2

走近ROS2.0

主讲人：胡春旭
时　间：2017年7月24日
邮　箱：huchunxu@aigalaxy.com
古月居：http://www.guyuehome.com

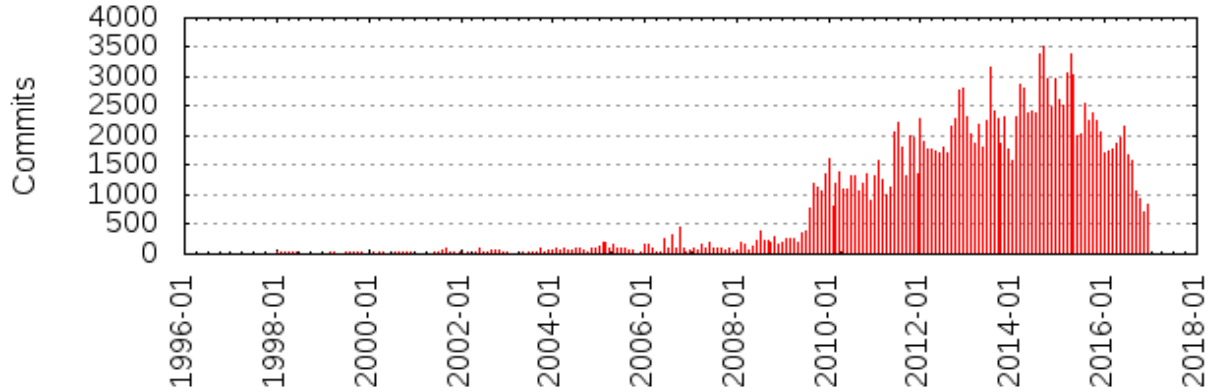
星河智能
AIgalaxy

目录 Contents

Why-ROS is ...

Why-ROS 9岁半了

- **起源** - 2007年11月
- **目标** - 提高代码复用率
- **成长** - 指数级
- **现状** - 机器人领域的事实标准



MoveIt!
pcl
GAZEBO
MOOS
OpenCV
THE OROCOS PROJECT
YARP

星河智能 Algalaxy

Why-ROS1的局限

- 独立的机器人（PR2）
- 工作站级别的计算资源
- 没有实时性的需求
- 良好的网络连接
- 主要应用于研究

Why-ROS1的困境

- ROS的应用领域已经不局限于学术研究
- 基于ROS的产品已经逐渐走向市场
- 支持机器人的种类繁多：轮式机器人，人形机器人,工业机械手,室外机器人（如无人驾驶汽车），无人飞行器, 救援机器人....
- 政府机构也开始注意ROS，例如NASA准备在Robonaut 2上搭载ROS
- 虽然还可以较好的使用，但是将逐渐陷入困境




Why-ROS1的困境

- **多机器人系统** - 没有构建多机器人系统的标准方法
- **跨平台** - 无法适用于windows、 RTOS等系统
- **实时性** - 缺少实时性方面的设计
- **网络连接** - 需要良好的网络环境保证数据的完整性
- **产品化** - 从科学研究到消费产品的过渡欠佳
- **项目管理** - 无法胜任完整生命周期下项目管理

Why-ROS2的曙光

ROS已经走过九个年头，伴随着机器人技术的大发展，ROS也得到了极大的推广和应用。尽管ROS还存在不少局限性，但无法掩盖ROS的锋芒，社区内的功能包还是呈指数级逐年上涨，为机器人开发带来了巨大的便利。不少开发者和研究机构还针对ROS的局限性进行了改良 ，但这些局部功能的改善往往很难带来整体性能的提升，机器人开发者对新一代ROS的呼声越来越大，ROS2.0的消息也不绝于耳。

终于在ROSCon 2014上，正式公布了新一代ROS的设计架构（Next-generation ROS: Building on DDS），2015年8月第一个ROS2.0的alpha版本落地，2016年12月19日，ROS2.0的beta版本正式发布。众多新技术和新概念应用到了新一代的ROS之中，不仅带来了整体架构的颠覆，更是增强了ROS2.0的综合性能。

Why-版本进化

Release Overview	Date
Version 1.0	13 December 2017
beta 3	13 September 2017
beta2	5 July 2017
beta1	19 December 2016
alpha8	4 October 2016
alpha7	11 July 2016
alpha6	1 June 2016
alpha5	5 April 2016
alpha4	17 February 2016
alpha3	18 December 2015
alpha2	3 November 2015
alpha1	31 August 2015

Why-Not just enhance

架构的颠覆

- ROS1的架构下，所有节点需要使用Master进行管理
- ROS2使用基于DDS的Discovery机制，和Master说拜拜

API的重新设计

- ROS1中的大部分代码都基于2009年2月设计的API
- ROS2重新设计了用户API，但使用方法类似

编译系统的升级

- ROS1使用rosbuild、catkin管理项目
- ROS2使用升级版的ament

What-ROS2 is ...

- Not going to break ROS 1, and will not be rolled out into ROS 1.
- Breaking API with ROS 1, but conceptually very similar.
- A chance to reevaluate design decisions.
- A chance to use more modern dependencies and tools.
- A chance to change fundamental parts of the system.
- Going to interoperate with ROS 1 (probably with a bridge).
- Going to continue using the ROS 1 .msg IDL format (genmsg).
- Embracing C++11 and Python3.
- Going to have a full feature C API.
- Will use DDS as the middleware.
- Aiming for real time, embedded, and Windows.
- Built on standards, like DDS’ s RTPS and X-Types.
- Gonna be awesome.

What-ROS2的目标

Support multi-robot systems
involving unreliable networks

Remove the gap between
prototyping and final products

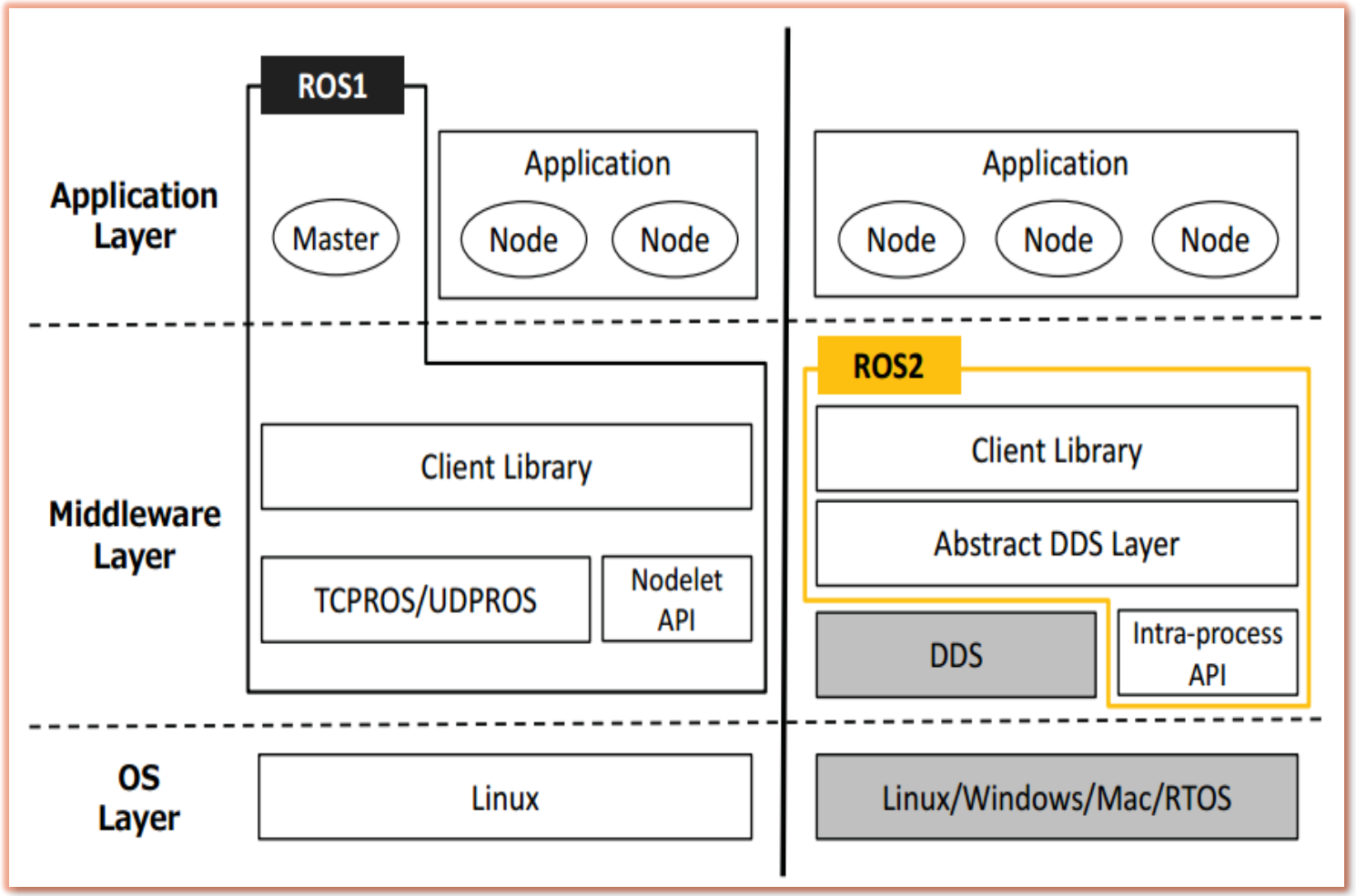
“Bare-metal”
micro controller

Support for
real-time control

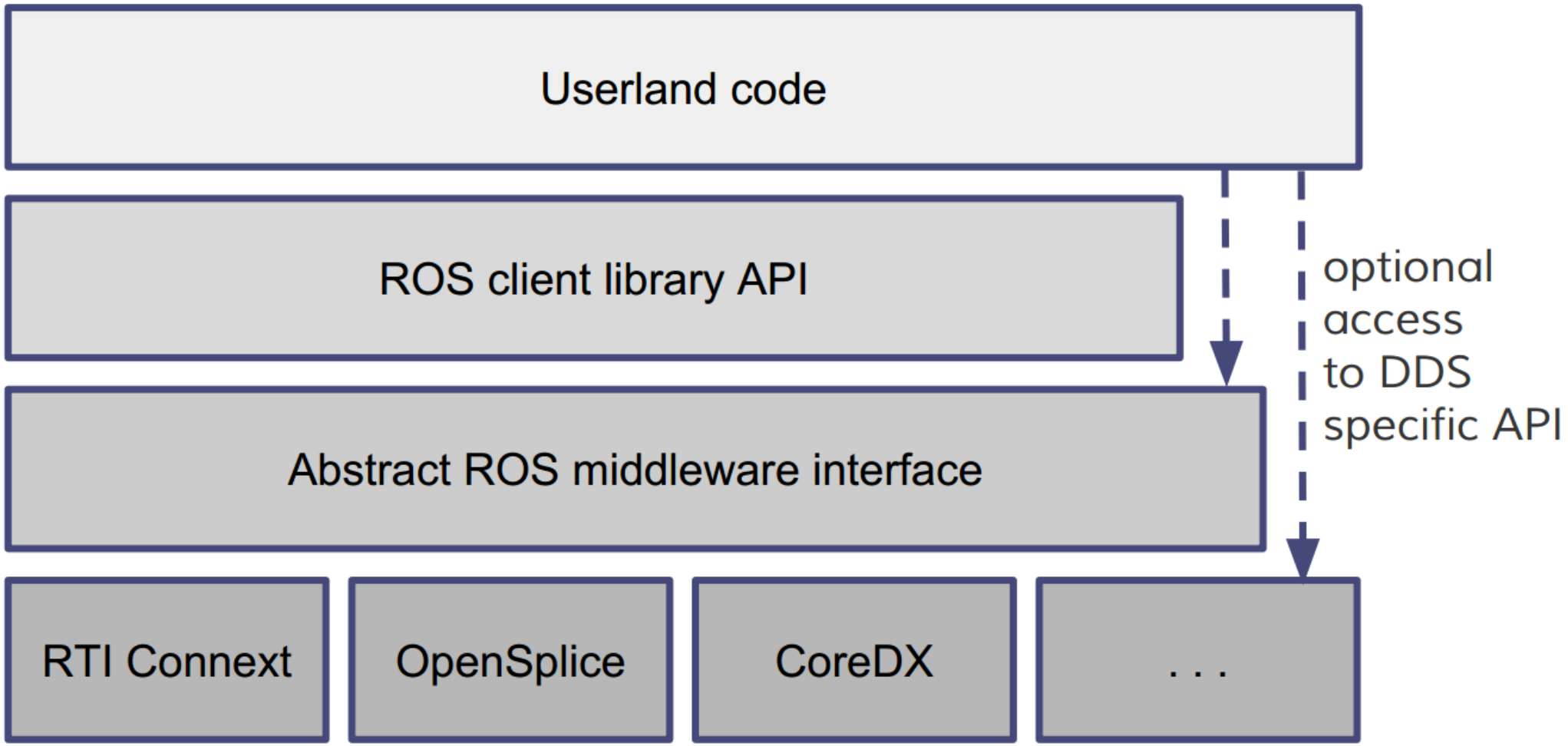
Cross-platform
support

What-ROS2的架构

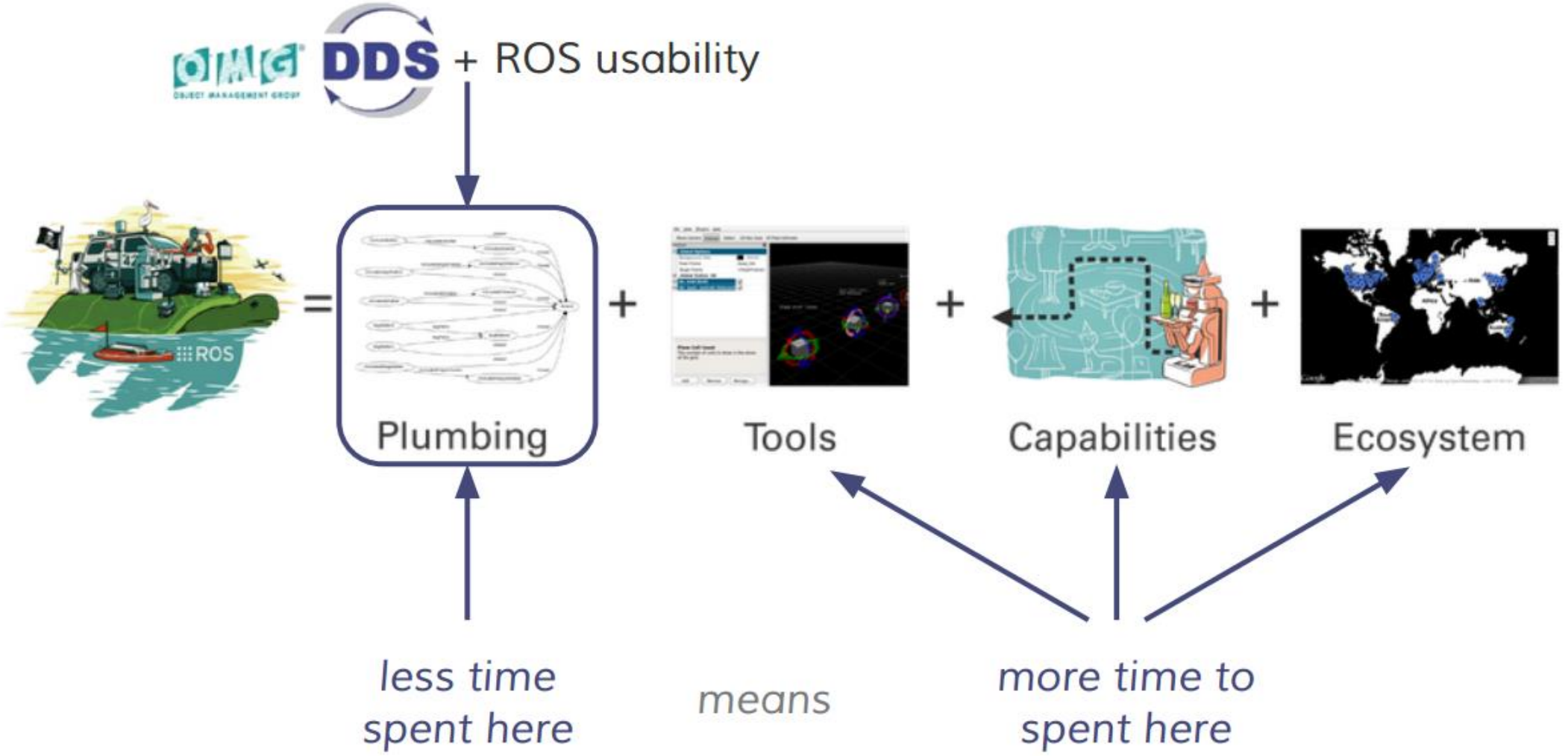
- **OS**
  - ROS1 : Linux
  - ROS2 : Linux、 Windows、 MAC、 RTOS
- **通讯**
  - ROS1 ：TCPROS/UDPROS
  - ROS2 ： DDS
- **节点模型**
  - ROS1 ：发布/订阅
  - ROS2 ：发现
- **进程**
  - Nodelet
  - Intra-process



What-ROS2的架构



What-ROS2的架构

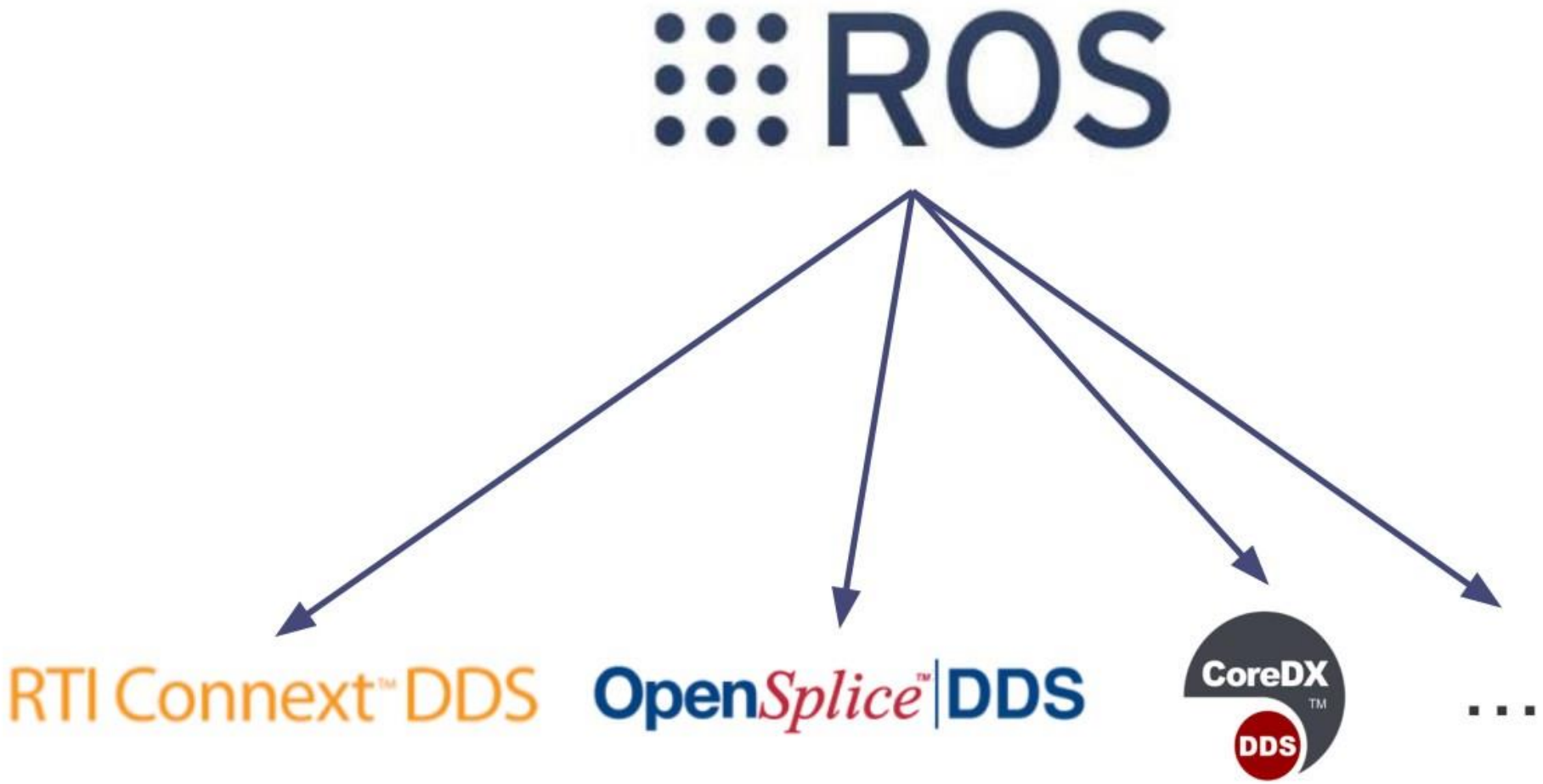


What-DDS是何方神圣

DDS，Data Distribution Service，即**数据分发服务**

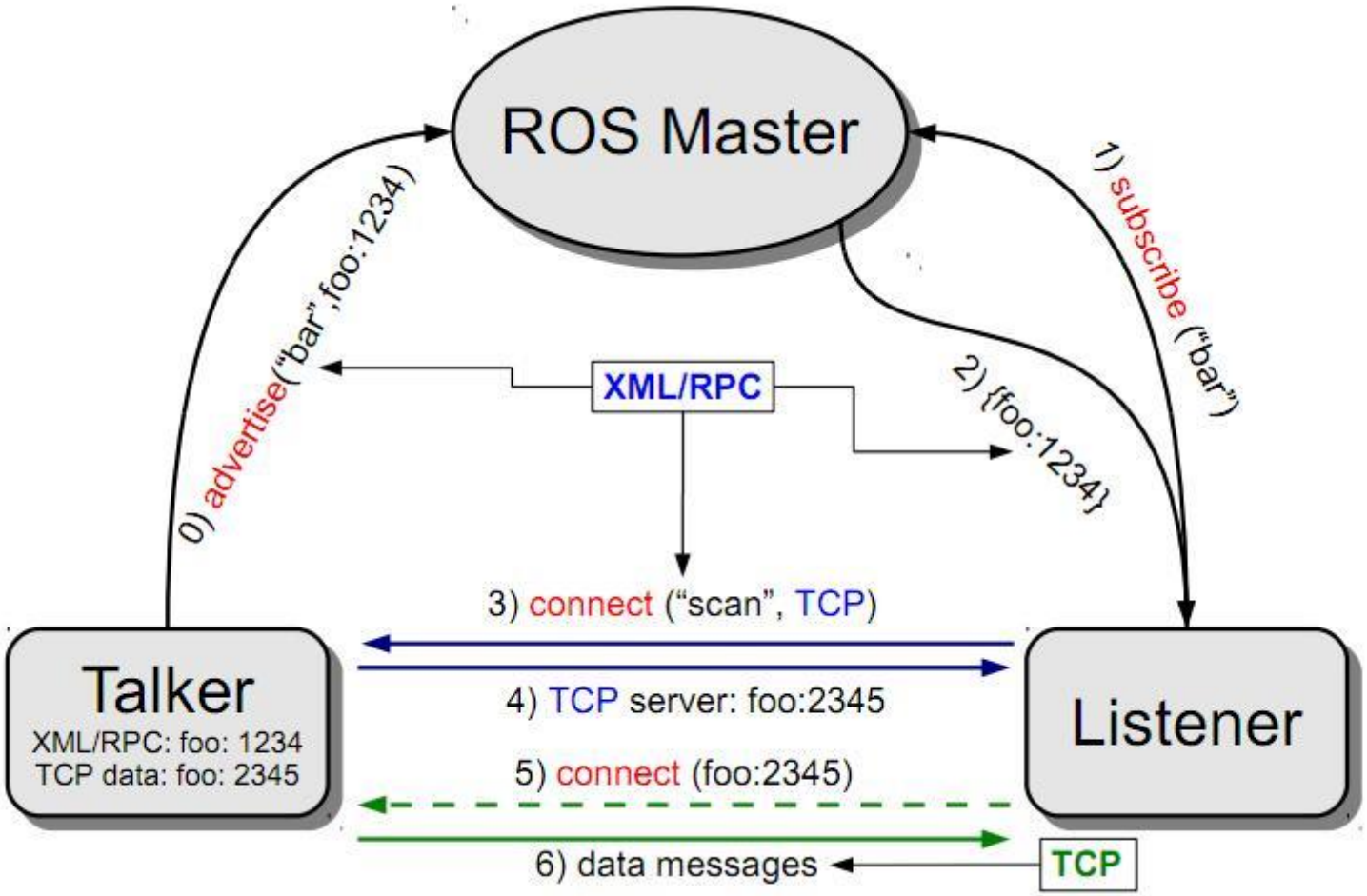
- 2004年由**对象管理组织**OMG（Object Management Group）发布
- 专门为**实时系统**设计的数据分发/订阅标准
- 最早应用于美国海军， 解决舰船复杂网络环境中大量软件升级的兼容性问题，目前已经成为美国国防部的强制标准，同时广泛应用于国防、民航、工业控制等领域，成为**分布式实时系统中数据发布/订阅的标准解决方案**。
- 技术核心是**以数据为核心的发布订阅模型**（Data-Centric Publish-Subscribe ，DCPS），这种DCPS模型创建了一个“**全局数据空间**”（global data space）的概念，所有独立的应用都可以去**访问**。
- 在DDS中，每一个发布者或者订阅者都称为**参与者**（participant），类似于ROS中节点的概念。每一个参与者都可以使用某种定义好的数据类型来读写全局数据空间。

What-DDS供应商



What-ROS1的通讯模型

- Talker注册
- Listener注册
- ROS Master进行信息匹配
- Listener发送连接请求
- Talker确认连接请求
- 建立网络连接
- Talker向Listener发布数据



What-ROS2的通讯模型

1.参与者（Domain Participant）：一个参与者Participant就是一个容器，对应于一个使用DDS的用户，任何DDS的用户都必须通过Participant来访问全局数据空间。

2.发布者（Publisher）：数据发布的执行者，支持多种数据类型的发布，可以与多个数据写入器（DataWriter）相联，发布一种或多种主题（Topic）的消息。

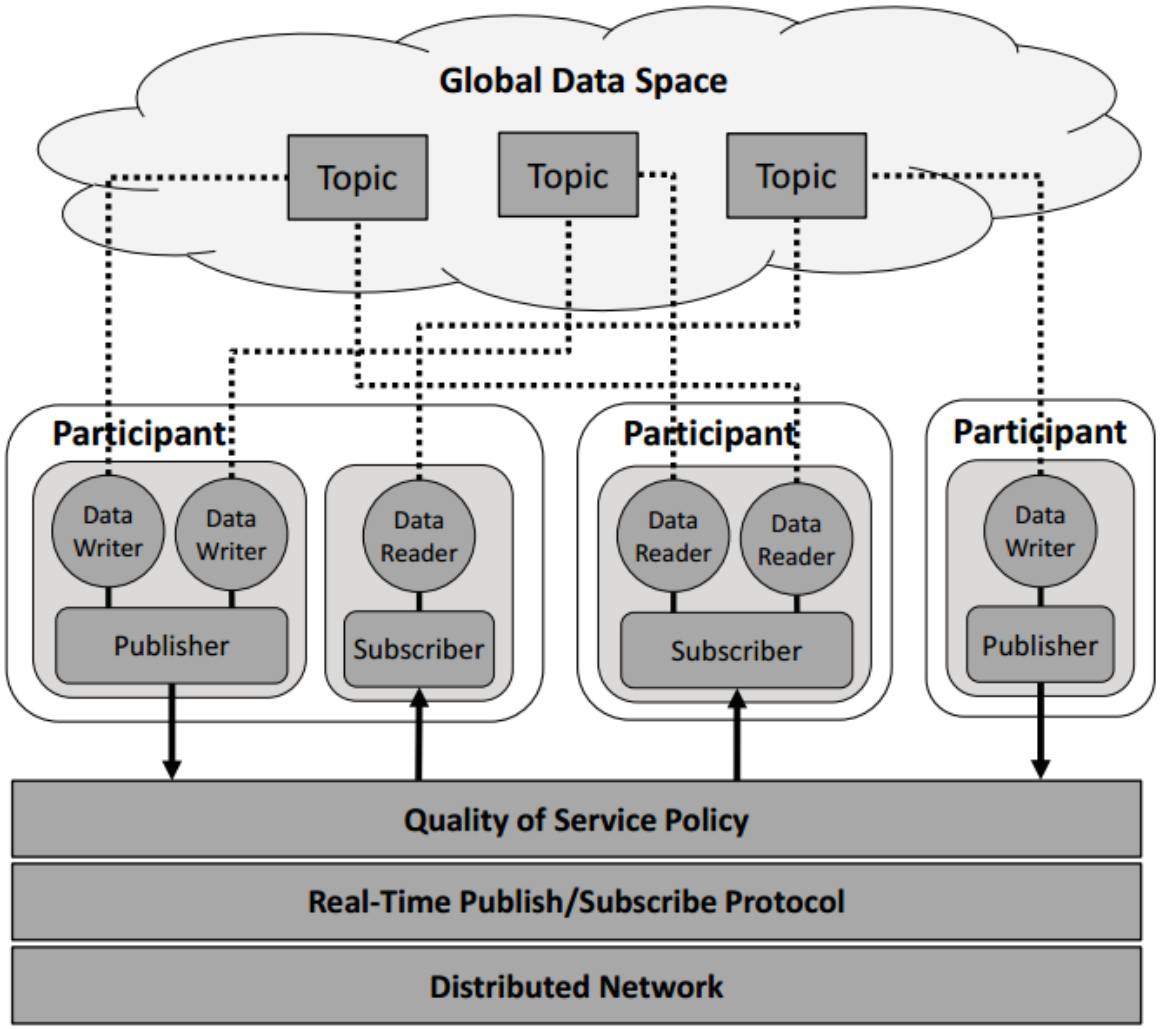
3.订阅者（Subscriber）：数据订阅的执行者，支持多种数据类型的订阅，可以与多个数据读取器（DataReader）相联，订阅一种或多种主题（Topic）的消息。

4.数据写入器（DataWriter）：应用向发布者更新数据的对象，每个数据写入器对应一个特定的Topic，类似于ROS1中的一个消息发布者。

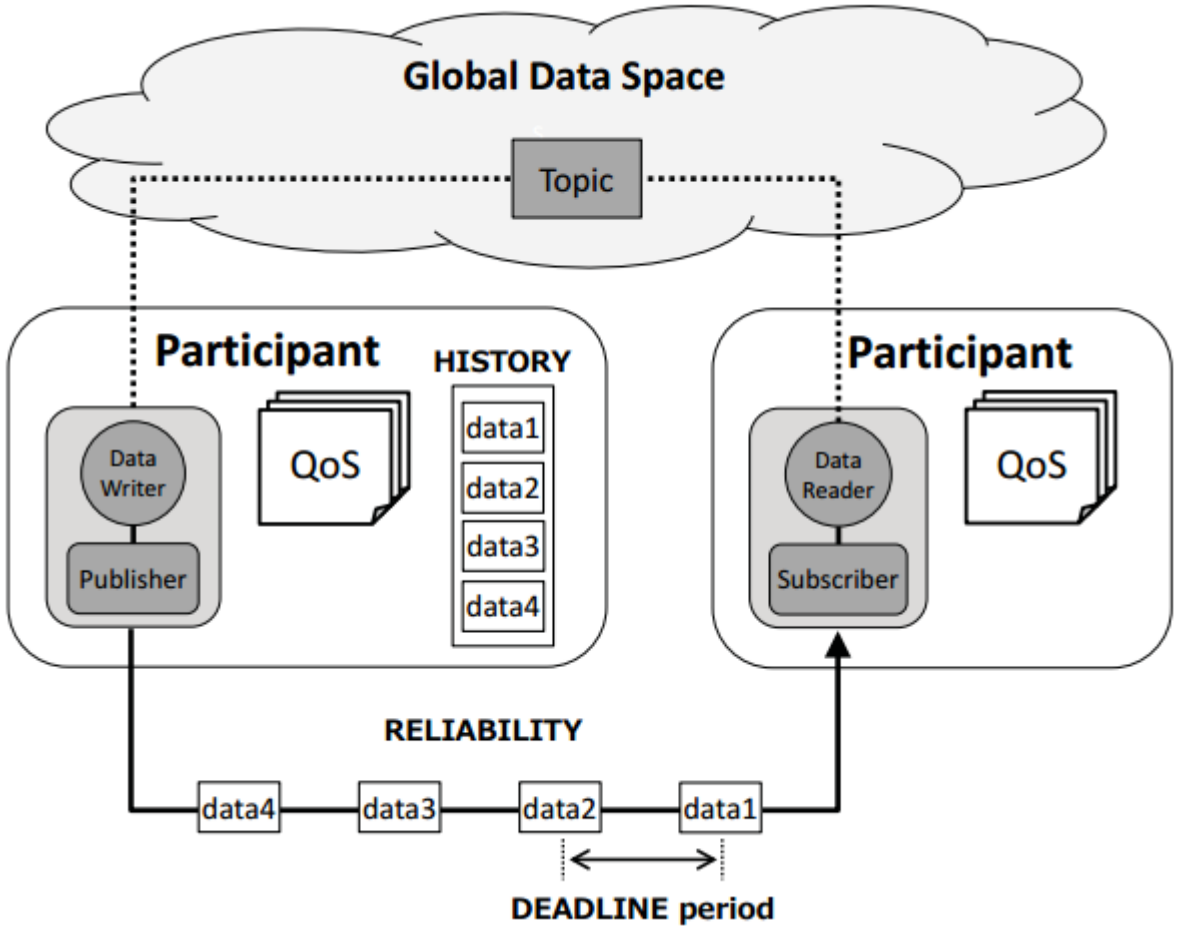
5.数据读取器（DataReader）：应用从订阅者读取数据的对象，每个数据读取器对应一个特定的Topic，类似于ROS1中的一个消息订阅者。

6.主题（Topic）：这个和ROS1中的Topic概念一致，一个Topic包含一个名称和一种数据结构。

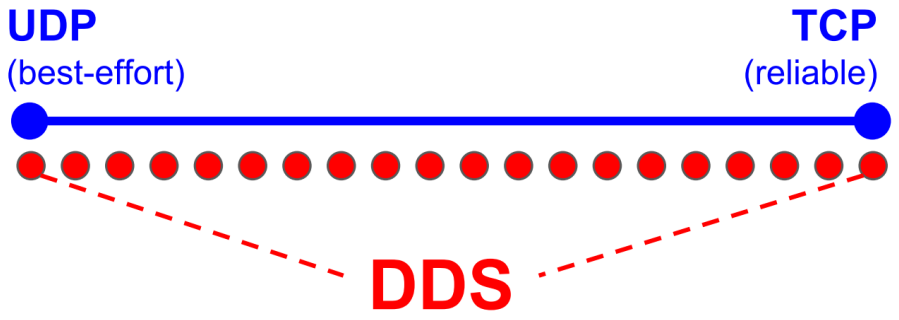
7.Quality of Service：质量服务原则，简称QoS Policy，控制各方面与底层的通讯机制，主要从时间限制、可靠性、持续性、历史记录几个方面，满足用户针对不同场景的数据应用需求。



What-ROS2的QoS



DEADLINE	A *DataWriter* and a *DataReader* must update data at least once every deadline period.
HISTORY	This controls whether the data transport should deliver only the most recent value, attempt to deliver all intermediate values, or attempt to deliver something in between (configurable via the `depth` option).
RELIABILITY	In `BEST_EFFORT`, data transport is executed as soon as possible. However, some data may be lost if the network is not robust. In `RELIABLE`, missed samples are retransmitted. Therefore, data delivery is guaranteed.
DURABILITY	With this policy, the service attempts to keep several samples so that they can be delivered to any potential late-joining *DataReader*. The number of saved samples depends on HISTORY. This option has several values, such as `VOLATILE` and `TRANSIENT_LOCAL`.



What-ROS2的QoS

QoS的数据结构

```
typedef struct RMW_PUBLIC_TYPE rmw_qos_profile_t
{
  enum rmw_qos_history_policy_t history;
  size_t depth;
  enum rmw_qos_reliability_policy_t reliability;
  enum rmw_qos_durability_policy_t durability;
} rmw_qos_profile_t;
```

- 每个原则都有对应的系统默认值
- 可以使用DDS厂商提供的配置工具修改QoS的设置
- 多商家的DDS可以并存

What-ROS2的编译系统

➢ 功能

ament是一种元编译系统，用来构建组成应用程序的多个独立功能包，catkin编译系统进一步演化的版本。

柔荑花序

➢ 组成

- 编译系统：配置、编译、安装独立的功能包
- 构建工具：将多个独立的功能包按照一定的拓扑结构进行链接

ament还处于开发中，目前并不稳定，也不提供并行构建的能力，将来应该会加入

What-ROS2的编译系统

- CMake centric

  catkin系统以 CMake为中心，所以只包含python代码的功能包也需要由CMake进行处理，但是CMake并不支持Python setuptools中的所有功能，而且也很难在Window上进行移植。

- Devel space

  在catkin系统构建完成后，会在工作目录下生成一个devel文件夹，里边是编译好的功能包，以及环境变量的设置等等，基本上等同于ROS安装完成后的目录结构和作用。但是相信很多初学者因为devel中的环境变量而苦恼过，这确实为用户带来了一些不必要的麻烦。

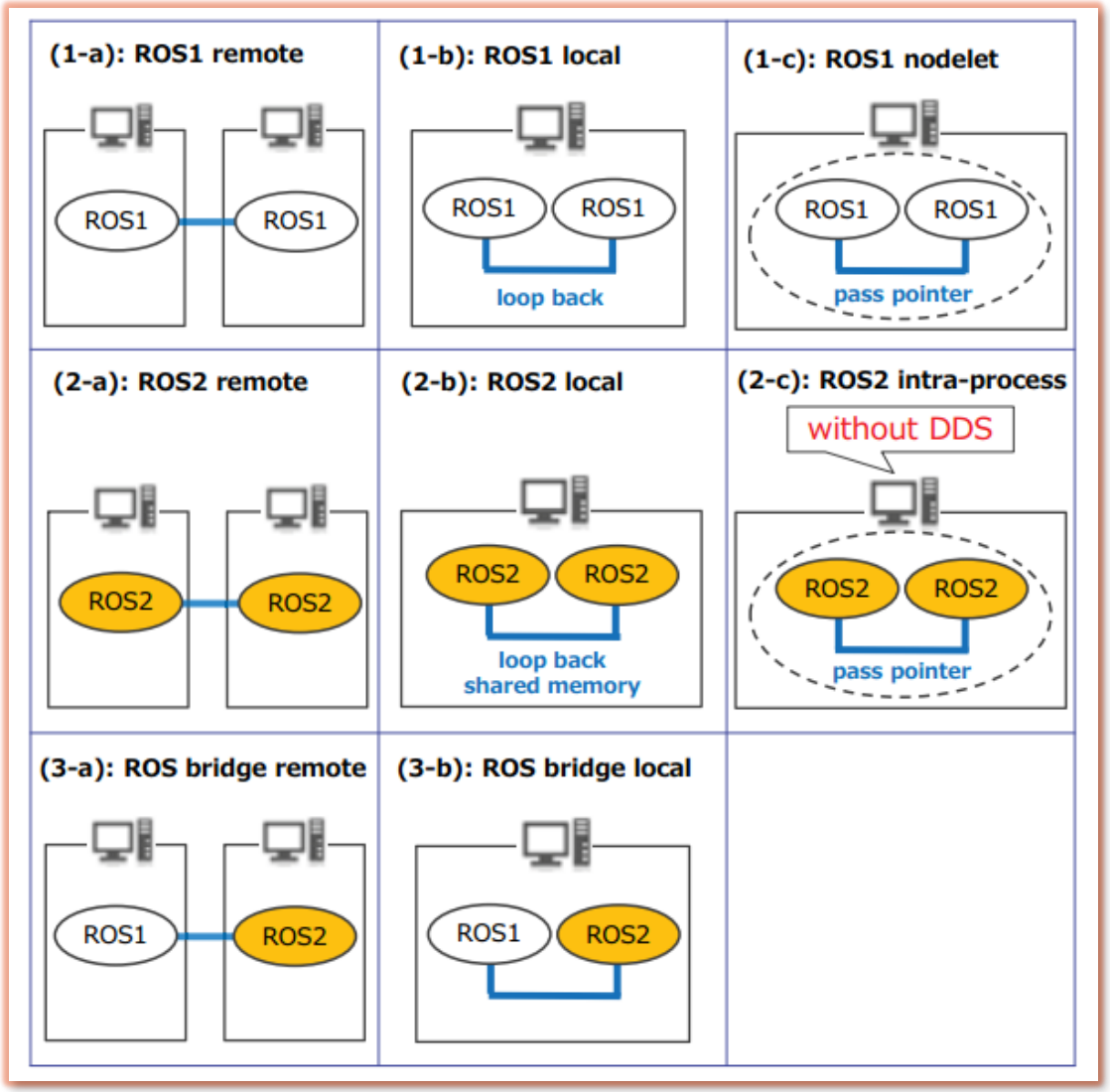
- catkin_simple

  catkin_simple是一个用于改善用户catkin体验的工具包，可以减少复杂的CMake代码，但是会存在不稳定的情况。ament也是实现了类似的功能，但是可靠性更强。

- Building within a single CMake context

  使用catkin_make命令可以一次性编译工作空间中的所有功能包，虽然方便，但如果存在相同命名的功能包时，会编译失败，ament在这方面也进行了改善。

What-ROS2的性能



"Exploring the Performance of ROS2"

What-ROS2的性能

Table 4: Capabilities of ROS1 and/or ROS2 for each Data Transport

				Initial loss	256 [byte]	512	1K	··· *	64K	128K	256K	512K	1M	2M	4M
ROS1	(1-a) remote			any	✓	✓	✓	···	✓	✓	✓	✓	✓	△[1]	△[1]
	(1-b) local			any	✓	✓	✓	···		✓	✓	✓	✓	✓	✓
	(1-c) nodelet			none	✓	✓	✓	···	✓	✓	✓	✓	✓	✓	✓
ROS2	(2-a) remote	Connext	reliable	none	✓	✓	✓	···	✓	▲[2]	▲[2]	▲[2]	▲[2]	▲[2]	▲[2]
			best-effort	none	✓	✓	✓	···	✓	✓	✓	✓	✓	△[1]	△[1]
		OpenSplice	reliable	none	✓	✓	✓	···	✓	✓	✓	✓	✓	✓	✓
			best-effort	none	✓	✓	✓	···	✓	✓	✓	✓	✓	△[1]	△[1]
		FastRTPS		none	▲[3]	▲[3]	▲[3]	···	▲[3]	▲[3]	▲[3]	▲[3]	▲[3]	▲[3]	▲[3]
	(2-b) local	Connext	reliable	none	✓	✓	✓	···	✓	▲[2]	▲[2]	▲[2]	▲[2]	▲[2]	▲[2]
			best-effort	none	✓	✓	✓	···	✓	✓	✓	✓	✓	△[2]	▲[1]
		OpenSplice	reliable	none	✓	✓	✓	···	✓	✓	✓	✓	✓	✓	✓
			best-effort	none	✓	✓	✓	···	✓	✓	✓	✓	✓	△[2]	△[2]
		FastRTPS		none	▲[3]	▲[3]	▲[3]	···	▲[3]	▲[3]	▲[3]	▲[3]	▲[3]	▲[3]	▲[3]
	(2-c) intra-process			none	✓	✓	✓	···	✓	✓	✓	✓	✓	✓	✓
ROS1 to 2	(3-a) remote	Connext		any	✓	✓	✓	···	✓	✓	✓	✓	✓	▲[1]	▲[1]
		OpenSplice		any	✓	✓	✓	···	✓	✓	✓	✓	✓	△[1]	△[1]
	(3-b) local	Connext		any	✓	✓	✓	···	✓	✓	✓	✓	✓	▲[1]	▲[1]
		OpenSplice		any	✓	✓	✓	···	✓	✓	✓	✓	✓	✓	✓
ROS2 to 1	(3-a) remote	Connext		any	✓	✓	✓	···	✓	✓	✓	✓	✓	▲[1]	▲[1]
		OpenSplice		any	✓	✓	✓	···	✓	✓	▲[1]	▲[1]	▲[1]	▲[1]	▲[1]
	(3-b) local	Connext		any	✓	✓	✓	···	✓	✓	✓	✓	✓	▲[1]	▲[1]
		OpenSplice		any	✓	✓	✓	···	✓	✓	△[1]	▲[1]	▲[1]	▲[1]	▲[1]

*: same behavior as 1 and 64 KB; ✓: data transport possible; △[1]: possible but missing the deadline; △[2]: data loss possible;
▲[1]: impossible due to a halt of process or too much data loss;
▲[2]: impossible with an error message (deficiency of additional configurations for large data);
▲[3]: impossible with an error message (unsupported large data for the DDS implementation)

What-ROS2的性能

- 不同数据量的场景下，表现呈指数级变化
- 不同厂商的DDS产品性能表现，相差巨大，需要根据不同的应用场景选用最佳的DDS产品
- QoS提供了多种配置选项，同样需要根据不同的应用场景，选择最适合的配置。
- ROS1会丢失初始数据，ROS2在这一点表现较好
- ROS2对DDS的支持有限，有待开发
- ROS2目前的整体性能（Alpha版本）并不如ROS1，还处于开发阶段

DDS Vendors

Company and product name	License	RMW impl.	Comments
RTI Connext	commercial, research	✔	stat. & dyn. impl.
PrismTech OpenSplice	commercial, LGPL	✔	only version 6.4 is LGPL
TwinOaks CoreDX	commercial	–	
eProsima FastRTPS	LGPL	✔	no support for fragmentation yet
OSRF FreeRTPS	Apache 2	partial	small part of DDS only aiming for emb. devices

How-ROS2的安装(Beta2版本)

添加源

```
sudo apt-get update && sudo apt-get install curl
curl http://repo.ros2.org/repos.key | sudo apt-key add –
sudo sh -c 'echo "deb http://repo.ros2.org/ubuntu/main xenial main" > /etc/apt/sources.list.d/ros2-latest.list'
```

安装

```
sudo apt-get update
sudo apt-get install ros-r2b2-*
```

环境变量

```
source /opt/ros/r2b2/setup.bash
```

安装功能包

```
sudo apt-get install ros-r2b2-ros1-bridge ros-r2b2-turtlebot2-*
```

How-ROS2的安装(Beta2版本)

```
Get:188 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-ceres-solver amd64 1.12.0-2xenial-20170701-001306-0800 [759 kB]
Get:189 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-cartographer amd64 0.1.0-3xenial-20170701-004440-0800 [825 kB]
Get:190 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-cartographer-ros-msgs amd64 0.1.0-4xenial-20170705-113631-0800 [60.9 kB]
Get:191 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-cartographer-ros amd64 0.1.0-4xenial-20170705-121055-0800 [1,056 kB]
Get:192 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-class-loader amd64 1.0.0-8xenial-20170701-051206-0800 [39.2 kB]
Get:193 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-example-interfaces amd64 0.0.2-0xenial-20170705-111158-0800 [22.3 kB]
Get:194 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-composition amd64 0.0.2-1xenial-20170705-113151-0800 [119 kB]
Get:195 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-demo-nodes-cpp amd64 0.0.2-1xenial-20170705-114821-0800 [136 kB]
Get:196 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-rclpy amd64 0.0.2-0xenial-20170705-112857-0800 [24.4 kB]
Get:197 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-demo-nodes-py amd64 0.0.2-1xenial-20170705-114648-0800 [7,590 B]
Get:198 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-image-geometry amd64 1.12.4-2xenial-20170705-121740-0800 [27.7 kB]
Get:199 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-depth-to-pointcloud amd64 0.0.2-0xenial-20170705-122218-0800 [30.6 kB]
Get:200 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-depthimage-to-laserscan amd64 1.0.7-3xenial-20170705-122219-0800 [48.0 kB]
Get:201 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-diagnostic-msgs amd64 0.0.2-0xenial-20170705-113925-0800 [43.3 kB]
Get:202 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-dummy-map-server amd64 0.0.2-1xenial-20170705-120049-0800 [18.1 kB]
Get:203 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-launch amd64 0.0.2-0xenial-20170701-051229-0800 [18.2 kB]
Get:204 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-ros2cli amd64 0.0.2-0xenial-20170705-121127-0800 [19.7 kB]
Get:205 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-ros2pkg amd64 0.0.2-0xenial-20170705-121947-0800 [6,908 B]
Get:206 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-ros2run amd64 0.0.2-0xenial-20170705-122300-0800 [5,482 B]
Get:207 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-dummy-robot-bringup amd64 0.0.2-1xenial-20170705-123001-0800 [6,750 B]
Get:208 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-dummy-sensors amd64 0.0.2-1xenial-20170705-120113-0800 [27.3 kB]
Get:209 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclcpp-minimal-client amd64 0.0.2-2xenial-20170705-113615-0800 [18.4 kB]
Get:210 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclcpp-minimal-composition amd64 0.0.2-2xenial-20170705-115158-0800 [40.5 kB]
Get:211 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclcpp-minimal-publisher amd64 0.0.2-2xenial-20170705-115213-0800 [26.3 kB]
Get:212 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclcpp-minimal-service amd64 0.0.2-2xenial-20170705-114626-0800 [13.6 kB]
Get:213 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclcpp-minimal-subscriber amd64 0.0.2-2xenial-20170705-115416-0800 [29.9 kB]
Get:214 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclcpp-minimal-timer amd64 0.0.2-2xenial-20170705-120149-0800 [10.9 kB]
Get:215 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclpy-minimal-client amd64 0.0.2-2xenial-20170705-114726-0800 [5,258 B]
Get:216 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclpy-minimal-publisher amd64 0.0.2-2xenial-20170705-115624-0800 [5,456 B]
Get:217 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclpy-minimal-service amd64 0.0.2-2xenial-20170705-114903-0800 [4,632 B]
Get:218 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-examples-rclpy-minimal-subscriber amd64 0.0.2-2xenial-20170705-115657-0800 [4,862 B]
Get:219 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-image-tools amd64 0.0.2-1xenial-20170705-115517-0800 [50.7 kB]
Get:220 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-intra-process-demo amd64 0.0.2-1xenial-20170705-115520-0800 [94.6 kB]
Get:221 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-joy amd64 1.11.0-2xenial-20170705-120431-0800 [22.3 kB]
Get:222 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-tinyxml-vendor amd64 0.0.2-0xenial-20170701-050838-0800 [2,832 B]
Get:223 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-urdfdom-headers amd64 1.0.0-1xenial-20170701-051045-0800 [11.2 kB]
Get:224 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-urdfdom amd64 0.3.0-5xenial-20170701-051421-0800 [72.4 kB]
Get:225 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-urdf amd64 1.12.7-5xenial-20170701-052921-0800 [16.5 kB]
Get:226 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-kdl-parser amd64 1.12.7-5xenial-20170701-053250-0800 [18.0 kB]
Get:227 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-launch-testing amd64 0.0.2-0xenial-20170701-051549-0800 [5,590 B]
Get:228 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-lifecycle-msgs amd64 0.0.2-0xenial-20170705-111201-0800 [58.6 kB]
Get:229 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-rcl-lifecycle amd64 0.0.2-0xenial-20170705-112748-0800 [16.3 kB]
Get:230 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-rclcpp-lifecycle amd64 0.0.2-0xenial-20170705-113208-0800 [43.8 kB]
Get:231 http://repo.ros2.org/ubuntu/main xenial/main amd64 ros-r2b2-lifecycle amd64 0.0.2-1xenial-20170705-122348-0800 [53.0 kB]
95% [231 ros-r2b2-lifecycle 28.5 kB/53.0 kB 54%]                                                                29.2 kB/s 1min 3s
```

正在安装...

星河智能 Algalaxy

How-ROS2的安装(Beta2版本)

```
➜  ~ sudo apt-get install ros-r2b2-
ros-r2b2-actionlib-msgs                          ros-r2b2-console-bridge                          ros-r2b2-robot-state-publisher
ros-r2b2-amcl                                    ros-r2b2-demo-nodes-cpp                          ros-r2b2-ros1-bridge
ros-r2b2-ament-clang-format                      ros-r2b2-demo-nodes-py                           ros-r2b2-ros2cli
ros-r2b2-ament-cmake                             ros-r2b2-depthimage-to-laserscan                 ros-r2b2-ros2msg
ros-r2b2-ament-cmake-auto                        ros-r2b2-depth-to-pointcloud                     ros-r2b2-ros2node
ros-r2b2-ament-cmake-clang-format                ros-r2b2-diagnostic-msgs                         ros-r2b2-ros2pkg
ros-r2b2-ament-cmake-copyright                   ros-r2b2-dummy-map-server                        ros-r2b2-ros2run
ros-r2b2-ament-cmake-core                        ros-r2b2-dummy-robot-bringup                     ros-r2b2-ros2service
ros-r2b2-ament-cmake-cppcheck                    ros-r2b2-dummy-sensors                           ros-r2b2-ros2srv
ros-r2b2-ament-cmake-cpplint                     ros-r2b2-example-interfaces                      ros-r2b2-ros2topic
ros-r2b2-ament-cmake-export-definitions          ros-r2b2-examples-rclcpp-minimal-client          ros-r2b2-rosidl-cmake
ros-r2b2-ament-cmake-export-dependencies         ros-r2b2-examples-rclcpp-minimal-composition     ros-r2b2-rosidl-default-generators
ros-r2b2-ament-cmake-export-include-directories  ros-r2b2-examples-rclcpp-minimal-publisher       ros-r2b2-rosidl-default-runtime
ros-r2b2-ament-cmake-export-interfaces           ros-r2b2-examples-rclcpp-minimal-service         ros-r2b2-rosidl-generator-c
ros-r2b2-ament-cmake-export-libraries            ros-r2b2-examples-rclcpp-minimal-subscriber      ros-r2b2-rosidl-generator-cpp
ros-r2b2-ament-cmake-export-link-flags           ros-r2b2-examples-rclcpp-minimal-timer           ros-r2b2-rosidl-generator-dds-idl
ros-r2b2-ament-cmake-flake8                      ros-r2b2-examples-rclpy-minimal-client           ros-r2b2-rosidl-generator-py
ros-r2b2-ament-cmake-gmock                       ros-r2b2-examples-rclpy-minimal-publisher        ros-r2b2-rosidl-parser
ros-r2b2-ament-cmake-gtest                       ros-r2b2-examples-rclpy-minimal-service          ros-r2b2-rosidl-typesupport-c
ros-r2b2-ament-cmake-include-directories         ros-r2b2-examples-rclpy-minimal-subscriber       ros-r2b2-rosidl-typesupport-cpp
ros-r2b2-ament-cmake-libraries                   ros-r2b2-fastcdr                                 ros-r2b2-rosidl-typesupport-interface
ros-r2b2-ament-cmake-lint-cmake                  ros-r2b2-fastrtps                                ros-r2b2-rosidl-typesupport-introspection-c
ros-r2b2-ament-cmake-nose                        ros-r2b2-fastrtps-cmake-module                   ros-r2b2-rosidl-typesupport-introspection-cpp
ros-r2b2-ament-cmake-pep257                      ros-r2b2-geometry-msgs                           ros-r2b2-ros-workspace
ros-r2b2-ament-cmake-pep8                        ros-r2b2-gmock-vendor                            ros-r2b2-rttest
ros-r2b2-ament-cmake-pyflakes                    ros-r2b2-gtest-vendor                            ros-r2b2-sensor-msgs
ros-r2b2-ament-cmake-python                      ros-r2b2-image-geometry                          ros-r2b2-shape-msgs
ros-r2b2-ament-cmake-ros                         ros-r2b2-image-tools                             ros-r2b2-sros2
ros-r2b2-ament-cmake-target-dependencies         ros-r2b2-intra-process-demo                      ros-r2b2-std-msgs
ros-r2b2-ament-cmake-test                        ros-r2b2-joy                                     ros-r2b2-std-srvs
ros-r2b2-ament-cmake-uncrustify                  ros-r2b2-kdl-parser                              ros-r2b2-stereo-msgs
ros-r2b2-ament-copyright                         ros-r2b2-launch                                  ros-r2b2-teleop-twist-joy
ros-r2b2-ament-cppcheck                          ros-r2b2-launch-testing                          ros-r2b2-teleop-twist-keyboard
ros-r2b2-ament-cpplint                           ros-r2b2-lifecycle                               ros-r2b2-test-communication
ros-r2b2-ament-flake8                            ros-r2b2-lifecycle-msgs                          ros-r2b2-test-rclcpp
ros-r2b2-ament-index-cpp                         ros-r2b2-map-server                              ros-r2b2-tf2
ros-r2b2-ament-index-python                      ros-r2b2-nav-msgs                                ros-r2b2-tf2-eigen
ros-r2b2-ament-lint-auto                         ros-r2b2-orocos-kdl                              ros-r2b2-tf2-geometry-msgs
ros-r2b2-ament-lint-cmake                        ros-r2b2-osrf-pycommon                           ros-r2b2-tf2-msgs
ros-r2b2-ament-lint-common                       ros-r2b2-pcl-conversions                         ros-r2b2-tf2-ros
ros-r2b2-ament-package                           ros-r2b2-pendulum-control                        ros-r2b2-tinyxml-vendor
ros-r2b2-ament-pep257                            ros-r2b2-pendulum-msgs                           ros-r2b2-tlsf
ros-r2b2-ament-pep8                              ros-r2b2-poco-vendor                             ros-r2b2-tlsf-cpp
ros-r2b2-ament-pyflakes                          ros-r2b2-python-cmake-module                     ros-r2b2-topic-monitor
```

安装功能包

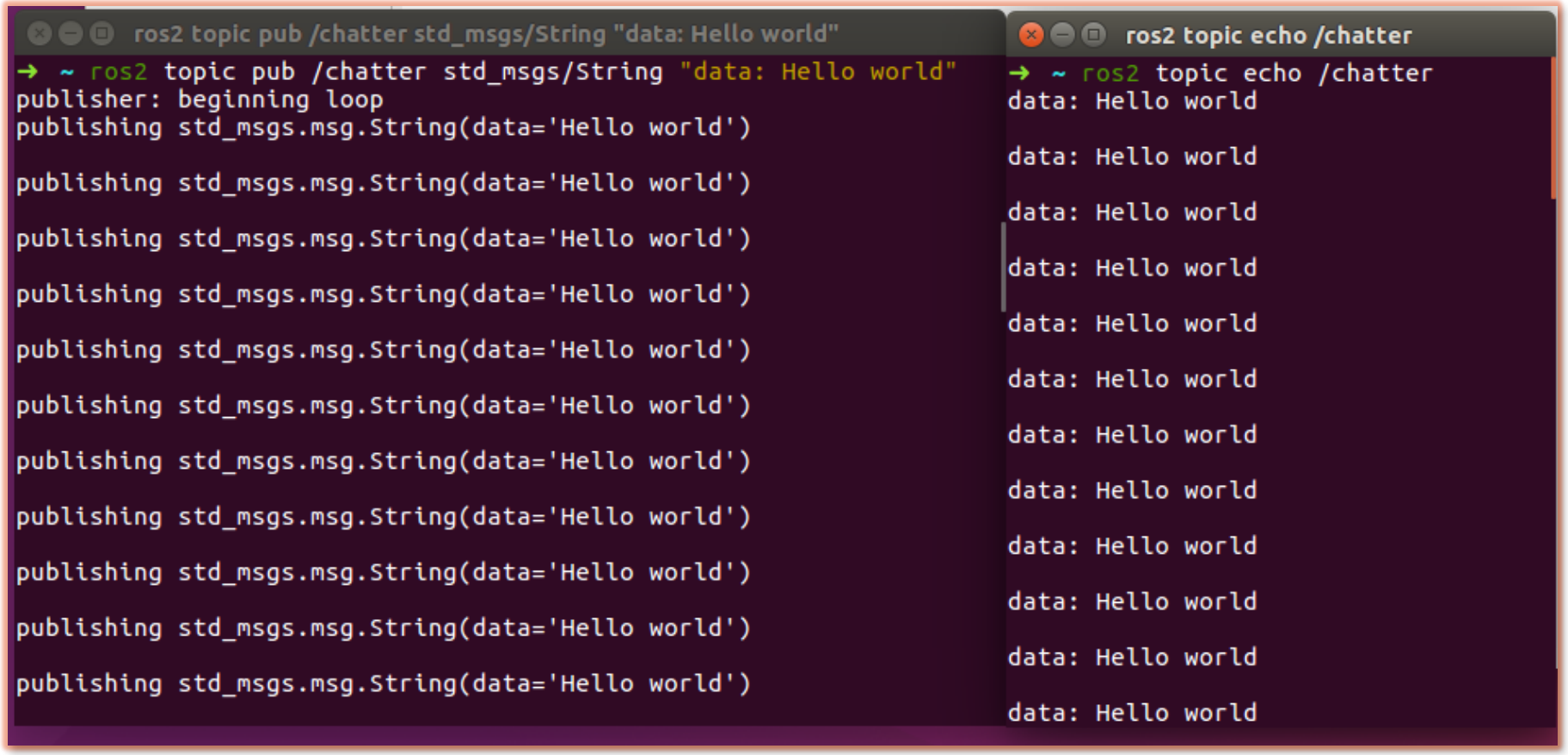
How-ROS2的命令工具

ROS2的命令帮助

source /opt/ros/r2b2/setup.bash

```
→ ~ ros2 --help
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  daemon    Various daemon related sub-commands
  msg       Various msg related sub-commands
  node      Various node related sub-commands
  pkg       Various package related sub-commands
  run       Run a package specific executable
  security  Various security related sub-commands
  service   Various service related sub-commands
  srv       Various srv related sub-commands
  topic     Various topic related sub-commands

  Call `ros2 <command> -h` for more detailed usage.
```

How-ROS2的命令工具



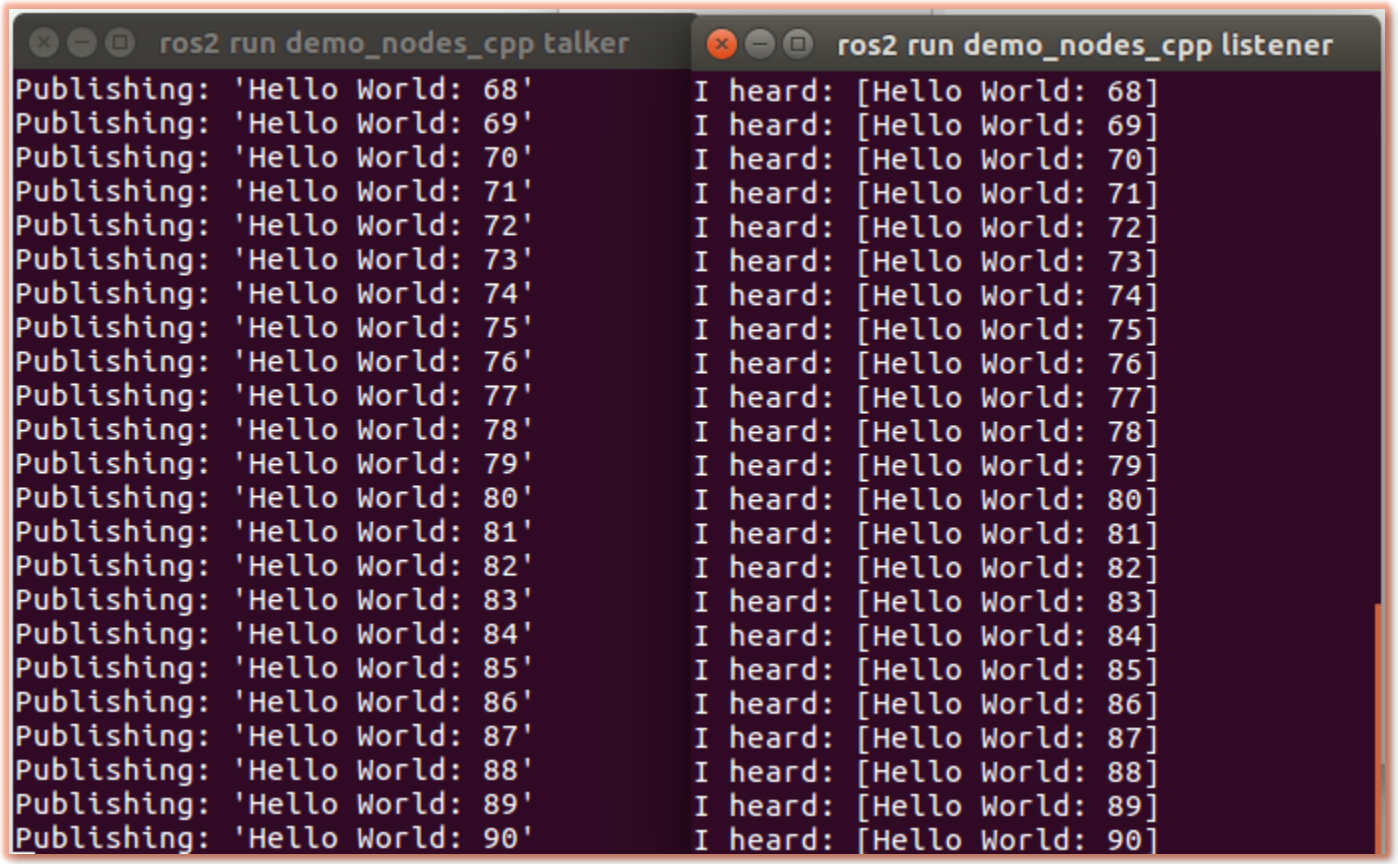
- ros2 node == rosnode（ROS1）
  - list
  - ….

- ros2 topic == rostopic（ROS1）
  - list
  - pub
  - echo
  - ….

How-ROS2的通讯测试

Publisher ros2 run demo_nodes_cpp talker

Subscriber ros2 run demo_nodes_cpp listener



How-创建工作目录

创建工作目录

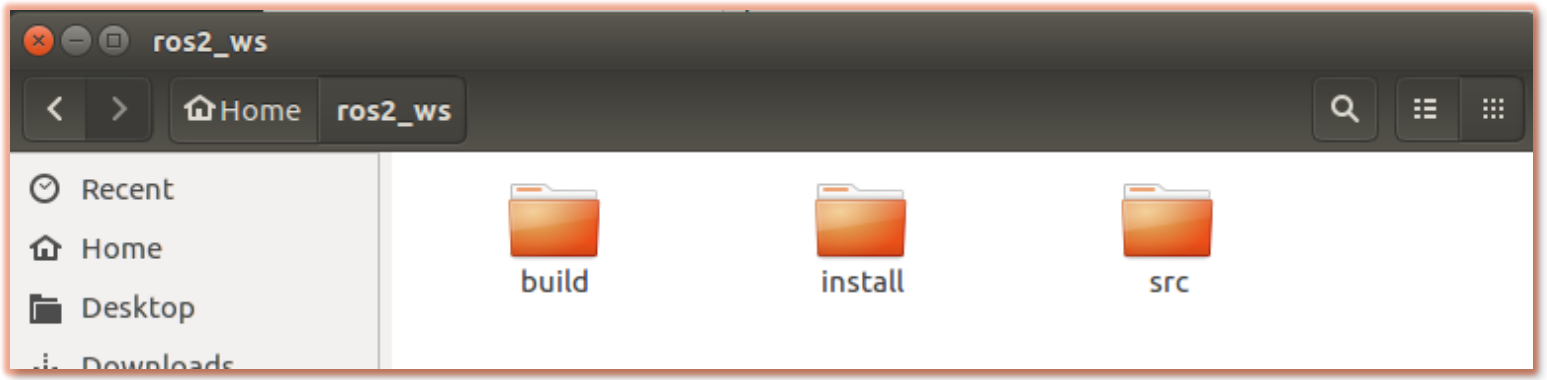
```
mkdir -p ~/ros2_ws/src
cd ~/ros2_ws
```

创建功能包

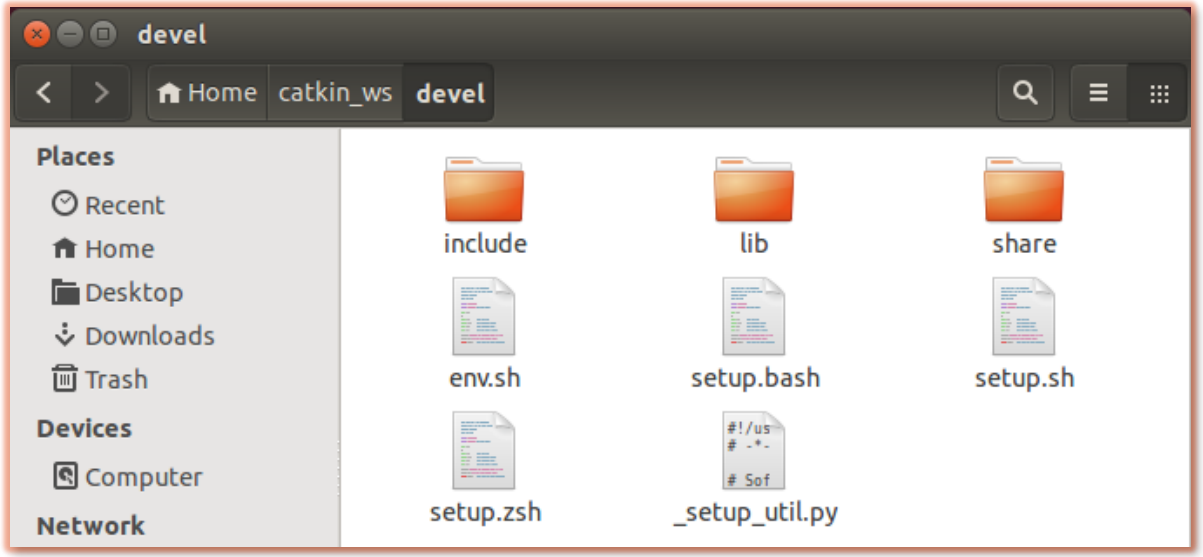
```
sudo apt-get update
sudo apt-get install ros-r2b2-*
```

编译

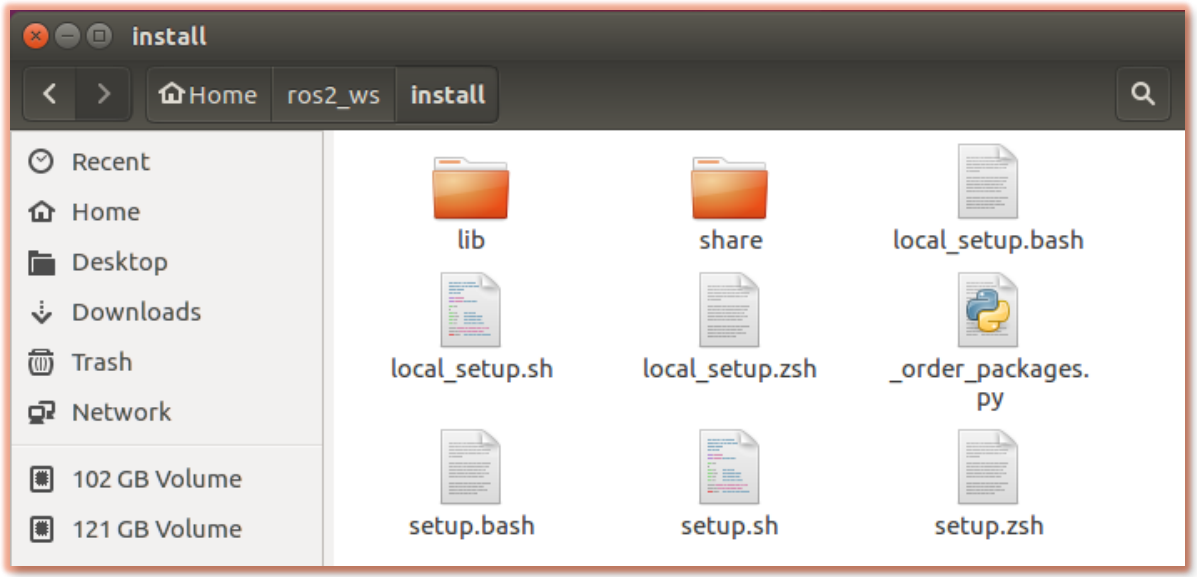
```
ament build
```



How-创建工作目录



devel文件夹（ROS1）



install文件夹（ROS2）

How-实现一个talker

```cpp
#include <iostream>
#include <memory>

#include "rclcpp/rclcpp.hpp"
#include "std_msgs/msg/string.hpp"

int main(int argc, char * argv[])
{
  //ros::init(argc, argv, "talker");
  rclcpp::init(argc, argv);

  //ros::NodeHandle n;
  auto node = rclcpp::node::Node::make_shared("talker");

  rmw_qos_profile_t custom_qos_profile = rmw_qos_profile_default;
  custom_qos_profile.depth = 7;

  //ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
  auto chatter_pub = node->create_publisher<std_msgs::msg::String>("chatter", custom_qos_profile);

  //ros::Rate loop_rate(10);
  rclcpp::WallRate loop_rate(2);

  auto msg = std::make_shared<std_msgs::msg::String>();
  auto i = 1;

  //while (ros::ok())
  while (rclcpp::ok()) {
    msg->data = "Hello World: " + std::to_string(i++);
    std::cout << "Publishing: '" << msg->data << "'" << std::endl;

    //chatter_pub.publish(msg);
    chatter_pub->publish(msg);

    //ros::spinOnce();
    rclcpp::spin_some(node);

    //loop_rate.sleep();
    loop_rate.sleep();
  }

  return 0;
}
```

注释是对应于ROS1的实现API

How-实现一个listener

```cpp
#include <iostream>
#include <memory>

#include "rclcpp/rclcpp.hpp"
#include "std_msgs/msg/string.hpp"

//void chatterCallback(const std_msgs::String::ConstPtr& msg)
void chatterCallback(const std_msgs::msg::String::SharedPtr msg)
{
  std::cout << "I heard: [" << msg->data << "]" << std::endl;
}

int main(int argc, char * argv[])
{
  //ros::init(argc, argv, "listener");
  rclcpp::init(argc, argv);

  //ros::NodeHandle n;
  auto node = rclcpp::Node::make_shared("listener");

  //ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);
  auto sub = node->create_subscription<std_msgs::msg::String>(
    "chatter", chatterCallback, rmw_qos_profile_default);

  //ros::spin();
  rclcpp::spin(node);

  return 0;
}
```

注释是对应于ROS1的实现API

How-两代ROS的代码对比

- ROS2中的API相比ROS1中发生了较大的变化，ROS2并不是在ROS1的基础上查漏补缺，而是完全从新设计。

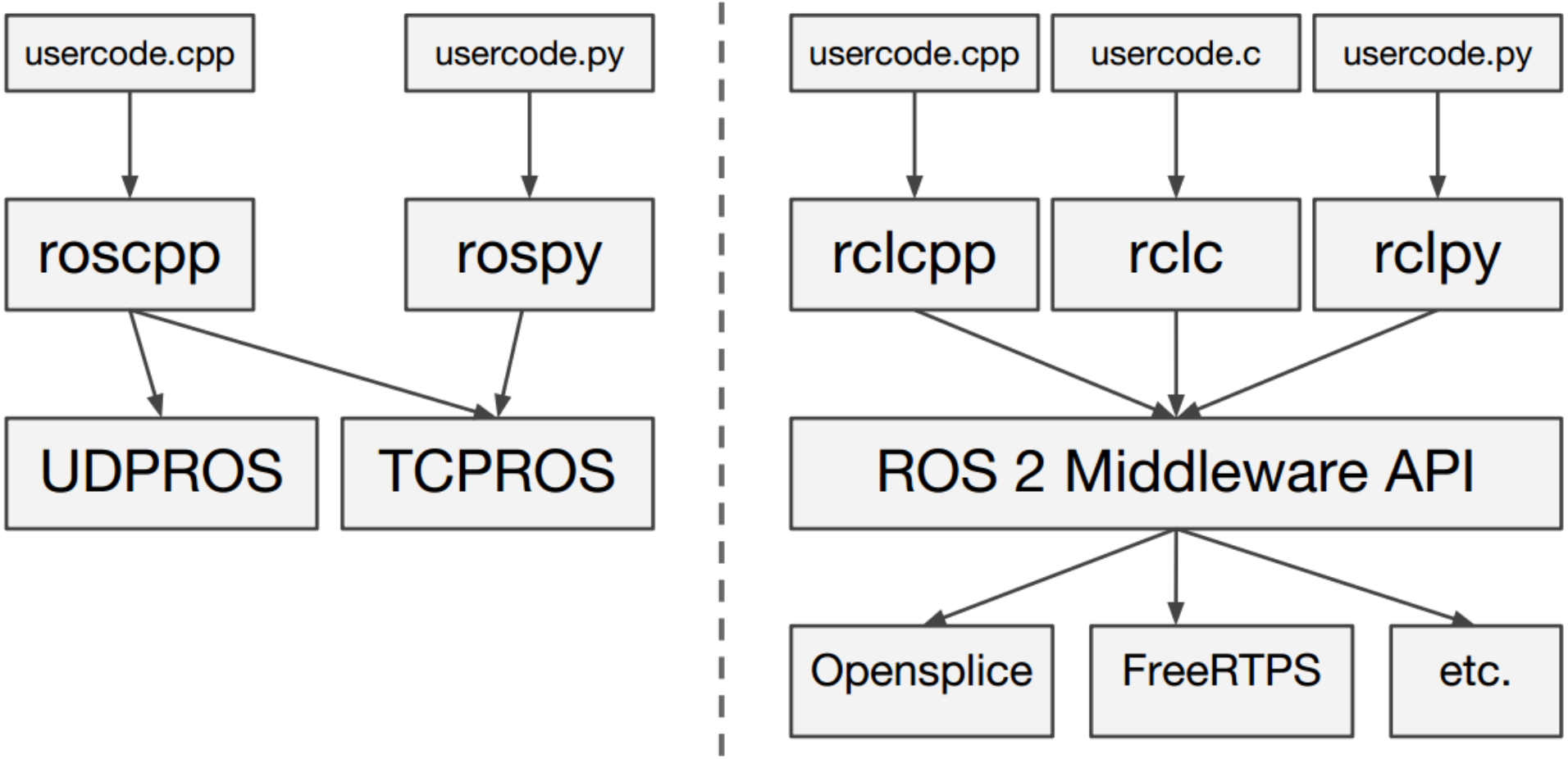
  关于ROS2的API说明，可以参考API文档：http://docs.ros2.org/beta1/api/rclcpp/index.html

- 使用了更多C++的特性，比如auto、make_shared等。

- 加入了QoS配置，从上边的代码中，我们可以看到QoS有默认的配置rmw_qos_profile_default ，而且talker将QoS的depth配置设置为“7”。

- 代码的总体架构还是与ROS1极为相似的。

How-两代ROS的代码对比



How-修改CMakeLists文件

```
cmake_minimum_required(VERSION 3.5)

project(demo_nodes_cpp)

# Default to C++14
if(NOT CMAKE_CXX_STANDARD)
  set(CMAKE_CXX_STANDARD 14)
endif()

if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
  add_compile_options(-Wall -Wextra -Wpedantic)
endif()

find_package(ament_cmake REQUIRED)
find_package(rclcpp REQUIRED)
find_package(std_msgs REQUIRED)

function(custom_executable target)
  add_executable(${target} src/${target}.cpp)
  ament_target_dependencies(${target}
    "rclcpp"
    "std_msgs")
  install(TARGETS ${target}
  DESTINATION lib/${PROJECT_NAME})
endfunction()

# Tutorials of Publish/Subscribe with Topics
custom_executable(talker)
custom_executable(listener)

ament_package()
```

CMakeLists文件的实现基本没变

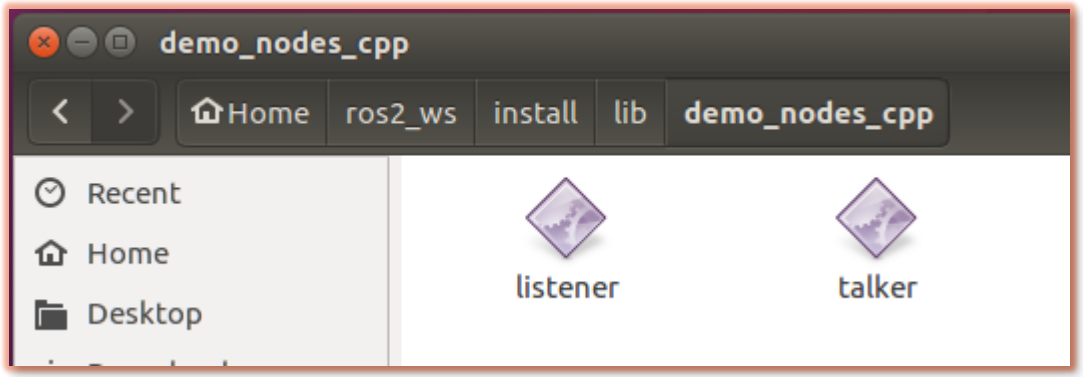
How-编译工作空间

```
hcx@hcx-pc: ~/ros2_ws
→ ros2_ws ament build
# Topological order
 - demo_nodes_cpp

Process package 'demo_nodes_cpp' with context:
--------------------------------------------------------------------------------
 source_space => /home/hcx/ros2_ws/src/demo_nodes_cpp
  build_space => /home/hcx/ros2_ws/build/demo_nodes_cpp
install_space => /home/hcx/ros2_ws/install
   make_flags => -j8, -l8
  build_tests => False
--------------------------------------------------------------------------------
+++ Building 'demo_nodes_cpp'
Running cmake because arguments have changed.
==> '. /home/hcx/ros2_ws/build/demo_nodes_cpp/cmake__build.sh && /usr/bin/cmake /home/hcx/ros2_ws/src/demo_nodes_cpp -DBUILD_TEST
ING=0 -DCMAKE_INSTALL_PREFIX=/home/hcx/ros2_ws/install' in '/home/hcx/ros2_ws/build/demo_nodes_cpp'
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found ament_cmake: 0.0.2 (/opt/ros/r2b2/share/ament_cmake/cmake)
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.5.2", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Found example_interfaces: 0.0.2 (/opt/ros/r2b2/share/example_interfaces/cmake)
-- Found rclcpp: 0.0.2 (/opt/ros/r2b2/share/rclcpp/cmake)
-- Found rmw_implementation_cmake: 0.0.2 (/opt/ros/r2b2/share/rmw_implementation_cmake/cmake)
-- Using RMW implementation 'rmw_fastrtps_cpp' as default
-- Found rmw_fastrtps_cpp: 0.0.2 (/opt/ros/r2b2/share/rmw_fastrtps_cpp/cmake)
-- Found fastrtps_cmake_module: 0.0.2 (/opt/ros/r2b2/share/fastrtps_cmake_module/cmake)
-- Found FastRTPS: /opt/ros/r2b2/include
-- Found sensor_msgs: 0.0.2 (/opt/ros/r2b2/share/sensor_msgs/cmake)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/hcx/ros2_ws/build/demo_nodes_cpp
==> '. /home/hcx/ros2_ws/build/demo_nodes_cpp/cmake__build.sh && /usr/bin/make -j8 -l8' in '/home/hcx/ros2_ws/build/demo_nodes_cp
p'
Scanning dependencies of target talker
Scanning dependencies of target listener
[ 25%] Building CXX object CMakeFiles/talker.dir/src/talker.cpp.o
[ 50%] Building CXX object CMakeFiles/listener.dir/src/listener.cpp.o
[ 75%] Linking CXX executable talker
[100%] Linking CXX executable listener
```

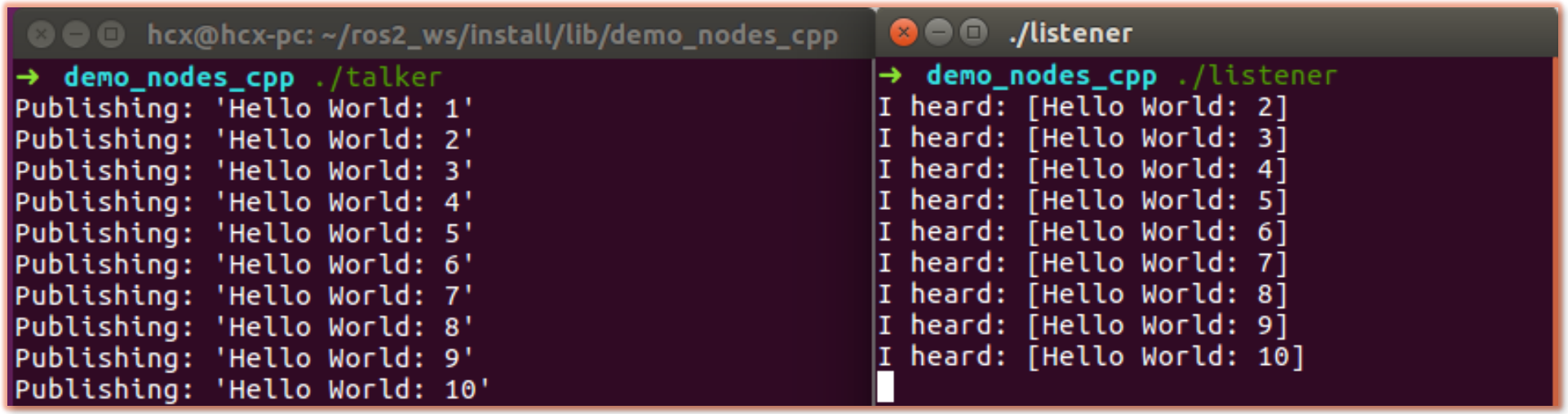
编译

ament build

How-运行可执行文件



打开install下的lib文件夹，即可找到编译成功的可执行文件。



How-ROS1与ROS2之间的桥梁

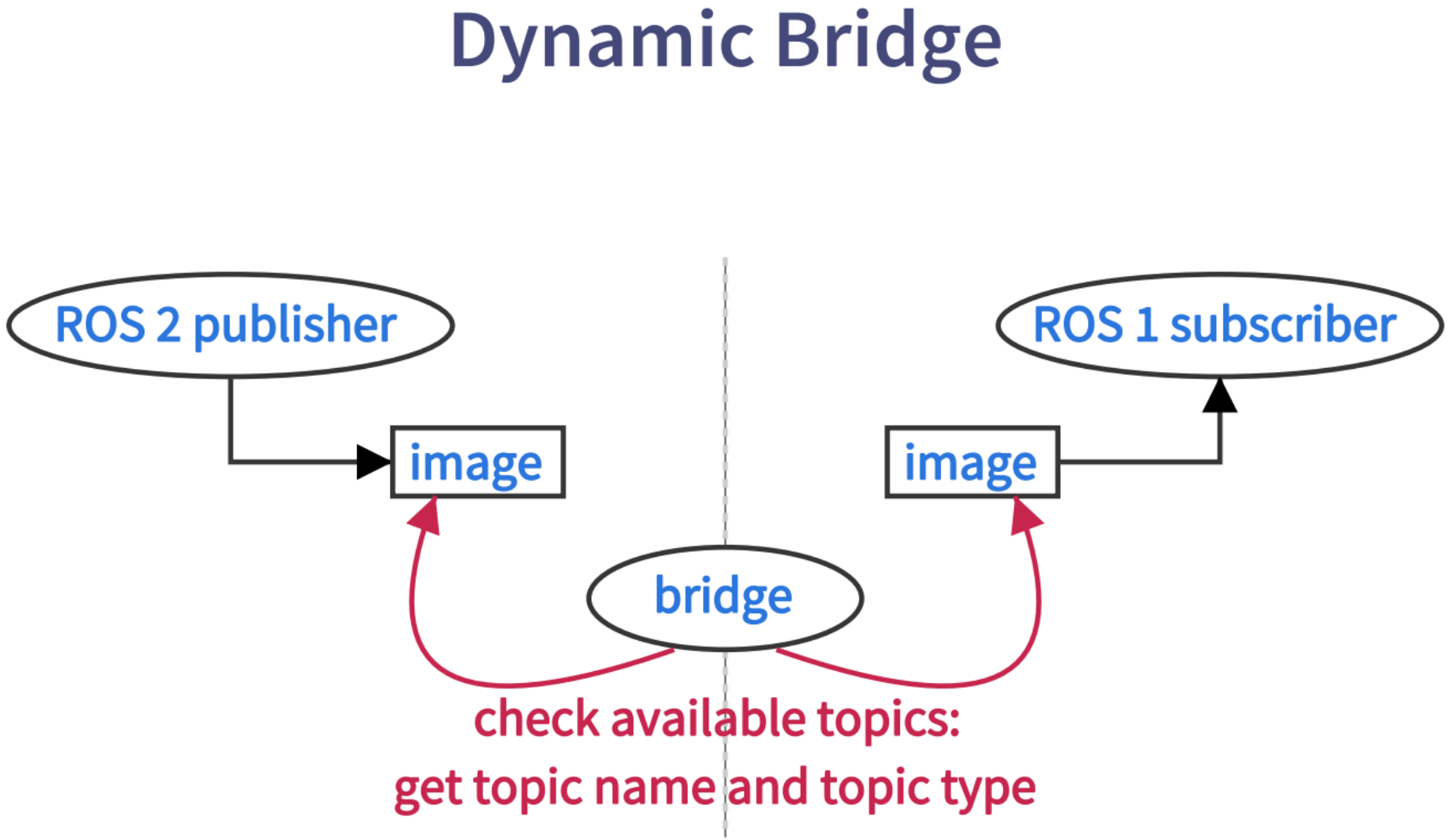
ROS 2

- New features
- Superior communication

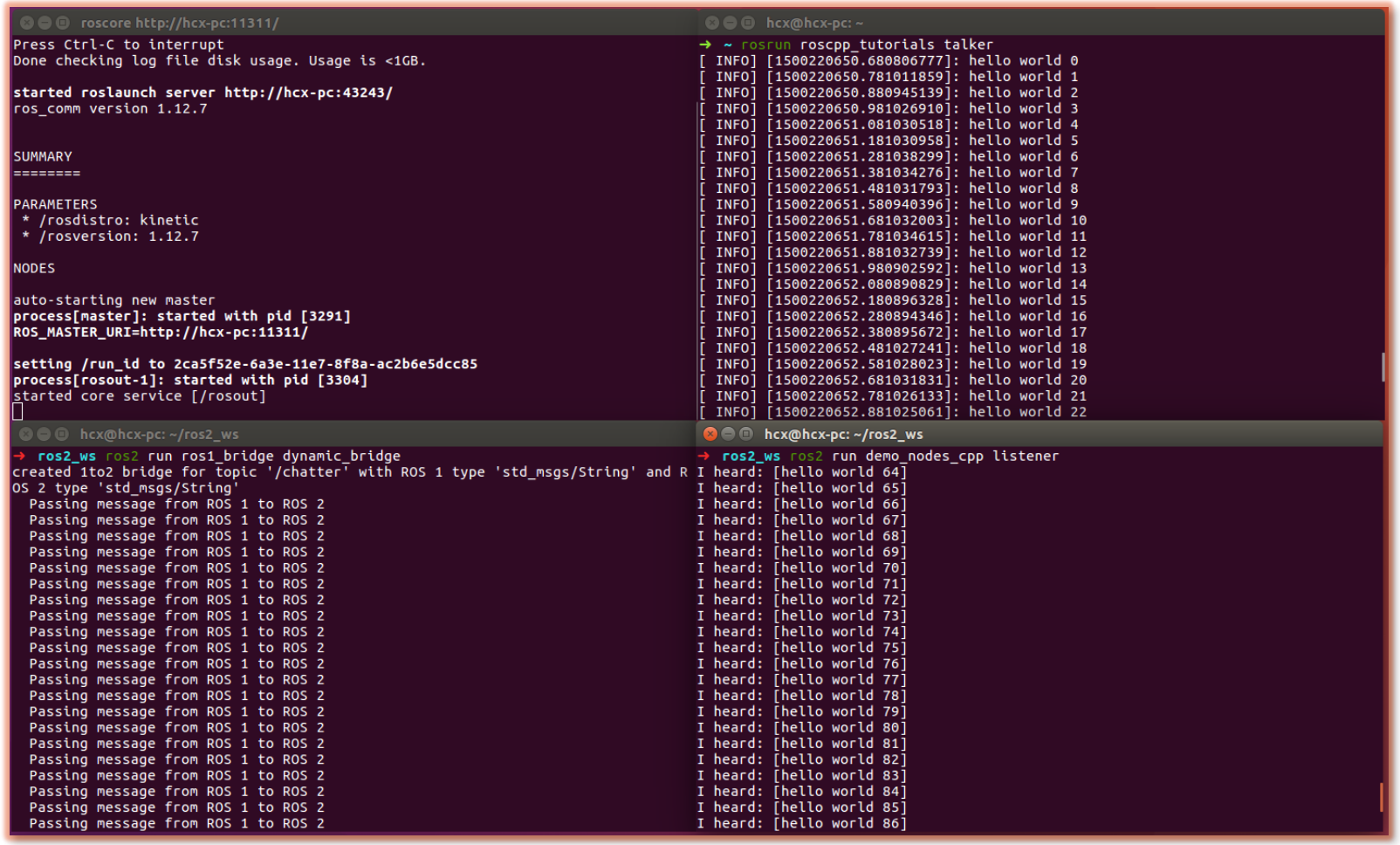
ROS 1

- Plenty of tools
- Existing funtionality

How-ROS1与ROS2之间的桥梁



How-ROS1与ROS2之间的桥梁



ROS1 Master

roscore

ros1_bridge

ros2 run ros1_bridge dynamic_bridge

talker

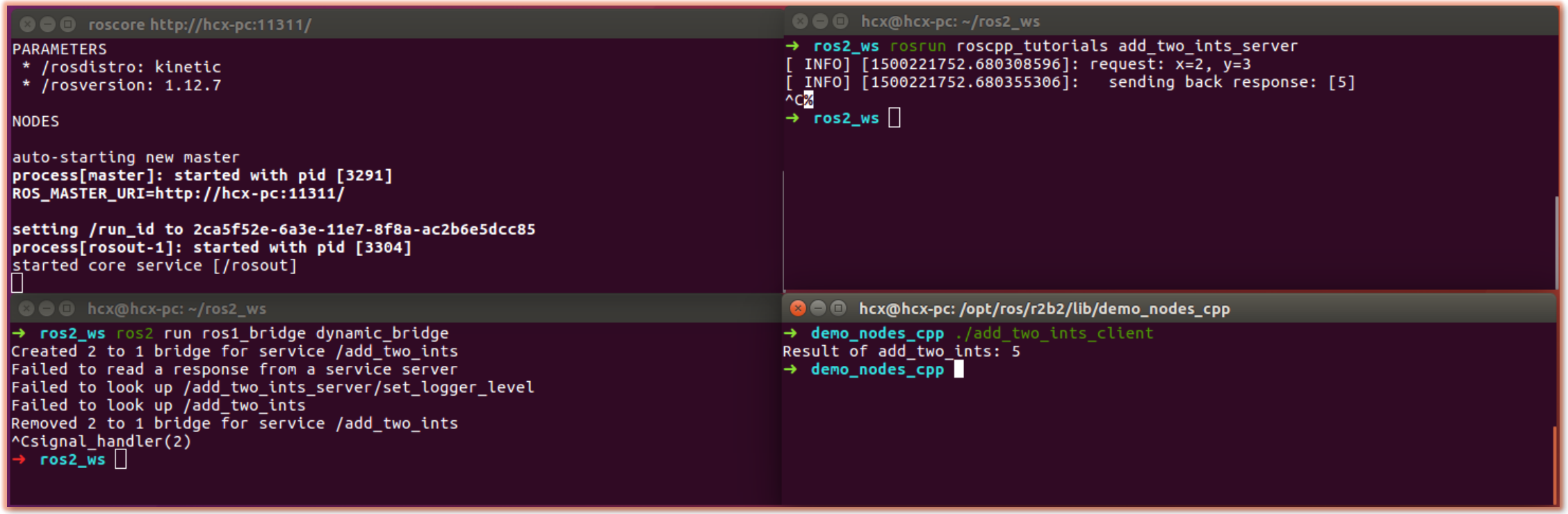
rosrun roscpp_tutorials talker

listener

ros2 run demo_nodes_cpp listener

How-ROS1与ROS2之间的桥梁

ROS1 Master	roscore
ros1_bridge	ros2 run ros1_bridge dynamic_bridge
Service server	rosrun roscpp_tutorials add_two_ints_server
Service client	ros2 run demo_nodes_cpp add_two_ints_client



总结 Summarize

- ROS1已成为机器人领域的事实标准，同时也存在一些局限性
- ROS2不是ROS1的升级版，而是颠覆性的重新设计
- 目前ROS2处于开发阶段，仍然建议使用ROS1

延展阅读

- ✓ http://robots.ros.org/
- ✓ http://design.ros2.org/articles/why_ros2.html
- ✓ https://github.com/ros2/examples
- ✓ http://www.guyuehome.com

Thank you

星河智能
AIgalaxy