Peripherals

Beyond Ros Core

Daniel Stonier
Yujin Robot

Contents

- Technologies
- Interactions
- Multi-Robot-Device
- Korean Group - ETRI

Connecting Technologies

RQT

Reusable tools and building blocks for ros with qt.

Java/Android

Bridging to a java-based ecosystem

OR

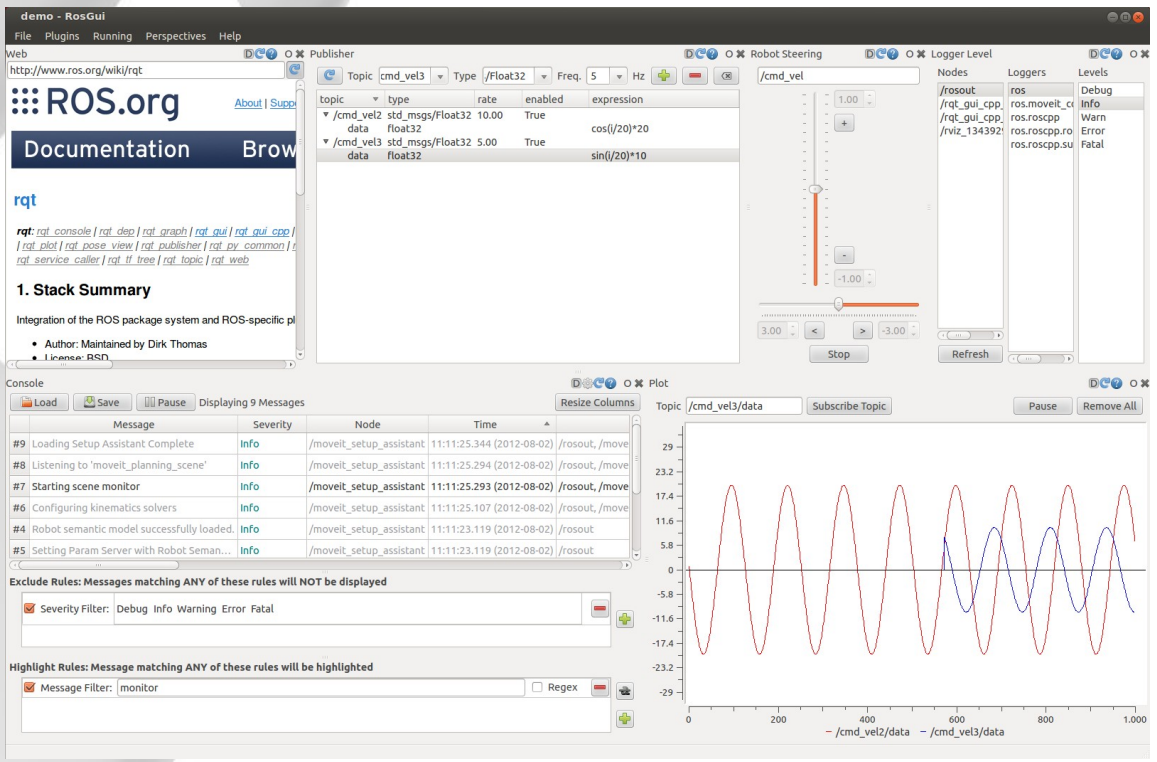
Hand held interactions with androids

Web Tools

Device agnostic
ui's

YUJIN ROBOT

RQT



The Official Graphical Framework for Ros Tools

C++ or Python
Your Choice!

Lots of Reusable Modules and Widgets

Full Access to Ros Core API and Tooling

Completely integrated catkin workflow

Low Learning Curve and Lots of Support

Potential to be cross-platform (linux-mac-win-tablet) but not enough critical mass

http://wiki.ros.org/rqt

YUJIN ROBOT

Java/Android

Reasonably Complete Comms and Core Libraries

New to Hydro

Java Debs
Individual Message Jars
Maven Repo On Github
Better Gradle Integration
The Android Studio Revolution!

What Does This Mean???

Build Side By Side with Ros (Catkin Make)
OR
Build Standalone Without Ros (Maven Repos)

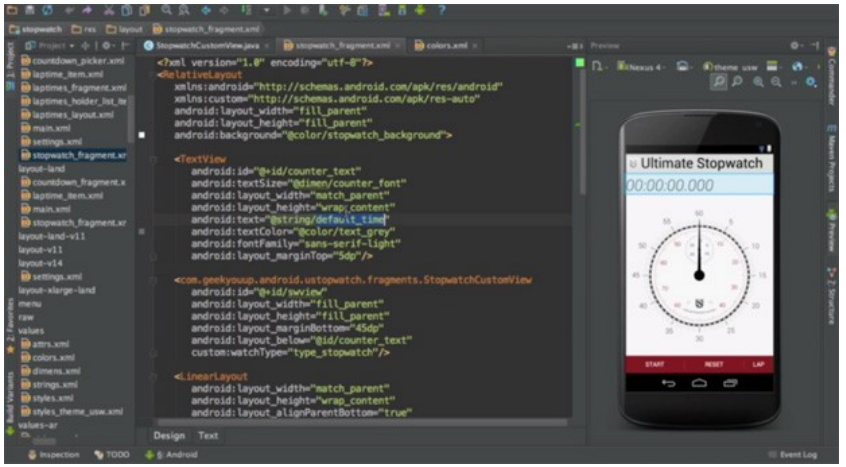
Manage a Collection of Android Libraries (.AARs)

Build an Android App without a Complicated Ros Install

Continuous Integration for Android Development

Coming to Indigo

Native Message Generation



Still Very Ros-Like

Missing Some Transports (Actions)
Not Currently Being Developed

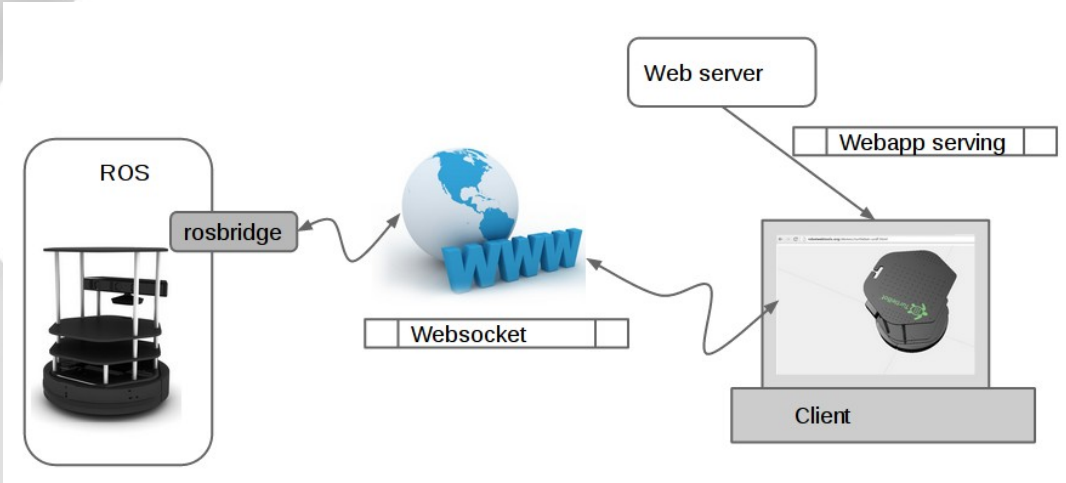
http://wiki.ros.org/rosjava

http://wiki.ros.org/android

YUJIN ROBOT

Web Tools

How It Works



Rapidly Growing Collection

Active Development

Zero Installation (User Side)

Very Cross Platform
Embeddable In Other
Web Applications

Requires a Bridge
Can't Easily Access the Device/PC

Comparison

RQT	Android	Web Apps
For Ros Devs	For Java Devs	For SW/Web Devs
Quick Prototyping & Low Maintenance	If you need the device NFC Speed	No Installation for Users
Can do very complex lots of library support	Can do complex algorithms scientific support	Embeddable in Web Application Suites
Dev Tools Factory SW Prototype UI Unstable UI	Android NFC Apps Web App Launchers	Customer Frontends, Embedded Web Apps, Remote Monitoring

@Yujin

YUJIN ROBOT

Rocon

Capabilities

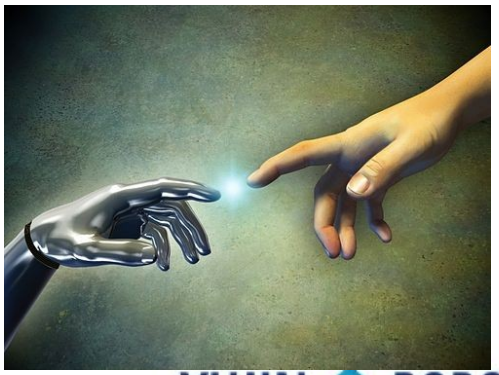
Laying down structure at the bootstrap level of a ros robot.

Robot App Management

A public task-like interface to your ros robot.

Interactions

Start interacting with your ros robot without needing ssh connections or remembering long uri's.



YUJIN ROBOT

Robot Capabilities

Q) “I need Navigation...”

What software stack?

What is the interface specification?

How do I launch it with my software?

What if I want to depend on a modified software stack?

Building Capabilities

Standard Capabilities (Interfaces)

Capability Providers (Implementations)

Runtime – The Capability Server

Start/Stop Capabilities

Remap the ROS api

Semantic Capabilities (head image, side image)

YUJIN ROBOT

Application Management

A Task-Like Interface for Your Robot

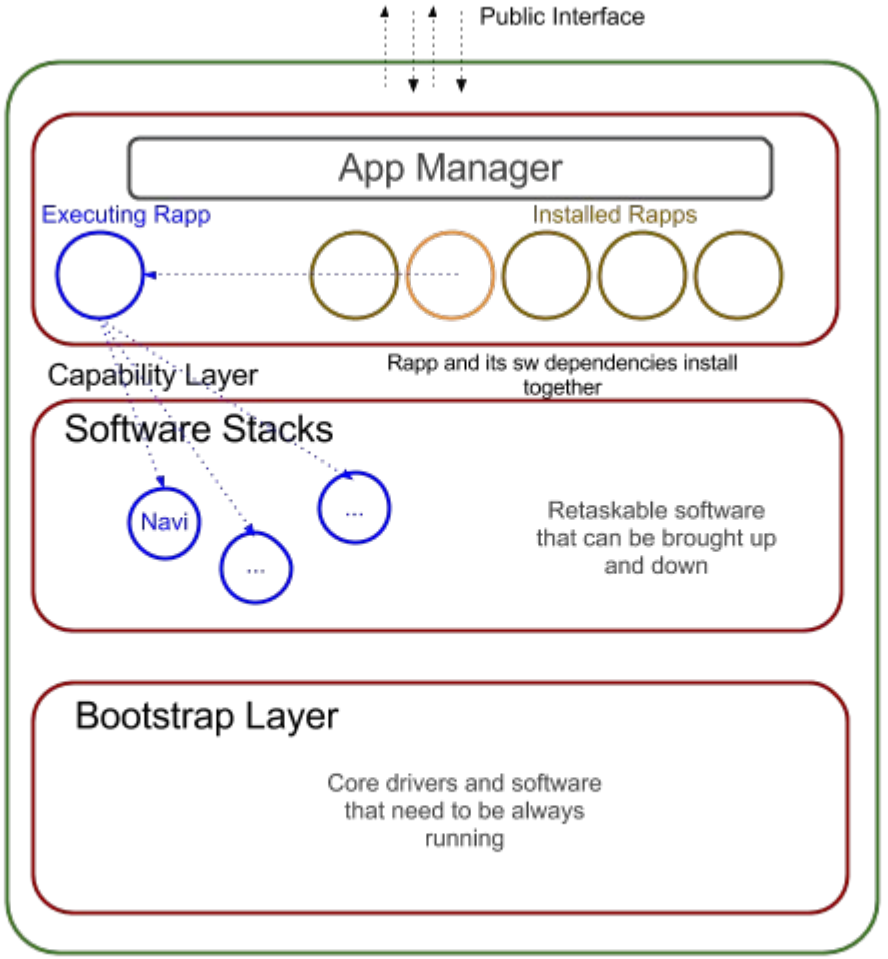
Tear Down and Build Up Entire Launch Environments

Rocon/Robot App Manager

- connect to a rapp repository
- advertise platform information
- install rapps plus their sw dependencies
- configure and launch rapps
- public interface to pairing or concert (multimaster) modes

Single App Execution Engine

Running multiple apps in parallel results in sw and hw resource conflicts making the system much more difficult to handle - start by considering a single app execution engine only.



Rapps

Rapp: robot or rocon app.

Possibly just meta packages providing configuration, launching rules and install dependencies.

Capabilities

Provides linking rules to the software underneath (remap style) and installation dependencies for various robot 'capabilities'.

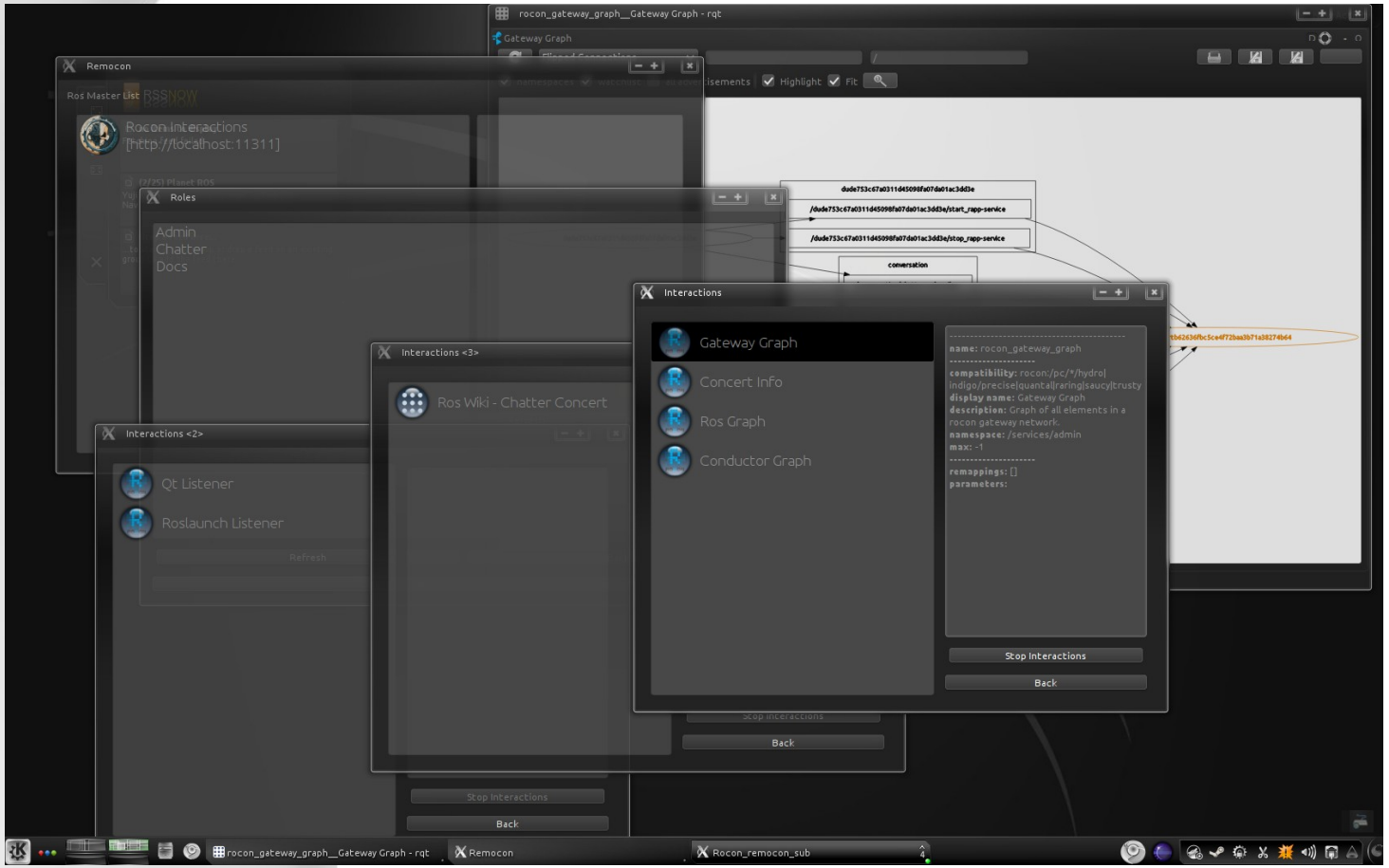
Allows simpler and more portable software development at the rapp level.

http://wiki.ros.org/rocon_app_platform

YUJIN ROBOT

Interactions

Human-Ros Master Software Interactions



Interaction Types

- Qt Tools
- Android Apps
- Web Apps
- Documentation Links
- RosRunnables
- RosLaunchers

Parameterise
and/or
Remap Launches

Pair With Robot App
Managers

Remocons - Interaction Launchpads

Android & Qt Remocons

http://wiki.ros.org/rocon_interactions

YUJIN ROBOT

Concert

Gateways

Distributed Multimaster - each ROS robot as a gateway on a hub.

Orchestration

Is there a right way to orchestrate/choreograph multiple robots?

Gazebo Concert

And you thought debugging one ROS robot was hard?
Enter gazebo!!!

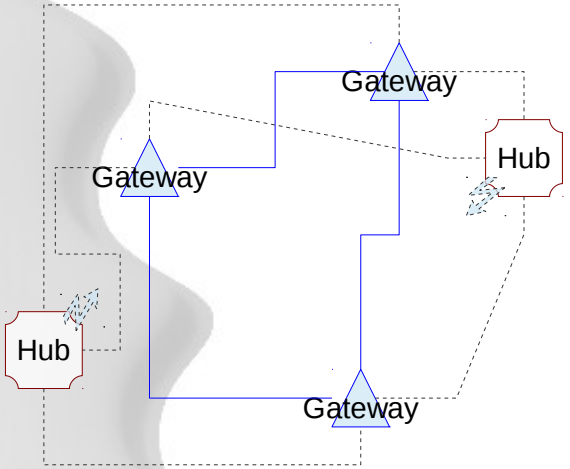
YUJIN ROBOT

Gateways

Multimaster Communications

- True multimaster
- Hides Ros Masters
- Public Interfaces
- Firewalls
- Auto-Discovery (Zeroconf)

Each Gateway is a Door to a Ros Master



Hubs are like Multimaster Name Servers

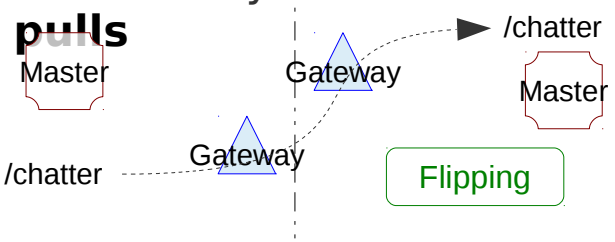
Now with Robust Wireless Handling

Flips, adverts & pulls are described by rules placed on watchlists.

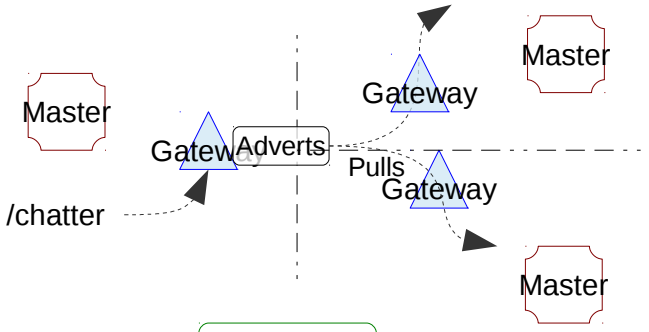
Gateways act on them as they locally appear and disappear.

Flips & Advertisements/Pulls

- Control where topics are sent with **flips**
- Share freely with **adverts** & **pulls**



Flipping



Adv & Pull

Build Your Own Multimaster System on Top of the Gateways

http://wiki.ros.org/rocon_gateway

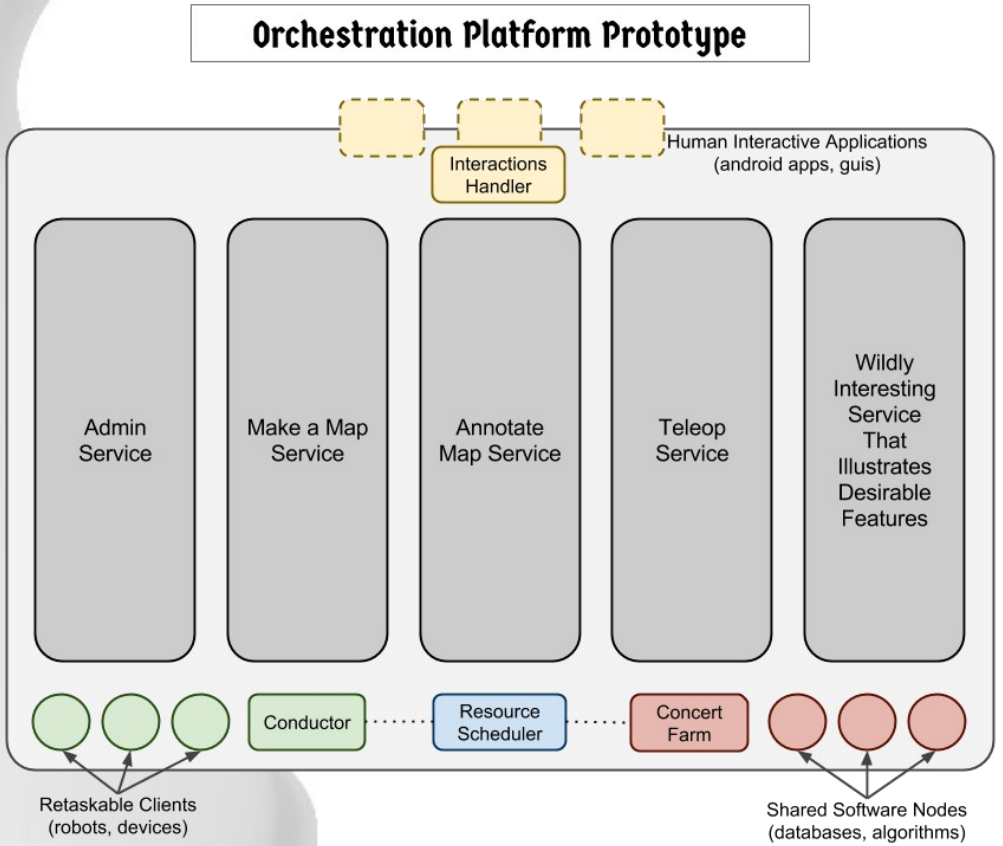
YUJIN ROBOT

Orchestration

Orchestrating Services with Multiple Robots-Devices-Humans

There is no Right Way!!!
Yet...

So Much Infra is Needed...



Internal features

- Wireless Handling
- Multiple Service Paradigms (ROS, Static Link Graphs, Orc, BPEL)

We Decided:

Parallel Services
(Like a Web Server)

Each service is an independent orchestration block

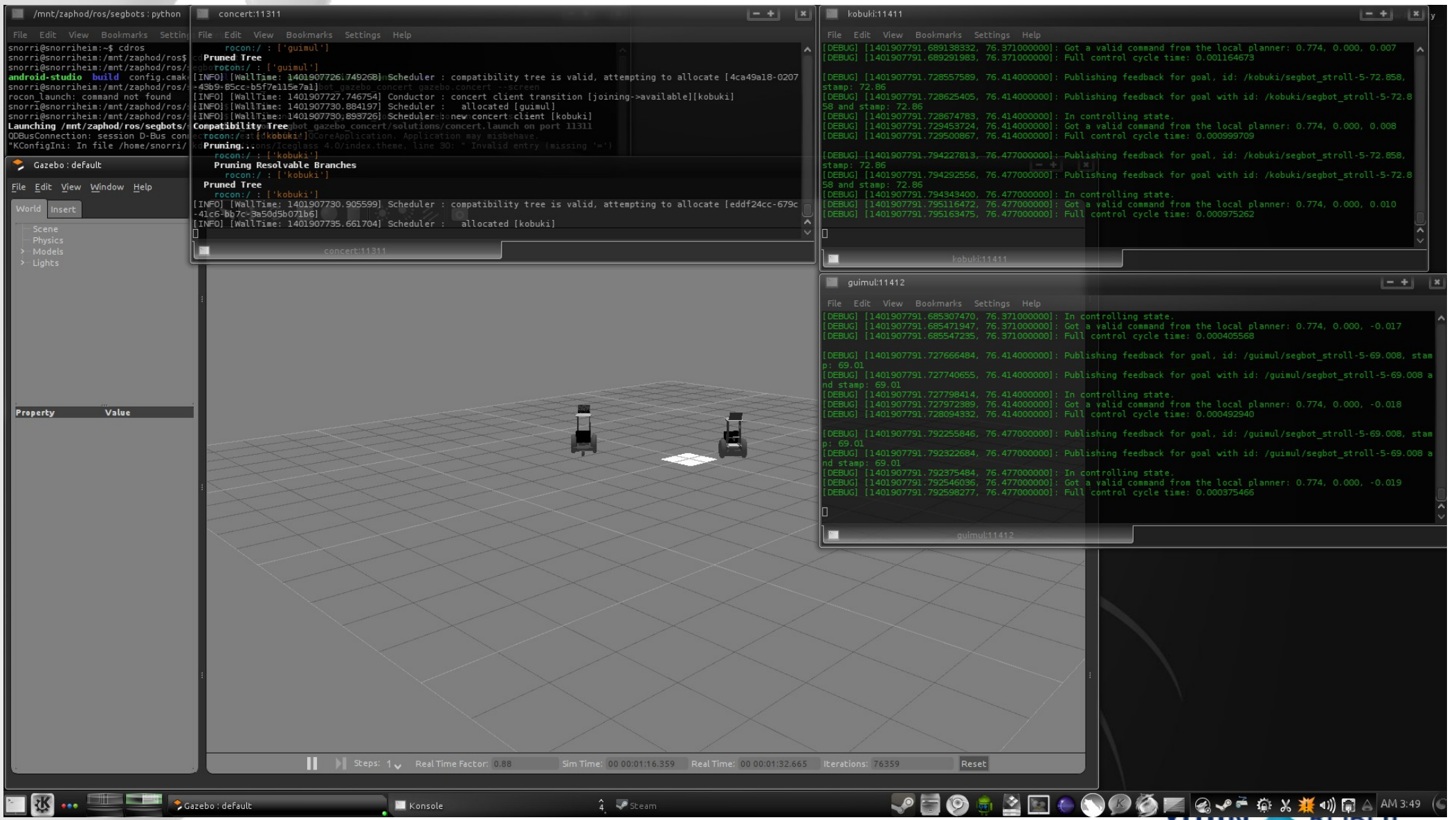
Give services the freedom to do their own orchestration

We Just build the Infra Around the Services
Communications,
Robot Task Scheduling,
Software Launching,
Human Interactions,

http://wiki.ros.org/rocon/indigo/Guide

YUJIN ROBOT

Gazebo Concert



Ros in Korea

Electronics and Telecommunications Research Institute
ETRI Co-Pilot System with Ros
YongBon Koo
Senior Researcher
Autonomous Driving System Research Section

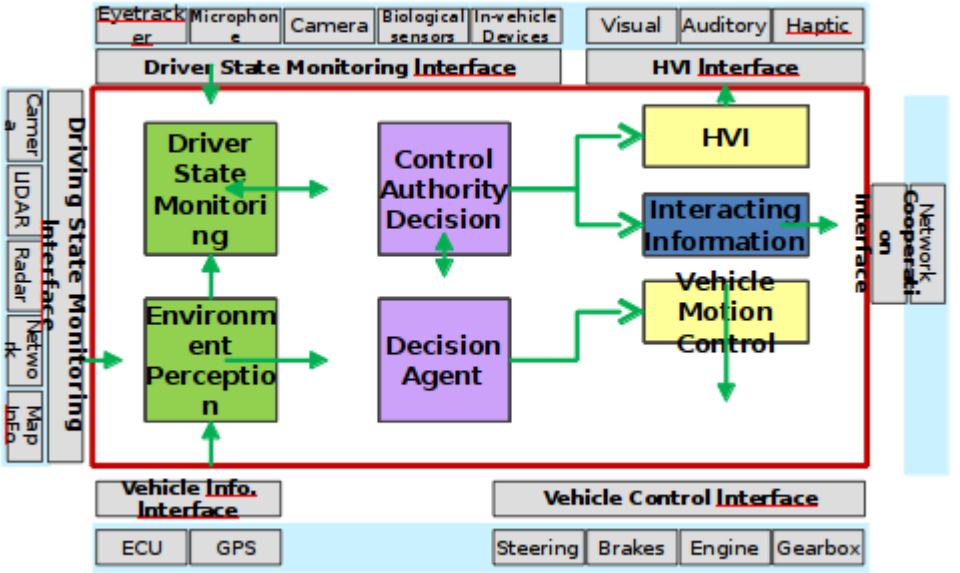
ETRI Co-Pilot System

Co-Pilot System is

- Basically, a fully autonomous vehicle system
- Like a google driverless car
- Human and autonomous vehicle cooperation

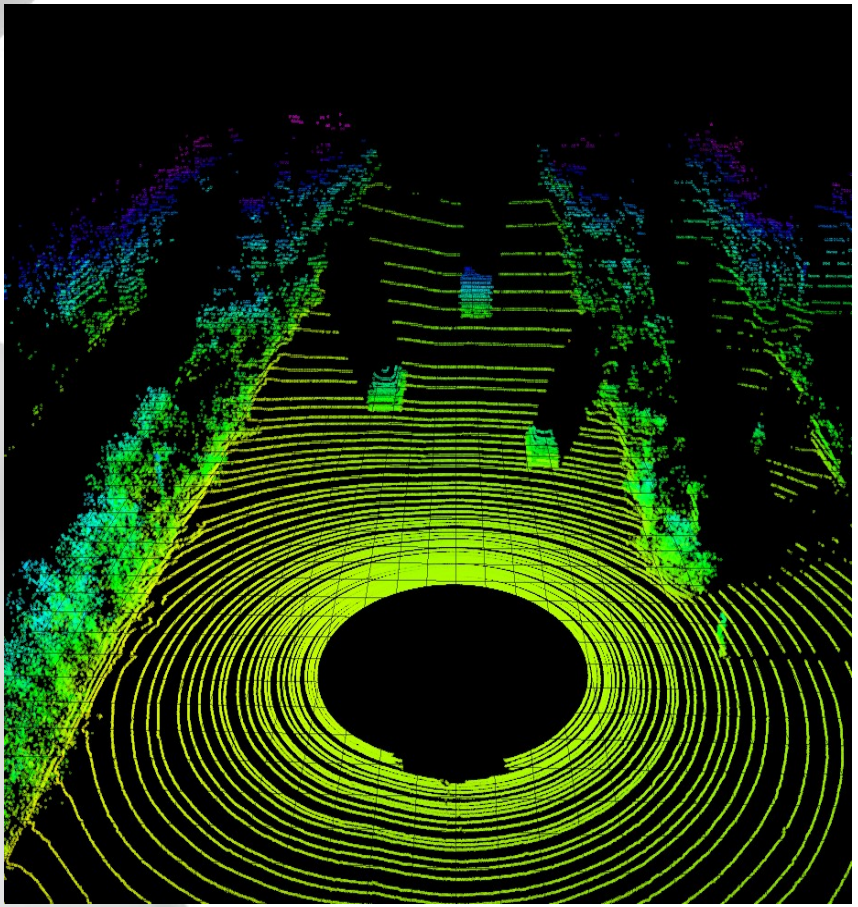
Ros for Co-Pilot System

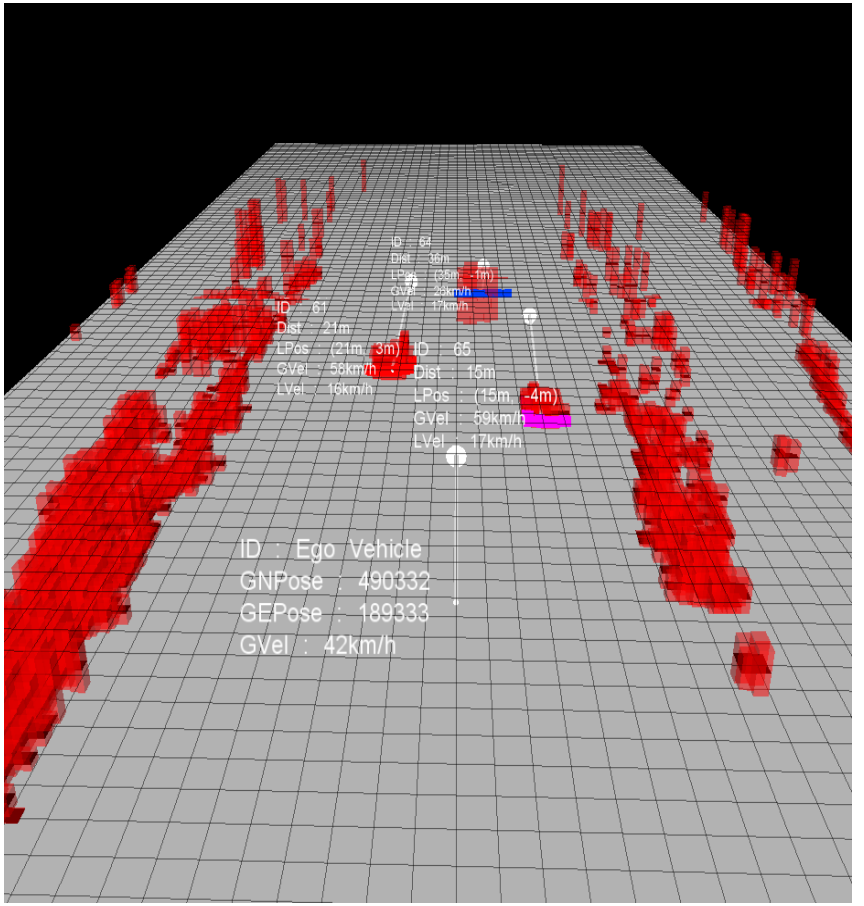
- Is a IPC framework between the b
- Offers some basic components



Environment Perception

Visualization using RViz

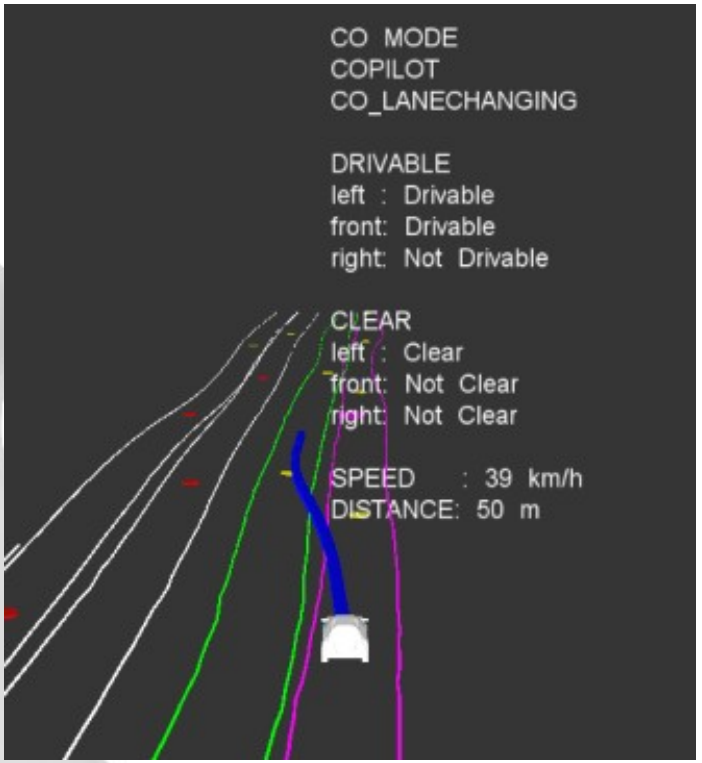


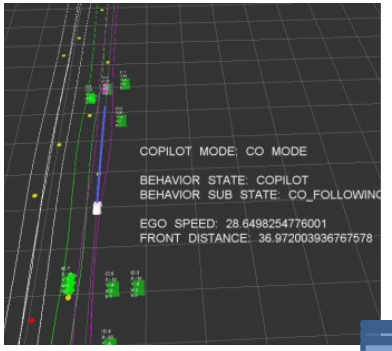


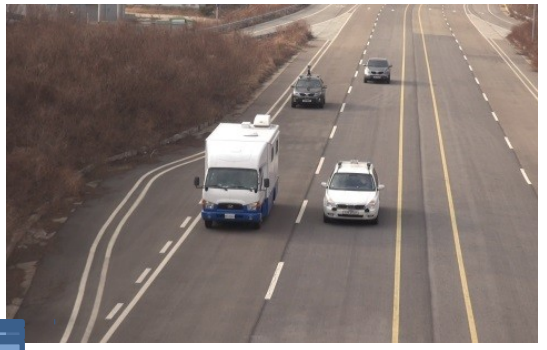
Decision Agent

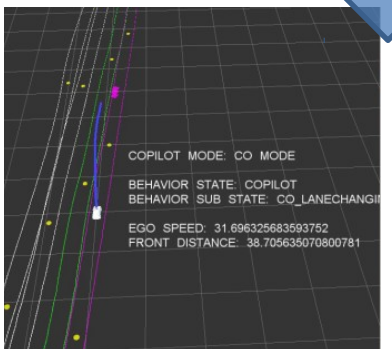
Based on Existing Univ. of Texas-Austin Code

Map Lanes, Observers









ETRI – Why We Chose ROS

Ros is Great For

- Who wants to make a working prototype rapidly
- You can use many components for
  - Communication, Sensor, Path Planning

- Fantastic for
  - Visualization, I love it!
  - Logging and Replay – I can't live without it!

But it Lacks

- Support for other platforms
  - We need ROS for LabView, Matlab, OSX, iOS.

- Documentation
  - Make it together

YUJIN ROBOT