

Open Source AceCon

2021 智能云边开源峰会

AI x Cloud Native x Edge Computing

人工智能 × 云原生 × 边缘计算

EdgeX Foundry新版本特性

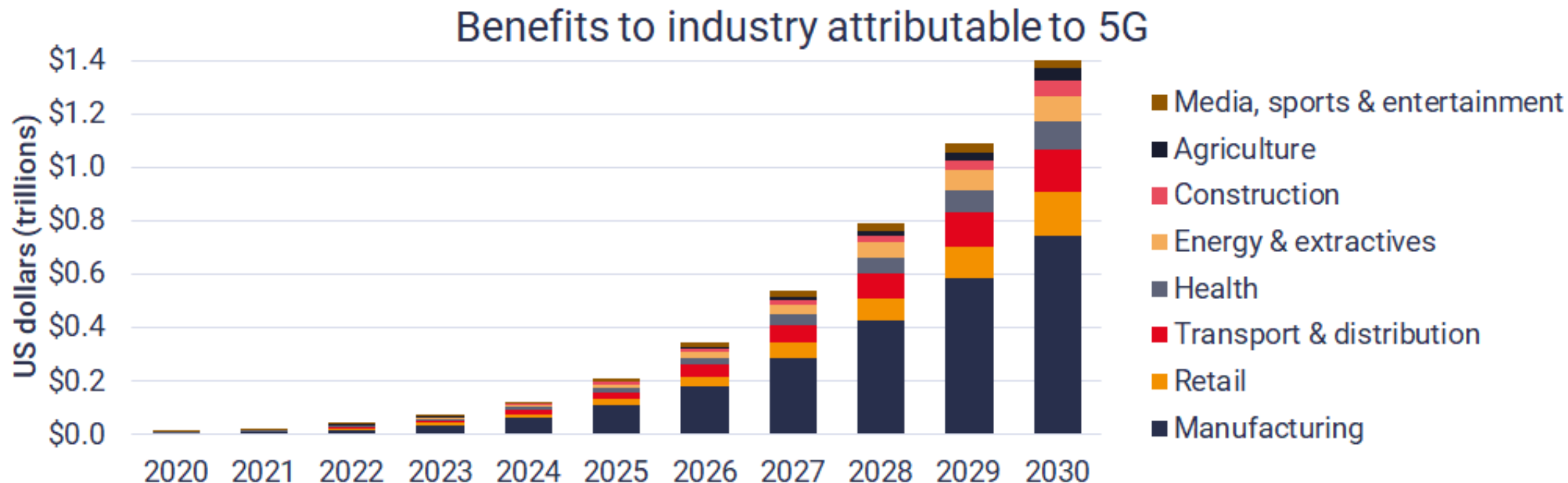
及中国社区的代码贡献

路广

VMware中国研发总监

EdgeX & eKuiper TSC成员

边缘计算是云计算的4倍并会冲击4万亿经济规模

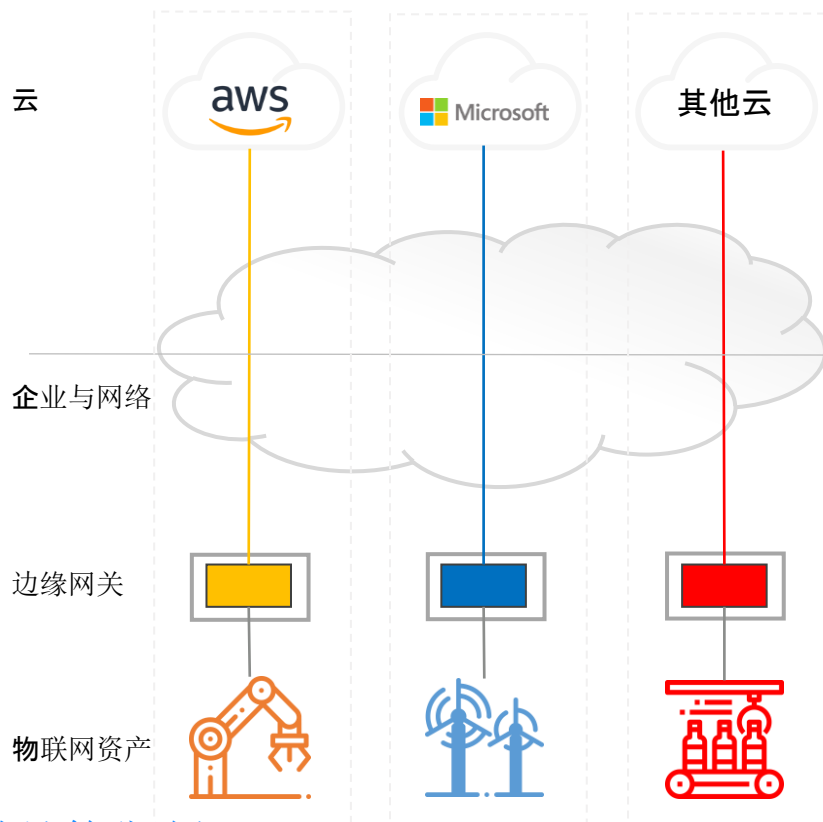


<http://www.chetansharma.com/publications/edge-internet-economy/>

多层边缘计算架构

今天的工业物联网

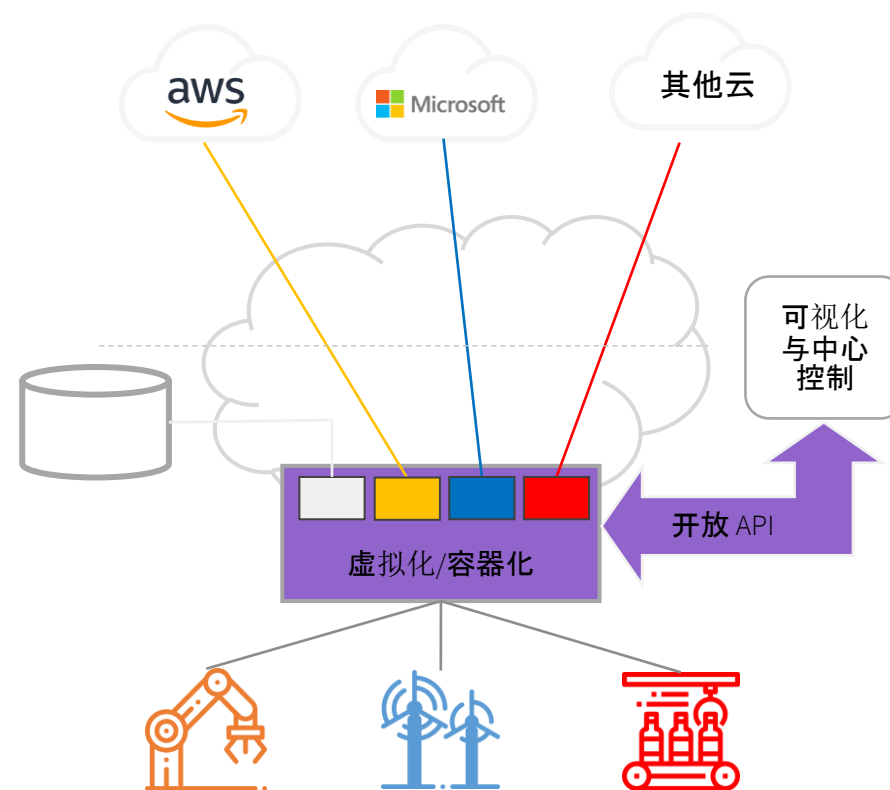
垂直数据竖井、平台锁定
数据/边缘治理与控制问题
硬件定义的、无法管理的边缘



边缘计算分论坛

未来的工业物联网

开放物联网数据架构，无锁定
数据与边缘属于企业
软件定义的边缘



边原生和云原生应用比较

云原生应用			边原生应用
构建	编程语言	最流行的语言和框架: Java, JavaScript, .Net, Go, Node.js, PHP, Python, and Ruby.	C, C++, Python, Go, JS.
	打包方式	轻量级的容器镜像	大多是二进制代码包, 最新有些容器/虚拟机镜像
	软件架构	松耦合的微服务架构, 有状态和无状态服务清楚分离	大多数是单块架构, 一些新应用以微服务设计
运行	应用程序接口	RESTful API	大多数是私有API, 有些新应用是REST
	CPU	x86-64	ARM 32/64, x86-64
	操作系统	大多数是Linux	Windows, Linux, 及其他各种实时OS
管理	生命周期	基于自服务、弹性的云基础设施, 自定义、基于策略的资源分配方式	对低延时、甚至硬实时相当强的需求, 管理能力相当有限
连接	I/O 接口	以太网, TCP/IP	TCP/IP, Wifi/蓝牙, ZigBee, 红外, Modbus, Can BUS, 及其他非常多的工业协议. 北向连接经常不稳定、非双向可见、不直接和窄带的.
保护	运行时隔离	硬件、虚拟机、OS级隔离	经常没有或者硬件隔离, 新应用有容器/虚拟机隔离
	安全	防火墙、IDS/IPS、杀毒、自动更新等全套保护	安全机制经常是不完整或不及时更新的

- EdgeX Foundry是Linux基金会运营的、厂商中立的开源软件项目，它是针对物联网边缘计算的通用开放框架。
- EdgeX Foundry的核心是一个互操作的框架，搭载于一个完全与硬件和OS无关的软件参考平台上，并使能即插即用的组件生态系统，统一化的市场，并加速物联网方案的部署。
- EdgeX Foundry架构与协议、硬件平台(x86/ARM)、OS(Linux, Windows, Mac OS)、编程语言(Java, JavaScript, Python, Go Lang, C/C++)无关，以支持客户差异化的选择。

以工业App生态系统带动事实标准

任意标准组合



互操作的边缘应用



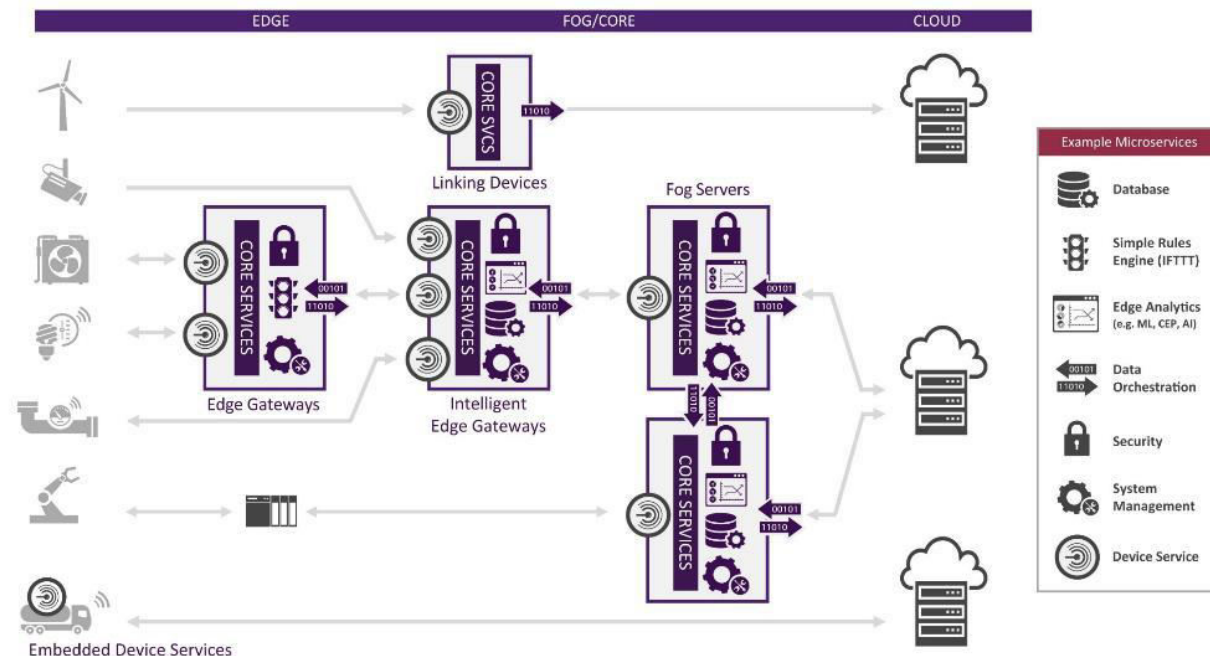
云端应用
任意选择



EdgeX框架使开放人员将边缘和云尽可能地解耦, 尽量避免重制采集数据、安全与管理所需的支撑部分, 从灵活性和选择多样性中收益。(自建还是采用开放中立生态中的现有技术)

EdgeX赋能分布式边缘侧部署

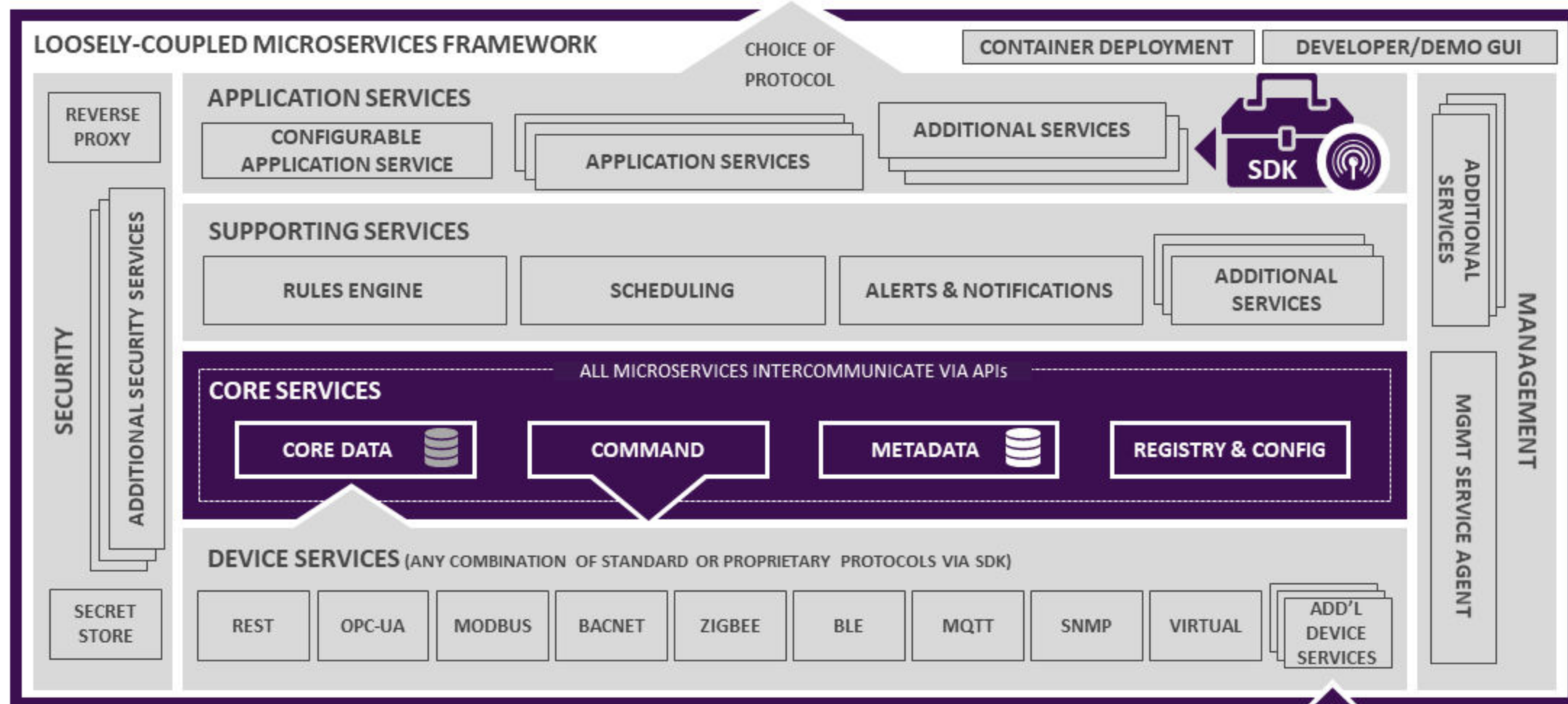
- 为分层边缘计算架构：松耦合架构使能跨节点部署
 - 应用于从智能传感器、控制器、边缘网关、路由器和服务器的设备上
- 高伸缩：指定节点上部署的微服务数量和功能依赖于硬件的用例和能力
 - 扩展云原生和函数即服务到边缘上，提高部署和生命周期管理的灵活性
- 不止于网关：整个平台可运行在单节点上、或分布于多节点
 - 离散设备服务可运行在有能力的智能传感器上，直接和其他后台系统通讯



与其他要么太死板、要么云中心化的开源软件不同，EdgeX Foundry是从头设计，赋能开放、中立、分布式的边缘计算生态系统。



"NORTHBOUND" INFRASTRUCTURE AND APPLICATIONS



"SOUTHBOUND" DEVICES, SENSORS AND ACTUATORS

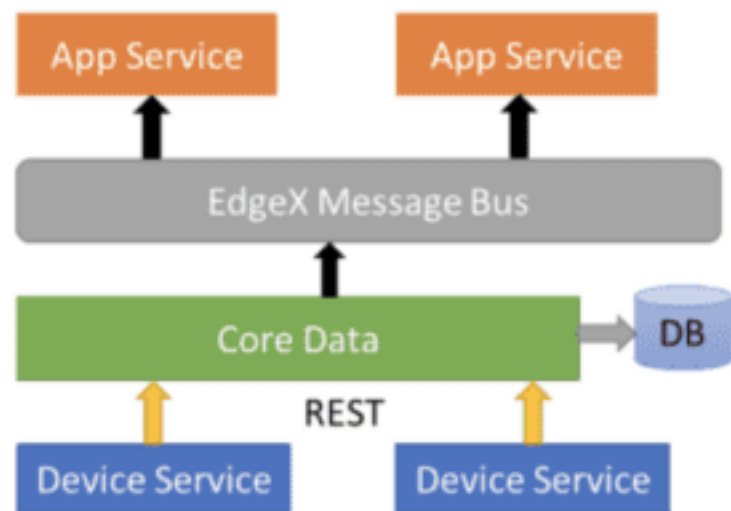
完全重写并改进的API v2.0

新 API 允许通信中的协议独立性、更好的消息安全性以及通过服务更好地跟踪数据。它们在 API 请求/响应消息方面也更加结构化和标准化。

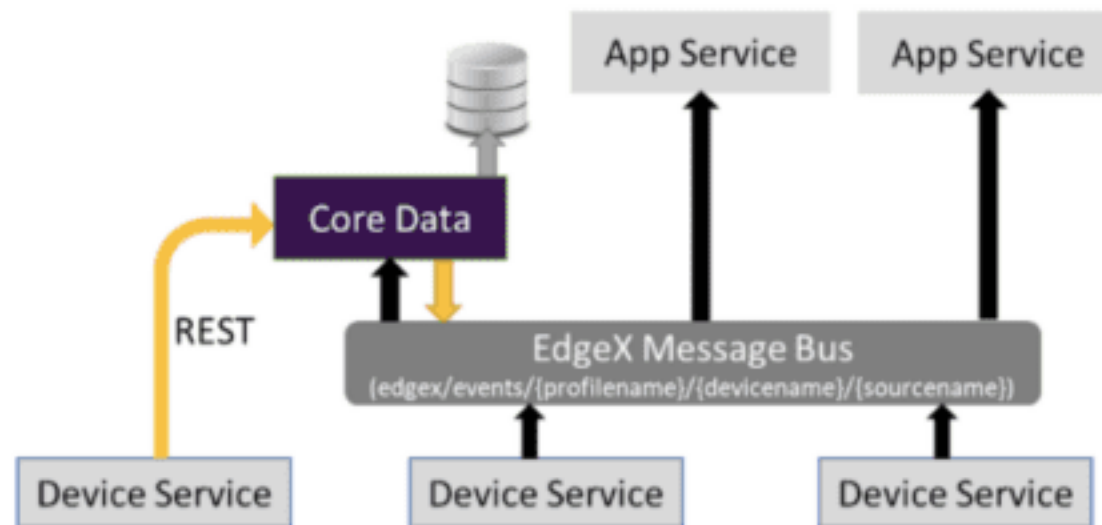
Request	Response
<pre>POST to /deviceservice Headers X-Correlation-ID 14a42ea6-c394-41c3-8bcd-a29b9f5e6835 Body [{ "requestId": "e6e8a2f4-eb14-4649-9e2b-175247911369", "apiVersion": "v2", "service": { "name": "home thermostat Device Service", "description": "manage homethermostats", "adminState": "UNLOCKED", "operatingState": "UP", "labels": ["hvac", "thermostat"], "baseAddress": "http://home-device-service:49990" }]</pre>	<pre>Headers X-Correlation-ID 14a42ea6-c394-41c3-8bcd-a29b9f5e6835 Body [{ "apiVersion": "v2", "requestId": " e6e8a2f4-eb14-4649-9e2b-175247911369 ", "statusCode": 201, "id": "6f52dc3c-5548-4142-baa6-052ac4bece93", "message": "" }]</pre>

消息总线和可选的核心数据

• EdgeX 1.x



• EdgeX 2.0



- 我们创建了一个模块和模式，为所有服务从秘密存储中检索秘密提供了一种通用方法（EdgeX 默认使用 Vault 作为秘密存储）。我们将此功能称为“所有人的秘密提供者”。
- EdgeX 使用 Consul 进行服务注册和配置。在此版本中，API 网关用于允许访问 Consul 的 API。EdgeX 1.0 中的 Consul API 被拒绝访问，这使得 Consul 的更改变得困难。
- Consul 现在从其访问控制列表系统起的方式启动——提供更好的服务认证/授权和更好的保护它存储的键/值配置。
- 在 Docker 容器中以 root 身份运行所需的进程和服务更少。
- API 网关 (Kong) 设置已得到改进和简化。
- EdgeX 现在可以防止 Redis（EdgeX 的持久存储）在不安全模式下运行。

简化的设备配置文件

EdgeX 1 Device Profile	EdgeX 2 Device Profile
<pre>name: "Simple-Device" manufacturer: "Simple Corp." model: "SP-01" labels: - "modbus" description: "Example of Simple Device" deviceResources: - name: "SwitchButton" description: "Switch On/Off." properties: value: {type: "Bool", readWrite: "RW", defaultValue: "true"} units: {type: "String", readWrite: "R", defaultValue: "On/Off"} deviceCommands: - name: "Switch" get: - {operation: "get", deviceResource: "SwitchButton"} set: - {operation: "set", deviceResource: "SwitchButton", parameter: "false"} coreCommands: - name: "Switch" get: path: "/api/v1/device/{deviceId}/SwitchButton" responses: - code: "200" description: "" expectedValues: ["SwitchButton"] - code: "503" description: "service unavailable" expectedValues: [] put: path: "/api/v1/device/{deviceId}/SwitchButton" parameterNames: ["SwitchButton"] responses: - code: "200" description: "" - code: "503" description: "service unavailable" expectedValues: []</pre>	<pre>name: "Simple-Device" manufacturer: "Simple Corp." model: "SP-01" labels: - "modbus" description: "Example of Simple Device" deviceResources: - name: "SwitchButton" coreCommand: true description: "Switch On/Off." properties: valueType: "Bool" readWrite: "RW" defaultValue: "true" deviceCommands: - name: "Switch" - rewWrite: "RW" - {deviceResource: "SwitchButton", parameter: "false"}</pre>

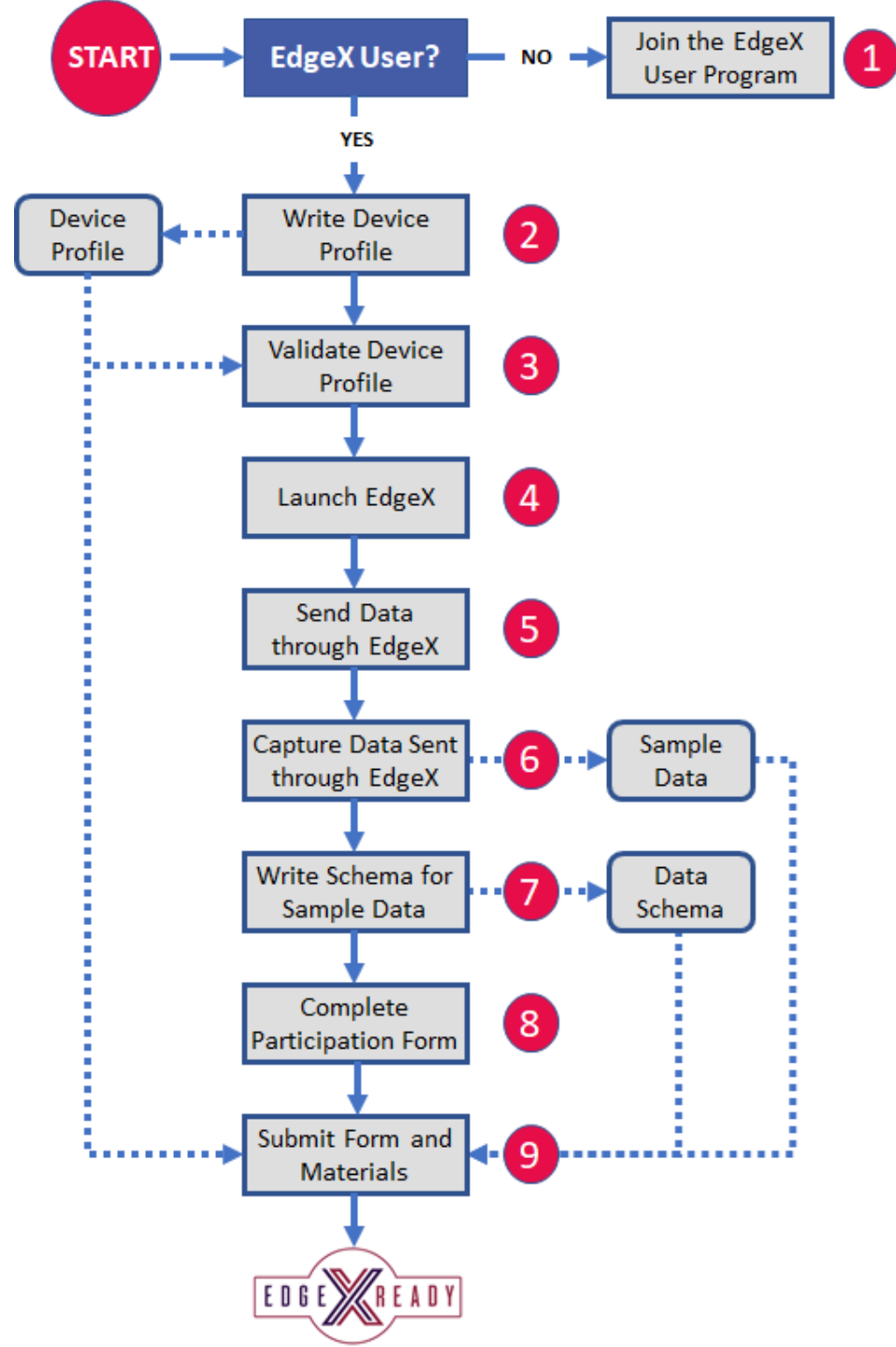
- 受限应用协议 (CoAP) 是一种用于资源受限节点和网络的 Web 传输协议。
- GPIO（或通用引脚输入/输出）是用于将微控制器连接到其他电子设备的标准接口。鉴于其在设备上的可用性，它也很受 Raspberry Pi 采用者的欢迎。
- 低级阅读器协议 (LLRP) 是许多 RFID 阅读器的标准化网络接口。
- 通用异步接收器/发送器 (UART) 是串行数据通信，用于调制解调器，可在 USB 到 UART 桥接器中与 USB 一起使用。

应用程序服务用于将 **EdgeX 数据**从边缘导出到企业和云系统和应用程序。应用服务过滤、准备和转换传感器数据以方便使用。

- 导出前按设备配置文件名称和设备资源名称过滤传感器读数的新功能
- **允许指定多个 HTTP 端点**以供一个应用程序服务导出
- 订阅多个消息总线（通过订阅启用多个过滤器）
- **提供了一个新模板，以便更轻松/更快地**创建自定义应用程序服务

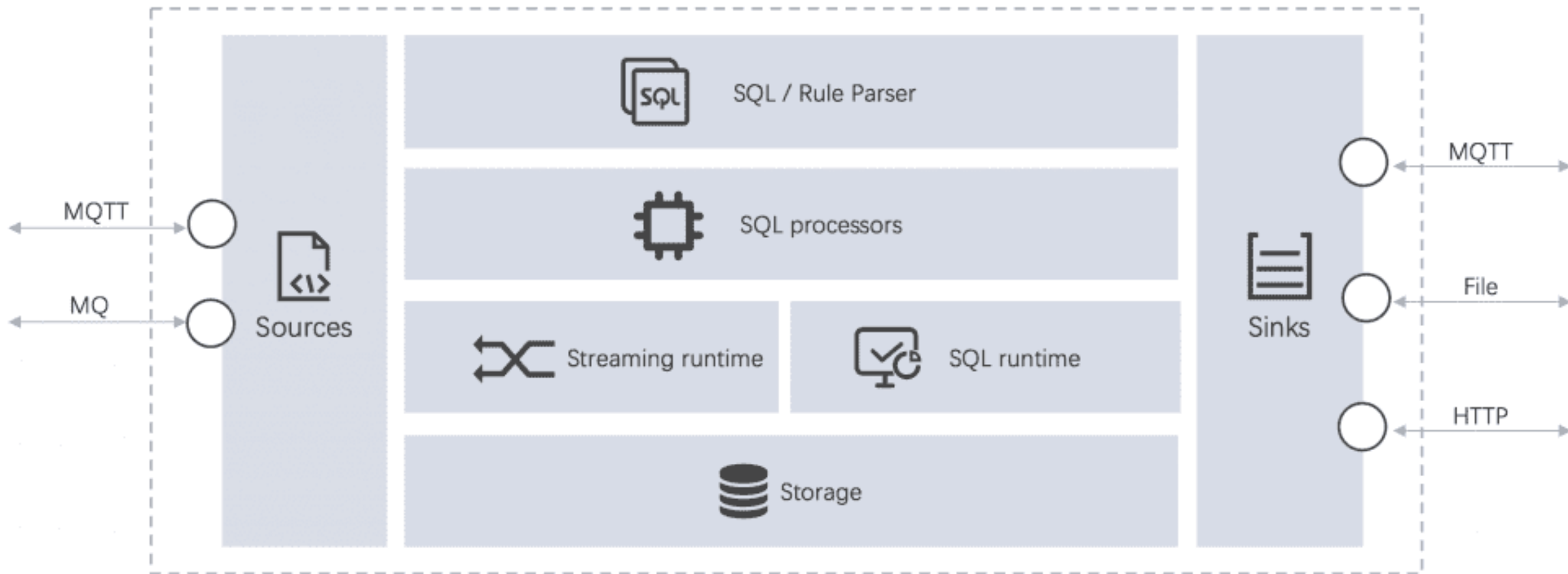
EdgeX 就绪

EdgeX 就绪是一项旨在突出展示能够与 EdgeX 合作的成员组织的计划。EdgeX 就绪计划随 EdgeX 2.0 版本一起推出，旨在提高拥有 EdgeX 专业知识的用户及其组织的意识。



EMQ将eKuiper贡献到LF Edge基金会

来自中国开源社区向LF Edge贡献的第二个项目



EdgeX Foundry支持Kubernetes部署和生命周期管理

由VMware 中国牵头贡献

EDGE X FOUNDRY™

特殊扩展

OpenYurt

Extending your native Kubernetes to Edge



KubeEdge



SuperEdge

轻量发行版



MicroK8s



Custom Resources



- DevOps 团队清理了 EdgeX Docker 镜像名称（删除了不必要的镜像前缀/后缀名称），并在 Docker Hub **中的所有**镜像上提供了描述和适当的标签——**帮助采用者更快速、更轻松**地找到他们需要的 EdgeX Docker 镜像。
- 测试/QA 团队添加了新的集成测试、性能测试和“冒烟”测试，以便在第三方库或组件（例如 Consul）的**新版本**出现问题时为项目提供早期指标——**允许社区解决不兼容或问题更快**。
- 营销团队改造和升级了我们的网站，并创建了我们网站的中文版cn.edgexfoundry.org，以支持我们不断增长的中国社区采用者。
- **改进**了设备服务贡献审核流程。
- **在代码**贡献过程中结合使用常规提交。
- **启动**了一个程序来审查 **第三方依赖项**；确保第三方依赖有足够的许可，质量和开发活动，以支持我们的项目。
- **帮助启动**开放零售参考架构项目，以促进为零售采用者开发 LF Edge **参考架构**。
- **与**Digital Twin Consortium**和**AgStack（**一个新的** LF 项目）签订了联络协议，以了解如何将 EdgeX **更好地集成到数字孪生系统中**并帮助促进农业生态系统中的解决方案。

小版本 2.0.1和雅加达版本 v2.1.0

小版本2.0.1

- 完成在爱尔兰版本 v2.0.0 上未完成的工作 (API v2.0)
- fix bug

雅加达版本 v2.1.0

- 只解bug的版本
- 稳定版本
- 第一个长期支持 (LTS) 版本：5年
- 替换ZMQ

Open Source AceCon

2021 智能云边开源峰会

AI x Cloud Native x Edge Computing

人工智能 × 云原生 × 边缘计算

Thank You