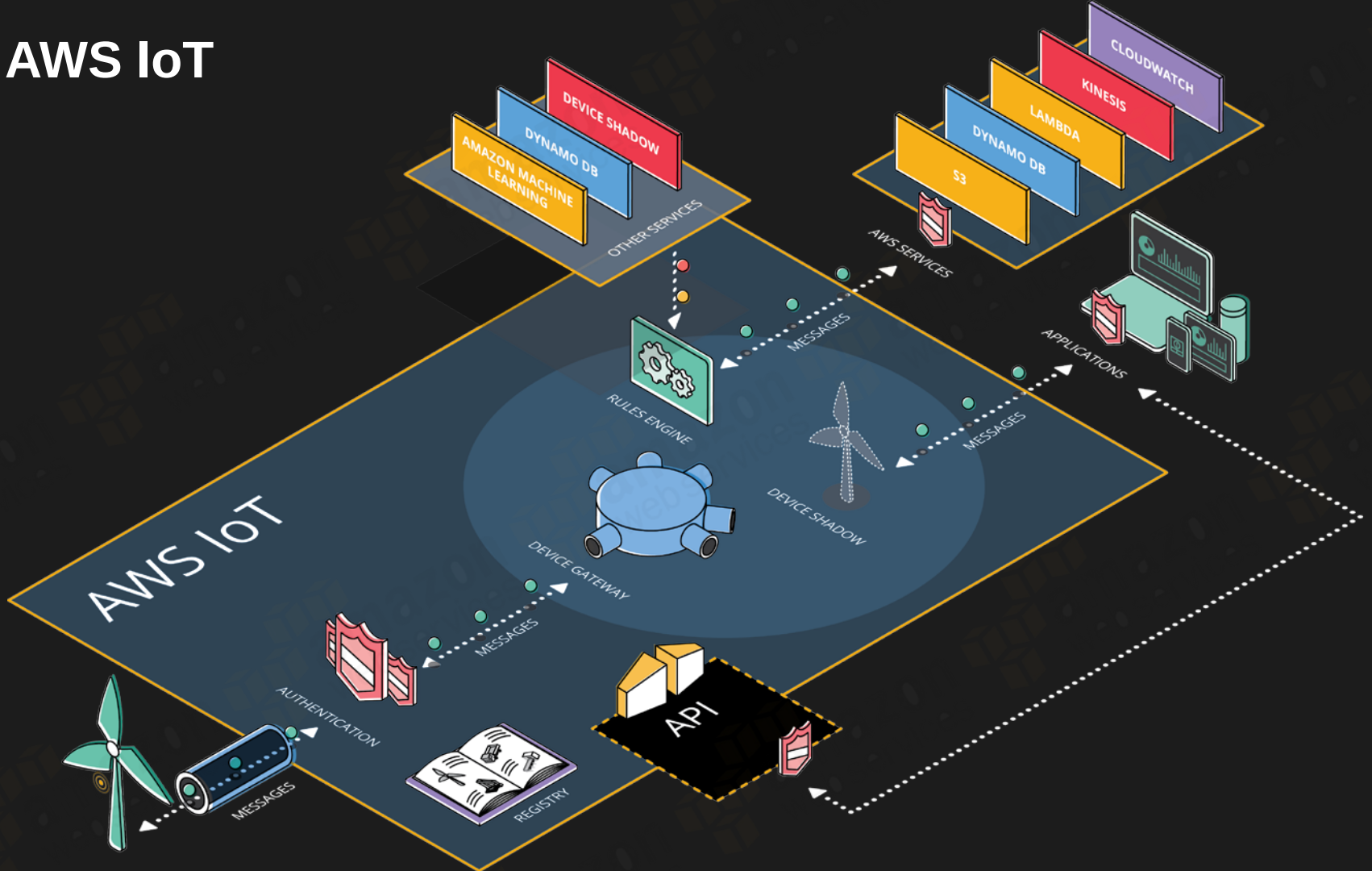
Deep Dive - AWS IoT

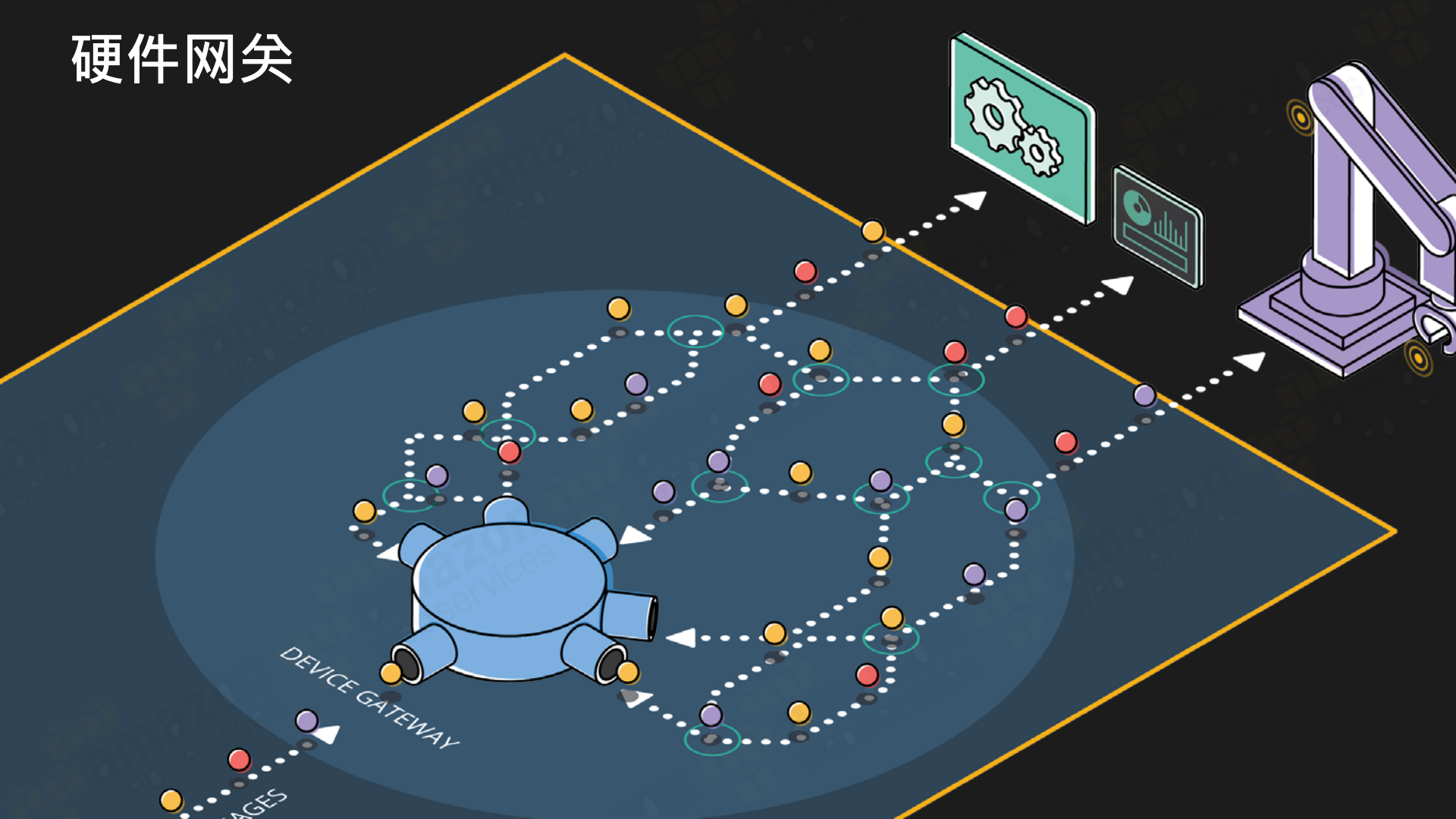
吴宗承 | 解决方案架构师

© 2016, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

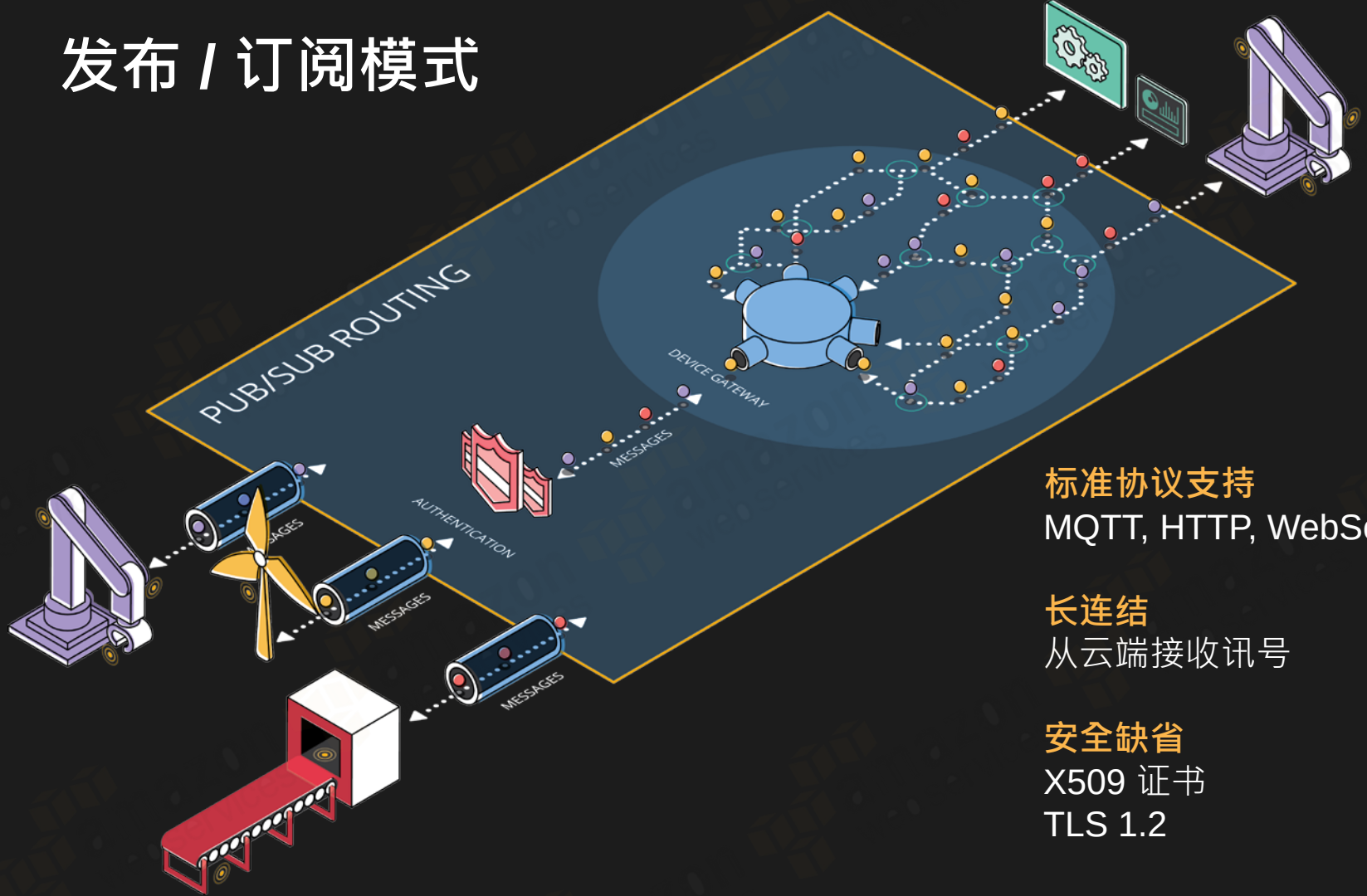
AWS IoT


DEVICE SHADOW
DYNAMO DB
AMAZON MACHINE LEARNING
OTHER SERVICES
CLOUDWATCH
KINESIS
LAMBDA
DYNAMO DB
S3
AWS SERVICES
APPLICATIONS
MESSAGES
MESSAGES
RULES ENGINE
DEVICE SHADOW
DEVICE GATEWAY
AWS IoT
MESSAGES
AUTHENTICATION
API
MESSAGES
REGISTRY

硬件网关



发布 / 订阅模式



标准协议支持
MQTT, HTTP, WebSockets

长连结
从云端接收讯号

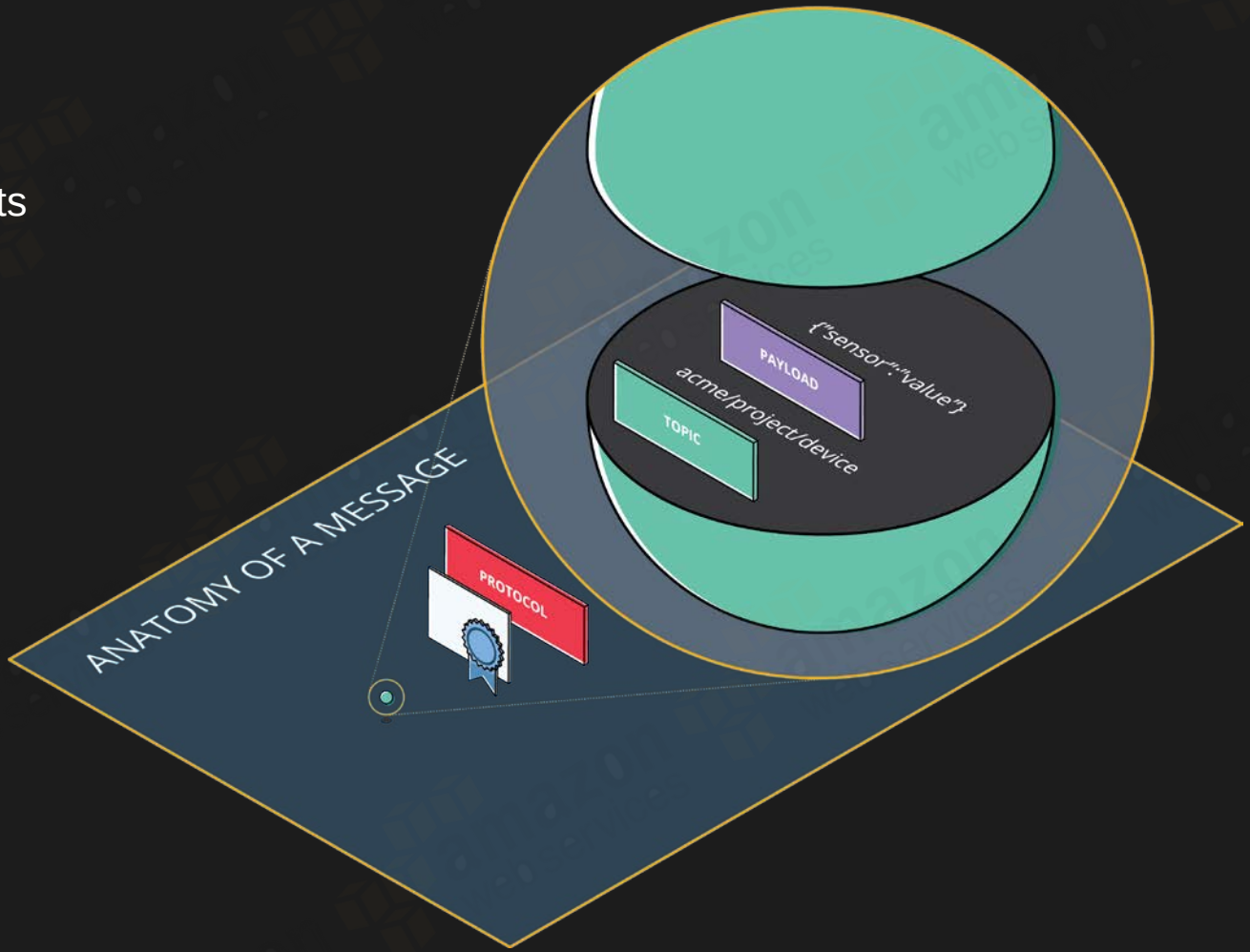
安全缺省
X509 证书
TLS 1.2

传感器讯息

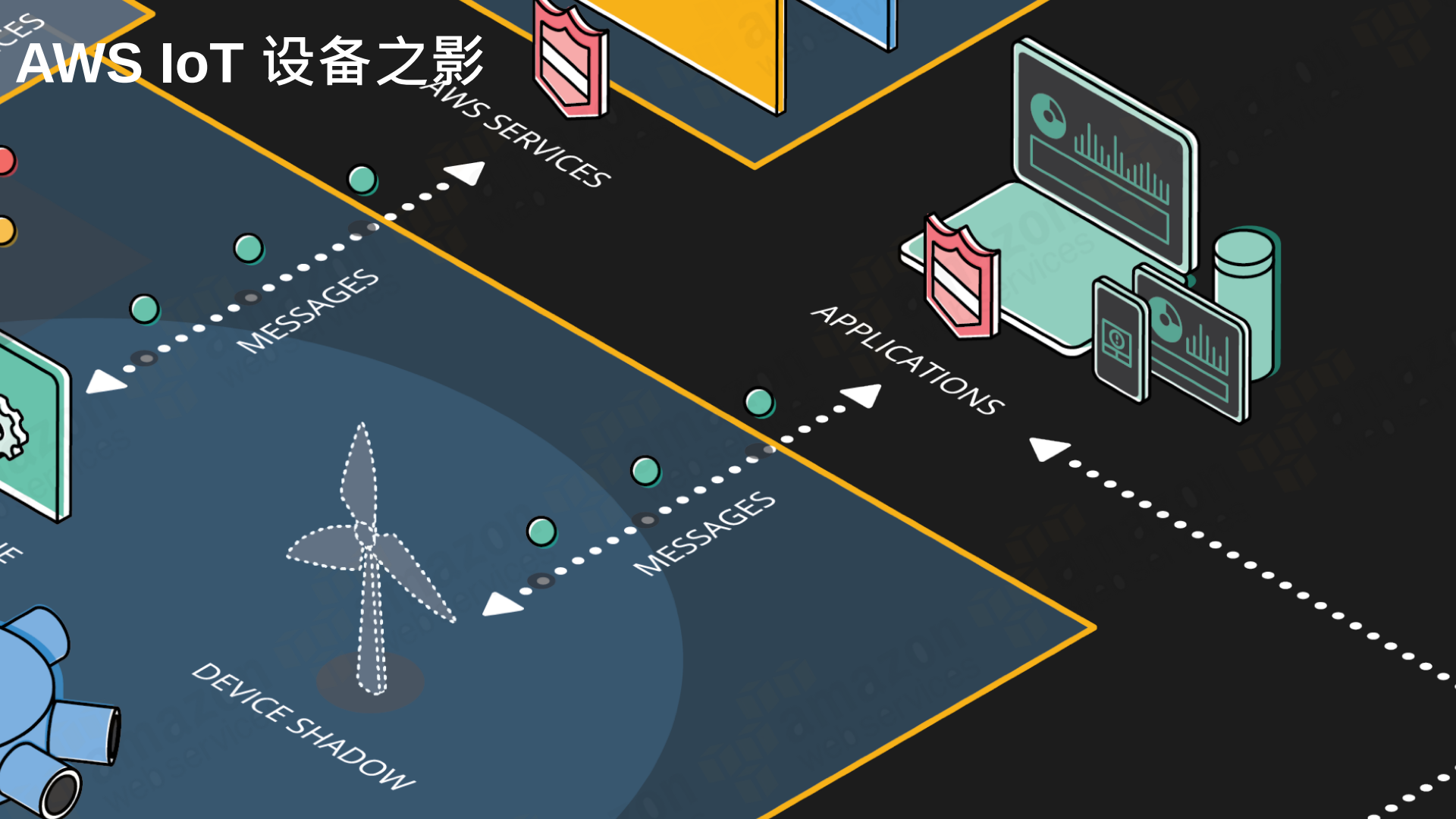
标准协议支持
MQTT, HTTP, WebSockets

主题/渠道
消息路由

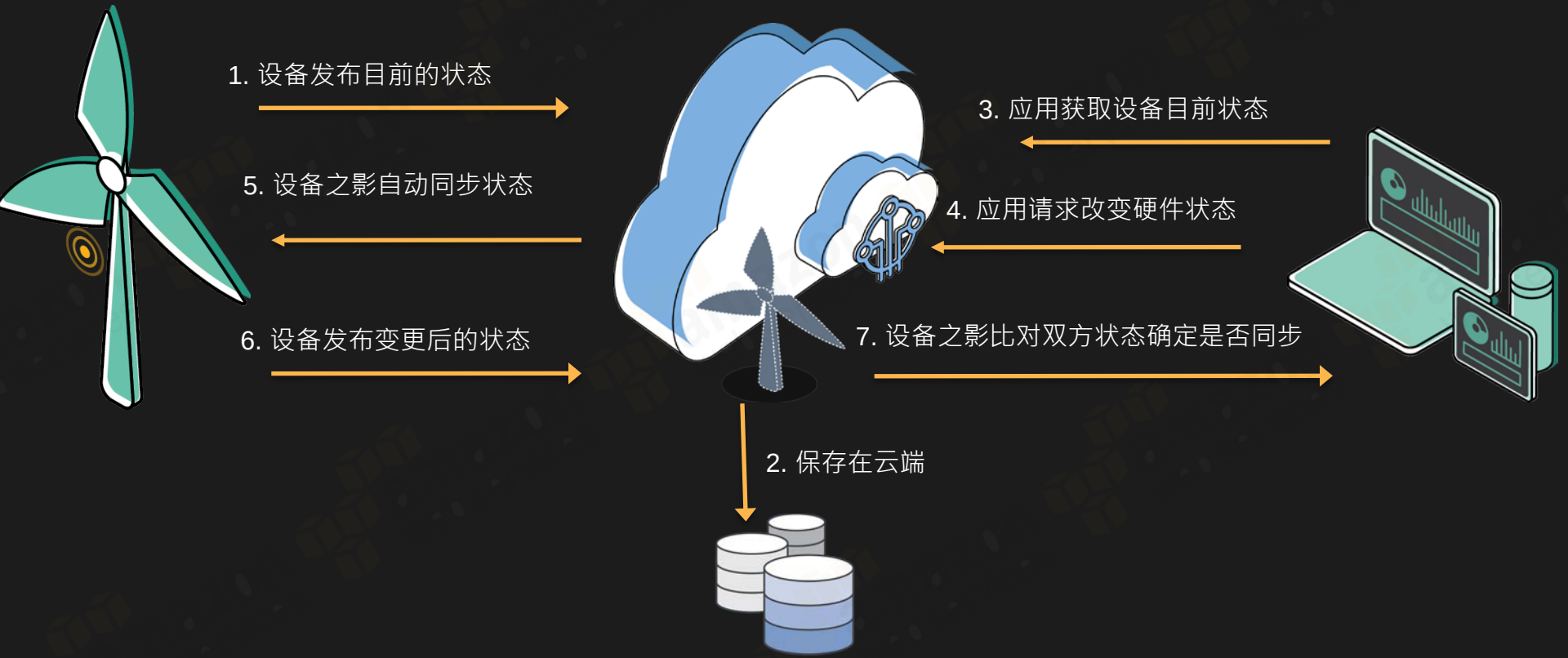
Payload (JSON)
可自定义的内容, 弹性



AWS IoT 设备之影


AWS SERVICES
MESSAGES
APPLICATIONS
MESSAGES
DEVICE SHADOW

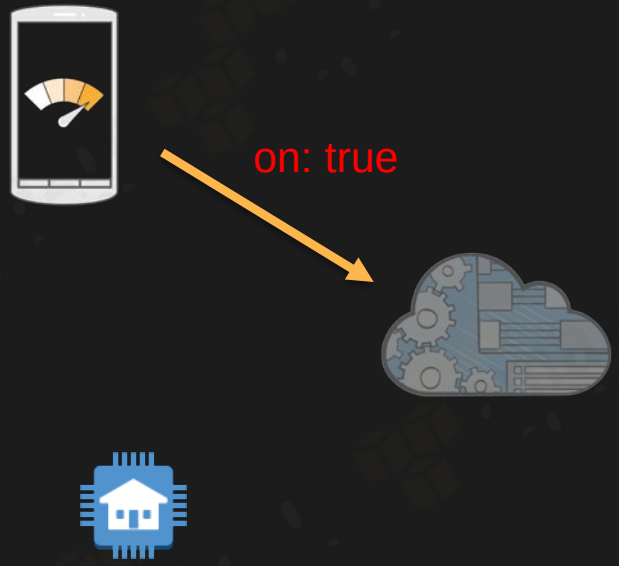
AWS IoT 设备之影


1. 设备发布目前的状态
5. 设备之影自动同步状态
6. 设备发布变更后的状态
3. 应用获取设备目前状态
4. 应用请求改变硬件状态
7. 设备之影比对双方状态确定是否同步
2. 保存在云端

on: false

‘$aws/things/abot-two/shadow/update’

```
{
        “state” : {
                "reported": {
                        "on": false
                }
        }
}
```



‘$aws/things/abot-two/shadow/update’

```
{
        "state" : {

                "desired": {
                        "on": true
                },
                "reported": {
                        "on": false
                }
        }
}
```

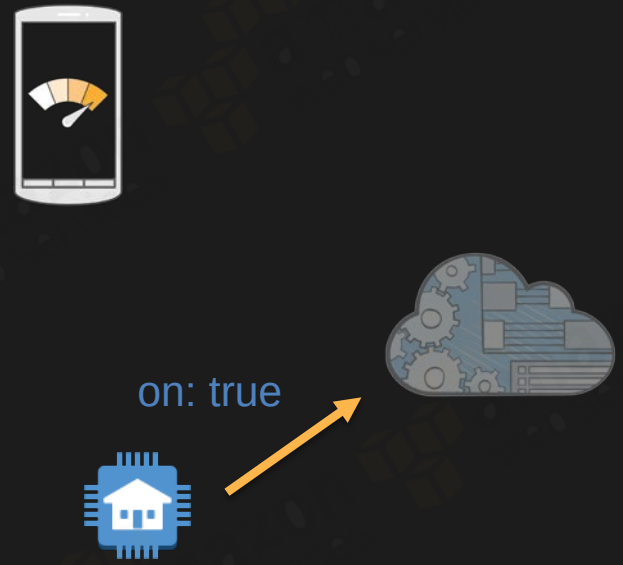
on: true

‘$aws/things/abot-two/shadow/update’

```
{
        "state" : {

                "desired": {
                        "on": true
                },
                 "reported" : {
                        "on": false
                },
                {
                 "delta": {
                        "on": true
                }
        }
}
```

out of sync



‘$aws/things/abot-two/shadow/update’

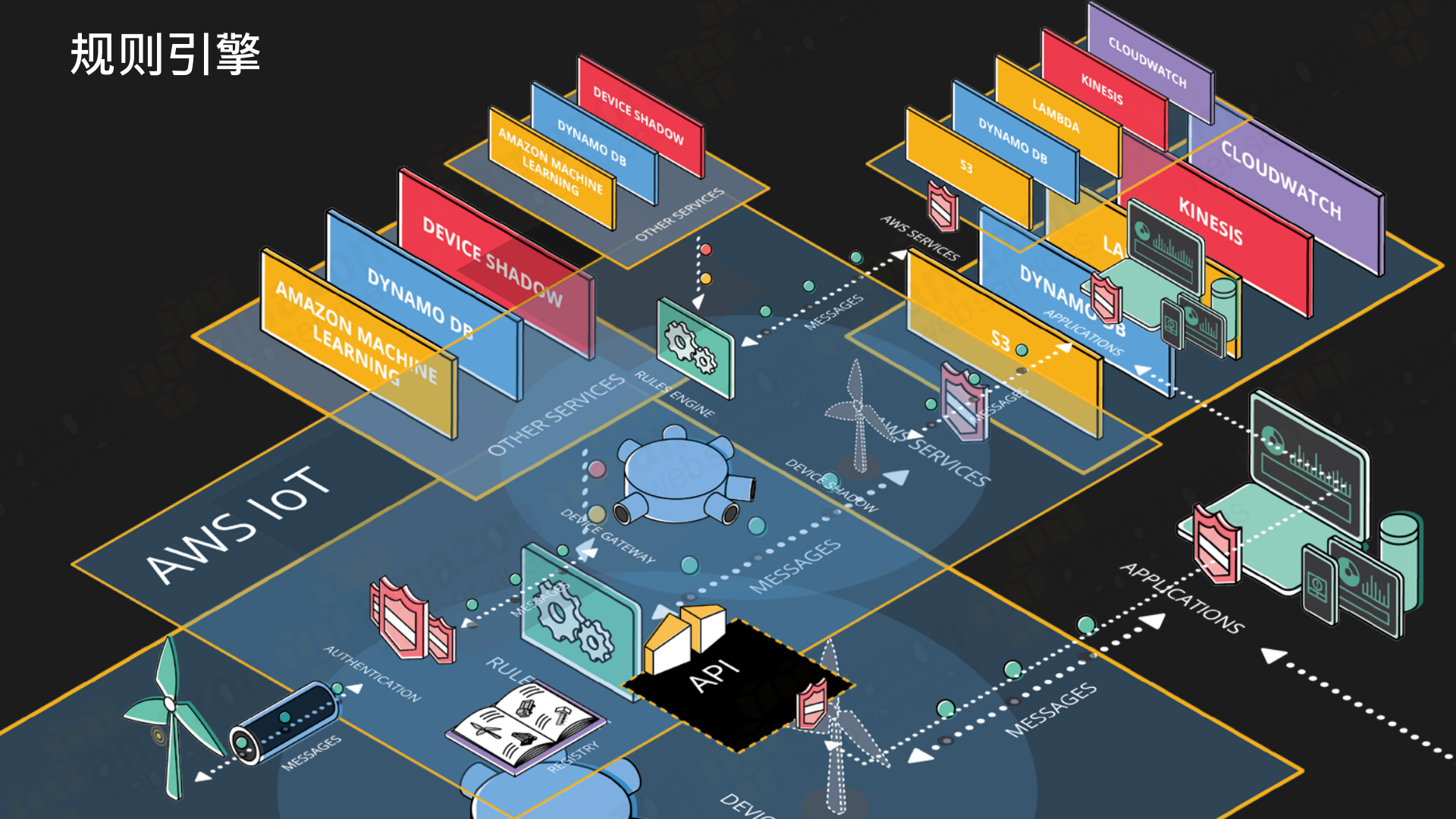
```
{
        "state" : {

                "desired": {
                        "on": true
                },
                "reported": {
                        "on": true
                }
        }
}
```

数据收取后?

1. 如何从我的数据中提取它的价值?
2. 如何展现我的数据?
3. 如何能未卜先知?

规则引擎


AWS IoT
DEVICE SHADOW
DYNAMO DB
AMAZON MACHINE LEARNING
OTHER SERVICES
CLOUDWATCH
KINESIS
LAMBDA
S3
AWS SERVICES
APPLICATIONS
RULES ENGINE
MESSAGES
DEVICE GATEWAY
AUTHENTICATION
API
REGISTRY

规则引擎：从数据中提取其价值!

- 过滤消息
- 转换消息的结构
- 基于消息做出响应
- 将消息发布至其他 主题
- 将消息发布至其他的系统
- 依据数据的趋势做出预测

什么是规则?

ruleArn: arn:aws:iot:<region>:<account_id>:rule/<uniq_rule_name>

rule:

- ruleName: human-readable name
- description: human-readable description
- ruleDisabled: true/false
- sql: e.g. SELECT * FROM ‘pm/topic’
- actions:
  - action 1:
  - action 2:

什么是规则?

- ruleName: human-readable name
- description: human-readable description
- ruleDisabled: true/false
- **sql: e.g. SELECT * FROM ‘pm/topic’**
- actions:
  - action 1:
  - action 2:

什么是规则?

- ruleName: human-readable name
- description: human-readable description
- ruleDisabled: true/false
- sql: e.g. SELECT * FROM ‘pm/topic’
- **actions:**
  - **action 1:**
  - **action 2:**

规则举例

```
{
   "rule": {
         "sql": "SELECT * AS message FROM 'sensors/#'",
         "description": "Store all sensor data into dynamodb and firehose",
         "actions": [{
            "dynamoDB": {
                "tableName": "sensor_data",
                "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDB",
                "hashKeyField": "sensor_id",
                "hashKeyValue": "${topic(2)}",
                "rangeKeyField": "timestamp“
                "rangeKeyValue": "${timestamp()}",
             }
         }, {
            "firehose": {
               "roleArn": "arn:aws:iam::123456789012:role/aws_iot_firehose",
               "deliveryStreamName": "my_firehose_stream"
            }
         }]
    }
}
```

规则引擎: 过滤与转换

- 过滤消息
- 转换消息的结构
- 基于消息做出响应
- 将消息发布至其他 主题
- 将消息发布至其他的系统
- 依据数据的趋势做出预测

规则引擎: 过滤与转换

- SQL
  - SELECT * FROM *topic* WHERE *condition*
  - SELECT status.space_id AS room_id FROM *'iot/tempSensors/#'* WHERE *temp > 50*
- 功能 (用于 SELECT 或 WHERE 内)
  - 字符串操作 (支援正則表達式)
  - 数学运算
  - Context-based 辅助函数
  - 加密计算
  - UUID, timestamp, rand, 等.

规则引擎: 响应与重新发布

- 过滤消息
- 转换消息的结构
- 基于消息做出响应
- 将消息发布至其他 主题
- 将消息发布至其他的系统
- 依据数据的趋势做出预测

规则引擎: 响应与重新发布

```
"rule": {
      "ruleName": "republishPredictions”,
      "ruleDisabled": false,
      "sql": "SELECT no AS id, cycle AS c_number FROM 'pm/topic'  WHERE failure = 1",
      "description": "Republish ids of engines that are predicted to fail within 30 cycles",
      "actions": [
        {
          "republish":  {
              "topic": "pm/failures",
              "roleArn": "arn:aws:iam::012345678901:role/iot-actions-role”
           }
        }
    ],
}
```

规则引擎: 响应与重新发布

```
"rule": {
      "ruleName": "republishPredictions”,
      "ruleDisabled": false,
      "sql": "SELECT no AS id, cycle AS c_number FROM 'pm/topic'  WHERE failure = 1",
      "description": "Republish ids of engines that are predicted to fail within 30 cycles",
      "actions": [
        {
          "republish":  {
             "topic": "pm/failures",
             "roleArn": "arn:aws:iam::012345678901:role/iot-actions-role”
           }
        }
    ],
}
```

规则引擎: 将消息发送到其他系统

- 过滤消息
- 转换消息的结构
- 基于消息做出响应
- 将消息发布至其他 主题
- 将消息发布至其他的系统
- 依据数据的趋势做出预测

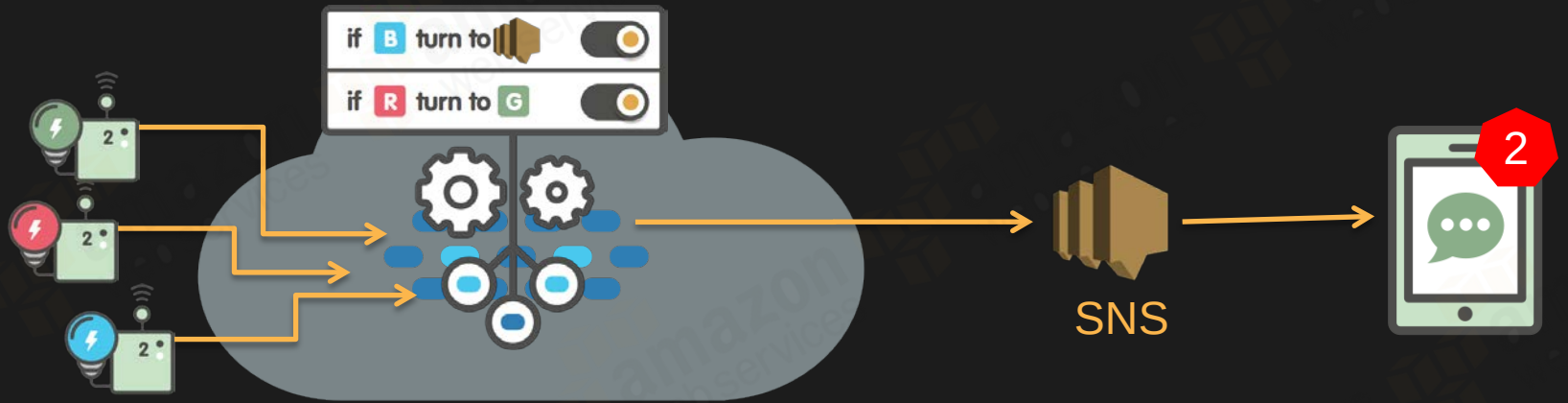
规则引擎: 将消息发送到其他系统

- 触发Lambda
- 存放对象在S3
- 新增或变更DynamoDB的数据
- 发布SNS 到主题
- 将消息送往流式存储
- 发布至 Firehose
- 或是重新发布到其他IoT 主题
- CloudWatch 监控
- 发布到搜寻引擎
- 写入SQS 消息对列

规则引擎: 将数据导入AWS存储

- 新增或修改DynamoDB 的表
- 发布至Firehose
- 把档案放入S3 的存储桶
- 发布至 Amazon ES
- 发布至Amazon Kinesis 数据流

AWS IoT 规则引擎 & Amazon SNS



推送消息通知

Apple APNS 端点, Google GCM Endpoint, Amazon ADM 端点, Windows WNS

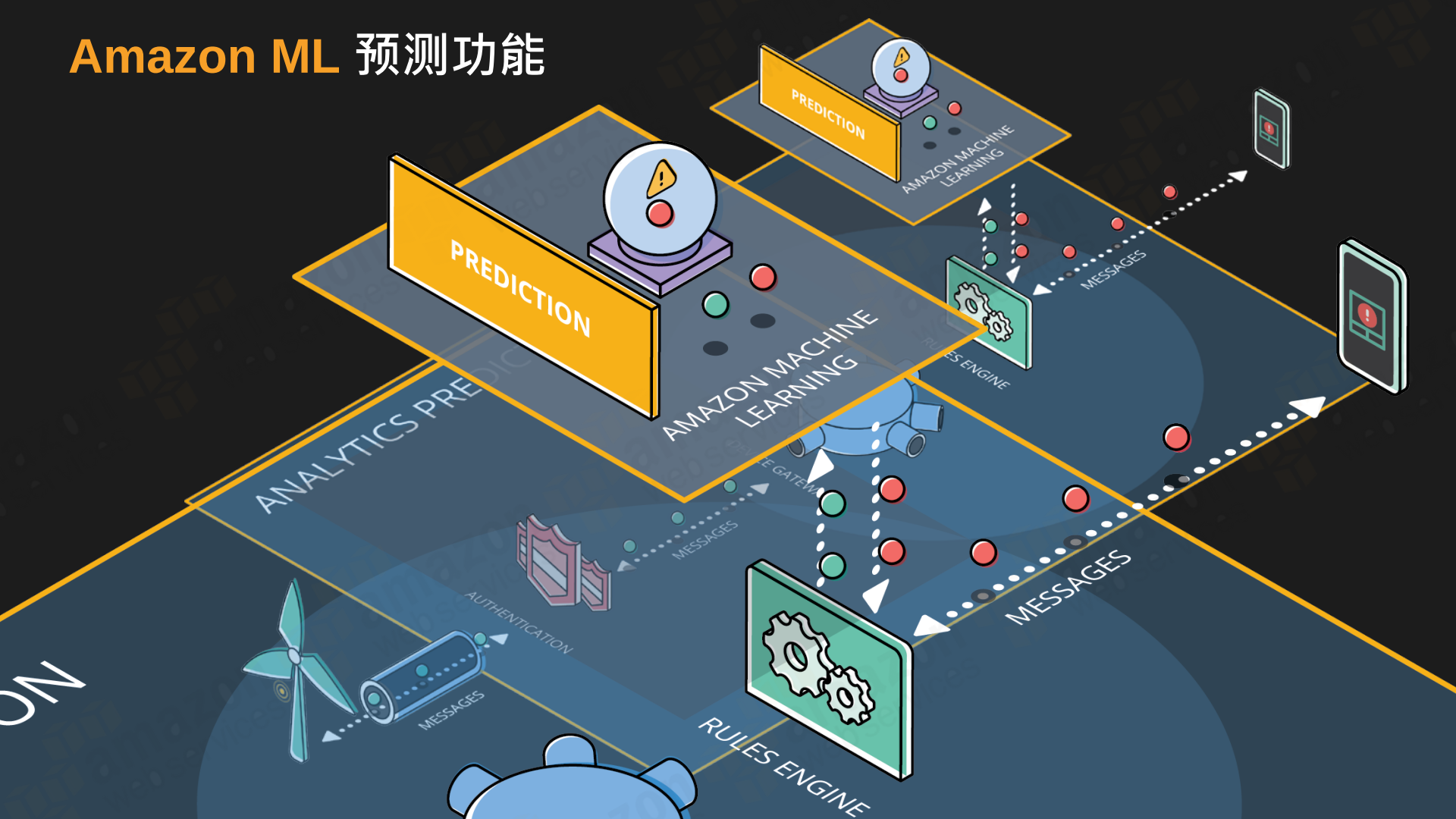
Amazon SNS -> HTTP 端点(或 SMS 或 Email)

基于HTTP 协议呼叫第三方或自己后台的端点

规则引擎：预知未来

- 过滤消息
- 转换消息的结构
- 基于消息做出响应
- 将消息发布至其他 主题
- 将消息发布至其他的系统
- 依据数据的趋势做出预测

Amazon ML 预测功能



未卜先知？！

机器学习赋予计算机学习的能力

机器学习算法:

监督学习 ← 训练项目

- 分类算法 ← 0/1, Yes/No, True/False
- 回归算法 ← 2个或多个变量之间的关系与强度?

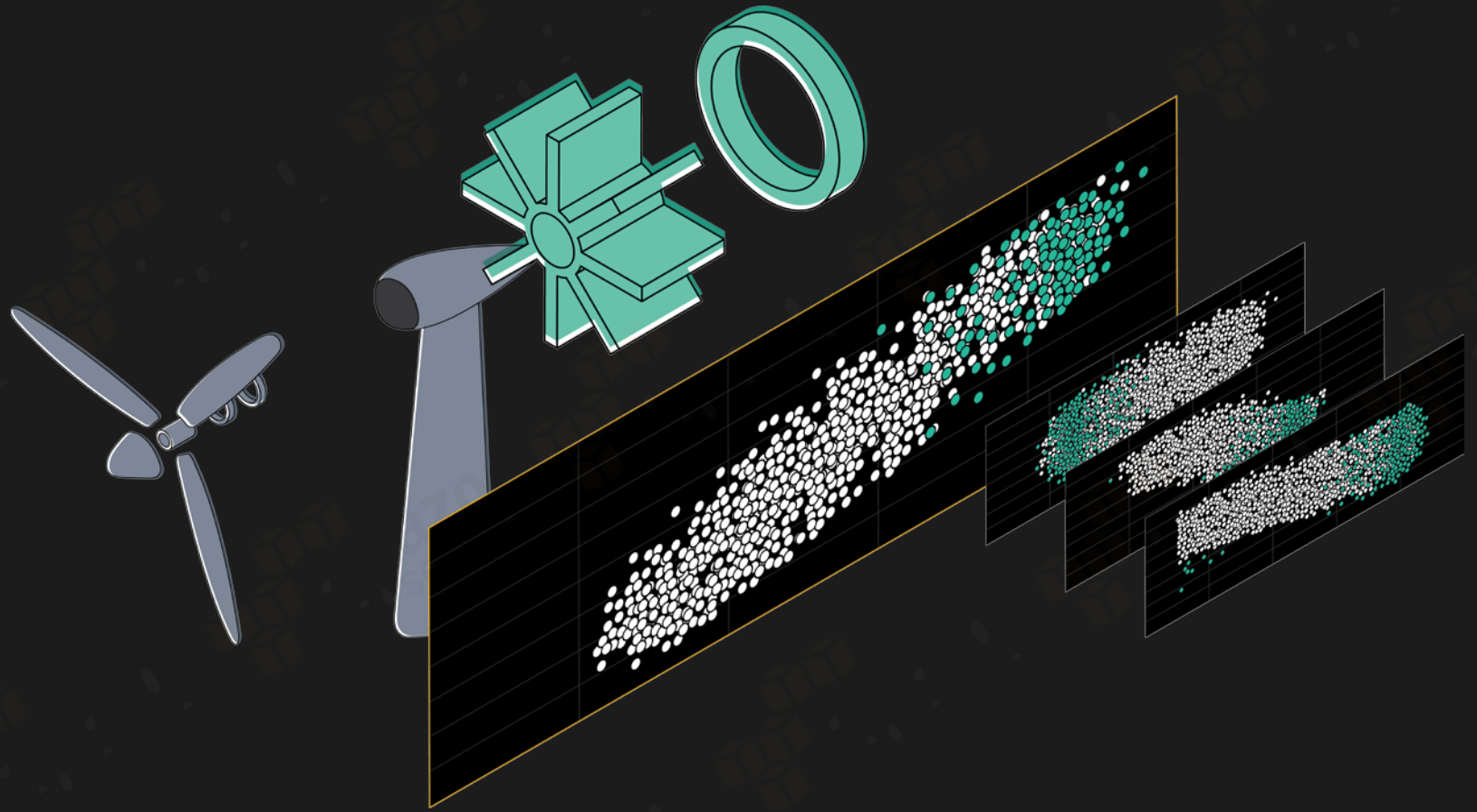
无监督学习 ← 自我学习

- 聚类算法 ← 广泛用于生物科技, 市场行销等

基本流程

1. 利用IoT 规则引擎 蒐集设备数据于 S3
2. 利用S3 的数据来训练 Amazon ML 的模型
3. 利用 IoT 规则 来:
    - 对实时数据做预测
    - 利用CloudWatch 的监控矩阵做报警
4. 利用IoT 规则 发出故障讯息的资讯

工业场景：预测硬件的故障！

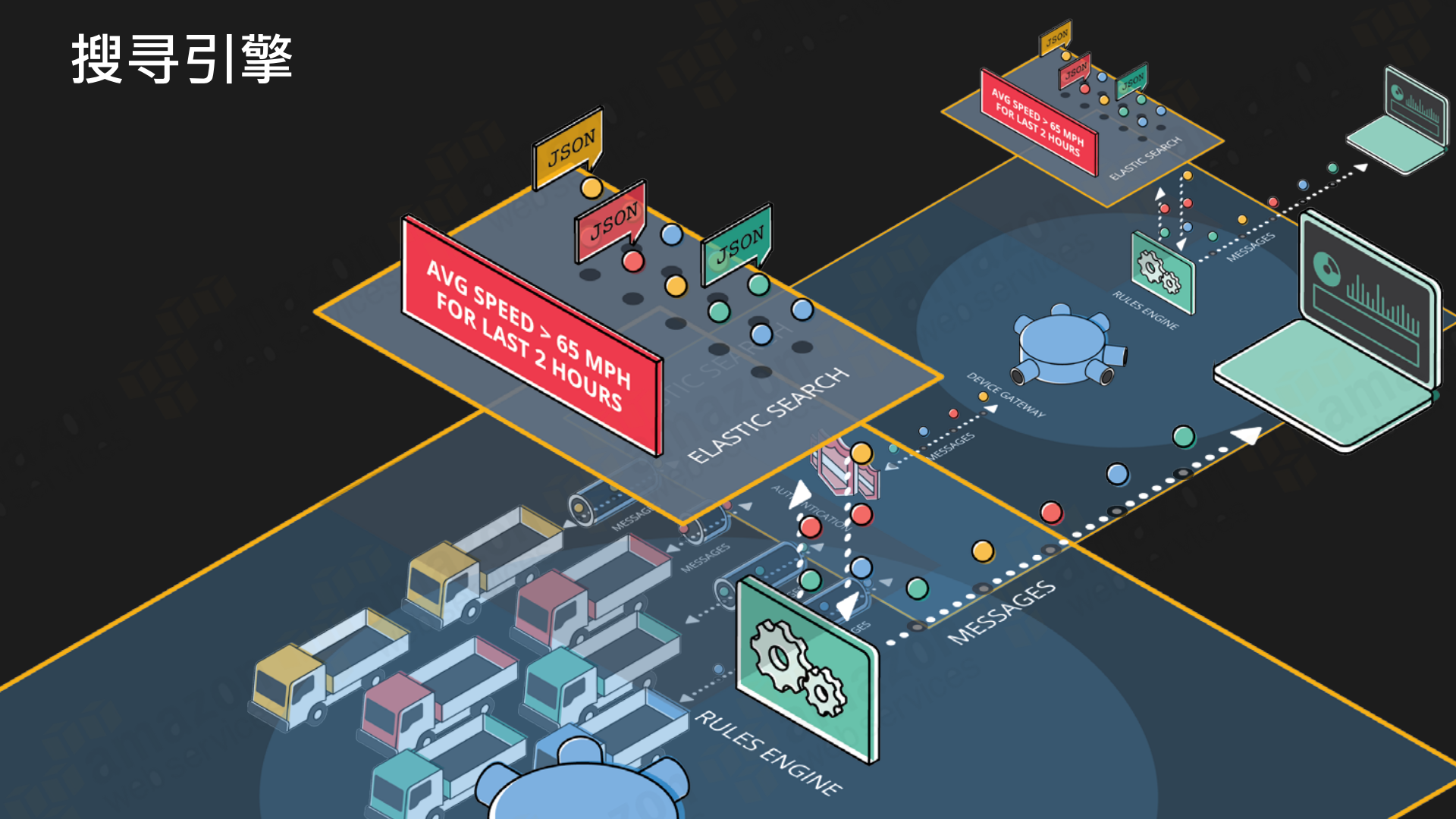


预测的重要性

危机/成本 的比例上取得好的平衡:

1. 预测发生的时间过早 = 造成一定的浪费
2. 不做预测 = 可能造成的损失更巨大

搜寻引擎


JSON
JSON
JSON
AVG SPEED > 65 MPH FOR LAST 2 HOURS
ELASTIC SEARCH
RULES ENGINE
DEVICE GATEWAY
MESSAGES
MESSAGES

演示 实时视觉化展现

语音的力量：Alexa

- **Alexa** – 利用Alexa Skills Kit, 你可以轻松的建立语音应用.
- **AWS IoT** 管理所有硬件的连接以及整合ASK与其他后台系统.
- 利用 **AWS Lambda** 建立你的语音应用. 只需写简单的代码上传即可.
- **Alexa 语音服务** 能利用语音驱动的能力连接你的硬件.

总结

- 提取数据的价值: 规则引擎
- 可视化你的数据: 规则引擎 + AWS 服务
- 利用设备之影的好处与Amazon 机器学习

amazon web services