# ROS2 Introduction

## TechDay - Tabit

**Hassan Umari - 26.09.2025**

# What is ROS?

# What is ROS?

- ROS is a framework for creating apps targeting Robotics

ROS = Libraries + Tools

📦 ⚒️

# What is ROS?

ROS = **Libraries** + Tools

📦 ⚒

Python

```
import rclpy
from tf2_ros import Buffer, TransformListener
```

C++

```
#include "rclcpp/rclcpp.hpp"
#include “std_msgs/msg/string.hpp”
```

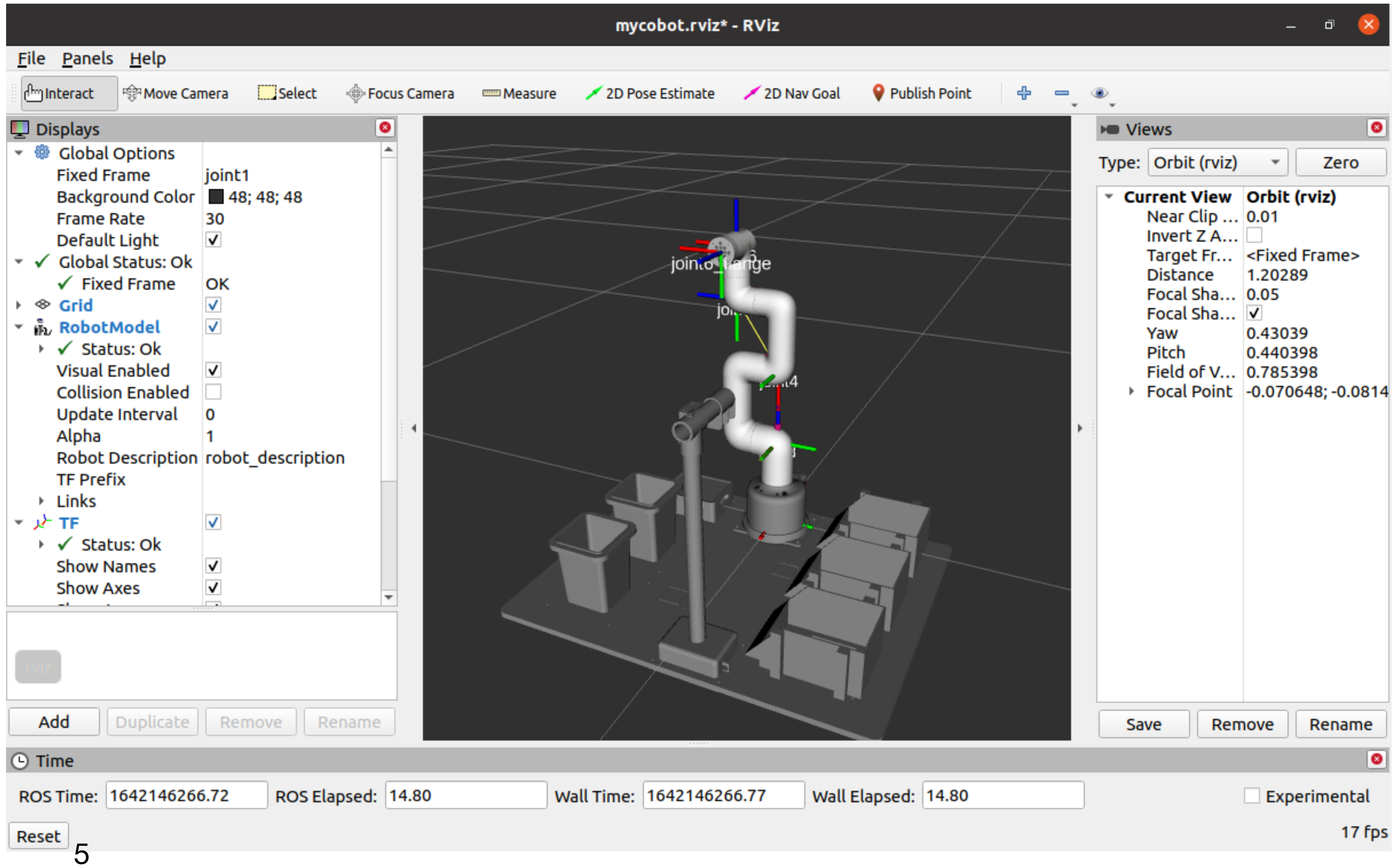
NodeJs

```
const rclnodejs = require('rclnodejs');
```

# What is ROS?

ROS = Libraries + **Tools**

RViz



Global Options
Fixed Frame joint1
Background Color ■ 48; 48; 48
Frame Rate 30
Default Light ☑
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ☑
RobotModel ☑
✓ Status: Ok
Visual Enabled ☑
Collision Enabled □
Update Interval 0
Alpha 1
Robot Description robot_description
TF Prefix
Links
TF ☑
✓ Status: Ok
Show Names ☑
Show Axes ☑

Type: Orbit (rviz) Zero
Current View Orbit (rviz)
Near Clip ... 0.01
Invert Z A... □
Target Fr... <Fixed Frame>
Distance 1.20289
Focal Sha... 0.05
Focal Sha... ☑
Yaw 0.43039
Pitch 0.440398
Field of V... 0.785398
Focal Point -0.070648; -0.0814

ROS Time: 1642146266.72 ROS Elapsed: 14.80 Wall Time: 1642146266.77 Wall Elapsed: 14.80 □ Experimental

17 fps

# What is ROS?

ROS = Libraries + **Tools**

📦 ⚒️

CLI tools

```
ros2 run <package> <node>
ros2 launch <package> <launch file>

ros2 node list
ros2 node info <node>

ros2 topic list
ros2 topic info <topic>
ros2 topic hz <topic>
```

# What is ROS?

- It offers a standard way to develop apps for robotics
- Because of that → hundreds of ROS packages

# Popular Packages

- Nav2

NAV 2

Powered By

OPEN NAVIGATION

# Popular Packages

MoveIt

MoveIt

Capabilities Overview

Brought to you by

PICKNIK

# Why?

# Why?

| | **Before** | **After** |
|---|---|---|
| Hardware Abstraction | ❌ | ✅ |
| Standard Data Types | ❌ | ✅ |
| Code re-use | ❌ | ✅ |

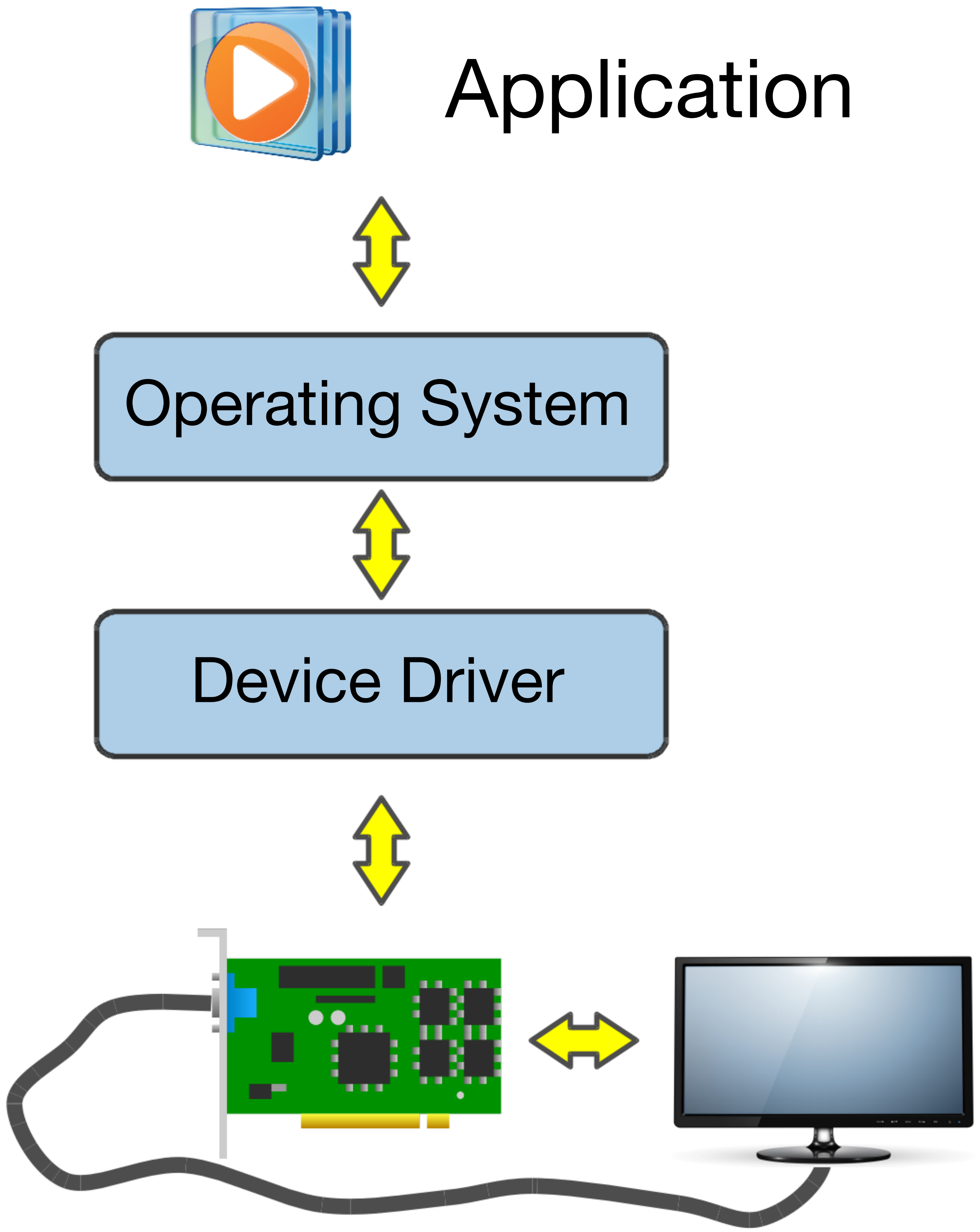
# Hardware Abstraction

Software apps

Work ok

Different hardware

# Hardware Abstraction

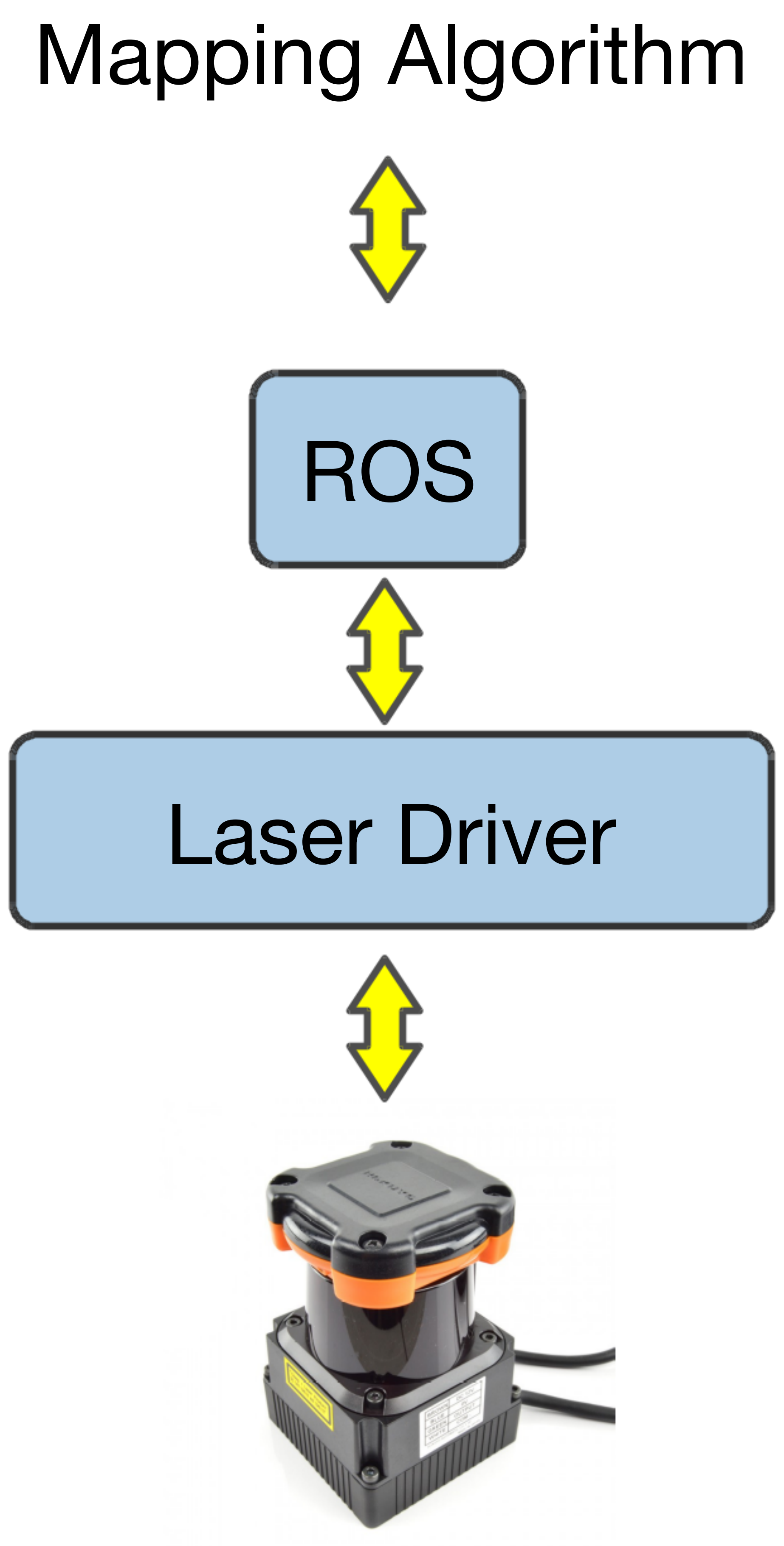


Application

Operating System

Device Driver

# Hardware Abstraction

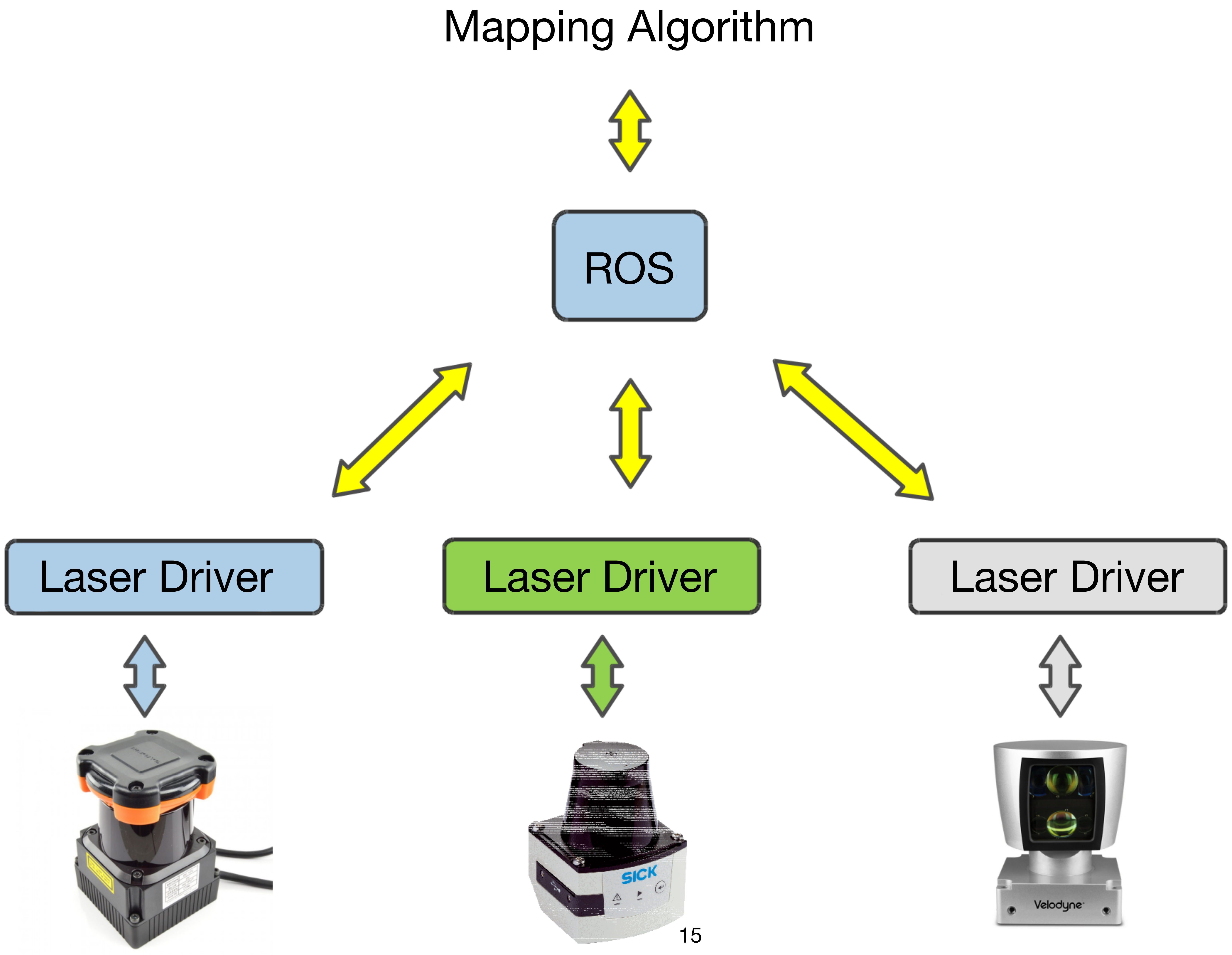


Mapping Algorithm

ROS

Laser Driver

# Hardware Abstraction



Mapping Algorithm

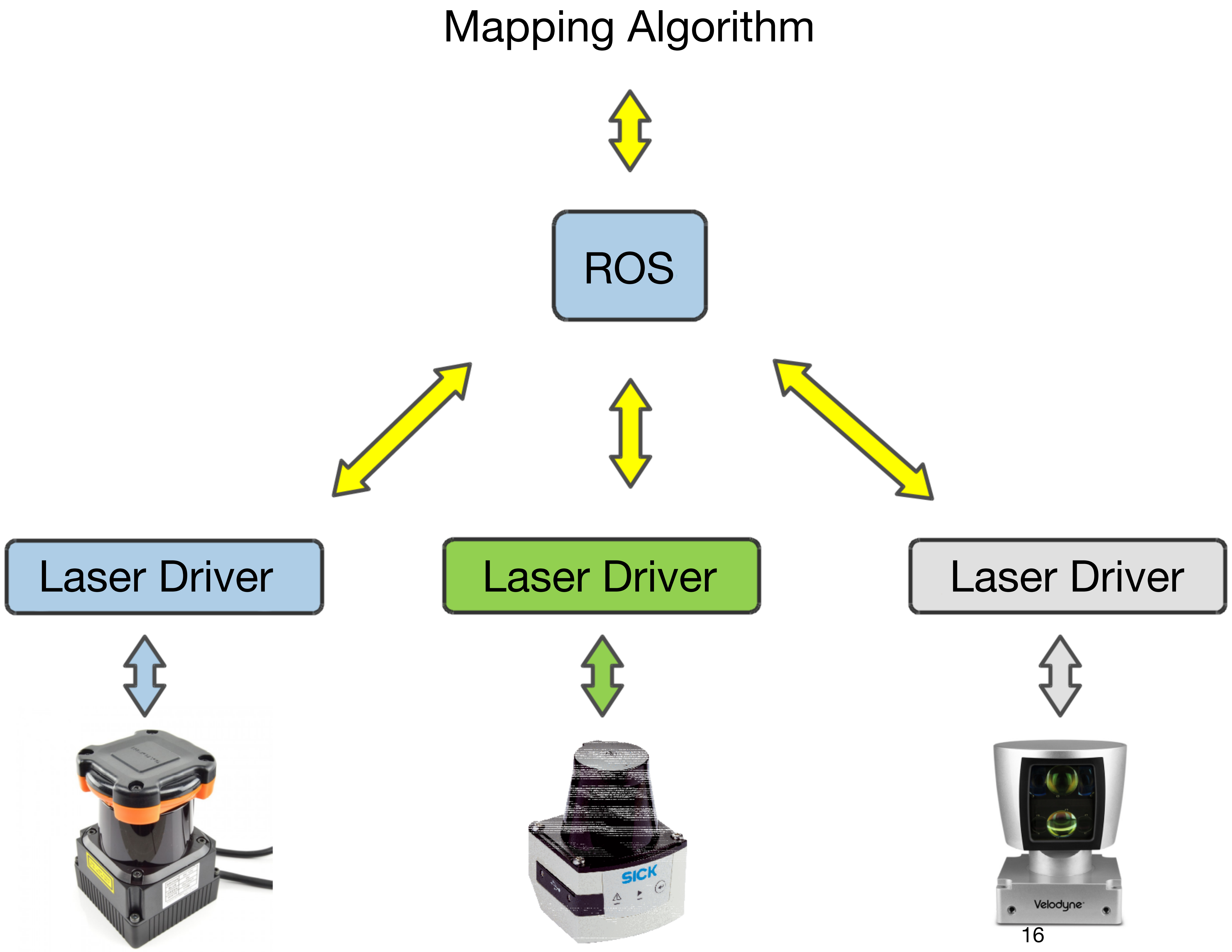
ROS

Laser Driver

Laser Driver

Laser Driver

# Hardware Abstraction



Mapping Algorithm

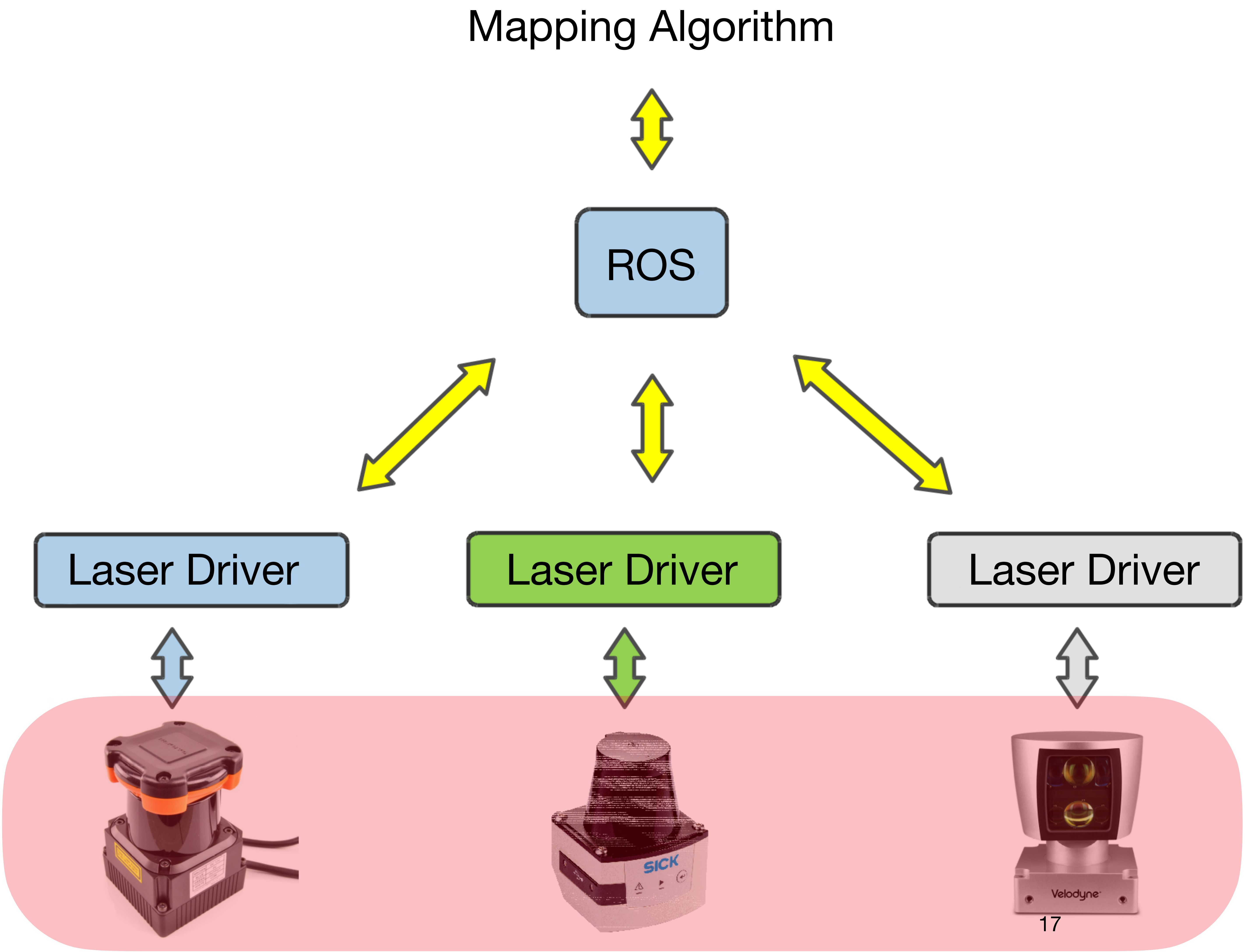
ROS

Laser Driver

Laser Driver

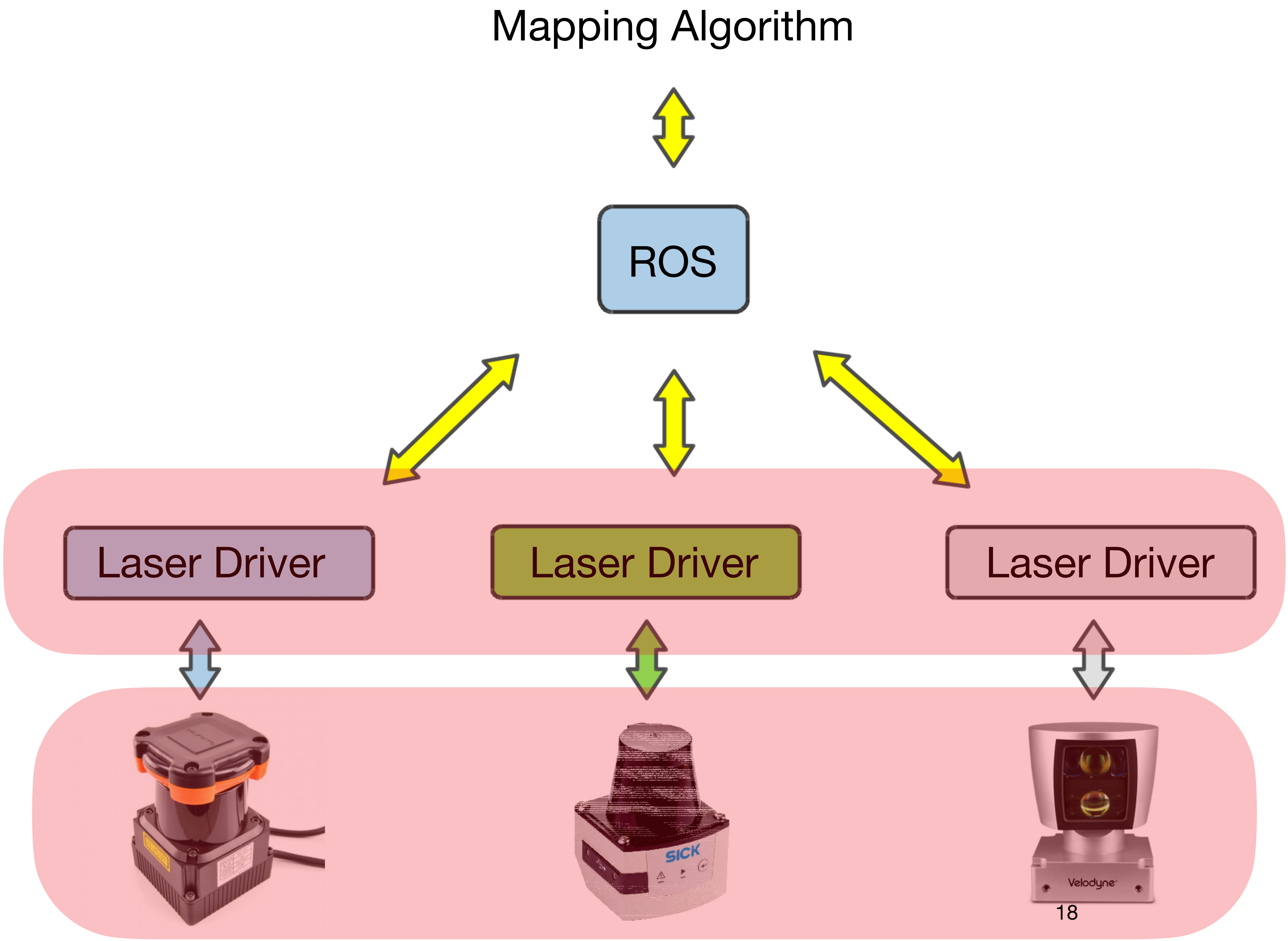
Laser Driver

# Hardware Abstraction



**Different hardware**

# Hardware Abstraction



Mapping Algorithm

ROS

Laser Driver

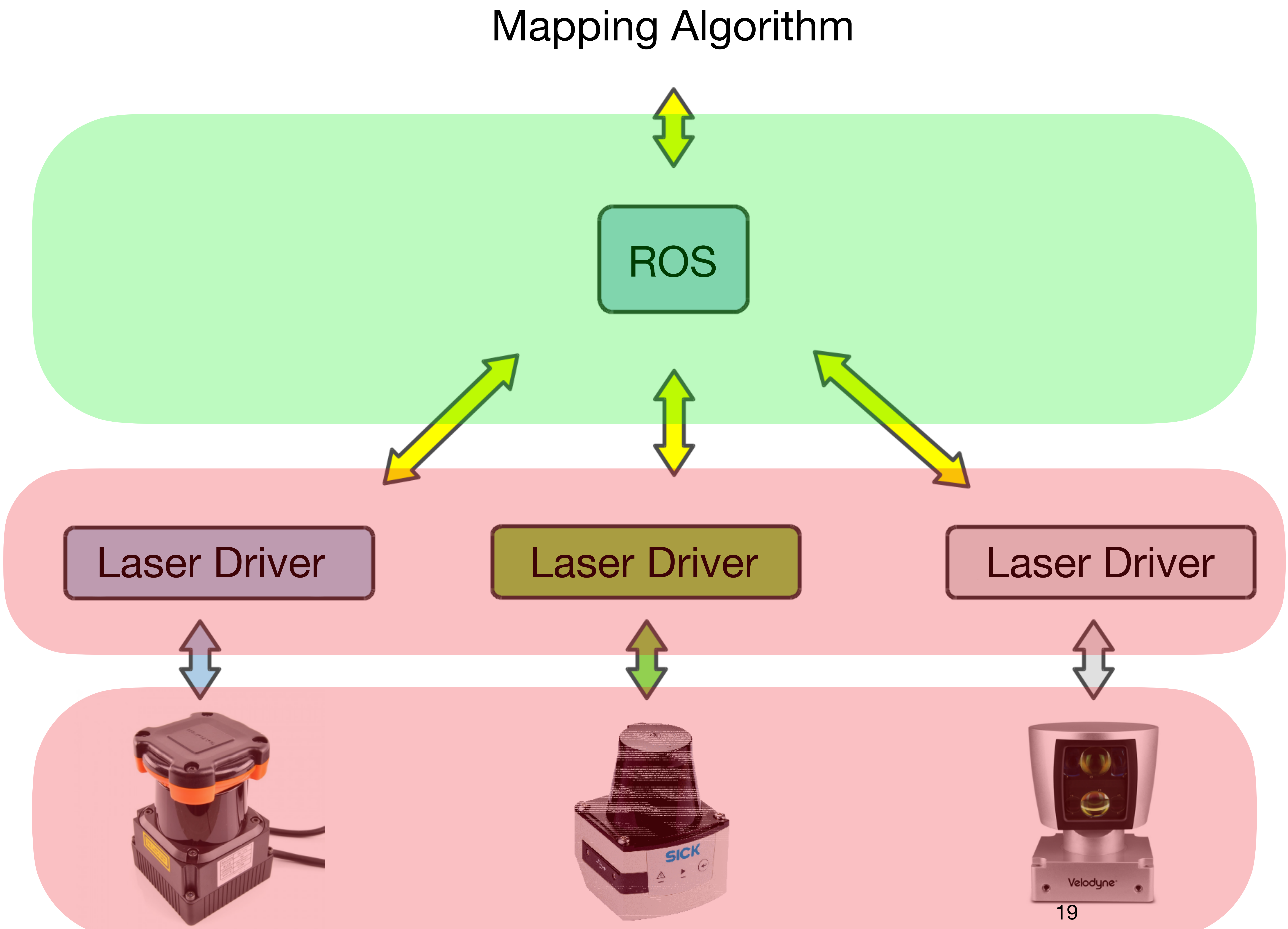
Laser Driver

Laser Driver

**Different Drivers**

**Different hardware**

# Hardware Abstraction

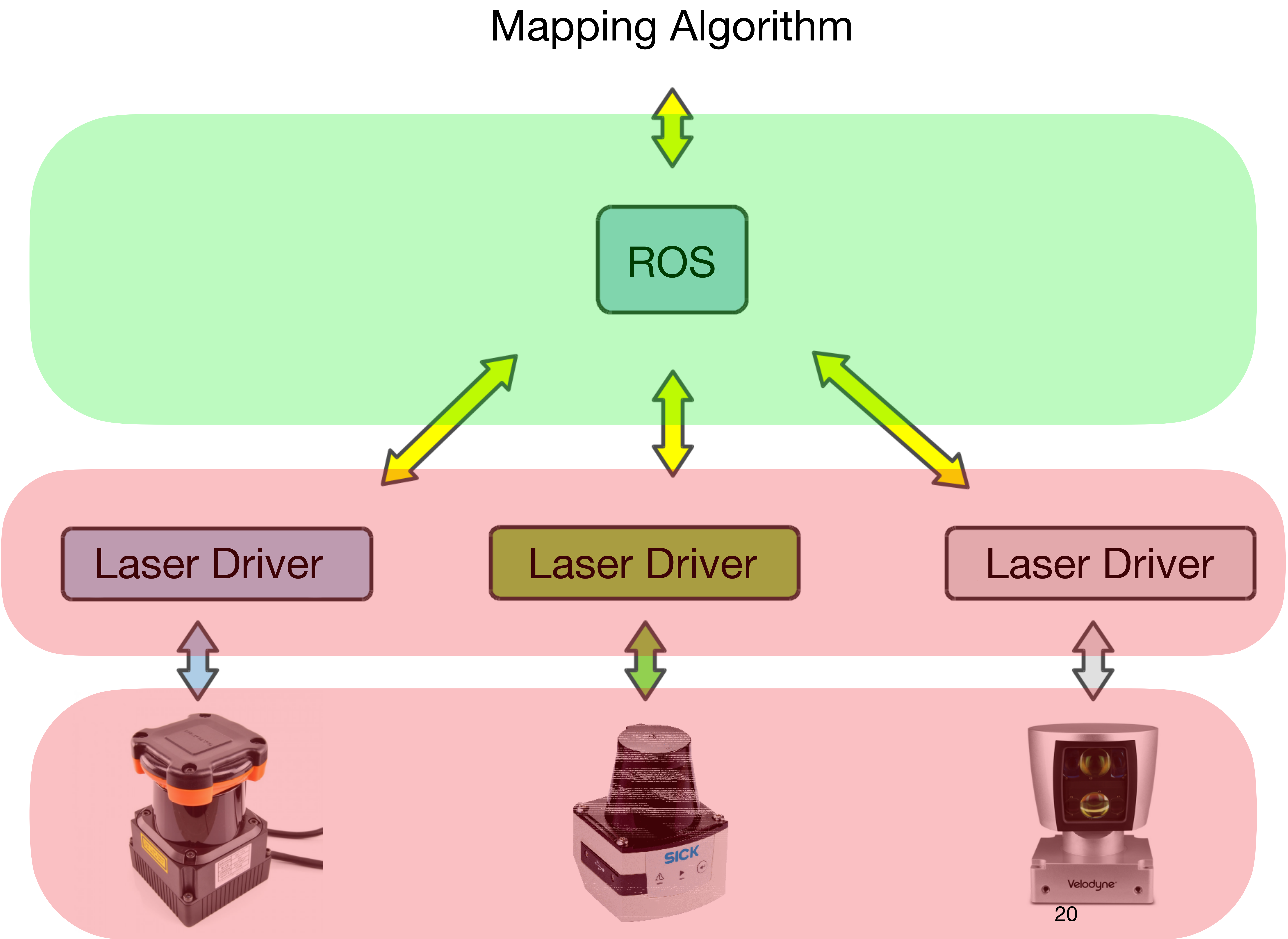


**Data received in the same message format**

**Different Drivers**

**Different hardware**

# Hardware Abstraction



✅ **No change**

**Data received in the same message format**

**Different Drivers**

**Different hardware**

# Hardware Abstraction

Mapping | Navigation | Pick & Place

Robot apps

Work ok

Different hardware

# Communication

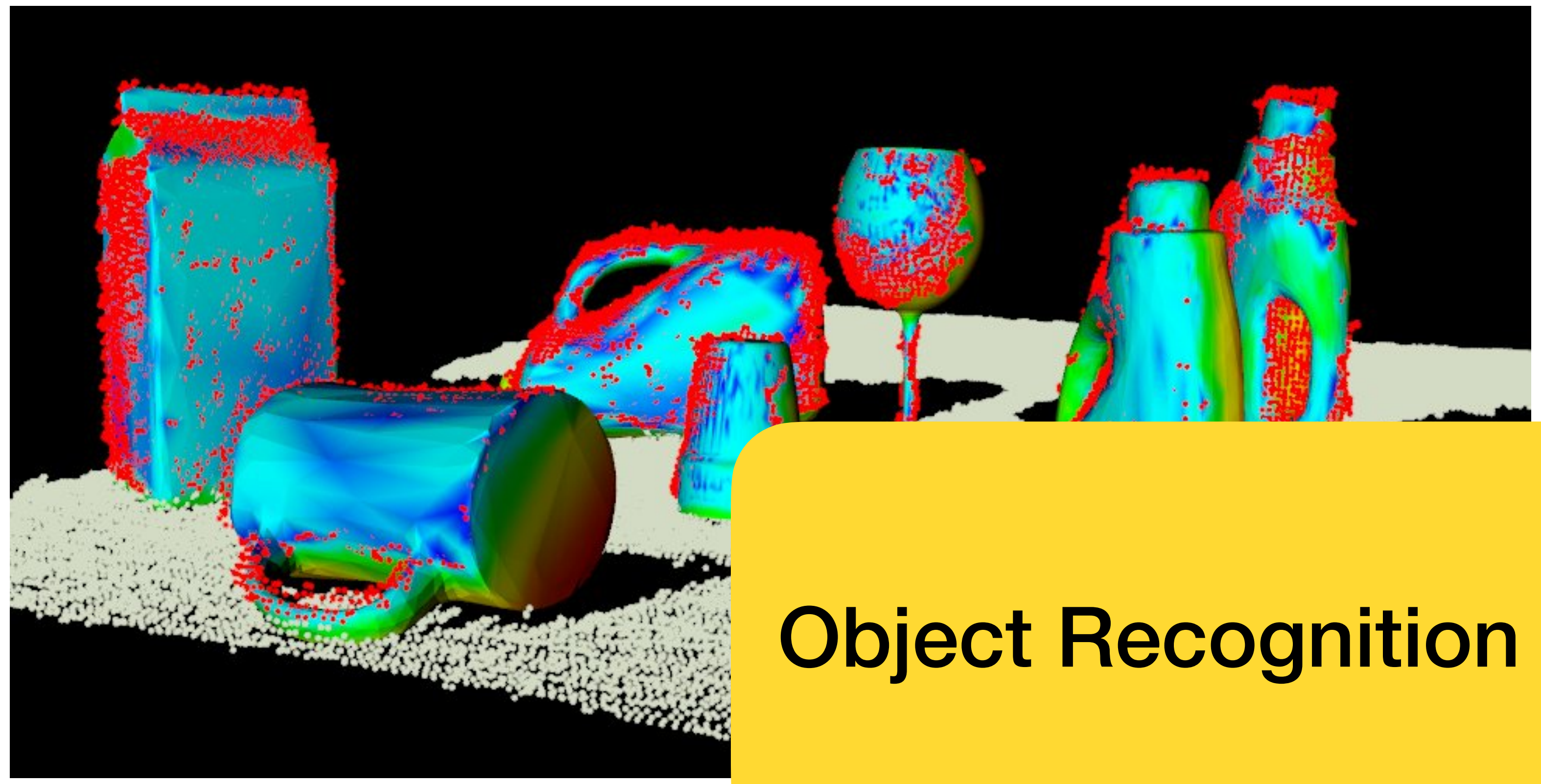
# Communication in ROS

- At it’s core, it offers several ways to **send** data between processes

- Follows multiple communication scheme
  - Publish/subscribe: nodes, topics, messages
  - Service client/server
  - Actions: long running services

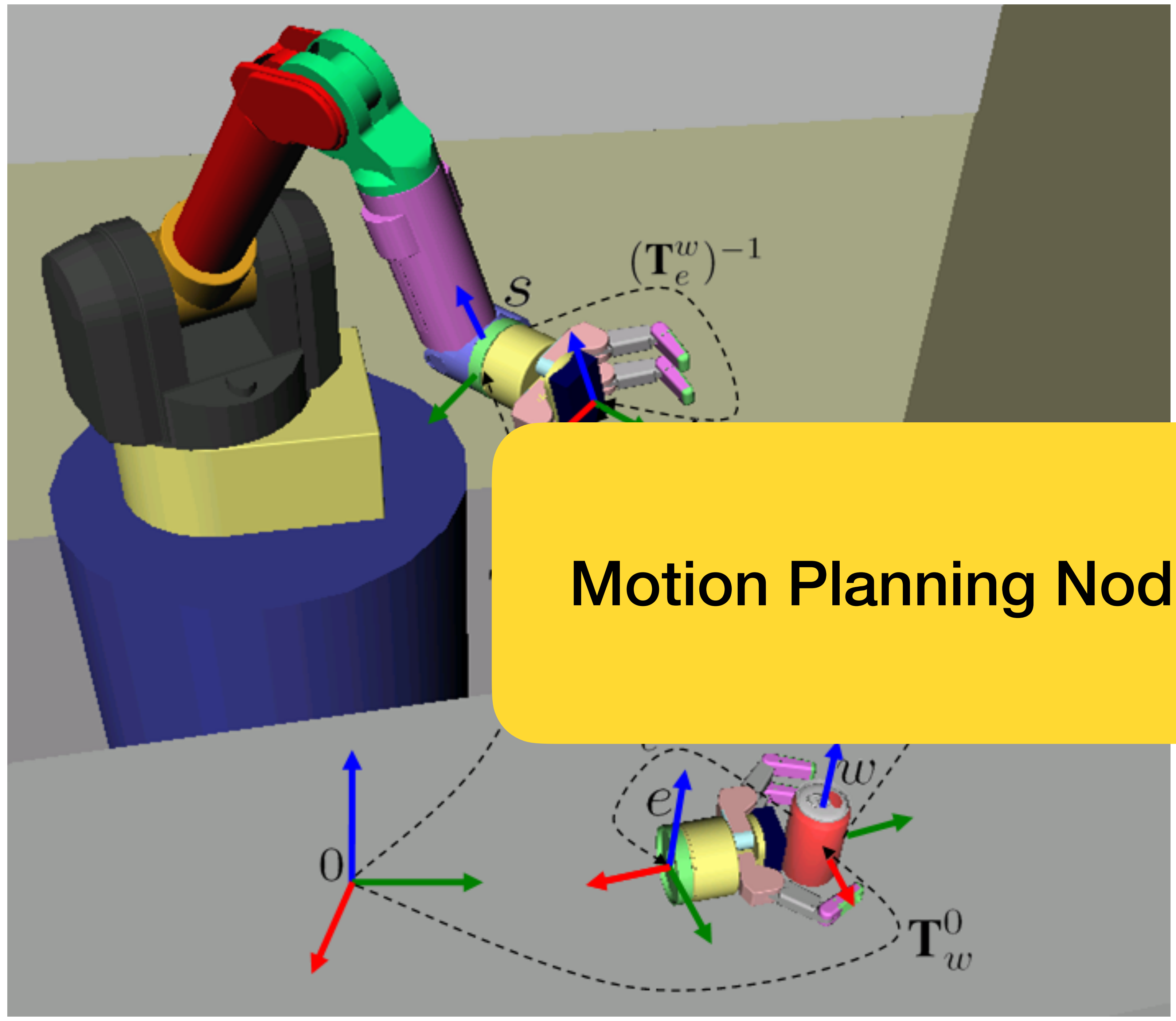
- Build on top of DDS (Data Distribution Service)

Camera Driver Node



Object Recognition Node

Robot Driver Node



Motion Planning Node

# Demo