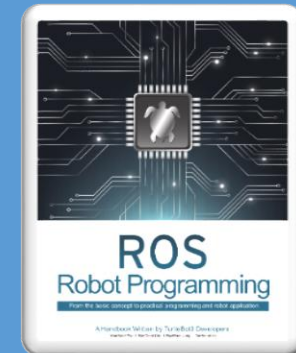


Robot Operating System ROS

ROBOTIS

KAIST



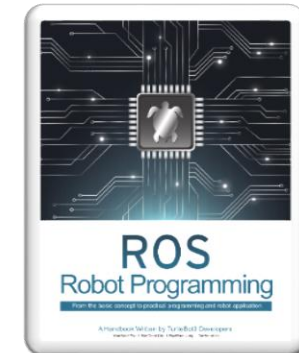
You Tube

Subscribe

Textbook
P. 9~21

Contents

- I. About ROS
- II. Meta operating system
- III. Purpose of ROS
- IV. Configuration of ROS
- V. ROS ecosystem
- VI. History of ROS
- VII. Version of ROS



You Tube

 **Subscribe**

Textbook
P. 9~21

Introduction to ROS

ROS?

ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

<http://www.ros.org/wiki/>



Software framework



- Software framework for developing robot software
 - It is possible to jointly develop complex programs by finely dividing them with message exchanging method between nodes.
 - Supports command tool, visualization tool Rviz, GUI toolbar rqt, 3D simulator Gazebo
 - Supports modeling, sensing, recognition, navigation, and manipulation functions commonly used in robotics
 - Create Robotics Ecosystem!

True purpose of ROS

Building an **ecosystem** that enables
robotics software development to be
collaborated on a global level!



Is ROS a new operating system (OS)?

- **Operating System**

- **General purpose computer**

- Windows(Windows XP, 7, 8 ...)
 - Linux(Ubuntu, Redhat, Fedora, Mint, Gentoo ...)
 - MAC(OS X ...) etc

- **Smart phone**

- Android, iOS, Windows Phone, Symbian, RiMO, Tizen etc

- **ROS** = Robot Operating System

- **ROS** is Meta-Operating System

Meta-Operating System

- Meta-Operating System (Meta-Operating System): It is not a precisely defined term but it can be regarded as a system that performs scheduling, loading, monitoring, error handling and utilizing distributed computing resources as a virtualization layer between applications and distributed computing resources.
- It is not a traditional operating system like Windows, Linux, and Android. Rather, ROS uses the traditional operating system (Linux, Windows, OS-X, and Android).
- It uses the existing operating system's process management system, file system, user interface, program utilities (compiler, thread model, etc.). In addition, it provides essential functions for developing robot application software such as data transmission / reception, scheduling and error handling among many different types of hardware in a library form.
- In addition, it develops, manages and provides various application programs based on the robot software framework, and has an ecosystem that distributes packages developed by users.

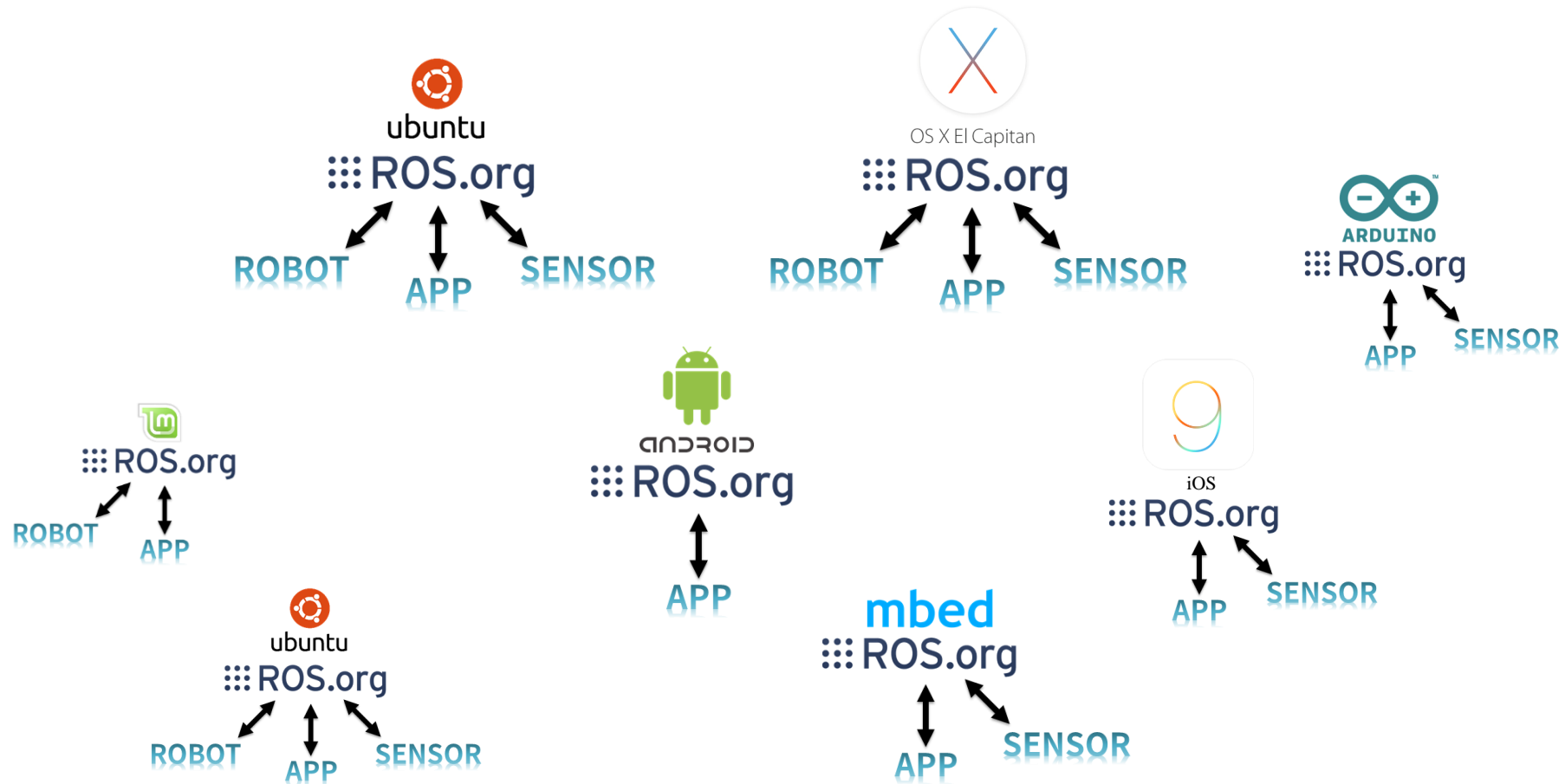
Meta-Operating System

- **Meta-Operating System (Meta-Operating System)**: It is not a precisely defined term but it can be regarded as a system that performs scheduling, loading, monitoring, error handling and utilizing distributed computing resources as a virtualization layer between applications and distributed computing resources.
- It is **not a traditional operating system** like Windows, Linux, and Android. Rather, **ROS uses the traditional operating system** (Linux, Windows, OS-X, and Android).
- It uses the existing operating system's process management system, file system, user interface, program utilities (compiler, thread model, etc.). In addition, **it provides essential functions for developing robot application** software such as data transmission / reception, scheduling and error handling among many different types of hardware in a library form.
- In addition, it develops, manages and provides various application programs based on the **robot software framework**, and has an **ecosystem** that distributes packages developed by users.

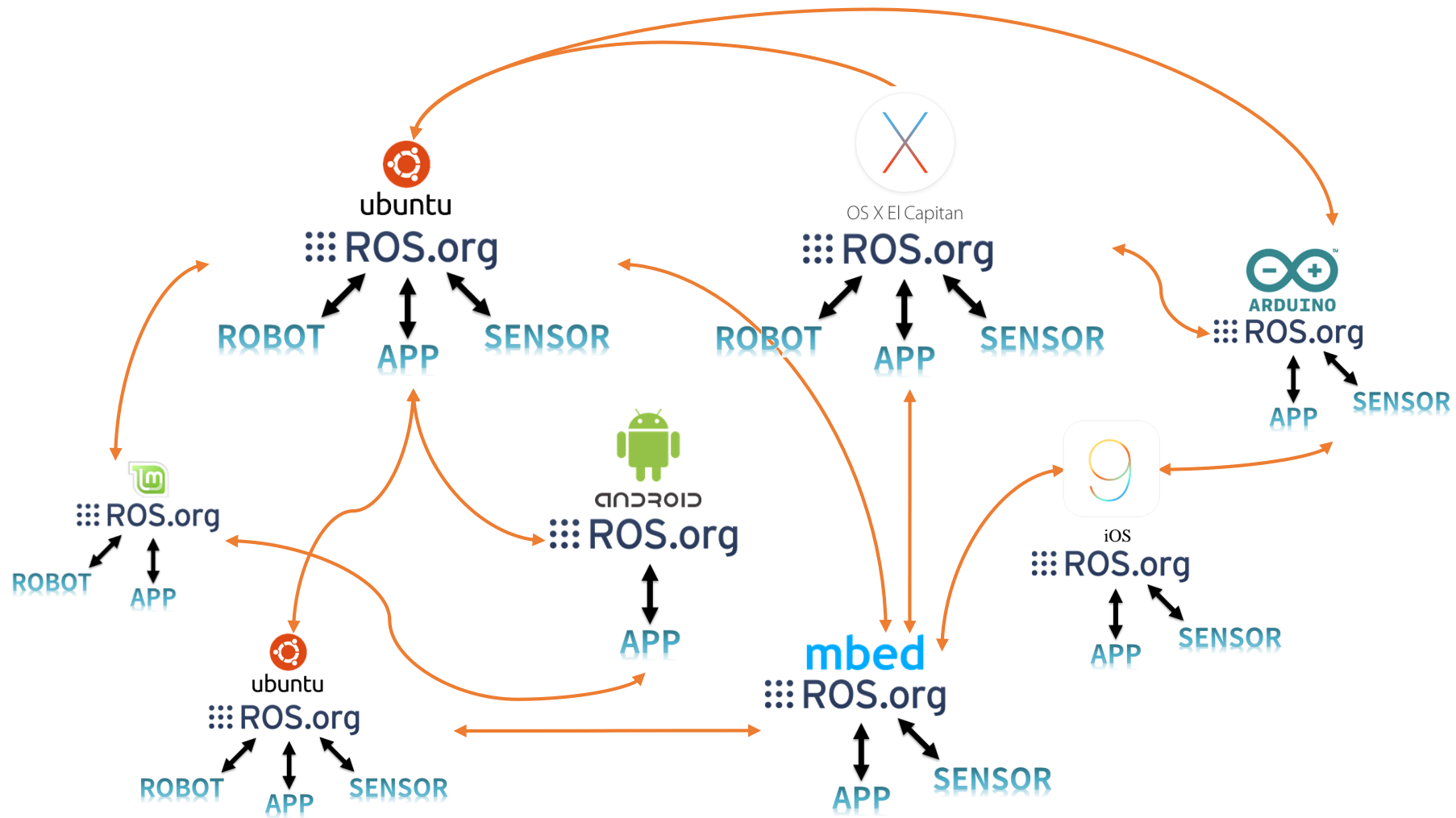
Meta-Operating System



Support communication between different devices



Support communication between different devices



ROS-enabled operating system

▪ Traditional operating system

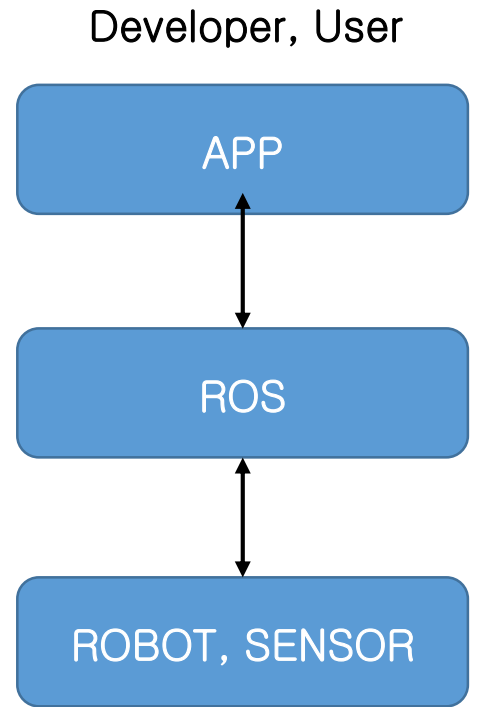
- Ubuntu, OS X, Windows, Fedora, Gentoo, OpenSUSE, Debian, Raspbian, Arch, and QNX Realtime OS. (There can be functional limitations in some OS)
- Partially available for Android and iOS, smartphone operating systems
- In case of microcontroller unit (MCU) which can not be equipped with OS, it provides a library to communicate via serial communication, Bluetooth, and LAN
- Basically it is recommended to run on **Ubuntu**, OS X!

ROS 2.0 supports
three major
operating systems

ROS Configuration

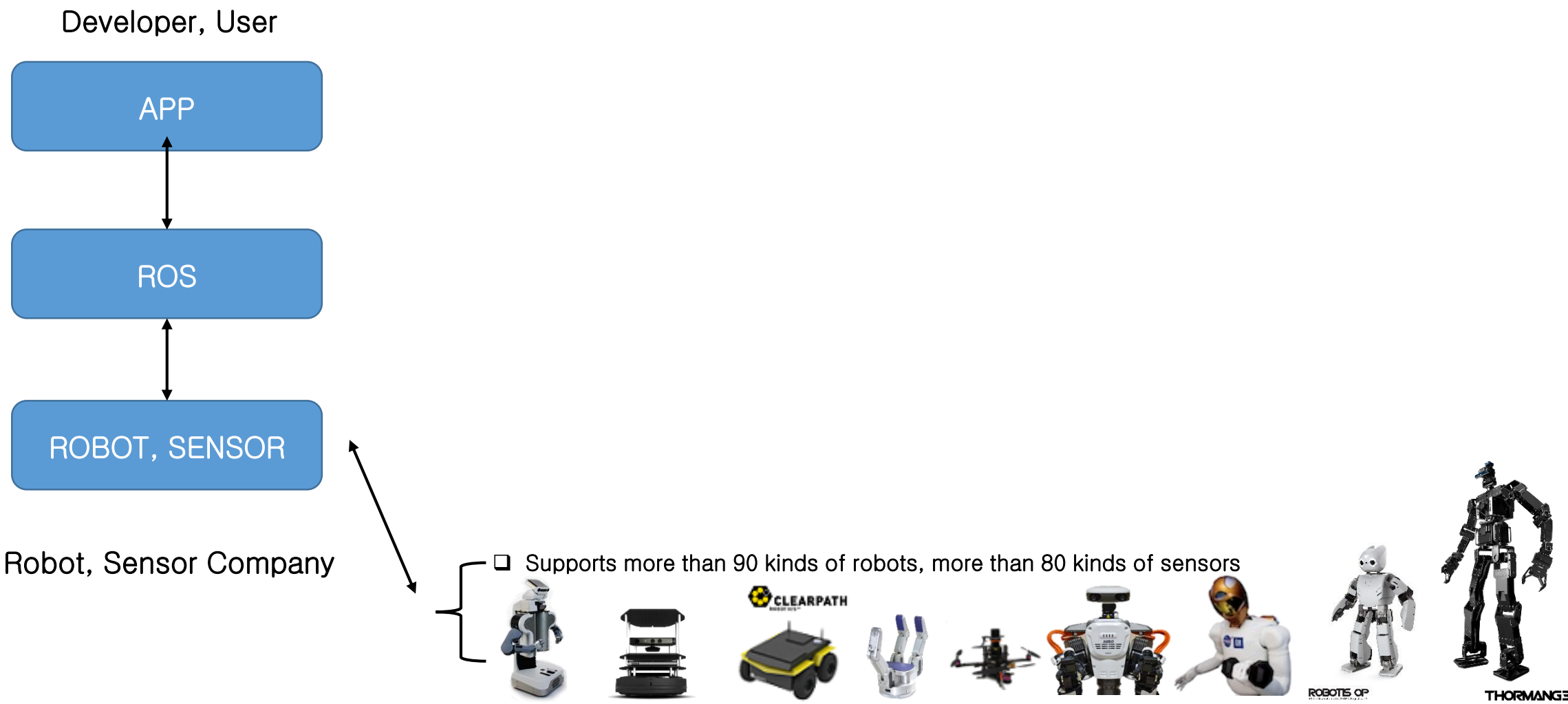
Client Layer	roscpp	rospy	roslisp	rosjava	roslibjs		
Robotics Application	MoveIt!	navigatioin	executive smack	descartes	rospieex		
	teleop pkgs	rocon	mapviz	people	ar track		
Robotics Application Framework	dynamic reconfigure	robot localization	robot pose ekf	Industrial core	robot web tools	ros realtime	mavros
	tf	robot state publisher	robot model	ros control	calibration	octomap mapping	
	vision opencv	image pipeline	laser pipeline	perception pcl	laser filters	ecto	
Communication Layer	common msgs	rosbag	actionlib	pluginlib	rostopic	rosservice	
	roscnode	roslaunch	roscparam	roscmaster	rosout	ros console	
Hardware Interface Layer	camera drivers	GPS/IMU drivers	joystick drivers	range finder drivers	3d sensor drivers	diagnostics	
	audio common	force/torque sensor drivers	power supply drivers	roscserial	ethernet drivers	ros canopen	
Software Development Tools	RViz	rqt	wstool	roscpack	catkin	rosdep	
Simulation	gazebo ros pkgs	stage ros					

Current ROS ecosystem

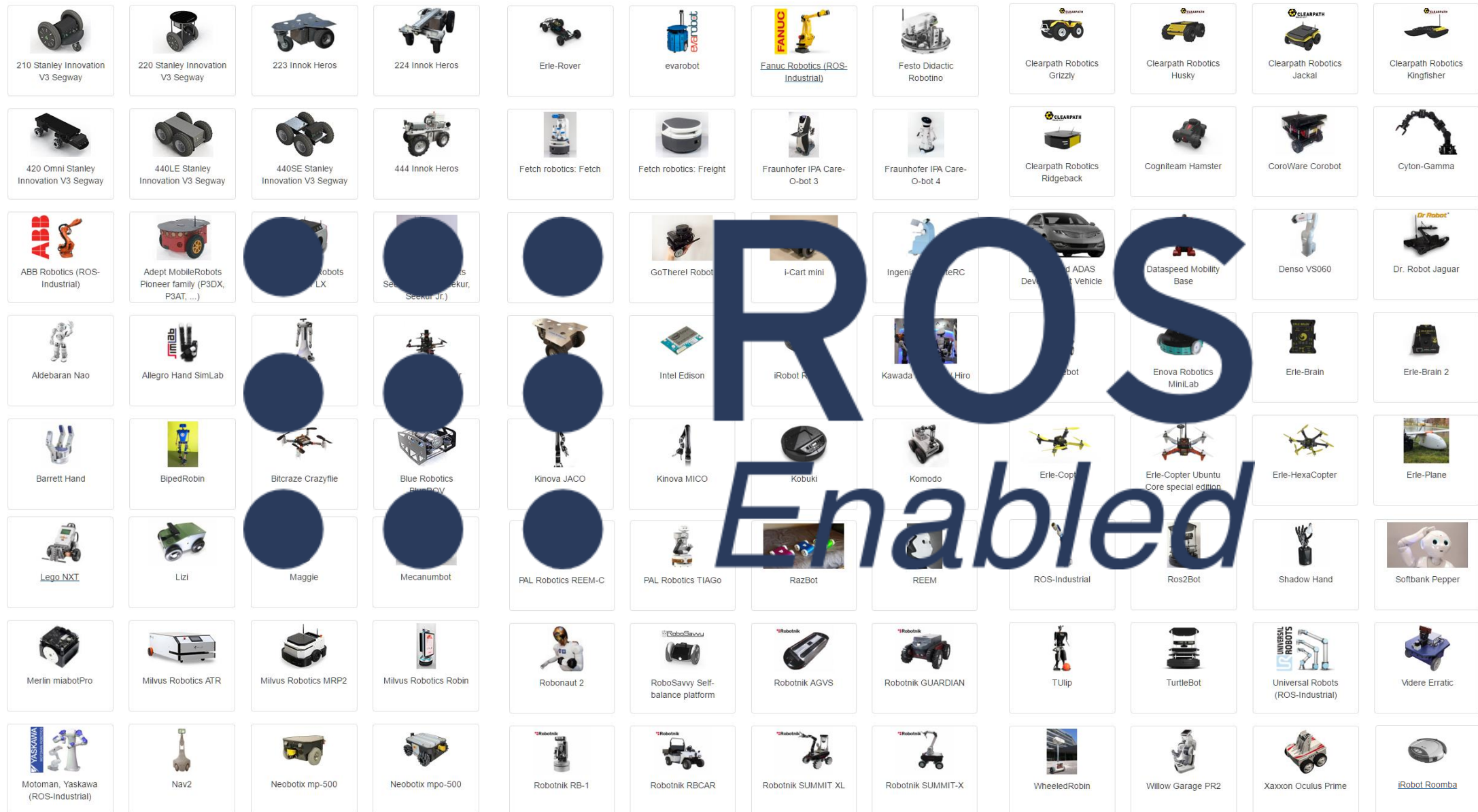


Robot, Sensor Company

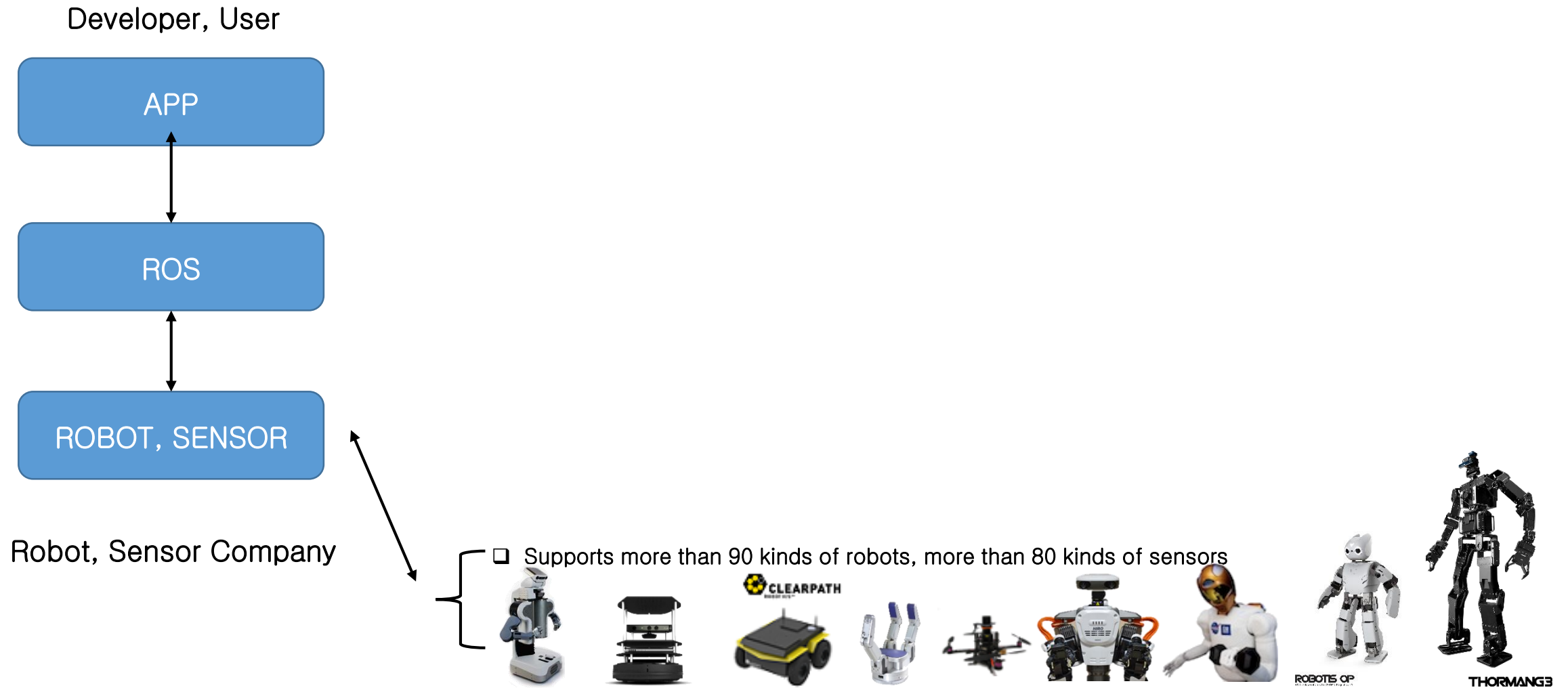
Current ROS ecosystem



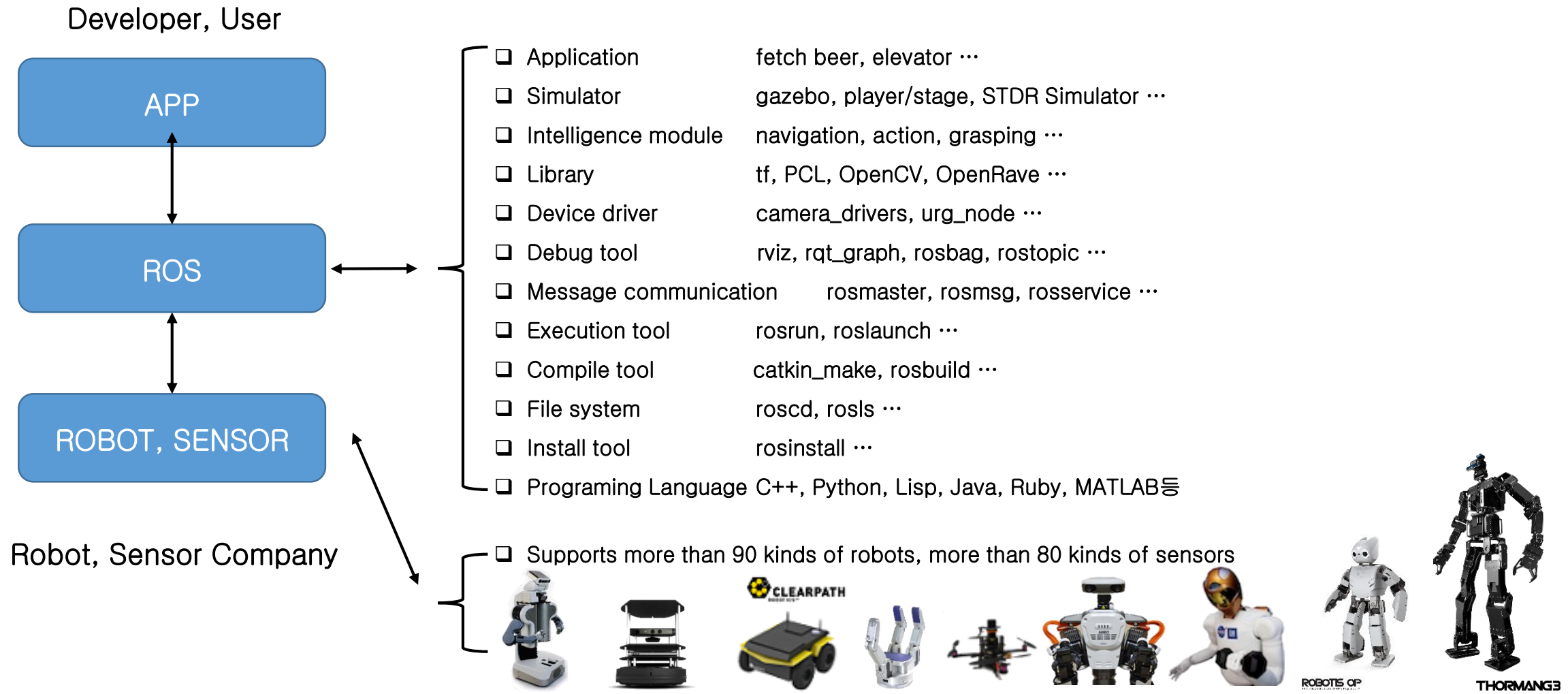
Current ROS ecosystem



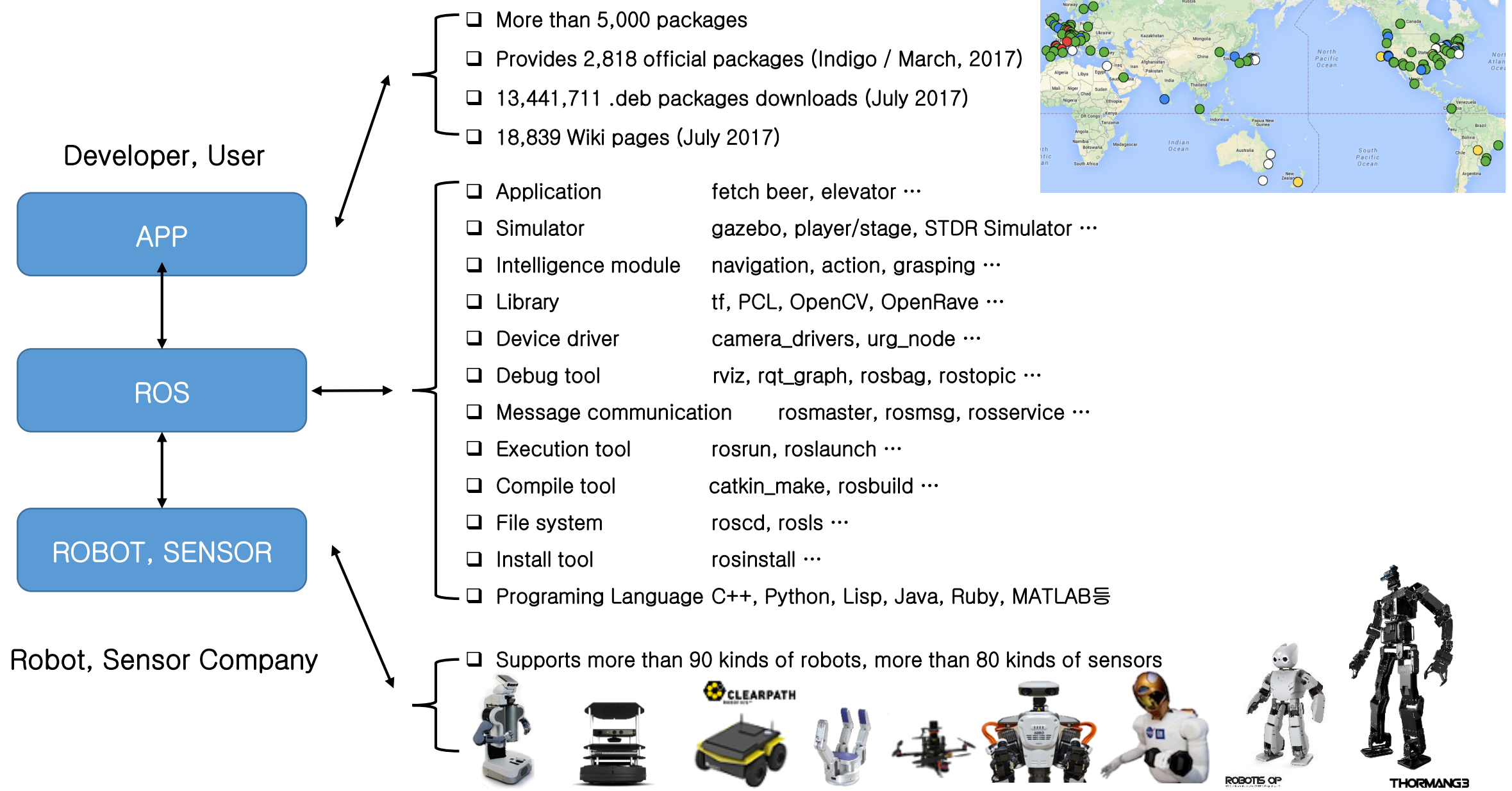
Current ROS ecosystem



Current ROS ecosystem



Current ROS ecosystem



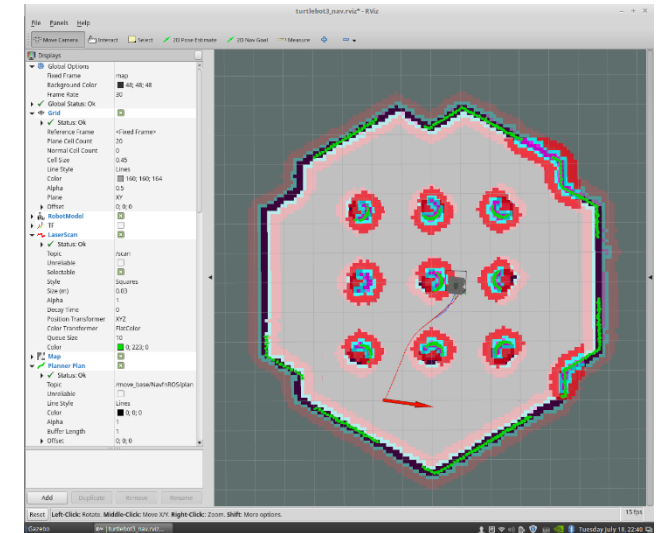
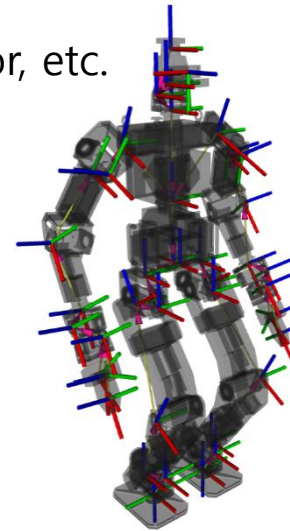
Features of Robot operating system ROS

Feature 1) Communication infrastructure

- Provides data communication between nodes
- Support for message transfer interface, which is commonly referred as the middleware
- **Message parsing function**
 - Provides communication system frequently used in robot development
 - Message transfer interface between nodes facilitating encapsulation and code reuse
- **Message Record and Play**
 - Messages that are transmitted/received between nodes can be stored and reused as needed
 - It is possible to repeat an experiment based on stored messages, and it is easy to develop algorithm
- **Use of various programming languages due to the use of messages**
 - Since data exchange between nodes use messages, each node can be written in different languages
 - Client libraries: roscpp, rospy, roslisp, rosjava, roslua, roscs, roseus, PhaROS, rosR
- **Distributed parameter system**
 - Variables used in the system are created as global key values so they can be shared, modified and applied in real-time

Feature 2) Various functions related to robots

- Define a standard message for a robot
 - Modularization by defining standard message such as camera, IMU, laser sensor / odometry, navigation data such as route and map, inducing collaborative work, and improving efficiency
- Robot geometry library
 - Provides TF to calculate the relative coordinates of robot, sensor, etc.
- Robot description language
 - XML document describing physical characteristics of the robot
- Diagnostic system
 - Provides diagnostic system to grasp the state of the robot
- Sensing / recognition
 - Sensor drivers, libraries for sensing / recognition
- Navigation
 - Estimation of poses (position / posture) of robots commonly used in robots, provision of self position estimation in the map
 - SLAM required for map creation, and Navigation library for navigating to destinations within the created map
- Manipulation
 - Provides various Manipulation libraries to support IK and FK used in robot arm as well as pick and place of application
 - Provides GUI manipulation tools (MoveIt!)



Feature 3) Various development tools

- Provides various development tools needed for robot development
- Improving the efficiency of robot development

■ Command-Line Tools

- Access to the robot and use ROS functions only with commands provided by ROS without GUI

■ RViz

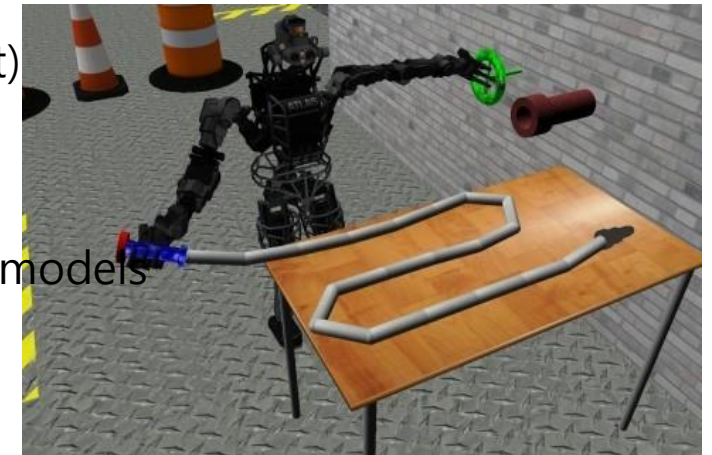
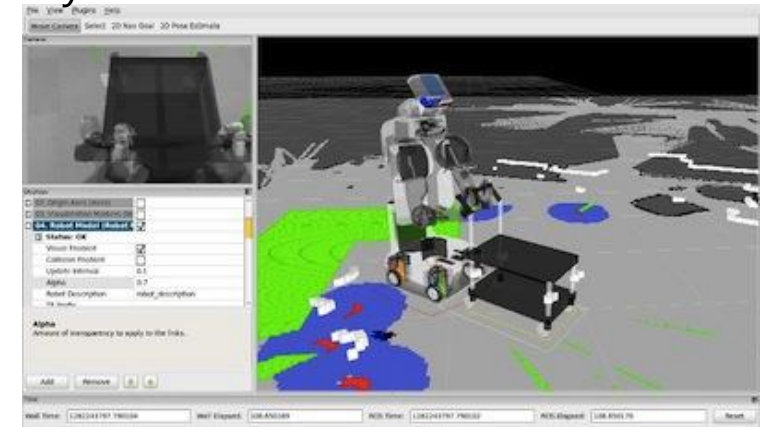
- Provide powerful 3D visualization tool
- Visualize sensor data such as laser, camera, etc.
- Represent robot outline and planned motion

■ RQT

- Provides Qt-based framework for developing graphic interface
- Displays connection information among nodes (rqt_graph)
- Values such as encoder, voltage, numbers that change over time (rqt_plot)
- Records and plays data in the form of message (rqt_bag)

■ Gazebo

- 3D simulator with physics engine. Supports robot, sensor, environmental models
- Highly Compatible with ROS



Selecting ROS version

ROS release schedule and version selection!

- Full version of ROS will be released once a year since Hydro!

- 2017.05.23 - Lunar Loggerhead
- **2016.05.23 - Kinetic Kame (LTS) Recommendation**
- 2015.05.23 - Jade Turtle
- 2014.07.22 - Indigo Igloo (LTS)
- 2013.09.04 - Hydro Medusa
- 2012.12.31 - Groovy Galapagos
- 2012.04.23 - Fuerte Turtle
- 2011.08.30 - Electric Emys
- 2011.03.02 - Diamondback
- 2010.08.02 - C Turtle
- 2010.03.02 - Box Turtle
- 2010.01.22 - ROS 1.0
- Lunar Loggerhead (EOL=May, 2019)
- **Kinetic Kame (EOL=April, 2021)**
- Jade Turtle (EOL=May, 2017)
- Indigo Igloo (EOL=April, 2019)

Version Selection

- 1) Choose LTS version of Ubuntu. It will be supported for 5 years
- Every 2 years in April, LTS version of Ubuntu will be released!
- 2) Latest LTS-supported ROS version
- 3) ROS is released three months after Ubuntu release (in general)
- 4) Select version after checking Gazebo "gazebo.org" information



- **Ubuntu 16.04.x Xenial Xerus LTS**
- **ROS Kinetic Kame**
- **Gazebo 7.x**

Question Time!

Advertisement #1



Free

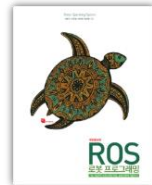


Download link



Language:

English, chinese, Japanese, Korean



“ROS Robot Programming”

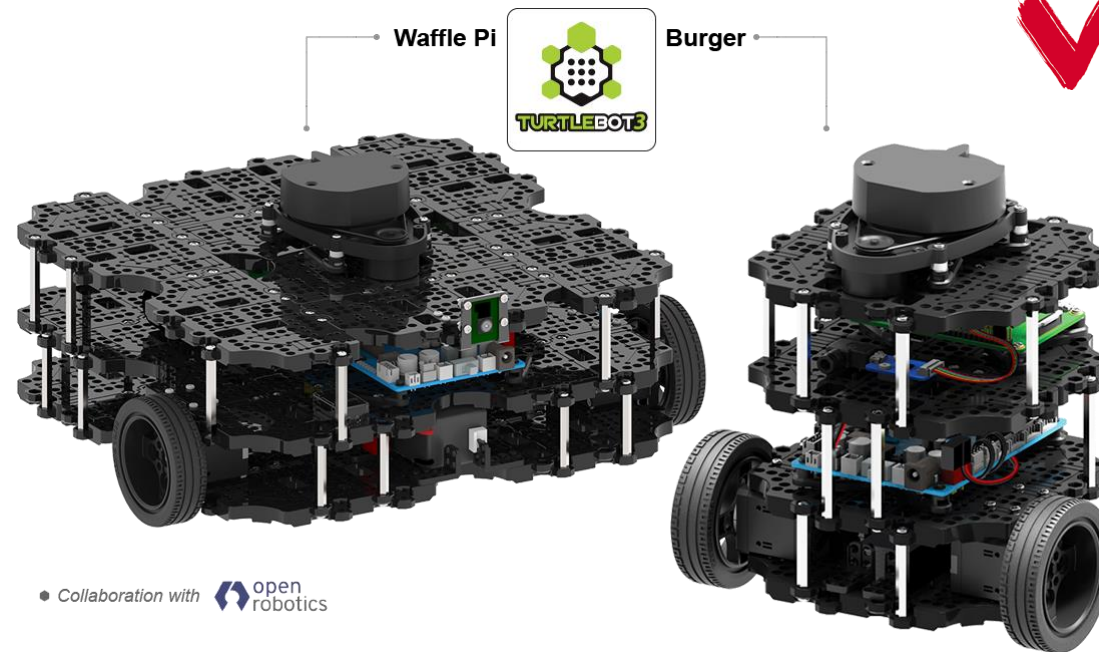
A Handbook is written by TurtleBot3 Developers

Advertisement #2

TURTLEBOT3

AI Research Starts Here
ROS Official Platform

TurtleBot3 is a new generation mobile robot that's modular, compact and customizable. Let's explore ROS and create exciting applications for education, research and product development.



✓ [Direct Link](#)

Advertisement #3



www.robotsource.org

The 'RobotSource' community is the space for people making robots.

We hope to be a community where we can share knowledge about robots, share robot development information and experiences, help each other and collaborate together. Through this community, we want to realize open robotics without distinguishing between students, universities, research institutes and companies.

Join us in the Robot community ~

END.