

# A HYBRID SIMULATION FOR PEPPER

---

*Softbank's Pepper Robot and RoboCup@Home*

*Florian Lier | LAAS, Toulouse, 2018*

# WHAT WE ARE DOING

---

*Short intro to RoboCup  
and our Pepper*



# ROBOCUP // SPEECH RECOGNITION & SYNTHESIS

[http://www.ais.uni-bonn.de/robocup.de/images/RC12/RC12\\_Home\\_Object\\_Recognition.jpg](http://www.ais.uni-bonn.de/robocup.de/images/RC12/RC12_Home_Object_Recognition.jpg) & <https://www.youtube.com/watch?v=YpjeNa8BAYg>

.....



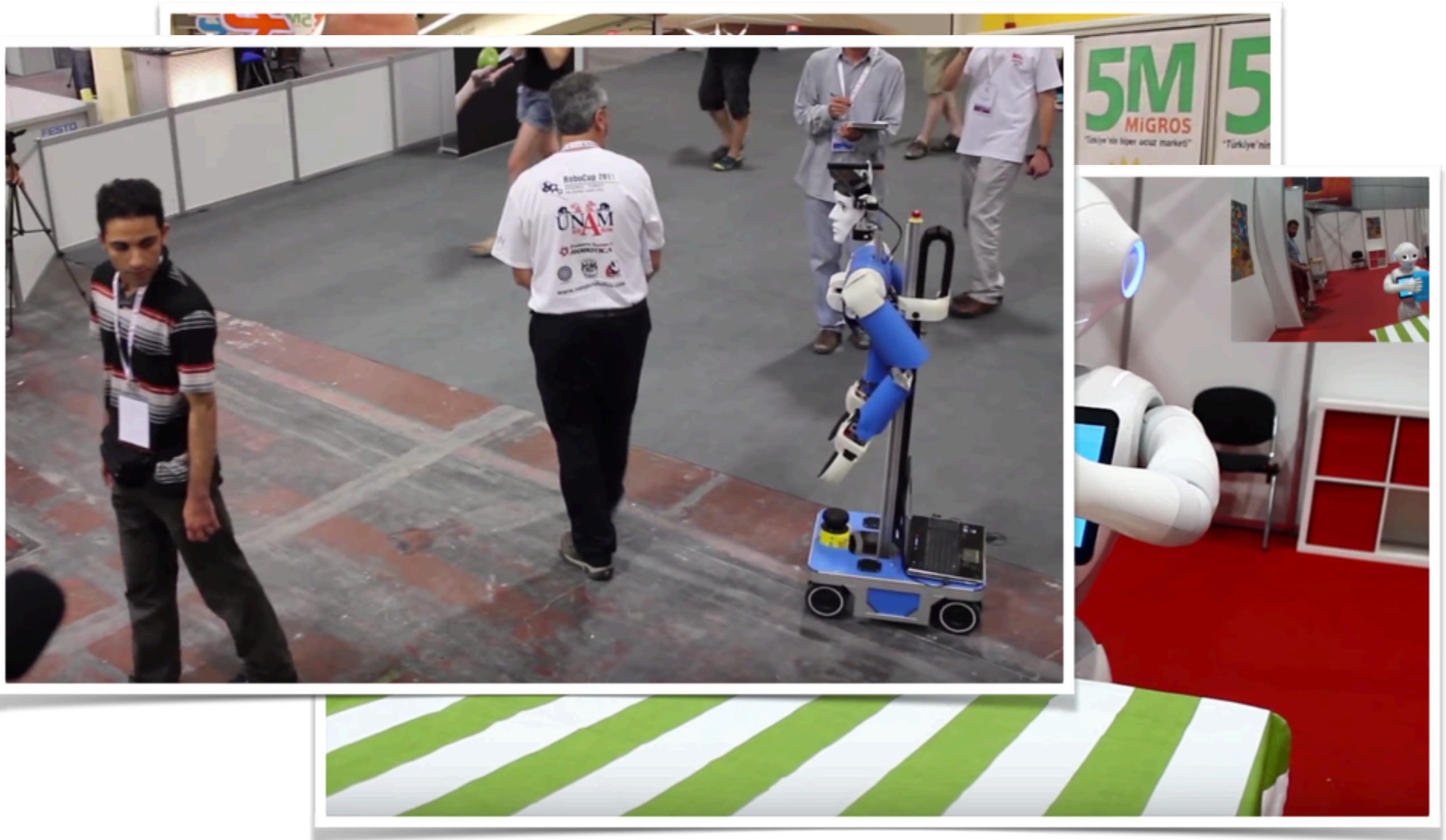
# ROBOCUP // OBJECT MANIPULATION

---



# ROBOCUP // AUTONOMOUS NAVIGATION & FOLLOWING

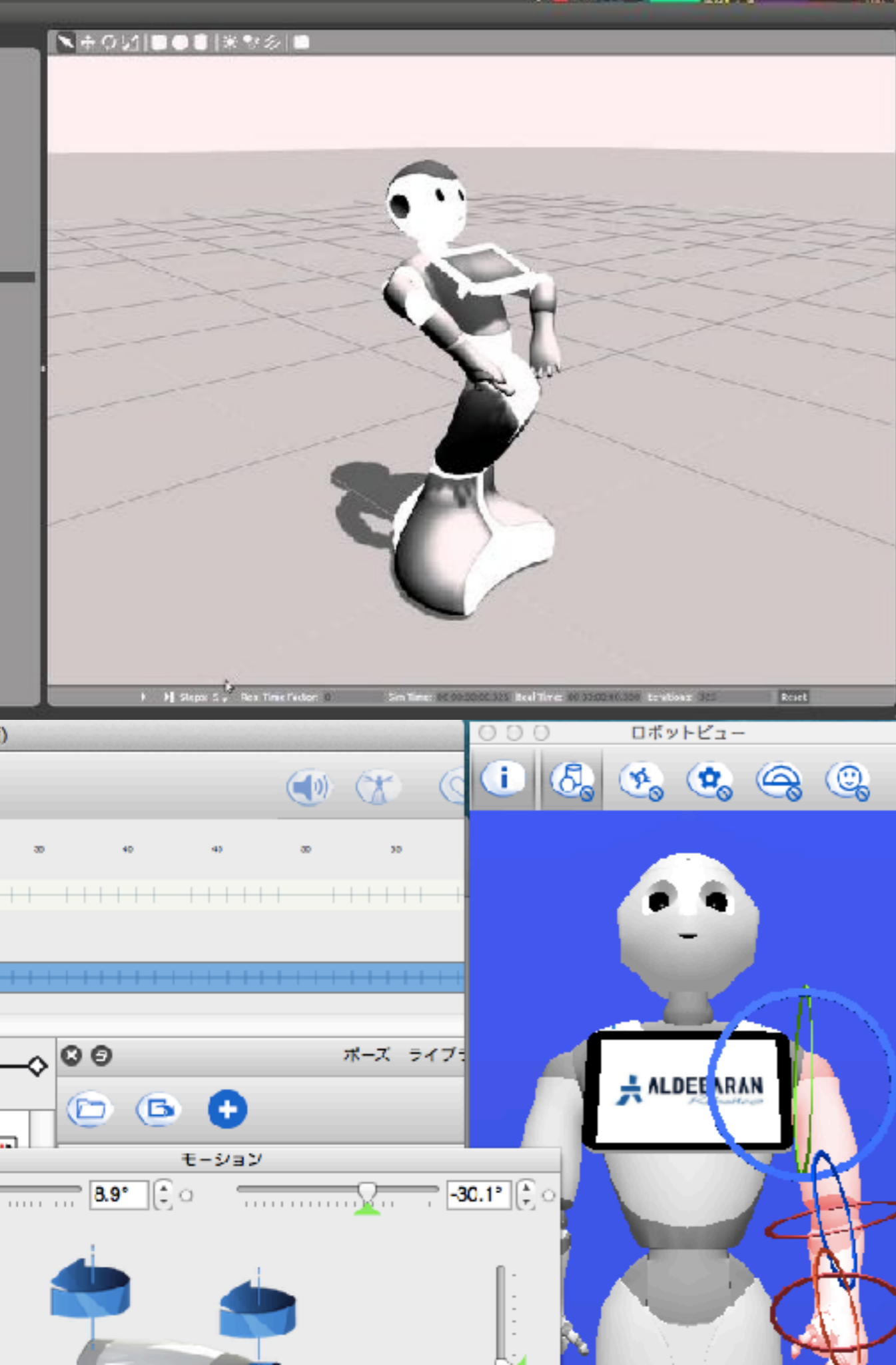
---



# ROBOCUP // OBJECT DETECTION & CLASSIFICATION

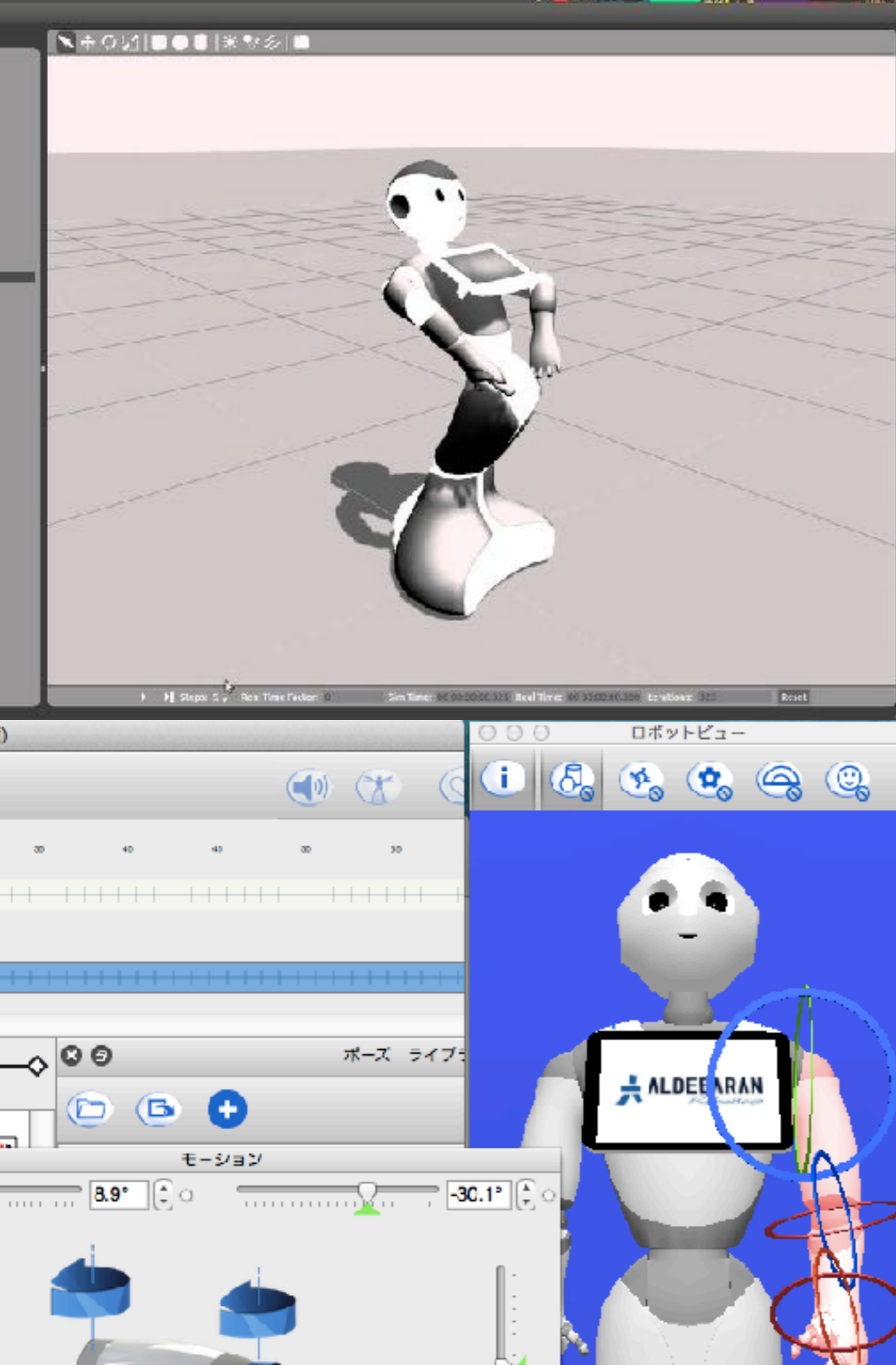
---





# PROBLEM STATEMENT

- No existing Pepper Simulation incorporating all of the following features:
- HRI capabilities
- ROS Interface
- ‘Programmable’ Scenarios/Setups
- Easy environment creation and integration wrt graphic modeling
- Simulated Sensors, e.g., Laser, RGB Cameras, Depth Sensors
- OpenSource (non-commercial)



# PROBLEM STATEMENT

- No existing Pepper Simulation incorporating all of the following features:
- HRI capabilities
- ROS Interface
- ‘Programmable’ Scenarios/Setups
- Easy environment creation and integration wrt graphic modeling
- Simulated Sensors, e.g., Laser, RGB Cameras, Depth Sensors
- OpenSource (non-commercial)

 **BUT MORSE!**



# OUR PEPPER

<https://www.youtube.com/watch?v=70YMCihD1ds&feature=youtu.be>

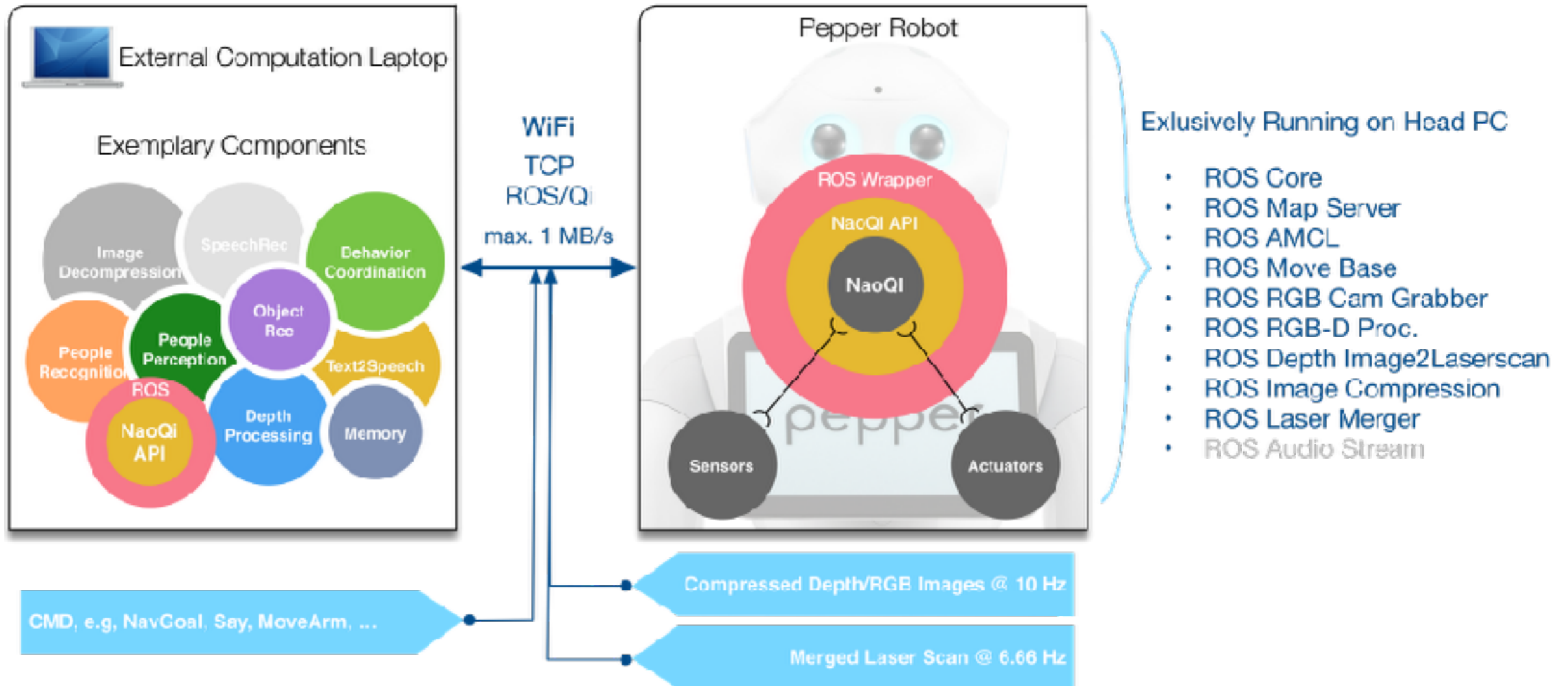
---



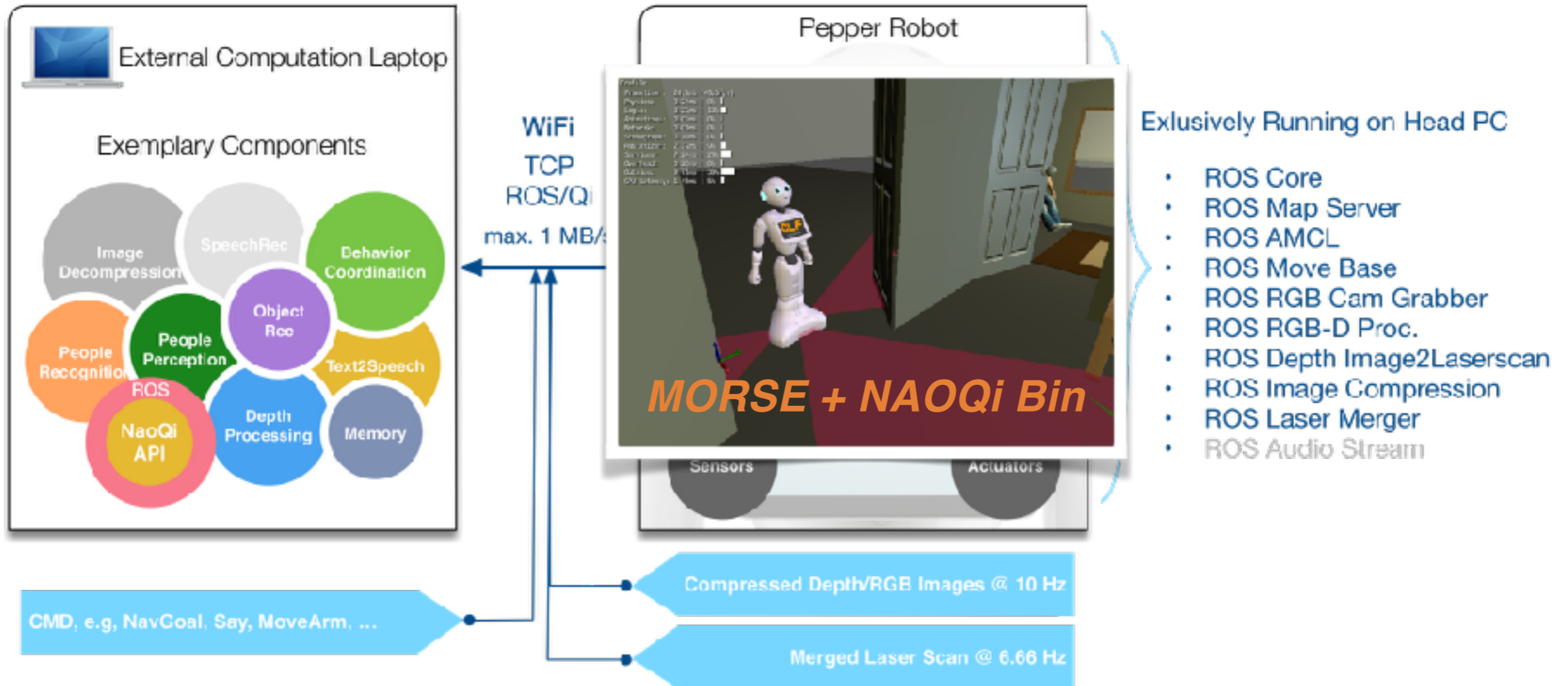
“

We need a Simulator that enables us to use our software stack without any modification and which enables us to test RoboCup related tasks.

# OUR SYSTEM PHYSICAL 'ARCHITECTURE'



# OUR SYSTEM SIMULATION 'ARCHITECTURE'



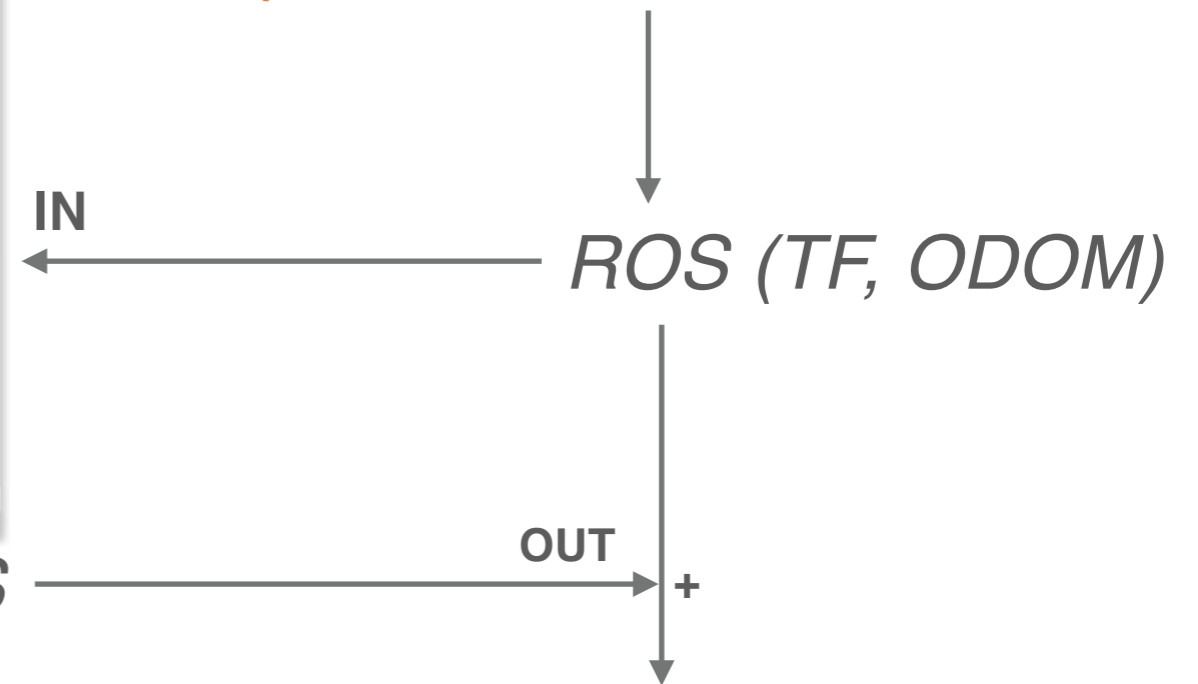
# OUR SYSTEM SIMULATION HYBRID 'ARCHITECTURE'

## MORSE



Exposes: Laser, RGB, Depth → ROS

**NAOQi Bin**  
Exposes: JointStates and QiAPI



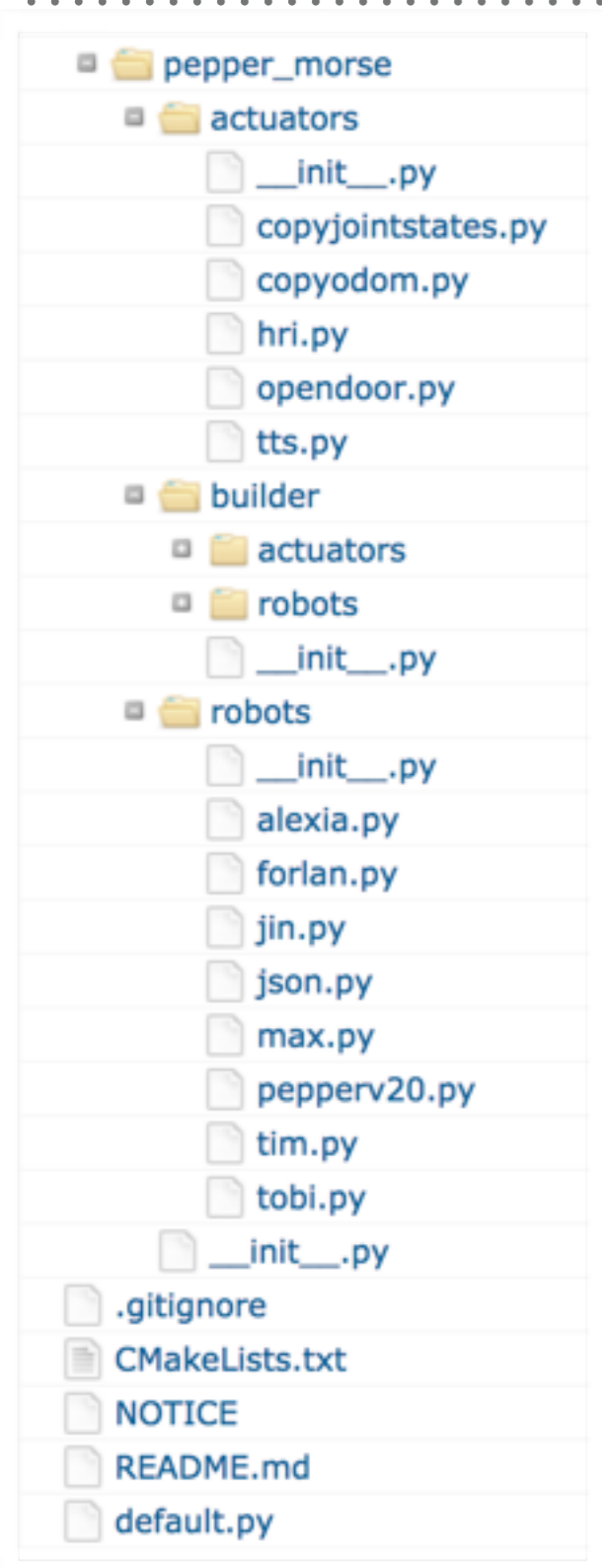
Same topics, e.g,  
*/pepper/laser/scan\_left*  
*/pepper/cmd\_vel*

LIVE DEMO

---

*YAY!*

# A FEW IMPLEMENTATION DETAILS



```
1  __author__ = 'Florian Lier [flieger@techfak.uni-bielefeld.de]'
```

```
2
```

```
3  import morse.core.robot
```

```
4  from morse.core import blenderapi
```

```
5  import logging; logger = logging.getLogger("morse." + __name__)
```

```
6
```

```
7
```

```
8  class Pepperv20(morse.core.robot.Robot):
```

```
9      """
```

```
10     Class definition for the pepperv20 robot.
```

```
11     """
```

```
12
```

```
13     _name = 'Pepper V20'
```

```
14
```

```
15     def __init__(self, obj, parent=None):
```

```
16         """ Constructor method
```

```
17         Receives the reference to the Blender object.
```

```
18         Optionally it gets the name of the object's parent,
```

```
19         but that information is not currently used for a robot.
```

```
20         """
```

```
21         logger.info('%s initialization' % obj.name)
```

```
22         morse.core.robot.Robot.__init__(self, obj, parent)
```

```
23         logger.info('Component initialized: %s' % obj.name)
```

```
24
```

```
25     def default_action(self):
```

```
26         """ Main loop of the robot
```

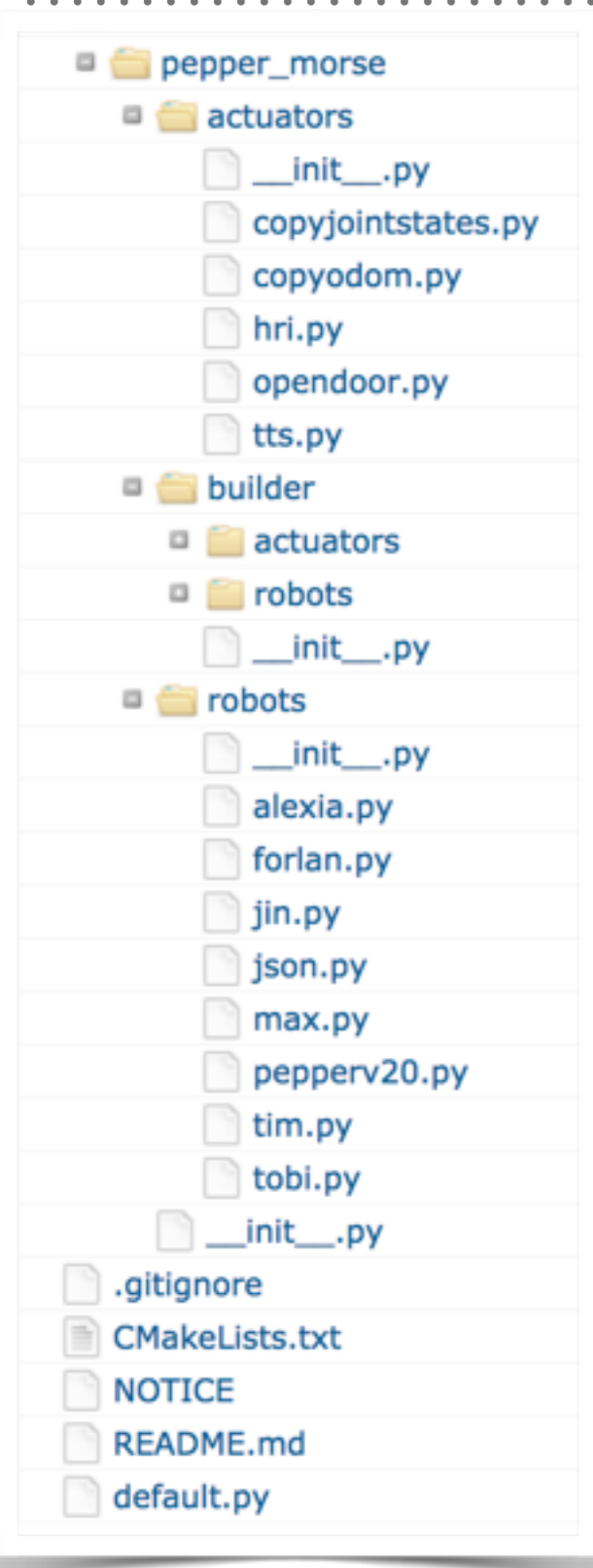
```
27         """
```

```
28         # This is usually not used (responsibility of the actuators
```

```
29         # and sensors). But you can add here robot-level actions.
```

```
30         pass
```

# A FEW IMPLEMENTATION DETAILS



```
14 # PEPPER MORSE
15 from pepper_morse.builder.robots import Pepperv20
16 from pepper_morse.builder.actuators.tts import TTS
17 from pepper_morse.builder.actuators.hri import HRI
18 from pepper_morse.builder.actuators.opendoor import OpenDoor
19 from pepper_morse.builder.actuators.copyjointstates import CopyJointStates
20 from pepper_morse.builder.robots import Json, Alexia, Forlan, Jin, Tim, Max
21
22 # Logging
23 logger = logging.getLogger("morserobots." + __name__)
24
25 FREQUENCY = 26.0
26 CAM_FREQUENCY = 10.0
27 LASER_FREQUENCY = 6.6
28
29 # Add a Clock for simulation time
30 clock = Clock()
31 clock.frequency(FREQUENCY)
32 clock.add_interface('ros', topic="/clock")
33
34 # Adding the Pepper Robot
35 pepper = Pepperv20()
36 pepper.properties(NoGravity=False, GroundRobot=True)
37 copy_ros_joints = CopyJointStates()
38 copy_ros_joints.properties(is_fake='no')
39 copy_ros_joints.frequency(FREQUENCY)
40 pepper.append(copy_ros_joints)
41 pepper.append(clock)
42
43 # Add a pose Sensor for Pepper
44 pose_pepper = Pose()
45 pepper.append(pose_pepper)
46 pose_pepper.add_interface('ros', topic="/pepper_robot/morse/pose")
47
```



**THAT'S IT!**

**MERCI BEAUCOUP!**