RosJava Android

Jordan Allspaw and Eric Marcoux
RosJava

ROSJava is an all Java implementation of ROS
Does not rely on native code

Publisher’s/ Subscribers referred to as NodeMain’s.
Managed by a “NodeMainExecutor”

Why use it?
ROSJava Android - Extensions to ROSJava for use with Android
Collision Logger
Joystick Teleop
Google Glass Teleop
RViz Point Cloud with Tango
Heartaches and Migraines

- The compute power in an android device != compute power of your computer, yet

- Need to seriously consider the publishing rate of messages received by a ROSJava node.
  - CPU time
  - Blocking UI Thread
  - Garbage Collection

- Two Options - Android RenderScript or the Native Development Kit (NDK)
Heartaches and Migraines (NDK)

- **Pros**
  - Possible to achieve nearer to real time processing of high rate messages
  - Garbage collection does not occur

- **Cons**
  - Garbage collection does not occur
  - Be careful when renaming native methods (use a FooLib class)
  - Make sure not to forget env and object parameters added to each method signature (or use javah)
  - Be cautious of JNI overhead from calling native methods from Java land or java methods from native land.
Gradle / Android Studio support for JNI is not there.
Thank you!

Jordan Allspaw and Eric Marcoux

http://robotics.cs.uml.edu/