

# Week2

☰ 태그	
☰ 내용	

## 5장 ROS2의 중요 콘셉트

### ROS2를 사용해야 하는 이유

1. 시장 출시 시간을 단축할 수 있습니다. 플랫폼과 프레임워크 소프트웨어를 사용하면 당연히 가능한 일입니다. ROS2 사용자는 인하우스 응용 기능 개발에만 집중할 수 있습니다.
2. 양산 가능한 수준의 설계이며, 최종 양산 제품에 적용 가능한 신뢰성과 안정성이 보장됩니다.
3. 멀티 플랫폼으로 확장이 가능하며, Windows와 macOS를 추가 지원합니다. 하드웨어 측면에서는 임베디드 시스템으로도 확장이 가능합니다.
4. 모바일 로봇, 드론, 자율 주행, 우주 항공 등 다양한 분야로 확장이 가능합니다.
5. 모듈화가 잘 되어 있어서 부분적으로 교체나 수정이 용이합니다. (예: DDS 변경)
6. 산업 표준인 DDS와 IDL을 채택합니다.
7. 오픈 소스 라이선스인 Apache 2.0을 사용합니다.
8. 커뮤니티를 통한 지원이 가능합니다.
9. 산업계에서도 지원하고 있습니다.
10. ROS 1과도 상호 운용성이 가능합니다. (다만, 브릿지가 필요합니다.)

## 6장 ROS1과 2의 차이점으로 보는 ROS2의 특징

### ROS1의 기술적 한계와 ROS 2에서의 개선 방향

1. 단일 로봇에서 두 대 이상의 로봇으로 확장
2. 고사양 요구사항 최적화

3. 리눅스 환경 외에도 다양한 OS 지원
4. 리얼타임 지원을 위한 DDS 및 QoS
5. 안정적인 네트워크 환경 구성을 위한 QoS

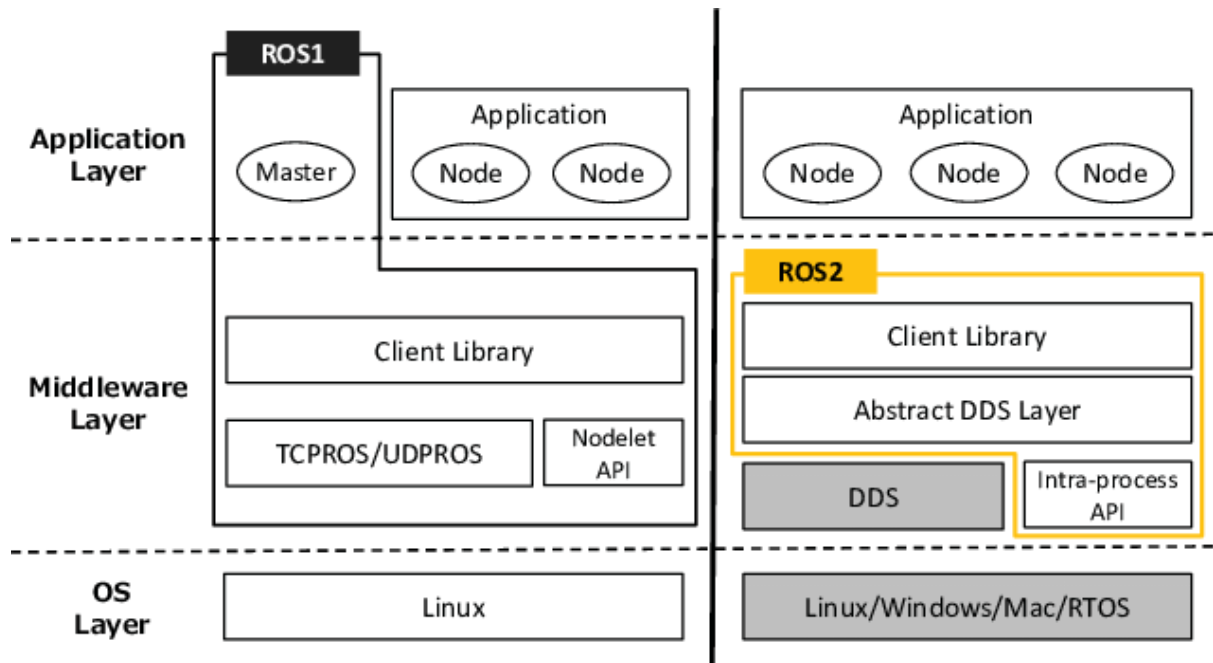
## ROS2의 특징

1. 리눅스, 윈도우, macOS 를 지원
  2. DDS 와 RTPS (Real-time Publish-Subscribe Protocol), QoS 등을 통해 리얼타임 을 지원
  3. ROS2에서는 DDS-Security 를 적용하여 통신 단부터 보다 강력한 보안을 적용
  4. RTPS 지원하는 DDS 의 적용, IDL 의 사용, QoS 사용
  5. 다양한 벤더의 DDS 를 사용 가능함
  6. ROS2에서는 ROS1 의 roscore 를 사용하지 않고, DDS 의 동적 디스커버리를 이용하여 노드를 검색하고 연결할 수 있다.
  7. C++14 (C++17) 과 Python 3.5+ 를 지원한다.
  8. ament 빌드 시스템의 적용 (CMake)
  9. colcon build 도구 제공
  10. Multiple workspace, No non-isolated build 지원, No devel space
  11. vcstool (\*.repos) 사용
  12. rclcpp, rcl, rclpy, rclobjc, rclada, rclgo, rclnodejs 등을 클라이언트 라이브러리 제공
  13. 라이프사이클, 상태 천이의 적용
  14. 하나에 프로세스에서 복수 개의 노드 지원
  15. 다양한 스레드 모델을 선택하여 사용 가능
  16. 토픽, 서비스, 액션의 메시지 지원
  17. 명령줄 인터페이스의 지원으로 다양한 ROS2 명령어 지원
  18. run 과 launch 를 통한 실행을 지원
  19. 실행 중에 재매핑을 지원 (예정)
  20. 임베디드 시스템 지원 (eg. micro-ROS)
-

# 7장. ROS2 와 DDS

## ROS2 에서 DDS 를 선택한 배경

ROS1에서는 TCPROS의 구조와 기술적 한계 때문에 DDS가 산업 표준으로 선정되었습니다. DDS를 사용하면 IDL(인터페이스 설명 언어)을 사용하여 메시지를 정의하고 직렬화할 수 있습니다.



## DDS(Data Distribution Service)

DDS 는 데이터 분산 서비스에 대한 규격 입니다. OMG(Object Management Group) 에서 산업 표준으로 정했습니다. 실시간 및 임베디드 시스템용 오픈소스 미들웨어 입니다.

1. OMG 에서 산업 표준으로 지정
2. 다양한 운영체제에서 사용 가능
3. 다양한 언어로 사용가능
4. UDP 기반의 전송방식, QoS 를 통한 신뢰성 보장
5. 데이터 중심적인 기능
6. 동적 디스커버리 기능의 제공
7. 확장 가능한 아키텍처, 다양한 응용 분야에서 사용이 가능
8. 다양한 벤더에서 개발 가능
9. 서비스 품질, QoS (22 가지의 옵션 제공)

## 10. 보안 기능의 강화

### DDS 사용해보기

ROS2를 설치한 후, talker와 listener 예제를 실행하면 DDS를 통한 프로세스 간 통신을 테스트해볼 수 있습니다.

```
makeplusscode@worktop: ~$ ros2 run std_msgs_demo talker
[INFO] [1681799920.570749631] [talker]: Publishing: 'Hello World: 62'
[INFO] [1681799921.570749030] [talker]: Publishing: 'Hello World: 63'
[INFO] [1681799922.570752503] [talker]: Publishing: 'Hello World: 64'
[INFO] [1681799923.570735237] [talker]: Publishing: 'Hello World: 65'
[INFO] [1681799924.570741991] [talker]: Publishing: 'Hello World: 66'
[INFO] [1681799925.570760515] [talker]: Publishing: 'Hello World: 67'
[INFO] [1681799926.570743109] [talker]: Publishing: 'Hello World: 68'
[INFO] [1681799927.570742954] [talker]: Publishing: 'Hello World: 69'
[INFO] [1681799928.570733425] [talker]: Publishing: 'Hello World: 70'
[INFO] [1681799929.570752131] [talker]: Publishing: 'Hello World: 71'
[INFO] [1681799930.570757860] [talker]: Publishing: 'Hello World: 72'

makeplusscode@worktop: ~$ ros2 run std_msgs_demo listener
[INFO] [1681799920.571513525] [listener]: I heard: [Hello World: 62]
[INFO] [1681799921.571693923] [listener]: I heard: [Hello World: 63]
[INFO] [1681799922.571584526] [listener]: I heard: [Hello World: 64]
[INFO] [1681799923.571734989] [listener]: I heard: [Hello World: 65]
[INFO] [1681799924.571517685] [listener]: I heard: [Hello World: 66]
[INFO] [1681799925.571588489] [listener]: I heard: [Hello World: 67]
[INFO] [1681799926.571584572] [listener]: I heard: [Hello World: 68]
[INFO] [1681799927.571505917] [listener]: I heard: [Hello World: 69]
[INFO] [1681799928.571469169] [listener]: I heard: [Hello World: 70]
[INFO] [1681799929.571499625] [listener]: I heard: [Hello World: 71]
[INFO] [1681799930.571505024] [listener]: I heard: [Hello World: 72]
```

### DDS 벤더리스트

1. Dassault Systemes- OMG Member (P)
2. Eclipse Foundation- OMG Member (C)
3. eProsima- OMG Member (P)
4. GurumNetworks, Inc.- OMG Member (P)
5. Kongsberg Defence & Aerospace- OMG Member (G)
6. MilSOFT- OMG Member (T)
7. Nanjing Zhenrong Technology Co., LTD- OMG Member (P)
8. Object Computing, Inc. - OCI- OMG Member (I)

9. Real-Time Innovations- OMG Member (C)
10. S2E Software Systems B.V.- OMG Member (P)
11. Sparx Systems- OMG Member (C)
12. Twin Oaks Computing, Inc.- OMG Member (P)
13. ZettaScale Technology- OMG Member (P)

## RTPS(Real Time Publish Subscribe)

DDS specification은 네트워크에서 메시지 교환을 위한 **전송프로토콜** 을 다루지 않습니다.

DDS는 점차 사용이 증가하면서 여러 벤더에서 DDS를 구현하게 되었고, 서로 다른 벤더에서 구현된 DDS는 서로 통신할 수 없는 문제가 발생했습니다. 이 문제를 해결하기 위해, 표준 wire-protocol로 선정된 것이 RTPS입니다.

RTPS는 산업 자동화를 위한 프로토콜로, 기존 산업용 디바이스에서 세계적으로 사용되어 검증된 방식입니다. RTPS는 Data-Distribution System 의 요구사항을 충족시키기 위해 설계되어, DDS의 타겟 application domain 중 하나입니다. DDS의 아키텍처와 RTPS 사이에는 많은 유사점이 있습니다.

- **QoS** 설정을 통해 best-effort와 reliable publish subscribe 통신을 선택할 수 있습니다.
- 프로토콜을 새로운 서비스로 확장할 수 있는 **Extensibility** 를 제공합니다.
- **Fault tolerance** 특정 지점의 장애에서도 네트워크를 구성할 수 있습니다.
- **Type-safety** 오류로 인해 원격 노드가 손상되지 않도록 보호합니다.
- **Plug-and-play** 애플리케이션과 서비스가 자동으로 검색되고 네트워크에 연결됩니다.

## 8장. DDS 의 QoS

ROS2 는 UDP 멀티캐스트 기반의 DDS 를 사용한다. DDS 사양에서는 22가지 QoS 옵션을 지원한다.

### ROS2 에서 사용되는 QoS 옵션

- 히스토리 (History)
  - DDS 메시지 큐의 메시지 보존 정책입니다.
- 신뢰성 (Reliability)
  - **BEST\_EFFORT** 는 속도를 중시하며 손실 가능성이 있습니다. 반면 **RELIABLE** 는 재전송 기능을 이용하여 손실을 막습니다.

- 내구성 (Durability)
  - 옵션에 따라서, 메시지를 영구 보관하거나, 삭제합니다.
- 데드라인 (Deadline)
  - 메시지가 도착하는 데 걸리는 최대 예상 시간
- 수명 (Lifespan)
  - 메시지를 발행하고 수신할 때 최대 시간
- 생존성 (Liveliness)
  - 정해진 주기 안에서 노드 또는 토픽의 생사를 확인하는 QoS 옵션