eProsima
The Middleware Experts

Enhancing Robotic Communication & Scalability with Topic Keys in ROS 2

Speaker:

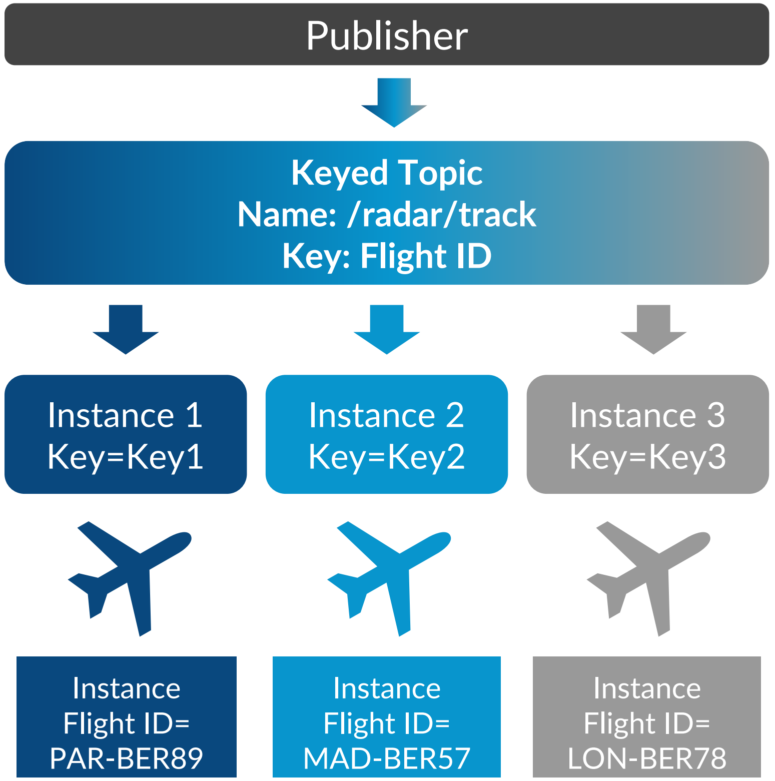
- Raúl Sánchez-Mateos, Project Manager @ eProsima
  raul@eprosima.com

October 23rd, 2024

Topic Keys - Overview

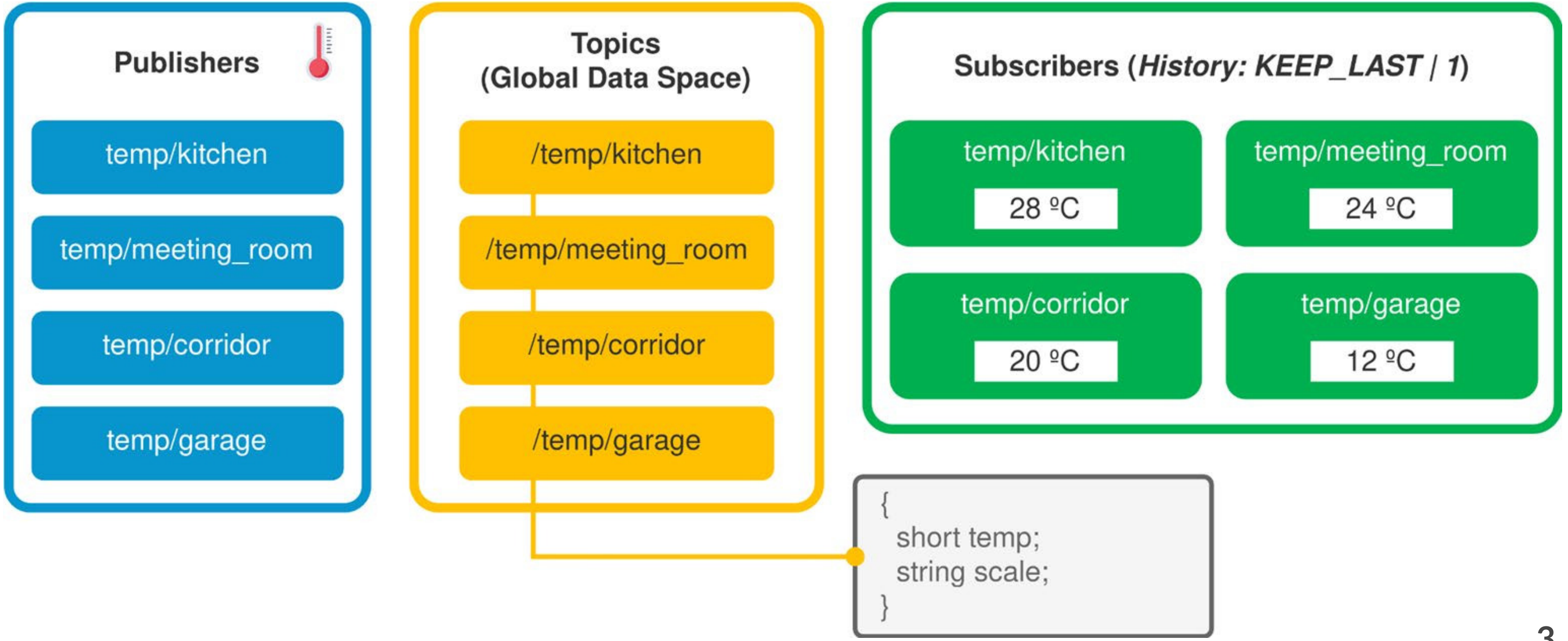
Overview of DDS Keyed Topics feature

- Similar objects with a common data type
- Objects identified using Keys
- One "Topic Instance" per Key
  Instances differentiates between different objects of the set
- QoS applied by instance
  Each instance keeps its own History, Deadline, Lifespan, etc.



Topic Keys - Overview

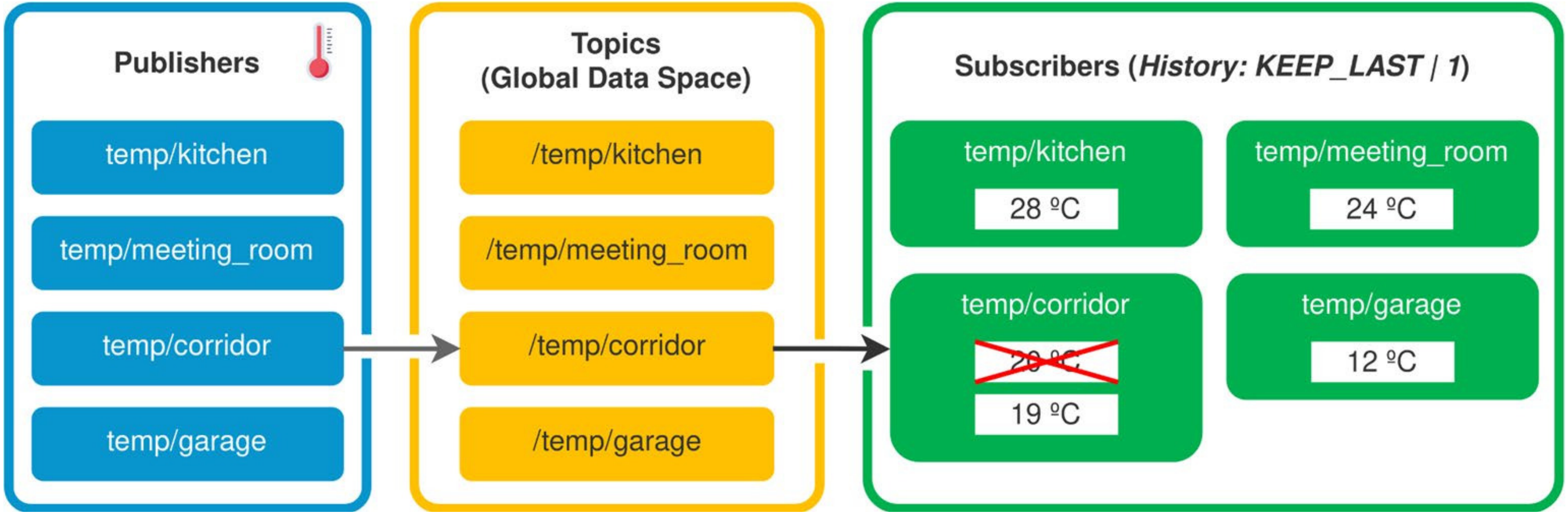
Topic Keys Example → Preserving QoS by topic instance



```
{
  short temp;
  string scale;
}
```

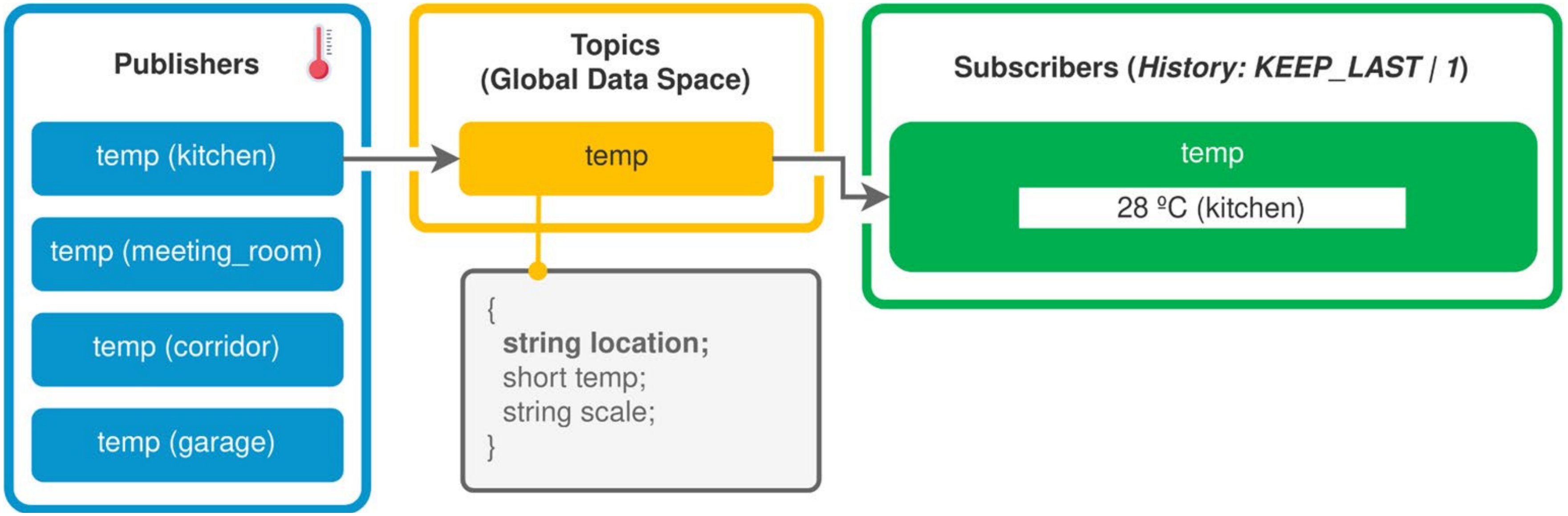
Topic Keys - Overview

Topic Keys Example → Preserving QoS by topic instance



Topic Keys - Overview

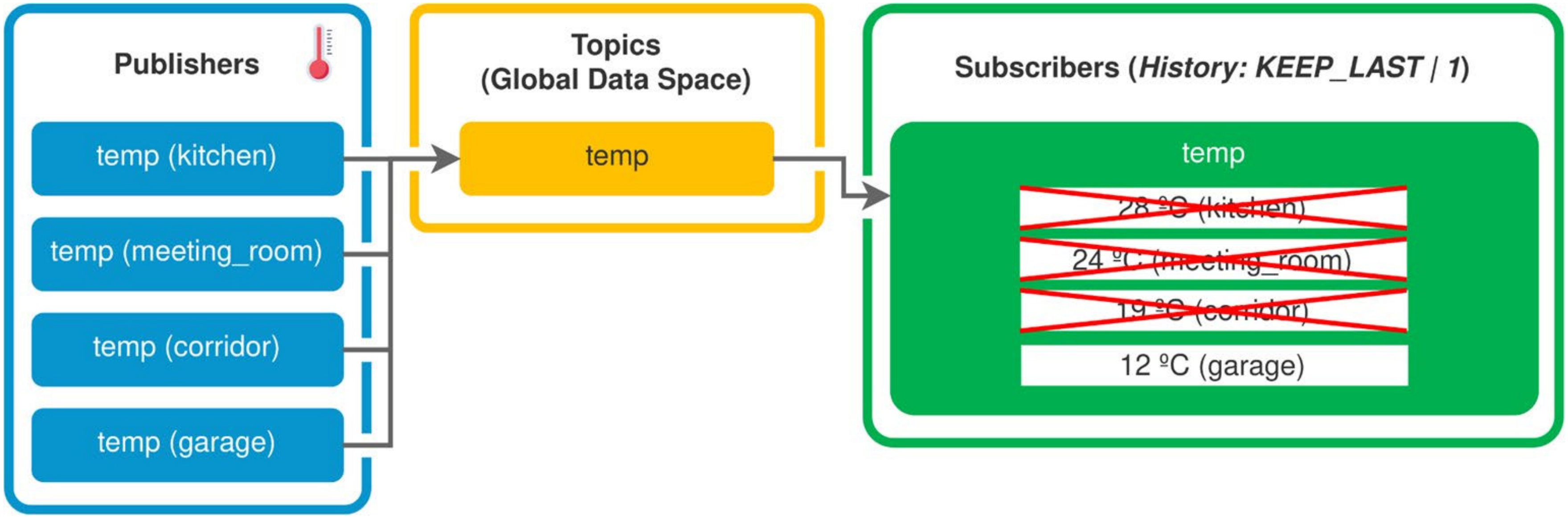
Topic Keys Example → Preserving QoS by topic instance



```
{
  string location;
  short temp;
  string scale;
}
```

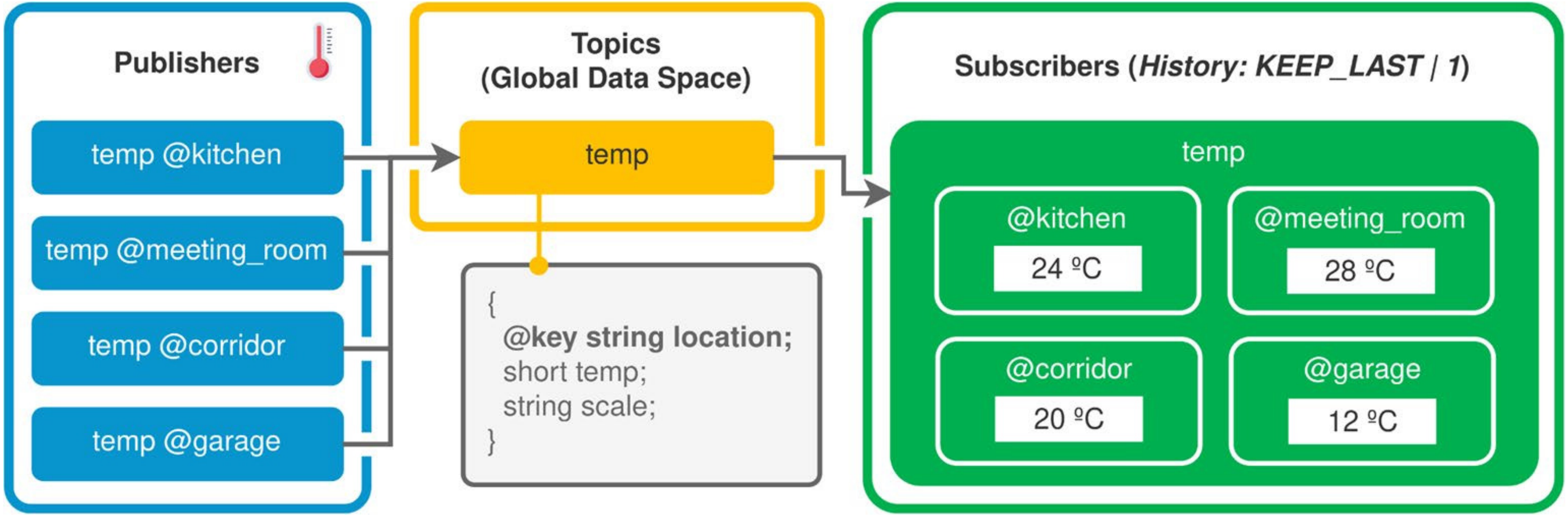
Topic Keys - Overview

Topic Keys Example → Preserving QoS by topic instance



Topic Keys - Overview

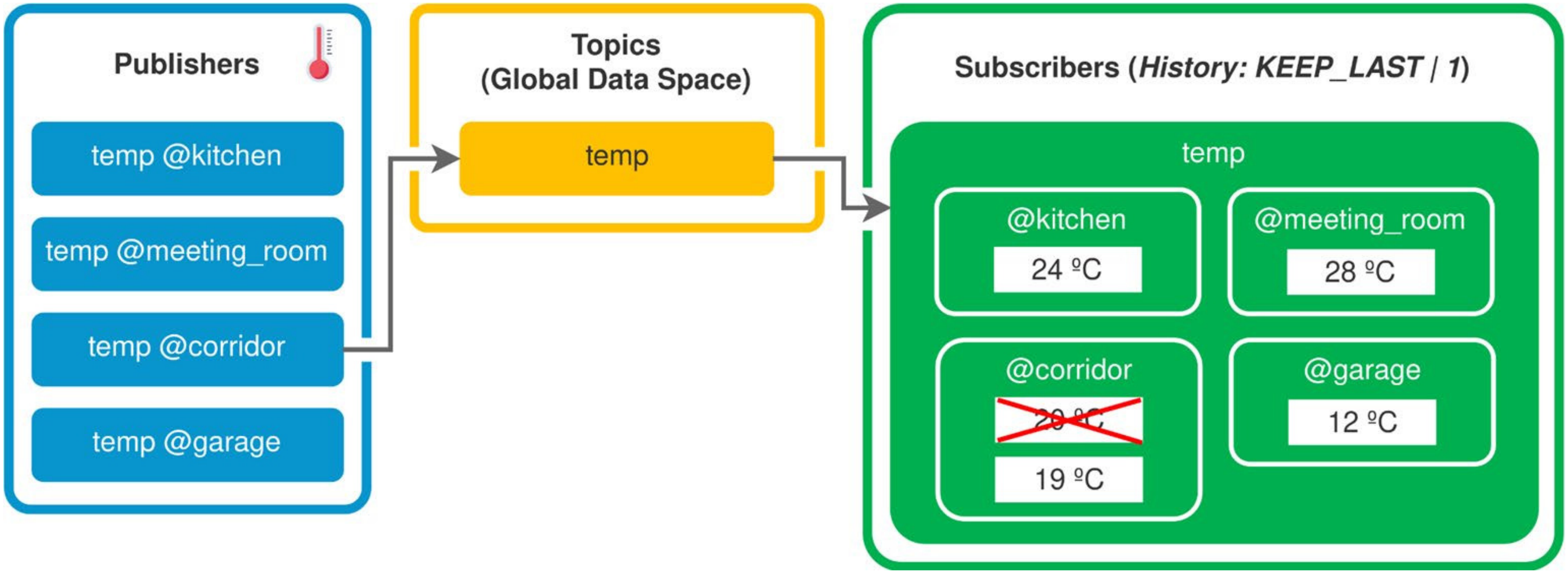
Topic Keys Example → Preserving QoS by topic instance



```
{
  @key string location;
  short temp;
  string scale;
}
```

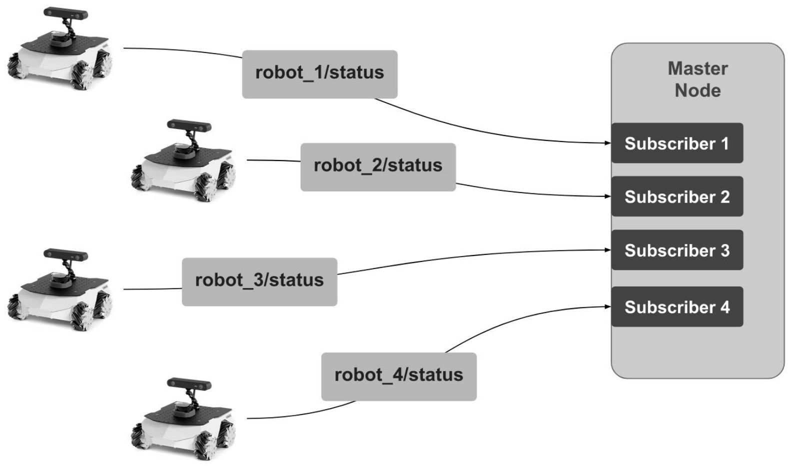
Topic Keys - Overview

Topic Keys Example → Preserving QoS by topic instance



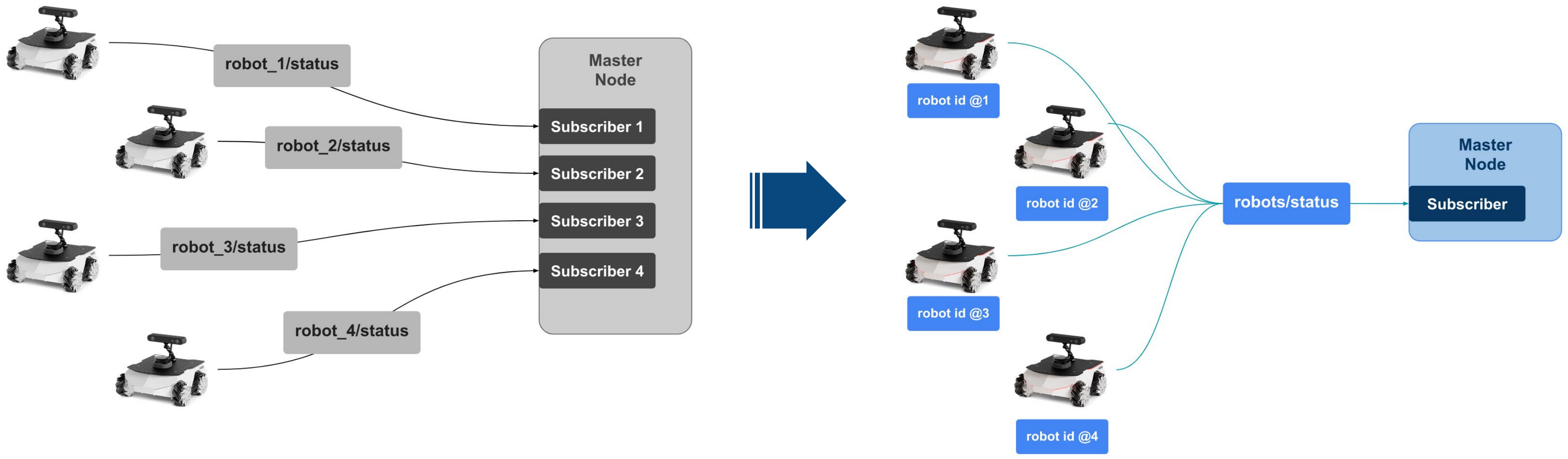
Topic Keys - Overview

Application of topic keys to ROS 2 systems



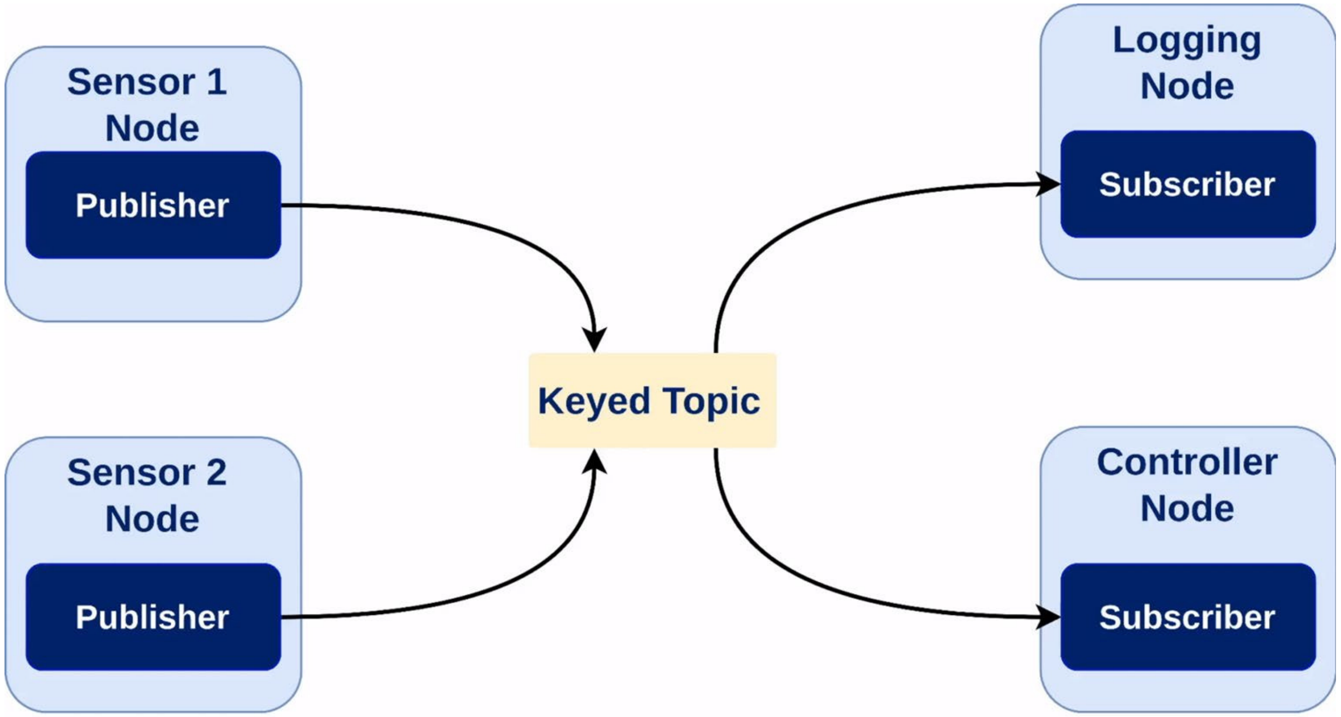
Topic Keys - Overview

Application of topic keys to ROS 2 systems



Topic Keys - Content Filtered Topic

Application of topic keys combined with Content Filter Topic to ROS 2 systems



Topic Keys - Overview

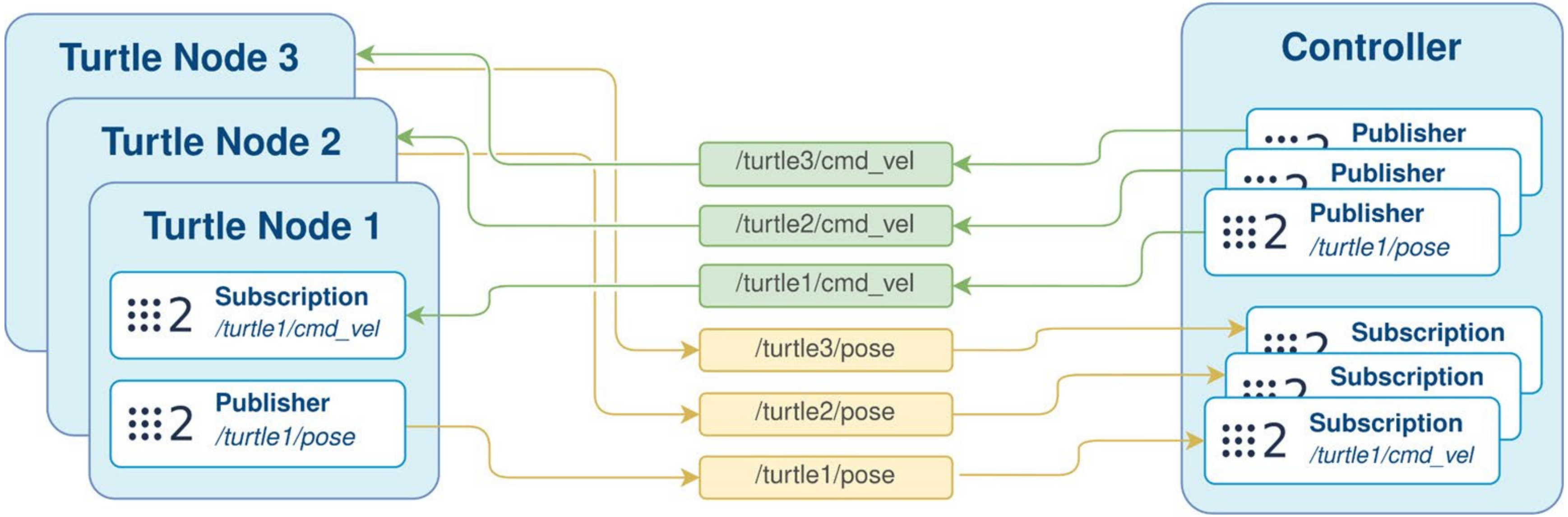
Overview of DDS Keyed Topics feature

KeyedTwist.idl

```
module docs_turtlesim {
  module msg {
    struct KeyedTwist {
       @key long turtle_id;
       docs_turtlesim::msg::Vector3 linear;
            docs_turtlesim::msg::Vector3 angular;
    };
  };
};
```

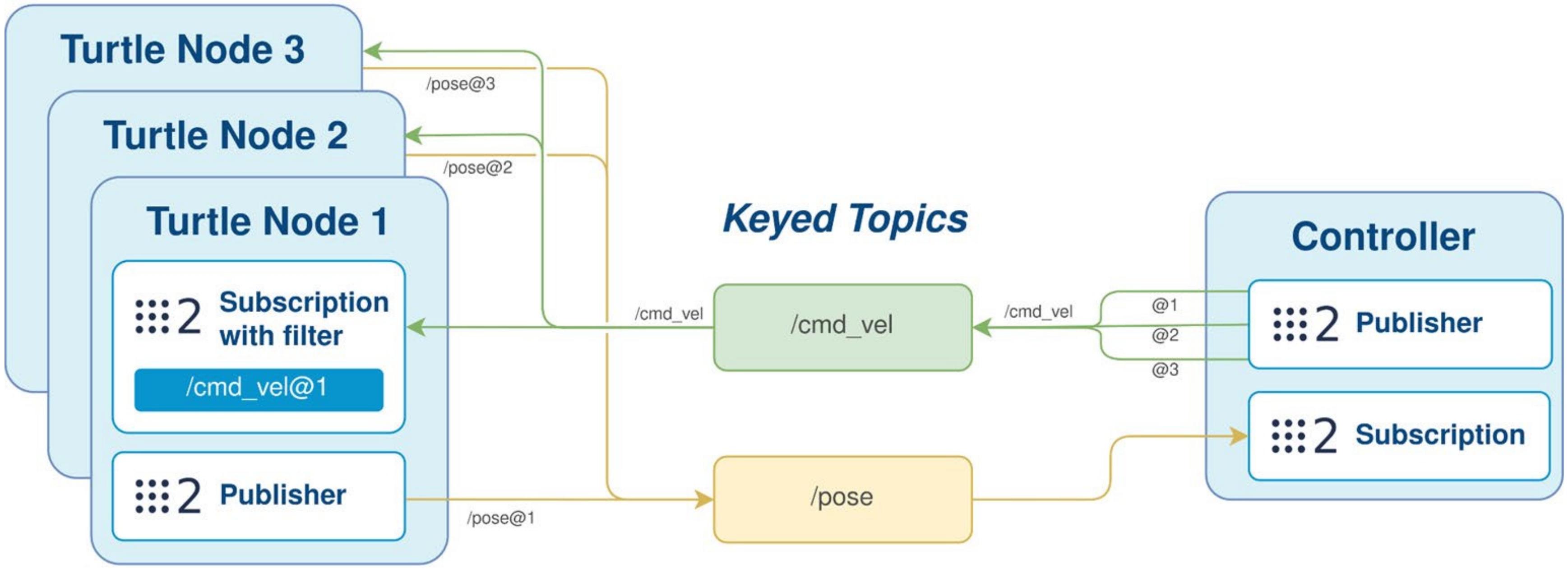
Topic Keys - Use Case

Turtlesim with keys demo



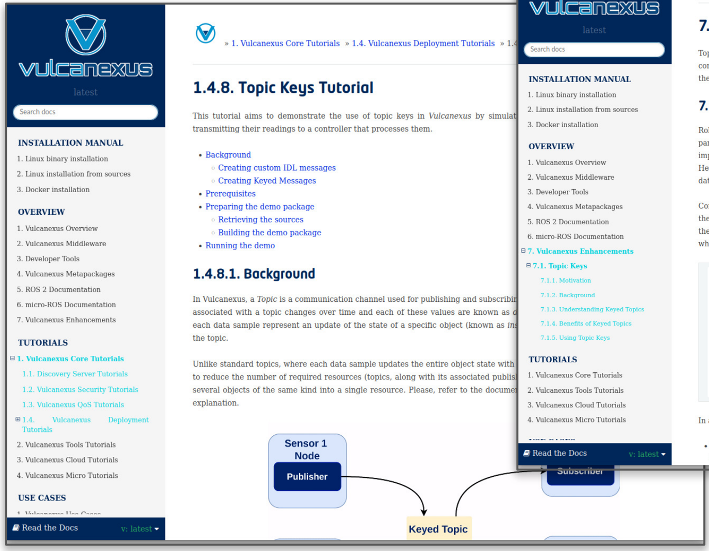
Topic Keys - Use Case

Turtlesim with keys demo



Topic Keys - Documentation

Documentation on how to use and setup the topic keys in Vulcanexus



vulcanexus

latest

Search docs

INSTALLATION MANUAL

1. Linux binary installation
2. Linux installation from sources
3. Docker installation

OVERVIEW

1. Vulcanexus Overview
2. Vulcanexus Middleware
3. Developer Tools
4. Vulcanexus Metapackages
5. ROS 2 Documentation
6. micro-ROS Documentation
7. Vulcanexus Enhancements

TUTORIALS

1. Vulcanexus Core Tutorials
   - 1.1. Discovery Server Tutorials
   - 1.2. Vulcanexus Security Tutorials
   - 1.3. Vulcanexus QoS Tutorials
   - 1.4. Vulcanexus Deployment Tutorials
2. Vulcanexus Tools Tutorials
3. Vulcanexus Cloud Tutorials
4. Vulcanexus Micro Tutorials

USE CASES

Read the Docs v: latest

» 1. Vulcanexus Core Tutorials » 1.4. Vulcanexus Deployment Tutorials » 1.4

1.4.8. Topic Keys Tutorial

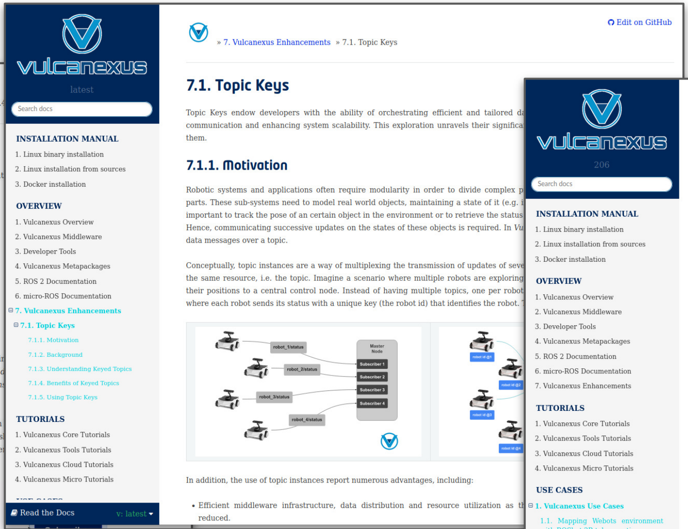
This tutorial aims to demonstrate the use of topic keys in *Vulcanexus* by simulat
transmitting their readings to a controller that processes them.

- Background
  - Creating custom IDL messages
  - Creating Keyed Messages
- Prerequisites
- Preparing the demo package
  - Retrieving the sources
  - Building the demo package
- Running the demo

1.4.8.1. Background

In Vulcanexus, a *Topic* is a communication channel used for publishing and subscribin
associated with a topic changes over time and each of these values are known as *d*
each data sample represent an update of the state of a specific object (known as *ins*
the topic.

Unlike standard topics, where each data sample updates the entire object state with
to reduce the number of required resources (topics, along with its associated publis
several objects of the same kind into a single resource. Please, refer to the docume
explanation.



vulcanexus

latest

Search docs

INSTALLATION MANUAL

1. Linux binary installation
2. Linux installation from sources
3. Docker installation

OVERVIEW

1. Vulcanexus Overview
2. Vulcanexus Middleware
3. Developer Tools
4. Vulcanexus Metapackages
5. ROS 2 Documentation
6. micro-ROS Documentation
7. Vulcanexus Enhancements
   - 7.1. Topic Keys
     - 7.1.1. Motivation
     - 7.1.2. Background
     - 7.1.3. Understanding Keyed Topics
     - 7.1.4. Benefits of Keyed Topics
     - 7.1.5. Using Topic Keys

TUTORIALS

1. Vulcanexus Core Tutorials
2. Vulcanexus Tools Tutorials
3. Vulcanexus Cloud Tutorials
4. Vulcanexus Micro Tutorials

Read the Docs v: latest

Edit on GitHub

» 7. Vulcanexus Enhancements » 7.1. Topic Keys

7.1. Topic Keys

Topic Keys endow developers with the ability of orchestrating efficient and tailored da
communication and enhancing system scalability. This exploration unravels their significa
them.

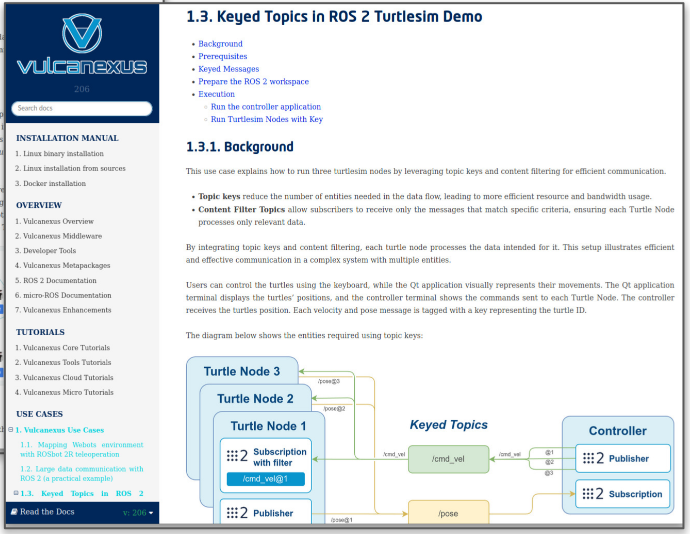
7.1.1. Motivation

Robotic systems and applications often require modularity in order to divide complex p
parts. These sub-systems need to model real world objects, maintaining a state of it (e.g. i
important to track the pose of an certain object in the environment or to retrieve the status
Hence, communicating successive updates on the states of these objects is required. In *Vu*
data messages over a topic.

Conceptually, topic instances are a way of multiplexing the transmission of updates of seve
the same resource, i.e. the topic. Imagine a scenario where multiple robots are exploring
their positions to a central control node. Instead of having multiple topics, one per robot
where each robot sends its status with a unique key (the robot id) that identifies the robot.

In addition, the use of topic instances report numerous advantages, including:

- Efficient middleware infrastructure, data distribution and resource utilization as th
  reduced.



vulcanexus

206

Search docs

INSTALLATION MANUAL

1. Linux binary installation
2. Linux installation from sources
3. Docker installation

OVERVIEW

1. Vulcanexus Overview
2. Vulcanexus Middleware
3. Developer Tools
4. Vulcanexus Metapackages
5. ROS 2 Documentation
6. micro-ROS Documentation
7. Vulcanexus Enhancements

TUTORIALS

1. Vulcanexus Core Tutorials
2. Vulcanexus Tools Tutorials
3. Vulcanexus Cloud Tutorials
4. Vulcanexus Micro Tutorials

USE CASES

1. Vulcanexus Use Cases
   - 1.1. Mapping Webots environment with ROSbot 2R teleoperation
   - 1.2. Large data communication with ROS 2 (a practical example)
   - 1.3. Keyed Topics in ROS 2

Read the Docs v: 206

1.3. Keyed Topics in ROS 2 Turtlesim Demo

- Background
- Prerequisites
- Keyed Messages
- Prepare the ROS 2 workspace
- Execution
  - Run the controller application
  - Run Turtlesim Nodes with Key

1.3.1. Background

This use case explains how to run three turtlesim nodes by leveraging topic keys and content filtering for efficient communication.

- **Topic keys** reduce the number of entities needed in the data flow, leading to more efficient resource and bandwidth usage.
- **Content Filter Topics** allow subscribers to receive only the messages that match specific criteria, ensuring each Turtle Node processes only relevant data.

By integrating topic keys and content filtering, each turtle node processes the data intended for it. This setup illustrates efficient and effective communication in a complex system with multiple entities.

Users can control the turtles using the keyboard, while the Qt application visually represents their movements. The Qt application terminal displays the turtles' positions, and the controller terminal shows the commands sent to each Turtle Node. The controller receives the turtles position. Each velocity and pose message is tagged with a key representing the turtle ID.

The diagram below shows the entities required using topic keys:

EPROSIMA

The Middleware Experts

www.eProsima.com

Linkedin.com/company/eProsima

Twitter.com/EProsima