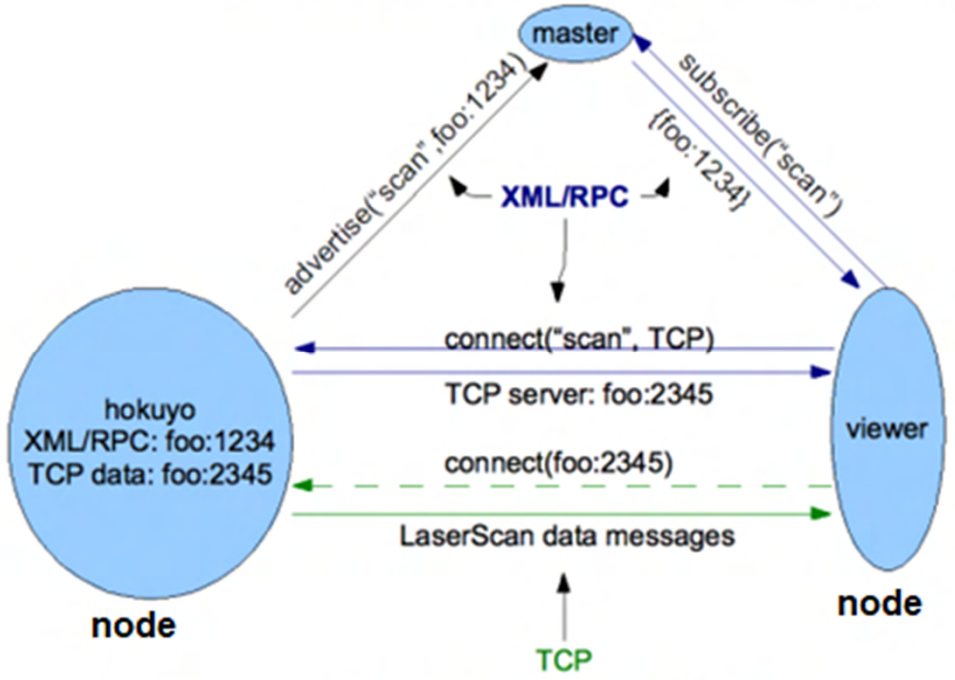
ROS 와 ROS2의 차이점: Real-Time[4]

- ROS1과 ROS2의 핵심적인 차이점
  - 데이터처리방식 (XMLRPS/TCPROS vs. DDS/RTPS)
  - Real-time Computing
  - 다양한 임베디드 시스템 지원 (MCU, ARM x86 등)
- XMLRPS/TCPROS: Master-Node 형태
  - Master에서 node의 정보를 갖고 있으면서, 이를 기반으로 discovery 실행함
  - Pub/Sub 필요한 node간의 1:1 통신을 가능하도록 만듬

publisher/subscriber node 정보 관리 with XMLRPC



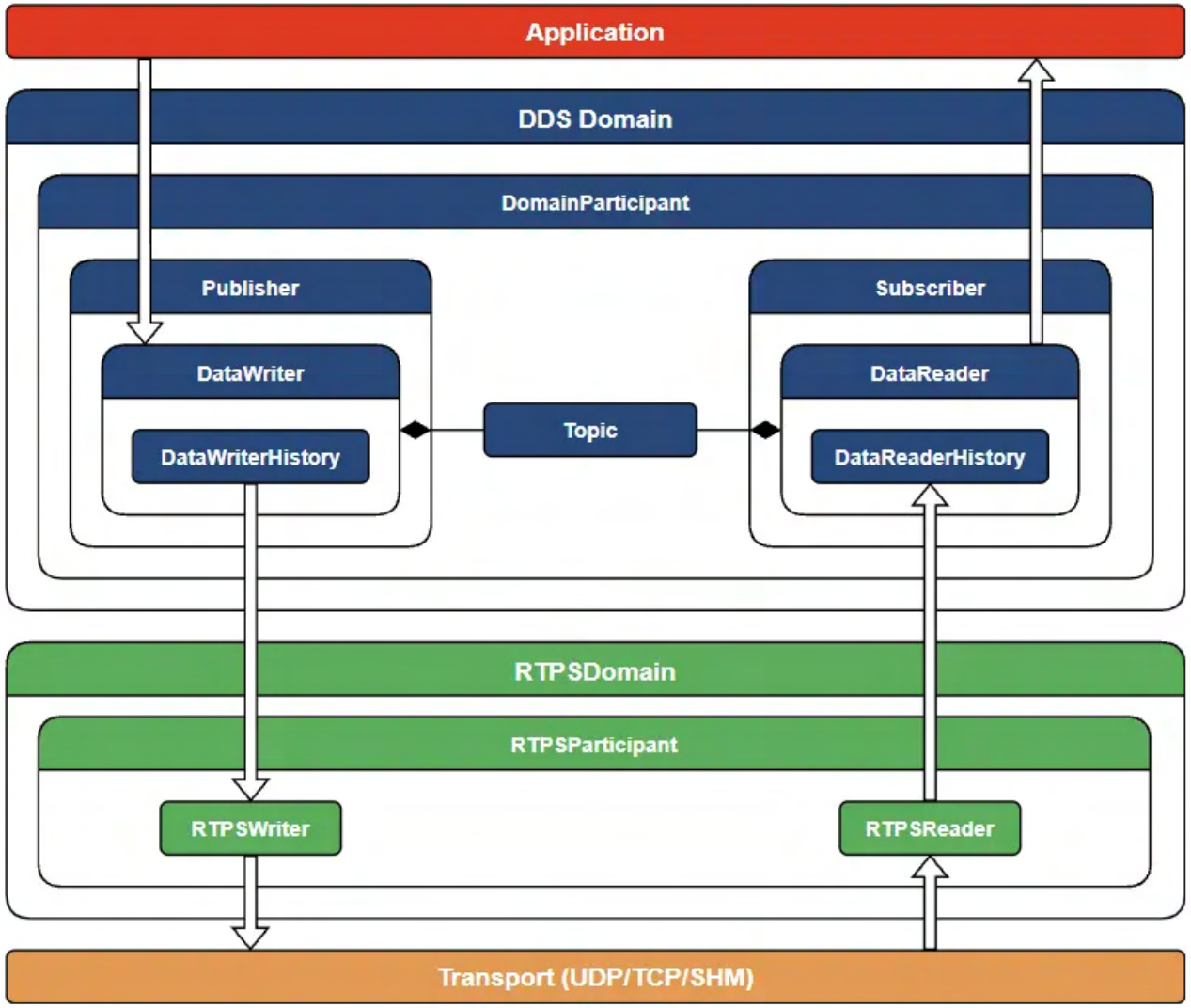
node 간 1:1 접속 with TCPROS

ROS 와 ROS2의 차이점: Real-Time[4]

- DDS/RTPS: Peer-to-Peer 통신 방식, TCP/UDP로 구현 가능, Master가 필요 없음
  - Node Discovery + Data Serialization + Transport로 구성됨
  - Node간의 역할을 Publish/Subscriber로 확실하게 정의하고 정보를 전달함
    - QoS 설정: 표준 IP 네트워크에서 best-effort와 reliable pub/sub 통신 선택
  - 각각의 node에서 실시간 discovery 기능을 갖고 있음
  - DDS는 msg 교환을 위한 transport protocol을 다루지 않지만, 표준이 필요해짐
    - 이를 위해 선정된 것이 RTPS(real-time publish subscribe)
    - 산업자동화를 위해 정의하여 검증된 프로토콜
  - RTPS의 특징
    ① QoS 설정과 Fault tolerance : single points failure에도 네트워크 구성 가능.
    ② Extensibility : 프로토콜을 새로운 서비스로 확장시킬 수 있음.
    ③ Plug-and-play : 새로운 application이나 service가 자동으로 discover되고 네트워크에 연결 됨.
    ④ Scalability : 매우 큰 네트워크로 시스템을 확장할 수 있음.
    ⑤ Type-safety : application의 programming 에러가 remote 노드들을 손상 시키지 못하게 함.

ROS 와 ROS2의 차이점: Real-Time[4]

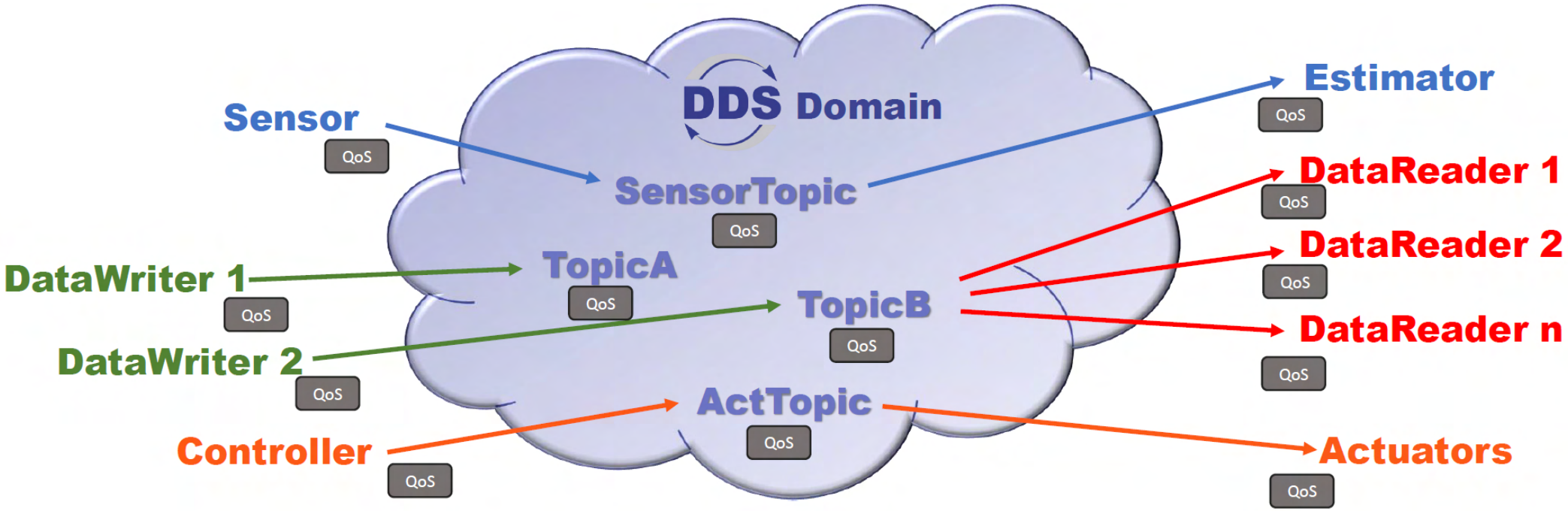
- Fast DDS 방식의 layer model
  - DDS Layer : DDS spec.을 위한 pub/sub comm. 구현된 high-lever layer
  - RTPS Layer : DDS 서비스 (e.g. discover, data update, push)를 위해 msg 구조와 컨텐츠를 활용하여 DDS의 DataWriter-DataReader가 서로 이해할 수 있는 msg를 교환하는 low-level Layer



Fast DDS layer model architecture

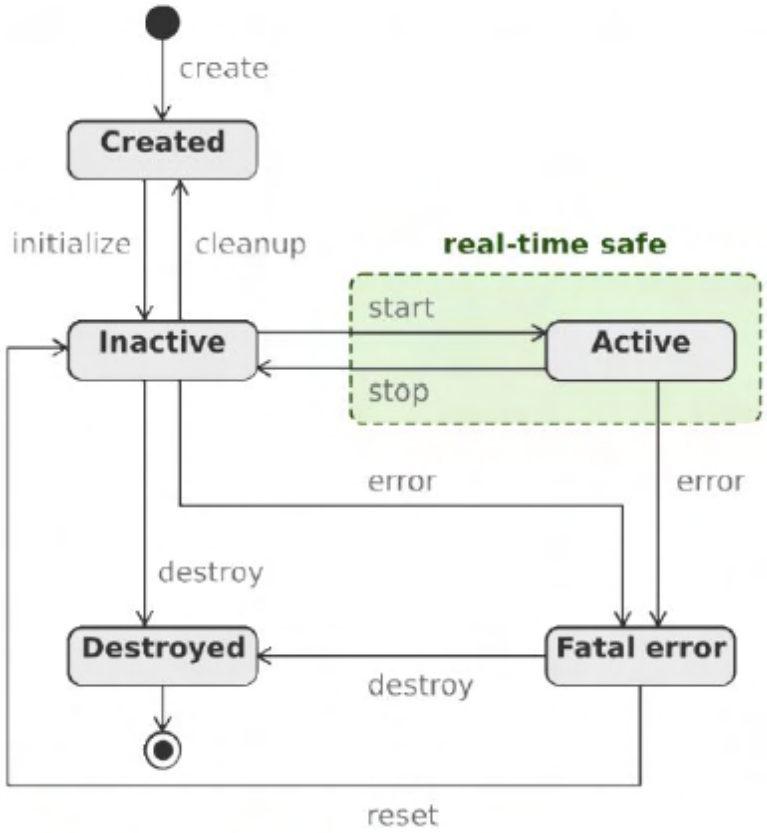
ROS 와 ROS2의 차이점: Real-Time[4]

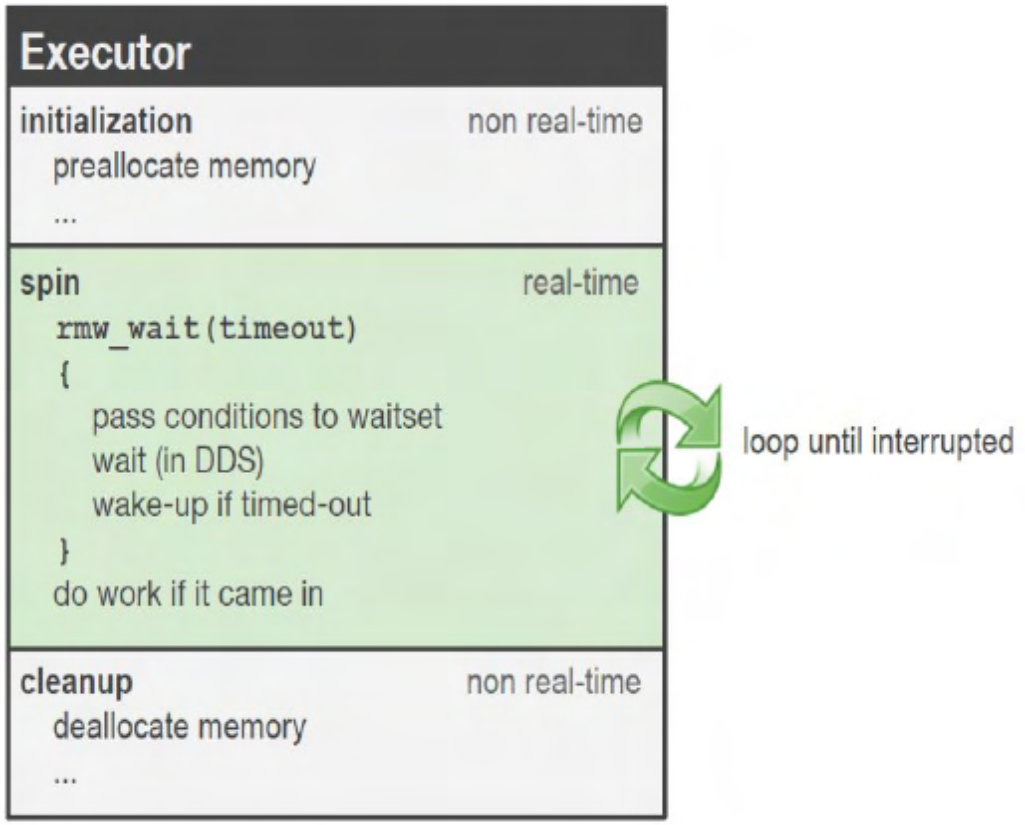
- DDS를 적용한 ROS2의 Pub/Sub 통신 개념



ROS 와 ROS2의 차이점: Real-Time[4]

- **Real-time control & deterministic execution & real-time communication**

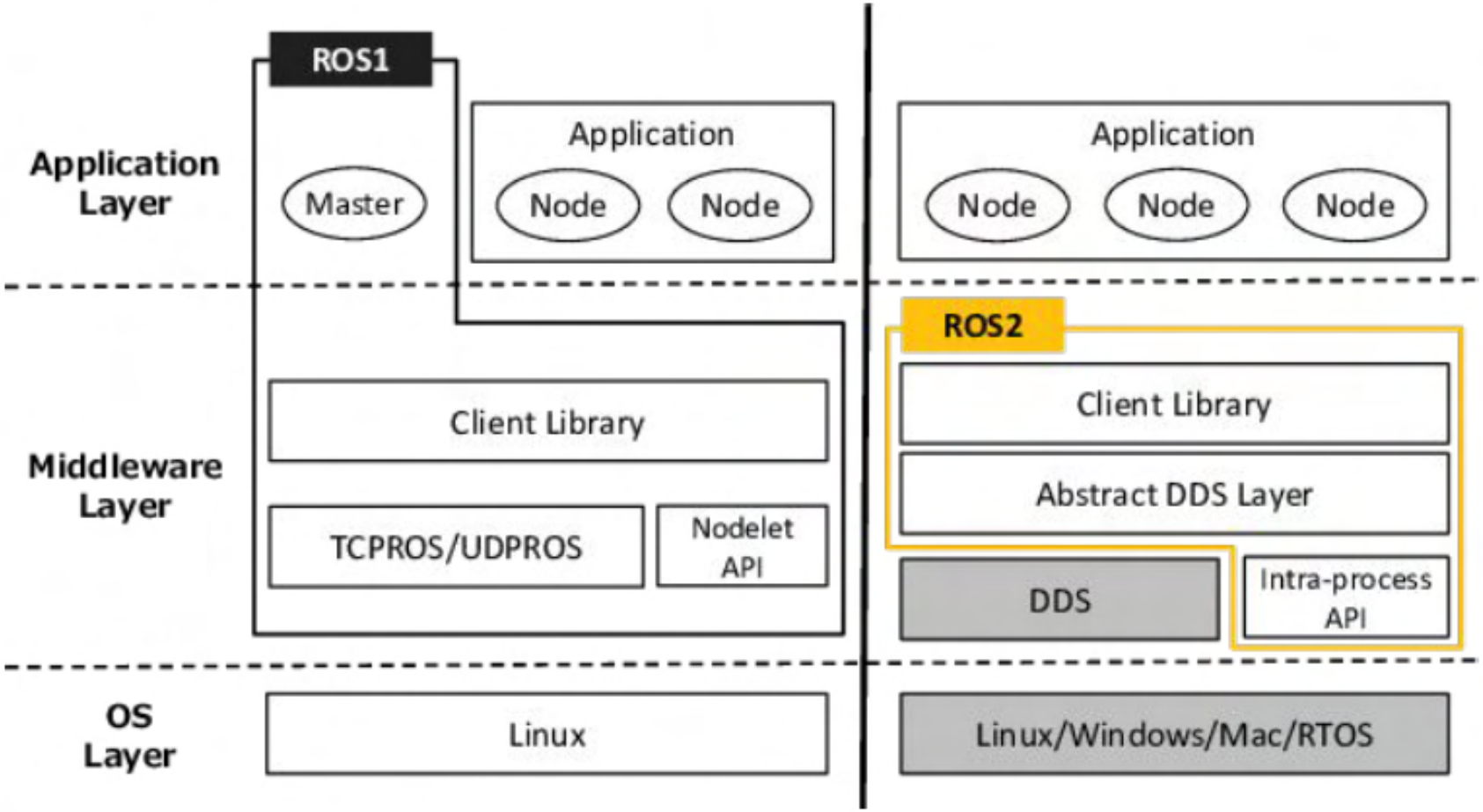




1) Jackie Kay (OSRF) et. al. "Real-time Performance in ROS 2", ROSCon2015

ROS 와 ROS2의 차이점: Real-Time[4]

- **ROS 1 vs ROS 2**



1) Maruyama, Yuya et al. "Exploring the performance of ROS2." *2016 International Conference on Embedded Software (EMSOFT)* (2016): 1-10.

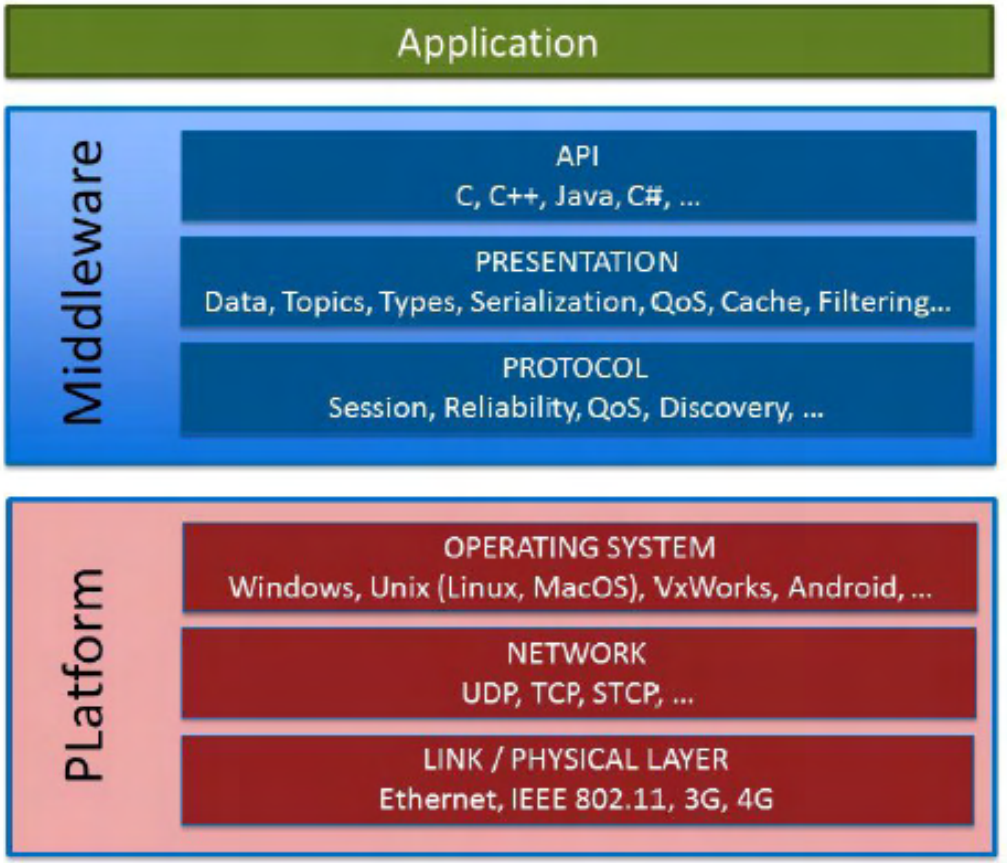
ROS 와 ROS2의 차이점: Real-Time[4]

- **Data Distribution Service**
  - OMG (Object Management Group) 에서 표준화한
    실시간(Real-Time) 발간자-구독자(Publisher-Subscriber) 방식 통신 미들웨어

i. OS Independent

ii. Architecture Independent

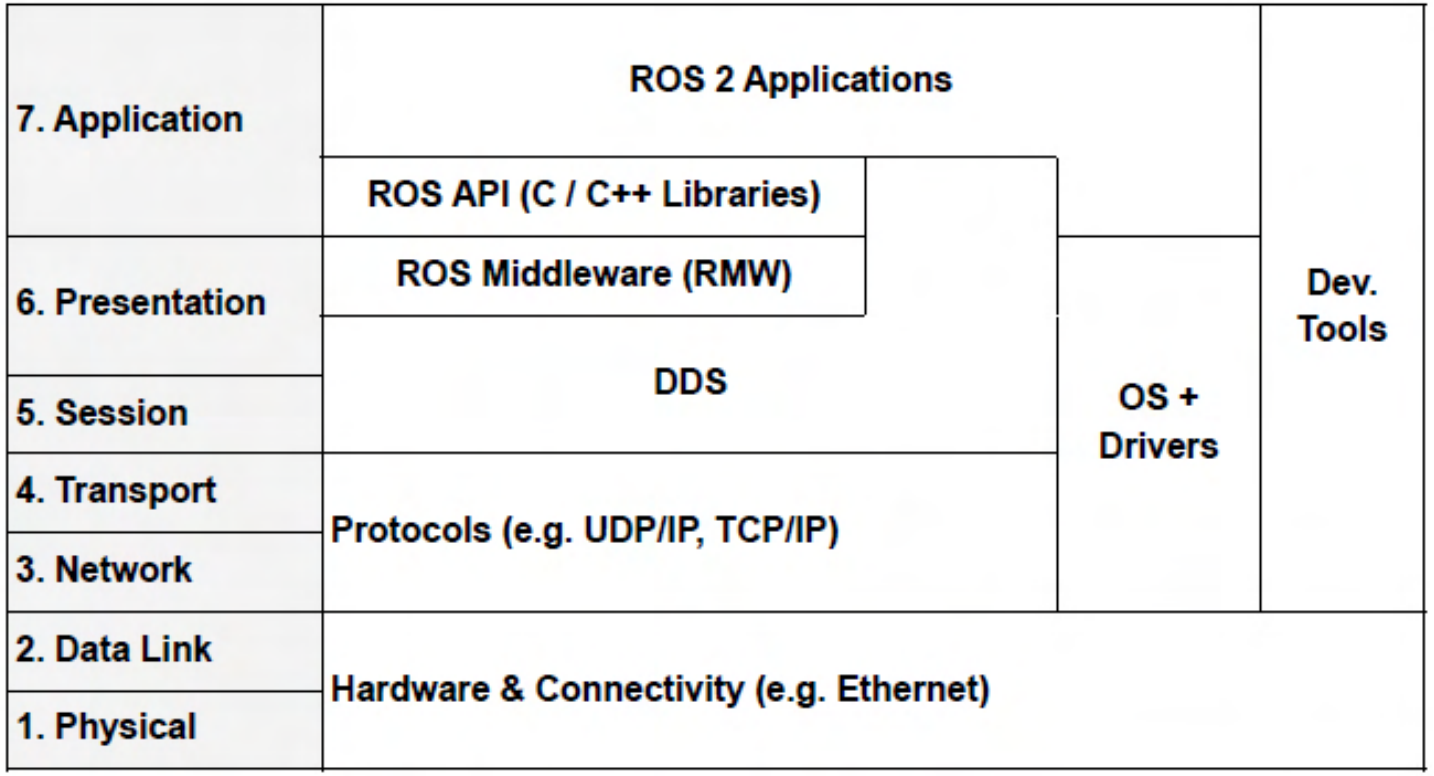
iii. Language Independent



1) https://www.dds-foundation.org/what-is-dds-3/
2) https://en.wikipedia.org/wiki/Data_Distribution_Service

ROS 와 ROS2의 차이점: Real-Time[4]

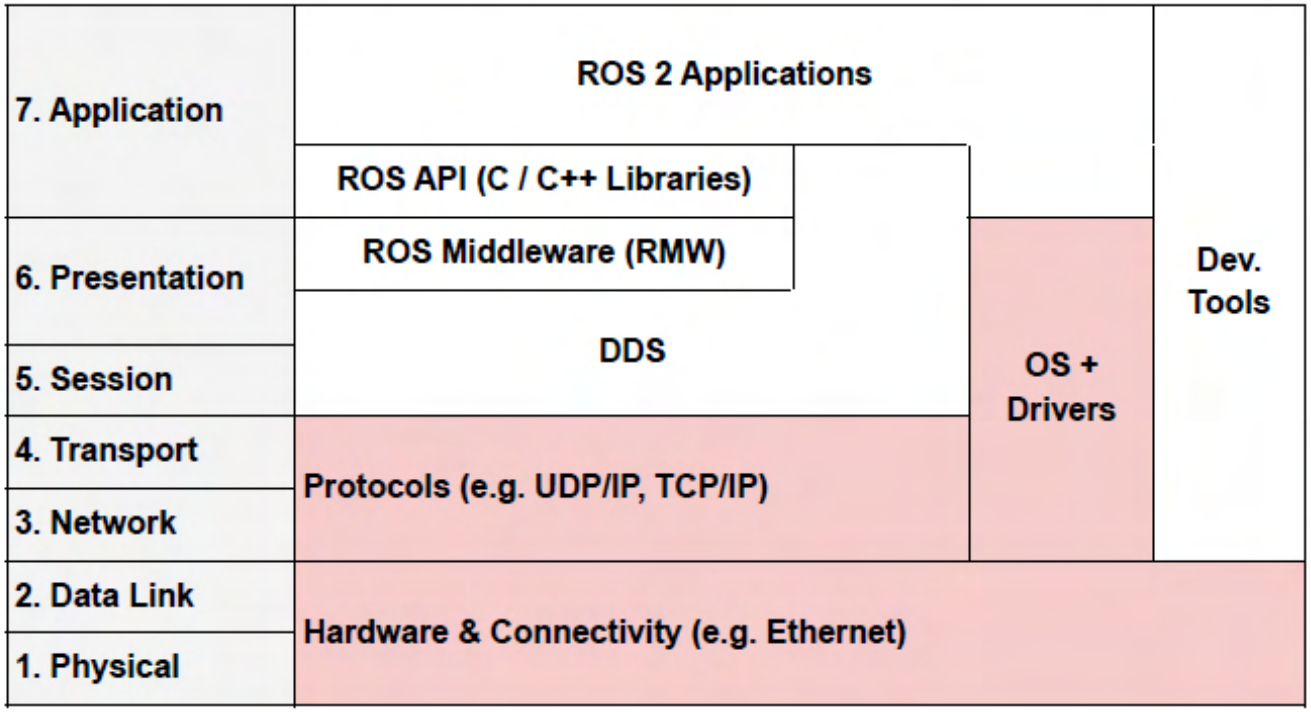
- **Real Time on OSI 7 layers**



1) https://roscon.ros.org/2019/ (Workshop by Apex.AI, Bosch, Alias Robotics, Ubiquity Robotics, Tier IV)
2) https://www.apex.ai/roscon2019 (Slides by Dejan, Víctor, David, Geoff, Ingo, Benjamin)

ROS 와 ROS2의 차이점: Real-Time[4]

- **Real Time on OSI 7 layers**



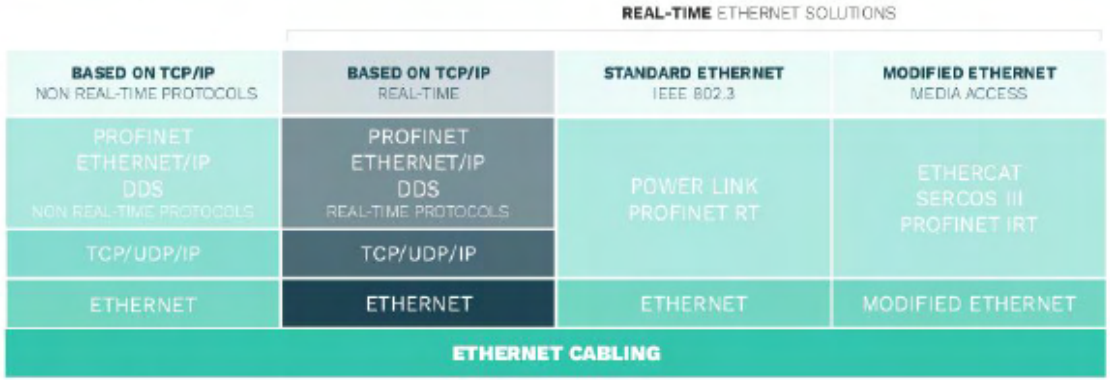
- Real Time device drivers
  - Sensors
  - Network Interface Cards
- Real Time protocol
  - TCP/IP or UDP/IP
- Real Time OS
  - QNX and VxWorks

1) https://roscon.ros.org/2019/ (Workshop by Apex.AI, Bosch, Alias Robotics, Ubiquity Robotics, Tier IV)
2) https://www.apex.ai/roscon2019 (Slides by Dejan, Víctor, David, Geoff, Ingo, Benjamin)

ROS 와 ROS2의 차이점: Real-Time[4]

- **Real Time on OSI 7 layers**

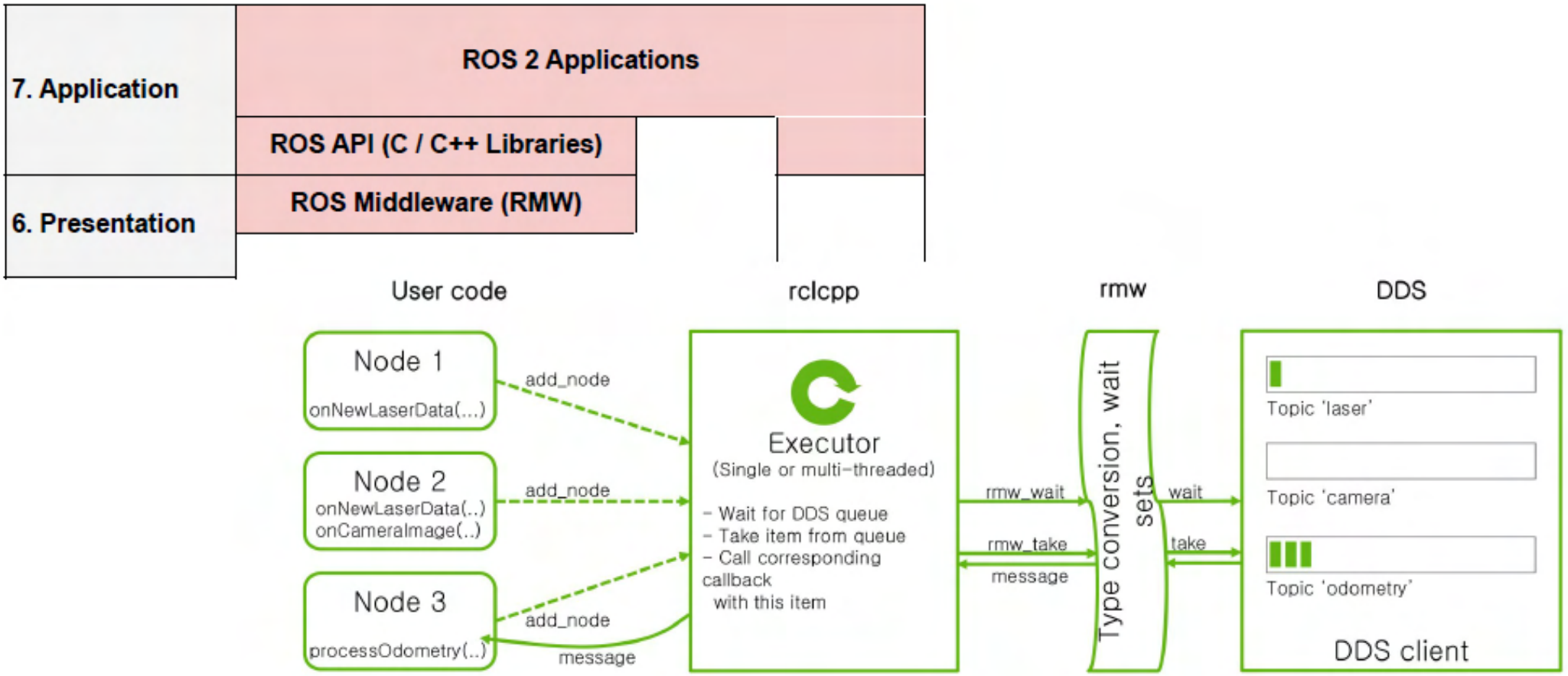
7. Application	ROS 2 Applications
	ROS API (C / C++ Libraries)
6. Presentation	ROS Middleware (RMW)
5. Session	DDS
4. Transport	Protocols (e.g. UDP/IP, TCP/IP)
3. Network	
2. Data Link	Hardware & Connectivity (e.g. Ethernet)
1. Physical	



1) https://roscon.ros.org/2019/ (Workshop by Apex.AI, Bosch, Alias Robotics, Ubiquity Robotics, Tier IV)
2) https://www.apex.ai/roscon2019 (Slides by Dejan, Víctor, David, Geoff, Ingo, Benjamin)

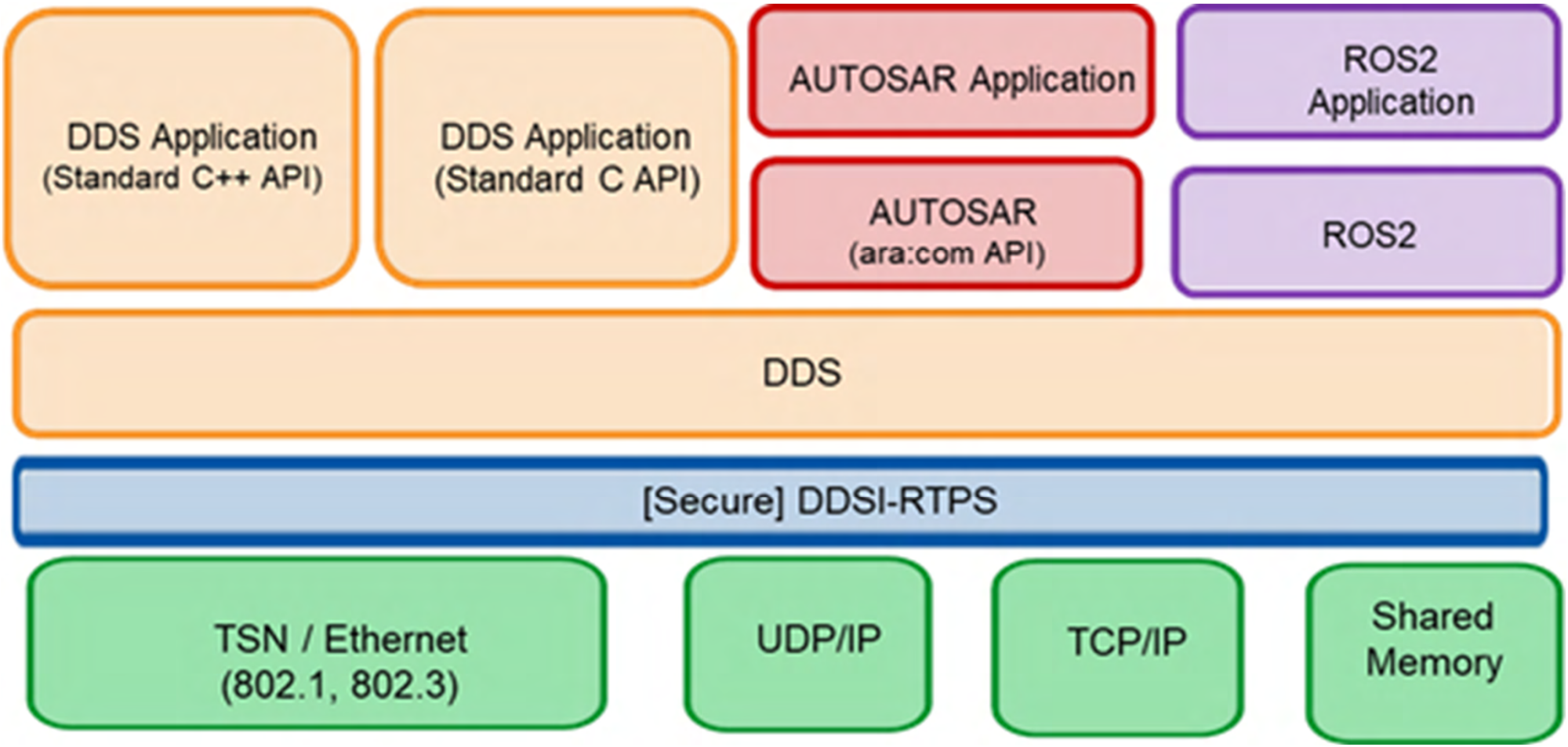
ROS 와 ROS2의 차이점: Real-Time[4]

- **Real Time on OSI 7 layers**

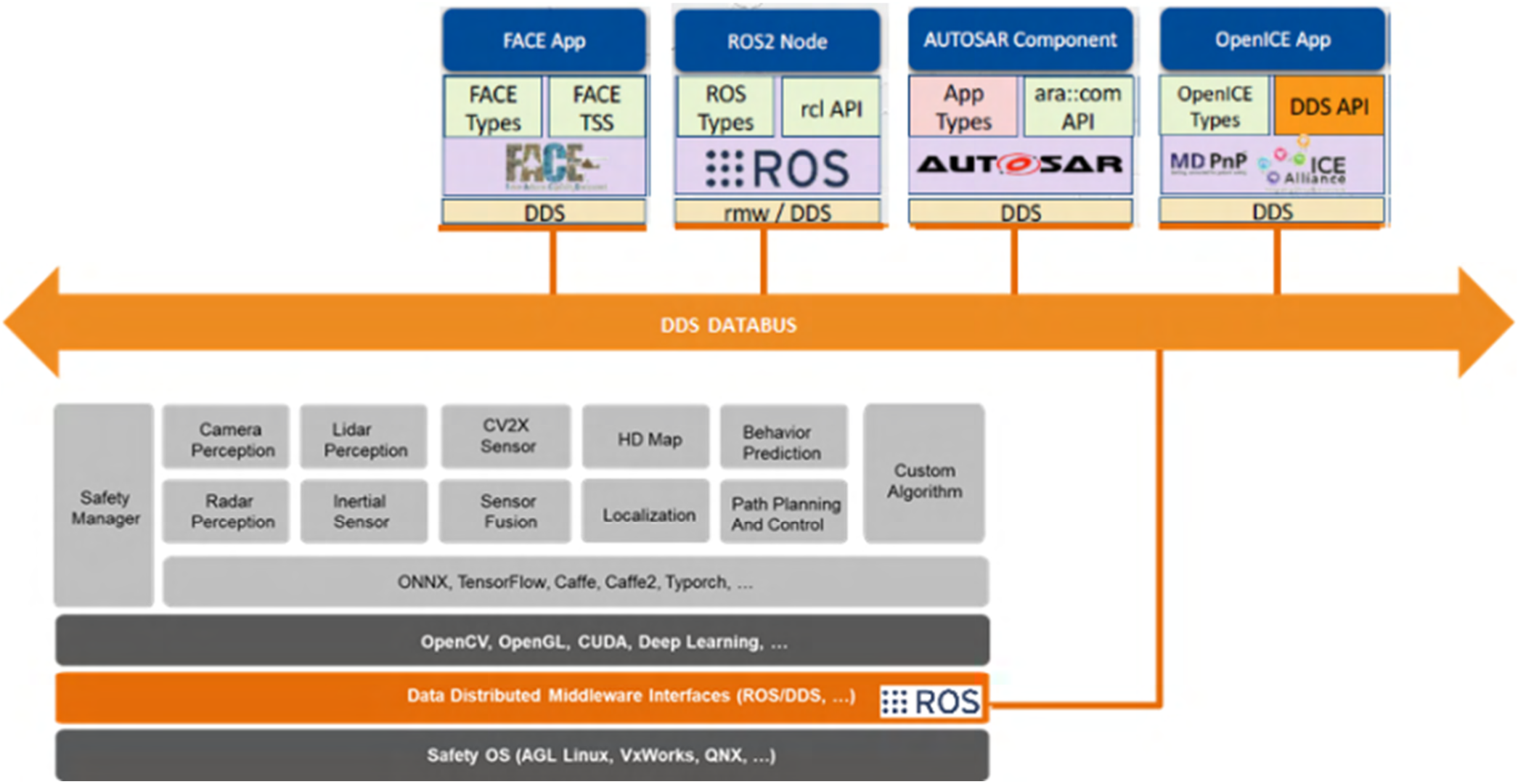


1) https://roscon.ros.org/2019/ (Workshop by Apex.AI, Bosch, Alias Robotics, Ubiquity Robotics, Tier IV)
2) https://www.apex.ai/roscon2019 (Slides by Dejan, Víctor, David, Geoff, Ingo, Benjamin)

ROS 와 ROS2의 차이점: Real-Time[4]

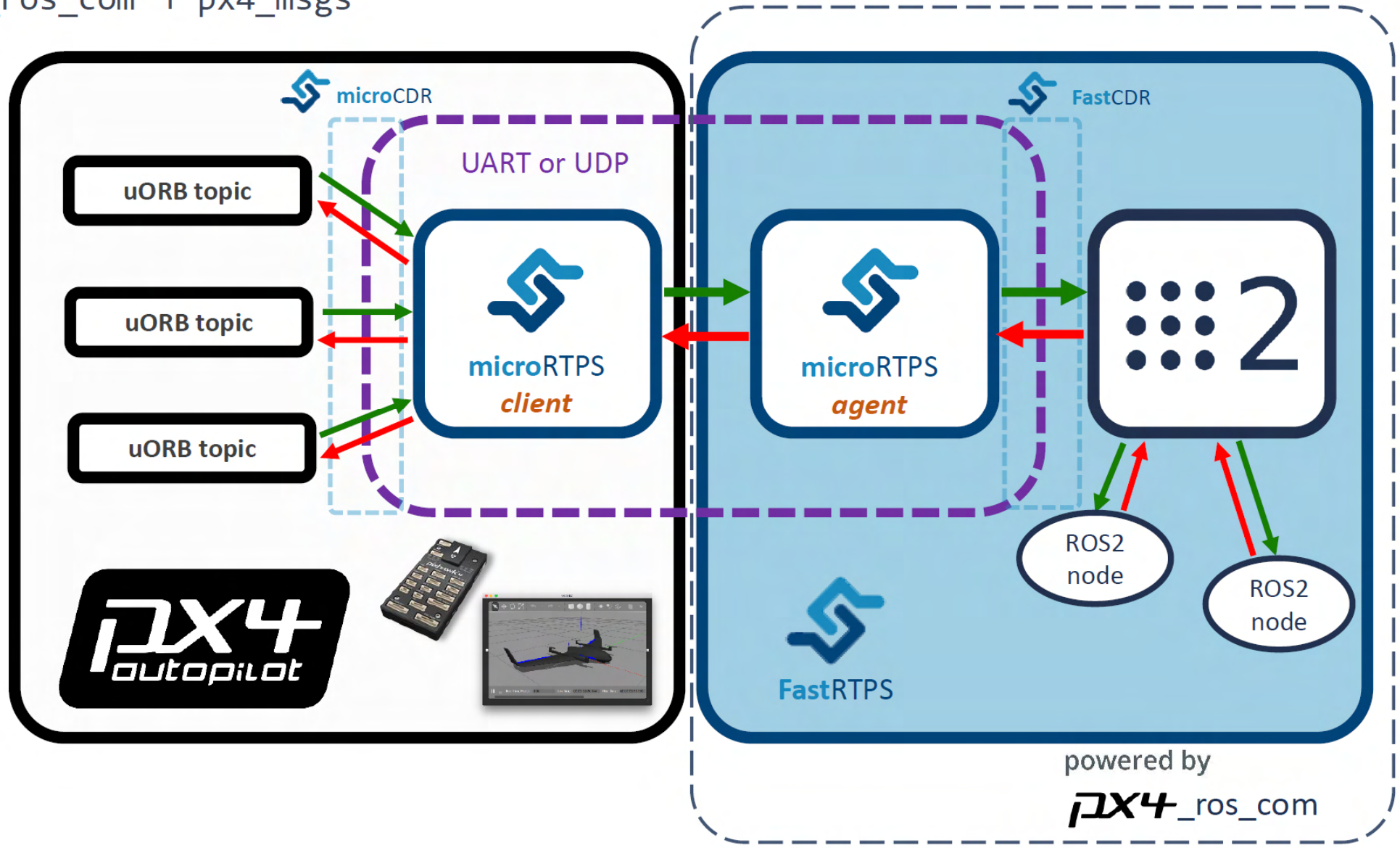


ROS 와 ROS2의 차이점: Real-Time[4]



ROS 와 ROS2의 차이점: Real-Time[4]

- px4_ros_com + px4_msgs



ROS 와 ROS2의 차이점: Real-Time[4]