

灵狐技术 Fox-Edge 软件 技术白皮书

文档版本 V 1.0

发布日期 2024-03-29



版权所有 © 广州灵狐技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，不得以任何形式传播。

商标声明



和其他灵狐商标均为广州灵狐技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您使用的产品、服务或特性等应受广州灵狐公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，广州灵狐公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

广州灵狐技术有限公司

地址：广州市天河区华景新城翰景路 197 号

网址：<http://www.fox-tech.cn>

前言




概述

本文档详细地描述了 Fox-Edge 软件的使用场景、系统架构、主要功能和典型组网，让用户对 Fox-Edge 有一个全面的了解。

读者对象

本文档主要适用于以下工程师：

- 营销工程师
- 技术支持工程师
- 维护工程师

 警告	表示如不可避免则可能导致死亡或严重伤害的具有严重风险的危害。
 须知	用于传递设备或环境安全警示信息。如不可避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。“须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修改记录:

版本	发布日期	说明
1	2024-03-29	第一次正式发布

1 产品简介	7
2 系统架构	9
2.1 架构概述	9
2.1.1 部署架构	9
2.1.2 软件架构	10
2.1.3 系统上下文对接方式	11
2.2 关键技术特性	12
2.3 部署方式	14
2.3.1 部署形态	14
2.3.2 部署位置	14
2.3.3 部署周期	14
2.4 升级方式	15
2.5 管理规模和推荐配置	15
2.5.1 入门配置	15
2.5.2 推荐配置	15
2.5.3 高级配置	16
3 产品功能	17
3.1 概述	17
3.2 组件仓库	17
3.2.1 服务模块	17
3.2.2 静态解码	18
3.2.3 动态解码	18
3.2.4 文件模板	19
3.2.5 前端模块	19
3.3 本地组件	19
3.3.1 服务模块	19
3.3.2 静态解码	20
3.3.3 动态解码	20
3.3.4 文件模板	20
3.4 系统管理	20
3.4.1 系统参数	20
3.4.2 服务模块	20
3.4.3 前台日志	21
3.4.4 用户管理	21
3.5 通道管理	21
3.5.1 通道列表	21
3.6 设备管理	22
3.6.1 设备列表	22
3.6.2 设备数值	22
3.6.3 设备映射	23
3.6.4 扩展信息	23
3.6.5 通信状态	23

3.7 任务管理.....	24
3.7.1 通道操作任务.....	24
3.7.2 设备操作任务.....	24
3.7.3 设备监控任务.....	24
3.8 周期管理.....	25
3.8.1 周期任务.....	25
3.8.2 周期记录.....	25
3.9 记录管理.....	25
3.8.1 设备记录.....	25
3.8.2 操作记录.....	26
3.8.3 历史记录.....	26
3.9 链路管理.....	26
3.10 模型管理.....	26
3.11 北向连接.....	26
3.11.1 第三方云平台 ThingsBoard.....	26
3.11.2 第三方云平台 ThingsPanel.....	27
3.11.3 第三方云平台 HuaWei.....	27
3.11.4 第三方云平台 WhZktl.....	28
3.11.5 灵狐云平台 Fox-Cloud.....	28
4 典型组网.....	29
4.1 上位机场景.....	29
4.2 智能网关场景.....	30

1 产品简介

Fox-Edge 为 x86 系列或 ARM 系列的服务器、边缘设备上，统一运行的物联网边缘计算软件，支撑公有云、企业客户对现场智能设备快速接入和高效的运维管理。

该软件实现在通用的 x86 系列或 ARM 系列设备上可部署后，提供了对现场侧的各类设备厂家生产的智能设备进行快速接入能力，提供设备纳管、设备配置、固件升级、设备监控等全生命周期的管理能力，有效帮助物联网生产企业、集成商、甲方用户提高运维效率、降低运维成本。

Fox-Edge 可广泛应用于能源、建筑、三表、基站、交通、船舶、校园、农业、林业、水利等领域的物联网项目，面向政府和企业客户的面向数字化运营场景下部署。

同时提供了企业自身开发的私有化模块、服务（JAVA、Python）的部署和管理能力，便于用户自行与 Fox-Edge 一起组成更为丰富的物联网边缘计算的能力。

图 1-1 Fox-Edge 应用场景。



产品特点：

- **通用化平台：**运行在通用的 x86 工控机或是 ARM 专用设备环境，不用担心专用设备对企业的绑定。
- **通用化系统：**运行的 LINUX 操作系统、数据库、运行时环境，均为通用的开源软件，这些软件系统均有对应的国产替代产品。
- **积木式架构：**用户可以根据自己的需要，可以从中央仓库中自行挑选组件和服务，搭建成适合自己项目的边缘计算系统。
- **开放式架构：**用户可以自行开发私有化服务，部署在同一硬件平台之上，与 Fox-Edge 构成更丰富的边缘计算应用。
- **共享的资源：**中央仓库可以为用户们互相分享自己开发的解码器组件和各类应用服务
- **方便的 ODM：**企业可以将 Fox-Edge 部署在通用工控机上后，以企业自己的产品形式对外销售。

2 系统架构

2.1 架构概述

2.1.1 部署架构

Fox-Edge 在物联网的【端-边-云】（如图 2-1）中，作为边缘计算的一个环节。

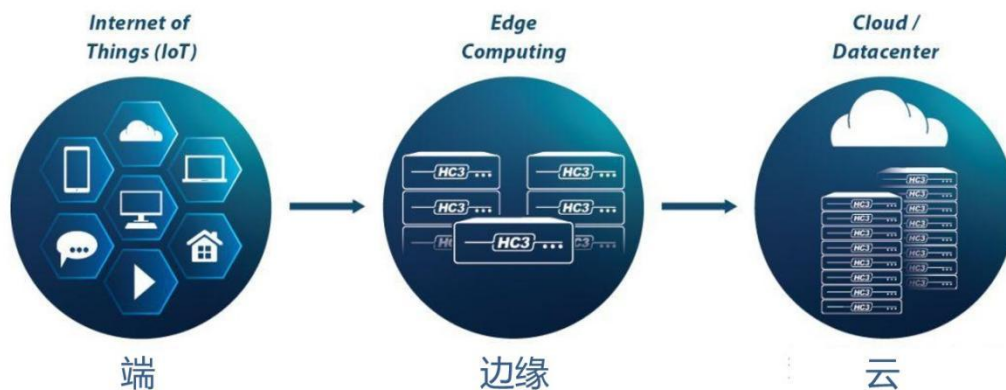


图 2-1

它通常与智能设备一起，被部署在用户现场（如图 2-2），对现场的设备进行就近管理。

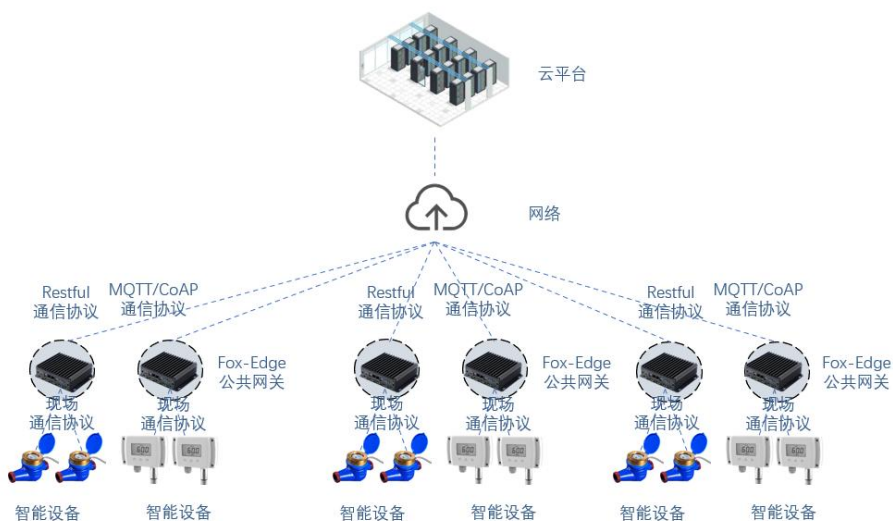
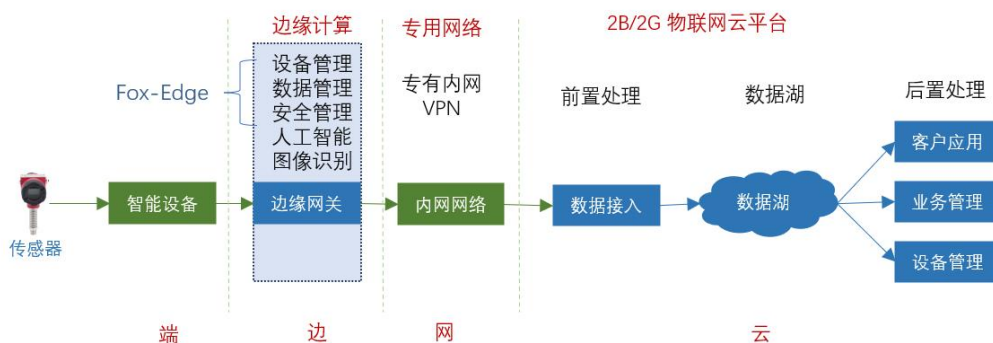


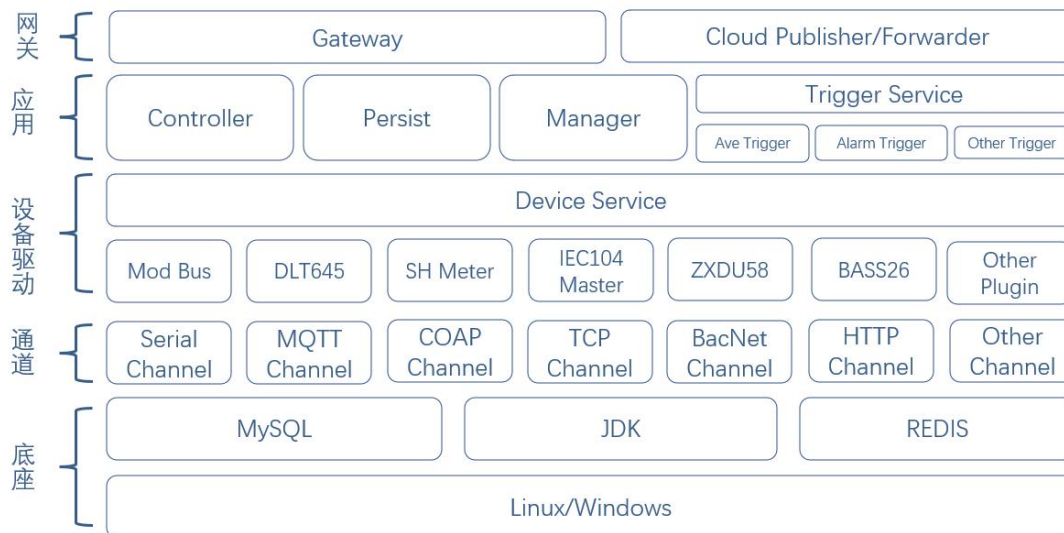
图 2-2

Fox-Edge 充当在现场的边缘端，最终通过现场总线，对传智能设备进行连接管理，然后通过网络与物联网云平台进行对接，构成一个完整的物联网方案（如图 2-3）。



2.1.2 软件架构

Fox-Edge 作为部署在 x86 或 ARM 通用硬件平台上的软件系统，采用的是微服务架构，方便根据项目的需要，进行积木式选择部署具体的部件，从而实现按需部署的要求。



架构特点：

- **通用化平台：**运行在通用的 x86 工控机或是 ARM 专用设备环境，不用担心专用设备对企业的绑定。
- **通用化系统：**运行的 LINUX 操作系统、数据库、运行时环境，均为通用的开源软件，这些软件系统均有对应的国产替代产品。

- **积木式架构:** 用户可以根据自己的需要, 可以从中央仓库中自行挑选组件和服务, 搭建成适合自己项目的边缘计算系统。
- **开放式架构:** 用户可以自行开发私有化服务, 部署在同一硬件平台之上, 与 Fox-Edge 构成更丰富的边缘计算应用。
- **共享的资源:** 中央仓库可以为用户们互相分享自己开发的解码器组件和各类应用服务

2.1.3 系统上下文对接方式

Fox-Edge 可以通过物理连接方式, 在南向跟各自智能设备进行建立连接, 在北向跟各云平台进行对接。

- 南向有线物理连接, 可以通过自带的接口或者插入各类通信模块后, 对接有线的 RS232、RS485、M-BUS、CAN、Ethernet、Profibus 智能设备。
- 南向无线物理连接, 可以通过自带的接口或者插入各类通信模块后, 对接无线的 WI-FI、LORA、NB-IOT、4G/5G、ZIGBB、TPUNB、RF、UWB、NFC 智能设备。
- 北向物理连接, 可以通过自带的接口或者插入各类通信模块后, 通过 Ethernet、WI-FI、4G/5G, 与物联网云平台对接。

Fox-Edge 跟南向的智能设备建立物理连接的基础上, 可以在部署指定的会话服务和解码器的方式, 跟南向的智能设备进行会话连接。

- 目前已经提供的连接会话服务, 包括 SerialPort、Serialport-Analyzer、TCP-Server、TCP-Client、UDP-Server、UDP-Client、OPC-UA、MQTT-Client、COAP-Client、HTTP-Client、SNMP-Client、S7PLC-Client、Hikvision-Fire, 更多的会话服务, 可在中央仓库中查看和获得。
- 目前已经提供的解码器, 包括 MODBUS、DLT645、CJT188、IEC104、OPC-UA、S7PLC、HJ-212、DAHUA-FIRE、HIKVISION-FIRE、HAIER-YCJ-A002、TCL-AIR、HAIWU-AIR、CET-UPS、ZS-SHT30-1T-1H、OMRON-FINS、MITSUBISHI-PLC-FX、RKCKTH-RS-M88, 更多的解码器, 可在中央仓库中查看和获得。

Fox-Edge 跟北向的云平台建立物理连接的基础上, 可以在部署指定的北向接入服务, 跟北向的各家云平台进行会话连接。

- WEB UI 接口, 这是 Fox-Edge 自带的前端页面 HTTP 接口, 它在作为 Fox-Edge 管理界面的时候, 同时也可以为第三方北向平台提供 HTTP/Restful 管理接口。
- IoT-Fox-Cloud 服务, 这是对灵狐的 Fox-Cloud 云平台的服务, 在 Fox-Edge 上部署它后, 可以把 Fox-Edge 上的数据推送给 Fox-Cloud 云平台
- IoT-ThingsBoard 服务, 这些对接第三方 ThingsBoard 云平台的北向服务, 部署它之后, 以把 Fox-Edge 上的数据推送给 ThingsBoard 云平台。
- IoT-ThingsPanel 服务, 这些对接第三方 ThingsPanel 云平台的北向服务, 部署它之后, 以把 Fox-Edge 上的数据推送给 ThingsPanel 云平台。

- IoT-HuaWei 服务，这些对接华为物联网云平台的北向服务，部署它之后，以把 Fox-Edge 上的数据推送给 HuaWei 云平台。
- IoT-WHZKTL 服务，这些对接中科图灵物联网云平台的北向服务，部署它之后，以把 Fox-Edge 上的数据推送给中科图灵物联网云平台。

Fox-Edge 可以通过 REDIS 的缓存和消息队列方式，为部署在同一个硬件平台之上的企业私有化服务，提供 API 接口。

Fox-Edge 同时也对符合安装包规范的企业私有服务，进行安装管理。当企业使用开发语言 JAVA、PYTHON 进行私有服务的开发后，按指定的 TAR 安装包规范进行打包后，可被 Fox-Edge 进行统一的安装、管理、卸载。

2.2 关键技术特性

通用化的硬件平台

- 可以部署在 x86 系列的服务器、工控机、软路由、虚拟机，这类通用的硬件平台之上。
- 可以部署在 ARM 系列的软路由，这类通用的硬件平台之上。
- 也可以部署在物联网设备生产厂商，自身开发的 ARM 系列私有化硬件平台之上。

通用化的软件平台

- 可以部署在通用的 LINUX 操作系统之上。
- 所依赖的 JDK、MySQL/MariaDB、Redis，均为被广泛应用的通用基础服务。
- 支持通用的 JAVA、PYTHON 开发语言。

丰富的硬件接入方式

- 支持通过硬件平台本身自带的串口、以太网接口，与现场智能设备进行对接。
- 支持通过插入市场上外购的 USB/RS-232、USB/RS-4852、USB/LORA、USB/CAN、USB/WIFI、USB/NB-IOT、USB/4G、USB/5G、USB/ZIGBEE 外置模块，与现场智能设备进行对接。

丰富的设备接入能力

- Fox-Edge 当前已经提供了市面上常用的 MODBUS、DLT645、CJT188、IEC104、OPC-UA、S7PLC、HJ-212 设备接入能力，并且在前面的项目实践中，为用户开发了众多的私有化设备的接入服务、解码器、配置模板。
- Fox-Edge 支持用户通信 JAVA、JavaScript 两种开发语言，快速的编写用户自己的私有通信协议的解码器，并在部署后，自行接入这些私有协议的通信设备。

丰富的设备交互方式

- Fox-Edge 通过各自会话服务，支持主从半双工、全双工、单工上报、单工下行的会话模式。
- Fox-Edge 通过各自会话服务，支持线下主动接入类型设备的自动注册、认证。

强大的设备管理能力

Fox-Edge 能够自动识别安装服务和解码器，并对这些会话服务和设备，进行统一的管理。

积木式的软件架构

Fox-Edge 系统采用微服务架构设计，让用户根据具体项目的需要，选择部署必要的软件服务和模块。像积木一样，为自己的项目搭建对应的边缘计算系统。同样，客户自己开发的私有化服务，也可以作为积木之一，共同部署在边缘计算的 LINUX 平台之上。

开放性的系统架构

Fox-Edge 采用的是开放性的系统架构，支持用户基于 JAVA、PYTHON 开发符合 Fox-Edge 规范的南向通道服务和设备解码器，北向云平台对接服务，以及自身项目特色需要的其他私有服务。

强大的北向接入能力

客户可以为自己的物联网云平台，开发私有接入服务后，将其直接部署到 Fox-Edge 系统之后，那么就可以完成对接自己的物联网云平台和 Fox-Edge 的对接。

强大的边缘计算能力

边缘计算，意味着可以在边缘端的通用平台之上，不但能够部署 Fox-Edge，还可以部署用户的其他各自应用，比如第三方的 NOD-RED 边缘计算，比如用户自己的 AI 应用、大数据应用、视频应用，共同构成一个更丰富的边缘计算应用系统，增强了用户自身的产品竞争力。

可信任的开源技术

为了取得用户的使用信任，Fox-Edge 对自己的代码全面开源，并进行免费授权。用户不用当心遇到技术供应问题。

极低的学习和开发成本

JAVA 拥有广泛的开发者群体和生态环境，同时它是在企业引用中被使用最多的开发语言。

Fox-Edge 相应的采用 JAVA 开发，那么使得上述本身拥有 JAVA 开发人员和团队用户，只要一个 JAVA 开发团队，即可直接学习、使用、开发、维护 Fox-Edge 产品。

高度稳定的系统

JAVA 的异常机制，使得开发者在开发中出现 BUG 的时候，会进行兜底。JAVA 开发的软件具有稳定性，在过去二十多年的大量企业和互联网项目，已经得到反复的实践验证。

这是 Fox-Edge 选择 JAVA 作为开发语言的重要依据，Fox-Edge 的稳定性同样在用户的诸多项目之中，得到了验证。

高性能的系统

JAVA 的 JIT (即时编译器) 技术，使得 JAVA 本身拥有非常高的性能。同时 Fox-Edge 采用的积木式架构，减少了传统 JAVA 开发的极大内存浪费，能够高效的利用硬件的性能。

极限测试数据：在阿里云的双核 CPU/4G 内存的模拟测试环境下，部署 20 多个 JAVA 服务，可以进行 1000 台模拟设备，10 万个采样点数据，10 秒周期的并发处理。

2.3 部署方式

Fox-Edge 可以根据业务场景，灵活的支持多种部署形态，以适应企业不同的运维管理需求。

2.3.1 部署形态

- 可部署在 x86/ARM 的物理设备或者服务器之中
- 可部署在 x86 的虚拟机镜像之中
- 可部署在 x86 的 Docker 镜像之中

2.3.2 部署位置

- 可部署在现场的 x86/ARM 的物理设备之中，对现场的智能设备就近管理
- 可部署在云端的服务器之中，对现场的智能设备远程管理

2.3.3 部署周期

- 可在出厂或出库前，在边缘设备上预部署好 Fox-Edge 的各服务组件。工程人

员携带边缘设备到现场后，完成设备的安装、配置和测试。

- 可在出厂或出库前，在边缘设备上预部署好 Fox-Edge 的核心组件。工程人员携带边缘设备到现场后，根据现场需要进行 Fox-Edge 的其他服务和组件的安装，然后进行配置和测试。
- 可在出厂或出库前，在边缘设备预配置 Linux 远程登录配置。工程人员携带边缘设备到现场后，完成边缘设备的安装、网络连通后离开。然后，再由管理中心的网管人员，进行 Fox-Edge 的远程安装、部署、配置、测试。

2.4 升级方式

Fox-Edge 边缘计算，可以连接数据中心的部件仓库，进行服务和部件的在线升级更新，这是推荐的日常升级方案。

也可以通过 Linux 远程登录的方式，进行初始化安装或者刷机升级。

2.5 管理规模和推荐配置

Fox-Edge 是面向边缘计算而设计的物联网平台。它既可以被使用在复杂的边缘计算场景，也可以作为传统的功能型智能网关使用。根据客户的具体业务场景的需求，相应的衍生出了很多部署方案。

2.5.1 入门配置

有很多小型的物联网项目，它的每个管辖片区中，智能设备规模非常小，往往只有几个到几十个智能设备，业务场景也非常单一，业务需要也仅仅为了获得现场数据，用户只会安装几个基础服务。

最低配置：2 核 CPU，4G 内存，32G 存储空间

业务场景：电信运营商的基站管理，中小企的办公室智能化管理

2.5.2 推荐配置

有很多的物联网项目，它的每个管辖片区中，智能设备规模开始上规模了，往往有上百个智能设备，采样点达到上万个，业务场景也开发多样化，业务需要不仅仅再是为了获得现场数据，而且用户会安装更多的应用服务。

最低配置：2 核 CPU，8G 内存，128G 存储空间

业务场景：水电煤气的三表管理，写字楼的智能建筑管理

2.5.3 高级配置

在现在的很多边缘计算项目，它们不仅仅是使用 Fox-Edge 进行现场设备的规模化
管理，而且还会在边缘计算平台上，部署用户自己的其他边缘计算应用，比如 AI
应用、视频应用、数据分析应用，Fox-Edge 仅仅是整个项目的应用场景之一。

配置通常要结合整个边缘计算的各应用，进行统一设计和规划。

最低配置：4 核 CPU，16G 内存，256G 存储空间

业务场景：市政的交通、环境项目

级别	资源规格	管理规模		备注
		设备	采样点	
入门配置	CPU: x86 2 核 内存: 4G 磁盘: 32G	≤128 台	12800 个	主要作为： 1、开发环境 2、测试环境 3、小型的物联网项目
	CPU: ARM 4 核 内存: 4G 磁盘: 32G	≤128 台	12800 个	
推荐配置	CPU: x86 2 核 内存: 8G 磁盘: 128G	≤256 台	25600 个	默认的生产环境配置
	CPU: ARM 4 核 内存: 8G 磁盘: 128G	≤256 台	25600 个	
高级配置	CPU: x86 4 核 内存: 16G 磁盘: 256G	≤512 台	51200 个	规模化的物联网项目
	CPU: ARM 4 核 内存: 16G 磁盘: 256G	≤512 台	51200 个	

3 产品功能

3.1 概述

Fox-Edge 是广州灵狐技术有限公司针对企业、政府的物联网项目，研发的一款面向数字化运营领域的边缘计算产品，具有以下特点：

- 灵活：按需的搭建能力
- 通用性：通用的软硬件平台，通用的配置、操作、管理方式
- 可信任：符合信创标准的技术，全开源的技术
- 开放性：全开放的平台
- 高性能：10 秒周期内，完成 1 千台设备，10 万采样点的单机并发能力

Fox-Edge 提供了组件管理能力、通道管理能力、设备管理能力、任务管理能力、服务管理能力、升级管理能力，以及其他常用的系统管理能力。

3.2 组件仓库

Fox-Edge 作为一种边缘计算产品，需要能够安装、配置、管理 Fox-Edge 自己的服务，也能安装、管理客户同机部署的私有化服务。

Fox-Edge 采用的是积木式架构，用户根据自己的项目需求，选择安装具体的服务、解码器、模板等组件，可以非常高效利用自己有限的硬件资源。并且，可以通过安装各种新的部件，来不断扩展 Fox-Edge 的能力。

初始安装的 Fox-Edge，只有两个核心模块，它的各种功能是通过 HTTP 连接到 Fox-Cloud 的中央组件仓库，根据具体的项目，进行按需各种组件的安装后，来实现的。

3.2.1 服务模块

服务模块，可以看到中央仓库提供的各种服务组件，用户可以按需选择下载和安装相应的服务组件。这些服务组件在选择下载和安装后，会被管理服务，以 Linux 进程的方式启动和管理起来。

需要安装的基础服务，至少包括如下：

- 设备服务：device-service 服务，这是系统服务，提供了南向智能设备的协议编码/解码的框架能力
- 通道服务：channel-xxx 规范命名的服务，至少包括一个通道服务，它们提供了对应的南向智能设备的接入能力
- 控制器服务：controller-service 服务，这是系统自带的数据采集任务，它提供了对设备进行周期任务的采集能力。用户也可通过开发更符合自己需要的私有控制器服务，来取代它。
- 持久化服务：persist-service 服务，这是系统服务，它提供了对采集到的各种数据，进行持久化的数据库保存能力。

在服务安装完成以后，可以在系统参数管理功能中，为各种服务进行工作参数配置。

在后续运维过程中，也可以通过在该功能中，看到是否有最新的服务组件，可以进行应用升级。

3.2.2 静态解码

静态解码，可以看到中央仓库提供的各种解码器组件，用户可以按需选择下载和安装相应的解码器组件，这些解码器是以 JAVA 的 JAR 包形式存在。

这些解码器组件在选择下载和安装后，可被设备服务冷加载，并在重启设备服务后，成为设备服务中运行代码，进行南向设备的通信协议编码/解码。

需要安装的基础模块，至少包括如下：

- 注解模块：fox-edge-server-protocol-anno
- 工具库模块：fox-edge-server-protocol-utils
- 核心模块：fox-edge-server-protocol-core。

各解码器模块，存在依赖关系，其他解码器均需要依赖上述模块。更多的依赖模块，可以在解码器安装到本地之后，在依赖文件中查看到，发现缺失的话，可以补充安装相应的 JAR 模块。

在后续运维过程中，也可以通过在该功能中，看到是否有最新的服务组件，可以进行应用升级。

3.2.3 动态解码

动态解码，可以看到中央仓库提供的各种解码器脚本，用户可以按需选择下载和安装相应的解码器脚本，这些解码器是以 JavaScript 脚本的形式存在。

这些解码器组件在选择下载和安装后，可被设备服务热加载，通过 JAVA 的 JVM 中的 JavaScript 引擎直接使用，进行南向设备的通信协议编码/解码。

3.2.4 文件模板

文件模板，可以看到中央仓库提供的 MODBUS、IEC104、OPC-UA、S7PLC、OMRON-FINS、Mitsubishi、SNMP 等框架类的协议，对应的协议 CSV 文件配置模板。

这些文件模板在选择下载和安装后，可被设备服务热加载，通过对应的静态解码器，进行南向设备的通信协议编码/解码。

说明：

上述通信协议均为框架类协议，需要在静态解码之中，提前安装相应的 JAR 解码器引擎。

在后续运维过程中，也可以通过在该功能中，看到是否有最新的服务组件，可以进行应用升级。

3.2.5 前端模块

前端模块，可以看到中央仓库提供的前端 WEB UI 包，用户可以按需选择下载和安装相应的 WEB UI 包，进行界面的升级。

在后续运维过程中，也可以通过在该功能中，看到是否有最新的服务组件，可以进行应用升级。

3.3 本地组件

Fox-Edge 作为一种边缘计算产品，需要能够安装、配置、管理 Fox-Edge 自己的服务，也能安装、管理客户同机部署的私有化服务。

Fox-Edge 采用的是积木式架构，用户根据自己的项目需求，选择安装具体的服务、解码器、模板等组件，可以非常高效利用自己有限的硬件资源。并且，可以通过安装各种新的部件，来不断扩展 Fox-Edge 的能力。

初始安装的 Fox-Edge，只有两个核心模块，它的各种功能是通过 HTTP 连接到 Fox-Cloud 的中央组件仓库，根据具体的项目，进行按需各种组件的安装后，来实现的。

3.3.3 服务模块

服务模块，可以看到已经安装到 Fox-Edge 本地的各种服务组件。这些组件会涉及到一些必要的程序启动参数配置，可在此进行配置。

3.3.2 静态解码

静态解码，可以看到已经安装到 Fox-Edge 本地的各种 JAR 解码器组件。这些组件涉及到对其他 JAR 包的依赖，以及包含哪些设备操作方法，是否选择加载，均可以在此查看。

3.3.3 动态解码

动态解码，可以看到已经安装到 Fox-Edge 本地的各种 JavaScript 解码器脚本。这些组件包含哪些设备操作方法和具体的脚本内容，可以在此查看。

3.3.4 文件模板

文件模板，可以看到已经安装到 Fox-Edge 本地的各种 CSV 解码器文件模板。这些组件包含哪些具体的 CSV 文件配置内容，可以在此查看。

说明：

这些文件模板，均依赖于特定的框架类 JAR 解码器的预先安装，在使用这些 CSV 模板之前，请提前在静态解码中，安装相应的 JAR 解码器引擎。

3.4 系统管理

Fox-Edge 作为一种边缘计算产品，需要能够安装、配置、管理 Fox-Edge 自己的服务，也能安装、管理客户同机部署的私有化服务。

它们必然会涉及到统一的参数配置、服务管理、日志信息的系统管理工作。

3.4.1 系统参数

Fox-Edge 下的各个服务模块，会涉及到各种工作参数配置，Fox-Edge 提供了统一的 JSON 风格的参数配置管理能力。可以在 Fox-Edge 通过为这些服务，单独配置它们规定的 JSON 配置参数。

系统参数，提供了用户对这些服务的系统配置能力。

3.4.2 服务模块

Fox-Edge 下的各个服务模块，作为 LINUX 下的服务进程，涉及到进程的启动、停止、重启、卸载等管理。同时还涉及到它们的工作状态管理，包括 PID、端口、内存占用、CPU 占用、启动时间。

服务模块，提供了用户对这些服务的进程管理能力。

3.4.3 前台日志

Fox-Edge 下的各个服务模块，在运行的过程之中，会产生一些前台日志信息，保存到 REDIS 中，然后通过前端呈现给运维人员，方便使用者排查问题。

3.4.4 用户管理

Fox-Edge 具有登录管理能力，可以通过用户管理，修改初始用户的账号和密码。

3.5 通道管理

Fox-Edge 在连接南向的各种各样的智能设备的时候，这些智能设备的物理连接方式，可能是 RS232、RS485、TCP、LORA、M-BUS、WIFI、4G、5G、NB-IOT 等各种物理连接，那么可以使用边缘设备自带的串口、以太网口、WIFI 直接建立物理连接，或者通过外置的 USB-XXX 模块适配器，转换为逻辑串口，然后进行连接。

这些现场设备，在建立物理连接之后，还需要跟它们进行会话，才能进行真正的通信。而各 Channel-xxx 类型的通道服务，即为跟这些设备建立连接会话的服务。当安装 Channel-xxx 通道服务后，Fox-Edge 管理界面，会出现对应的 xxx 通道管理。

例如：

现场有个 TCP-SERVER 的设备，那么在 Fox-Edge 上安装好 channel-tcp-client 服务后，即可通过以太网或者 WIFI 跟该 TCP-SERVER 设备进行连接，并展开会话。

3.5.1 通道列表

当安装 channel-xxx 的服务组件后，在现场如果有对应通信接口的智能设备，那么可以添加一个对应的通道会话，并进行对应的参数配置，那么 Fox-Edge 就可以跟该设备建立会话连接。

例如：现场有一个 RS485 接口的 DLT645 智能电表，用户想通过 Fox-Edge 对它进行管理。用户手头上有一个自带 USB 的工控机，以及一个 USB-RS485 模块，并与该智能电表通过电缆进行了连接。

步骤：

- 将 USB-RS485 模块插入工控机的 USB 口
- 登录 LINUX 后，使用命令行 `ls -al /dev/ttyU*`，查看该模块识别为 `/dev/ttyUSB0` 文件
- 在 Fox-Edge 安装 channel-serial 服务部件，并配置为开机启动

- 此时进行新增通道时，会发现已经支持了 serial 类型的通道
- 新增一个 serial 类型的通道，并配上 JSON 格式的串口参数，其中 serialName 配置为上面查询到的 ttyUSB0
- 在任务管理-通道操作任务中，给该通道发送一串 DLT645 的通信报文，可以看到 DLT645 设备会返回一段响应报文。

3.6 设备管理

Fox-Edge 在连接南向的各种各样的智能设备的时候，这些智能设备的在建立物理连接和会话连接之后，在安装对应的通信协议解码器之后，可以对这些智能设备进行各种远程查询和控制操作。

例如：

现场有个支持 RS485 的 DLT645 的智能电表设备，那么 Fox-Edge 上安装好 channel-serial 服务，建立好通道连接后，即再安装一个 DLT645 解码器，对该设备进行通信编码/解码的远程操作。

3.6.1 设备列表

当安装 xxx 的解码器模块后，就可以支持对 xxx 类型的设备进行添加、配置和其他管理操作了。

例如：现场有一个已经通过 USB 模块和 Channel-Serial 通道服务连接的 DLT645 智能电表，用户想通过 Fox-Edge 对它进行管理。那么用户再 Fox-Edge 的静态解码仓库中安装 DLT645 解码器之后，此时设备列表中，就会出现对 DLT645 的支持。

步骤：

- 将 USB-RS485 模块插入工控机的 USB 口
- 在 Fox-Edge 安装 channel-serial 服务部件
- 新增一个 Serial 通道，与该 DLT645 电表建立会话连接
- 在 Fox-Edge 上选择安装静态解码组件，dlt645-core 和 dlt645-1997
- 将这两个模块设置为启动，并重启设备服务，然后会出现支持新增 DLT645 设备类型
- 在设备列表中，添加一个 DLT645 设备，并配置好 JSON 风格的设备地址参数
- 那么，在设备操作任务中，即可新建一个读取数据的任务，读取 DLT645 电表的数据

3.6.2 设备数值

当在 Fox-Edge 设备列表之中添加了一批智能设备之后，同时在任务管理中的设备监控任务中，添加了监控任务。

例如：现场有一个已经通过 USB 模块和 Channel-Serial 通道服务连接的 DLT645 智能电表，并且配置了 DLT645 设备解码器，以及一个 DLT645 数据采集任务。那么当用户部署并启用控制器服务的时候，控制器服务会根据监控任务不断收集 DLT645 设备的数据，并在获得设备数据后，通知持久化服务保存起来。用户可以在设备数值这边，看到 DLT645 设备的采样数据了。

步骤：

- 将 USB-RS485 模块插入工控机的 USB 口
- 在 Fox-Edge 安装 channel-serial 服务部件，并创建一个设备连接通道
- 在 Fox-Edge 上选择安装 DLT645 解码组件，然后在设备列表中添加一个 DLT645 设备对象
- 在 Fox-Edge 安装 controller-service 服务，并在设备监控任务中，创建一个 DLT645 监控任务
- controller-service 服务会根据监控任务，对 DLT645 设备不断的采集数据，并通知给 persist-service 服务将读取的设备数据，保存起来
- 那么，设备数值列表中，可以查看到该 DLT645 设备数据不断的更新

3.6.3 设备映射

当在 Fox-Edge 对监控设备进行数据采样之，获得了设备第一份数据之后，会把该设备的数据作为设备数据的模板，展示在设备映射列表之中。并在后续的设备数据存储过程中，对后续数据进行存储前的预处理。

那么用户，可以对这个设备模板的设备采样数据对象，进行修改配置，包括别名化、映射、备注等各种操作，使之更符合用户需要的数据格式。

3.6.4 扩展信息

Fox-Edge 对采样的对象，只预定义了数值和名称两个字段。但是用户可能会需要更多的自定义字段，作为自己的业务描述信息。那么用户可以通过为设备定义扩展信息，新增逻辑字段。

例如：用户需要在每个数据采样对象上，增加单位和备注，那么可以在扩展信息这边添加一个 unit 和 description 两个字段，那么就可以在设备映射中，看到这新增的两个字段。

3.6.5 通信状态

Fox-Edge 对现场的设备进行通信的时候，现场的智能设备可能某个时刻，突然连接中断。通信状态记录了设备最近通信成功的时间和通信失效的时间，那么用户可以以此判断，这个设备是否已经断联了。

3.7 任务管理

Fox-Edge 在安装完成各种服务组件、解码器组件、设备模板，并添加设备和通道配置之后。Fox-Edge 在任务管理中，为用户需要对设备进行相关的操作任务。

3.7.1 通道操作任务

当 Fox-Edge 与设备建立连接之后，并不知道 Fox-Edge 与设备之间是否真的联通，此时通道操作任务，为用户提供了测试通信功能。

例如：现场有一个已经通过 USB 模块和 Channel-Serial 通道服务连接的 DLT645 智能电表，用户想通过 Fox-Edge 对它进行连通性测试。

步骤：

- 在 Fox-Edge 的通道列表中，为该设备建立一个串口通道
- 在通道操作任务中，为该新建的串口通道新建一个测试任务
- 在模板参数中，填写一个 DLT645 的通信测试报文
- 在该通道操作任务上点击发送，可以看到 DLT645 返回的十六进制报文

3.7.2 设备操作任务

当 Fox-Edge 与设备建立连接之后，并不知道 Fox-Edge 与设备之间是否可以互相操作，此时设备操作任务，为用户提供了测试操作的功能。

例如：现场有一个已经通过 USB 模块和 Channel-Serial 通道服务连接的 DLT645 智能电表，用户想通过 Fox-Edge 对它进行设备操作测试。

步骤：

- 在 Fox-Edge 的通道列表中，为该设备建立一个串口通道
- 在 Fox-Edge 的设备列表中，为添加一个设备，并绑定上面的串口通道
- 在设备操作任务中，新建一个读数据操作的任务
- 在该设备操作任务上点击发送，可以看到 DLT645 返回数据，并解码后获得设备数据。

3.7.3 设备监控任务

当 Fox-Edge 能够对设备操作之后，用户往往需要对设备进行定期数据采集，此时设备监控任务，为用户提供了周期性数据采集的功能。

例如：现场有一个已经通过 USB 模块和 Channel-Serial 通道服务连接的 DLT645 智能电表，用户用户也在设备操作任务中，对该设备测试成功了。

步骤：

- 在 Fox-Edge 的通道列表中，为该设备建立一个串口通道
- 在 Fox-Edge 的设备列表中，为添加一个设备，并绑定上面的串口通道
- 在设备监控任务中，新建一个读数据操作的任务，其中配置参数可以参考前面的设备操作任务
- 那么在设备数值页面，可以看到从 DLT645 获得的数据，不断的更新变化之中。

3.8 周期管理

Fox-Edge 在经过配置监控任务之后，控制器服务会不断的采集数据，保存最新的设备数据到 REDIS 之中。

很多用户，需要对保存在 REDIS 中的快照，按一定的周期，不断的进行数据的快照抓取，并保存到数据库之中，作为一种历史数据。此时，可以安装 period-service 服务，那么该服务在经过周期抓取配置之后，不断的从 REDIS 的快照中，抓取数据，并保存到数据库之中。

3.8.1 周期任务

当 Fox-Edge 在安装完成 period-service 之后，可以配置周期快照抓取任务。当配置完成之后，period-service 会根据任务，不断的抓取数据，并保存到数据库中。

3.8.2 周期记录

当 Fox-Edge 在安装完成 period-service，并配置完成周期任务之后，不断抓取的历史快照记录，可以在周期记录之中，查询获得。

3.9 记录管理

Fox-Edge 支持设备上报的事件，比如火灾的告警、门禁的开门，会将这些事件作为设备的记录，保存起来。另外，Fox-Edge 同样支持对用户的手动操作，以及设备的数值变化，进行历史记录。

3.8.1 设备记录

Fox-Edge 支持设备上报的事件，比如火灾的告警、门禁的开门，会将这些事件作为设备的记录，保存起来。用户可以在设备记录中，查询到设备上报的记录类型数据。

3.8.2 操作记录

用户会通过 Fox-Edge 对设备进行一些控制类型的操作，这些动作往往牵涉到责任和管理上的敏感性。用户可以在这边进行历史上的手动操作记录。

3.8.3 历史记录

当 Fox-Edge 会对设备采样的数字类数据，进行周期的记录，方便用户对设备运行状态的监测管理。

3.9 链路管理

Fox-Edge 提供了链路服务，用来解决数据的透传问题。在某些场景下，用户希望设备的数据在云端进行报文处理，那么可以在安装了对应的链路服务之后，通过链路，对 Fox-Edge 管理的现场设备数据，进行云端获得。

3.10 模型管理

某些第三方云平台，需要跟线下的智能网关，约定好上传设备的数据格式。该数据格式是由第三方云平台指定的，可以将第三方平台他的数据模型，配置在这边，提供给北向连接服务使用。

3.11 北向连接

Fox-Edge 能够对南向设备进行管理，同样也能把这些数据，推送给北向的第三方云平台。

可以从组件仓库中，安装 `iot-xxx` 的北向连接服务，就可以跟北向的第三方云平台对接，并把数据推送给它们。

同样的，用户也可以为自己的私有云平台，开发对应的北向连接服务，然后部署到 Fox-Edge 之中，把 Fox-Edge 的数据推送给自己的云平台。

3.11.1 第三方云平台 ThingsBoard

背景: ThingsBoard 是国外一家知名的 2C 类第三方物联网云平台，它支持通过 HTTP 接口将设备数据推送给它。

Fox-Edge 在安装 `iot-thingsboard` 服务之后，会周期性的将设备数据推送给第三方云平台 ThingsBoard。



3.11.2 第三方云平台 ThingsPanel

背景: ThingsPanel 是国内一家知名的 2C 类第三方物联网云平台, 它支持通过 HTTP 接口将设备数据推送给它。

Fox-Edge 在安装 `iot-thingspanel` 服务之后, 会周期性的将设备数据推送给第三方云平台 ThingsPanel。



3.11.3 第三方云平台 HuaWei

背景: HuaWei 是国内一家知名的 2C 类第三方物联网云平台, 它支持通过 MQTT 接口将设备数据推送给它。

Fox-Edge 在安装 `iot-huawei` 服务之后, 会周期性的将设备数据推送给第三方云平台 HuaWei。



3.11.4 第三方云平台 WhZktl

背景：WhZktl 是国内一家物联网云平台，它支持通过 MQTT 接口将设备数据推送给它。

Fox-Edge 在安装 `iot-whzktl` 服务之后，会周期性的将设备数据推送给第三方云平台 WhZktl。

3.11.5 灵狐云平台 Fox-Cloud

背景：Fox-Cloud 是灵狐提供的一个物联网云平台，它支持通过 MQTT 或 HTTP 接口将设备数据推送给它。

Fox-Edge 在安装 `iot-fox-cloud` 服务之后，会周期性的将设备数据推送给云平台 Fox-Cloud。

4 典型组网

企业用户对甲方项目进行系统设计和项目实施的时候，输出的是一个帮甲方能够具体解决问题的解决方案。

在一个解决方案中，通常包括应用场景和网络部署。简单说，就是如何使用 Fox-Edge 产品。

Fox-Edge 可以应用于，但不限于下列场景。

4.1 上位机场景

有很多小型物联网项目，它们并没有复杂的管理需求，对现场的智能设备，主要是希望有一个能够接入智能设备的现成可用的上位机系统。

需求特点：

- 智能设备的规模小，通常没有十几个
- 能管理智能设备
- 有一个现成的管理界面
- 能够开箱即用

业务场景：

- 中小企业办公室的智能化管理
- 小型酒店的智能化管理

Fox-Edge 能够直接作为一台上位机系统，对现场的智能设备进行简单设备管理。

水平拓扑：

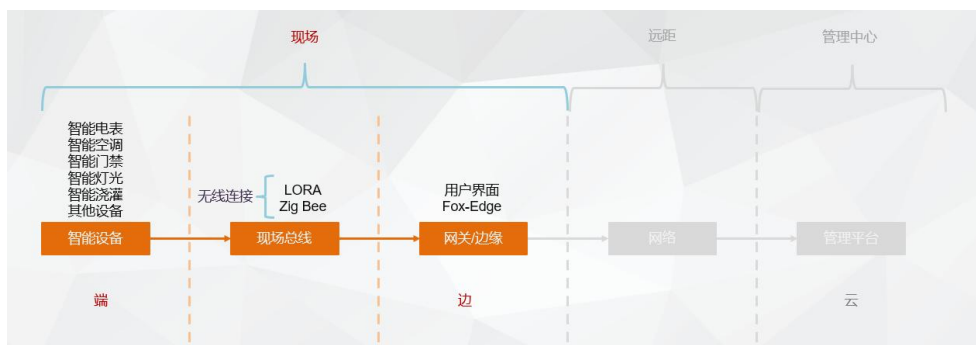


图 4-1

平面拓扑:

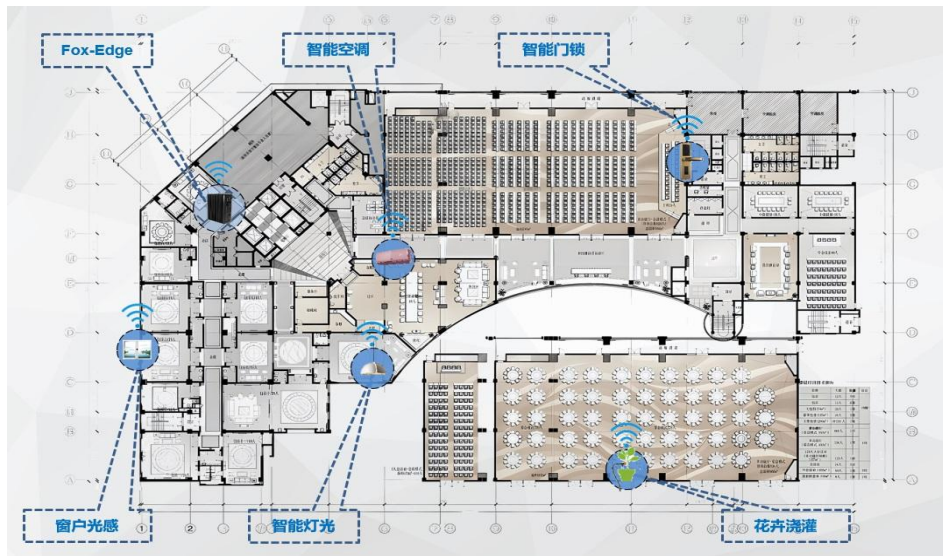


图 4-2

4.2 智能网关场景

有很多中型物联网项目，它们本身有自己独立的上位机业务管理系统。同样也有数量和种类较多的现场的智能设备。

它们此时主要是需要一批能够充当智能网关的设备，在南向接入对现场的智能设备进行按片区接入管理，在北向能够被上位机业务系统管理。

需求特点:

- 智能设备数量比较多，种类比较多
- 有一个面向业务管理的上位机系统
- 需要一个能南向接入智能设备，北向接入上位机的智能网关

业务场景:

- 物业小区的水电煤气的智能化管理
- 写字楼的安防、空调、水电的智能化管理

水平拓扑（写字楼）:



图 4-3

平面拓扑（写字楼）：

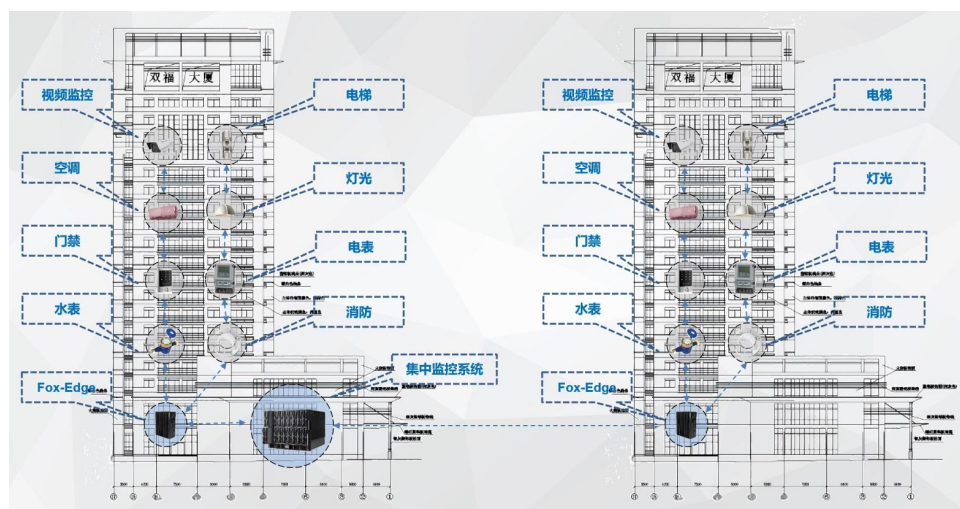


图 4-4

水平拓扑（物业小区）：



图 4-5

平面拓扑-室内（物业小区）：

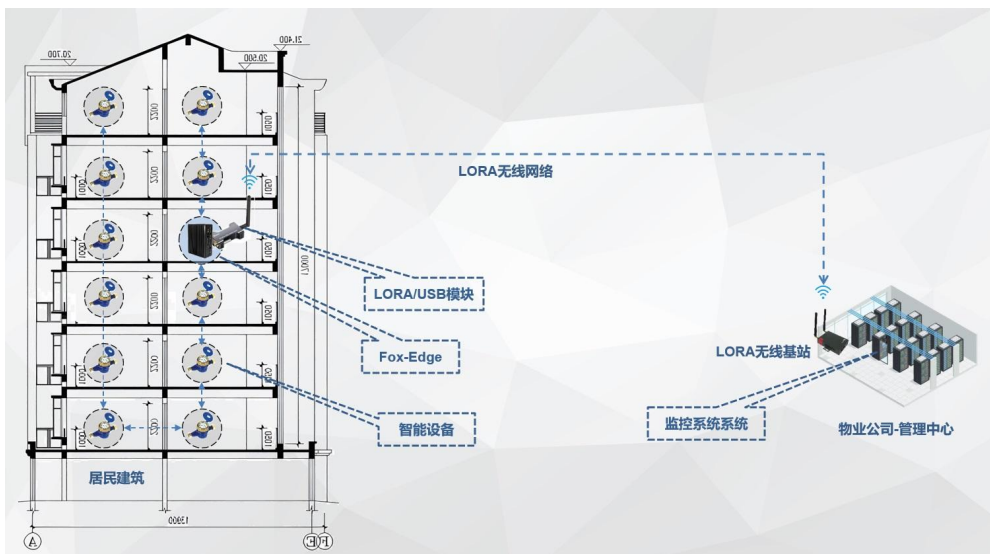


图 4-6

平面拓扑-室外（物业小区）：

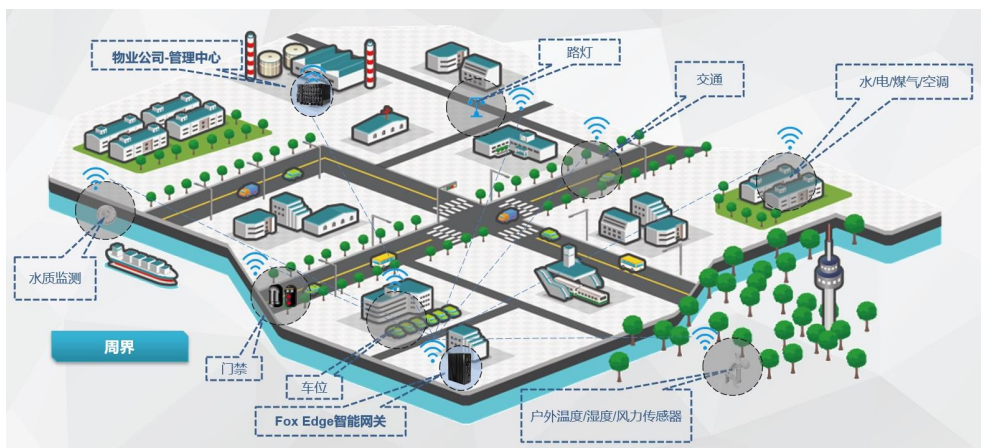


图 4-7

4.3 边缘计算场景

有很多大型物联网项目，它们已经不在局限于传统的上位机-智能网关-智能设备的场景。

它们在现场有海量的智能设备，业务能够在云端集中处理，也能下沉到现场的边缘端去处理。云端、边缘端不在视为一个功能的划分，而是视为不同位置的业务容器。

它们此时，需要一个既能够根据需要，将一些业务应用部署在云端，也同样能够根据需求将这些业务应用部署在边缘端。

Fox-Edge 能够把云端下沉过来的业务服务纳入管理，也能够自己的服务上浮到云端，被云端系统管理。

需求特点:

- 设备众多，业务复杂
- 业务管理灵活
- 业务子系统能够上浮到云端，也能下沉到边缘端
- 需要一个与云端采用同技术栈的边缘端系统

业务场景:

- 物联网的 AI 智能管理
- 边缘端的视频、设备的综合业务管理