ROS / ROS 2 with Kubernetes

Tomoya Fujita

Sony R&D US Laboratory

by *Tomoya.Fujita@sony.com* / *fujitatomoya@github*

Slides are available

by *Tomoya.Fujita@sony.com* / *fujitatomoya@github*
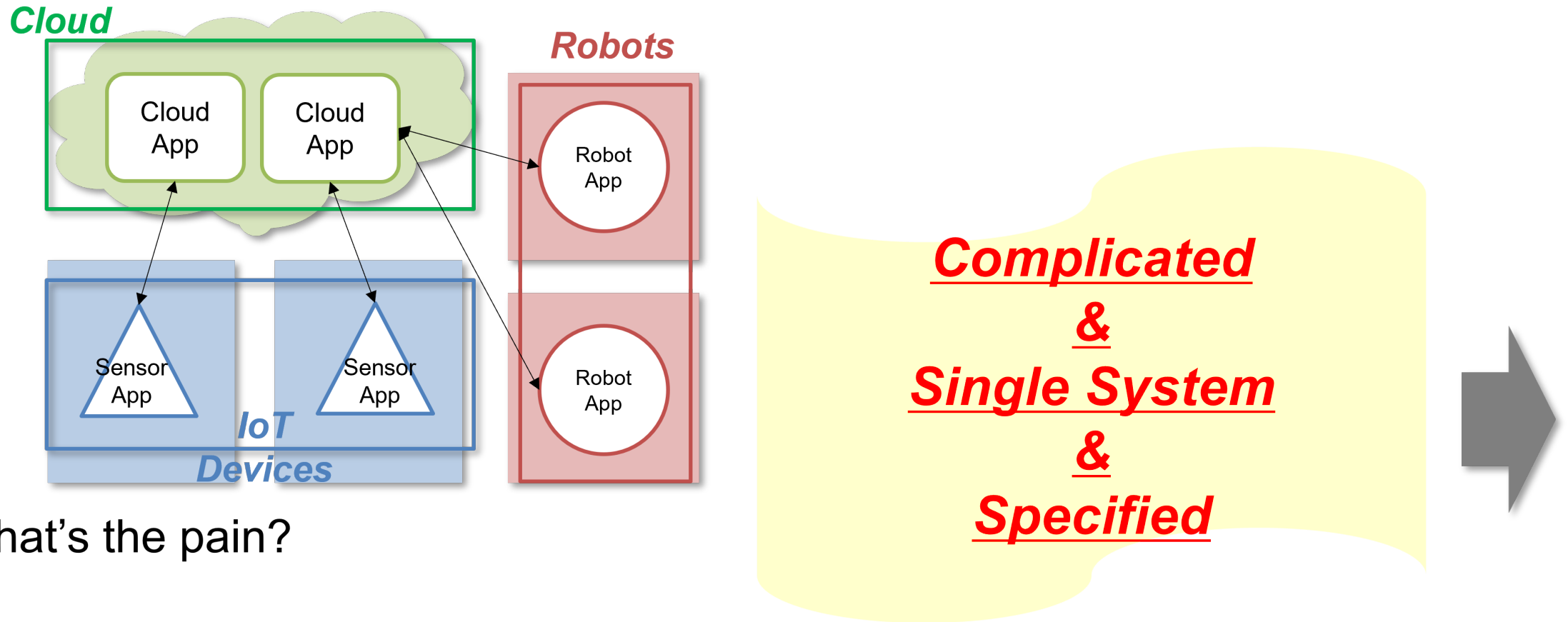
References

- ROSCon 2023 ROS with Kubernetes Talk / Slides
- Community Meeting
  - ROS Deployment Meeting
  - *Cian Donovan* : *containerised userspace "drivers" using udev/systemd/podman*
  - Stay tuned for next meeting?

Background / Problems

- Broad use cases (factory, logistics, entertainment, rescue, car, drone...)

- ROS and ROS 2 Distributed and Connected System.

- Collaborative and Orchestrated Application.

- AI/ML Ops Application lifecycle

- Hardware Acceleration / Platform Dependencies.

Cloud

Robots

Cloud App

Cloud App

Robot App

Sensor App

Sensor App

IoT Devices

Robot App

Complicated & Single System & Specified

What's the pain?

• Different architecture for cloud and edge device.
• Take time and effort to setup environment and run application.
• IoT Device is really statistic implementation.

Cloud

Robots

IoT Devices

Simple/Common
&
Distributed System
&
Platform Agnostic
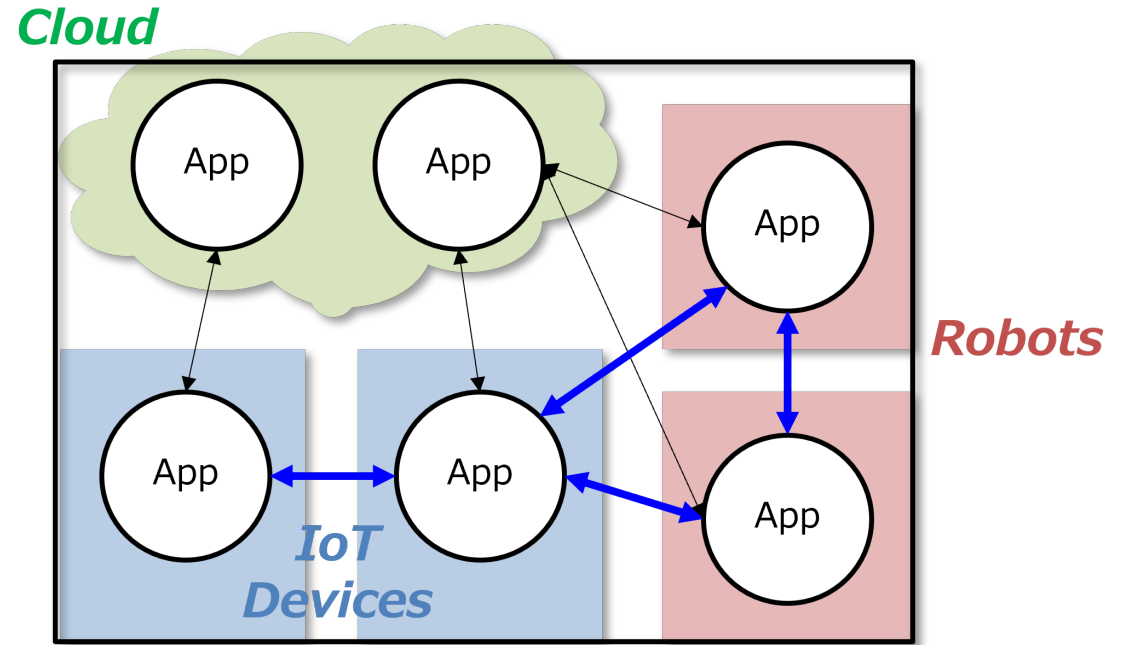
What we want is…

- Common base architecture for everyone and everywhere.
- Application can be deployed on anywhere.
- Eco-system for Application.

Goal / Requirements

- Flexible Application Deployment.

- Network, Device and Storage Abstraction.

- Application Modularity / Platform Agnostic.

- Sustainability / Efficiency

- Observability / Network Policy
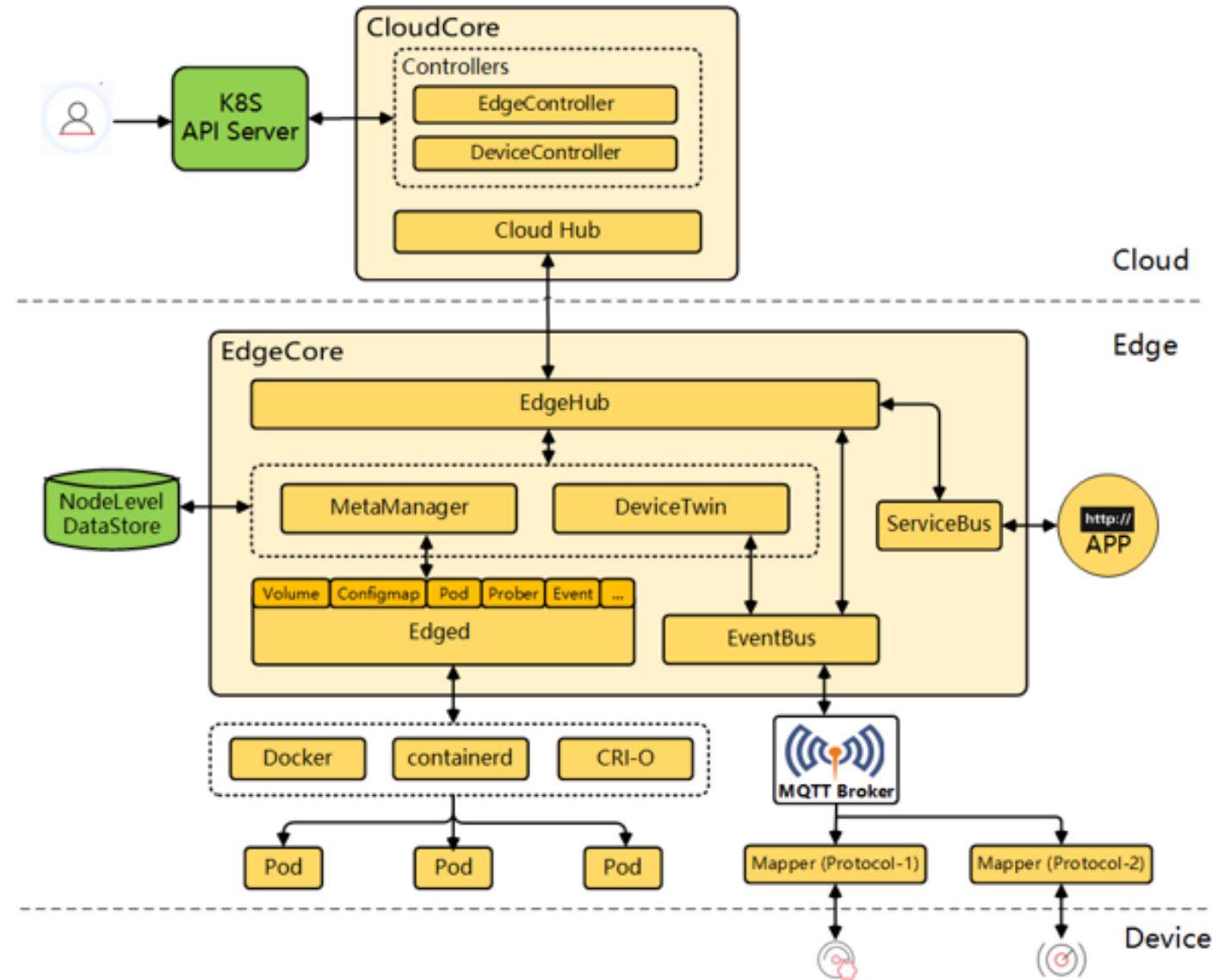
Kubernetes

- Application Deployment and Orchestration.

- Device Capability and Label Control.

- Auto-Scaling and Healing.

- Roll Up/Down, Canary Test.

- Role Based Access Control.

- Device-Plugin / Container Device Interfaces.

- Container Network Interfaces.

- Observability / Network Policy.

KubeEdge

- Cloud-Edge Coordination
- Edge Autonomy
- Kubernetes-native Support
- Resource Efficient

Device-Abstraction (Device-Plugin, Container Device Interface)

- Kubernetes Custom Resource Definition, that allows us to plugin vendor specific hardware and device to the containers.
- After advertising the custom resource to Kubernetes, Kubernetes controls those resources with workload based on application requirements.
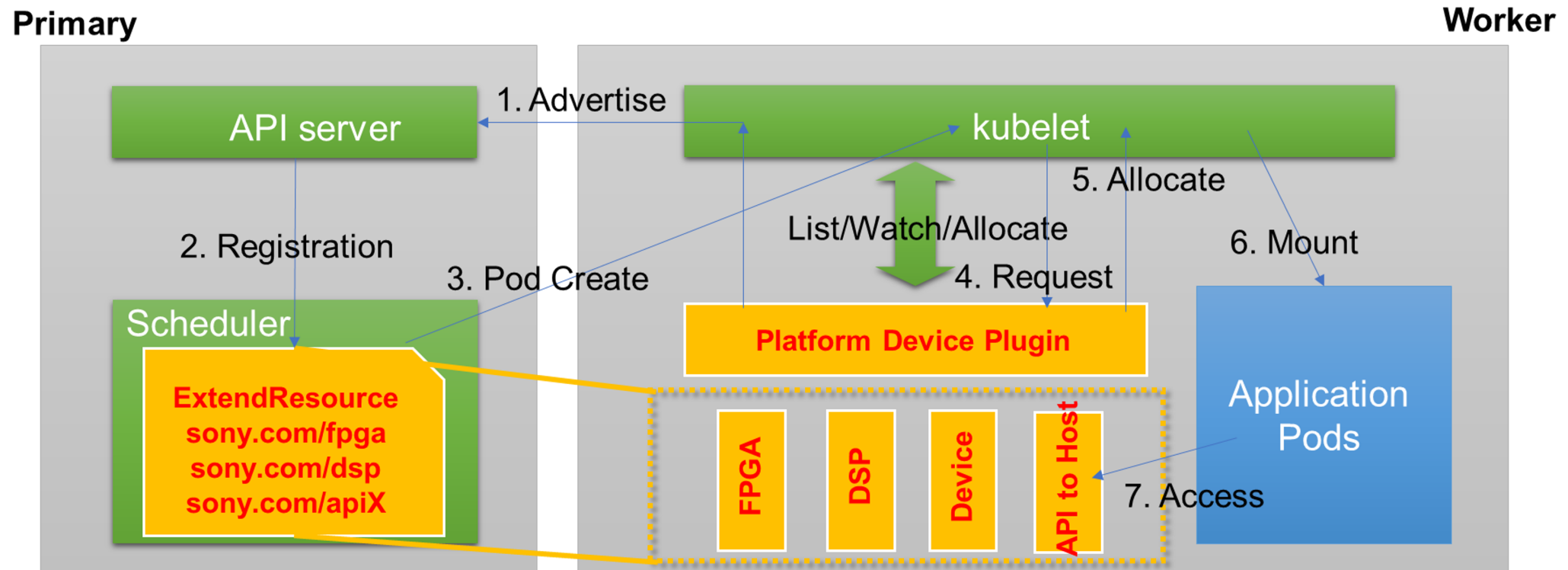
What's missing?

- Missing CNI supports multicast (Cilium)

- KubeEdge CNI support (Cilium)

- Edge Optimization / Configuration

- KubeEdge Cloud-Native Digital Twin RobotDevOps
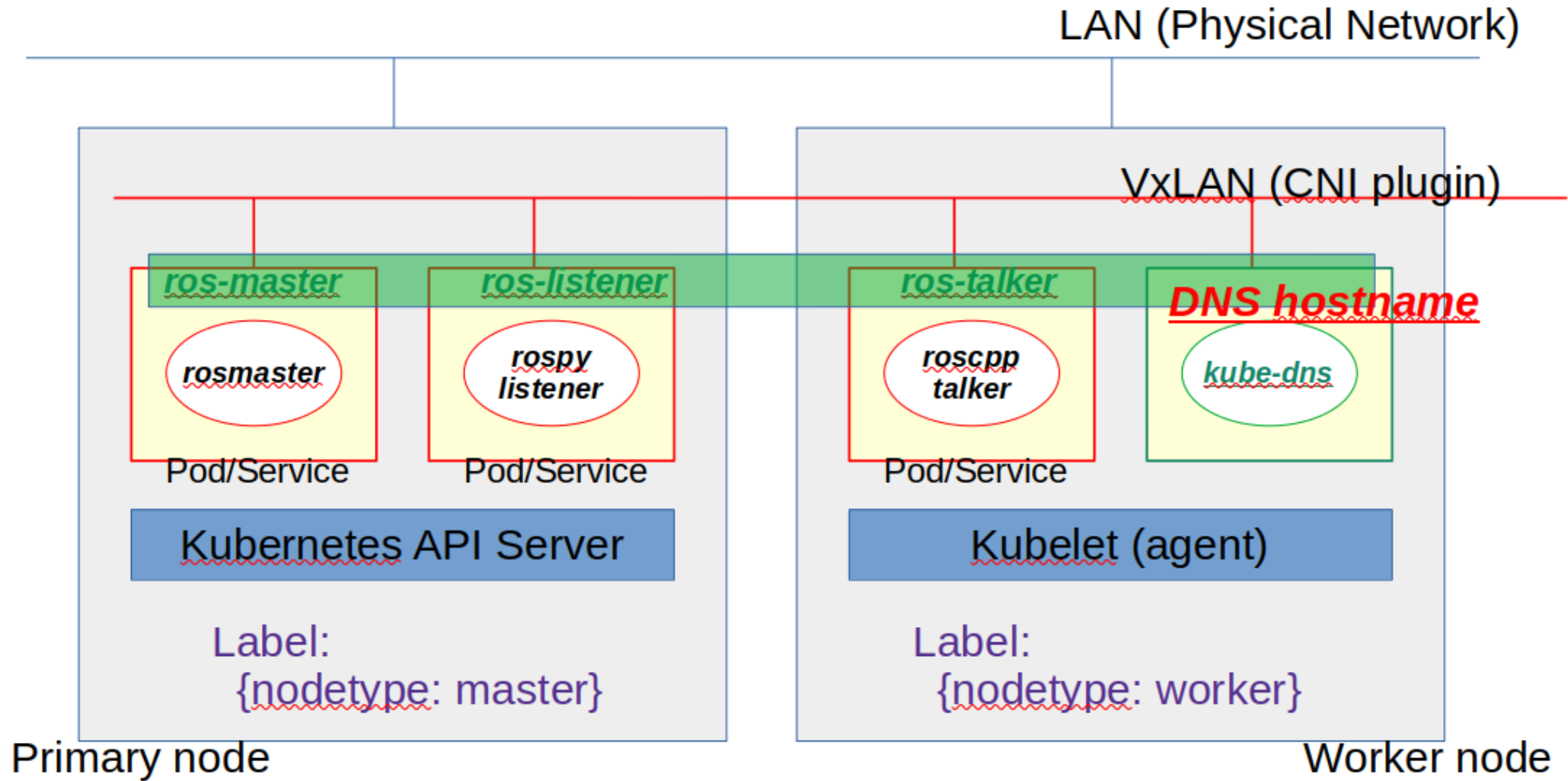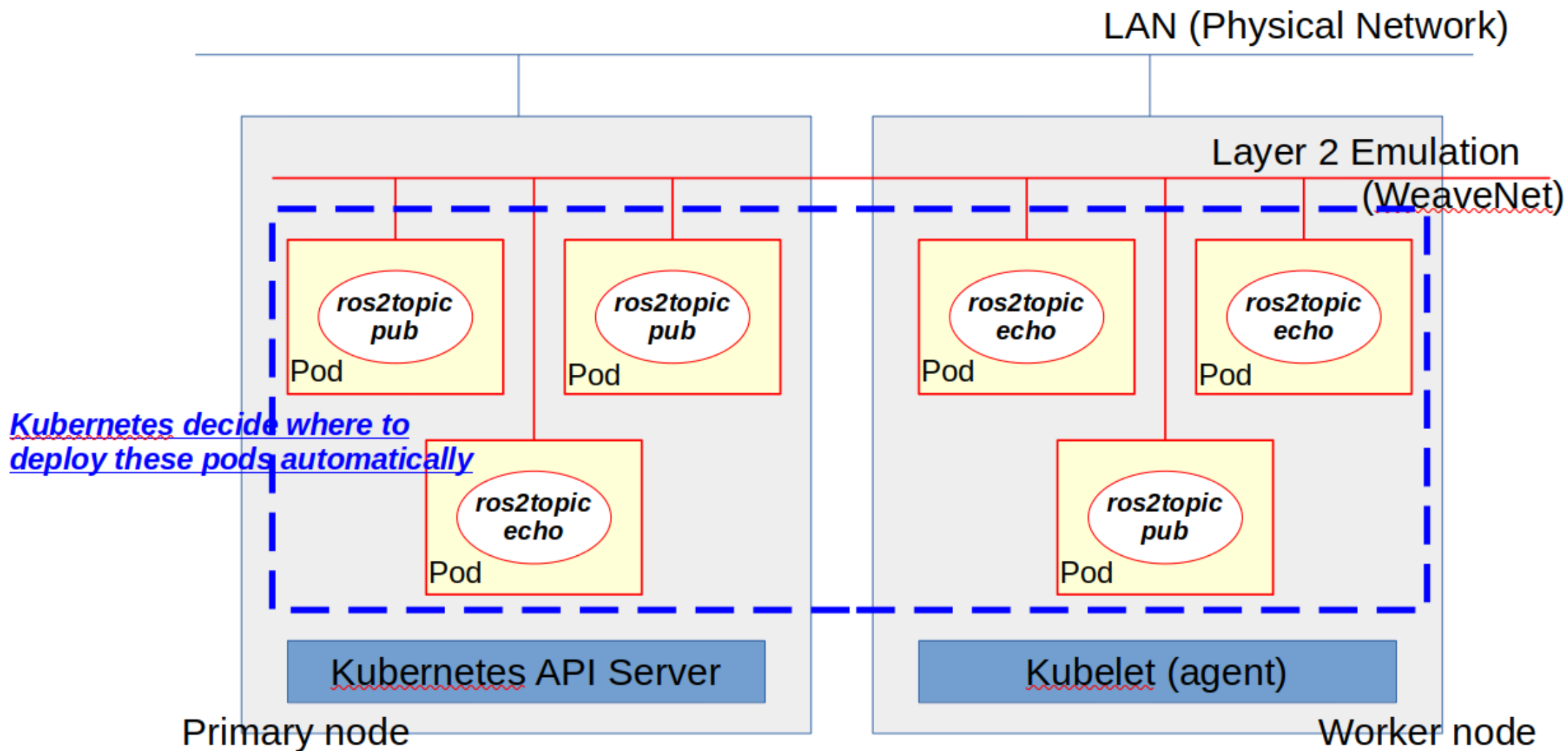
- Far Edge Devices? WasmEdge?

Open Issues

- Add example deployment with Data Sharing via LoanedMessage

- support systemd kubelet to start the cluster automatically

- Edge worker device offers limited resource to be managed by Kubernetes

- Support SROS2 security enclaves via ConfigMap

- Support Nvidia Device-Plugin

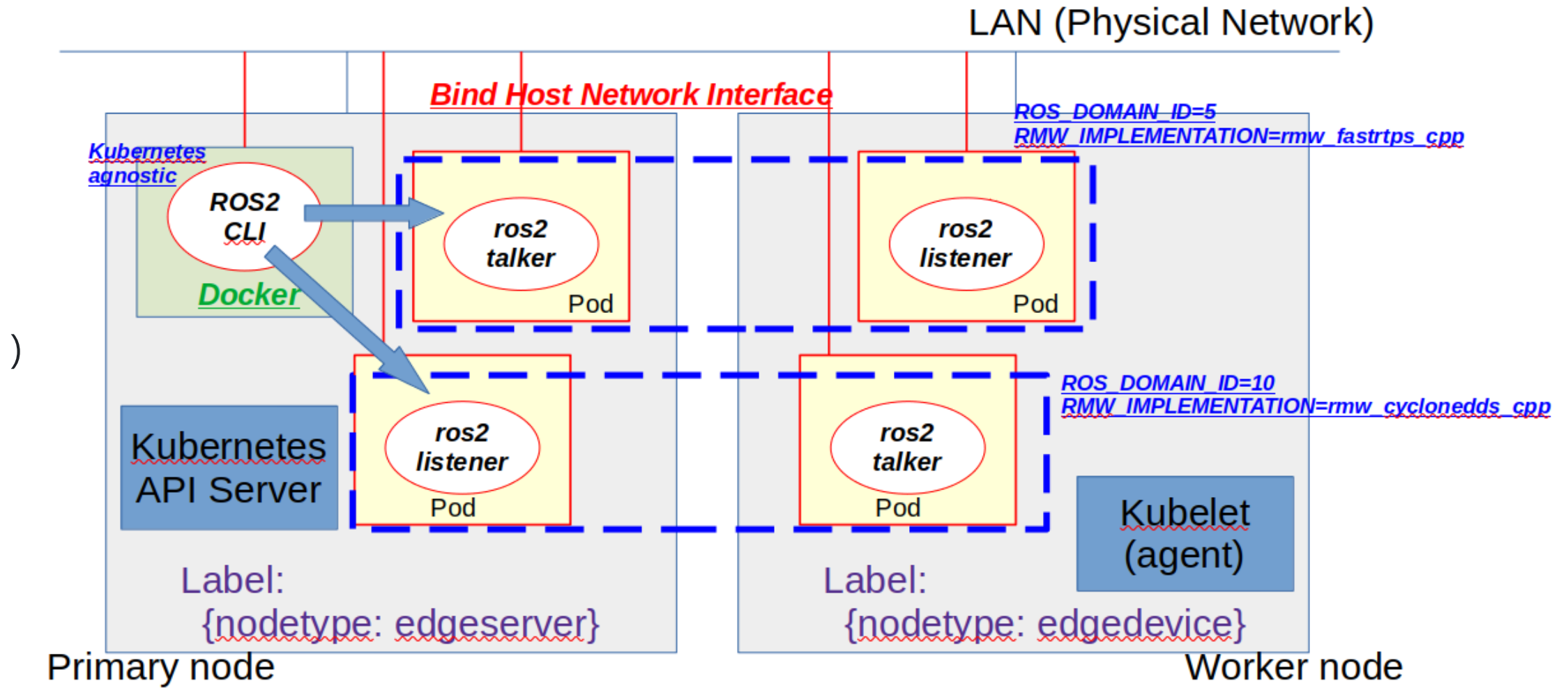- Support KubeEdge tutorial with far edge micro-ROS

Demo

- ROS 1 Multiple Node Deployment

- ROS 2 Simple Distributed System

- ROS 2 Domain ID / RMW Configuration

Check Kuberntes Cluster

LAN (Physical Network)

Bind Host Network Interface

ROS_DOMAIN_ID=5
RMW_IMPLEMENTATION=rmw_fastrtps_cpp

Kubernetes agnostic

ROS2 CLI

Docker

ros2 talker

Pod

ros2 listener

Pod

Kubernetes API Server

ros2 listener

Pod

ROS_DOMAIN_ID=10
RMW_IMPLEMENTATION=rmw_cyclonedds_cpp

ros2 talker

Pod

Kubelet (agent)

Label:
{nodetype: edgeserver}

Label:
{nodetype: edgedevice}

Primary node

Worker node

by Tomoya.Fujita@sony.com / fujitatomoya@github

**Kubernetes
IoT Edge WG**

**KubeEdge
SIG Robotics**

**ROS Kubernetes
Tutorials**

by *Tomoya.Fujita@sony.com* / *fujitatomoya@github*

Questions?

by *Tomoya.Fujita@sony.com* / *fujitatomoya@github*