Introduction to Robot Operating System (ROS)

Application to mobile robots

Amr Elhussein
Advisor: Dr. Suruz Miah

Department of Electrical and Computer Engineering
Bradley University
1501 W. Bradley Avenue
Peoria, IL, 61625, USA

Friday, May 31, 2019

BRADLEY University

Outline

- Introduction
  - Historical Background
  - Robot Programming Before ROS
  - ROS is ..
  - ROS Equation
  - Applications
- ROS Concepts
  - Filesystem
  - Computation Graph
  - Community level
- ROS installation
- Future of ROS

BRADLEY University

Intrduction

History and Legacy

- Started in 2007 by researches from Stanford AI Robot (Stair) and the Personal Robots (PR) Program and was sponsored by Willow Garage a visionary robotics incubator.
- Used Worlwide in Research and Industry.
- Currently supported by the Open Source Robotics Foundation.

Figure: Stair

Intrduction

Robot Programming Before ROS

- No common platform for develeoping robotics
- Build every thing from scratch
- Algorithm implementation

Intrduction

ROS is ..

A flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

Intrduction

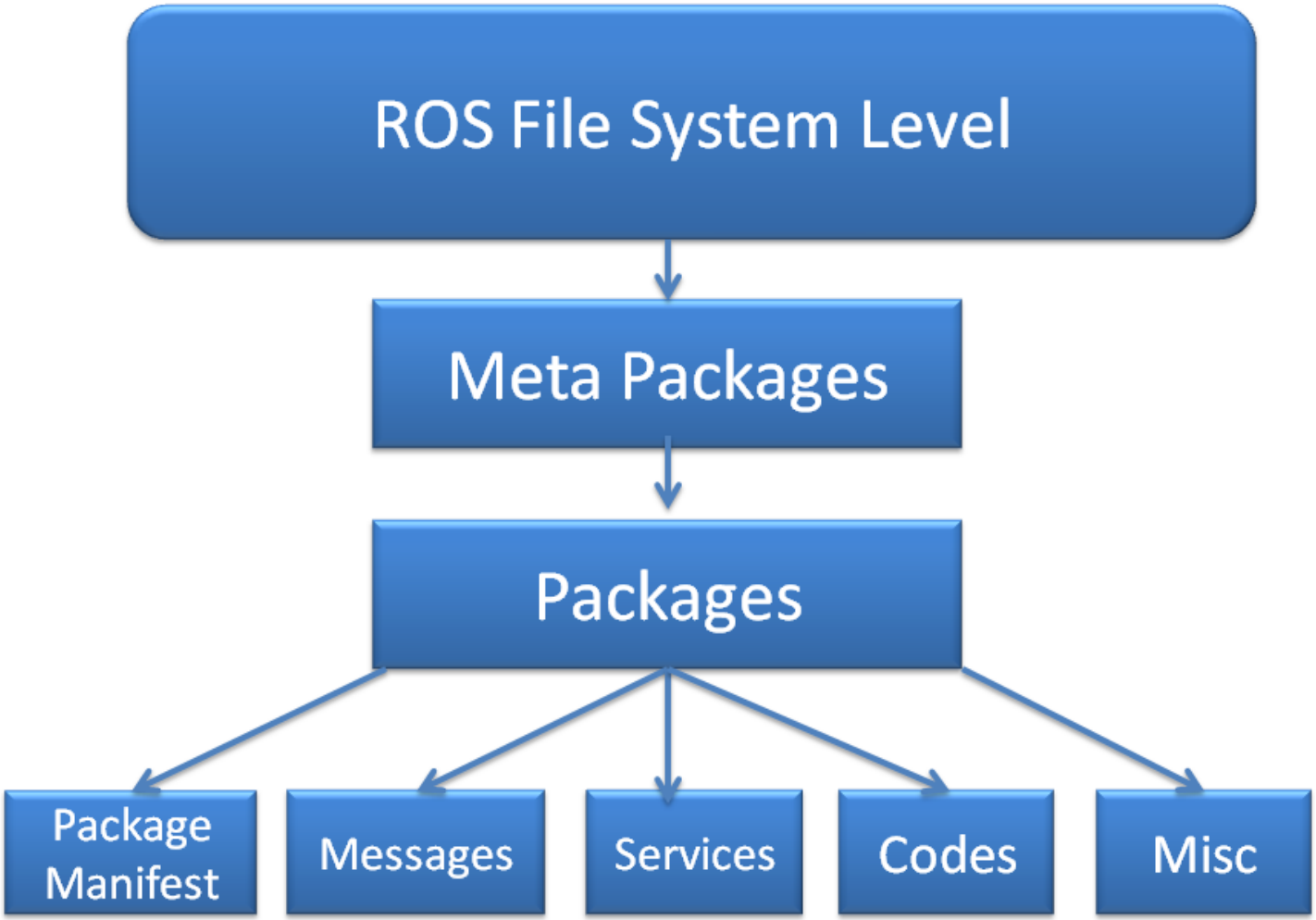
Ros Equation

Intrduction

Applications

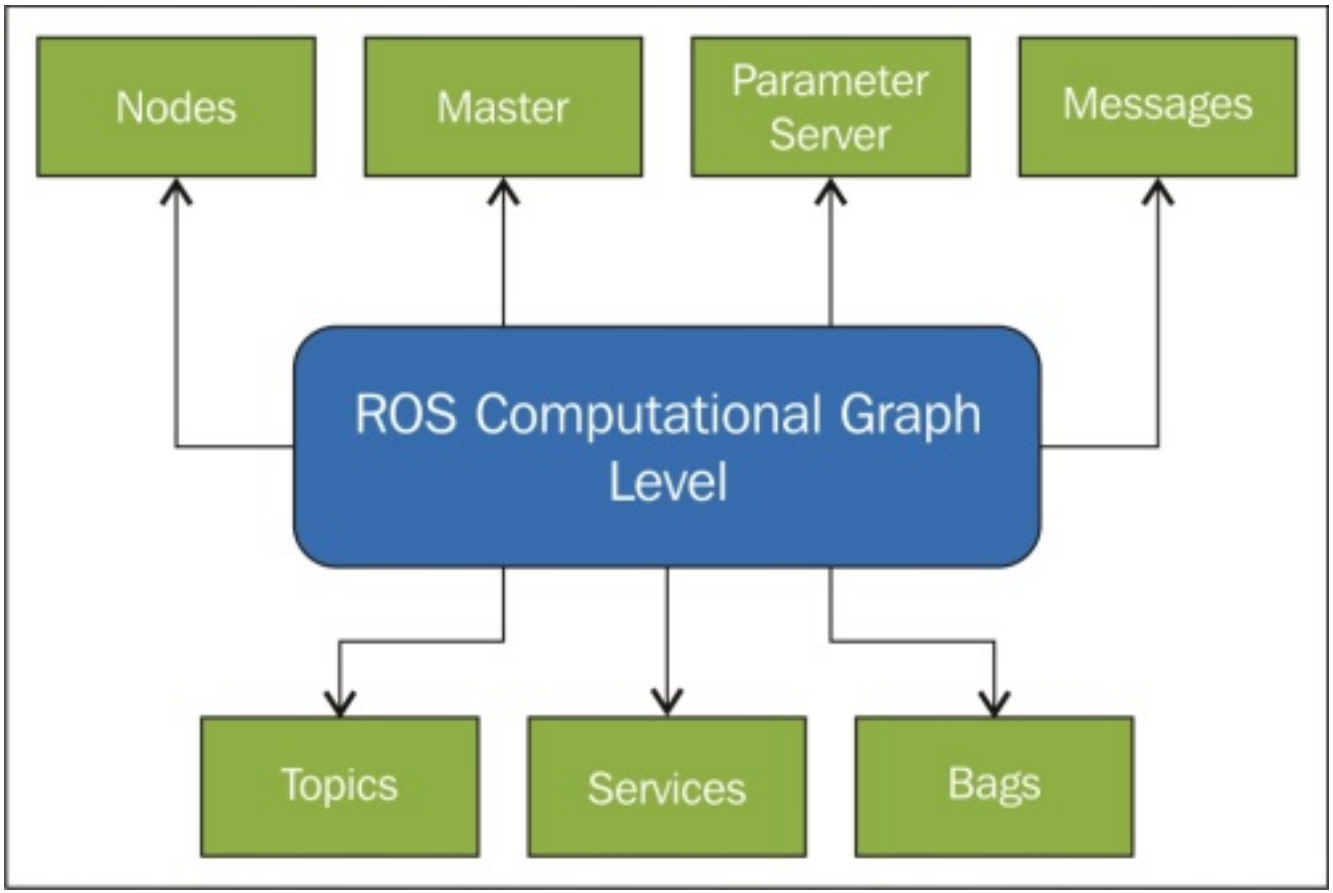
ROS Concepts

Filesystem



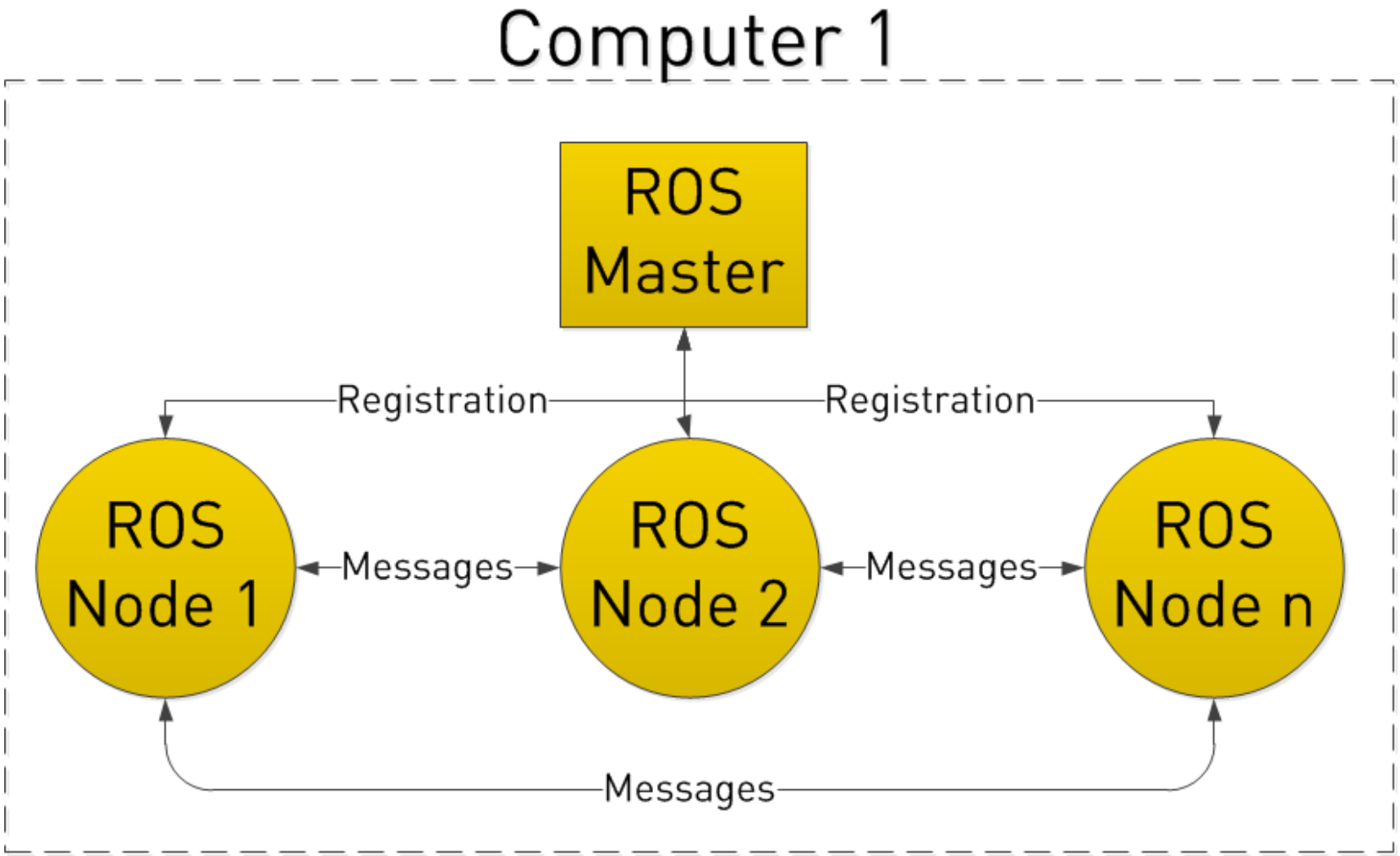
ROS Concepts

Computation Graph



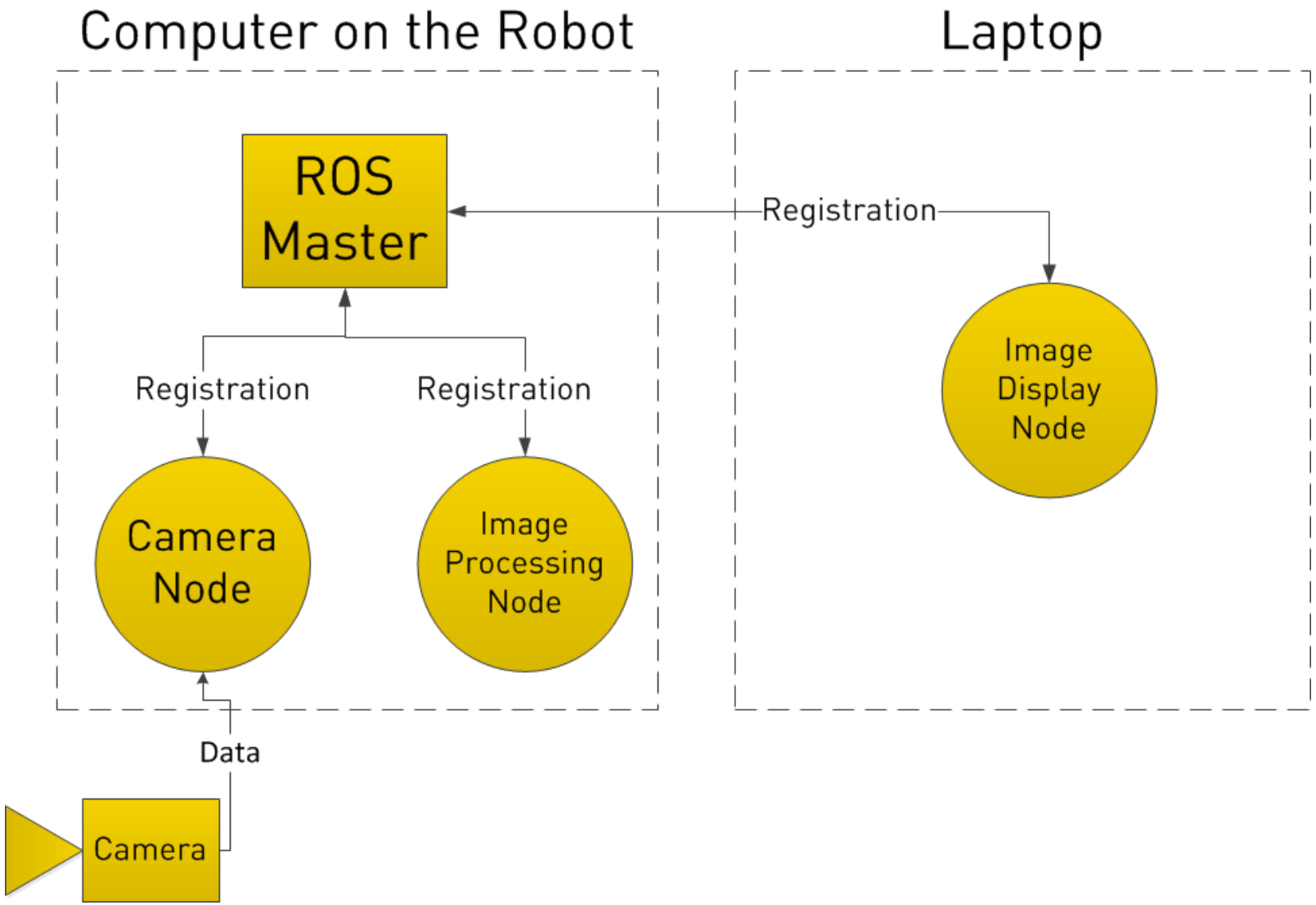
ROS Concepts

Computation Graph: Master



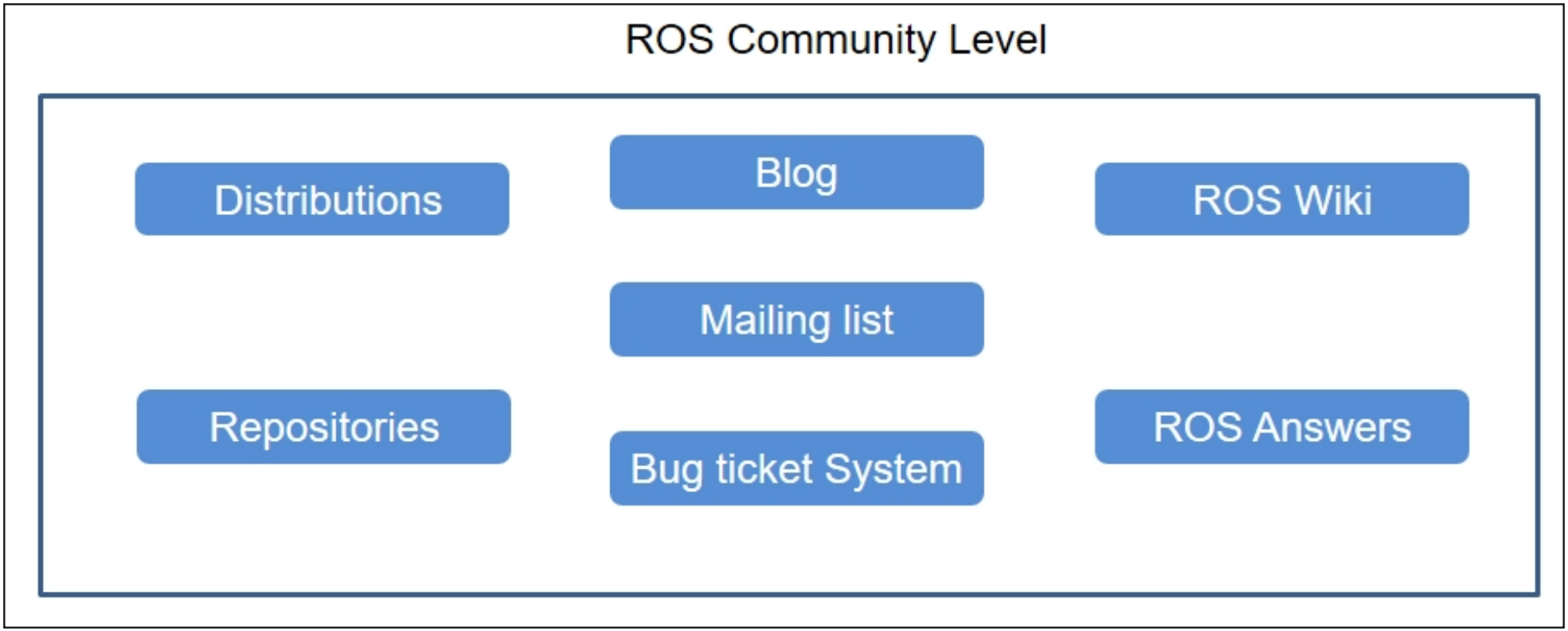
ROS Concepts

Computation Graph: Master



ROS Concepts

Community level



Installation

- Debian-based distributions such as Ubuntu.
- Many robots.
- Current supported distributions
  - ROS Kinetic Kame, Released May, 2016.
  - ROS Melodic Morenia, Released May, 2018

Installation

After choosing the distribution follow the instruction on ROS Wiki which start by:

- Configure your Ubuntu repositories.
- Setup your sources.list.
- Set keys.
- Install with "sudo apt-get install ros-kinetic-desktop-full".

Future of ROS

- Security
- Critical Missions
- Distributed Processing

Thanks !

Matlab Robotics Systems Toolbox

Application to mobile robots

Amr Elhussein
Advisor: Dr. Suruz Miah

Department of Electrical and Computer Engineering
Bradley University
1501 W. Bradley Avenue
Peoria, IL, 61625, USA

Friday, June 4, 2019

BRADLEY University

Outline

- Introduction
- Workflow
  - Desktop prototyping
  - Standalone ROS Nodes
- Examples

Introduction

Robotics System Toolbox provides algorithms and hardware connectivity for developing autonomous robotics applications for aerial and ground vehicles, manipulators, and humanoid robots. Toolbox algorithms include path planning and path following for differential drive robots, scan matching, obstacle avoidance, and state estimation. For manipulator robots, the system toolbox includes algorithms for inverse kinematics, kinematic constraints, and dynamics using a rigid body tree representation.

BRADLEY University

Workflow

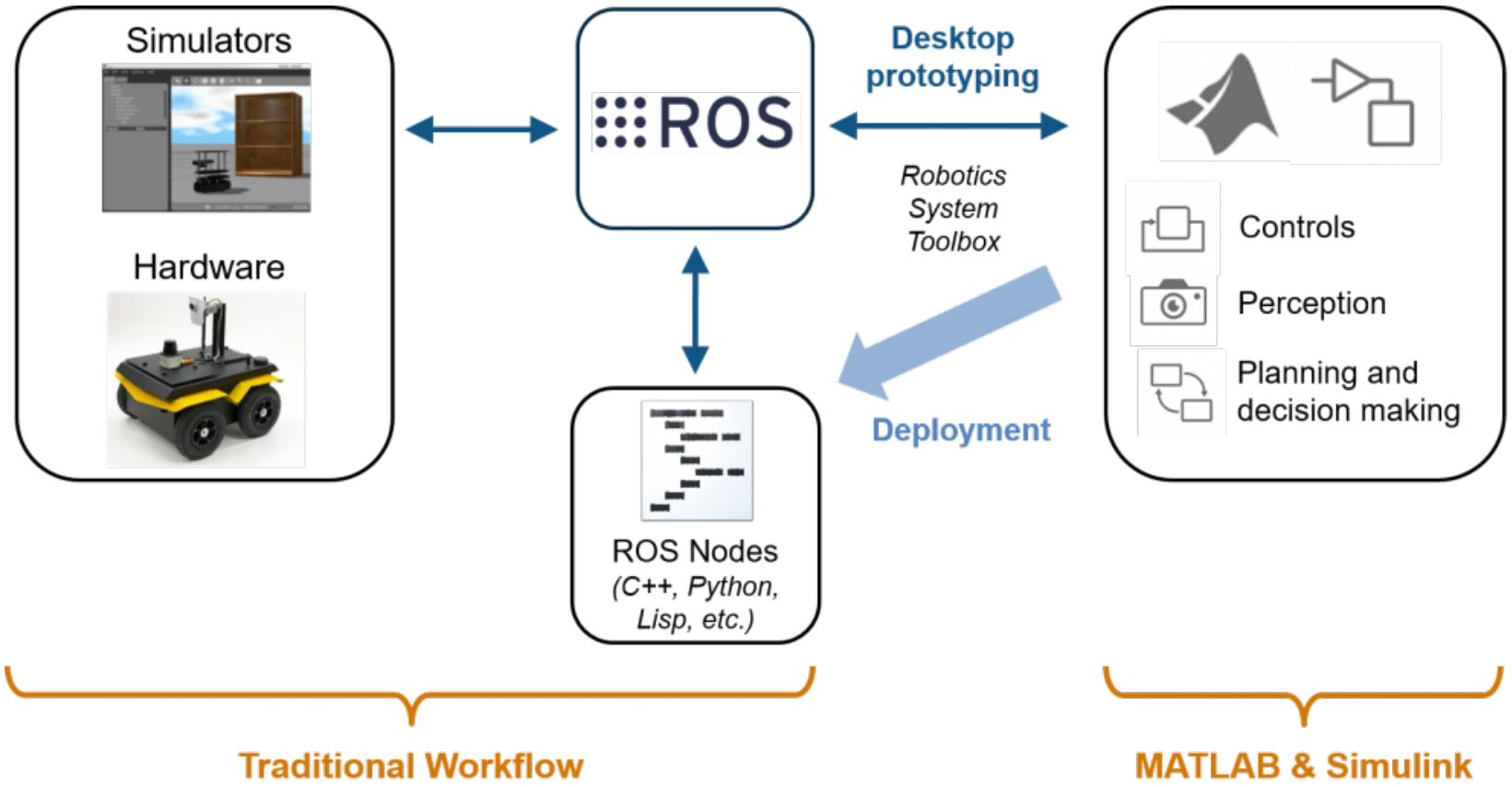


Figure: Matlab robotics tool box and ROS workflow. courtesy of mathworks.com

Workflow

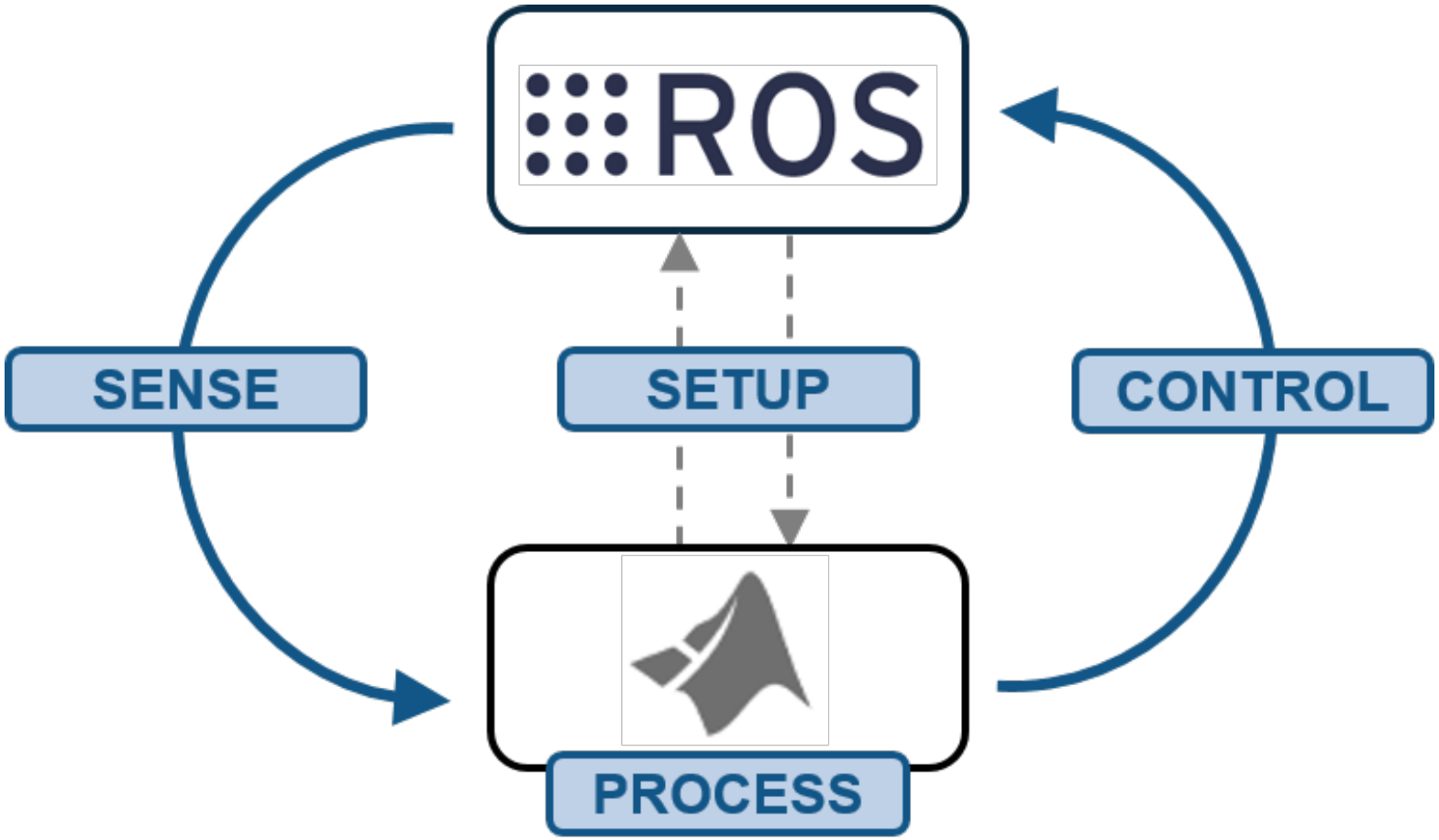


Figure: Matlab and ROS integration, courtsey of mathworks.com

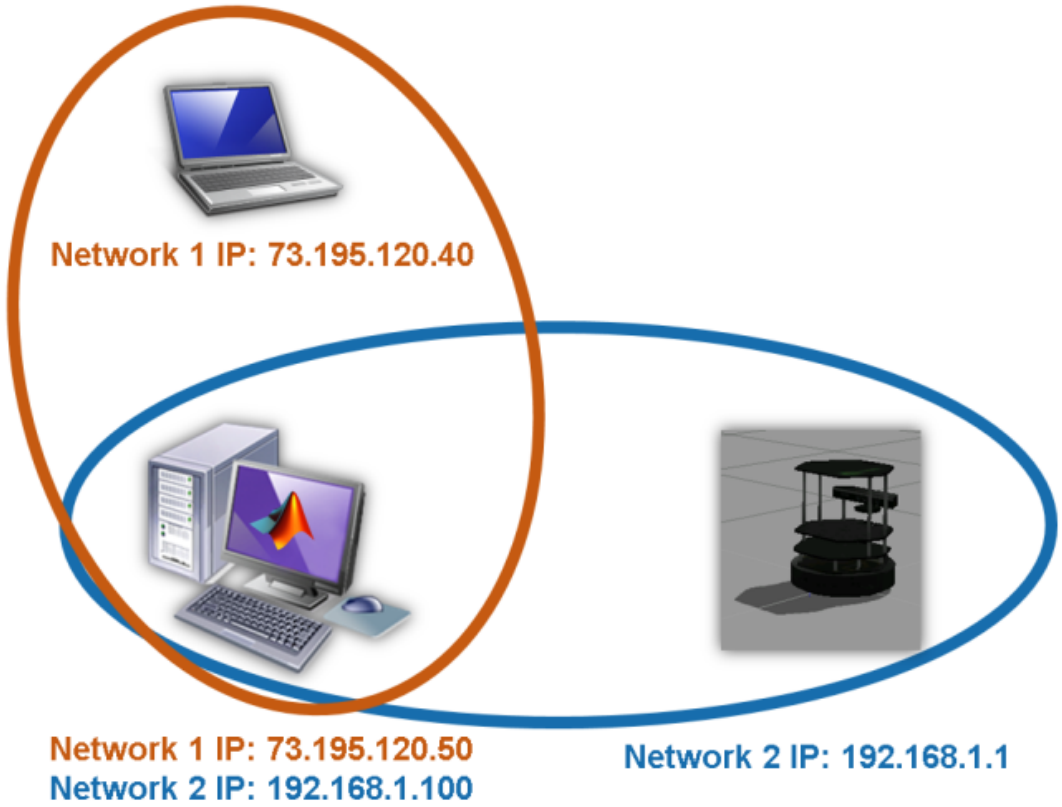
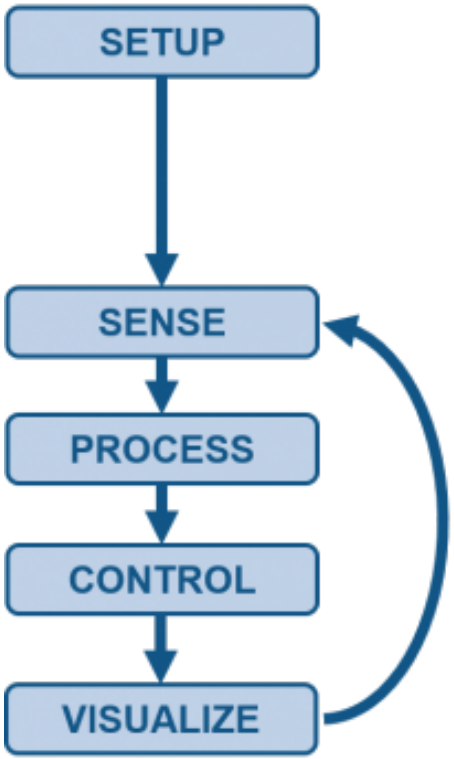


Figure: Matlab ROS desktop prototyping, mathworks.com

Workflow

Desktop prototyping

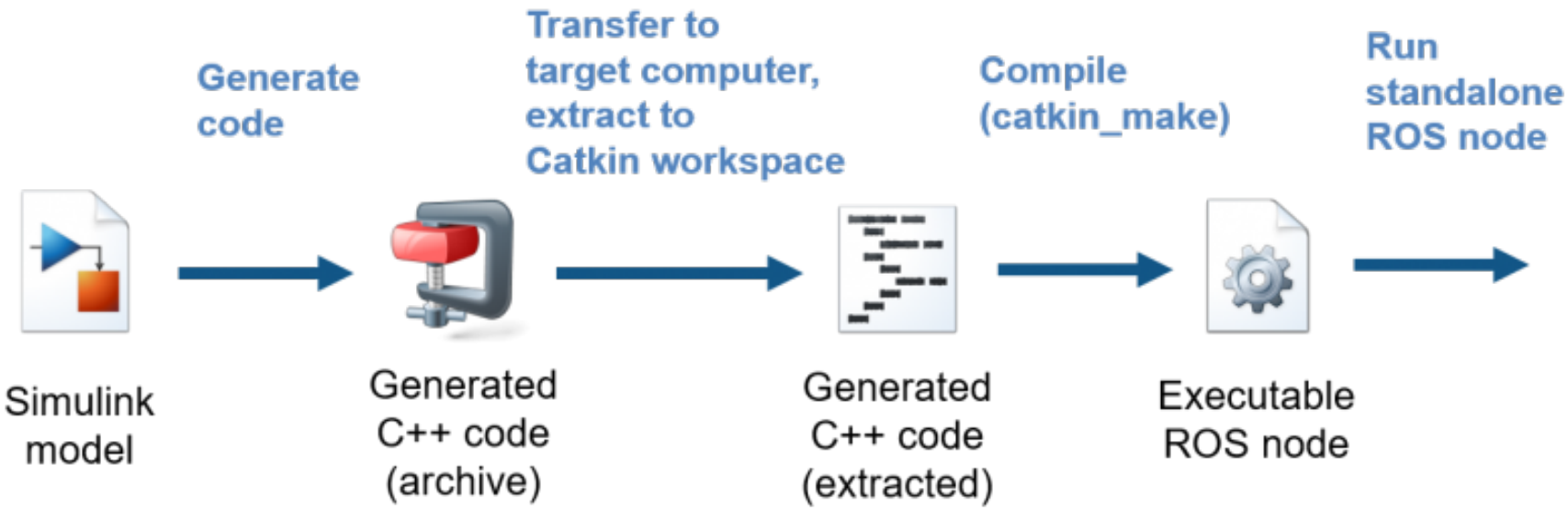



```
rosinit('ipAddress')
mySub = rossubscriber('/sub_topic');
[myPub,pubMsg] = rospublisher('/pub_topic');
currentTime = 0;

tic
while(currentTime < 10)
  recvMsg = mySub.LatestMessage;

  ctrlOut = myAlgorithm(recvMsg);

  pubMsg.FieldName = ctrlOut;
  send(myPub,pubMsg);

  currentTime = toc;
  plot(currentTime,ctrlOut)
end
```

BRADLEY University

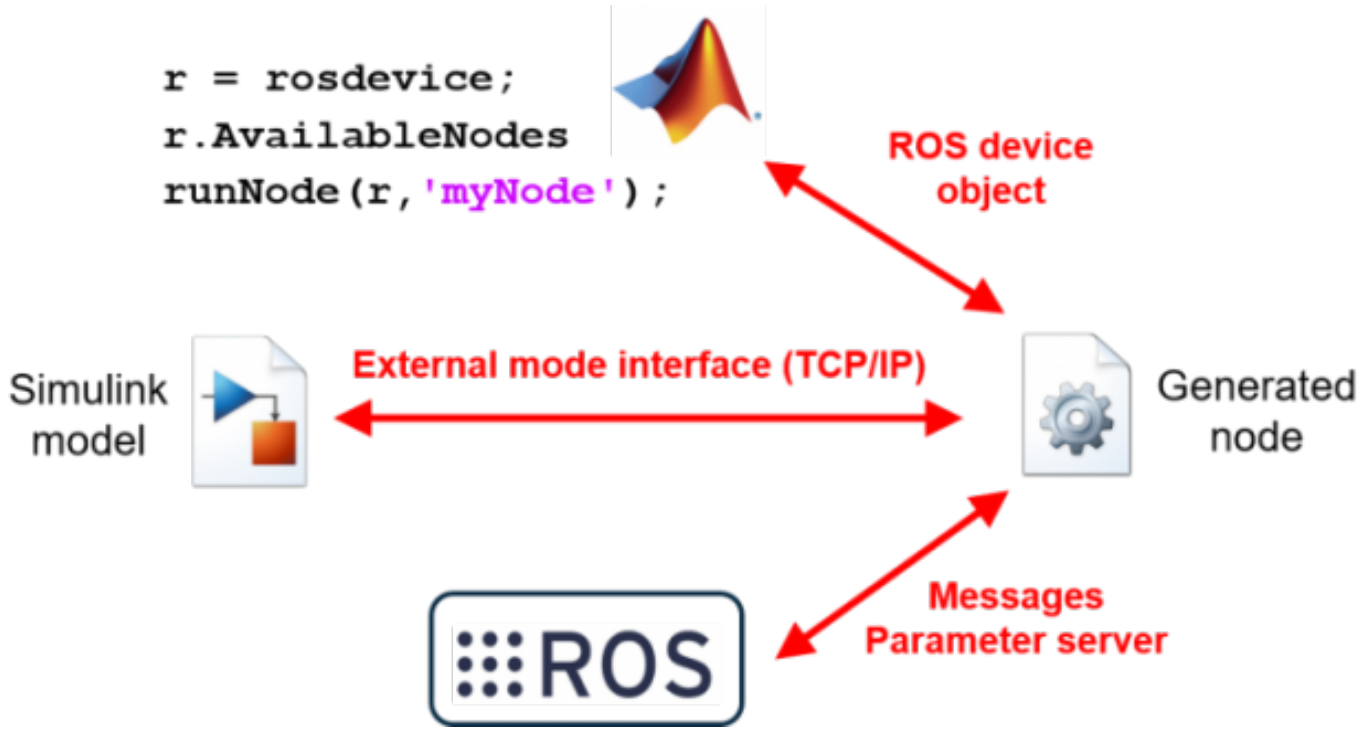
Worflow

Standalone Node



Workflow

Standalone Node



Examples