

AWS Greengrass介绍

陈雪杰，AWS解决方案架构师

2017年6月27日

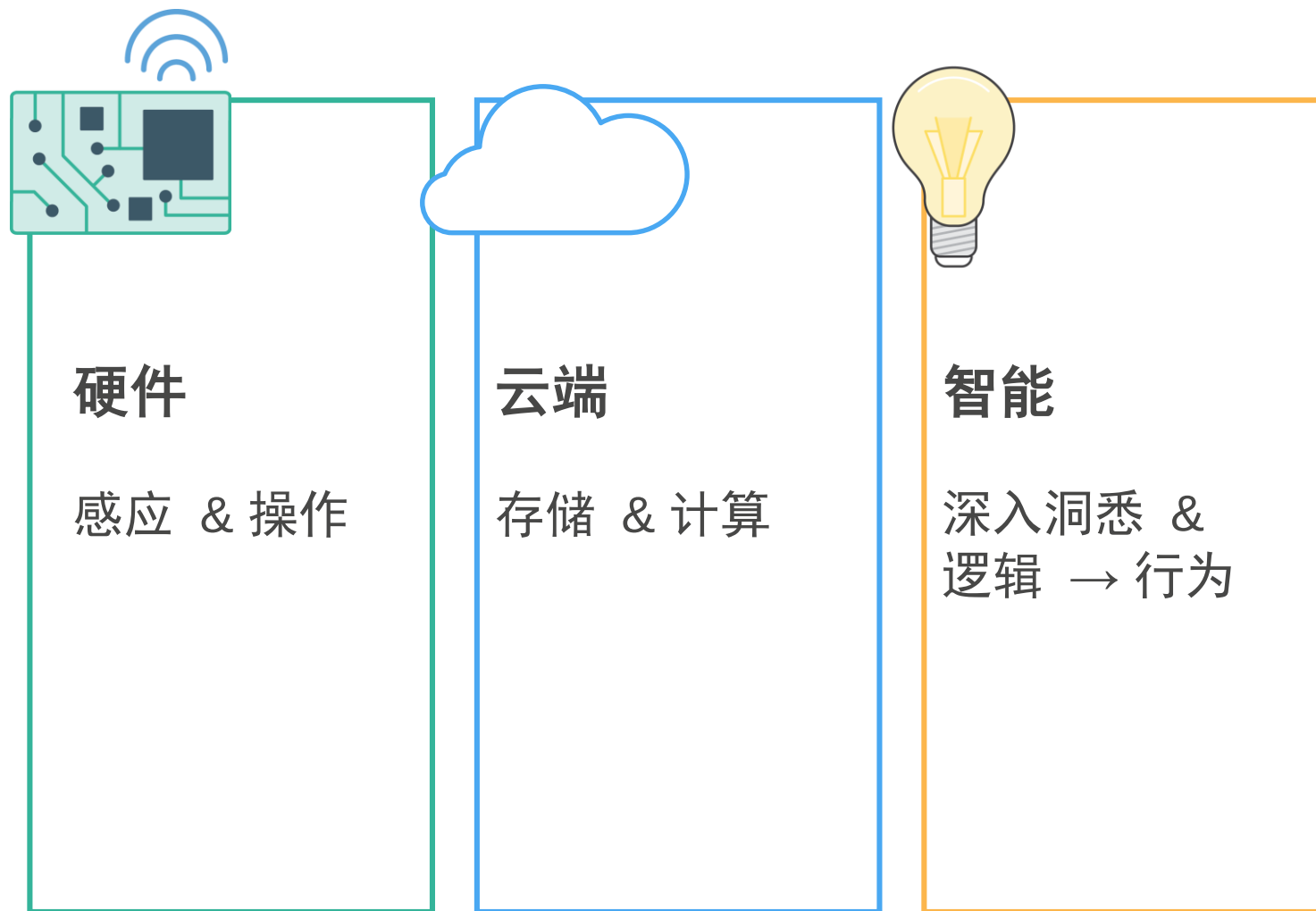
- ❖ 为什么需要AWS Greengrass?
- ❖ 什么是AWS Greengrass?
- ❖ AWS Greengrass的组件
- ❖ AWS Greengrass的功能
- ❖ AWS Greengrass代码示例
- ❖ 案例分享



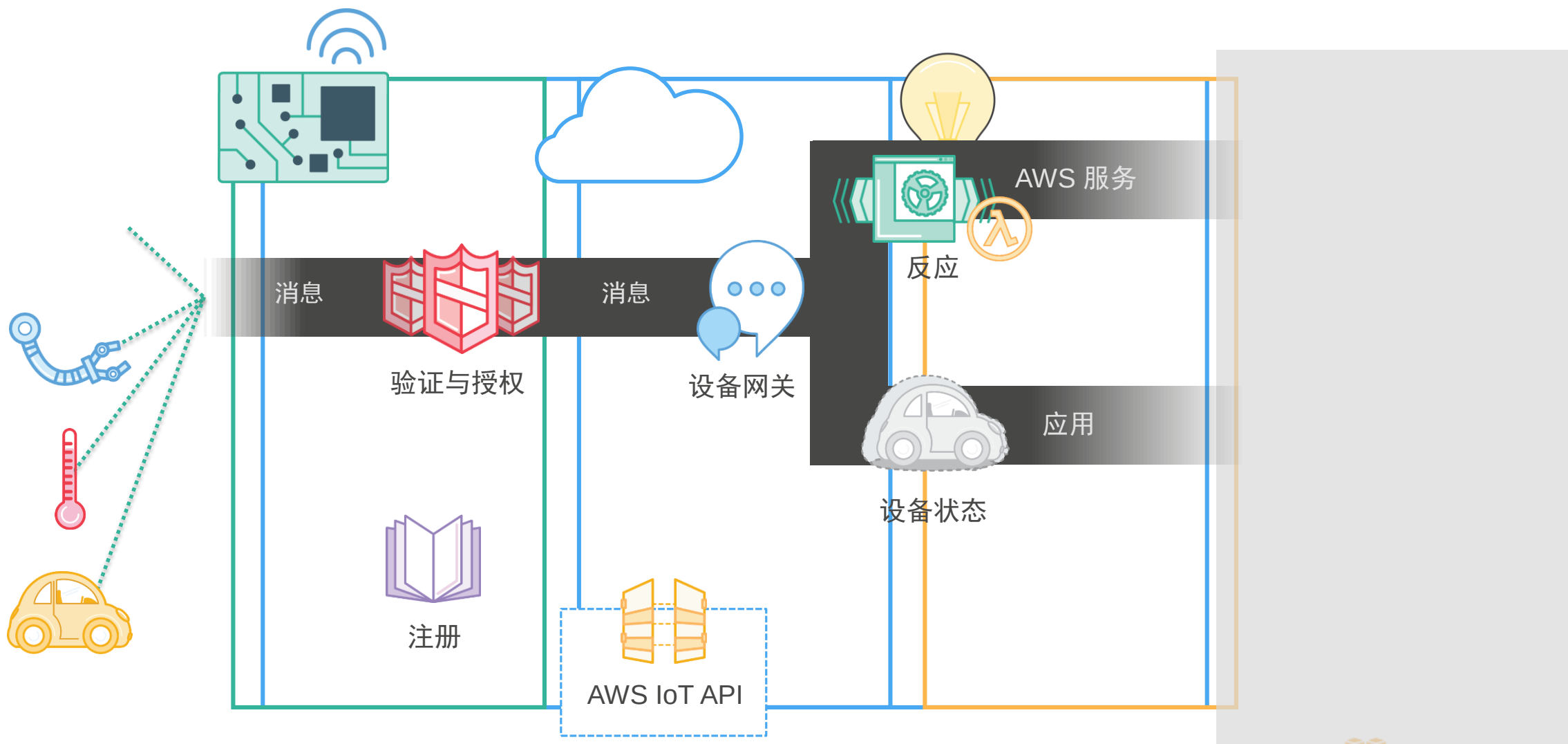


为什么需要AWS Greengrass ?

物联网三大支柱



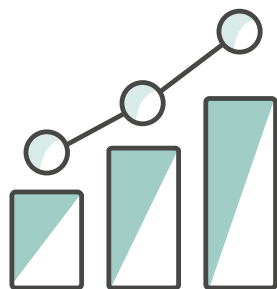
AWS IoT 从云开始



在源头处理数据的价值



物理的限制



经济的考虑



法律的约束

AWS IoT 从Edge开始



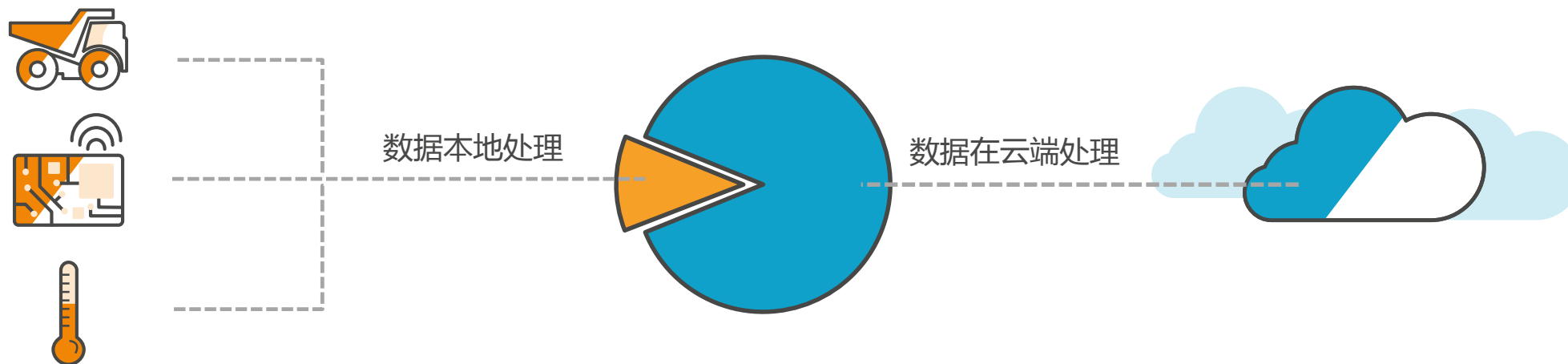


什么是AWS Greengrass?

AWS Greengrass

将计算移到Edge

AWS Greengrass 将AWS扩展到你的设备里, 所以你既可以在本地处理数据, 又可以利用云端的优势。



AWS Greengrass

特性



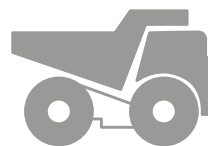
本地响应

本地Lambda函数



本地触发

本地消息代理



状态同步

本地设备影子



安全

本地安全认证

AWS Greengrass

收益



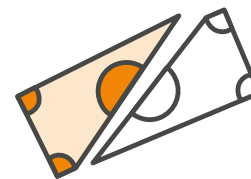
本地事件快速响应



离线操作



简化设备编程



减少IoT应用成本



安全通信

AWS Greengrass应用场景



合作伙伴生态



客户和合作伙伴

accenture

acer

Aricent

Ayla

BCG
Digital
Ventures

Bright Wolf

BSQUARE

CANONICAL

ctp
Cloud
Technology
Partners

DIGI

enel

intel

lenovo

lightbits
labs

LUXOFT

machineshop

MONGOOSE OS

NOKIA

PENTAIR

QNAP

QUALCOMM

RaspberryPi

resin.io

RioTinto

Saguna

SAMSUNG

solstice

StanleyBlack&Decker

ST
life.augmented

technicolor

TELUS

ThingLogix™

ThunderSoft®

trek10

wistron

amazon

annapurna
labs
an amazon company

AWS
Snowball Edge

amazon
web services | Webinars

AWS Greengrass价格

活跃设备

价格/每个

3

1年免费

3–10,000

\$0.16/月
\$1.49/年

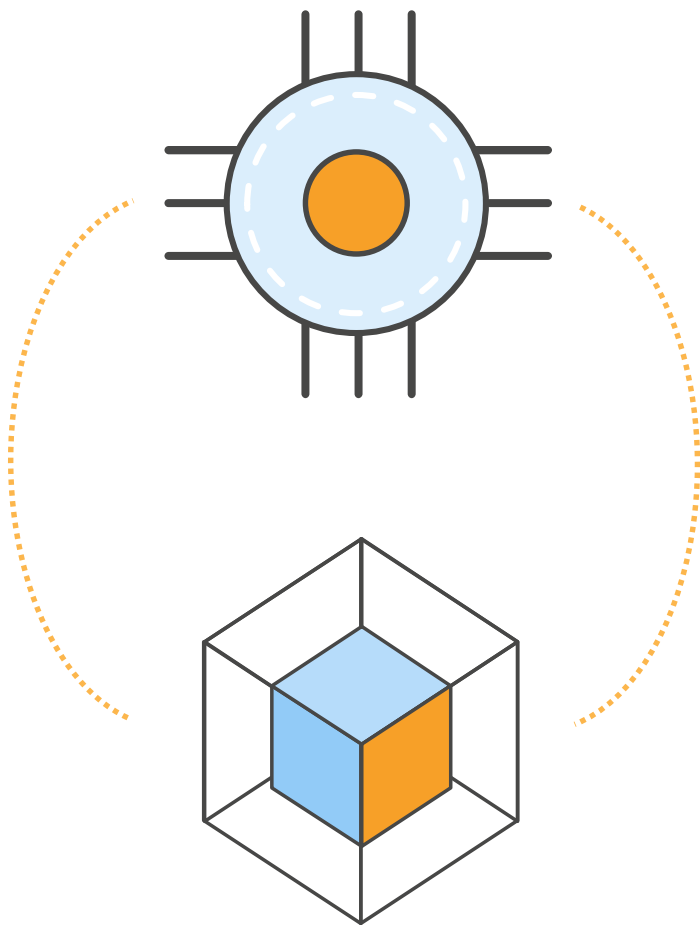
10,000+

请与我们联系



AWS Greengrass的组件

Greengrass 组件

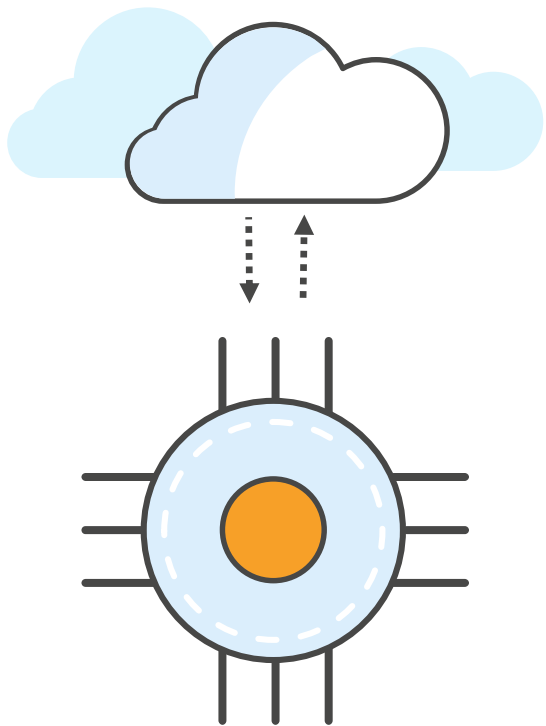


Greengrass是软件, 非硬件形式

由两个主要组件构成:

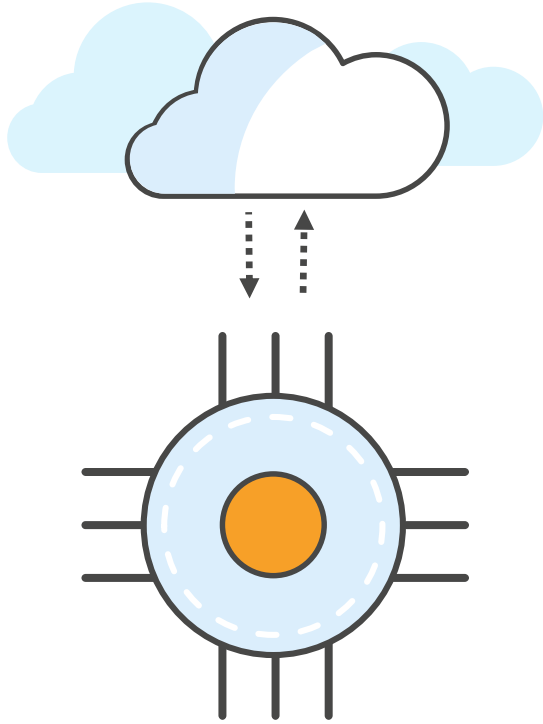
1. Greengrass Core
2. IoT Device SDK

AWS Greengrass Core (GGC)



负责执行Lambda
消息传递 (pub/sub)
安全
设备影子
可直接与云端交互

AWS Greengrass Core (GGC)



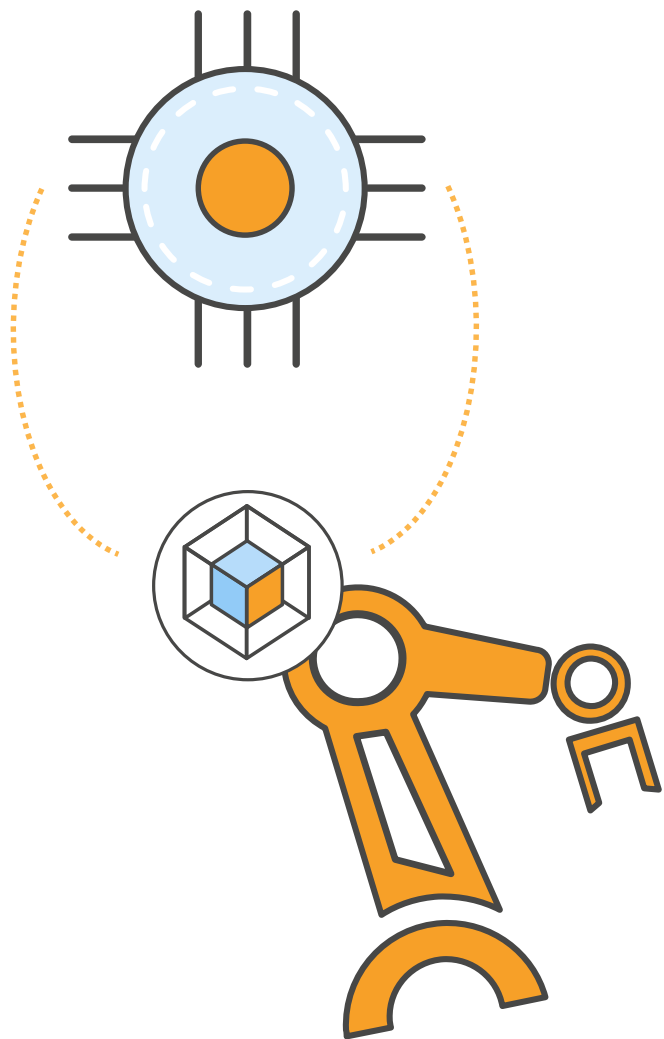
最低 单核 1 GHz

最低 128 MB RAM

x86 & ARM

Linux (Ubuntu 或 Amazon)

IoT device SDK

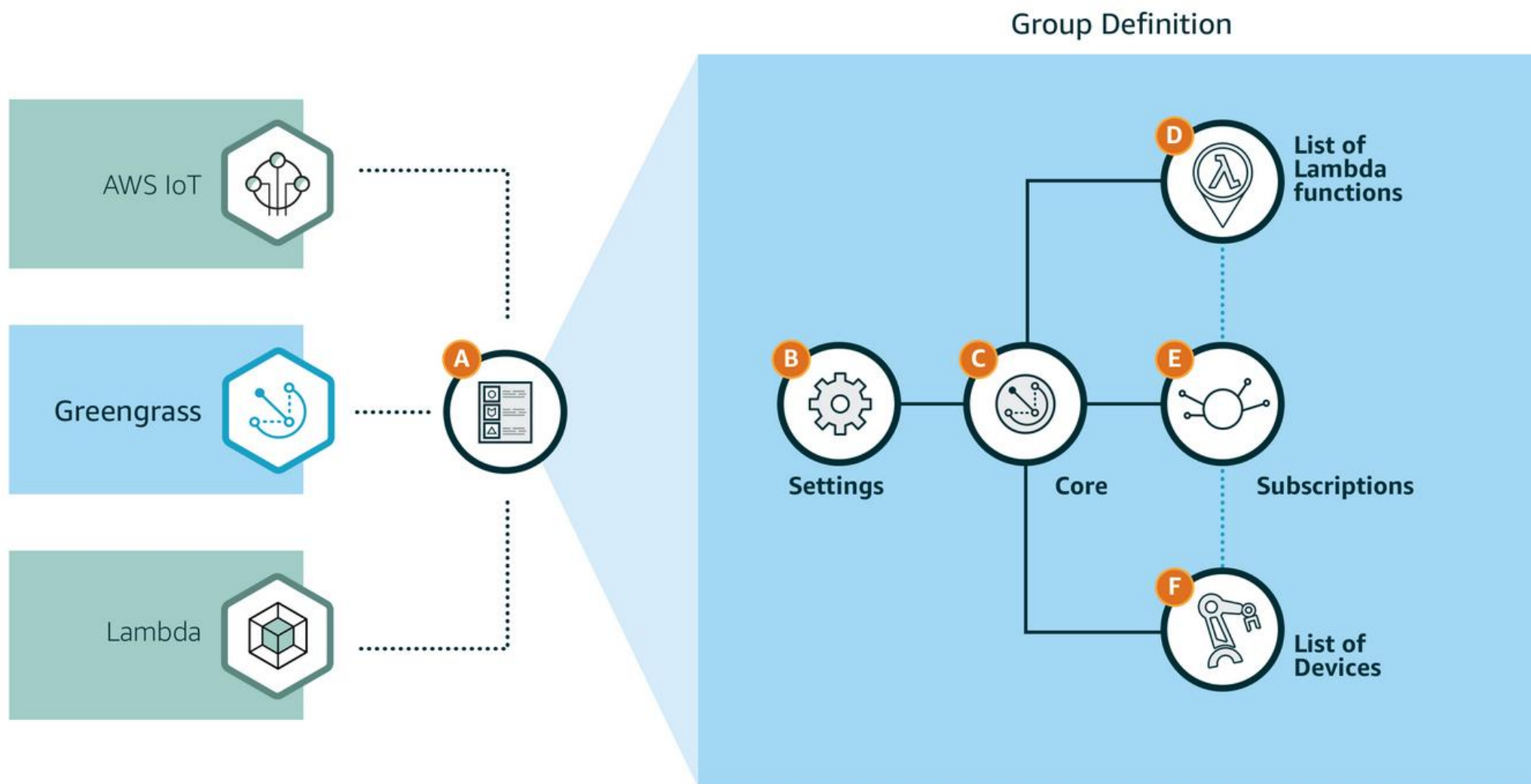


任何设备使用IoT device SDK可透过本地网络与GGC 交互

连接到AWS Greengrass 的硬件可大可小(microcontroller-based).

从C++ 开始支持, 其他语言会陆续发布

Greengrass相关概念



Greengrass组

GREENGRASS 组

GGGroup1

● 部署成功完成

操作 ▾

部署

组历史记录概述

按部署 ▾

订阅	已部署	版本	状态
核心	2017年6月20日 下午6:51:09 ...	d953d76d-f489-472f-b53f-...	● 部署成功完成 ...
设备	2017年6月20日 下午6:46:48 ...	e78cd02c-33bd-4f24-aa4c-...	● 部署成功完成 ...
Lambda	2017年6月20日 下午6:36:16 ...	603984e1-6135-4c24-a6a5...	● 部署成功完成 ...
设置	2017年6月20日 下午6:25:36 ...	0f2868fd-22ea-43aa-9919-...	● 部署成功完成 ...
	2017年6月20日 上午11:41:2...	d6ba0e67-a24f-49c1-9447...	● 部署成功完成 ...

Greengrass 订阅

GREENGRASS 组

GGGroup1

● 部署成功完成

操作 ▾

部署

订阅

核心

设备

Lambda

设置

添加订阅

源	目标	主题
 Grove_Sensor	 Local Shadow Service	\$aws/things/Grove_Sensor/... ...
 esp32_055B18	 Local Shadow Service	\$aws/things/Grove_Sensor/... ...
 Local Shadow Service	 Grove_Sensor	\$aws/things/Grove_Sensor/... ...
 Local Shadow Service	 Grove_Sensor	\$aws/things/Grove_Sensor/... ...

Greengrass Core

GREENGRASS 组

GGGroup1

● 部署成功完成

操作 ▾

部署

订阅

核心

设备

Lambda

设置

核心

GGGroup1_Core

核心

Greengrass 设备

GREENGRASS 组

GGGroup1

● 部署成功完成

操作 ▾

部署

订阅

核心

设备

Lambda

设置

设备

添加设备

RobotArm_Thing

设备

仅限本地影子

esp32_055B18

设备

仅限本地影子

Grove_Sensor

设备

仅限本地影子

Greengrass Lambda

GREENGRASS 组

GGGroup1

● 部署成功完成

操作 ▾

部署

订阅

核心

设备

Lambda

设置

Lambda

添加 Lambda

messageLambda

LAMBDA 函数

使用 V1

updateGroveSensor

LAMBDA 函数

使用 V2

uptimeLambda

LAMBDA 函数

4, 3, 7, 5, 6

Greengrass 设置

GREENGRASS 组

GGGroup1

● 部署成功完成

操作 ▾

部署

订阅

核心

设备

Lambda

设置

组角色

GreengrassServiceRole

...

策略

CloudWatchFullAccess

AWSLambdaReadOnlyAccess

AWSIoTFullAccess

GreengrassPolicyInline

证书颁发机构 (CA) 和本地连接配置

设备证书生命周期

通过更改此设置，您可以控制设备能够与其核心建立通信的时间段。下一个新时间段将为 30 天。

7 天

30 天

30+

AWS中国（北京）区域由光环新网运营



Webinars



AWS Greengrass的功能

本地Lambda函数



Lambda函数是事件驱动的计算函数

使用AWS Greengrass, 您可以在云中编写Lambda函数, 并在本地部署

本地Lambda函数



AWS Greengrass 运行Python 2.7
编写的Lambda 函数

透过消息以及设备影子的变化来调
用Lambda 函数

本地Lambda能做什么？



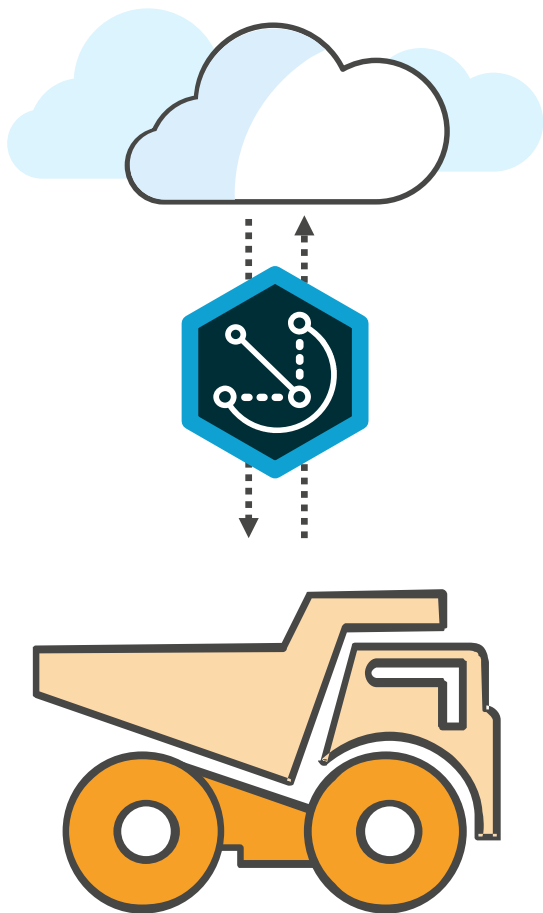
命令和控制

离线作业

数据过滤和聚合

迭代学习

设备影子

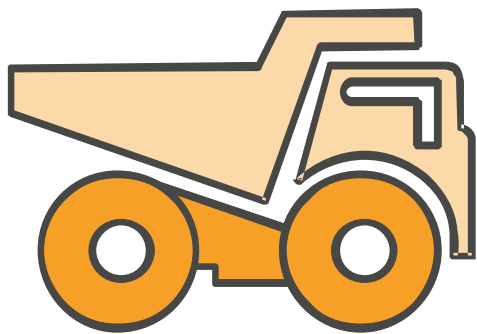


JSON 文件，描述您的设备与Lambda函数的状态

您可以自由的定义，例如一辆车，一颗引擎，甚至是一组车队

可以将状态同步回云端或是保存在本地端

设备影子能做什么？



设备状态(current & desired)

粒度设备状态
(仅与云进行同步以进行除错)

动态配置

消息



本地端的MQTT pub/sub 模式

定义pub/sub 规则

MQTT 主题过滤器

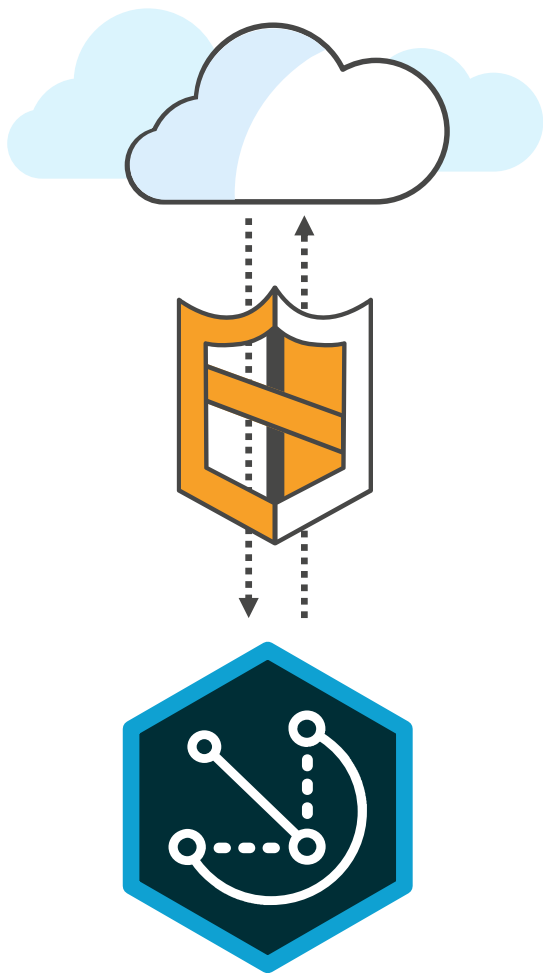
消息能做到什么？



云端的桥接器

本地的消息分发系统

安全



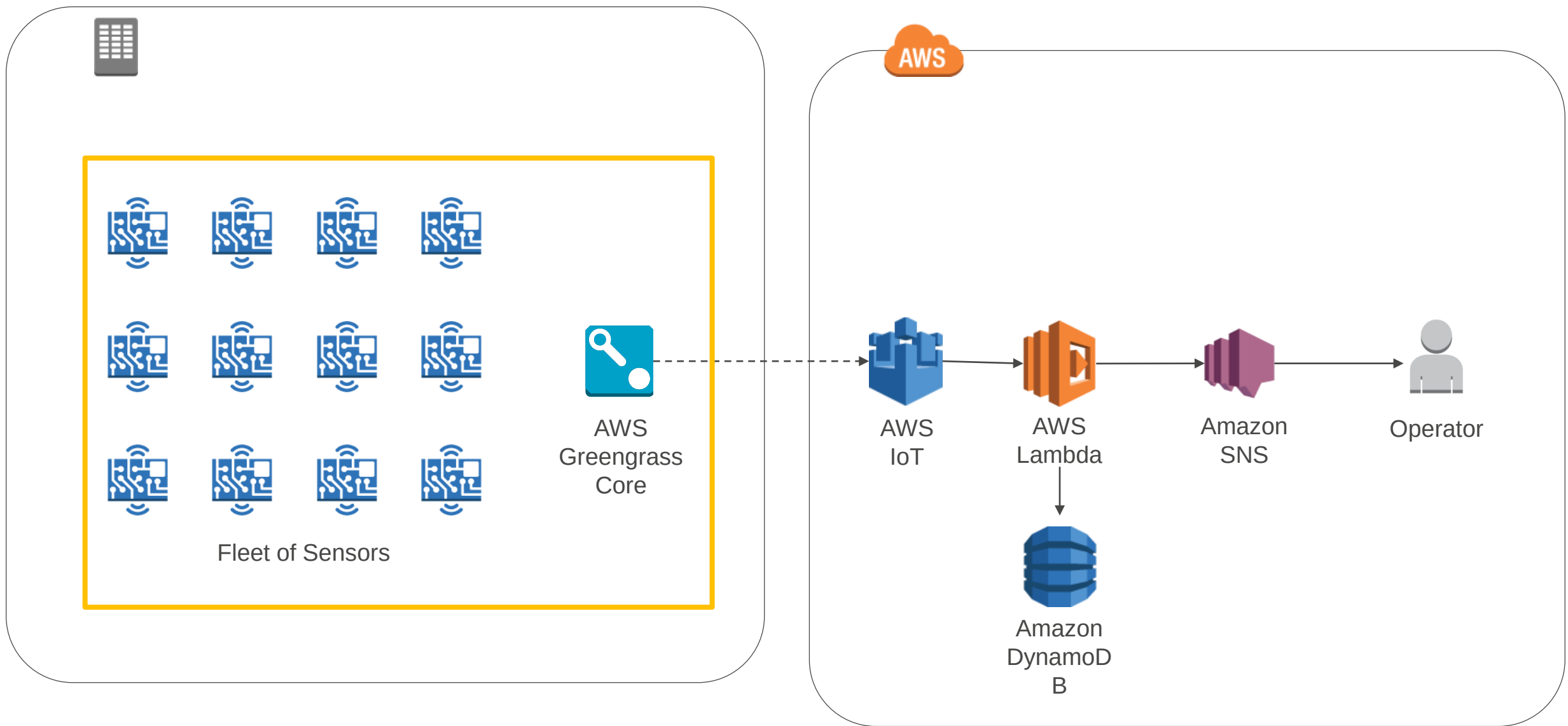
相互认证, 包含本地端与云端

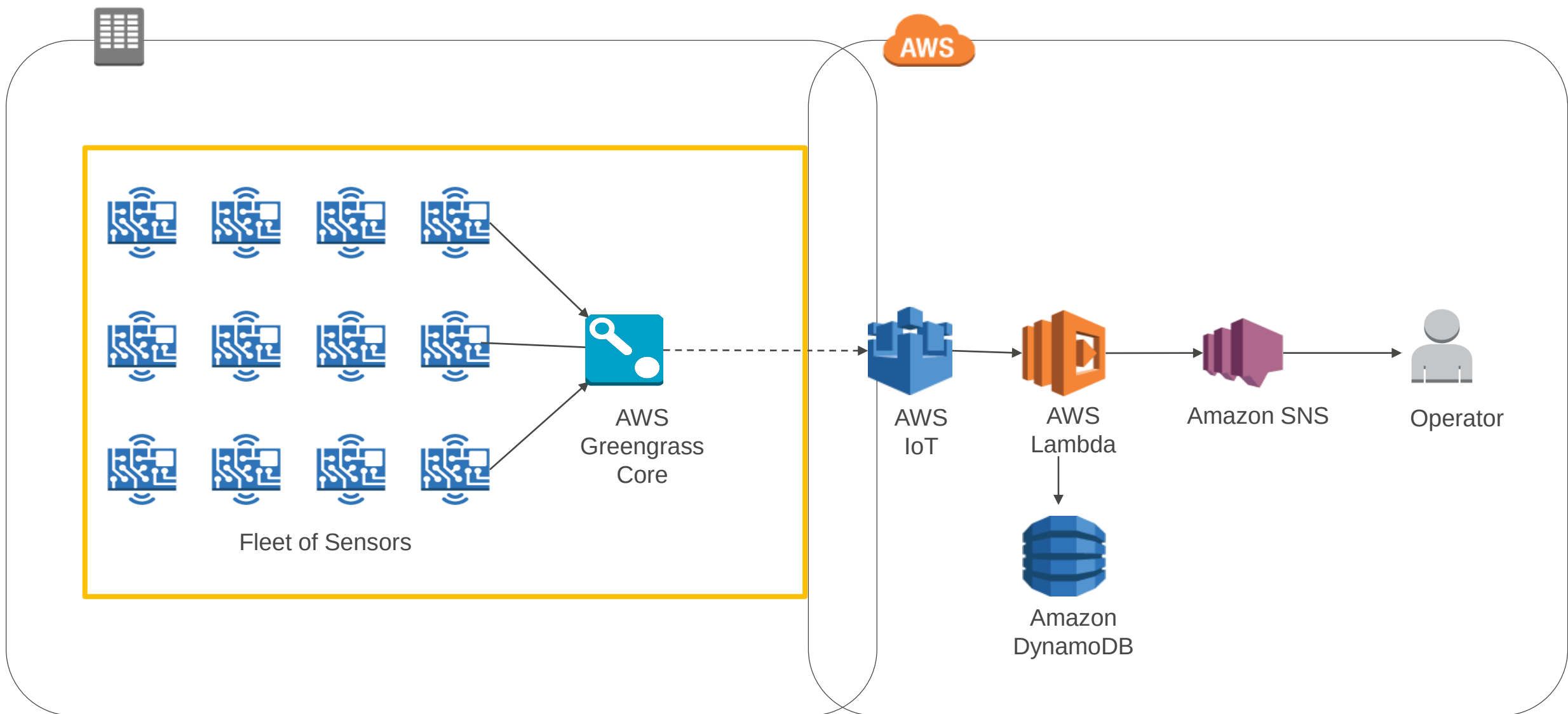
您设备上的证书可以与云中的SigV4凭据相关联

透过安全协议, 您可以从Greengrass呼叫任何AWS 的服务



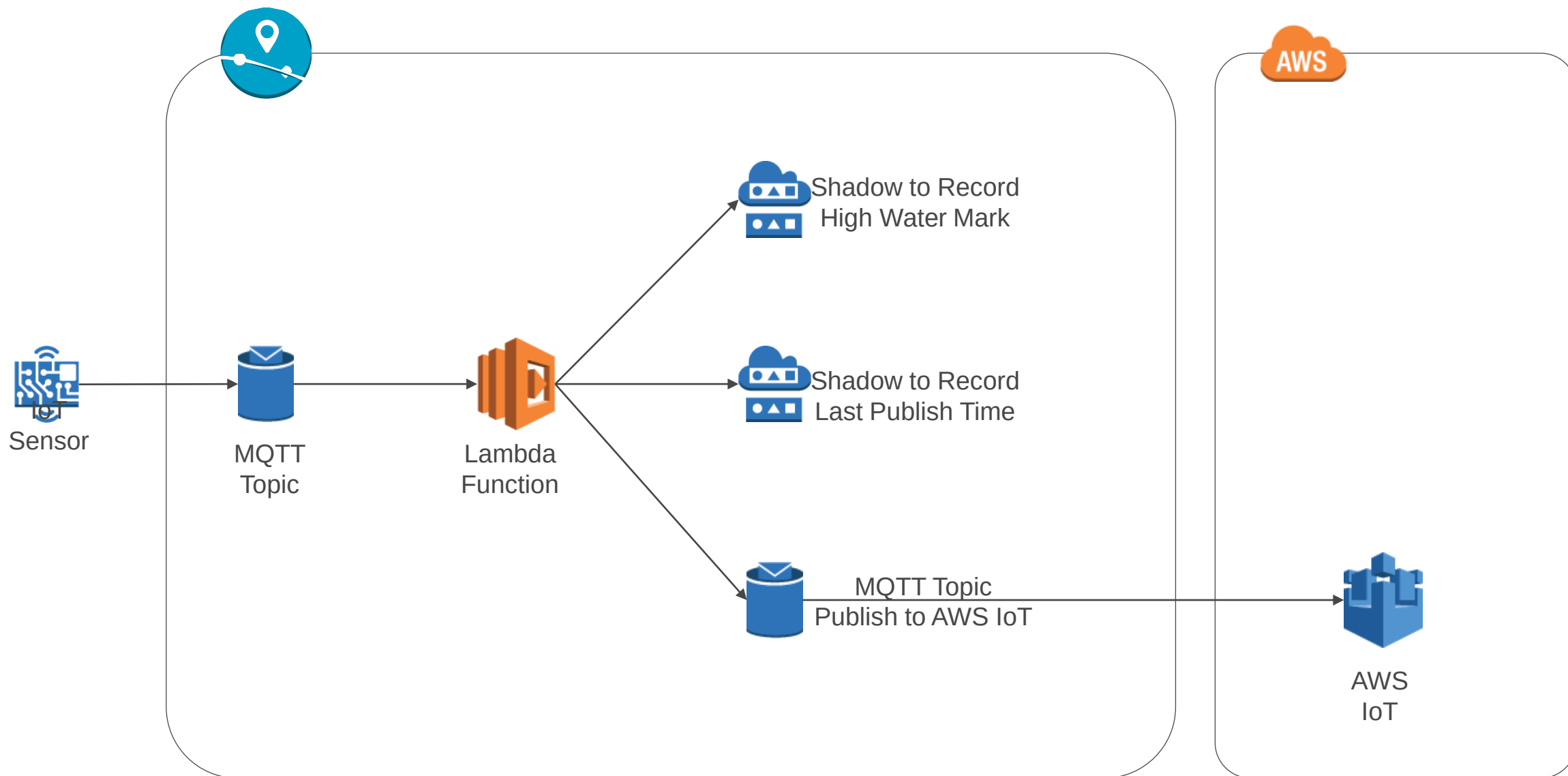
AWS Greengrass代码示例





载入与分析本地端设备影子

```
def current_record_data(cls):  
    try:  
        # The get_thing_shadow API loads a shadow  
        shadow = iot_client.get_thing_shadow(thingName=cls.name)['payload']  
        record_shadow = json.loads(shadow)  
        version = record_shadow['state'].get('version', 0)  
        ...  
        return record_value, last_push_time, version  
  
    except ShadowError as e:  
        # The shadow doesn't exist yet, so return some default values  
        if str(e).startswith('Request for shadow state'):  
            return None, None, 0  
        else:  
            raise e
```

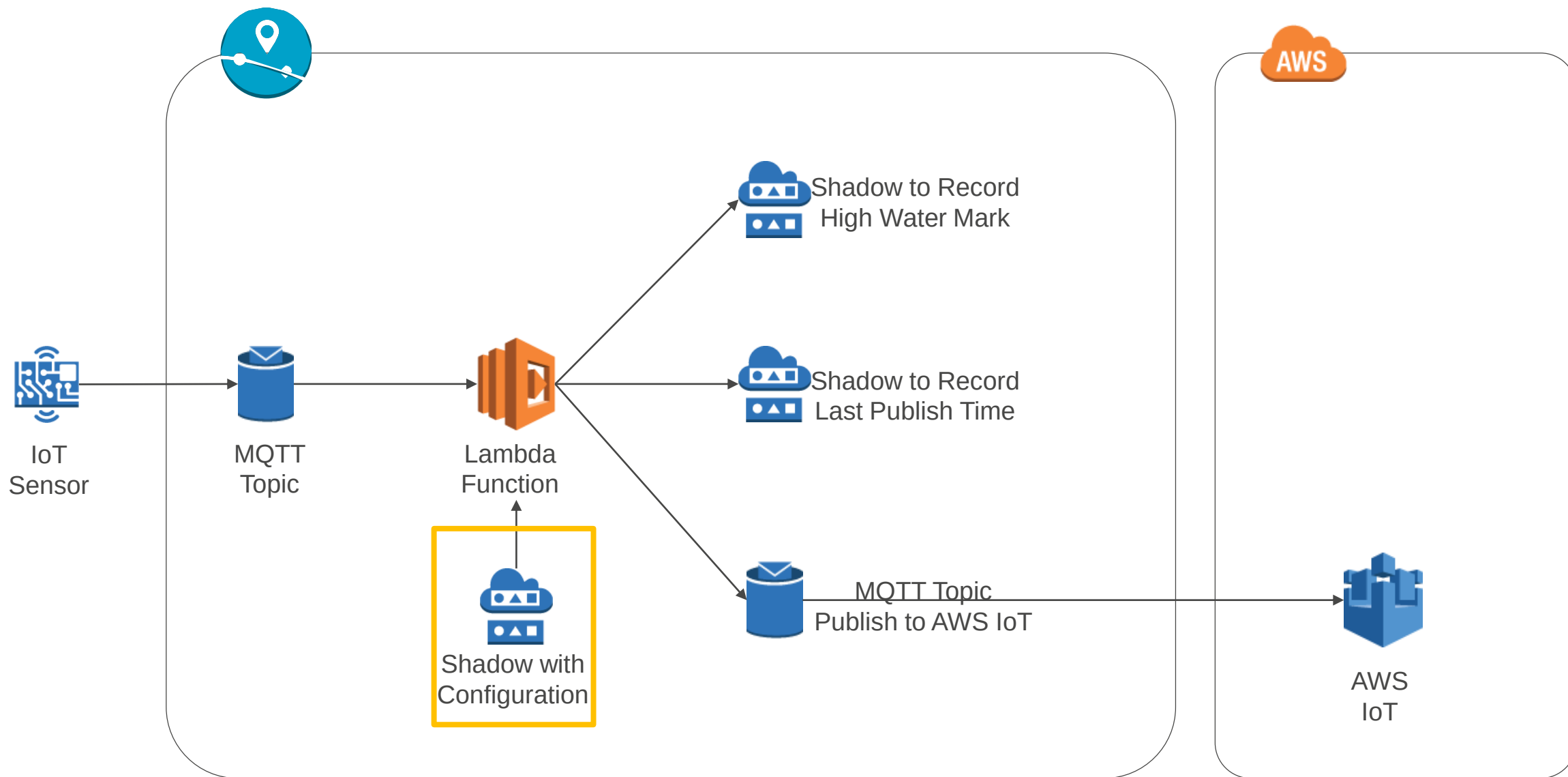



更新本地端设备影子

```
def update_shadow(self):  
    # Loop until shadow state update successfully occurs  
    while True:  
        current_shadow, last_push, version = self.current_record_data()  
  
        if not self.should_update(self, current_shadow):  
            return False  
  
        new_shadow = self.new_shadow(version)  
        if self._update_shadow(new_shadow):  
            return True  
  
    # Sleep with jitter  
    time.sleep(random.random())
```

透过MQTT发布消息到云端

```
def push_record(self):  
    # Loop until shadow state update is successful  
    while True:  
        record, last_push_time, version = self.current_record_data()  
  
        if not self._should_push_record(last_push_time):  
            # Another Lambda has already posted. Job done!  
            return False  
  
        # Try to update the thing shadow  
        if self._try_update_shadow(version):  
            # If that works, publish the message  
            iot_client.publish(topic=self.record_reporter_topic, payload=str(record))  
            return True  
  
        # Sleep for a short while, with jitter  
        time.sleep(random.random())
```





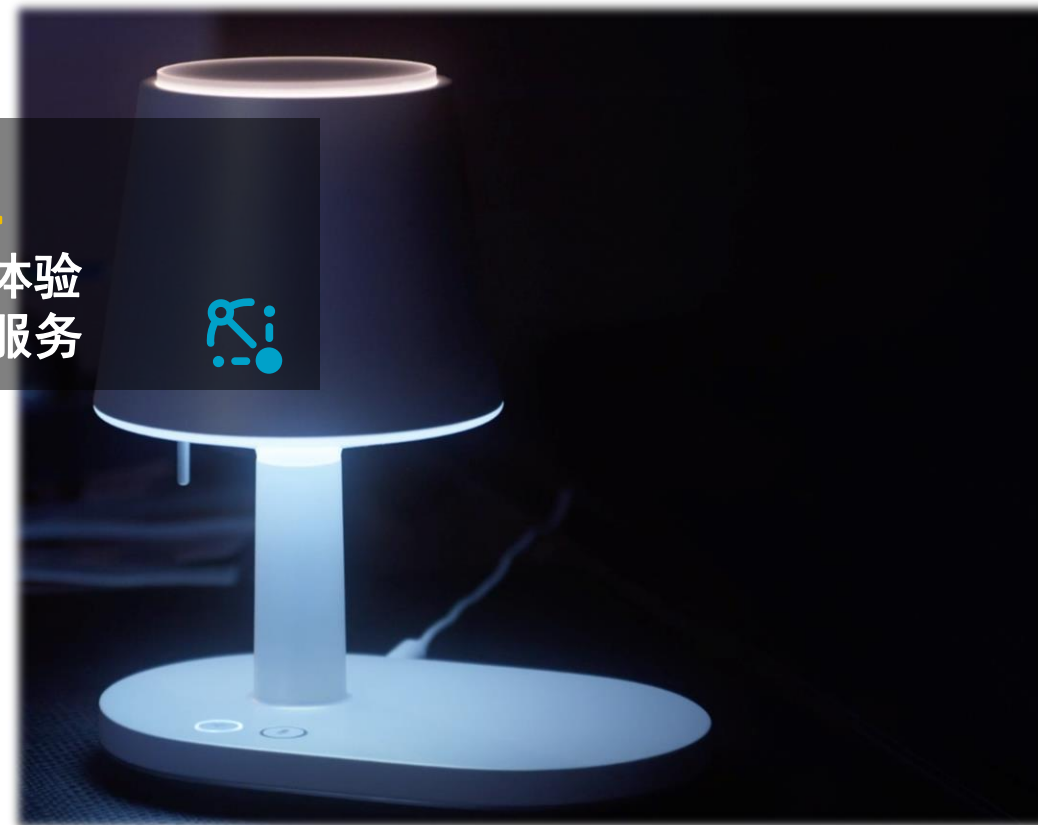
AWS Greengrass案例分享

新产品与服务中接入Greengrass

个人助理

语音 - 激活用户体验
应用 - 优化客户服务

10Gb Broadband GW



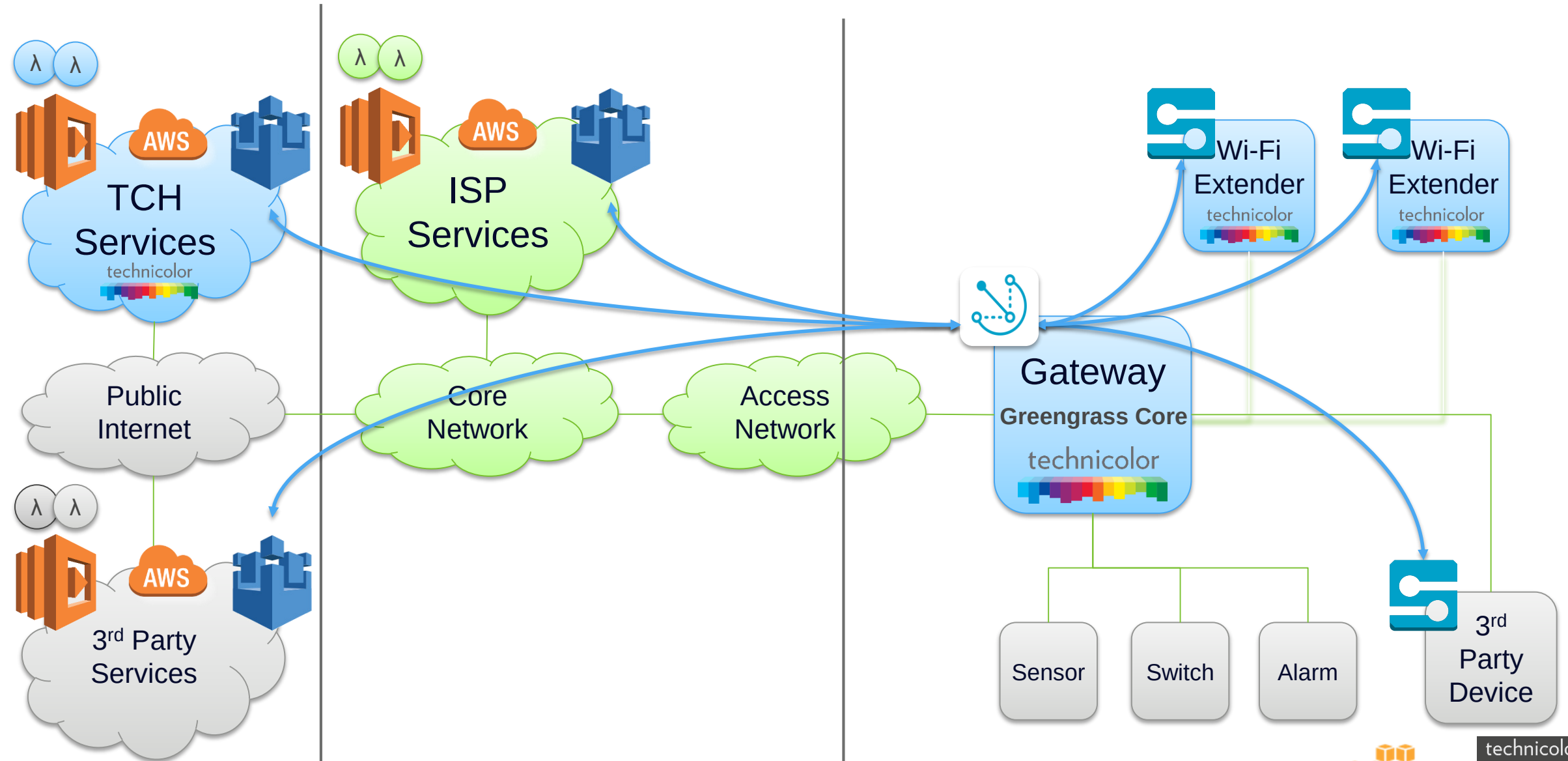
Tri-band Wi-Fi
Extender



接入Greengrass 后的好处



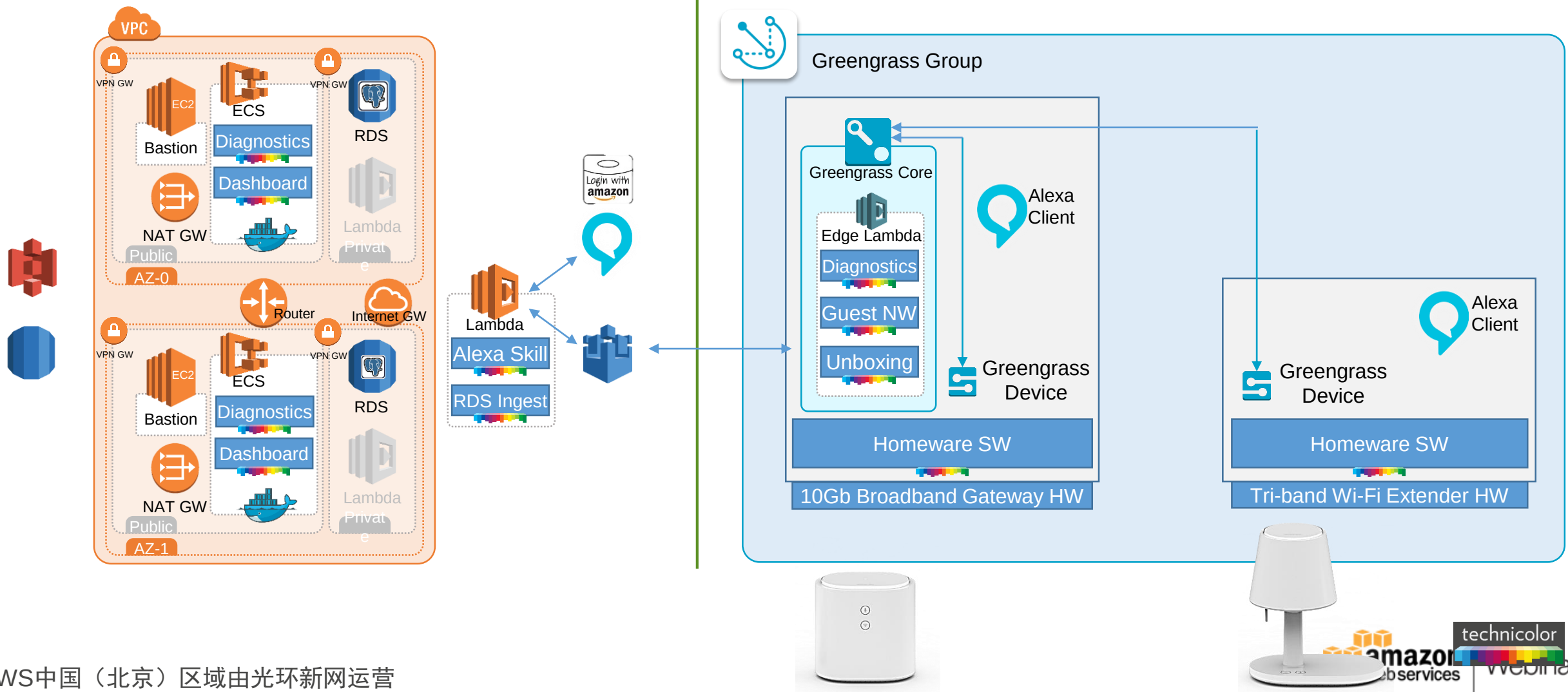
环境解耦



AWS中国（北京）区域由光环新网运营

详细架构与使用案例

CLOUD ← → EDGE



AWS中国（北京）区域由光环新网运营

结论

新产品

- ▶ 作为家用设备的领先厂商, TECHNICOLOR很高兴能介绍运用Greengrass 接入新的产品和服务

优势

- ▶ Greengrass能帮助我们更快的切入市场(TTM), 降低总成本, 提高灵活性和隐私安全, 让我们的服务提供商和消费者受益

机会!

- ▶ 通过在CLOUD和EDGE之间建立桥梁, 我们为创新和改善消费者体验创造了机会

Thank You!