

AWS IoT Services

이세현

IoT Specialist 솔루션즈 아키텍트
AWS



Agenda

- IoT 기술 Trend
- FreeRTOS - edge에서 디바이스를 구축하고 Cloud에 연결
- IoT Core – 디바이스 연결 관리 및 Cloud application과의 상호작용
- AWS IoT Greengrass - Cloud의 서비스와 기능을 edge까지 확장
- AWS Partner Device Catalog

IoT 기술 Trend



IoT 기술 Trend



다른 device와 상호 작용



Cloud와의 연결 및 통신



Cloud의 기능들을 edge로 확장



지속적으로 학습 및 진화

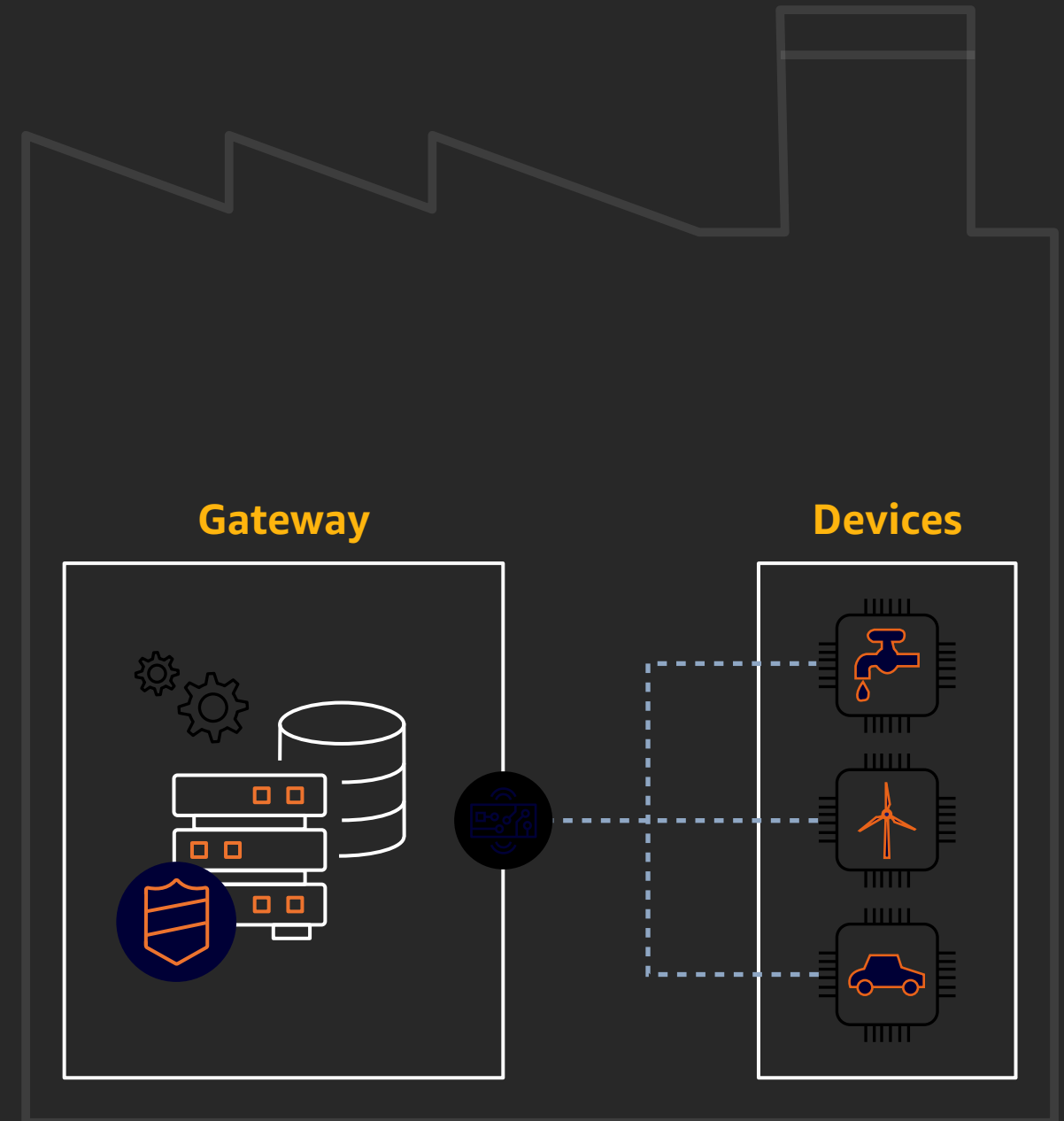
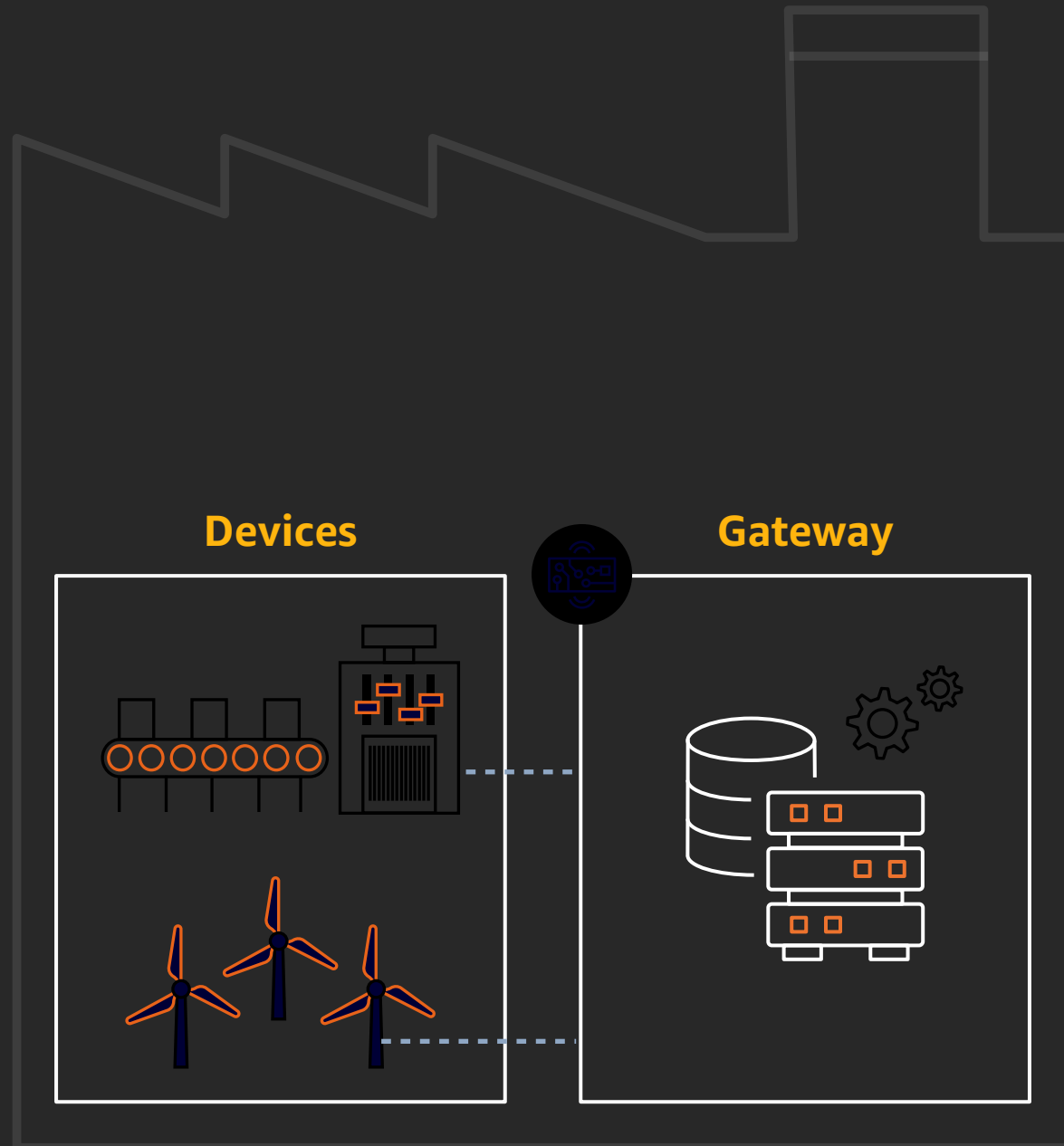


주변 데이터를 활용하여 추론

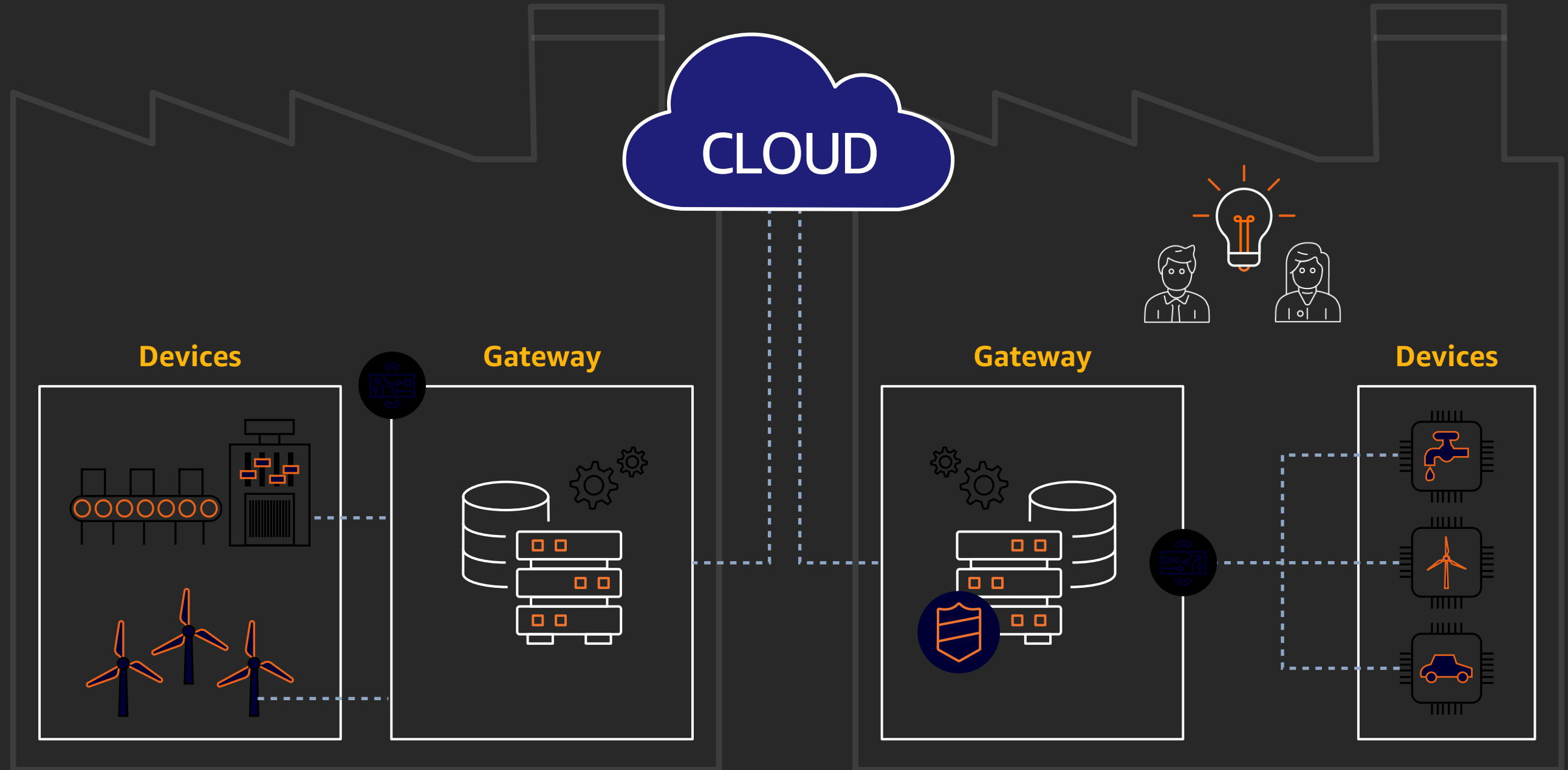


보안

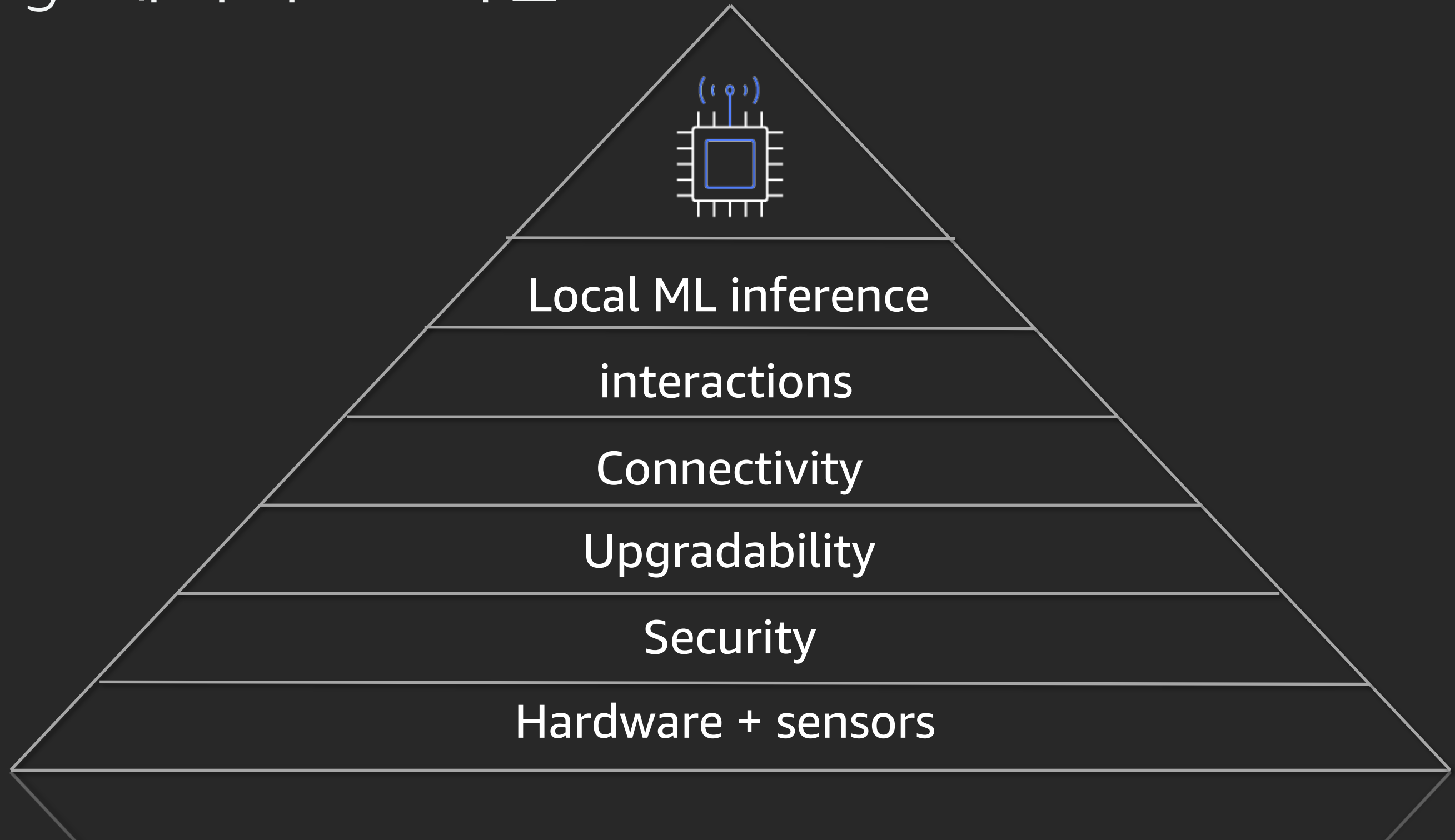
Edge data



Edge 에서 cloud로의 데이터 흐름



Edge에서의 IoT 기술





Analytics
Services

어떻게 IoT 데이터에서 가치를 어떻게 추출합니까?



Connectivity
& Control
Services

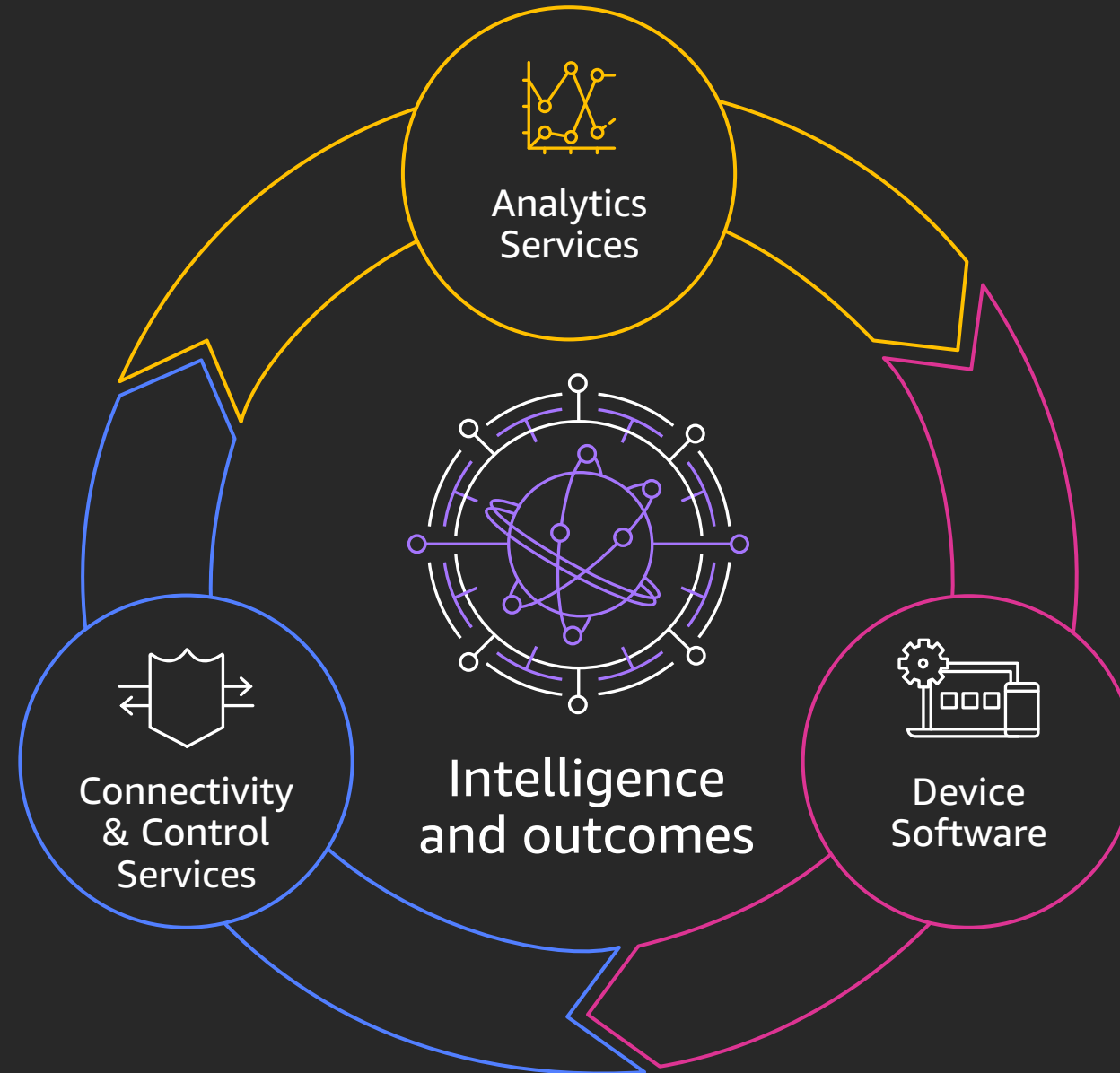
대규모의 장치를 제어, 관리 및 보호하려면
어떻게 해야합니까?

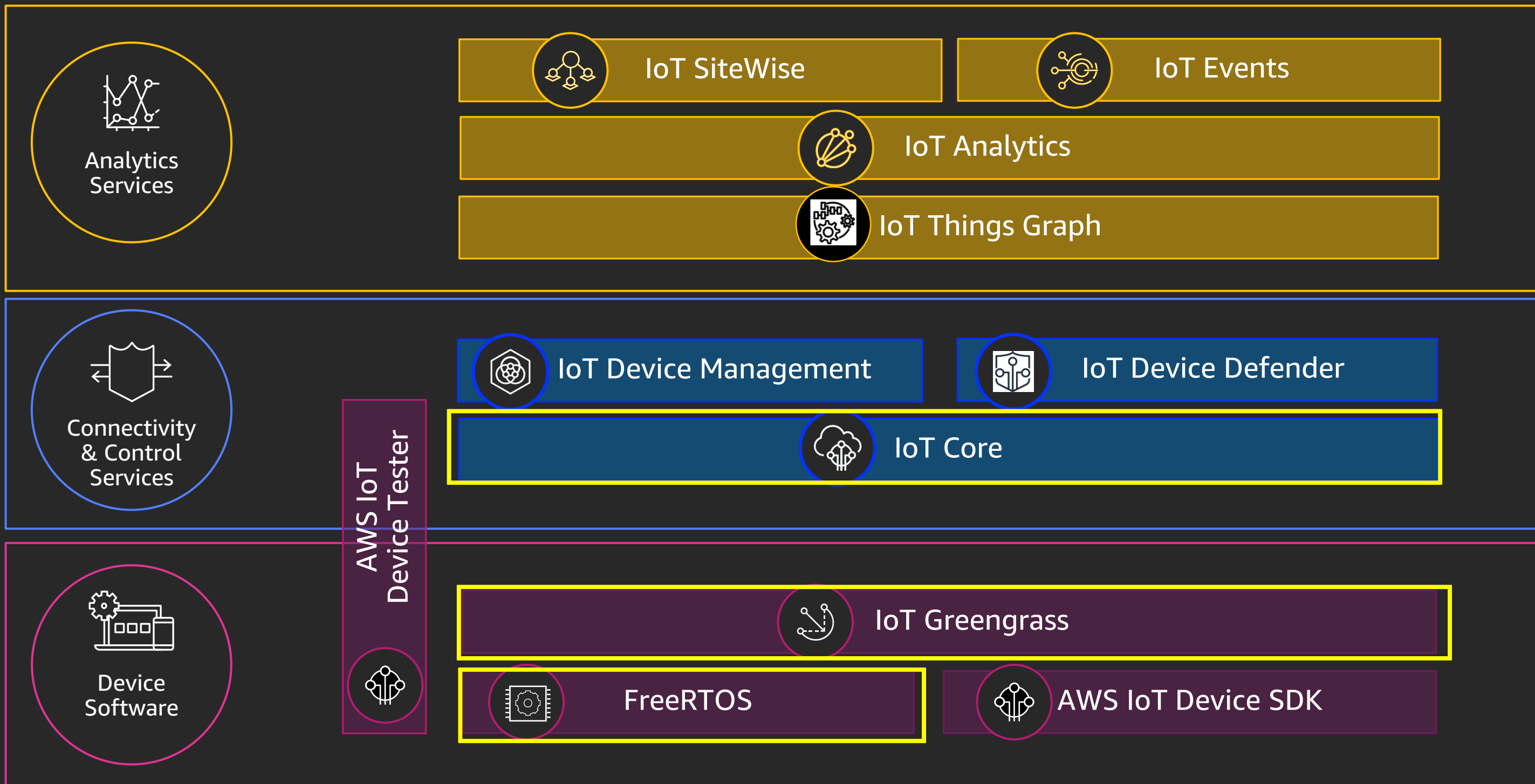


Device
Software

장치를 연결하고 edge에서 작동하려면
어떻게 해야합니까?

IoT 선순환



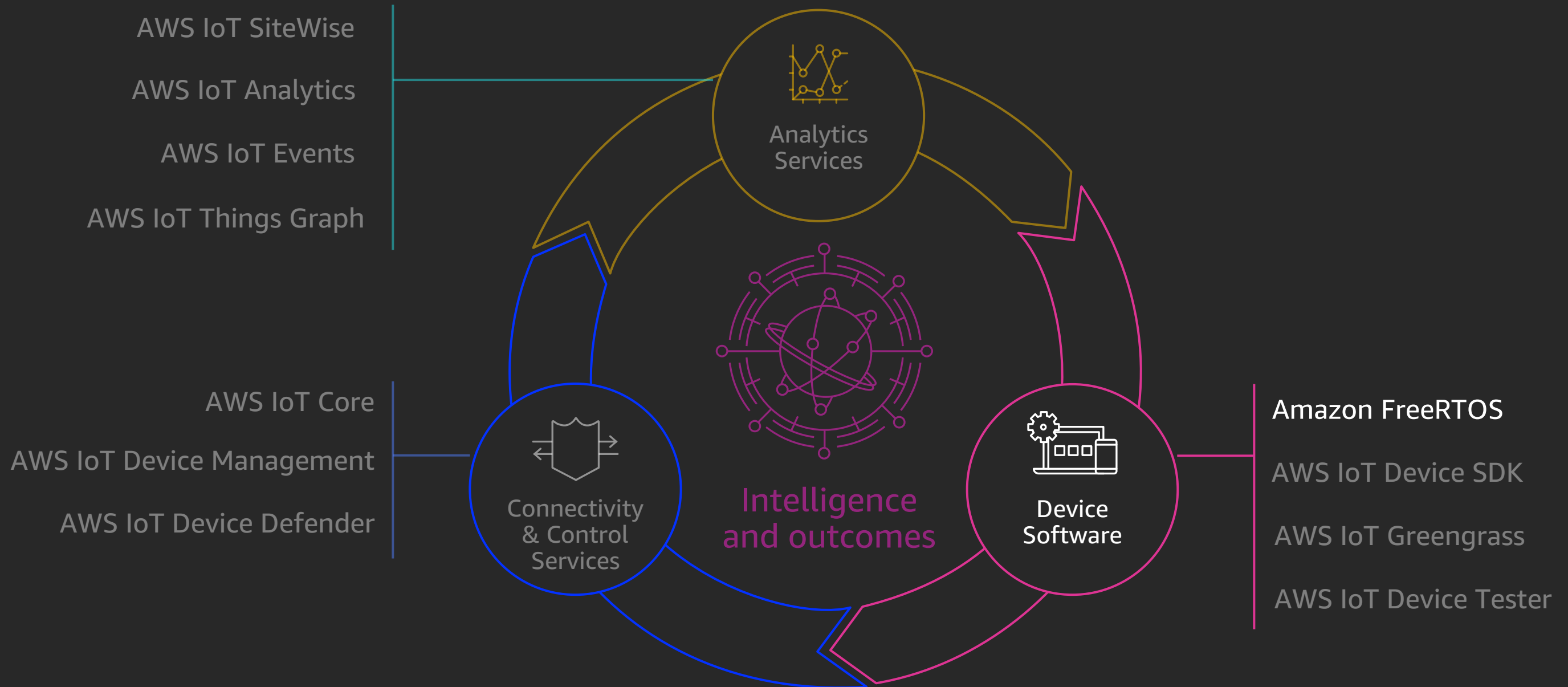


FreeRTOS

- edge에서 디바이스를 구축하고 Cloud에 연결



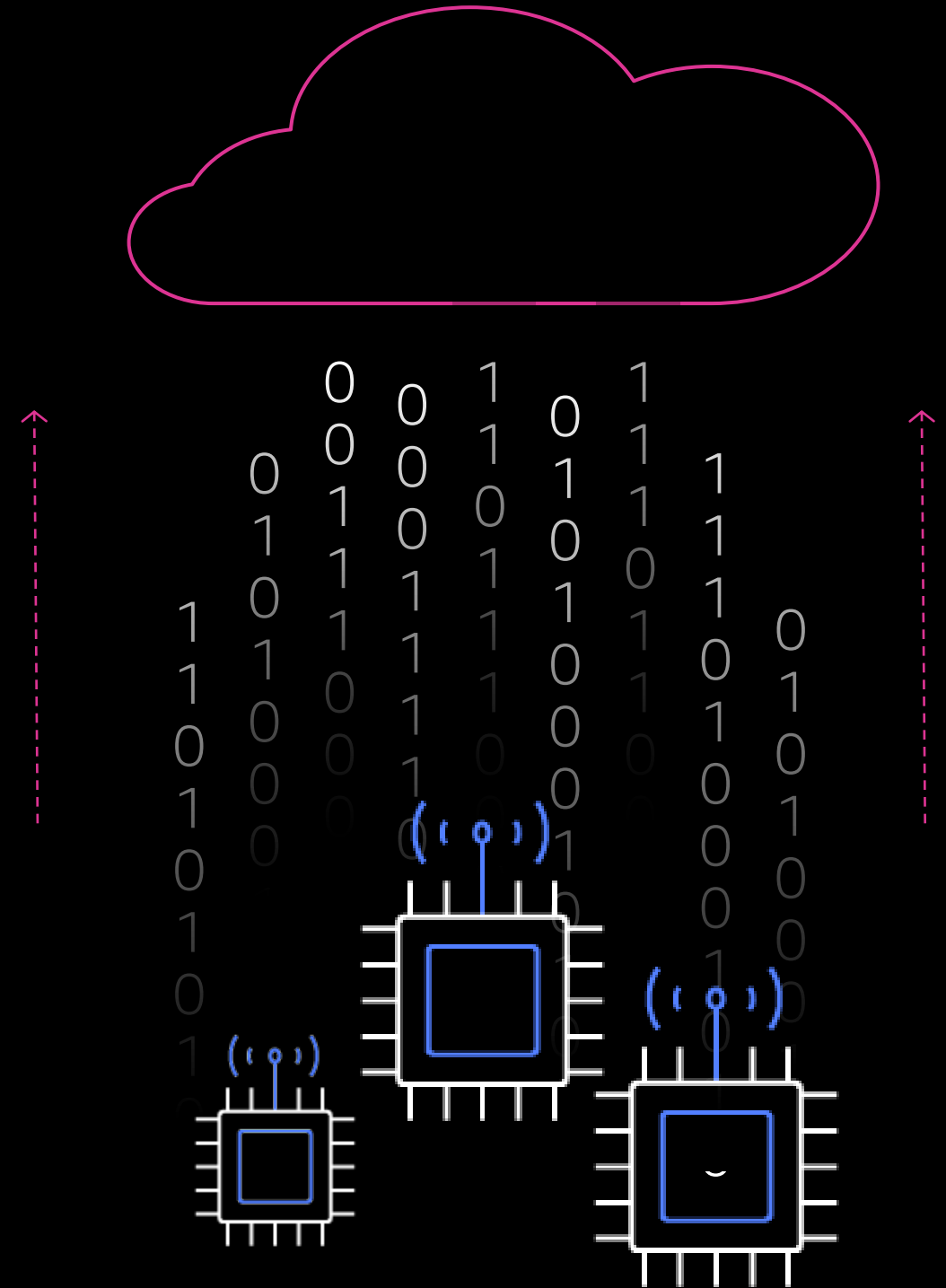
장치를 연결하고 edge에서 작동하려면 어떻게 해야합니까?



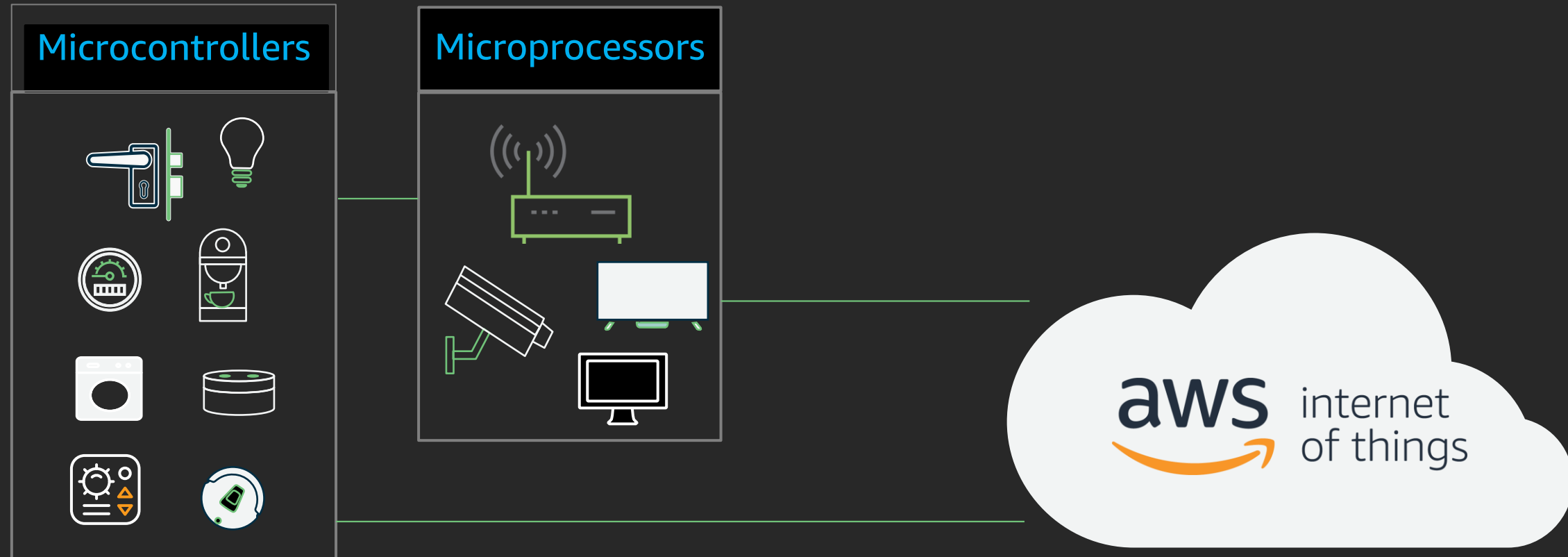
AWS IoT와 함께 작동하는
edge에서 디바이스를
구축하려면 어떻게 해야
합니까?



Device
Software

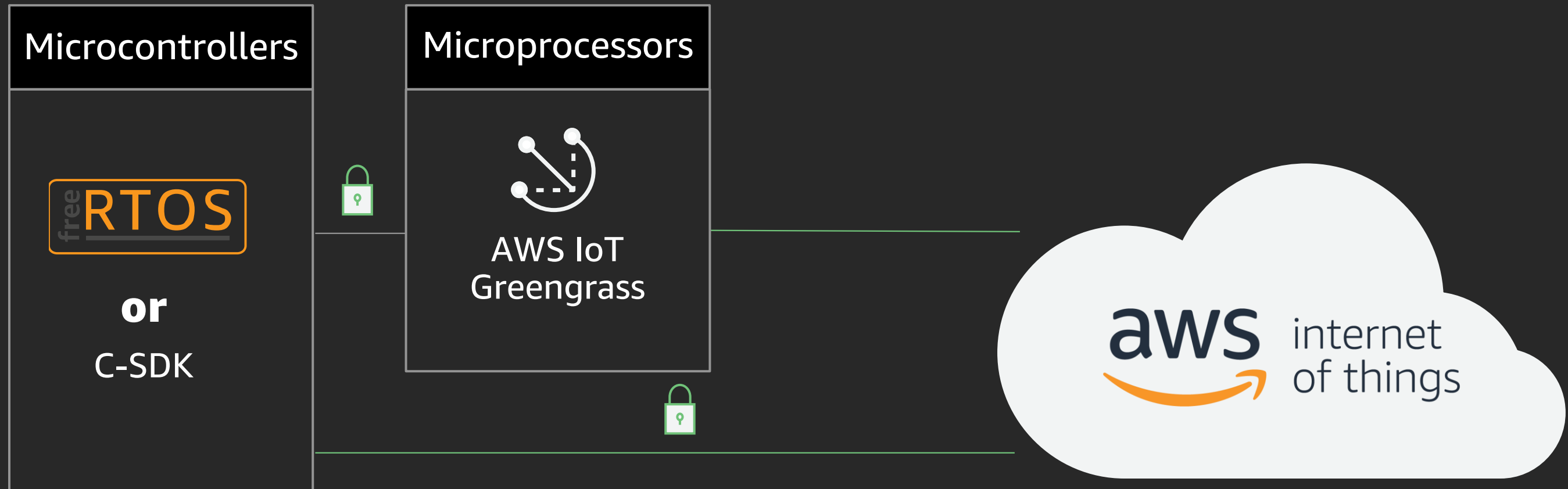


수십억 개의 IoT 장치에 전력을 공급하는 마이크로 컨트롤러와 마이크로 프로세서



Device
Software

AWS IoT와 함께 작동하는 디바이스를 쉽게 구축

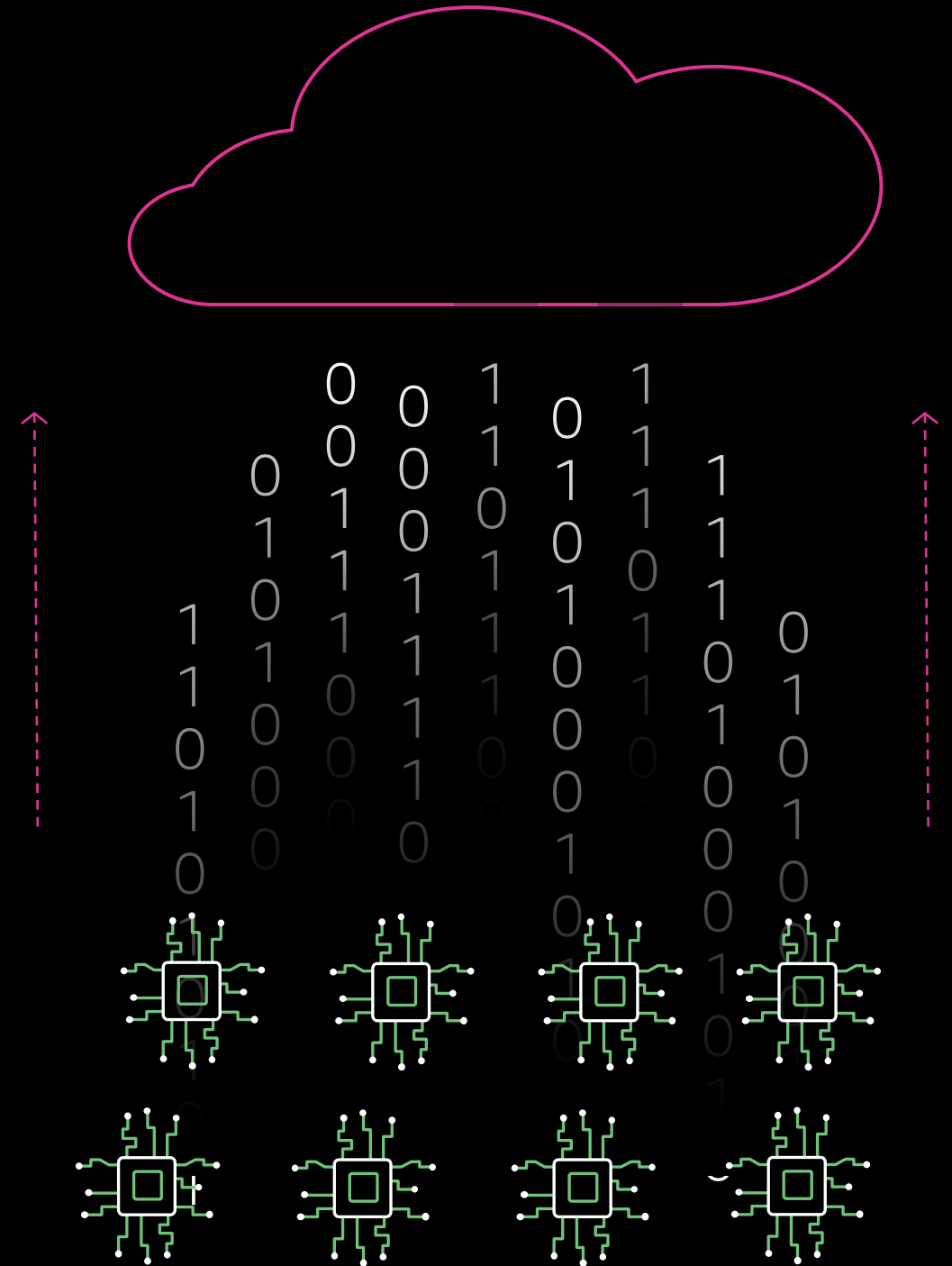


Device
Software

AWS와 함께 작동하는
마이크로 컨트롤러 기반
디바이스를 구축하려면
어떻게 해야 할까요?



Device
Software



FreeRTOS



15 년 이상, 신뢰할 수 있고 널리 배포됨

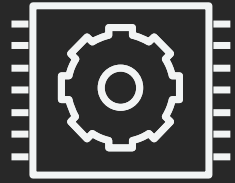
RISC-V 및 Arm v8-M을 포함한 40개 이상의 지원 아키텍처

광범위한 생태계 지원

무료 및 오픈 소스

MIT 오픈 소스 라이선스

스트림 및 메시지 버퍼를 사용하여
향상된 프로세스 간 통신 (IPC) 기능 지원



FreeRTOS는 IoT endpoint를 지원합니다.



MCU가 "지능형" 워크로드를 처리할 수 있습니까?

- MCU가 GHz 속도에 도달함
 - NXP 는 Arm Techcon 2019에서 i.MXRT1170 를 발표
 - 최대 1GHz의 clock 속도
- 머신 러닝 프레임 워크는 MCU 클래스 장치를 대상으로 합니다
 - Keyword spotting/wake word detection
 - TinyML
 - Tensorflow Lite

➔ MCU는 점점 더 복잡해지는 워크로드를 처리 할 수 있습니다

FreeRTOS에서 secure elements 사용

Secure Element를 사용하면 암호화 키의 변조 방지가 가능합니다.

개방형 표준 PKCS#11 인터페이스를 사용하여 호환성 극대화

고객의 물리적 위치에서 떨어진 위치에 디바이스를 설치할 수 있습니다.

마이크로 컨트롤러에서 암호화 기능을 오프로드하여 메모리 공간을 확보하고 전력 소비를 줄입니다.



Secure Element를 사용하지 않을 경우

MCU Hardware

User (Application) Code

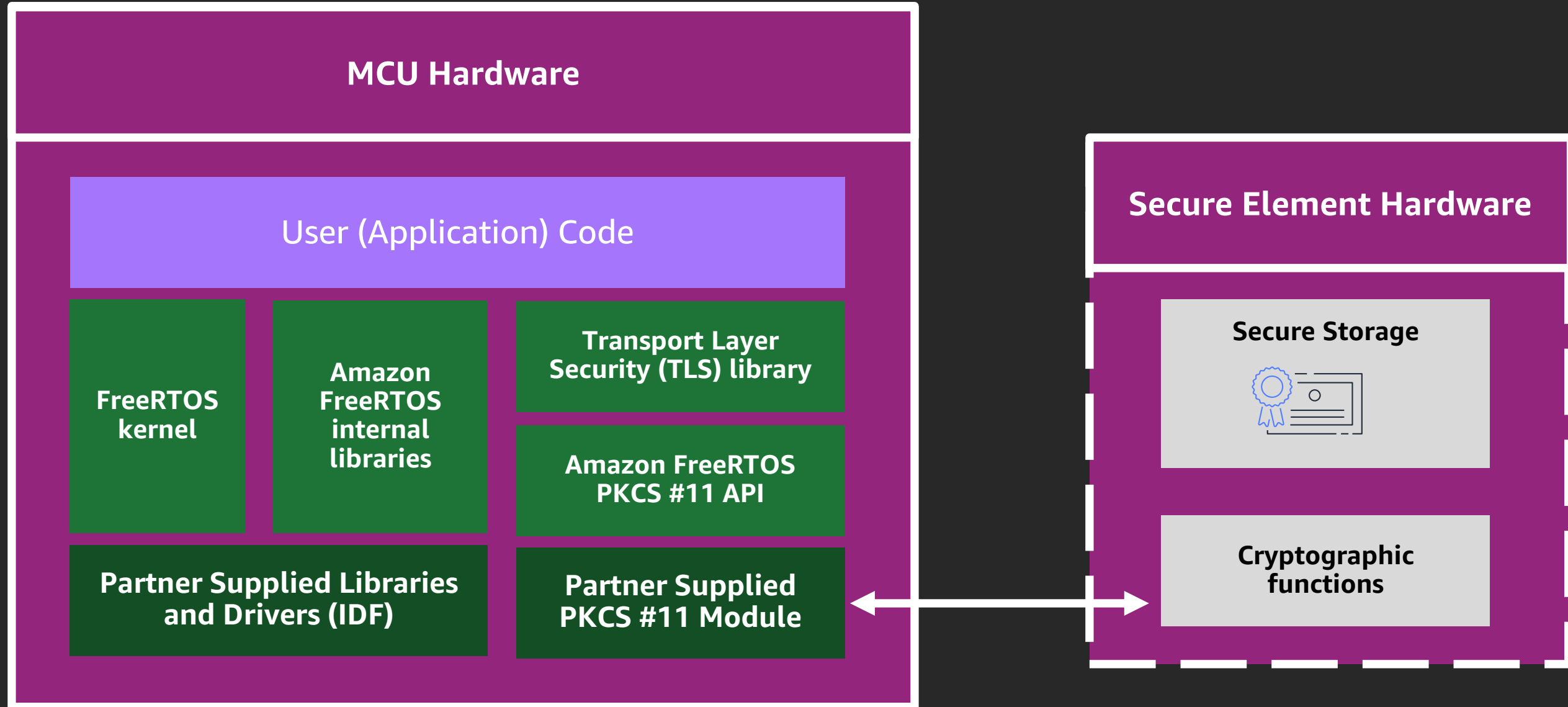
FreeRTOS
kernel

Amazon FreeRTOS internal
libraries

Transport Layer Security (TLS) library

Partner Supplied Libraries and Drivers (IDF)

암호화를 위하여 secure element 사용

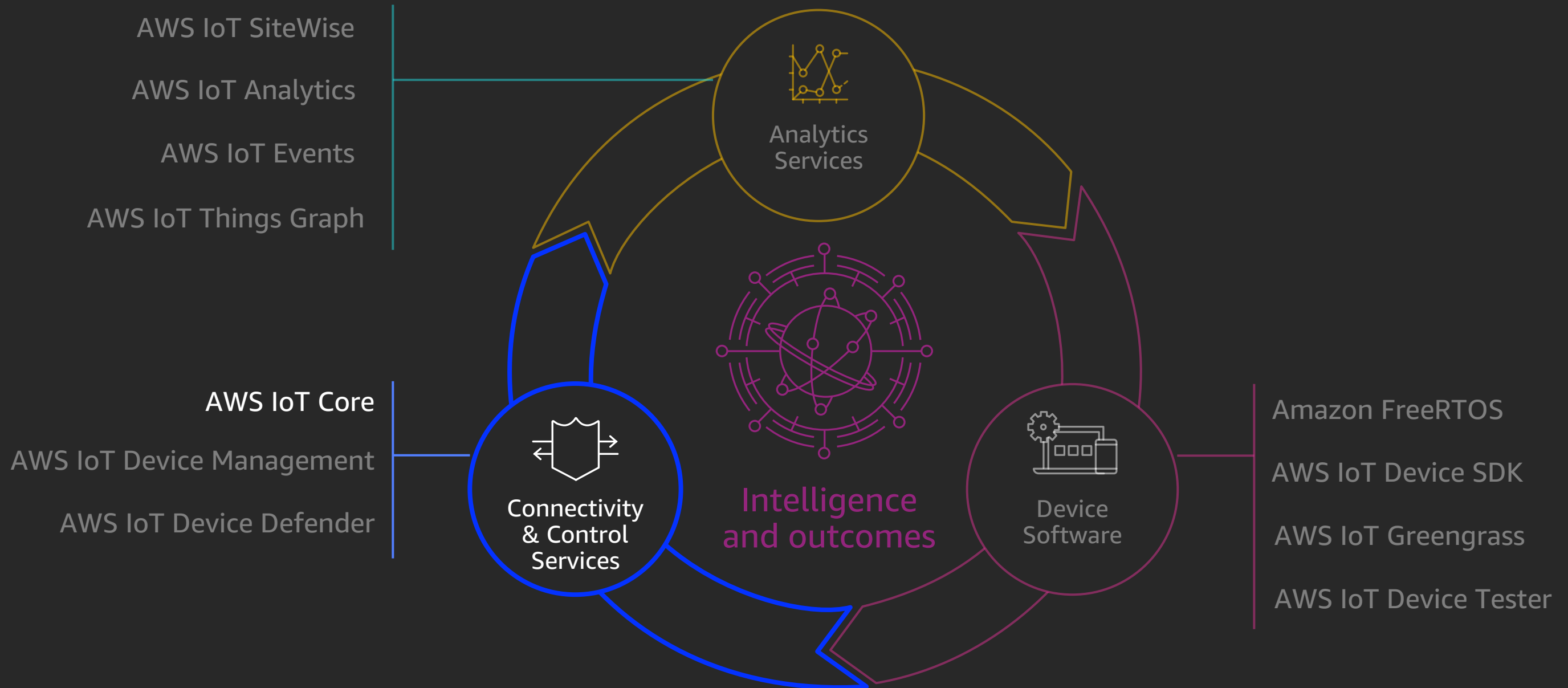


IoT Core

- 디바이스 연결 관리 및 Cloud application과의 상호작용



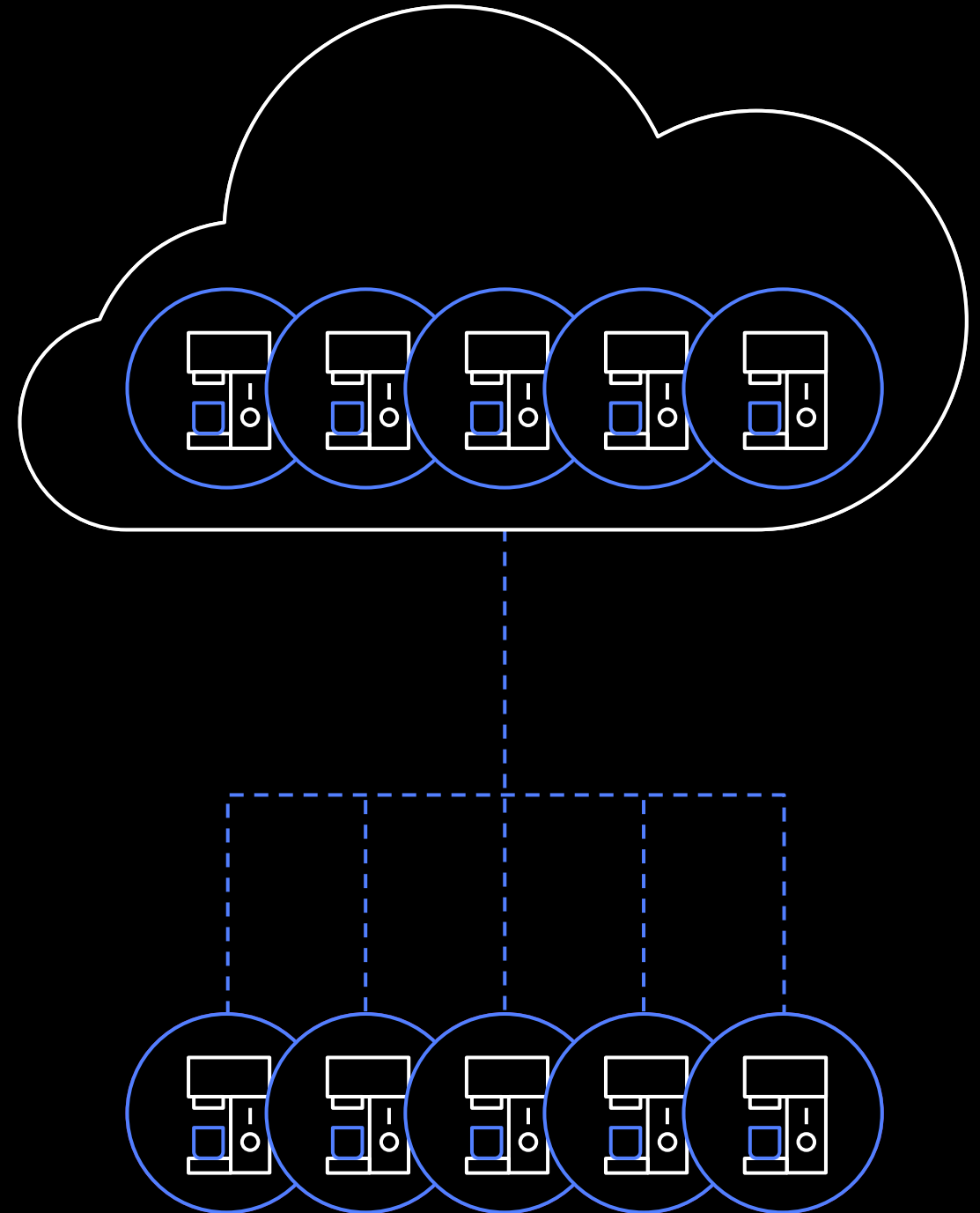
대규모의 장치를 제어, 관리 및 보호하려면 어떻게 해야합니까?



대량의 IoT 디바이스들의
연결 관리 및 Cloud상의
애플리케이션들과
상호작용하도록 하려면
어떻게 해야 할까요?

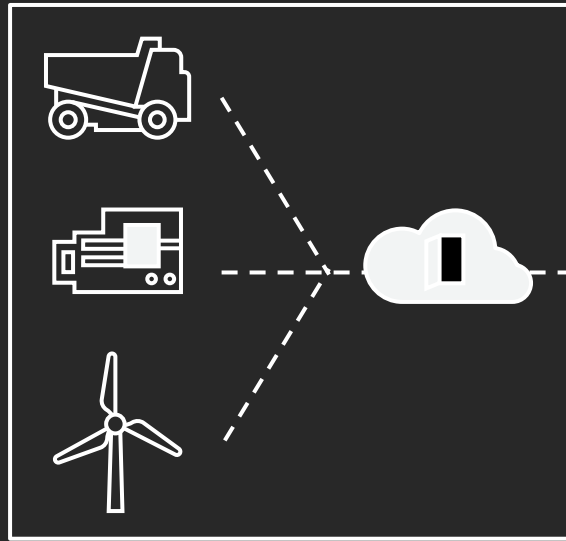


Connectivity
& Control
Services

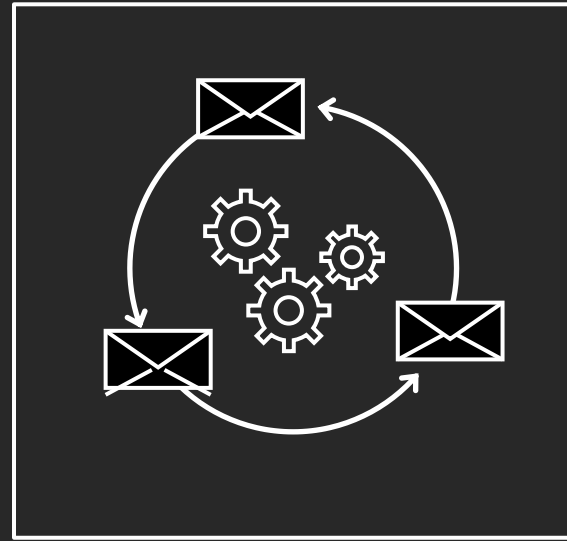


AWS IoT Core

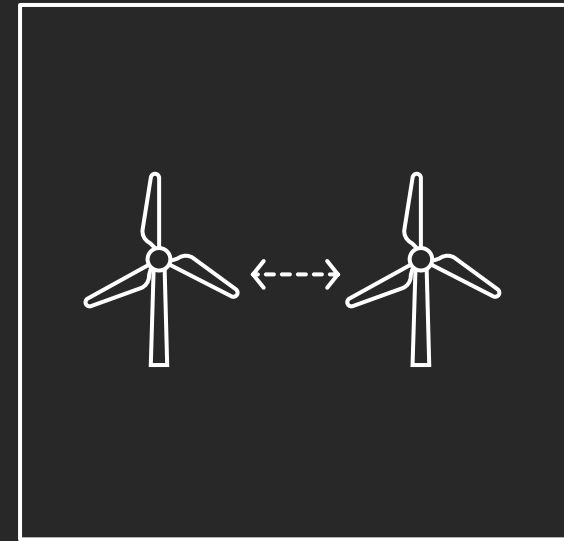
AWS IoT Core는 연결된 디바이스가 cloud 애플리케이션 및 기타 디바이스와 쉽고 안전하게 상호 작용할 수 있도록 하는 관리형 서비스입니다.



대규모로 디바이스를 AWS cloud 및 기타 디바이스에 안전하게 연결하기 위하여



연결된 장치의 데이터를 라우팅 및 처리하기 위하여



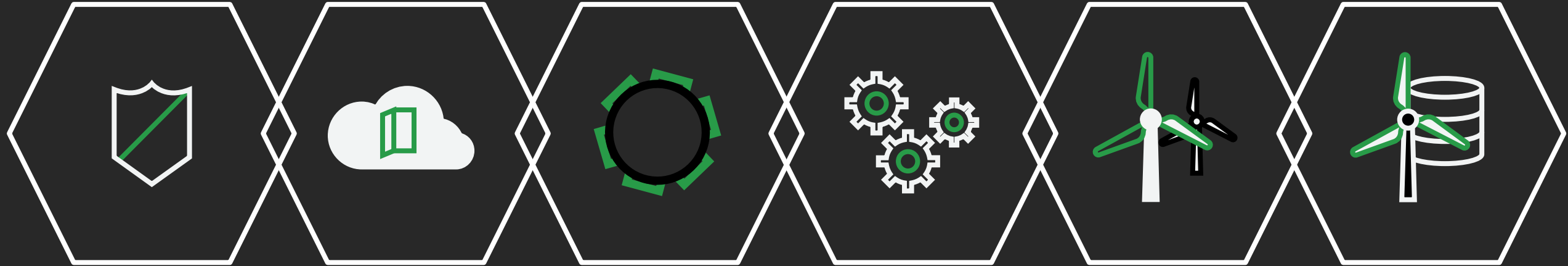
오프라인 상태에서도 응용 프로그램이 장치와 상호 작용할 수 있도록 하기 위하여



다른 AWS 서비스와 완전히 통합하여 데이터를 기반으로 추론하기 위하여 (Analytics, Databases, AI, etc.)



AWS IoT Core



Identity Service

규모에 따라
기기 인증
관리 및 고유
한 ID로
provision

Device Gateway

IoT 워크로드에
최적화된 연결
관리

Message Broker

IoT fleet
전체에서
안정적이고
빠른 통신

Rules Engine

저렴한 비용으로
대량의 IoT
데이터를 수집하고
전처리 분석, 보고
및 시각화를 위해
10 개 이상의
서비스에 제공

Device Shadow

언제든지
장치 상태를
이해하고
제어

Registry

AWS
서비스에서
쉽게 사용할
수 있도록
디바이스 정의
및 카탈로그화



Connectivity
& Control
Services

Identity Service

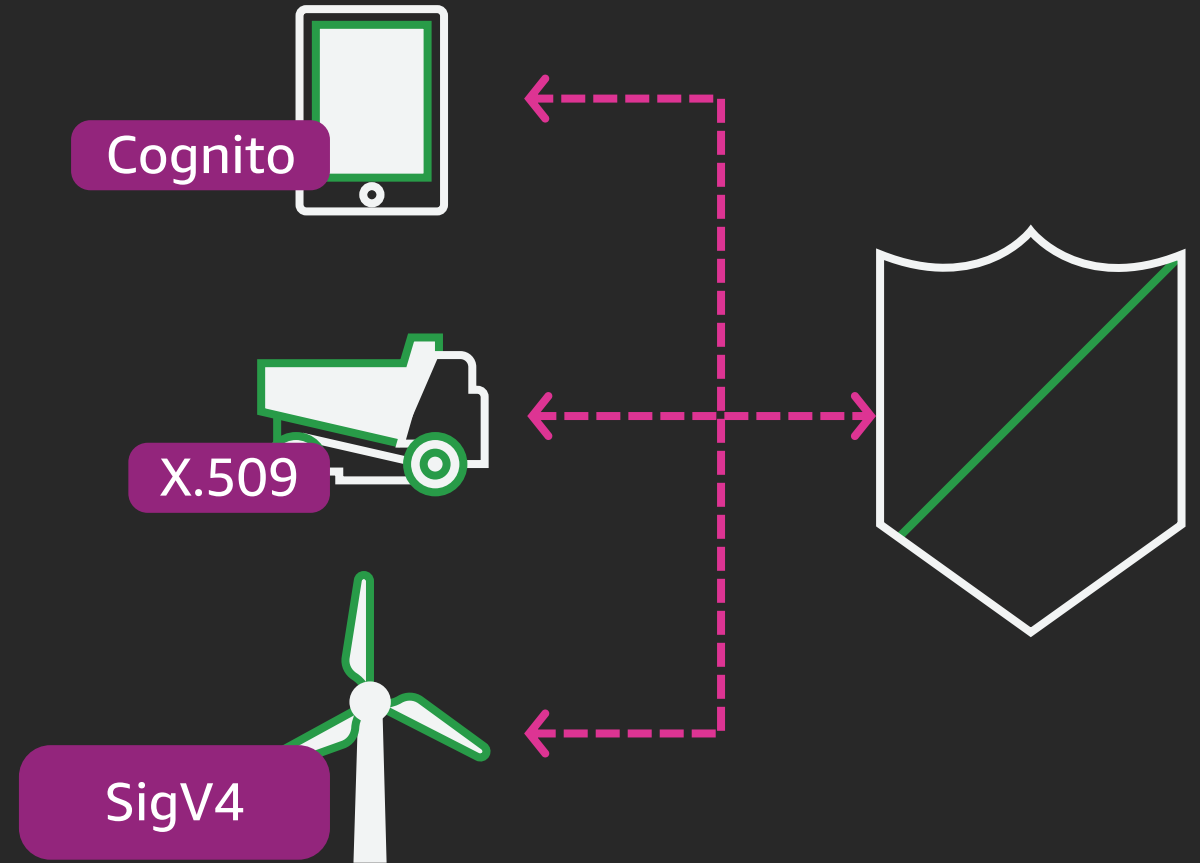
규모에 따른 장치의 인증 관리 및 고유한 ID 제공

자체 root CA 및 client 인증서를 가져와서
사용하거나 AWS IoT Core가 사용자를
위한 인증서를 생성

자동화된 device provisioning 방식 지원

certificate-based mutual authentication,
SigV4, Amazon Cognito Identity,
custom authorizers(token based
authentication) 지원

IoT정책을 통한 유연하고 세분화된 액세스
제어



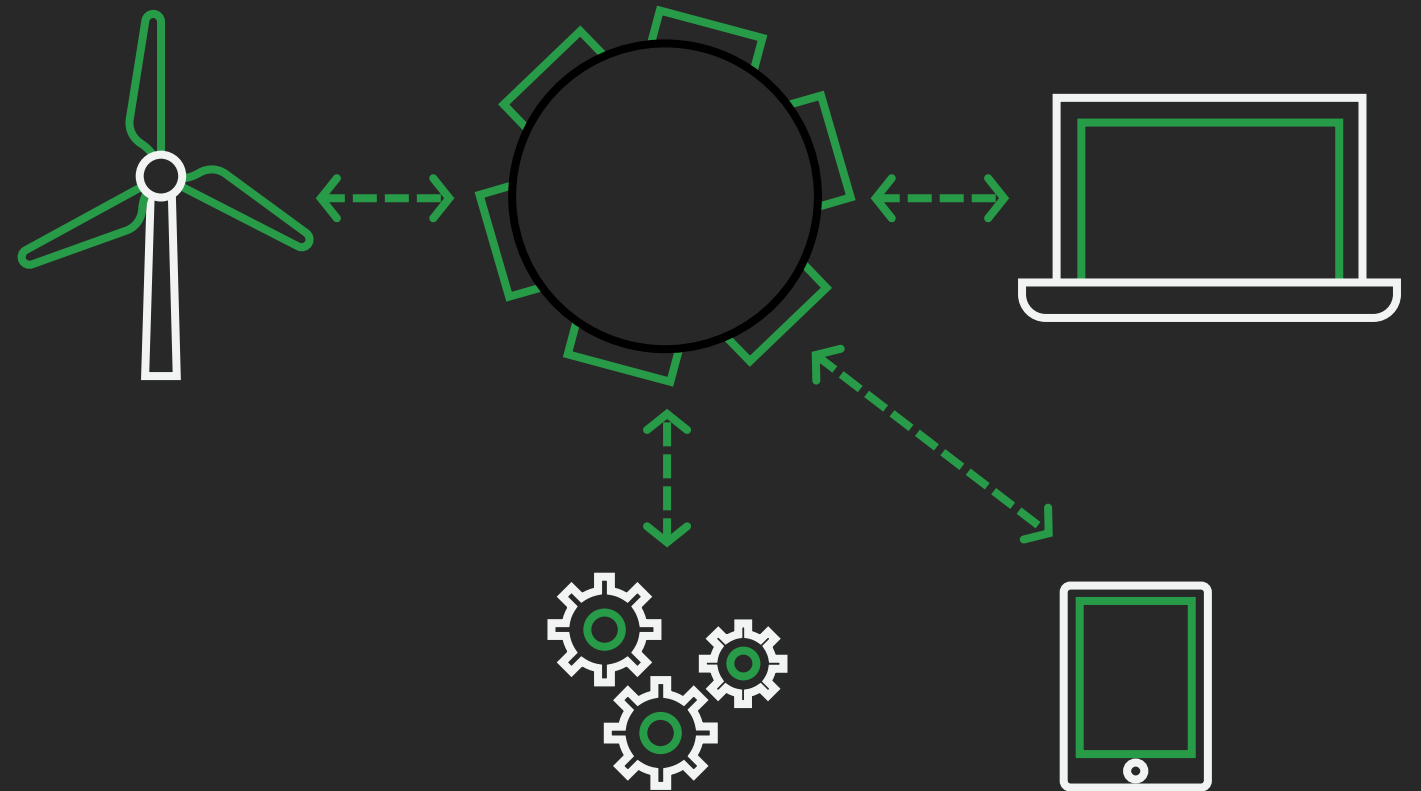
Connection Requirements (certificate-based mutual authentication)

장치를 연결하는데 필요한 구성 요소

x509 Device Certificate

Security Policy

Device를 등록



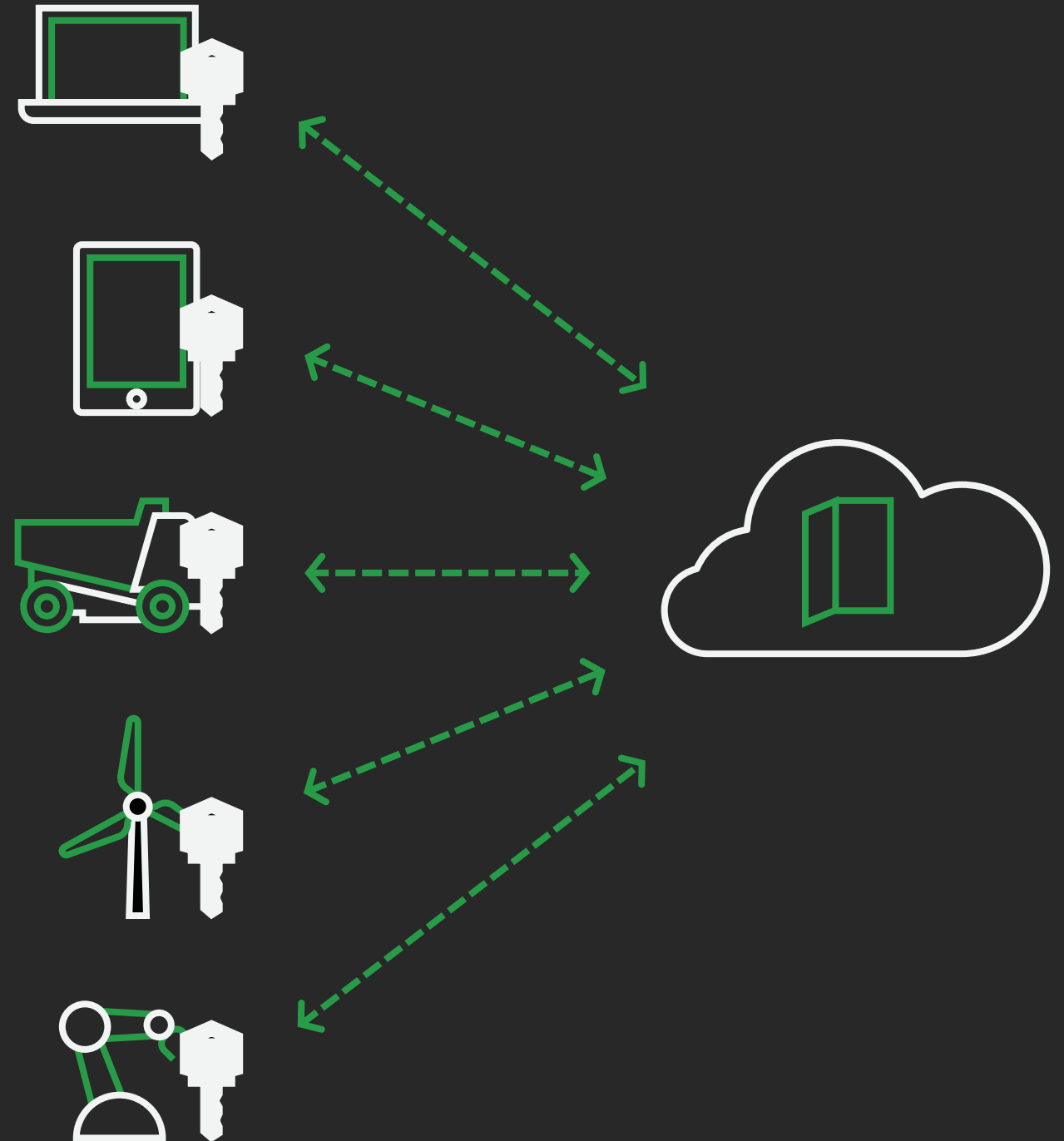
Device Gateway

IoT 워크로드에 최적화 된 연결 관리

MQTT, WebSockets, HTTP1.1 프로토콜
지원

TLS 1.2 기반 암호화 통신

디바이스 게이트웨이는 완전 관리형
서비스로, 사용자가 인프라를 관리할
필요 없이 수십억 개의 디바이스를
지원하도록 자동 확장



Message Broker

확장 가능하고 강력한 publish-subscribe broker

독립된 디바이스 및 application에 대한
Publish/Subscribe

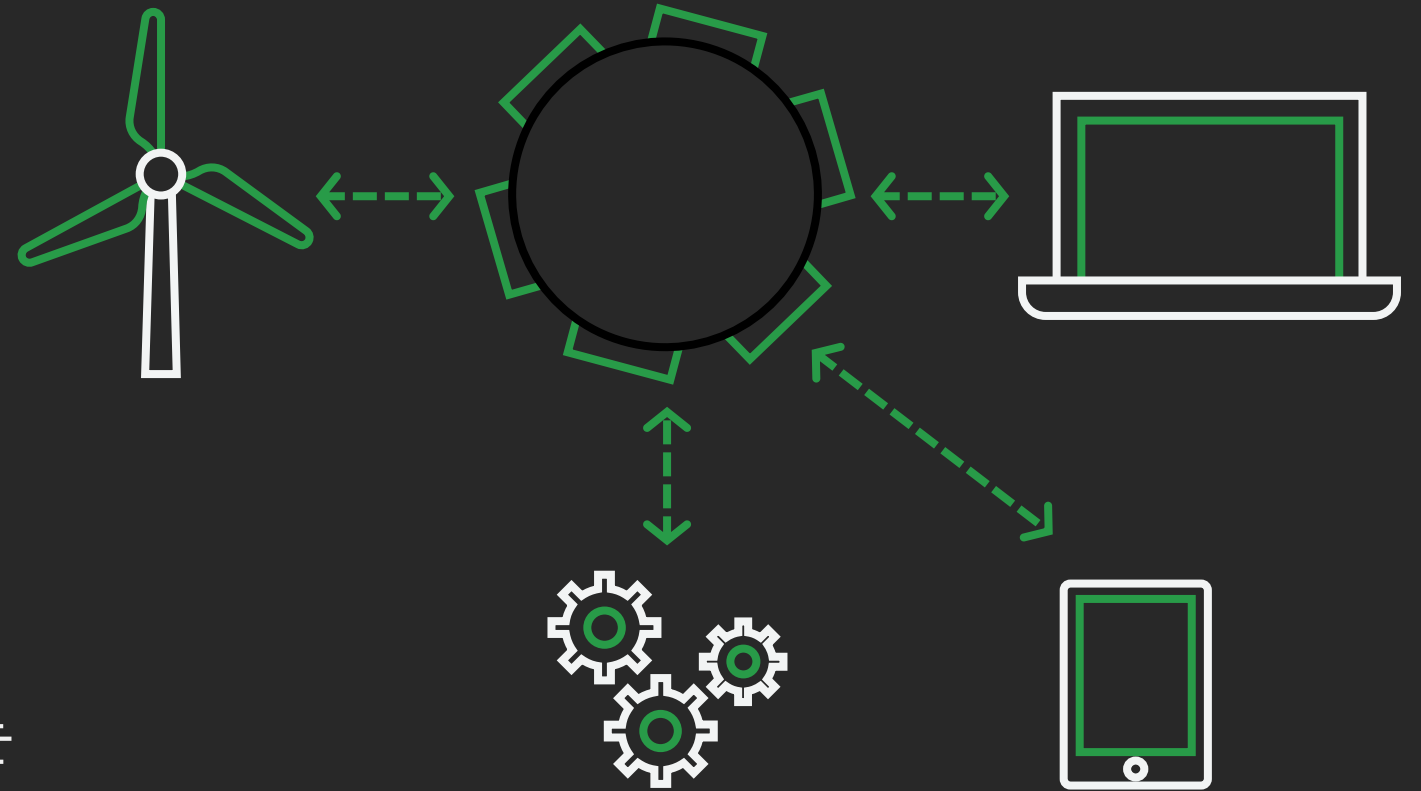
wildcard topic filter 지원

QoS0 와 QoS1 메시징

JSON과 Binary payload 지원

SDK 없이 모든 기능 사용 가능
+ 필요할 경우 일반적인 프로그래밍 언어를 지원하는
device SDK 제공

IoT Core 기능은 topic 및 reserved topic에 대해 구현되며,
SDK를 사용하지 않아도 기능이 손실되지 않습니다.

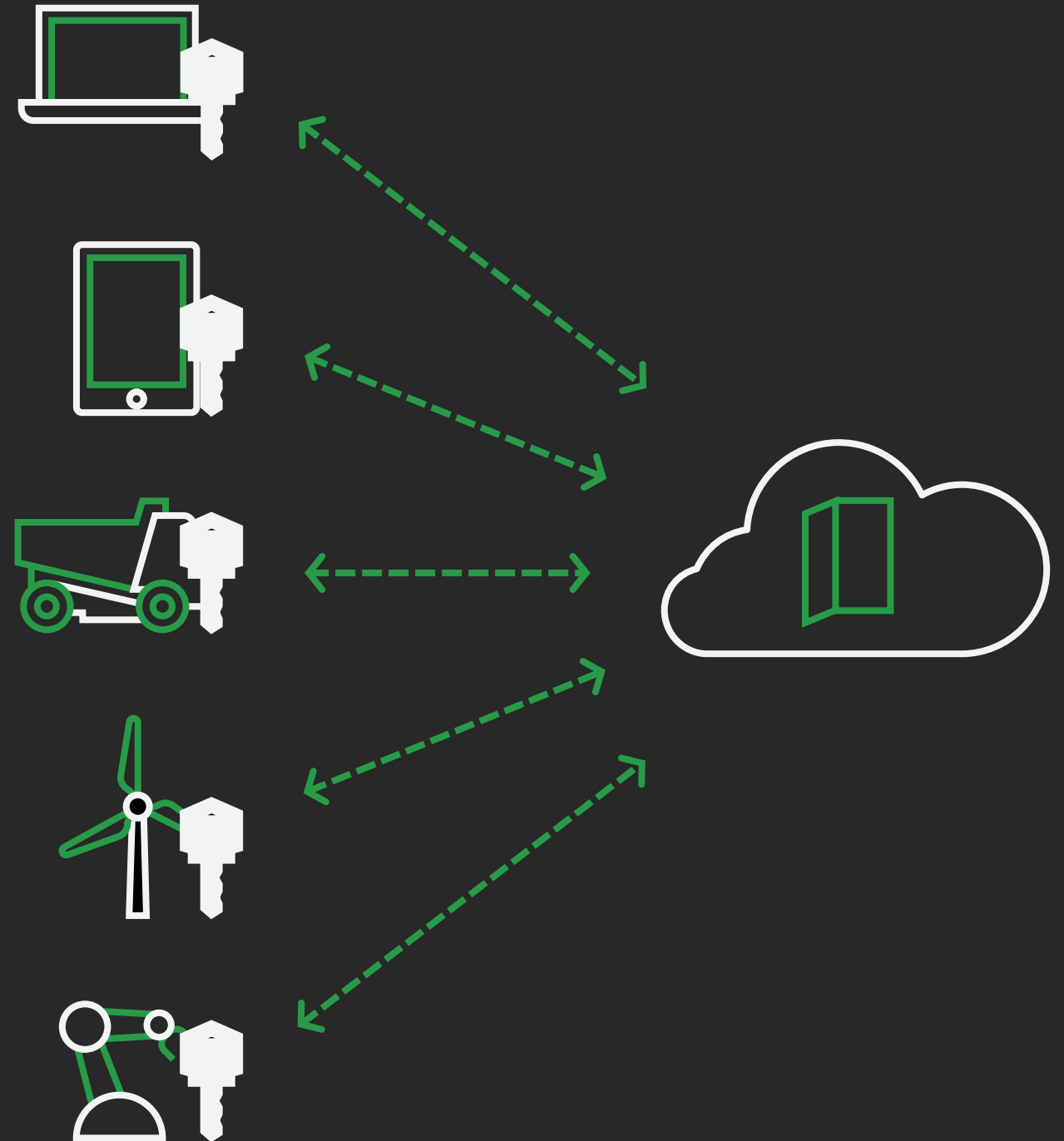


What can data look like?

사용자 지정 JSON payloads

```
PUBLISH turbines/ev-gen/123 (qos: 0)
{
  "timestamp": "2016-11-29T10:00:00",
  "temperature": 125,
  "humidity": 95,
  "rotor-freq": 6455,
  "output": 480,
  "output-freq": 60
}
```

What about binary payloads?
Supported but limited



Rules Engine

저렴한 비용으로 많은 양의 데이터를 수집하고 사전 처리하며 분석, 보고 및 시각화를 위한 10 개 이상의 서비스에 제공

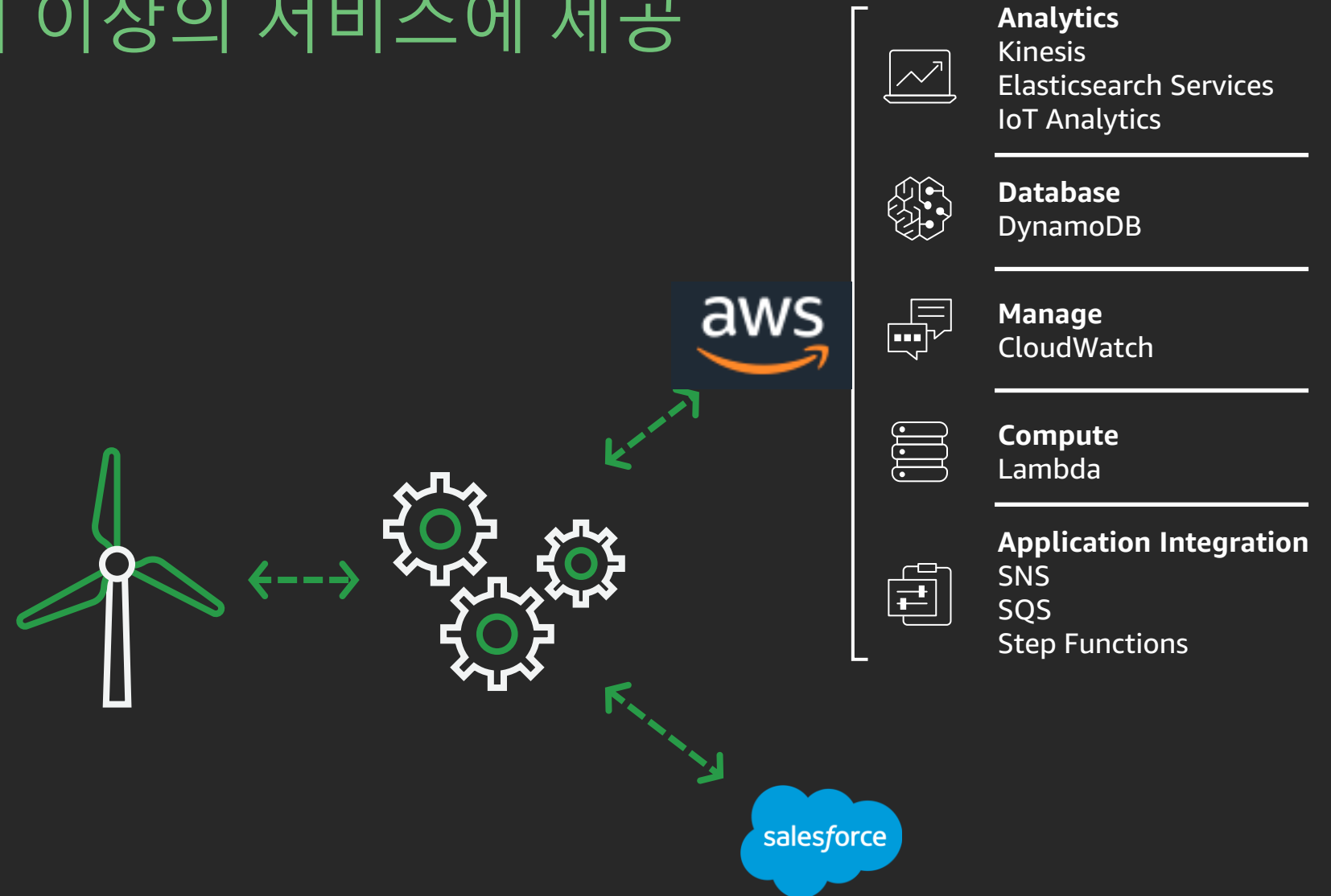
Transform—수학, 문자열 조작, 날짜 등을 위한 내장 함수

Filter—WHERE 절을 사용하여 필요한 데이터만 추출

Enrich—인라인 AWS Lambda 실행을 통해 Device Shadow 및 Amazon Machine Learning 또는 외부 소스에서 컨텍스트를 가져옵니다.

Route—10개가 넘는 AWS 서비스 및 Salesforce 등과 같은 3rd 서비스로 데이터를 전송

Basic Ingest



Device Shadow

언제든지 장치 상태를 이해하고 제어

디바이스의 최종 상태를 보고

- 예를 들면, 전구의 최종 상태는 빨간색(red)

디바이스의 상태를 변경

- 예를 들면, 전구의 색상을 파란색(blue)으로 변경

MQTT를 사용하여 상태 변경을 실시간으로 알림

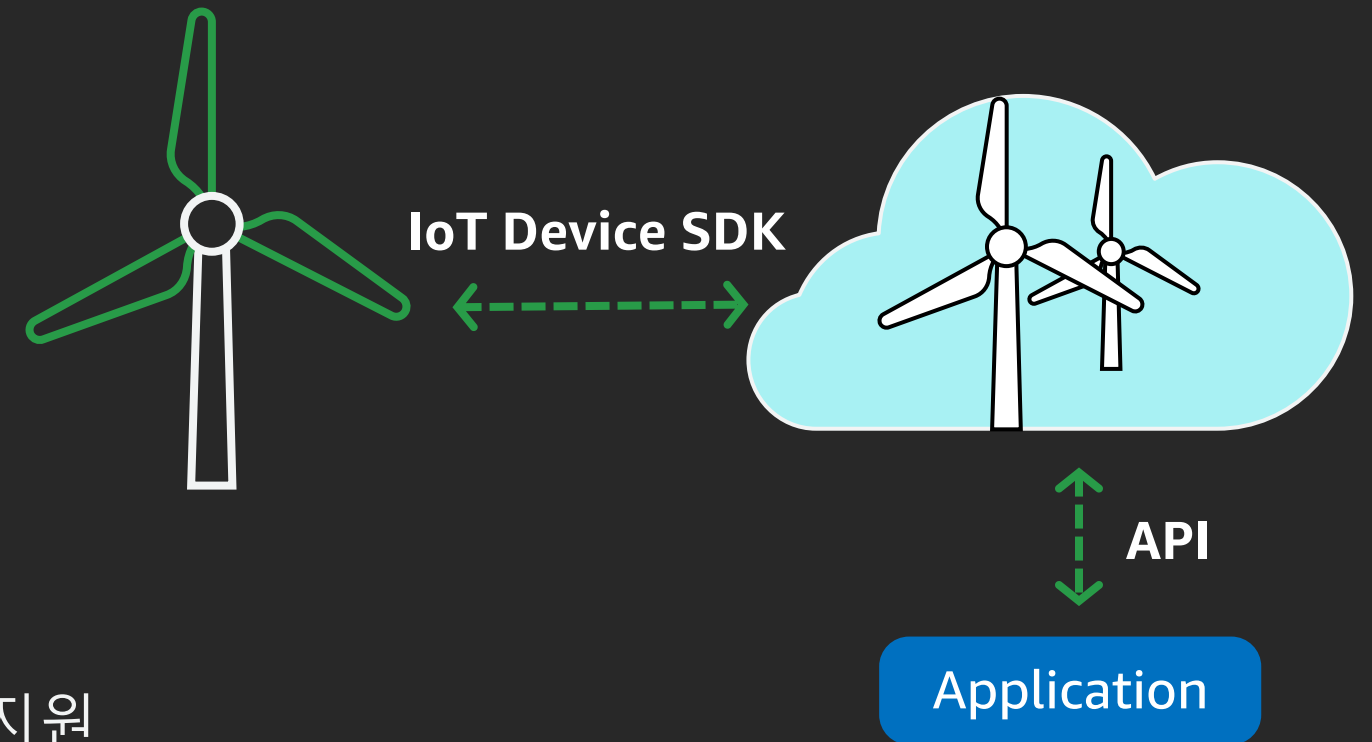
오프라인 디바이스와의 비동기 통신

장치에서의 손 쉬운 개발을 위한 Device SDK 통합

Application이 디바이스와 상호 작용할 수 있는 REST API 지원

모든 명령 및 제어 작업을 위한 기본 기능

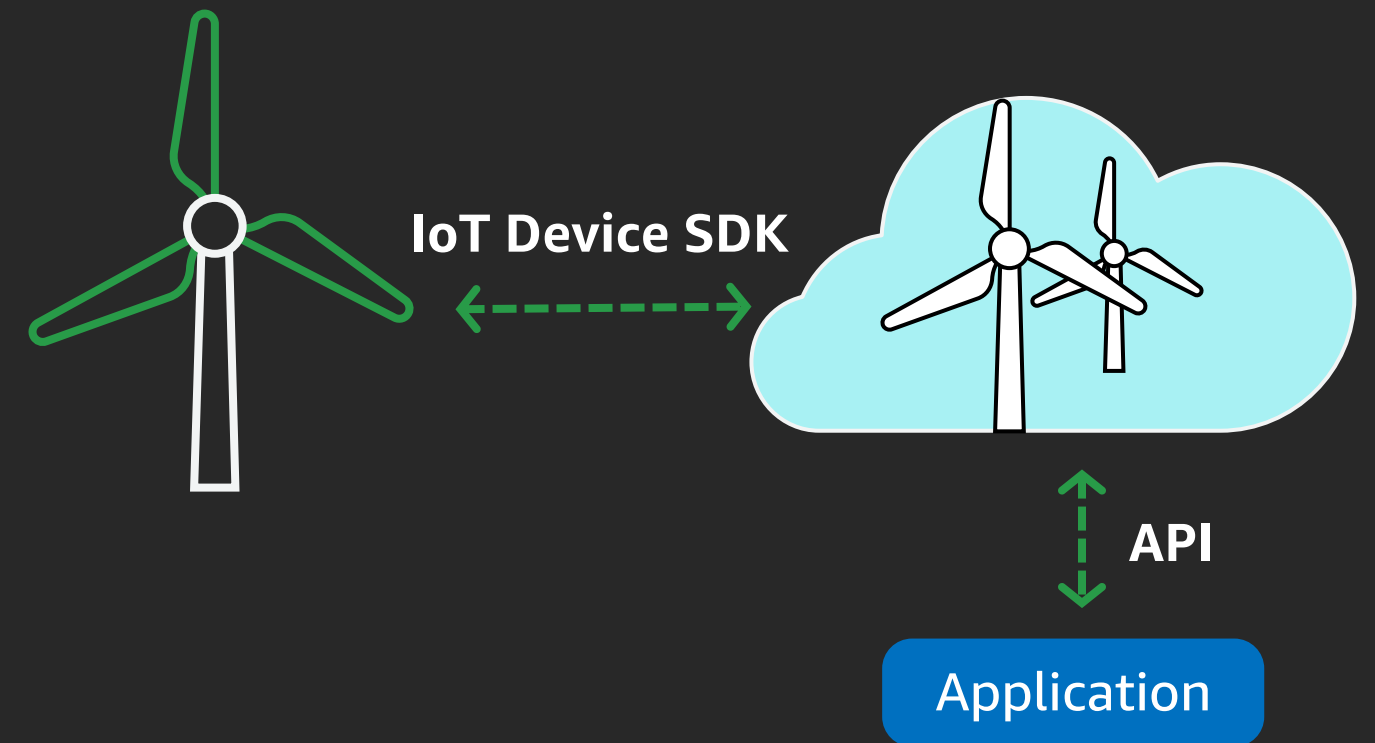
여러 개의 named shadow와 하나의 unnamed shadow (classic shadow) 지원



Device Shadow

The shadow is the most critical feature to understand...

```
"state" : {  
  "desired" : {  
    "lights": { "color": "RED" },  
    "engine" : "ON"  
  },  
  "reported" : {  
    "lights" : { "color": "GREEN" },  
    "engine" : "ON"  
  },  
  "delta" : {  
    "lights" : { "color": "RED" }  
  }  
},  
"version" : 10  
}
```



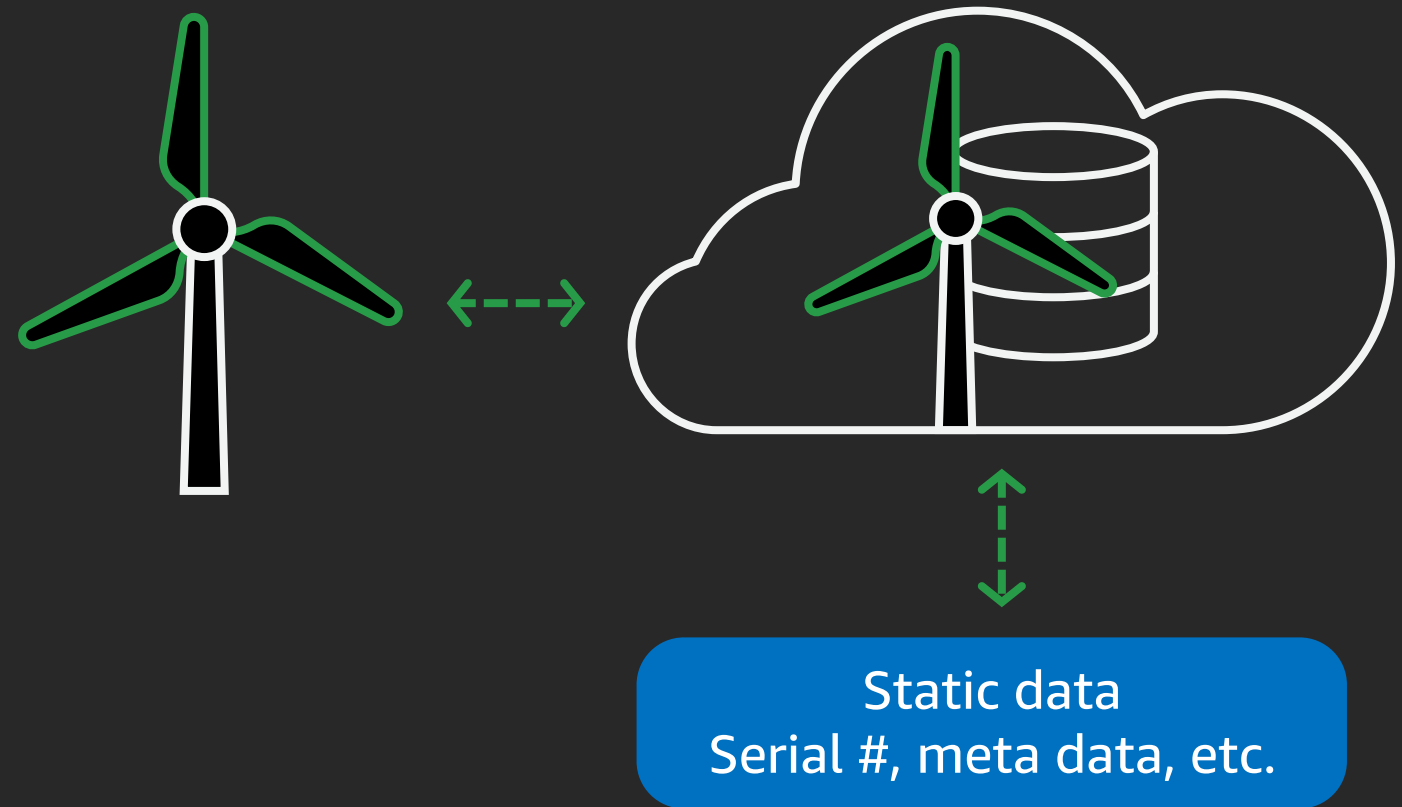
Registry

AWS 서비스에서 쉽게 사용할 수 있도록 디바이스 정의 및 카탈로그

간단한 검색 (예를 들면, 2010에
생성된 디바이스들은?)

여러 기기에서 속성 및 정책을
표준화하기 위해서 *ThingTypes*
정의 (예를 들면, Hyundai 및 Kia 를
Car 라는 ThingType으로 정의함)

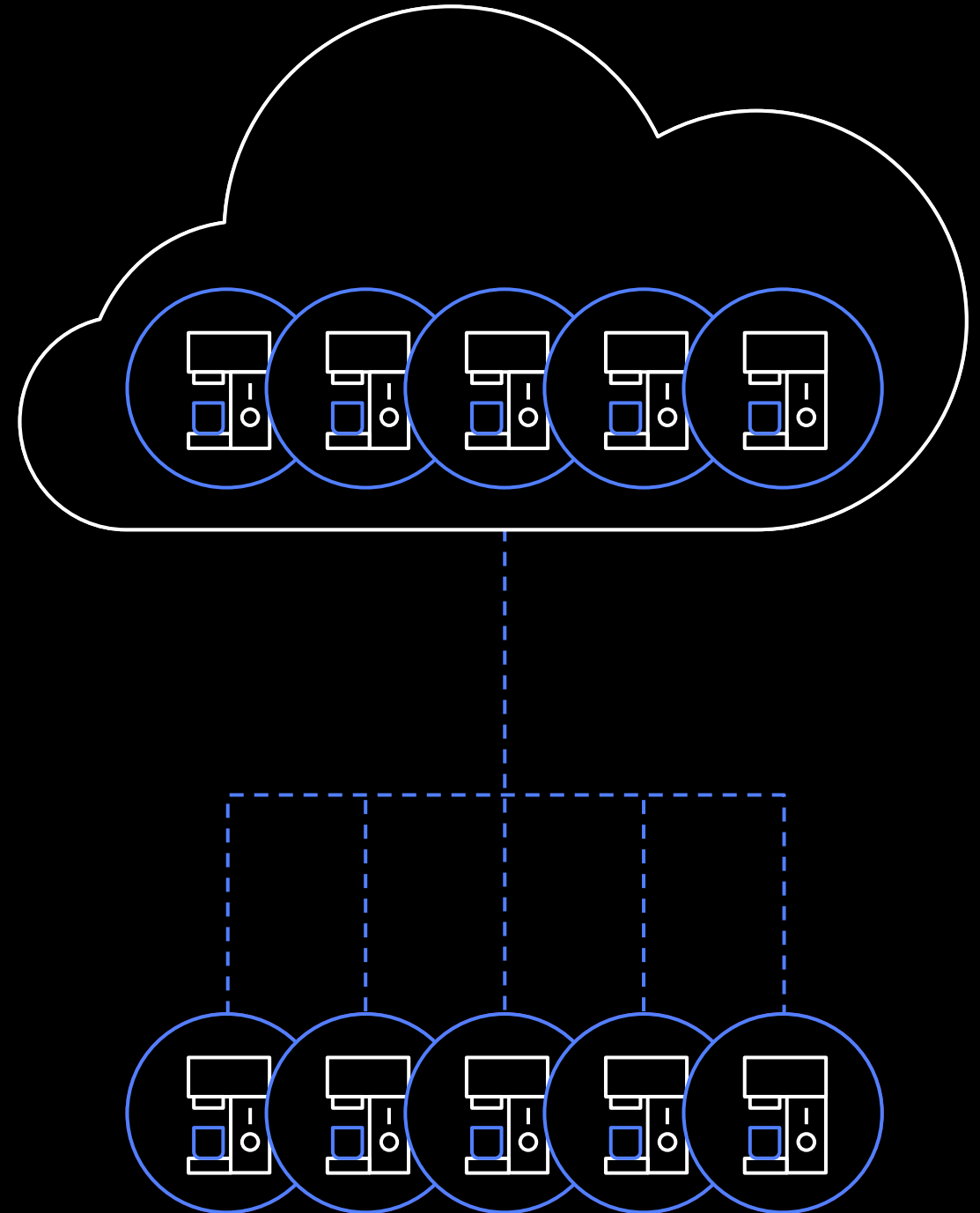
보다 간단한 관리 (작업 실행, 정책
설정 등)를 하기 위하여 *Group*을
정의/ (예를 들면, 차 안의 sensor들)



데모: Murata Type 1SC를 활용한 AWS IoT Core와 통신



Connectivity
& Control
Services



Type 1SC

by Murata Manufacturing Co., Ltd.



Type 1SC is a compact, low cost, LTE-M module that supports the cellular LTE-M and NB1 network protocols. This stand-alone module measures just 11.1 x 11.4 x 1.5 mm (max), constructed in a silver resin shielded package and is comprised of an Altair ALT1250 transceiver with an FEM module and external flash memory. An integrated TCXO, that has robust low drift thermal characteristics, provides an accurate clock source for the RF transceiver.

Type 1SC supports Multi-band (low and mid-band) to cover most of the globe, VoLTE to deliver voice calls directly over LTE-M, an iUICC to select MNO profiles, GPS and Glonass for accurate positioning, and features low current consumption (eDRX and PSM) to achieve a battery life of 10+ years. It is Ideally suited for small, battery operated IoT devices and applications.

AWS Service

AWS IoT Core

Industry

Agriculture, Consumer Goods, Energy / Utilities,
Healthcare / Life Sciences, Industrial, Retail,
Security, Smart City, Smart Home,
Transportation

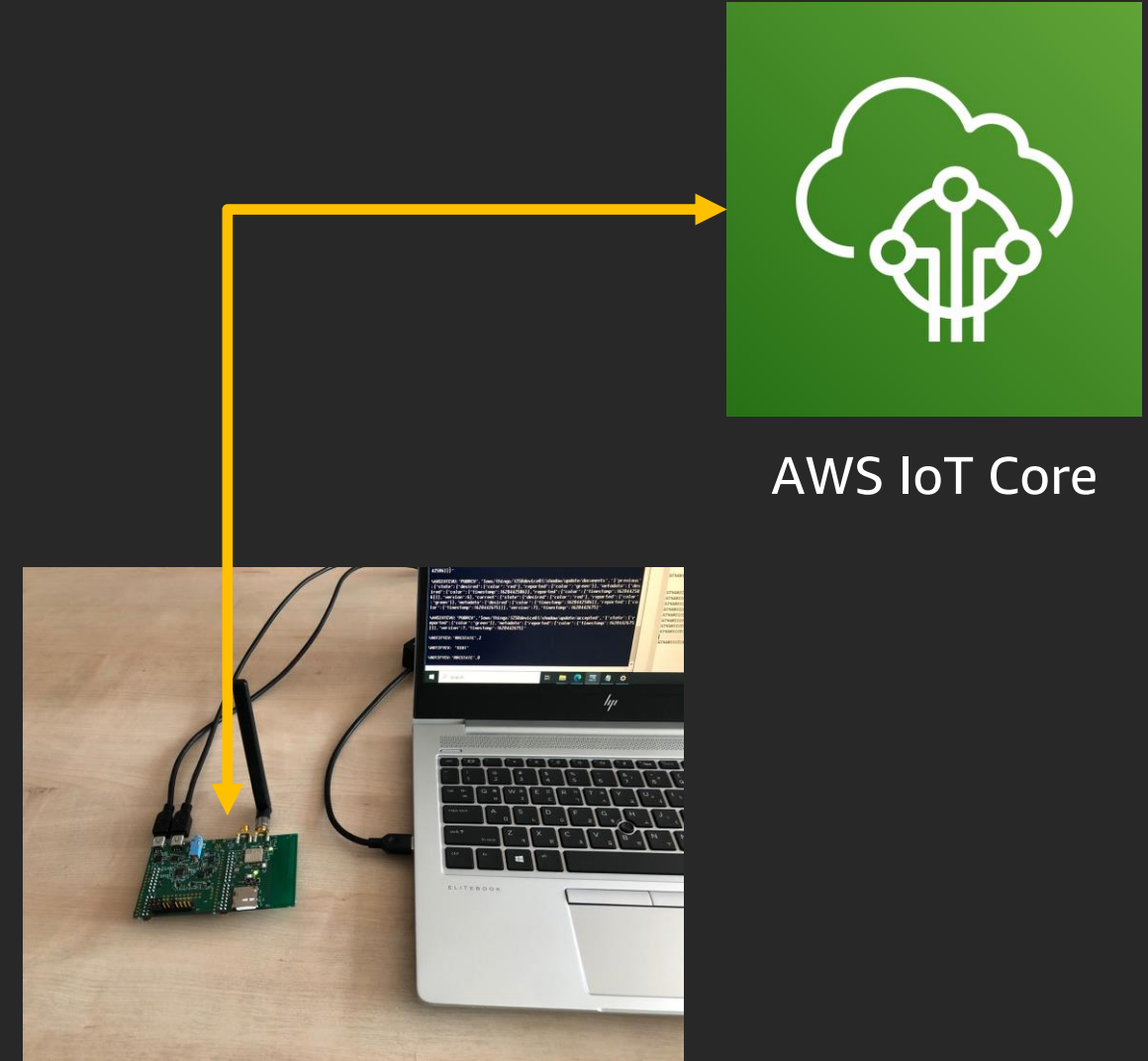
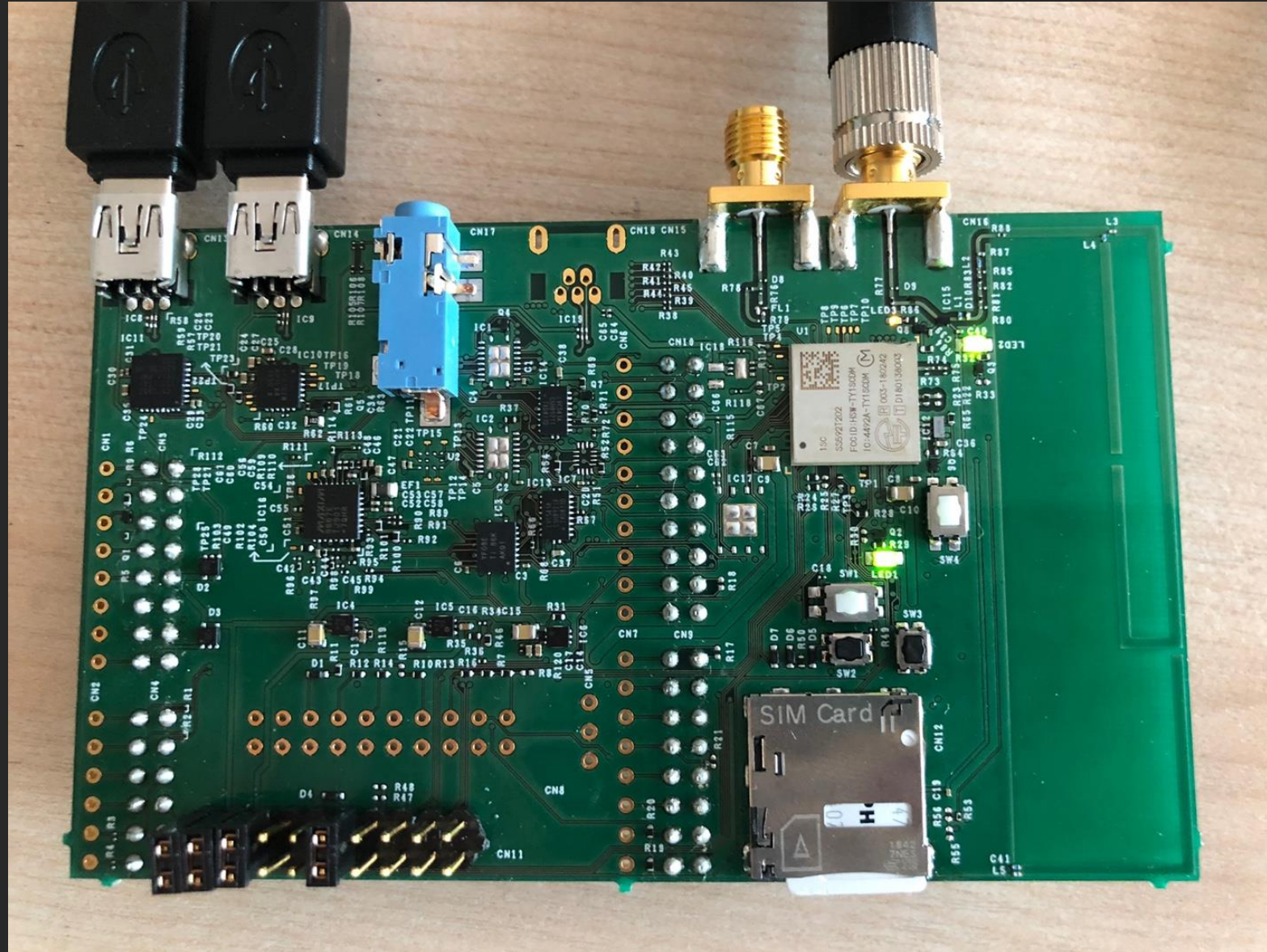
Device Type

RF Module

Application

Appliances, Asset Management / Tracking,
Building Automation, Connected Vehicle,
Contact / Call Center, Data Logging,
Diagnostics, Emergency Response, Fleet
Management, Food Service, Heavy Equipment,
Home Automation, Industrial Automation,
Irrigation, Lighting, Logistics, Medical Devices,
Oil / Gas, Point-of-Sale, Predictive Maintenance
/ Quality, Remote Monitoring, Security / Access
Control, Smart Grid, Tank Monitoring, UAV,
Video Surveillance, Water / Wastewater,
Wearables

Murata Type 1SC를 활용하여 AWS IoT Core와 통신



AWS IoT Core 서비스 콘솔

AWS IoT

Monitor

Activity

▶ Onboard

▶ Manage

▼ Greengrass

Getting started

Core devices

Components

Deployments

▶ Classic (V1)

▼ Secure

Certificates

Policies

CAs

Role Aliases

Authorizers

▼ Defend

Intro

Audit

▼ Detect

Alarms

AWS IoT > MQTT test client

MQTT test client

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Subscribe to a topic

Publish to a topic

Topic filter

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

Enter the topic filter

▶ Additional configuration

Subscribe

Subscriptions	Topic
You have no topic subscriptions.	Subscribe or select a topic to view incoming messages.

MQTT test client

Info

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Subscribe to a topic

Publish to a topic

Topic filter

Info

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

tracker/location

▶ Additional configuration

Subscribe

Subscriptions

tracker/location

Pause

Clear

Export

Edit

tracker/location



Session View Tools Help Documents

4,115200 Serial(COM) Port

COM4@115200,8,n,1

```
%ANSIOTEVU: "SUBCONF", 3, 0
AT%ANSIOTCMD="SUBSCRIBE", "$aws/things/1250device01/shadow/update/rejected"
%AWSCMD: 4
OK

%ANSIOTEVU: "SUBCONF", 4, 0
AT%ANSIOTCMD="SUBSCRIBE", "$aws/things/1250device01/shadow/update/delta"
%AWSCMD: 5
OK

%ANSIOTEVU: "SUBCONF", 5, 0
AT%ANSIOTCMD="SUBSCRIBE", "$aws/things/1250device01/shadow/update/documents"
%AWSCMD: 6
OK

%ANSIOTEVU: "SUBCONF", 6, 0
AT%ANSIOTCMD="SUBSCRIBE", "$aws/things/1250device01/shadow/get/accepted"
%AWSCMD: 7
OK

%ANSIOTEVU: "SUBCONF", 7, 1
AT%ANSIOTCMD="SUBSCRIBE", "$aws/things/1250device01/shadow/delete/accepted"
%AWSCMD: 8
OK

%ANSIOTEVU: "SUBCONF", 8, 1
AT%ANSIOTCMD="SUBSCRIBE", "$aws/things/1250device01/shadow/delete/rejected"
%AWSCMD: 9
OK

%ANSIOTEVU: "SUBCONF", 9, 1

%NOTIFYEV: "SIB1"
```

모든 유형의 연결된
디바이스에 Amazon
Alexa를 저렴하게
추가하려면 어떻게
해야합니까?

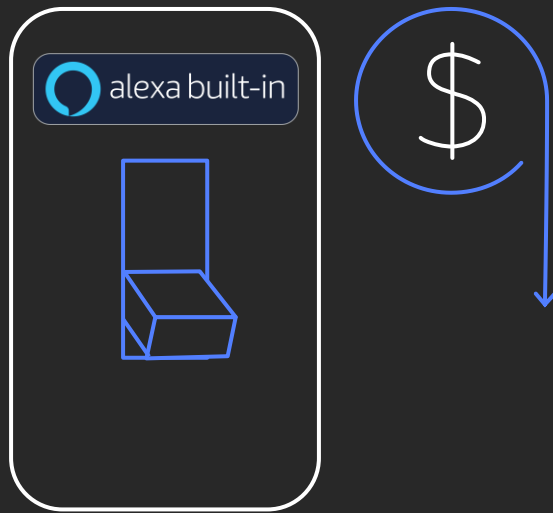


Connectivity
& Control
Services



AIA (Alexa Voice Service Integration for AWS IoT Core)

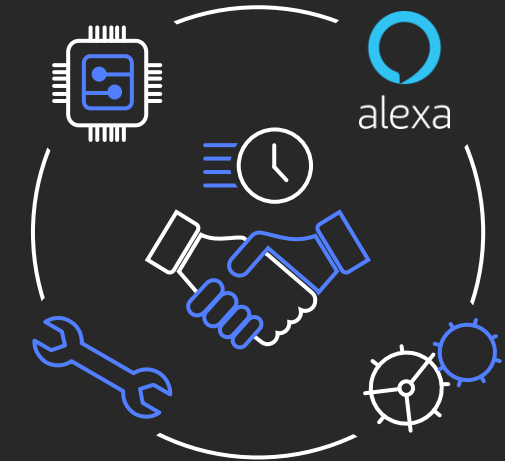
조명 스위치, 온도 조절기 및 소형 가전 제품과 같은 새로운 범주의 제품에 Alexa 내장 기능을 사용하여 빠르고 비용 효율적으로 시장에 출시하십시오.



필요한 컴퓨팅 및 메모리 공간을 줄여 Alexa Voice 통합 비용을 최대 50 %까지 낮춥니다.



리소스가 제한된 장치 (예 : 임베디드 RAM이 1MB 미만인 ARM 'M'클래스 마이크로 컨트롤러)에서 Alexa 내장 제품의 새로운 범주를 생성합니다.



기본적으로 IoT Core 용 AVS Integration과 함께 작동하는 인증된 파트너 개발 키트로 시장 출시 시간을 단축하세요.



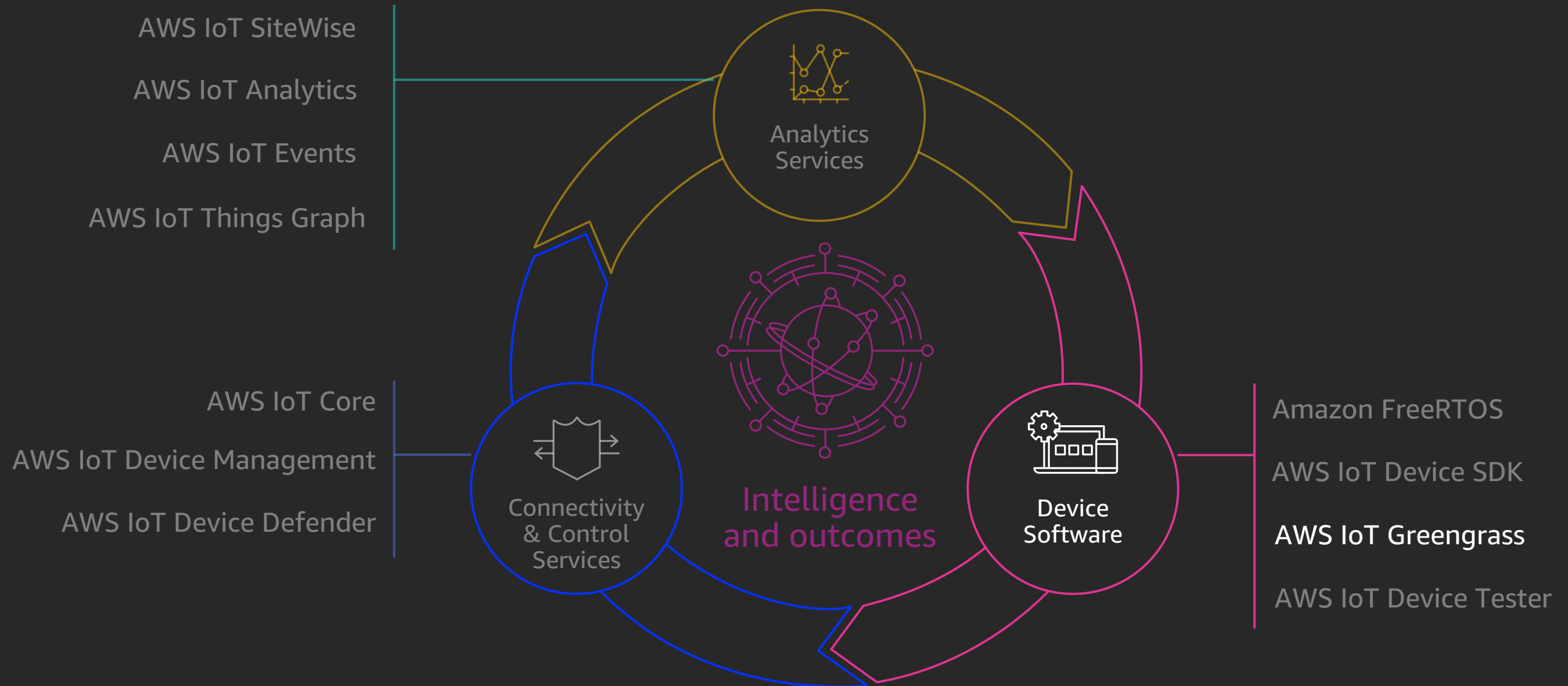
Connectivity
& Control
Services

AWS IoT Greengrass

- Cloud의 서비스와 기능을 edge까지 확장

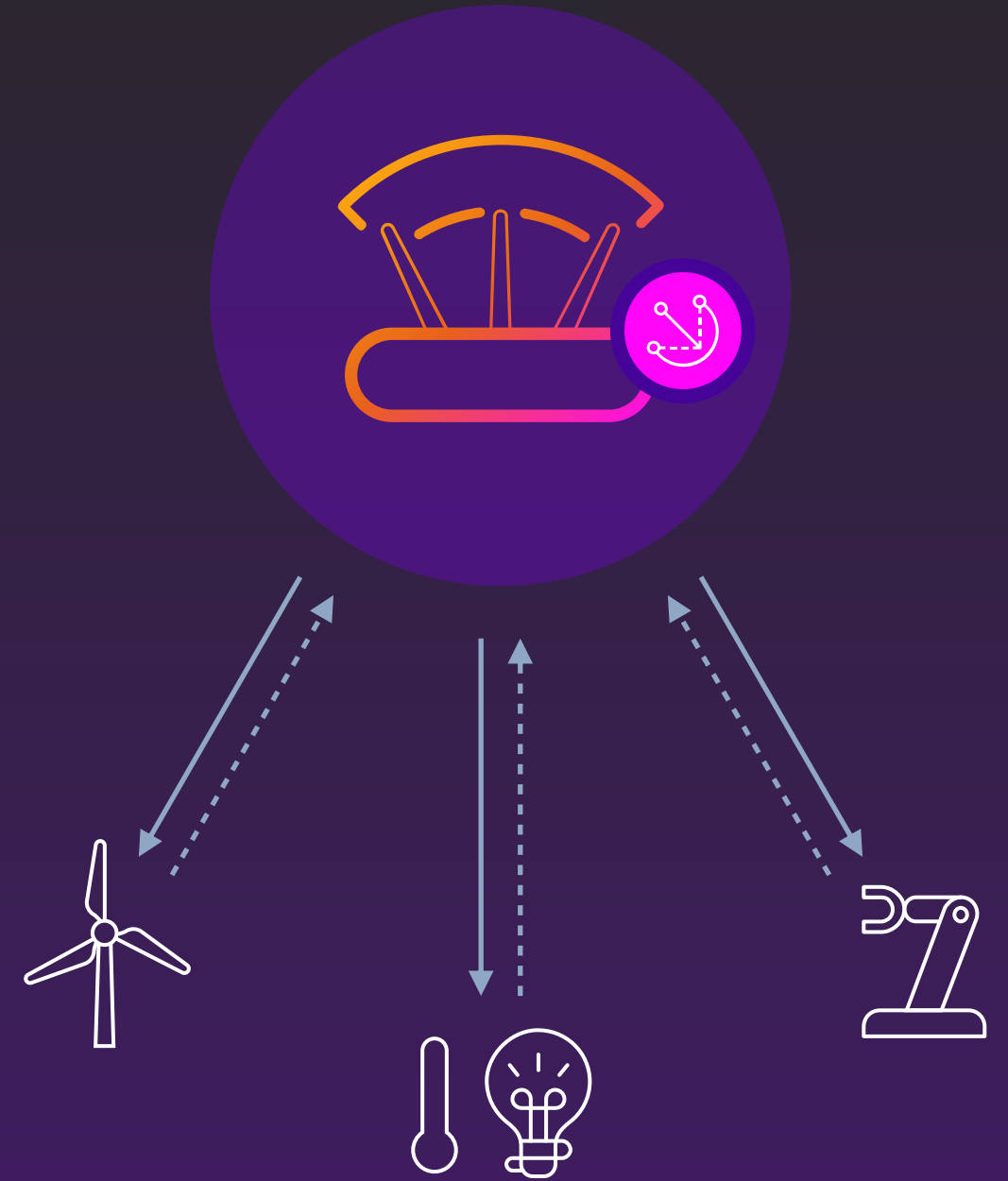


장치를 연결하고 edge에서 작동하려면 어떻게 해야합니까?



“

Edge에서
IoT 디바이스 소프트웨어를
쉽게 구축하고 관리하려면
어떻게 해야 할까요?



AWS IoT Greengrass



보안 및
AWS Secrets Manager



로컬 메시징



로컬 액션



Container 지원



스트림 매니저



ML 추론

AWS IoT Greengrass 2.0

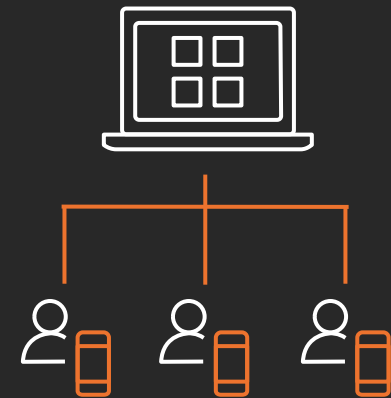
수백만 개의 장치에서 장치 소프트웨어를 쉽게 구축, 배포 및 관리 할 수 있는 유연한 edge 플랫폼



완전한 모듈화



규모에 맞게
device software를 배포

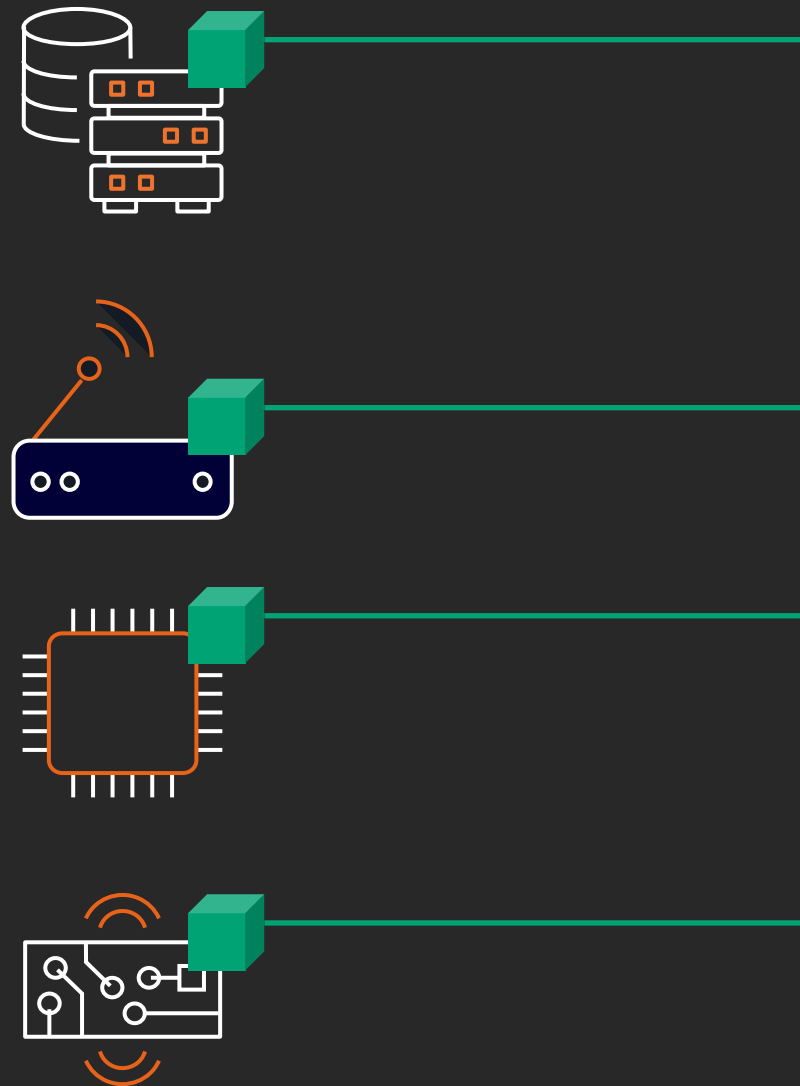


로컬에서 배포 및 설정 가능

AWS IoT Greengrass는 이제 open source입니다

AWS IoT Greengrass Core software

Software distributions



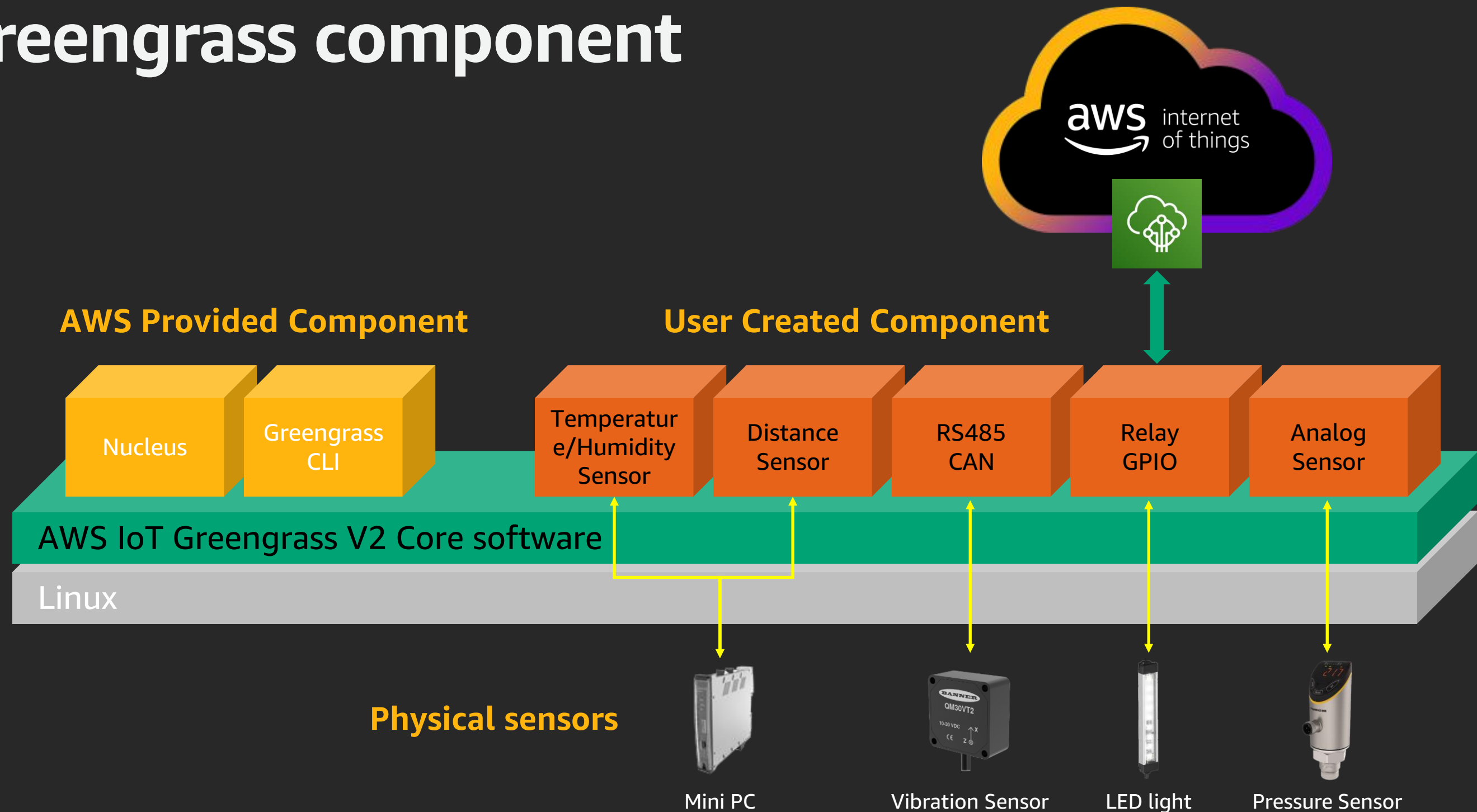
AWS IoT
Greengrass
Core software

AWS Cloud service



AWS IoT Greengrass V2 API
AWS IoT Greengrass V2 console

Greengrass component



Create your component - Recipes

Recipes

모든 component에는 metadata를 정의하는 recipe file이 포함
Recipe는 구성 요소의 매개 변수, 종속성, 수명주기 및 플랫폼 호환성 지정

```
---
RecipeFormatVersion: 2020-01-25
ComponentName: com.example.HelloWorld
ComponentVersion: '1.0.0'
ComponentDescription: My first AWS IoT Greengrass component.
ComponentPublisher: Amazon
ComponentConfiguration:
  DefaultConfiguration:
    Message: world
Manifests:
- Platform:
  os: linux
  Lifecycle:
    Run: |
      python3 {artifacts:path}/hello_world.py '{configuration:/Message}'
```

Create your component - Artifacts

Artifacts

스크립트, 컴파일된 코드,
정적 리소스 등을 포함

```
import sys
import datetime

message = f"Hello, {sys.argv[1]}! Current time: {str(datetime.datetime.now())}."
message += " Greetings from your first Greengrass component."
# Print the message to stdout.
print(message)

# Append the message to the log file.
with open('/tmp/Greengrass_HelloWorld.log', 'a') as f:
    print(message, file=f)
```

Local에서 component를 개발 및 테스트

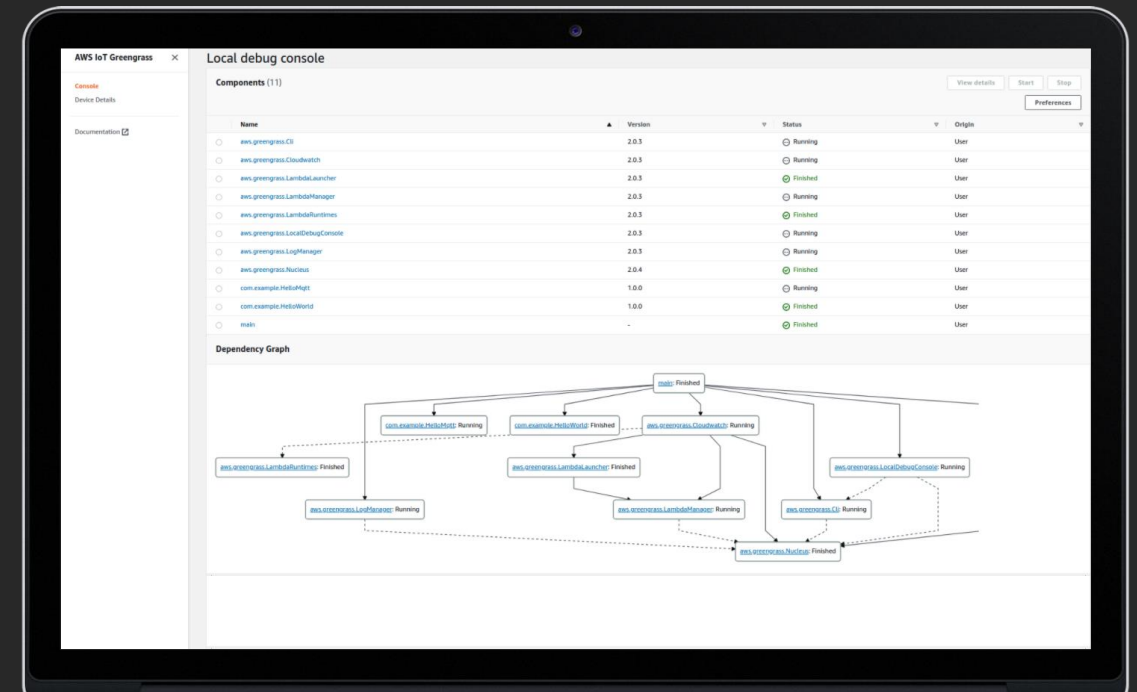
AWS IoT Greengrass CLI

디바이스에서 로컬로 애플리케이션을 개발하고 디버깅할 수 있는 새로운 CLI

```
setch@setch-NUC10i3FNK: ~  
setch@setch-NUC10i3FNK:~$ /greengrass/v2/bin/greengrass-cli help  
Usage: greengrass-cli [-hv] [--ggcRootPath=<ggcRootPath>] [COMMAND]  
Greengrass command line interface  
  
    --ggcRootPath=<ggcRootPath>  
        The AWS IoT Greengrass V2 root directory.  
-h, --help      Show this help message and exit.  
-V, --version    Print version information and exit.  
Commands:  
  help          Show help information for a command.  
  component      Retrieve component information and stop or restart  
                  components.  
  deployment     Create local deployments and retrieve deployment status.  
  logs           Analyze Greengrass logs.  
  get-debug-password Generate a password for use with the HTTP debug view  
                  component.  
setch@setch-NUC10i3FNK:~$
```

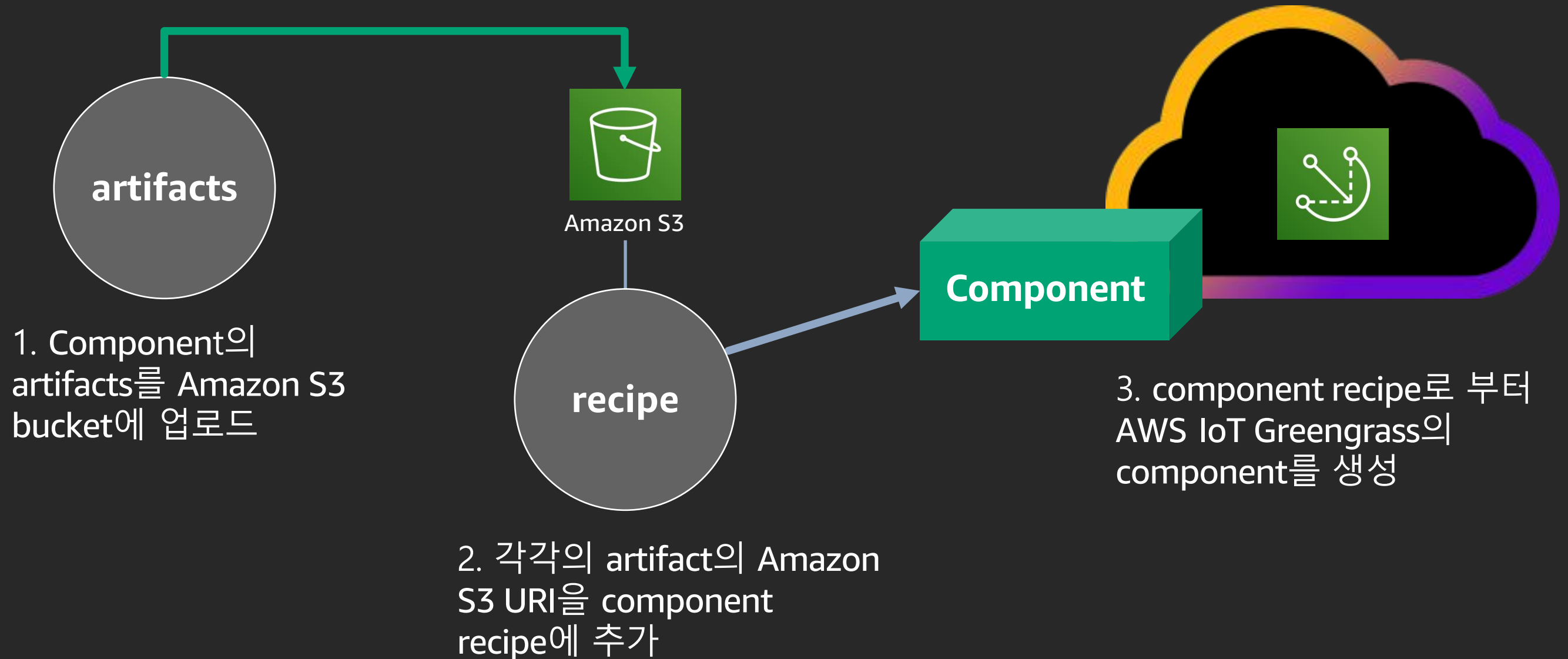
Local debug console

디바이스에서 애플리케이션을 시각적으로 디버깅



Upload your component

✓ Component 개발을 완료하면 AWS 클라우드에 업로드할 수 있습니다.



Deploy component

Improved fleet deployment features

AWS IoT Thing group을 이용하여 deployment를 수행할 코어 디바이스의 목록을 지정

Deployment가 롤 아웃(roll-out)되는 속도와 deployment가 실패하는 조건을 구성 가능

Overview

AWS IoT Greengrass uses continuous AWS IoT Jobs to deploy to thing groups.

Target UbuntuGreengrassCoreGroup ↗	Target type Thing group	Deployment created 17 days ago
IoT Job 26ead4a1-3ae6-4bf8-951d-095c149f833b ↗	Status Active	

Target core devices (1)

Add core device to target group

Find by device name

< 1 > ⚙

Core device name	Status	Status reported
GG-NUC10i3FNK	Healthy	17 days ago

Components (1)

View configuration

< 1 >

Name	Version
aws.greengrass.Cli	2.0.3

Advanced deployment configurations

Rollout configuration
This deployment doesn't specify a rollout configuration.

Timeout configuration
This deployment doesn't specify a timeout configuration.

Cancel configuration
This deployment doesn't specify a cancel configuration.

Component update policy
Notify components, Component update response timeout: 60 seconds

Configuration validation response timeout
60 seconds

Failure handling policy
Don't roll back

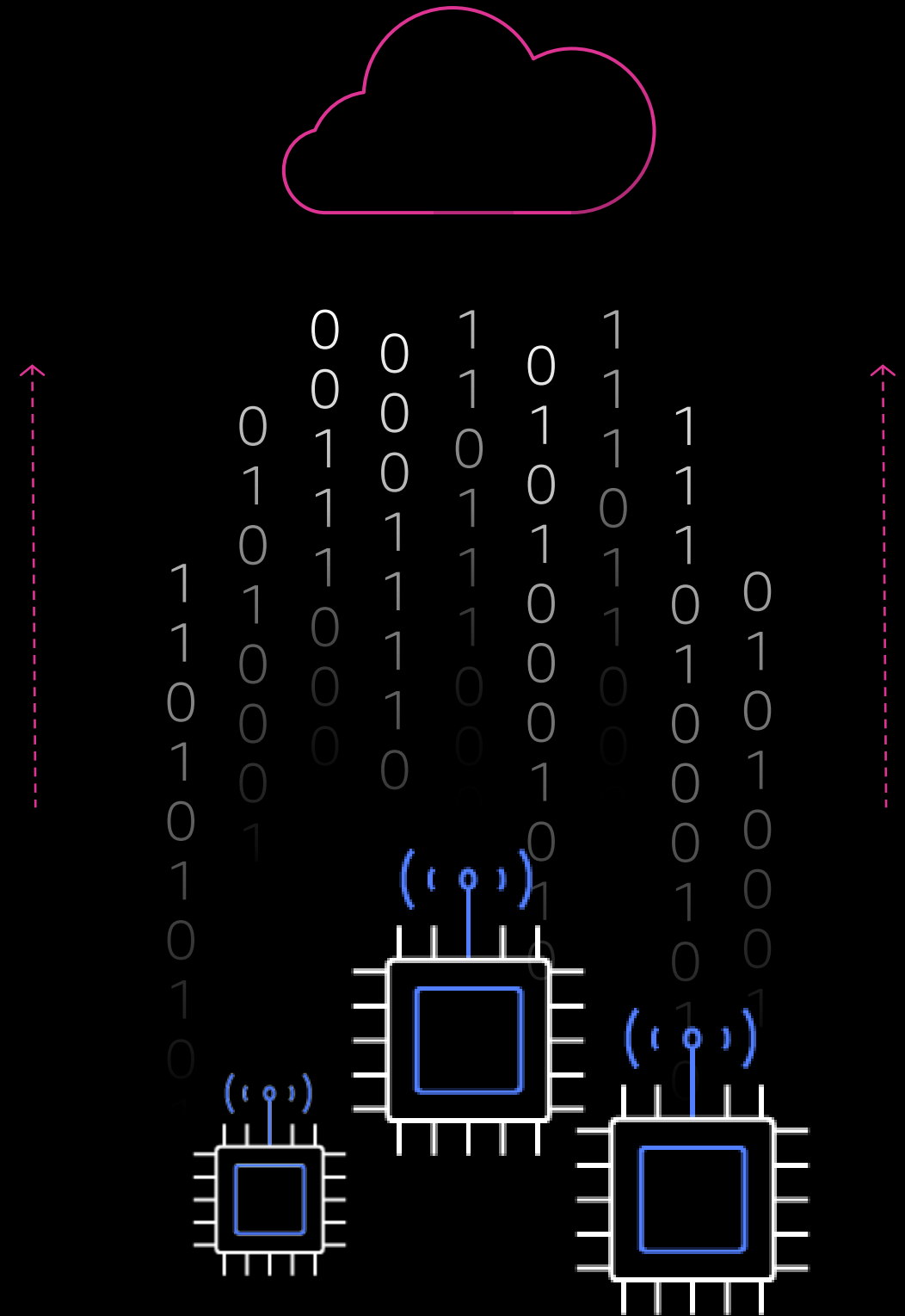
AWS Partner Device Catalog



AWS IoT와 호환되는 올바른 디바이스를 어떻게 찾습니까?

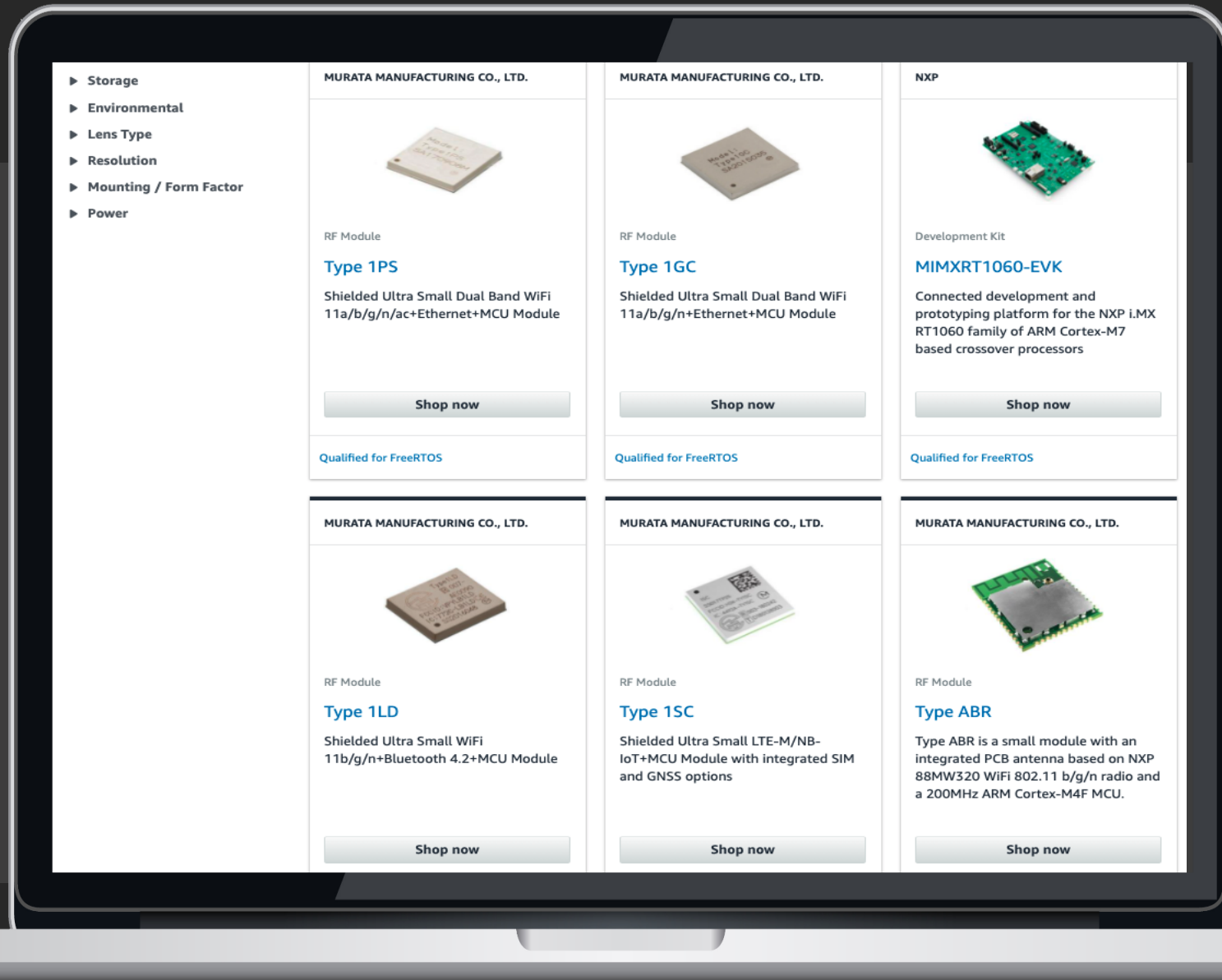


Device
Software



AWS Partner Device Catalog

기본적으로 AWS와 함께 사용할 수 있는 검증된 파트너 하드웨어 및 디바이스를 검색하십시오.



340+

Qualified IoT
devices in the Partner
Device Catalog.

<https://devices.amazonaws.com/>

감사합니다

