

# ROSBAG 2 UPDATES

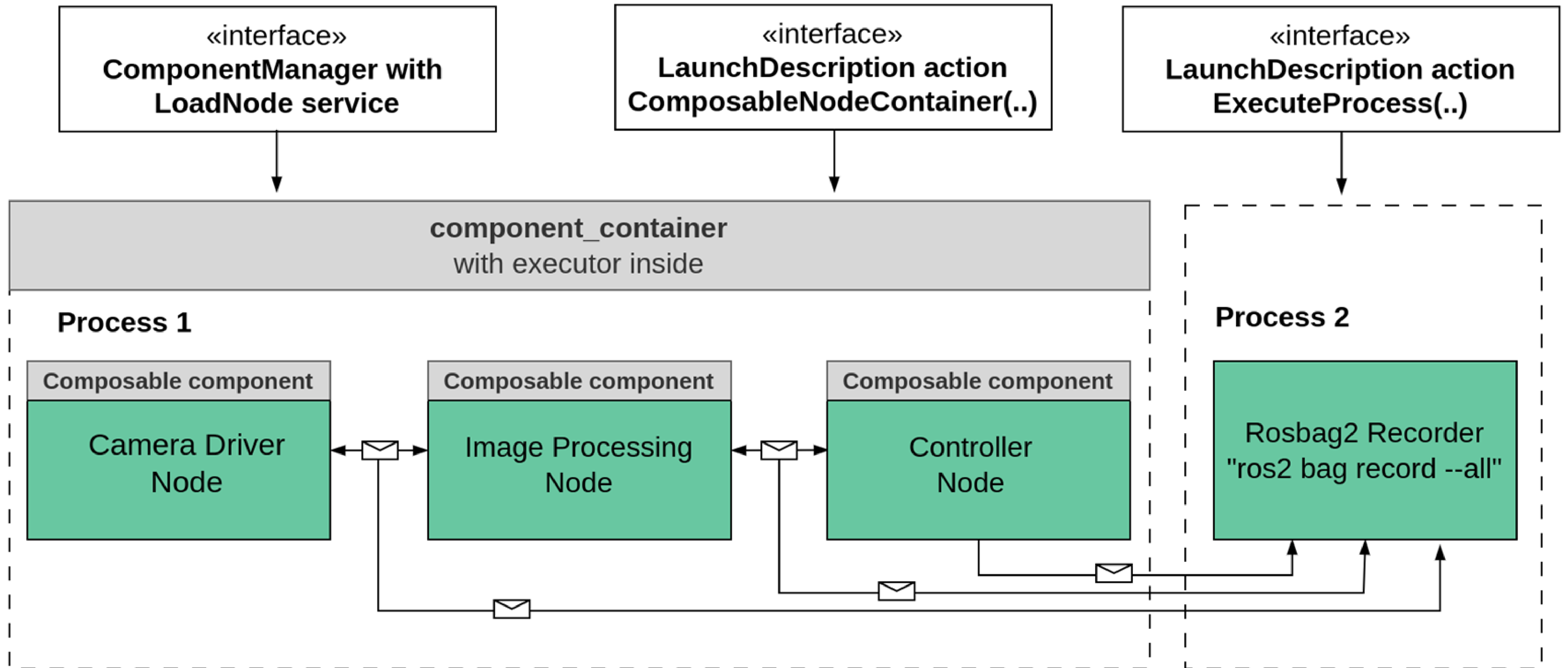
New features and improvements in Rosbag2 Jazzy Jalisco release

**Michael Orlov (Apex.AI)**  
**Tomoya Fujita (Sony)**

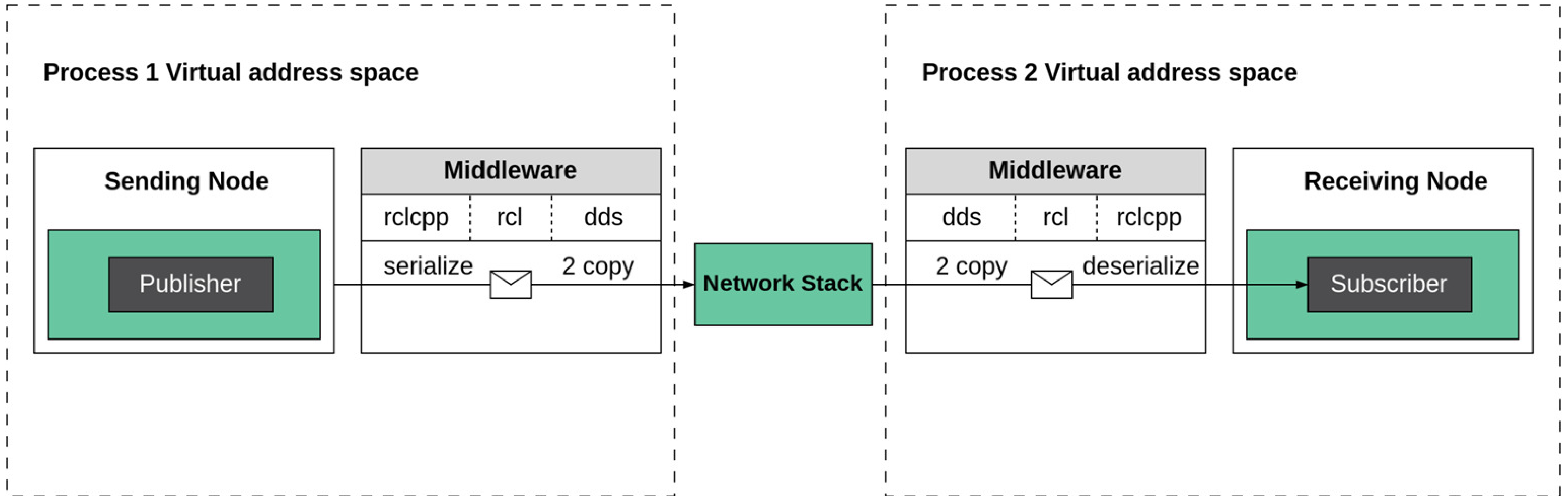
# Notable new features in Rosbag2

- 1. Composable Player and Recorder nodes.**
- 2. Recording the “message sent” timestamp.**
- 3. Enhanced message filtration options in recorder.**
- 4. Ability to store serialized metadata in bag files.**
- 5. Use enum values for “offered\_qos\_profiles”.**
- 6. Ability to split already existing bags by time.**
- 7. Services recording and replay.**

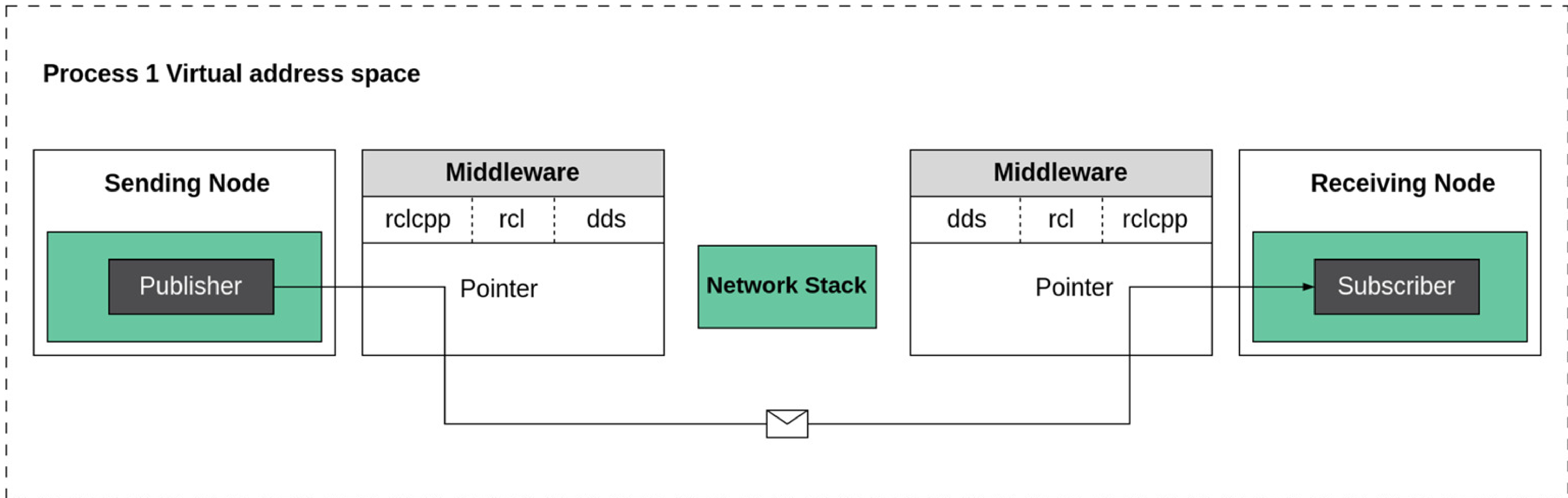
# 1. Composable Player and Recorder nodes



# How the inter-process communication typically works



# How the intra-process communication works





## 2. Saving sent and received timestamps for each message during recording

- Rosbag2 now uses the sent and received timestamps from "message\_info" during recording.
- These timestamps are more indicative of when the data was actually sent and received, respectively.
- Saving the sent timestamps is currently only supported for MCAP files (the default storage format).

# 3. Enhanced message filtration options in recorder

## Inclusive options:

**--all-topics, --all-services, --all** - Will record all topics (excluding hidden topics), all services or all topics and services respectively.

**--topics** - Space-delimited list of topics to record.

**--services** - Space-delimited list of services to record.

**--topic-types** - Space-delimited list of topic types to record.

**--regex** - Record only topics and services containing provided regular expression. Note: **--all, --all-topics** or **--all-services** will override **--regex**.

# 3. Enhanced message filtration options in recorder

## Excluding options:

**--exclude-topics** - Space-delimited list of topics to exclude from recording. Works on top of --all, --all-topics, --topics or --regex.

**--exclude-services** - Space-delimited list of services not being recorded. Works on top of --all, --all-services, --services or --regex.

**--exclude-regex** - Exclude topics and services matching the provided regular expression. Works on top of --all, --all-topics, --all-services, --topics, --services or --regex.

**--exclude-topic-types** - Space-delimited list of topic types to exclude from recording. Works on top of --all, --all-topics, --topics or --regex.



## 4. Store serialized metadata in bag files directly

Metadata is now also stored in each bag file, once when opening the file and a second time when closing the written bag file.

This allows bag files to be self-contained, and used without the “metadata.yaml” file in the Rosbag2 player or third-party applications.

However, the “metadata.yaml” file is still written and “ros2 bag reindex” can be used to restore the “metadata.yaml” file, if desired.

In addition we also now store ROS\_DISTRO name in the metadata.

# 5. Human-readable string values for QoS settings in metadata, and in the overriding QoS profile YAML files

## Old version

```
/topic_name:
  history: 1
  depth: 10
  reliability: 1
  durability: 2
  deadline:
    # unspecified/infinity
    sec: 0
    nsec: 0
  lifespan:
    # unspecified/infinity
    sec: 0
    nsec: 0
  liveliness: 0
  liveliness_lease_duration:
    # unspecified/infinity
    sec: 0
    nsec: 0
  avoid_ros_namespace_conventions: false
```

## New version

```
/topic_name:
  history: keep_last
  depth: 10
  reliability: reliable
  durability: volatile
  deadline:
    # unspecified/infinity
    sec: 0
    nsec: 0
  lifespan:
    # unspecified/infinity
    sec: 0
    nsec: 0
  liveliness: system_default
  liveliness_lease_duration:
    # unspecified/infinity
    sec: 0
    nsec: 0
  avoid_ros_namespace_conventions: false
```

## 6. Added ability to split existing bags by time

Added “**start\_time\_ns**” and “**end\_time\_ns**” to the “**StorageOptions**” to exclude messages not in [start\_time; end\_time] during the “**ros2 bag convert**” operation.

Example of usage when need to merge multiple bag files:

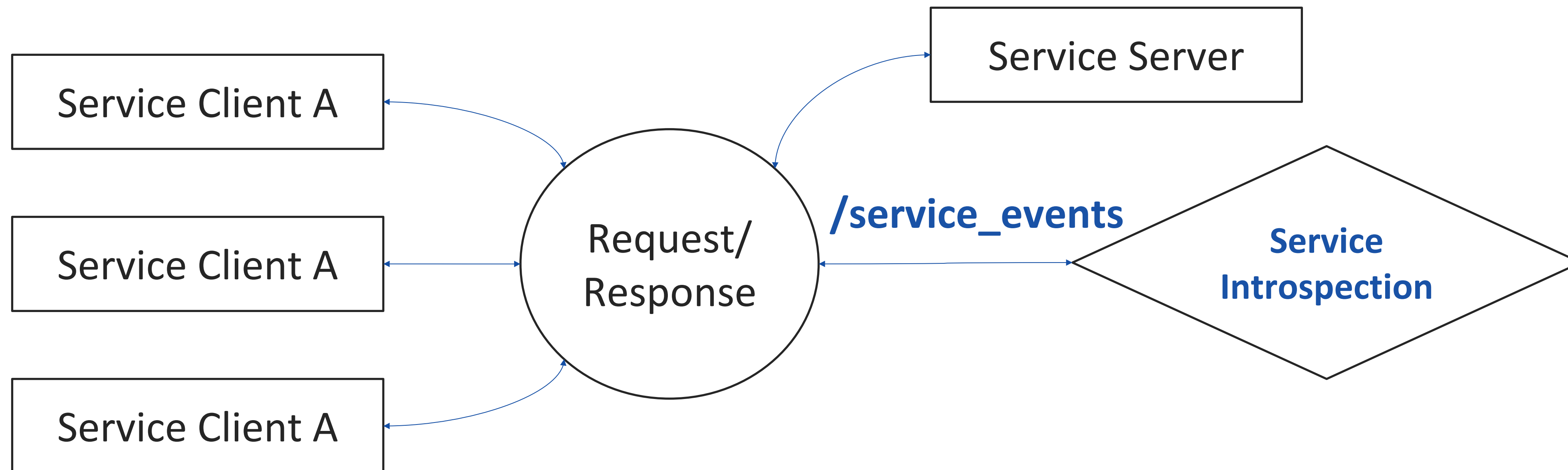
```
$ ros2 bag convert -i bag1 -i bag2 -o out.yaml
```

Where out.yaml file could be as:

```
output_bags:  
- uri: merged_bag  
  all: true  
  start_time_ns: 9223372036854775807  
  end_time_ns: 9223372046800000000
```

# Service Introspection

- Allows users to introspect the metadata and contents associated with the service, such as requests and responses. (Available since Iron Irwini)
- Introspection must be configured explicitly for the service.
- All information is published on a hidden topic generated from the name of the service. (e.g **/myservice service** creates **/myservice/\_service\_event**)

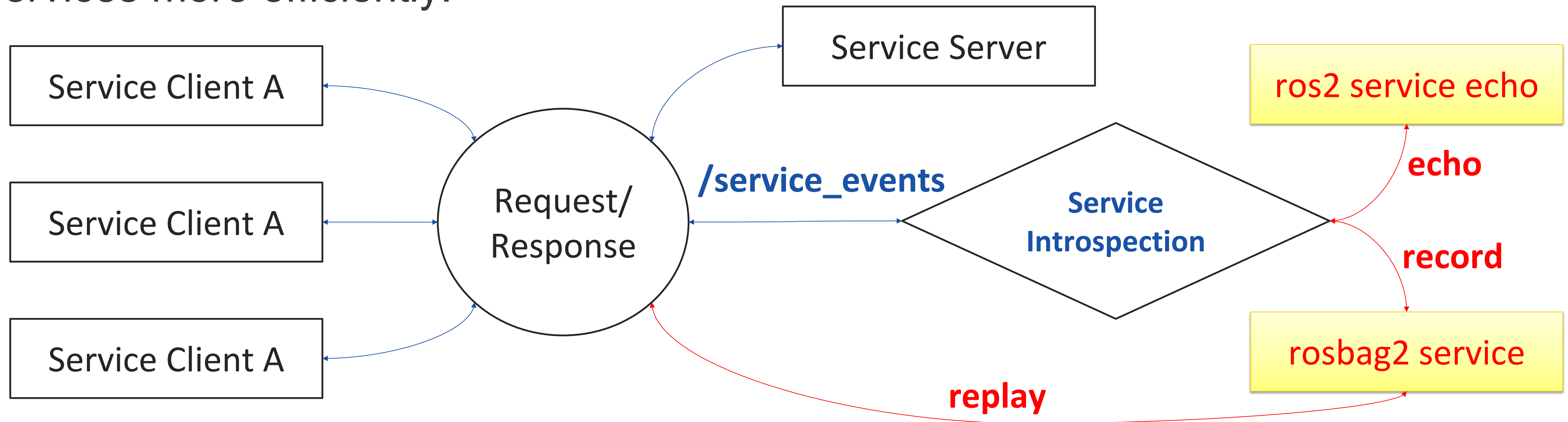


# 7. Services recording and replay in Rosbag2

- This feature depends on Service introspection, available from Jazzy Jalisco.
- Rosbag2 takes advantage of it to record all service requests and responses, also replays service data from the bag file.
- This feature allows you to debug/enhance/test/simulate the ROS 2 services more efficiently.



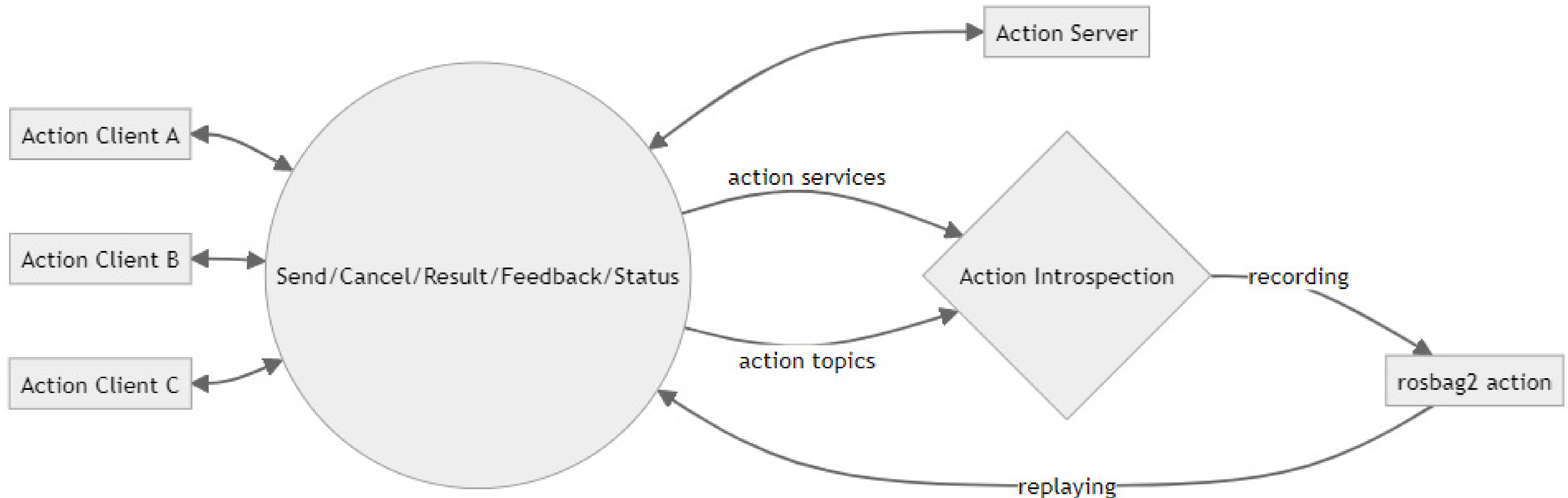
Tutorial





# Action Introspection / Rosbag2 support

- [\[Feature Request\] Action Introspection and Record/Play Support](#)
  - <https://github.com/ros2/ros2/issues/1543>
- [\[ROS Enhancement Proposal\] Action introspection rosbag2 support](#)
  - <https://github.com/ros-infrastructure/rep/pull/405>



# What's next?

- Recording and replay for actions.
- Playback from multiple files.
- Statistics about recording (at run-time and in the metadata).
- Overhaul the snapshot feature.
  - Writing each snapshot to a separate file.
  - Saving snapshots for N seconds before and after event.
- Store channels in topics on per-publisher bases.
- Focus on quality and usability.
- More improvements and bugfixes.

# Thank you!

---

**Special thanks to all the Rosbag2 contributors out there!**

**Join us!**

**Join Rosbag2 and tooling working group meetings, every other Friday at 9:00 AM PST.**

**Calendar:** See the OSRF community-organized events calendar

**Code and documentation:** <https://github.com/ros2/rosbag2>

