SOAFEE

PICCOLO

Vision of SDV through PICCOLO and
Current Progress of PICCOLO

Chulhee Lee
2024-05-09

Who is speaker?

Chulhee Lee (이철희)

Leader of PICCOLO Project
Software Defined Vehicle Task, LG Electronics

- **Project Management & SW Architect**
    - In-Vehicle Container Ecosystem Project for 4 years

- **SW Engineer** (Integration & SOA Framework)
    - IVI & Cluster Integrated System Project for 5 years
    - Development Android Applications for 4 years

Software Defined Vehicle

Vehicle Software should be added, updated, deleted continuously, even if released.
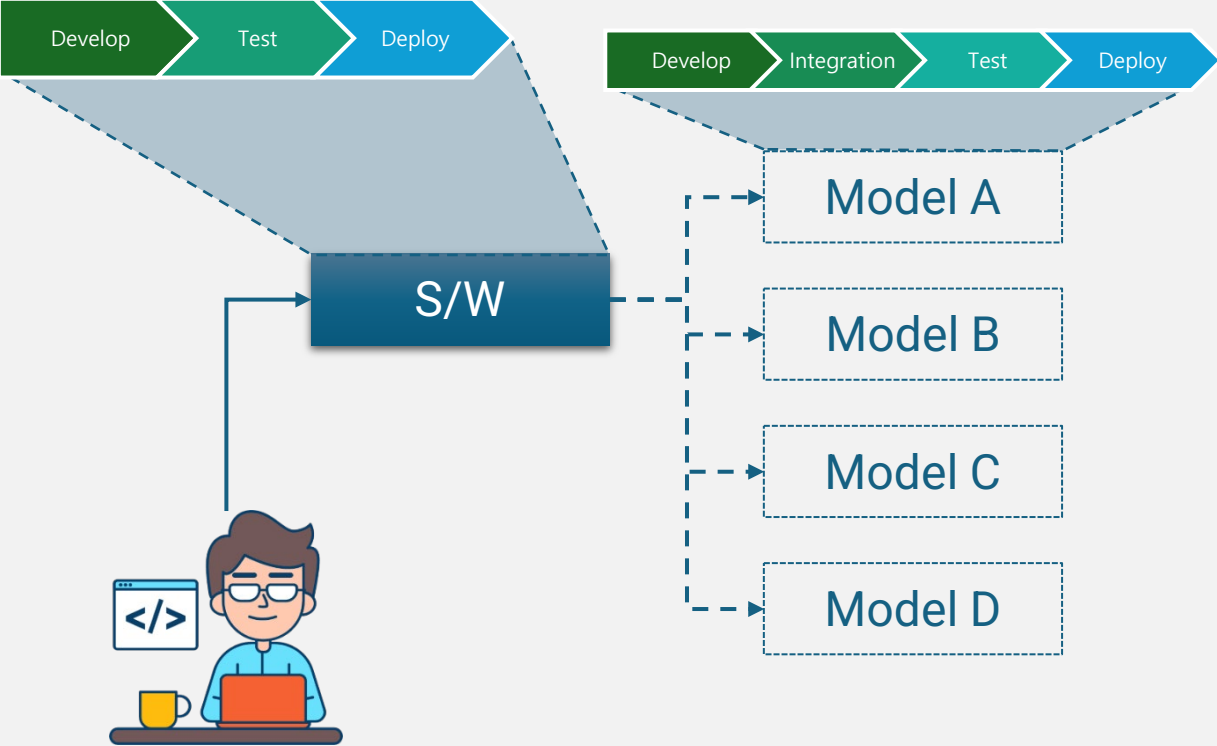
PICCOLO

Mixed Critical Orchestrator

- ## Scenario managing according to manifest
  : Support to add and modify User Scenario after released

- ## Support provisioning for workloads
  : Device & Resource allocations / limits (In development)
  : Workload Manifest Converting (Cloud → Vehicle)
      * According to SOAFEE Specification

- ## Support API for managing workloads
  : High-level API (PICCOLO) / Low-level API (Bluechi)
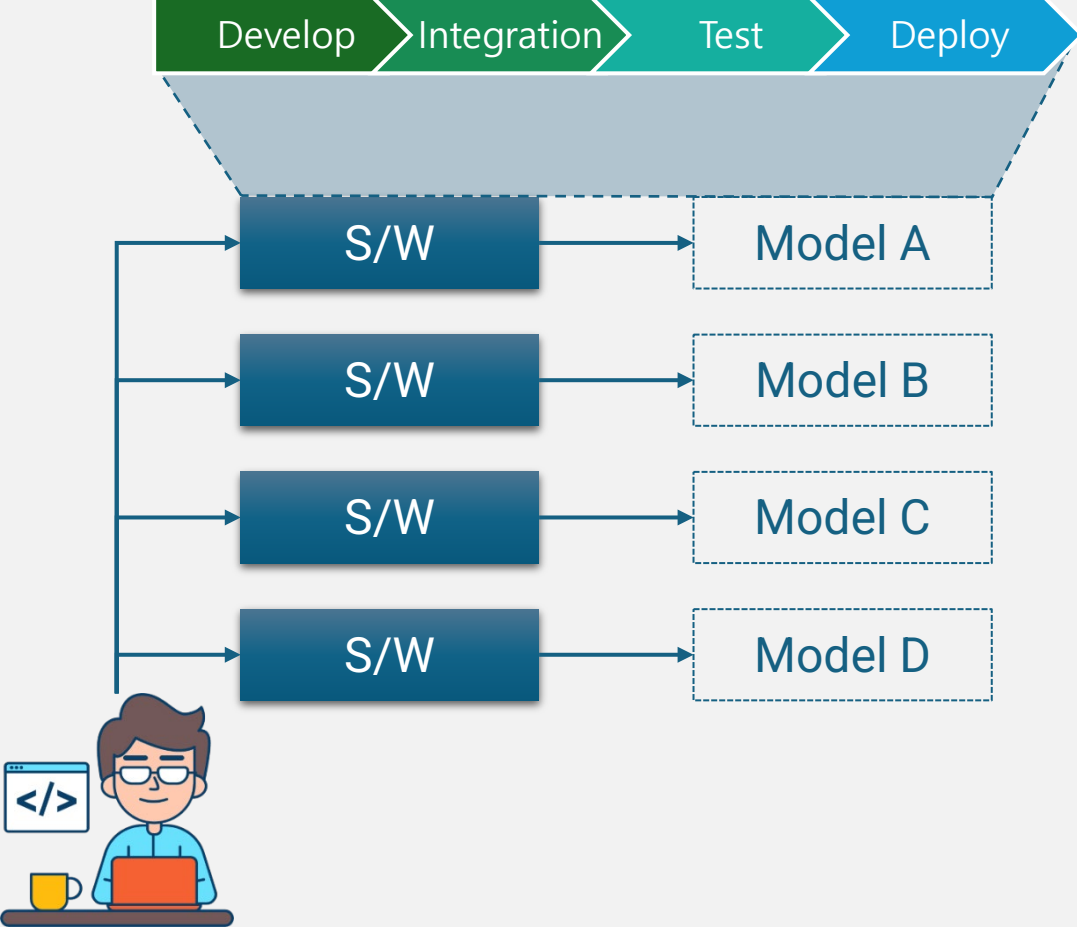 : Implementation Layer of Bluechi

SW Development Process

Cloud Native

Develop | Test | Deploy

Develop | Integration | Test | Deploy

S/W

Model A

Model B

Model C

Model D

Independent Software Development Process

Vehicle

Develop | Integration | Test | Deploy

S/W → Model A

S/W → Model B

S/W → Model C

S/W → Model D

Dependent Software Development Process

When a problem occurs in the software

<u>Cloud Native / Mobile</u>

<u>Vehicle</u>

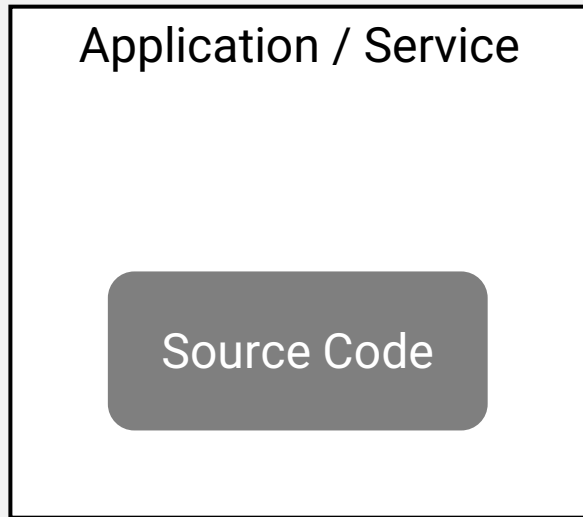Inconvenience

SAFETY Problem

Mixed Critical Orchestration
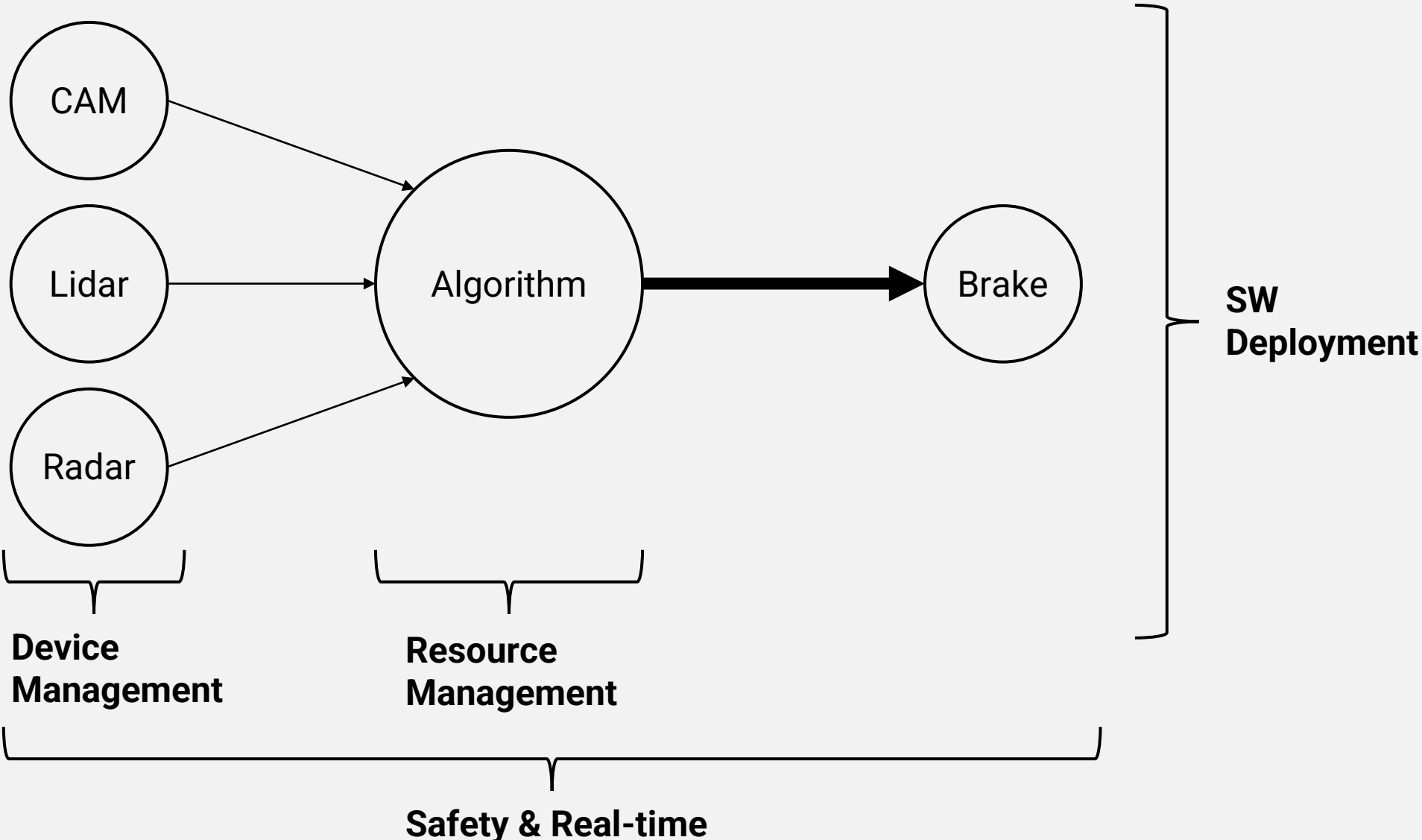
How to manage criticalities

Application / Service

OS / Middleware

SoC

Devices

2. Service & Policy Management

3. Device & Resource Allocation

1. Safety & Real-time

Software   Hardware
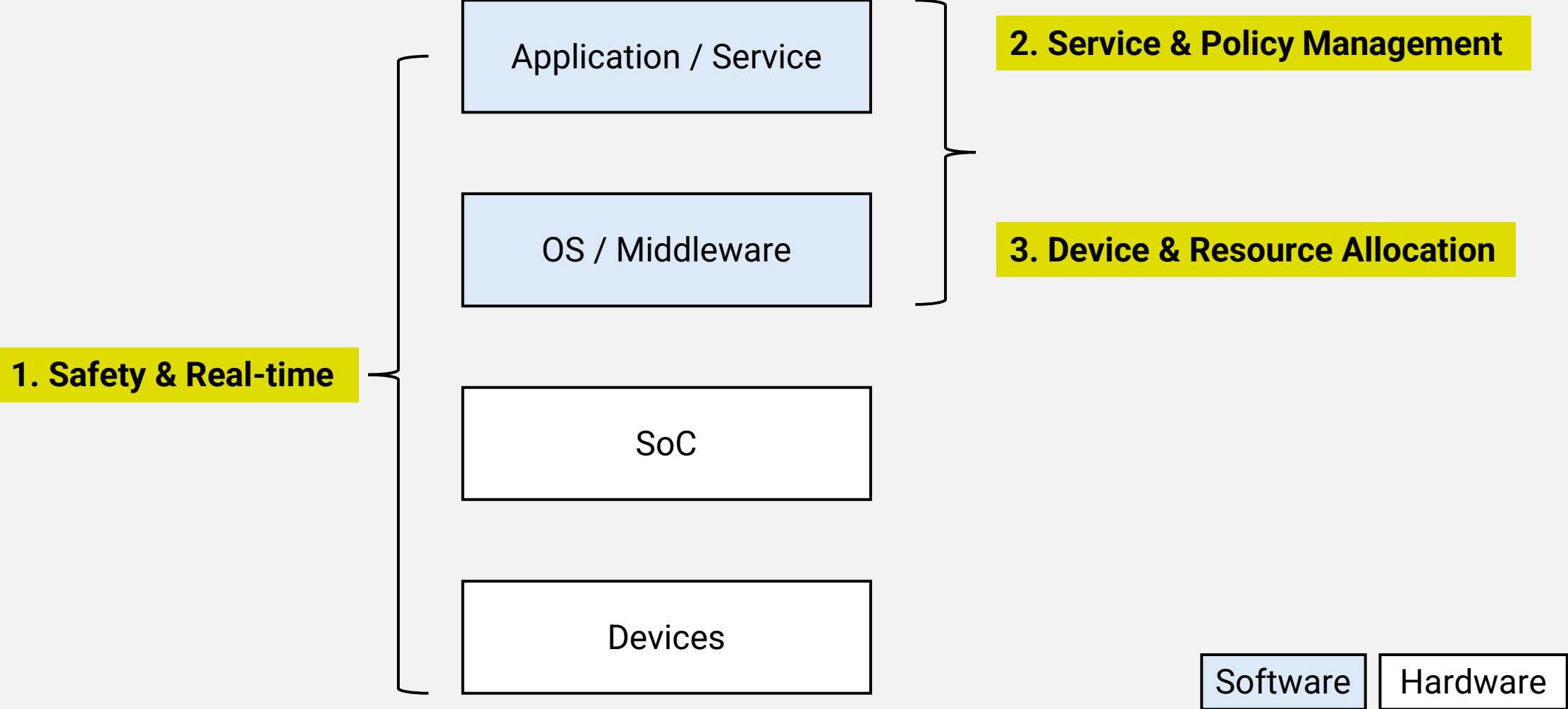
Which points could be changed?

1. Source Code
* Legacy Style

2. Manifest of Service

3. Containers & Manifest

Application / Service (Source Code)

Application / Service (Source Code + Manifest)
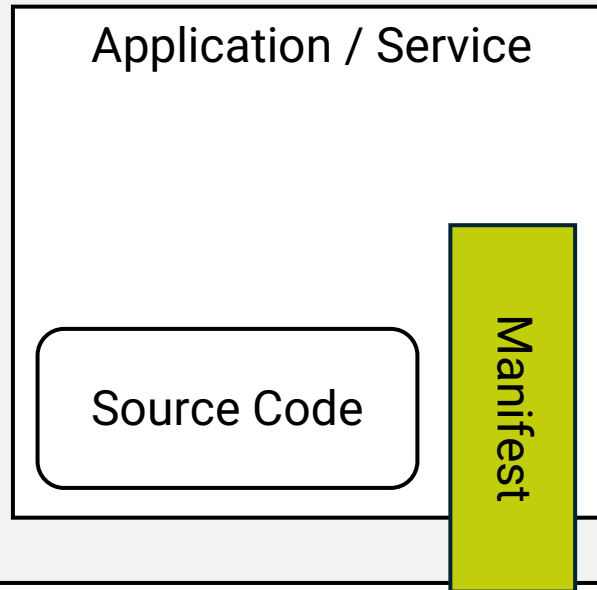
Application / Service (Container → Container → Container + Manifest)
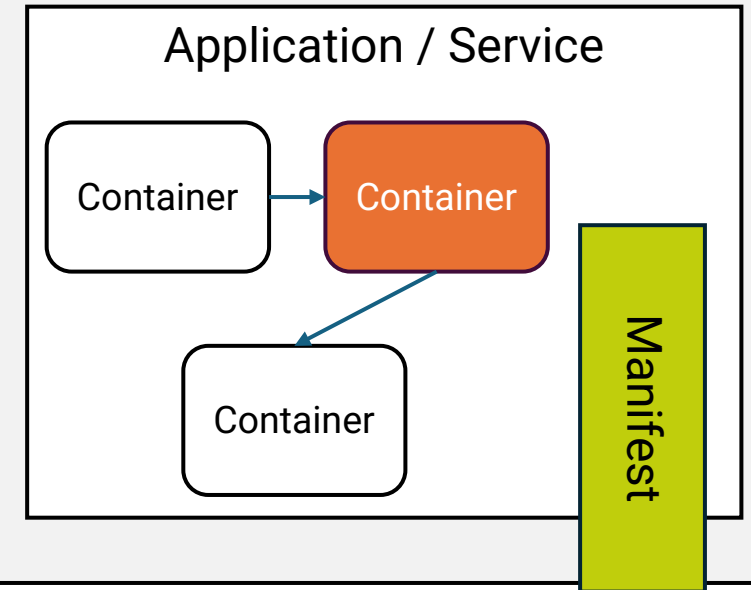
OS / Middleware / Orchestrator (PICCOLO)

The PICCOLO project suggests methods 2 and 3.
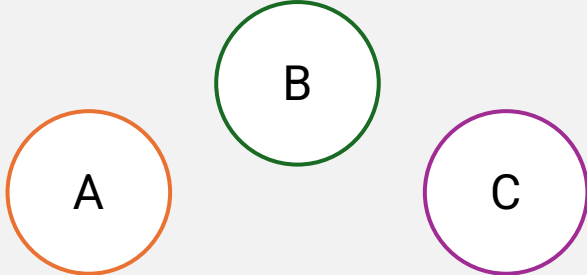
Which attributes could be included in manifest?
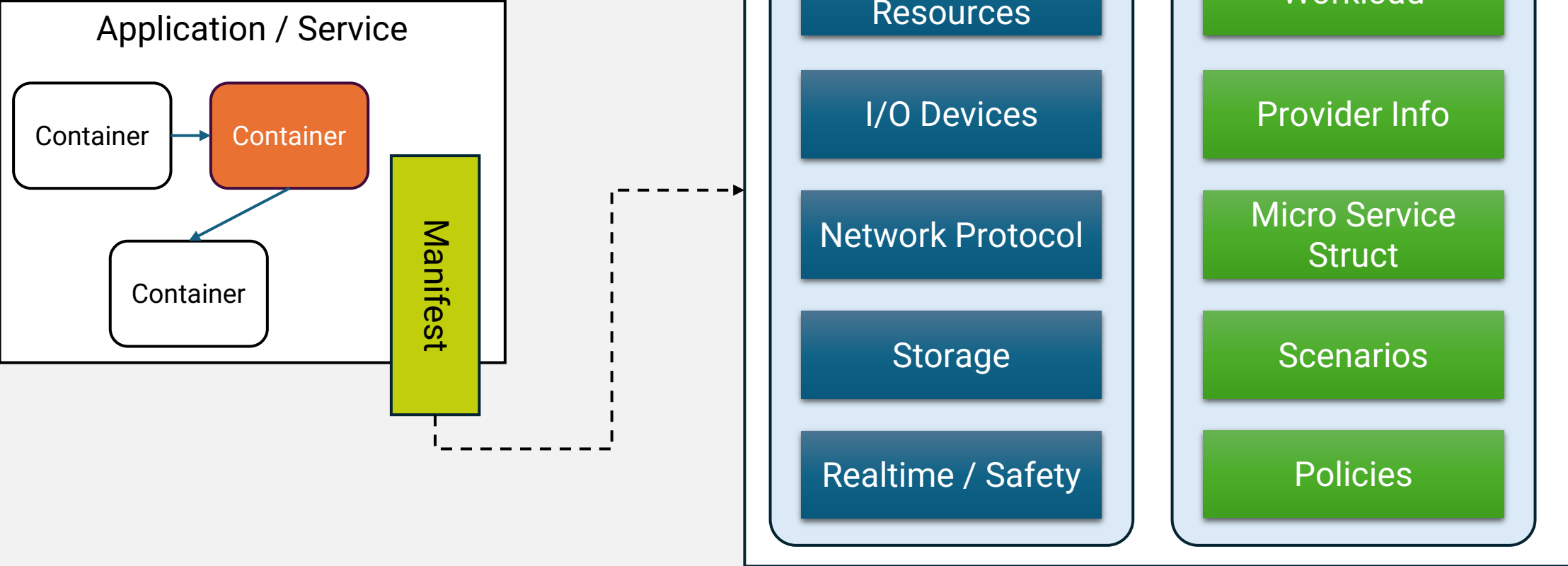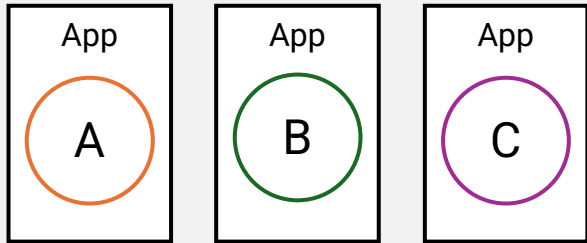

Needs help from companies in various fields.
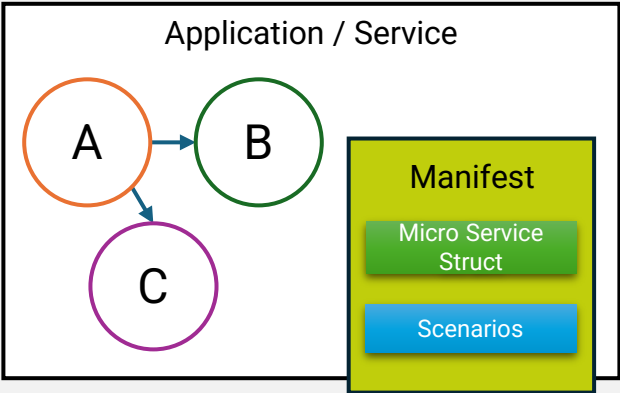
Add / Update / Delete

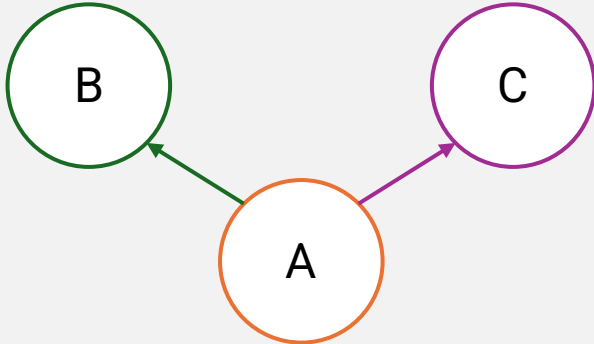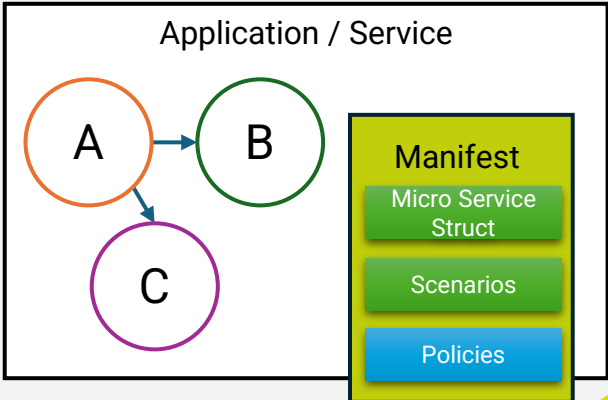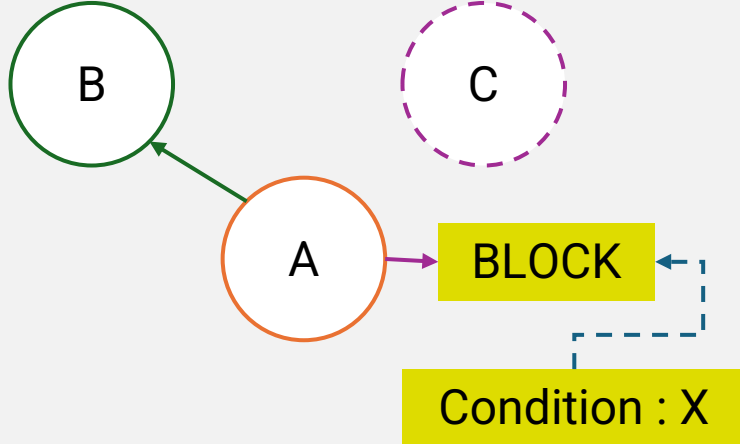# '23 PICCOLO PoC (Switching Scenario)

Current Progress of PICCOLO

Complete

- PICCOLO – Bluechi API Integration
- Scenario managing according to manifest
- Support API-Server based on gRPC
- Workload converter (K8S → PICCOLO & Bluechi)
- Example USE-CASE
  - Managing download workload timing according to manifest

Plan
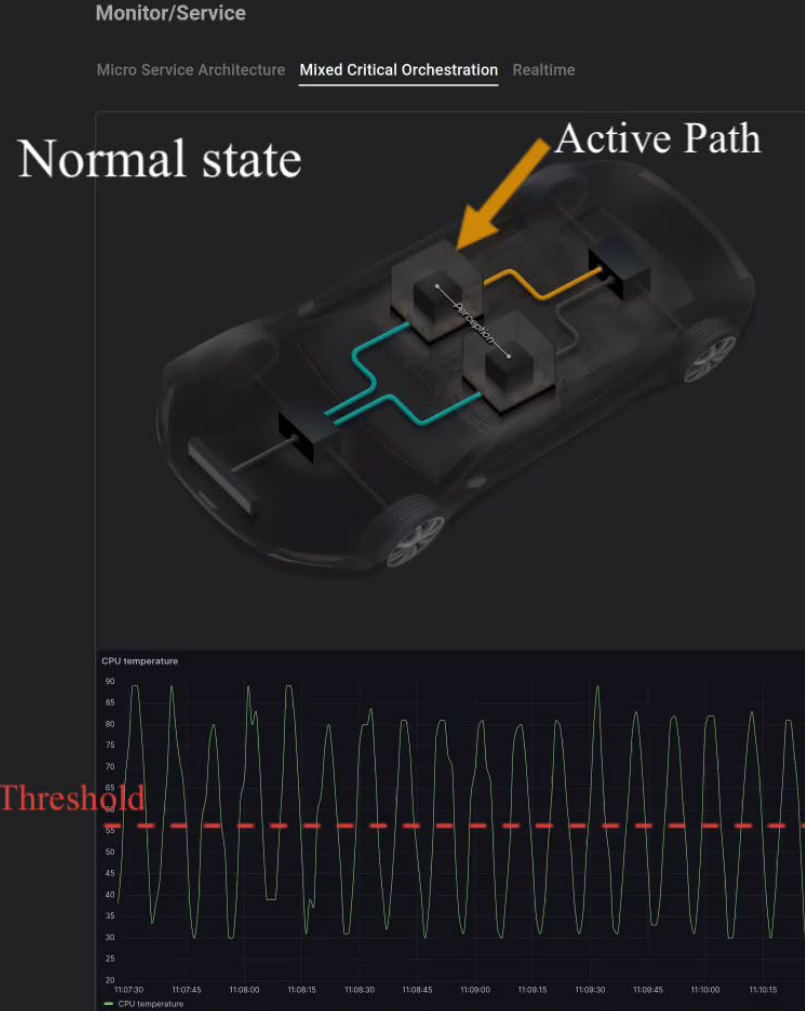
- Contribute to opensource until end of June (~6/30)
- Update additional scenario that supported by PICCOLO

Sample Scenario (Update)


```
 1 apiVersion: v1
 2 kind: Scenario
 3 metadata:
 4   name: version-display
 5 spec:
 6   conditions:
 7     express: Equal
 8     value: "parking"
 9     operands:
10       type: DDS
11       name: gear_state
12       value: "rt/piccolo/gear_state"
13   actions:
14     - operation: update
15       podSpec:
16         containers:
17         - name: display-container
18           image: sdv.lge.com/library/version_display:2.0
19           volumeMounts:
20           - name: x11
21             mountPath: /tmp/.X11-unix
22           env:
23           - name: DISPLAY
24             value: :0
25         volumes:
26         - name: x11
27           hostPath:
28             path: /tmp/.X11-unix
```
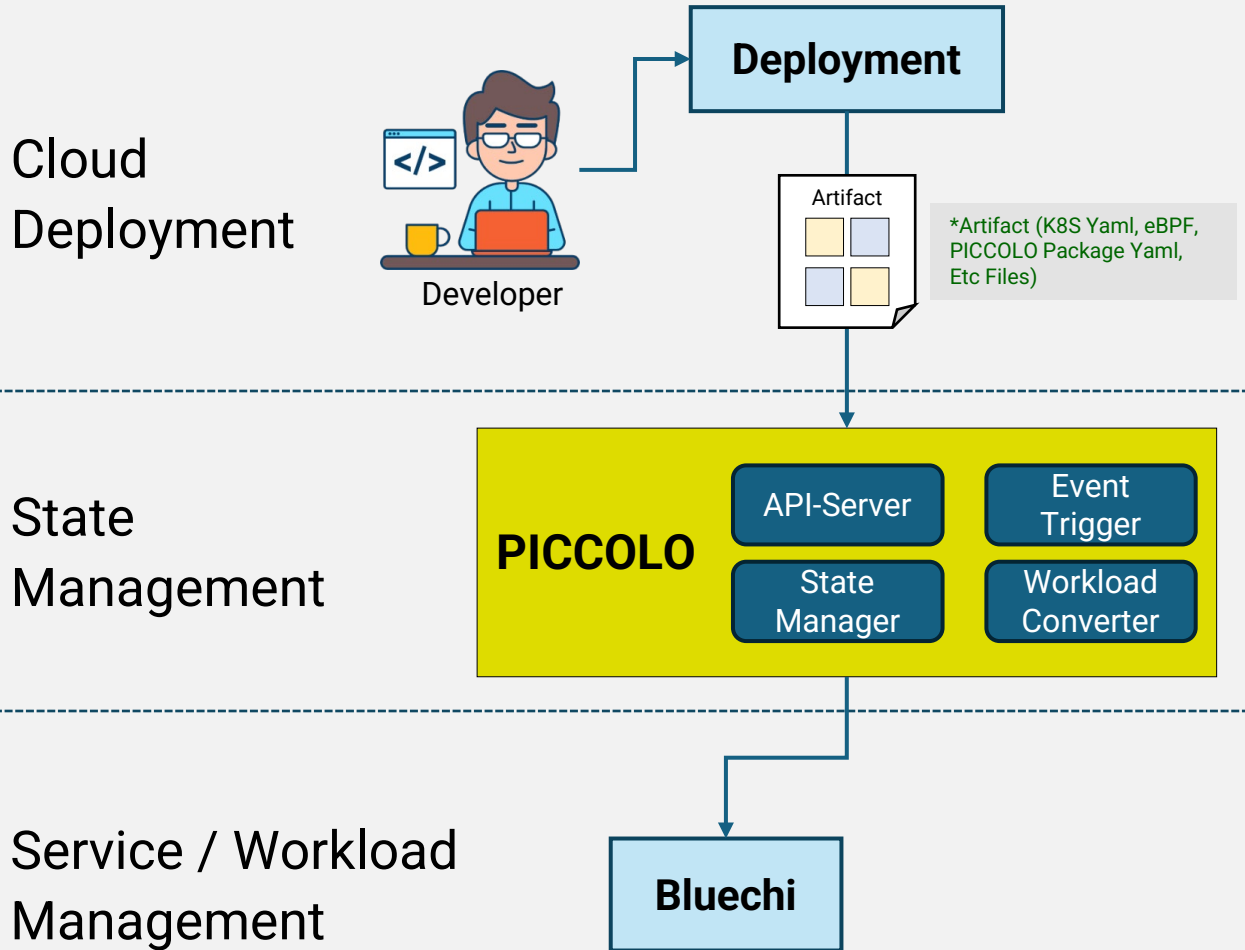
Handwritten annotations:
- Editable Attributes
- Condition (lines 7-8)
- Vehicle Data (lines 9-12)
- Action (PICCOLO API) (line 14)
- Workloads (lines 15-28)

'24 Target Goal

Cloud
Deployment

State
Management

Service / Workload
Management

Deployment

Developer

Artifact

*Artifact (K8S Yaml, eBPF,
PICCOLO Package Yaml,
Etc Files)

PICCOLO

| API-Server | Event Trigger |
| State Manager | Workload Converter |

Bluechi

Opensource Based

Full stack (Cloud – Vehicle)

Workloads Orchestration Platform

SOAFEE

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה

Chulhee Lee (chulhee1.lee@lge.com)
PICCOLO Projects (SDV-PICCOLO@lge.com)