开放数据中心委员会
Open Data Center Committee

云边计算系统架构、方案和工程实践

[编号 ODCC-2020-04002]

开 放 数 据 中 心 委 员 会
Open Data Center Committee

目录

前　言 I
1 云边协同应用于 V2X 领域的总体思考 1
1.1 V2X 基本概念和技术 1
1.2 V2X 的演进和发展 2
1.3 V2X 与边缘计算 5
2 V2X 总体设计 8
2.1 整体架构 8
2.2 百度移动边缘计算平台 OTE 10
2.2.1 OTE Stack 介绍 10
2.2.2 OTE Stack 架构 12
2.2.3 OTE Stack 商业应用 14
2.2.4 OTE Stack 未来规划 14
2.3 边缘云侧 14
2.3.1 物理层 14
2.3.2 IaaS 层 14
2.3.3 应用层——OTE 组件 15
2.3.4 应用层——BIE 组件 15
2.4 边缘路侧 17
2.4.1 物理层 17
2.4.2 IaaS 层 18
2.4.3 应用层——V2X 感知算法 18
2.4.4 应用层——OTE 组件 18
2.4.5 应用层——BIE 组件 19
2.5 V2X 安全 19
2.5.1 Docker 安全实践 19
3 倡议与愿景 23

前　言

道路如同城市的血管，血管的通畅程度与城市活力息息相关。但在城市规模越来越大、人口数量不断增大、交通拥堵程度加剧的当下，仅靠加快道路建设等方式来构建便捷高效的交通体系，已经变得越来越困难。5G 时代，作为城市和车辆的连接点，道路也成为通信网络、云计算、智能传感器融合创新的交汇点，如何利用新的技术来提升城市智能化水平，增强城市路网与车辆的协同效率和安全性，从而降低城市拥堵、改善出行体验，成为技术改变生活的新机遇和挑战。

车路协同是将道路、车辆以及技术进行有效融合，通过先进的无线通信和互联网技术，实现车与车、车与路、车与人、车与网络的实时数据交互，帮助乘客和车辆选择更好的出行路径，改善道路规划、建设和管理，提升交通效率。

目前百度车路协同技术业界领先，车路协同实际项目落地除感知算法外，还包括 RSU 服务器订制、网络实施、基础 IaaS 管理、摄像头管理与报警、感知算法部署和监控、容灾预案等。作为保证感知算法 24 小时稳定运行的 IaaS 平台，它是车路协同运作的基石。百度边缘计算平台 OTE 为 V2X 系统提供了完整的运行环境，它主要负责集群化交付、应用管理、自动化运维、实时监控、日志回传等。

本白皮书主要介绍百度边缘计算平台 OTE 在打造车路协同基础能力和保障系统稳定高效运行方面所做的一些工程实践，同时探索 5G 时代车路协同的工作模式，为车、路、人、云的整体协作提供云边协同支持。

本白皮书主要内容包括：

介绍了 V2X 发展，调研了 V2X 最新技术，论证了边缘计算对 V2X 贡献

详细阐述了百度 V2X 整体架构、边缘路侧、百度 5G 移动边缘计算平台 OTE、边缘云侧技术架构和解决方案

介绍了 V2X 在工程实践落地中遇到的容器安全问题及优化方案

本报告感谢以下起草单位（排名不分先后）：百度、英特尔、中国信息通信研究院

起草人（排名不分先后）：

陈刚、张贺纯、许振华、刘世堃、张骏、徐佳靓、吴美希、芦帅

1 云边协同应用于 V2X 领域的总体思考

1.1 V2X 基本概念和技术

■ V2X 基本概念

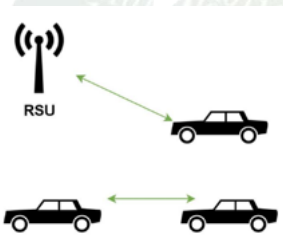
根据 3GPP 描述，V2X 术语是指基于车到车、车到基础设施、车到网络、车到人等实体间的通信。

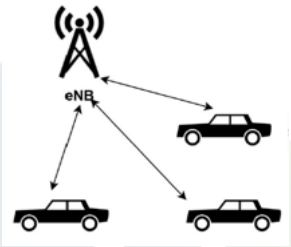
- ✓ 车到车(V2V, Vehicle to Vehicle)：正在靠近的车辆之间的 V2V 应用信息交互，如车辆位置、速度、方向和运动状态等，目的是提供安全预警，从而提高驾驶安全性。
- ✓ 车到基础设施(V2I, Vehicle to Infrastructure)：车辆与通信范围内的路边基础设施（RSU，Road Side Unit）之间的信息交互，如红绿灯信息、交通标志等交通信号信息，从而让车辆能够实时感知到所处交通环境信息。
- ✓ 车到网络（V2N, Vehicle to Network）：车辆与蜂窝基础设施、与中心云、与应用服务器之间的应用信息交互，主要用于提升交通效率和避免拥堵。
- ✓ 车到行人（V2P, Vehicle to Pedestrian）：车辆与行人间应用信息交互，如为车辆及行人提供安全预警，预防交通事故的发生。

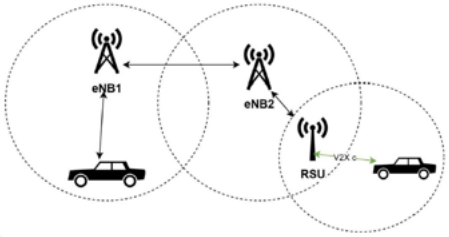
■ V2X 通信技术

支持 V2X 通信的技术技术目前主要有两种，IEEE802.11p 和 C-V2X。除了标准的 V2X 通信技术外，也有一些正在使能的技术，如可见光通信 VLC，毫米波等。VLC 及毫米波方式都存在着一些技术上的局限性，如只能在可见光条件下使用，并且会受到恶劣天气的严重影响等等，更多被作为补充技术。

通常 V2X 通信划分直接通信、蜂窝网络辅助通信和混合通信三类，







如下图所示：

直接通信(802.11p/PC5)　　蜂窝网络辅助通信

混合通信

在直接通信中，车辆直接与相邻车辆通信。在蜂窝网络辅助的通信中，必须通过 RSU 或基站（BS）进行通信。在混合通信中，车辆通信通过使用 802.11p 和蜂窝网络（例如 RSU 和 BS）将直接通信和网络辅助通信相结合来实现彼此之间的通信。

诸如 IEEE 802.11p 和 C-V2X 等最新的车载网络技术无法满足对关键和安全应用需要超可靠通信的下一代自动驾驶汽车的需求。因此，下一代的智能和自动驾驶汽车需要能够处理超可靠情况的增强型车辆服务。随着车辆的发展以及向智能和自动驾驶汽车的发展，V2X 的 QoS 要求变得更加严格。因此，现有的 V2X 技术可能无法满足未来 V2X 通信的性能要求。

1.2 V2X 的演进和发展

车辆是我们日常通行的重要组成部分，它们已经成为必需品，而不是奢侈品。V2X 通信可以在由车、路、人、云构成的整体网络中实时提供安全驾驶、交通效率和路况相关的信息。目前，汽车行业正在经历着重大技术进步和变革，网联化、智能化和自动化成为未来的发展方向。

作为未来智能交通系统 ITS 的开发和部署工作的一部分，V2X 通

信和服务已经引起了从政府机构到汽车制造商等整个产业链极大的兴趣。这主要是因为 V2X 能够带来一系列的优点，如减少与降低交通事故发生率、引入新型商业模式及减少车队运营支出。同时，V2X 通信能够提供各种服务，如车辆自动驾驶、交通流量优化、车内娱乐等一系列的功能及服务。这些应用和服务通常对端到端时延、带宽、算力、存储等方面有着非常严格的要求。

■ V2X 的发展

✓ 通过 2G、3G、4G 等技术提供车载信息服务，如车载导航、车辆诊断和车载娱乐等信息。

✓ 通过 LTE-V 等技术辅助驾驶应用，如实时交通路况信息共享、碰撞预警和避免等

✓ 通过 5G 技术逐步引入自动驾驶应用，如远程驾驶、编队行驶、环境感知和控制决策、高精度地图下载等 V2X 应用

DSRC 时代刚开始时，3G 蜂窝基础设施也开始运营。但是，3G 并不能满足 V2X 通信所需的严格规范。在 4G 时代，车载 LTE（LTE-V）于 2016 年作为 DSRC 的替代技术推出，用于支持智能车路协同。但是，LTE-V 仍然不能支持 V2X 应用程序对超低延迟、超高可靠性和超高带宽的要求，例如关键消息和队列的协调。

之后，学术界和产业界引入了 5G NR 技术来增强 C-V2X 应用。同时，边缘计算被认为是推动 5G 生态系统向前发展的技术。它在网络边缘向附近的车辆或其他设备提供计算服务，从而在算力和存储方面降低了通信延迟。边缘计算通过超低延迟的数据传输提供了一系列服务，例如边缘数据存储、任务处理、实时网络访问和服务质量（QoS）改进。V2X 系统中的 MEC 平台具有各种优势，包括无处不在的网络连接、增加的可伸缩性和降低的网络操作复杂性，并有助于 5G V2X 服务的开发。

■ V2X 标准演进

考虑到未来更高等级的自动驾驶和车路协同需求，3GPP 定义了面向 5G 增强的 V2X 业务需求及 5G-V2X 无线接入标准。

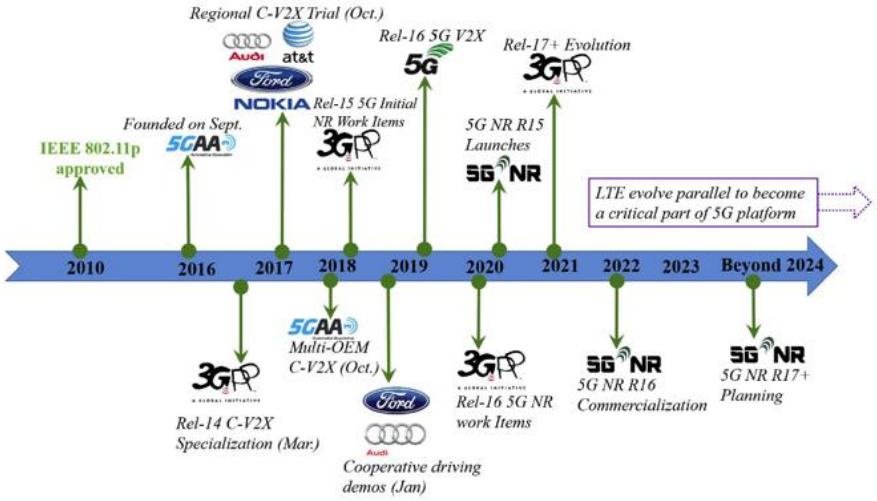


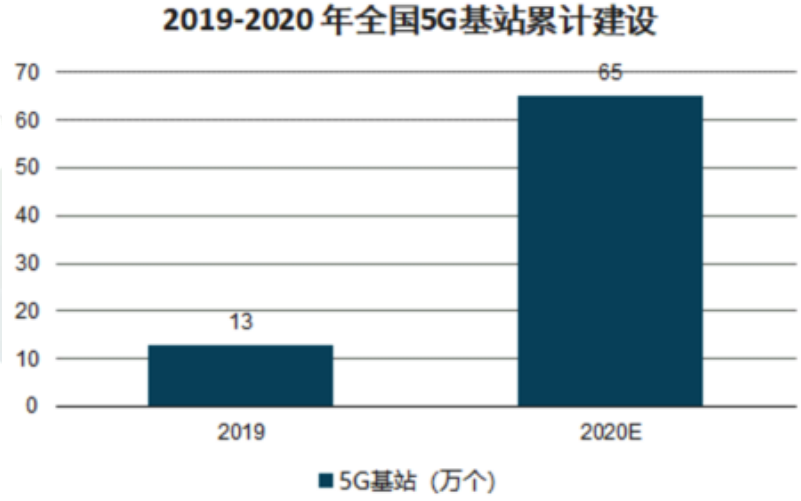
图 1-1 C-V2X 向 5G NR V2X 的演进

1.3 V2X 与边缘计算

■ 边缘计算与 V2X

3GPP R15 版本针对 V2X 的场景需求分析中（TR38.913），时延要求最为严格的自动驾驶和传感器共享场景，对时延的要求最低达到了 3ms。同时，带宽需求最大的传感器共享场景，对带宽的要求最高达到了 1Gbps。此外，全局路况分析场景对服务平台的计算能力也提出了要求，要求能够快速对视频、雷达信号等感知内容进行精准分析和处理。

边缘计算通过在靠近用户及终端的位置部署计算、存储、网络、加速等能力，使得应用不再受限于终端电池容量、散热、算力等限制，从而可以获得计算与性能的较好平衡。同时，因为更靠近部署了计算、存储、网络及加速等能力的边缘，终端及应用对大带宽、低时延、大



连接的需求也得到了较好的满足。

图 1-2 2019-2020 年全国 5G 基站累计建设

边缘计算赋能 V2X 的核心是将 V2X 业务部署在边缘计算平台上，借助网络及空口支持实现了车、路、人、云的整体协同。因此，边缘计算是实现 V2X 产业变革最好的和必须的方式。

■ 边缘计算对 V2X 场景赋能

针对边缘计算的架构特点及 V2X 对边缘计算在时延、带宽、存储等方面的强烈需求，边缘计算对如下 V2X 场景应用有着非常明显的赋能作用：

- ✓ **超视距感知：**单车感知存在不可避免的盲区或漏检，通过在 V2X RSU 附近部署的边缘计算平台可以在路侧实现对感知数据的融合分析，将 RSU 提供的转角、路口侧向等车辆感知盲区状态信息实时分享和发布，有效扩展车辆的感知范围。

- ✓ **自动泊车：**自动驾驶车辆需要停车场端提供停车场地图、车位信息、路径感知信息等数据，以及自动泊车调度与指令数据。边缘计算可有效降低该场景下数据回传与下发的时延。
- ✓ **高精度地图：**通过在边缘计算平台部署和存储高精度地图数据，实现了高精度地图向车辆的实时分发。
- ✓ **远程驾驶和编队驾驶：**通过边缘计算与 5G 能力开放架构支持，可实现对车辆的实时监控和异常接管，从而使远程驾驶和编队驾驶成为可能。

2 V2X 总体设计

V2X 总体设计需考虑到端到端时延要求、带宽、物理部署位置等多个因素，以满足智能车路协同对车端、路侧、边缘站点和中心云侧等各方面的需求。

在设计过程中，将 V2X 智能车路协同整体架构划分为车端、边缘路侧(RSU、感知摄像头、监控摄像头、RSCU)、边缘云侧（边缘节点、核心交换机）、中心云侧等四大部分。

本白皮书重点关注边缘路侧、边缘云侧两部分的核心功能。

2.1 整体架构

V2X 架构从基础设施层面上看分为两部分，其中一部分属于边缘云侧，该部分一般是园区提供的一个或多个中心 IDC，用于提供对边缘路侧设备的管控，以及作为边缘数据的汇聚中心。另一部分为边缘路侧设备，一般由路侧的基本计算单元（RSU、RSCU）和路侧摄像头、以及各种感应采集设备组成。无论是中心还是路侧，其组成部分都为基础设施层——IaaS 层——应用层，另外云中心侧还要提供若干可视化平台，提供用户操作与观察界面。

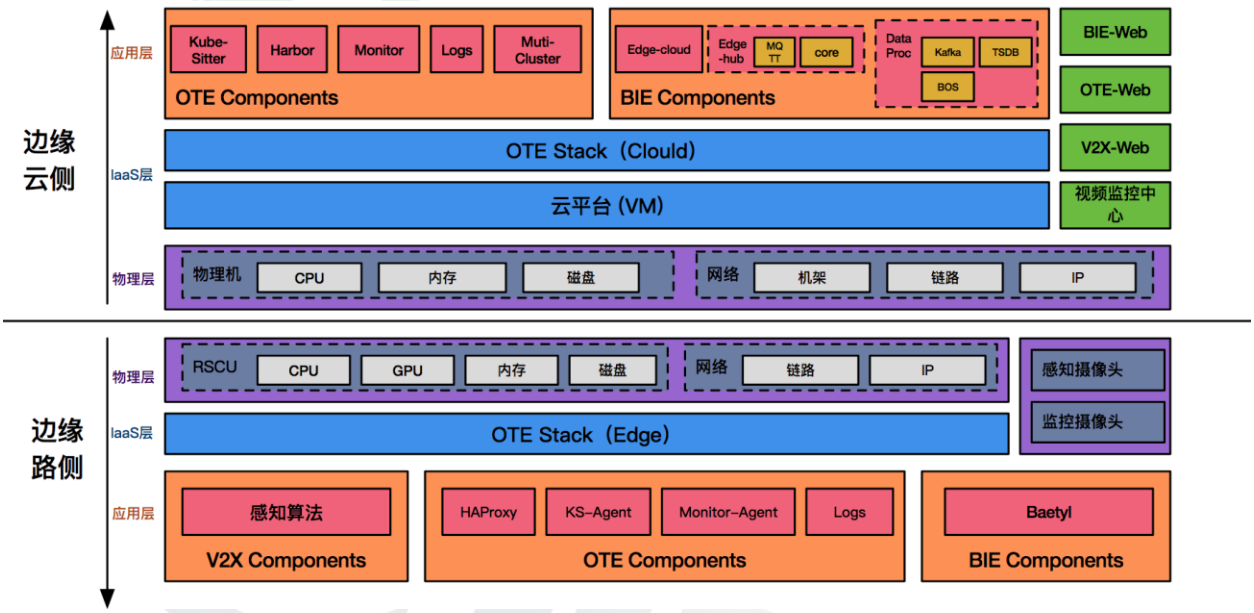

边缘
云侧
应用层
laaS层
物理层
Kube-Sitter
Harbor
Monitor
Logs
Muti-Cluster
OTE Components
Edge-cloud
Edge-hub
MQTT
core
Data Proc
Kafka
TSDB
BOS
BIE Components
BIE-Web
OTE-Web
V2X-Web
视频监控中心
OTE Stack（Clould）
云平台 (VM)
物理机
CPU
内存
磁盘
网络
机架
链路
IP
边缘
路侧
物理层
laaS层
应用层
RSCU
CPU
GPU
内存
磁盘
网络
链路
IP
感知摄像头
监控摄像头
OTE Stack（Edge）
感知算法
V2X Components
HAProxy
KS-Agent
Monitor-Agent
Logs
OTE Components
Baetyl
BIE Components

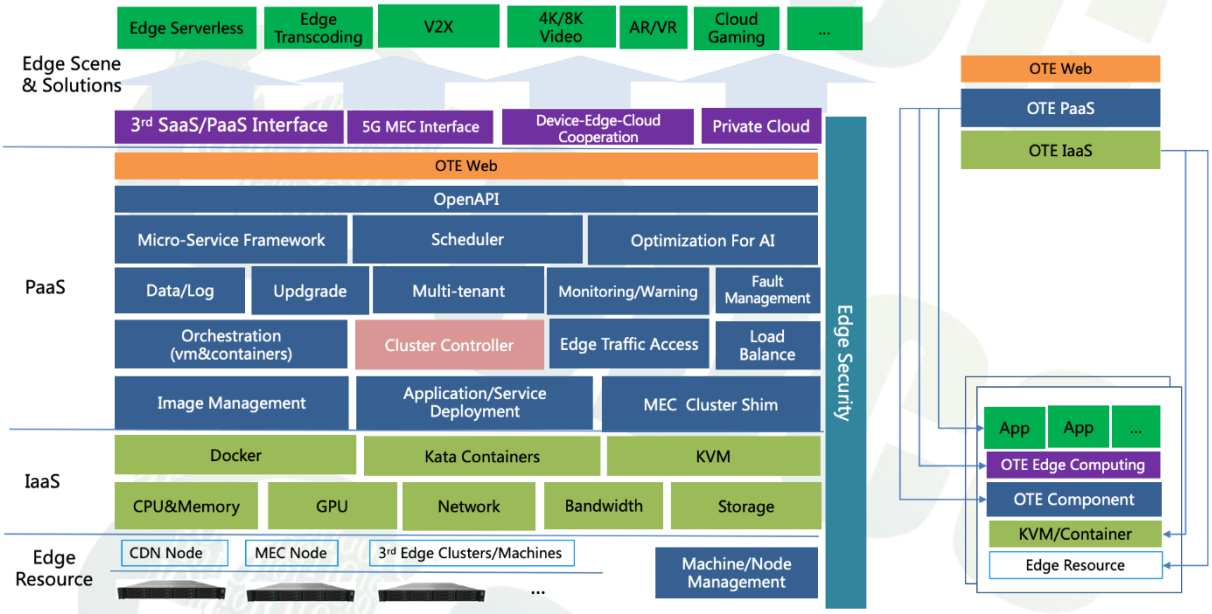
图 2-1 V2X 整体架构

2.2 百度移动边缘计算平台 OTE

2.2.1 OTE Stack 介绍

百度 OTE Stack 边缘计算平台，最初源于 CDN 的边缘云化，2018 年 OTE Stack 正式定位为服务于 5G 移动边缘的边缘计算平台。OTE 集中致力于多运营商、多种边缘资源的统一接入，通过虚拟化和智能调度屏蔽底层异构的特性、提高资源利用率、降低接入门槛。

面向百度内部，OTE 作为边缘基础设施，支撑百度 Device-Edge-Center 算力的统一调度，构建新计算体系结构，为百度 AI 战略提供低延时，高可靠，成本最优的边缘算力支持。面向生态合作伙伴，在边缘提供尽可能全的运行环境，并进行统一管理，无论何时何地何种



网络，计算服务总能及时出现在离终端最近的位置并提供持续可靠的服务。

图 2-2 百度 5G 移动边缘计算平台 OTE 架构

2.2.2 OTE Stack 架构

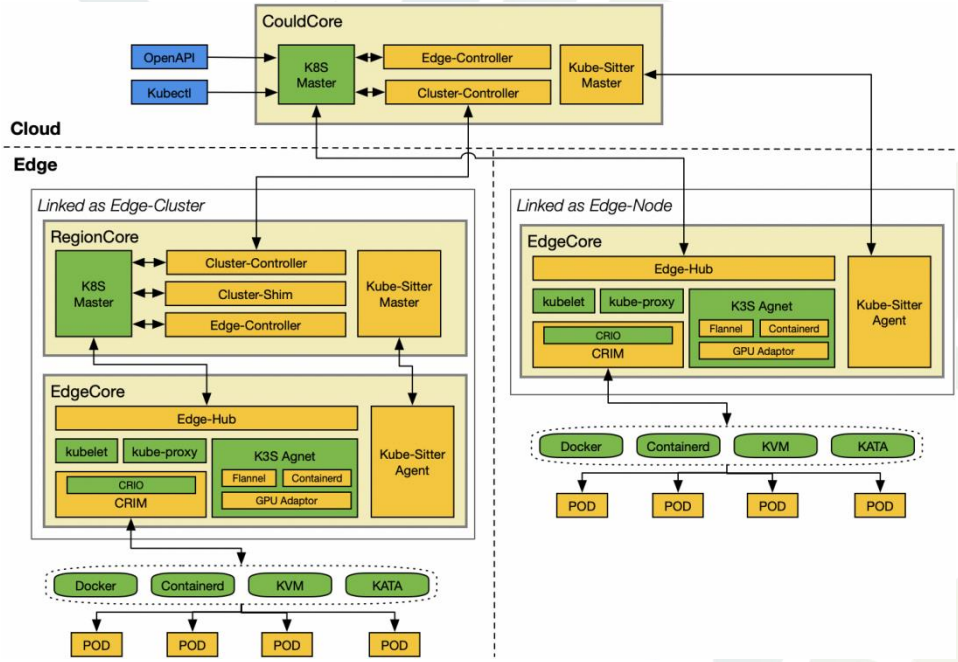


图 2-3 OTE 分层架构

- **纳管海量边缘节点：**集群以树形的方式级联起来，理论上可以做到无限扩展
- **云边协同：**边缘集群和边缘节点皆可被纳管，使得可以在云端一键部署，便能下发至海量的边缘节点。
- **边缘节点自治：**提供了 Kubelet/K3S agent 的缓存和代理，能够使得节点访问 master 能够故障自动切换。
- **边缘集群轻量化：**引入了 K3S 轻量化集群，剔除了不必要的插件，使得 k3s 可以在 512M 的内存空间中正常运行。
- **自动化集成交付：**自研的 Kube-Sitter 组件使得在边缘项目中能够迅速完成 OTE Stack 整体环境的交付。

- **屏蔽底层异构：**通过云原生技术（K8S + Docker）完成底层资源差异的屏蔽。边缘节点上完成 GPU 适配，在 CPU 架构上完成 X86 和 ARM 的适配。

2.2.3 OTE Stack 商业应用

在自动驾驶济车辆网方向，支持了百度自动驾驶团队多个 V2X 项目，包括长沙、沧州、阳泉等等项目的交付，总规模数亿元人民币。

在智慧园区方向，目前正在对接某重能源行业钢铁厂、铝厂的智慧园区视频安防项目，地产行业的多个智慧社区及未来社区项目。

在 CDN Edge 支撑方面，已接入了全量 CDN 数百个节点 Relay 机，总量达数千台机器的资源管理和调度，同时落地百度网盘动态加密等多个产品，完成 100G 动态加密带宽。

2.2.4 OTE Stack 未来规划

展望未来，百度 OTE 在进一步完善云边协同、监控组件和 web 平台功能的同时，将结合内部 AI Edge 使得 AI 能够迅速落地到边缘。同时，将继续扎根 5G MEC、CDN 和私有边缘，致力于提供全场静边缘计算服务。

2.3 边缘云侧

2.3.1 物理层

以长沙 V2X 为例，长沙园区提供了 10 台机器的小型机房作为中心，由于物理机资源有限，不能满足实际业务部署需求。所以使用了某电信设备商的云平台，对这部分物理机进行虚拟化。如果物理机资源足够，则可以不需要额外搭建虚拟机云平台。

2.3.2 IaaS 层

除去 VM 云平台之外，该层主要由百度 OTE 提供容器 IaaS 运行环境，容器环境由 K8S 提供。为了满足后续边缘场景的轻量化需求，我们后续会采用轻量化集群方案替代 K8S，以实现边缘集群

轻量化。

2.3.3 应用层——OTE 组件

- ✓ 多功能交付组件——kube-sitter，由于边缘资源非常有限，不能像传统数据中心 IDC 那样为每样的功能单独开发一套 agent 系统，这会给边缘设备带来沉重的负担和巨大的管理成本。Kube-sitter 旨在把机器设备管理、监控采集、开机环境初始化、软件包部署管理等主要功能集成于一身，精简功能满足轻量化要求。
- ✓ 多集群管理组件——muti-cluster，随着 V2X 技术越来越成熟，V2X 将会全面开花，届时将有越来越多的路侧设备接入，以及越来越多的城市实现 V2X 无人驾驶。每个集群只能管理单个区域内有限数量的路侧设备，那么将会有越来越多的集群使得运维管理难度陡增，我们推出多集群管理方案，目的是通过多层级的分级自治管理，实现一次部署下发，全量集群实现部署的功能。
- ✓ 集群监控系统组件群，由自研数据汇聚端 data-server、promethues、alert-manager 组成，实现了对多维数据的汇集、存储、查询以及自定义告警等功能。
- ✓ harbor，集成了 chart 仓库和 docker 镜像仓库，为每个 V2X 私有项目提供稳定可靠的镜像源。
- ✓ logs 组件是有比较主流的 EFK 方案组成的，服务端主要是 elasticsearch 作为日志存储引擎，提供接口给 web 平台展示。

2.3.4 应用层——BIE 组件

BIE 组件主要由 IoT Stack 组成，它在 V2X 中的作用主要在于对边缘路侧的感知数据进行回传并存储。IoT Stack 架构图如下所示：

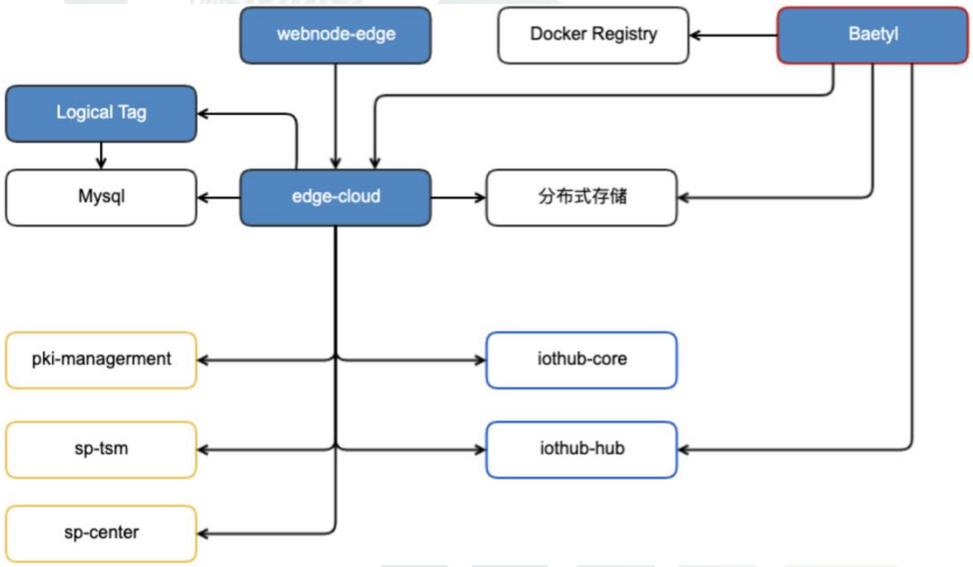


图 2-4 应用层 BIE 组件

目前 V2X 在 RSCU 侧使用 Baetyl 采集感知算法预测结果数据，通过 MQTT 协议并回传到边缘云中心的 Edge-hub，最终存储在 TSDB。除此之外，BIE 还提供 BOS 存储功能，用于存储视频结构化数据，以备上层应用需要。

2.4 边缘路侧

2.4.1 物理层

路侧的物理设备有 RSCU（Road Site Compute Unit）、感知摄像头、监控摄像头组成。RSCU 是经过改良以满足路侧灯杆的电压低、温度高、湿度大等极端条件的小型服务器。三者组成局域网，RSCU 作为网络中心，摄像头把数据先回传至 RSCU，再回传到边缘云中心。感知摄像头采集感知数据，一方面给感知算法模型做预测，做出决策响应给车辆；另一方面将视频原始数据落盘，用来进一步训练模型。下图为从路侧到云中心侧的网络拓扑。

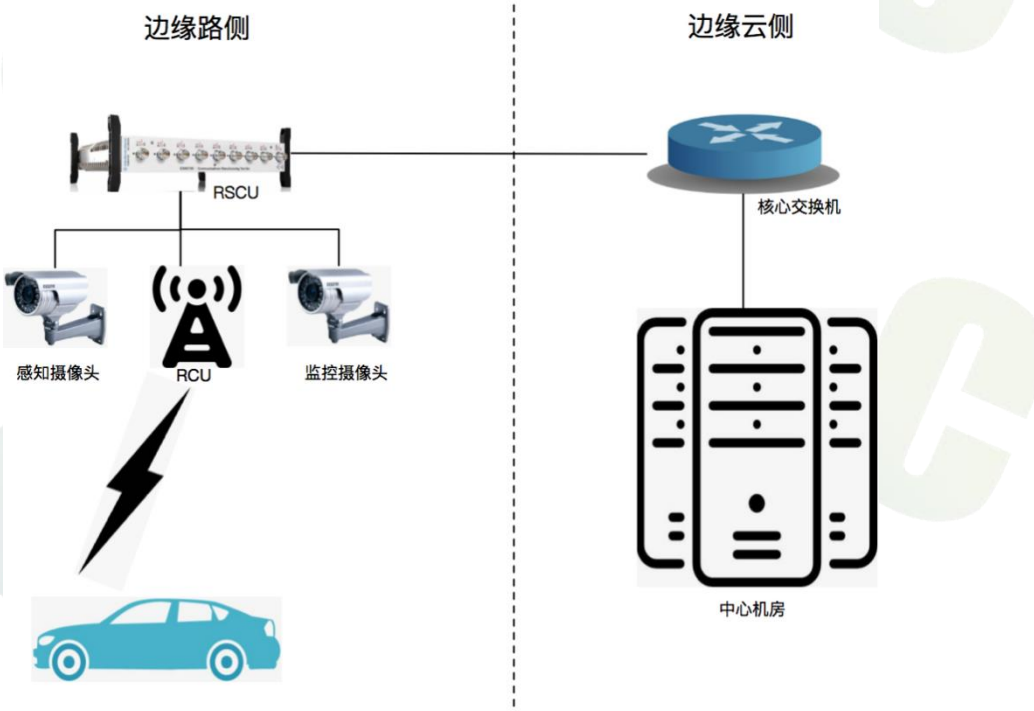


图 2-5 边缘路侧

与边缘云侧网络拓扑

2.4.2 IaaS 层

路侧 RSCU 直接通过 OTE 的 K8S 容器资源提供运行环境。

2.4.3 应用层——V2X 感知算法

V2X 感知算法是一个集实时视频采集与存储、AI 模型预测服务、关键数据回传等功能的路侧核心组件。感知摄像头产生的数据直接输入至感知算法用作预测决策，并直接落盘，落盘的数据会定期取出磁盘用作训练数据。另外感知算法还会采集将 RSCU 本身的状态信息、红绿灯等信息回传到云中心，提供给 V2X 设备管理平台用作交互展示。

2.4.4 应用层——OTE 组件

- ✓ HAProxy 提供了 4 层正向代理，由于园区内部缺少 4 层 LB 组件，即使 master 侧拥有高可用集群的能力，对于单个 RSCU 接入点也只能选多个 master 其中的一个作为入口。为了解决这个问题，OTE 提供了 node 侧的 HAProxy，将客户端对多个 master 入口的探活与动态切换转交给 HAProxy，RSCU 上的 kubelet 直接与本地 proxy 的地址连接即可。
- ✓ Kube-sitter agent 端是整套交付组件的核心执行引擎。Kube-sitter master 负责定义规则、服务和监控，并分发到对应的机器的 agent 上，agent 获取到对应的任务，通过任务类型交由对应的 handler 处理。
- ✓ 监控采集组件，目前有内置到 kubelet 的 cadvisor、node-exporter、gpu-exporter 等采集端，另外还可以通过 kube-sitter 分发自定义监控脚本采集所需的数据。

✓ Logs 组件在 RSCU 侧主要是通过 fluentbit 来采集容器日志。

2.4.5 应用层——BIE 组件

BIE 在 RSCU 只部署 Baetyl 一个组件，Baetyl 是 IoT Stack 中的边缘核心，其部署在边缘远端的物理设备或者服务器上，旨在将云计算能力拓展至用户现场，提供临时离线、低延时的计算服务，包括设备接入、消息路由、消息远程同步、函数计算、设备信息上报、配置下发等功能。感知算法把预测结果投递给 Baetyl，然后由 MQTT 协议回传到云中心。

2.5 V2X 安全

本章节主要关注 V2X 在外场工程实践中遇到的容器加固内容，外场部署所使用的镜像遵循本文档的相关约束。

2.5.1 Docker 安全实践

✓ 选用最小化基础镜像

选用最小化基础镜像，即只包含项目确实需要的系统工具和库的镜像，就能最小化系统的攻击面，确保所用操作系统是安全的。

✓ 设定最小权限的 USER

如果 Dockerfile 中没有指 user ，Docker 默认将会以超级用户 root 身份运行容器，为了尽量降低安全威胁，创建专门的用户和用户组，在 Dockerfile 中使用 user 指定用户，确保以最小权限的用户身份运行容器应用。

例如：RUN groupadd -r redis && useradd -r -g redis redis

✓ 签名和校验镜像，防范中间人攻击

必须保证拉取的容器镜像确实是发布者发布的镜像，没有被任何人篡改。发生镜像篡改，有可能是因为 Docker 客户端和镜像中心之间的中间人攻击，或者是发布者的身份被人盗用并在镜像中心发布了恶意镜像。

✓ 找出、修正和监控镜像漏洞

防范安全软件漏洞的一种方法是使用像 Snyk 这样的工具，持续扫描和监控 Docker 镜像各层可能存在的漏洞。

✓ 不要在容器镜像中包含机密信息

如果 Dockerfile 中包含复制机密信息的命令，构建镜像时，这行命令对应的中间容器会被缓存，导致机密数据也被缓存，有可能造成机密信息泄漏。因此，像令牌和密钥这样的机密信息必须保存在 Dockerfile 之外。

✓ 设定镜像的标签，保证镜像的不可更改性

理由如下：

1) 可重现。一般来说，希望容器再每次运行时都是一样的。如果使用了“latest”，那么哪个版本的镜像会被下载并执行将不受我们的控制。

2) 可追踪。当在诊断问题时（尤其是安全问题），希望找到执行的代码是从何而来。如果基础镜像使用了“latest” tag，想要追踪到具体的版本和源代码是非常困难的。

具体做法：

1) 优先选用最详细的镜像标签，例如，镜像有 8、8.0.1 和 8.0.1-alpine 等标签，选择最后这个，因为它提供了最详

细的信息。Adrian 推荐格式是使用 MAJOR.MINOR 来指定版本号。

2）使用比签名更具体的 SHA256 引用指明要使用的镜像，这能保证每次拉取都是相同内容的镜像。

✓ 使用 COPY，不要使用 ADD

1）COPY - 将本地文件或者目录（递归）复制到容器镜像中的目标目录，复制来源和目标都必须明确指定。给定明晰的源和目标文件或目录，递归地复制本地文件。要使用 COPY 命令，必须声明位置。

2）ADD - 以递归方式复制本地文件，在目标目录不存在时隐式创建目标目录。ADD 也可以将远程 URL 指定的文件下载到目标目录。

为什么这么做：

1）使用 ADD 从远程 URL 下载文件，存在中间人攻击的风险，文件内容有可能因此被篡改。

2）如果复制的是本地压缩文件，ADD 自动将它解压缩到目标目录，这有可能触发 zip 炸弹或者 zip 任意文件覆盖漏洞。

3）使用 COPY 复制文件或目录，会创建一个缓存的中间镜像层，优化镜像构建的速度。

✓ 使用 LABEL 指定镜像元数据

镜像元数据有助于用户更好地理解和使用该镜像。最常见的元数据是 maintainer，它说明了镜像维护者的电邮地址和名字。

✓ 使用多阶段构建小而安全的镜像

使用 Dockerfile 构建应用容器镜像时，会生成很多只是构建时需要的镜像层，包括编译时所需的开发工具和库，运行单元测试所需的依赖、临时文件、机密信息等等。保留这些镜像层，不仅会增加镜像的大小，影响镜像下载速度，而且会因为安装更多软件包而面临更大的攻击危险。

最佳实践：

1）第一个镜像——非常大的镜像，包含了构建应用和运行测试所需的所有依赖。

2）第二个镜像——非常小的镜像，只包含运行应用所需的极少数依赖。

✓ 使用静态分析工具

使用静态分析工具，能够避免常见的错误，建立工程师自动遵循的最佳实践指南。

可使用 hadolint 分析 Dockerfile 并列出不符合最佳实践规则的地方

3 倡议与愿景

融合了 5G、AI、边缘计算与云计算等关键技术的云边协同在未来必将深入的影响车联网、互联网、物联网、消费、制造、零售等各行各业的发展模式。ODCC 作为开放生态的倡导者，我们将促进边缘计算、5G、AI 和 V2X 等技术的深度融合，推动边缘计算与智能车路协同的技术发展及产业合作。



WWW.ODCC.ORG.CN

本页为规范最后一页