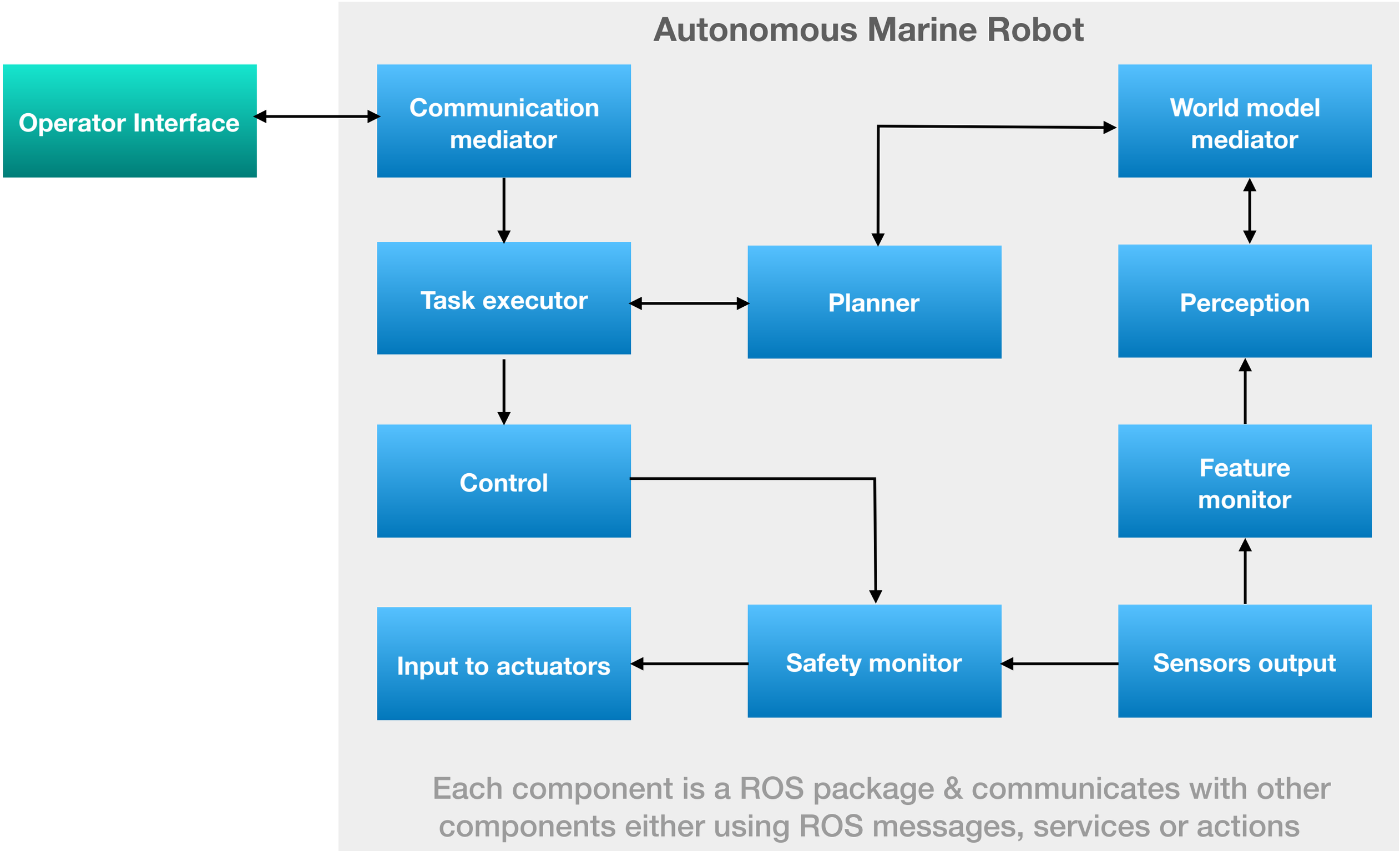
# Software component modelling for autonomous marine robots

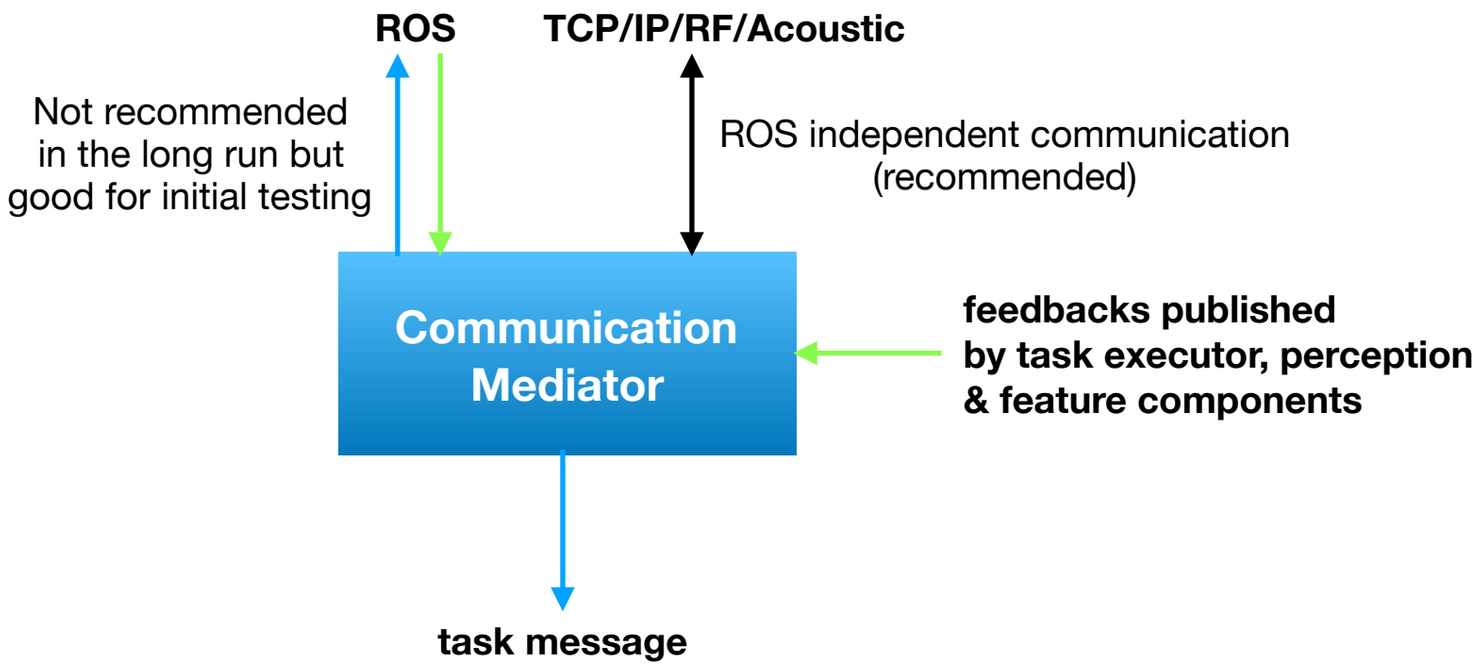
Lakshadeep Naik

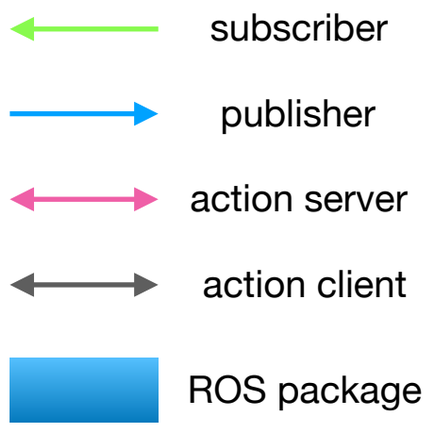
# Software component diagram



# Communication mediator

**Communication link between operator interface and robot**



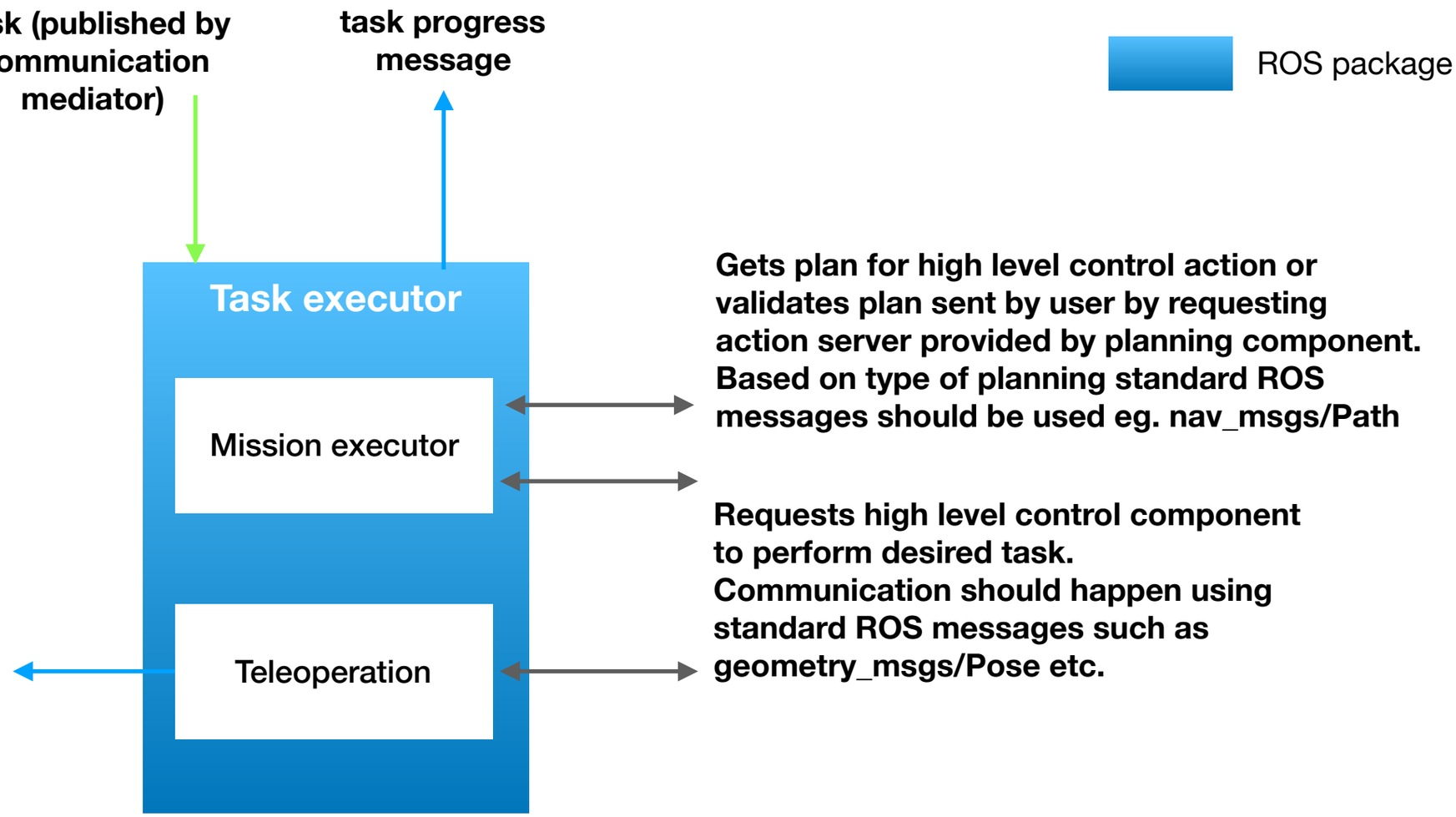




**Code structure**

- Adapter module for converting messages from channel specific to ROS format
- Multiple nodes for listening on each channel & publishing a task message

# Task executor

**Responsible for executing different tasks**



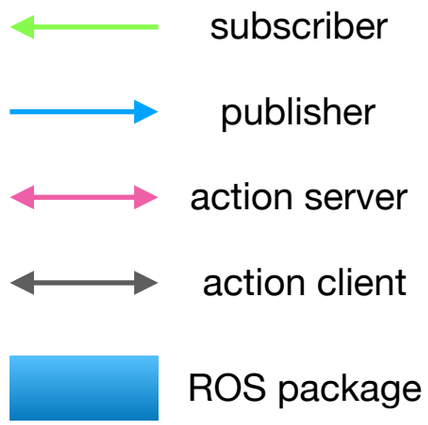

**Code structure**

- Bridge module for decoding a task message into missions, high level and low level control actions
- Bridge module for decoding a mission into high level control actions
- Module for high level control action execution (planning, validation & execution)
- Node for mission execution and teleoperation

# World model mediator & Planner

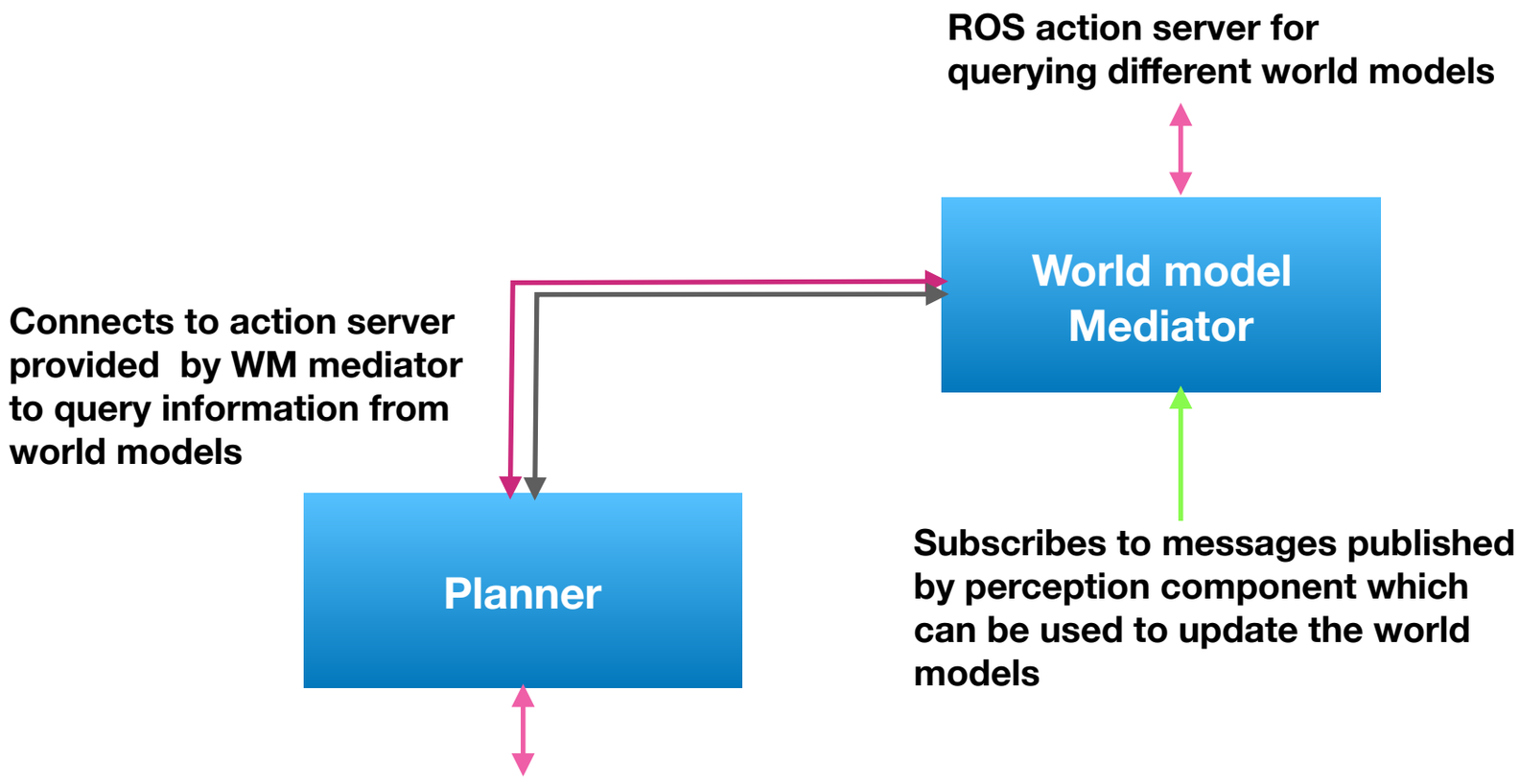
**World model mediator acts as communication link between different world models (knowledge about the environment)**

**Planner uses information from world models to create and validate plans**




## Code structure

- Module for each world model (create, update, query)
- Node for querying and updating world models
- Modules for planning
- Node for planning

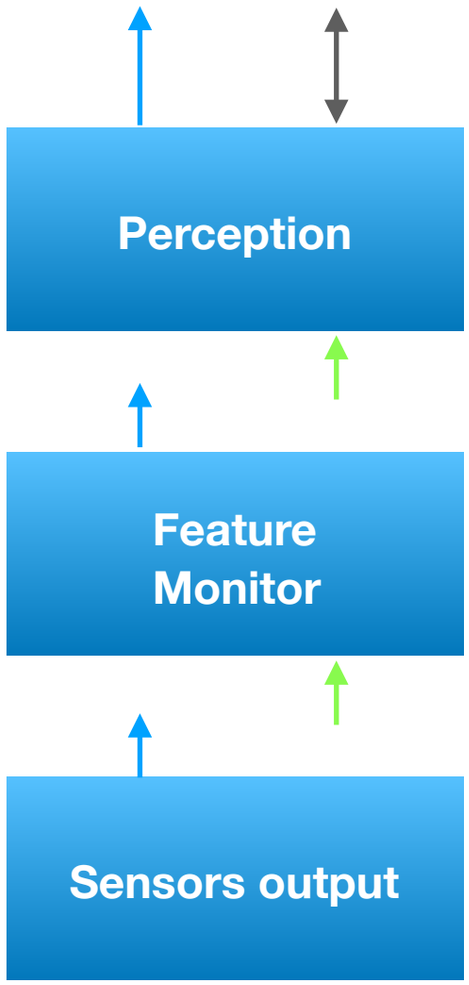


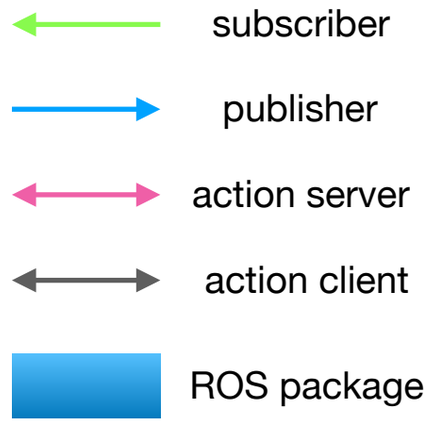
# Sensors, feature monitor & perception

**Reads information from sensors and use it for updating the world model**

**Subscribes to monitored features and use them to draw information about how world model has changed and publish it over ROS topic**

**Uses action client to request information from world models & associating it with monitored features**




**Subscribes to sensor messages & publish extracted features**

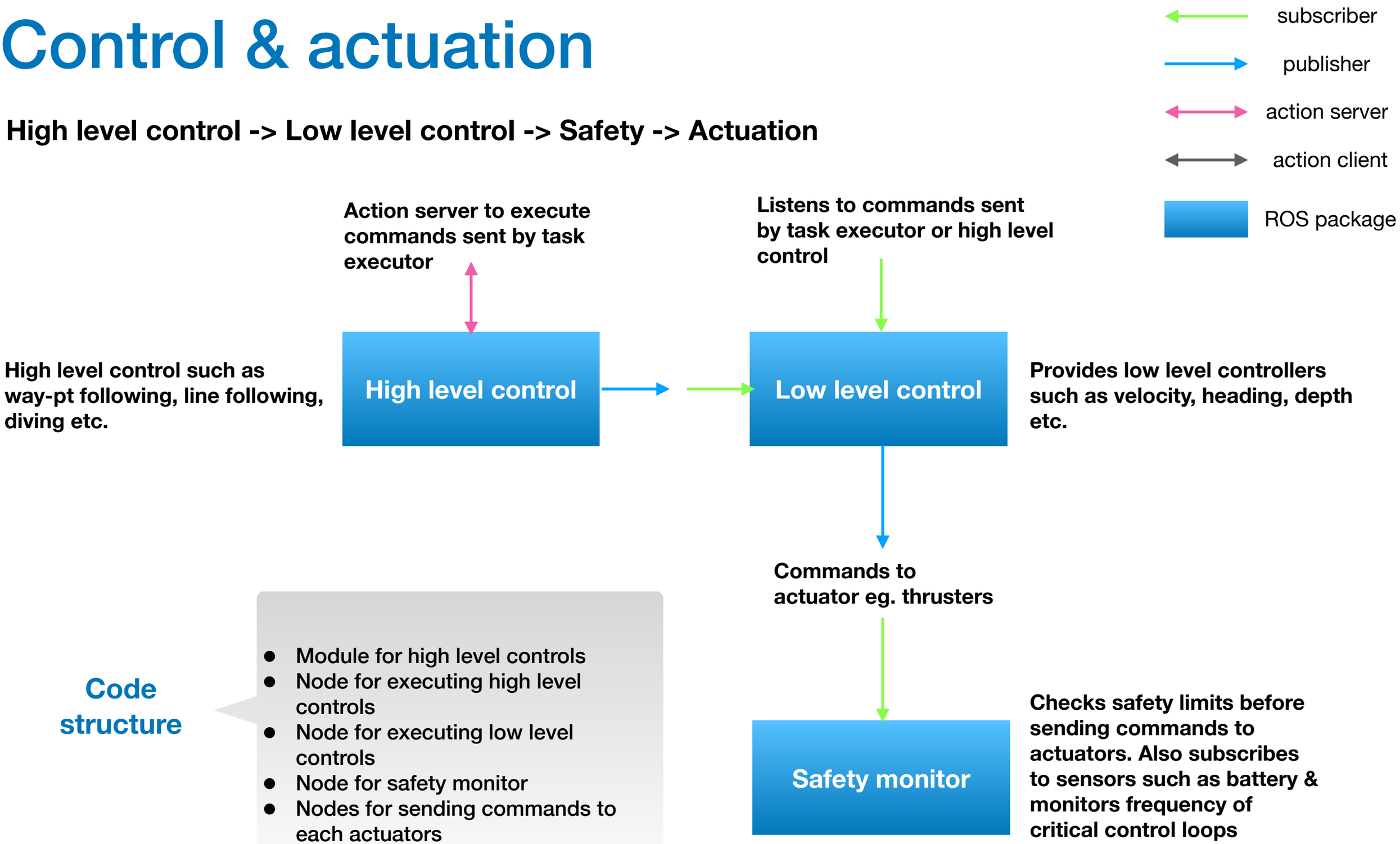
**Reads and publishes sensor data using standard sensor messages provided by ROS**

## Code structure

- Adapter modules for reading and converting each sensor data to ROS
- Nodes for reading & publishing data from each sensor
- Modules for feature monitoring eg. filters
- Node for different feature monitoring
- Modules for perception eg. localization
- Node for perception

# Control & actuation

**High level control -> Low level control -> Safety -> Actuation**




## Code structure

- Module for high level controls
- Node for executing high level controls
- Node for executing low level controls
- Node for safety monitor
- Nodes for sending commands to each actuators