



0基础ROS2应用开发实战

基于Atlas 200I DK A2摄像头目标检测

主讲人：华为海思昇腾CANN 杜鹏



课程大纲

- **准备工作**
- ROS2通信框架
- CANN加速模型推理
- 摄像头目标检测运行演示

准备工作

- Atlas_200I_DK_A2小站制卡（内置Ubuntu22.04）及联网指南，请参见昇腾社区

<https://www.hiascend.com/forum/thread-0257118120733068063-1-1.html>

- ROS与Ubuntu版本对应关系

| Ubuntu | ROS | Release date | End of Life |
|--------|--------------|--------------|-------------|
| 16.04 | ROS1 Kinetic | 2016.5 | 2021.4 |
| 18.04 | ROS1 Melodic | 2018.5 | 2023.5 |
| 20.04 | ROS1 Noetic | 2020.5 | 2025.5 |
| 20.04 | ROS2 Foxy | 2020.6 | 2023.5 |
| 22.04 | ROS2 Humble | 2022.5 | 2027.5 |

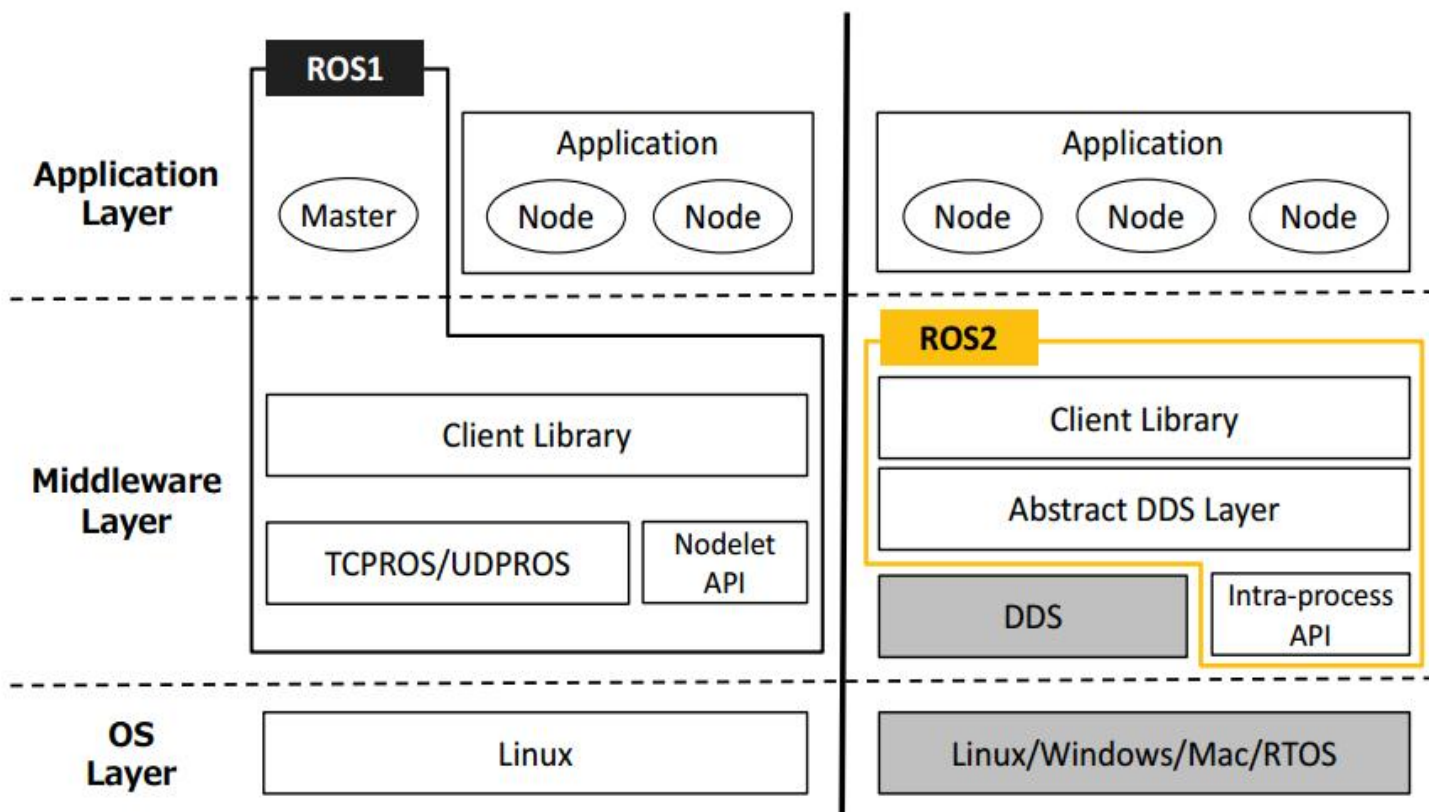
- ROS2安装

- (1) 安装ROS2-humble桌面版
- (2) 配置ros2环境变量

课程大纲

- 准备工作
- **ROS2通信框架**
- CANN加速模型推理
- 摄像头目标检测运行演示

ROS2与ROS1对比



- 框架重构：ROS1框架下，所有节点需要使用Master进行管理；ROS2框架使用基于DDS (Data Distribution Service) 的隐藏发现机制。
- API重构：ROS1的API大多是2009年设计的，ROS2对API做了重构，但保持大部分使用方法不变。
- 编译系统升级：ROS1使用cakin，ROS2升级成ament、colcon。

ROS2功能包创建流程

➤ 创建工作空间

```
$ mkdir -p ~/dev_ws/src
```

➤ 下载或创建功能包

```
$ cd ~/dev_ws/src
```

```
$ git clone https://gitee.com/guyuehome/ros2_21_tutorials.git # 下载功能包
```

```
$ ros2 pkg create --build-type <build-type> <package_name> # 创建功能包
```

➤ 自动安装依赖

```
$ rosdep install -i --from-path src --rosdistro humble -y
```

➤ 编译工作空间

```
$ colcon build --packages-select <package_name>
```

➤ 设置环境变量

```
$ source install/local_setup.sh # 仅在当前终端生效
```

```
$ echo " source ~/dev_ws/install/local_setup.sh" >> ~/.bashrc # 所有终端均生效
```

ROS2编写&运行节点 (Python)

- 新建hello_world/node_helloworld.py文件

```
import rclpy
```

```
from rclpy.node import Node # 导入库文件
```

```
def main(args=None):
```

```
    rclpy.init(args=args) # 初始化客户端
```

```
    node = Node( "helloworld" ) # 新建节点对象
```

```
    rclpy.spin(node) # spin循环节点
```

```
    rclpy.shutdown() # 关闭
```

- 打开功能包的setup.py文件，加入如下入口点的配置：

```
entry_points={
```

```
    'console_scripts': [
```

```
        'node_helloworld = hello_world.node_helloworld:main',
```

```
    ],
```

- 运行：\$ ros2 run hello_world node_helloworld

ROS2编写&运行节点 (C++)

- 新建hello_world/node_helloworld.cpp文件

```
#include "rclcpp/rclcpp.hpp" //导入库文件
```

```
int main(int argc, char** argv){
```

```
    rclcpp::init(argc, argv); // 初始化客户端
```

```
    auto node = std::make_shared<rclcpp::Node>("helloworld"); // 新建节点对象
```

```
    rclcpp::spin(node); // spin循环节点
```

```
    rclcpp::shutdown(); // 关闭
```

```
}
```

- 打开功能包的cmakelists.txt, 加入如下入口点的配置:

```
add_executable(node_helloworld src/detection_example.cpp)
```

```
ament_target_dependencies(node_helloworld rclcpp)
```

- 运行: \$ ros2 run hello_world node_helloworld

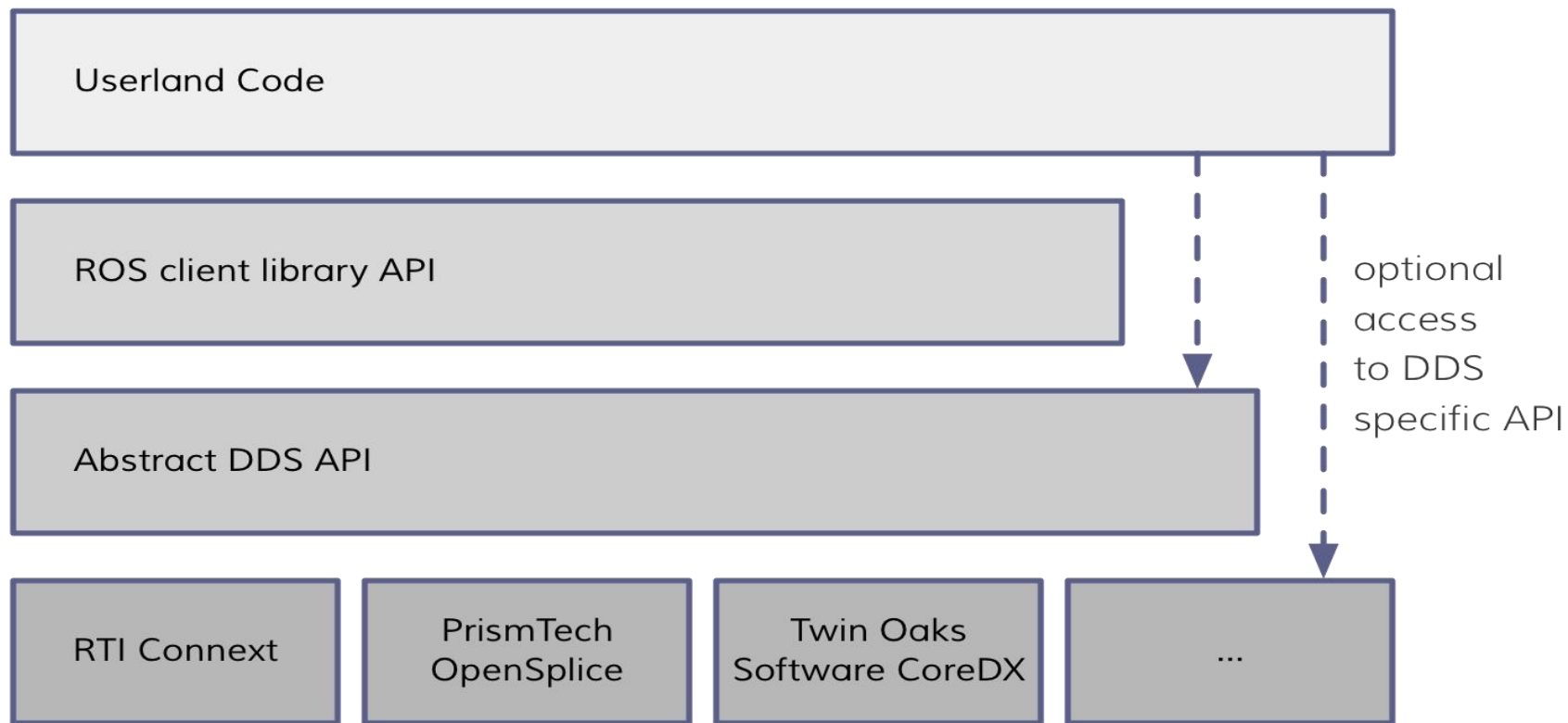
ROS底层通信标准-DDS



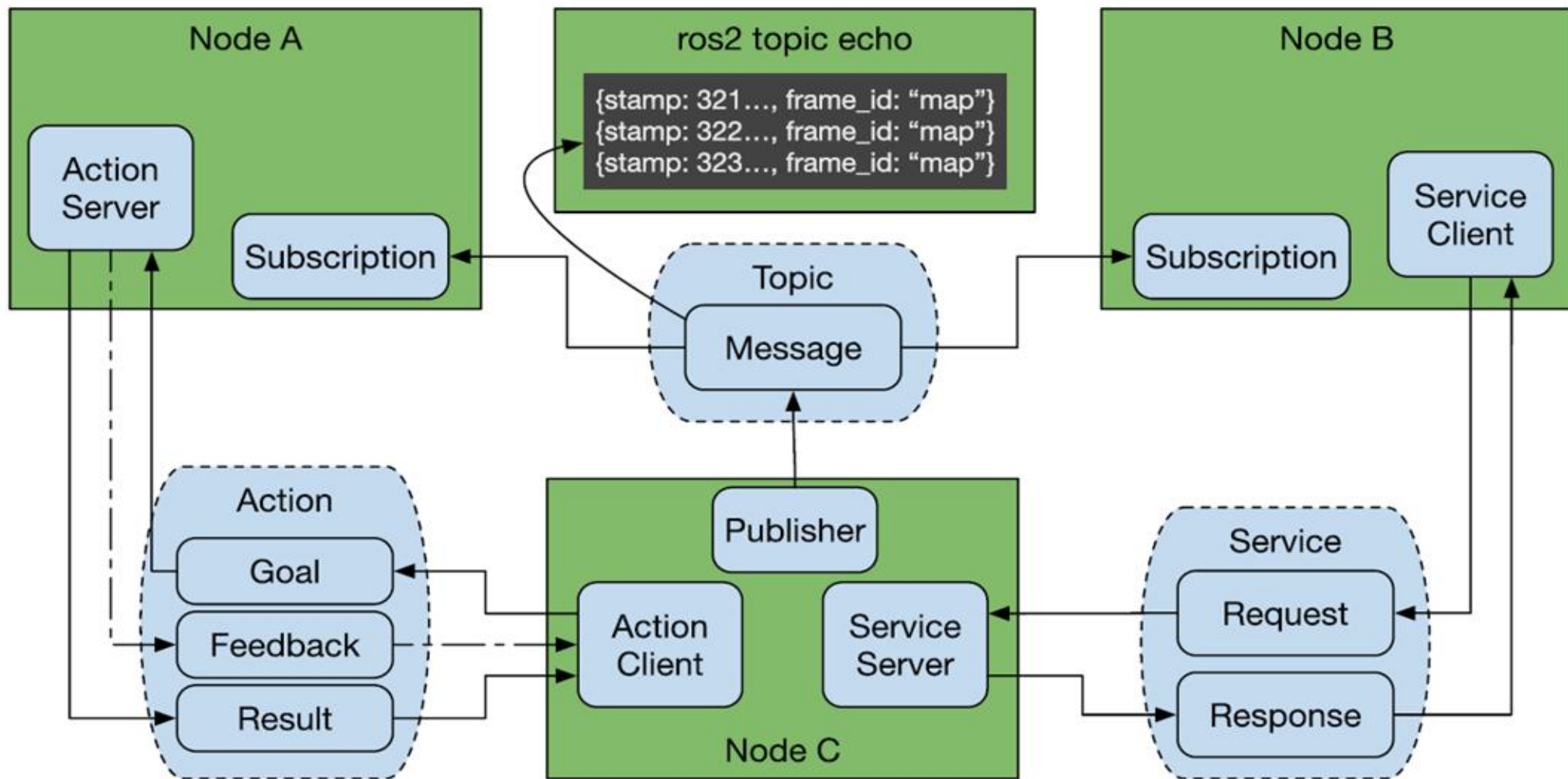
- DDS的全称是Data Distribution Service，也就是数据分发服务，2004年由对象管理组织OMG发布和维护，是一套专门为实时系统设计的数据分发/订阅标准，最早应用于美国海军，解决舰船复杂网络环境中大量软件升级的兼容性问题，现在已经成为强制标准。DDS强调以数据为中心，可以提供丰富的服务质量策略，以保障数据进行实时、高效、灵活地分发，可满足各种分布式实时通信应用需求。
- 我们在后续课程中介绍的话题、服务、动作，他们底层通信的具体实现过程，都是靠DDS来完成的，它相当于是ROS机器人系统中的神经网络。

DDS中间件

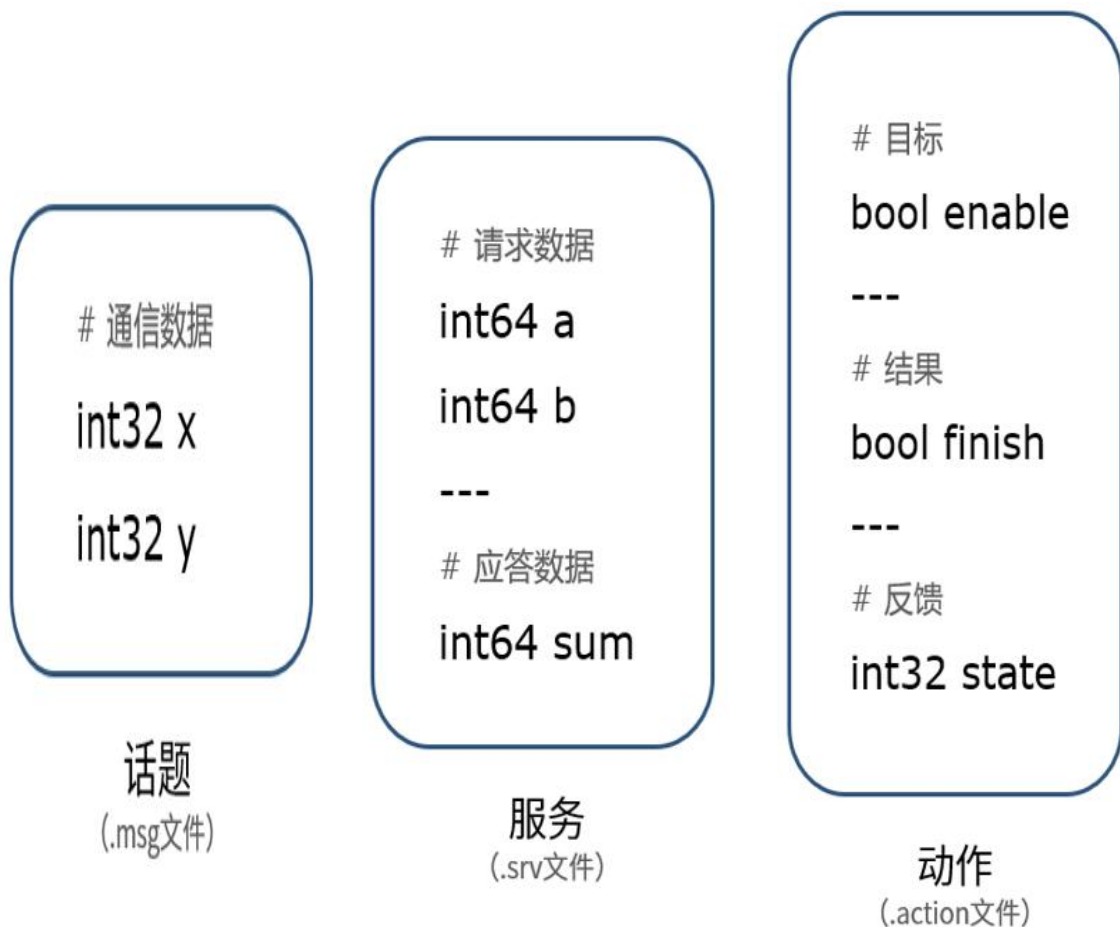
进程间采用共享内存通信，分布式通常采用UDP通信



ROS2三种消息通信方式（话题、服务、动作）



ROS2消息格式



- 话题通信接口的定义使用的是.msg文件，单向传输。
- 服务通信接口的定义使用的是.srv文件，包含请求和应答两部分定义，通过中间的“---”区分。
- 动作是另外一种通信机制，用来描述机器人的一个运动过程，使用.action文件定义。

举例：ROS2话题接口定义

- 在learning_interface/msg/ObjectPosition.msg定义接口

int32 x # 表示目标的X坐标

int32 y # 表示目标的Y坐标

- 在功能包的CMakeLists.txt中配置编译选项

```
find_package(rosidl_default_generators REQUIRED)
```

```
rosidl_generate_interfaces(${PROJECT_NAME}
```

```
  "msg/ObjectPosition.msg"
```

```
)
```

ROS2话题发布接口使用

➤ 发布者接口调用

```
from learning_interface.msg import ObjectPosition
```

```
class PublisherNode(Node):
```

```
    def __init__(name)
```

```
        self.pub = self.create_publisher(\n            ObjectPosition, 'position', 10)
```

```
# 创建订阅者对象（消息类型、话题名、队列长度）
```

```
    def broadcast():
```

```
        position.x, position.y = int(self.objectX), int(self.objectY)
```

```
        self.pub.publish(position) # 发布目标位置
```

```
def main(args=None): # ROS2节点主入口main函数
```

```
    node = PublisherNode("topic_position_pub") # 创建ROS2节点对象并进行初始化
```

```
    node.broadcast()
```

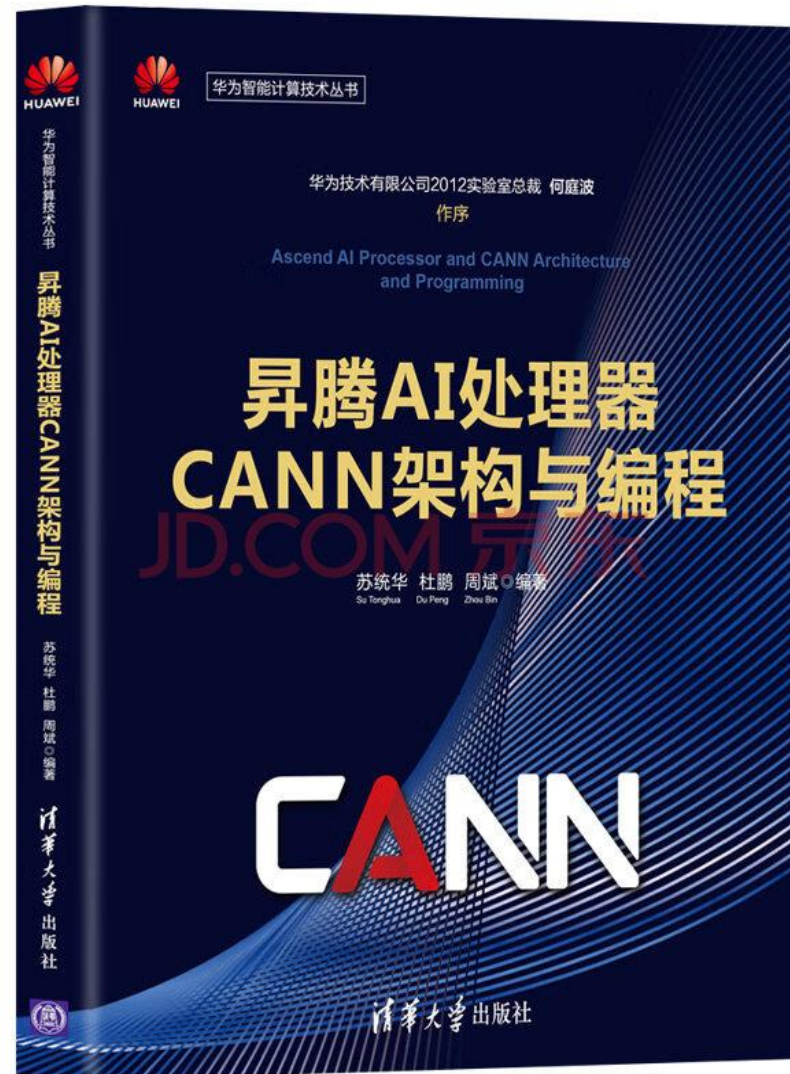
ROS2话题订阅接口使用

➤ 订阅者接口调用

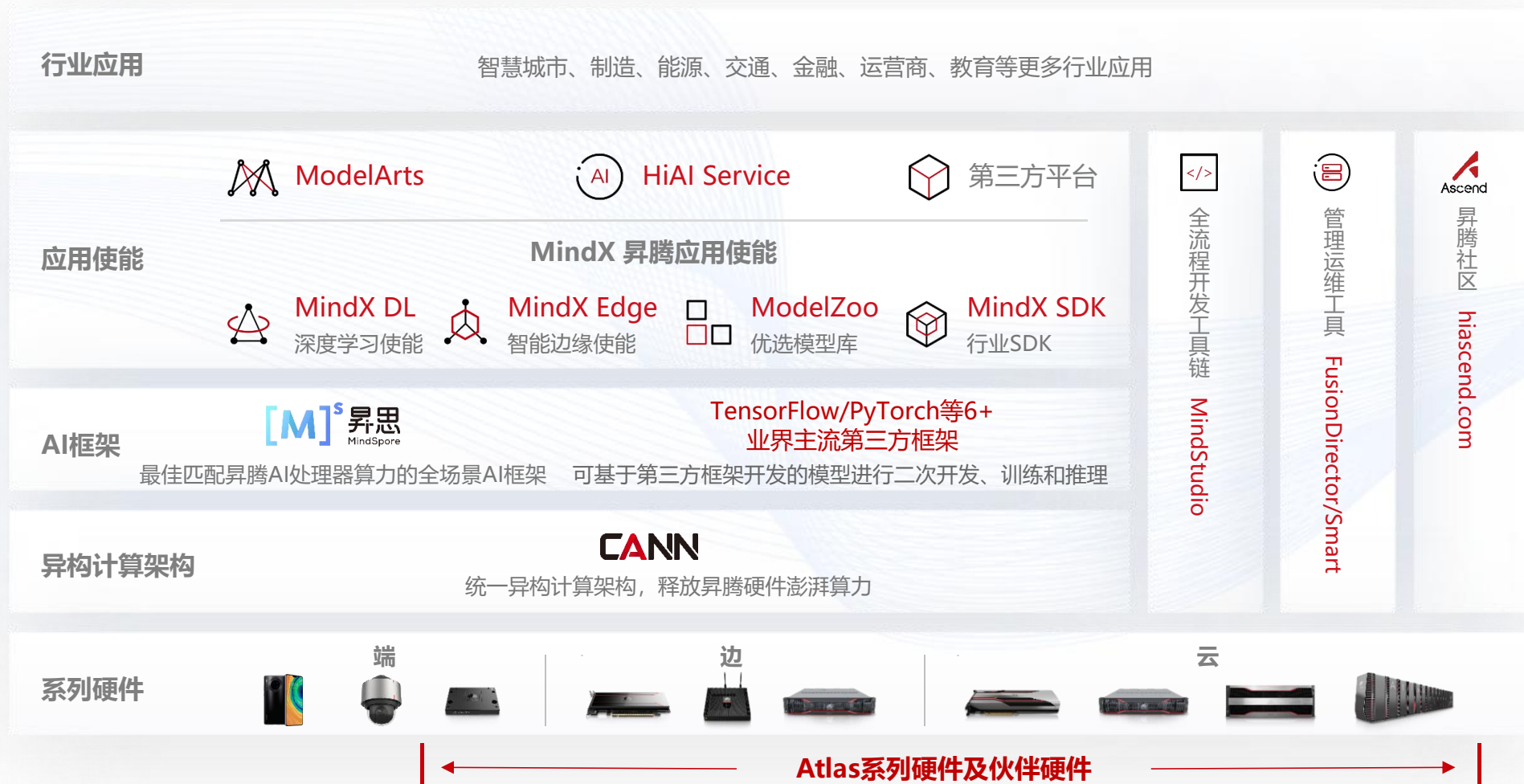
```
from learning_interface.msg import ObjectPosition
class SubscriberNode(Node):
    def __init__(self, name)
        self.sub = self.create_subscription(\
            ObjectPosition, "position", self.listener_callback, 10)
# 创建订阅者对象（消息类型、话题名、订阅者回调函数、队列长度
    def listener_callback(self, msg):
# 创建回调函数，执行收到话题消息后对数据的处理
        self.get_logger().info('Target Position: "(%d, %d)" % (msg.x, msg.y))
def main(args=None):                                # ROS2节点主入口main函数
    node = SubscriberNode("topic_position_sub") # 创建ROS2节点对象并进行初始化
    rclpy.spin(node) # spin循环节点
```


课程大纲

- 准备工作
- ROS2通信框架
- CANN加速模型推理
- 摄像头目标检测运行演示



昇腾异构计算架构CANN：昇腾软件栈，向下对接昇腾硬件，向上使能开发者



2019年8月

华为成为国家新一代人工智能

开放创新平台

基础软硬件平台

唯一建设单位

异构计算架构CANN：提供多层次开发接口

CANN (Compute Architecture for Neural Networks)

是华为针对AI场景推出的异构计算架构，通过提供多层次编程接口，支持用户快速构建基于Ascend平台的AI应用和业务。



统一APP编程语言

提供一套标准的AscendCL编程接口，对应用程序开发者屏蔽底层多种芯片差异，提升用户APP编程易用性



统一网络构图接口

提供标准的昇腾计算AIR，支持多框架，支持用户在昇腾处理器上快速部署神经网络业务



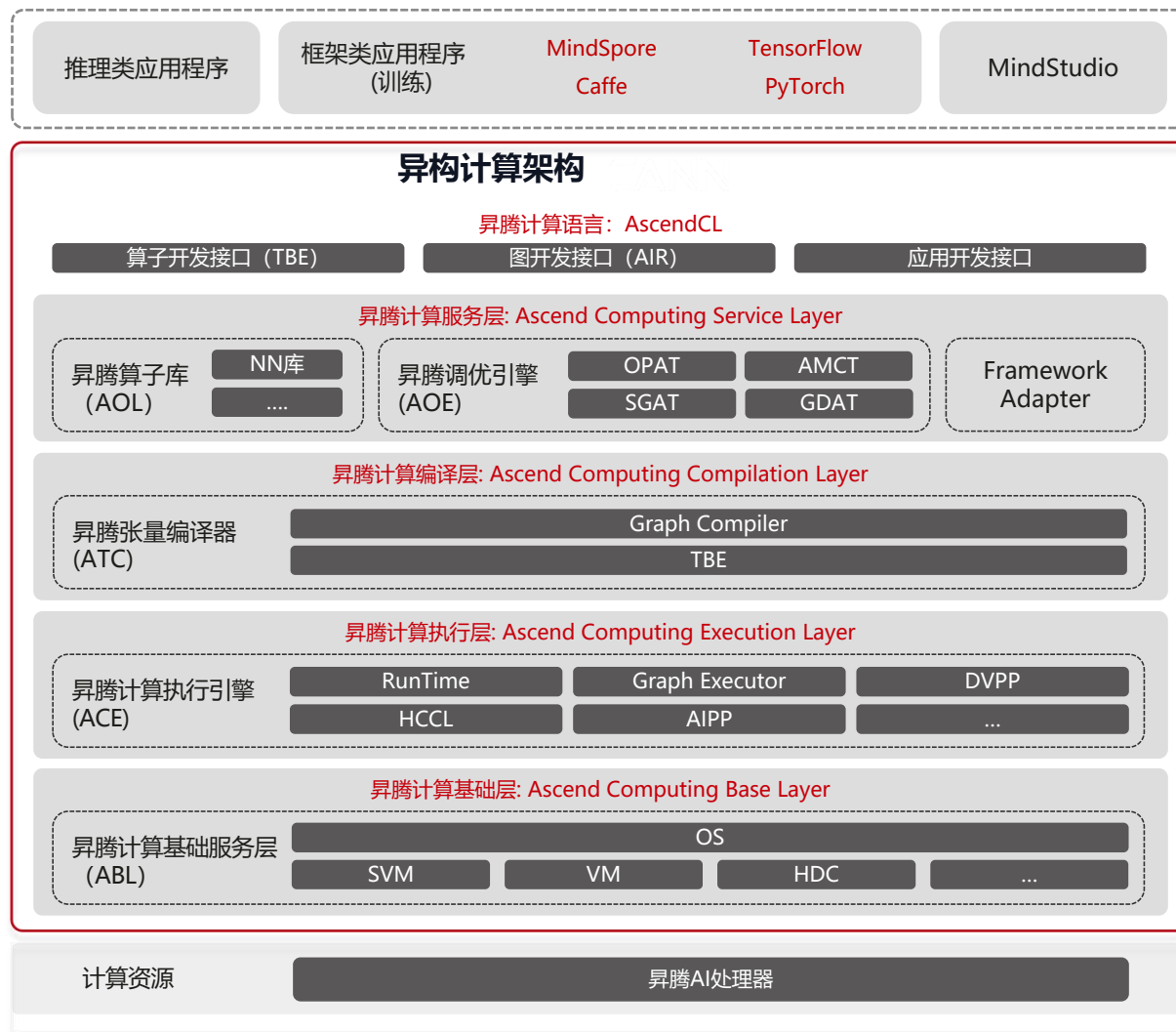
高性能计算引擎以及算子库

通过高性能编译引擎/执行引擎/调优引擎和预置高性能算子库，支撑客户快速部署神经网络业务、降低部署成本并最大程度发挥昇腾计算能力



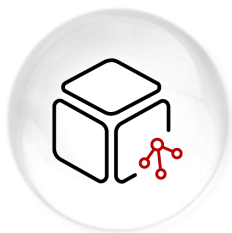
基础服务

提供驱动、虚拟化、媒体、集合通信等能力



从模型优化到推理业务部署，全方位使能应用开发

<https://www.hiascend.com/zh/marketplace/mindx-sdk/case-studies/b2391ab9-53d6-4794-8d44-2a2286fed2e3>



模型压缩AMCT

Ascend Model Compression Toolkit

降低模型数据量和计算量，提升计算性能

模型量化

IN8/INT4量化 | 支持多框架 | 自动混合精度量化

张量分解

昇腾亲和分解 | 支持多框架

通道稀疏

支持多框架 | 自动通道稀疏 | 结构化4选2稀疏



模型优化ATC

Ascend Tensor Compiler

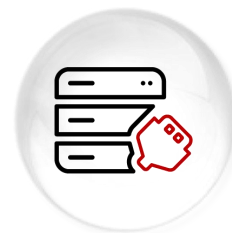
将三方框架模型编译成可在昇腾加速执行的格式

高效编译

一键式编译命令
支持主流AI框架

编译优化

常量折叠 | 图/算子融合
权重数据重排 | 内存使用优化
编译时直接量化



部署执行AscendCL

Ascend Computing Language

调用AscendCL编程接口高效完成模型部署和执行

高性能加载执行

数据驱动的高效模型推理
高性能DVPP/CBLAS加速库
高性能单算子执行

友好易用

支持C++/Python编程
支持Host/Device灵活部署
支持单算子二进制直调

课程大纲

- 准备工作
- ROS2通信框架
- CANN加速模型推理
- **摄像头目标检测运行演示**

Atlas 200I DK A2开发者套件 (华为商城有售)



8 TOPS INT8

4 TFLOPS FP16

LPDDR4X, 4GB, 支持 ECC

4core * 1.0GHz

支持 H.264 / H.265 Decoder 硬件解码, 20路 1080P 30FPS, 2路 4K 75FPS
JPEG 解码 1080P 512FPS, 编码 1080P 256FPS, 分辨率最大 16384 x 16384

MIPI-CSI 51针连接器 *2, 可接摄像头模组

MIPI-DSI 51针连接器 *1

RJ45 网口 *2, 支持自适应 100 / 1000M

HDMI 接口 *2

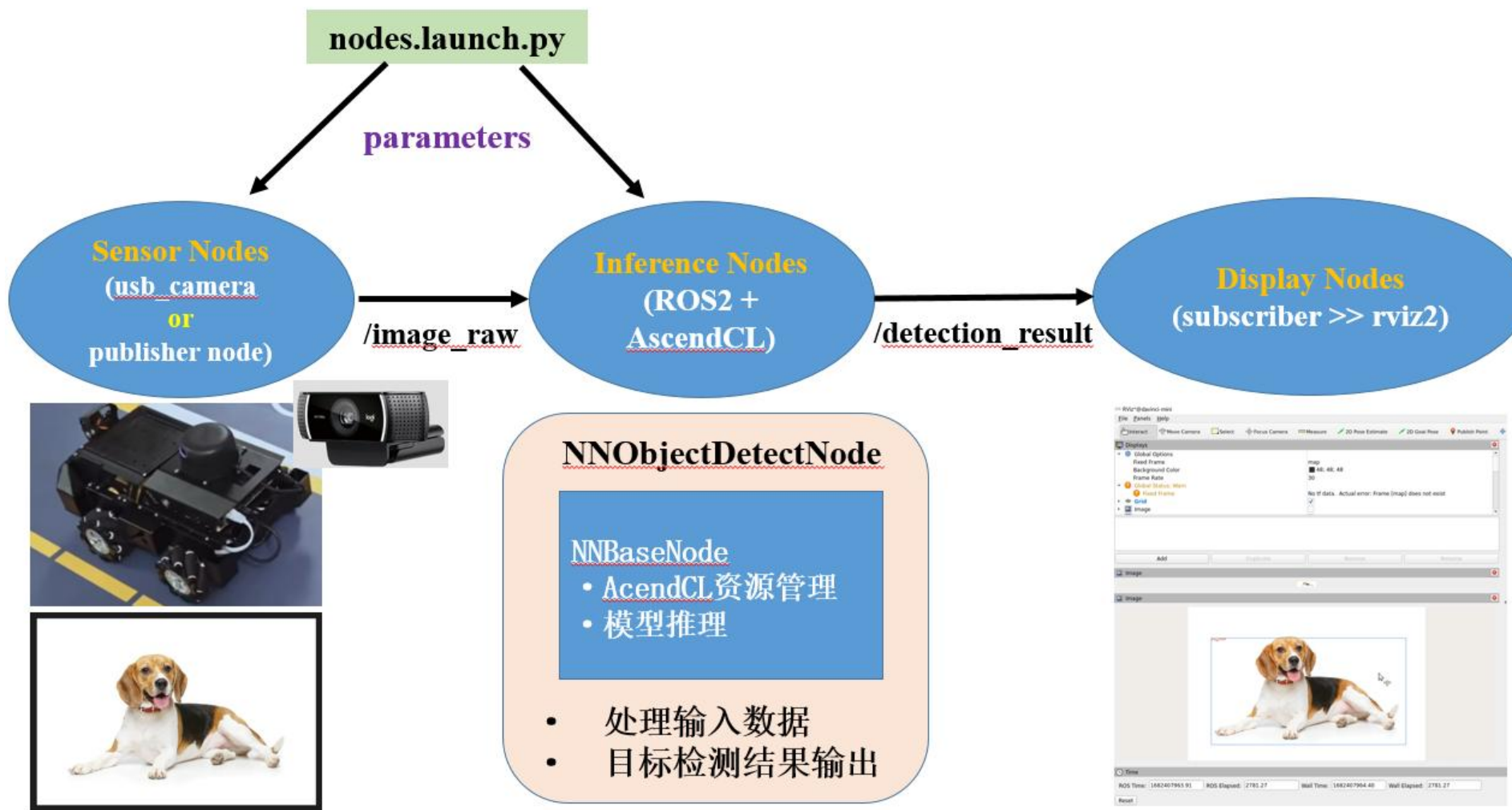
USB 3.0 Type-A 接口 *2, 兼容 USB 2.0

USB 3.0 Type-C 接口 *1

40针扩展连接器 *1

摄像头目标检测运行演示

https://e.gitee.com/HUAWEI-ASCEND/repos/HUAWEI-ASCEND/ascend-devkit/tree/master/src/E2E-Sample/ros2_acl_inference



下期预告：四足机器人运动控制&解决方案



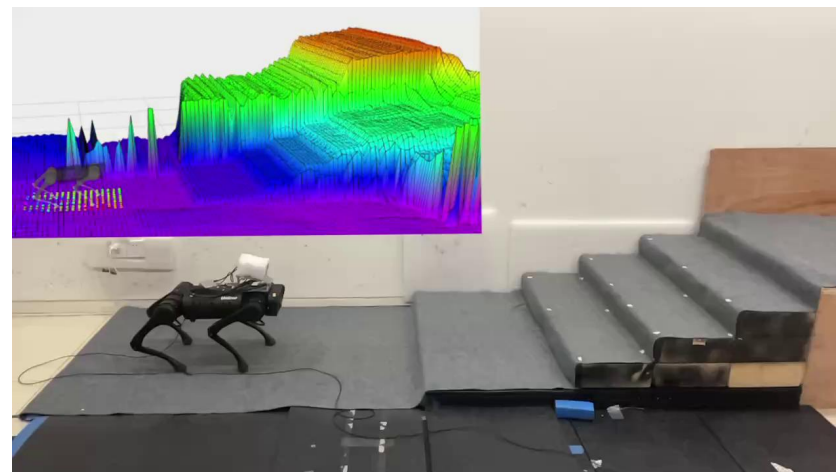
MPC+WBC运动控制



强化学习运动控制



自主避障+目标追踪



强化学习结合地形感知运动控制

以下哪个不是ROS2相比于ROS1的改进点:

- A、框架重构: ROS1框架下, 所有节点需要使用Master进行管理; ROS2框架使用基于DDS (Data Distribution Service) 的隐藏发现机制。
- B、API重构: ROS1的API大多是2009年设计的, ROS2对API做了重构, 但保持大部分使用方法不变。
- C、通信方式升级: ROS2取消了ROS1的通信方式, 引入全新通信API。
- D、编译系统升级: ROS1使用cakin, ROS2升级成ament、colcon。

以下哪个不是ROS2的通信方式

- A、话题
- B、服务
- C、动作
- D、消息

Thank you.

昇腾开发者社区



<http://hiascend.com>

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

*Bring digital to every person, home, and
organization for a fully connected,
intelligent world.*

Copyright©2021 Huawei Technologies Co., Ltd.
All Rights Reserved.

*The information in this document may contain predictive
statements including, without limitation, statements
regarding
the future financial and operating results, future product
portfolio, new technology, etc. There are a number of
factors that
could cause actual results and developments to differ
materially
from those expressed or implied in the predictive
statements.
Therefore, such information is provided for reference
purpose
only and constitutes neither an offer nor an acceptance.
Huawei
may change the information at any time without notice.*

