

# Basic ROS building blocks

- Packages
- Nodes
- Topics
- Messages
- Services

# ROS Packages

- ROS software is divided into packages that can contain programs, images, data and even tutorials.
- A package provides a useful function for ease of use and especially reuse.
- ROS package contains a [package.xml](#) file (sometimes called a manifest).
- ROS packages are built with [catkin](#) build process.
- Multiple related packages are combined into a [metapackage](#).

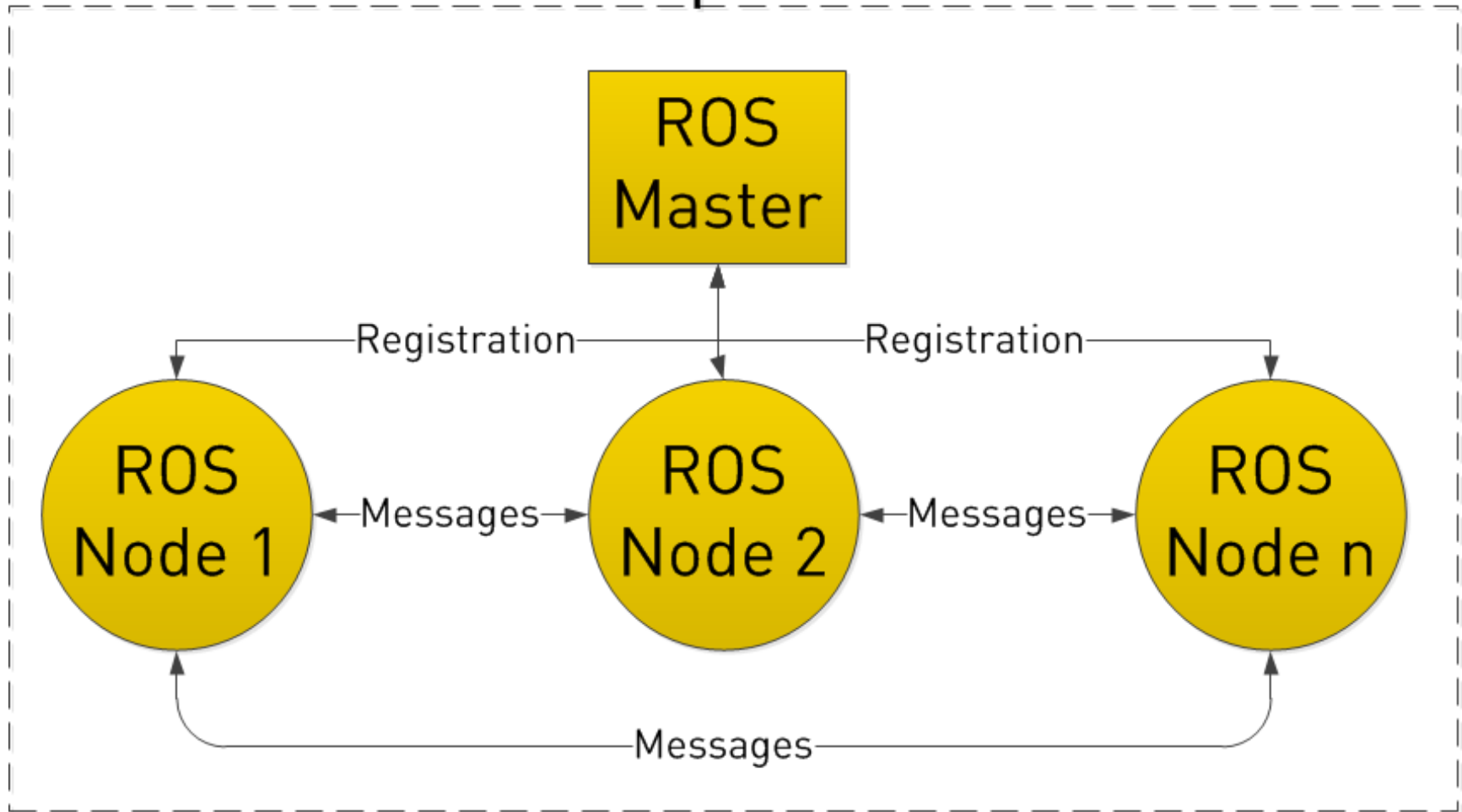
# ROS Nodes

- A ROS system is comprised of a number of independent nodes, each of which communicates with the other nodes using a publish/subscribe messaging model.
- ROS nodes are built small to control one aspect of a robotic system.
- For example, a USB camera driver is implemented as a node, which publishes image data in a stream of messages. These messages can be consumed by any number of other nodes.
- Nodes in ROS do not have to be on the same computer or even of the same architecture!

# ROS Master

- ROS must start with a ROS Master.
- The Master allows all other ROS software processes (nodes) to find and talk to each other.
- All nodes must register with the Master when they startup.
- After registration, Node 1 can send and receive messages from Node2.

# Computer 1



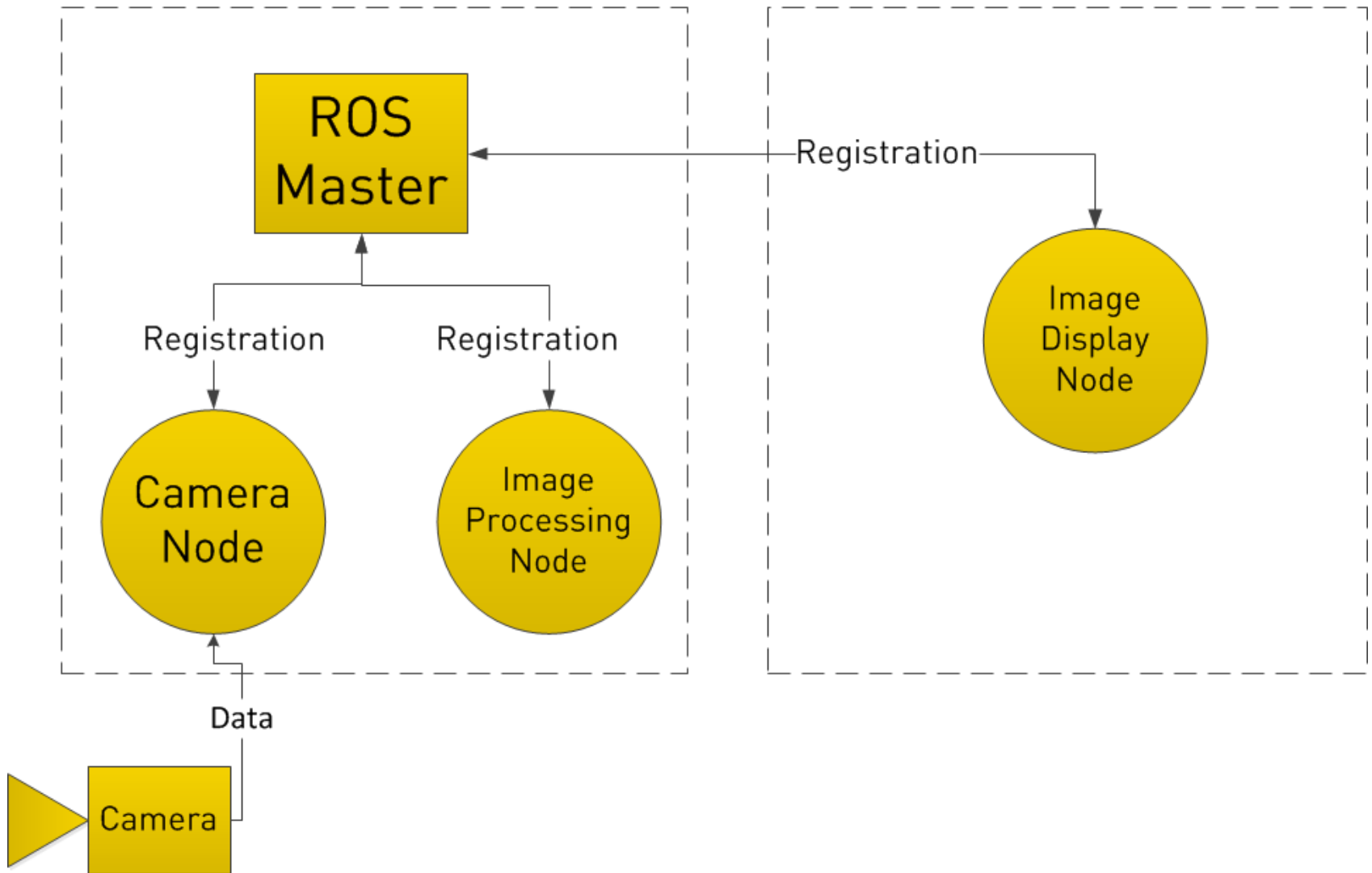
Images from <https://www.clearpathrobotics.com/2014/01/how-to-guide-ros-101/>

# ROS Topics

- ROS topics are published and subscribed to by nodes.
- To see images from the camera, a **Camera Node** communicates with the camera and an **Image Processing Node** processes the image data. A third node, an **Image Display Node** displays images on the computer screen.

# Computer on the Robot

# Laptop



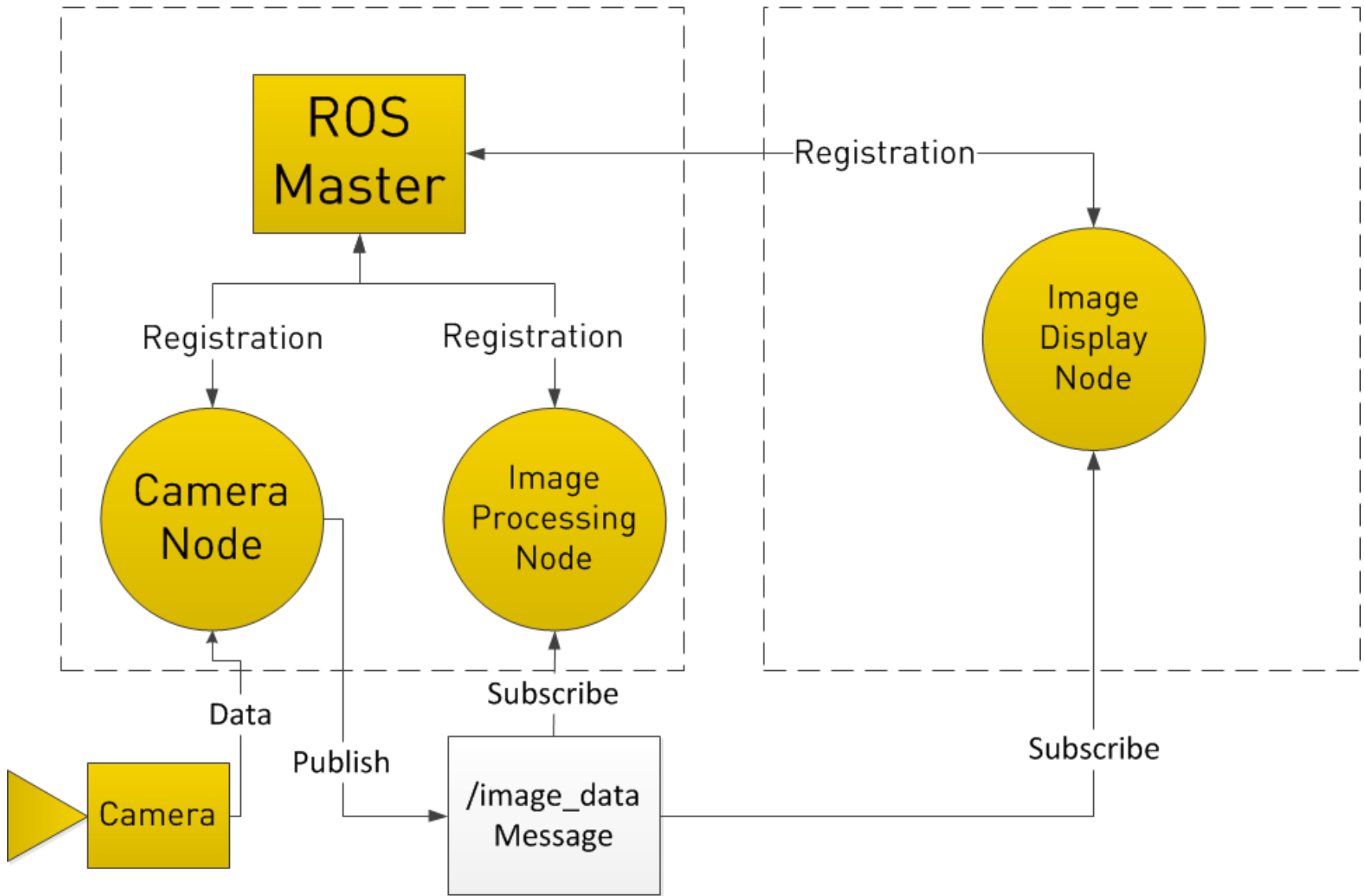
# ROS Master (continued)

- As these nodes register with the Master:
  - The **Camera Node** declares that it publishes a topic called */image\_data*.
  - The **Image Processing Node** and the **Image Display Node** declare that they subscribe to the topic */image\_data*.
- **Camera Node** interfaces with the camera and creates and sends */image\_data* messages directly to the other two nodes.



# Computer on the Robot

# Laptop



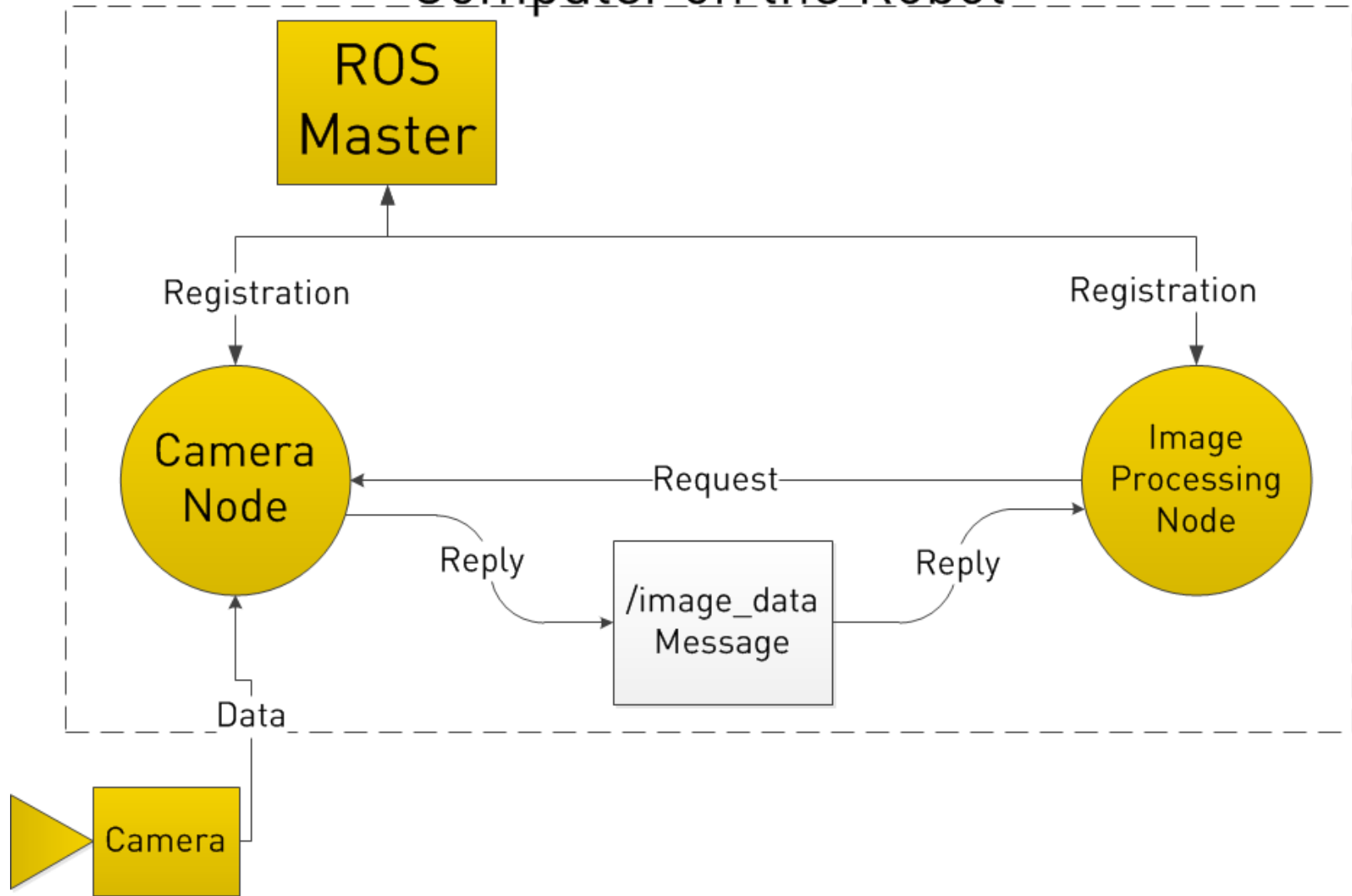
# ROS Messages

- ROS message is a strictly typed data structure.
- Analogy:
  - Topic is the suitcase
  - Messages are the contents

# ROS Services

- Nodes register a specific service with the ROS Master (just like topics).
- ROS services work on a request and response cycle.
- For example, the **Image Processing Node** requests the service for */image\_data*. The **Camera Node** acquires the data from the camera and sends the reply.

# Computer on the Robot



# ROS Parameter Server

- ROS Parameter Server is a dictionary of parameters shared between the nodes.
- Accessed by the ROS Master (seamless)
- Exchange of parameters (data) between nodes at runtime

# ROS help

- ROS wiki <http://wiki.ros.org/>
- ROS getting started  
<http://wiki.ros.org/ROS/StartGuide>
- ROS tutorials <http://wiki.ros.org/ROS/Tutorials>
- ROS answers  
<http://answers.ros.org/questions/>
- ROS blog <http://www.ros.org/news/>