

## 5-2 ROS2 기본 개념

# 강의 요약

01

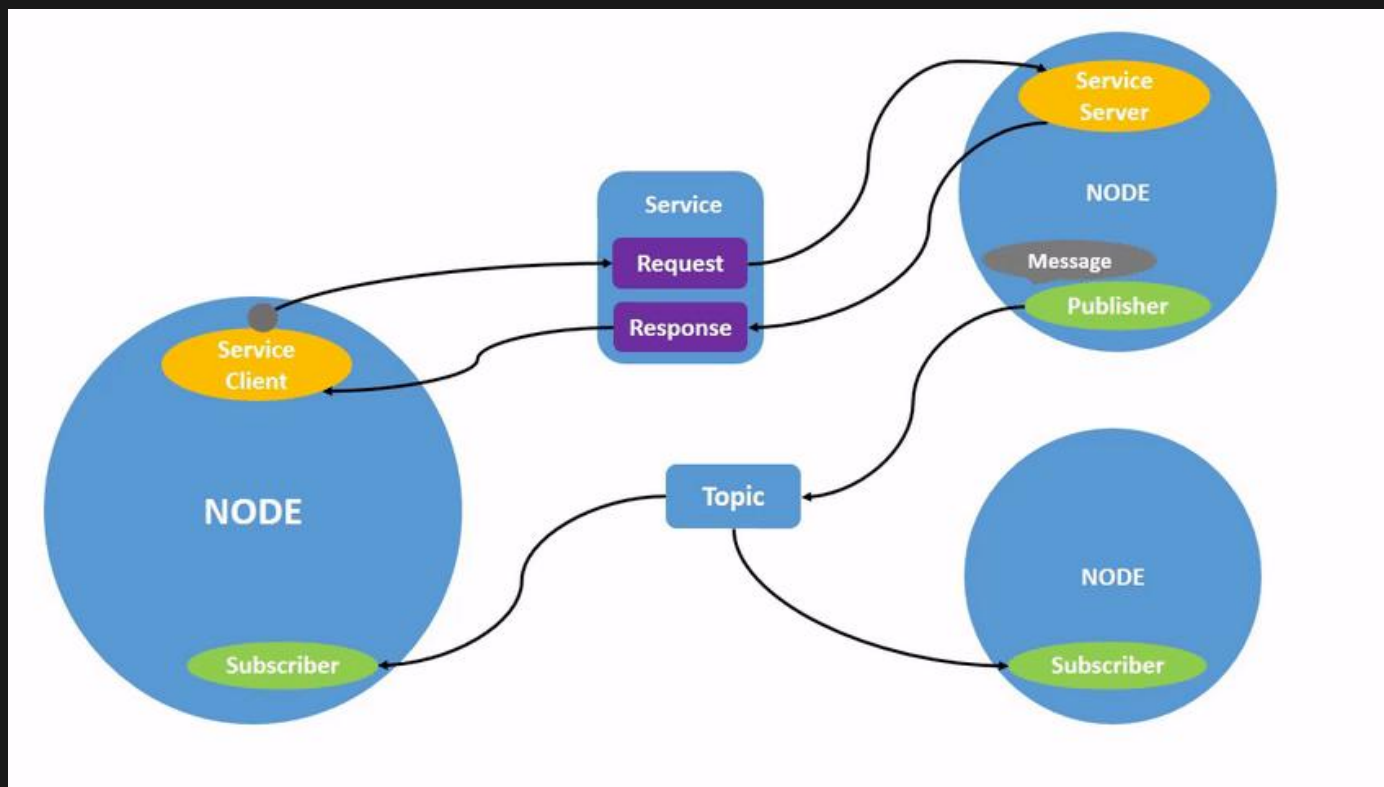
## ROS 의 특징 및 필요성

- 하드웨어와 소프트웨어를 모듈화  
및 연결
- 기업과의 협업

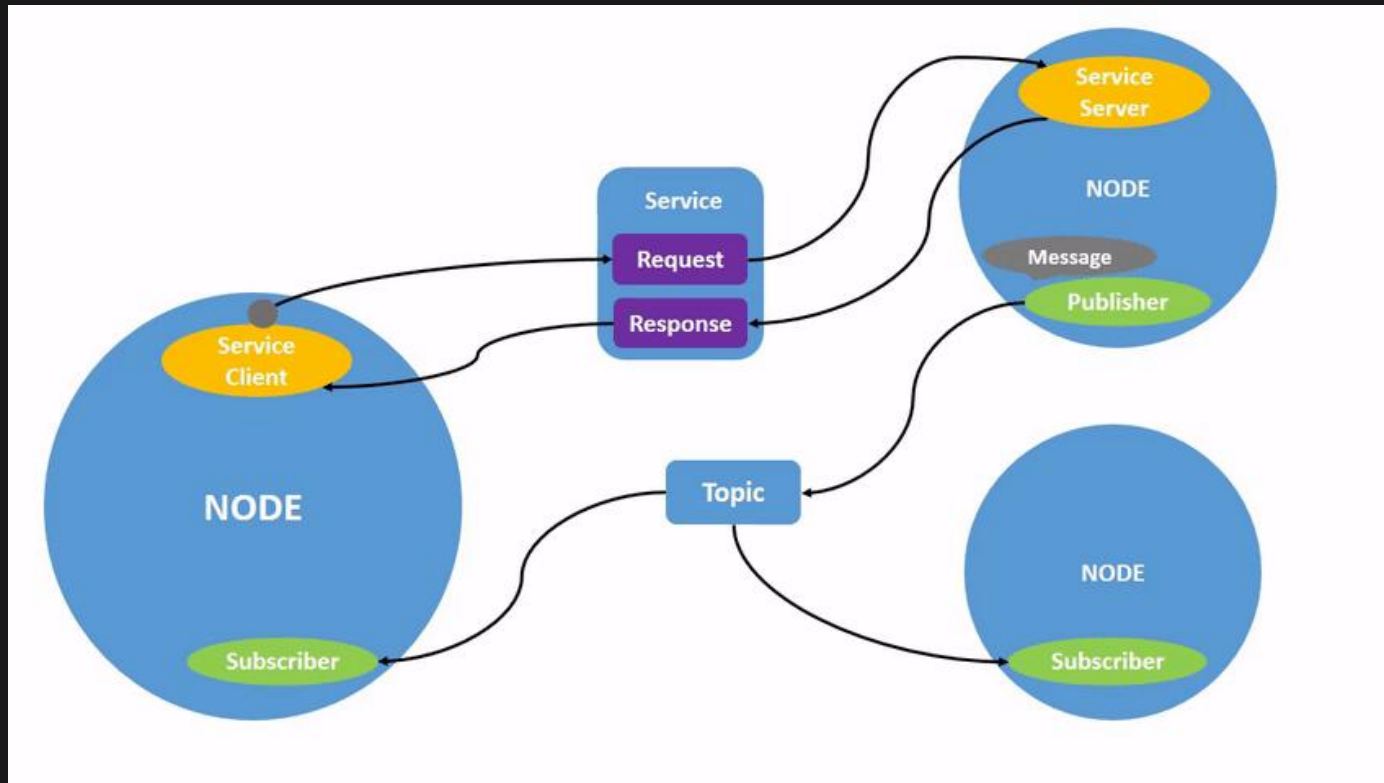
02

## ROS1 vs. ROS2

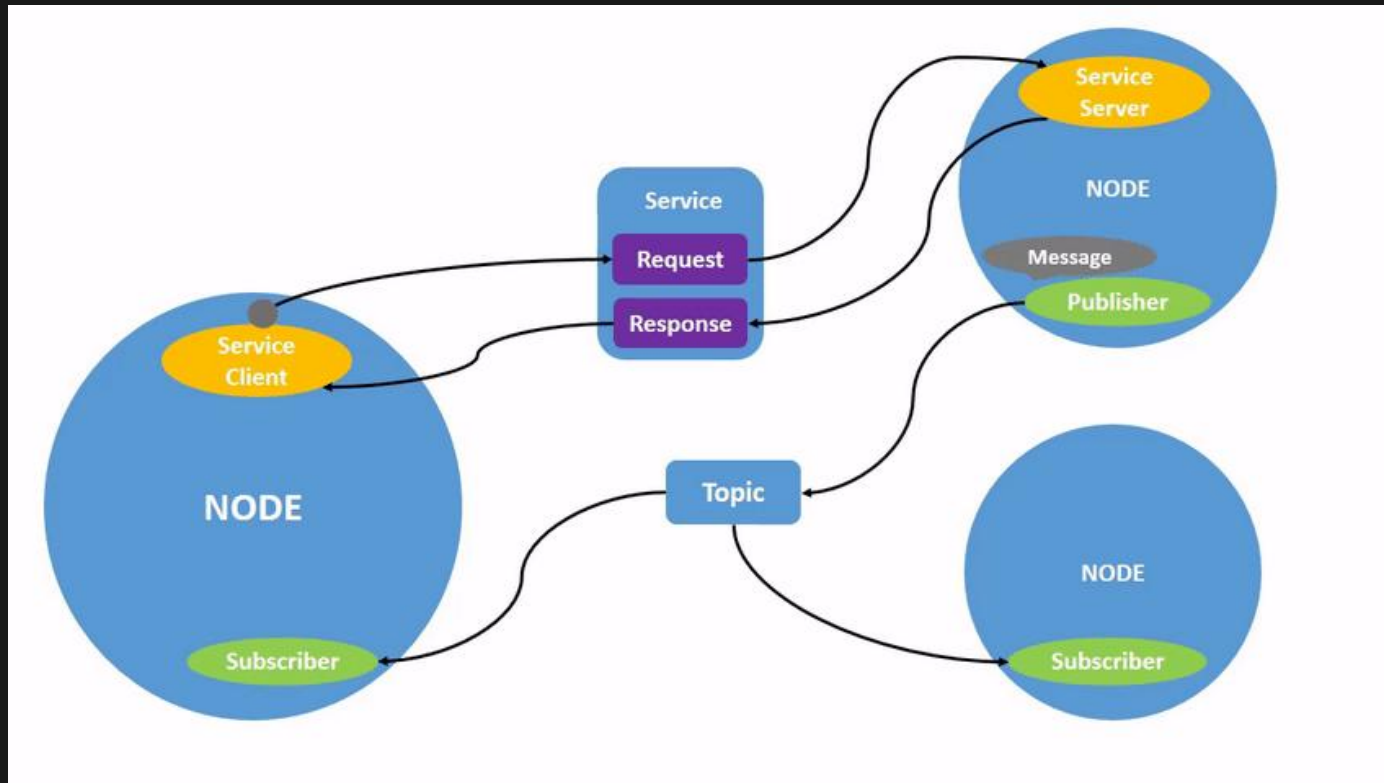
- 구조
- 실시간성
- 로봇 ID
- Python 기반 실행



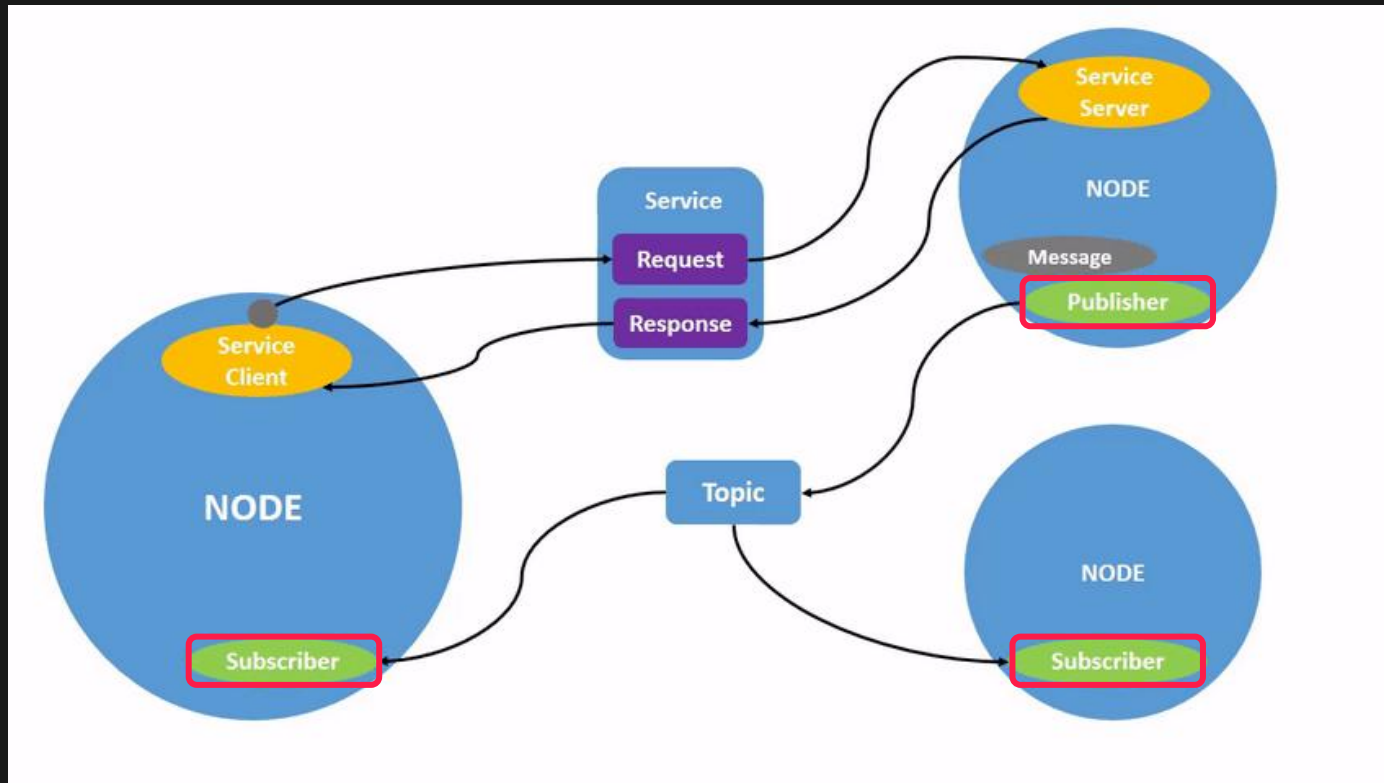
# Message



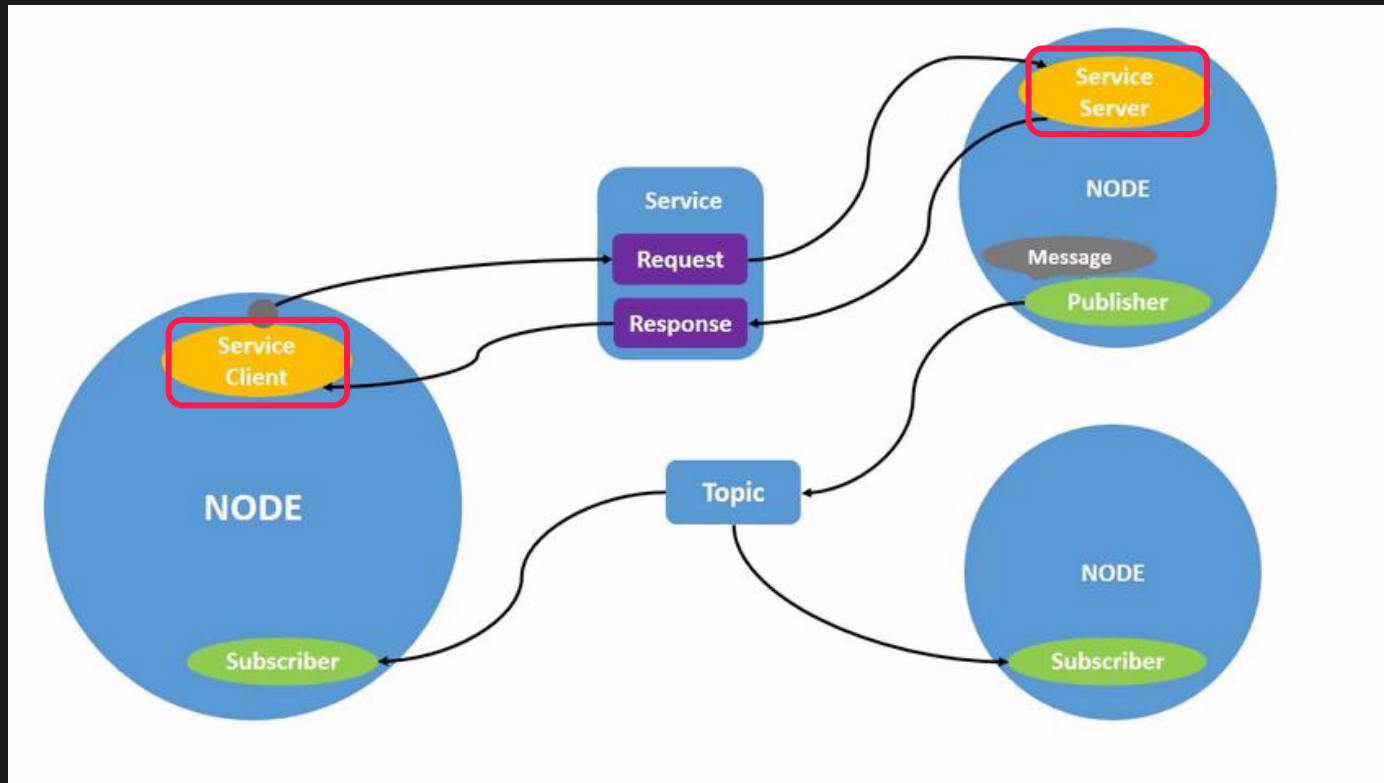
# Message, Node



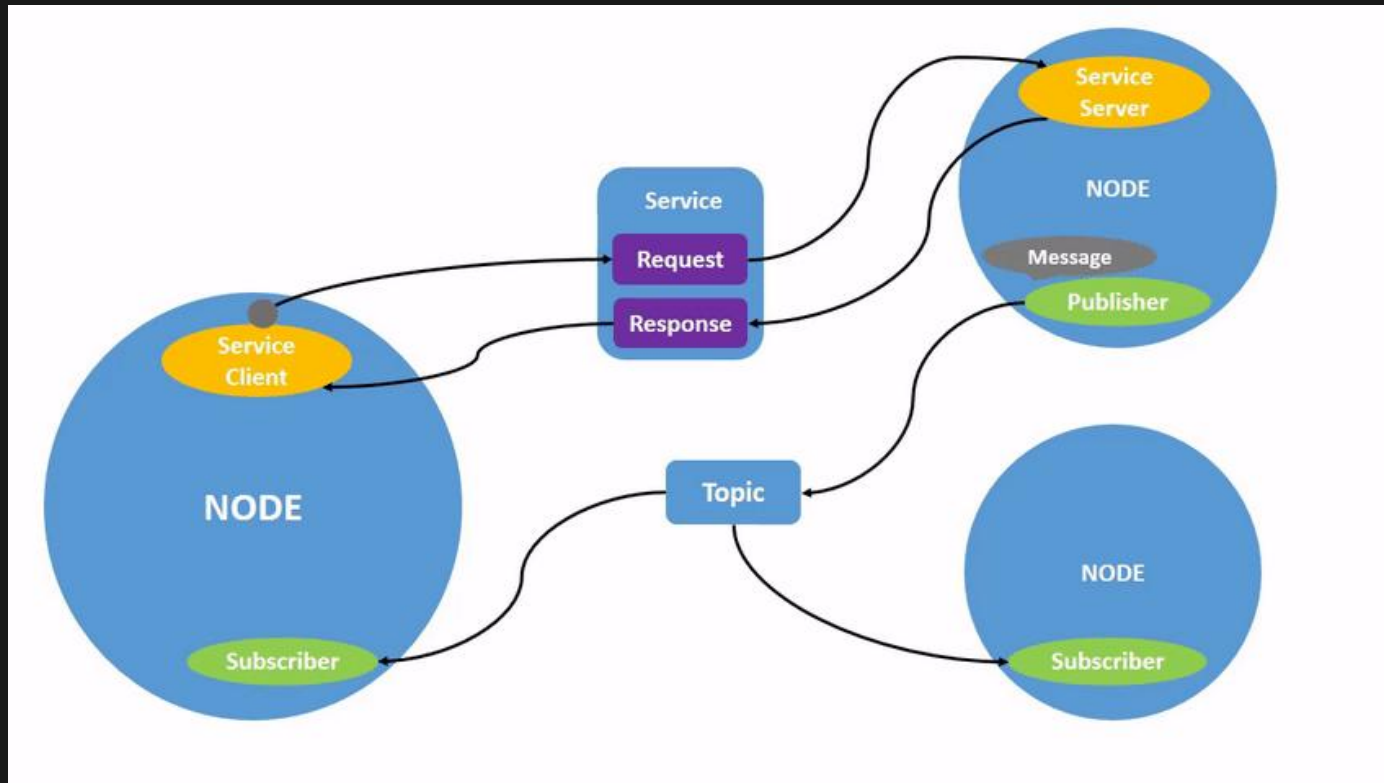
# Message, Node



# Message, Node

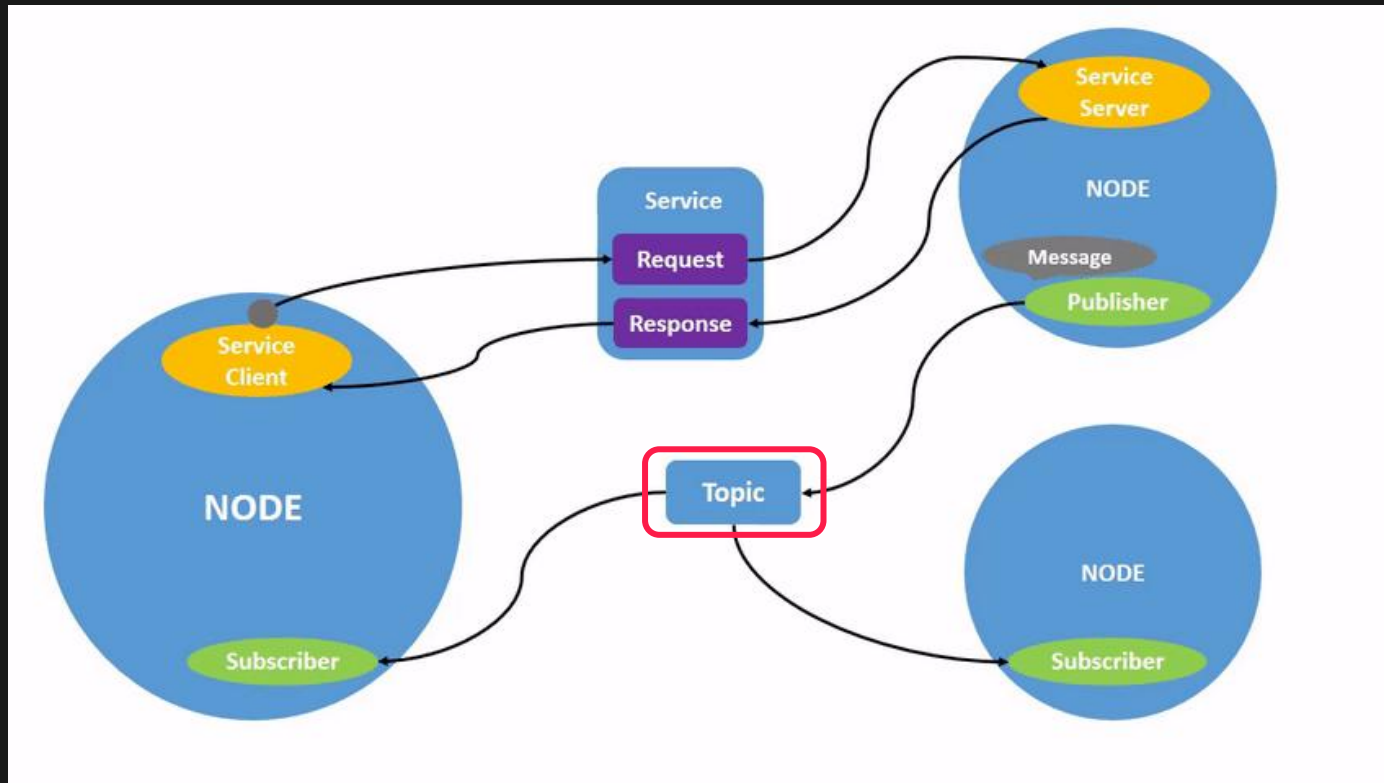


# Message, Node



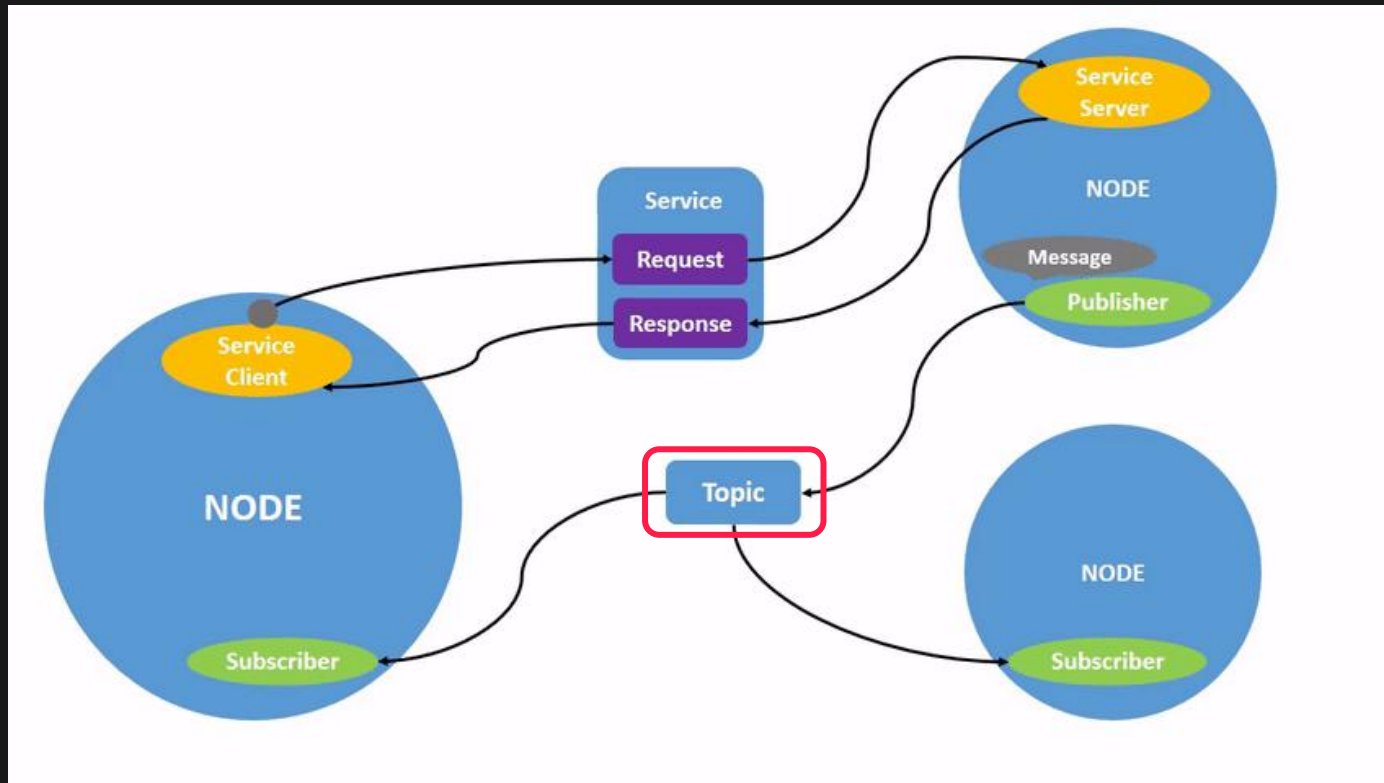


# Message, Node, Topic



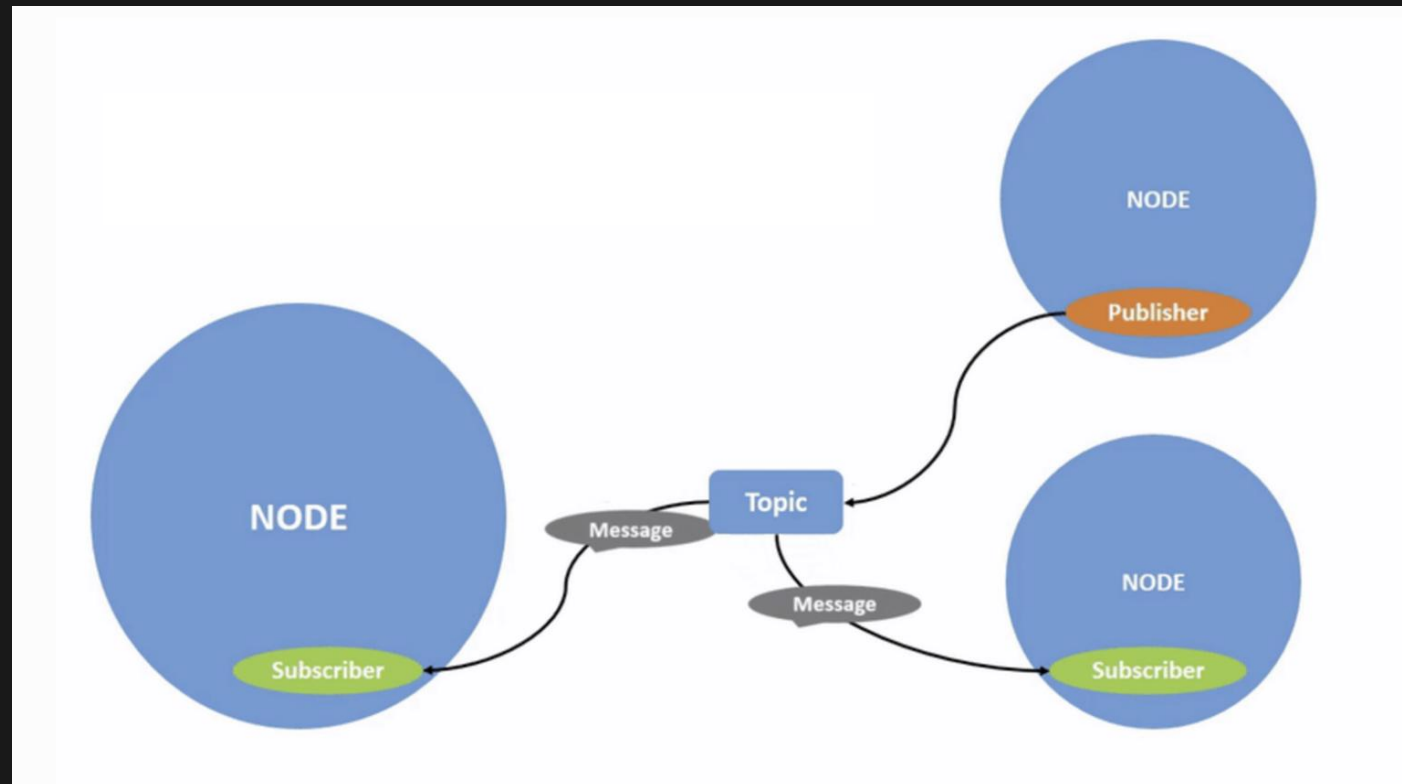
# Message, Node, Topic

Subscriber  
Publisher



# Message, Node, Topic

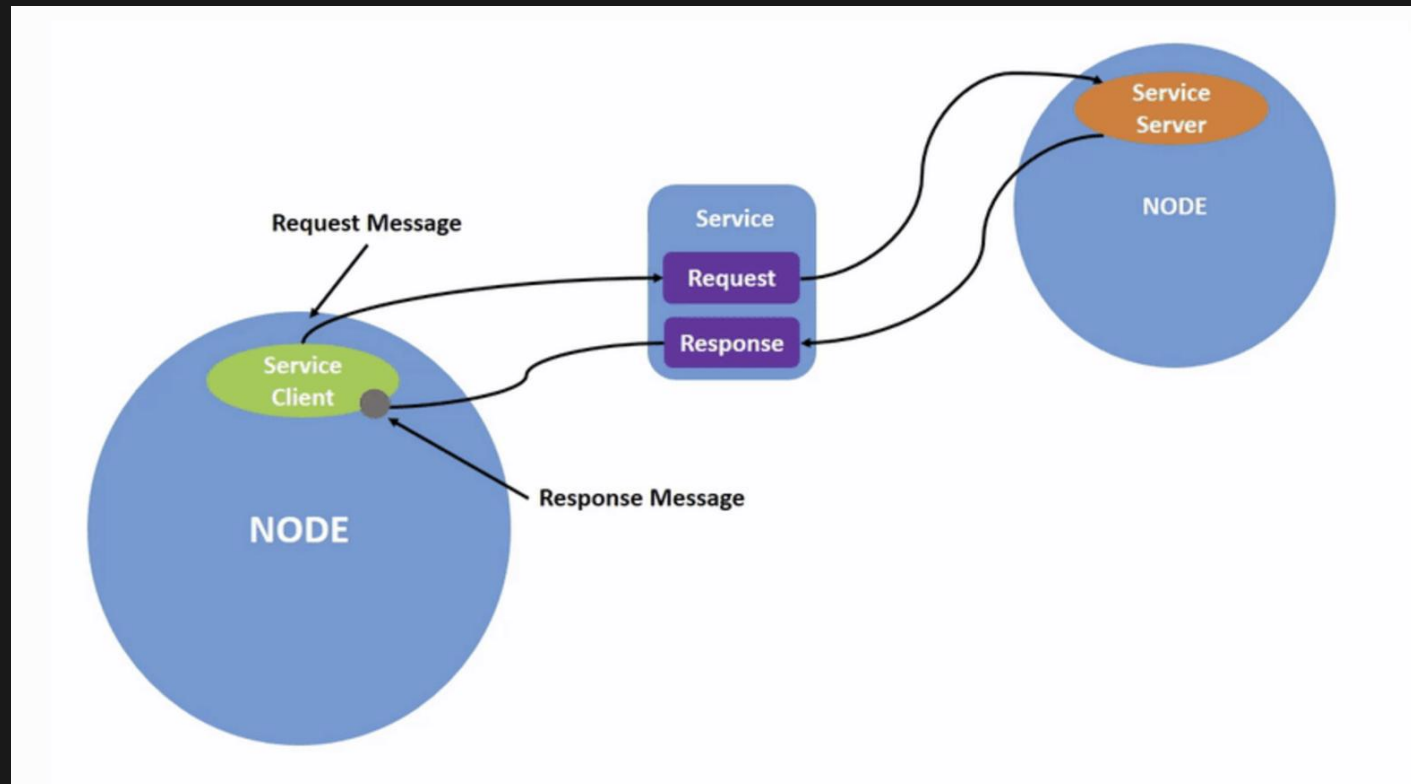
Subscriber  
Publisher



# Message, Node, Topic, Service

Subscriber  
Publisher

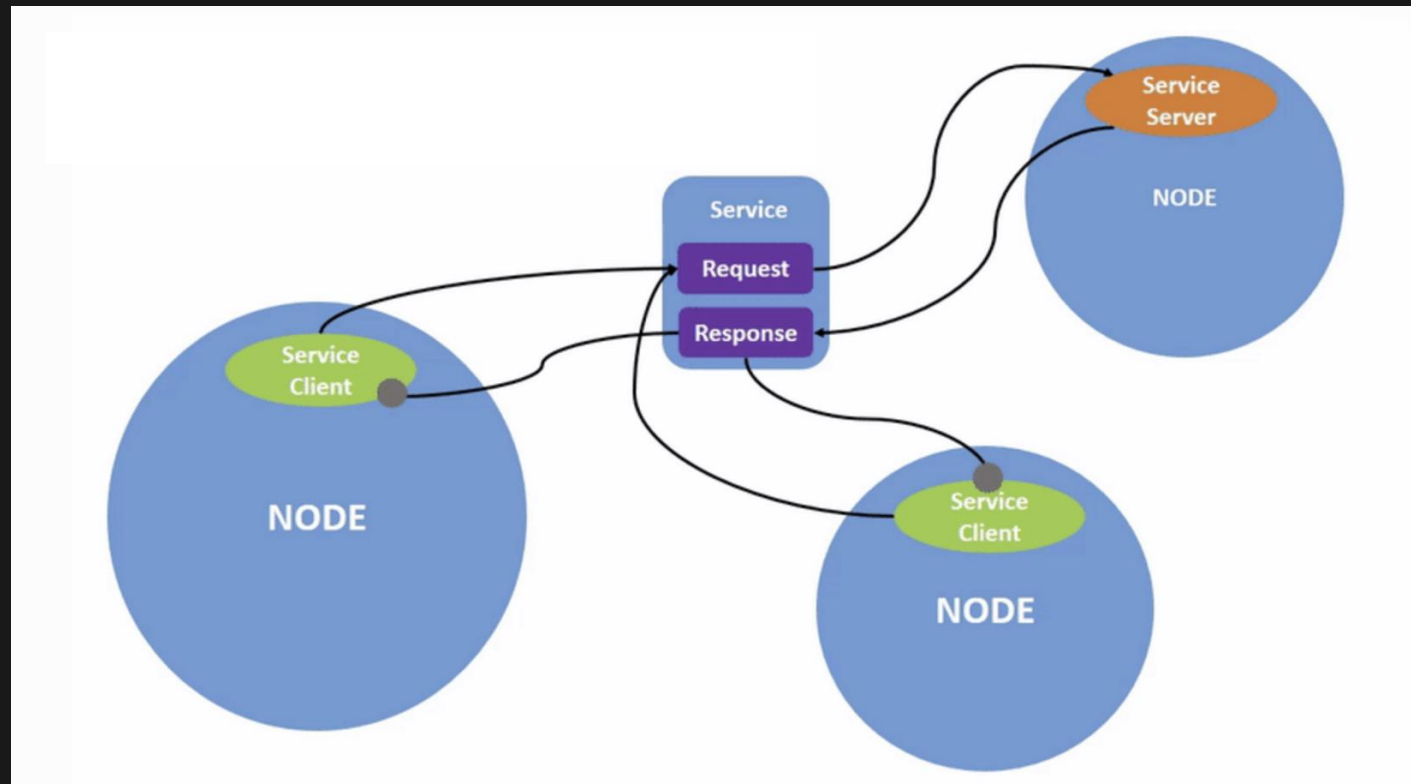
Server  
Client



# Message, Node, Topic, Service

Subscriber  
Publisher

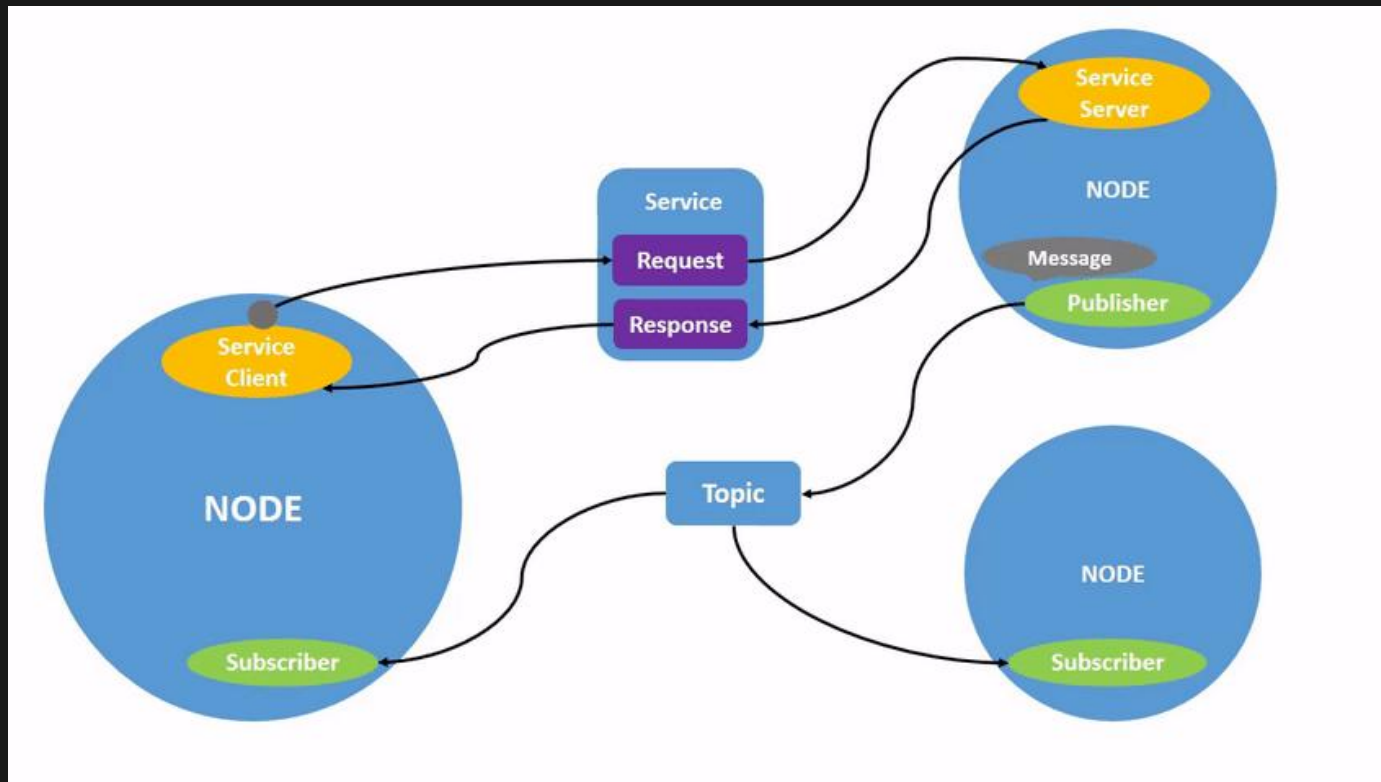
Server  
Client



# Message, Node, Topic, Service, Parameters

Subscriber  
Publisher

Server  
Client

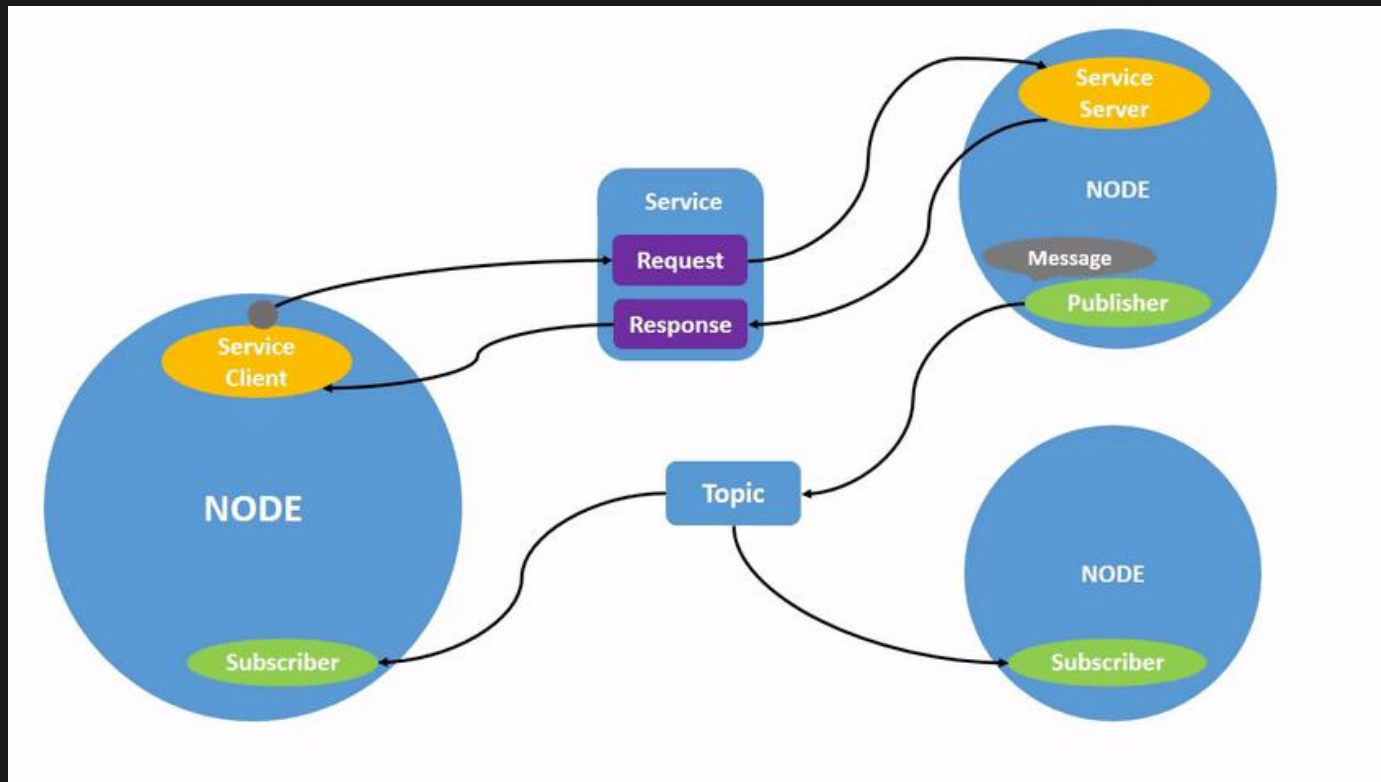


# Message, Node, Topic, Service, Parameters

Subscriber  
Publisher

Server  
Client

진행상황을 알 수 없고 취소가 불가능



# Message, Node, Topic, Service, Parameters, Actions

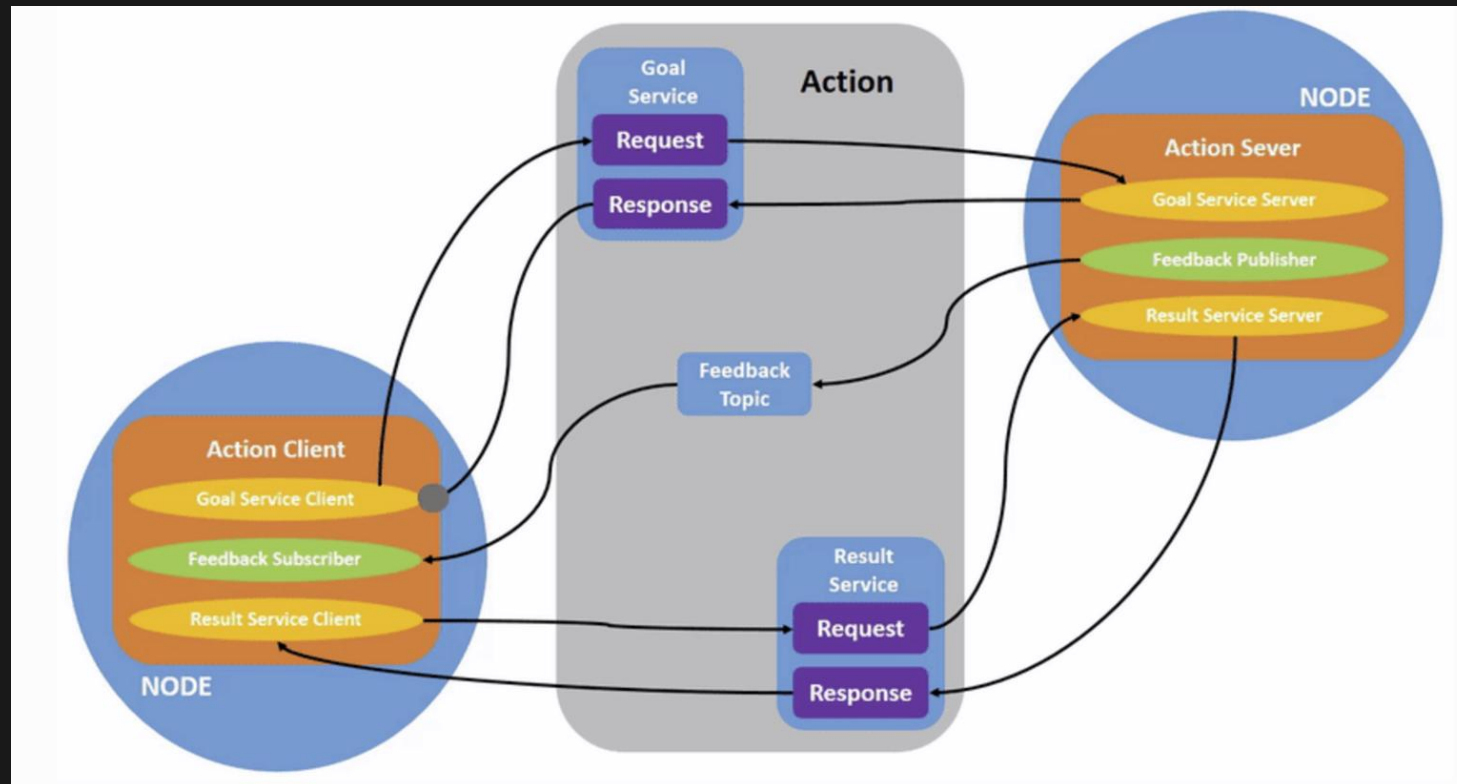
Subscriber  
Publisher

Server  
Client

진행상황을 알 수 없고 취소가 불가능

Server  
Client

Goal → Feedback → Result  
진행 상황을 알 수 있고 취소가 가능





# Message, Node, Topic, Service, Parameters, Actions

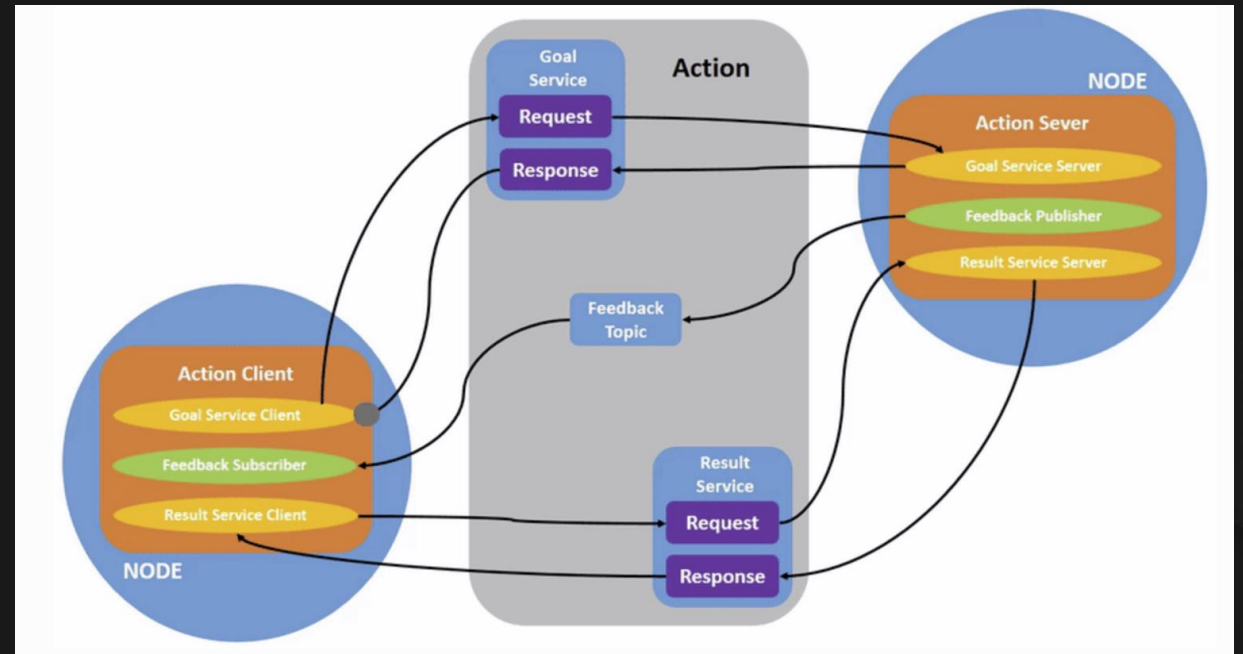
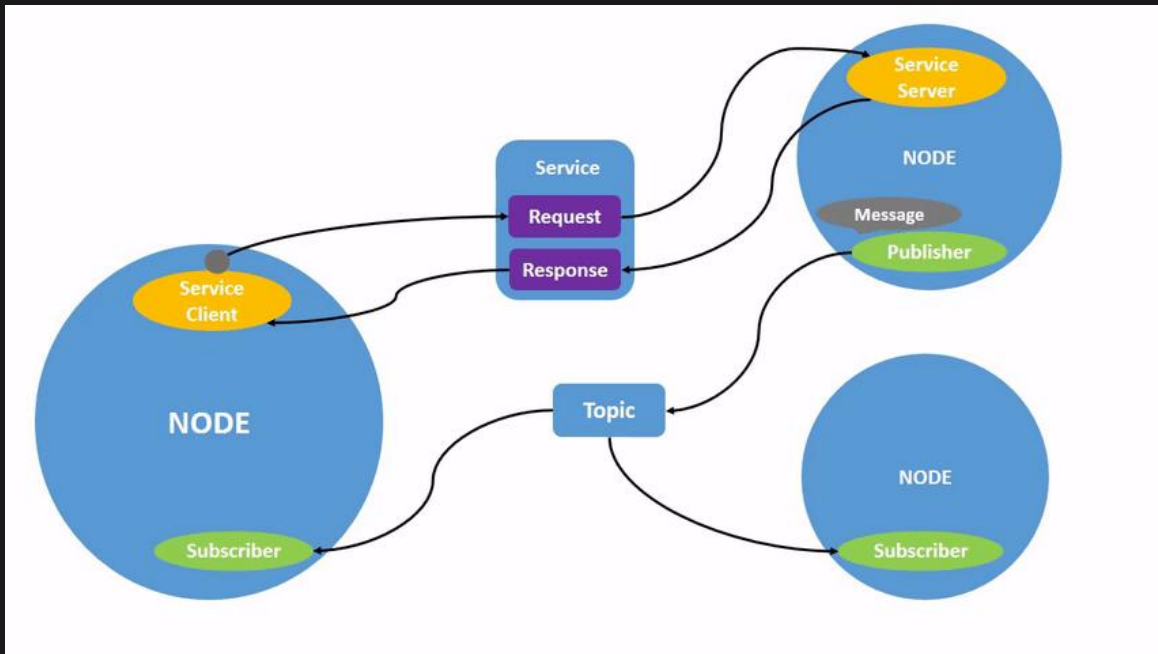
Subscriber  
Publisher

Server  
Client

진행상황을 알 수 없고 취소가 불가능

Server  
Client

Goal → Feedback → Result  
진행 상황을 알 수 있고 취소가 가능



# Package

- 기능별 프로젝트를 관리하는 폴더
- 기능 통합, 분리 등 단위에 제약이 없음

```
my_robot_pkg/  
├─ package.xml  
├─ setup.py  
├─ setup.cfg  
├─ resource/  
│   └─ my_robot_pkg  
├─ my_robot_pkg/  
│   └─ __init__.py  
│   └─ my_node.py
```

# Package (Python)

- 기능별 프로젝트를 관리하는 폴더
- 기능 통합, 분리 등 단위에 제약이 없음

```
my_robot_pkg/  
├─ package.xml  
├─ setup.py  
├─ setup.cfg  
├─ resource/  
│   └─ my_robot_pkg  
├─ my_robot_pkg/  
│   └─ __init__.py  
└─ launch/  
    └─ my_launch.py
```

- **package.xml** : 패키지 정보
- **setup.py** : Python 모듈로 설정하기 위한 스크립트
- **setup.cfg** : setup.py 보조 파일
- **resource/my\_robot\_pkg** : 패키지 리소스 관리
- **my\_robot\_pkg/\_\_init\_\_.py** : Python 패키지로 인식하기 위한 설정파일
- **launch** : 패키지 실행 파일

# Package (C++)

- 기능별 프로젝트를 관리하는 폴더
- 기능 통합, 분리 등 단위에 제약이 없음

```
my_cpp_pkg/  
├─ CMakeLists.txt  
├─ package.xml  
├─ include/  
│   └─ my_cpp_pkg/  
│       └─ my_cpp_node.hpp  
├─ src/  
│   └─ my_cpp_node.cpp  
└─ launch/  
    └─ my_launch.py (선택)
```

- CMakeLists.txt : C++ 핵심 빌드 설정 파일
- package.xml : 패키지 정보
- include : C++ 헤더파일
- src : 사용자가 작성한 실행 코드
- launch : 패키지 실행 파일

# Package (C++)

- 기능별 프로젝트를 관리하는 폴더
- 기능 통합, 분리 등 단위에 제약이 없음
- ROS2 설치와 동시에 기본적으로 제공되는 패키지
  - 코어: rclpy, rclcpp, ros2launch, ...
  - 통신 관련 도구: ros2topic, ros2service, ...
  - 시각화 및 시뮬레이션: rviz2, gazebo, ...
  - 기본 메시지: std\_msgs, geometry\_msgs, sensor\_msgs, ...
  - 기타 툴 패키지: tf2, ...

# Workspace

- ROS2 전체 프로젝트 구조

```
ros2_ws/  
├─ src/  
│   ├── my_robot_pkg/  
│   ├── my_sensors_pkg/  
│   └─ custom_msgs/  
├─ install/  
└─ build/
```

- `ros2_ws` : 워크 스페이스 최상단
- `src` : 패키지를 포함하는 폴더
- 패키지
- 빌드된 실행 파일 (빌드 결과물)
- 중간 빌드 결과물

# 명령어 소개

- Node
  - `ros2 node list`
  - `ros2 node info <node>`
- Topic
  - `ros2 topic list`
  - `ros2 topic echo <topic>`
  - `ros2 topic info <topic>`
  - `ros2 topic type <topic>`
  - `ros2 topic pub <topic> <type> '{data}'`
  - `ros2 interface show <pkg/msg/type>`
- Service
  - `ros2 service list`
  - `ros2 service type <service>`
  - `ros2 service call <srv> <srv_type> '{request}'`
  - `ros2 interface show <pkg/srv/type>`
  - `ros2 service info <srv>`
- Parameter
  - `ros2 param list`
  - `ros2 param get <node> <param>`
  - `ros2 param set <node> <param> <value>`
  - `ros2 param dump <node>`
  - `ros2 param load <node> <file.yaml>`
- Action
  - `ros2 action list`
  - `ros2 action info <action>`
  - `ros2 action send_goal <name> <type> '{goal}'`
  - `ros2 interface show <pkg/action/type>`
- Package
  - `ros2 run <pkg> <executable>`
  - `ros2 launch <pkg> <launch.py>`
  - `ros2 interface list`
  - `colcon ...`

# 강의 요약

01

## ROS 2 구성요소

- Message
- Node
- Topic
- Service
- Parameter
- Action

02

## 패키지 구성

```
my_robot_pkg/  
├─ package.xml  
├─ setup.py  
├─ setup.cfg  
├─ resource/  
│   └─ my_robot_pkg  
├─ my_robot_pkg/  
│   ├── __init__.py  
│   └─ my_node.py
```

03

## 워크 스페이스

```
ros2_ws/  
├─ src/  
│   ├── my_robot_pkg/  
│   ├── my_sensors_pkg/  
│   └─ custom_msgs/  
├─ install/  
└─ build/
```