ROS – Robot Operating System

Embedded Motion Control 2012

Jos Elfring

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

What is ROS?

- Open-source meta-operating system for robots
- Primary goal: support code reuse in robotics R&D
- Implemented in C++, python, lisp
- Allows running code on multiple computers

Basic Concepts (1/3)

- **Nodes:**
  processes that perform computation

- **Master:**
  provides name registration and lookup

- **Parameter server:**
  allows data storage by key in a central location

Basic Concepts (2/3)

- **Messages:**
  nodes communicate with each other by passing messages

- **Topics:**
  named buses over which nodes exchange messages

- **Services:**
  request/reply communication

Basic Concepts (3/3)

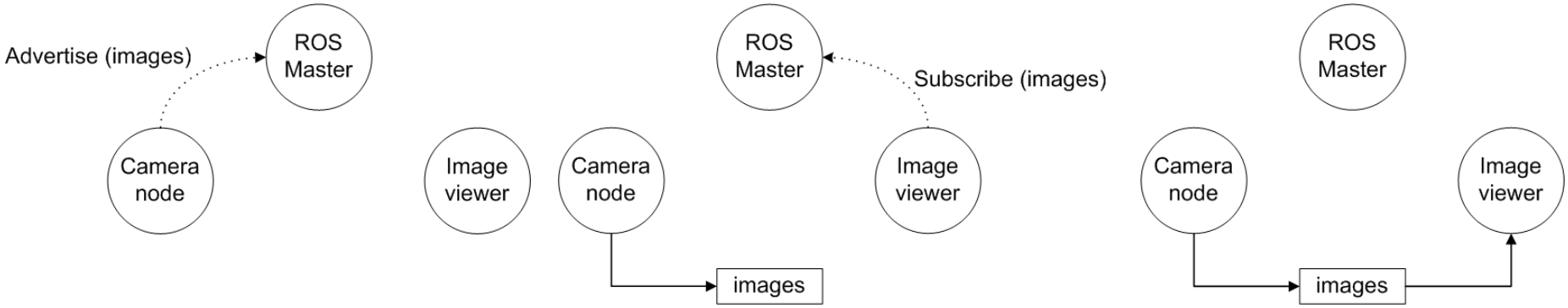
- **Bags:**
  Format for storing and playing back message data

Basic Concepts: examples

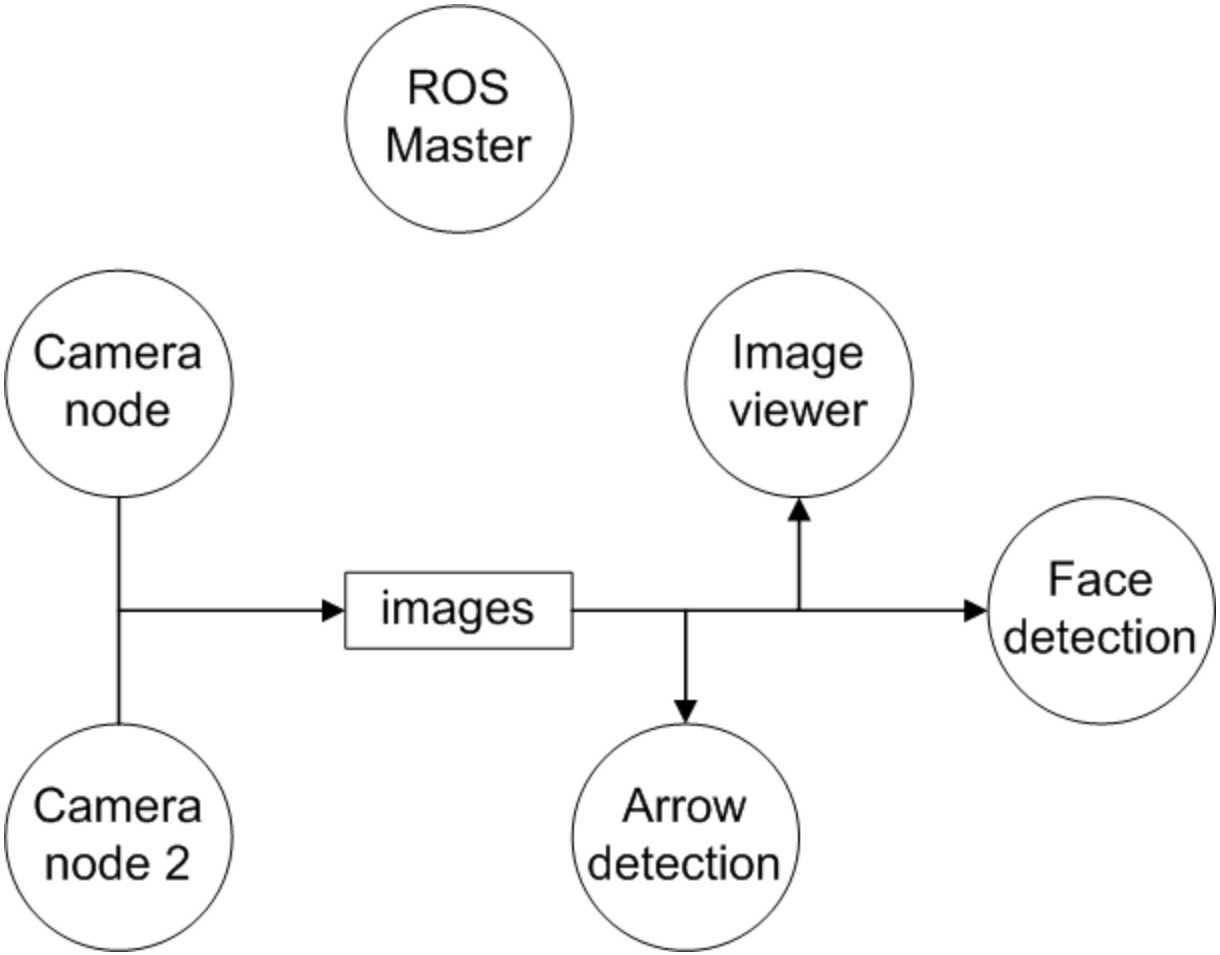
“Camera” node wants to publish on topic “images”

“Image viewer” node wants to subscribe to topic “images”

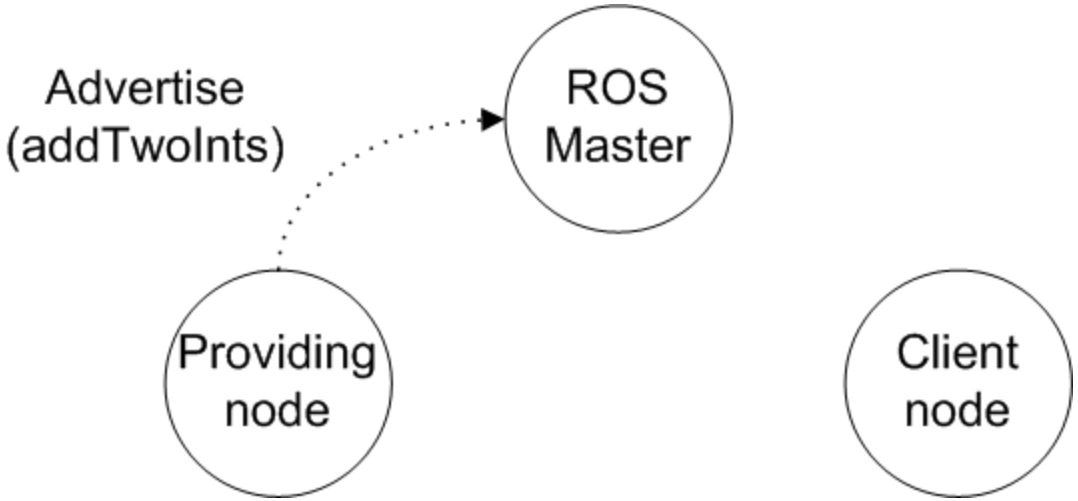
Master informs nodes about each others existence



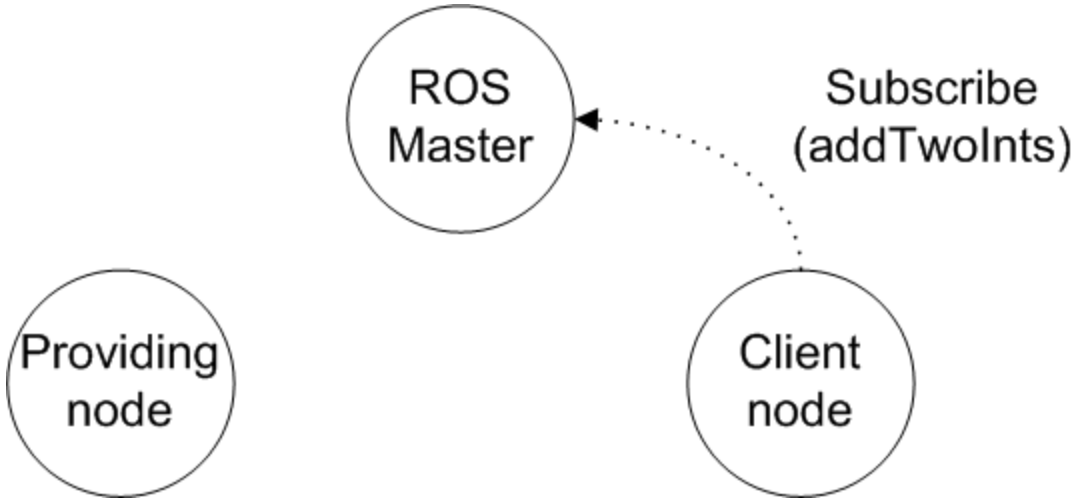
Basic Concepts: examples



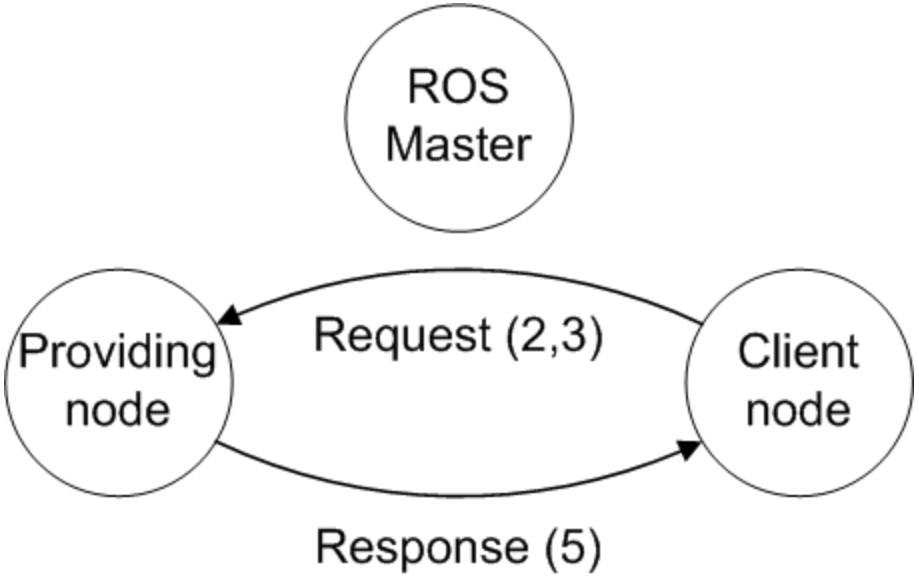
Basic Concepts: examples



Basic Concepts: examples



Basic Concepts: examples



ROS Filesystem (1/2)

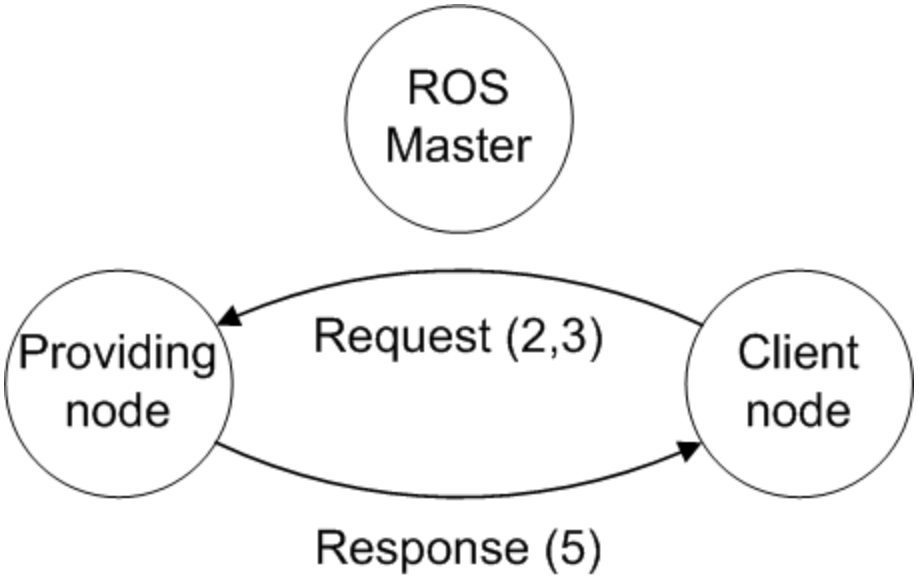
- **Packages:**
  Contains anything that is usefully organized together
- **Manifest:**
  Contains meta data about a package
- **Stack:**
  Collection of packages that provides functionality
- **Stack manifest:**
  Data about the stack

ROS Filesystem (2/2)

- **Message types:**
  Message description, data structure for sending messages in ROS

- **Service types:**
  Defines request/response data structures for services in ROS

Basic Concepts: examples



AddTwoInts.srv:

```
int64 A
int64 B
---
int64 Sum
```

More Information

- **Tutorials and documentation: www.ROS.org**

- **Course wiki: http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control**