

---

# 「ROS」の基礎と ROS 2プログラミングの実践

## 3. ROS 2概論

高瀬 英希

(京都大学／JSTさきがけ)

[takase@i.kyoto-u.ac.jp](mailto:takase@i.kyoto-u.ac.jp)



# プログラム

---

## 3. ROS 2概論 [day2 09:30-10:00]

- ROS 2の開発背景と目指すところ
- ROS 2の新機能

# ROSのユースケースの変遷

- 単体ロボット
- ワークステーション級の計算資源
- 非リアルタイムな制御
- 理想的かつ安定的なネットワーク環境
- 研究用途



- 複数ロボット(協調動作)
- 組込み規模での動作実現の要求
- リアルタイム制御
- 不安定なネットワーク環境(遅延や損失への対応)
- 実製品への適用

ROSをver.2として  
フルスクラッチから  
開発し直す！

⋮⋮⋮ 2

# ROS 2の新機能

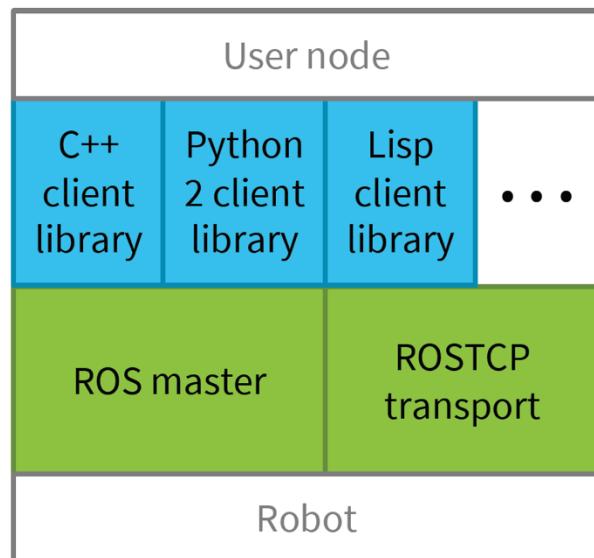
---

- DDS (Data Distribution Service)
- Quality of service
- Lifecycle state machine
- Actions
- Intra-process communication
- ros1bridge
- Multi platform support
- colcon build tool
- Python launch system

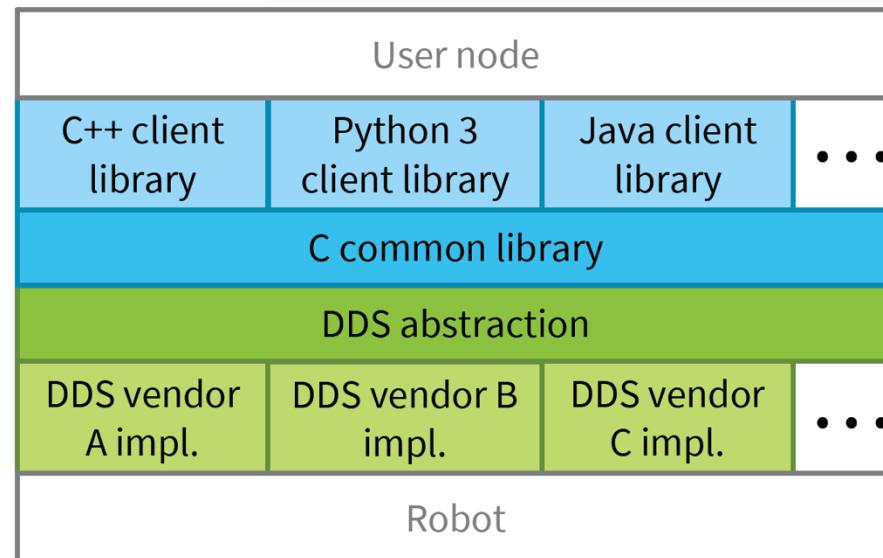
# DDSとソフトウェア構造の変更

- Data Distribution Service
  - (ROSに似た) pub/sub通信プロトコル
    - ✓masterが不要となる
  - OMG標準仕様として規定されている

<https://speakerdeck.com/youtalk/dds>



ROS 1



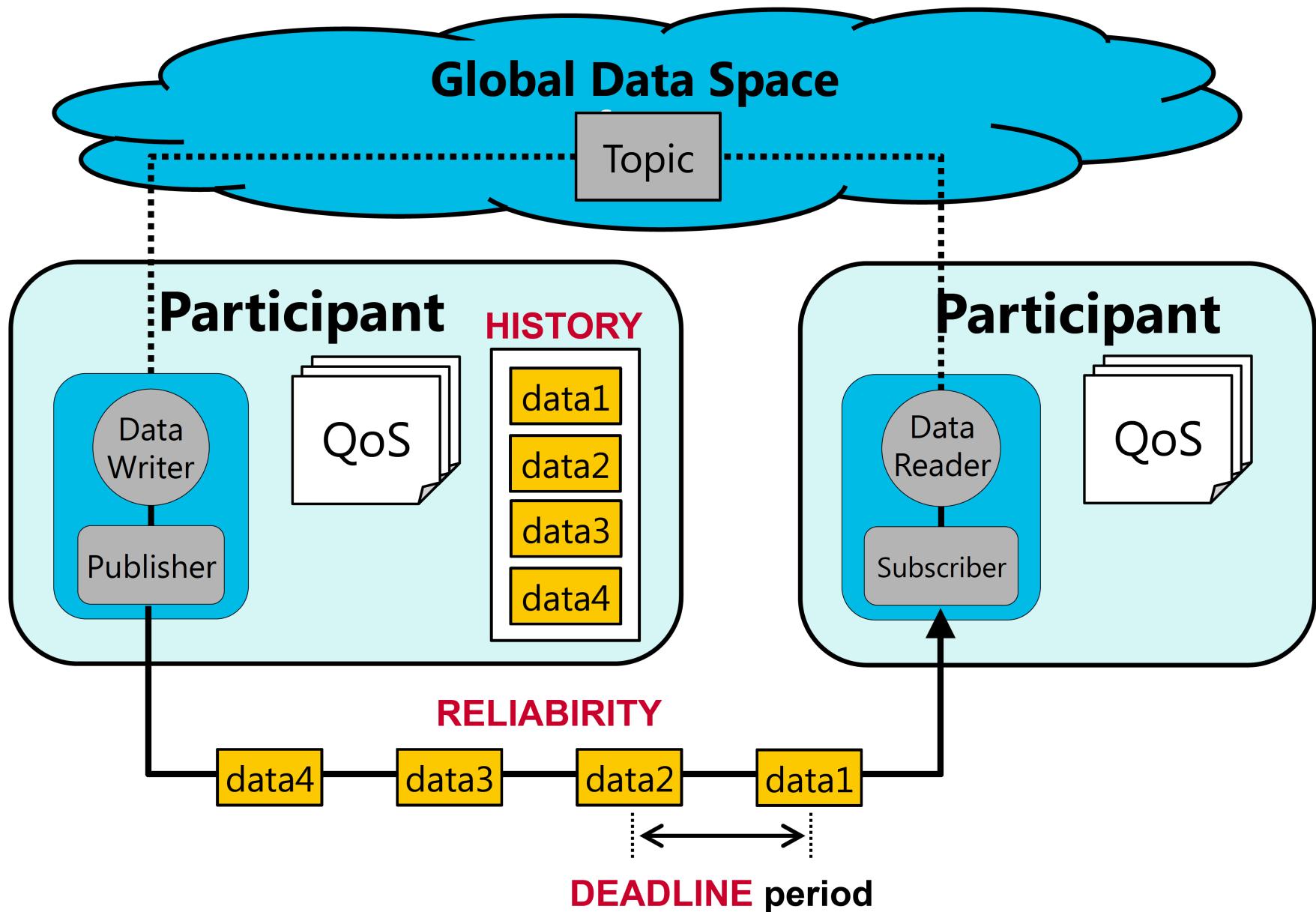
ROS 2

# ROS 2対応DDSの実装

[REP-2000](#) を基に  
作成・整理

Middleware Library	Provider	Support	ライセンス・備考等
<code>rmw_fastrtps_cpp</code>	eProsima Fast-RTPS	Tier 1	Apache 2.0, ROS 2デフォルト, 軽量実装
<code>rmw_connexx_cpp</code>	RTI Connexx	Tier 1	商用／研究(機能制限)ライセンス, No.1 ベンダ, PF/Arch.サポートは限定的
<code>rmw_cyclonedds_cpp</code>	Eclipse Cyclone DDS	Tier 2	Eclipse Public License 2.0(Open), 高性能・高信頼な評価結果
<code>rmw_opensplice_cpp</code>	ADLINK OpenSplice	Tier 2	商用／オープン(機能制限, Apache 2.0), PFサポートは限定的, Foxyで対象外
<code>rmw_fastrtps_dynamic_cpp</code>	eProsima Fast-RTPS	Tier 2	Apache 2.0, 型の実行時の解釈・変換 (type introspection)をサポート

- 実装機能やライセンス形態で複数のDDSから通信層を選択できる
- 異種DDSを選択したノード間で通信できる (rmw層で抽象化)
  - Tier 1: Open Robotics公式の手厚いサポート
  - Tier 2: 公式だが限定的なサポート

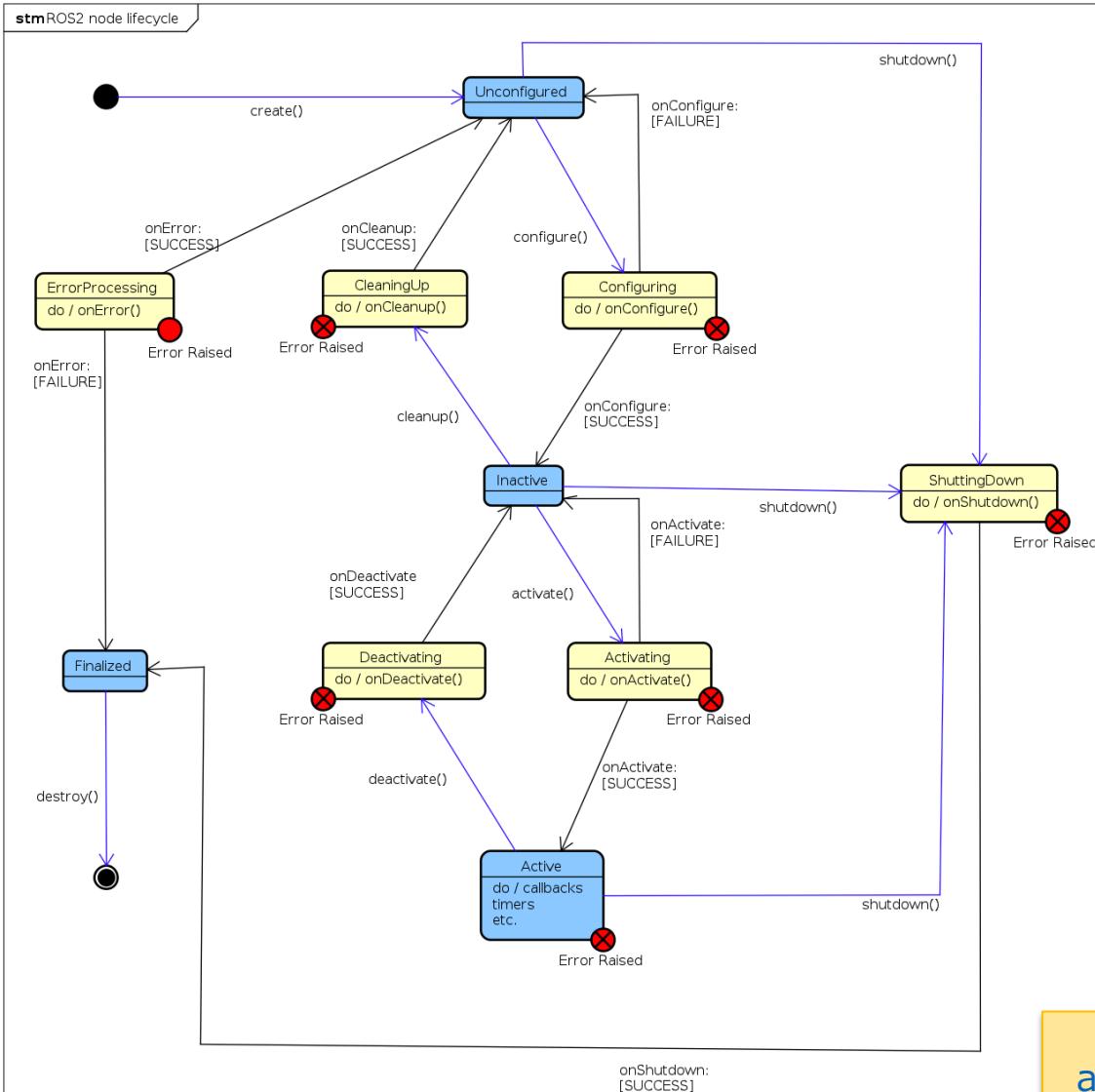


# QoS Control

- 通信経路の品質を指定する
  - Pub/Sub node間で互換性が必要  
(SubのQoSプロファイルのほうが厳しい必要がある)
- 指定できるパラメータの例 (下線はデフォルト設定)
  - History (履歴): Keep last (任意の値数を保持) or Keep all
  - Depth (深さ): Size of the queue (Keep lastの個数 10)
  - Reliability (信頼性): Best effort or Reliable (リトライ試行)
  - Durability (耐久性): Transient local or Volatile (遅延非許容)

Profile	History policy	Reliability	Durability
Default	Keep last 10	Reliable	Volatile
Services	Keep last 10	Reliable	Volatile
Sensor data	keep last 5	Best effort	Volatile
Parameters	keep last 1000	Reliable	Volatile

# Lifecycle

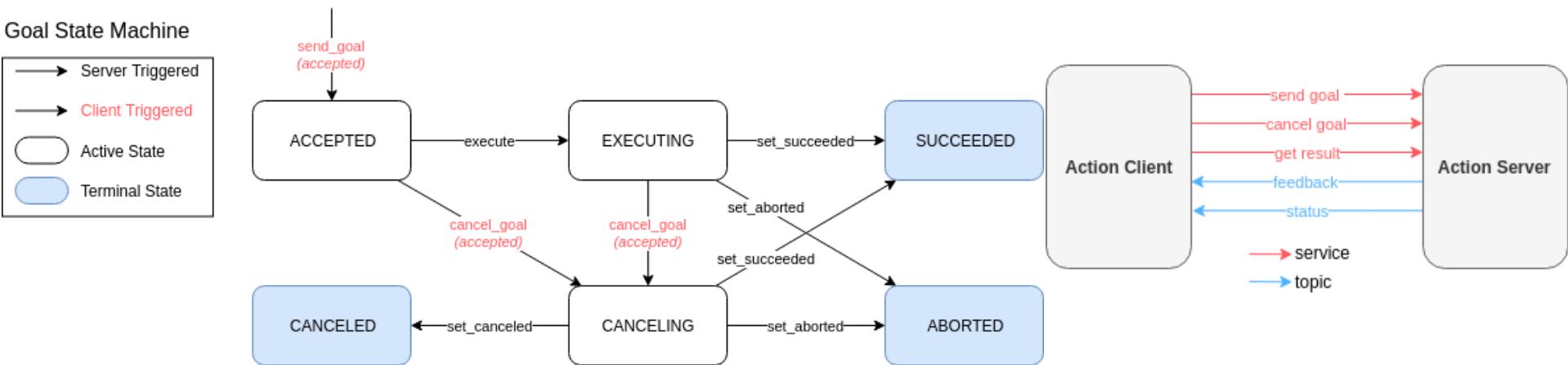
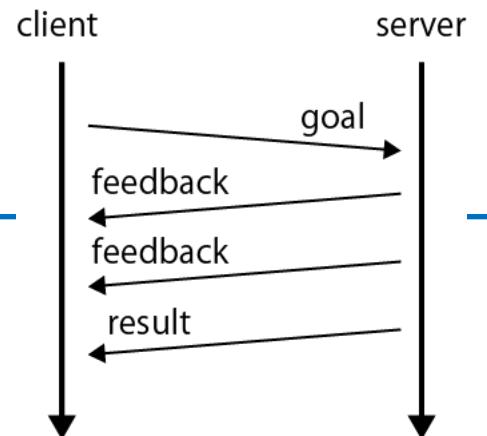


- 基本状態
    - Unconfigured
    - Inactive
    - Active
    - Finalized
  - 遷移状態
    - Configuring
    - Activating
    - Deactivating
    - CleaningUp
    - ShuttingDown
    - ErrorProcessing
- ✓ transitionsの通知で  
状態遷移する

[http://design.ros2.org/  
articles/node\\_lifecycle.html](http://design.ros2.org/articles/node_lifecycle.html)

# Actions

- 非同期送信／同期受信の組合せ
  - Entities: action server / client
  - Interface Definition: Goal / Result / Feedback
- ✓ TopicとServiceの名前空間は分離できる



<http://design.ros2.org/articles/actions.html>

# Intra-Process Comm.

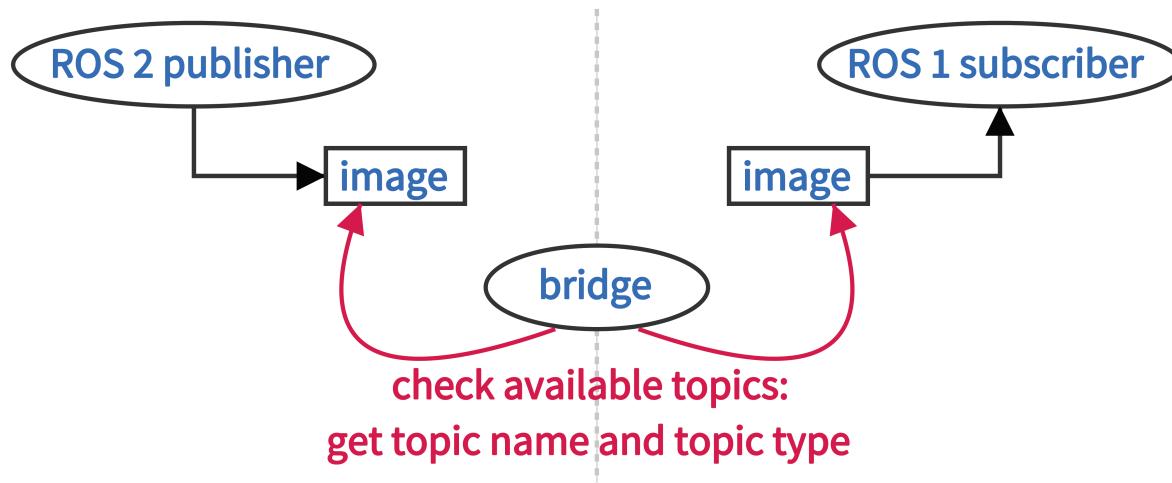
- 複数ノードを1プロセスとして動作させること
  - Pub/Sub通信が高速かつ軽量に
  - std::unique\_ptr によるゼロコピーと所有権移譲
    - ✓ ROS 1のnodelet (boost::shared\_ptr) にあった潜在的バグも解消
  - SingleThreadedExecutor によるノード登録と実行

```
class Producer : public rclcpp::Node {...}
class Consumer : public rclcpp::Node {...}

rclcpp::executors::SingleThreadedExecutor executor;
auto producer = std::make_shared<Producer>("producer");
auto consumer = std::make_shared<Consumer>("consumer");
executor.add_node(producer);
executor.add_node(consumer);
executor.spin();
```

# ros1bridge

- ROS 1とROS 2を共存させる仕組み
- bridgeノードが topic/service の“橋渡し”をする
  - dynamic\_bridge: 双方に同名のトピック／サービスが存在する時のみブリッジを作成
  - static\_bridge: 常にブリッジを作成



# Multi Platform Support

Arch.	Ubuntu Bionic (18.04)	MacOS Mojave (10.14)	Windows 10 (VS20129)	Debian Buster (10)	OpenEmbedded Thud (2.6) / WebOS OSE [s]
amd64	Tier 1 [d][a][s]	Tier 1 [a][s]	Tier 1 [a][s]	Tier 3 [s]	
arm64	Tier 1 [d][a][s]			Tier 3 [s]	
arm32	Tier 2 [a][s]			Tier 3 [s]	Tier 3 [s]

- Tier 1: Open Robotics公式の手厚いサポート
- Tier 2: 公式だが限定的なサポート
  - [d]: Debianパッケージとして配布
  - [a]: 必要パッケージが含まれたアーカイブとして配布
  - [s]: ソースからビルドが必要
- Tier 3: コミュニティお任せのサポート
- 組込み環境の対応も進行中



# colcon build tool

- **collective construction**
- colcon = ament + catkin
  - ament: build for ROS 2 pkgs \$ ament build --isolated
  - catkin: build for ROS 1 pkgs \$ catkin\_make\_isolated --install

TL;DR

```
$ mkdir -p /tmp/workspace/src          # Make a workspace directory with a src subdirectory
$ cd /tmp/workspace                     # Change directory to the workspace root
$ <...>                                # Populate the 'src' directory with packages
$ colcon list -g                         # List all packages in the workspace and their dependencies
$ colcon build                           # Build all packages in the workspace
$ colcon test                            # Test all packages in the workspace
$ colcon test-result --all               # Enumerate all test results
$ . install/local_setup.bash            # Setup the environment to use the built packages
$ <...>                                # Use the built packages
```

# Python launch system

```
def generate_launch_description():
    server = launch_ros.actions.Node(
        package='demo_nodes_cpp',
        node_executable='add_two_ints_server',
        output='screen')
    client = launch_ros.actions.Node(
        package='demo_nodes_cpp',
        node_executable='add_two_ints_client',
        output='screen')
    return launch.LaunchDescription([
        server, client,
        launch.actions.RegisterEventHandler(
            event_handler=launch.event_handlers.OnProcessExit(
                target_action=client,
                on_exit=[launch.actions.EmitEvent(
                    event=launch.events.Shutdown())]),
        )),
    ])
```

# ROS versionとdistribution

distribution	Release	EOL date	Ubuntu
Kinetic Kame	2016/05/23	April, 2021	Xenial (16.04)
<b>Melodic Morenia</b>	2018/05/23	May, 2023	Bionic (18.04)
Noetic Ninjemys	2020/05	May, 2025	Focal (20.04)
Crystal Clemmys	2018/12/14	Dec, 2019	Bionic (18.04)
<b>Dashing Diademate</b>	2019/05/31	May, 2021	Bionic (18.04)
Eloquent Elusor	2019/11/22	Nov, 2020	Bionic (18.04)
Foxy Fitzroy	2020/05	3+ years	Focal (20.04)

<http://wiki.ros.org/Distributions>

<https://index.ros.org/doc/ros2/Releases/>

- 追記：Gazebo 11 (3D sim tool)は2020年で最終リリース
  - 2025年1月までのLTS release
  - 次世代版の Ignition が開発進行中

# 最新安定版のROS 2 distribution

- ROS 2 Dashing Diademata
  - code name: dashing
  - 2019/05/31に正式リリース
  - EOLは2021年5月まで
    - ✓これまでのものは半年だった
- Dashingでの主なUpdate
  - componentスタイルの推奨
  - Actionsの仕様変更
  - rosidl, parameters機能の更新



# 最新のROS 2 distribution

- ROS 2 Eloquent Elusor
  - code name: eloquent
  - 2019/11/22に正式リリース
  - EOLは2020年11月まで（1年）
    - ✓ Crystalまでは半年だった
- Eloquentでの主なUpdate
  - launchのXML/YAML記述対応
  - ros2doctor コマンドの追加（環境情報や問題の確認）
  - create\_publisher/\_subscription の仕様変更(API統一)
    - ✓ QoSの指定が必須化
  - RMW実装の改善（zero-copy, CycloneDDS推奨）

