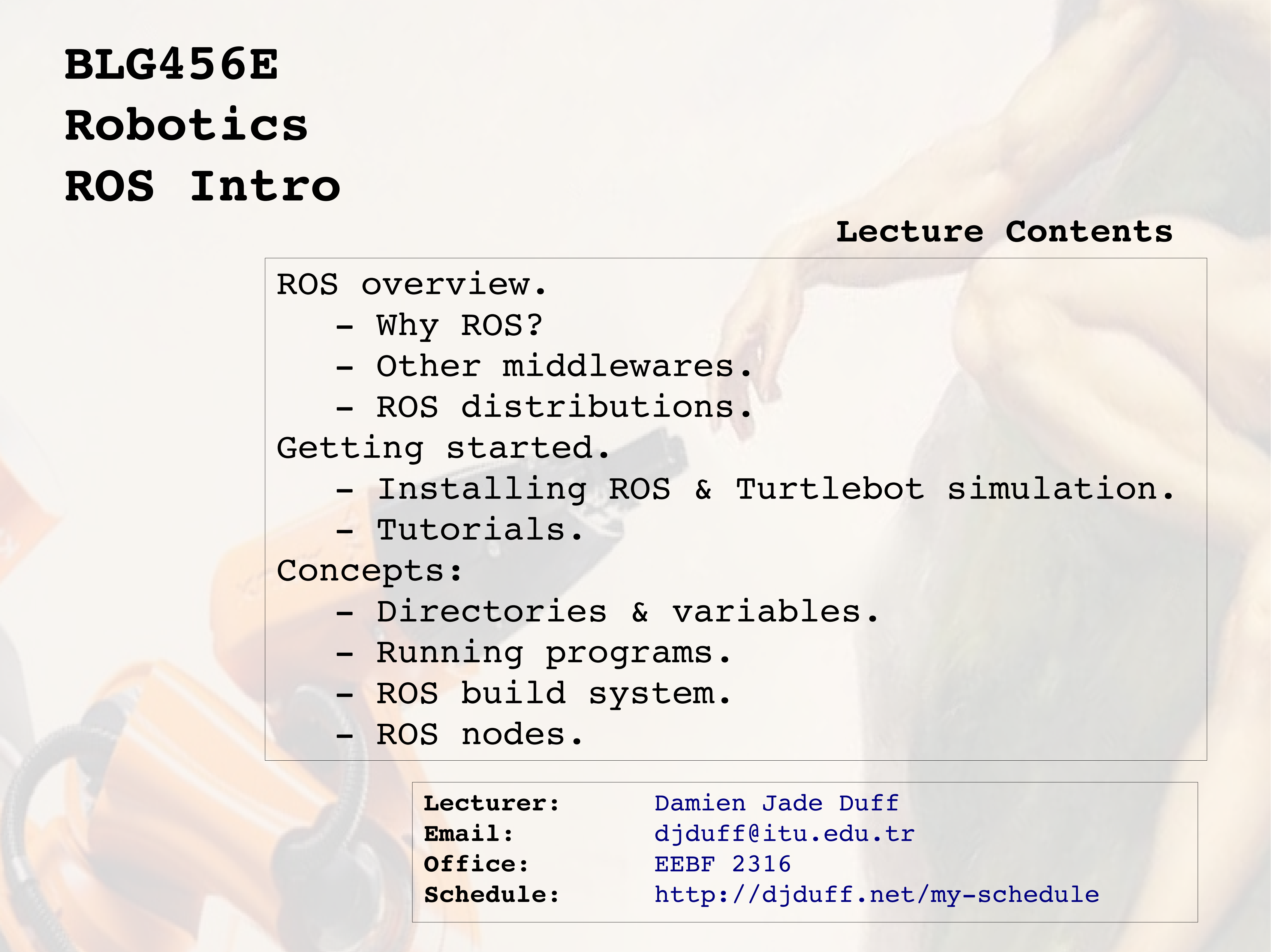


BLG456E
Robotics
ROS Intro

Lecture Contents

ROS overview.
- Why ROS?
- Other middlewares.
- ROS distributions.

Getting started.
- Installing ROS & Turtlebot simulation.
- Tutorials.

Concepts:
- Directories & variables.
- Running programs.
- ROS build system.
- ROS nodes.

Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule

ROS Overview

- ROS = "*Robot Operating System*"
  - Not an operating system!
- Contains:
  - Middleware & tools.
  - Build/packaging system.
  - Core packages.
    - E.g. geometry tools.
  - Peripheral packages.
    - E.g. mapping.

Why ROS?

- Open source.
- Big ecosystem.
  - Many robots.
  - Many users.
  - Many tools.
- Common environment.
- Separation of concerns.
- Willow garage / OSRF.

Willow Garage

Open Source Robotics Foundation

ROS Distributions

- Built on Ubuntu.
- Distributions rolled out approx yearly:
  - Kinetic May 23rd, 2016.

    → (L/K/X)**Ubuntu** 15.10 or **16.04**
  - Jade May 23rd, 2015.
  - Indigo July 22nd, 2014.

BLG456E
Robotics
ROS Intro

Lecture Contents

ROS overview.
- Why ROS?
- Other middlewares.
- ROS distributions.

Getting started.
- Installing ROS & Turtlebot simulation.
- Tutorials.

Concepts:
- Directories & variables.
- Running programs.
- ROS build system.
- ROS nodes.

Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule

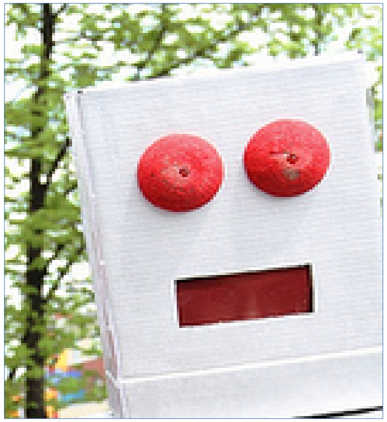
Installing ROS & Turtlebot

- Step 1: Install Ubuntu.
  - Option 1: Hard disk install.
  - Option 2: Virtual machine install (slow).
  - Option 3: External hard disk install.
- Step 2: Install ROS & Turtlebot.
  - Run shell script **install_456_students.sh** from https://bitbucket.org/damienjadeduff/456_kinetic_turtlebot/src

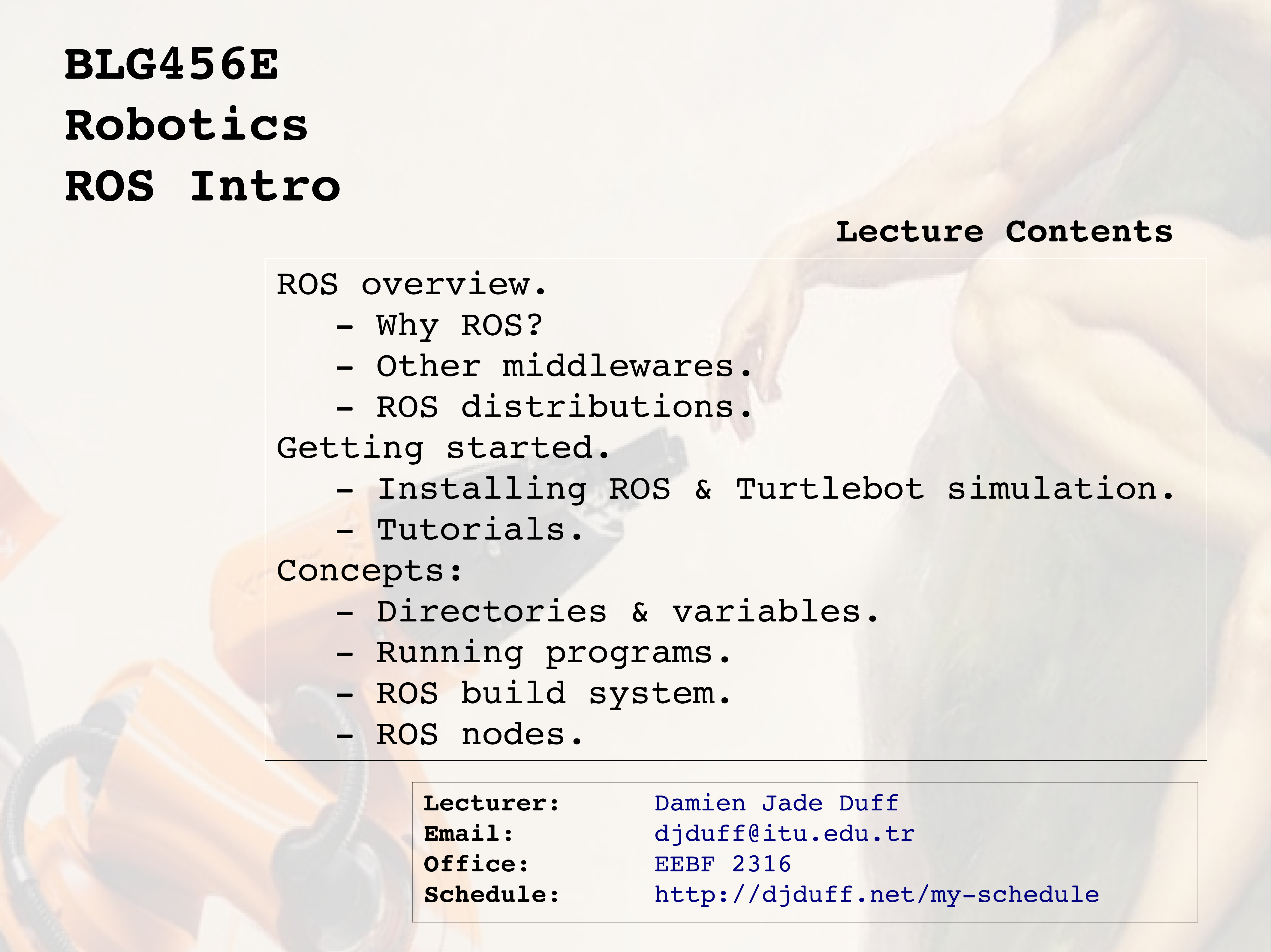
Detailed instructions can be found on Ninova under Course Files / Ders Kaynaklar

How to use ROS

- Follow the tutorials to learn the basics.
  - Choose Kinetic/Catkin tutorials.



- Suggested tutorials:
  - Installing & Configuring your ROS Environment.
  - Navigating the ROS Filesystem.
  - Creating a ROS Package.
  - Building a ROS Package.
  - Understanding ROS Nodes.
  - Understanding ROS Topics.
  - Writing a Simple Publisher & Subscriber (C++).
  - Examining the Simple Publisher & Subscriber.
  - Using rqt_console & roslaunch.



BLG456E
Robotics
ROS Intro

Lecture Contents

ROS overview.
- Why ROS?
- Other middlewares.
- ROS distributions.

Getting started.
- Installing ROS & Turtlebot simulation.
- Tutorials.

Concepts:
- Directories & variables.
- Running programs.
- ROS build system.
- ROS nodes.

Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule

Directories & variables

- ROS Kinetic installed in
  `/opt/ros/kinetic`
- To make use of it:

  ```
  source /opt/ros/kinetic/setup.bash
  ```

  - Sets up environment variables.

Directories & variables

- Your code will be in

```
~/catkin_ws/
```

- To intialise it:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
source /opt/ros/kinetic/setup.bash
catkin_init_workspace
cd ~/catkin_ws
catkin_make
```

- To make use of it:

```
source ~/catkin_ws/devel/setup.bash
```

ROS concepts:

ROS build-system

- Functionality comes in “packages”.
- **Your** programs will be built as packages.
- Toolchains for building:
  - **rosbuild** – older, deprecated.
  - **catkin** – newer, cmake-based ←

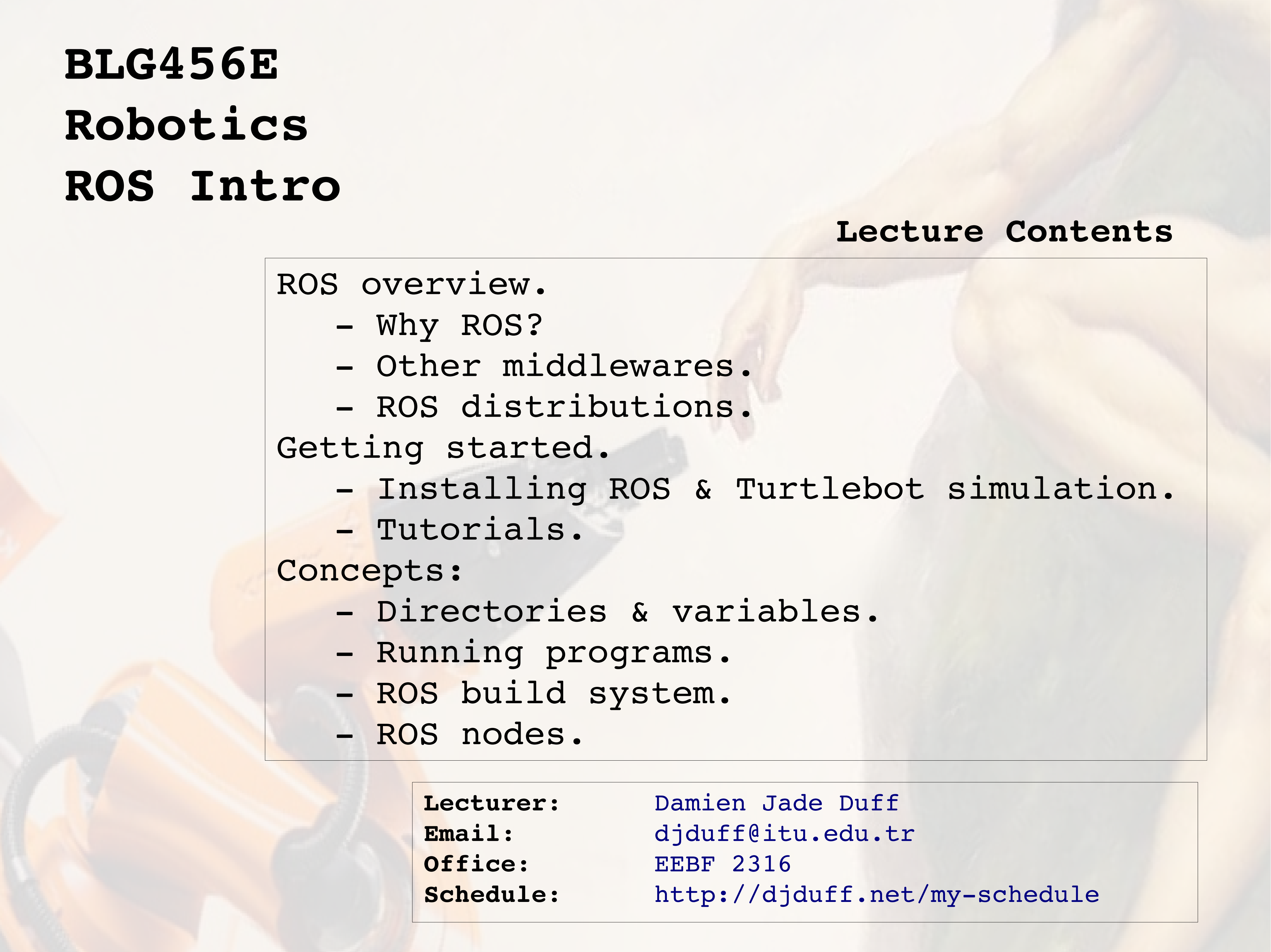
Running programs

- Run (launch) a bundle of programs:

```
roslaunch turtlebot_gazebo turtlebot_world.launch
```

- Run a single executable:

```
roscore # coordinate node communication
# and in a different terminal window:
rosrun rviz rviz
```



BLG456E
Robotics
ROS Intro

Lecture Contents

ROS overview.
- Why ROS?
- Other middlewares.
- ROS distributions.

Getting started.
- Installing ROS & Turtlebot simulation.
- Tutorials.

Concepts:
- Directories & variables.
- Running programs.
- ROS build system.
- ROS nodes.

Lecturer: Damien Jade Duff
Email: djduff@itu.edu.tr
Office: EEBF 2316
Schedule: http://djduff.net/my-schedule

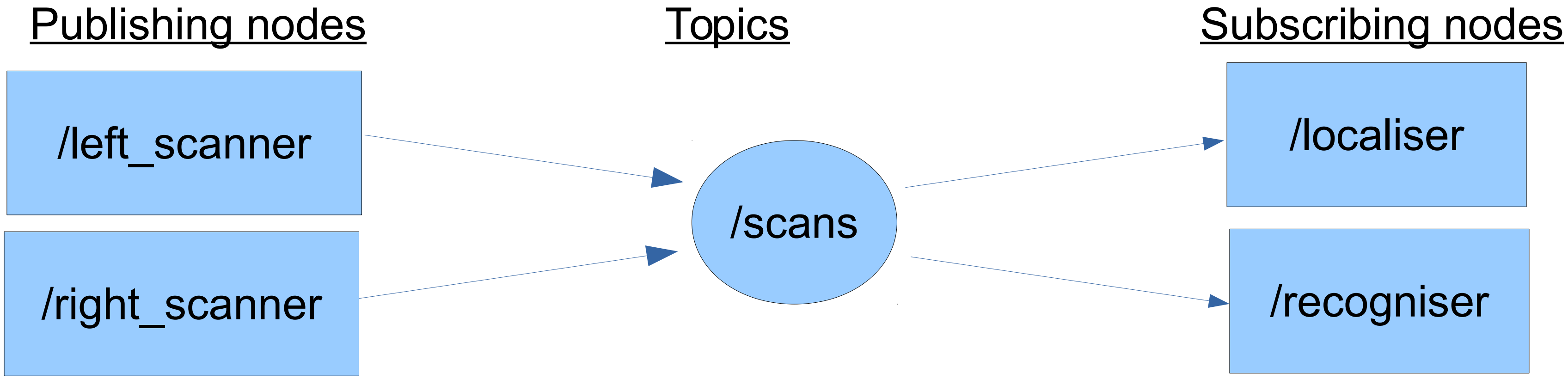
ROS concepts:

A ROS program consists of communicating nodes.

- ROS uses a publish-subscribe model.
- Programs construct nodes.
- Publishing nodes send messages to a topic.
- Subscribing nodes take messages from a topic.

ROS Node graph

Conceptual node graph



Publishing nodes

Topics

Subscribing nodes

/left_scanner

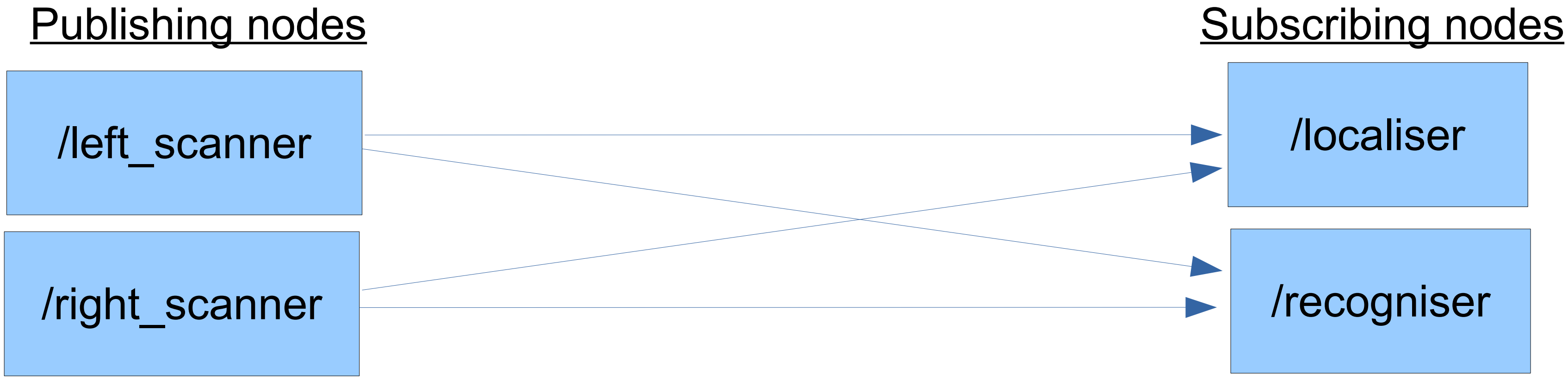
/right_scanner

/scans

/localiser

/recogniser

Graph of message routes



Publishing nodes

Subscribing nodes

/left_scanner

/right_scanner

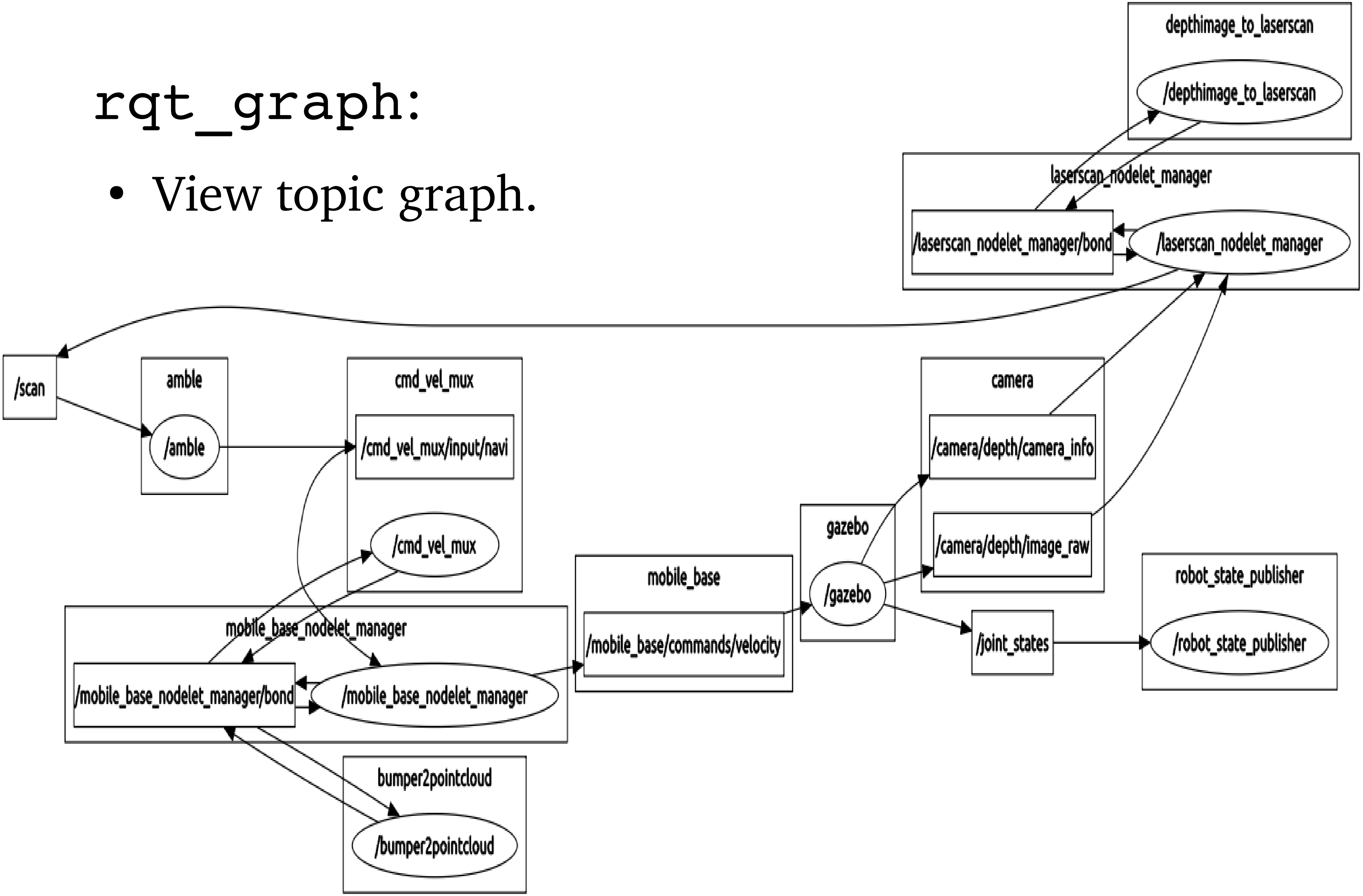
/localiser

/recogniser

More ROS Tools

rqt_graph:

- View topic graph.



ROS concepts:

A ROS program is a set of communicating nodes.

When nodes connect:

- ROS master program TCP/IP address is an environment variable.
- Publishing & subscribing nodes contact ROS master over TCP/IP.
- Master coordinates communication over topics.