SOAFEE

SDV Challenges and Cloud-native System Design Approach

SOAFEE APAC Seminar

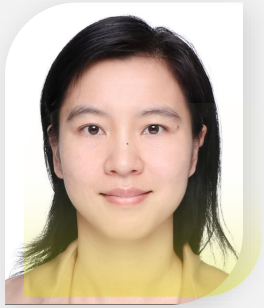
Jian Zhu[1], Wenhung Kevin Huang[2]

[1]DENSO SHANGHAI SMART MOBILITY TECHNOLOGYCO.,LTD.
[2]DENSO CORPORATION

9th May, 2024

Personal Introduction

■ *Who am I*：



JIAN ZHU（朱健）

Optimistic software developers

■ *What I do*

·12 years of experience in car software development. The products related to intelligent cabins and connected vehicles

· 3 years of experience in promoting standardization activities related to AI and SDV

■ *What I love*：

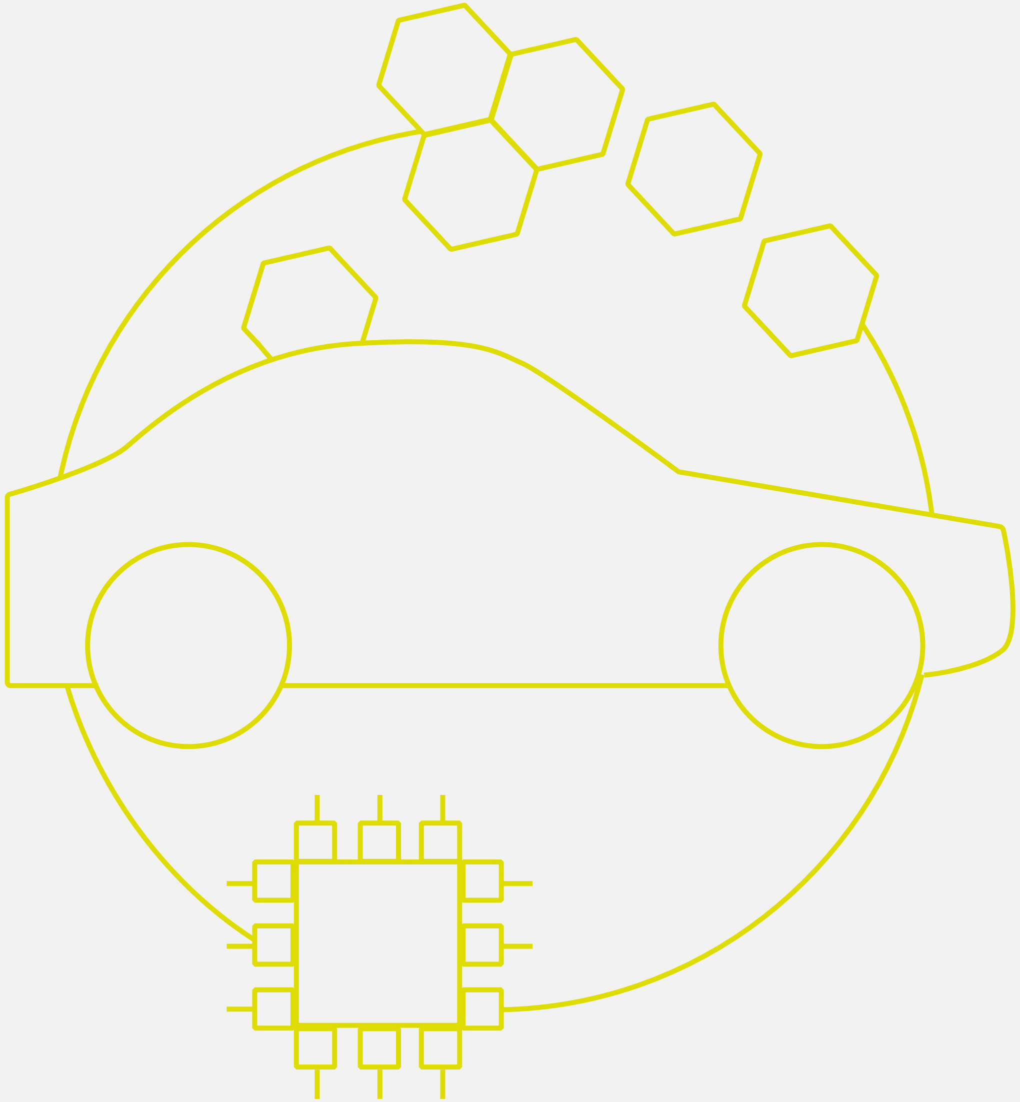
·I enjoy traveling to different cities and experiencing their unique charm

· I like animals that are free and unrestrained in nature

Agenda

1. Introduction to DENSO Corporation
2. SDV Challenges and Ecosystems
3. Cloud-native System Design Approach
4. Conclusions and Future Works

1. Introduction to DENSO Corporation



Introduction to DENSO Corporation

DENSO
Crafting the Core

DENSO is a global company focused on advanced mobility that positively changes how the world moves and contributes to greater well-being. As a global Fortune 500 company, we have a broad product portfolio and widespread global impact.

DENSO's Future Direction

4 core technologies

- Electrification
- Advanced Safety and Automated Driving
- Connected Driving
- Factory Automation /AgTech

The DENSO Group **190** companies

Total number of employees **164,572** people

35 countries and regions

denso.com/global/home/about-us/at-a-glance/

DENSO China

1987 Starting to develop	manufacture main products	About 17000 employees

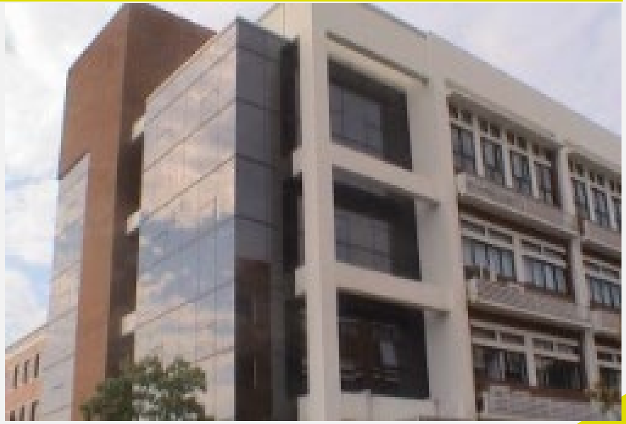
DENSO(China) Investment Co.,LTD

2003 Established	974 Employee	Investment &Omnibus

DENSO SHANGHAI SMART MOBILITY TECHNOLOGYCO.,LTD.

2002 Established	182 Employee	Software development

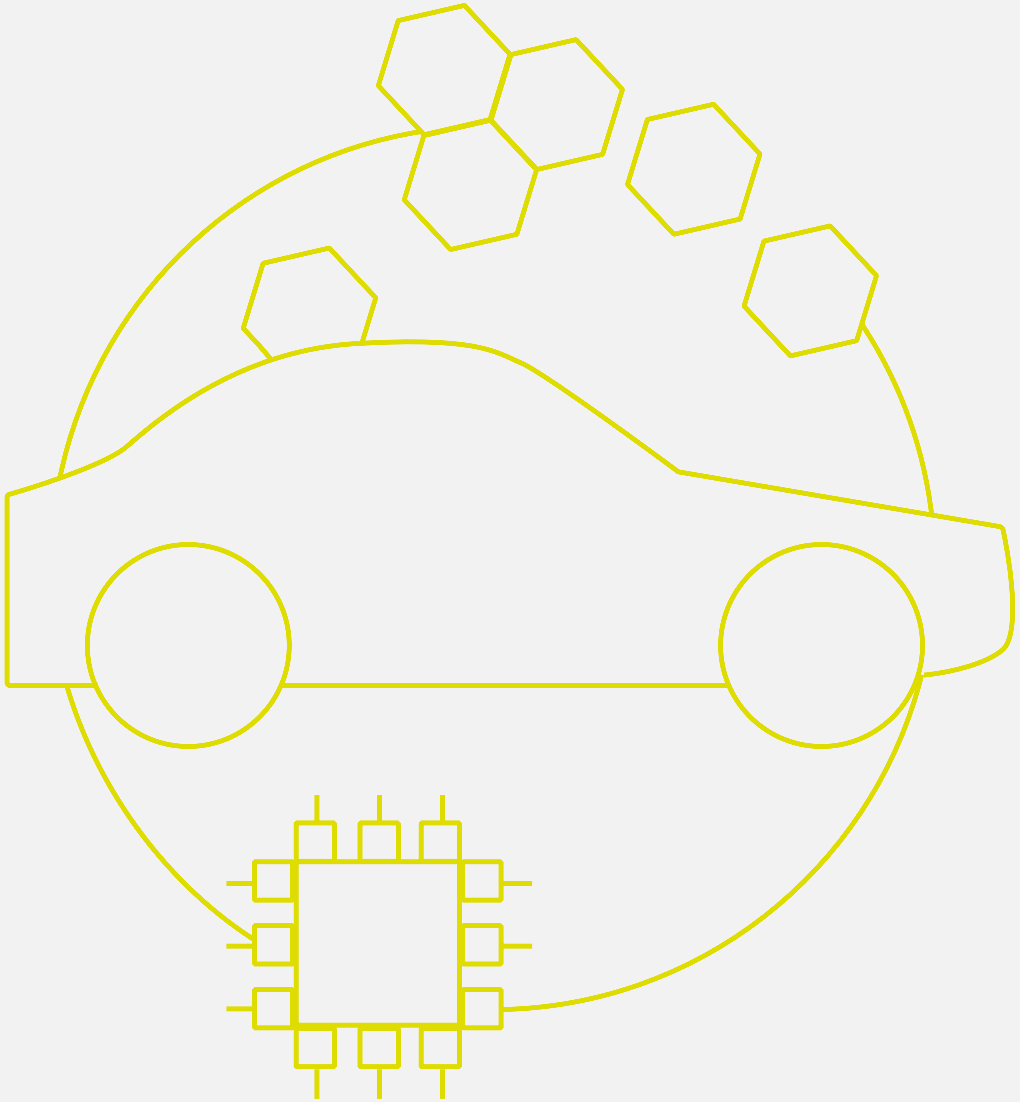
Business Management
System Hardware



Core Technology Development
System Software

SOAFEE

2. SDV Challenges and Ecosystems



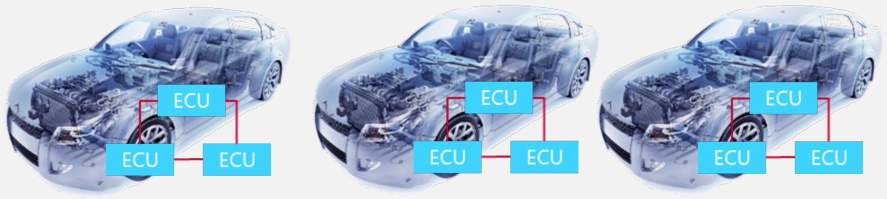
SDV (Software Defined Vehicle)

- SDV stands for Software Defined Vehicle, which means "a car defined by software.".
- In the past, cars improved their performance by improving the hardware centered around internal combustion engines and engines, but in the future, in car software will determine the value of cars.

Concept structure of SDV system

SDV App | SDV App | SDV App

SDV Software Platform(Virtual ePF)



ePF-A ePF-B ePF-C

- The concept and mechanism of **abstracting vehicle hardware** (electronic Platform -> Virtual ePF))
  - ECUs, in-vehicle networks, sensors, and actuators with **virtualization technology**
- **Software controlling** these computer resources
- In other words, "How to **separate apps, software, and hardware**"

The development/production of automobiles has shifted from hardware centric to software centric.

The changes and challenges brought by SDV to the automotive industry

◆ ***The changes brought by SDV***

➢ **Reduced the difference between hardware and software development**

More standardized development reduces the complexity of software and hardware integration

➢ **Hardware and software decoupling**

Under the SDV concept, the OTA is given more attention, promoting the decoupling of hard and soft.

➢ **Creating new business models**

The automotive industry is no longer just selling hardware but bringing new profits to OEMs by providing services.

◆ ***The challenges brought by SDV***

➢ **Architectural design**

In order to achieve rapid development and iteration, it is necessary to design a multi-modular automotive software architecture with low correlation between modules,.

➢ **Functional safety**

In the process of software upgrades, it is necessary to test and verify functional modules with different safety requirements to ensure safety.

➢ **Information Security**

Information protection and control technology are required. For example, verification of data sources, and verification of data correctness and timeliness.

SDV has had a significant impact on the development of automobiles, posing multiple challenges.

SDV Ecosystem in China and OSS

In Global：

- **Global organizations are committed to establishing open source technology standards and aiming to build a sustainable ecosystem.**
- **e.g. Standardized and open E/E system Architecture;** Connected vehicle systems; Open Technology Platform for SW; Cloud-native architecture .

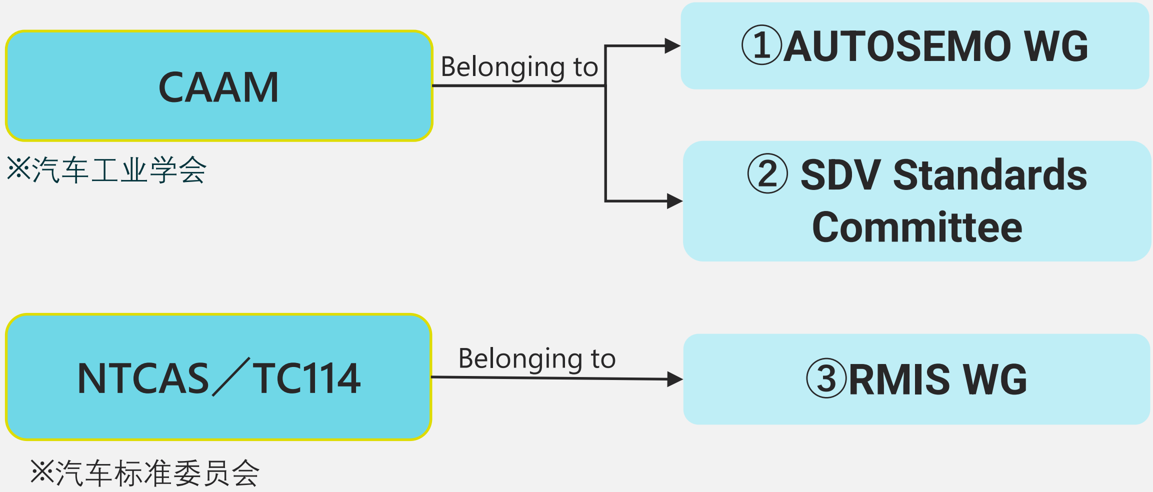
In China:

- **Participants in the automotive industry hope to work together to build a Chinese automotive ecosystem, while also promoting their own standardization system to respond to changes in demand in the Chinese market.**
- **e.g.**
- **With the nationalization of chips, it is necessary to establish a development ecosystem that can adapt to domestic chips.**
- **With the improvement of laws related to automotive data storage, the standardization of localized cloud services is also accelerating.**

Denso contributes to SDV ecosystems both globally and in China.

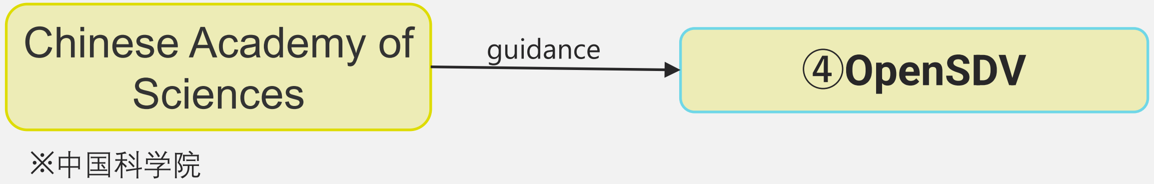
Status of SDV Standardization in China

◆ ***Standardization organization***



- AUTOSEMO actively convened forums and committees, advocating for intra industry cooperation and ecological construction.
- The SDV Standards Committee advocates the establishment of a technical expert committee to ensure the resolution of inconsistent technical standard definitions and internal coordination.
- The RMIS WG comprehensively considers the relationships between different products, functions, and technology types, and has formed a new standard system for intelligent connected vehicles in 2023.

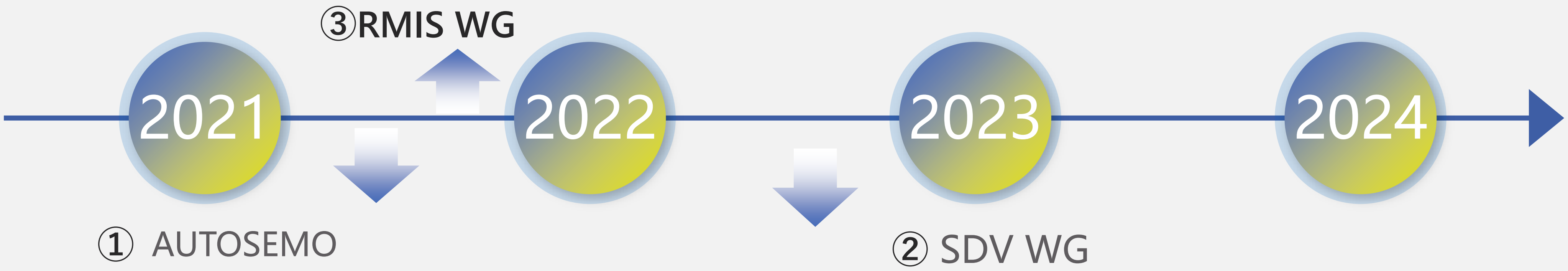
◆ ***Technology Alliance***



- In the field of autonomous driving, As a open source WG to promote technological breakthroughs, code co construction, standard cooperation, and technological applications under the open source approach.

The Chinese automotive industry is calling for “intra industry cooperation and ecological construction”, and is currently in the stage of practical and verification of achievements

DN's Standardization activities in China



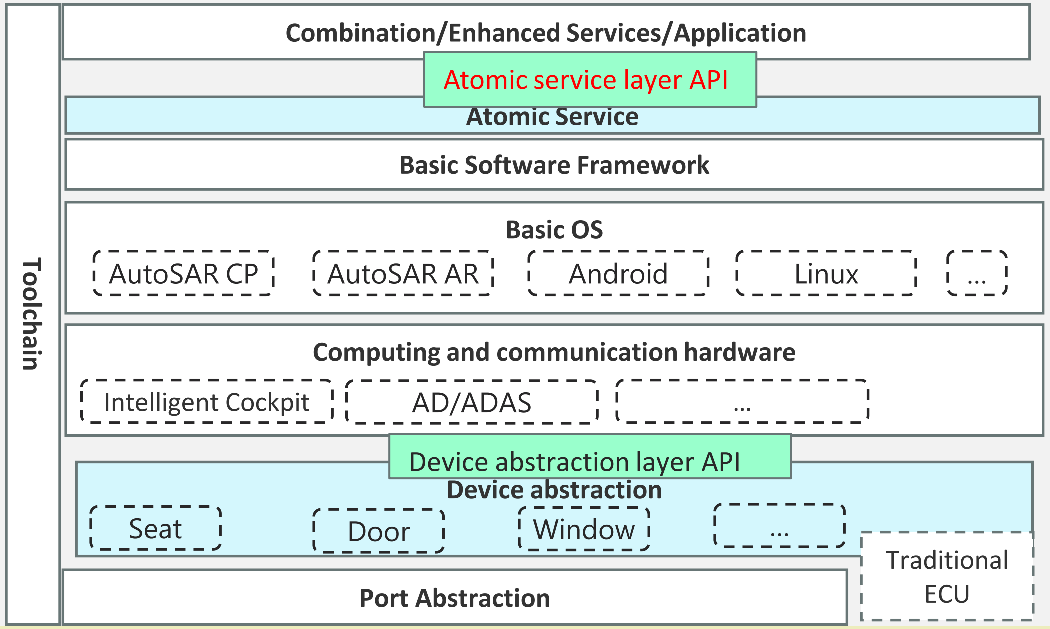
As DN China, we have participated in activities ①, ②, ③ and are currently conducting trend analysis to prepare for future standardization promotion.

DN’s Standardization Activities in「SDV WG」

◆ API improvement proposal

Suggestions for improving the API of the Atomic service layer(※原子服务层)

e.g. modify interface description of VCS TqCtr (lift torque request ->control torque request)



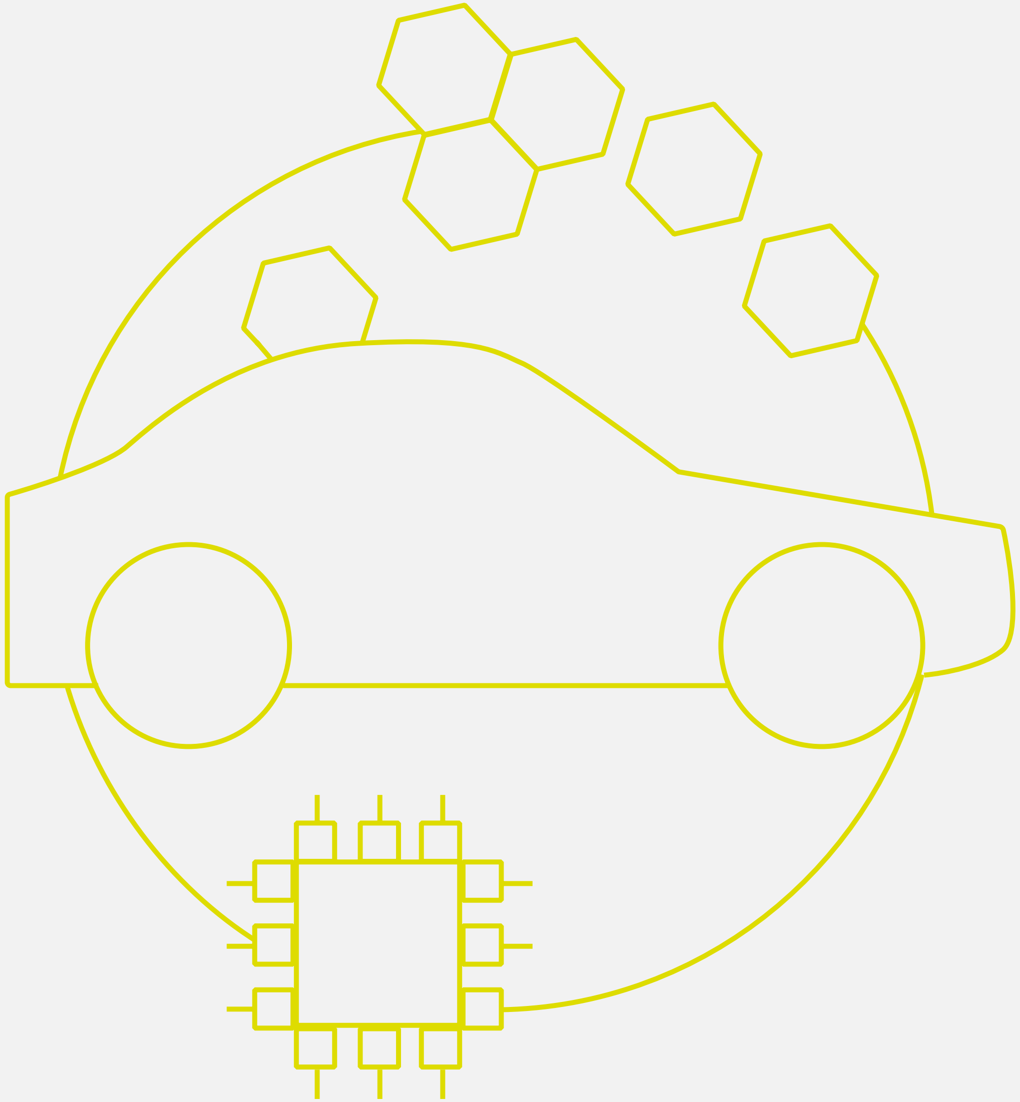
◆ Participate in industry research

Actively participate in the 「Notice on Conducting SDV Industry Application Research」.Continuously share the challenges encountered in product development.

e.g. a complete toolchain that supports SOA does not yet exist, and each tool itself is not sufficient to support distributed development across multiple regions. Some AUTOSAR design tools currently do not support comparing differences between versions, making it difficult to ensure development quality。

In China, DN actively participates in pre standard discussions and industry research, striving to promote standardization.

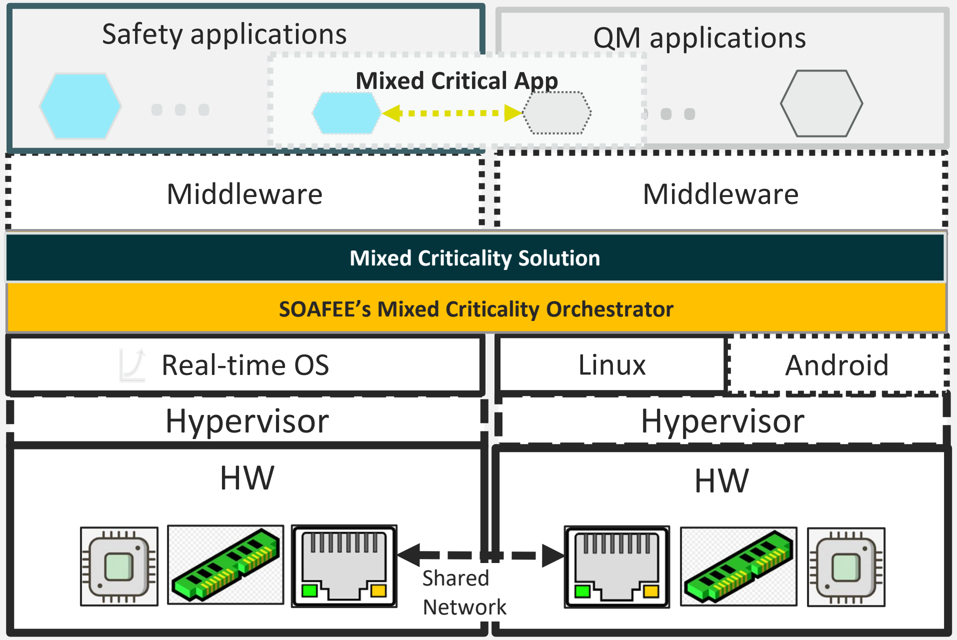
3. Cloud-native System Design Approach



Mixed Criticality (MC)

- **SOAFEE's Mixed Criticality orchestrator concept**
  - Hardware abstractions for criticality agnostic application interface
  - Advanced virtualization methods involving resource management

Proposed Mixed Criticality Runtime



- **DENSO's Mixed Criticality solution:**
  - Provides an **application-level safety** envelope for handling uncertainties
  - **Deterministic scheduling** methods for handling real-time requirements at the application interface
  - **Safety violations** detected at runtime

The combination of the two concept is key to the realization of MC applications

Lingua Franca

An actor-based synchronous reactive programming paradigm with a logical model of time

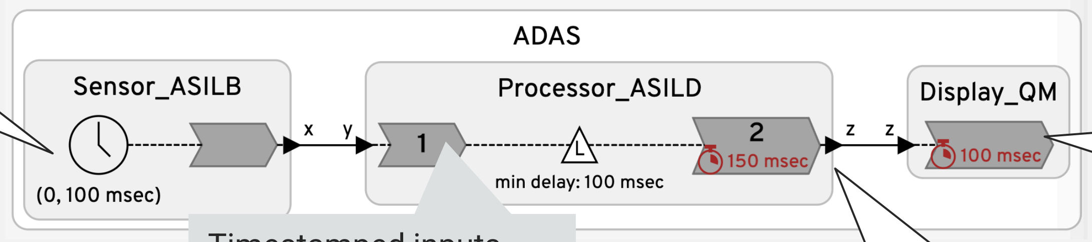
→ System Modeling

- Modeling software as reusable components

→ Deterministic scheduling

- Provide a runtime that enables efficient deterministic concurrency
- Support deadline-based error detection

Periodic sensor input



Deadline handler invoked if expected event not triggered within 100 msec

```
reactor Sensor_ASILB{
    timer t(0, 100 msec);
    output x:int;
    state count:int=0;
    reaction(t) -> x {=
        lf_set(x, self->count);
        self->count++;
    =}
}
```

Timestamped inputs

Logically instantaneous outputs

Local state variables

Reaction describing time encoded event behavior

Target language code

Factoring in WCET of this **workload**

```
main reactor {
    sensor = new Sensor_ASILB();
    processor = new Processor_ASILD();
    ivi = new Display_QM();
    sensor.x -> processor.y;
    processor.z -> ivi.z;
}
```

Lingua Franca semantics allow us to model and develop deterministic application code

System Modeling

Model as Code VS Model is Code

Model as Code

Templating Language YAML/JSON/TOML - -
k8s,CloudFormation

→ Pros
- Easy to read for human

→ Cons
- Too complex to be used for production-grade manifests

```
application:
  name: Sensor
  asil: B
  output:
    name: x
    type: int
    targetPort: Processor.y
 --
application:
  name: Processor
  input:
   name: y
```

Model is Code

Programming language/DSL - ✓
CDK8s, Lingua Franca

→ Pros
- More readable and production-grade manageable

→ Cons
- learning curve is steeper

IDL

```
reactor Sensor_ASILB{
        timer t(0, 100 msec);
        output x:int;
}
```

wiring

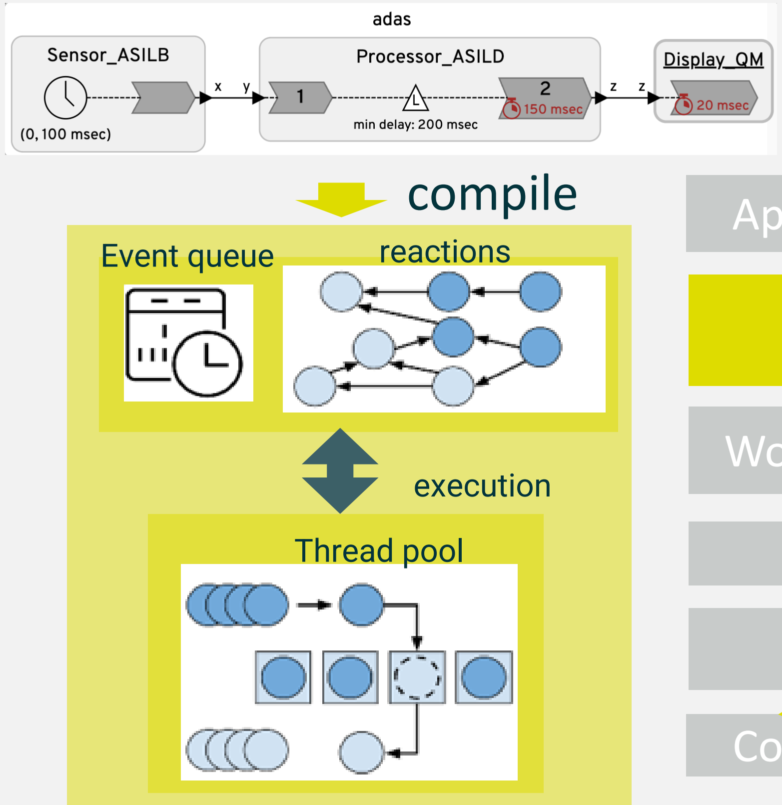
```
main reactor {
        sensor = new Sensor_ASILB();
        processor = new Processor_ASILD();
        sensor.x -> processor.y;
}
```


Deterministic Scheduling

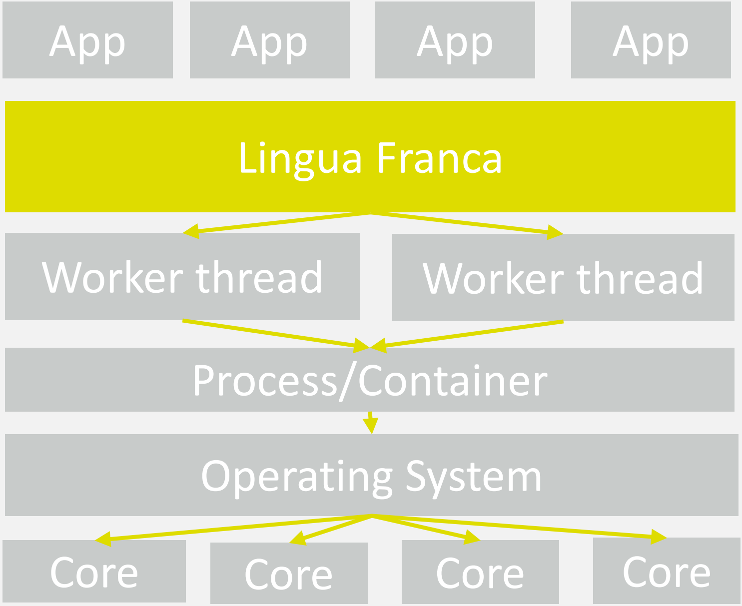
A runtime with determinism and parallelism enable

→ **Determinism**
- Components inform the scheduler at what logical time to trigger reactions

→ **Parallelism**
- The runtime exploits parallelism by the dependencies between reactions in the dependency graph



Control event flow through scheduling algorithms



Automated Valet Parking: Problems and Approach

→ Automated Valet Parking

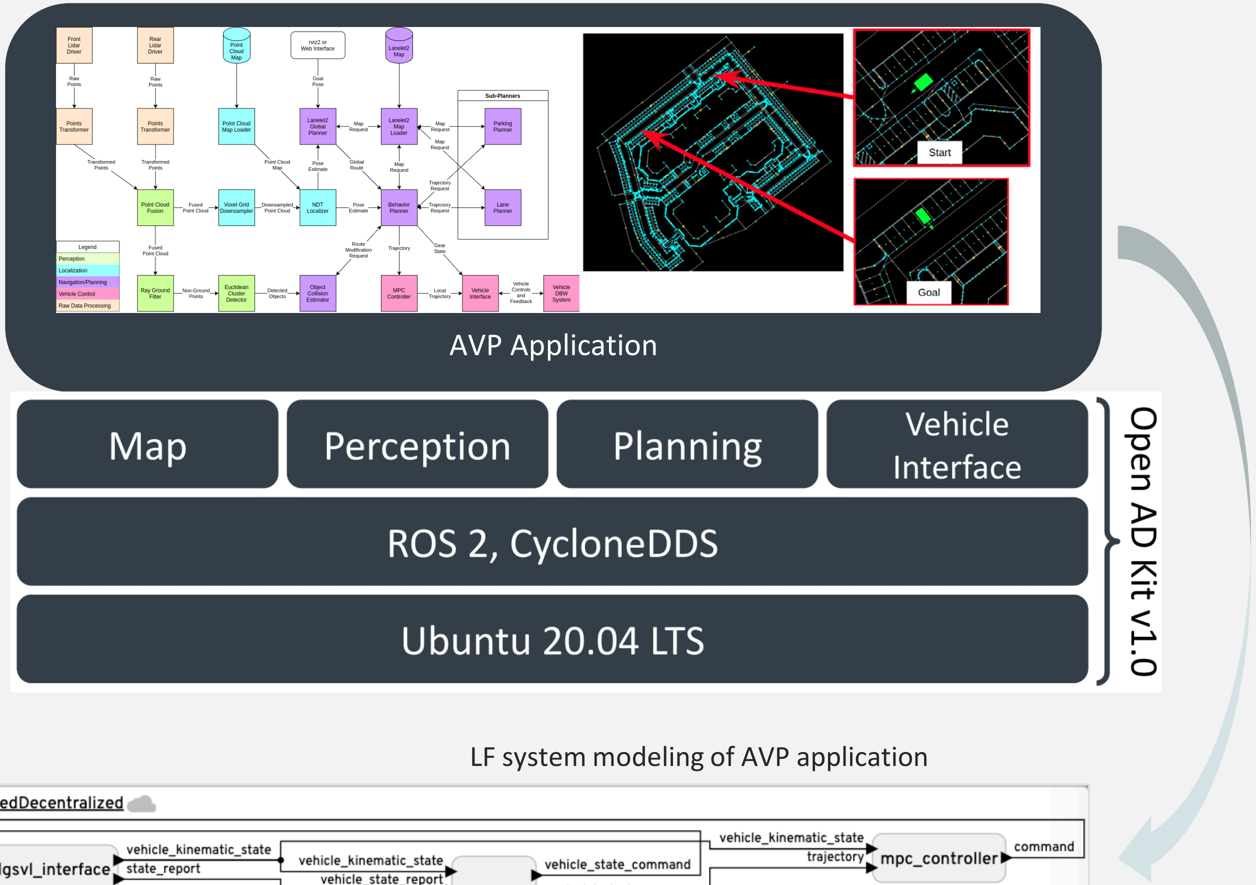
- AD application to autonomously park and return to a pick-up/drop-off area in a parking lot
- Autoware Foundation provided blueprint to show how such a service can be integrated with SOAFEE SDV reference architecture

Problems

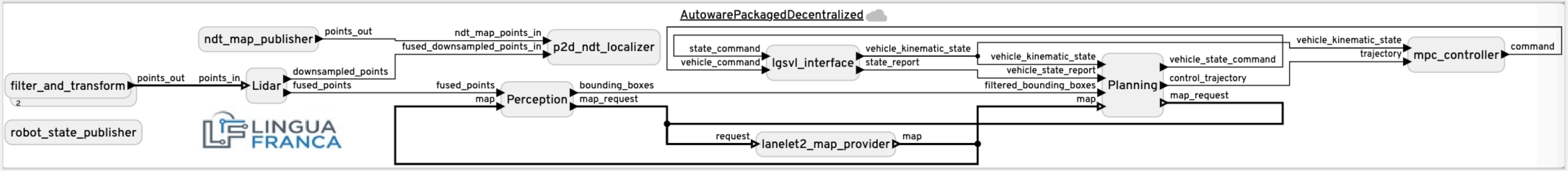
- AVP exhibits non-deterministic behavior (Eg: unresponsiveness, jitteriness, etc.) on SDV platform ▶
- This problem highlights the importance of deterministically scheduling various interacting subcomponents

Approach

- LF enforced deterministic scheduling to suppress observed issues in original demo



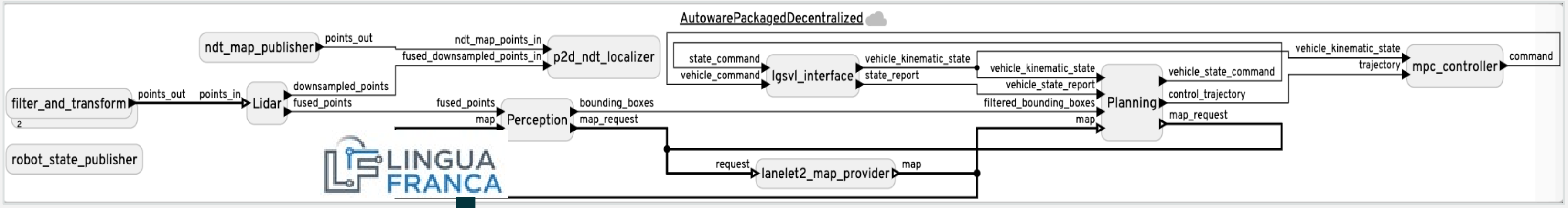
LF system modeling of AVP application

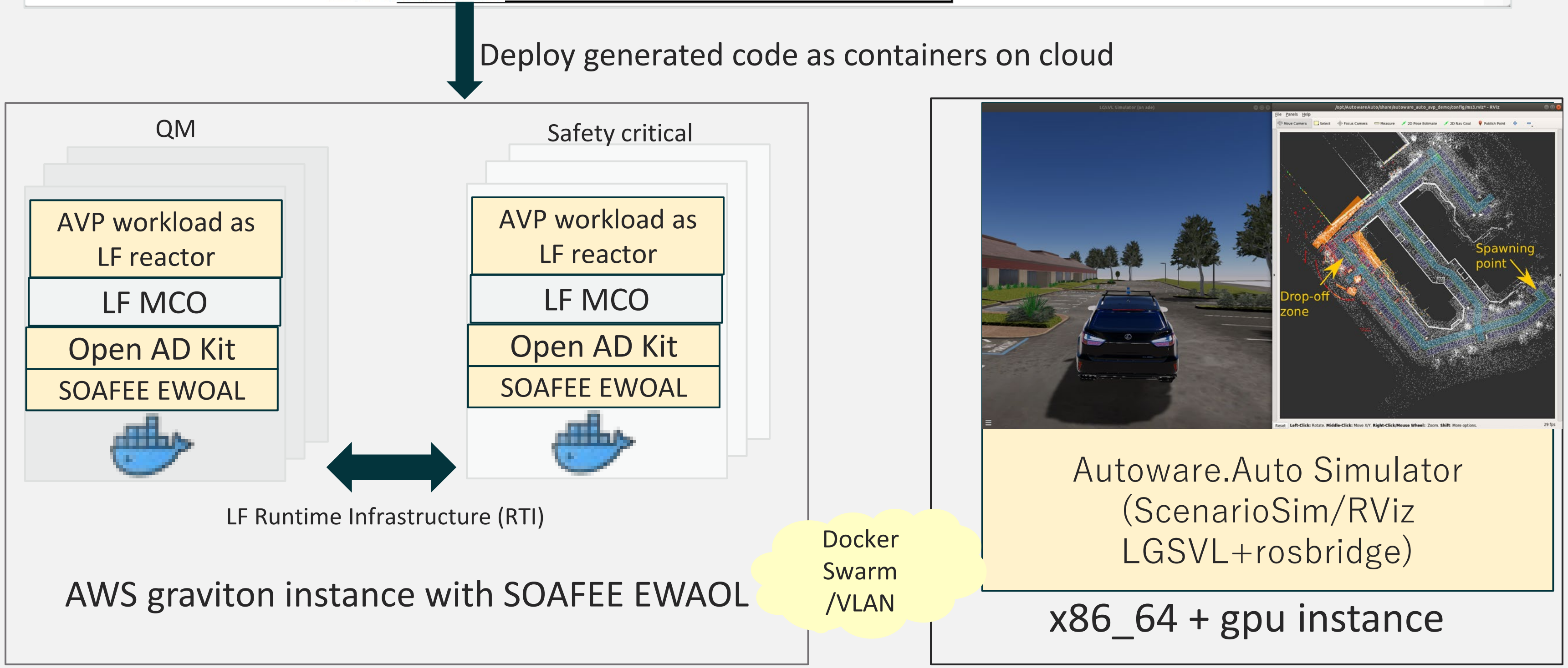


Integrated LF and Open AD Kit application on SDV

Demonstrate LF as a mixed critical orchestrator solution on SOAFEE reference architecture using AVP

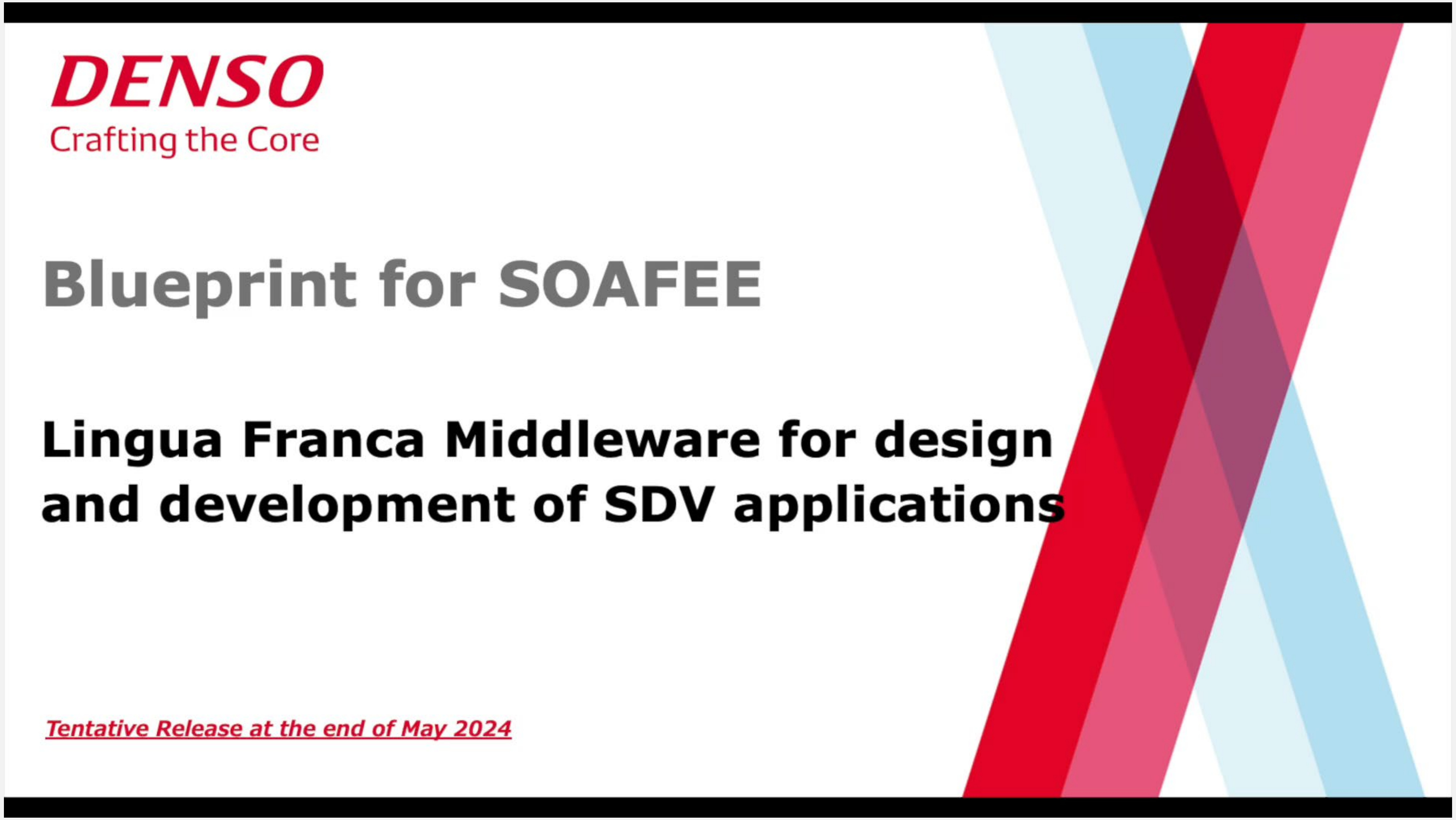
- LF Mixed Critical Orchestrator (MCO) manages the scheduling across containerized workloads
- Porting ROS2 nodes to LF
- In current configuration, safety critical and QM containers run on virtual High Performance Compute (HPC)
- The default Autoware simulator LGSVL is used





*Evaluation on mixed criticality hardware setup is the next step.
Testbed: NVIDIA Orin (as HPC) + (R-car S4 as Safety Island)

Demo of Automated Valet Parking using LF



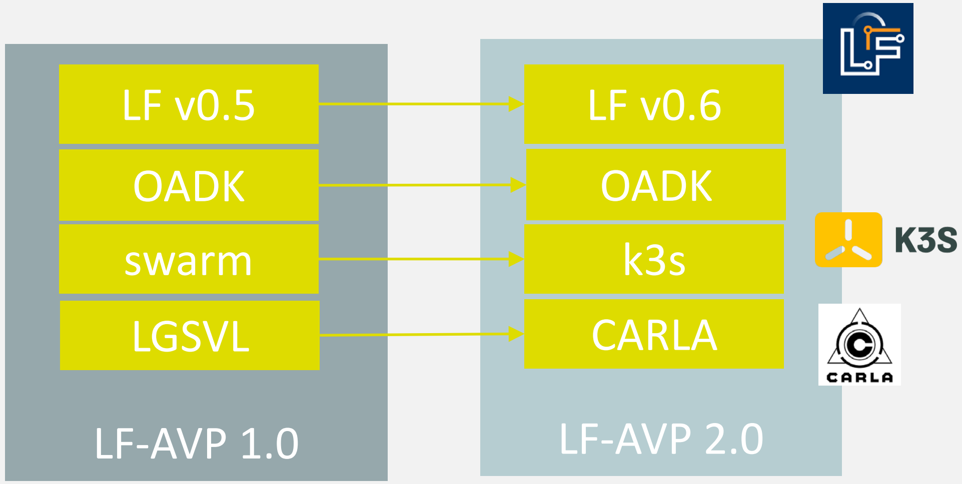
Ongoing work for Blueprint of AVP v2.0

Lingua Franca

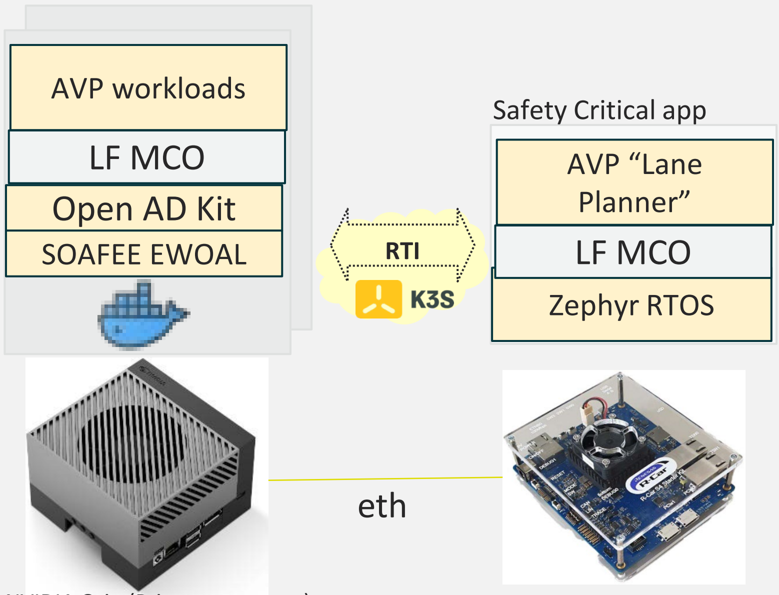
Open AD Kit

Container orchestrator

Simulator



Mixed criticality hardware testbed

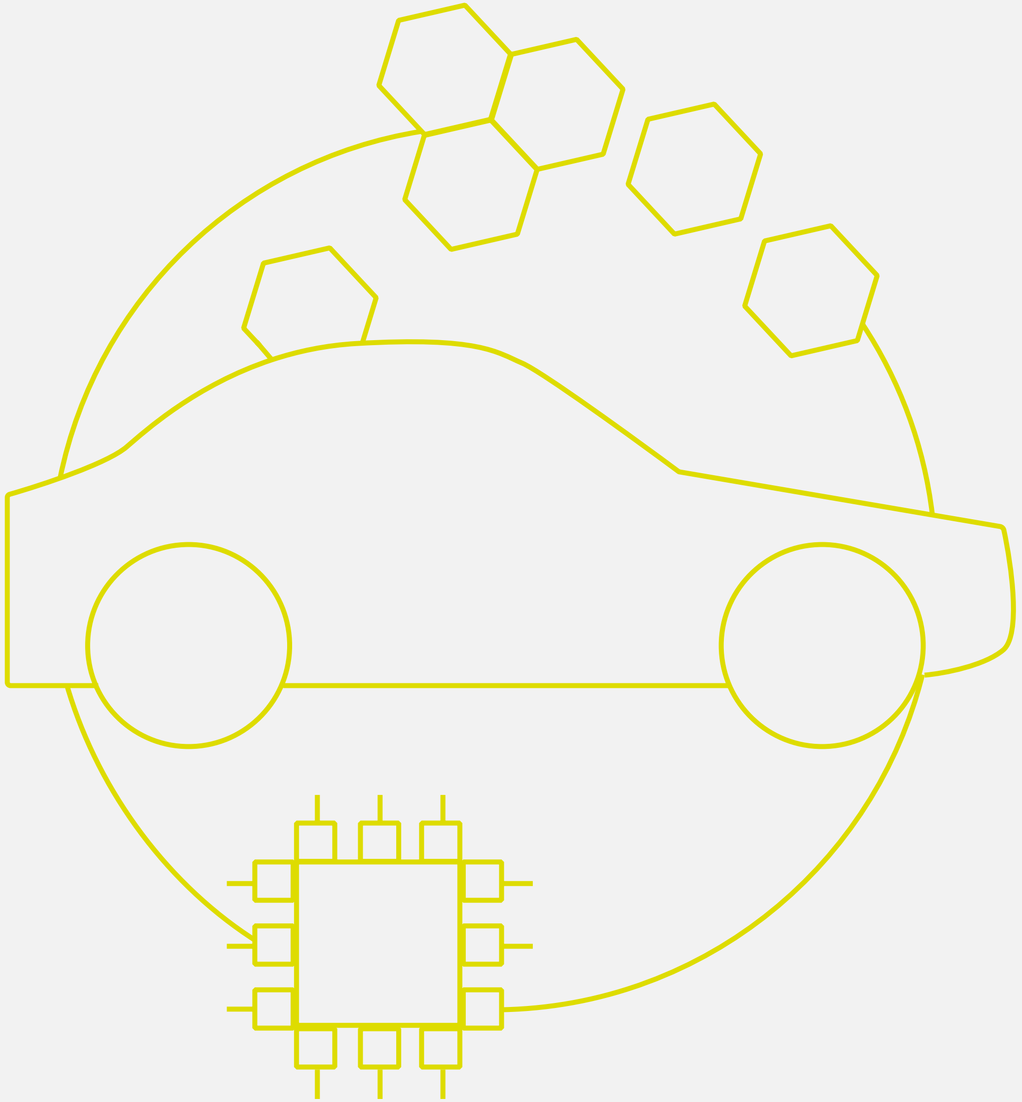


NVIDIA Orin (Primary compute)

R-car S4 (Safety Island)

- Port critical workload to safety island and monitor LF performance

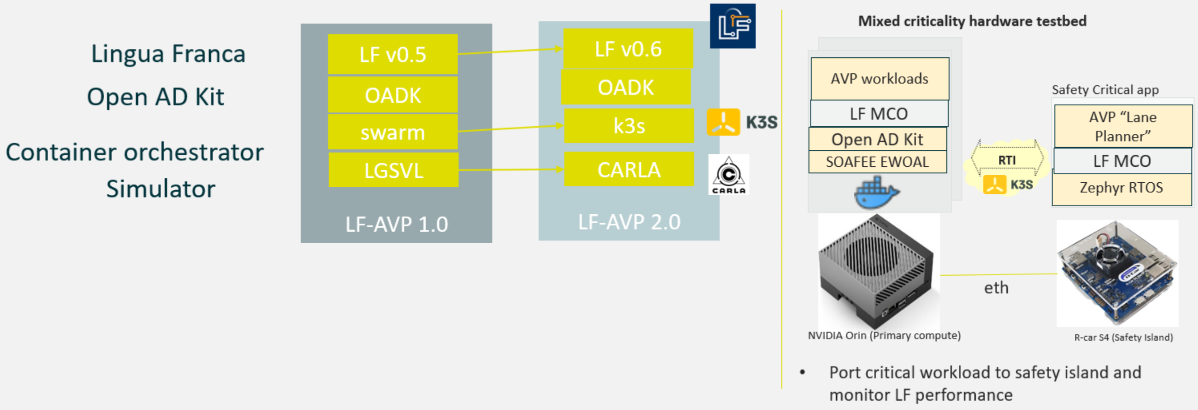
4. Conclusions and Future Works



Conclusions and Future Works

- Conclusion
  - Ecosystem collaboration and standardization are crucial to accelerate towards SDV
  - We demonstrated LF as a mixed critical orchestrator solution on SOAFEE reference architecture using AVP

- Future works
  - AVP v2.0 is ongoing



SOAFEE

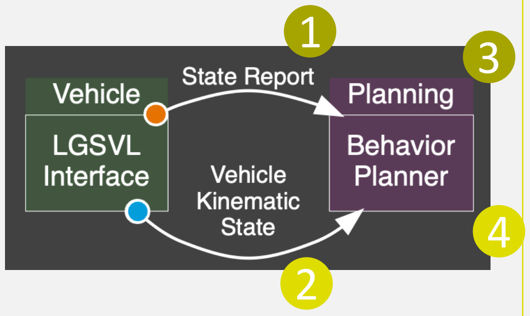
Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
· Ăĕą Ŀ̄
תודה

Non-determinism in AVP demo

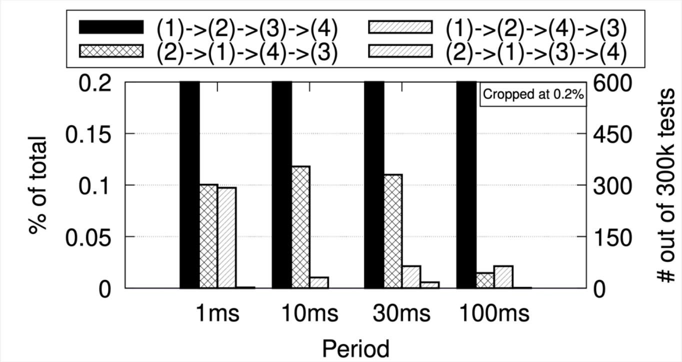
Testability: Given an initial state and a set of inputs to the system, there can only be one correct behavior.

Consistent Global State: An agreement among the software components on the order in which state changes occur in the environment.

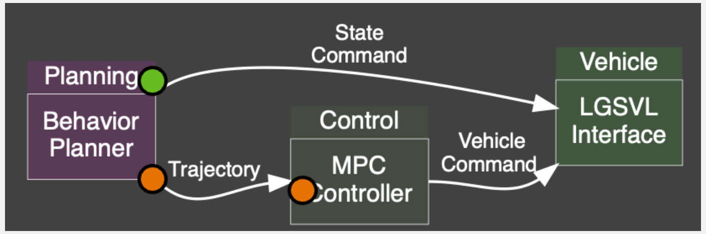
- LGSVL simulator Interface: Intended parking sequence
  - ① Produce a “forward” gear on State Report
  - ② Produce a kinematic state (+ve Velocity)
  - ③ Produce a “reverse” gear on State Report
  - ④ Produce a kinematic state (-ve Velocity)



- Behavior Planner: What will it see due to non-determinism in node-to-node arrival of messages?
  - 1 -> 3 -> 2 -> 4
  - 2 -> 1 -> 3 -> 4
  - 2 -> 4 -> 1 -> 3
  - 1 -> 2 -> 4 -> 3
  - 1 -> 2 -> 3 -> 4
  - 2 -> 1 -> 4 -> 3



- LGSVL interface executes gear change command before corresponding trajectory executed at MPC controller.



State machine-based approach to fix corner cases

Problems:

- Guarantees provided by ROS Point-to-point in-order delivery of messages not sufficient
- Sheer complexity of application

End-end coordination guarantees using LF achieves testability and consistent global state

