



下载DZone的2019 Scaling DevOps趋势报告，了解如何在扩展DevOps时确保安全性。

下载报告▶

独立5G网络中MEC部署的UE应用启动和卸载

通过 Harpreet Kaur · 2月21日，19 · IoT Zone · 介绍

Download the 2019 Guide to Containers to see why container-related tools have become indispensable to software c and how you can make the most of them. 译
Presented by DZone

5G是为满足未来网络的容量和性能要求而强制要求的颠覆性技术。新兴的应用程序（如AI，IoT，AR / VR）需要大量的带宽需求和极低的延迟要求，需要通过其他新兴技术（如SDN / NFV和多接入边缘计算（MEC））来促进5G。通过使计算更接近用户，MEC承诺满足所需的延迟和带宽限制。标准化机构，如3GPP（用于5G）和ETSI（用于MEC），一直致力于简化5G核心和MEC系统互通的程序。5G和MEC规范提供了对未来预期集成策略的深入了解 - 使MEC作为5G应用功能与3GPP 5G系统进行交互，以实现流量转向和移动事件的接收。但是，MEC功能实体与5G核心网络功能之间在应用程序启动和UE移动性方面的完整信息流似乎在此时缺失。本文旨在深入研究其中一些交互工作问题，并解释参与实体在整个应用程序生命周期中的交互。

关键词 - MEC（多接入边缘计算），5G（第5代），UE应用卸载，5G应用功能

介绍

多访问边缘计算有助于在5G部署中实现超级交互式，丰富用户应用程序的所需KPI。为了实现这种性能，UE应用程序被分成模块，其中CPU密集型和延迟敏感型应用程序托管在最靠近消费者的网络边缘。由于这是网络结构的一部分，因此在将这些MEC主机部署到5G网络时，需要UE应用前端，MEC边缘主机上部署的网络功能和5G核心网络功能之间的互通。另外，期望UE是移动的。由于UE位置的改变，处理UE应用的MEC主机可能变得非最佳。因此，为了维护应用程序的性能要求，可能需要将应用程序从一个MEC主机卸载到另一个主机。检测何时需要应用程序卸载以及将UE流量引导到新MEC主机所需的改变将再次需要来自5G核心网络功能的支持。本文旨在展示MEC功能实体与5G核心网络功能之间的互通，以便在部署在5G网络中的MEC主机上运行UE应用。

本文的其余部分安排如下。第二节提到了UE应用程序卸载将发挥关键作用的一些实际用例。第三节介绍了MEC-5G集成以及此类部署面临的挑战。第VI节至第VIII节详述了每当UE应用程序处理被卸载到边缘时，MEC系统和5G核心之间的完整程序流程。

在继续之前，我们建议您快速阅读附录A.1（介绍5G核心网络功能实体）和附录A.2（介绍MEC功能实体），以熟悉本文后面使用的术语。表1列出了本文中使用的缩写。

II. 用例

当底层网络支持UE移动性时，UE可以移动到连接到与当前服务ME主机不同的ME主机的网络实体。启用新MEC主机以服务UE需要维持UE与移动边缘应用实例之间的连接以及应用实例重定位或UE状态/上下文重定位。UE移动性可能触发这种应用卸载到新MEC主机的少数情况是底层网络中的UE承载路径改变，这导致UE移出源ME主机的覆盖区域到另一ME主机的覆盖区域（目标）。

任务关键型低延迟应用程序需要几毫秒的延迟。为了支持这种低延迟，如果用户从一个小区移动到另一个小区，则需要将ME应用程序重新定位在用户附近。

3GPP	第三代合作伙伴计划	NAS	非接入层
5GC	第五代核心	NEF	网络曝光功能
AF	应用功能	NF	网络功能
AMF	访问和移动管理功能	NRF	网络存储库功能
APPD	应用描述符	OSS	运营支持系统

CN	核心网络	PCC	政策和收费控制
CUPS	控制和用户平面分离	PCF	政策控制功能
DN	数据网络	PDU	协议数据单元
DNAI	数据网络访问标识符	PGW	分组数据网关
DNN	数据网络名称	PLMN	公共陆地移动网络
EPC	演进分组核心	RAN	无线接入网
FE	功能实体	鼠	无线接入技术
FQDN	完全合格的域名	SGW	服务网关
GPSI	通用公共订阅标识符	SMF	会话管理功能
GUAMI	全球独一无二的AMF ID	S-NSSAI	订阅的网络片选择辅助信息
GUTI	全球唯一的临时标识符	TA	跟踪区域
KPI	关键绩效指标	TAI	跟踪区域标识
LADN	局域网数据	UDM	统一数据管理
LCM	生命周期管理	UDR	统一数据存储库
MEC	多访问边缘计算	UE	用户设备
MEO	Mobile Edge Orchestrator	UL	上行
MEP	移动边缘平台	UPF	用户平面功能
MME	移动管理实体	URI	统一资源标识符

表1：缩写列表

III。定义问题陈述

事实证明，4G的进步是5G网络的强大基础。4G EPC利用CUPS架构 - 控制平面和用户平面分离。通过虚拟化，网络节点SGW和PGW的独立可扩展性变得简单。5G是对这种方法的进一步改进。SGW-C和PGW-C与SGW-U和PGW-U完全分离，将所有用户平面功能（例如 - 分组路由/转发，策略规则实施，QoS处理，流量使用报告）移至新的5G功能称为“用户平面功能”或UPF。4G的MME，SGW-C和PGW-C功能在5G中移至AMF和SMF。为了实现基于NFV的功能所期望的模块化，访问管理和移动性管理被移动到AMF，并且用户会话管理被移动到SMF。

独立的5G网络基于基于服务的体系结构（[1]的第4.2节） - 5G中的每个网络功能（AMF，SMF，NEF和其他）是一个独立的服务，可以为其他组件提供某些服务。可以独立访问和更新这些服务。这种能力对于5G-MEC部署互通也是至关重要的，因为MEC功能实体可以使用5G核心网络功能的基于服务的接口并检索/更新相关信息。例如，MEC功能实体可以使用这些接口来检索UE位置或将流量规则传递到5G核心网络。这使得有可能实现，例如，移动应用程序从一个LADN卸载到另一个LADN，如我们的原始用例（II。用例）中所讨论的。

将用户平面完全分离到UPF有助于在5G中单独放置和缩放用户平面和控制平面。UPF现在可以放置在靠近网络边缘的位置，并且可以与接入网络中的MEC主机集成。MEC主机位于UPF和数据网络之间的路径中，如图1所示。为了将数据包传递到靠近服务于UE的UPF的MEC主机，5G系统引入了将上行链路分类器与UPF。该UL分类器可以触发向MEC主机的流量转向。5G CN中的SMF配置UL分类器上的流量规则以实现此流量控制。

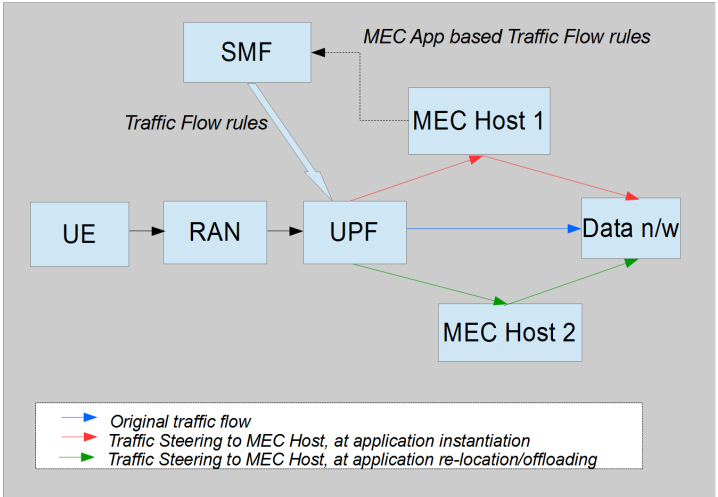


图1：问题陈述

5G的这些固有功能为与MEC部署的成功集成铺平了道路。虽然5G 3GPP规范解释了5G核心网络功能如何相互作用并允许外部应用程序功能访问它们，但是缺少MEC FE和5G核心网络功能之间的完整流程。MEC FE需要一种机制来将流量规则（基于UE应用程序要求）提供给SMF。为此，MEC FE需要检测为UE服务的SMF。他们还需要发现所选择的MEC主机何时由于UE移动性而变得非最佳，并且由于应用程序卸载到新的最佳MEC主机而触发流量规则更新。本文的其余部分将深入探讨其中一些问题，并解释参与实体在整个应用程序生命周期中的交互。

应用程序生命周期分为三个阶段 - 所选MEC主机上的应用程序实例化，由于UE移动性而向新MEC主机上的应用程序迁移以及应用程序终止。在应用实例化之前，作为UE注册和PDU会话建立的一部分，一些UE上下文被保存在5G核心网络中。需要此上下文信息来选择MEC主机以实例化应用程序实例或将其移动到来自核心网络的位置更新的新MEC主机。

在应用程序实例化之前

要在MEC主机上触发应用程序实例化或连接到核心的任何服务，UE需要进行注册并与核心网络建立用户上下文，与AMF建立信令连接，最后通过DN建立PDU会话SMF，如图2所示。SMF为UE选择UPF，并根据在UE注册阶段获得的订阅信息，配置业务规则以将来自所选UPF的业务引导到本地DN。这是UE可以将PDU发送到DN的时间。UE注册和PDU会话建立在附录A.3中更详细地解释：（5G中的UE注册和PDU会话建立）。

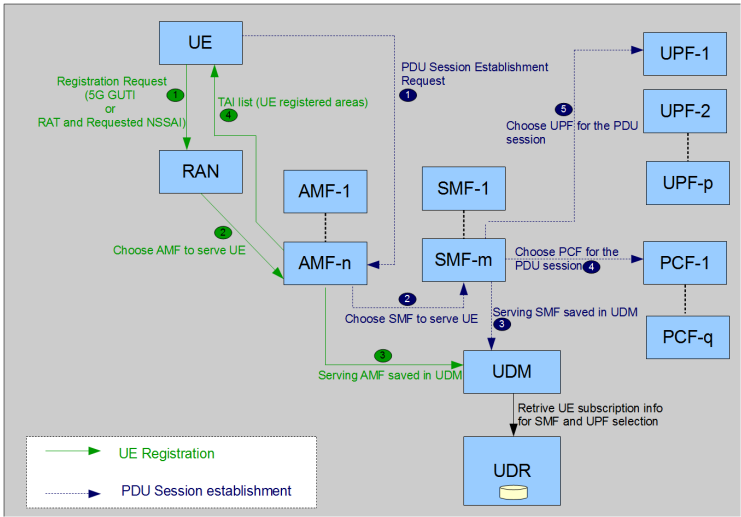


图2：MEC App启动前的5G CN程序

在UE注册之后，在UE注册的接入网络中为UE服务的AMF被保存在UDM中。服务AMF注册其自身以从RAN接收UE的位置更新。在服务于UE的小区不在作为UE注册的一部分接收的TAI列表（跟踪区域列表）的情况下，该服务AMF还处理来自UE的移动性注册更新。

类似地，为PDU会话建立选择的SMF保存在UDM中。该服务SMF从服务AMF注册UE的位置更新。NEF或PCF可以从UDM检索服务SMF详细信息，以便在应用程序实例化和重定位时提供流量规则，如本文后面部分所述。

MEC主机上的应用实例化

MEC功能实体通常分为两部分 - 位于核心网络的MEC系统级实体（包括用户应用程序LCM代理，OSS和MEO）和MEC主机级实体（包

<https://dzone.com/articles/ue-application-initiation-amp-offloading-on-mec-de>

括部署的实际Mobile边缘主机）到UE和移动边缘平台）。一旦建立了PDU会话，UE就可以在MEC系统上触发应用实例化。

在应用程序实例化之前，应用程序包需要加入MEC系统。存在两种替代方案 - 通过OSS将应用程序包上的应用程序提供者提供给MEO（[5]的第5.2.2节）或者将 appPackageSource URI（应用程序包的URI）提供给UE应用程序（由应用程序提供者提供）然后UE应用程序在发送实例化应用程序的请求时将此URI传递给OSS（[9]的第5.1.3和6.2.3节）。OSS可以使用此URI触发包的入门。

一旦对应于期望的UE应用的应用包在MEO上加入并启用，UE就可以通过用户应用LCM代理向MEO发送“实例化应用”或“应用上下文创建”请求。该板载应用包包含应用程序描述符（AppD）指定应用程序要求和所需的流量导向规则。这些流量导向规则包括用于识别数据包的流量过滤器，要采取的操作以及用于接收数据包的目标接口。

MEO选择最佳MEC主机（基于约束，例如延迟，可用资源和可用服务）来为该请求提供服务。但是为了触发流向最佳MEC主机的流量，MEO需要一种与5G核心网络功能接口的机制，如图3所示。

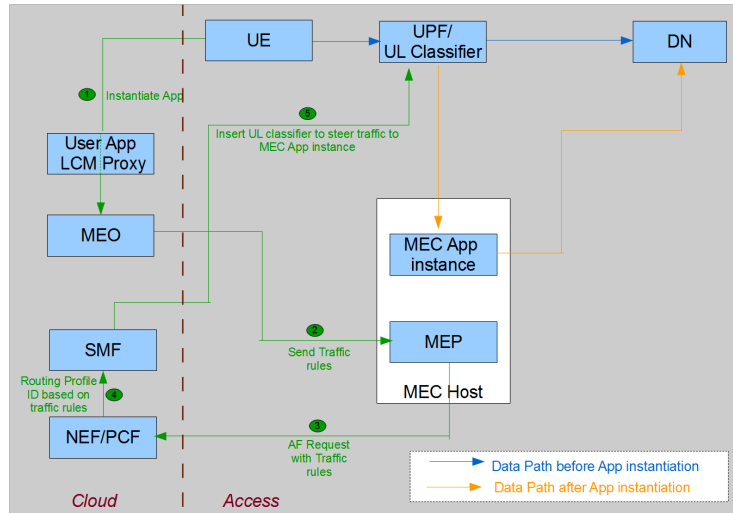


图3：MEC App启动时的5G CN程序

交通转向MEC主机

应用 AppD 程序包中的应用程序描述符 包含数据包到达MEC主机上运行的MEC应用程序实例所需的流量规则。当应用程序包被启用时，MEO获取此应用程序描述符。作为处理应用程序实例化或上下文创建的一部分，MEO将此应用程序描述符中的流量控制规则传递给所选MEC主机的MEP。

MEP充当“应用功能”并通过NEF与5GC交互。MEP触发针对NEF的HTTP POST AF请求（[6]流量影响程序的第4.4.7节）。为了使5GC识别UE并执行适当的流量导向，从MEP到NEF的应用功能请求（AF请求）包含以下信息：

- UE在GPSI或Ipv4 / v6地址方面的身份
- 流量描述 - 流量过滤器，DNN和S-NSSAI - 用于对属于应用程序的数据包进行分类
- 流量路由信息 - N6流量路由信息或预先配置有5GC的路由配置文件ID，指定所选择的MEC主机和接收流量的接口。此路由信息标识在其中部署所选MEC主机的数据网络的IPv6地址以及数据网络中此MEC主机的N6隧道端的UDP端口号

从AppD（在应用程序包中）检索的流量导向规则和与所选择的MEC主机相对应的N6接口信息用于填充该AF请求的流量描述和流量路由信息。TrafficInfluSub [6]中的数据模型清楚地定义了完整的AF请求格式。NEF负责将来自MEP的AF请求提供的信息映射到5GC所需的信息，并将该信息传递给PCF。

PCF接收所接收的AF请求并识别相应的路由简档Id（指定PCC规则）。可以在5GC和AF之间预先配置这些路由简档ID。在这种情况下，AF请求可以传递这些路由简档ID而不是N6流量路由信息。这些ID也在SMF上预先配置。PCF将选定的策略/配置文件ID传递给SMF。因此，SMF在为UE服务的UPF上配置流量规则。

为了在UPF上配置这些MEC触发的业务导向规则，SMF配置UL分类器，其被插入UE的数据路径中。最终将流量转向规则安装在此UL分类器上。UL分类器基于“业务描述”字段对应用的数据分组进行分类，并基于业务导向规则的“业务路由”信息将它们转发到所选择的MEC主机。

位置更新来自5G CN

在5G CN中，AMF负责监视UE位置并向感兴趣的实体发送位置更新通知（UE移动性事件通知）。AMF有多种方法可以识别UE位置是否已更新：

- AMF可以请求RAN进行位置报告 - 然后RAN将向AMF发送位置报告消息，包括UE的最后已知位置和时间戳。AMF可以请求连续

报告模式以获得定期更新。或者它可以基于报告请求“感兴趣区域（根据UE注册的TAI列表）”。在这种情况下，如果RAN检测到感兴趣区域中的UE存在与报告的最后一个不同，则RAN可以向AMF发送通知。

- UE在注册响应中传递TAI列表。在UE检测到服务于UE的小区不在该列表中的情况下，它可以用AMF触发“移动性注册更新”。

通过5G中基于服务的架构，AMF向作为消费者的其他服务或网络功能提供UE移动性事件通知。任何NF消费者（SMF，PCF或NEF）都可以订阅AMF，以接收报告“感兴趣区域”中“UE存在”变化的通知。AMF跟踪UE位置并确定“感兴趣区域”中的“UE存在”。在检测到改变时，AMF将新UE位置通知给订阅的NF消费者（SMF或NEF）。

如果SMF订阅AMF以在AMF检测到UE位置相对于感兴趣区域的变化时接收通知，则AMF向SMF通知DNAI的任何变化。SMF可以进一步触发用户平面管理事件朝向订阅这些事件的AF或5G NF（NEF）。

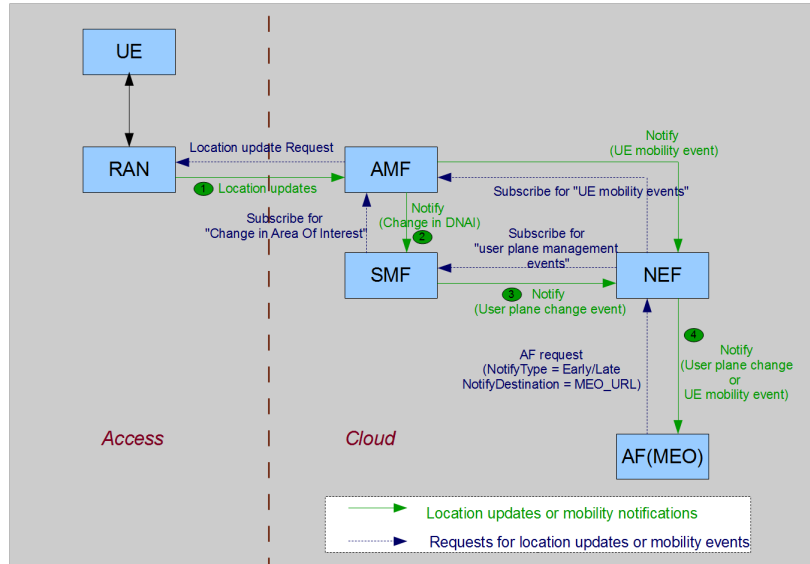


图4：MEC FE的位置更新和移动性事件

为了在MEC功能实体处获得这些通知，MEO可以充当AF并订阅NEF，如上面的图4所示。从MEO到NEF的AF请求中的“TrafficInfluSub”[6]字段定义了用于向NEF注册关于用户平面管理事件的订阅的参数。其中一个参数是 notificationDestination - 它包含用于接收来自NEF的通知的回调URL。MEO可以包含自己的URL notificationDestination。每当NEF从SMF接收用户平面管理事件通知时，NEF将包括所通知的事件的HTTP发布消息发送到MEO。类似地，NEF可以使用MEO URL作为通知地址直接从AMF订阅UE移动性事件。

应用程序卸载到新的MEC主机

在获得新UE位置（经由UE移动性事件或用户平面管理事件）时，MEO检查UE移动到的新区域是否由相同的MEC主机服务。如果不是，则MEO需要将应用程序实例从旧的MEC主机重新定位到服务于UE的当前位置的新MEC主机。在应用程序是无状态的情况下，在新的MEC主机上不需要UE状态信息。但是对于有状态应用，系统需要UE上下文传送以支持到UE的服务连续性。该上下文信息可以存储在先前MEC主机中的UE app或app实例中。此应用程序实例重定位还将触发流量导向更新，以将流量发送到新的MEC主机[8]。此交通转向更新的MEC系统和5G CN交互如下图5所示：

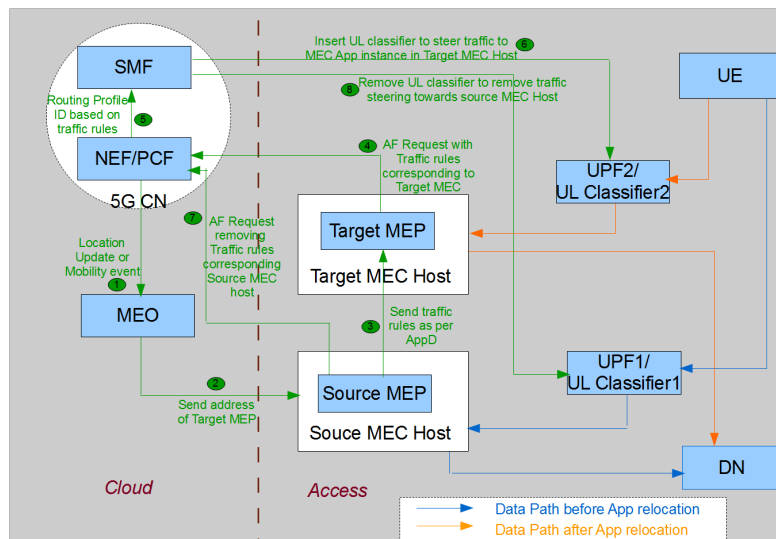


图5： MEC App重定位时的5G CN程序

MEO将目标MEP的地址传递给源MEP，作为其应用程序重定位过程的一部分。源MEP将应用的流量控制规则传递给目标MEP。目标MEP充当5G AF以触发朝向NEF的AF请求，将该目标MEC主机指定为N6流量路由信息中的目的地。NEF将此新请求传递给PCF。PCF将触发针对SMF的更新规则，并且SMF将在UE的当前数据路径中插入新的UL分类器。该UL分类器将引导分组朝向目标MEC主机。

一旦配置了最新规则以将数据包发送到目标MEC主机，源MEP也可以发送AF请求以移除先前配置的流量控制规则。

摘要

该文件阐明了5G核心网络实体和MEC功能实体之间在MEC边缘主机上实现UE应用的紧密交互。由于UE移动性触发的应用程序重定位，它还试图引出这些实体之间的程序流程。

附录A.1（介绍5G核心网功能实体）

3GPP 5G系统架构[1]详细解释了所有网络功能。图6显示了这些网络功能如何取代4G EPC功能。虽然所有网络功能在UE应用程序启动和卸载中都有一定的作用，但为了更好地理解本文，表2列出了对于知道至关重要的功能。

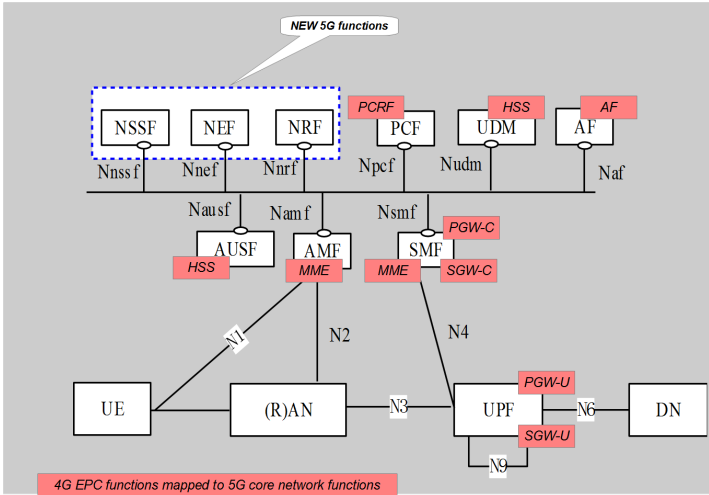


图6：映射4G EPC功能和5G核心网络功能

5G CN功能	映射到4G EPC	功能
访问和移动管理功能（AMF）	MME - 移动管理	一个。处理UE注册（初始/周期/移动性触发） b. 建立/释放与UE的NAS信令连接以用于会话管理和移动性消息 c. 跟踪UE位置并向其他网络功能提供UE移动性相关事件
会话管理功能（SMF）	MME - 会话管理 S-GW和P-GW的控制平面功能（SGW-C, PGW-C）	一个。建立/终止UE与数据网络（DN）之间的PDU会话 b. UE IP地址分配和管理 c. 选择UPF来处理UE流量 d. 在UPF配置流量控制以将流量路由到所选数据网络。
用户平面功能（UPF）	S-GW和P-GW（SGW-U, PGW-U）的用户流量传输功能	一个。外部PDU与数据网络互连的会话点。 湾 执行SMF配置的流量导向规则，以将流量路由到所选的数据网络实例
网络曝光功能（NEF）	不存在	一个。向外部实体公开5G核心网络功能的能力和事件（运营商不信任直接访问5G核心网络功能） b. 可用于监控（例如，UE位置相关事件）和配置（PDU会话的流量规则）
应用功能（AF）	应用功能（AF）	一个。5G核心网外部的网络功能，并使用这些5G核心网络功能 b. 操作员信任的应用程序功能直接与5G核心网络功能交互 c. 其他应用功能通过NEF与5G核心网络交互
政策控制功能（PCF）	PCRF	一个。处理应用程序功能的流量转向请求，并将其映射到SMF理解的流量控制策略

网络存储库功能（NRF）	不存在	一个。根据类型，PLMN Id, FQDN或IP地址维护5G核心网络功能实例的配置文件。 湾 帮助发现网络功能的实例
统一数据管理（UDM）	HSS	一个。提供UE订阅数据和用户上下文 b. 存储为UE服务的AMF和SMF ID。 C. 保持正在进行的PDU会话的SMF / DNN映射，以实现会话连续性
统一数据存储库（UDR）	HSS	一个。存储要由UDM b 访问的UE订阅数据。存储PCF要访问的策略数据 c. 存储应用程序功能AF共享的应用程序数据（流量检测和流量控制方面的要求）

表2：介绍5G核心网络功能

附录A.2（MEC功能实体介绍）

ETSI发布的MEC规范[3]展示了在移动网络中运行边缘应用所需的多接入边缘计算的框架和参考架构。表3提到了MEC参考架构中的功能块，这些功能块与理解这篇文章有关。

MEC功能实体	功能
UE app	一个。应用程序在UE设备中运行。
用户应用程序 LCM代理	一个。充当UE app和MEC系统级实体之间的接口。所有UE请求检索MEC主机上已经运行的应用程序实例或触发应用程序入门/实例化都由User app LCM代理授权并进一步传递到MEC系统以进行操作。
移动边缘主机（MEC主机）	一个。UE的局域网数据网络（LADN）中的边缘主机。 。截取/处理从UE到DN的分组，反之亦然
移动边缘协调器（MEO）	一个。从UE b 的LADN中的所有可用MEC主机确定UE的特定MEC主机。在通过UE app LCM模块或通过OSS获取UE应用程序请求时，触发MEC主机上的应用程序实例化
移动边缘应用（MEC应用）	一个。在MEC主机中运行的应用程序用于处理用户流量
移动边缘平台（MEP）	一个。根据UE应用程序的流量规则要求，帮助配置数据平面以引导进出MEC应用程序的流量

表3：介绍MEC功能块

附录A.3 :(5G中的UE注册和PDU会话建立)

从UE到RAN的注册请求包含由PLMN分配给UE的5G-GUTI（全球唯一临时标识符）。该5G-GUTI由GUAMI（全球独特的AMF ID）和5G-TMSI组成。RAN基于GUAMI识别的AMF ID选择AMF来服务该UE。如果5G-GUTI不可用，它可以根据RAT选择AMF并在注册请求中请求NSSAI。在成功注册时，在UDM中针对UE注册服务AMF-ID，并且UE接收TAI列表（跟踪区域标识）。这些跟踪区域用作UE的注册区域。在任何时间点，如果服务于UE的小区不在该TAI列表中，则UE发起移动性注册更新 - 这触发了5G核心网络中的移动性事件。

在成功注册之后，UE建立到AMF的信令连接以交换NAS信令消息（经由AMF从UE到SMF的PDU会话建立）。AMF还选择将管理UE与DN之间的PDU会话的SMF。AMF可以从NRF检索可用的SMF实例列表，也可以使用AMF上本地配置的信息。基于DNN（数据网络名称），S-NSSAI，存储在UDM / UDR中的UE的订阅信息，运营商策略以及SMF实例的负载条件来选择SMF。用于UE的PDU会话的该服务SMF也存储在UDM中。

作为会话管理的一部分，SMF选择UPF来处理PDU会话数据流量。PDU会话建立请求包含DNN和S-NSSAI。SMF向NRF发送DNN，S-NSSAI和SMF区域标识。作为回报，NRF返回可用的UPF实例的IP地址或FQDN。SMF根据UPF实例的负载条件，UPF位置，UE位置，UDM / UDR中的UE订阅信息等从这些实例中选择一个UPF。

根据DNN和运营商策略，SMF还为此PDU会话选择PCF [4]。PCF从UDR检索策略并将这些流量处理策略传递给SMF，SMF最终将它们

传递给UPF。

参考

- [1] 3GPP TS 23.501: “5G系统的系统架构;第2阶段（版本15）”。
- [2] 3GPP TS 23.502: “5G系统的过程;阶段2（版本15）”。
- [3] ETSI GS MEC 003: “移动边缘计算（MEC）;框架和参考架构”。
- [4] 3GPP TS 23.503: “5G系统的策略和计费控制框架;第2阶段（版本15）”。
- [5] ETSI GS MEC 010-2 V1.1.1（2017-07）: “移动边缘管理;第2部分: 应用程序生命周期, 规则和需求管理”。
- [6] ETSI TS 129 522 V15.0.0（2018-07）: “5G; 5G系统;网络暴露功能北向API;第3阶段（3GPP TS 29.522第15.0.0版本15）”。
- [7] ETSI第28号白皮书: “5G网络中的MEC”
- [8] ETSI GR MEC 018: “移动边缘计算（MEC）;端到端移动性方面”。
- [9] ETSI GS MEC 016 V1.1.1（2017-09）: “移动边缘计算（MEC）; UE应用程序接口”。

Download DZone's 2019 Scaling DevOps Trend Report to learn how to ensure security as you scale DevOps, why “it's all retrospectives” are a myth, and the key considerations for getting the most out of DevOps as you scale. [Download N](#)

Presented by DZone

喜欢这篇文章？阅读更多来自DZone



什么是边缘计算？



趋势警报：边缘计算如何推动物联网



开源物联网边缘项目



免费DZone Refcard
边缘计算

话题：5G，边缘计算应用程序，边缘计算，边缘计算，IOT，MEC，UE，UE应用程序

DZone贡献者表达的意见是他们自己的。

用于工业物联网和自动化的Apache Kafka，KSQL和Apache PLC4X

通过 凯瓦纳 · MVB · 9月06日，19 · 物联网区 · 教程





通过Apache Kafka, KSQL和Apache PLC4X了解有关IIoT自动化的更多信息

Data integration and processing is a huge challenge in Industrial IoT (IIoT, aka Industry 4.0 or Automation Industry) due to monolithic systems and proprietary protocols. Apache Kafka, its ecosystem (Kafka Connect, KSQL), and Apache PLC4X are a great open-source choice to implement this IIoT integration end-to-end in a scalable, reliable, and flexible way.

This blog post covers a high-level overview of the challenges and good, flexible architecture to solve the problems. In the end, I share a video recording and the corresponding slide deck. These provide many more details and insights.

Challenges in IIoT/Industry 4.0

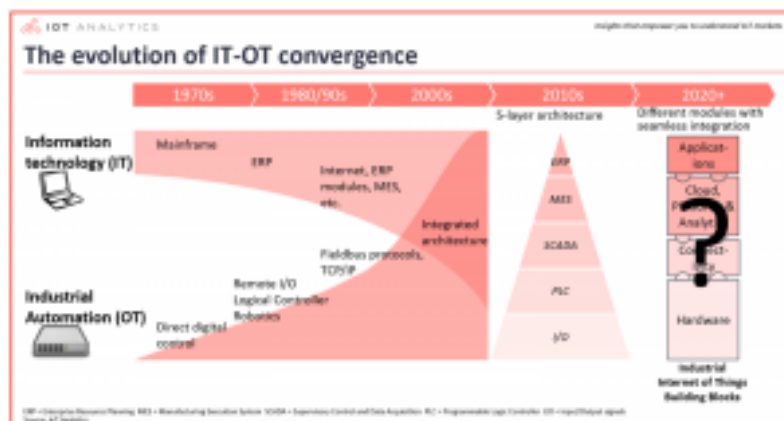
Here are some of the key challenges in IIoT/Industry 4.0:

- **IoT = IIoT:** Automation industry does not use MQTT or other standards, but is slow, insecure, not scalable, and proprietary.
- **Product lifecycles** are very long (tens of years), no simple changes or upgrades
- IIoT usually uses **incompatible protocols, typically proprietary, and just built for one specific vendor.**
- Automation industry uses **proprietary and expensive monoliths which are not scalable and not extendible.**
- Machines and PLCs are **insecure by nature with no authentication, no authorization, no encryption.**

This is still state of the art in the automation industry. This is no surprise with such long product life cycles, but still very concerning.

Evolution of Convergence Between IT and Automation Industry

Today, everybody talks about cloud, big data analytics, machine learning, and real-time processing at scale. The convergence between IT and automation industry is coming, as the analyst report from IoT research company IoT Analytics shows:



There is huge demand to build an open, flexible, scalable platform. Many opportunities from a business and technical perspective:

- Cost reduction
- Flexibility
- Standards-based
- Scalability
- Extendibility

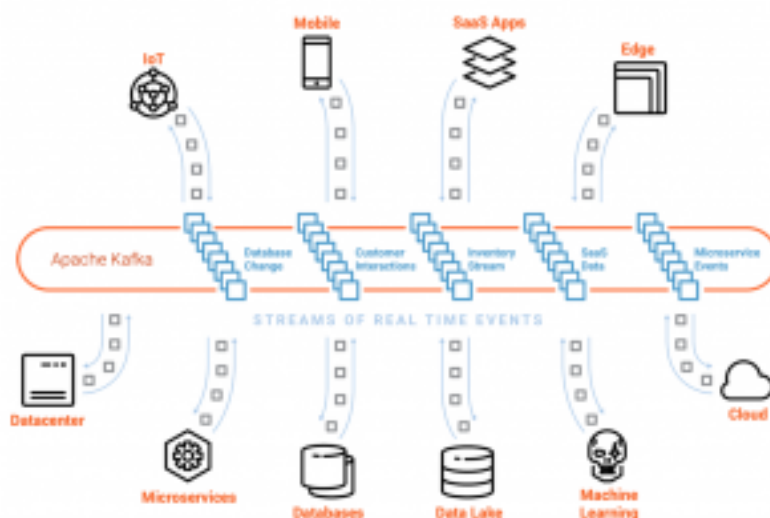
- Security
- Infrastructure-independent

So, how to get from legacy technologies and proprietary IIoT protocols to cloud, big data, machine learning, real-time processing?
How to build a reliable, scalable, and flexible architecture and infrastructure?

Apache Kafka and Apache PLC4X for End-to-End IIoT Integration

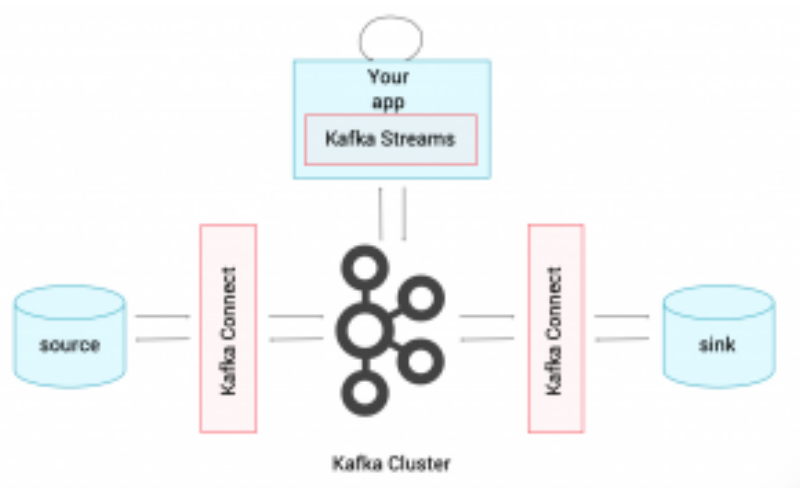
I assume you already know it: **Apache Kafka** is the de-facto standard for real-time event streaming. It provides

- Open-source (Apache 2.0 License)
- Global-scale
- Real-time
- Persistent storage
- Stream processing



If you need more details about Apache Kafka, check out the Kafka website, the extensive Confluent documentation, or some free video recordings and slides from any Kafka Summit to learn about the technology and use cases.

The only very important thing I want to point out is that **Apache Kafka includes Kafka Connect and Kafka Streams**:



Kafka Connect enables reliable and scalable integration of Kafka with other systems. Kafka Streams allows writing standard Java apps and microservices to continuously process your data in real-time with a lightweight stream processing API. And finally, KSQL enables Stream Processing using SQL-like Semantics.

Apache PLC4X for PLC Integration (Siemens S7, Modbus, Allen Bradley, Beckhoff ADS, etc.)

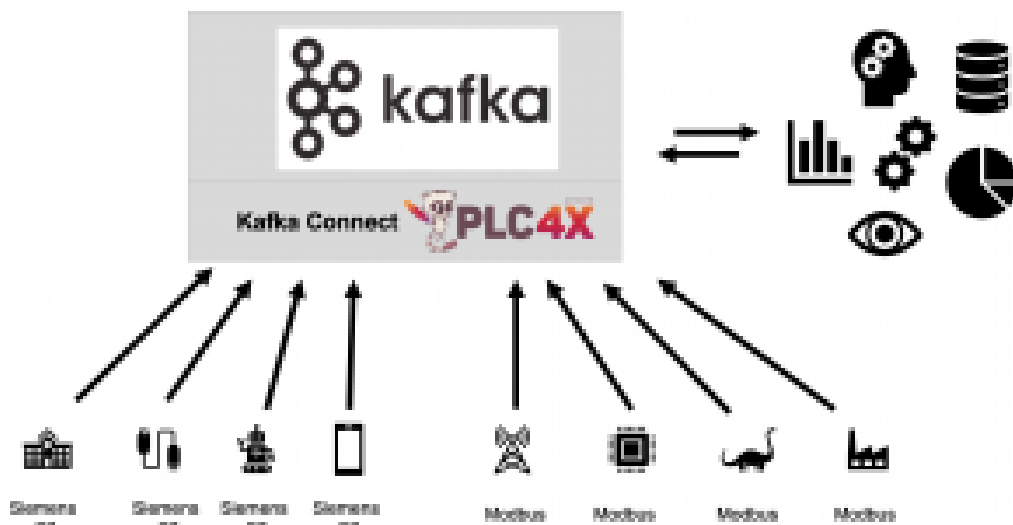
Apache PLC4X is less established on the market than Apache Kafka. It also "just covers a niche" (a big one, of course) compared to

Apache PLC4X is less established on the market than Apache Kafka. It also just covers a niche (a big one, of course) compared to Kafka, which is used in any industry for many different use cases. However, PLC4X is a very interesting top-level Apache project for the automation industry.

The goal is to open up PLC interfaces from IIoT world to the outside world. PLC4X allows vertical integration and to write software independent of PLCs using JDBC-like adapters for various protocols like Siemens S7, Modbus, Allen Bradley, Beckhoff ADS, OPC-UA, Emerson, Profinet, BACnet, Ethernet.

PLC4X provides a Kafka Connect connector. Therefore, you can leverage the benefits of Apache Kafka (high availability, high throughput, high scalability reliability, real-time processing) to deploy PLC4X integration pipelines. With this, you can build one single architecture and infrastructure for

- Legacy IIoT connectivity using PLC4X and Kafka Connect
- Data processing using Kafka Streams/KSQL
- integration with the rest of the enterprise using Kafka Connect and any other sink (database, big data analytics, machine learning, ERP, CRM, cloud services, custom business applications, etc.)



As Kafka decouples the producers from the consumers, you can consume the IIoT machine sensor data from any application - some might be real-time, some might be batch, and some might be request-response communication for human interaction on a web or mobile app.

Apache PLC4X Vs. OPC-UA

A little bit off-topic: How to choose between Apache PLC4X (open-source framework for IIoT) and OPC-UA (open standard for IIoT). In short, both are different things and can also be complementary. Here is a comparison:

OPC-UA

- Open standard
- All the pros and cons of an open standard (works with different vendors; slow adoption; inflexible, etc.)
- Often poorly implemented by the vendors
- Requires app server on top of PLC
- Every device has to be retrofitted with the ability to speak a new protocol and use a common client to speak with these devices
- Often over-engineering for just reading the data
- Activating OPC-UA support on existing PLCs greatly increases the load on the PLCs
- With licensing cost for every machine

Apache PLC4X

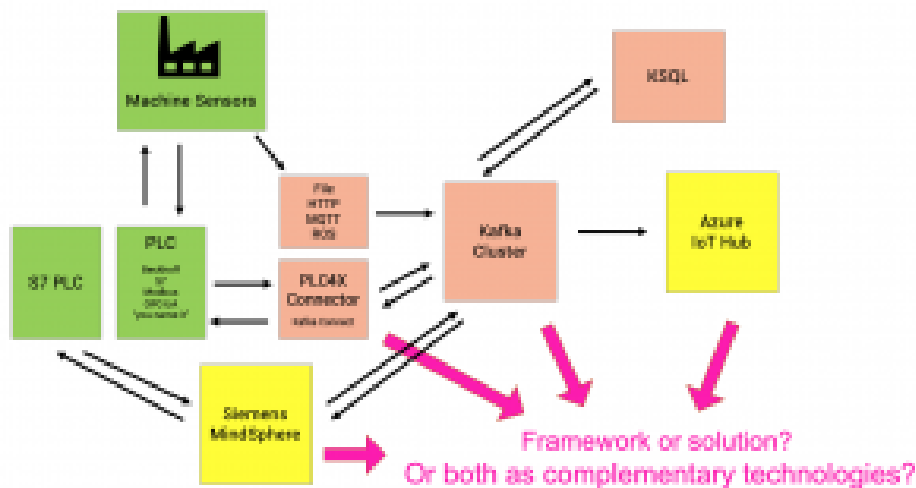
- Open-source framework (Apache 2.0 license)
- Provides unified API by implementing drivers for communicating with most industrial controllers in the protocols they natively understand

- No need to modify existing hardware
- No increased load on the PLCs
- No need to pay for licenses to activate OPC-UA support
- Drivers being implemented from the specs or by reverse engineering protocols in order to be fully Apache 2.0 licensed
- PLC4X adapter for OPC-UA available —> Both can be used together!

As you see, both have their pros and cons. To me, and this is clearly my subjective opinion, PLC4X provides greater alternatives with high flexibility and low footprint.

Confluent and IoT Platform Solutions

Many IoT Platform Solutions are available on the market. This includes products like Siemens MindSphere or Cisco Kinetic, and cloud services from the major cloud providers like AWS, GCP, or Azure. And you have Kafka + PLC4X as you just learned above. Often, this is not a "neither ... nor" decision:



You can either use

- Just Kafka and PLC4X for lightweight and flexible IIoT integration based on a scalable, reliable, and open event streaming platform
- Just an IoT Platform Solution if the pros of such a specific product (dedicated for a specific vendor protocol, nice GUI, etc.) outperform the cons (like high cost, proprietary and inflexible solution)
- Both together where you use the IoT Platform Solution to integrate with the PLCs and then send the data to Kafka to integrate with the rest of the enterprise (with all the benefits and added value Kafka brings)
- Both together where you use Kafka and PLC4X for PLC integration and one of the consumers is the IoT Platform Solution (while other consumers can also get the data from Kafka — fully decoupled from the IoT Platform Solution)

All alternatives have their pros and cons. There is no single solution which fits every use case! Therefore, no surprise that most IoT Solution Platforms provide Kafka source and sink connectors.

Apache Kafka and Apache PLC4X — Slides/Video Recording/GitHub Code Example

If you got curious about more details and insights, please check out my video recording and slide deck.

Slide Deck — Apache Kafka and PLC4X:

<https://www.slideshare.net/KaiWaehner/iiot-industry-40-with-apache-kafka-connect-ksql-apache-plc4x/KaiWaehner/iiot-industry-40-with-apache-kafka-connect-ksql-apache-plc4x>

GitHub Code Example — Apache Kafka and PLC4X:

We are also building a nice and simple demo on GitHub these days:

Kafka-native, end-to-end IIoT Data Integration, and Processing with Kafka Connect, KSQL, and Apache PLC4X

PLC4X gets most exciting if you try it out by yourself and connect to your machines or tools. So, check out the example and adjust it to connect to your infrastructure.

Feedback and Questions?

Please let me know your feedback and questions about Kafka, its ecosystem, and PLC4X for IIoT integration. Let's also connect on LinkedIn to discuss interesting IIoT use cases and technologies in the future.

Like This Article? Read More From DZone


IoT Integration With Kafka Connect, REST/HTTP, MQTT, and OPC-UA [Video]

Edge Computing: Unlocking the Business Value of the IoT

Top Three Industrial IoT Implementation Challenges

Free DZone Refcard Edge Computing

Topics: KAFKA, PLC, INTERNET OF THINGS, INDUSTRIAL IOT, IIOT, KSQL, KAFKA CONNECT PLATFORM, INTEGRATON, MIDDLEWARE, EDGE

Published at DZone with permission of Kai Wähler , DZone MVB. [See the original article here.](#) 
Opinions expressed by DZone contributors are their own.

Apache Kafka and Machine Learning for Real-Time Supply Chain Optimization in IIoT [Video]

by Kai Wähler  MVB · Sep 05, 19 · IoT Zone · Tutorial



Learn more about implementing real-time analytics for supply chain optimization in IIoT

I did a webinar with Confluent's partner Expero about "**Apache Kafka and Machine Learning for Real-Time Supply Chain Optimization.**" This is a great example for anybody in the automation industry/Industrial IoT (IIoT) industries, like automotive, manufacturing, logistics, etc.

You may also like: Creating a Competitive Edge in IIoT

In this presentation, we explain how a real-time event streaming platform can integrate in real-time with the legacy world and proprietary IIoT protocols (like Siemens S7, Modbus, Beckhoff ADS, OPC-UA, et al). You can process the data at scale and then ingest it into a modern database (like AWS S3, Snowflake, or MongoDB) or analytic/machine learning framework (like TensorFlow, PyTorch, or Azure Machine Learning Service).

We leverage various components from the **Apache Kafka ecosystem**. This includes:

- Kafka Connect as a scalable and reliable integration framework
- Kafka Connect connectors like PLC4X — a great Apache framework to integrate with IIoT protocols
- KSQL for continuous processing (filter, transform, aggregate) of the sensor data
- Kafka Streams to deploy the trained analytic models to a real-time streaming scoring engine

Use Case: Supply Chain Optimization Using Real-Time and Batch Processes

Automating multifaceted, complex workflows requires hybrid solutions including streaming analytics of IoT data and batch analytics. This includes machine learning solutions and real-time visualization. Leaders in organizations who are responsible for global supply chain planning are responsible for working with and integrating with data from disparate sources around the world. Many of these data sources output information in real-time. This assists planners in operationalizing plans and interacting with manufacturing output. IoT sensors on manufacturing equipment and inventory control systems feed real-time processing pipelines to match *actuals* productions figures against planned schedules to calculate yield efficiency.

Using information from both real-time systems and batch optimization, supply chain managers are able to economize operations and automate tedious inventory and manufacturing accounting processes. Sitting on top of all of these systems is a supply chain visualization tool. This enables users' visibility over the global supply chain. If you are responsible for key data integration initiatives, join for a detailed walkthrough of a customer's use of this system built using Confluent and Expero tools.

What will you learn?

- See different use cases in the automation industry and Industrial IoT (IIoT) where an event streaming platform adds business value
- Understand different architecture options to leverage Apache Kafka and Confluent Platform in IoT scenarios in the cloud, on-premise data centers and at the edge
- Learn how to leverage different analytics tools and machine learning frameworks in a flexible and scalable way
- How real-time visualization ties together streaming and batch analytics for business users, interpreters, and analysts
- Understand how streaming and batch analytics optimize the supply chain planning workflow.
- Conceptualize the intersection between resource utilization and manufacturing assets with long term planning and supply chain optimization.

Industrial Internet of Things (IIoT) in Real-Time at Scale With Apache Kafka

Here is the slide deck and video recording. Have fun watching it. Please let me know if you have any feedback or questions in the comments!

[Slides]

Video Recording:

Sorry

Because of its privacy settings, this video cannot be played here.

[Watch on Vimeo](#)

Further Reading

Pros and Cons of Implementing IoT in Supply Chain

[DZone Refcard] Getting Started With the Industrial Internet

Like This Article? Read More From DZone


**IoT Integration With Kafka Connect,
REST/HTTP, MQTT, and OPC-UA [Video]**

Fundamentals of Anomaly Detection

**A Comparison of Streaming Analytics
Frameworks and Products**

**Free DZone Refcard
Edge Computing**

Topics: KAFKA, INTERNET OF THINGS, INDUSTRIAL INTERNET OF THINGS, OPEN SOURCE, MACHINE LEARNING, MQTT, MODBUS, OPC UA, ANOMALY DETECTION, IOT

Published at DZone with permission of Kai Wähler , DZone MVB. [See the original article here.](#) 
Opinions expressed by DZone contributors are their own.