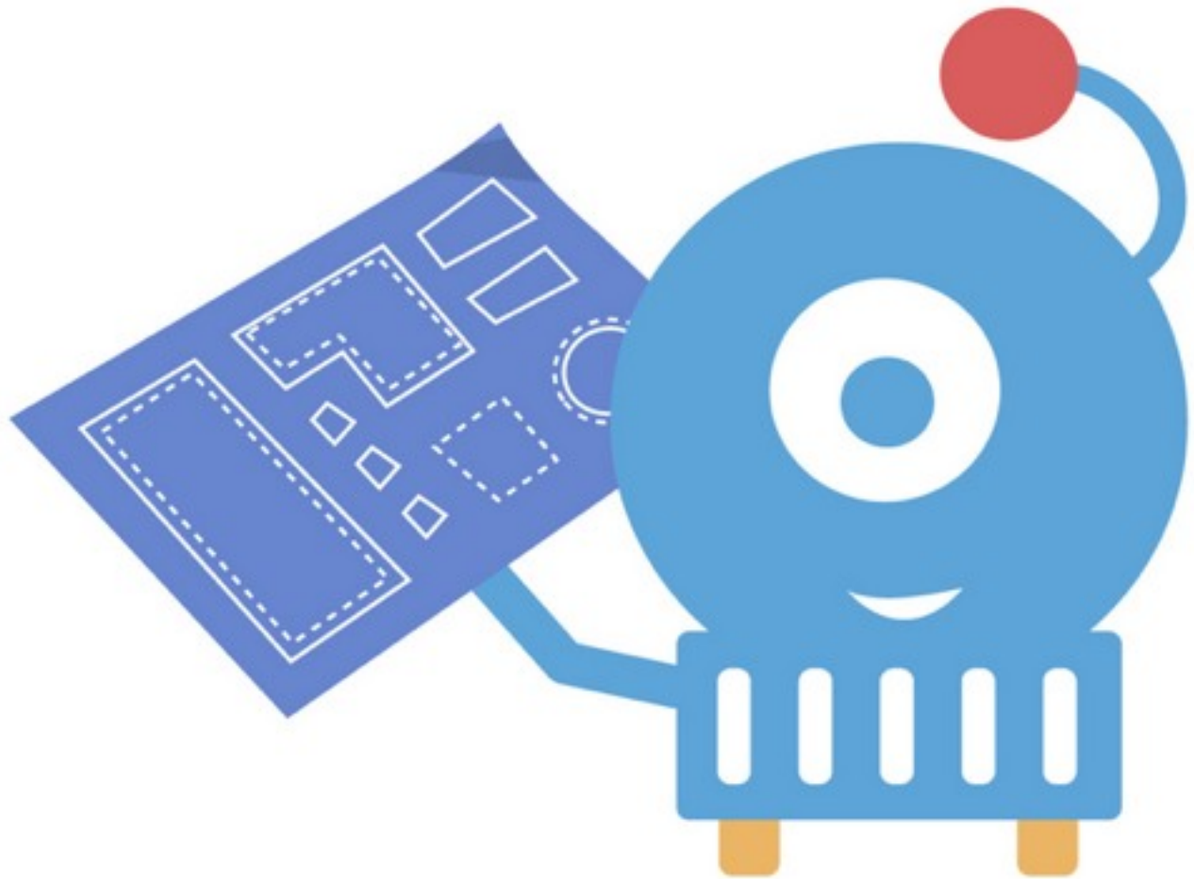

Google™
11
I/O

Cloud Robotics

Damon Kohler, Ryan Hickman (Google)
Ken Conley, Brian Gerkey (Willow Garage)
May 11, 2011

Overview

- Introduction (Ryan)
- ROS (Ken and Brian)
- ROS on Android (Damon)
- Closing Remarks (Ryan)
- Q&A



Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

Cloud-connected robots can...

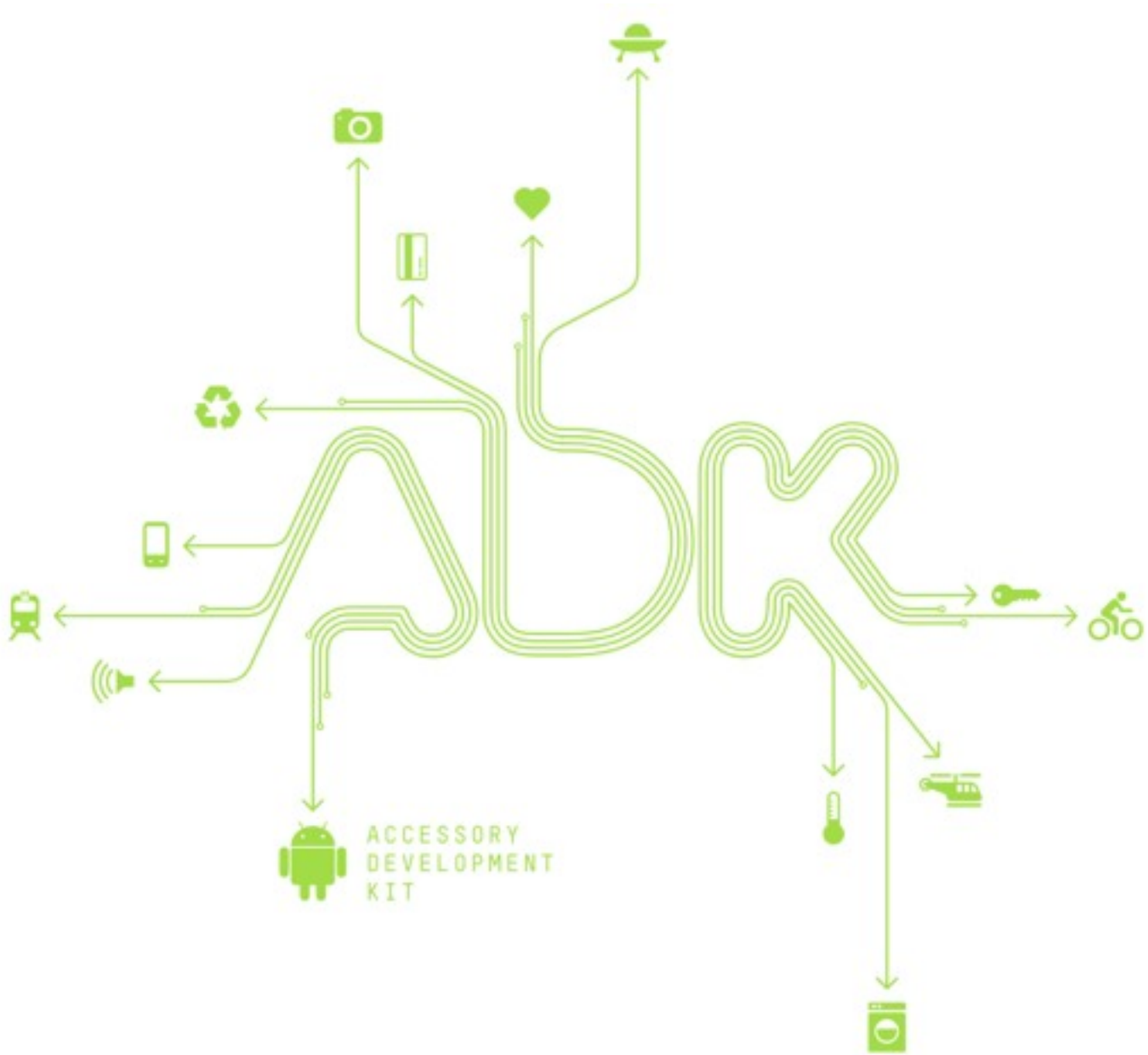
perceive, understand, share, and react.

Connect hardware to the cloud.

Feedback: http://goo.gl/tTQmW

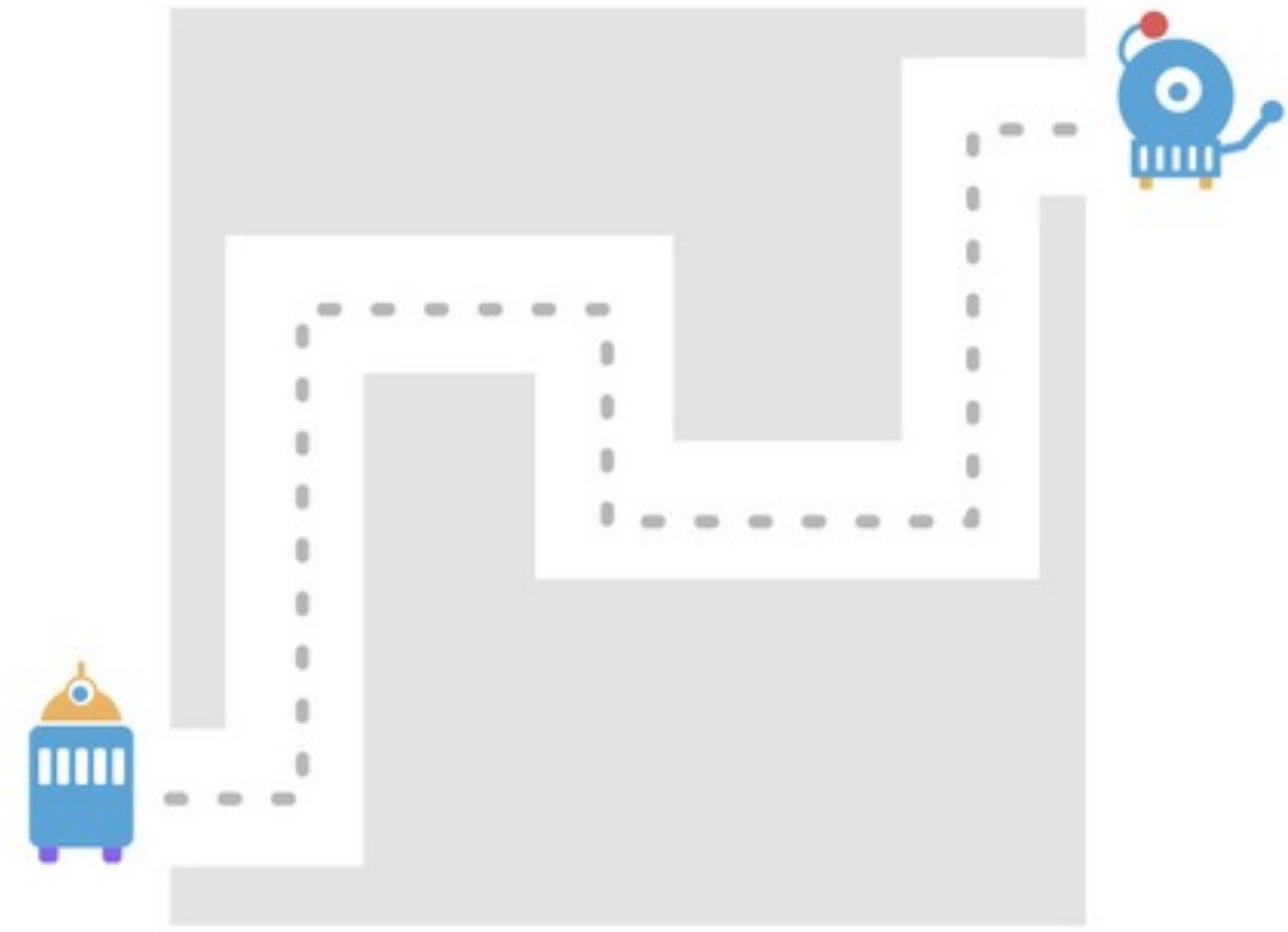
Hashtags: #io2011 #TechTalk

Android Open Accessory API



Object Recognition

Mapping and Navigation



Voice and Text Services

- Voice recognition
- Optical character recognition
- Language translation
- Text to speech

Google I/O 11

The cloud enables smarter robots.

- Off the shelf hardware means affordable robots
- Lower the barrier to entry for robotics
- Scalable CPU, memory, and storage

ROS

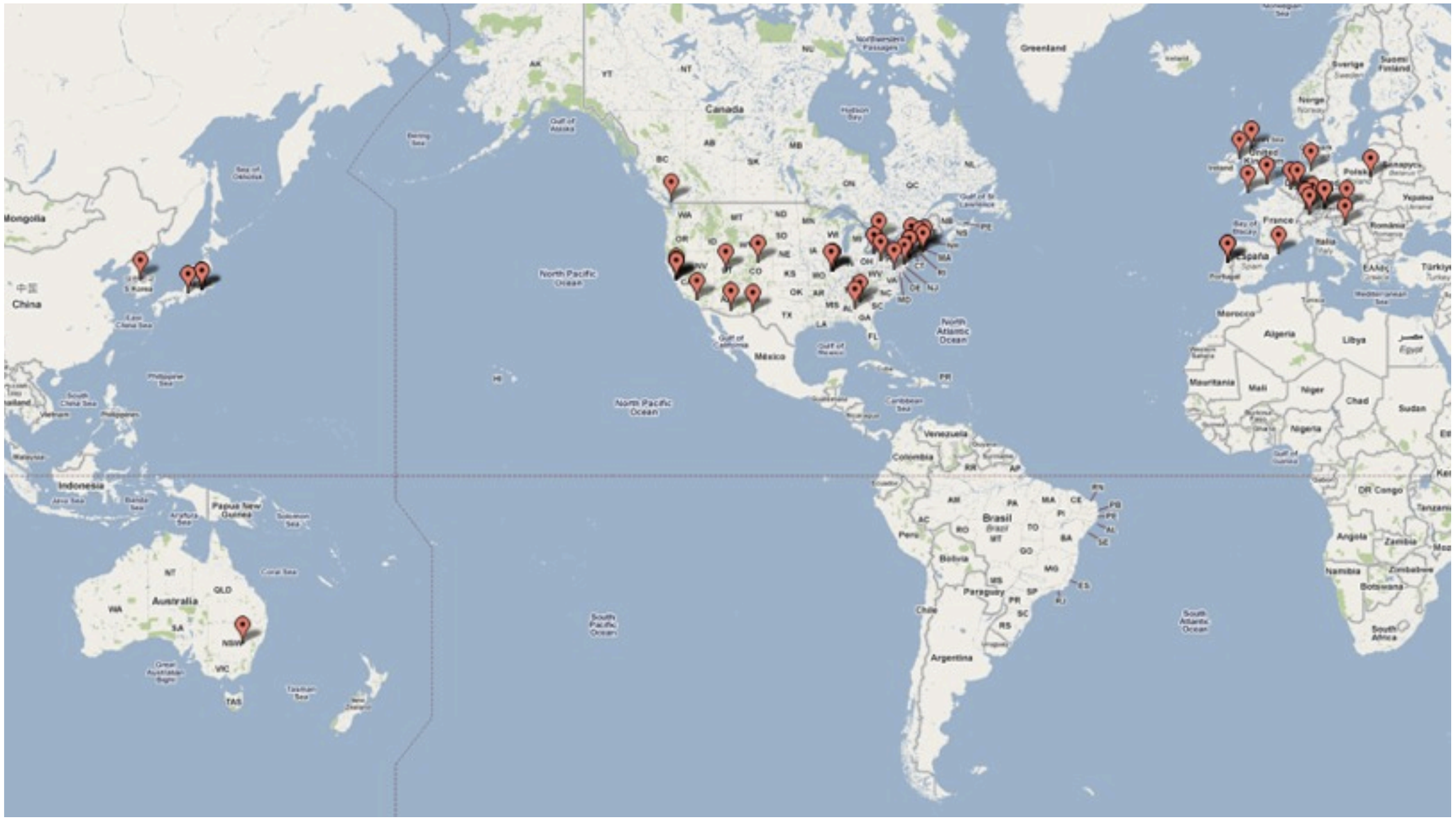
Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

Google I/O 11

ROS Code Repositories

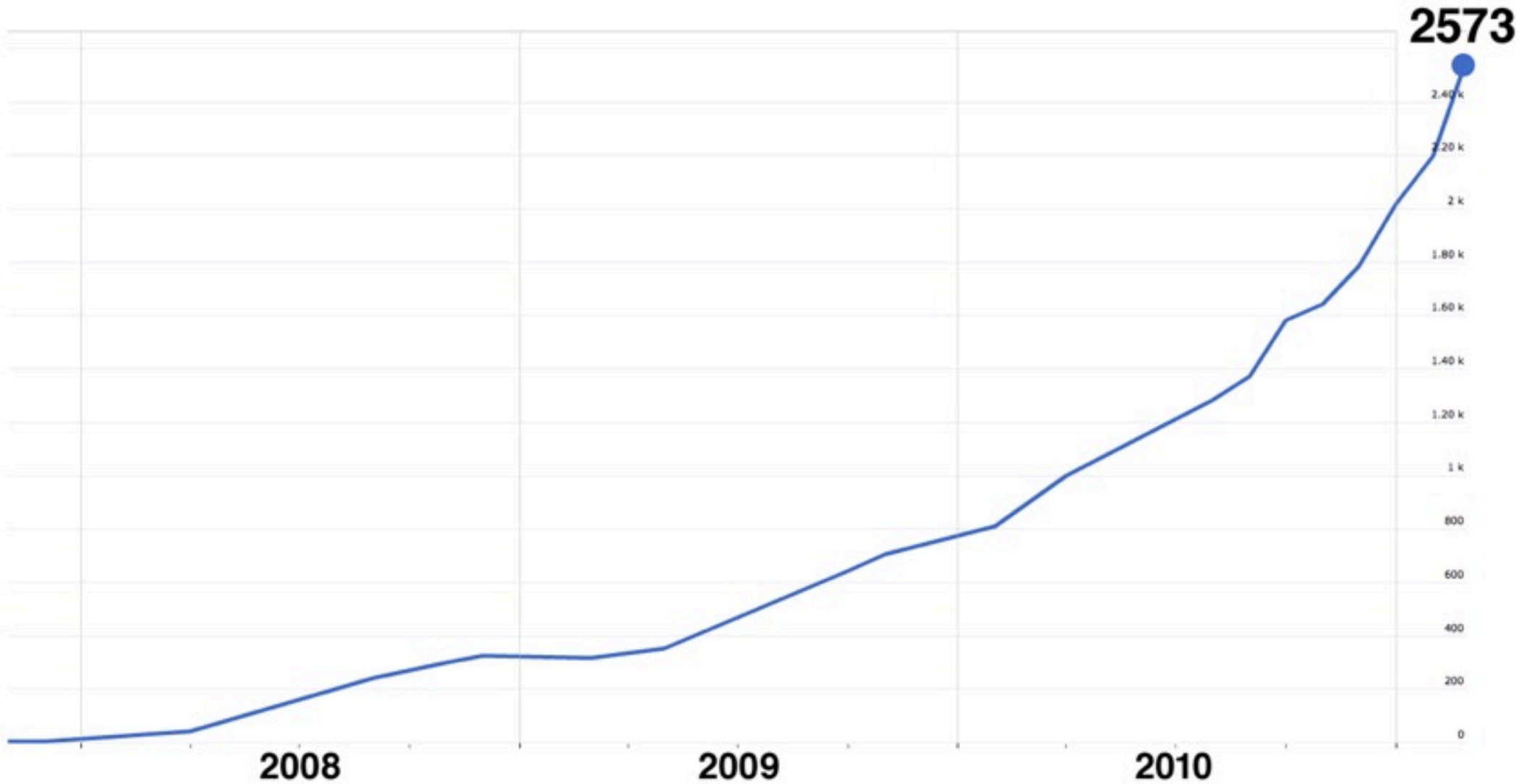
November 2007 - March 2011



Google I/O 11

ROS Code Packages

November 2007 - March 2011



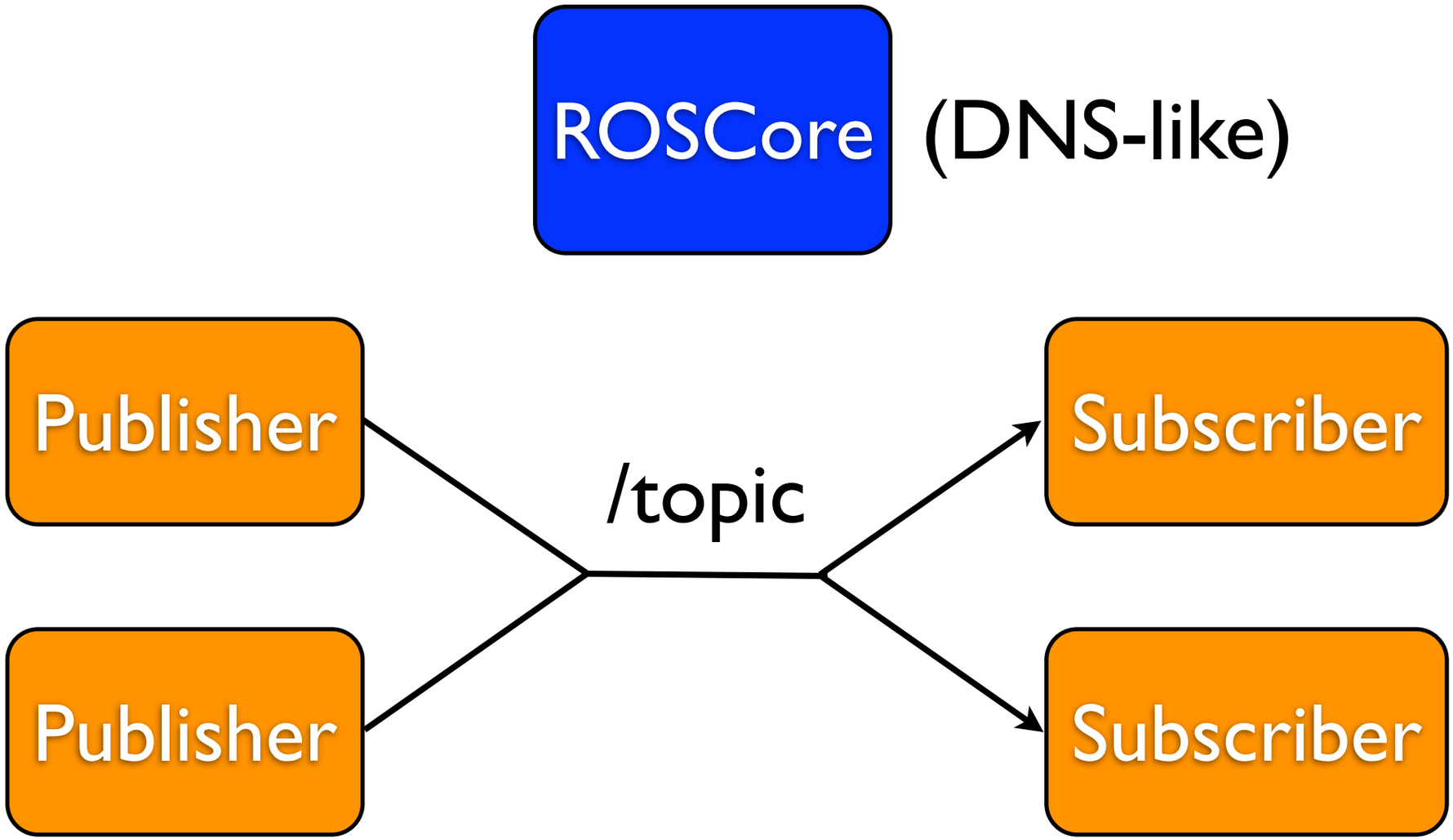
Google I/O 11

Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

Nodes

Anonymous Publish and Subscribe

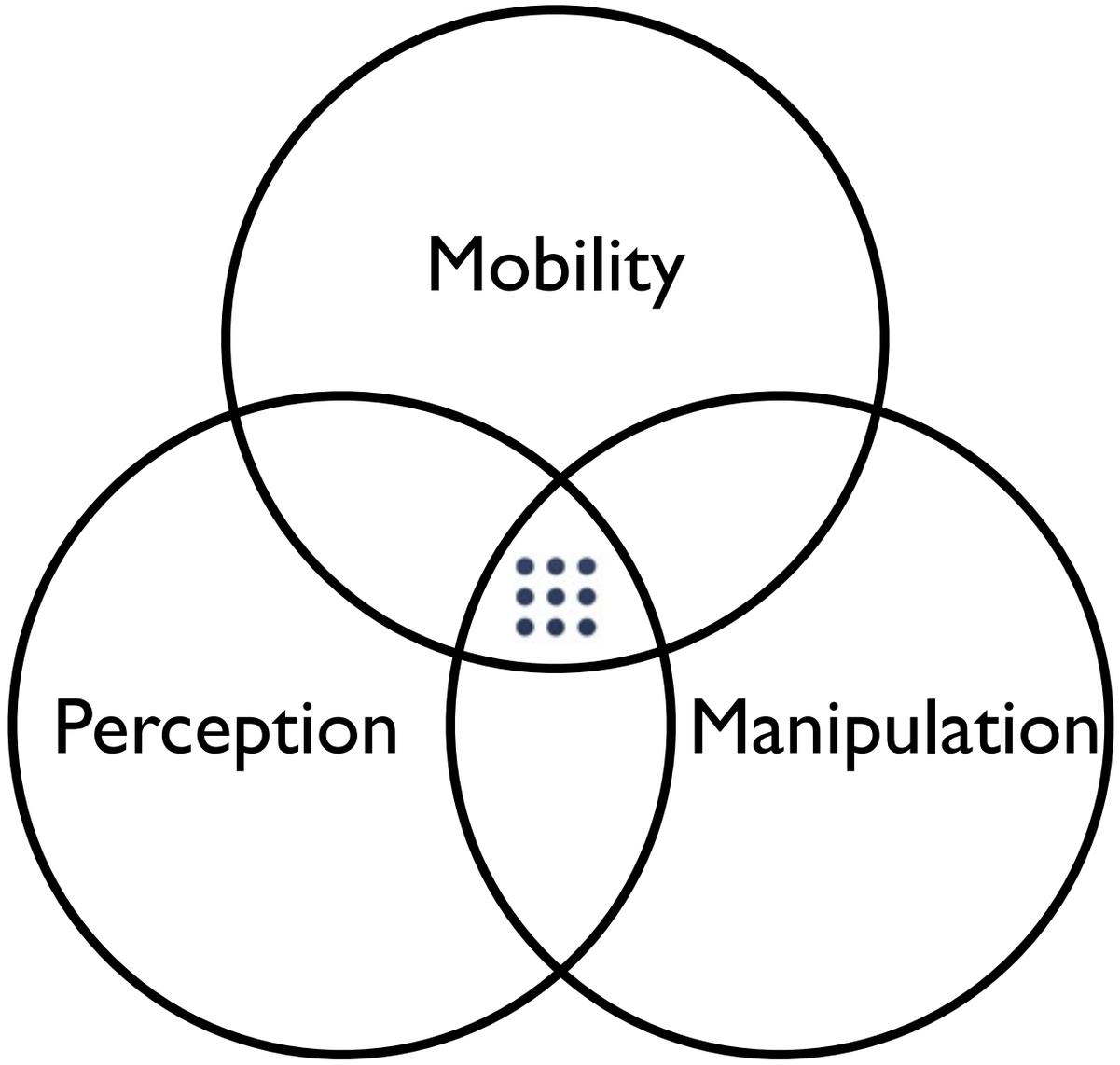


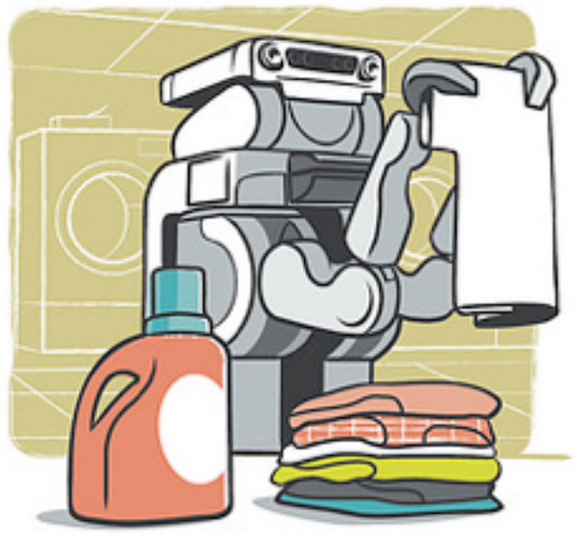
Google I/O 11

ROS Publish and Subscribe Example

```
$ rostopic pub /chatter std_msgs/String 'Hello World' -r 10
...
$ rostopic echo /chatter
data: Hello World
---
data: Hello World
---
data: Hello World
---
```

Functionality





PR2

Feedback: http://goo.gl/tTQmW

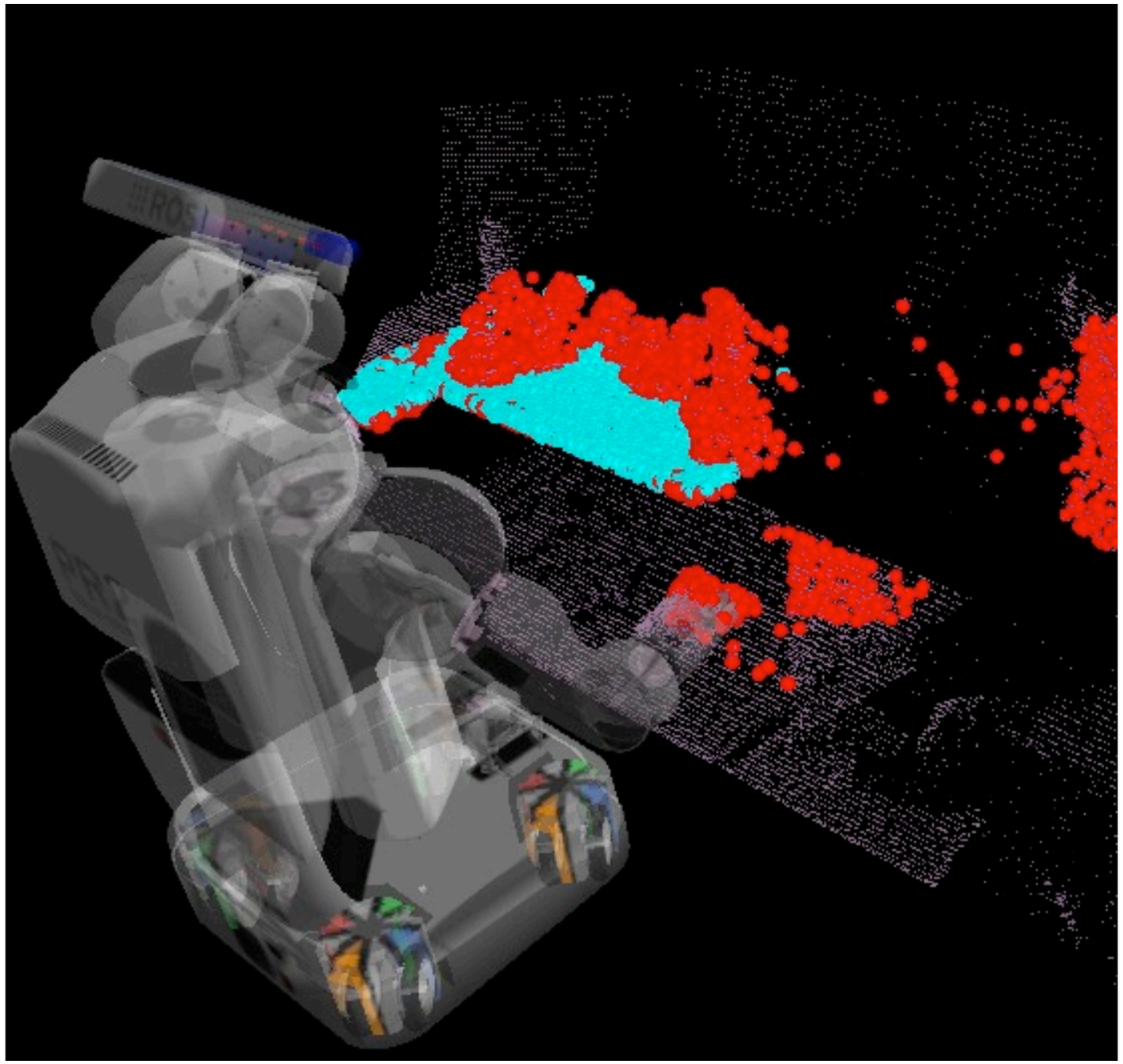
Hashtags: #io2011 #TechTalk

Specifications

- Costs $400,000
- 2x Servers (8 core i7)
- 2x 24GB RAM
- 7x Cameras (2x Stereo)
- 2x Laser rangefinders
- 2x 4 DOF Arm, 3 DOF Wrist
- Omnidirectional base

Robots in the Home

RViz



Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

The message is the medium.

Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

Head Tracking

```
# create goal message: 1 meter in front of base
target = PointStamped(x=1.0, frame_id='base_link')
goal = msg.pr2_controllers_msgs.PointHeadGoal(
                            target=target,
                            min_duration=Duration(1.0))

# publish
topics.head_traj_controller.point_head_action.goal(goal=goal)
```

Head Tracking

```
# create goal message: left gripper
target = PointStamped(frame_id='l_gripper_tool_frame')
goal = msg.pr2_controllers_msgs.PointHeadGoal(
                          target=target,
                          min_duration=Duration(0.2))
```

```
# publish loop
r = Rate(10)
while ok():
  topics.head_traj_controller.point_head_action.goal(goal=goal)
  r.sleep()
```

Hello World Redux

```
import math, time

# create pose message: 1 meter in front of shoulder
p = PoseStamped(frame_id='l_torso_lift_side_plate_link')
p.pose.orientation.y = -math.sqrt(2)/2.0
p.pose.orientation.w = math.sqrt(2)/2.0
p.pose.position.x = 0.5 # meters

# publish loop
r = Rate(10)
while ok():
  p.pose.position.y = math.sin(1.5*time.time())*0.3
  topics.r_cart.command_pose(p)
  r.sleep()
```

ROS +

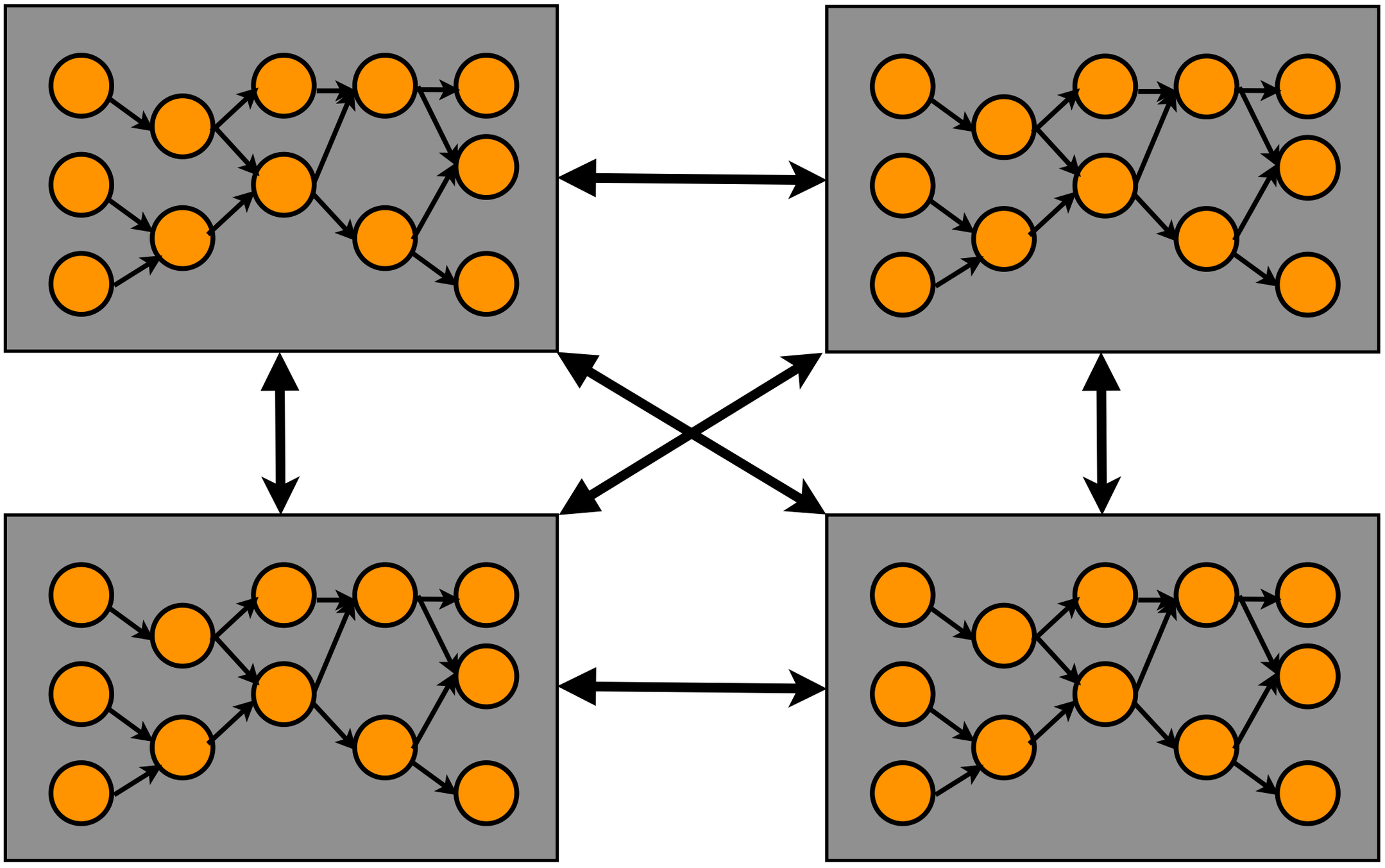
ROS in the Cloud

Feedback: http://goo.gl/tTQmW

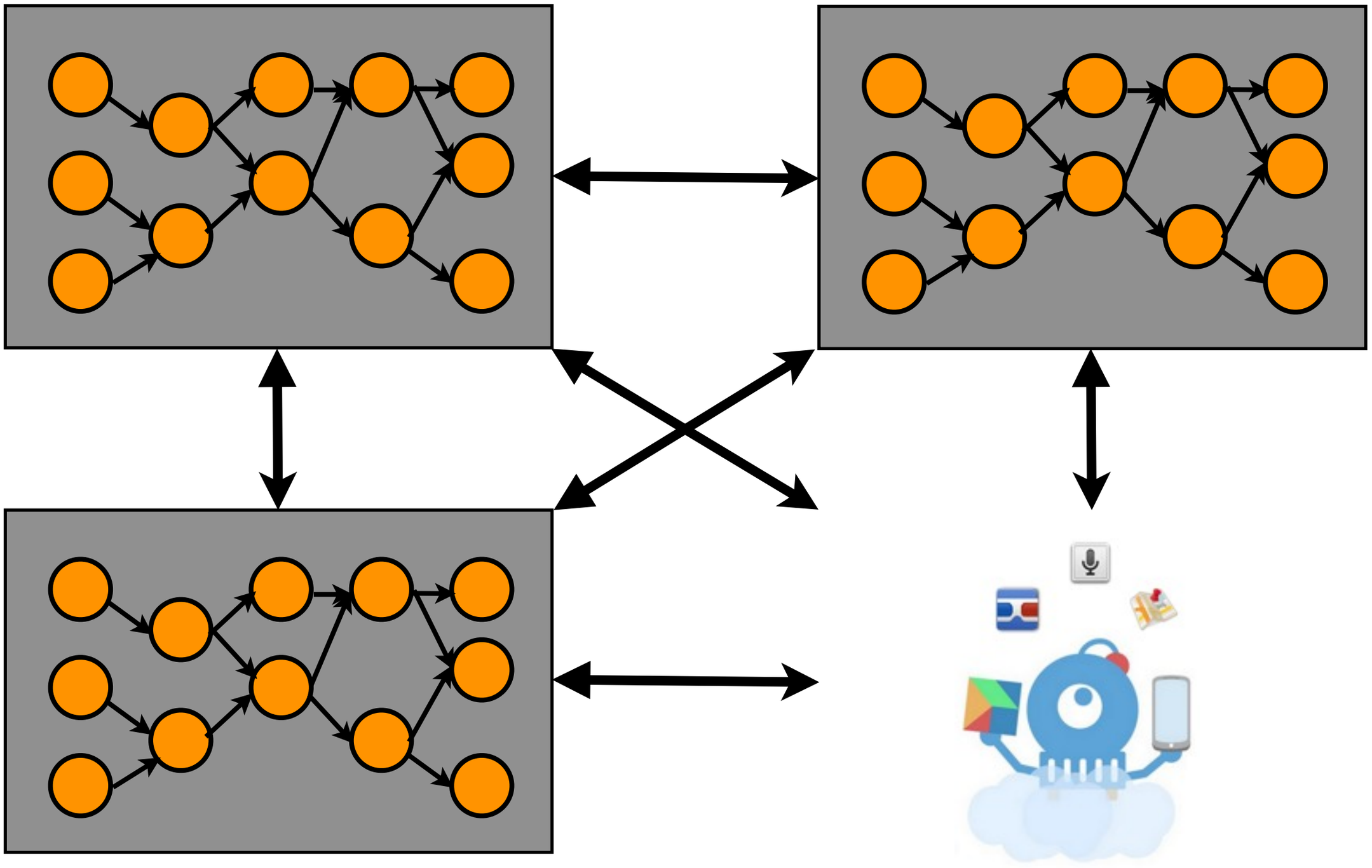
Hashtags: #io2011 #TechTalk

Google I/O 11

ROS in the Cloud



ROS in the Cloud



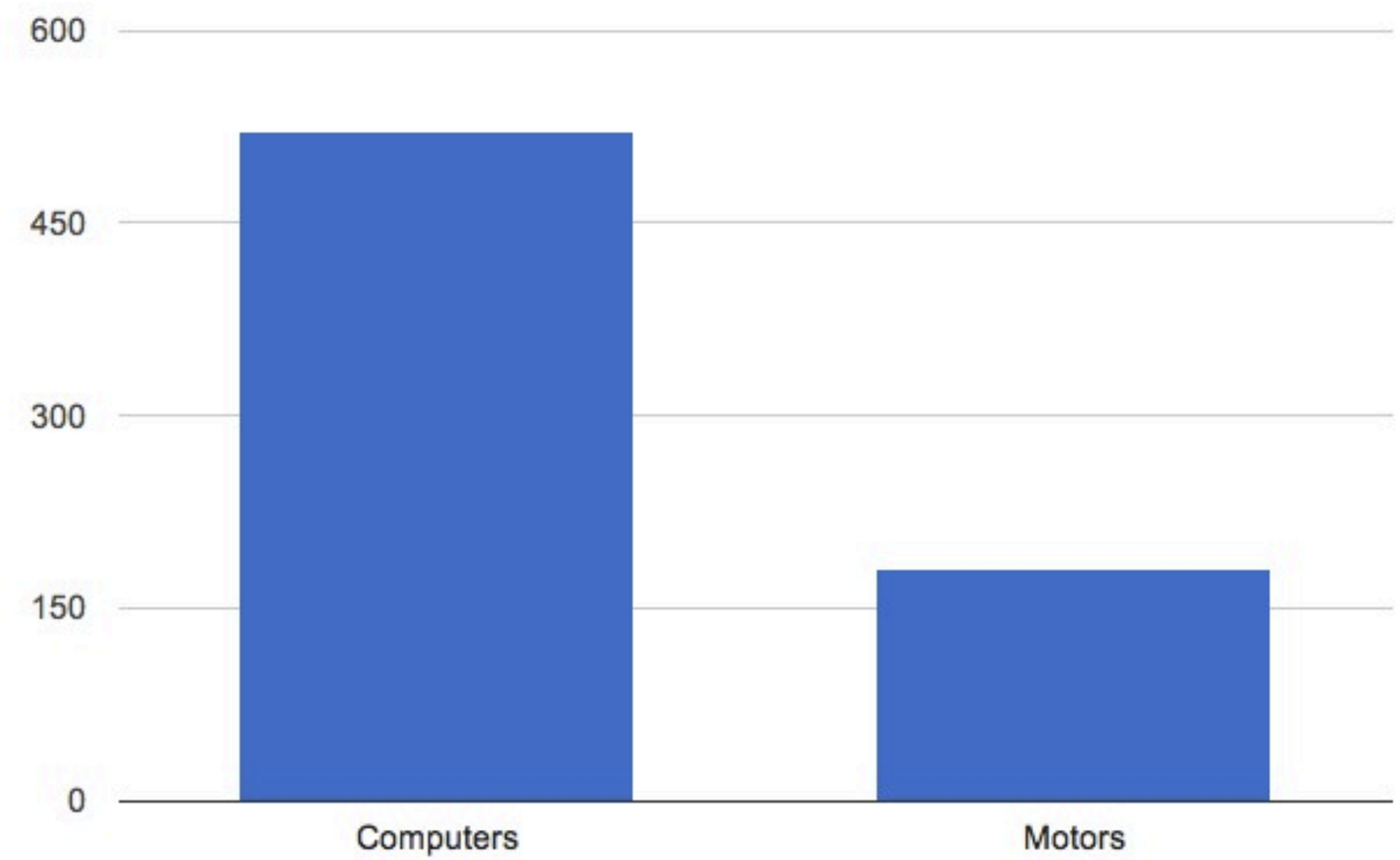
Personal robots need to be inexpensive.

Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

PR2 Power Consumption

Max Load (watts)



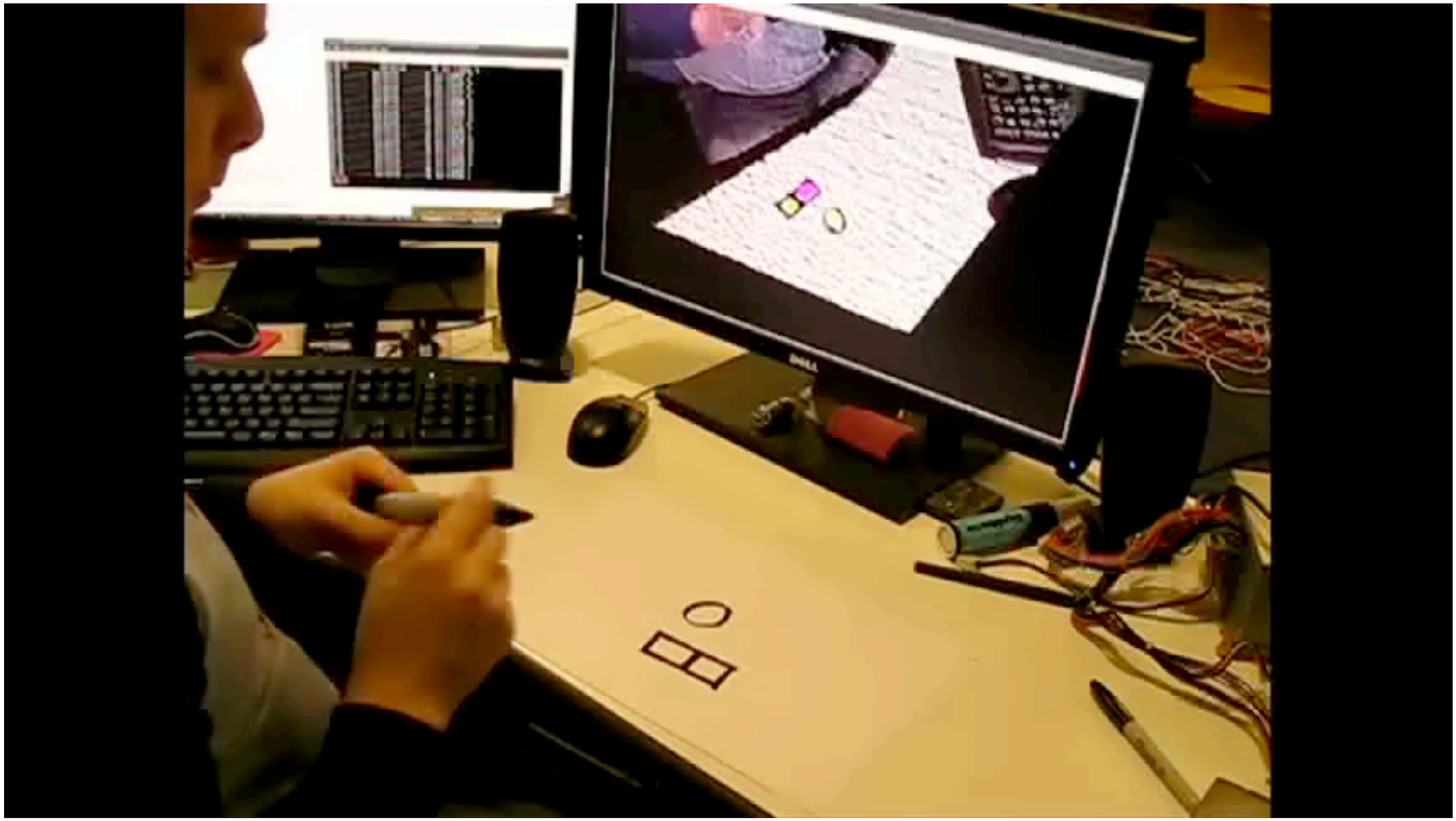
Google I/O 11

LIDAR: >$!000

(Robot not included)

ROS 3D Contest

ROS 3D Contest



Turtlebot Kit

Google StreetView Robot

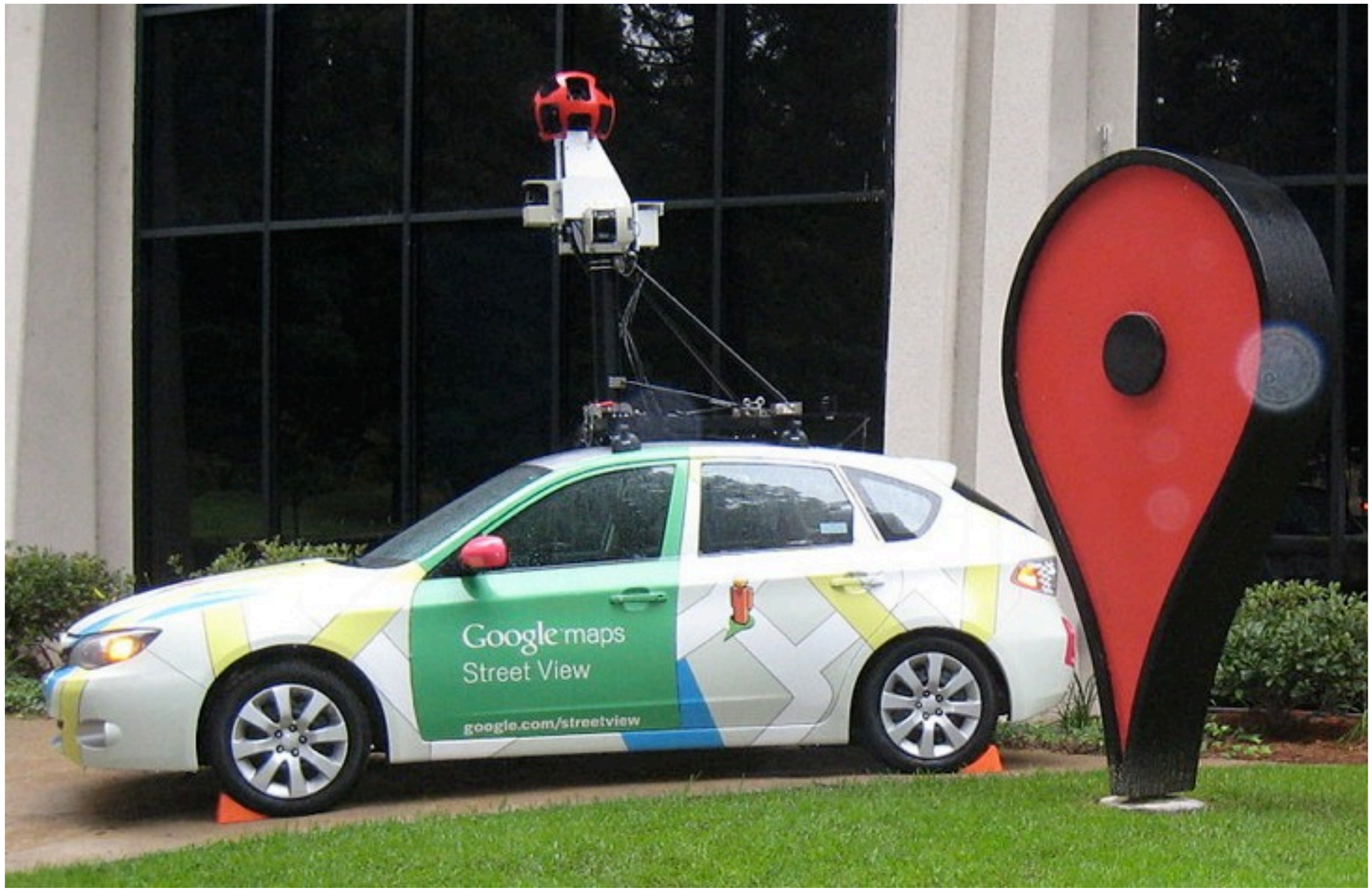


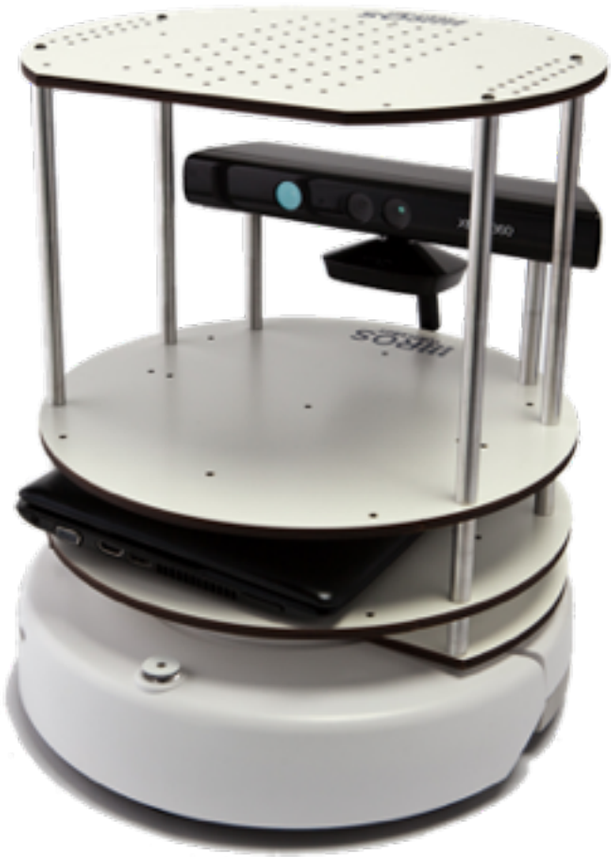
Image Source: Wikipedia, Public Domain

Mapping

Google I/O 11

Home Search Engine

Personal Replicators



+

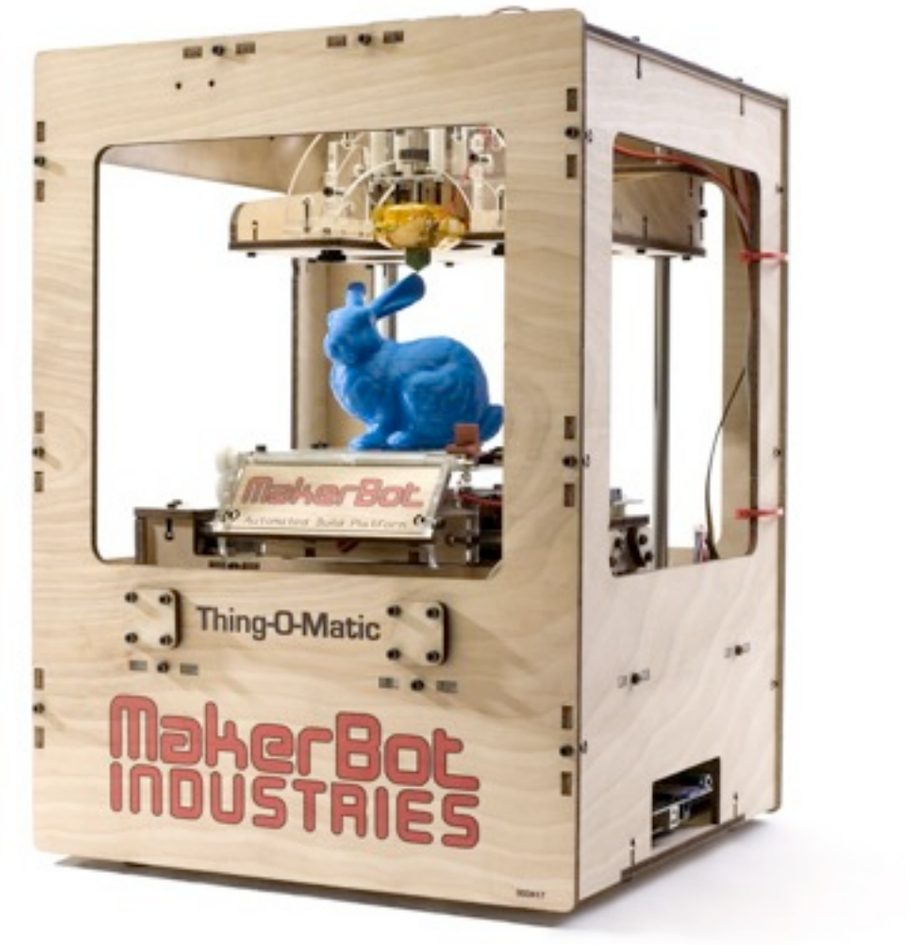
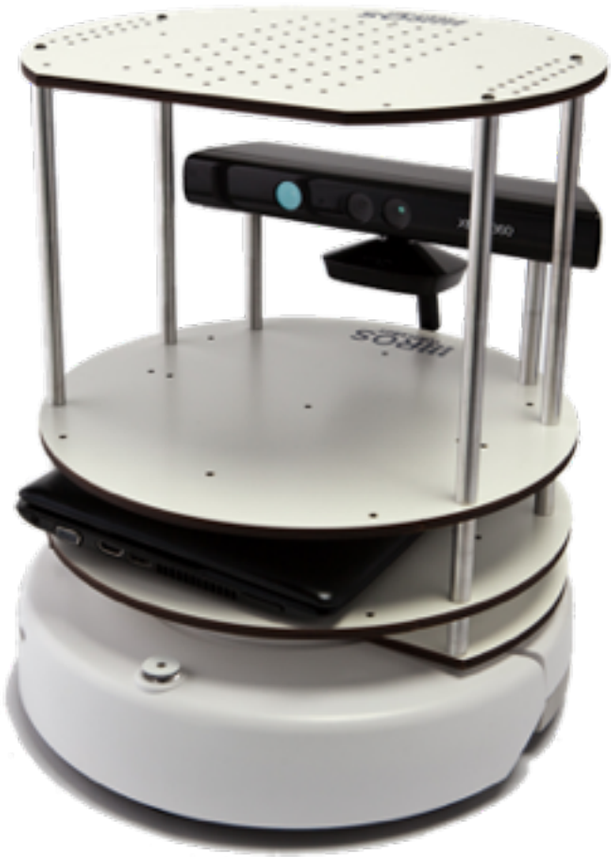


Image Source: Makerbot Industries

Google I/O 11

ROS and Android



+

Google I/O 11

ROS on Android

Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

Google I/O 11

What is rosjava?

- ROS framework written in pure Java
- Android compatible
- Open source
- Early release

Google I/O 11

Nodes

```
public class Talker implements NodeMain {

  @Override
  public void main(NodeConfiguration config) {
    ...
  }

}
```

Publishers

```
@Override
public void main(NodeConfiguration config) {

  Node node = new Node("talker", config);
  Publisher<std_msgs.String> publisher =
      node.createPublisher("/chatter");
  while (true) {
    std_msgs.String msg = new std_msgs.String();
    msg.data = "Hello world!";
    publisher.publish(msg);
    Thread.sleep(1000);
  }

}
```

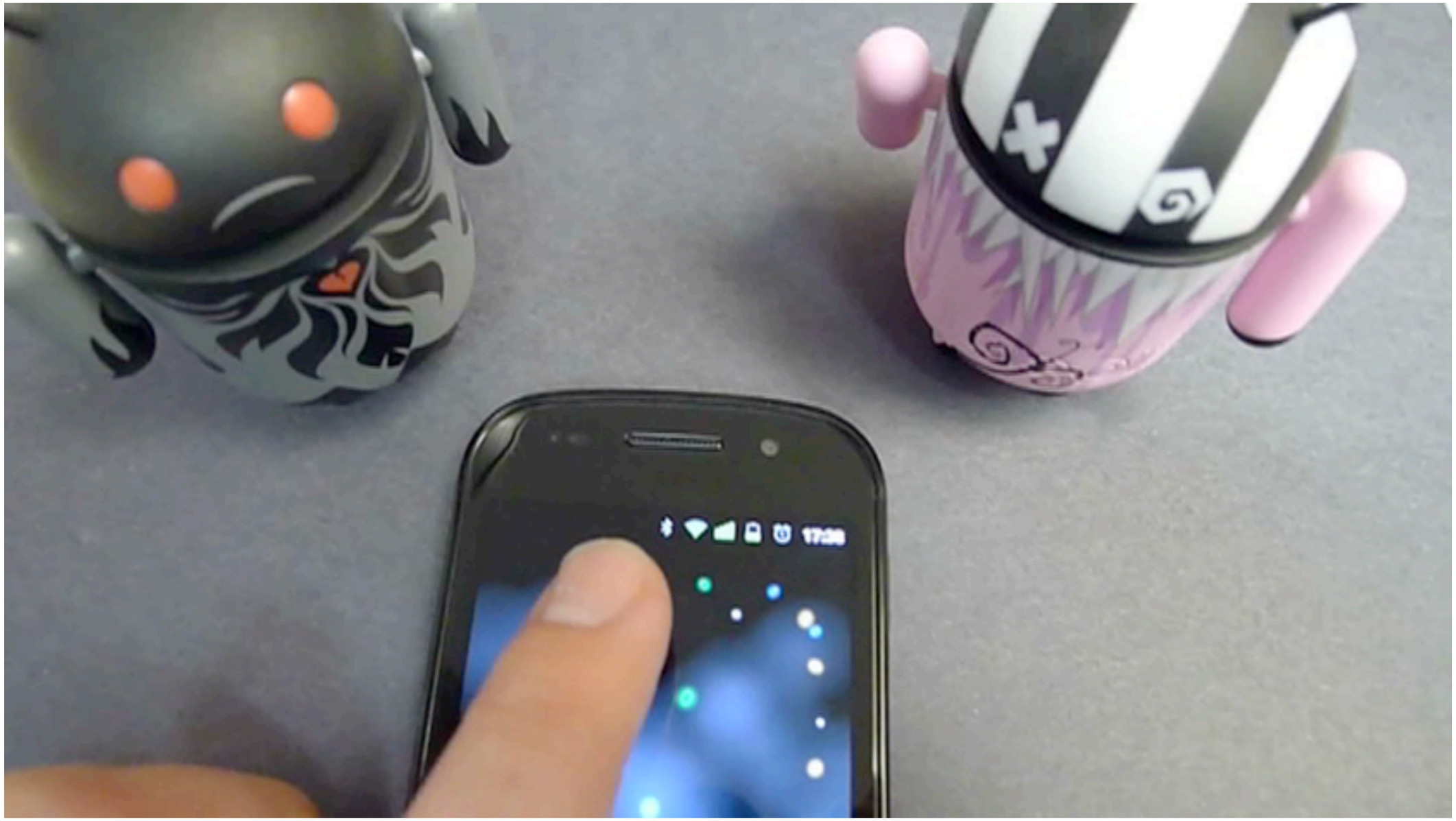
Subscribers

```
@Override
public void main(NodeConfiguration config) {

  Node node = new Node("listener", config);
  node.createSubscriber("/chatter",
      new MessageListener<std_msgs.String>() {
        @Override
        public void onNewMessage(std_msgs.String msg) {
          System.out.println(msg.data);
        }
      });

}
```

Starting rosjava Nodes

```
$ rosrun rosjava run org.ros.Talker
...
$ rosrun rosjava run org.ros.Listener
Hello, world!
Hello, world!
...
```

Hello, World!



Google I/O 11

Running Nodes

```
public class MainActivity extends Activity {

  private final NodeRunner nodeRunner;

  public MainActivity() {
    super();
    nodeRunner = NodeRunner.createDefault();
  }

  ...

}
```

ROS Views

```
@Override
public void onCreate(Bundle savedInstanceState) {

  super.onCreate(savedInstanceState);
  RosTextView<std_msgs.String> txt =
      findViewById(R.id.txt);
  txt.setTopicName("/chatter")

  nodeRunner.run(new Talker());
  nodeRunner.run(txt);

}
```

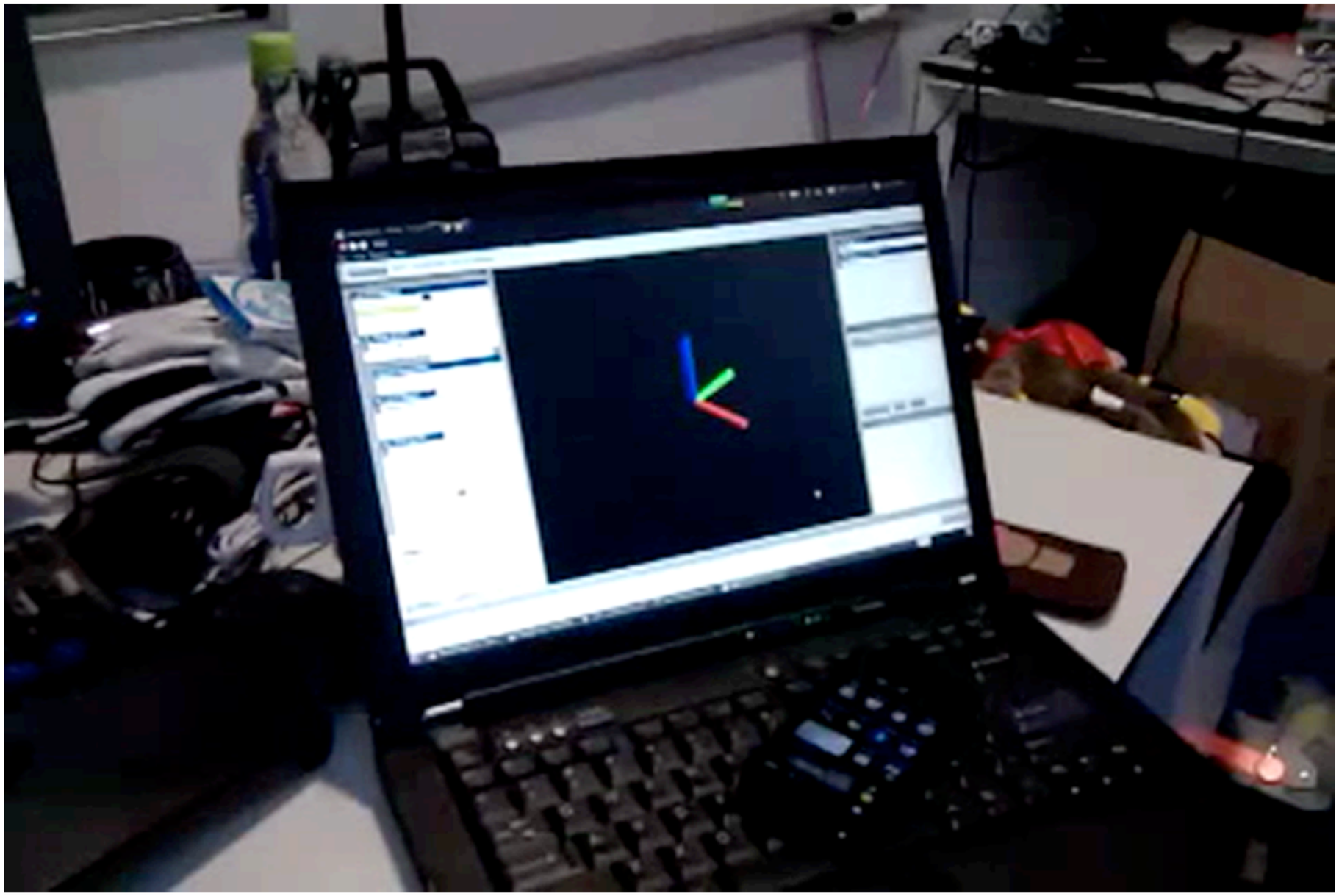
RosTextView

```
public class RosTextView<T> extends TextView
    implements NodeMain {

  private String topicName;
  private Node node;

  @Override
  public void main(NodeConfiguration config) {
     ...
  }

  ...

}
```

ROS Views are Nodes

```
@Override
public void main(NodeConfiguration config) {

  node.createSubscriber(topicName,
      new MessageListener<T>() {
        @Override
        public void onNewMessage(final T message) {
          post(new Runnable() {
            @Override
            public void run() {
              setText(message.toString());
            });
        }
      });

}
```

Visualizing Orientation Readings



Integration with Sensors

```
@Override
public void main(NodeConfiguration config) {

  SensorManager mgr = getSystemService(SENSOR_SERVICE);
  Sensor s = mgr.getDefaultSensor(
      Sensor.TYPE_ROTATION_VECTOR)
  mgr.registerListener(new SensorEventListener() {

    ...

  }, s, SensorManager.SENSOR_DELAY_FASTEST);

}
```

Publishing Orientation Readings

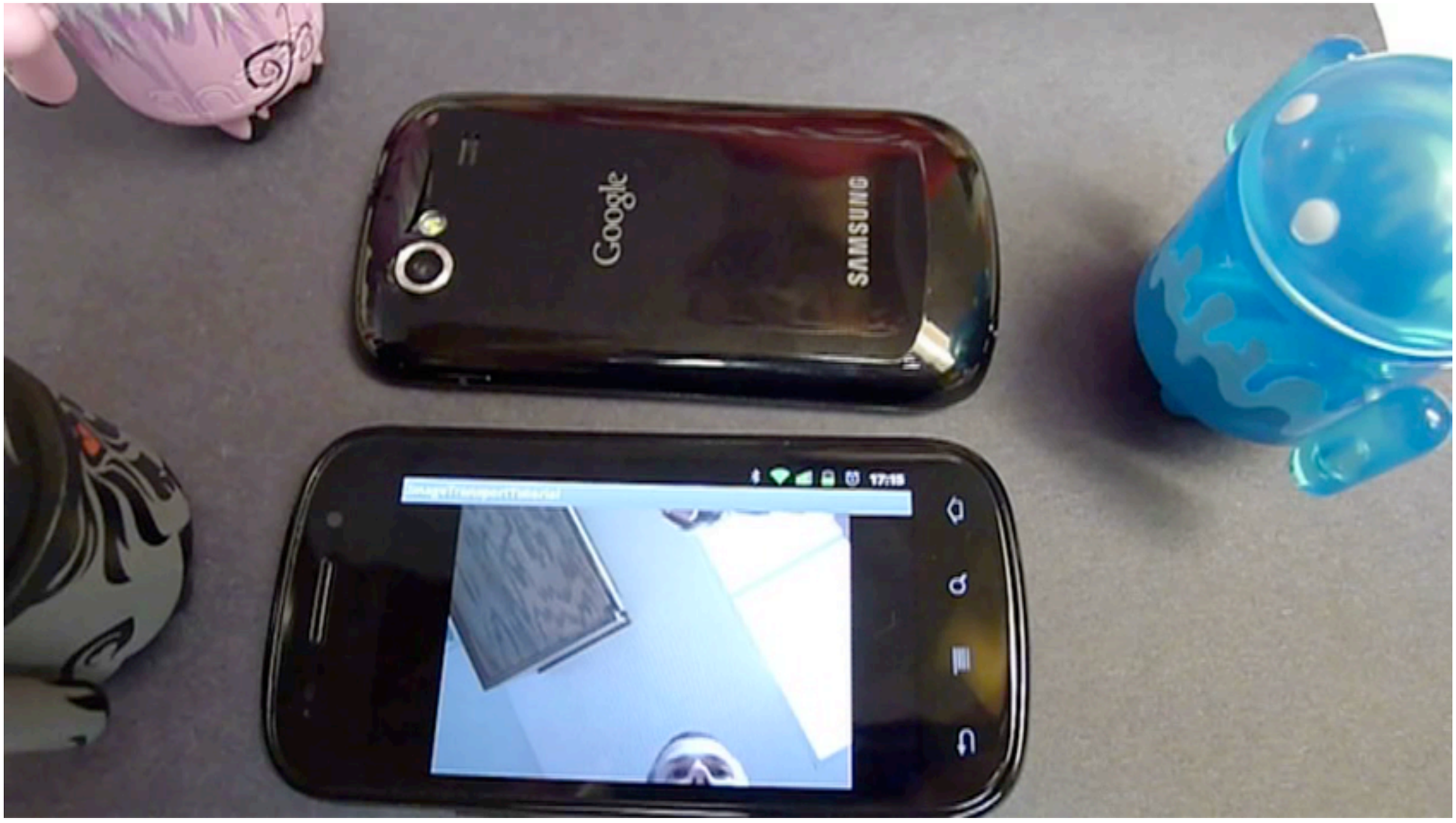
```
@Override
public void onSensorChanged(SensorEvent e) {

  float[] vec = new float[4];
  SensorManager.getQuaternionFromVector(vec, e.values);
  Quaternion orientation = new Quaternion();
  orientation.w = vec[0];
  ...
  PoseStamped pose = new PoseStamped();
  pose.position = new Point(0, 0, 0);
  pose.orientation = orientation;
  publisher.publish(pose);

}
```

Image Transport: Subscribers

RosImageView

```
@Override
public void onCreate(Bundle savedInstanceState) {

  super.onCreate(savedInstanceState);
  RosImageView<CompressedImage> image =
      findViewById(R.id.image);
  image.setTopicName("/camera")

  nodeRunner.run(image);

}
```

Image Transport: Publishers



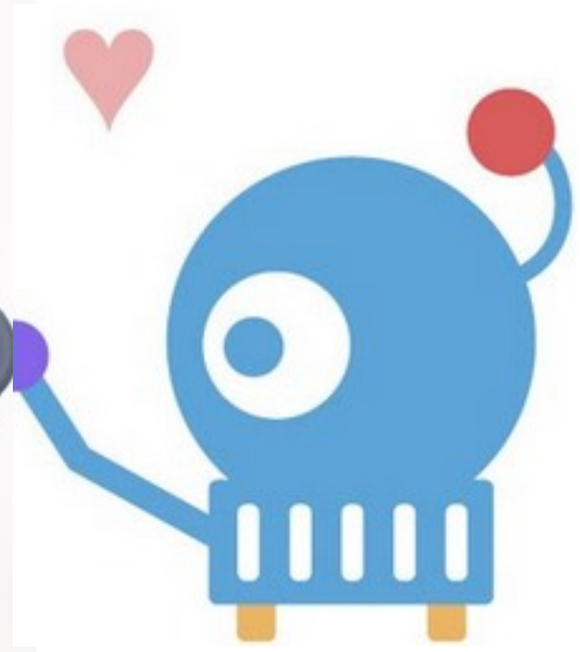
Google I/O 11

Camera Publisher

```
@Override
public void onCreate(Bundle savedInstanceState) {

  super.onCreate(savedInstanceState);
  RosCameraPreviewView camera =
      findViewById(R.id.camera);
  camera.setTopicName("/camera")

  nodeRunner.run(camera);

}
```

PR2 and Android Demo

- Tablet subscribes to PR2 camera images
- PR2 subscribes to tablet orientation



Google I/O 11

Android Possibilities

- Open Accessory API
- Accelerometer
- Gyroscope
- Compass
- GPS
- Camera
- Touch screen
- Wireless network

cloud robotics

Closing Remarks

Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

Google I/O 11

Object Recognition Demo

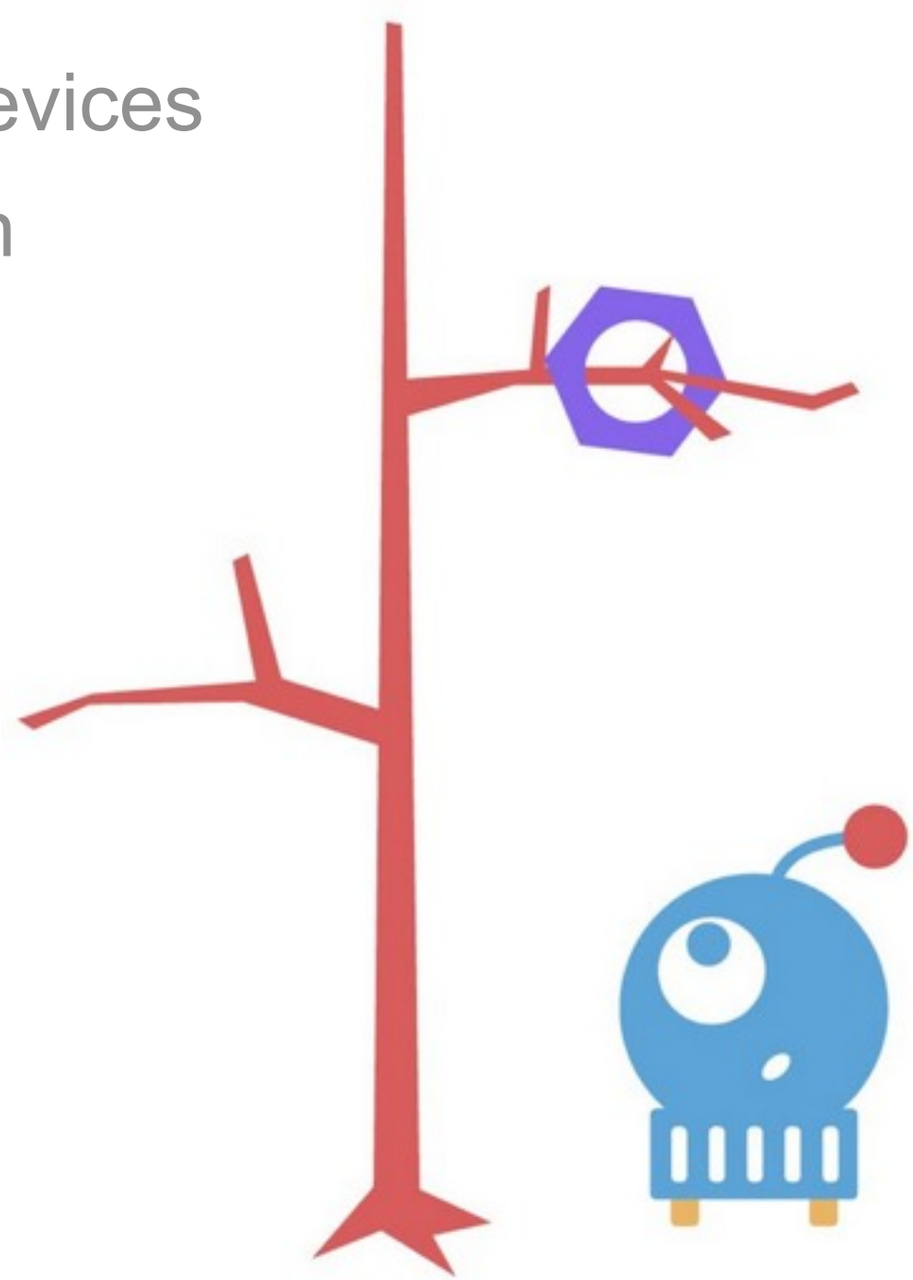
- Train a custom corpus of objects
- Store the knowledge in the cloud
- Any robot can access the shared objects

Object Recognition Demo

Project Phōndox

What robotics problems can you tackle?

- Process large amounts of data
- Share knowledge among multiple devices
- Create new forms of user interaction

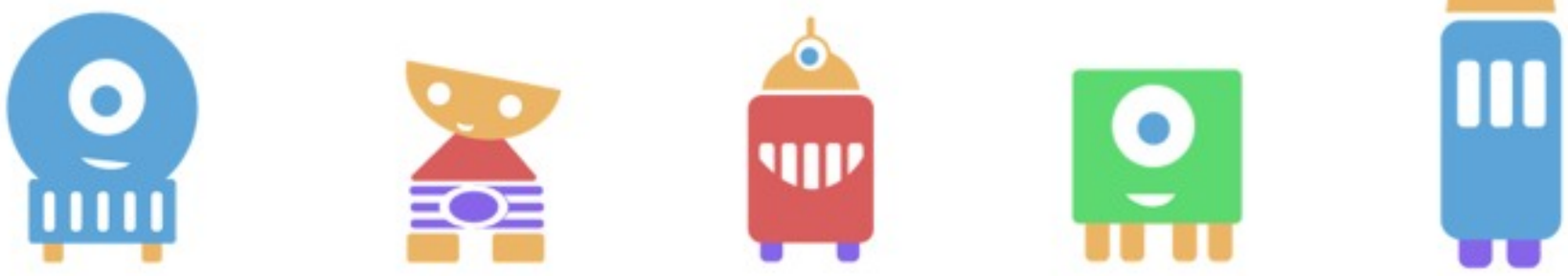


Enable your apps for cloud-connected robots!

- ROS-enable web services and applications
- ROS-enable Android apps
- Connect hardware to Android devices
- Connect the physical world to the cloud

- Visit http://www.cloudrobotics.com/
- See us in the Android Interactive Zone on the 3rd floor
- See us at Maker Faire on the 21st and 22nd

Q&A



Feedback: http://goo.gl/tTQmW

Hashtags: #io2011 #TechTalk

Google I/O 11