Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Introduction to ROS

Foundation Course

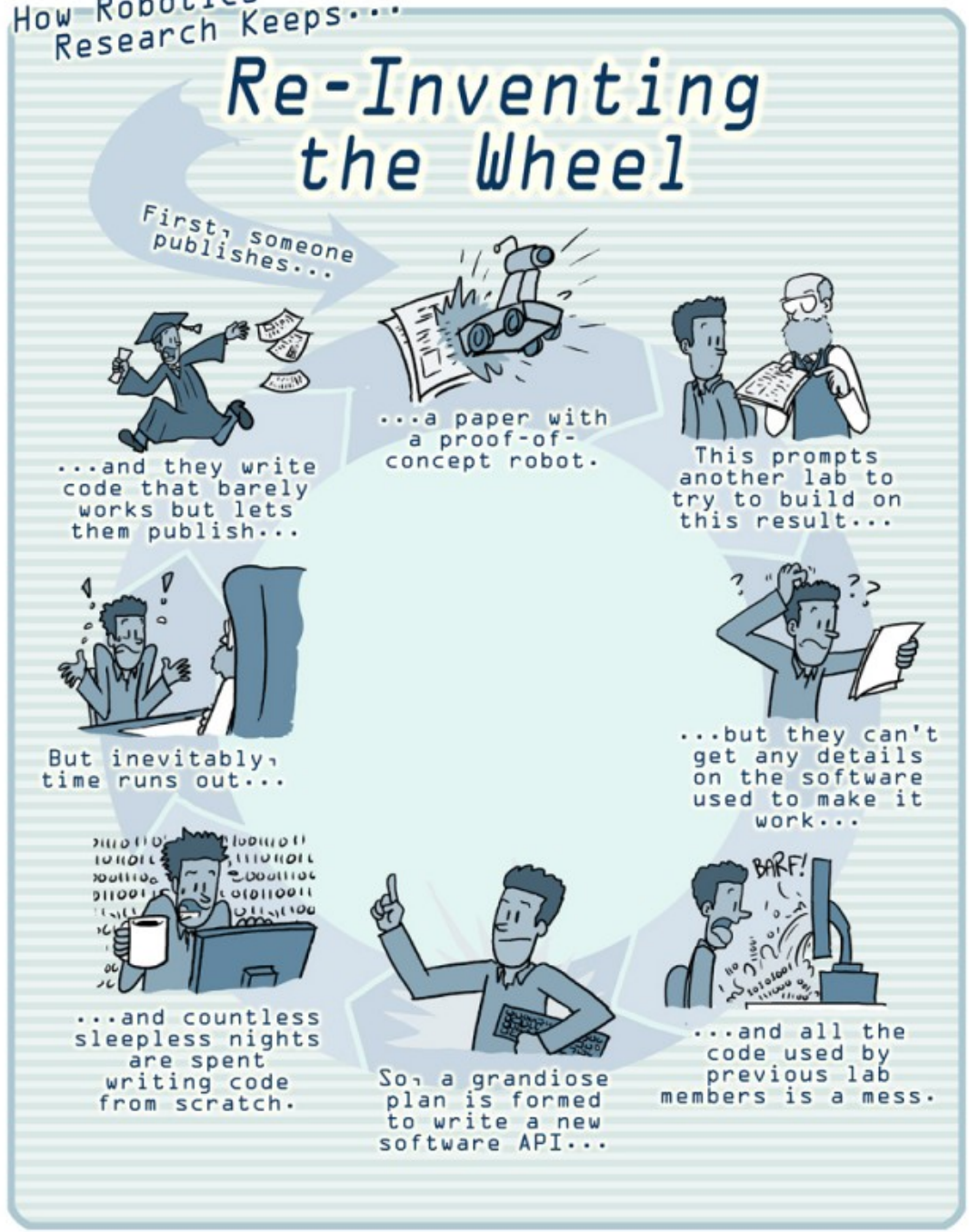
August 20, 2019

Hassan Umari

What ROS is

Robot Operating System

- Short for: Robot Operating System.
- A collection of libraries and tools.
- It helps software developers create robot applications.



http://www.willowgarage.com/blog/2010/04/27/reinventing-wheel

What ROS is

Robot Operating System

- A way to standardize writing software for robots.
- It enhances code reusability  .
- ROS is open-source  .
- It is a meta-operating system.
- ROS can be installed on Ubuntu and Debian (so it's currently supported on Linux only).

What ROS is NOT

Robot Operating System

- It is NOT a programming language.
- It is NOT an integrated development environment (IDE).
- It is NOT a stand-alone operating system

1. What is ROS?
1.1 What ROS is
1.2 What ROS is NOT

2. Analogy Between ROS and Operating Systems

3. Features of ROS
3.1 Language independent
3.2 Distributed and Modular
3.3 A lot of libraries and tools
3.4 Bad Thing About ROS

4. ROS Concepts

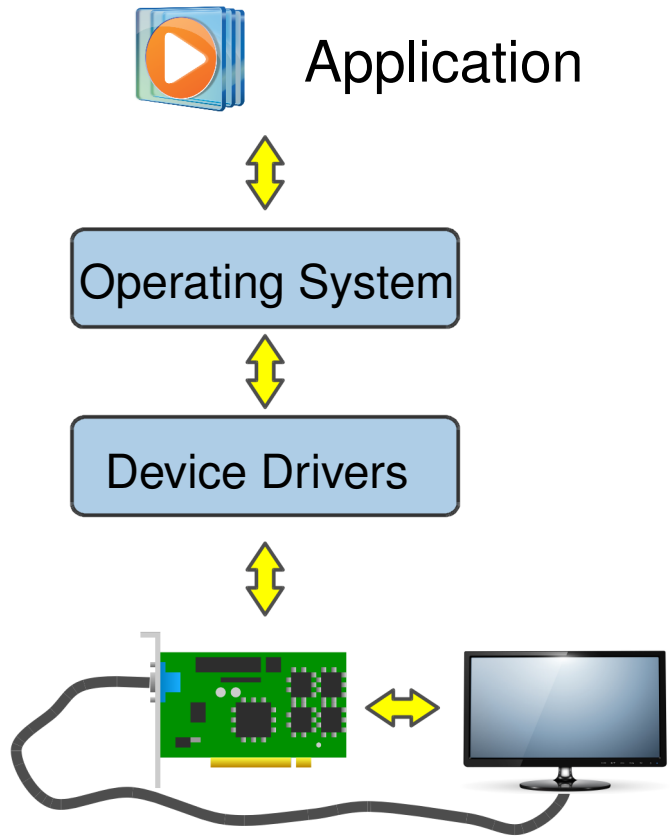
Analogy Between ROS and Operating Systems

Software Applications

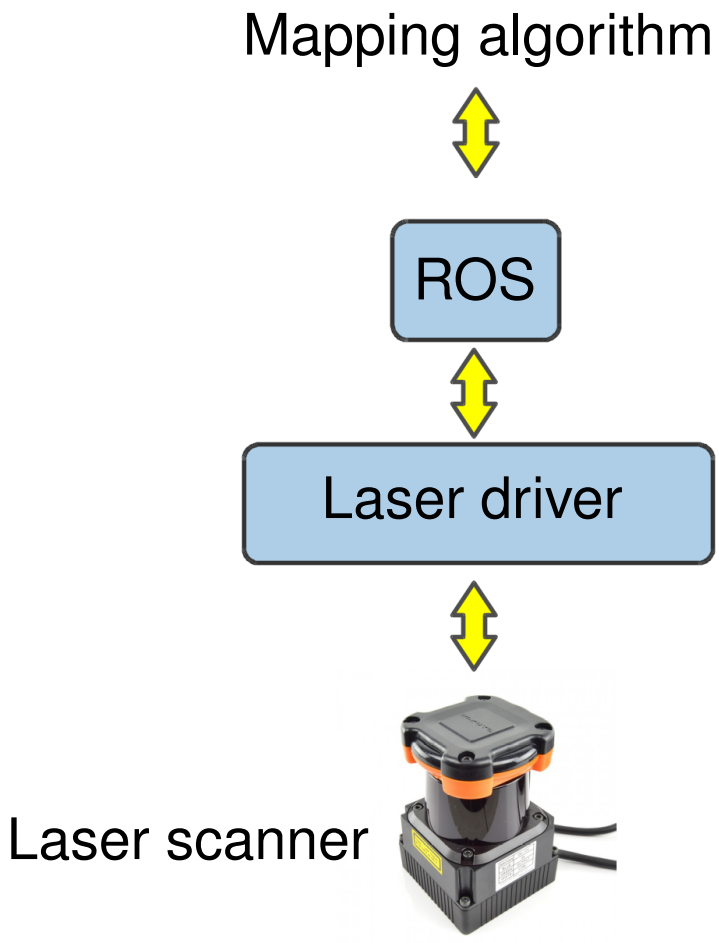
work on

Different hardware

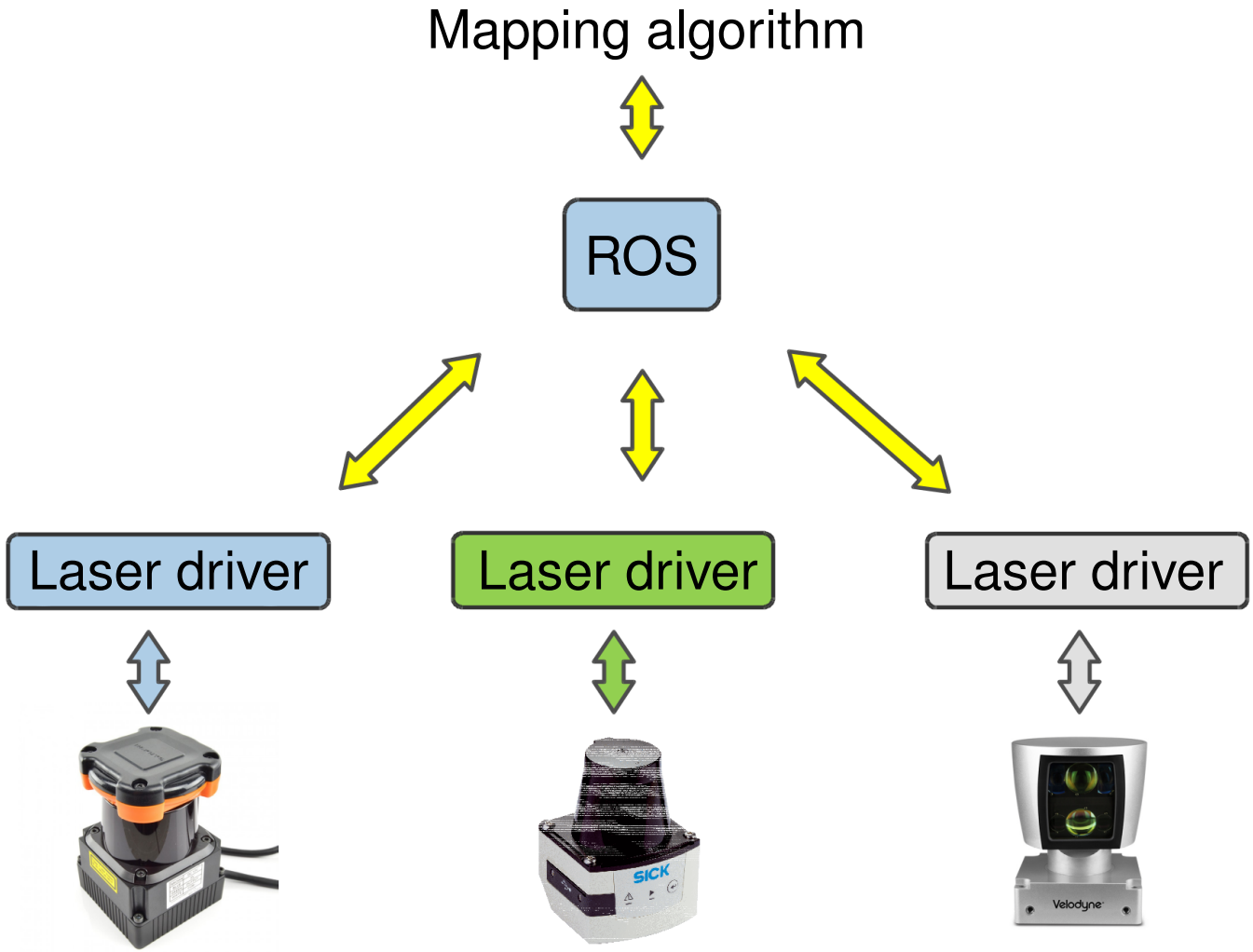
Analogy Between ROS and Operating Systems



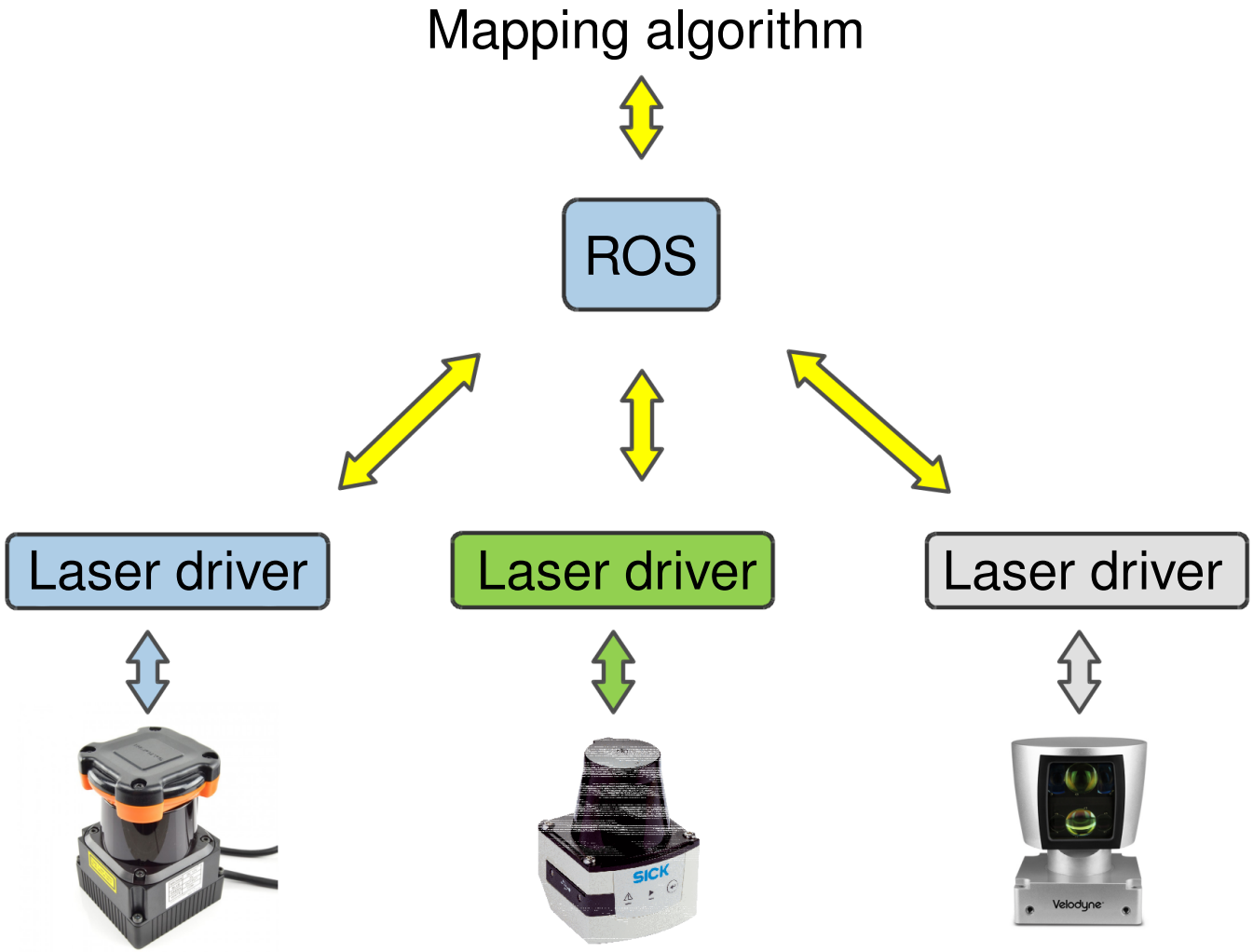
Analogy Between ROS and Operating Systems



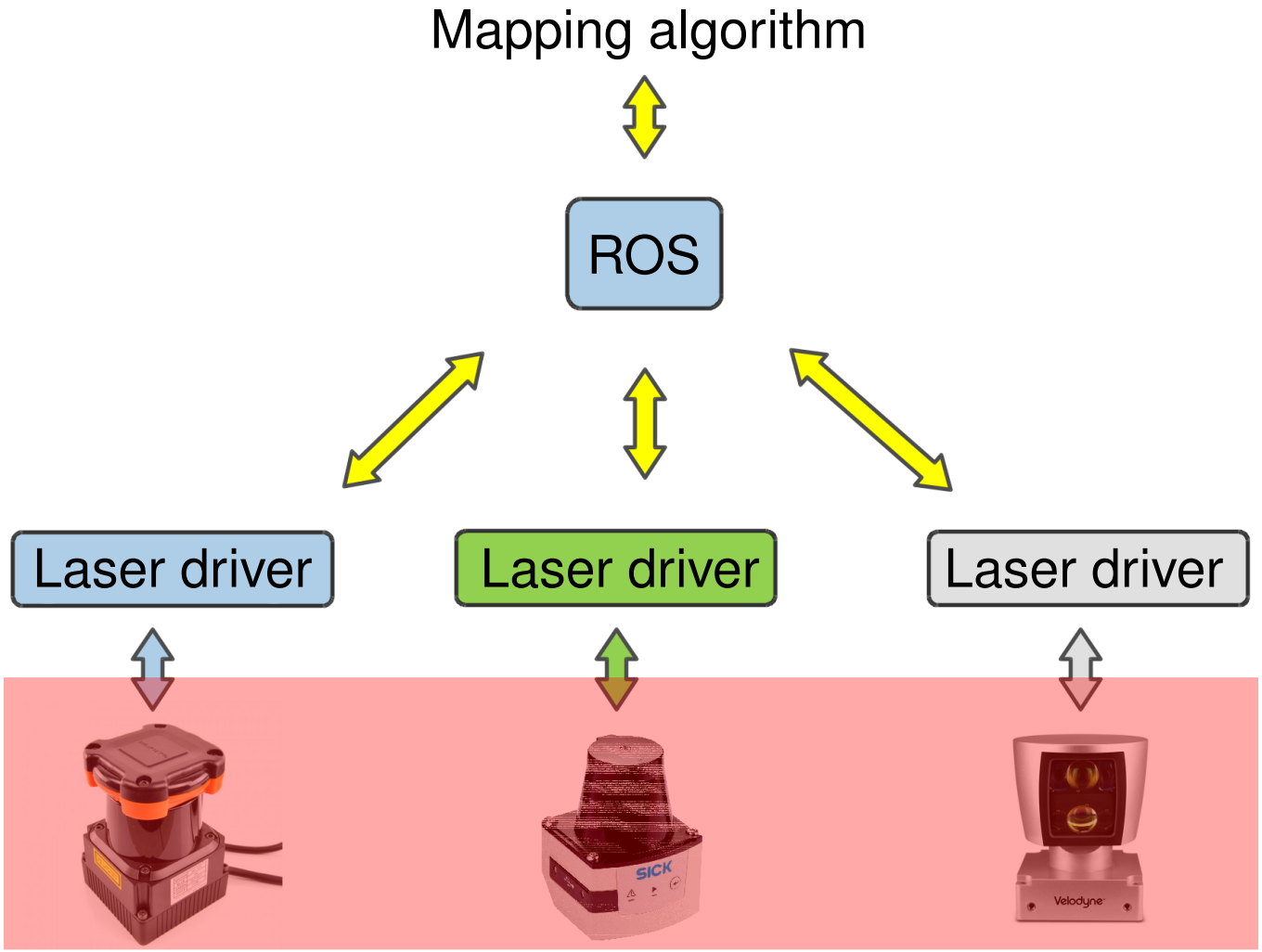
Analogy Between ROS and Operating Systems



Analogy Between ROS and Operating Systems

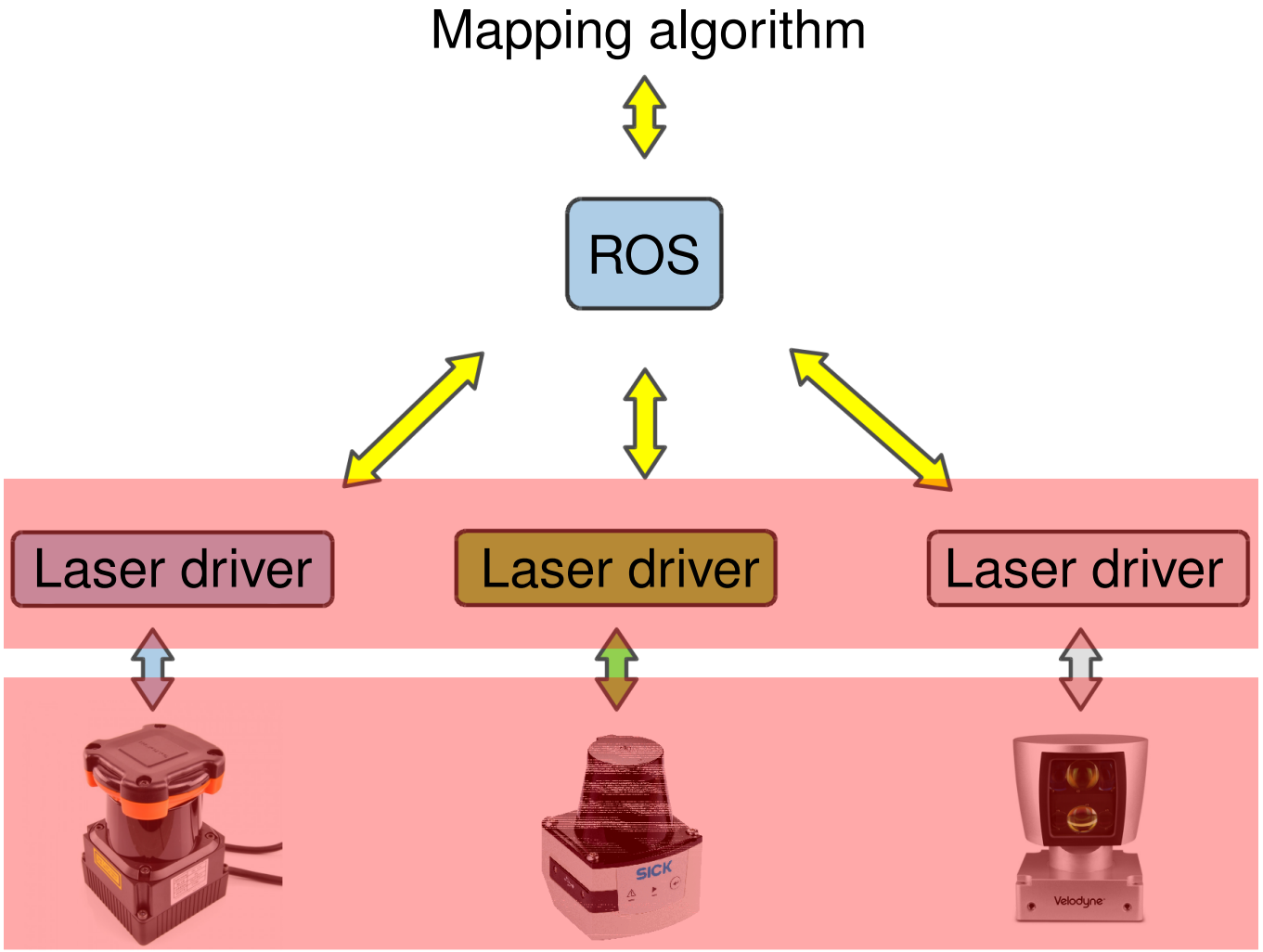


Analogy Between ROS and Operating Systems

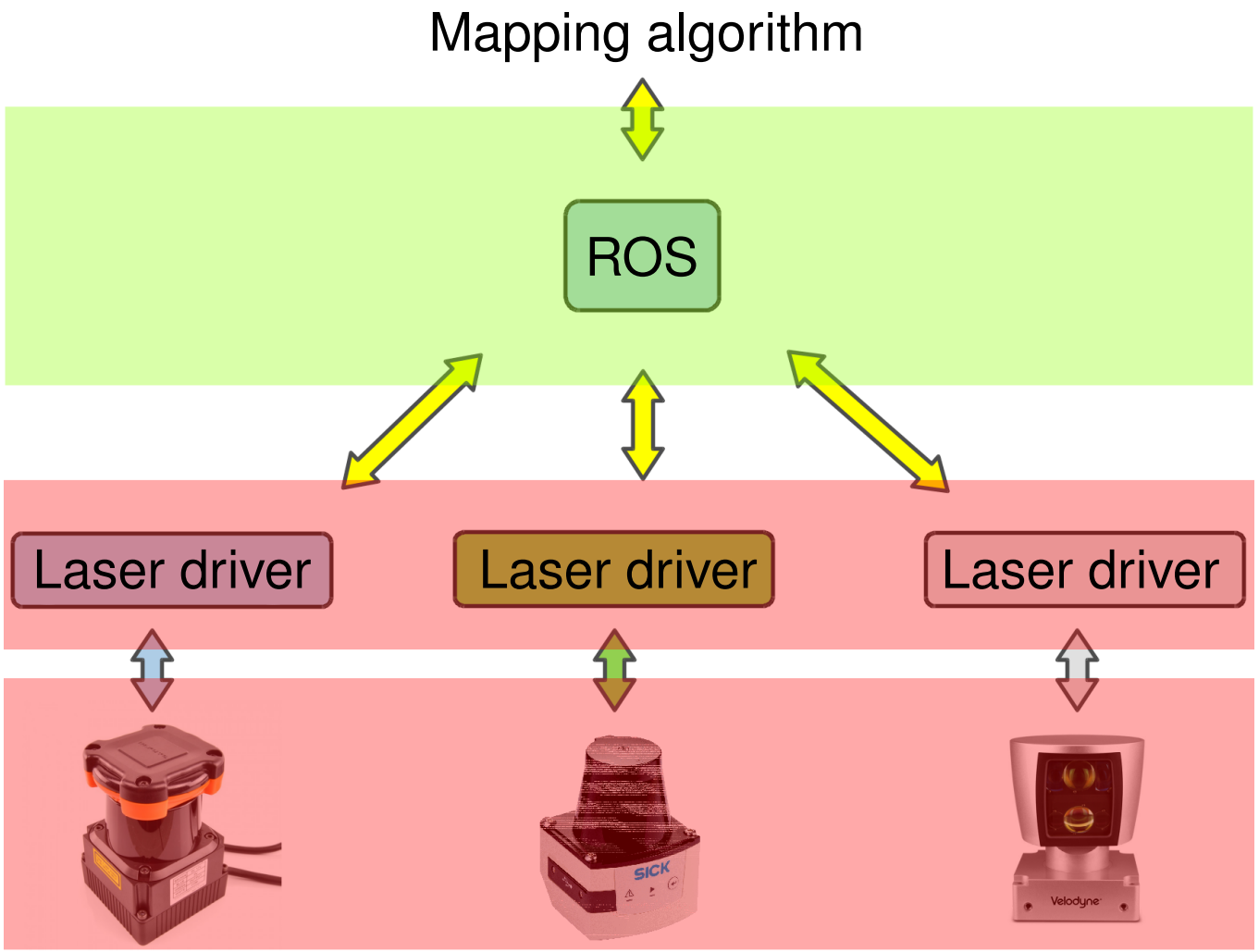


Different hardware

Analogy Between ROS and Operating Systems



Analogy Between ROS and Operating Systems

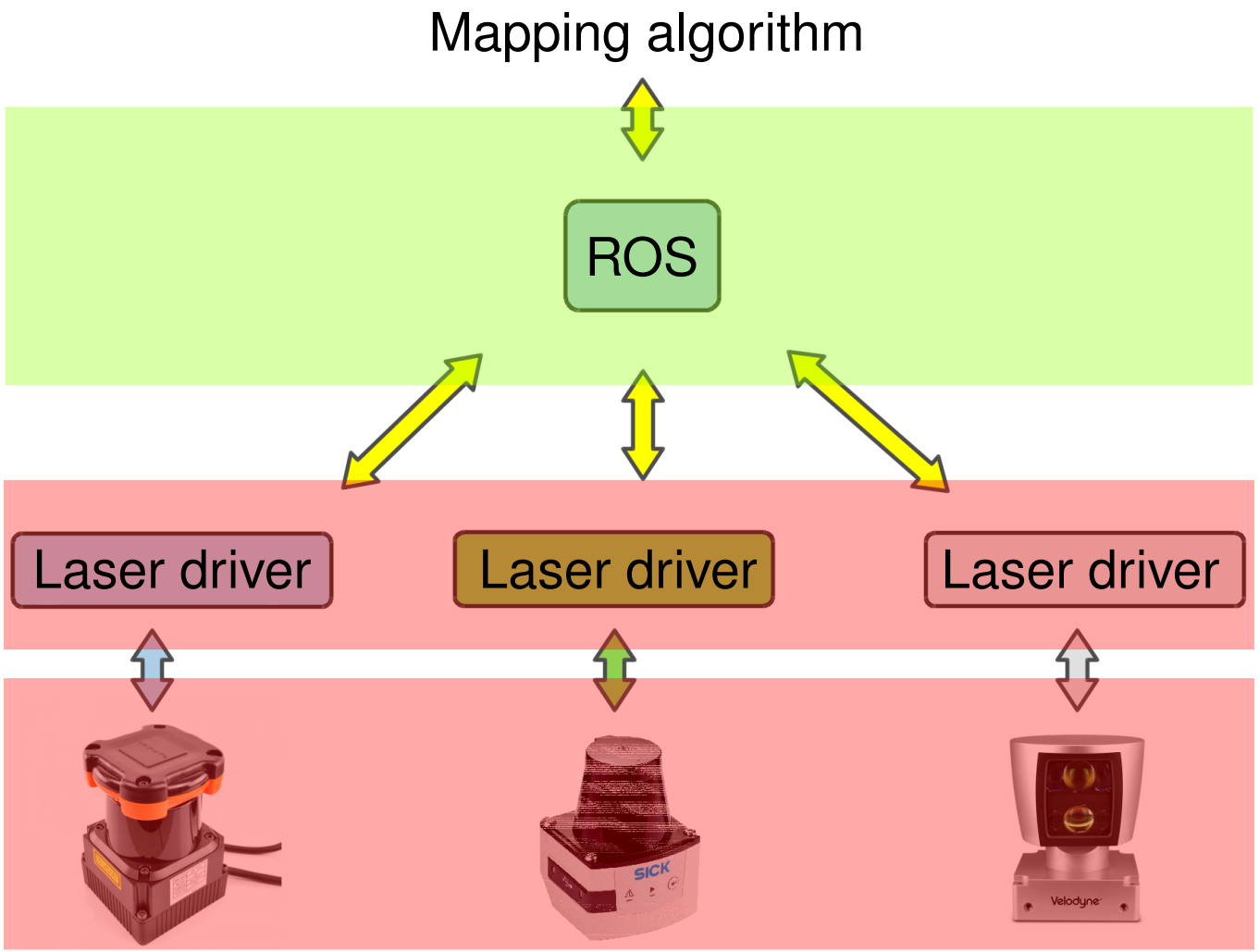


Data received in the same message format

Different drivers

Different hardware

Analogy Between ROS and Operating Systems



Analogy Between ROS and Operating Systems

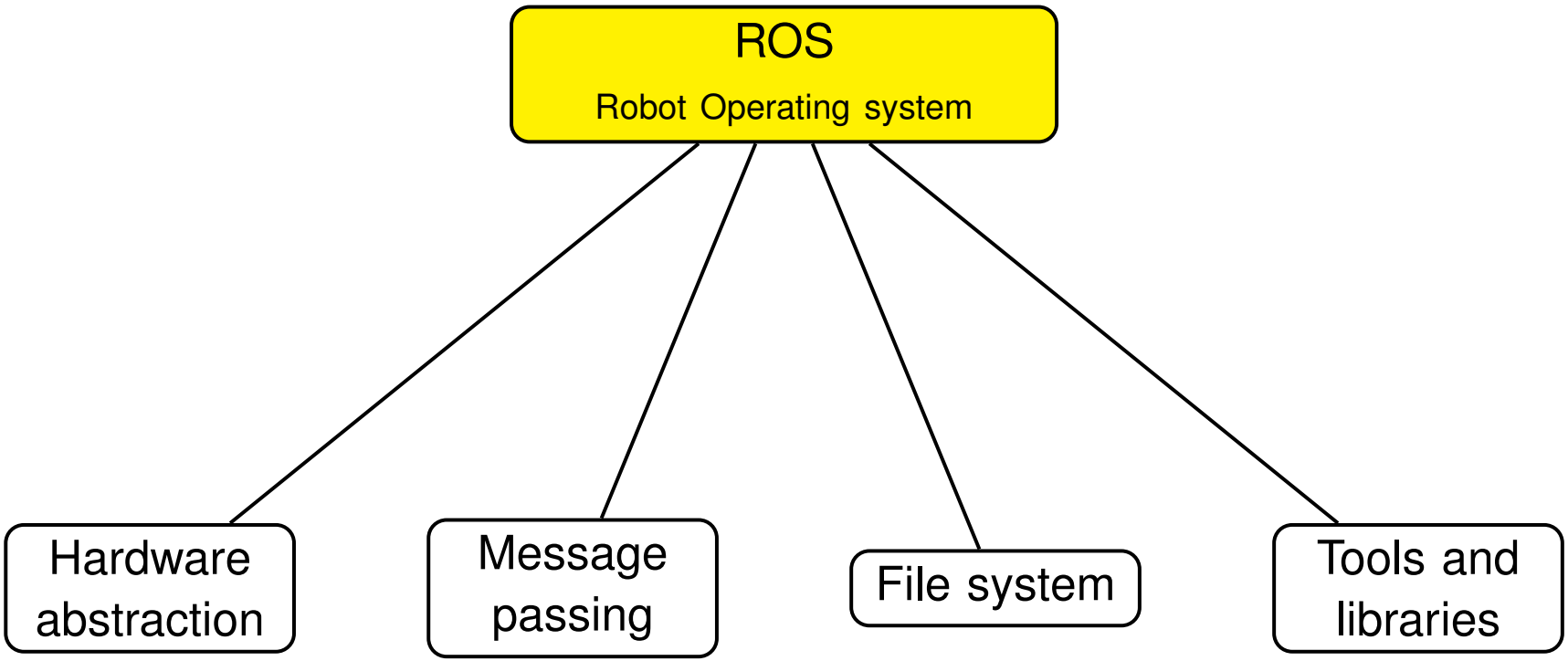
Mapping | Navigation | pick & place

Robot Applications

work on

Different hardware

Analogy Between ROS and Operating Systems



Features of ROS

- Language independent.
- Distributed and Modular.
- A lot of libraries and tools.
- Open Source.
- Active Community.

Features of ROS

Language independent

- ROS functionalities are implemented as a library in different programming languages.
- These libraries are referred to as ROS client libraries.

Language independent

Features of ROS

ROS client libraries.

- Main ROS Client libraries:
  - roscpp
  - rospy
  - roslisp
- Experimental ROS client libraries:
  - rosjava
  - rosruby
  - and some others..
- ROS support on MATLAB:
  - Robotics System Toolbox


Distributed and Modular

Features of ROS

- ROS supports running processes on multiple computers connected together through a LAN.
- In a system running ROS, there will be multiple of processes where each process can do certain task. A process can be changed without altering the remaining processes.

A lot of libraries and tools

Features of ROS

- Examples of libraries:
  - Navigation stack.
  - SLAM (gmapping, hector SLAM, etc..).
  - Localization (amcl, etc..).
  - Motion planning for manipulators (MoveIt)
  - Support for popular libraries (OpenCV, PCL).
- Examples of tools:
  - RVIZ:3D Visualization.
  - ROS bag files: Logging Sensor Data.
  - Catkin: A Build System.
  - Command line tools.

Bad Thing About ROS

- Learning ROS needs time.
- It needs a computer. Does not work on a microcontroller!
- Not optimized for multiple robots.
- Supported only on Linux, no support for Windows or macOS.