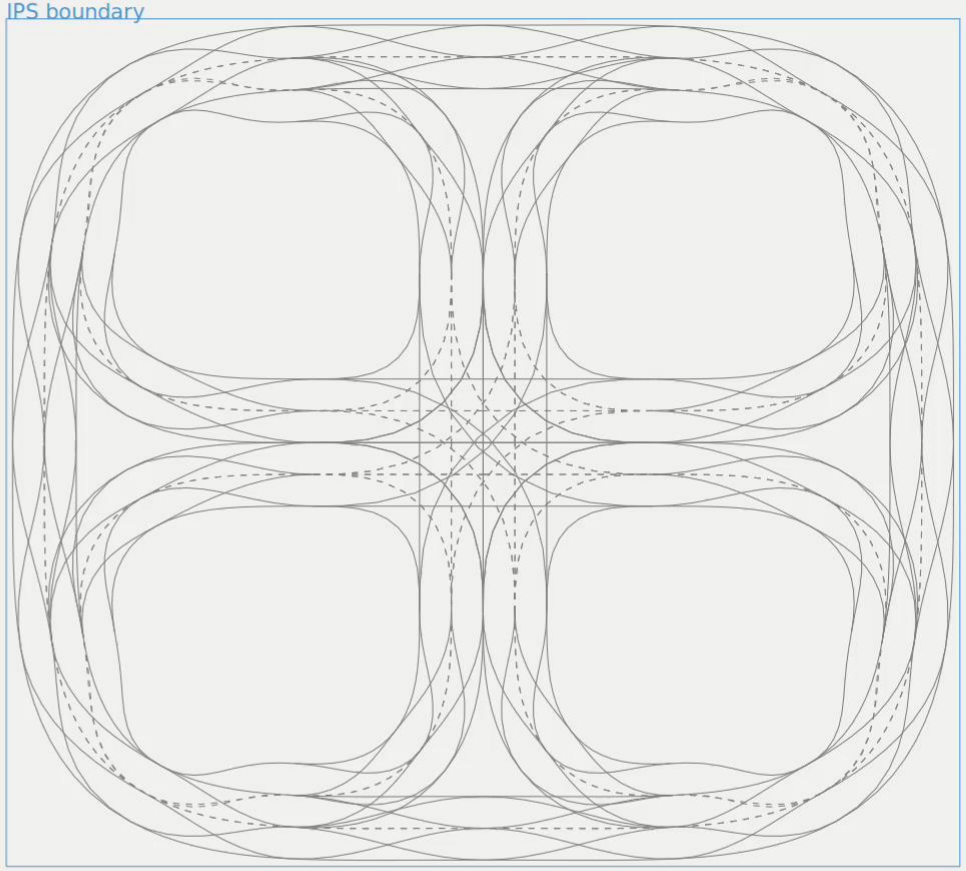
Player

Activities | Lab Control Center | Wed 13:15 | en

lab_control_center

Parameters | Map View

IPS boundary



Setup | Commonroad | Manual Control | Parameters | Timer | Logs | LCC Errors

Lab mode OFF

Diagnosis OFF

Script Path

../high_level_controller/examples/cpp/central_routing/build/central_routing

Open

Script Command Line Parameters

Simulated Vehicles

Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4	Vehicle 5	Vehicle 6	Vehicle 7	Vehicle 8
Simulate	Simulate	Simulate	Simulate	Simulate	Simulate	Simulate	Simulate
Vehicle 9	Vehicle 10	Vehicle 11	Vehicle 12	Vehicle 13	Vehicle 14	Vehicle 15	Vehicle 16
Simulate	Simulate	Simulate	Simulate	Simulate	Simulate	Simulate	Simulate
Vehicle 17	Vehicle 18	Vehicle 19	Vehicle 20				
Simulate	Simulate	Simulate	Simulate				

Select none | All simulated

Simulated Time OFF

Deploy Remote OFF

Lab Camera

Record OFF

Camera Status Not Initialized

Kill

Deploy

Reset view | Exp time: — | HLCs online: 0 | Reboot HLCs | HLC RTT (ms): — | Vehicle RTT (ms): —