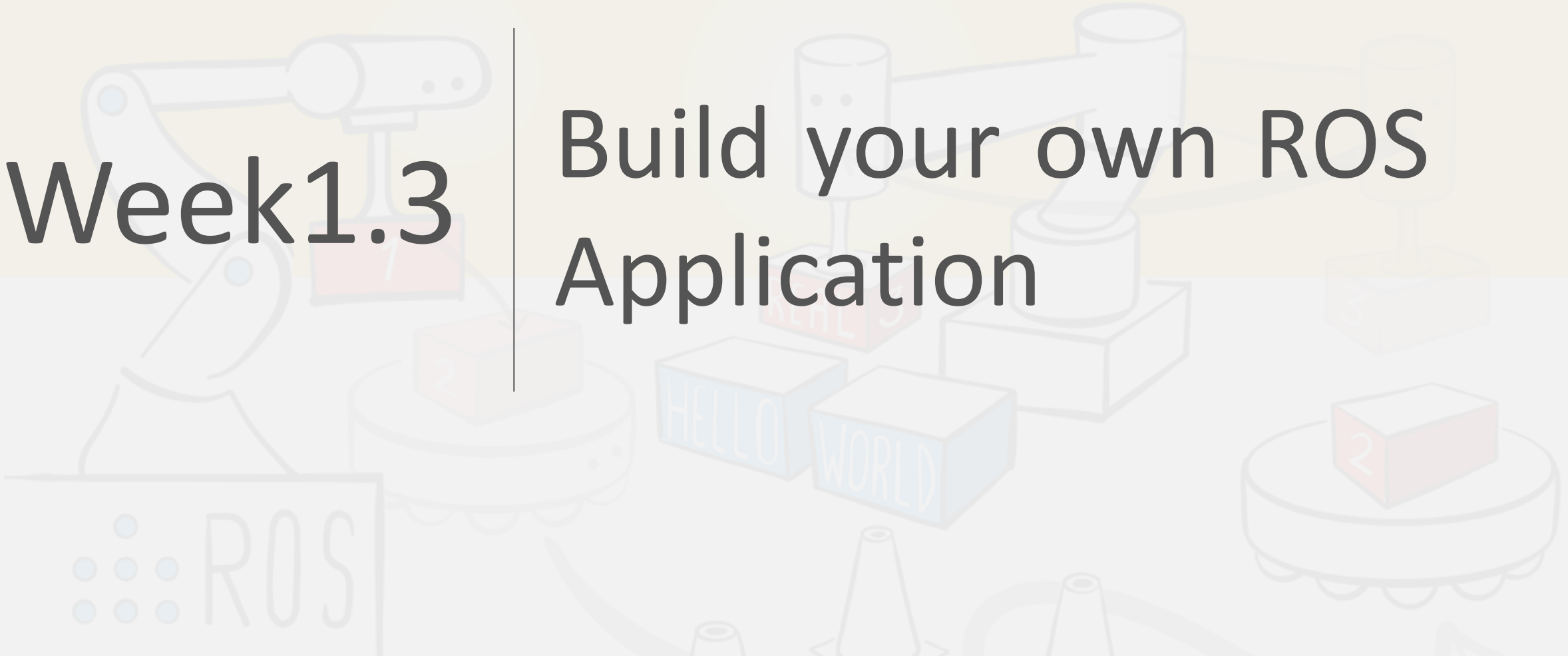
Week1.3

Build your own ROS Application



Build your own ROS application

- We learned how to find our way around, “given” an already running ROS application.
- We will now learn about the ingredients required to build our own ROS application.
- Write different types of ROS nodes – Publishers, Subscribers, Services and Actions.

Remember?

The names publisher and subscriber do ring a bell!

```
donnie@tudelft:~$ rostopic list
/rosout
/rosout_agg
/topic_1
donnie@tudelft:~$ rostopic info /topic_1
Type: std_msgs/String

Publishers:
 * /node_1 (http://tudelft:35439/)

Subscribers:
 * /node_2 (http://tudelft:34711/)


donnie@tudelft:~$
```

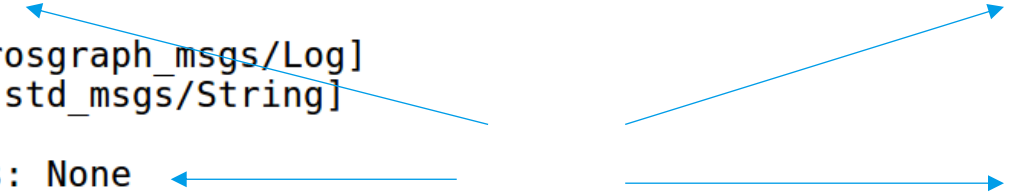
Remember?

```
donnie@tudelft:~$ rosnode info /node_1
--------------------------------------------------------------------------------
Node [/node_1]
Publications:
 * /rosout [rosgraph_msgs/Log]
 * /topic_1 [std_msgs/String]

Subscriptions: None

Services:
 * /node_1/get_loggers
 * /node_1/set_logger_level


contacting node http://tudelft:35439/ ...
Pid: 3874
Connections:
 * topic: /topic_1
    * to: /node_2
    * direction: outbound
    * transport: TCPROS
 * topic: /rosout
    * to: /rosout
    * direction: outbound
    * transport: TCPROS
```



```
donnie@tudelft:~$ rosnode info /node_2
--------------------------------------------------------------------------------
Node [/node_2]
Publications:
 * /rosout [rosgraph_msgs/Log]

Subscriptions:
 * /topic_1 [std_msgs/String]

Services:
 * /node_2/get_loggers
 * /node_2/set_logger_level


contacting node http://tudelft:34711/ ...
Pid: 3922
Connections:
 * topic: /rosout
    * to: /rosout
    * direction: outbound
    * transport: TCPROS
 * topic: /topic_1
    * to: /node_1 (http://tudelft:35439/)
    * direction: inbound
    * transport: TCPROS
```