OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

自动驾驶系统与架构

System and Infrastructure

Dr. Hongyang Li

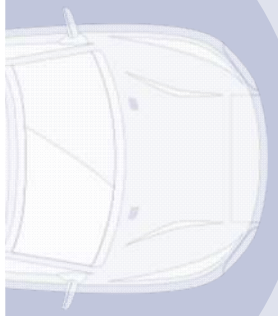
OpenDriveLab / 浦江实验室

Feb 28 2024

@上海交通大学

Slides credit partially from members at OpenDriveLab

Outline



First 45 min

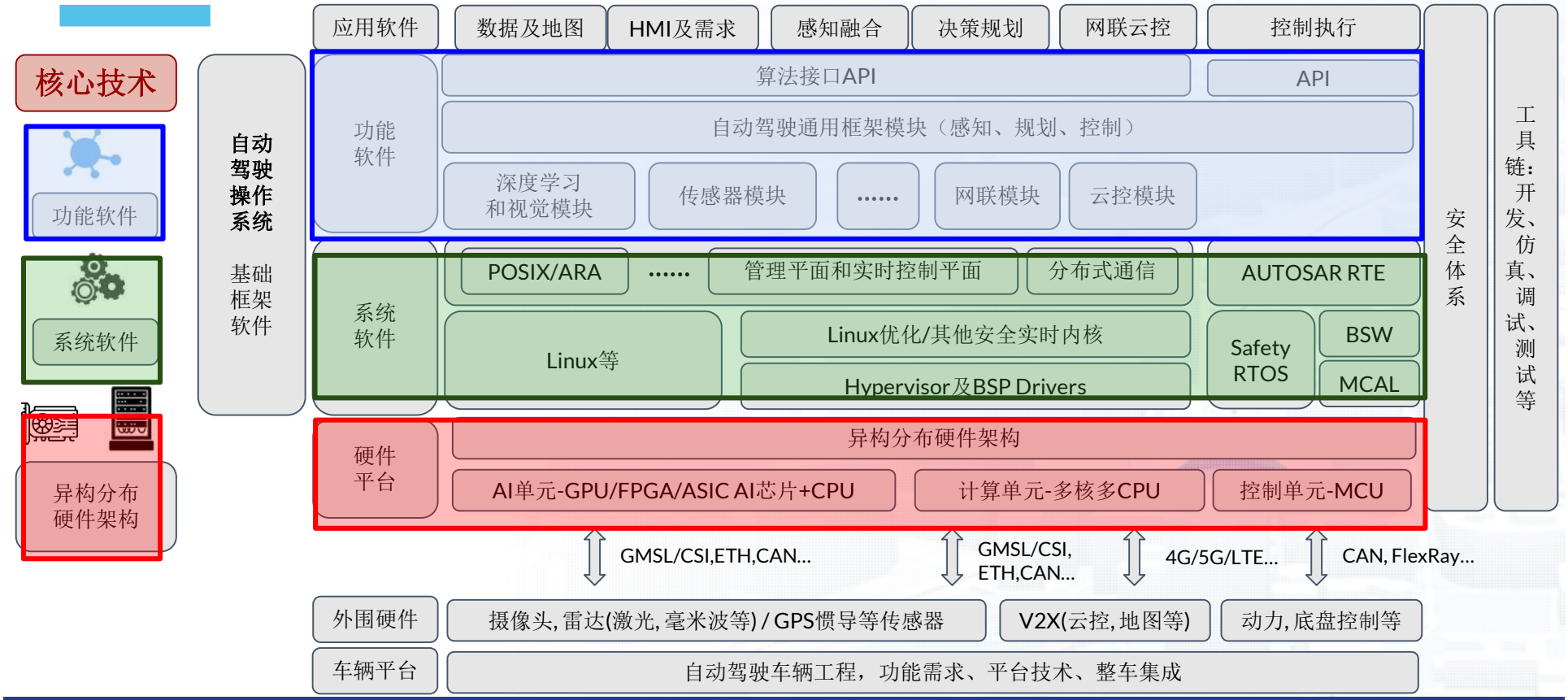
Second 45 min

Third 45 min

- 自动驾驶系统概述 / Infrastructure
- 硬件系统 / Hardware
- 软件系统 / Software
  - 操作系统 / Operating System
  - ***AutoSAR*** 简介 / Middleware
  - 功能软件&应用软件 / Functional Software & Application
- 云服务 / Cloud Service
- ***V2X*** 简介 / V2X Infrastructure

自动驾驶系统

https://www.sohu.com/a/577938734_121124366

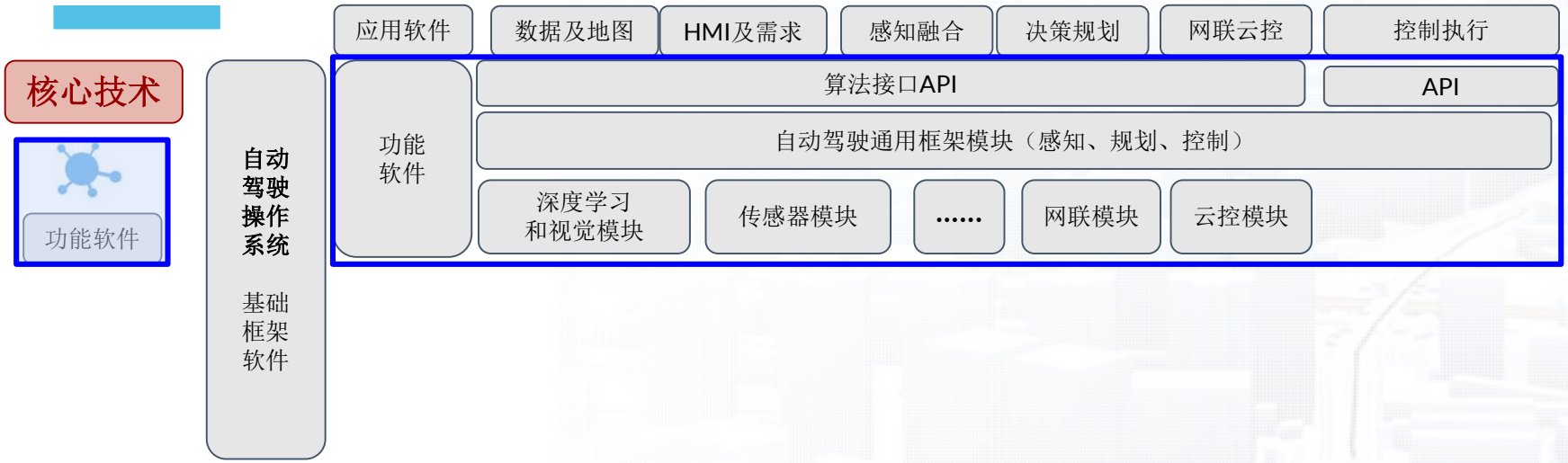


自动驾驶系统 - 核心技术

https://www.sohu.com/a/577938734_121124366

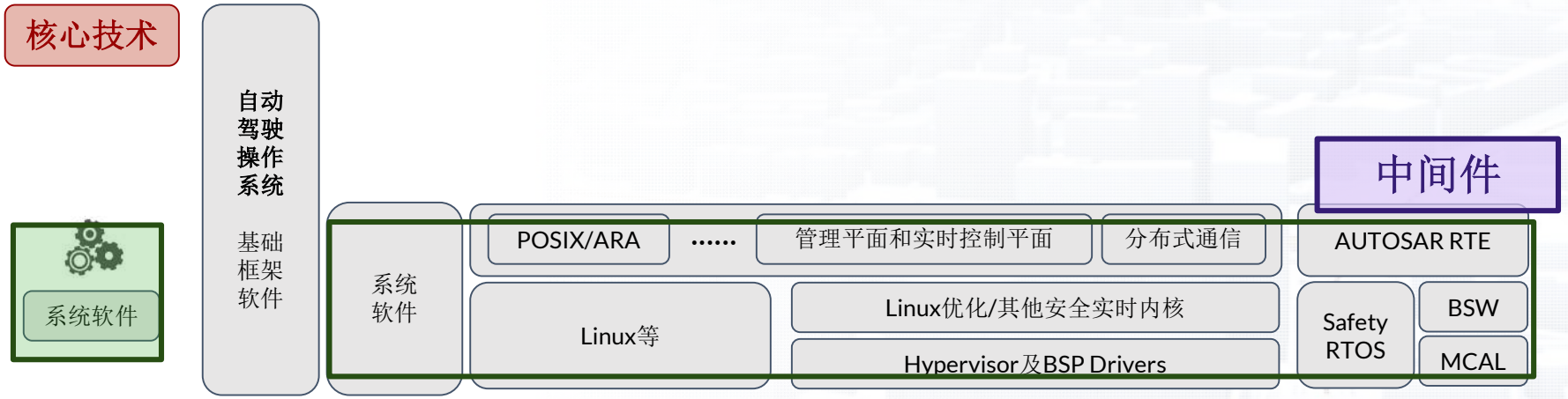
自动驾驶系统 - 功能软件 / 应用软件

https://www.sohu.com/a/577938734_121124366



自动驾驶系统 - 系统软件

https://www.sohu.com/a/577938734_121124366



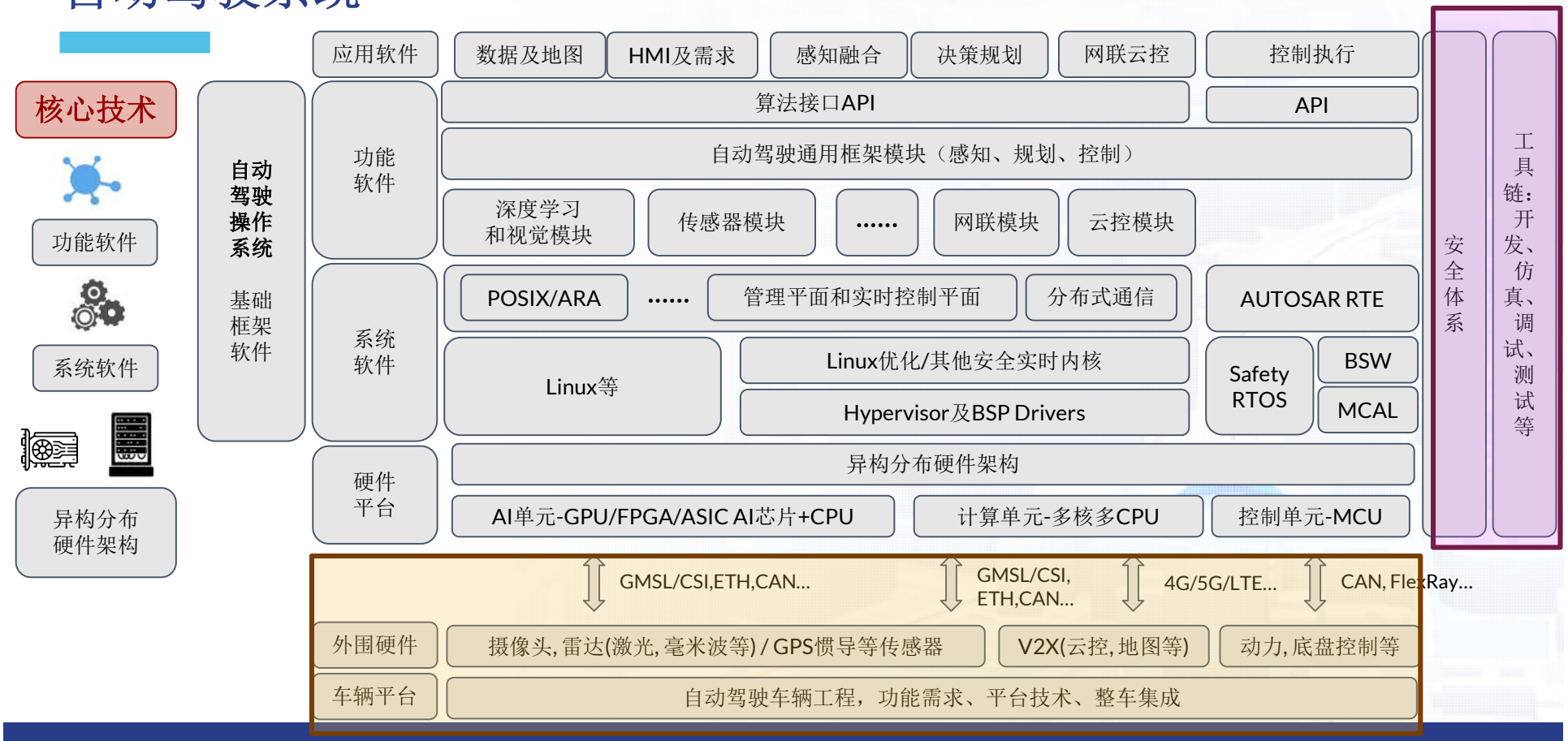
自动驾驶系统 - 硬件平台

https://www.sohu.com/a/577938734_121124366

核心技术

自动驾驶系统

https://www.sohu.com/a/577938734_121124366



自动驾驶系统

自动驾驶汽车软硬件架构中，最底层的是**车辆平台以及外围硬件**（传感器、V2X、动力及底盘控制等）。

在其之上的是**自动驾驶计算平台**，而这也是实现汽车智能化的核心。

进一步来看，自动驾驶计算平台可以分为**硬件平台和软件平台**两大部分。

OpenDriveLab

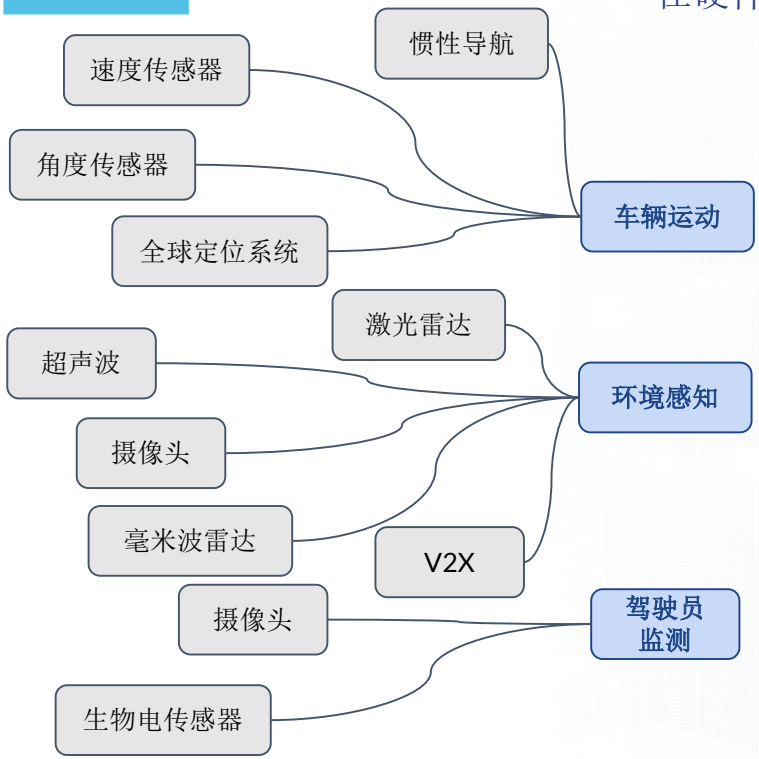
上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Hardware / 硬件

2 System and Infrastructure

自动驾驶硬件系统

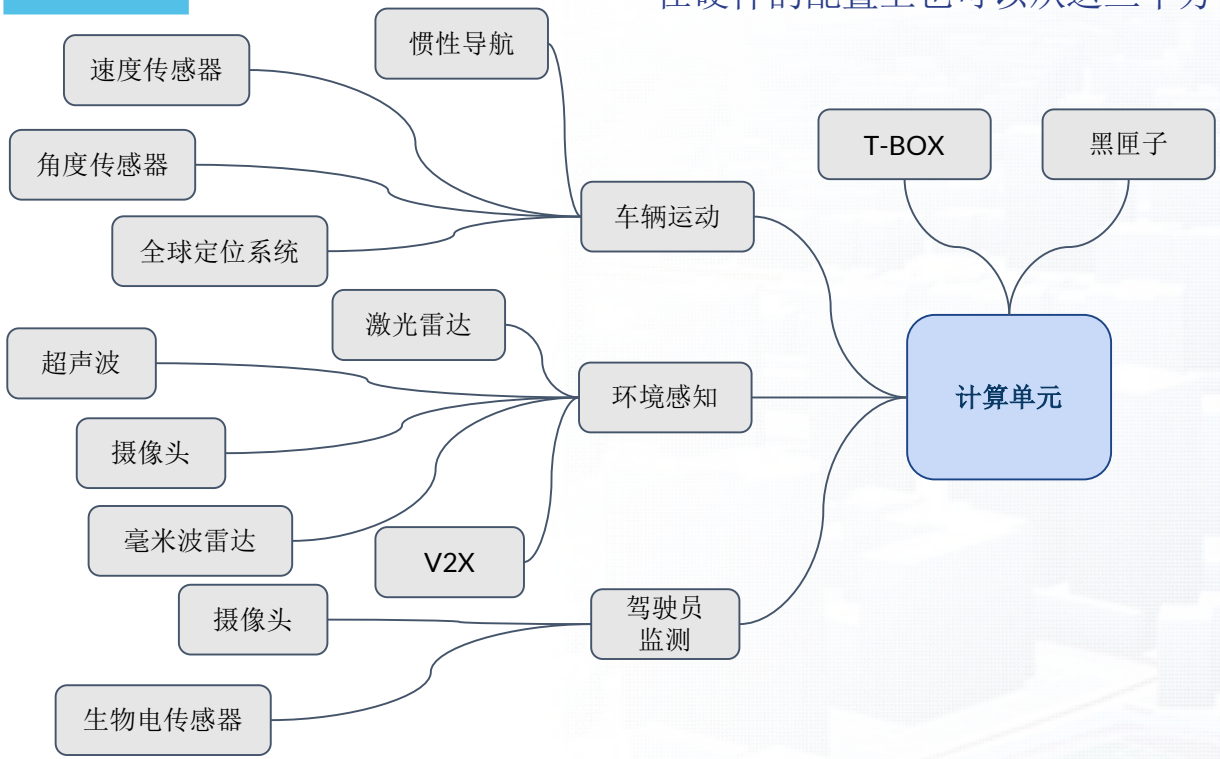
自动驾驶系统主要包含**感知、决策、控制**三个部分，因此在硬件的配置上也可以从这三个方面展开。



感知　　决策　　控制

自动驾驶硬件系统

自动驾驶系统主要包含**感知、决策、控制**三个部分，因此在硬件的配置上也可以从这三个方面展开。

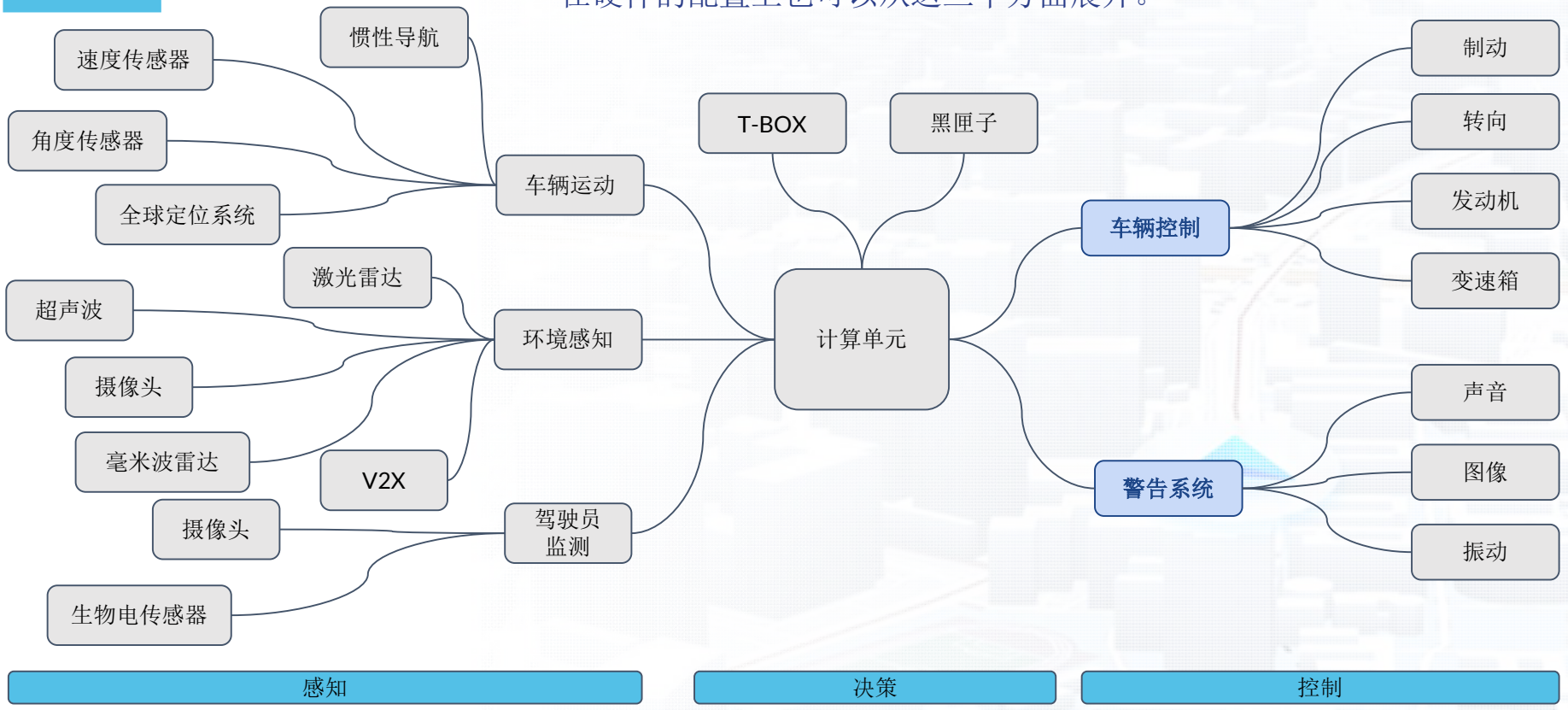


T-Box（Telematics Box）的主要作用是为**车辆提供远程通信接口，实现车辆与用户之间的信息互联互通。**

T-Box通过CAN总线与车辆的主机系统进行通信，并允许用户通过手机APP来获取车辆信息和控制车辆功能，如远程启动车辆、远程关闭窗户和空调等。此外，T-Box还能提供车辆故障监测、驾驶行为分析、驾驶数据采集、胎压状态、车门状态等信息，以及实现车辆遥控、安全报警、紧急救援等功能。T-Box作为车联网系统的重要组成部分，能够通过移动4G通信网络，完成卫星定位系统、车辆数据的实时采集、车辆行驶路线的记录、车辆相关故障的监控，以及通过网络对车辆进行远程控制等功能。

自动驾驶硬件系统

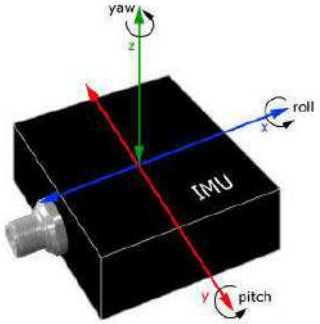
自动驾驶系统主要包含**感知、决策、控制**三个部分，因此在硬件的配置上也可以从这三个方面展开。



自动驾驶硬件系统

感知层配置了安装大量的传感器，包括车辆运动传感器、环境感知传感器和驾驶员监测传感器。

- **车辆运动传感器**包括惯性导航（IMU）、速度传感器、角度传感器和全球定位系统。
- **环境感知传感器**包括激光雷达(LiDAR)、毫米波雷达(Radar)、摄像头(Camera)和超声波雷达。
- **驾驶员检测传感器**包括摄像头和生物电传感器，用于在危险状况下提醒驾驶员接管车辆。



惯性导航IMU

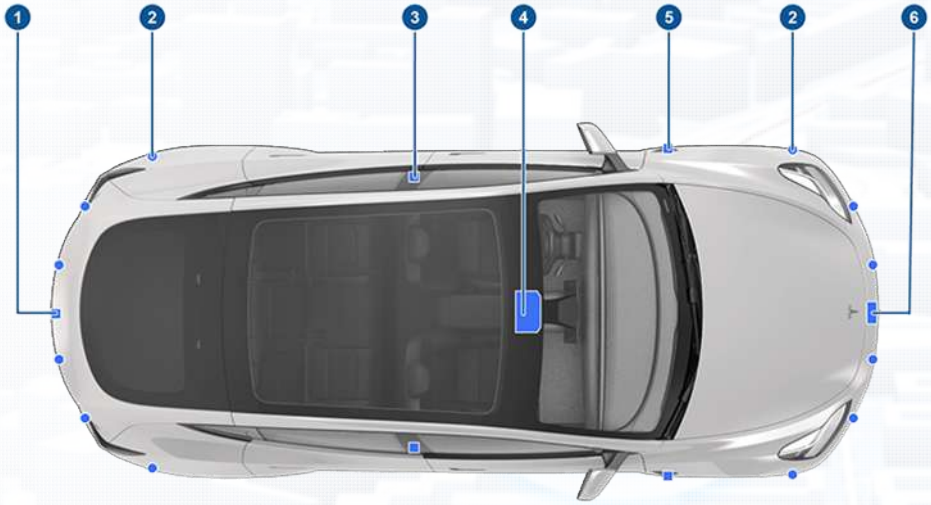
大疆 Livox 觅道 Mid-360 激光雷达

华为 问界M5 侧视摄像头

自动驾驶硬件系统 - 特斯拉

以特斯拉Model Y车型为例，其配备：

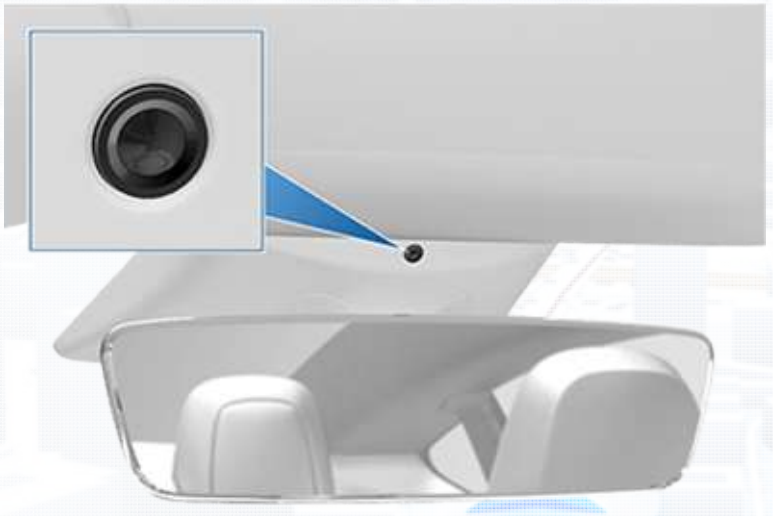
1. 后车牌的上方装有一个摄像头。
2. 超声波传感器位于前后保险杠中。
3. 各门柱均装有一个摄像头。
4. 后视镜上方的挡风玻璃上装有三个摄像头。
5. 每块前翼子板上装有一个摄像头。
6. 雷达安装在前保险杠后侧。



https://www.tesla.com/ownersmanual/modely/zh_cn/GUID-682FF4A7-D083-4C95-925A-5EE3752F4865.html

自动驾驶硬件系统 - 特斯拉

Tesla 车型也在后视镜上方装配一个驾驶室摄像头，用于在驾驶员使用Autopilot辅助驾驶系统时进行监控，可以分析驾驶员视线方向，判断驾驶员有没有使用手机，有没有带墨镜等

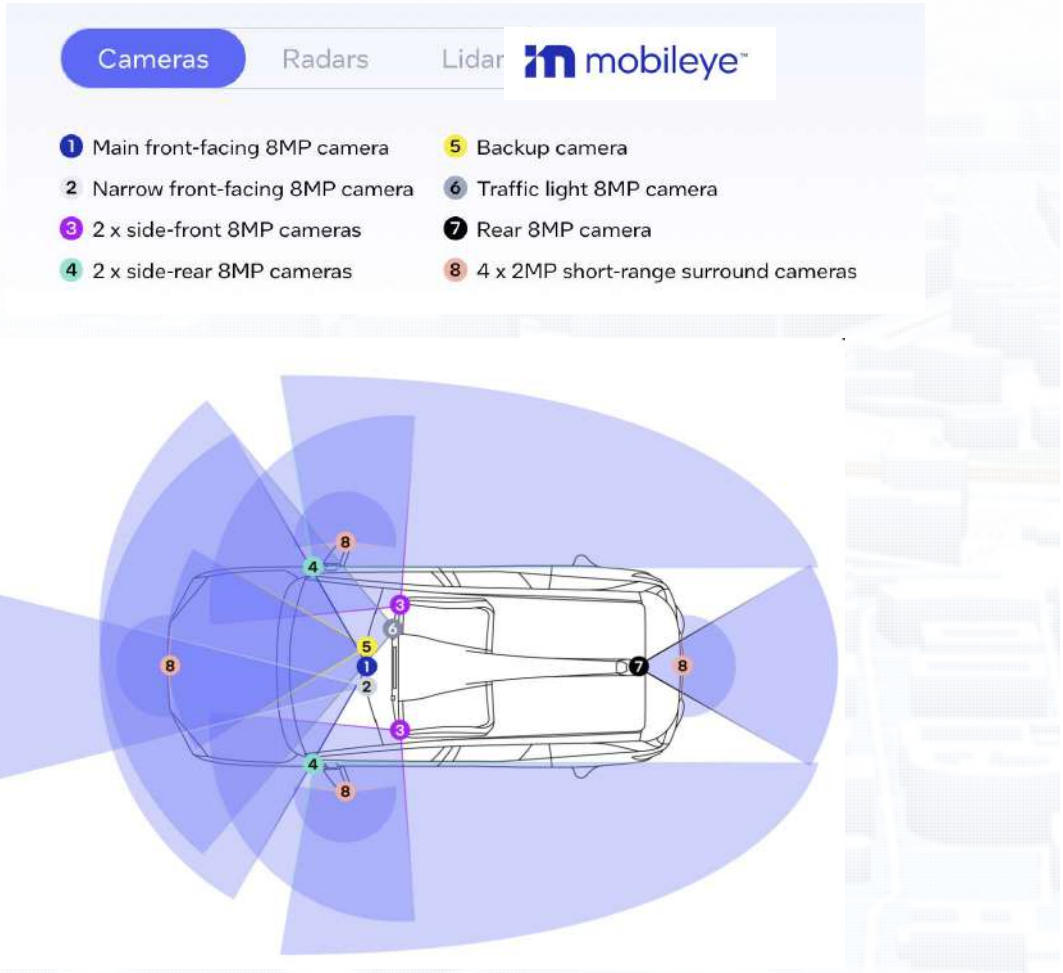


https://www.tesla.com/ownersmanual/modely/zh_cn/GUID-682FF4A7-D083-4C95-925A-5EE3752F4865.html

https://www.mobileye.com/ces-2024/

自动驾驶硬件系统 - *Camera*

Mobileye Drive™ is our turnkey self-driving system, designed to turn almost any service or vehicle autonomous. It is already being integrated across industries and around the globe into autonomous public transit, autonomous goods delivery, and a full Robotaxi offering.



https://www.mobileye.com/ces-2024/

自动驾驶硬件系统 - *Camera*

影响自动驾驶相机性能的有几个关键的因素：

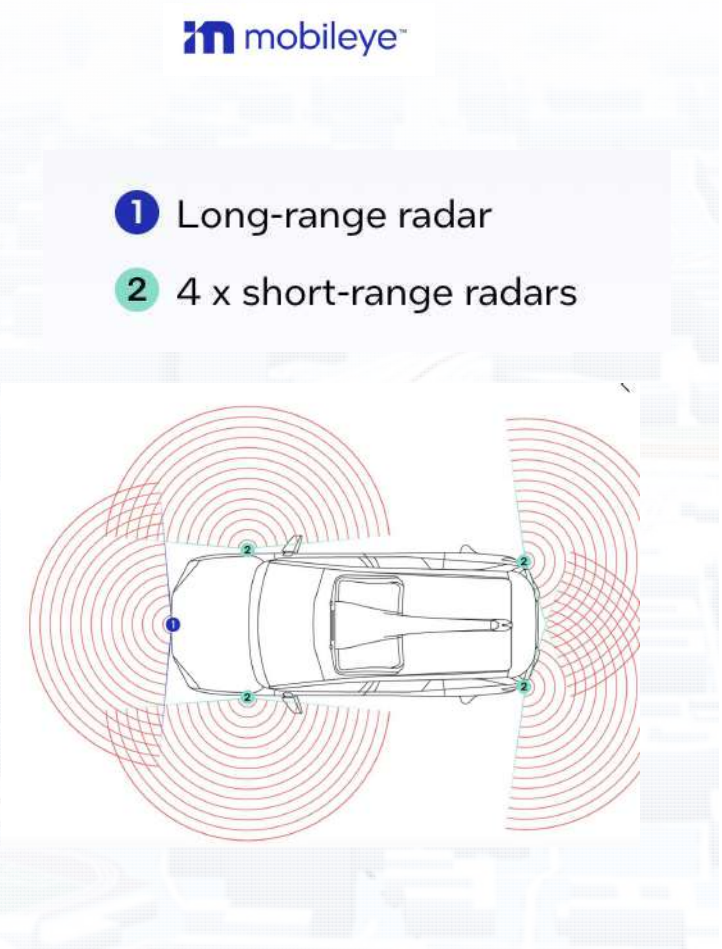
1. **镜头** 决定了相机的***FOV***（视场），常见的FOV有30度、60度、70度、120度、鱼眼等。
2. **图像传感器** 将光信号转化为电信号，现在主流的是***CMOS***工艺。索尼、OmniVision是主流的CMOS提供商，常说的IMX-290、IMX-390等是CMOS的型号。
3. **畸变参数** 分为径向畸变和切向畸变，分别是由镜头不完美、安装不精确引入的。需要通过标定计算相机内参去畸变。
4. **图像分辨率** 常用的有1080P、2K、4K。对于远距离物体（大于600米），4K的相机是很有必要的，例如600米外的一个2米宽的物体对于一个FOV30的相机图像上只有五六个像素。

https://www.mobileye.com/ces-2024/

自动驾驶硬件系统 - *Radar*

Rada全称Radio Detection and Ranging。自动驾驶车辆一般使用的是工作在**毫米波波段**（millimeter wave）探测的雷达。工作频段一般为30GHz ~ 300 GHz， 波长1~10mm，介于微波和厘米波之间，兼具有微波雷达和光电雷达的一些优点。

- 相比传统厘米波雷达具有体积小、易集成和空间分辨率高的特点
- **鲁棒性强，**不良天气环境下，雨、雾、灰尘等对毫米波雷达干扰较小

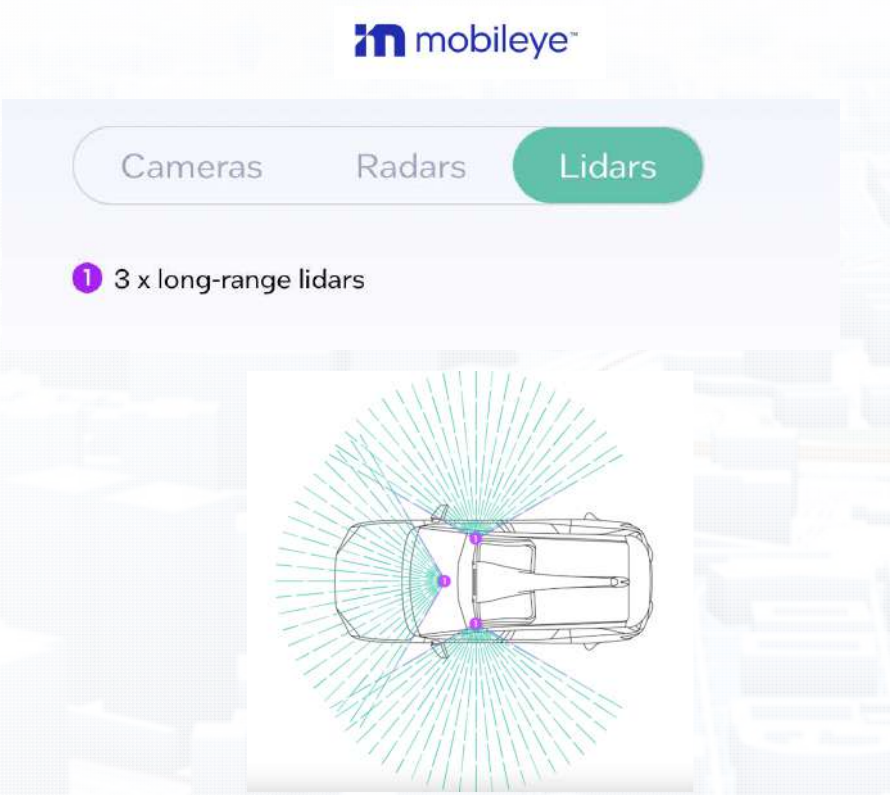


https://www.mobileye.com/ces-2024/

自动驾驶硬件系统 - *LiDAR*

LiDAR全称Light Detection and Ranging。激光雷达通过各个方向发射激光脉冲，波长在 **纳米** 范围，这些脉冲在抵达物体表面之后反射回来，并被接收器接收。因其波速接近于光速，只需记录激光的首发时间，通过简单的公式就能进行测距。

- LiDAR波长较短，相比Radar更加精准，可以检测**更小的**物体
- 同时由于波长较短，很容易收到空气介质中的各种杂质干扰，**恶劣天气下效果影响很大**

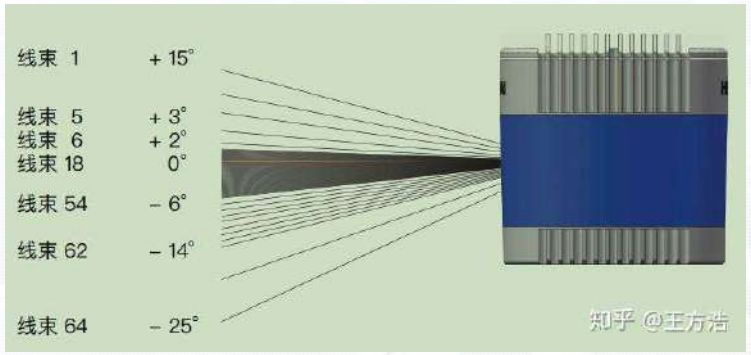


自动驾驶硬件系统 - *LiDAR*

LiDAR在实际应用中，有两个常见的参数：

- **垂直方向上**，激光收发模块的数量被称为线数。**线数越高，扫描结果越精确。**常用的线数有16线，32线，64线。
- **水平方向上**，由于激光雷达在旋转扫描，扫描的点数和激光雷达的扫描频率有一定的关系，这个参数一般被称为水平分辨率。列如水平分辨率为0.2°，那么扫描的点数为360°/0.2°=1800，即水平方向会扫描1800次

禾赛64线激光雷达光束分布



Credit to FangHao Wang @ Zhihu

	Velodyne					Hesai		Ouster		RoboSense
	VLS-128*	HDL-64S2	HDL-32E	VLP-32c	VLP-16	Pandar64	Pandar40p	OS1-64	OS1-16	RS-LiDAR-32
Channels	128	64	32	32	16	64	40	64	16	32
FPS[Hz]	5-20	5-20	5-20	5-20	5-20	10,20	10,20	10,20	10,20	5,10,20
Precision[m]	±0.03	±0.02[a]	±0.02	±0.03	±0.03	±0.02[c]	±0.02[c]	±0.03[d]	±0.03[d]	±0.03[c]
Max.Range[m]	300	120	100	200	100	200	200	120	120	200
Min.Range[m]	N/A	3	2	1	1	0.3	0.3	0.8	0.8	0.4
vFOV[deg]	40	26.9	41.33	40	30	40	40	33.2	33.2	40
vRes[deg]	0.11[b]	0.33[b]	1.33	0.33[b]	2.0	0.167[b]	0.33[b]	0.53	0.53	0.33[b]
hRes[deg]10hz	0.2	0.16	0.16	0.2	0.2	0.2	0.2	0.35	0.35	0.2
λ[nm]	903	903	903	903	903	905	905	850	850	905
d[mm]	165.5	223.5	85.3	103	103.3	116	116	85	85	114
Weight(kg)	3.5	13.5	1.0	0.925	0.830	1.52	1.52	0.425	0.425	1.17
Firmware Ver.	—[e]	4.07	2.1.7.1	N/A	3.0.29.0	5.10	4.29	—[f]	1.12.0	1.12.0
Price[g][USD]	$$$$	$$$	$$	$$	$	$$$	$$$	$	$	$$

10款主流的旋转式机械激光雷达在自驾场景中的对比

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9142208

自动驾驶硬件系统

计算单元

是自动驾驶车辆的大脑，所有数据都要通过计算单元运算处理，然后做出决策，下发执行。为了保证自动驾驶的实时性要求，**软件响应最大延迟必须在可接受的范围内**，这对计算的要求非常高。

目前主流的解决方案有基于GPU、FPGA、ASIC等。

自动驾驶硬件系统

计算单元

是自动驾驶车辆的大脑，所有数据都要通过计算单元运算处理，然后做出决策，下发执行。为了保证自动驾驶的实时性要求，软件响应最大延迟必须在可接受的范围内，这对计算的要求非常高。

目前主流的解决方案有基于GPU、FPGA、ASIC等。

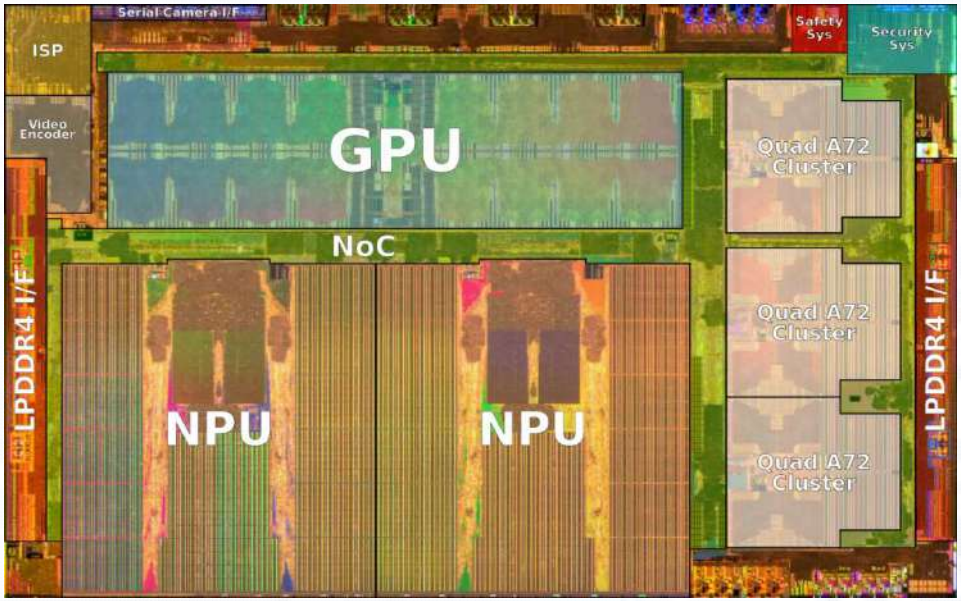
FPGA：现场可编程门阵列，它是作为专用集成电路领域中的一种半定制电路而出现的，既解决了定制电路的不足，又克服了原有可编程器件门电路数有限的缺点。

ASIC：一种为专门目的而设计的集成电路。是指应特定用户要求和特定电子系统的需要而设计、制造的集成电路。ASIC 的特点是面向特定用户的需求，在批量生产时与通用集成电路相比具有体积更小、功耗更低、可靠性提高、性能提高、保密性增强、成本降低等优点。

自动驾驶硬件系统

“CPU+GPU+ASIC”方案的主要代表有：英伟达、特斯拉 FSD 以及高通 Ride。

特斯拉 FSD 芯片以 NPU（ASIC）为计算核心，有三个主要模块：CPU、GPU 和 Neural Processing Unit（NPU）。

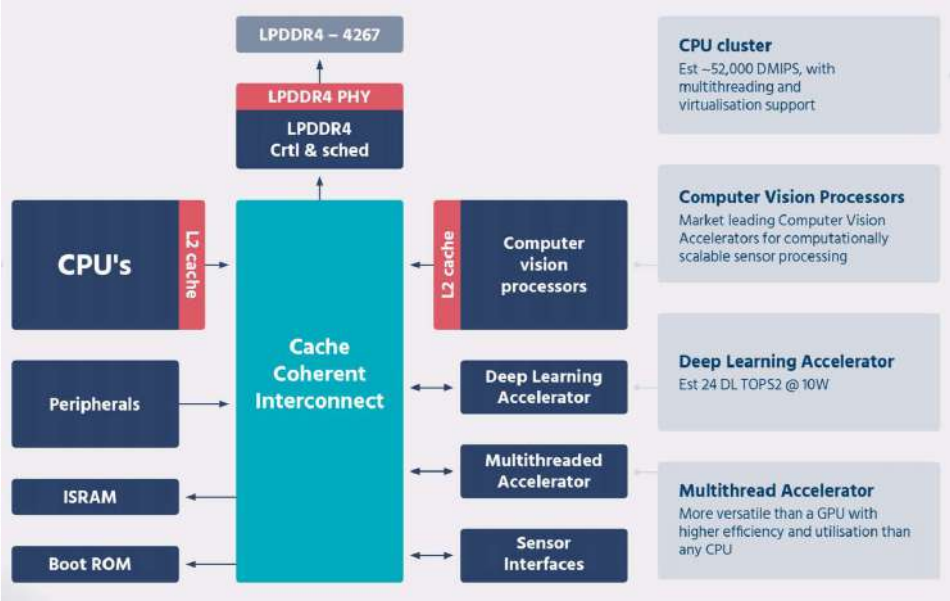


https://en.wikichip.org/w/images/6/6d/tesla_fsd_die_%28annotated%29.png

自动驾驶硬件系统

“CPU+ASIC”方案的主要代表有：***Mobileye*** EyeQ5 系列 和 **地平线**征程系列。

Mobieye EyeQ5主要有 4个模块：CPU、Computer Vision Processors（CVP）、Deep Learning Accelerator（DLA）和 Multithreaded Accelerator（MA），其中 CVP 是针对传统计算机视觉算法设计的 ASIC。

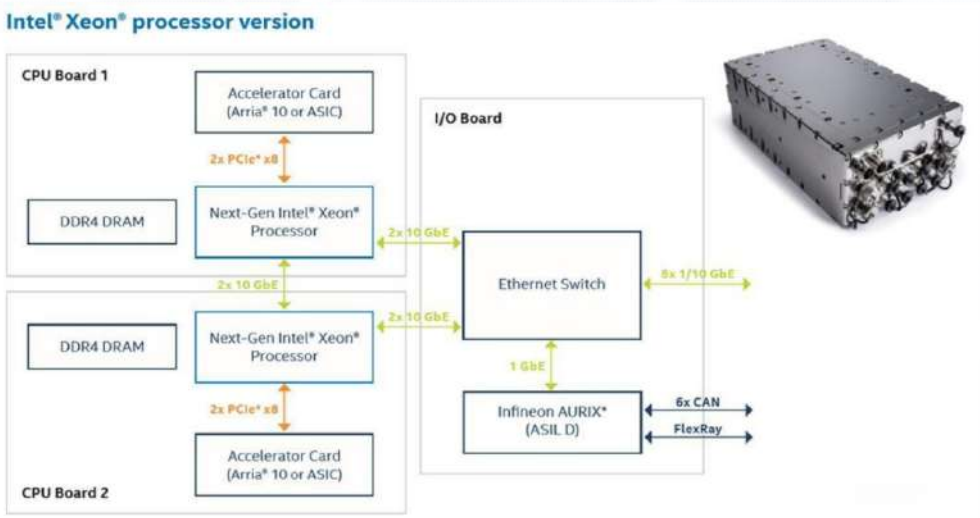


https://s21.q4cdn.com/600692695/files/doc_presentations/2019/01/Mobileye_CES2019.pdf

自动驾驶硬件系统

“CPU+FPGA”的主要代表是 Waymo。

与其余厂商不同，***Waymo*** 采用“CPU+FPGA”的架构，其计算平台采用英特尔 Xeon12 核以上 CPU，搭配Altera 的 Arria 系列 FPGA。



http://news.eeworld.com.cn/qcdz/ic618781.html

自动驾驶硬件系统

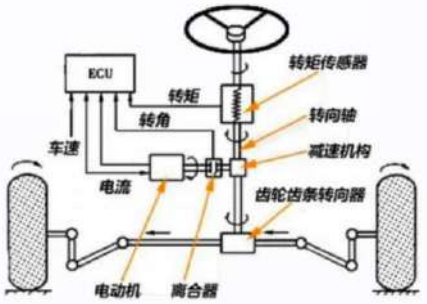
车辆控制系统是控制部分的重要组成，自动驾驶需要用**电信号**控制车辆的转向、制动、油门系统，其中涉及到车辆低盘的线控改装，目前在具备自适应巡航、紧急制动、自动泊车功能的车上可以直接借用原车的系统，通过CAN总线控制而不需要过度改装。

警告系统主要是通过**声音、图像、振动**提醒司机注意，通过HMI的设计有效减少司机困倦、分心的行为。

线控系统

线控 就是Control by Wire的直译 。简单理解，就是车辆的控制都是由一系列命令而执行的，而不是物理的操作进行执行的。

电子助力转向（*EPS*）与线控转向最大的区别在于，EPS方向盘与车轮之间链接并未参与线控技术，依然采用的机械链接。从电信号控制角度看EPS也可以看成是一种线控转向系统。



EPS

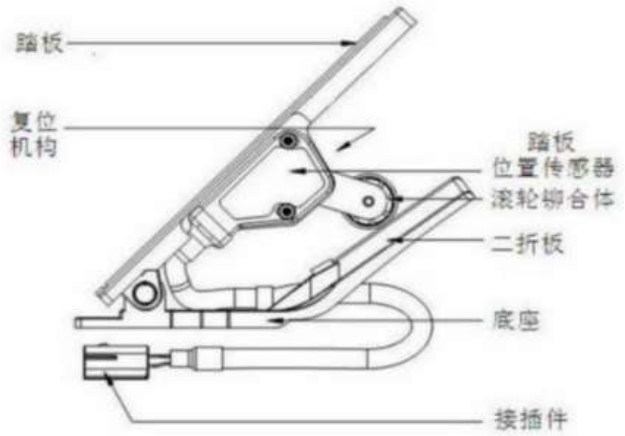
线控转向

线控系统

线控油门就是电子油门，通过位置传感器传送油门踩踏深浅与快慢的讯号，从而实现油门功能的电子控制装置。

电子油门目前已大量普及，凡具备定速巡航即可认定有电子油门，早期电子油门为接触式，近来已经改为**非接触式**。电车依靠电机扭矩实现，直接发扭矩信号即可，油车依靠发动机管理系统（EMS）发扭矩信号实现。

线控油门构成



线控系统

转向 的最早改装是在转向管柱端截断加装转向电机进行改造。之后利用原车转向助力系统进行转向控制。

制动 的最早改装是加装电机踏板，后续利用原车的ESC系统进行控制，未来会选用MK C1之类的线控控制系统。

加速 的最早改装都是发扭矩信号依靠EMS实现，后续的改装方案都是借用原车ACC接口由电子油门来执行。

减速控制　　转向控制　　加速控制

OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Software

2 System and Infrastructure

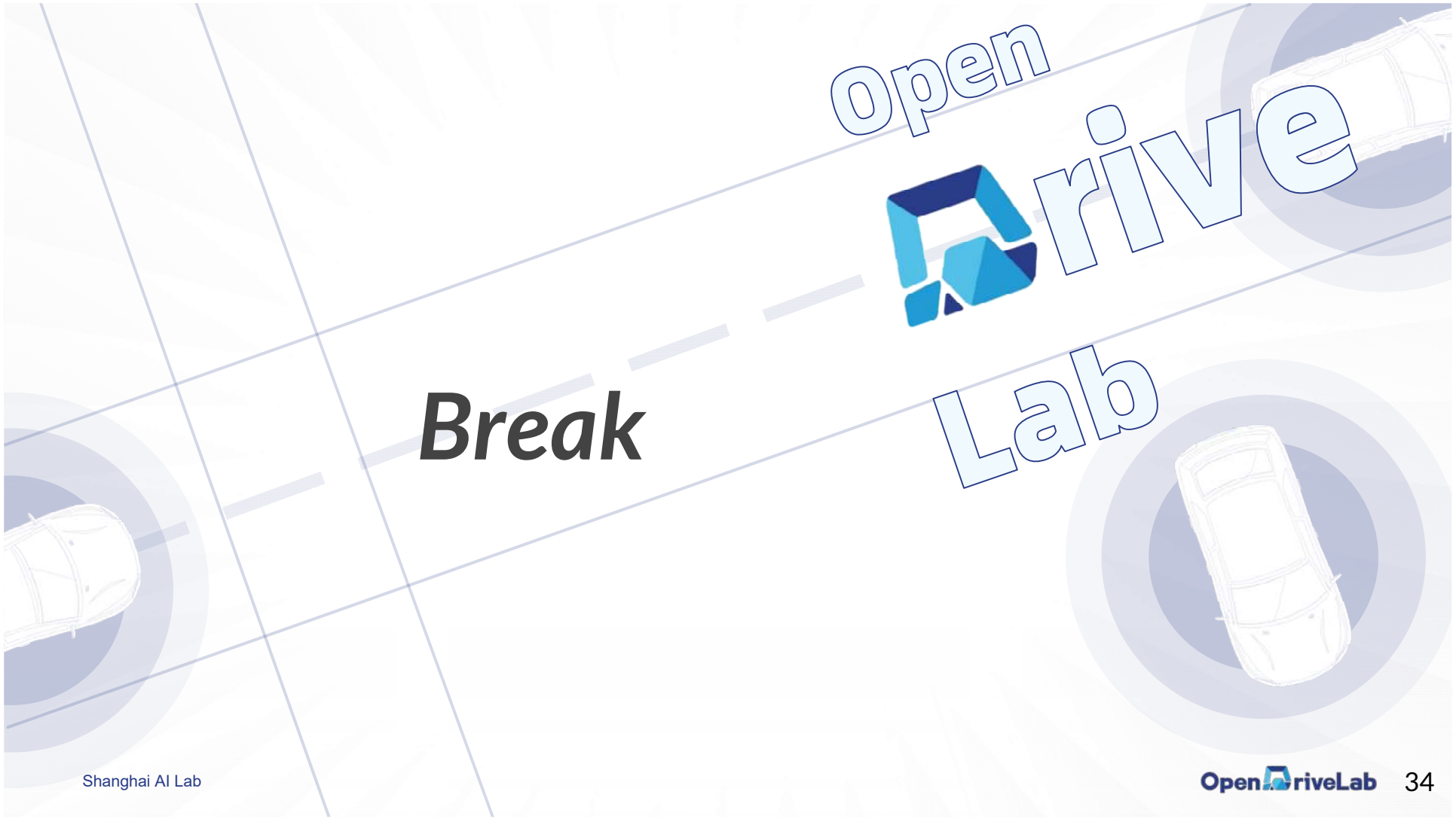
软件架构

自动驾驶计算平台中的软件系统是智能化的直接体现，核心功能和算法的定制化和差异化是各大品牌发力的重点，其中涉及多个软件层面，自下而上分别为：

- 系统软件
- 功能软件
- 应用软件

软件架构

- **系统软件：**由硬件抽象层、OS 内核（狭义上的操作系统）和中间件组件构成，是广义操作系统的核心部分。
- **功能软件：**主要为自动驾驶的核心共性功能模块，包括自动驾驶通用框架、AI 和视觉模块、传感器模块等库组件。**系统软件与功能软件构成了广义上的操作系统。**
- **应用软件：**主要包括场景算法和应用，是智能座舱（HMI、应用软件等）以及自动驾驶（感知融合、决策规划、控制执行等）形成差异化的核心。



Break

OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Operating System

2 System and Infrastructure

自动驾驶操作系统

自动驾驶车辆的软件算法都运行在操作系统之上，**对操作系统最主要的要求是稳定性 和 实时性**。

- **稳定性**体现在操作系统占用的资源少，出现故障之后系统不会奔溃，能够长时间运行 。

- **实时性**要求系统能够及时响应控制指令，**工业设备、汽车电子、航空航天**等领域都要求采用实时操作系统，因为在这些领域操作系统如果不能及时响应控制指令，会产生很严重的后果。

自动驾驶操作系统

主流车载操作系统具有不同的特点和应用场景

操作系统	简介	优势	劣势	合作主机厂/零部件供应商
QNX	属于黑莓公司，是全球第一款通过ISO 26262 ASIL level D 认证的车载操作系统	安全性，稳定性极高，符合车规级要求，可用于仪表盘	需要授权费用，只应用在高端车型上	通用、克莱斯勒、凯迪拉克、雪佛兰、雷克萨斯、路虎、保时捷、奥迪、大众、别克、丰田、捷豹、宝马、现代、福特、日产、奔驰、哈曼、伟世通、大陆、博世等
Linux	基于POSIX和UNIX的多用户、多任务、支持多线程和多CPU的操作系统	免费、灵活性、安全性高	应用生态不完整，技术支持差	丰田、日产、特斯拉等
Android	谷歌开发的基于Linux架构的系统，属于“类Linux系统”	开源，易于OEM自研，移动终端生态完善	安全性、稳定性较差，无法适配仪表盘等安全要求高的部件	奥迪、通用、蔚来、小鹏、吉利、比亚迪、博泰、英伟达等
WinCE	微软发布的32位的多任务嵌入式操作系统，具有多任务抢占、硬实时等特点	在当时实时性出色，Windows应用开发便利	现在开发者和应用以及非常少，即将推出历史舞台	福特 Sync1、Sync2等

自动驾驶操作系统

各OS内核比较

	QNX	Linux	Vxworks
开放性	半封闭	源代码开放	源代码开放
是否可裁剪	否	是	是
软件生态丰富程度	较丰富；商业公司提供，或自己从开源软件移植	非常丰富，主要来自开源软件社区	在自动驾驶产业影响较小
实时性	微秒级	毫秒级（打开CONFIG_PREEMPT_RT后为微秒级）	微秒级
功能安全等级	ASIL-D	无	ASIL-D
开发工具和使用费用	昂贵	免费	昂贵
易用性	容易	最难	比较难

自动驾驶操作系统

以Apollo 5.0为例，采用的操作系统是Linux操作系统，而Linux不是实时操作系统，需要打上系统补丁之后，才能成为实时操作系统。

那么如何保证系统任务的实时调度呢？

接下来我们先看Linux采取的调度策略。

*Linux*进程调度

- ***Linux*内核分为抢占式内核和非抢占式内核。**抢占式内核的实时性更好。
- CPU给任务划分时间片，通过时间片轮转，使CPU看起来在同一时间执行多个任务。
- **内核把进程做了区分，分为交互型和脚本型。**Linux通过抢占式的方式，对任务的优先级进行排序，交互型进程的优先级要比脚本型进程的优先级更高。还有一类进程是实时进程，这类进程的优先级最高，实时进程必须要保证立刻执行，因此会有限抢占其它进程。

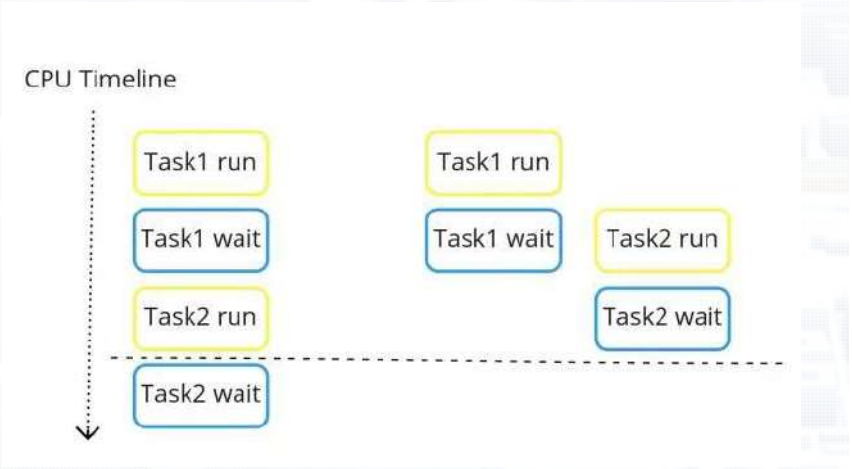
*Linux*进程调度

如果单纯根据优先级，低优先级的任务可能很长一段时间都得不到执行，因此需要更加公平的算法，在一个进程等待时间过长的时候，会动态的提高它的优先级。当一个进程已经执行很长一段时间了，会动态降低它的优先级。这样带来的好处是，不会导致低优先级的进程长期得不到CPU，而高优先级的进程长期霸占CPU。Linux采用CFS算法来保证进程能够相对公平的占用CPU。

*Linux*进程调度

左图展示了Linux的进程调度采用的***CFS (Completely Fair Scheduler)*算法**。

如图所示，单个CPU核心的情况下，左侧是没有进程调度的情况，任务1在执行完成之后，会读取IO（内存、硬盘等）数据，这时候CPU会进入等待状态，CPU在等待的过程中没有做任何事情。而右边采用了调度策略，在CPU等待的过程中，任务1主动让出CPU，下一个任务就可以在当前任务等待IO的过程中执行。



无人驾驶进程调度 Case Study

参考Linux的进程调度，我们思考下如何进行无人驾驶进程调度。

假设无人驾驶系统有以下几个进程：**定位、感知、规划、控制、传感器、日志和地图**，而CPU只有2个核心，那么应该如何规划这些任务的优先级呢？

无人驾驶进程调度 Case Study

- 首先应该假设**定位、感知、规划、控制和传感器读取的优先级比日志和地图更高**。
- 接下来对于优先级高的模块，因为目前只有2个CPU核心，所以不可能同时执行上述所有模块，只能通过**时间片轮转**来实现。

无人驾驶进程调度 Case Study

- **无人驾驶对模块的算法复杂度也有要求。**

  如果感知模块采用了复杂度较高的算法来提高准确率，导致的结果是感知模块会占用更多的CPU时间，其它模块需要和感知模块竞争CPU，系统总的执行时间会变长。

  解决的办法有2种：

  一种是升级硬件，增加*CPU*的核数；

  另外一种是降低系统算法复杂度，每个模块尽可能的高效，占用较少的系统时间。

- **系统的算法复杂度还要尽可能的稳定，**不能波动很大，或者执行超时。

  如果各个模块的算法都不太稳定，当遇到极端情况，每个模块需要的时间都会变长，系统的负载会一下子突然变高，导致CPU的响应不及时，出现很致命的问题。

无人驾驶进程调度 Case Study

- **除此之外，无人驾驶车还需要考虑极端情况下**，系统的进程会奔溃或者一直占用CPU的情况。

  1. **找不到最优解，死循环**。大部分情况下程序没有响应是因为找不到最优解，或者死循环，这种情况可以通过代码保证。

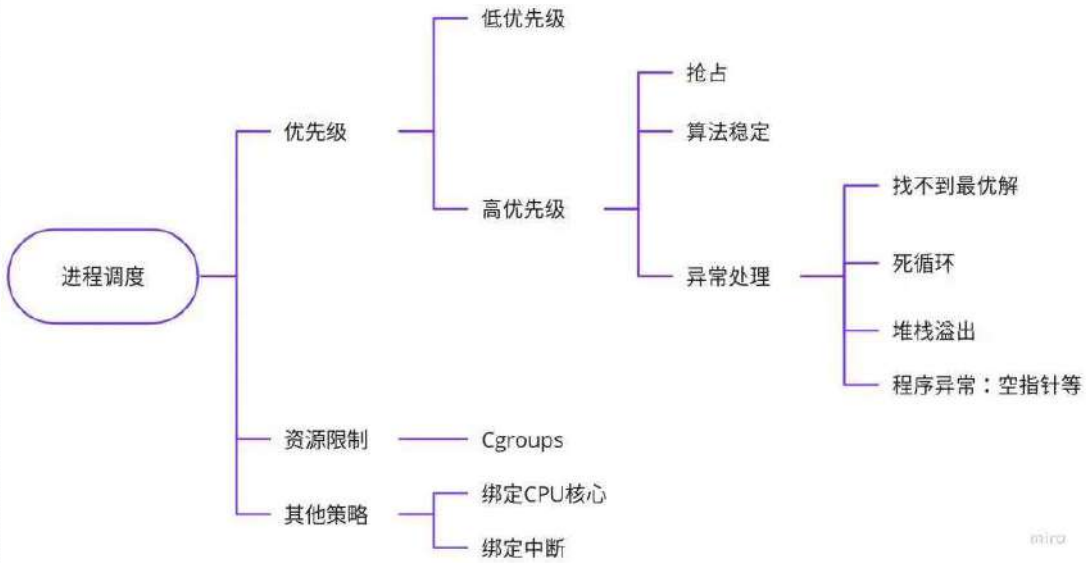
  2. **堆栈溢出、内存泄露、空指针**。这种情况属于程序编码错误，也可以通过代码保证。

  3. **硬件错误**。极小概率的情况下，CPU的寄存器会出错，嵌入式的CPU会有冗余校正，而家用和服务器级别的CPU没有这种设计，这种情况下只能重启进程，或者重启硬件。

无人驾驶进程调度 Case Study

根据上述的思路，可以得到无人驾驶的进程调度策略，如图所示。

把控制模块的优先级设置到最高，规划模块其次，感知和定位模块的优先级设置相对较低。因为控制和规划模块必须马上处理，感知和定位模块如果当前帧处理不过来，大不了就丢弃，接着处理下一帧。当然这些进程都需要设置为实时进程。而地图、日志等模块的优先级设置为最低，在其它高优先级的进程到来之时会被抢占。



OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Middleware / 中间件

2 System and Infrastructure

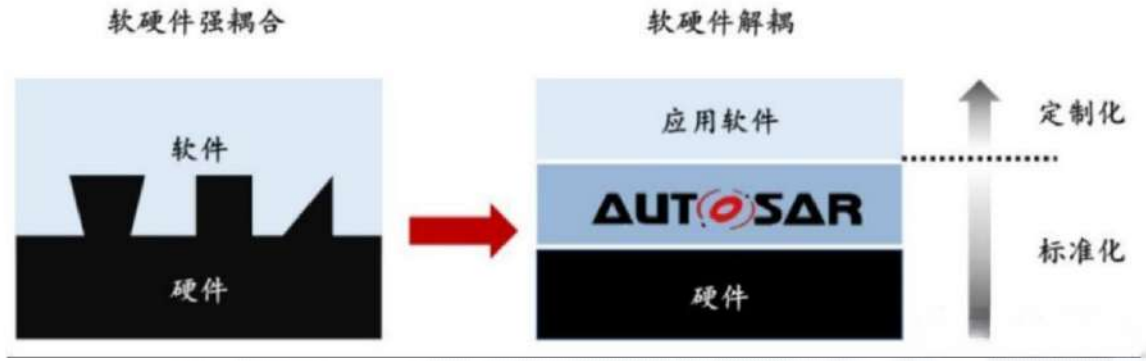
中间件：*AutoSAR*标准应用最为广泛

中间件（*Middleware*）隔离应用层与底层硬件，助力软硬件解耦。

中间件位于操作系统、网络和数据库之上，应用软件的下层，作用是为处于自己上层的应用软件提供运行与开发的环境，帮助用户灵活、高效地开发和集成复杂的应用软件，实现软硬件的解耦分离。在所有中间件方案中，AUTOSAR是目前应用范围最广的车载电子系统标准规范。

AutoSAR - What is it?

- AutoSAR 是AUTomotive Open System ARchitecture的简称（翻译：汽车开放系统架构）。
- AutoSAR是由全球汽车制造商、零部件供应商以及各种研究、服务机构共同参与制定的一种**汽车电子系统的合作开发框架**，并建立了一个开放的**汽车控制器（*ECU*）标准软件架构**，规范了车载操作系统标准与 API 接口。



AutoSAR

https://www.autosar.org/

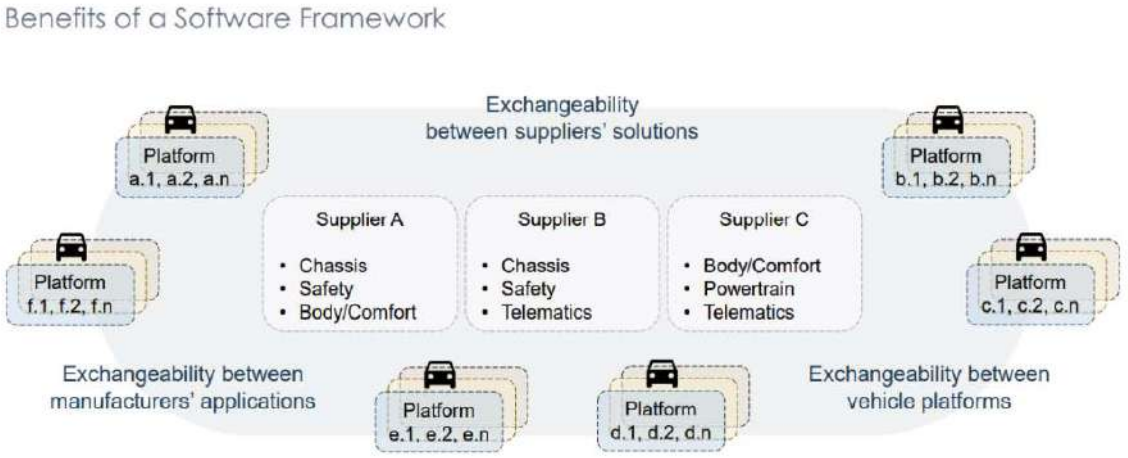
2003 年 7 月，宝马、博世、大陆、戴姆勒、福特、通用、PSA、丰田、大众等 9 家汽车行业巨头发起了AUTOSAR联盟的建立，这 9 家公司后来也成为 AUTOSAR 联盟的核心成员。截至 2021 年 7月，AUTOSAR 联盟已经拥有了 300 多家合作伙伴，包括全球各大主流整车厂、一级供应商、标准软件供应商、开发工具与服务提供商、半导体供应商、高校以及研究机构等，其中也包括了华为、百度、长城、东风、一汽、上汽、吉利、蔚来、拜腾、宁德时代等国内厂商。

Core Partner

Nine companies founded the AUTOSAR partnership to consolidate the expertise of partner companies in the automotive industries and define an automotive open system architecture standard to support the needs of future in-car applications.

AutoSAR

AUTOSAR 旨在通过提升 OEM 以及供应商之间软件模块的可复用性和可互换性来改进对复杂汽车电子电气架构的管理。



The **AUTOSAR Software Framework** promotes software module reuse and exchangeability.

AutoSAR

目前，*AUTOSAR* 拥有 *Classic Platform* 和 *Adaptive Platform* 两大平台，分别对应传统控制类车辆电子系统与对应自动驾驶的高性能类车载电子系统。

AutoSAR

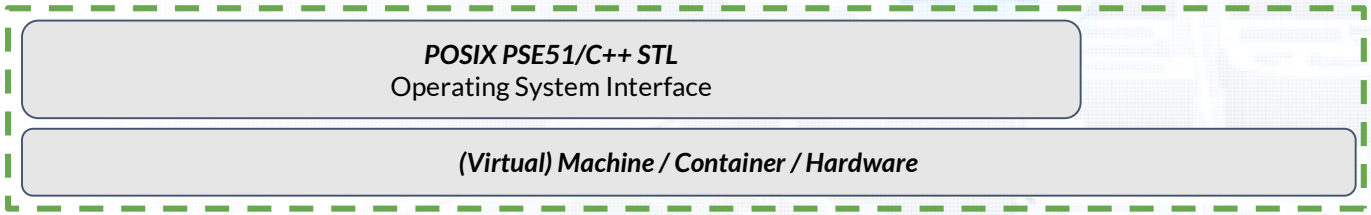
目前，***AUTOSAR*** 拥有 ***Classic Platform*** 和 ***Adaptive Platform*** 两大平台，分别对应传统控制类车辆电子系统与对应自动驾驶的高性能类车载电子系统。

- ***Classic Platform（CP）***：Classic Platform 是 AUTOSAR 针对传统车辆控制嵌入式系统的解决方案，具有严格的实时性和安全性限制。
- ***Adaptive Platform（AP）***：Adaptive Platform 是 AUTOSAR 面向未来自动驾驶、车联网等复杂场景而提出的一种新型汽车电子系统软件架构标准。

AutoSAR - AP 架构

Adaptive Platform 自下而上可分为三层：

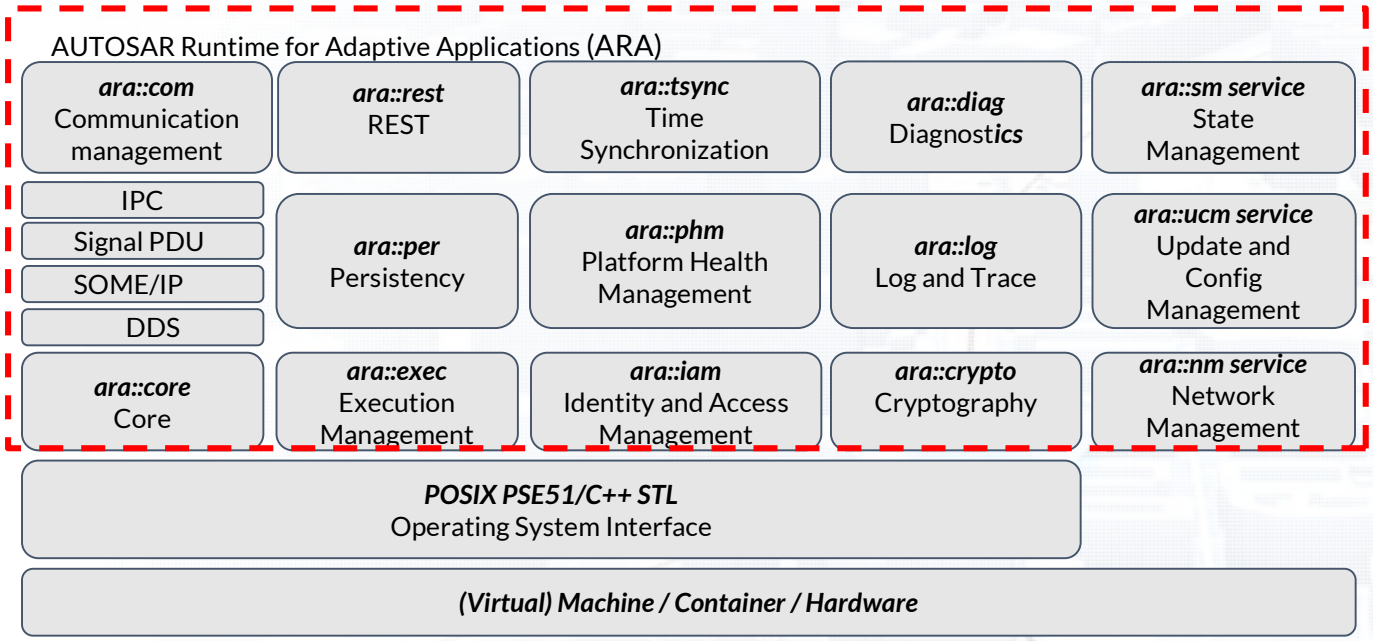
- 硬件层：AP 可将运行的硬件视~~Machine~~
- 实时运行环境层(ARA)：由功能集群提供的一系列应用接口组成，分为API和 service 两种接口类
- 应用层：运行在 AP 上的一系列应用。



AutoSAR - AP 架构

Adaptive Platform 自下而上可分为三层：

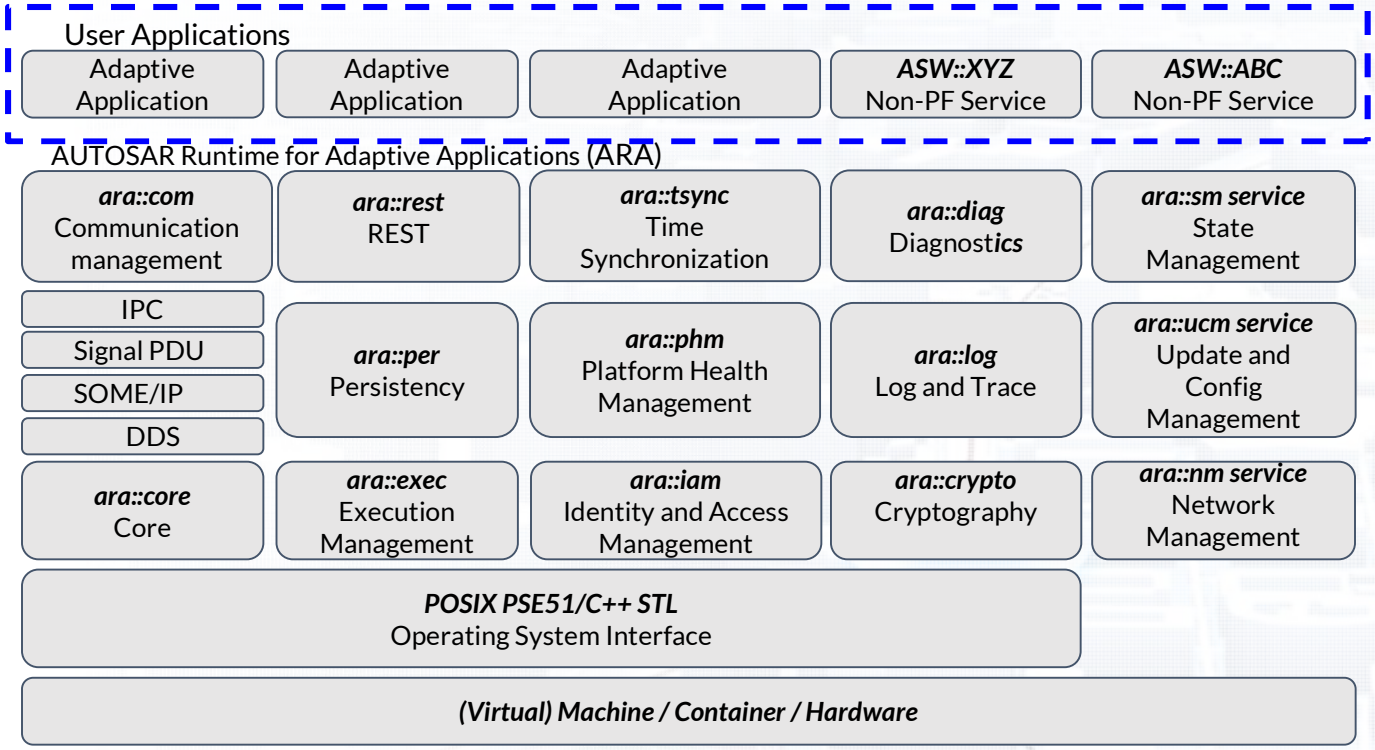
- 硬件层：AP 可将运行的硬件视Machine
- 实时运行环境层(ARA)：由功能集群提供的一系列应用接口组成，分为API和 service 两种接口类
- 应用层：运行在 AP 上的一系列应用。



AutoSAR - AP 架构

Adaptive Platform 自下而上可分为三层：

- 硬件层：AP 可将运行的硬件视Machine
- 实时运行环境层(ARA)：由功能集群提供的一系列应用接口组成，分为API和 service 两种接口类
- 应用层：运行在 AP 上的一系列应用。



AutoSAR

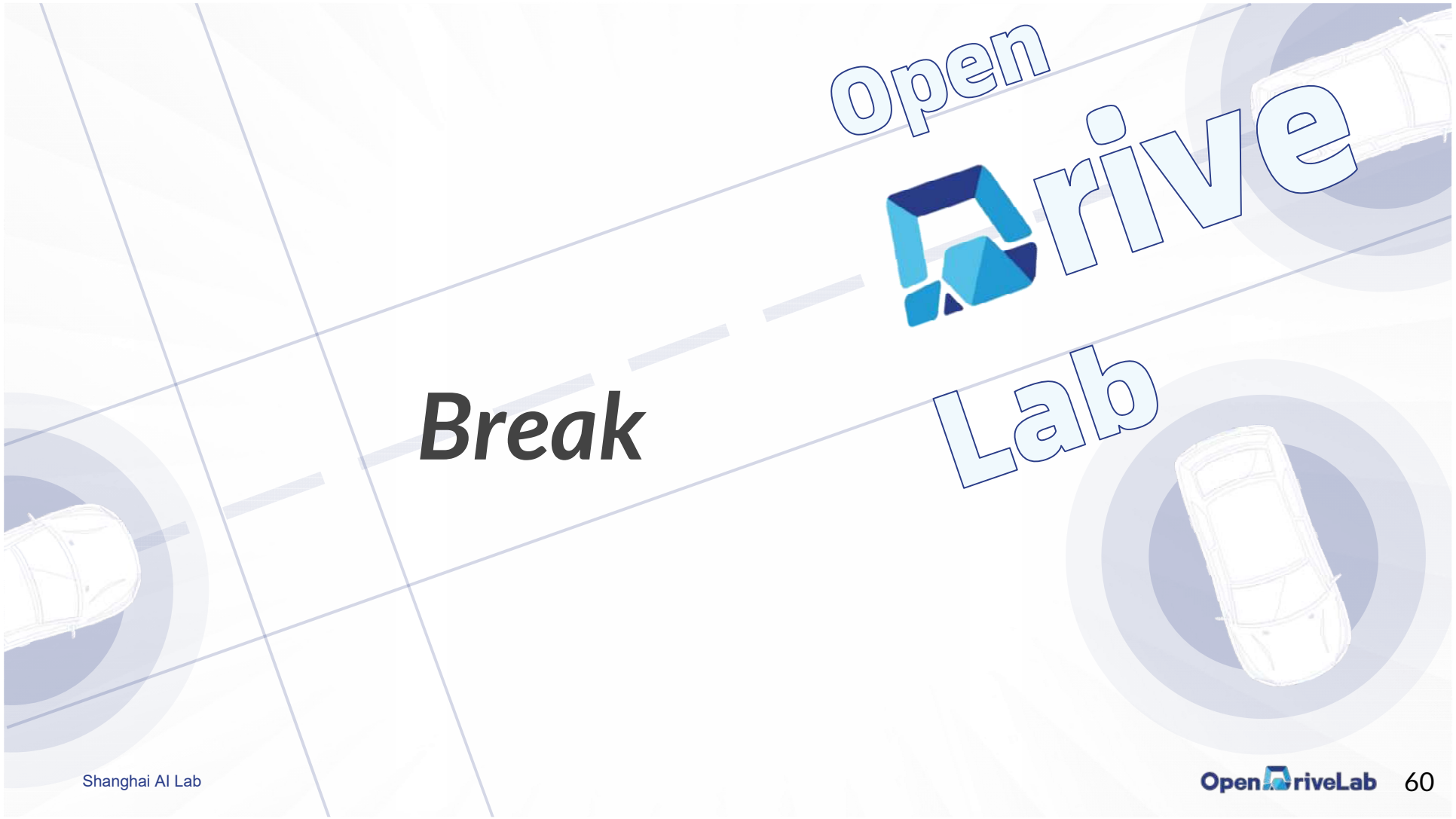
	Classic Platform	Adaptive Platform
使用语言	C	C++
实时性	硬实时	软实时
适用场景	传统 ECU（如 ECM、VCU、BMS、MCU 等）	自动驾驶、辅助驾驶、车联网
应用架构	面向信号的架构（SignalOriented Architecture）	面向服务的架构（Service-Oriented Architecture）
功能升级	一般 ECU 开发后比较固定	可灵活在线升级
安全等级	最高到 ASIL D	最低 ASIL B（最高可到 D）
主要通信方式	CAN、LIN	以太网
操作系统	Autosar OS（OSEK OS）	POSIX OS（Linux、QNX 等）

AutoSAR

***Adaptive Platform* 相较 *Classic Platform* 更加适应于当前汽车架构向 *SOA* 转型的大趋势。**

- Adaptive Platform 支持虚拟化技术以及多系统并存的架构，只要是 POSIX OS 都可使用，包括那些达到 ASIL-D 的操作系统，兼容性广，可移植性高，相较 CP 更有优势。

- CP 主要支持面向信号的架构（Signal-Oriented Architecture），以基于信号的静态配置的通信方式（CAN、LIN）为主，开发后功能升级较为困难；AP 主要支持面向服务的软件架构（SOA），以基于服务的 SOA 动态通信方式（SOME/IP）为主，硬件资源间的连接关系虚拟化，不局限于通信线束的连接关系，软件也可实现灵活的在线升级。



Break

OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

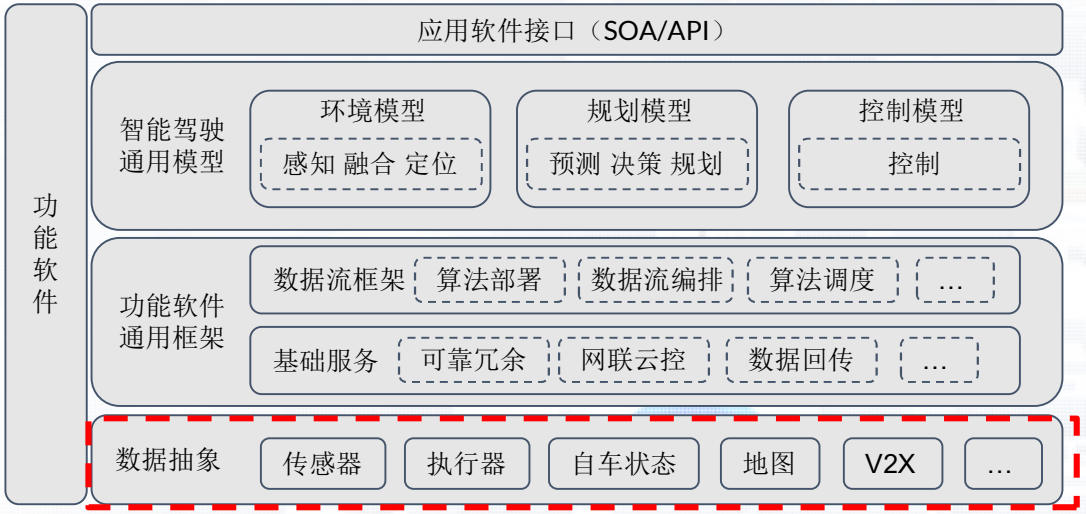
Functional Software & Application

2 System and Infrastructure

功能软件

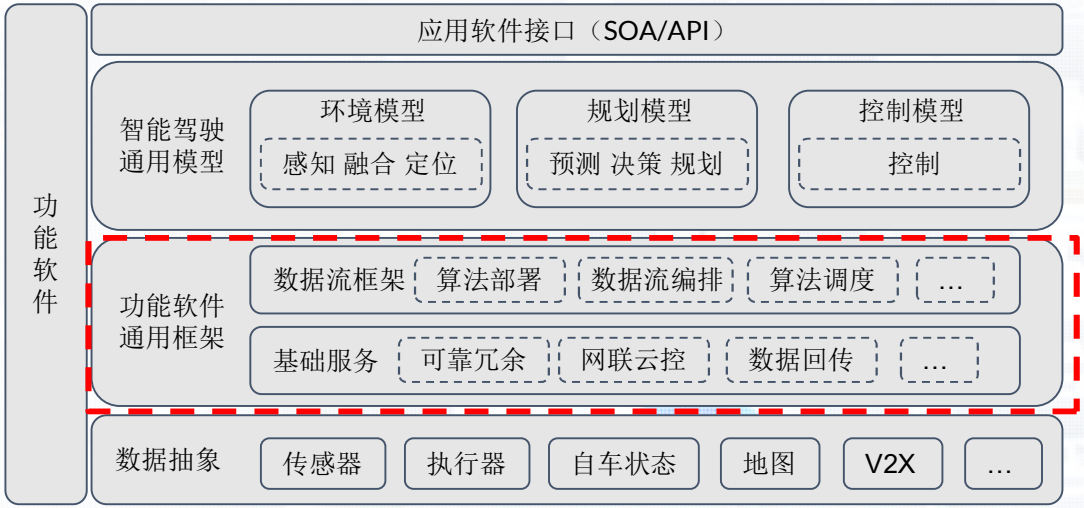
功能软件主要包含自动驾驶的核心共性功能模块。自下而上可分为三个主要部分：

- **数据抽象**对传感器、执行器、自车状态、地图以及来自云端的接口等数据进行标准化处理，为上层各模型提供数据源



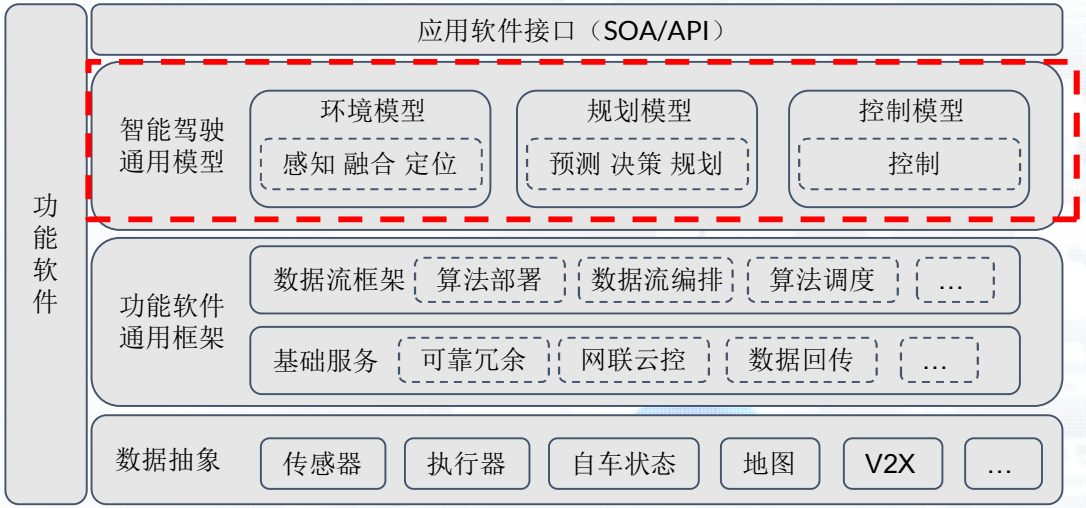
功能软件

- **功能软件通用框架**是承载智能驾驶通用模型的基础，可以分为**数据流框架**和**基础服务**两部分。
  - **数据流框架**向下封装不同的智能驾驶系统软件和中间件服务，向智能驾驶 通用模型中的算法提供与底层系统软件解耦的算法框架。
  - **基础服务**是功能软件层共用的基本服务，包括可靠冗余组件、信息安全基 本服务以及网联云控服务等。



功能软件

- **智能驾驶通用模型**是对智能驾驶中智能认知、智能决策和智能控制等过程的模型化抽象。对应于自动驾驶中环境感知、决策与规划、控制与执行三大部分，通用模型也可以分为**环境模型**、**规划模型**和**控制模型等**。



应用软件

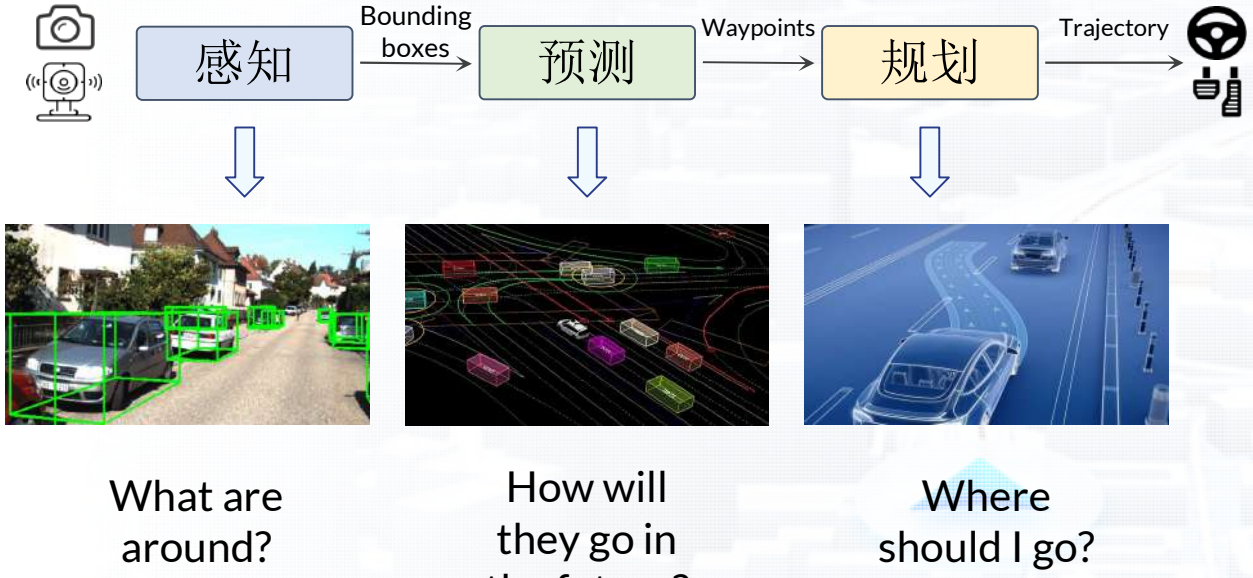
应用软件作为系统软件与功能软件之上独立开发的软件程序，更是品牌智能化产品力的直接体现。应用软件主要包括**面向自动驾驶算法、地图导航类、车载语音、*OTA* 与云服务、信息娱乐**等。

其中，**自动驾驶算法**是决定车辆智能化水平的**关键所在**。自动驾驶算法覆盖**感知、决策、执行**三个层次。

自动驾驶算法

难点 | 多样化的天气, 照明条件, 场景

自动驾驶任务/*Autonomous Driving (AD) Tasks*



自动驾驶算法

传统方案是基于模块化设计 / *Modular Based*

优势

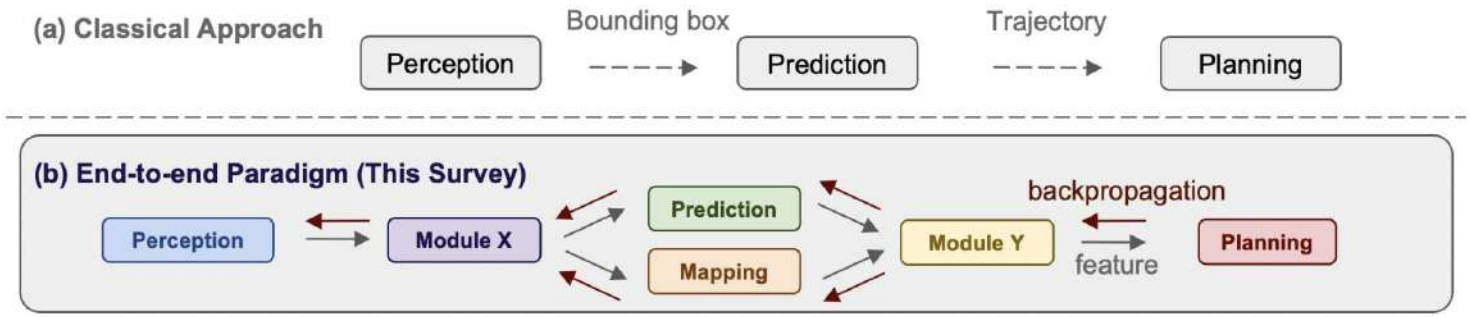
+ 便于**独立**的团队对不同模块进行开发
  + 对数据集友好
  + 可以对中间结果进行定量评估
  + 可解释性强
+ 可以并行实车部署

劣势

- 容易累积错误，丢失信息。**模块间传递的是结果而非特征**
- 数据集标注开销巨大
- 独立模块计算成本大

自动驾驶算法

端到端（*End-to-end*）方案给出了另一种可能



https://github.com/OpenDriveLab/End-to-end-Autonomous-Driving

端到端自动驾驶系统：

- 将原始传感器数据作为输入
- 输出轨迹规划，或低级别的控制信号

自动驾驶算法

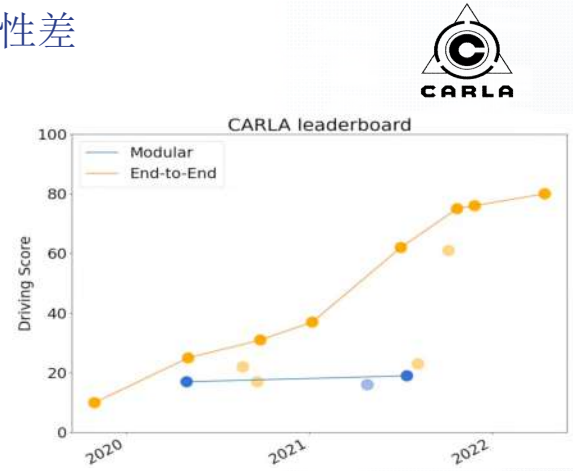
优势

+ 将所有模块合并为一个可**联合训练**的单一模型带来的**便利性**
+ 避免模块化设计带来的级联错误
+ 直接**针对最终任务进行优化（规划/轨迹预测）**
+ 计算效率高 (共享 backbone), 对最终产品友好

自动驾驶算法

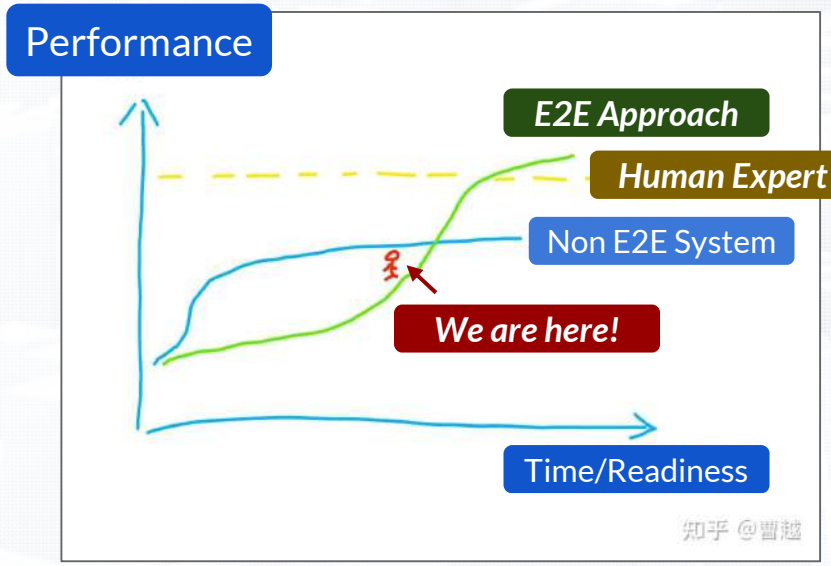
劣势

- 只能在模拟器和机载测试中进行闭环评测(Closed-loop evaluation)
- 缺少真实世界数据
- 可解释性差



Credit to Andreas Geiger @ CVPR Workshop 2023

E2E vs Non-E2E



Credit to Dr. Yue Cao @ Zhihu

自动驾驶算法

工业界

E2E Vehicle

...
v12 is reserved for when FSD is **end-to-end AI**, from images in to steering, brakes & acceleration out.

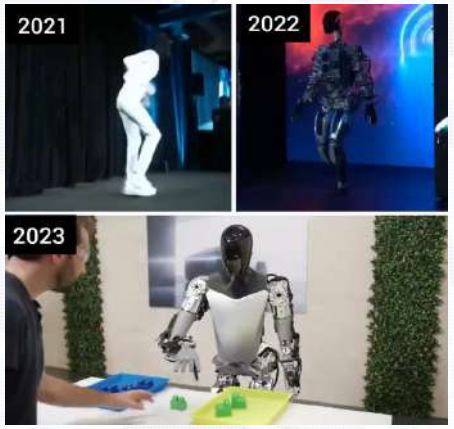
No hard-code.
Completely learning on its own.
End-to-end, video to neural network to controls.
Don't need map data at all, only coordinates!
No cellular connection needed.

- Probably e2e as a backup module
- Massive high-quality data prevail
- Mapless is promising and feasible

E2E Robot

Tesla Optimus @Tesla_Optimus · Sep 24
Optimus can now sort objects autonomously

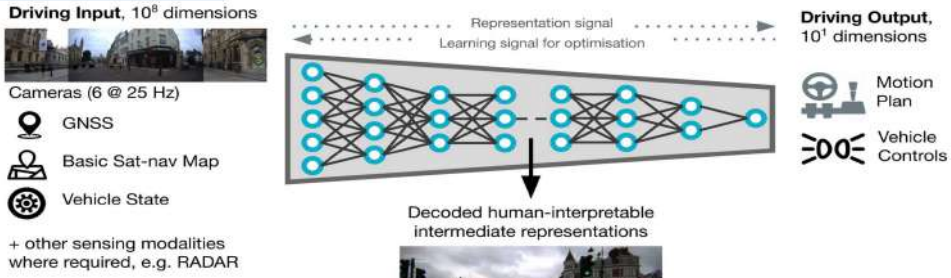
Its neural network is trained fully **end-to-end**: video in, controls out.



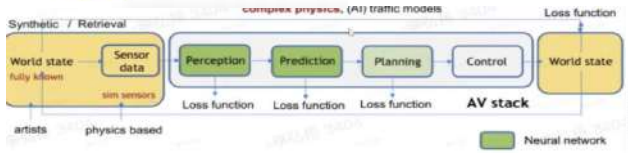
自动驾驶算法 工业界

And many others ...

WAYVE



NVIDIA



comma.ai

- *Openpilot is an open-source driver assistance system.*
- *Openpilot performs the functions of Automated Lane Centering (ALC) and Adaptive Cruise Control (ACC) for 250+ supported car makes and models.*

https://arxiv.org/abs/2206.08176

自动驾驶算法 学术界

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Transfuser, KING, Misconceptions, DA-RB, etc.

TEXAS The University of Texas at Austin

LBC, WoR, LAV, etc.

UNIVERSITY OF TORONTO Uber Waabi

NMP, P3, MP3, UniSim (sort of)

BOSTON UNIVERSITY

LbW, SelfD, CaT, AnyD, etc.

ETH zürich

Roach, etc

香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

MMFN, Carl-lead, PMP, etc

Valeo

GRI

Horizon Robotics

VAD

UCLA

MetaDrive, ACO, CAT, etc

清華大學 Tsinghua University

ADAPT, etc

OpenDriveLab

OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Cloud Service

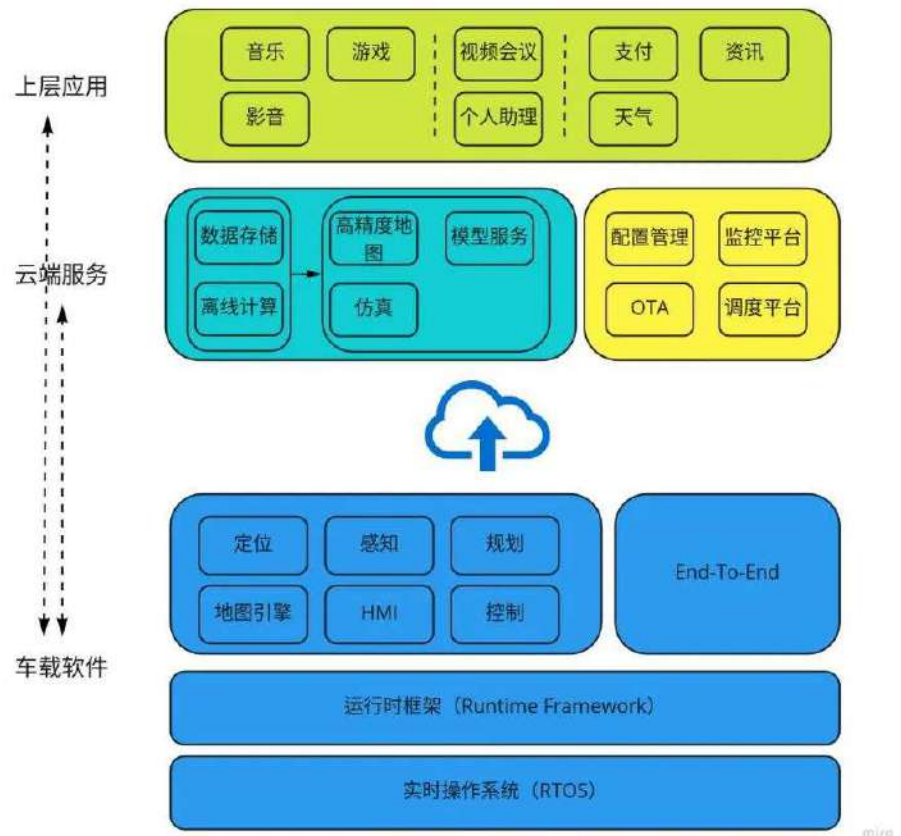
2 System and Infrastructure

Cloud Service

自动驾驶软件除了在车辆端运行的操作系统和无人驾驶系统，还包括云端提供的自动驾驶所需要的各种服务。

Cloud Service

自动驾驶软件除了在车辆端运行的操作系统和无人驾驶系统，还包括云端提供的自动驾驶所需要的各种服务。



Cloud Service

云服务也是自动驾驶不可或缺的一环，自动驾驶相关的高精度地图、数据存储、模型训练、自动驾驶仿真等都依赖于云服务。

目前已经宣布能提供自动驾驶服务的云平台有百度Apollo、亚马逊AWS和华为Octopus，提供的主要功能包括：

- 数据采集和存储
- 数据Pipeline
- 模型训练部署
- 自动驾驶仿真。

Cloud Service

https://www.apollo.auto/

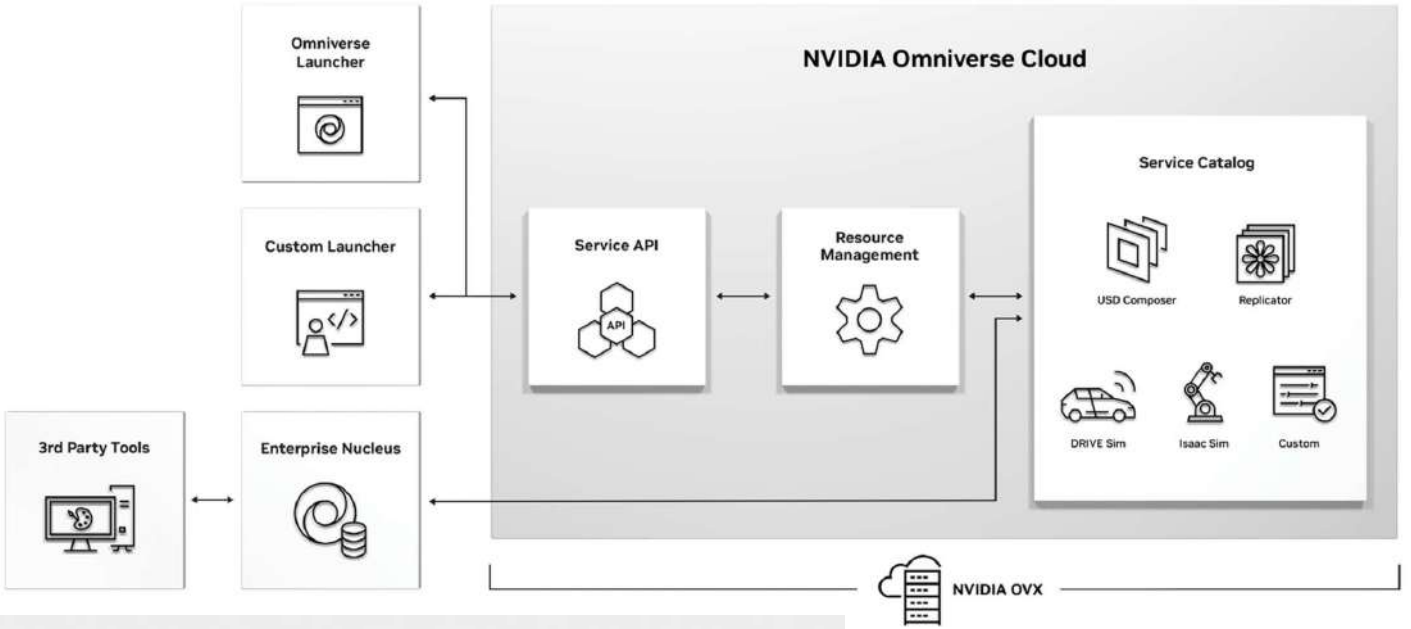
百度*Apollo*平台架构图

Apollo开放平台9.0全新架构图

Cloud Service

英伟达

https://www.nvidia.cn/omniverse/cloud/



Omniverse Cloud 的主要优势

即时安全访问

快速访问安全的云服务基础设施和NVIDIA Omniverse软件和服务。

单一来源解决方案

只需订阅一次，即可访问经过全面优化的基础设施和开发平台，以适应要求严苛的 3D 和数字化工作负载。

通过设计实现互操作和可扩展

真正的 PaaS 体验，助力企业团队构建先进的 Universal Scene Description（通用场景描述）美元基于）的工具和应用，3D 工作流程以及数字生。

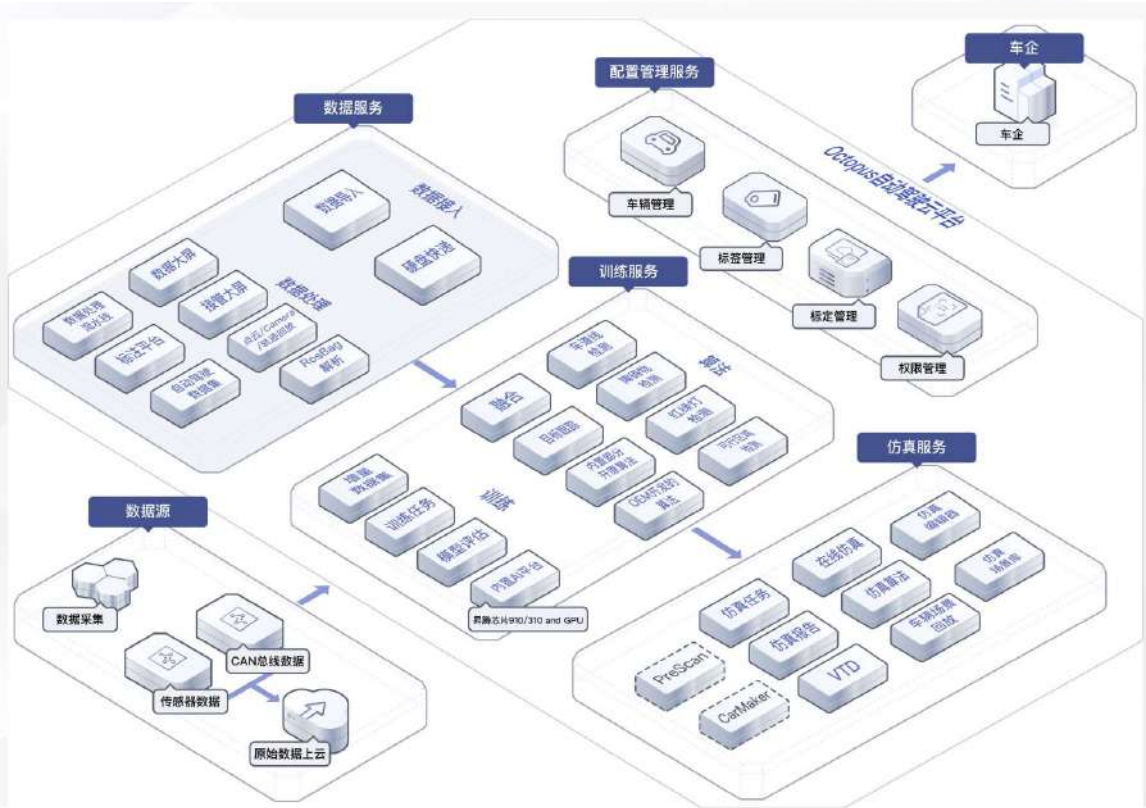
轻松扩展

随着工业数字化项目的增长，轻松扩展用户和应用程序。

Cloud Service

华为*Octopus*平台架构图

https://developer.huaweicloud.com/techfield/car/oct.html



数据的采集与存储

一辆无人驾驶车配置了多种传感器包括摄像头、激光雷达、毫米波雷达、GPS、IMU等。每天使用到的数据量高达4000GB，这些数据需要收集并存储，用于高精度地图制作和模型训练。

大数据存储一般涉及到两方面的问题：**一是数据传输，二是数据存储**。

数据的采集与存储

1、数据传输

- **数据传输需要对不同的数据类型进行分别对待。**
  例如，需要实时处理的数据，对传回数据中心的网络带宽提出了要求，而5G网络支持的上行速率约为50Mbps，即每分钟可以上传375MB数据，远远低于无人驾驶车的数据上传需求。这就要求我们对无人驾驶车产生的数据做分级，把高优先级的数据优先发送，低优先级的数据先保存在本地，等网络空闲之后再上传。
- **为了保证数据传输的经济性，还需要先在本地对数据做预处理。**

数据的采集与存储

2、数据管理

- 数据上传到数据中心之后，**需要对数据进行元信息管理**。由于自动驾驶的数据都是二进制文件，不具备可视化；场景也多种多样，路况、时间、天气不一样，产生的数据也不一样，所以详细记录数据的各项信息，包括数据录制时间、车辆编号、软件版本、天气情况以及行驶路段等，便于我们快速的找到所需的数据。
- 数据管理的另一个应用场景是**测试数据管理**，每天测试产生的大量数据，什么时候因为什么原因接管的，在回归测试中很关键。如果没有这些信息，假如我们需要查找所有因为红绿灯引发的问题进行回归测试，就会无从下手。

数据的采集与存储

3、数据存储

数据存储首先需要的是一个**分布式的文件系统**，大数据时代已经被广泛证明了分布式文件系统的好处，最主要的好处是容量可以水平扩展，而且可靠性高。

接下来对于数据库的选择，我们先分析下自动驾驶大数据应用场景和传统互联网的区别。

数据的采集与存储

3、数据存储

互联网	自动驾驶
数据生产方式是几亿用户，每人每天产生几条数据，合起来几个T的数据	
针对几亿用户，一般是选择key-value结构的数据库，例如HBase	
应用场景是拿用户的ID作为key，如果同时频繁的命中相邻的ID，被称为单点问题，导致容量难以提高	

数据的采集与存储

3、数据存储

互联网	自动驾驶
数据生产方式是几亿用户，每人每天产生几条数据，合起来几个T的数据	一辆车每天产生几个T数据
针对几亿用户，一般是选择key-value结构的数据库，例如HBase	
应用场景是拿用户的ID作为key，如果同时频繁的命中相邻的ID，被称为单点问题，导致容量难以提高	

数据的采集与存储

3、数据存储

互联网	自动驾驶
数据生产方式是几亿用户，每人每天产生几条数据，合起来几个T的数据	一辆车每天产生几个T数据
针对几亿用户，一般是选择key-value结构的数据库，例如HBase	HBase的单条数据最好是10M以内，因此需要按照地理位置或时间，拆分成很多key-value结构的小数据
应用场景是拿用户的ID作为key，如果同时频繁的命中相邻的ID，被称为单点问题，导致容量难以提高	

数据的采集与存储

3、数据存储

互联网	自动驾驶
数据生产方式是几亿用户，每人每天产生几条数据，合起来几个T的数据	一辆车每天产生几个T数据
针对几亿用户，一般是选择key-value结构的数据库，例如HBase	HBase的单条数据最好是10M以内，因此需要按照地理位置或时间，拆分成很多key-value结构的小数据
应用场景是拿用户的ID作为key，如果同时频繁的命中相邻的ID，被称为单点问题，导致容量难以提高	如果按照地理位置或时间组织key-value对刚好会导致单点问题；**如果把key做哈希散列，把地理位置信息打散，这样容量提高上去了，而这又恰恰和应用场景有冲突**。

数据的采集与存储

3、数据存储

- 自动驾驶需要的不是高并发读取，即同时几十万的并发，**而是一个用户连续读取大量数据，单台机器能够对数据做预取**。
- **高精度地图也不应该直接以*XML*格式保存，**占用的空间太大，应该把地图分块序列化之后再保存，压缩之后存储的效率会高很多。
- 日志文件则采用时间序列型数据库保存，当通过日志文件获取无人驾驶车位置的时候，可以准确的反应出无人车从起点到终点的时间序列。

综上所述，不同的数据需要选择不同类型的存储和数据库。自动驾驶的一些大数据场景可能根本不需要数据库，只需要文件系统就可以了，如果需要管理结构化的数据，可以用数据库存储文件路径，而把文件本身放到文件系统中。

数据处理*Pipeline*

深度学习模型训练、高精度地图生成以及自动驾驶仿真等都需要进行数据处理。**自动驾驶的数据处理流程包括：收集、清洗、标注、训练和部署**。

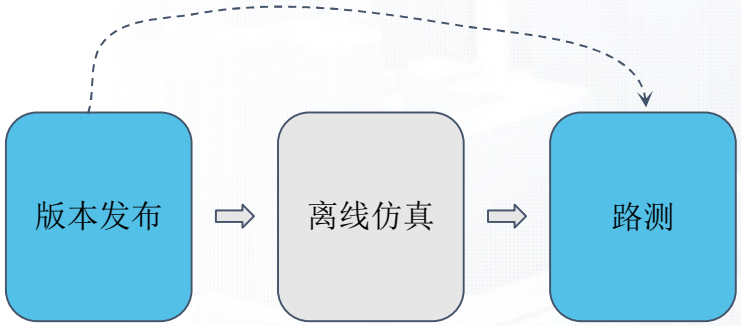
- **数据的自动化标注是很大的挑战**，通过工程的方法尽量减少人工标注，可以大幅度提高标注效率。实现自动标注通常有2种方法：一是通过机器自动标注，然后人工修正部分数据；二是通过仿真模拟生成大量标注好的数据。
- 数据处理的**另一个挑战是大规模并行处理数据**，由于数据量巨大，如何快速的处理数据是瓶颈。有很多优秀的分布式计算框架，其中Apache Spark可以构建大规模集群并发执行多个任务，在大规模数据处理中有非常好的实践。
- **还有一部分离线计算是利用空间换时间**，比如planning模块reference line的生成；routing线路事先计算保存；感知的ROI区域；定位用到的点云数据等。

地图服务

- 云端提供的高精度地图的道路信息比传统地图的要求更加精细，不仅仅包括道路信息，还包括车道信息、红绿灯信息、交通标志信息等。同时高精度地图的精度也比传统地图要求更高，需要达到厘米级。
- **一些动态信息可以通过地图服务的方式下发给无人车**，在高精度地图中，这部分信息被称为动态图层。动态图层包括：交通管制、交通拥堵状态、交通规则等，还包括周围的银行、医院、便利店等生活信息。
- **高精度地图的维护是目前面临的最大问题，因为涉及到整个地图的采集、加工和标注，实时维护大体量的高精度地图目前来说成本高昂**，一些高精度地图服务提供商提出采用众包的方式更新高精度地图。

仿真

- 自动驾驶仿真的首要目的是为了更早的发现问题，**业界预测要确保安全，自动驾驶的安全性测试需要行驶至少2.5亿英里**。如果全部采用真实环境测试，需要1000辆无人驾驶测试车每天测试100英里，不间断测试6.8年，短期内不可能实现。如果采用自动驾驶仿真，通过模拟真实场景的数据，让无人车大规模部署在虚拟环境中测试，然后再去真实场景路测，可以极大提高发现问题的效率。



仿真

- 除了要求能够大规模部署，**仿真的另一个需求是问题快照，在测试出现问题的时候能够保存现场**。
- 除了自动驾驶功能测试，仿真还可以通过生成数据来帮助模型训练。**数据的生成方式有2种，一种方式是生成标注好的数据**，通过在仿真环境中模拟真实的车辆、行人、建筑物等，这些信息在仿真环境中都是已知的，可以直接生成标注好的数据用来进行模型训练。**另一种方式是利用强化学习在仿真环境中模拟开车**，进行端到端的自动驾驶模型训练。

仿真 - 英伟达

在公共道路上进行测试时，自动驾驶汽车不可能遇到所有的交通情况。在 ***NVIDIA DRIVE Sim™*** 中，虚拟汽车可以在各种场景（从常规驾驶到罕见乃至危险情况）中进行数百万英里的仿真驾驶，并且比现实世界中的驾驶测试更高效、更经济实惠、更安全。高保真平台还可以生成基于物理效果的合成传感器和真值数据，从而根据任意开发需求定制场景。

什么是合成数据？

在训练任何 AI 模型时，都需要用到仔细标记的多样化数据集。这些数据集通常包含数千乃至数千万个元素，其中一些元素超出可见范围。在实际中，采集和标记此数据不仅费时费力，而且成本高昂。这可能会阻碍 AI 模型的开发，并增加解决问题所需的时间。

合成数据由计算机仿真生成，组成要素为 2D 图像或文本。您可以结合使用合成数据和实际数据来训练 AI 模型。合成数据生成 (SDG) 可以大幅节省时间和成本。

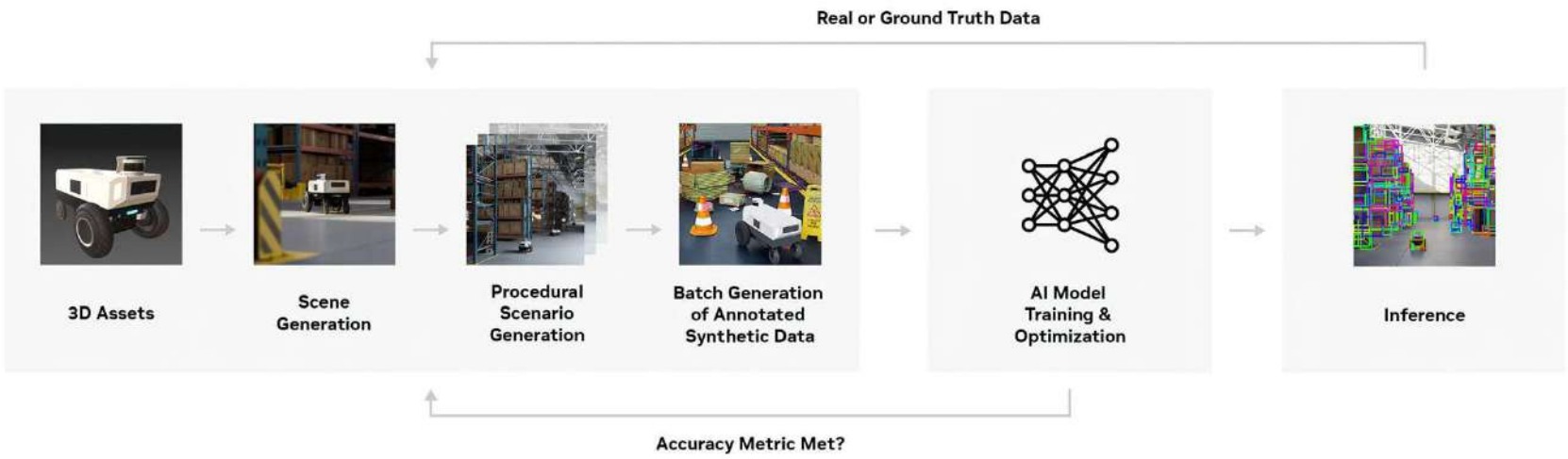
https://www.nvidia.cn/self-driving-cars/simulation/

仿真 - 英伟达

Youtube 视频

https://www.nvidia.cn/self-driving-cars/simulation/

仿真 - 英伟达


Real or Ground Truth Data
3D Assets
Scene Generation
Procedural Scenario Generation
Batch Generation of Annotated Synthetic Data
AI Model Training & Optimization
Inference
Accuracy Metric Met?

仿真 - 英伟达

合成数据的优势

节省成本

弥补数据差异，降低为训练 AI 模型获取和标记数据所需的总体成本。

保护隐私

通过生成多样的数据集来模拟现实世界，解决隐私问题并避免受偏见影响。

提高精度

在训练中使用包含罕见但重要的极端案例的数据（通常不可能收集到），以创建高度准确的通用 AI 模型。

实现扩展

生成可根据制造、汽车、机器人等领域的具体用例进行扩展的数据。

OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

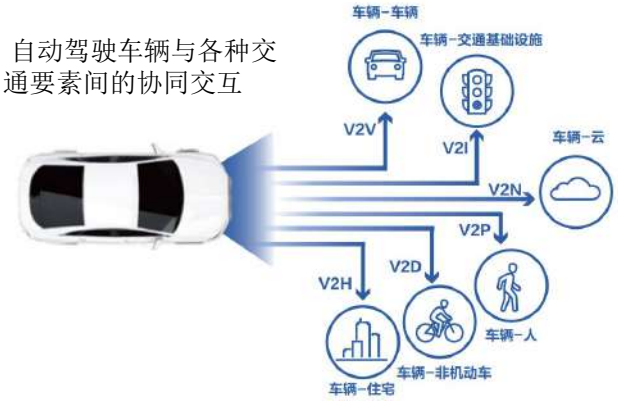
Vehicle-to-everything (V2X)

2 System and Infrastructure

V2X: An Overview

车路协同系统整体架构

自动驾驶车辆与各种交通要素间的协同交互



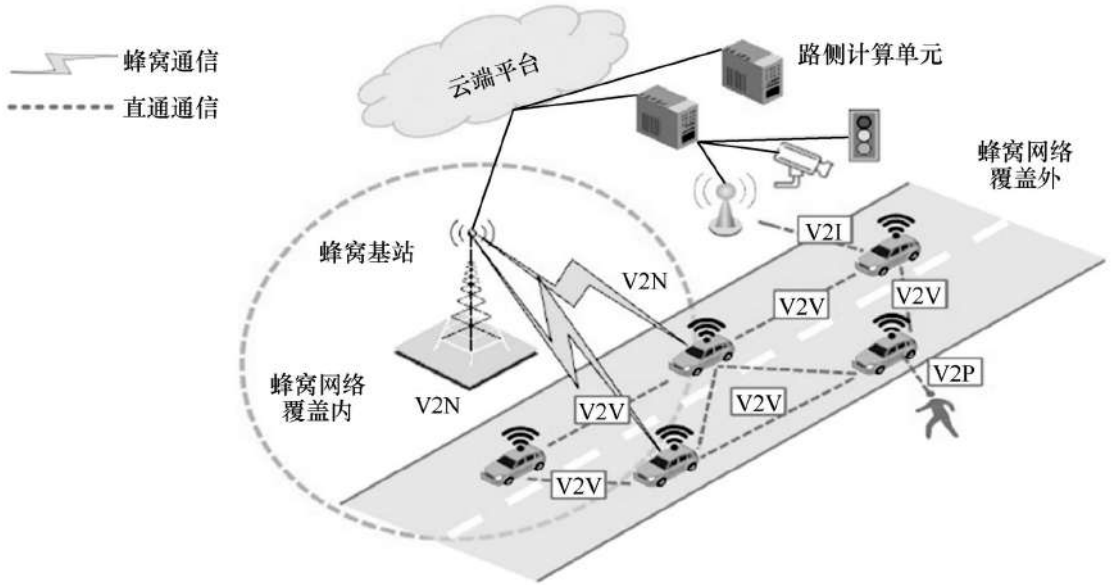
V2X旨在通过车辆与众多交通参与者的交互来帮助实现智能驾驶，目前以车路协同技术路线为主：

- 包含包括车端、路端和云端多方参与
- 实现协同感知、协同控制等多个协作层次

V2X: 通信模块

V2X通信主要由两种相互独立、互为补充的工作模式组成：

- 直连无线通信直接实现端到端的低时延通信，满足车车、车路等设备间近程信息交互需求
- 蜂窝移动通信通过基站转发来进行数据传输，具有高可靠性、大带宽等特点，解决远程的信息服务需求



V2X: 应用场景

V2X在多种典型场景下可以提升

安全和效率

- 超视距/盲区协同感知帮助提前做出预判和决策控制，降低事故风险
- 意图冲突车辆的协同控制，提升路口等复杂环境下的通行效率
- 施工等交通事件的提前感知，避免车辆的长时间阻塞等待
- 多车的联合编队行驶，减少能源消耗、缓解交通拥堵

盲区障碍物感知

交叉路口车辆的协调通行

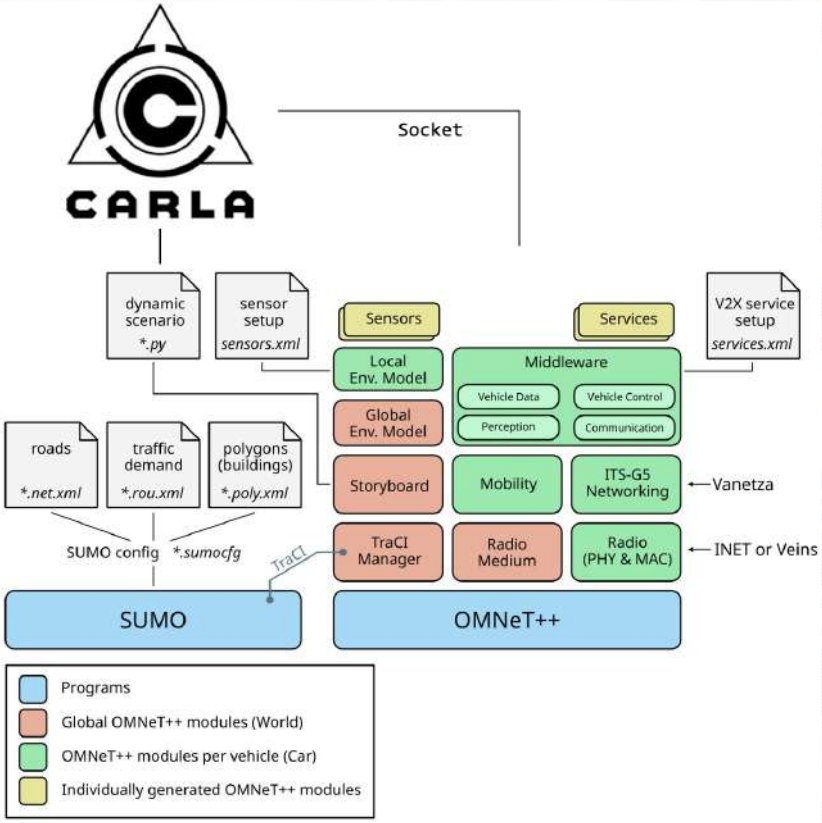
施工占道的提前绕行

多车的联合编队行驶

V2X: 前沿研究

https://github.com/riebl/artery

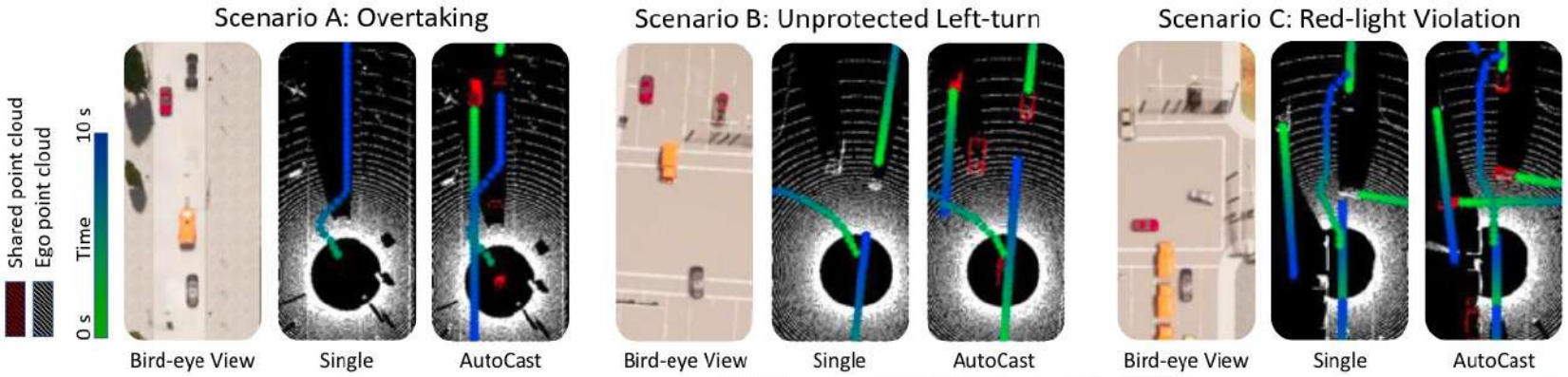
- *Carla + Artery*：基于*OMNeT++*的通信仿真工具
  - **优势**：可以实现通信的全栈式建模，理论上最真实，包括从消息层到物理层的整个通信栈
  - **劣势**：是基于标准的通信协议开发，目前字段只支持CAM和CPM Message Type，难以传输原始点云和中间特征
  - 实现方式：
    - 通过SUMO（交通流仿真软件）作为全局Agent的生命周期管理和运动控制
    - 对每个agent都会在OMNeT++中实例化一个CarService作为通信服务
    - 对每个agent都会在Carla中实例化一个car



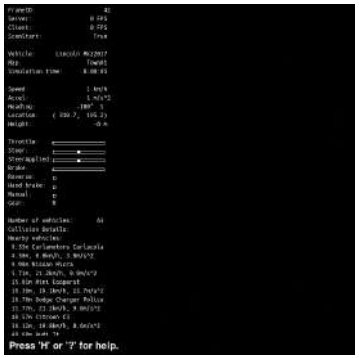
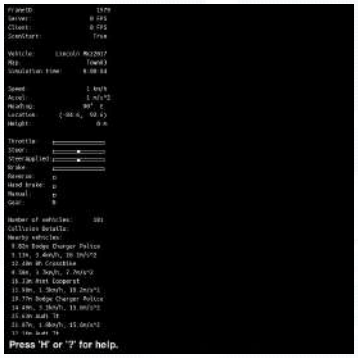
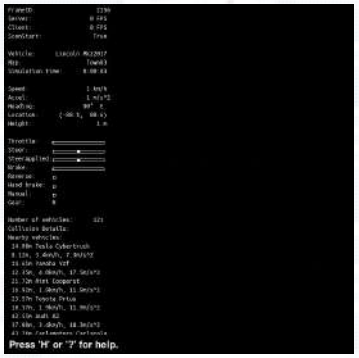
V2X: 前沿研究

https://github.com/hangqiu/AutoCastSim

- ***AutoCastSim***：基于*Carla*实现的场景级别的仿真
  - 基于此的工作：AutoCast，COOPERNAUT等
  - 实现方式：通过MQTT构建消息队列，通过本地Scock之间的传输来模拟通信
  - 优势：可以在Carla仿真框架里面实现真实的数据传输过程
  - 劣势：实际上的传输时延距离真实情况差距较大，场景比较简单



V2X: 前沿研究

Overtaking	Unprotected Left-turn	Red-light Violation
		
		

OpenDriveLab

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

课后部分

思考题

如何在通信出现丢包、干扰等情况下，保证*V2X*服务的稳定性？

参考资料

[1] Ren S, Lei Z, Wang Z, et al. Interruption-Aware Cooperative Perception for V2X Communication-Aided Autonomous Driving[J]. arXiv preprint arXiv:2304.11821, 2023.

[2] Li Y, Fang Q, Bai J, et al. Among us: Adversarially robust collaborative perception by consensus[J]. arXiv preprint arXiv:2303.09495, 2023.

[3] Li J, Xu R, Liu X, et al. Learning for vehicle-to-vehicle cooperative perception under lossy communication[J]. IEEE Transactions on Intelligent Vehicles, 2023.

References

[1] 国信证券-软件与服务行业汽车智能化系列专题 https://pdf.dfcfw.com/pdf/H3_AP202202111546286745_1.pdf?1644567954000.pdf

[2] 浅析自动驾驶硬件系统 https://zhuanlan.zhihu.com/p/142986670

[3] 万字长文介绍无人驾驶软件和云服务 https://apollo.baidu.com/community/article/197

[4] 自动驾驶系统概述 https://zhuanlan.zhihu.com/p/527580633

[5] 百度 Apollo 5.5 自动驾驶软件系统概述 https://dingfen.github.io/apollo/2020/10/14/apollo-intro.html

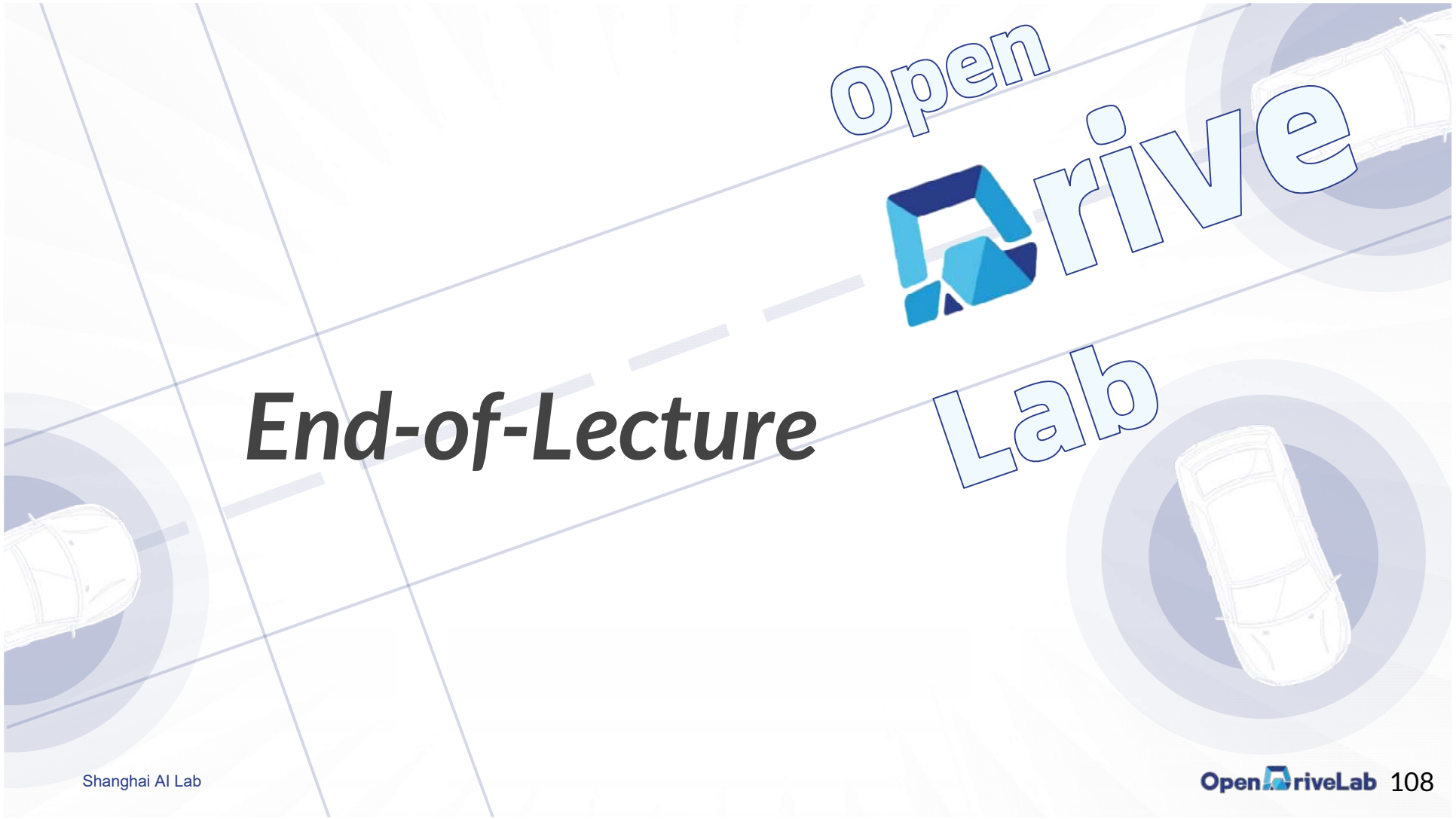
[6] 东方证券-智能汽车深度系列之一：汽车软件的星辰大海 https://pdf.dfcfw.com/pdf/H3_AP202202221548462770_1.pdf

[7] 自动驾驶系统设计的那些底层软件开发中的重点解读 https://www.dongchedi.com/article/7164639202259517952

[8] 基于 AUTOSAR 的自动驾驶软件架构 www.uml.org.cn/car/202309264.asp

[9] 国内外自动驾驶OS盘点 https://www.sohu.com/a/577938734_121124366

[10] 面向自动驾驶的车路协同关键技术与展望2.0
https://apollo-new.cdn.bcebos.com/means/document/air/%E3%80%8A%E9%9D%A2%E5%90%91%E8%87%AA%E5%8A%A8%E9%A9%BE%E9%A9%B6%E7%9A%84%E8%BD%A6%E8%B7%AF%E5%8D%8F%E5%90%8C%E5%85%B3%E9%94%AE%E6%8A%80%E6%9C%AF%E4%B8%8E%E5%B1%95%E6%9C%9B2.0%E3%80%8B.pdf



End-of-Lecture