车云对等环境加速软件定义汽车

Girish Shirasat
Arm 汽车业务软件战略和架构师总监

Stefano Marzani
亚马逊云科技自动驾驶汽车业务熟悉专业解决方案架构师

arm

亚马逊云科技

白皮书

愿景

2025 年。一家大型汽车供应商的软件工程师 Judith 正在家里工作，以解决客户对自动巡航控制功能的反馈。她的工作均在一个云中工作区内进行，在这里可以访问与有问题行为相关的大量数据，并利用这些数据调整控制算法。她使用一个包含车辆动力学模拟器的闭环软件测试平台评估并验证新功能，随后将这个软件包标记为已准备好可以部署。与此同时，数千英里外另一片大陆上，就职于一家设计公司的 Mark，正在对 UI 软件进行最后润色，这些软件需要为一个新市场进行本地化，并搭载到该市场下周即将上市的一款汽车中。Mark 在 OEM 厂商提供的云平台中修改了软件，并通过虚拟代理对其进行了大规模的并行测试。测试通过后，他将新的 UI 软件包标记为已经准备好可以部署。就职于这家 OEM 厂商总部的 DevOps 验证工程师 Kay 看到了这两个新软件包，随后进行了最终的硬件闭环测试，借此全面验证并认证即将部署给公司未来几周即将发布的所有量产车辆的内容。

简介

随着汽车行业逐渐拥抱软件定义的未来，很多 OEM 厂商开始追求这样一种愿景：借助上文所描述的这种敏捷性和灵活性来开发软件，进而在不影响质量或安全性的前提下，循序渐进地向车辆提供新功能。

云原生方法 [1] 能够在功能安全理念和实时执行方面保留汽车的某些特定特征，已成为通过现代化数字服务和简单易用的车载应用程序，打造此类以软件为中心的生态系统的关键所在。通过创新、高效的工作流程，使得越来越多的开发者能够参与到开发过程中，同时还能帮助汽车公司缩短开发时间，获得快速发展和功能更新所需的敏捷性，进而满足现代消费者快速变化的预期。

实现汽车云原生开发管道的第一个关键技术推动因素是在云环境和作为目标的嵌入式汽车边缘平台之间构建一种对等环境，并在此基础上部署工作负载。正如领先的信息架构师 Kevin Hoffman 所描述的 [2]：

*“ 将严谨和纪律应用于**对等环境**，这样做的目的在于让您的团队和整个组织确信应用程序将能**随处运行**。”*

这种对等使得厂商可以直接在云中开发、验证并确认，而无需使用开发者桌面上摆放的嵌入式硬件，进而可从根本上缩短整个汽车价值链中解决方案投放市场所需的时间，开发出可移植性更强的应用程序，并让开发工作流面向未来做好准备。

为实现这样的对等环境，并为车载软件日益增加的需求提供支持 [3]，车辆内部需要具备一种尺寸适中的多核心嵌入式系统。本文我们将探讨如何在汽车系统的开发过程中实现云原生方法，并将重点探讨如何在云中的执行环境和车辆边缘位置之间实现环境对等，同时我们还将探讨这种方法在加速软件定义的汽车的上市速度方面所能产生的影响。

软件定义的汽车（SDV）是什么？

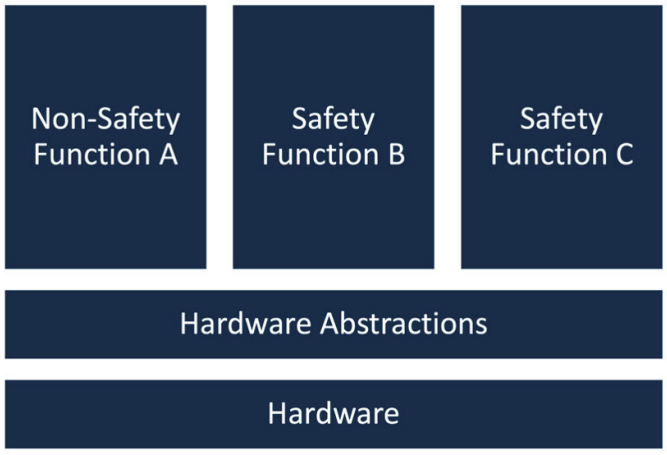


图1：在抽象的硬件基础上实现的车辆功能

在软件定义的汽车这个语境下，“ 软件定义 ” 是指通过共享或集中部署的计算设备上运行的软件功能和服务，来体现或实现汽车功能，而不再以单独的物理 ECU 控制单元或类似机制来实现的做法。此外，“ 软件定义 ” 还意味着在整个汽车系统开发生命周期（包括生产前和生产后）中，开发和部署这些软件定义的功能所需的能力需要以敏捷的方式实现。如图 1 所示，理想情况下，这些软件服务应该是与特定硬件或供应商无关的，可将其理解为为了实现特定功能而需要的一系列数据提供（传感器）、数据处理（应用程序逻辑）以及数据消耗（执行装置）服务。这种程度的灵活性可以让 OEM 厂商以及纯粹的软件公司围绕特定应用程序领域打造全新功能。最近十多年来，在智能手机这样的细分市场中，给已售出产品添加新功能已经成为一种普遍做法，现在也有越来越多的人希望能为包括汽车在内的传统嵌入式设备实现类似的方法。

SDV 可以概括为一种具备以下特征的系统：

- 具备网络连接能力的车辆，能够以 “ 大闭环 ” 的视角通过无线通信技术持续传输数据并接收软件更新 [5]。
- 从底层硬件中抽象而来的软件。
- 车辆功能与能力均通过软件实现，在车辆生命周期内，软件可升级，可管理。
- 从汽车开发运维（DevOps）角度来看，需要从云端到车辆边缘，采用能跨越不同硬件平台的云原生设计范式。
- 为重要程度各异的汽车应用程序提供混合管理能力，为工作负载提供不同级别的防故障保证 [6]。

SDV中的云原生

“ 软件定义 ” 并非新概念。除了上文列举的智能手机，这种概念早已通过软件定义的网络、软件定义的存储以及软件定义的计算等设计范式广泛应用于通信和数据中心行业，并且在企业应用程序领域也极为普遍。在这些领域中，云原生概念已成功用于通过丰富的设计模式和生态系统工具交付软件定义的系统 [12]。因此探索云原生概念在汽车领域的应用也就成了一种非常合理的做法，借此，汽车行业可以通过围绕云原生开发的庞大技术和商业生态系统提高创新的效率和速度。

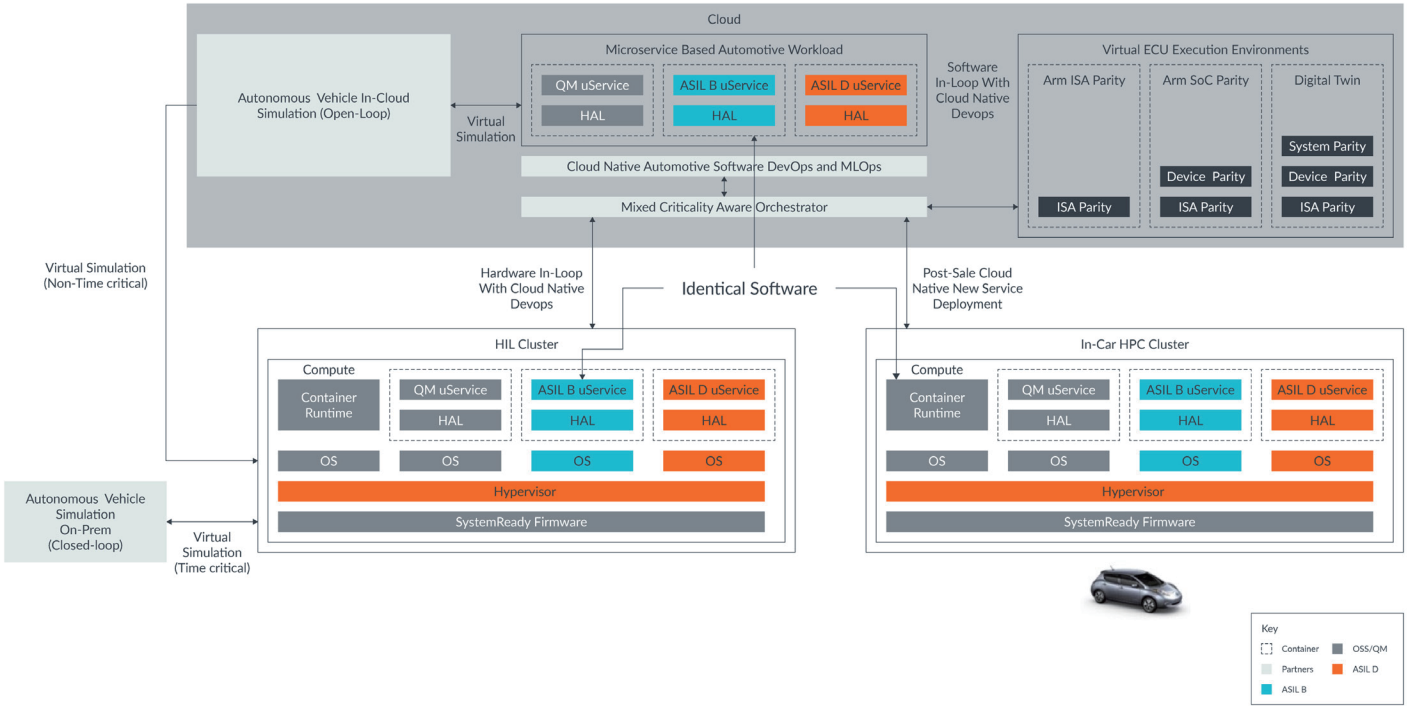


图2：云原生概念在汽车系统开发中的运用

图2所示场景很容易会让我们想到：一位开发者正坐在咖啡馆里操作笔记本电脑连接到云端，开发信息娱乐或高级驾驶辅助系统（ADAS），例如自适应巡航控制或横向线路控制应用程序，提交代码后即可在云端的虚拟执行环境中触发构建和集成周期，只要在云和汽车边缘之间建立对等环境，这个虚拟环境就能与现实世界中的目标汽车系统完全一致。云和汽车边缘之间可实现的环境对等程度以及模拟速度会对开发者的效率产生直接影响。

随着汽车系统软件的复杂性开始超过波音 787 梦想客机 [7]，汽车行业的开发者可能更愿意使用经过验证的，基于面向服务的架构 / 微服务的软件设计模式，并使用容器为工作负载的可移动性和复杂性奠定技术基础 [11,12]。汽车应用程序的开发带来了独特挑战，但在容器的使用已成为常态的企业或智能手机领域，根本未曾考虑过这些挑战。取决于有关安全性和实时性的要求，汽车的工作负载从本质来看可能是混合且关键的，构成工作负载的大量微服务除了要复合相应的安全要求之外，还会受到特定空间与时间要求的约束。根据应用程序的不同，其中一些微服务可能还需要考虑额外的要求，例如质量管理（QM）或达到 ISO 26262 规范所定义的 ASIL-B/Avcv-D 完整性级别。这就需要将在安全性方面获得认证的编译器和相关工具集成到云端工具框架中。此外，部署汽车应用程序的硬件原本是高

度分布式的（现代车辆往往会搭载上百个电子控制单元，即 ECU），并且这些硬件有着多样化的本质特征，因此很难实现软件与硬件的解耦。最后，现有的云原生基础架构（包括根据企业级平台的感知，将微服务部署到最佳硬件节点的编排器）也需要进行拓展，以便使其能更深入地了解这些独特的嵌入式汽车硬件系统的功能，进而以最优化的方式部署微服务。

当所有这些要求和问题都找到适合的解决方案后，开发者即可在云端安全地运行模拟并利用云所提供的固有优势，例如可扩展性和弹性。举例来说，作为 DevOps 基础架构的一部分，开发者可能希望使用云中运行的模拟器来运行各种模拟的操作设计领域（ODD），进而进行完整的软件在环（Software-in-the-loop，SiL）验证，借此只需为待测软件提供大量模拟数据即可快速验证输出结果。而如此庞大的真实或模拟环境还可以进一步扩展，通过同时执行数千个内核同时验证上千个场景。这种规模是硬件在环（Hardware-in-the-loop，HiL）环境中的嵌入式系统完全无法实现的。

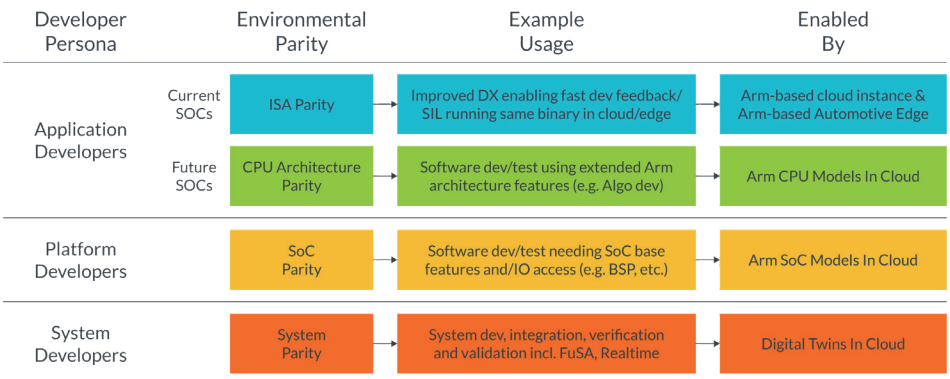
基于对客户的初步调查，我们估计在能够实现环境对等的情况下，目前通过 HiL 环境执行的测试中约有 70% 可以迁移到云端 SiL 环境，并充分利用云提供的可扩展性。

某些功能始终需要在硬件上进行验证，例如与嵌入式系统的物理特性息息相关的输入 / 输出测试，但通过从云转向车辆边缘，从系统开发的角度来看，我们有望借助云原生编排技术将在云中通过了验证的软件部署到实体 ECU 中，进而在实验室内进行 HiL 验证，随后再对其进行扩展，通过售后服务部署给已售出车辆。

云到车辆边缘的对等环境帮助汽车行业实现云原生开发

云原生汽车系统开发对于基础架构有一个关键要求：需要借助多种与物理环境高度相似且尽可能与车辆边缘环境实现对等的虚拟系统执行环境进行开发、集成和验证工作。实现这种高水平的环境对等，可对开发者的反馈环路产生直接影响，帮助他们进一步提高开发效率 [8]。接下来让我们从不同开发者的角度进一步看看这种对等环境，以及该如何帮助不同的参与者和角色在云中实现这样的对等环境。

图3：云到车辆边缘的对等环境



应用程序开发者

从汽车开发者的角度来看，他们对云端执行环境的主要期待是在云端和汽车边缘环境之间实现指令集架构（ISA）和 CPU 架构的对等。以基于 Arm 架构的汽车计算平台为例，现在已经可以使用 Amazon Graviton 提供的，基于 Arm 架构的云实例托管云环境。这是一个重要的里程碑，借此可实现一种基于 Arm 架构，完整且端到端的云到汽车边缘计算连续体。从以下几方面考虑，这是一种非常理想的开发环境：

- 开发者效率：正如开发者有效性专家 Tim Cochran 的观点，减少开发者反馈环路对提高开发者效率，进而改善公司盈利能力至关重要 [9]。在 Arm 架构的环境中为 Arm 设备开发代码所能获得的好处包括：
  - 避免交叉编译。交叉编译是指在一个架构的开发平台上编译代码，并在另一种架构的目标平台上运行。本质上，这种做法为开发 - 部署 - 测试过程增加了一个额外的步骤，扩大了开发者的反馈环路。
  - 交叉编译还为软件引入了额外的潜在错误来源。
  - 在云中托管 Arm CPU 模型还可适用于单凭 ISA 对等已无法满足要求，同时还要实现 CPU 架构对等的场景。在不同架构的系统上进行模拟，这是一种缓慢低效的做法。但在“Arm 上的 Arm”环境中可以直接使用 Arm 虚拟机监视程序扩展，进而更快速地执行模型。

- 性能优化。有一种观点认为编译器会实现所需的全部优化，但在真正为软件库或功能模块进行优化时，开发者依然需要偶尔以手工方式针对特定架构进行优化，以便让关键组件能够发挥出全部性能。目前的很多合作伙伴与第三方生态系统提供商恰恰就是这样做的。这种情况下，云和边缘的对等可以帮助优化措施实现从开发到部署的“ 无缝 ”过渡。

平台开发者

除了应用程序的开发，现代化的车辆还涉及另一种主要的软件层，即基础平台系统软件，这包括固件、OS/RTOS、设备驱动程序以及相关组件。平台开发者需要一种能使用 CPU 模拟 SoC（片上系统）的虚拟环境，要借助额外特性满足对某些功能的需求，同时还要针对相应 SoC 系统架构提供必要的设备接口。为了实现这些要求，可在云中托管虚拟的 SoC 模型并将其作为组件集成到整个汽车软件 CI/CD 框架中。这方面的一个重要问题是让最基础的系统架构实现标准化，借此无需针对每个平台进行改动，或只需要很少量的改动，即可用市售操作系统镜像直接引导。这也是 Arm SystemReady 认证项目希望解决的一个直接挑战，目前该项目已经获得了广泛的行业支持。

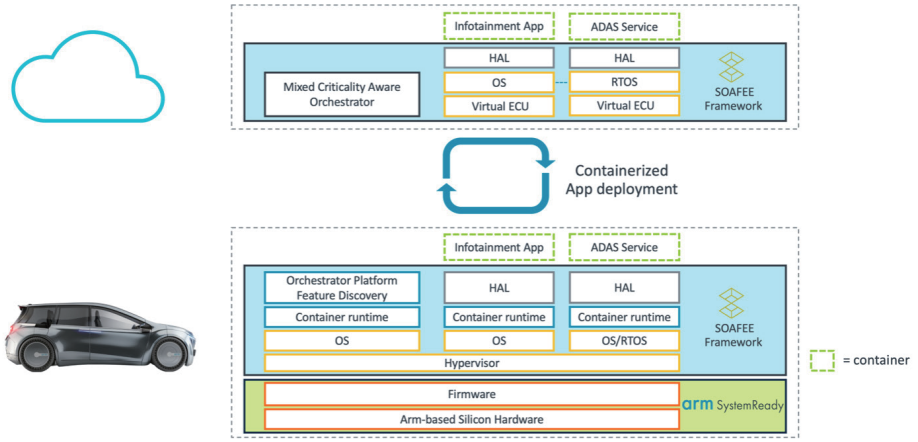
系统开发者

正如上文所述，汽车系统上运行的工作负载从本质上来看是混合且关键，并且是分布式的。为了能够开发、验证并确认这些工作负载，必须有一个托管在云中的车辆数字孪生体，并通过该数字孪生体模拟系统在功能性和非功能性方面的一切，例如实时性、安全性以及性能特征。在云中运行的过程中，这种数字孪生体还必须能对物理环境中遇到的不同操作设计领域进行建模。目前已经有很多大型的生态系统合作伙伴在开发这样的汽车数字孪生体技术，并在弥合环境对等性缺口方面取得了显著进展。

SOAFEE —— 行业主导的汽车云原生开发推动倡议

为了让本文开篇所畅想的咖啡馆中的 ADAS 开发者场景变为现实，整个价值链中的合作伙伴生态系统必须紧密结合并相互协调。Arm 与亚马逊云科技等重要合作伙伴携手，最近公布了 SOAFEE（Scalable Open Architecture for Embedded Edge，面向嵌入式边缘的可扩展开放架构）倡议。SOAFEE 为汽车应用程序提供了针对混合且关键能力增强的云原生架构，还将提供开源的参考实现，借此为商用和非商用产品提供支持。SOAFEE 由一个行业主导的特殊兴趣小组（SIG）推动，该小组正在基于开放标准定义 SOAFEE 架构并提供开源的实现。有关该倡议的更多信息请访问 SOAFEE 官网。

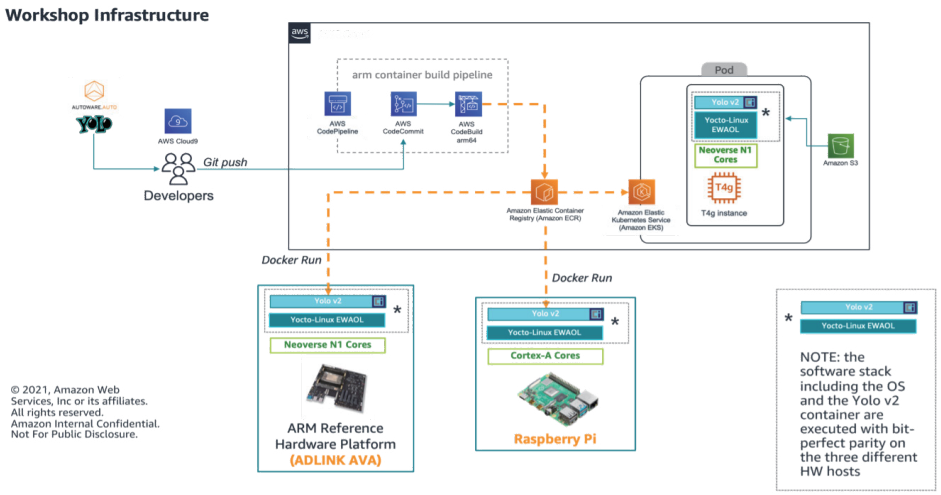
图5：SOAFEE 高级参考架构



Arm DevSummit Workshop —— 基于 SOAFEE 初始实现的技术演示

Arm DevSummit 2021[9] 通过现场演示研讨会展示了 SOAFEE 的首个参考实现。

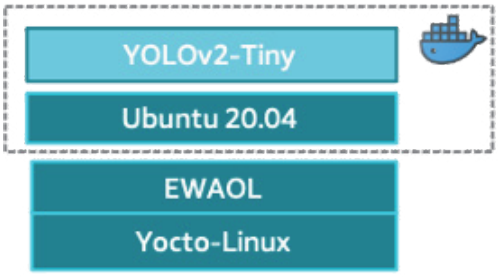
图6：云和车辆边缘对等的汽车开发管道



如上述架构所示，本次研讨会介绍了一种新颖的汽车原生软件开发基础架构，该架构可通过一系列目标计算元素（包括 Amazon EC2 Graviton2 实例、Raspberry Pi 以及 AVA Developer Platform）对等的环境执行相同的容器化工作负载。

在该架构中，我们使用亚马逊云科技服务创建 CI/CD 管道，并借此通过云和嵌入式设备组成的大规模环境构建、容器化、评估、部署了一种名为 YOLO 的感知网络。这个网络可充当汽车应用程序工作负载的替身，借此演示设计范式。本次研讨会所用的 YOLO 版本为 (YOLOv2-Tiny)，运行在 Ubuntu Linux 20.04 上。请注意，这是完整的待测系统（SUT）栈首次包含嵌入式操作系统（Yocto-Linux 发行版）并以原生方式在云中运行，这都要归功于 Arm 的 Edge Workload Abstraction and Orchestration Layer (EWAOL)，SOAFEE 架构的一种参考实现。

图7：待测系统，以全栈方式包含操作系统以及之上的所有组件



操作系统之上的所有 SUT 组件均以指令集对等的方式执行，并为所有目标系统使用了相同的 Aarch64 底层架构。Arm 的 EWAOL 为用户提供了一种基于标准的框架，借此即可使用容器在多种嵌入式平台上部署并编排应用程序。这一系列功能相结合可实现：

- 不仅开发中的软件单元（即 YOLOv2-Tiny 感知模型本身），嵌入式操作系统以上的整个嵌入式软件栈可在云中执行。

- SUT 可从云端无缝移植到嵌入式边缘，不再需要交叉编译或模拟（也不再遇到相关问题，如编译错误或性能下降）。

实际上，这种方法使得开发者可以在云端开始编写并测试嵌入式代码，实现嵌入式开发工作流的左移，并充分利用云平台强大的可扩展性显著扩大测试的覆盖范围（在本例中我们使用 Amazon Batch 并行启动多个 SUT 的执行）。

结论和后续工作

总而言之，软件定义的汽车很明显已经不再是对未来的畅想，而是一个正在发生的关键趋势，整个汽车生态系统的例子已经证明了这一点。未来，软件将成为 OEM 厂商之间竞争的关键性差异化因素，软件开发者则会成为未来汽车系统开发过程中不可或缺的关键角色之一。随着汽车行业逐渐开始进行这个关键转变，所遇到的很多挑战都可以通过现有的汽车云原生技术顺利解决。考虑到这一点，诸如 SOAFEE 这样的倡议开始在汽车行业对云原生技术的采用和加速运用过程中发挥出关键作用。

云原生汽车软件开发方法在技术方面的第一个关键推动因素是，能够在云端开发环境和汽车边缘之间实现对等环境，借此汽车软件开发者即可在云端开发并验证自己的软件，随后将其部署到边缘。借此形成的左移、横向扩展和售后部署模式可有效提升汽车系统开发效率，推动创新的业务模式，从而实现汽车行业的转型。基于 Arm 架构的汽车目标系统和 Amazon Graviton 实例就是这种对等的一个例子，这种方式目前已正式可用，并在 Arm DevSummit 2021 活动中进行了端到端的演示 [9]。

随后，随着我们向着开发新技术和概念不断努力，此时最重要的是能够通过代码来演示这些概念，并为开发者提供平台以进一步培养这个生态系统并以此为基础进行创新。除了与亚马逊云科技联手进行 DevSummit 研讨会演示，Arm 还与 Autoware Foundation（AWF）合作于最近公布将要打造 Open AD Kit[13]，该工具旨在提供完整的端到端开发工具包，借此可在 Amazon Graviton 实例上以 Autoware 作为参考的自动驾驶软件栈进行原生开发，并借助云原生编排技术将软件无缝部署给汽车边缘平台。

汽车行业的转型之旅就此开始。Arm、亚马逊云科技以及 SOAFEE SIG 成员将继续深入合作，帮助汽车行业加速实现云原生。如果您同样对这个远景的实现充满激情，诚邀您加入 SOAFEE SIG。详情请访问 soafee.io/。

参考

[1] Girish Shirasat，用云原生的方法实现软件定义的汽车，2021 年 9 月

[2] Kevin Hoffman，超越 Twelve-Factor 应用，2016 年 4 月，O'Reilly Media, Inc.

[3] Robert N. Charette，软件是如何蚕食汽车的，2021 年 6 月

[4] Robert Day，软件定义的汽车需要能跑得更远的硬件，2021 年 6 月

[5] Constantin Gillies，巨大的闭环：人工智能和机器学习，2021 年 7 月

[6] Alan Burns 与 Robert I. Davis，混合关键系统——评述，2019 年 3 月

[7] Information is Beautiful —— 百万行规模的代码

[8] Martin Fowler 有关开发者有效性的一篇精彩文章

[9] " 云原生汽车软件开发上手 " 研讨会，录像请点击这里观看，循序渐进教程请点击这里查阅

[10] Netflix 迁移至云原生架构

[11] Computer Weekly —— 容器化技术如何帮助大众开发汽车软件

[12] 捷豹路虎的 DevOps 之旅

[13] The Autoware Foundation —— Autoware Foundation 发布快速上手工具包加速云原生自动驾驶技术的开发

[14] Arm 面向嵌入式边缘的可扩展开放架构：Soafee