第 3 章

边 缘 计 算

Mohammad Hossein Zoualfaghari[1] *,Simon Beddus*[1] *,and Salman Taherizadeh*[2]

1 British Telecommunications plc , Ipswich , UK

2 Jožef Stefan Institute , Ljubljana , Slovenia

3.1 引言

如今，Amazon、Google 和 Azure 等超大规模云计算服务商为物联网数据的存储、处理和分析提供了高性价比的解决方案。云服务的经济性依赖于数量有限且远离物联网终端设备的大型数据中心。这种云服务运作模式适用于网络数据丢包、延迟和失真不敏感的云计算应用，例如，网页浏览和电子邮件等。

边缘计算模型为需要预测网络可靠性、安全性和低数据处理延迟的应用提供了解决方案，实现了靠近传感器、传动装置、物联网设备和用户等资源的数据处理。这种托管模式可以缩短传输距离、降低网络延迟，并且在许多情况下可以降低解决方案的复杂性，从而为客户和终端用户带来更好的服务。例如，边缘视频分析处理可以减少视频数据传输到云端的需求，从而降低网络负载。

由于边缘资源通常位于私有网络中，因此，在考虑数据隐私控制和重要任务应用的可靠性等因素时，可以选择边缘计算模式。另外，边缘计算的本地处理可以显著减少广域网流量，尤其在当广域网连接成本高昂或覆盖稀疏时，这可能成为主要因素。

按照惯例，边缘计算可以托管工业、零售和物联网应用的关键业务。随着网络安全功能虚拟化的出现，边缘计算可用于托管虚拟网络功能（Virtual Network Functions，VNFs）和安全功能。此外，边缘计算的独特位置还支持物联网安全性和虚拟网络功能。该场景下，边缘设备或终端可视为小型私有云。

实际上，边缘计算设备既可以紧邻用户，也可以位于接入网附近的通信服务提供商（Communication Service Provider，CSP）的网络边缘。其中，位于消费者附近

的设备通常专用于特定消费者，而在通信服务提供商范围内的基础设施则由许多用户共享。

通常，边缘计算设备基于工业化功率控制和刀片系统的 X86 架构，而基于精简指令集（Advanced RISC Machine，ARM）的设备性能较低，如流行的 Raspberry Pi，BeagleBoard 和 pcDuino3 Nano 等。

3.2 边缘计算基础

如图 3-1 所示，边缘计算的部署通常包括 4 部分：

- **IoT 设备**。包括一些简单的网络设备，例如，靠近数据源或控制接口的传感器和传动装置。物联网设备中大量的传感器、传动装置和物体，可以通过多种接口连接到用户边缘设备。例如，边缘计算应用场景中的 3G、4G、5G、Wi-Fi、PCIe、USB 或以太网。
- **用户边缘设备**。这些设备从 IoT 设备接收数据，并向其发送指令。用户边缘设备提供有限的本地存储、处理和网络功能，并可以安装在用户或通信服务提供商的范围内。每个用户边缘设备可在其传输范围内为传感器和传动装置提供无线接入服务。在网络边缘，用户边缘设备可以提供传感器数据的获取、收集、过滤、规范化服务，以及传感器和传动装置的指挥或控制功能。
- **移动边缘计算（Mobile Edge Compute，MEC）**。这些服务器能够降低传输成本，并在计算卸载服务中提供快速的交互响应。相比之下，部署在骨干网的传统云服务资源具有海量的计算能力，而 MEC 服务器则资源受限。因此，MEC 服务器专注于数据聚合、压缩和转换工作。
- **集中式 IoT 云平台**。可以为 IoT 场景提供强大的集中存储和处理能力，包括数据互操作和数据统一访问等重要能力，相关细节将在第 4 章中进行讨论，同时还支持 IoT 设备的远程管理功能。值得注意的是，集中式云计算仍然是边缘计算模型的重要组成部分。无论私有云或公有云基础设施，以及 MEC 服务器都是相互补充、互惠互利、相互依存的服务整体。一些功能适合在云端执行，而另一些功能更适合在边缘端运行。

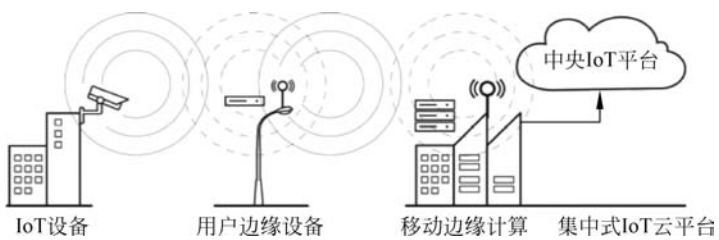

图 3-1 边缘计算的生态拓扑

3.2.1 边缘计算策略

由图 3-1 可知,边缘计算服务可以部署在用户域或通信服务提供商域中。因此,可以抽象出两种边缘计算的主要策略。

1. 用户域边缘计算(Customer Premises Edge Compute,C-PEC)策略

与用户域计算设备或用户资源附近的传感器、局域网(Local Area Network,LAN)和物联网终端相关。这种方式的特点是时延短(小于 10ms),单租户应用程序可在专用设备上执行,且计算负载适中。这种方式的好处是数据可以保留在本地,因此用户可以更好地控制端到端服务的安全性。与现有的云计算范例相比,C-PEC 策略是改进程度最高的物联网使能计算模型。该策略可以在更靠近数据生成端的传感器附近分析延迟敏感型数据,并在某些情况下减少网络通信流量。尽管这种边缘策略主要针对用户域应用,但可以应用于智能车辆或智能手机。例如,传感器附近的智能手机可以充当本地物联网数据的处理器。

2. 通信服务提供商域边缘计算(Communication Service Provider Premises Edge Compute,CSP-PEC)策略

与 CSP 域中的 MEC 服务器等计算资源相关,并且与 5G 场景下的 eNodeB 和无线接入网(Radio Access Network,RAN)有关。这种策略可视为低延迟云服务,使延迟敏感应用程序在传感器附近执行。这种方式的特点是低延迟(小于 20ms),终端设备可为多个租户运行应用程序,以及计算负载高。MEC 型系统的优势是具有大规模的计算设备,因此可以根据用户需求扩展云服务规模。相比于用户边缘域策略,部署在 CSP 域的资源需要更多的处理、存储和通信能力。

在某些情况下,上述两种策略是互补的。可以利用 C-PEC 解决方案对数据隐私和应用程序的完整性进行更高层次的控制。同时,在某些情况下,尤其是在工作负载变化时,大规模的云服务更为重要。此外,MEC 在汽车领域拥有大量的应用场景,汽车间可以通过 5G RAN 通信,并基于 MEC 应用程序进行动作协同。可以预期,未来这些方法将会共存,而且,MEC 将成为 C-PEC 使能方案中解决计算迁移问题的重要基础。

有时,由于工作量的急剧增加,计算能力有限的用户边缘设备会出现过载状况,需要将计算负载从用户边缘设备迁移到 MEC 服务器。因此,如图 3-2 所示,在某些情况下,计算负载可能会在连续的计算层之间卸载或加载。此外,边缘计算场景的高度动态环境可能涉及 IoT 设备在不同地理位置间的变换。在这种情况下,为降低边缘计算应用程序的延迟响应时间,需要将计算负载从一个用户边缘设备转移到另一个更靠近 IoT 设备的位置。因此,计算负载可以提供从一个节点到另一个节点的服务迁移。例如,在用户边缘设备或 MEC 服务器间的服务迁移。

与 MEC 服务器相比,用户边缘设备的存储、网络和计算能力有限。因此,用户边缘设备可以当作原始数据的网关,例如,在运行时对本地数据流进行过滤、编

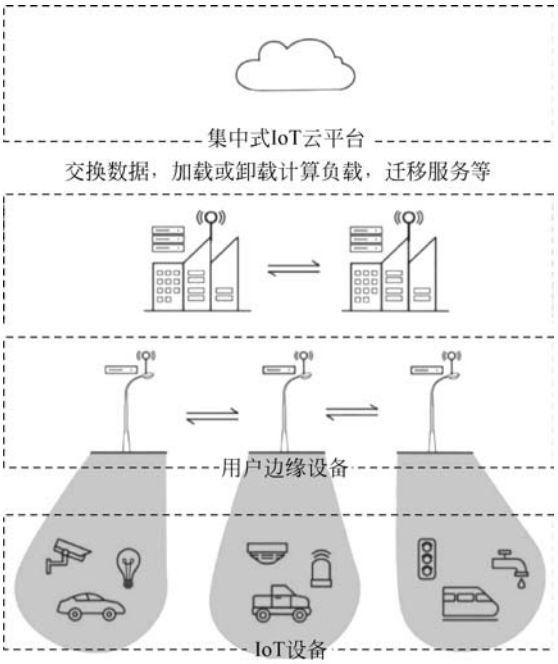

图 3-2　边缘计算生态策略

码和加密。另外，MEC 服务器可以提供诸如数据压缩、聚合和转换之类的服务。此外，部署在云端的集中式物联网平台，能够提供承受大规模边缘计算的无限工作负载能力。

3.2.2　网络连接

边缘计算设备通常支持 WAN 和 LAN 等多种网络连接。

在 LAN 连接方面，边缘设备充当连接 IoT 设备的代理（集线器），通常支持 Wi-Fi、以太网、蓝牙、ZigBee 和工业系统的控制局域网络总线技术（Controller Area Network，CAN）。在传感器物理连接方面，可以通过设备本身（例如 Raspberry PI）或支持 PCIE 的 GPIO 扩展卡提供通用输入/输出功能（General-purpose Input/Output，GPIO）。GPIO 连接器还允许将其他设备（如传感器或灯泡（Light-emitting Diode，LED））连接到控制板上。

在集中式 IoT 云平台连接方面，可以采用 WAN 连接的方式，并通过 3～5G 移动蜂窝网，xDSL 和 LORA 等技术实现。该技术路线中，目前正利用网络功能虚拟化（Network Functions Virtualization，NFV）和软件定义网络（Software-defined Networking，SDN）技术支持边缘节点（无论是 MEC 服务器还是用户边缘设备）与集中式 IoT 云平台间的数据迁移。这两种互补的网络技术是网络构建、设计和运维的最新方法。NFV 和 SDN 技术可以显著增强网络的管理和动态性。例如，当网络质量差时，可以动态更改边缘节点和集中式 IoT 云平台间的数据路径。

3.3　边缘计算构架

3.3.1　设备概述

C-PEC 边缘计算设备采用 X86 或 ARM 处理器架构，通常配有坚固外壳以满足部署在复杂环境中的需求。较小的双核和四核处理器足以满足室内环境下的 C-PEC 部署要求。但需兼顾托管 VNF 的边缘设备可能会使用八核或更多核心设备。此外，除了专用神经网络处理器和人工智能（Artificial Intelligence，AI）芯片，新一代图形处理器单元（Graphic Processor Units，GPUs）可以使边缘设备能够执行更复杂的实时 AI 处理和分析功能。

边缘设备主要使用 Centos 等 Linux 操作系统或 Wind River 等公司提供的安全加固 Linux 变种操作系统。后者通过修改操作系统内核来防止对设备的恶意攻击。为了处理和分析数据，边缘计算资源需要使用轻量级虚拟化技术，例如，便于服务开发、部署、实例化、终止和迁移的容器技术，这也是选择容器化技术（Container）运行 IoT 应用程序的主要原因。容器化对于边缘计算模型非常重要，因为它比传统隔离机制（如基于 hypervisor 的虚拟化）使用资源显著减少[①]。与虚拟机（Virtual Machine，VM）相比，容器具有更高的非侵入性和更少的虚拟化开销。与基于 VM 的虚拟化技术不同，容器不需要为每个容器实例启动操作系统（OS）。容器化管理技术示例包括 CoreOS、Kubernetes、OpenShift Origin 和 Docker Swarm。

MEC 设备更像基于云计算的同类产品，可以将多用户间共享的可用计算资源部署在具有多核处理器的刀片服务器中。这样，MEC 设备可以为用户提供包括多个虚拟机管理程序[②]在内的多种计算服务，并支持多种应用程序。因此，MEC 资源需要诸如 OpenStack 等更高级的基础设施管理工具，尤其因为 MEC 基础设施的计算环境是高度动态的，工作负载会随时间不断变化。例如，资源会在运行时更改状态，并且 IoT 设备有时会频繁变得可用/不可用或更改地理位置。

3.3.2　边缘应用模块

如前所述，应用通常使用容器进行发布，例如，Web 服务、IoT 和动态分析音/视频的特定用户服务。一方面，某些应用可以是专用的，例如，从本地传感器收集温度数据，并进行转换后发送到特定云服务提供商。另一方面，某些应用可以通过执行单个任务实现多种用途，如图 3-3 所示。信息代理应用程序可提供跨应用

---

① 与基于虚拟机的应用相比，使用容器技术后的计算量、内存和存储资源占用情况可降低为（10～20）分之一。

② 支持 VMware、KVM 和 XEN 虚拟机。

程序的互操作，并按需将信息存储并转发给多个云服务提供商。后续的应用样式将在 3.3.3 节讲解。

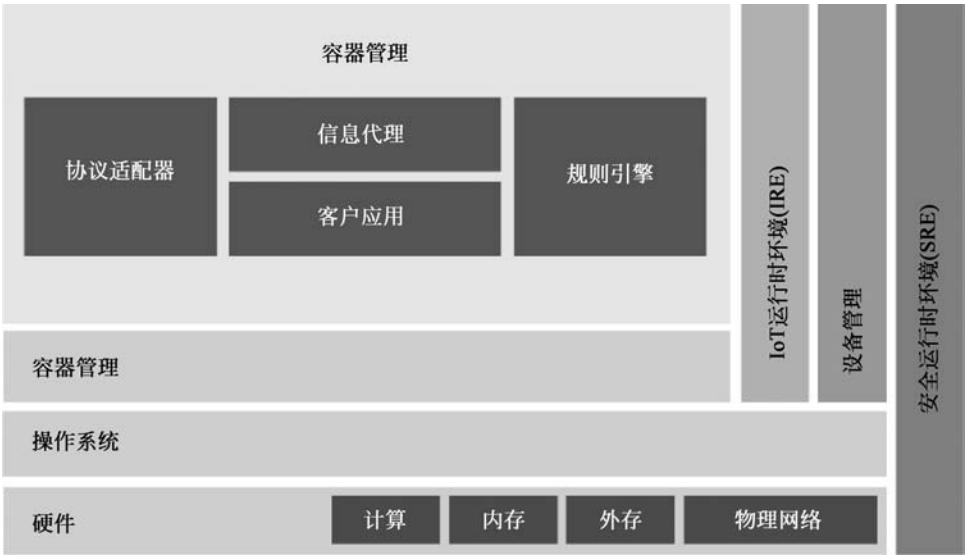

图 3-3 IoT 边缘架构

3.3.3 IoT 运行时环境

如图 3-3 所示，物联网运行时环境（the Internet of Things Runtime Environment，IRE）在容器管理层之上提供了附加功能，实现了多个 IoT 应用和外部端点之间信息流的远程管理。IRE 允许特定应用程序的链接，从而支持数据转换、上下文分析，并向其他系统的信息传递。

3.3.4 设备管理

设备管理功能包括初始设备配置和持续管理。设备管理的目标期望是仅使用一套操作系统和一个设备管理代理即可将设备运送到用户站点。首次启动时，设备将接入到 WAN 网络，并下载设备初始状态所需的软件和配置信息。设备管理组件可以远程获取设备、网络、应用程序状态、吞吐量和运行状况等信息，并可以远程启动、关闭和重启设备。

3.3.5 安全运行时环境

从定义角度讲，边缘计算设备必须支持大量物联网和虚拟网络功能应用。为此，边缘计算设备提供了丰富的计算、存储和多样的网络服务。然而，在边缘计算环境中的上述服务容易受到各类本地和远程恶意攻击。同样地，与从物理角度保护数据中心的计算资源不同，边缘计算设备位于更易受到物理攻击的暴露位置，例

如，USB、无线和固定网络连接的攻击等。

为保护边缘计算设备，可以建立安全运行时环境（Secure Runtime Environment，SRE）。该组件本质上是一系列缩小被攻击空间的工具集合，包括身份访问管理（Identity Access Management，IAM）、安全启动、设备证明、基于硬件的可信平台模块（Trusted Platform Modules，TPMs）和可信任执行环境（Trusted Execution Environments，TEEs）等措施。例如，SRE 可以保证：

- 仅运行可信容器和 VM，例如，来自可信资源池。
- 只能从设备启动软件和配置升级，禁止从设备外部启动。
- 只有有限的可信任身份才能对计算、网络和存储资源进行访问。
- 保护关键配置数据（例如，设备证书和衍生证书）免受篡改。
- 安全启动机制与软件和硬件认证结合使用，以防止启动被篡改/更改的系统。

3.4 边缘计算方案实践

许多读者希望通过边缘计算解决方案实践，并建立原型系统来进一步拓展其知识。本节旨在满足学生或实验人员的实验需求，并聚焦于商业解决方案等成熟产品。

3.4.1 新手配置

对于学生和实验人员，建议使用 Raspberry PI 系列设备作为简单 C-PEC 解决方案的基础。Raspberry PI 3（RPI 3）具有支持物联网应用程序开发阶段所需的计算能力，支持连接传感器和传动装置的 GPIO 引脚，支持具备许多强大开发应用的基本 Raspbian Linux 操作系统。RPI 3 具有 1GB 内存和 ARM 架构的四核 CPU。在短时间内，新手即可完成简单的传感器应用程序，并在本地和云端发布数据。

3.4.2 开发工具

当开发直接在计算设备操作系统上运行的边缘计算应用程序时，有许多可供选择的开发工具。鉴于 C-PEC 的有限资源，以下工具或编程语言可供选择。

Python 是 Linux 平台上常见的开源语言解释器，可以很好地访问网络、输入/输出（Input/Output，I/O）设备以及丰富的轻量内置数据格式化库。Google、NASA、Yahoo 和 CERN 等许多大公司都在使用 Python，并广泛用于边缘计算应用程序开发，例如，科学计算、信息安全、嵌入式应用程序、AI 算法和 Web 开发。同时，Python 在教育领域的垄断地位也确保了稳定的开发人员队伍。

NodeJS 源自将 JavaScript 语言从 Web 浏览器端迁移到服务器端的理念。与 Java 之类的语言相比，这种开源代码运行时的环境具有占用空间小的优势。应当

注意，由 Java 应用程序组成的容器实例都需要一些包，而且，Java 虚拟机（Java Virtual Machine，JVM）也会占用一定内存。

基于 NodeJS 的 Node-RED 是一种基于流的开发工具，专门用于编写 IoT 应用程序。它提供了易于使用的拖曳式开发环境，可以最大限度地减少 JavaScript 的编程工作。如图 3-4 所示，Node-RED 预先配置了消息队列遥测传输（Message Queuing Telemetry Transport，MQTT）协议和 GPIO 引脚（对于 RPI 3）的节点（或适配器）。MQTT 是一种轻量级的 IoT 信息发布/订阅传输协议，尤其适用于需要降低代码占用空间或着重考虑网络带宽的远程连接场景。

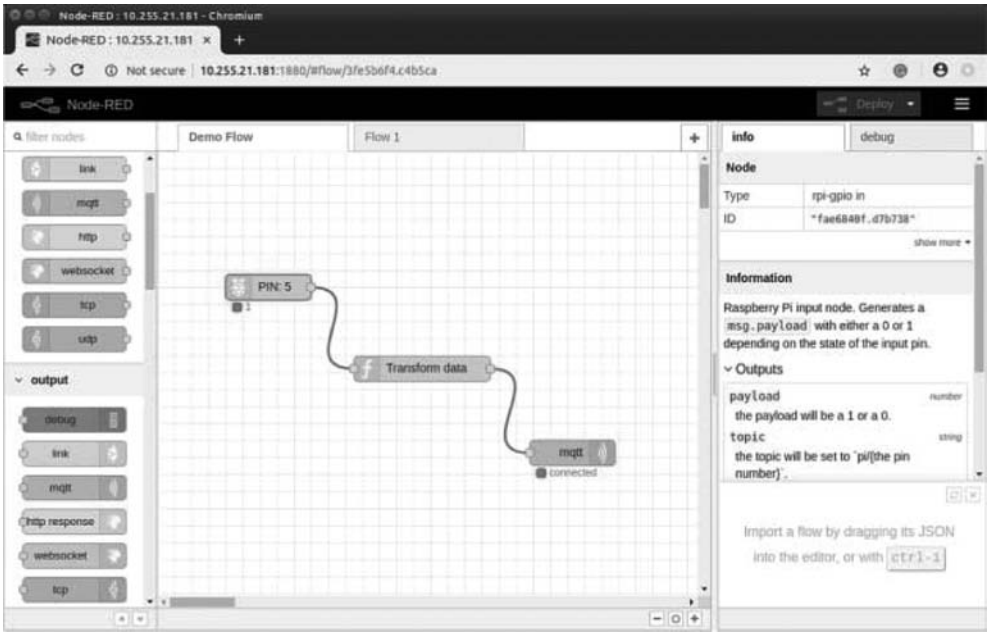

图 3-4　具有工作流示例的 Node-RED 界面

3.4.3　边缘计算构架

边缘计算框架（Edge Compute Framework）代表了针对 C-PEC 风格设备的下一波软件基础架构浪潮，尤其是针对物联网解决方案。这些现代框架的目的是为边缘计算设备应用程序的开发、操作和管理提供高度标准化的规范。进而提高软件的复用性、创新性、计算资源的利用率。一般来说，边缘计算构架支持的功能包含以下模块。

- **协议适配器**，协议特定的模块，可将传入的传感器数据或发出的传动装置命令转换为通用格式。例如，在进行相应配置后，Modbus 协议适配器可以用于读取或设置电机的每分钟转数（Revolutions per Minute，RPM）或转向。

- **信息代理**，一种板载数据存储模块，允许存储来自传感器、云端或其他模块最近接收到的数据。
- **规则引擎**，该模块用于根据预定义规则将来自其他模块的传入数据进行路由转发。例如，每小时将电机的 RPM 数据传送到云端，或者当转速超过 800rpm 时，立即将电动机 RPM 数据转发到云端。
- **专用用例**，开发人员制定的用于执行专业功能的模块，例如，视频分析模块通过处理传入的 MP4 或实时传输协议（Real-Time Protocol，RTP）视频流对道路上的车辆进行计数。
- **管理和安全性**，该模块负责 IoT 设备的系统注册、管理和配置。如图 3-5 所示，该模块通常还涉及身份管理、访问控制和软件堆栈证明等安全功能。

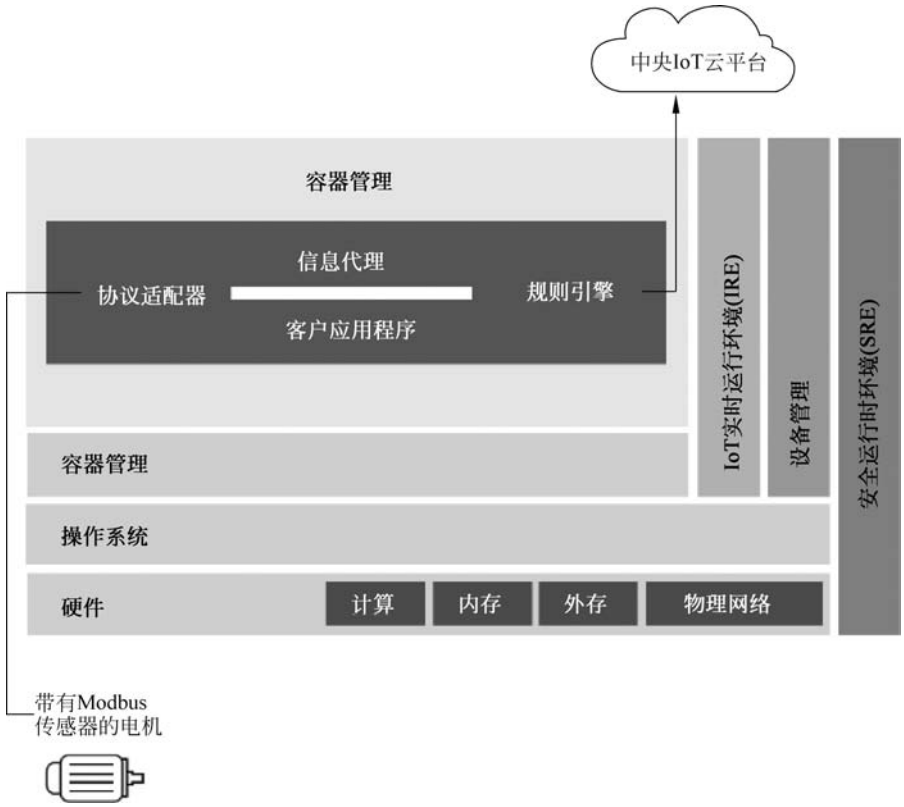

图 3-5　边缘计算架构

像 Azure 和 Amazon Web Services 等供应商提供了大量边缘计算框架。Azure IoT Edge 框架指定了一个边缘代理和一个管理代理，可以轻松映射到图 3-5 中的通用模块。AWS Greengrass 提供了一种不同的方法，使用其无服务器的 Lambda 计算框架来允许开发人员构建有效的计算解决方案，并且与 Azure 一样，它具有一个管理代理。开源的 EdgeX Foundry 项目定义了一个类似的 IoT 框架，

它独立于云服务提供商，具备板载存储、规则引擎、协议适配器，以及向云和其他模块信息分发的功能。

3.5 零接触设备上线

如今，有超过 200 亿个联网设备，并且设备的数量不断增长。例如，据报道，制造业市场中的物联网规模以每年 29%的速度增长。当这些边缘设备交付到用户端时，需要先由专家安装和手动配置，然后才能连接到网络。之后，还涉及常规的硬件和软件维护流程——不幸的是，所有维护过程都需要人工干预。

大量 IoT 边缘设备的部署使设备管理成为 IoT 平台提供商的重要问题。物联网平台提供商越来越希望改变当前的设备管理流程，这种劳动密集型且耗时的工作需要为每台设备或用户单独提供在线解决方案，并向在线自动化和远程管理转型。这将大大减少部署时间，减少人力资源需求，并降低安装所需的专业知识水平。

自动化在线设备面临的主要问题是远程建立边缘设备和 IoT 平台之间的初始信任，这就是安全设备上线的证明过程。如今，出现了可以实现零接触并确保物联网终端安全上线的前沿技术。Intel 公司的安全设备上线（Secure Device Onboarding，SDO）技术、思科公司的 aSSURE 和微软公司的 Azure Sphere 都是物联网安全领域的旗舰技术。英国电信（British Telecommunications，BT）集团的合作实验室正在研发一种新型边缘技术，可以远程、安全地建立信任关系，并完全自动化地证明 IoT 终端安全性。该解决方案中，当设备首次连接到网络时，将会自动、安全地以合法且完全可信的身份注册到集中式 IoT 云平台中，并按照如图 3-6 所示的流程立即在无线网络中实现安全的自我监控和远程维护（Over the Air，OTA），即零接触设备上线（Zero-touch Device Onboarding，ZDO）过程。

如图 3-6 所示，作为应用使能平台的一部分，设备引入了证明服务器和引导服务器两个新组件。在证明服务器与第三方交互的解决方案中，使用不同技术建立起自各种供应商的 IoT 管理服务器与远程终端间的信任关系。引导服务器会根据其类型、资源、使用目的和其他功能自动为每个设备准备并封装必要的协议、固件、应用程序和设备管理代理。然后，经由服务器建立的安全通道，应用程序和配置信息会自动发送到原始设备上。

ZDO 支持许多新引入和常用的 IoT 标准，例如，开源移动联盟标准（Open Mobile Alliance，OMA）、基于约束应用协议（Constrained Application Protocol，CoAP）的轻量级机器对机器标准（Lightweight Machine to Machine，LWM2M）、用户数据包协议（User Datagram Protocol，UDP），具备安全防护的数据传输层安全协议（Datagram Transport Layer Security，DTLS）。

基于此，设备在首次开机时会自动认证并接收所有必要的应用程序和配置信

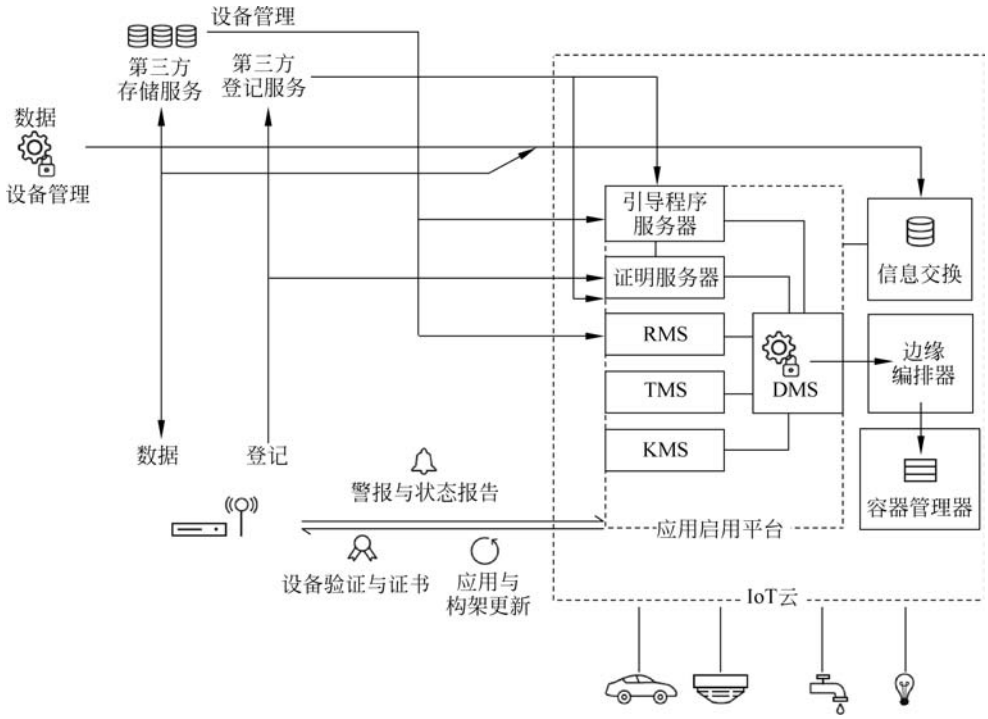

图 3-6　安全与零接触上线流程

息，并通过一组专用的协议和证书引导到相应的 IoT 终端。换句话说，原始设备开机并自动进行安全配置，并按预期开始工作，同时向相应的 IoT 平台报告。ZDO 已在各种概念证明原型中应用，涵盖了 IoT 设备的整个生命周期。这包括：在制造商工厂中的构建过程、运送到配送中心、设备的购买和采购（物理和数字方式）、所有权链、零售、到达用户端、首次启用时自动配置、认证、引导程序（协议、固件和应用程序）、自动化设备管理和远程维护。为此，英特尔公司的 SDO 技术既可以运行在具有英特尔增强隐私 ID（Enhanced Privacy ID，EPID）芯片的 IoT 终端物理层（例如，芯片级安全），又可以运行于固件/软件。

3.6　边缘计算应用

边缘计算可以应用于从工业到零售、企业到智能家居的各种用例。以可解决集中式物联网问题的智能边缘相机为例，英国有数百万个用于监视道路上 2500～4000 万辆汽车车牌号的 CCTV 摄像机，因而面临着可以支持大流量的网络基础设施，以及传输和存储大数据的成本需求。此外，监视这些 CCTV 所需的人力资源和人工监视的效率也是棘手的难题。每年用于监视摄像机的费用约为 22 亿英镑。

通过在摄像头（如固定的，移动式或穿戴式）边缘或附近部署 AI 功能（包括第

5章中讨论的机器视觉功能），从网络成本和人力资源的角度来看，自动人脸识别（Automated Facial Recognition，AFR）、自动车牌识别（Automated Number-Plate Recognition，ANPR）之类的分析功能非常有效。这得益于将所有智能分析和检测功能放在本地和靠近摄像头的边缘计算设备上，并且仅将分析结果、统计信息和异常情况反馈到集中式 IoT 平台。这样既可以节省成本和带宽、提高隐私和安全性，还可以将执行速度提高几个数量级。例如，如果警察在特定区域内寻找犯罪嫌疑人，可以立即将 AFR 应用程序推送到相关位置的所有闭路监控中，并在边缘设备上优先执行此任务。所有闭路监控都可以同时扫描犯罪嫌疑人的脸部或汽车，当且仅当检测到犯罪嫌疑人时，才将高分辨率镜头回传。同时，可以向该地区周围的警察发出警报，告知犯罪嫌疑人的位置以及去向。最后，事发地边缘设备可以操纵相邻路口的交通信号灯，并通过制造人为的交通拥堵困住犯罪嫌疑人，直到警察到达。

类似的技术可以用于工业、零售业或服务业等场景，在这种情况下，普通摄像头被视作多用途智能传感器，既可以计算停车场人数、汽车数量、自行车数量和可用停车位数量，又可以测量交通流速和道路拥挤程度，并检查工人是否穿着醒目的背心和外套。

3.7 总结

物联网解决方案已经融入设备并不断生成大量数据，然后传输到数据中心的高度分布式环境中。这不仅导致网络带宽和计算资源利用率低下，还会导致物联网应用程序的高延迟时间响应。为了减少服务响应时间和网络流量，可以将计算资源从云基础设施扩展到紧邻 IoT 设备的网络边缘。

本章全面介绍了边缘计算相关技术，包括边缘计算基础、边缘计算架构、边缘计算解决方案实践，以及利用零接触设备上线的现代化方法，使设备能够快速开机，并自动在集中式 IoT 平台内注册，并构建了隐私保护和设备安全的基准。最后概述了边缘计算的各种实际应用。未来研究的重要领域将涉及大量其他现代计算技术，例如，雾计算和渗透计算，以及如何利用这些方法扩展和增强边缘计算能力。

参考文献