

元邀 CARSMOS 开源智能出行项目 白皮书

2022 版



元邀 CARSMOS 开源智能出行项目组

目录 CONTENTS

项目综述 Executive Summary	01
1. 百年未有之大变局催生新出行产业	03
2. 新出行产业的“三零”愿景：零排放、零事故、零成本	04
3. 智能汽车和新的出行产业在教育、研究、标准、产业化方面的挑战和机会	06
3.1 软件定义智能汽车之转型与挑战	06
3.1.1 智能汽车软件是人类创造的最复杂的软件	06
3.1.2 自动驾驶是AI应用的集大成者	06
3.1.3 智能网联汽车是实现全自动驾驶的必由之路	06
3.1.4 软件定义汽车(SDV)是智能汽车一切皆服务商业模式的基础	06
3.1.5 电动滑板底盘和线控技术将会重塑汽车产业的生产制造	07
3.2 智能汽车教育转型的巨大挑战	08
3.3 智能汽车研究的挑战	10
3.4 产业缺芯少魂战略选择的挑战	11
3.4.1 面向算法定义的AI芯片架构和芯粒集成	11
3.4.2 自动驾驶统一数据集和大模型	12
3.4.3 可信安全编程语言和中间件	15
3.5 产业标准的挑战	16
4. 发展开源社区，融合产学研力量，构建统一软件和计算平台	17
4.1 开源创新力量和生态价值已经被IT产业反复验证	17
4.2 基于开源建立全栈的智能汽车统一平台	18
4.2.1 EDIT EV项目：面向低速无人应用场景的电动滑板底盘	18
4.2.2 PULSAR项目：模块化、虚拟化、集中化的开放电子电气架构	20
4.2.3 AutoExpress项目：敏捷DSA和D2D互联接口标准	23
4.2.4 DORA：数据流计算中间件	25
4.2.5 ESSA：弹性带状态无服务器计算架构	29
4.2.6 EtymOS项目：自动驾驶的软件2.0平台	29
4.2.7 OASIS：仿真测试平台和场景描述语言	35
4.3 采用在线仿真平台促进高质量智能汽车教育	41
4.4 智能汽车全栈参考设计平台，研究转型为数据场景驱动加开源竞赛	42
4.5 产学研结合：统一平台，加速落地	44
4.6 形成关键产业标准	44
5. 结束语	46

项目综述 Executive Summary

新能源智能汽车是汽车产业诞生百余年来最大的变革，将会满足全球对节能减排，产业升级以及发展中国家工业化的诉求，建立以零排放、零事故、零成本为特征的下一代智能清洁出行体系。像其他数字产品在全球的普及一样，新能源智能汽车将会改变全球汽车和出行产业的格局，普惠全世界。新一代的出行产业是汽车、IT、通信、交通、能源产业的大融合，当前汽车产业的领先公司和国家将会发生逆转，产业会发生大规模重构和转移，是全球第四次工业革命的王冠。

同时，智能网联汽车产业的发展需要解决一系列的挑战。

- ◎ 智能汽车软件全栈将有5亿行代码，是人类创造的最复杂的软件之一，也是AI应用的集大成者。为智能汽车建立软件技术体系是前所未有的挑战。
- ◎ 智能汽车产业需要数以百万计的高端软件人才。高校教育、职业教育与产业的需求差距巨大。同时，智能汽车也需要理论研究的支撑，尤其是AI的理论研究，但是学术界在AI研发基础设施上和产业界的差距越来越大。
- ◎ 智能汽车产业面临着缺芯少魂的供应链安全问题，但是在未来技术路线上，存在类似攀登珠穆朗玛峰北坡和南坡的战略选择。
- ◎ 单车智能vs车路协同两条发展路线，各有千秋，适合不同国情和发展阶段。
- ◎ 最后，智能汽车需要监管和标准的保驾护航才能降低新技术不成熟带来的安全和信任问题。

这个过程中，我们可以建立和智能汽车产业规模相匹配的标准和平台体系。

解决上述挑战，需要调动产业和社会的力量。开源平台作为中立、非盈利、全球协作平台，已经在云计算、智能手机、AI等全球产业，形成了事实标准，构建出了价值万亿美元的生态价值，所以，智能汽车的诸多挑战也需要通过开源平台来解决。

元遨 CARSMOS (寓意 Cars' Mobility Operating System), 旨在提供一个技术先进的、标准的、开源的平台，解决智能汽车研究、教育和产业合作中的平台封闭和碎片化问题。本项目群包含了来自产、学、研的贡献、赞助和参与：

- EDIT EV 低速电动滑板底盘,
- PULSAR 开放电子电气架构,
- AutoExpress 芯粒互联技术和敏捷 DSA 芯片设计标准,
- DORA 数据流中间件和 SDV (软件定义汽车) 业务开发平台,
- EtymOS 自动驾驶大模型和数据飞轮平台,
- OASIS 场景描述语言和仿真测试平台。

开源软件部分采用 Apache 2.0, 兼容木兰宽松许可证 v2(MulanPSLv2), 开源硬件部分采用 CERN-OHL-W-2.0 协议。我们期待通过这些统一平台技术，服务智能汽车教育和科研的基础需求，也成为和产业对标对话的平台，让教育和研究紧密对标产业需求，加速研究成果到产业的落地速度。



1. 百年未有之大变局催生新出行产业

1876年德国人Otto制造出第一台现代意义上的内燃机，开启了现代汽车工业的序幕；1908年美国人福特采用大规模流水线生产model T型汽车，标志着汽车开始大规模进入家庭。这些伟大的发明彻底改变了人类社会的面貌，指数倍地降低了交通运输成本，使得大规模的产业分工和商品流通成为可能，汽车产业也随之成长为万亿美元规模的产业，是发达工业化国家的支柱产业之一。

但是，人类历史上伟大的发明往往是双刃剑，汽车也让人类社会付出了昂贵代价。根据联合国世界卫生组织的统计，全球每年有约135万人死于交通事故，每年的交通事故产生的医疗和保险费用大约占全球GDP的2%-3%。在城市里，交通拥堵造成每年人均浪费96小时，30%左右的空间用于停放90%空闲的车辆，是城市病的主要原因之一。车辆每年排放超过1亿吨的二氧化碳，占全球温室气体整体排量的1/3，是全球变暖的重要贡献方。

另一方面，当前汽车技术在安全、绿色、成本方面无法满足全球人口的出行需求，导致发展中国家有数十亿贫困人口负担不起也享受不到现代出行方式的便利性。

随着全球各国在气候变暖议题达成一致行动方案，以电动汽车为代表的新能源汽车替代传统燃油汽车迫在眉睫，发展迅速。在这个过程中，以软件和芯片为代表的数字化和智能技术，也将彻底赋能汽车、交通网络和出行服务，汽车产业正在经历类似功能手机被智能手机替代的破坏性式创新（Disruptive Innovation）过程，包括动力系统从内燃机到电动机的变革，从人工驾驶为主向自动驾驶模式的演进，从拥有和自驾私家汽车到享受安全、便捷、经济的出行服务方向拓展。

新一代的出行产业是汽车、IT、通信、交通、能源产业的大融合，当前汽车产业的领先公司和国家将会发生逆转，产业会发生大规模转移和重构，是全球第四次工业革命的王冠。



2. 新出行产业的“三零”愿景：零排放、零事故、零成本

我们认为零排放、零事故、零成本是新一代新能源智能汽车和交通出行系统的典型特征。

首先，全球电动汽车（包括插电式混动汽车和纯电动汽车）对燃油车的替代可以实现出行的零排放。2020年以来，售价10万元人民币左右，单次充电行驶里程400公里以上的电动汽车开始规模上市，可以完全对标主流燃油汽车的性能和使用成本，标志着电动汽车已经到达超越燃油车的临界点。预计2022年中国电动汽车产销量突破600万台，提前三年完成了新能源汽车在新车销售占比超过20%的目标。随着全球能源危机加剧，传统石化能源价格剧烈波动，电动汽车替代内燃机汽车的步伐将会进一步加快，2030年在全球范围停产内燃机汽车的目标已经指日可待，零排放的电动汽车的全面普及将会使全球石油消耗量减少约30%，将成为减缓全球变暖和实现巴黎气候议定书目标最重要的成就之一。

其次，智能汽车的自动驾驶能力能够带来出行安全性的极大提升，最终达到零事故出行的愿景。目前，全世界每年有135万人死于车祸，相当于每24秒就有1人因交通事故丧命，另外还有2000万至5000万人受到非致命伤害。与车祸相关的支出约占全球GDP的0.5%。

根据中国交通事故深度调查（China In-Depth Accident Study, CIDAS）项目对8000起交通事故的统计，81.5%的交通事故是由驾驶员因素引起的，包括驾驶员主观忽视交通规则等违章行为，也包括驾驶员因为能力限制，没有留意其他交通参与者行为和驾驶技术导致的错误驾驶行为，还有外部因素导致的无法感知其他车辆的危险驾驶行为和天气因素等。自动驾驶技术可以通过多模式和冗余传感器达到对环境和交通参与人的高精度和高可靠性感知，解决因为环境因素、驾驶员能力和疲劳等问题导致的感知失效，也可以完全避免驾驶员忽略交通规则

等主观行为导致的驾驶错误。

根据Waymo公司的最新研究，Waymo的AI司机能够规避75%的碰撞事故发生，减少93%的严重受伤，均高于理想状态下人类司机模型的62.5%和84%。随着辅助驾驶和自动驾驶系统的进一步完善和安装，零事故的出行体系指日可待。

第三，电动汽车的维护保养和行驶成本相比燃油车可以大幅度降低，甚至趋近于“零”。首先，燃油汽车的运动部件超过2000个，而电动汽车的运动部件少于200个，电动汽车在生命周期内的保养成本不到燃油车的1/2。电动汽车的电机系统、传动系统和车身设计可以支撑160万公里的运行，在正常行驶场景下，电动车的动力电池可以支持50-80万公里的行驶，电池寿命到期之后也可以通过更换电池持续使用，所以电动汽车的使用寿命是燃油汽车的10倍以上，其生命周期内的折旧费用也可以减少90%。



Replies to @Gfilche

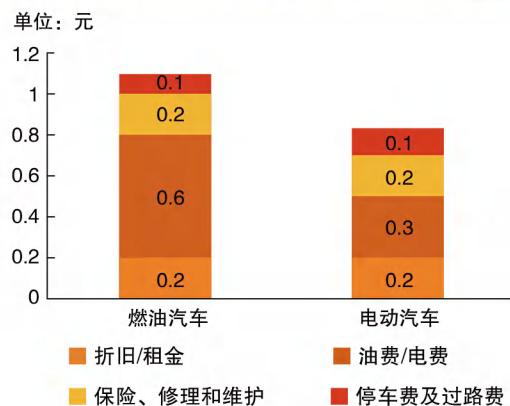
Model 3 drive unit & body is designed like a commercial truck for a million mile life. Current battery modules should last 300k to 500k miles (1500 cycles). Replacing modules (not pack) will only cost \$5k to \$7k.

9:18 AM · Apr 13, 2019 · Twitter for iPhone

在使用成本方面，今天，电动汽车的每公里出行的能源成本为0.2-0.3元，相当于燃油车能源成本的1/4-1/6，随着清洁能源和电网资源的有效利用，有望进一步降低。

因为电池占电动汽车成本的40%-60%，所以电池技术的持续改善，包括能量密度、可充电次数、容量衰减等，都可以直接降低电动汽车的电池使用成本。电动汽车也可以作为一种储能设备配合智能电网来调峰消谷优化电网使用效率，尤其可以有效利用太阳能、风能等绿色能源，进一步降低充电的成本。基于换电技术，电池换电服务（Battery-As-A-Service）带来了另一个维度的降低出行成本的方法，尤其在商用车领域，换电服务可以降低因为充电时间带来的服务中断，可以通过大规模的标准化降低电池采购成本，并且也可以优化电池维护和保养，换电也可以更好的利用电网的价格低谷。综上所述，电动汽车的折旧成本、维护成本和使用成本都相当于传统燃油车的1/10，在共享出行的场景，每公里成本从今天的1元左右，降低到0.1元。

图：电动化能进一步降低网约车每公里成本



图：电动汽车相较燃油汽车优势明显



参考：<https://ecocostsavings.com/electric-car-cost-per-mile/>,
<https://news.energysage.com/electric-vehicle-charging-cost-vs-gas/>,
<https://auto-time.36kr.com/p/942283254635653>

品牌	技术名称	描述	价格（元）
特斯拉	AP	基础版辅助驾驶功能	免费
	FSD	完全自动驾驶能力	64000
蔚来	ACC	自动泊车等	15000
	NOP	车道保持、召唤等	39000
小鹏	NAD	点到点自动驾驶	680/月（订阅）
	XPILOT 2.5	自动泊车等	免费
小鹏	XPILOT 3.0	NGP自动导航辅助驾驶等	标准36000；交车前够2000；年度订阅12000
理想	驾驶辅助系统	全速域自适应巡航等	免费

制图/网易科技

共享出行的另一个主要开支是司机费用，未来自动驾驶模式趋于成熟，AI司机可以替代人类驾驶员，按照目前蔚来的自动驾驶订阅模式，每月为680元，特斯拉的自动驾驶订阅每月为1400元，假设共享出行每月里程为6000公里，则每公里的AI司机费用约为0.01-0.02元。综合上述数据，通过电动化和自动驾驶，共享出行每公里的综合成本可以从今天的1.5元降低到0.1元左右，百公里出行的成本为10元，折合为一杯冷饮的价格，如果是多人一起出行，则成本进一步降低，和消费者使用水、电、网络服务的使用体验并无差别。

从历史来看，技术革命在增进人类福祉上发挥着最重要的作用，让大量现代生活必须品的使用成本大大降低，变为人人可获得的公用事业（utility service），例如：自来水网、电网、公路、铁路的大规模建设，成为公用事业，是第二次工业革命的象征；个人电脑、智能手机和无所不在的互联网让数字基础设施成为公用事业，是第三次工业革命的象征。新的智能汽车和出行服务革命是由数字技术即软件、芯片、人工智能算法和电池能源技术驱动的变革，本质上都有零成本的边际效益，廉价、舒适、绿色和安全的新一代出行服务将在这次变革中成为普惠全球人民的公用事业，是第四次工业革命的标志性产业。

3. 智能汽车和新的出行产业在教育、研究、标准、产业化方面的挑战和机会

3.1 软件定义智能汽车之转型与挑战

3.1.1 智能汽车软件是人类创造的最复杂的软件

汽车产业的转型面临的最大挑战之一是转型到软件定义汽车的新模式。智能汽车的软件代码量将超过5亿行，是有史以来最复杂的软件系统之一。相比之下，航天飞行器有40万行代码、波音787有1400万行代码，Android有1500万行代码，Windows有5000万行代码，Facebook有6200万行代码，现代汽车有1亿行代码。同时，智能汽车的软件也是现代软件的集大成者，包含了实时嵌入式软件、云计算软件、消费者软件、仿真软件、AI算法和大规模多智能体协同系统软件。而且，这么庞大和复杂的软件系统对安全性、可靠性要求也极高，因为软件的失效会导致性命攸关的严重后果。因此，智能汽车软件需要使用软件工程和软件测试验证等领域最先进的理论及实践。

3.1.2 自动驾驶是AI应用的集大成者

在自动驾驶系统中部署了几乎所有主要的人工智能技术，用于感知、路径规划、控制、人机交互等应用。传统的基于规则和专家知识推理的 AI 系统将被基于数据驱动的软件 2.0 方法取代。

软件 2.0 基于“Coding by Example”方法，不同于由程序员向计算机发出具体指令的软件 1.0 模式，软件 2.0 模式是通过向计算机提供示例数据集，由计算机按照机器学习方法给出解决方案，这更像人类在现实世界中学习解决问题的方式，通过老师监督不断学习各种知识并通过训练减少错误。软件 2.0 是软件产业的新兴领域，它不是简单地按照深度学习早期那种黑盒子模式，暴力训练模型参数，而是需要在理论上给出机器学习和模型训练的指导和论证，如：优化理论对训练中迭代算法的发展，知识图的引入和模型的可解释性等，以及相

应的软件工程 MLOps 理论，其核心目标是使得 AI 模型从训练到布署的整条端到端链路能够稳定、高效地运行在生产环境中，满足客户的终端业务需求，持续训练（continuous learning）开发、测试、发布和调优流程。

3.1.3 智能网联汽车是实现全自动驾驶的必由之路

现实世界的复杂性和大量的边缘案例导致单车的自动驾驶可能永远都不能完全实现，基于 5G V2X 等通信技术实现多车之间以及车和道路基础设施之间实时共享其位置、感知和意图，是实现L4和L5级无人驾驶的必由之路。

3.1.4 软件定义汽车(SDV)是智能汽车一切皆服务商业模式的基础

智能汽车产业也在进行商业模式的变革，一切皆服务的服务订阅模式（Everything-As-A-Service, XaaS），即电池、驾驶体验、数字内容、自动驾驶和维修保养等业务的全面软件化和订阅化，已经成为了大部分车企的共识。



要支持XaaS商业模式，智能汽车的平台需要进行硬件和软件功能的分层抽象和解耦，演变为轮子上的计算机平台，智能汽车内部的功能模块要通过面向服务的架构（SOA）提供统一API，汽车功能的开发将会演变为基于统一电子电气架构平台进行的软件功能开发，新车的开发周期从而可以缩短到1-2年，而且汽车的功能将会在出厂之后，通过软件升级通道继续提升，导致汽车越用功能越先进的反直觉现象。但是汽车软件对实时性、确定性和安全性的要求远高于消费者软件，如何融合汽车软件和消费者软件的开发工具和开发方法，在确保可靠性、安全性基础上提供软件定义汽车的能力，是对车企软件转型的最大挑战之一。

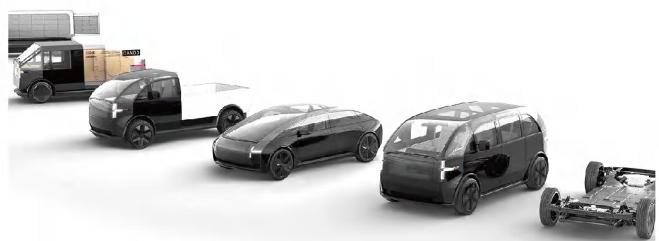
3.1.5 电动滑板底盘和线控技术将会重塑汽车产业的生产制造

传统燃油车企的竞争力核心是底盘技术，包括了发动机、变速箱、悬挂、刹车和转向机构。头部企业和顶层供应商在底盘技术领域有上百年的技术经验积累，对后发者是一个几乎不可逾越的屏障。而电动汽车的出现完全改变了底盘的设计。首先，底盘本身和电池要高度融合，在乘用车领域，通过电池和车身更紧密的融合（Cell To Body），来提高功率密度，提高车身安全性和

乘坐舒适性，在商用车领域，则出现了标准化的换电仓，通过换电来满足全天候运营的需求。

其次，电动车的驱动、刹车和转向功能，可以大量利用电动机100多年来在其他工业领域的使用经验和技术，完全基于数字化的理念和技术来设计实现，即完全软件化，通过调用软件API来控制，成为全面线控化的底盘（Drive by Wire, Brake by Wire, Steer by Wire），甚至还出现了轮毂电机这样的高度集成化的设计，使驱动、转向、刹车、悬挂系统和电池系统进一步解耦。

标准的底盘和线控技术，将底盘和车身设计彻底解耦，出现了大规模、低成本、标准化的滑板底盘供应商和面向应用场景的整车集成商，基于软件定义方式来开发定义汽车的基础驾驶能力、智能驾驶能力和信息娱乐功能，让汽车的生产更加类似于智能手机：电动汽车代工企业会提供基于标准软件API的底盘和计算机系统，而汽车企业则会全面软件化，面向用户群体提供不同的产品设计和软件组合，这将彻底改变汽车产业的面貌和格局。



滑板底盘的优势：独立性、通用性、大空间

独立性

- 上部车舱和滑板底盘平台可独立开发，降低研发成本，缩短研发时间



通用性

- 板式底盘平台能适配多种不同的车型，使车型开发多元化



大空间

- 滑板式底盘取消机械连接，为上部车舱保留更多空间



3.2 智能汽车教育转型的巨大挑战

自动驾驶“人才荒”是结构性的，是“质”与“量”的双重缺位。据中国汽车工程学会人才工作部部长在新能源汽车国家大数据联盟2021年高峰论坛中发布的《智能网联汽车产业人才需求预测报告》透露，基于其工作组调研，目前我国智能网联汽车行业现存人才约在7.2万人左右，若按照低速发展、中速发展和高速发展对于智能网联进行预测，至2025年该行业人才需求约在9.2万到11.6万之间，在扣除每年进入这一领域的毕业生人数后，仍存在1.3万至3.7万人的研发人才缺口。除了研发人才，行业所需的其他支撑人才（例如维修服务工程师）有更大缺口。

根据教育部公布的数据，2020届全国高校毕业生人 数达874万，相比去年增长40万，创历史最高纪录，其中就业难上加难的是汽车类专业毕业生。中国人民大学中国就业研究所与智联招聘发布的《2020年大学生就业力报告》显示，汽车、摩托车行业位列高校毕业生就业景气较差行业第9名。究其原因，汽车工程专业教授的绝大多数课程是即将被淘汰的燃油汽车平台技术，如发动机、变速箱、底盘等，课程以机械电子为主，软件只是辅助角色，并且不包括通信、实时系统软件、AI、自动控制等智能汽车急需的软件知识。从2018年开始，有一些企业纷纷不再招聘车辆工程专业学生，改为大量储备计算机人才，其中以一汽-大众为代表。2019年9月，一汽-大众表示2020年暂无车辆工程、工业设计（车身结构）等多个专业学生招聘计划，这引起业内热议。

目前国内高校主要是通过参加自动驾驶相关科研项目来培养自动驾驶人才，以硕、博研究生为主，本科生、博士后相对较少。当下的自动驾驶人才大多来自于计算机系、自动化系、电子系、车辆工程等学科专业。这种人才培养模式无法解决社会上普遍存在的自动驾驶用人荒，也不能取得最佳的人才培养效果。

为了填补智能汽车软件工程师的市场缺口，职业培训也扮演了重要的角色，全国有多家职业培训学校提供自动驾驶的课程培训，但是水平参差不齐，也缺乏统一

的课程和认证。

综上所述，智能汽车的人才培养是制约车企软件转型的最迫切因素。

从全球范围来看，欧美发达国家已经率先开展了智能汽车的学校教育和职业教育。例如，美国的MIT，UC Berkeley, Stanford, CMU等一流高校都提供自动驾驶相关课程。

美国的Udacity从2015年起就开始提供在线的自动驾驶课程，结合了视频讲解，在线编程和仿真技术，为学生提供了一个高互动性的教育平台，虽然全套课程收费高达数千美元，但是仍然供不应求，吸引了大量的在职学生，以至于需要申请才能加入。

另一个在线教育平台Coursera也提供多伦多大学的自动驾驶课程，有超过13万名学生加入学习。在线教育平台通过交互式的学习和测试平台帮助在职工程师在6-12个月内掌握自动驾驶的基础软件和算法，可以短期内为车企提供可用人才。



培训机构	课名及网址	联合授课	课程介绍
Coursera	Self-Driving Cars Specialization https://www.coursera.org/specializations/self-driving-cars	加拿大 多伦多大学	面向已经在自动驾驶行业人士。 建议学习时间，每周 5 小时，7 个月，月费 59 美元。
Udacity 优达学城	Self-Driving Fundamentals: Featuring Apollo https://www.udacity.com/course/-self-driving-car-fundamentals-featuring-apollo-ud0419	百度	面向初学者,建议 4 周完成,免费。 学习自动驾驶的基础模块,特别是百度阿波罗架构。
Udacity 优达学城	Intro to Self-Driving Cars https://www.udacity.com/course/intro-to-self-driving-cars-nd113	N/A	面向中等程度学习者,建议每周学习 10 小时,4 个月完成,月费 1356 美元。学习自驾基本概念和需要的编程技能。本课并非预备自驾工作岗位技能的一门培训课程。
Udacity 优达学城	Self-Driving Car Engineer https://www.udacity.com/course/self-driving-car-engineer-nanodegree-nd0013	奔驰公司 Waymo	面向高级程度学习者,为进入自动驾驶行业提供扎实培训。学费 933 美元。建议每周学习 10 小时,5 个月完成。这个自动驾驶工程师微型学位包含教学内容和多个实习项目,横跨自动驾驶能力需要的各个领域。
SAE 国际汽车 工程协会	Robotics for Autonomous Vehicle Systems Bootcamp https://discover.sae.org/av-robotics-bootcamp-google	Clemson University, Argo AI	12 周的训练营,包括自驾基本概念,编程技能,和一个实习项目。
深蓝学院	无人驾驶技术线下实战特训营 https://www.shenlanxueyuan.com/course/532	N/A	5 天线下教学实践,2 个月线上答疑辅导,从仿真环境到实车验证。 学费 15999 元人民币。

业内有建议高校应该设置独立的自动驾驶专业。因为培养自动驾驶人才所需的教学大纲、课程体系、师资、教学实习实验设备、产业环境等都与现有专业不同，需要重新组织才能满足专业建设的要求。

3.3 智能汽车研究的挑战

智能汽车的发展提出了大量需要做理论研究和应用实践研究的课题，但是因为智能驾驶的数据、算法、计算平台和软件工程的标准还不成熟，导致学术界和产业难以对标，相比头部智能驾驶企业，学术界也缺乏智能驾驶研究的基础平台，在算力和数据方面远远落后于企业，因而目前领先的自动驾驶研究都被头部企业垄断，不利于产业的健康发展。

自动驾驶算法的研发是围绕场景和数据进行的，头部企业已经构建了包括路测车辆采集数据，数据筛选和清洗，数据标注和扩增，模型训练，仿真环境和影子模式验证，模型发布的完整数据循环，如下图所示。



在IT技术领域，学术界在软件工具和计算基础设施方面和工业界的区别并不大，尤其是公有云提供了非常方便的基于使用量的计费模式，基本消除了双方的算力差别。可是在智能汽车领域，企业已经进入了自动化的数据大工业时代，而学术界还在手工作坊阶段。构建面向学术研究的自动驾驶基础研究设施迫在眉睫。下面是目前存在的问题：

- 1.自动驾驶系统是一个软件2.0系统，意味着大量的

高质量数据是开展研究所必须的。这些数据来自路测采集的传感器数据，（例如：相机、lidar、雷达数据），足够多样化的条件，（例如：城市、乡村、不同的天气和拥堵情况），多维度的数据标注，（例如：做感知研究所需要的实例分割和边界场景、做预测需要的动态场景等）。头部的自驾企业如waymo，motional等提供了比较好的数据集，但是这些数据也存在如下的问题：

• 各国的交通基础设施，驾驶行为存在巨大差异，目前的数据集多样性不够；

• 数据集的数量有限，因为这些数据集只是车企为了获取外部研究思路的一种方式，并不具备服务产业的目的；

• 数据集和算法有较强的耦合性，在公共数据集上练出来的模型，移植到不同传感器配置的车型上，往往需要重新训练，大大增加了研究到工程落地的时间和成本。

2. 据兰德公司估计，自动驾驶技术需要测试110亿英里（=176亿公里）才能获得足够好的性能和消费者信心，物理车辆直接测试如此规模的里程无论从成本和场景多样性上都不可能，所以必需采用仿真技术，事实上99%的自动驾驶测试是在仿真环境中验证的，但是仿真环境也存在如下的问题：

• 欧美汽车产业从标准到商业仿真软件系统，控制了仿真产业链，商业智能汽车仿真系统，成本昂贵，需要数以百万人民币的商业许可；

• 需要购置较高端的GPU服务器，增加了研究成本；
• 仿真环境的地图和接口自驾软件和硬件系统的接口缺乏统一标准；

• 缺乏仿真数据和场景的标准，不同的仿真系统数据格式不统一，无法相互比较数据质量。

3. 路测数据和仿真数据还不能完全取代实际的物理环境测试，虽然有不少智能网联汽车测试场，但是只能服务企业需求，缺乏像美国密歇根大学的MCity这样面向研究的测试场地，测试场地也不提供对应的仿真环境和公共数据集。

4. 目前自动驾驶算法研究还有一些理论上的问题亟待解决，比如感知模块中存在的问题有小感知（多模型多任务）和大感知（一个模型多个任务）、从距离视角（range view）到鸟瞰视角（BEV）的感知和融合、多模态传感器的融合以及时域融合等，下游的预测、规划决策模块存在的问题包括上游感知不确定性的处理、交通环境中的智体交互如何建模、不遵守交通规则的行为

如何预报以及安全危害的场景如何检测等。

5. 算法竞赛也是推进AI研究的有效方式，竞赛的前提是要有场景和性能指标。自动驾驶领域已经有不少驾驶数据数据集和指标，例如：nuScene，Waymo Open Dataset，华为ONCE dataset等，但是还缺乏端到端的自驾性能评估机制。基于开源仿真软件Carla实现的leaderboard自驾竞赛，提出了一套端到端自驾打分标准，虽然还比较粗糙，但也已经能够较好评估不同算法架构的自驾能力，是一个值得继续探索的方向。

3.4 产业缺芯少魂战略选择的挑战

智能驾驶是综合性全产业的竞争，尤其是ICT领域的体系竞争，包括芯片、传感器、电子电气架构、基础软件、数据技术、算力平台、5G通信、软件工程等。以当前我国的产业准备度而言，确实面临缺芯少魂的问题。解决这些问题犹如攀登珠穆朗玛峰，但是也存在南坡和北坡的选择问题。南坡是在既有技术路线上攀登，北坡则另辟蹊径，使用另类架构和技术。

3.4.1 面向算法定义的AI芯片架构和芯粒集成

自驾计算平台的核心是AI芯片，当前通用AI芯片厂家通过多年的技术积累和生态投资，在生态通用性上做出了护城河，既能较好地支持新的AI算法模型，训练和推理的性能也不错，但是其每个版本都需要考虑上百客户的需求，架构必然是复杂的，需要先进的半导体制程来容纳更多的特性，必然导致其流片成本、工具成本增高，例如：nvidia的Xavier车载计算芯片芯片投入了2000名工程师做了4年开发，总投资估计超过20亿美元。

虽然one-size-fits-all的AI芯片架构通用性最好，但是以效能和成本衡量的性价比是最低的。因为AI算法模型对AI芯片的加速器架构有决定性影响，例如：片上内存、带宽、加速器的算法内核等，如果能放弃通用性，而只为少数的几种算法架构优化，必然可以降低芯

参考：<https://mcity.umich.edu/>,

<https://iot.ofweek.com/2019-04/ART-132216-8120-30323836.html>

<https://blogs.nvidia.com/blog/2018/01/07/drive-xavier-processor/>

片的复杂度，可以选择较低的制程，工具投资也会降低，从而达到较高的性价比。例如，Tesla的FSD自驾芯片，投入就远远小于Xavier，但是通过算法和芯片的联合优化，性能和成本要优于Xavier，基于14nm和144 TOPS算力的FSD HW 3.0的芯片已经展示出初级的L4自动驾驶能力。

芯粒是具有标准接口和特定功能的已经制成的未封装的裸片（die），他们可以和其他芯粒一起被封装为更大的系统。实际上，芯粒是传统IP的物理表现形式。如下图所示，传统IP集成为SoC的过程并不能节省EDA设计和流片成本，而预先制造的芯粒则不需要产生额外的EDA工具、工艺设计和光罩成本，而且流片速度更快。以芯粒方式设计芯片，就相当于设计一个基于硅片集成的硬件板卡，大量的货架性的裸片（die）都可以买来集成，用户只需要关注系统功能的达成，设计的成本和质量保证工作都分摊给了封装工厂。除了芯粒集成的密度更高，芯粒集成和传统的PCB设计的经济学原理是完全一样的，是后摩尔时代工艺水平停滞情况下，通过封装方式的变革提升芯片性能，降低设计和投片成本，加快芯片设计速度的重要技术路线。

一次性 投入成本 (NRE)	方案				
	开源工具	开源IP	无光罩光刻	外包	芯粒
EDA工具	✓	✗	✗	✗	✓
IP成本	✗	✓	✗	✗	✓
设计成本	✗	✓	✗	✓	✓
光罩成本	✗	✗	✓	✗	✓

芯粒技术和算法定义的AI芯片可以改变自动驾驶芯片设计的格局。把AI芯片的通用CPU计算部分和定制化要求较高的加速器分别实现，两者的裸片采用芯粒工艺在封装厂做DIE互联和封装，则可以进一步降低流片成本和周期。而且通用的CPU计算部分不需要反复流片，可

以只付出封装成本而支持不同的加速器，从而获得比较好的通用性。

	流片 成本	工具 成本	通用性	性价比
通用 AI 芯片	高	高	高	低
算法定义的 AI 芯片	中	低	低	中
CHIPLET+ 算法 定义的加速器	低	低	中	高

3.4.2 自动驾驶统一数据集和大模型

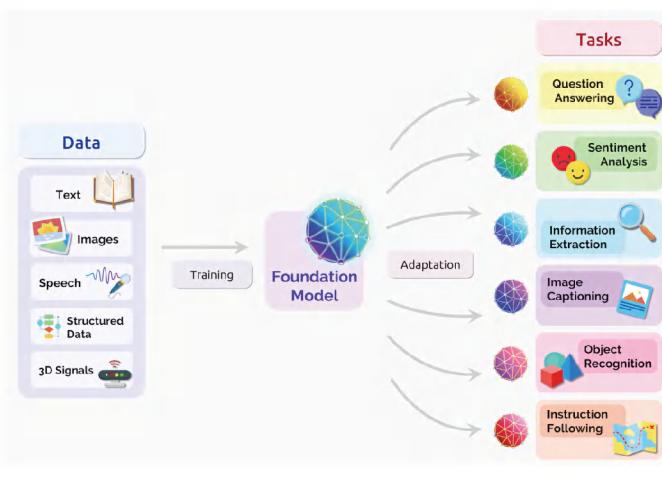
自动驾驶本质上是一个软件2.0的应用，不能直接使用软件1.0的架构去做分层，而应该以数据和模型的视角来定义自动驾驶应用数据和应用抽象。这个架构可以映射到产业链的结构中去，把目前自动驾驶从烟囱型的各自为战转变为依据技术架构和产业竞合为原则的层次化抽象，形成自动驾驶的“OS”和应用生态。

AI领域已经进入预训练大模型竞争的时代。尤其在自然语言理解领域和图像领域，有海量的互联网数据集可以使用，大模型可以通过无监督或者弱监督学习来利用海量的互联网数据集，参数已经达到千亿规模，使得大模型达到很高的通用性。众多的下游任务只需要局部调优就可以达到很高的性能，例如GPT3大模型可以用于做翻译、文本生成、语言理解等多样的自然语言应用，而且大模型可以同时学习多模态数据，（例如：语言和图像和其他结构化数据），用于多模态的识别，甚至是内容生成，（例如OpenAI的DALL-E实现了文字到图像生成），在视觉艺术创作，人机交互领域具备很大的应用潜力。预训练大模型和生长在其之上的应用，可以简单映射为软件2.0的操作系统和应用程序。

参考 : <https://cleantechnica.com/2019/06/15/teslas-new-hw3-self-driving-computer-its-a-beast-cleantechnica-deep-dive/>

<https://www.intrinsix.com/blog/the-impending-hsoc-revolution-potential-solutions-to-the-asic-cost-barrier>

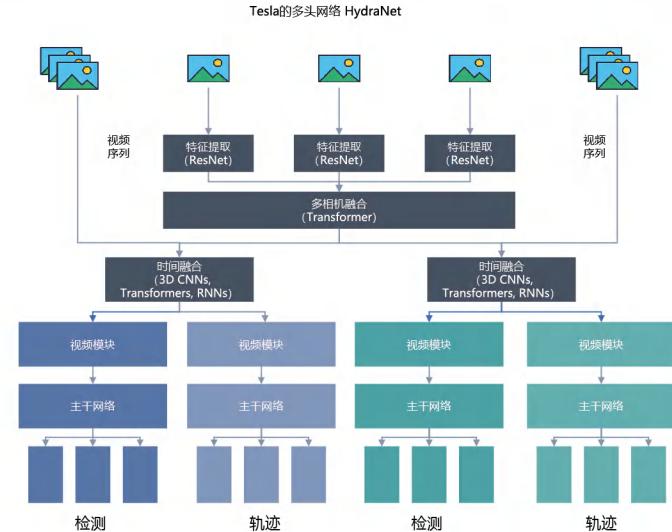
<https://arxiv.org/pdf/2108.07258.pdf>



在自动驾驶领域，训练大模型的问题要比自然语言理解和一般性的图像识别和分类更具挑战性。自动驾驶的感知任务需要输出非常准确的位置、速度等信息，对数据集的要求远高于其他领域，传感器参数（例如：分辨率、动态范围，像素密度等）对模型有直接影响。多模态的传感器需要做校准和融合，因此传感器安装的位置角度等对模型也有直接影响。基于此，自动驾驶的模型训练难以利用公开领域的数据素材，数据规模不能支撑训练大模型。其次，已经开源的数据集，彼此之间传感器的规格、配置以及标注方法不统一，缺乏标准，在各数据集上训练出的模型，也不通用。

但是对头部自驾企业而言，这恰恰是竞争优势，例如，Tesla研发的HydraNet使用了基于transformer架构的多层次多结构大模型，基于这个大模型实现了上千个下游感知任务，如：多传感器融合、时空融合、语意分割、目标识别和跟踪等。随着不断加入新场景，大模型的能力与日俱增，下游任务也同时受惠。Tesla打造的这个HydraNet就是自动驾驶软件2.0系统的操作系统。

大模型的兴起也导致业界对可能出现的AI能力垄断



的担忧，所以对应商业闭源版的预训练大模型，全球AI社区也非常快速的推出了对应的开源版本。

	闭源版本	开源版本	数据集	性能
自然语言	GTP3	GTP3-NEO, BLOOM	开放语料库	可以对标商业版
图像生成	DALL-E	Stable Diffusion	开放图像数据集	可以对标商业版
BEV多头模型	Tesla HydraNET	BEVFormer, BEVFusion	nuScene, 数量少, 无标准	无法对标商业版

这样做能够有效节约算力成本，因为大模型训练需要海量数据和算力。据统计大模型的训练成本已经突破百万美元的水平，OpenAI的GTP3语言大模型的算力成本至少是460万美元，Tesla的DOJO训练集群的算力也可以排进全球HPC排行榜的前10名。开源预训练大模型可以有效整合社区的数据、人才和算力，服务于社会和产业，拉平和平台企业的竞争能力。

参考：<https://www.thinkautonomous.ai/blog/how-tesla-autopilot-works/>

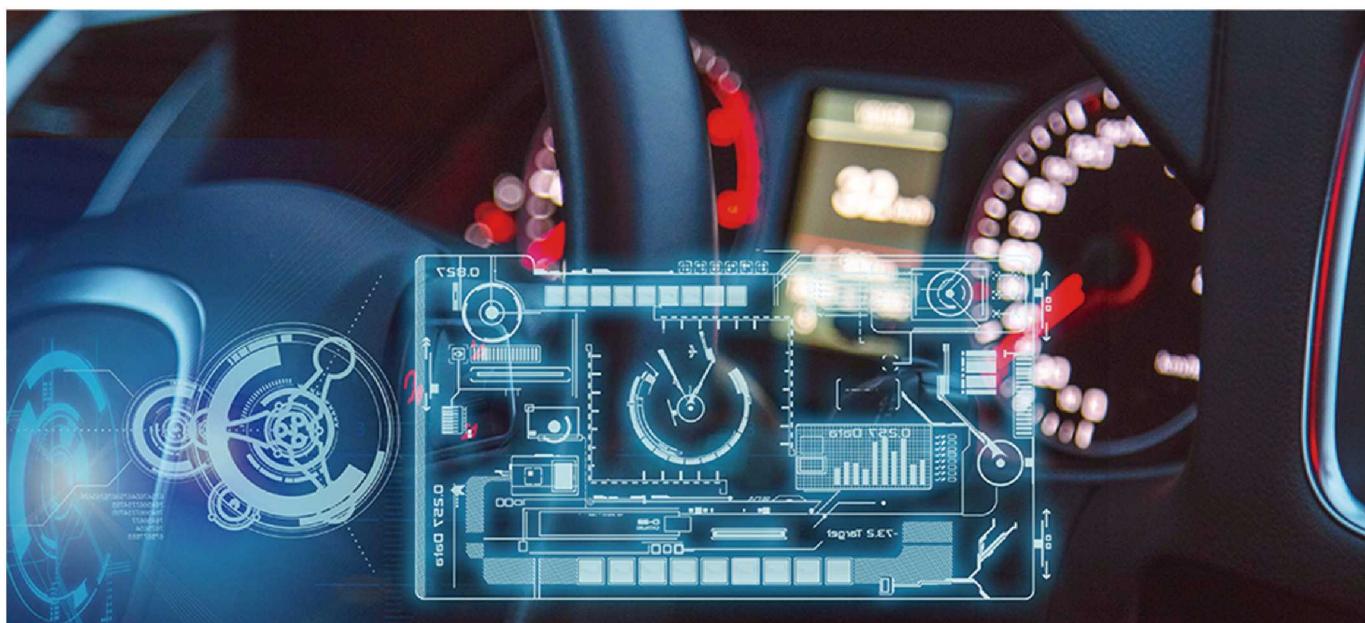
<https://lambdalabs.com/blog/demystifying-gpt-3/>,

<https://www.hpcwire.com/2022/08/16/tesla-goes-its-gpu-powered-ai-super-is-dojo-next/>,

<https://www.economist.com/interactive/briefing/2022/06/11/huge-foundation-models-are-turbo-charging-ai-progress>

自动驾驶领域的开源大模型主要是由学术界通过有限的开源数据集，例如nuScene数据集，训练出来，但是开放数据集规模有限，而Tesla已经有百万辆车辆在贡献数据集，估计已经获得52亿英里（=80亿公里）以上的数据集，业界领先的Waymo自驾公司大概拥有3000万英里（=4800万公里）数据集，而用于学术研究的公开数据集规模不到10万英里（=16万公里）。数据集的规模太小，加上不同数据集不通用，导致学术界难以训练出和Tesla HydraNet对标的大模型。

要解决数据集的问题，不仅仅要解决上述传感器和数据标准问题，还需要建立产业对自动驾驶大模型操作系统的共识，通过开源和开放数据集来提升产业相对头部企业垄断企业的竞争力。这种策略在开源Android对标iOS，开源K8S对标私有云计算平台领域都得到过很成功的实践。尤其智能驾驶成为国家之间产业竞争的焦点，如果头部企业和国家产业政策结合，可能会更加快速的形成平台的垄断，例如：Tesla可能开放其自驾能力给其他车企，而中国企业可能被排除在外。



参考：<https://electrek.co/2021/01/27/tesla-talks-automakers-licensing-self-driving-software-elon-musk/>

3.4.3 可信安全编程语言和中间件

汽车操作系统可以包含底层做硬件抽象和内存与进程管理的操作系统，但是一般特指提供应用编程环境的中间件。

AutoSAR是汽车行业广泛应用的中间件平台，基于定制C++语言编写，广泛应用于车控平台的开发，也有完善的车规认证产品，但是在支持软件定义汽车和自动驾驶方面，面临挑战，也没有影响力很大的开源项目。

ROS（包括ROS1和ROS2）开源项目是2008年诞生于斯坦福大学的世界领先的机器人计算平台，它借鉴了历史上机器人的软件架构，构建了一套基于pub/-sub分布式消息机制的计算模型，在机器人产业得到了广泛的接受和应用。大量的传感器、执行器都提供ROS标准API，ROS还紧密集成了感知、规划、控制的算法框架，基于ROS生态，很容易构建不同形态的机器人，也被很多车企改造用于支持SDV和自动驾驶应用开发。ROS采用C++开发，也有厂商提供车规级的解决方案。但是ROS开源项目的治理机制比较封闭，技术演进速度也较慢。

汽车软件对安全性和性能要求极高，这就创造出两难的问题，因为高性能的C/C++语言有不可解决的内存使

用安全问题，例如：重复内存释放，空指针等，尤其在多CPU多线程并发场景下，会造成更多的内存使用带来的软件缺陷。据微软统计，70%的微软软件安全缺陷都是由于内存问题引起的。

Rust语言是新兴的安全编程语言，它提供给开发者安全使用内存的语义，例如：所有者（ownership）、借用检查（borrow checker）、生命周期（lifetime）等，编译器可以对内存使用做严格的检查，编译通过说明开发者的内存使用基本是没有问题的。Rust作为C/C++语言的替代，已经进入到Linux内核开发、网络服务，数据库等系统开发领域。AutoSAR组织和SAE也都开始关注Rust语言在汽车软件领域的应用。面向未来，采用Rust语言开发基础的汽车软件操作系统和中间件是大势所趋。

DORA是基于Rust编程语言开发的面向SDV和无人驾驶的数据流计算平台。相比ROS、AutoSAR，还引入了数据流控制面和数据面分离，可定义数据面的设计架构，是目前面向自动驾驶和机器人领域比较好的计算中间件开源项目。

	编程语言	消息中间件	数据流计算	安全性	开源生态	网络性能	车规认证
AutoSAR	定制版 C++	SOME/IP	无	中	低	TSN	支持
ROS	C++	DDS	中	低	高	以太	支持
DORA	Rust	多种选择	高	高	中	高	暂无

参考：<https://msrc-blog.microsoft.com/2019/07/22/why-rust-for-safe-systems-programming/>

3.5 产业标准的挑战

智能汽车进入软件和算法、算力的竞争时代，必然导致了头部企业可以高效整合研究和工程落地，通过垂直整合带来迅速超越友商的功能和性能差异性，导致已经在互联网发生过的，平台企业垄断市场的高度可能性。

这个问题的复杂性在于，智能汽车转型事关万亿美元的新市场的经济利益，鼓励领先企业迅速占领市场是必须的，但是和互联网不同，智能汽车作为一个全新的技术，一旦监管跟不上，可能会出现严重的公共安全事故，反倒会影响消费者对自动驾驶技术的接受。例如，20世纪初，美国开始大规模建设铁路，但是因为对高速运行的火车的安全性认识不足，导致了严重的脱轨等安全事故，反倒出现了火车使用的倒退，直到压缩空气刹车和信号体系等安全设施建设到位之后，火车系统才在全社会大规模普及。所以，对自动驾驶技术的安全监管反而有益于它的良性发展，逐步建立消费者的使用习惯和信心。

中国新能源汽车和智能汽车在产业政策引领下，获得了高速发展，新能源智能汽车的出货量占据全球70%以上，拥有完整的领先世界的供应链，世界一流的车路协同基础设施，还有大量新兴智能汽车品牌，势必成为未

来全球新能源智能汽车市场的重要势力。

另一方面，欧洲和美国过去百年中不仅建立了庞大的汽车产业，也是全球汽车标准话语权的制定者。汽车产业的软件化、智能化将会对智能汽车的标准产生巨大的影响，基于燃油车的标准体系不能满足电动体系架构的要求。

在软件领域，AutoSAR等标准在支持SDV和自动驾驶业务上力不从心。在自动驾驶领域，自驾能力认证和评估的标准和工具非常重要，成为各大标准组织关注的重点。不同于传统以机械电子为核心的硬件标准和认证体系，软件领域遵循事实标准，即软件平台是以开发者和用户数量来竞争的。历史来看，无论是消费者端的PC和手机，还是企业服务端的云计算，都不会出现三个以上的软件平台，因为软件的边际成本为零，领先的平台一旦获得竞争优势，后发的软件平台需要付出更高昂的成本来获取开发者和用户。

这对中国的智能汽车带来了超越的机会，即可以通过市场的体量和先发优势，迅速建立平台体系。目前智能汽车产业还处在各自为战，竞争大于合作的阶段，如何吸取ICT产业软件转型的历史教训，在智能汽车的基础OS平台上建立开放开源的平台，是极具战略价值的挑战。



4. 发展开源社区，融合产学研力量，构建统一软件和计算平台

无疑，如此巨大的技术、商业和产业链转型，对现有的汽车企业更多是挑战而不是机会，而非汽车企业又很难短时间具备汽车企业百年建立起来的生产制造、供应链管理、测试认证等基础能力。所以，我们需要对智能汽车和出行产业建立一个愿景共识，在关键性的瓶颈技术上面调动全社会和全产业的力量进行有效投资。

教育系统需要为汽车产业输送数以百万计的智能汽车软件人才，学术界需要提供基础理论和加速成果转化，产业生态需要合力解决卡脖子的缺芯少魂问题，同时，也要建立符合我们市场地位的标准体系话语权。

我们认为开源平台是整合教育、研究和产业力量的有效方式，开源社区是中立的、服务社会、产业和国家战略的。通过开源社区既可以通过开放协作的方式催化软件和研究创新，也能够通过平台的生态建立事实标准。开源平台还是目前逆全球化大势下全球还可以进行有效合作的国际性平台。

下面，将就开源社区和生态如何服务智能汽车的教育、研究、产业合作和标准制定来做逐一阐述。

4.1 开源创新力量和生态价值已经被IT产业反复验证

因为软件存在零成本边际效益，意味着用户的数量决定了软件生态的成败，越多的用户能够更多摊平开发的成本，也能获得更多的用户使用体验和需求的反馈，进一步提升软件的价值，从而进一步吸引了更多的用户。而开源是增加用户数量和获得用户反馈的最好手段，软件产业已经进入了开源生态竞争的时代。

- 开源是软件创新的最重要的方式。自动驾驶的软件根技术例如：编程语言Rust、Python，操作系统Linux，数据库、中间件ROS、DDS、DORA、Zenoh，AI计算平台Pytorch、Tensorflow、Mindspore、OneFlow、TVM等，都是通过开源打造的，通过国际合作进行的。中国参与开源项目的开发者已经接近8百万，居全球第二，但是在软件根技术领域的贡献还需要加强。

- 开源也是AI研究创新的最重要平台。顶级论文的代码都基本是开源的，开源数据集也成为AI研究的不可或缺的组成部分。头部企业例如：waymo，cruse，motional，argo都在利用开源数据集，影响学术研究的方向，获得最新最好的研究成果。

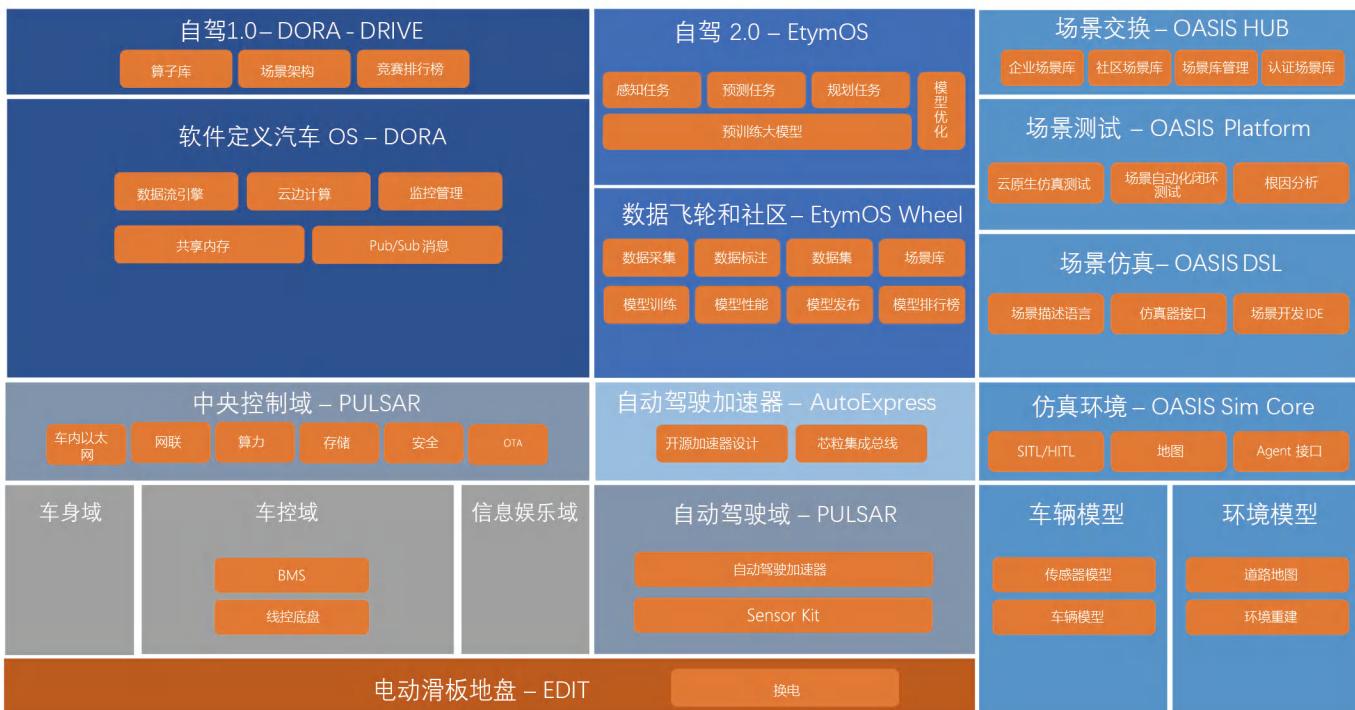
- 开源已经成为IT云计算产业的基础平台，包括云计算领域的Kubernetes，大数据领域的Hadoop、Spark、Flink，数据库领域的MySQL、Postgres、高斯等。IT产业已经高度云化、服务化，开源软件成为获取客户构筑生态价值的最战略性举措，业界领先的软件平台企业都把开源作为其技术和商业战略的一部分。微软Github几乎垄断了开源软件托管和协作，成为微软云计算领域的重要资产。Google的Android平台创造了千亿美元的收入，Facebook的Pytorch平台成为AI模型训练领域的事实标准。国内的鸿蒙、欧拉生态方兴未艾，蓬勃发展。

- 开源才能形成生态，成为事实标准。IEEE SA建立了自己的开源平台，希望在未来的标准制定过程中融入开源软件的成分和治理模式。Linux Foundation也并购了面向标准开发的开源基金会JDF，启动了OpenSSF做软件供应链的标准。LF Energy基金会做新能源领域的软件参考架构，Green Software Foundation建立软件碳排放的标准等。

参考：<https://www.chinadaily.com.cn/a/202208/30/WS630d631ba310fd2b29e74e75.html#:~:text=According%20to%20statistics%20from%20CSDN,participated%20in%20open%20source%20projects.>

4.2 基于开源建立全栈的智能汽车统一平台

CARSMOS元遨项目是面向产、学、研合作，构建统一智能汽车硬件、计算、软件和生态平台的项目群，下图是CARSMOS项目的立项范围。下文将逐一阐释各个领域的项目内容和规划。



4.2.1 EDIT EV项目：面向低速无人应用场景的电动滑板底盘

智能汽车产业是否会分化为标准化的滑板底盘结构和上层的应用开发生态是一个被热烈讨论的课题。电动滑板底盘的典型特征如下：

- ◎ 滑板底盘必须是“全集成+全线控”，从而实现上下分体式造车；
- ◎ 滑板底盘需具备“不仅上下分离解耦开发，更具备上下一体共同承载”的能力；
- ◎ 滑板底盘需具备“空间释放，成就移动大平层”的能力。

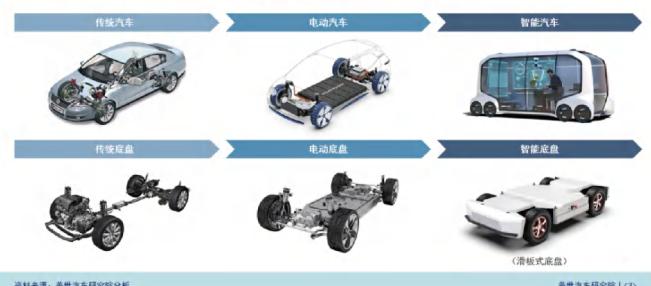
第一个特征也是滑板底盘最大的不同于纯电平台的特征。

在乘用车领域，因为动力性能和个性化设计是车企的核心竞争力，短期内不太可能走向统一的滑板底盘的技术路线。但是随着智能汽车商业模式从卖硬件进一步XaaS服务化，电动滑板底盘可能会成为未来的一种重要选择。

而商用车市场是为了使用成本做极致优化的，电动滑板底盘能通过标准化极大降低商用车设计和运营的成本，注定是一个重要的选择。开源的电动滑板底盘将首先在商用车市场得到接受，因为可以进一步优化供应链。在商用车服务领域，会出现类似IT产业走向云化服务化时候大规模采纳开源白牌硬件的产业供应链变革。

汽车底盘演变

电动化、智能化的发展对汽车底盘系统产生了巨大的冲击，在汽车从传统燃油车到智能电动车的演化过程中，底盘形态也随之发生了改变，集成化程度越来越高



资料来源：盖世汽车研究院分析

EDIT开源项目是一个电动车的滑板底盘设计，符合欧洲L7E标准（重型四轮机动车，载人干重<400kg，载物干重<550kg，若是电动车不包括电池重量，最大净功率<15kw）和中国M1标准（至少有4个车轮或3个车轮的乘用车，出厂确定的最大总质量不超过1吨，除驾驶员座椅外不超过8个乘客座椅），其主要目标是

- ◎ 对标MINI EV形态的商用车服务，充分模块化的设计，鼓励社区基于标准底盘构建不同的商用车形态，成为低速电动滑板底盘领域的事实标准。



- ◎ 集成换电标准。换电对商用车24x7的商业运行至关重要，EDIT通过创新的电池仓机械和电气结构设计，可

以快速安全实现换电能力，配合标准集装箱尺寸的换电站可以方便的实现移动换电或者模块化的换电站。

- ◎ 线控能力（Drive by Wire, Steer by Wire, Brake by Wire）能力是电动滑板底盘的标志性特性。EDIT提供了多样性的供应商选择，可以使用主流供应商的方案，满足低速商用市场的安全性需要，也提供基于开源实现的方案，希望和产业和学术界一起来打造对标商用方案质量的X by Wire 方案。

按照联合国欧洲经济委员会的分类标准，EDIT EV 支持以下车辆类别（当然也包括更低的级别）：

- ◎ L7带有四轮的低速(<45km/h)车辆，其空载质量不超过400千克（用于运载货物的车辆为550千克），其中不包括电动汽车的电池质量，并且其最大连续额定功率不超过15千瓦，载重（乘客）不超过200kg，载重（货物）不超过1000kg，例如：ATV、微型汽车、微型皮卡。

- ◎ M1用于载客的轻型车辆，除驾驶员之外，还包括不超过8个座位（比标准汽车大，例如：伦敦出租车/ E7型车辆8座位+驾驶员）。

L - 4 轮机动车，最高速度 45km/小时		
M – 至少4轮的机动车，用于 乘客运输		

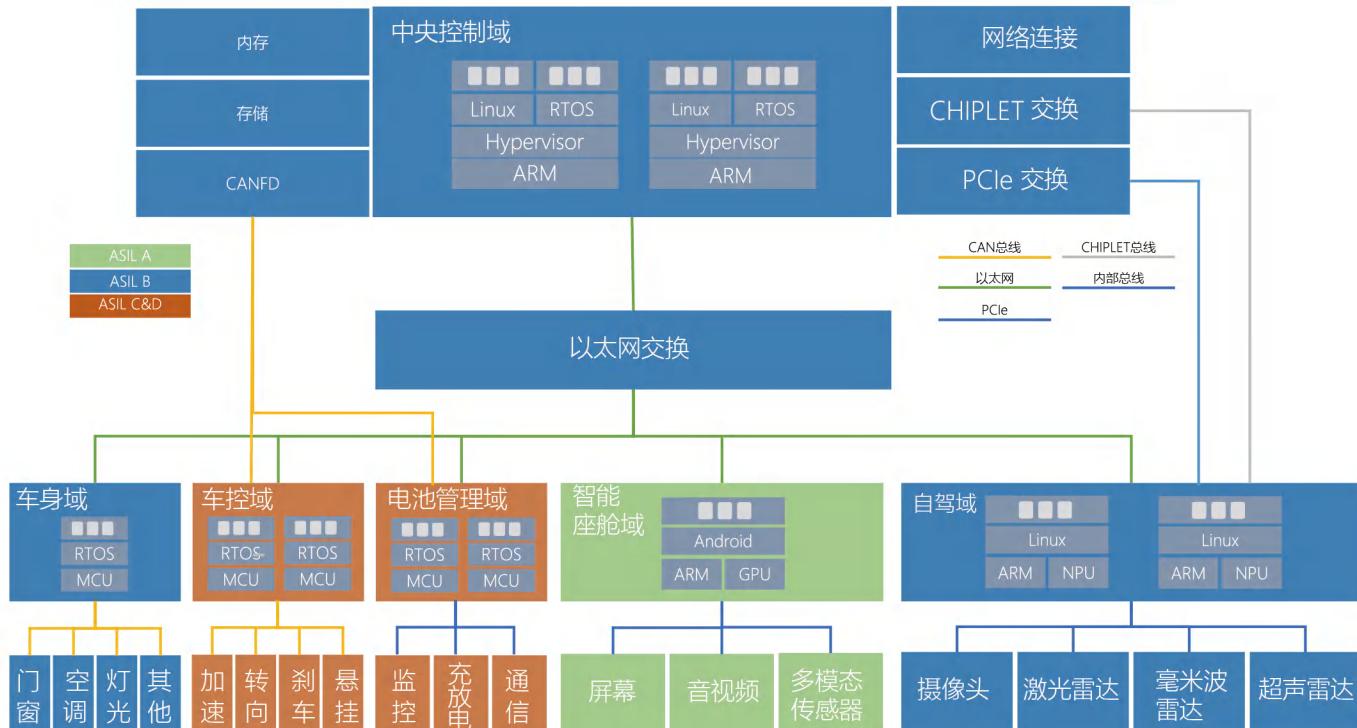
4.2.2 PULSAR项目：模块化、虚拟化、集中化的开放电子电气架构

电子电气架构（Electrical/Electronic Architecture）是实现软件定义汽车的关键性架构，终极目标是将今天以硬件为核心的嵌入式软件平台，转变为基于标准计算平台、标准化通信架构、标准化数据、标准化硬件与软件接口、云原生微服务和 DevOps 理念构建的新型计算平台。汽车工业现有功能安全的标准，例如：ISO 26262，

是经过行业几十年的经验积累的安全共识，所以必须遵守，但是这个安全标准和采纳 IT 软件和云原生应用开发方式产生了较大的矛盾。下表比较基于传统 ISO 26262 功能安全的汽车软件开发方法，和软件定义汽车所要求的电子电气架构的区别和演进路线。

	ISO 26262	PULSAR E/EA 架构
计算平台能力	lock-step MCU, 高可靠性，工艺成熟	通用计算平台，异构计算技术，共享服务器和智能手机的计算技术和生态。通过状态和功能分离的负荷分担等方式解决软件可靠性问题。
	性能低，难以满足智能驾驶舱和自动驾驶的算力要求	
通信能力	CAN BUS 提供点对点，确定性带宽和延时的通信能力	使用以太网统一传输技术，支持10-1000Mbps的灵活传输速度，通过VLAN和交换链路冗余等技术支持功能隔离和可靠性、确定性
	缺乏传输层和协议层的解耦，不支持大带宽的传输，不支持网络层的冗余	
硬件环境	AEC-Q100等车规环境要求包括：温度范围，湿度、震动/跌落，硬件生命周期大于10年	通过模块化设计，把不同安全等级的硬件集成在一个计算架构，统一数据通信接口和软件部署运行模式，可以逐步演进。
	供应商选择范围少，成本高，进化慢	通过Chiplet封装技术，可以把先进制程高算力CPU和成熟制程、高可靠性MCU做封装集成，达到成本、性能最优
软件可靠性	AutoSAR 编程规范，限制C/C++语言的非安全使用，私有的RTOS和中间件系统，基于model编程等自动代码生成	使用Rust语言，杜绝内存错误使用和异步并发带来的软件缺陷，使用高性能WebAssembly轻量级虚拟化技术集成不同的功能模块。使得可以使用通用编程技术和生态，降低软件定义汽车开发成本。
	不支持新的编程语言和工具，开发者生态小，开发速度慢、成本高	
软件工程	认证流程驱动的瀑布式开发方法，软件生命周期长、变化慢	通过虚拟化技术，解耦硬件和软件，支持不同软件应用的安全隔离、集中运算、迭代升级，对接云原生软件的DevOps生态，解决汽车软件新功能快速迭代开发的速度和成本。
	不支持快速的需求变化，难以满足智能驾驶舱、自动驾驶的快速功能迭代	

PULSAR 的架构如下图所示，从功能上划分可以简单分为6个域，包括：



• 提供整车计算和网络资源的中央控制域

首先通过以太网交换技术，解耦多个域，中央控制域提供基础计算能力，核心是多核冗余的服务器级的高算力ARM CPU平台，提供内存、存储、网络连接（WiFi、4G/5G）、PCIe总线等功能，支持type 1/2 hypervisor，可以运行商业发行版的Linux和RTOS，中央控制域平台的功能安全要符合ASIL D等级，但是因为ARM SoC的复杂性尚无法能达到此级别，但是最低也要符合 ASIL B 等级。平台的可靠性要符合ISO16750，冗余备份可采用零部件备份，功能重叠，硬件模块板备份等方式，其虚拟化计算资源池可以提供基于内存状态同步的负荷分担计算方式，支持其他域的计算需求。中央计算域主要通过板载的以太网交换机和其他域做互联。同时也提供和车控域直接连接的 CAN FD 接口的冗余连接方式，车控域的全部或部分软件可以以虚拟化实例方式部署在中央计算域，提供从完全嵌入式部署的车控域软件到集中化、虚拟化部署的车控域软件的升级路线。中央控制域还提供和自驾控制域的PCIe直接连接方式，因为主流的自驾NPU/GPU加速器都支持PCIe高带宽、低延时的连接方

式。同时，基于以太网的连接方式作为冗余备份方式。也支持和NPU基于最新的Chiplet总线，例如：AutoExpress标准的封装集成方式，提供数十倍优于PCIe PCB板级连接的速度、延时和能效比。

中央控制域也提供WiFi和移动4G/5G网络连接能力，网络平台需要提供对外部网络环境的防火墙能力和VPN能力，保障对外沟通的信息安全。对内部，提供基于域网络策略的外网访问能力。

中央控制器也管理和配置以太网交换机。交换机提供100–1000Mbps的交换带宽接口，目前车规级的车载以太网标准100BASE-T1（IEEE.802.3bw），1000BASE-T1（IEEE 802.3bp）标准还处在早期，为了保障可靠传输的 802.1 TSN（Time Sensitive Network）协议组也没有商用，相比之下基于数据中心和工业以太网交换需求的以太网交换机可以在抗干扰方面做加固达到车规的要求，同时通过采用数据中心采用的多链路互联的方式，通过带宽换取服务质量。

中央控制域也支持平台级的软件DevOps能力，为全域软件提供软件部署、监控、升级等能力。

• 自动驾驶域

自动驾驶域控制器平台需能满足单车智能或/和基于C-V2X的车路云协同的自动驾驶功能，需要处理大量的感知传感器输入，所以需要采用高带宽、高可靠性、低延时的连接方式来连接摄像头、激光雷达、雷达、超声雷达和其他传感器，例如：GSML (Gigabit Multimedia Serial Link) 可以在15米的传输距离内提供高达6Gbps的传输速度，USB 3.0则可以在18米的最大距离提供最高4.8Gbps的传输速度。需要提供算力支持计算机视觉算法处理、感知数据融合、交通参与人行为预测、宏观和微观路径规划等计算负载，所以需要配置合理的CPU、GPU、DSP、NPU的异构处理器组合配合算法达到低延时、低功耗、低成本的目标。自动驾驶域控制器通常通过SoC方式集成上述异构计算组件，通过负荷分担方式提供冗余计算能力。目前只能支持ASIL B的安全等级。

• 车控域

行车控制需要最高等级的ASIL-D支持，在目前阶段还以支持lock-step的嵌入式MCU作为主要计算单元，并通过本地CAN总线来控制驱动、转向、制动、悬挂等系统。中央控制器通过以太网和车控域连接，可以通过中间件消息来完成线控指令的发送。随着软件定义汽车技术的演进，车控域的MCU软件功能可以通过虚拟化技

术运行在中央控制器，通过以太网来直接控制线控能力单元，或者本地MCU软件和虚拟化的MCU软件互为备份，达到高可靠性的目的。

• 电池管理域

电池管理也涉及整车能源和动力系统，需要支持至少ASIL-C 等级，和车控域一样，在目前阶段还以支持lock-step 的嵌入式 MCU 作为主要计算单元，并通过本地 CAN 总线来控制电池端的传感器实现监控、充放电控制等能力，这些能力通过中间件消息以 API 方式提供给中央控制域。

• 车身域

车身域完成车辆的舒适性控制，例如：座椅调整，空调管理、车灯管理、门窗管理等功能，需要支持ASIL B等级。这部分的MCU软件可以直接迁移到中央控制域作为虚拟化的软件实例运行。

• 智能座舱域

智能座舱域提供对整车的配置、信息显示和控制能力，提供多模态人机交互能力，例如：语音识别、手势识别，人脸识别等，提供数字娱乐内容，包括：音乐、视频、游戏和其他app。这部分只需要达到ASIL-A等级，即使出现了功能故障也不影响基本的行车安全。智能座舱域一般采用移动SoC的车载版本，来更好配合Android生态系统。



4.2.3 AutoExpress项目：敏捷DSA和D2D互联接口标准

Chiplet（芯粒）技术是后摩尔时代在同样工艺水平情况下，提升芯片集成度的重要途径。在过去几十年中，半导体产业一直按照摩尔定律的规律发展，凭借着芯片制造工艺的迭代，使得每18个月芯片性能提升一倍。但当工艺演进到5nm, 3nm节点，提升晶体管密度越来越难，同时由于集成度过高，功耗密度越来越大，供电和散热也面临着巨大的挑战。Chiplet（芯粒）技术是SoC集成发展到当今时代，摩尔定律逐渐放缓情况下，持续提高集成度和芯片算力的重要途径。相比传统Monolithic芯片技术，Chiplet技术背景下，可以将大型单片芯片划分为多个相同或者不同的小芯片，这些小芯片可以使用相同或者不同的工艺节点制造，再通过跨芯片互联和封装技术进行封装级别集成，降低成本的同时获得更高的集成度。

Chiplet技术研究的意义在于：

- ◎ 通过开源拉通产学研合作，面向后摩尔时代芯粒（Chiplet）互联的技术，构建完全基于国内的工艺情况的物理层标准；
- ◎ 催生国内基于Chiplet的芯片敏捷开发模式，加速汽车芯片快速落地，缓解汽车领域“缺芯”困境；
- ◎ 催生Chiplet形式NPU开发，促进AI算法芯片化的快速迭代；
- ◎ 降低芯片开发公司的进入门槛，吸引新的人才和年轻工程师自主创业，激发国内半导体行业的活力，推动我国集成电路快速发展。

与传统的消费级和工业级的芯片不同，汽车电子需要车规工艺，同样也需要车规IP（接口类），然而车规IP和消费电子IP差别很大。

◎ 首先是可靠性的区别。车规根据等级不同，环境温度的容限差别非常大，在使用寿命和失效率方面差别也很明显。AEC组织有很多不同的标准，适用于芯片IP，业界广泛使用的是AEC-Q100的车规验证标准。

除此之外，汽车电子的设计使用寿命至少是10年，一般会以15年以上，失效率在使用寿命期基本上是在

1ppm，也就是百万分之一以下。

- ◎ 车规IP的可靠性要求相较于消费类IP，会有大量的、额外的复杂工作。主要是在工作温度、耐久性和可靠度上的差异。不同的Tier 1厂商对于Mission Profile的定义也不尽相同，车规IP可通过与客户、foundry合作，共同定义切合实际的Mission Profile，并结合车规的工艺和车规相关的设计数据进行严格的仿真验证来达成设计要求，这也间接的延长了车规芯片的开发周期。

参数要求	消费级	工业级	汽车级
温度	0~40°C	零下10~70°C	零下40~155°C
湿度	低	根据使用环境而定	0~100%
验证	JESD47 (Chips) ISO16750(Modules)	JESD47 (Chips) ISO16750(Modules)	AEC-Q100 (Chips) ISO16750(Modules)
出错率	<3%	<1%	0
使用时间	1~3年	5~10年	15年

- ◎ 功能安全性对于车规级芯片也是至关重要。特别是对人身安全有关联的部件和模组，国际标准化组织定义了ISO 26262标准来规范不同的安全等级。所以对IP的要求，就像可靠性AEC-Q100那样需要很多的额外工作和流程的介入。

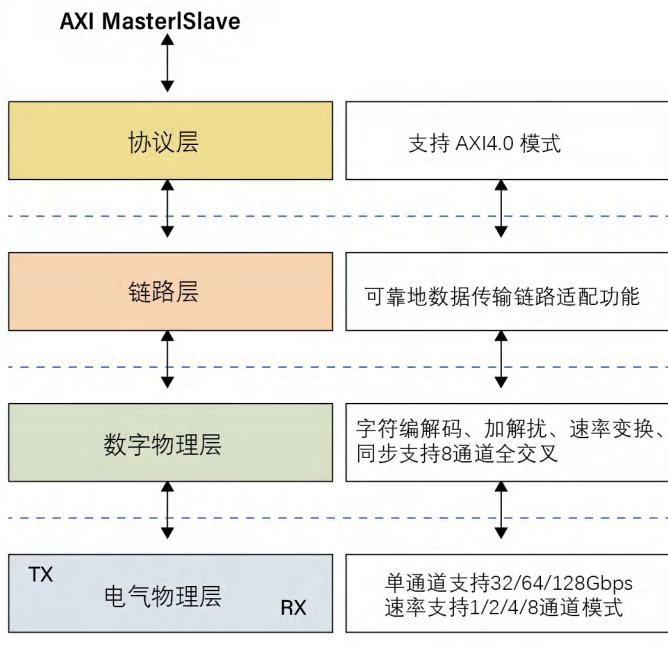
为了提升功能安全性一般会采用两种策略，一种是保护性的设计，另外一种是冗余性的设计。保护性的方法有多种，实现起来较为复杂，但具有针对性强，效率高的特点，如片上存储的ECC纠错、存储器BIST优化、数据包CRC检错、Watchdog防锁死、执行单元的Lockstep等。冗余设计的方法就相对比较简单，适用面也广，但是成本开销就大。业界通常会将两种方法融合使用，各取所长。车规级IP设计挑战主要包含了以下几个方面，可控制质量和可溯源的质量管理体系QMS、行业标准要求的可靠性设计AEC-Q100、功能安全性设计ISO 26262、流程中各个环节的审查报告、签核交付件和第三方权威机构的评估报告或者认证书。

AutoExpress项目是面向芯片模块化敏捷开发的Chiplet模式中Die-to-Die (D2D) 互联接口，满足多裸片短距高带宽低延时互联，并针对汽车领域做了车规级设计，项目目标为：

- ◎ 可靠性：选择车规工艺路线,按照AEC-Q100的可靠性标准实现。
- ◎ 功能安全：满足ISO 26262 标准，提供各个层级的功能安全设计，主要包括错误检测、心跳字、健康状态字、前向纠错、通道冗余备份等设计。
- ◎ 速率：最大支持8Lane × 128Gbps传输率，向下兼容64Gbps、32/28Gbps。
- ◎ 低误码：32Gbps下，误码率<10⁻¹²。
- ◎ 低延迟：8Lane × 32Gbps模式下，端到端延迟为50~60ns。
- ◎ 低成本：相较并口方案成百上千个PAD的互联方案，AutoExpress的PAD数量仅为几十个，低成本少层数封装即可满足传输要求，相同面积下，成本仅为高级封装的1/3。
- ◎ 封装方式灵活：支持2.5D封装，2D封装，极短距离PCB封装。
- ◎ 兼容广：协议层可配置，可以支持AXI以及其他标准总线。
- ◎ 与配套的NPU搭配，互联接口针对AI场景优化。AI传输对延时不敏感，但是对带宽敏感。AutoExpress的优化方向更加偏向于带宽，对AI优化。

AutoExpress采用分层协议架构实现，主要包含协议层、链路层、数字物理层和电气物理层，各层独立提供指定的业务功能，从而降低了标准实现的复杂度。具体实现可以基于业务需求优化某个层次的功能特性，进而开发出独具特色的产品，同时亦能复用已有成熟层次的具体实现，降低成本。其协议层默认支持AXI4.0协议，后续可根据业务需求加入对更多协议的支持。链路层提供了快速的建链流程，同时支持自动检测、纠错以及重传机制，以提供可靠传输。物理层包含数字物理层和电气物理层，其中电气物理层采用超短距SerDes技

术，满足高性能D2D的高带宽、低延迟传输；数字物理层承担SerDes传输必须的加解扰、编解码、数据对齐、链路交叉、极性反转等功能。



AutoExpress技术研究的意义在于：

- ◎ 技术层面：通过D2D接口将多颗不同裸NPU die进行合封。
- ◎ 仿真层面：D2D接口仿真模型。
- ◎ 质量层面：提供D2D接口signoff标准与支持。
- ◎ 生态层面：不同工艺结点、不同foundry技术可互联。
- ◎ 技术价值 - 弥补目前国内Chiplet芯片开发模式下缺乏统一且高效的Die-to-Die互联接口，避免产业的重复冗余投入。
- ◎ 业务价值 - 通过实现D2D接口标准化，有力推动Chiplet技术发展，对于中国的半导体产业链上下游企业都会带来直接的经济效益。
- ◎ 生态价值 - 通过实现D2D接口标准化，将使得Chiplet技术在整个产业链应用得更加广泛；新的Chiplet技术将会催生出新的商业模式，打破传统SOC设计模式，重构供应链市场。

AutoExpress应用场景：

场景1：异构芯粒互联集成

- ◎ 用于实现基于主芯粒（称之为Hub Die，包括CPU与主要IO接口）与各功能化芯粒（称之为Side Die）进行互联。
- ◎ 可服务于系统级芯片的模块化NPU设计，按需迭代速度不同进行迭代，加速芯片产品的上市。

场景2：同构芯粒扩展互联

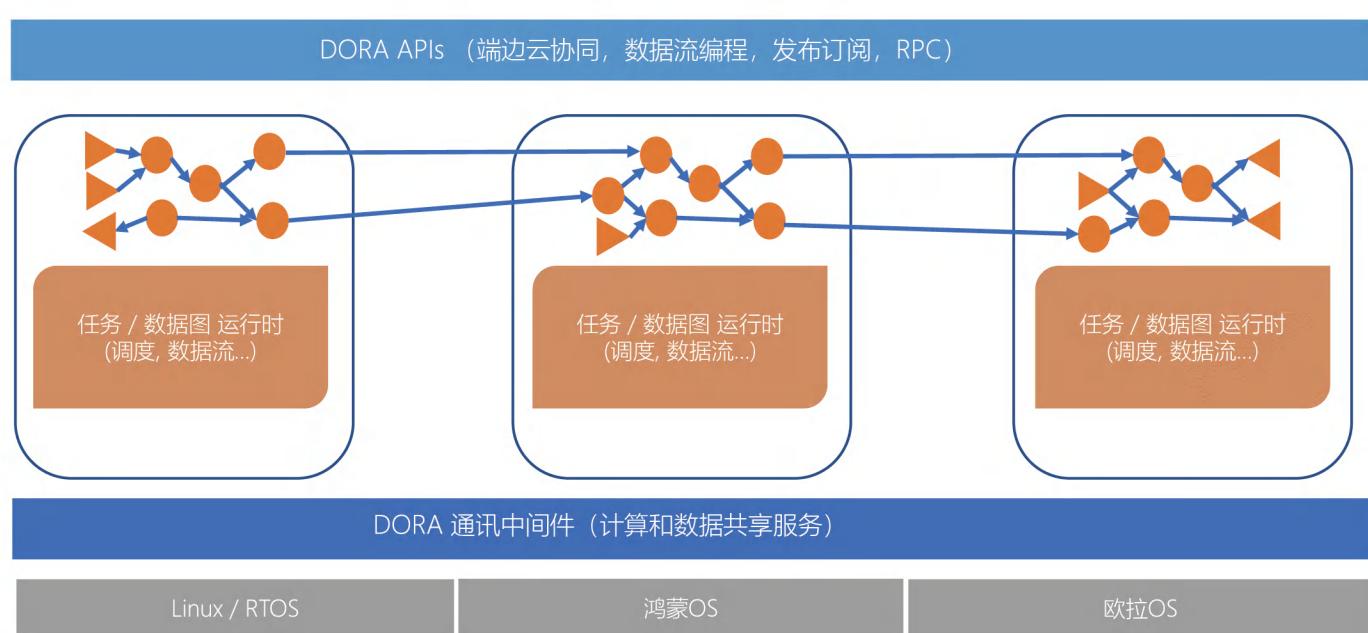
- ◎ 用于实现多个相同功能芯粒通过互联网络实现扩展互联，从而提供不同规模的算力芯片。
- ◎ 不同档次的产品，以及不同功能级别的场景对算力会有不同的需求，通过同构扩展可以基于一次流片实现多种算力需求。

4.2.4 DORA：数据流计算中间件

自动驾驶实际上就是在限定场景使用的四轮机器人，因此大多数的商用自动驾驶系统或直接使用目前已经比较成熟的ROS (Robotics Operating System)，或以ROS为基础来构建计算平台，例如汽车产业SDV的中间件标准提议SOME/IP也大量参考了ROS的架构和API设计。但是，ROS核心社区比较封闭，缺乏汽车行业成员，有商业利益的公司对项目有过高比例的决定权，而且主要服务机器人社区，对软件定义汽车和自动驾驶的特性需求考虑较少。ROS基于15年前ICT基础技术上搭建，现代的云计算技术尚未引入平台，例如云原生技术、低成本开发、DevOps 实践等。

DORA项目 (<https://github.com/dora-rs>) 应运而生。它以软件定义汽车，智能驾驶为目标场景，集成了大量云原生的架构理念和生态工具，并且采用了安全可信的Rust语言开发，支持弹性数据流计算框架、多样化的中间件选择、云端协同计算，为智能汽车的软件开发提供了又一个更加现代化的平台选择。同时，它也提供了一条对ROS1/ROS2升级、替代及超越的技术路线。

DORA的参考架构如下图所示，它的核心是一个跨端边云的数据流计算框架。

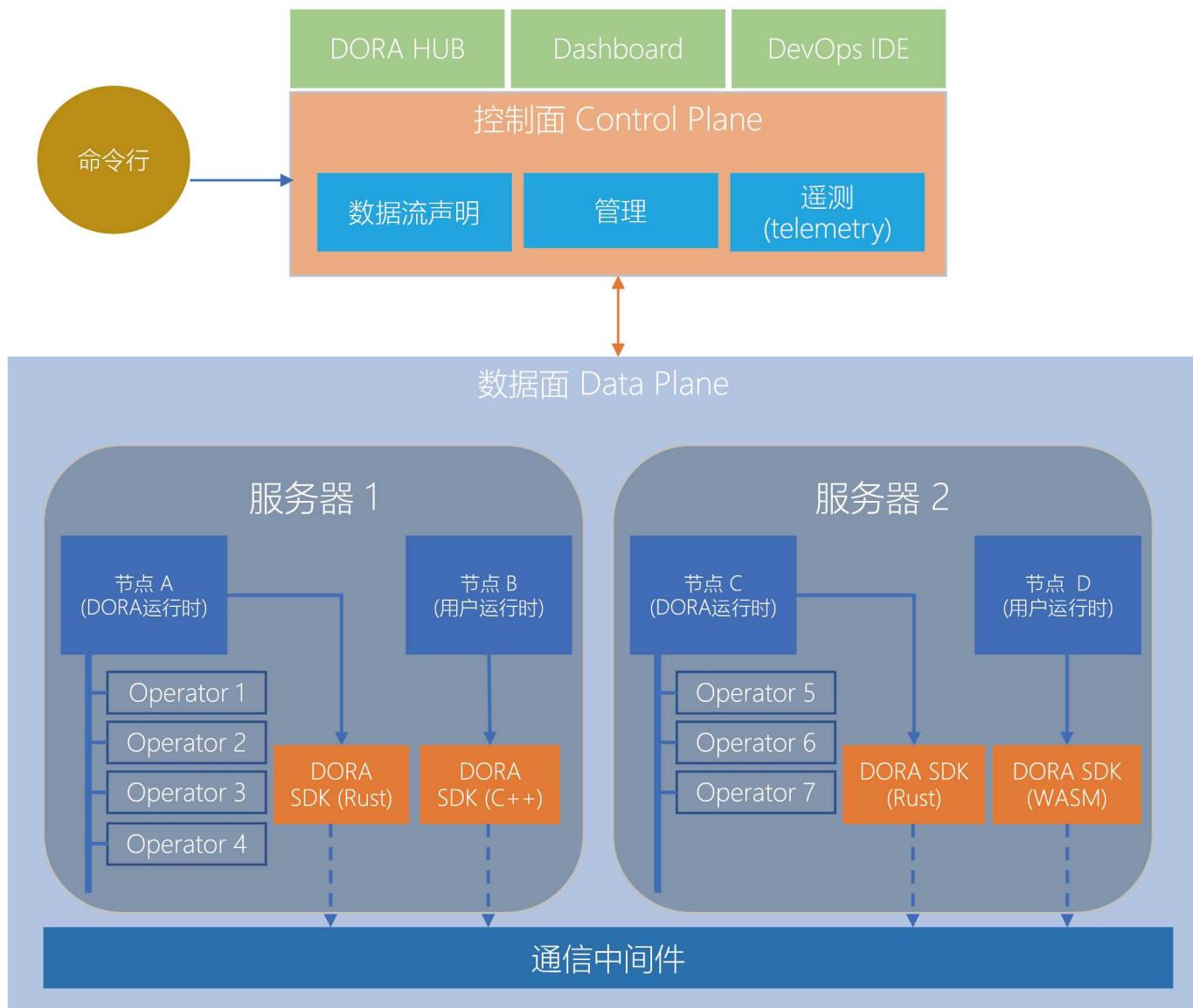


下表比较了 DORA 与 ROS2 的主要差异。

	ROS2	DORA
中间件开发语言	C/C++	Rust
API	使用C/C++接口的命令式编程，分布式计算为核心理念的架构，统一pub/sub消息接口，消息数据格式不统一	使用YAML脚本的声明式数据流编程，以计算为核心理念的架构，定义标准数据交换接口，对通信中间件透明
应用开发语言	C/C++为主，通过C/C++二进制接口集成其他语言，包括Python	Python优先，结点和算子库支持Python原生编程，支持C/C++，Rust算子库
通信中间件	支持RTPS的DDS标准的实现，包括Cyclone DDS, FastRTPS, Zenoh, ICEORYX	通过YAML脚本来描述通讯中间件的选择，支持Pub/Sub, RPC, P2P, 共享内存，比如Zenoh, ICEORYX，以及数据序列化库，Cap'n Proto, ProtoBuf等
实时性和可靠性要求	缺乏	开发者可以描述算子和结点的实时性约束和可靠性需求，由运行时来保障
云边计算	DDS不能跨越二层局域网，无法和云计算平滑整合，不支持车队管理	节点可以跨越本地计算环境，通过选择云边协同通信中间件来共享数据。
云原生 DevOps	不支持	以YAML脚本为核心做开发、部署、运维，支持完备的DevOps命令行，遥测能力
算法社区	大颗粒的领域算法实现，例如：Autoware, Nav2, NDT	支持细粒度的算子灵活集成，ROS算法可以被重构为DORA算子或节点
硬件生态	大量支持ROS接口的传感器和动作器	支持ROS消息转换为DORA结点，从而支持对ROS硬件的读取和控制
日志数据格式	rosbag是传感器数据的标准	支持云原生的open telemetry日志数据格式，也计划支持以ROSBAG格式存取日志数据
仿真	Gazebo, Carla	和Carla优化集成

下图给出DORA更加细节的软件架构，主要分成控制面（Control Plane）和数据面（Data Plane）。控制面（Control Plane）组件主要有：数据流YAML声明的处理，数据流的管理和遥测；负责解析YAML蓝图，按照蓝图部署、管理和监控计算图的运行，同时提供命令行和API接口。在其之上，可以利用云原生生态构建更多的工具，比如：DORA Hub，Dashboard和DevOps开

发工具等。数据面的组件包括节点（node），算子（operator）和 DORA SDK库。节点node是一个单独的程序进程，可以运行一个或者多个算子功能，一台计算机可以运行一个或者多个node，所以node部署是可以跨越单机的，operator实现一个算法或者函数处理。在多个node和operator之间的通信经由通讯中间件实现。



DORA提供多种编程语言SDK，目前提供Python, Rust, C/C++ SDK来开发node和operator。尤其是把对Python语言的支持作为一等公民来对待。这是因为大量的数据科学家和算法工程师都是以Python语言作为主要工作语言。这样开发者可以使用DORA Python API, 通过低编码IDE来编程，可以快速测试、验证ML模型和算法。因为DORA支持共享内存，在数据流交互中尽量避免数据拷贝，即使是Python开发的应用，也有良好的性能保证。开发者可以充分利用和使用来自研究和开源社区的最新ML模型和算法（例如: <https://paperswith-code.com/>, <https://huggingface.co/>）来快速构建机器人以及自驾系统原型，用于研究和教育。经过验证的ML模型和算法通过DORA C/C++/Rust API快速引入机器人和自驾系统的生产环境，然后生产环境提供对新的ML模型和算法的反馈Python API用于研究、教育和快速原型设计C/C++/Rust API 用于高性能、低延迟、异步并行的生产环境；两者有机结合，相辅相成，形成正向循环。

DORA支持发布/订阅、RPC、共享内存、点对点通道的不同通信中间件实现，node和operator基于统一的数据结构来实现数据的输入和输出；DORA通过声明式语言YAML定义了数据流图标准的node和operator API接口（SDK），以及节点内（例如任务之间的管道）和跨节点之间端到端数据流图的数据面（data plane）拓扑关系和实时制约（比如，自驾系统中特定的任务调度算法：软实时、截止时间戳deadline，和水位watermark等）；YAML描述的部署需求由控制面Control Plane负责通过选择不同的通信中间件的实现，满足YAML蓝图的部署要求。

为开发者提供一个对开发和生产透明的环境也是云原生的核心价值，从模拟仿真环境移植到实际的车载计算环境，性能一定会有差异，DORA通过Control Plane控制面和云原生生态工具进行云集成（北向），充分利用云原生技术来对数据流组件（Nodes、Operators、Sources、Sinks）进行管理、遥测（telemetry）；以及软件组件的生命周期管理。这些工具让开发者能够有

效调试组件异常或不安全的行为，并且提供数据记录保存和离线再现运行场景的支持。使得在 DORA 上开发的组件可以在模拟中进行测试，并毫不费力地部署到真实车辆上，ML模型和算法的行为也忠实地再现在真实车辆中。

DORA提供类似Docker/Docker Hub的基于最新IT技术开发的DevOps工具集。开发者基于DORA SDK开发DORA模块（节点nodes或者算子operators），编译测试DORA模块，把测试通过的DORA模块打包（packaging），上传托管到DORA HUB平台，最后部署时，DORA根据其YAML申明式描述，下载所需的DORA模块，构建端到端的数据流执行环境。

在DORA中间件之上，DORA-DRIVES是一个面向自动驾驶和软件定义汽车的应用开发社区，提供大量经过验证的算子库和传感器、控制器接口库，基于这些库，提供了典型自驾应用的模版实现，例如：Autoware的大部分算法，可以通过接口改造变为DORA算子，Autoware可以从ROS2架构迁移到DORA架构。DORA-DRIVES目前已经充分利用最新的开源ML模型和算法，包括感知，规划和控制，构建了完整的端到端的自驾数据流实现。通过DORA Python API，开发者可以快速迭代其中的ML算法模块，DORA的“即插即用plug-and-play”的模块化架构设计，对引入新模块和组件，提供了非常好的开发体验。同时在可移植性上，通过DORA的性能监控能力，开发者能够在不同的模拟器和现实世界的机器人或者车辆之间进行转换，以便在各种驾驶环境中评估他们的组件。例如，开发者应该能够确保他们的规划算法在交通模拟器（如SUMO）、自驾模拟器（如CARLA）以及真实世界的车辆上提供类似的行为。



4.2.5 ESSA：弹性带状态无服务器计算架构

车路协同，是提升汽车智能化程度终极实现无人驾驶的重要一步，也是我国发展智能网联汽车的重要战略之一。车路协同技术方案，需要端 - 边 - 云协同，高效分配和执行计算资源。ESSA (Elastic Stateful Serverless Computing Architecture) 项目，是支持端边云高性能计算的一个加速器技术。

无服务器计算 (Serverless Computing) 日渐流行，因为它简化了开发人员构建和部署应用程序的复杂过程。目前主要的无服务器计算架构平台都是部署在大的公有云上，而且都是无状态的应用。而大量的 V2X 应用要求低延时，有状态。ESSA 弹性带状态无服务器计算架构就是为了满足边缘部署和有状态应用而开发的，它主要包括两个子项目：

- Anna-RS: 这是一个支持地域分布的低延迟、自动缩放 (Auto Scaling) 和高效的分布式键值 (KV) 数据库，通过 Anna-RS KV 来支持 ESSA 弹性无服务器计算架构所需要的状态管理；
- Essa-RS: 这是一个基于 WebAssembly 的无服务器函数运行环境，相比其他无服务器函数的运行环境，WebAssembly 具有速度快、多语言 (C/C++、Rust 等语言) 支持、非常安全的沙箱隔离，更少的函数调用开销等优点。



4.2.6 EtymOS项目：自动驾驶的软件2.0平台

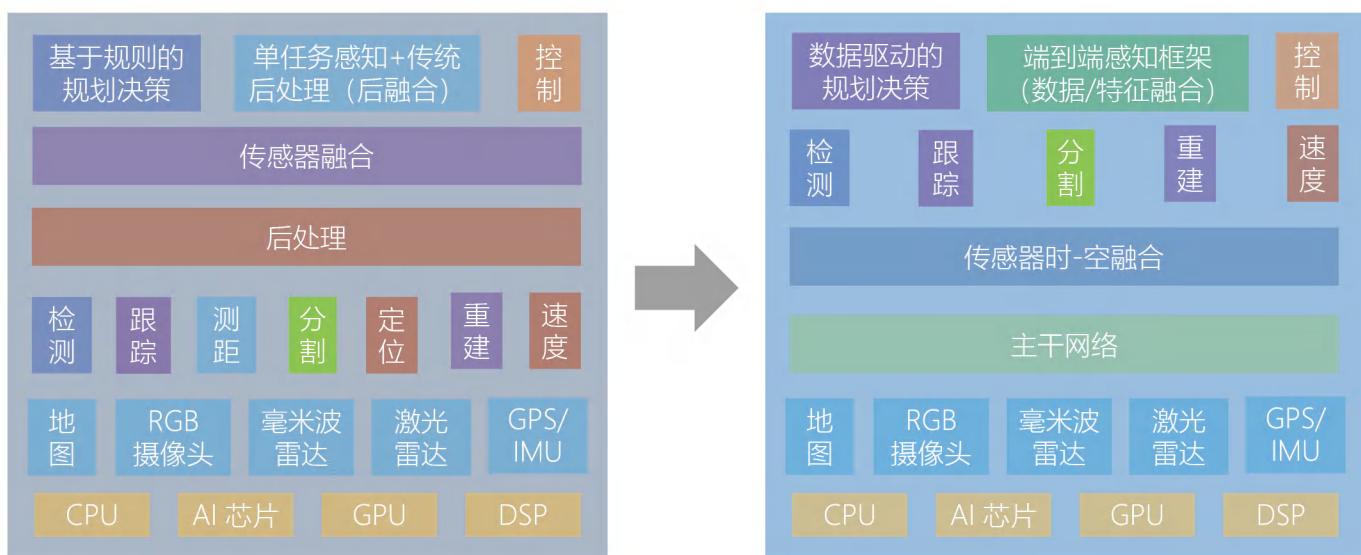
自动驾驶基本存在1.0和2.0两个发展阶段。自动驾驶1.0时代采用多种传感器构成感知输入，比如激光雷达 (Lidar) 、视觉相机、雷达 (Radar) 、惯导IMU、轮速计和GPS/差分GPS等，各个传感器在感知能力上存在差异，针对性用多模态传感器融合架构，目前多是采用后融合策略，把各个传感器在相关任务的结果进行一次滤波，达到互补或者冗余的效果。这方面存在两种路线，一是依靠激光雷达加高清地图的做法，成本高，主要是 robotaxi 等L4公司采用，另一个是视觉为主、轻高清地图的做法，成本低，多是L2/L2+自动驾驶公司为量产落地的思路。两种路线都是会有很传统的后处理步骤（特别是视觉），大量调试工作和问题也来自于此。另外，这个阶段的规划决策多是采用基于规则的方法，实际上没有数据驱动的模型，比如开源的Autoware和百度Apollo。L4公司由于运行场景在一些具备高清地图的固定地区，本身感知投入的传感器精度较高，已经做过数据训练规划决策模型的探索；相对来说，L2/L2+公司还没有建立数据驱动的规划决策模块开发模式，多是采用优化理论的解决方案，普遍从高速场景入手，升级到特斯拉那种“闹道-到-闹道”的模式，很少能支持城市自动驾驶的复杂场景（比如环岛路口和无保护左拐弯操作等）。

自动驾驶2.0时代应该是以数据驱动为标志，同时对于1.0阶段的感知框架也有比较大的改进。数据驱动的开发模式倾向于端到端的模型设计和训练，对于规划决策而言就是需要大量的驾驶数据去学习“老司机”的驾驶行为，包括行为克隆的模仿学习，以及通过基于模型强化学习 (M-RL) 估计行为-策略的联合分布等，不再靠各种约束条件下的最优问题求解。其中轨迹预测是一个重要的前奏，需要对智体交互行为做良好的建模，并分析存在的不确定性影响。对于感知而言，2.0时代需要考虑机器学习模型取代那种传统视觉或信号处理（滤波）的部分，真正做到采集数据来解决问题的开发模式。比如特斯拉最近的BEV和Occupancy Network，都直接通

过深度学习模型实现所需信息输出，而不是采用传统视觉和融合理论去二次处理模型输出。传感器融合理论也从后融合升级到模型中特征级融合甚至数据融合（如果同步和标定有一定的先验知识）。这里可以看到Transformer网络在这个感知框架下扮演着重要的角色，同时也给计算平台提出了更高的要求。基于这种数据驱动的自动驾驶平台需求，大模型的设计思路也被引入，因为大量数据的获取，包括高效的数据筛选、自动标注和仿真技术辅助，需要在服务器维护一个“老师”大模型支持各种布置到车端的“学生”小模型训练和迭代升级。

CARSMOS项目支持上述自动驾驶1.0和2.0两个阶段的自动驾驶平台。DORA和DORA-DRIVES为1.0项目的实现提供了很好的底座和参考，通过不断扩充算子(operators)，通过DORA可以编排出更高性能和多样化的1.0应用场景。

EtymOS围绕数据和模型来支持自动驾驶2.0项目，包括构建标准化的数据采集EtymOS Kit，通过自动化的数据飞轮EtymOS Wheel做数据集共享和社区协同开发，来快速提升自动驾驶模型的性能，打造标准化的自驾感知大模型EtymOS。



4.2.6.1 EtymOS Kit: 统一Sensor Kit 项目

EtymOS Kit 是一个标准化的自动驾驶感知 sensor 平台，可以和 EDIT 电动滑板车，PULSAR EEA 架构配合，面向低速无人商用场景，提供标准化的传感器硬件，例如，基于视觉的多摄像头方案，和基于高清地图的 Lidar+ 摄像头方案，以及数据接口，例如：基于万兆以太网和高速差分总线的物理层数据接口和 ROS bag 数据输出。

基于深度学习的感知算法大多是是基于数据驱动，而数据的依赖性在某些应用是非常敏感的，比如目标检测，在一个物流大卡车上安装的摄像头和激光雷达和一个乘用车安装的同样传感器，它们得到的数据有很大的不同；摄像头视角的高低直接影响目标遮挡的效果，也影响 3D 目标的识别精度；同样，激光雷达获取的点云也是不全面的，除了信息残缺的 L- 形状之外，得到目标（汽车为例）顶部信息的多少也不一样，这样对于点云聚类和深度学习的特征图提取都造成影响。另外，还有安装的传感器数目和角度等，会直接影响感知的融合框架和性能。有时候，对于数据前融合或者特征融合的感知算法，时间同步是一个刚需，而且摄像头的感知真值获取，也要靠时域同步的激光雷达点云提供。时间同步要兼顾：

- 配置和安装标准：提供不同车辆类型（卡车、客车、乘用车、物流小车和清扫车等等）的传感器安装的数目、位置、角度、视角和观测距离（比如摄像头焦距和激光雷达可达距离）等。关于时间同步的需求，要尽量通过硬件控制传感器的采集时刻，特别是不同模态的传感器（如激光雷达和摄像头）。
- 标准可互换数据集：对于同模态但不同设置的传感

器获取的数据，建议设定规范化方法，比如特斯拉的虚拟摄像头技术，去畸变-重新变换-加畸变；激光雷达的规范化包括点云空间分辨率和时间采样率的调整，今后也可以加入点云补全技术，填补“黑洞”等。在时间同步的要求得不到硬件支持情况下，必要的软件同步，即数据后处理，也是有价值的。比如，激光雷达点云在时域的内插，以确定摄像头拍摄时刻的修正点云，包括自运动补偿和其他目标运动的轨迹修正（基于刚体假设）。

4.2.6.2 EtymOS Wheel: 基于数据飞轮理念的模型工厂

在软件 1.0 系统中，所有测试和验证都是围绕源代码构建的，而软件 2.0 系统的核心是数据，模型是数据的产品（artifact），因此数据是事实真相的唯一来源（single source of truth）。由于数据不包含显式逻辑，我们不能应用形式化的方法对软件 2.0 项目进行功能完备性验证，因而软件 2.0 模型总是存在失效的长尾场景。在软件 2.0 实践中，模型首先是利用作为基本事实（ground truth）的标记数据集进行训练的。然后模型将针对测试数据集进行测试。在测试数据集上可以生成足够好的预测时，该模型被认为是有效和通用的。模型部署到生产场景，会被持续做质量跟踪，实效的场景会不断被添加到训练数据集中。这个过程一直持续到训练数据集满足应用领域所需要的安全性和可靠性。这很像云原生软件中发展出来的 DevOps 方法，包括大规模的自动化测试、更高频度的软件版本发布，A/B 测试和实时的软件功能和性能观测。在软件 2.0 模型开发中，为了加快验证过程，可以使用数据增强技术来人为地合成接近但不同的新数据集，以有效地探索相邻的角落场景，这也很像软件 1.0 中的模糊测试。所以软件 2.0 可以利用 DevOps 的很多理

参考：<https://auro.ai/blog/2019/12/a-modular-vehicle-agnostic-sensor-housing-unit-for-autonomous-vehicle-platforms/>,
<https://werideai.medium.com/weride-releases-brand-new-generation-of-sensor-suite-weride-ss-5-0-412408a874ba>,
<https://www.quanser.com/products/self-driving-car-studio/>, <https://www.pixmoving.com/pixkit>
https://aws.amazon.com/cn/campaigns/dam_contact_us/

念构建围绕数据和模型开发的DataOps流程，这个流程被亚马逊更形象化描述为数据飞轮概念，因为启动飞轮转动需要较大的力，而一旦飞轮转动起来，维持其高速转动的力会越来越少。如前文所述，大模型的训练也需要大量的数据和算力的投入，而一旦获得了大模型能力，则可以极大增快下游任务的训练开发速度，甚至通过大模型本身的无标注、弱标注学习能力加快大模型本身的进化速度，也是一个构建数据飞轮的过程。

EtymOS Wheel的目标是构建一套融合边缘和云的软件平台，低成本、高效能完成自动驾驶场景数据的采集、筛选和清洗、数据标注、数据扩增、模型训练优化、模型版本验证到模型版本发布的全流程，同时对管理数据和模型的对应关系，包装模型训练过程的可重复性。

EtymOS Wheel可以部署在云端，提供一个对数据迭代、模型训练的统一平台和社区。EtymOS Wheel也可以在车队和开发社区侧，给予开发者更好的对本地数据的控制，更有效利用本地的算力。为了能够实现社区合力构建一个统一数据集和大模型的目的，EtymOS

Wheel CLOUD和EDGE之间需要同步融合。

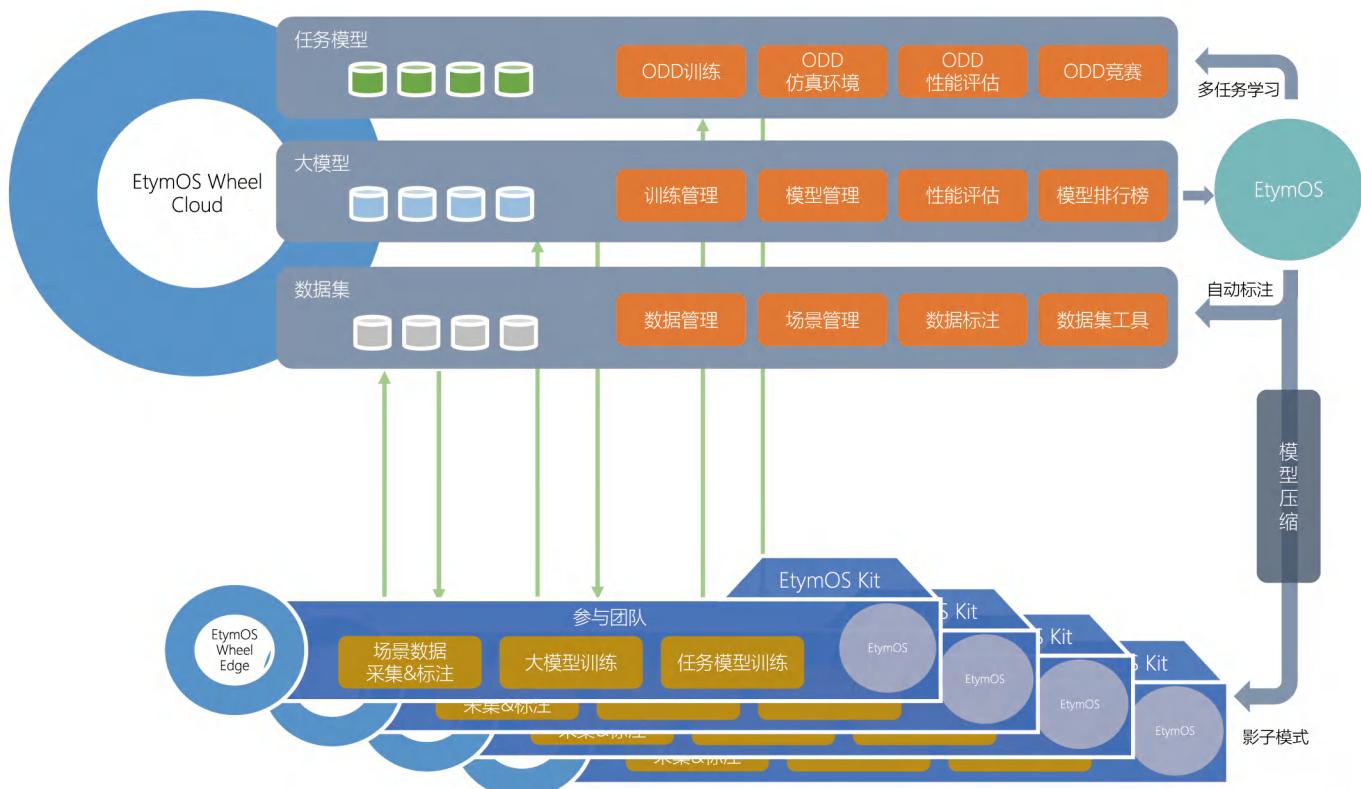
EtymOS Kit 产生的标准数据集，和对数据集的标注、数据集的发布格式，需要统一。鼓励开发者共享自己的数据集到EtymOS Wheel Cloud社区，达到聚合社区力量，构建统一标准数据集，构建大模型的目的。

EtymOS Wheel 提供对共享数据集的存储管理、元数据管理，提供自动化和交互式的标注工具，鼓励社区参与数据的标注，降低标注成本，标注后的数据集按照版本发布到社区，用于模型训练。

基于统一开放数据，社区可以训练大模型，为了降低算力成本，EtymOS Wheel提供分布式模型训练管理，这样可以聚汇社区的算力来训练统一的模型。同时，提供模型和数据集的版本对应关系管理，保证了从数据到模型的可重复性。

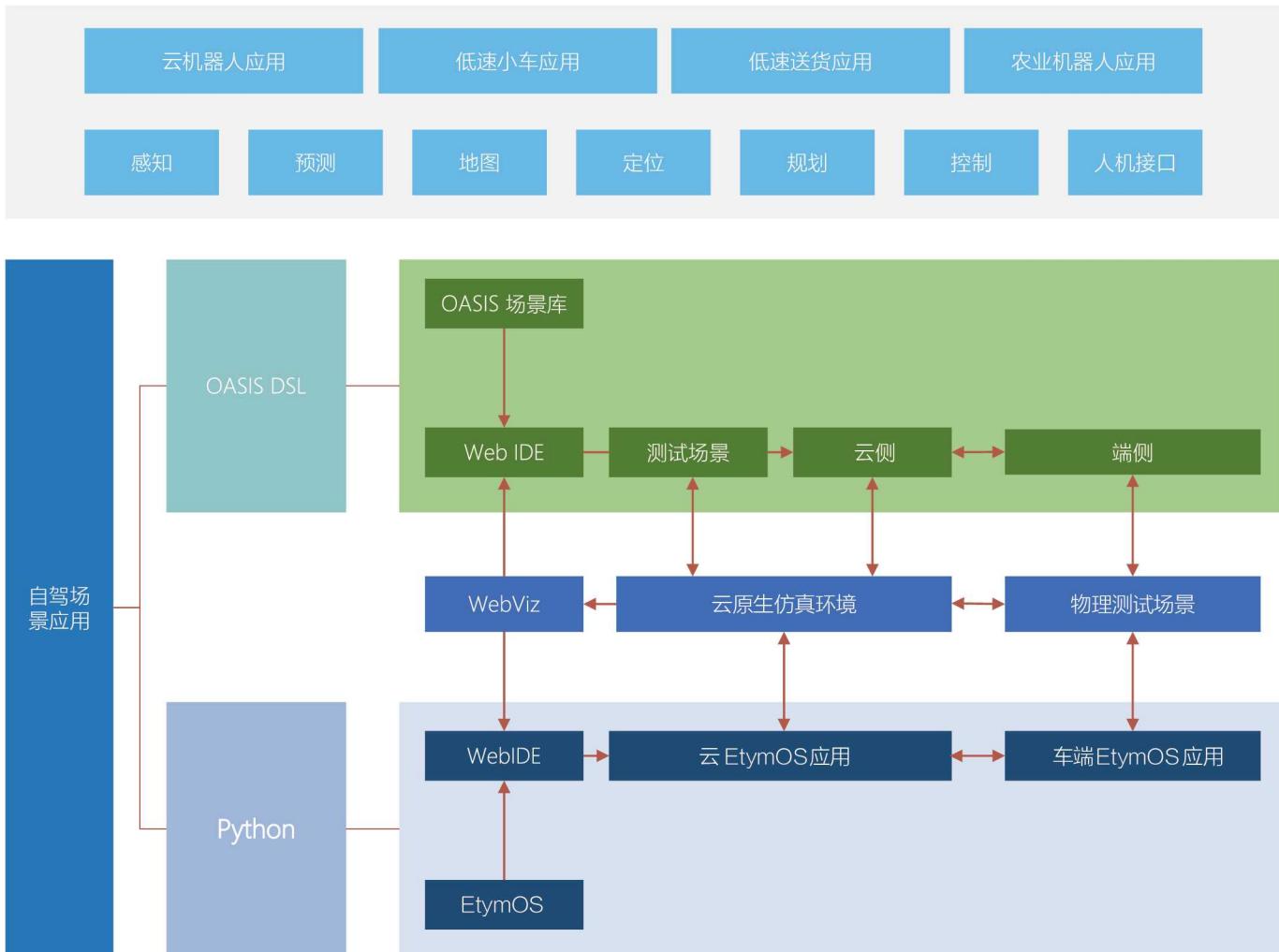
EtymOS Wheel社区提供模型性能评估测试和排行榜，激励社区产生更高性能的模型。

通过社区的竞赛迭代产生出性能最优大模型（SOTA）可以发布为EtymOS版本，EtymOS强调极致性能，可以进一步被用于对数据集的自动标注，同时也作为基础模型，用于下游自驾任务的进一步训练。



在自驾业务开发层面，EtymOS Wheel还提供基于DORA、EtymOS Wheel和OASIS仿真平台的低速无人驾驶应用开发IDE，可以方便的调用HUB中社区分享

的模型，或者自己托管的模型，来组合成一个面向ODD场景的全栈方案，可以在OASIS仿真环境中做场景测试，快速迭代算法。



4.2.6.3 EtymOS：自动驾驶大模型

大模型是自动驾驶业务解决方案中重要的一环。小模型往往在计算资源有限的客户端（比如车端）实时运行，而大模型则是在计算资源强大的服务器端非实时地进行，其包括的数据输入和结果输出也往往多于客户端的小模型。大模型所能达到的AI算法性能基本是小模型的性能“天花板”，但是其作用可以是两方面：

- 数据的预标注甚至是自标注，比如特斯拉的工作，特别是针对缺乏3-D传感器的量产车辆（非数据采集

车），其获取的数据往往难以标注3-D信息，而大模型基本能够完成这个艰巨的标注任务，也提高了包括小模型的AI算法迭代速度；

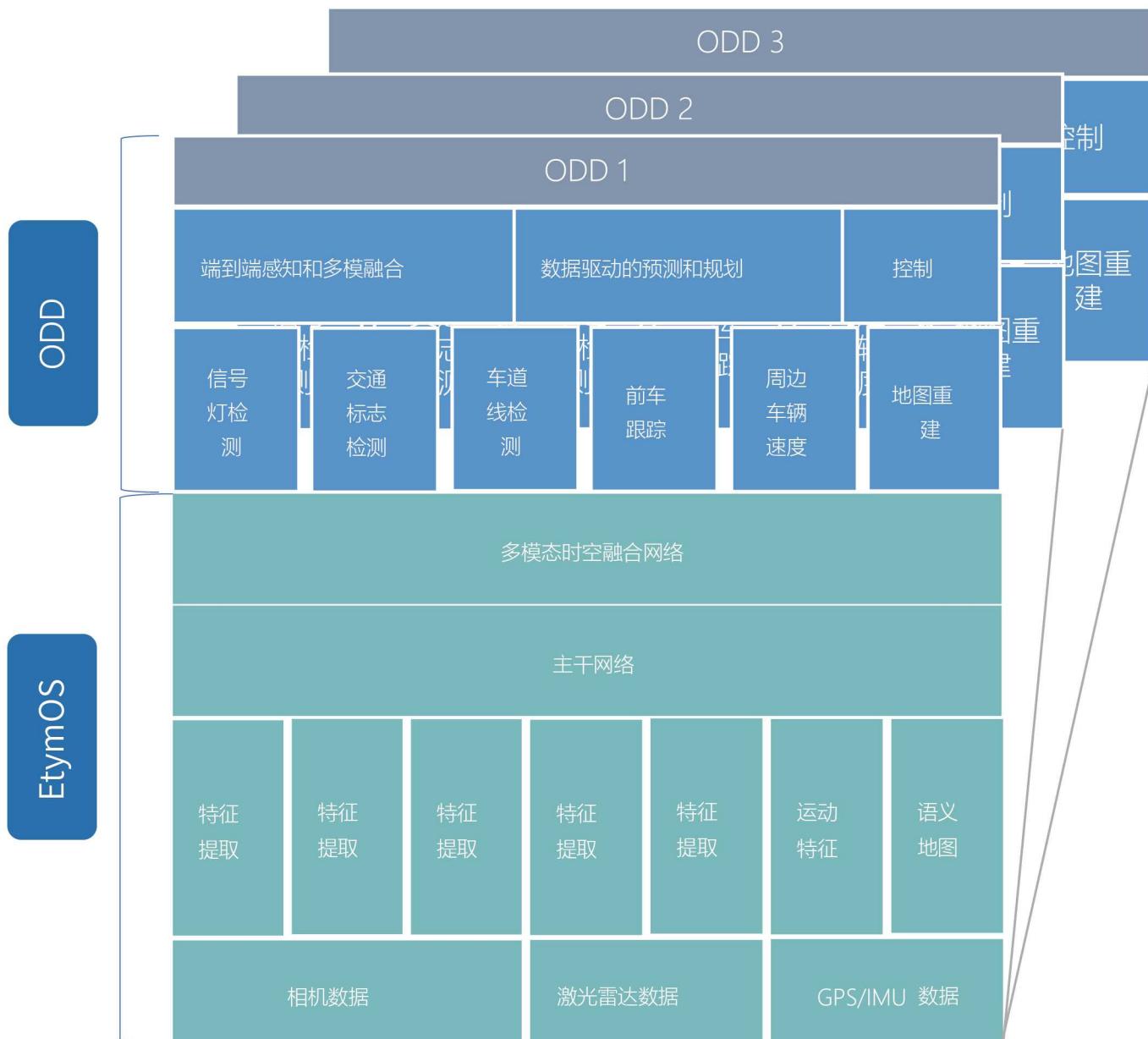
- 帮助客户端的小模型实现性能提升。这里小模型可能是一个个自动驾驶任务的单独模型，而大模型经常采用多任务训练（MTL）和多头网络结构，但在“教师-学生”的框架下，可以实现大模型->小模型的知识蒸馏。

通过EtymOS Wheel平台和社区合作，通过统一EtymOS Kit产生的数据集，社区不断迭代出最好性能的大模型，发布为EtymOS。EtymOS是基于社区产生的最好（STOA）算法构建的具有多头结构的多任务预训练网络。

- 基于EtymOS可以开发大量的感知和预测任务，这些任务把EtymOS作为一个预训练模型加入训练的管线，用来训练自己的局部参数，仍然保持EtymOS的参数冻结，这样EtymOS可以支撑多样化的任务开

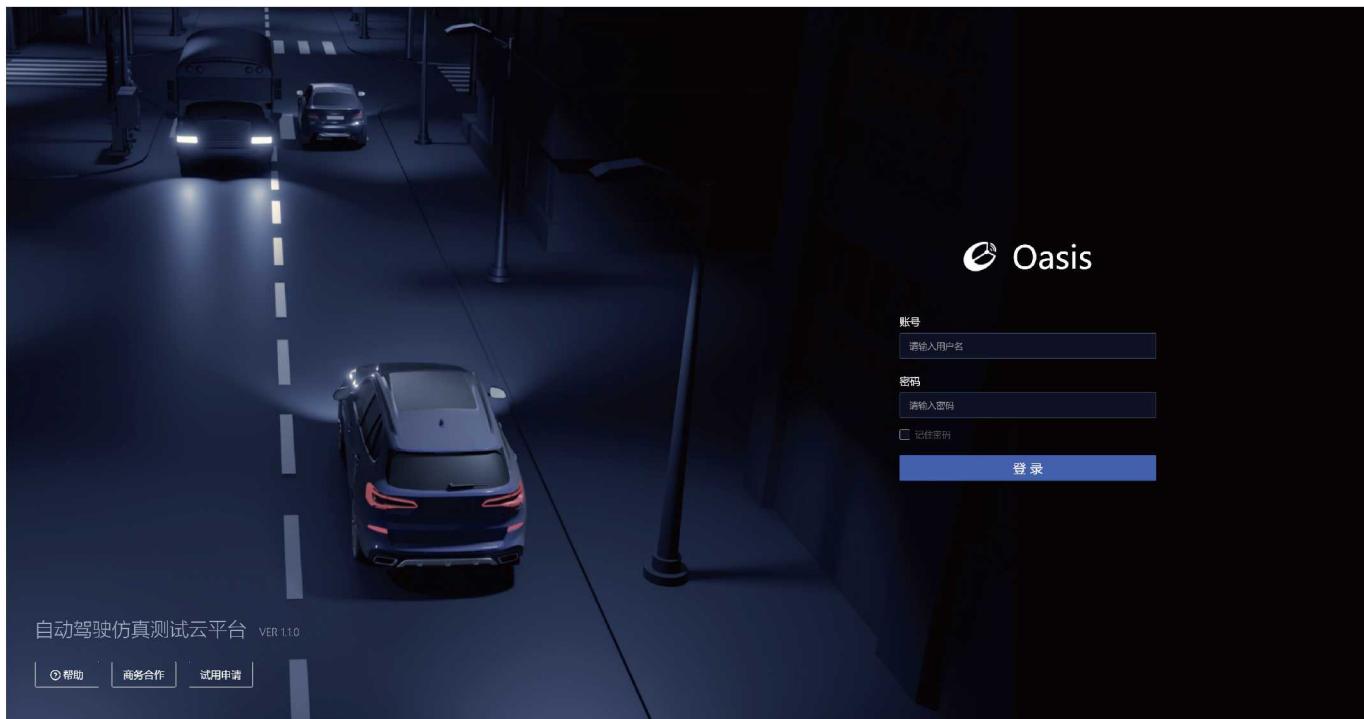
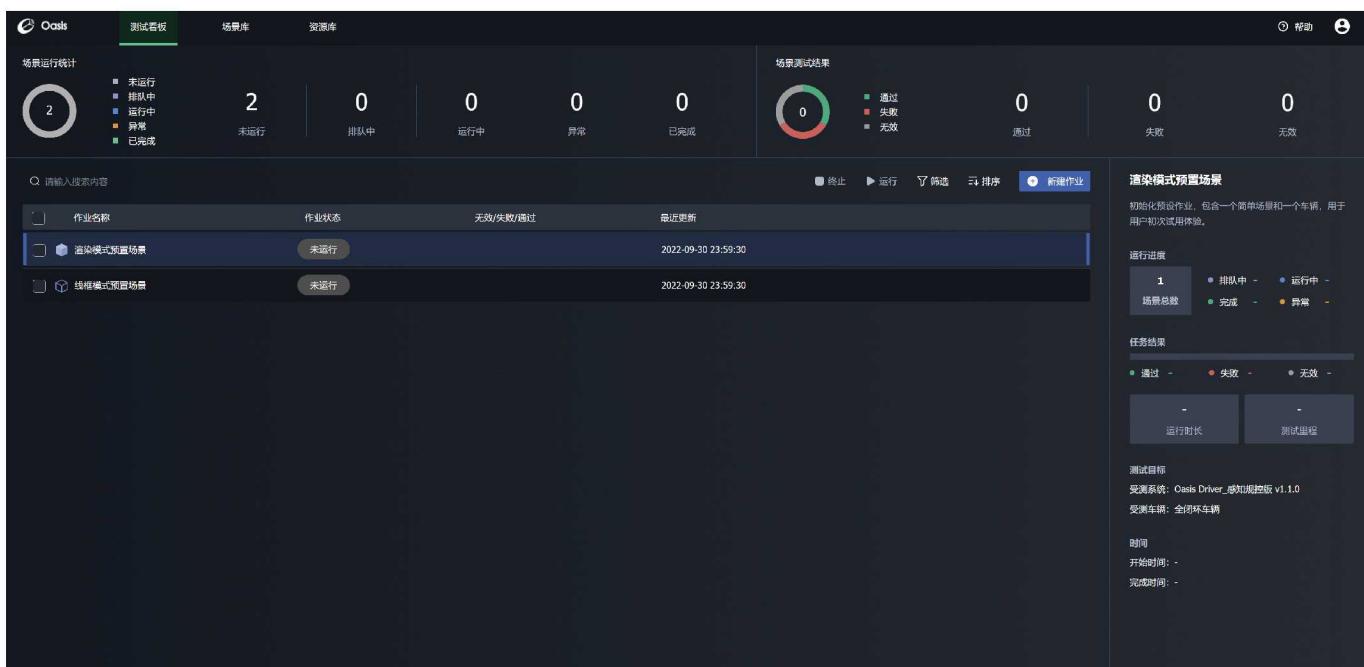
放，这些任务彼此独立，可以按照自己的版本发布。

- 基于EtymOS 和ODD业务场景，会形成一个模型组合，这个组合可以在ODD仿真场景中评估完整性能。性能如果达标，则这个EtymOS+ODD任务组合可以作为这个ODD业务场景EtymOS APP的一个云端版本发布。但是这个版本是性能优先的，不考虑计算效率和应用环境要求的实时性、可靠性等指标，需要通过对EtymOS APP做进一步的模型优化，才能运行在真实车端的软件和硬件计算环境。



4.2.7 OASIS：仿真测试平台和场景描述语言

4.2.7.1 OASIS Sim 仿真测试平台

The screenshot shows the OASIS test management interface. At the top, there are tabs for '测试看板' (Dashboard), '场景库' (Scene Library), and '资源库' (Resource Library). The '测试看板' tab is active.

场景运行统计

状态	数量
未运行	2
排队中	0
运行中	0
异常	0
已完成	0

场景测试结果

状态	数量
通过	0
失败	0
无效	0

渲染模式预置场景

初始化预设作业，包含一个简单场景和一个车辆，用于用户初次试用体验。

运行进度

场景总数	排队中	运行中
1	-	-

任务结果

状态	数量
通过	-
失败	-
无效	-

测试目标

受测系统: Oasis Driver_感知控制版 v1.1.0
受测车模: 全闭环车辆

时间

开始时间: -
完成时间: -

复杂的智能驾驶（L2/L2+/L4级自动驾驶）系统在设计、实施、集成和部署运营阶段都需要强大的仿真系统提供测试验证的能力。一方面保证系统的功能完整性和可靠性，另一方面提早暴露系统缺陷，降低开发和维护成本。OASIS Sim作为一个开放可扩展的平台为智能驾驶系统研发的完整生命周期提供不同等级，不同规模的仿真测试服务：

目前在企业中使用较为广泛的仿真测试解决方案主要基于VTD, PreScan, Carmaker等闭源商用软件。生态较为封闭，定制化开发及系统集成难度大。原本面向单机使用场景的软件架构在向云计算迁移的过程中面临诸多性能局限。高校、科研机构多使用开源仿真器，例如Carla, LGSVL, AirSim等，在此基础上二次开发或集成其他工具。经常出现重复造轮子，实验平台维护成本高，科研成果不可复现且难以共享等阻碍研究工作效率和成果转化的问题。

近年来，国内外出现了以TADSim（腾讯），51Sim-One（五一视界），SimPro（赛目），Cognata（以色列Cognata公司），Simian（美国Applied Intuition）为代表的新一代仿真测试平台，具有如下特性：

- 云原生软件架构支持大规模仿真；
- 基于游戏引擎的高保真渲染能力有望实现端到端智能驾驶系统的仿真测试。

这些新一代仿真测试平台大多按互联网产品的敏捷方式开发迭代：特性更新迅速，运维体系完整，用户体

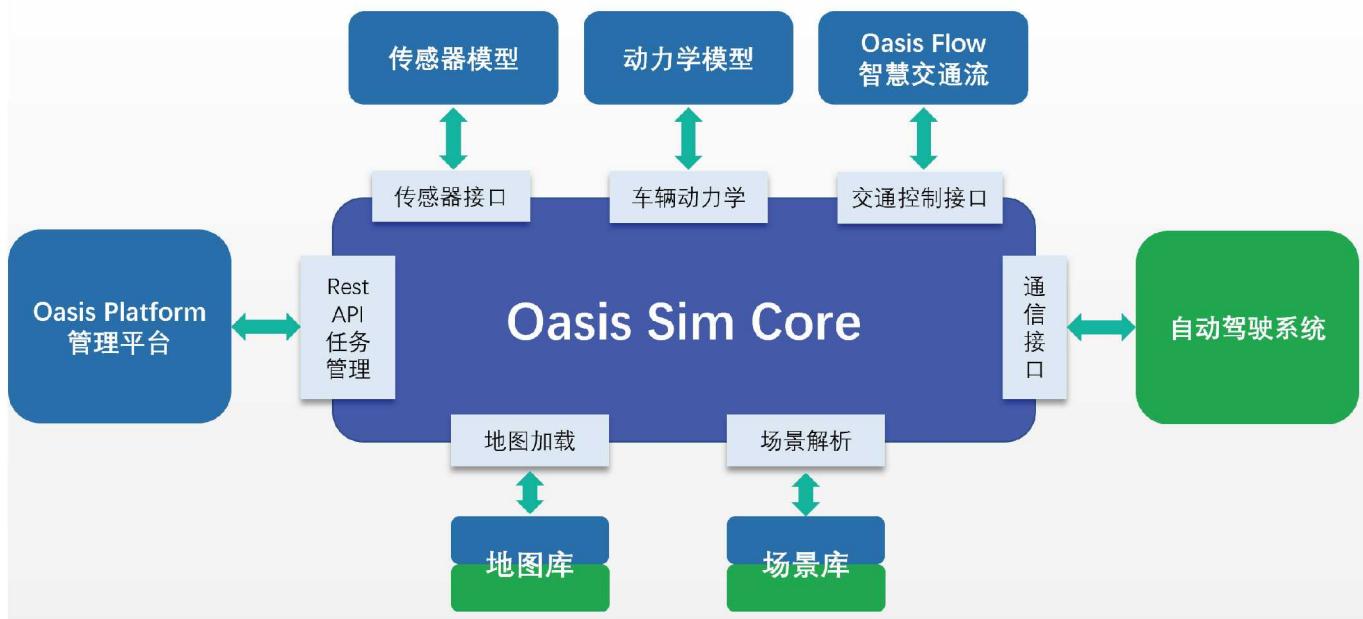
验流畅。功能完整度和性能上可比肩一线L4自动驾驶企业（例如Waymo, Tesla等）内部自研的仿真工具链。但这些先进平台还未在产业界，尤其是主机厂广泛应用。究其原因，可归结为仿真平台与企业现阶段智驾功能开发需求的不匹配：

- 业界对仿真渲染保真度的怀疑使得主机厂大多只针对规控系统进行仿真测试。这样使用场景下新一代仿真平台和传统商用软件相比无优势可言；
- 有限的测试场景库无法发挥新一代仿真平台在规模化仿真测试使用场景下的优势；
- 企业中不完整的智驾开发流程管理和工具链使得完整仿真测试平台集成难度加大，收益不明确。

以上需求与产品的失配可总结为新一代仿真平台通过对标头部L4自动驾驶企业及自我迭代走得过快，主机厂在人才配备、研发方式、工具链建设方面还未完成转型。加上智能驾驶系统测试方法论及相关标准法规的不确定性，很难预料目前走在前方的仿真产品形态是否会与企业完成转型后的需求吻合。

在这样的碰撞和机遇之下，采用开放开源形式研发的仿真测试平台及其生态有望连接科研，监管，产业三方，伴随智驾研发企业的成长和转型。OASIS Sim 定位为这样一个开放的仿真基础平台，以最小化的组织形式实现仿真关键功能。OASIS Sim 与 CARSMOS内其他开源项目组成完整智能驾驶研发工具链，降低研发门槛，提高研发自由度和可维护性。



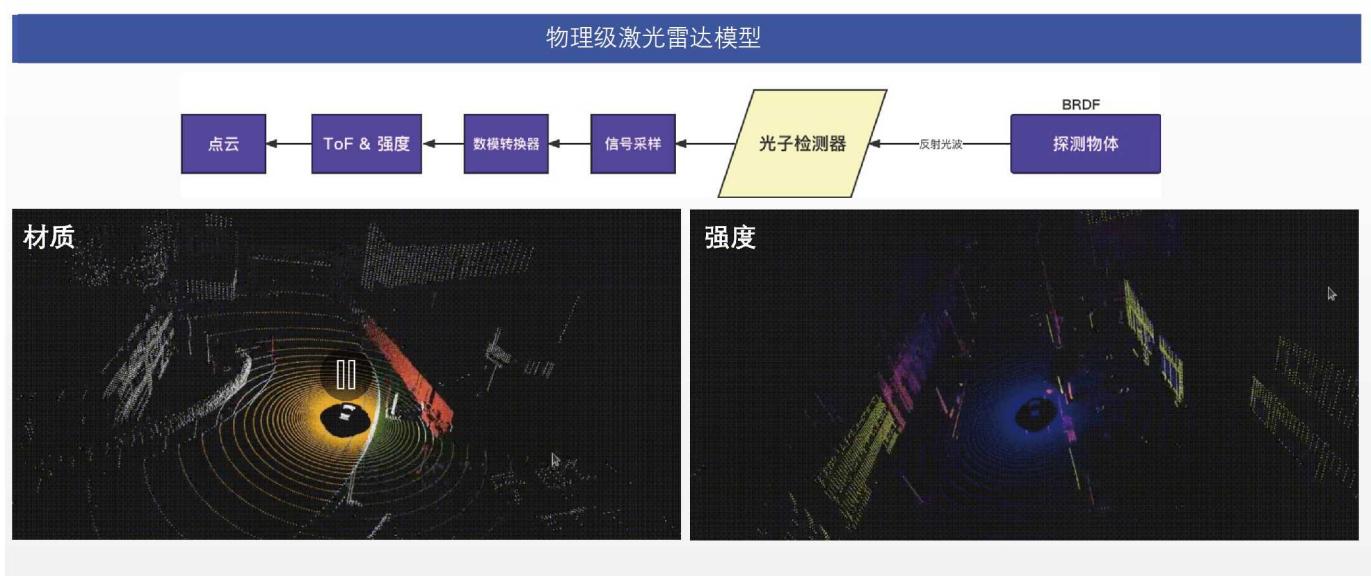


OASIS Sim由核心仿真器OASIS Sim Core，仿真测试管理平台OASIS Platform以及与传感器模型，车辆动力学模型，待测自动驾驶系统交互的各个接口组成。

OASIS Sim Core提供了仿真世界的执行逻辑。根据被测对象的不同，OASIS Sim Core可提供基于目标状态的低渲染仿真和基于游戏引擎的高保真渲染仿真。围绕核心仿真器，OASIS Sim平台定义了统一的接口形式，支持高精度传感器模型，车辆动力学模型，场景描

述，交通参与者行为模型和被测系统的集成。OASIS Sim实现了由测试定义，测试执行和测试结果分析组成的完整仿真业务流程。目前OASIS Sim Core基于开源仿真器Carla二次开发实现，但亦可替换为LGSVL，AirSim甚至VTD等商业仿真器。

传感器仿真插件通过传感器接口与OASIS Sim Core交互，可实现激光雷达，视觉传感器，毫米波雷达模型与OASIS Sim Core的联合仿真。



动力学仿真可通过仿真器自带的简单动力学模型实现，OASIS Sim Core通过Carla同时支持Carsim，EchoSim等专业动力学仿真软件。

场景描述通过OASIS DSL场景描述语言及位于OASIS Platform上的交互式场景编辑器实现。OASIS DSL语法兼容OpenScenario 2.0，同时OASIS Sim Core通过二次开发的Carla scenario_runner也支持90%以上的OpenScenario 1.0场景标签。静态场景描述支持OpenDrive 1.1。

交通流仿真支持Sumo、PTV Vissim等专业微观交通仿真软件。

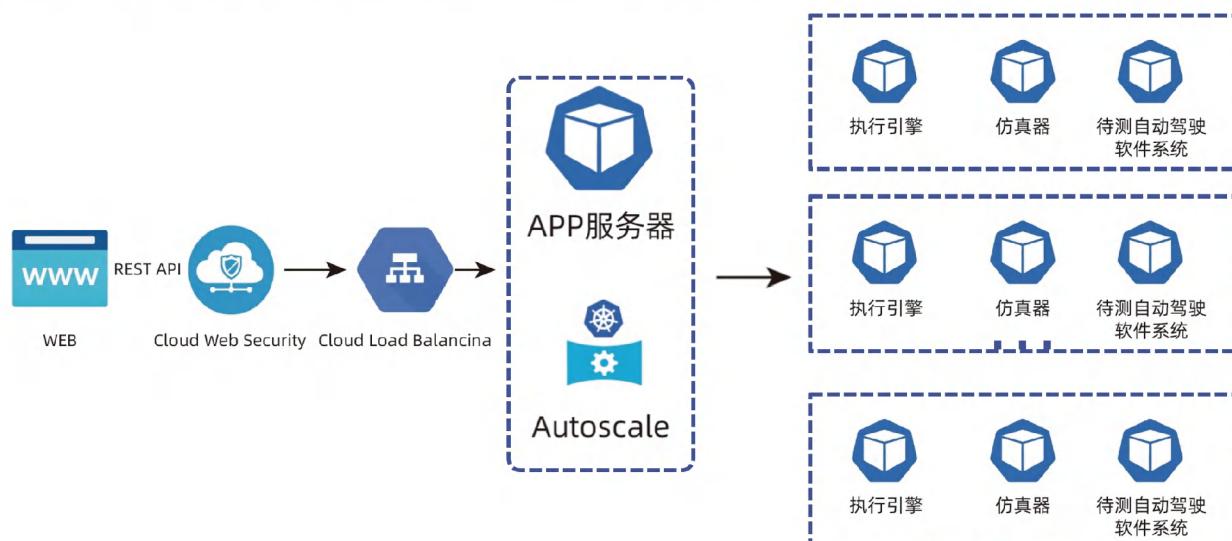
系统通信接口包含Dora_bridge,Ros_bridge及CyberRT_bridge。支持基于Dora，ROS1/2 (Au-

tware) 及CyberRT (Apollo) 开发的智能驾驶系统。对私有协议的支持可通过新的桥接器的开发完成。

OASIS Platform 负责与用户的交互，实现仿真任务定义，任务状态监控，仿真结果分析（例如日志查看与仿真回放）等业务。用户定义的仿真任务被发送至任务队列，由后端包含OASIS Sim Core的任务执行器拾取并执行。

OASIS Sim采用云原生架构，各种仿真组件以容器化微服务的形式部署，可被容器编排器统一管理。池化的任务执行器可根据负载在云端横向扩展。

针对本地的硬件在环和车辆在环仿真测试，OASIS Sim本地版将提供实时的目标级仿真和近实时的传感器仿真能力。



4.2.7.2 OASIS DSL 场景描述语言

功能场景: npc车辆切入并减速

```

scenario cut_in_and_slow:
# 定义测试路段，通过with关键字添加约束
Road path with:
  map(name: 'Town01.xodr') # 设置地图
  min_lanes(2) # 车道数量约束
# ego是待测车辆，通过with配置车辆参数
Car ego with:
  right_lane(path)
  speed(80km/h)
# 放置其他交通参与者
Motorcycle mc with:
  left_lane(path)
  behind(ego, 100m)
  speed(100km/h)
# 场景行为定义
behavior: parallel:
  ego.drive(path)
  serial:
    accelerate: mc.drive(path) with:
      position([5..100]m, behind: ego, at: start)
      position([5..15]m, ahead_of: ego, at: end)
      change_lane_to: mc.drive(path).with:

```



测试规约

障碍物识别:

$G(\text{ground(obs)} \wedge d(\text{obs}, \text{ego}) \leq d_{\text{range}} \rightarrow \text{detect}(\text{obs}) = \text{ground}(\text{obs}))$

碰撞避免:

$G(\text{obs} \rightarrow d(\text{ego}, \text{obs}) > p)$

红灯必停:

$G(\text{light} = \text{red} \rightarrow F_{[0, \tau_1]}(v_e = 0 \wedge \text{position}))$

超速检查:

$G(\text{speedlimited} = v_{\text{lim}} \rightarrow F_{[0, \tau_2]}(v_e < 0))$



基于场景的仿真测试是目前被广泛采用的测试方法。对场景的有效描述是实现规模化、自动化仿真测试的重要前置条件。

自动驾驶仿真测试场景库中通常包含由场景描述语言描述的各种具体场景、泛化生成的场景、及借助验证分析技术生成的缺陷场景及安全关键场景。通过对自动驾驶场景的标准化描述，不仅做到了使计算机能够解析及计算场景，而且满足了不同开发设计阶段对场景描述的需求，实现了场景重用，有利于形成高效的场景库管理机制，提高数据资源的利用率。

目前使用较广泛的场景描述方式包括自然语言、结构化数据以及形式化语言等。自然语言主要用于描述感性的和抽象的场景，不具备可执行性；结构化语言适合对简单、确定性场景的描述，但是不能够描述复杂的不确定场景；形式化语言可以描述复杂系统、不确定性事件的预期和行为，也具备可执行性，被认为是描述自动驾驶仿真测试场景的最佳方法。形式化场景描述语言不仅可以有效支持自动驾驶系统的仿真测试，还可以广泛

应用于其它人工智能系统的仿真测试验证，例如机器人、智能制造、无人机等。

OASIS DSL是一套面向动态交通场景描述的形式化领域特定语言，具备以上形式化语言的优势。OASIS DSL开源项目提供了语法设计，领域模型及语言在OASIS Sim仿真平台上的实现，包括基于OASIS DSL的交互式场景编辑器。

OASIS DSL的语法结构能够描述语言元素之间的组织关系以及如何通过这些关系推导出符合语法的场景表示。为了满足语言的表达能力以及语义对应关系，OASIS DSL 的语法结构采用上下文无关语法（Context-Free Grammar）的设计原则，以及采用巴科斯范式（BNF: Backus Normal Form）的形式化语法描述方式。作为一种元语言，BNF不仅能严格地表示语法规则，而且所描述的语法是与上下文无关的。BNF具有语法简单，表示明确，便于语法分析和编译的特点。此外，在设计与实现上述语法结构过程中，OASIS DSL还满足了以下6方面的语言特性：

• 表达能力：OASIS DSL支持不同抽象程度的场景描述，如功能场景、逻辑场景、具体场景以适应系统研发不同阶段对场景描述的不同需求。例如在系统早期设计阶段，系统架构师和功能安全专家主要围绕系统功能讨论场景。对功能场景的描述通常会简化为“跟车”、“上下高速匝道”、“泊车”这样的简单语句；而在系统集成测试阶段，测试工程师需要实现精确的具体测试场景，例如前车在特定相对距离，相对速度的条件下切入主车前方，以及沿特定的切入轨迹等。OASIS DSL从设计上满足以上用户对场景描述抽象层次的需求。

• 组合性：通过场景运算符支持任意粒度的场景组合，组合方式包括顺序组合、并行组合及选择组合等，将原子场景组合为测试场景。支持结构化场景构建的。

• 复用性：支持导入已实现的场景或功能模块，减小新场景搭建的工作量。鼓励场景描述的共享及二次开发。

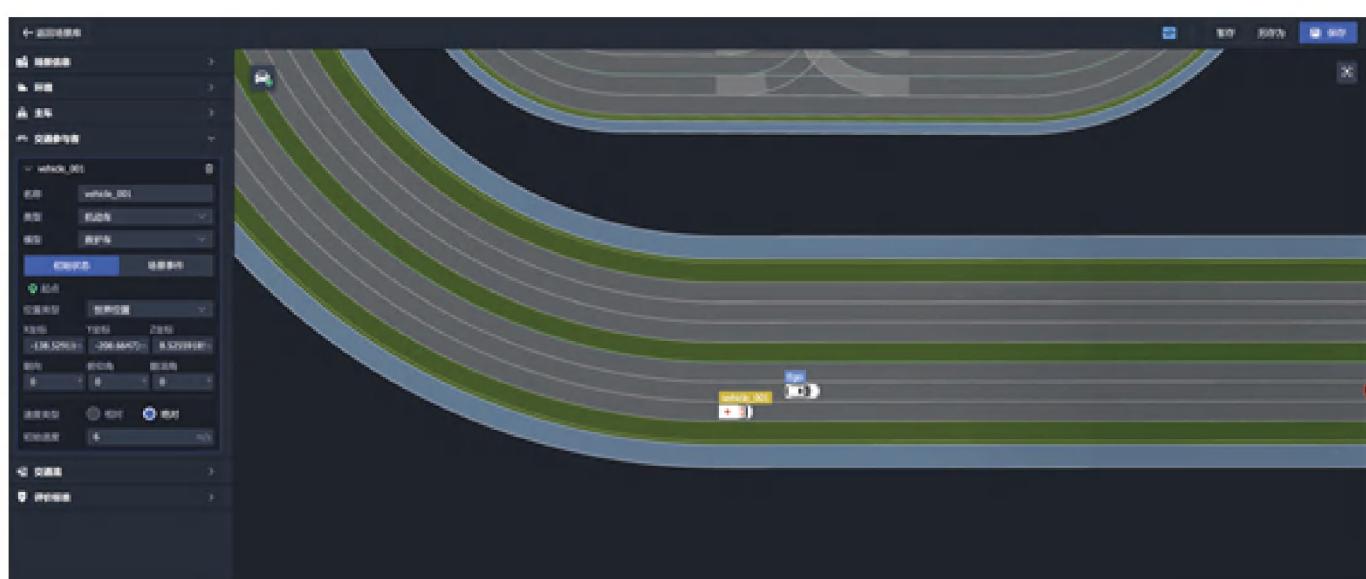
• 混合编程：支持调用外部程序或连接外部模型以达到扩展场景描述语言功能的目的。例如可调用python的统计功能实现场景参数的抽样，调用外部交通流仿真工具生成具有拟人驾驶行为的环境车辆(NPC :Non-PlayerCharacter)，以提高测试场景的丰富度和真实度。

• 可观测性：支持系统日志的收集，用户可通过外部

日志分析工具对仿真过程进行分析。OASIS DSL不仅包含形式化的场景描述，还包括了基于时序逻辑的测试规约描述。支持对复杂测试目标的定义可自动化判断，例如测试主车是否在红灯时停在了停车线之前。基于OASIS DSL的测试规约可实现完整的仿真测试评价体系；

• 支持库及配套工具：除上述基础语言特性外，OASIS DSL还包含相关的库函数及配套工具。例如对基础场景的预定义，包括简单功能场景，标准法律法规场景，危险工况场景等。配套工具包含与仿真测试平台适配的接口，场景的图形化编辑及预览工具等，帮助用户快速开展工作。

为了尽快推动自动驾驶技术及产业发展，亟待建立基于标准化的仿真测试验证和认证体系框架，并利用开源+商业的合作模式，来实现云服务的开放式合作社区，形成良性循环。OASIS DSL场景描述语言能够在标准化的仿真测试评估中充分发挥其作用和价值，有助于推动自动驾驶技术产品快速迭代优化，形成可闭环、可落地的实际应用能力。达成这样的目标还需要主机厂、自动驾驶方案供应商、研究机构和标准组织等产学研单位，充分利用CARSMOS开放平台集思广益、共同合作完成。



4.3 采用在线仿真平台促进高质量智能汽车教育

智能汽车涉及的技术和学科是多样化的，涉及到电子、软件、AI、自动化控制、汽车等综合知识。按照传统科系的划分办法，必然培养出瞎子摸象式的人才，有必要为智能汽车编写一套符合智能汽车知识体系的系统化教材，这需要汇集产业界和学术界的师资力量、训练数据和应用场景。

在课程教授上，也需要创新。例如自动驾驶涉及大量的算法开发，虽然软件教育也大量采用计算机辅助教育，但是自动驾驶的算法开发一定需要物理世界的仿真系统，提供算法训练需要的数据和模拟环境，而且可以自动判断算法的质量。学生们可以直接在仿真环境测试出结果，通过实践项目来学习自驾技术。课程的考试也可以通过在线平台完成。

需要借鉴Udacity等领先在线平台，基于云平台实现高度交互的学习，练习，测试和竞赛功能，自动化完成作业的批改，课程管理，考试打分等功能，降低老师的工作强度，例如：

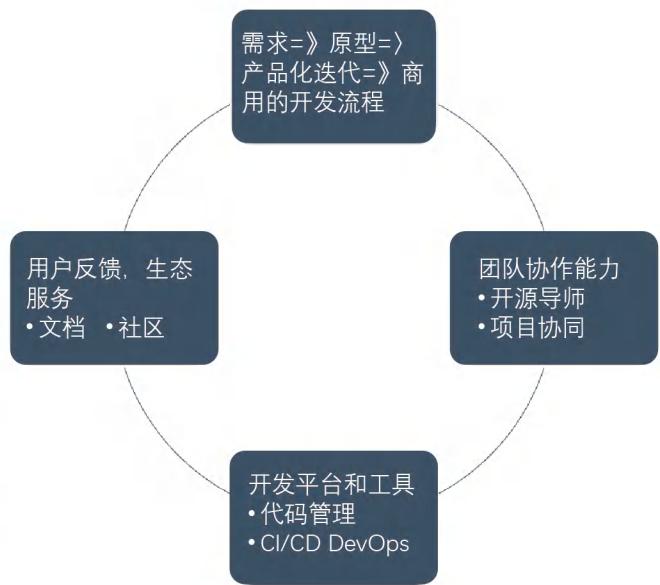
- ◎ 课程要点可以通过观看视频和老师在线讲解完成，包括以自动驾驶为场景学习最新的编程语言，编程框架，系统集成等等。
- ◎ 学生学习完知识点后，马上可以在在线平台完成作业。
- ◎ 学生可以使用仿真环境提供的训练环境编写测试自己的代码，增加学习趣味性。
- ◎ 仿真环境也具备测试能力，可以自动化判断答案的对错和质量。
- ◎ 学生看到系统给出的判分，又可以进一步修改错误，改进方案质量，这样快速的反馈迭代和能够让学生在学习过程中保持高度专注力达到很好的效果。
- ◎ 学生学习的过程中产生的作业和考试提交都可以存档到版本管理平台，代码还可以做重复度扫描等，避免作弊。

这个平台也可以用于科研和竞赛，学生可以在这个平台完成毕业设计，硕士论文和博士科研。

智能汽车教育要提供完整的学习路径，从简单ODD（Operation Design Domain）到复杂ODD，从仿真到实车的连续体验，也通过竞赛选拔优秀学生，参与开源项目贡献，为科研和企业培育人才。

智能汽车教育离不开开源平台和社区。通过引入开源硬件/软件，打造统一的起步教育平台，提升计算机教育水平，引入开源导师，培训学生做开源贡献，即锻炼了能力，也增加了声誉度。

开源软件协作和开发也是一项重要的软件技能，学生通过智能汽车教育也可以参与到CARS莫斯社区贡献中。在参与项目贡献的过程中，学习现代开源软件工程的全流程，包括从如何管理和制定项目需求，产生产品开发路标，通过和用户的不断互动完善软件性能，最终交付功能完善的产品，学习使用各种开源平台工具，通过在线平台和其他开源开发者协作沟通和开发，同时培养服务客户、建设软件生态的能力，学子如何撰写技术文档，使用说明，演示程序，如何利用在线讲座，线下交流和社交媒体宣传项目，持续获得用户反馈。



4.4 智能汽车全栈参考设计平台，研究转型为数据场景驱动加开源竞赛

CARSMOS构建的全栈智能汽车平台，使得研究人员可以使用开源的软硬件平台来和产业对标，填补了产业和学术之间的沟壑，使得智能汽车的研究和IT产业一样。研究人员可以聚集解决基础性的算法、软件平台、软件工程方面的挑战，而研究成果可以很快被产业采用。

- EDIT EV 开源电动滑板底盘设计和PULSAR统一EEA架构，通过标准化的X-by-Wire设计，兼容车企的SDV API标准，填补了研究和产业平台的鸿沟，通过标准化，极大降低了成本，满足了智能驾驶的教学和科研需要。学生可以一人一车，相比之下MIT开源的F1TENTH的智能小车，物理尺寸太小，使用的传感器和摄像头也比较简单，同时缺乏配套的仿真环境和软件2.0基础工具的支持，所以对科研的意义不大。

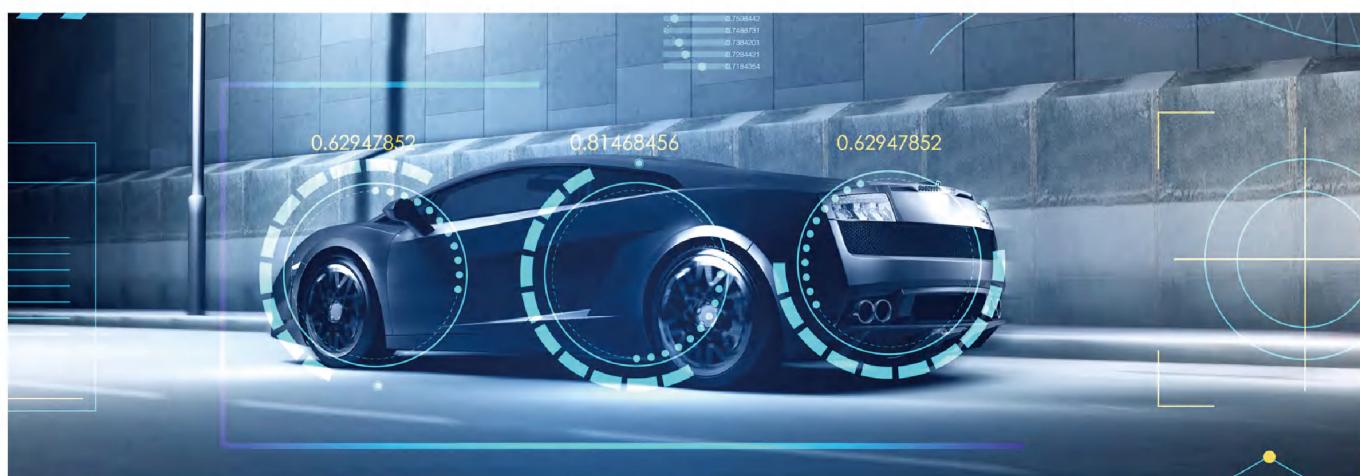
- 统一的EtymOS Kit配合 EtymOS Wheel 的数据飞轮工具链，让EDIT滑板底盘的用户可以采集标准化的全量数据集，这些数据集可以开源方式在社区共享，这些数据集规模很容易超越目前头部企业开放的数据集，学术界可以基于这个数据集训练对标头部公司的大模型。

- 基于 EtymOS Kit 产生的标准数据集，学术界可以通过竞赛方式产生出对标头部车企的大模型，形成了对EtymOS的快速迭代。EtymOS可以对标其他领域的大模

型。基于EtymOS，会形成自驾任务的社区，针对不同的领域训练不同的任务，使得学术界不需要在不同的不兼容的小数据集上重复造轮子。下表是全球领先的面向自驾业务的开放数据集，这些数据集提供了详尽的标注、高清2D和3D语意地图，以及开发SDK，并且在数据集上定义了感知、预测和规划任务的基准benchmark。然后开放给全球的研究者百花齐放公开打擂，各个研究团队的成绩实时进入公开的全球排行榜，其中的优胜者以科研论文的方式投稿发表自己的研究方法和成果。可以说在自动驾驶领域，数据集是产生高质量科研的基础。

- EtymOS Kit 产生的场景数据可以被OASIS场景描述语言抽象泛化为场景库。场景库可以驱动仿真系统产生大量相似场景，对EtymOS及其子任务模型做验证。场景库也可以是车企提供的，用于验证EtymOS的成熟度，成为车企和学术界对标的另一条途径。场景库也可以作为端到端自驾能力评估的挑战标准，为不同算法架构性能比较提供一个基准。

- 基于EtymOS，也可以更好的设计硬件加速器。AutoExpress提供的基于chiplet封装标准的NPU HUB接口标准和DSA参考设计，和DSA敏捷设计的开源社区衔接起来，为学术界提供了一个可以把算法工程化、芯片化的路径。



参考：<https://f1tenth.org/>

数据机构	比赛项目	传感器配置	数据量	最新比赛冠军
KITTI	2D/3D 多传感器感知 任务	6xcameras, 1x LIDAR, 5xRADAR, GPS, IMU	德国卡尔斯鲁厄的几个郊区，在 73.7 公里的行驶距离内对应超过 32 万张图像和 10 万次激光扫描 2D/3D 同步标注数据，HD地图	自驾全量数据先驱
Motional	2D/3D 多传感器感知 任务 nuScene	6xCamera, 5xRadar, 1xLidar	15 小时在波士顿和新加坡市区的驾 驶数据，提供包括 6xCamera+5xRadar+1xLidar 在 内的全量数据，提供同步的 3D/2D 标注和产生的 HD 地图 https://www.nuscenes.org/nuscenes#data-collection	BEVFormer 等 BEV 框架
	预测和规划任务 nuPlan	6xCamera, 5xRadar, 1xLidar	来自波士顿、匹兹堡、拉斯维加斯 和新加坡4个城市的1500小时驾驶 数据。 大约1000小时来自拉斯维加斯， 包含了2D高清的语意地图	领先的预测和 规划数据集
Waymo	预测和规划任务	5xCamera, 5xLidar	该数据集由 103,354 个片段组成， 每个片段包含 20 秒的 10Hz 对象 轨迹和该片段所覆盖区域的地图数 据。这些片段进一步分为 9 秒的窗 口（1 秒的历史数据和 8 秒的未来 数据），重叠 5 秒。标注好的高清 2D语意地图	在预测和规划领域出现 了领先的DenseTNT 等算法
Carla Leaderboard	基于高清地图 的赛道	任意配置	提供包括高清地图和全量的 GPS+Lidar+Camer 的合 成数据	出现了 MMFN+, Pilot 等模型
	基于纯传感器 的赛道	任意配置	提供全量的 GPS+Lidar+Camer 的合成数据	在端到端的大模型领域 领先，小出现了 LAV, TCP, Interfuser 等领 先的模型

4.5 产学研结合：统一平台，加速落地

智能车企面临转型的巨大压力，需要高校和培训机构能够大量培养面向智能汽车开发的中高级软件人才，为此也需要做和车企类似的转型。

- EDIT EV电动滑板底盘，PULSAR EEA和DORA中间件，可以支持车企定义的API和软件平台。车企也可以通过开源的方式，在这个计算平台上集成自己的中间件架构，做到教育培训和产业的无缝对接。

- 基于数据闭环的自驾开发方法是大多数车企缺乏的软件2.0能力。EtymOS基于统一sensor数据的大模型迭代，可以为车企赋能。EtymOS算法社区可以不断通过竞赛方式产生新的大模型，车企可以提供自己的场景数据来验证大模型的性能和效率。这样加快了研究成果在企业落地的速度。

- 企业也可以提供自己的ODD场景，CARSMOS社区可以帮助车企做擂台赛。客户的ODD的物理环境可以建模为地图和环境模型，通过OASIS场景描述语言构建测试和竞赛场景，同时社区也可以对客户的ODD场景提供基线的算法性能评估，并推荐最优算法。接下来，社区可以针对客户的ODD场景定义性能指标，搭建竞赛平台，鼓励学术界来打擂台，最终帮助企业选出最优性能的算法。算法也可以通过DORA计算平台在实际车载计算平台在物理环境验证性能，进一步通过EtymOS Wheel数据飞轮架构的循环来优化大模型。

4.6 形成关键产业标准

如前所述，智能汽车产业的发展需要解决一系列的挑战，需要调动产业和社会的力量，全面合作形成开放标准化的产业体系和生态。由于智能汽车产业特别是自动驾驶技术还在持续迭代发展演进中，相关产业标准化也要打破传统的标准定义方式，既要避免各自为政的碎片化开发，也要防止按部就班的教科书式定义，而是应该产学研合作，以商业需求为目标，以研究创新为驱动力，以技术开发为实现方式，采用学术研究、开源开发以及产业标准相结合的快速递归迭代模式。

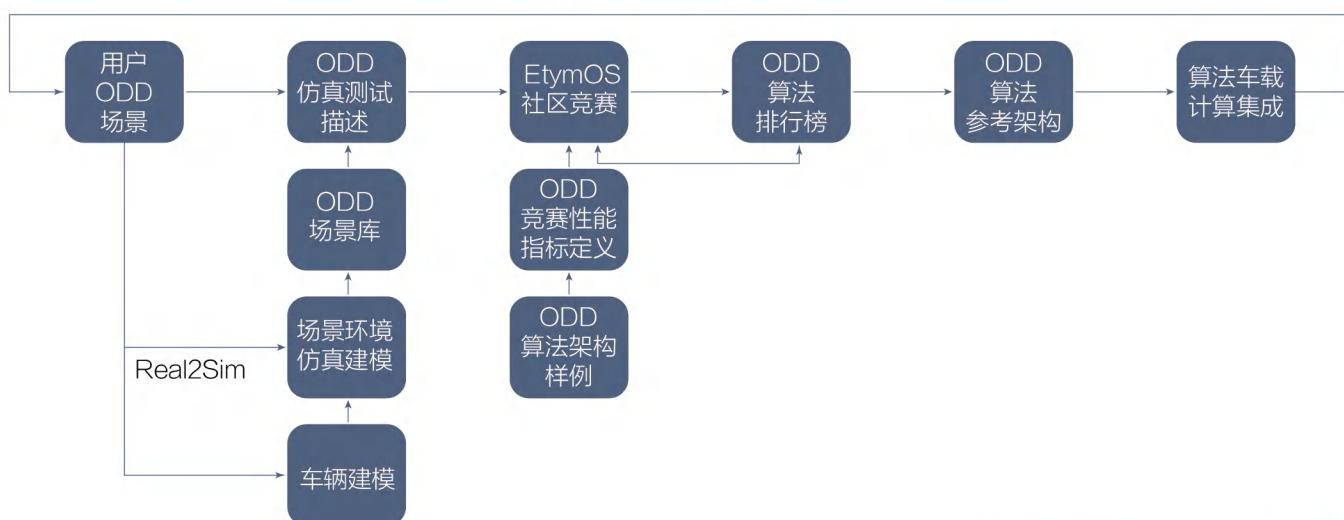


整体原则采用“从大处着眼，从小处着手”（Think Big, Start Small）：

- 从大处着眼（Think Big）：综合考虑国家战略、产业愿景、商业需求和技术发展趋势，形成总体目标框架，明确各个业务、功能和技术层面的整体逻辑关系及交互合作方式，依托事实标准进行迭代，并避免对各个具体实现功能模块能力的过度标准化。

- 从小处着手（Start Small）：基于总体目标框架，着重于各个实现功能模块能力的快速迭代开发，形成事实标准，并对各个大模块例如硬件平台和软件平台的接口，

实际场景验证+算法迭代



模拟仿真、测试和认证规范，以及统一开放数据集等关键能力进行国家和国际标准化。其中最关键的问题，是硬件平台的开放性，以及软件平台的开源化。纵观历史，IT产业的竞争主要体现在开放硬件架构和与之配套的开源软件生态两个方面。

- 硬件平台开放性：IBM PC之前的计算机，无论是大型机还是刚出现的个人电脑（Apple, Commodore）都是私有架构的，意味着用户必须从一家公司采购所有的主机、外设、操作系统和软件。1981年IBM发售的IBM PC标志着全新的开放式电脑时代的到来，IBM PC第一次采用了开放硬件和软件架构，它采用货架（off-the-shelf）硬件器件，主板设计提供了内存、外设、存储的可扩展性，并且采用了微软的MS-DOS操作系统，PC的硬件设计文档如此细致，以至很快出现了兼容PC的第三方配件和外设，甚至PC兼容机，可以运行PC上可以运行的软件。而软件层面基于MS-DOS,出现了文字处理，数据库，游戏等丰富的软件生态。今天，遵循开放架构的个人计算机和服务器设计依然是IT产业的基石。

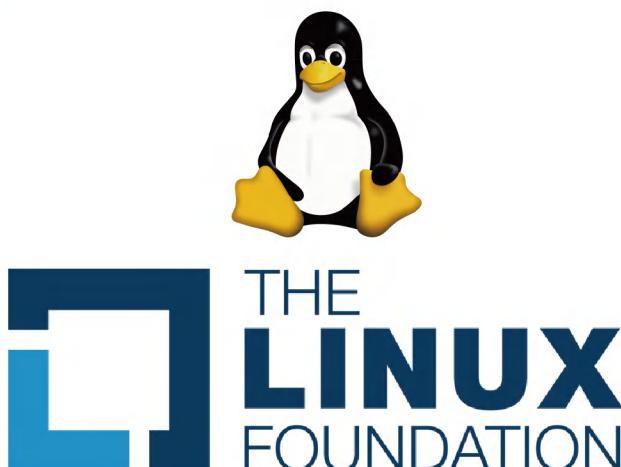


- 软件平台开源化：1991年Linux 1.0的发布标志着大规模开源软件时代的到来。Linux南向接口提供了对硬件系统的抽象，硬件厂家提供配套自己硬件设备的驱动程序，实现Linux的系统API，Linux北向接口提供了操作系统服务的API和开发工具，应用开发可以在抽象化的计算机系统上进行，可以在不同的计算机上运行。经过30多年的发展，Linux也成为IT软件产业的基石，广泛应用

于消费者设备、云计算中心和工业设备。管理Linux的Linux基金会也成为全球最大的开源基金会。

那么智能汽车在成为安着轮子的计算机的转型中也必然遵循IT产业的发展规律，其产业标准必然走向开放硬件架构和开源软件生态。如下表所示，CARS莫斯项目和其他开源项目已经在关键的开放硬件架构和开源软件生态上形成生态和标准布局，借助中国智能软件汽车产业的高速发展带来的市场规模，和开源平台促成的产、学、研的紧密合作，形成智能汽车的事实标准，为全球智能汽车产业做出符合中国市场地位的贡献。

	私有架构、闭源	开放架构、开源
电动滑板底盘	私有电动滑板底盘	EDIT, PIX Moving
指令集	ARM	RISC-V
AI加速器	NVidia和其他	开源DSA
CHIPLET互联	各大SoC厂商私有架构	AutoExpress, nCIE
OS	QNX	Linux
中间件	AutoSAR	DORA
自动驾驶OS	私有	EtymOS
仿真系统	Ominiverse	OASIS平台
场景描述语言	OpenScenario	OASIS DSL



5. 结束语

千里元遨，始于足下。CARS莫斯开源智能出行项目，愿景清晰，任重道远。在此诚邀业界公司、机构、专家学者、工程师、学生们积极参与，共同努力，构建一个平等自治的生机勃勃的开源社区，实现在开源智能驾驶领域教育、研究、产业化落地的历史使命。

附录: CARS莫斯项目网站

CARS莫斯项目介绍:

<http://www.carsmos.ai/>

开源代码库:

<https://github.com/carsmos>

<https://gitee.com/carsmos>

附录: CARS莫斯贡献单位、白皮书撰写组

深信科创	杨子江、潘余曦、黄浴
元毅智行	丁新民、李伟军
Futurewei Technologies	何勇、孙栋
北极熊芯	马恺声、高冲霄
江苏波霎科技	徐匡一
Open Motors	廖天行 (Tin Hang Liu)

附录: 专业术语

ODD	Operation Design Domain	运营设计域
DORA	Dataflow-Oriented Robotics Architecture	面向数据流的机器人架构
MaaS	Mobility-as-a-Service	出行即服务
AI	Artificial Intelligence	人工智能
EEA	Electrical/Electronic Architecture	电气电子架构
Chiplet	Tiny chip modules to build up a SoC	芯粒
ESSA	Elastic Stateful Serverless Architecture	弹性无服务器服务架构
DSA	Domain-Specific Architecture	特定领域计算架构
SDV	Software-Defined Vehicle	软件定义汽车
XaaS	Everything-as-a-Service	一切皆服务
DSL	Domain-Specific Language	特定领域语言

免责声明

本白皮书由元遨 CARSMOS 项目工作组集体编写，力图综合描述智能出行行业的历史、现状和愿景，仅供参考，可能随时变更。我们无法保证本白皮书中所作陈述或结论的准确性，对于因使用、参考或依赖本白皮书内容而引起的损失或损害（无论直接、间接、后果性或任何其他类型的损失或损害），我们不承担任何责任。本白皮书可能包含对第三方数据和行业出版物的引用。据我们所知，本白皮书中复制的信息是准确的，其中包含的估计和假设是合理的。但是，我们不保证这些数据的准确性或完整性。虽然我们相信本白皮书中复制的信息和数据是从可靠来源取得，但我们并未独立审核本白皮书中提到的第三方来源的任何信息或数据，也没有确定此类信息来源所依据的基础假设。

