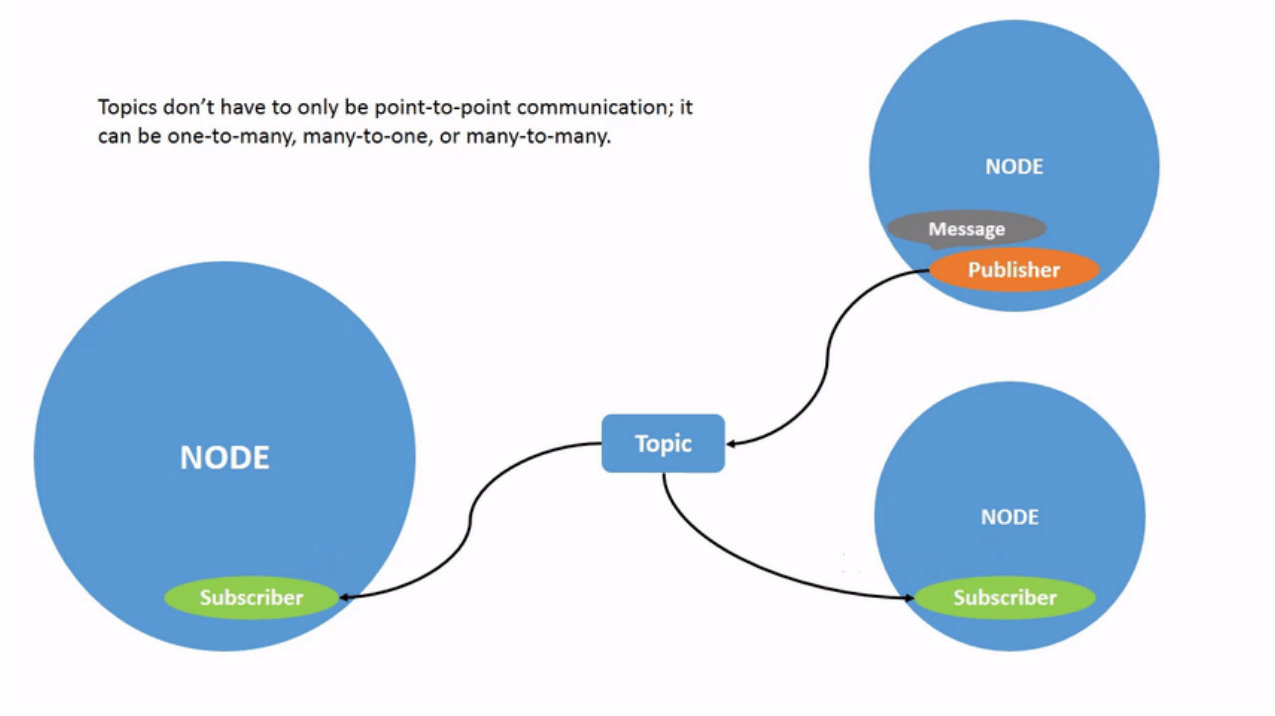
1. Theoretical background

Oficial documentation

- Unidirectional data streaming
- A topic based communication is composed by 2 parts:
  - Publisher.- Element that continuously publish a message
  - Subscriber(s).- Element(s) that subscribe to the topic to get the message



3. Python topics

3.1 Publisher

3.1.1 Structure

1. Structure a node
2. Import the interfaces
3. In the constructor:

    - Instanciate a publisher defining
    - Create **message instances**
    - *If the topic is periodic*, create a **timer** instance as a member of the class, defining the publishing frequency and the publisher callback.
4. Save the message in its attribute ".data"
5. Publish the message
6. In the main function:
    - *If the topic is periodic*, spin the node

3.1.2 Methods

- API

- Publisher instantiation

```
my_publisher = Node.Publisher(<interface>, <topic>, <queue>)
```

- Timer instantiation

```
my_timer = Node.Timer(<seconds>, <callback>)
```

- Modify a message content

```
my_msg = String()
my_msg.data = <new_message>
```

- Publish

```
def callback(self):
        my_publisher.publish(<my_message>)
```

3.2 Subscriber

3.2.1 Structure

1. Structure a node
2. In the **constructor**:
    - Instanciate a subscriber defining: type of message, topic, callback method and the queue size.
3. Create a **subcriber callback**:
    - The message will be received as an object in the first non-self argument
    - Use the atribute".data" to access the message content.
4. In the main function:
    - Spin the subscriber

3.2.2 Methods

- API

- Subscriber instanciation

```
my_subscriber = self.create_subscription(<interface>, <topic>, <callback>, <queue>)
```

3.2.3 Callbacks

- Parameters:
    - Message_object
- Contents:
    - Get the message-value through the attribute **.data**
- Returns:

- Empty

```
def callback(self, msg):
        self.get_logger().info(my_msg.data)
```

4. Topics in C++

5. Remappings (CLI debugging)

- Allow to change the name of a topic when it is executed:

```
ros2 run pkg <node> -ros-args -r <old_topic_name>:=
<new_topic_name>
```