

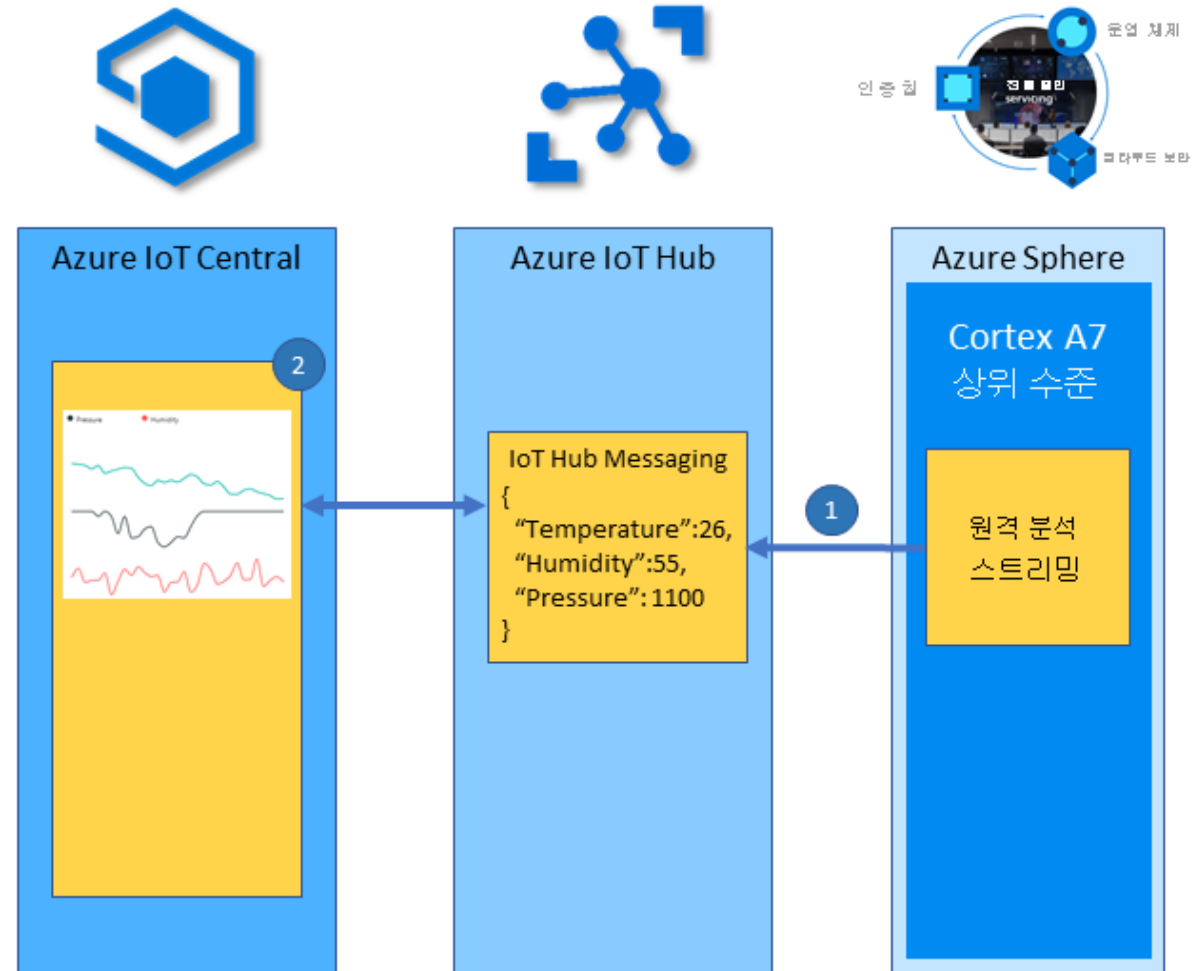
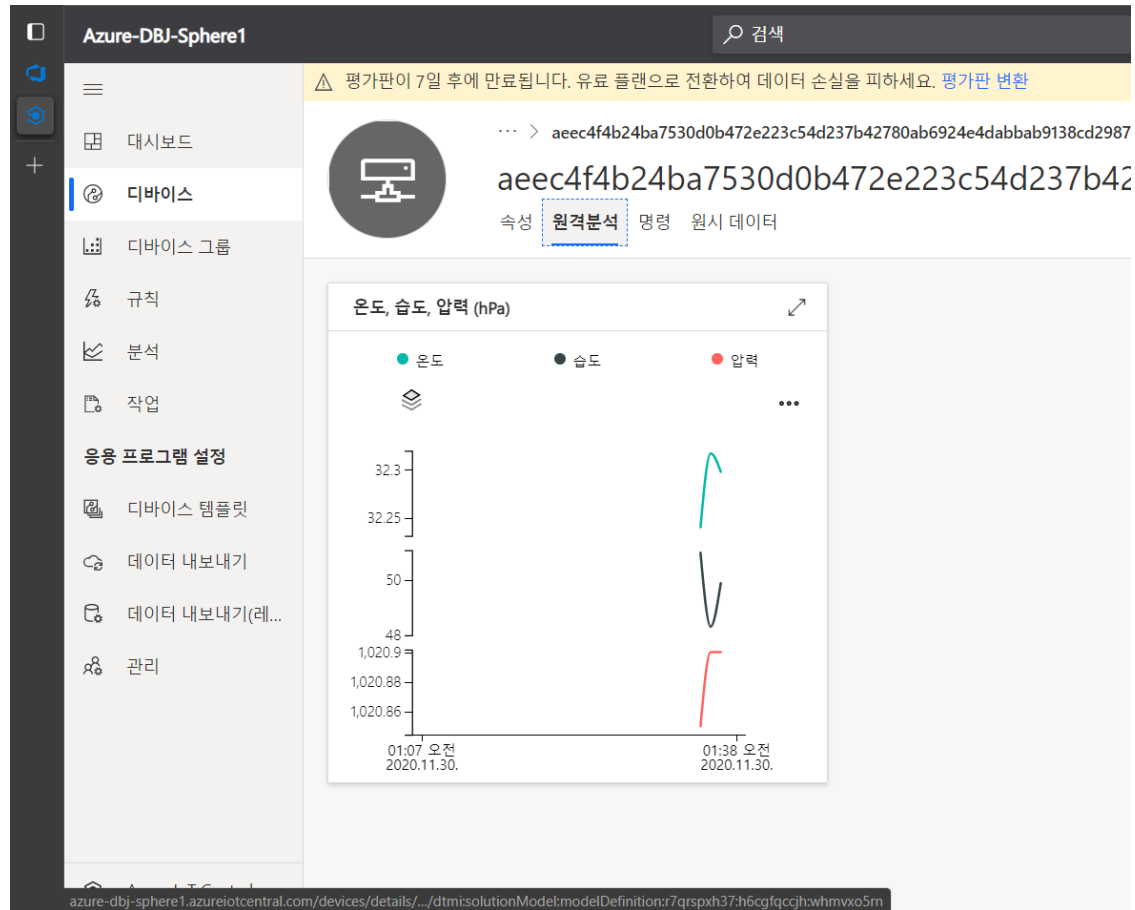


# Azure Sphere

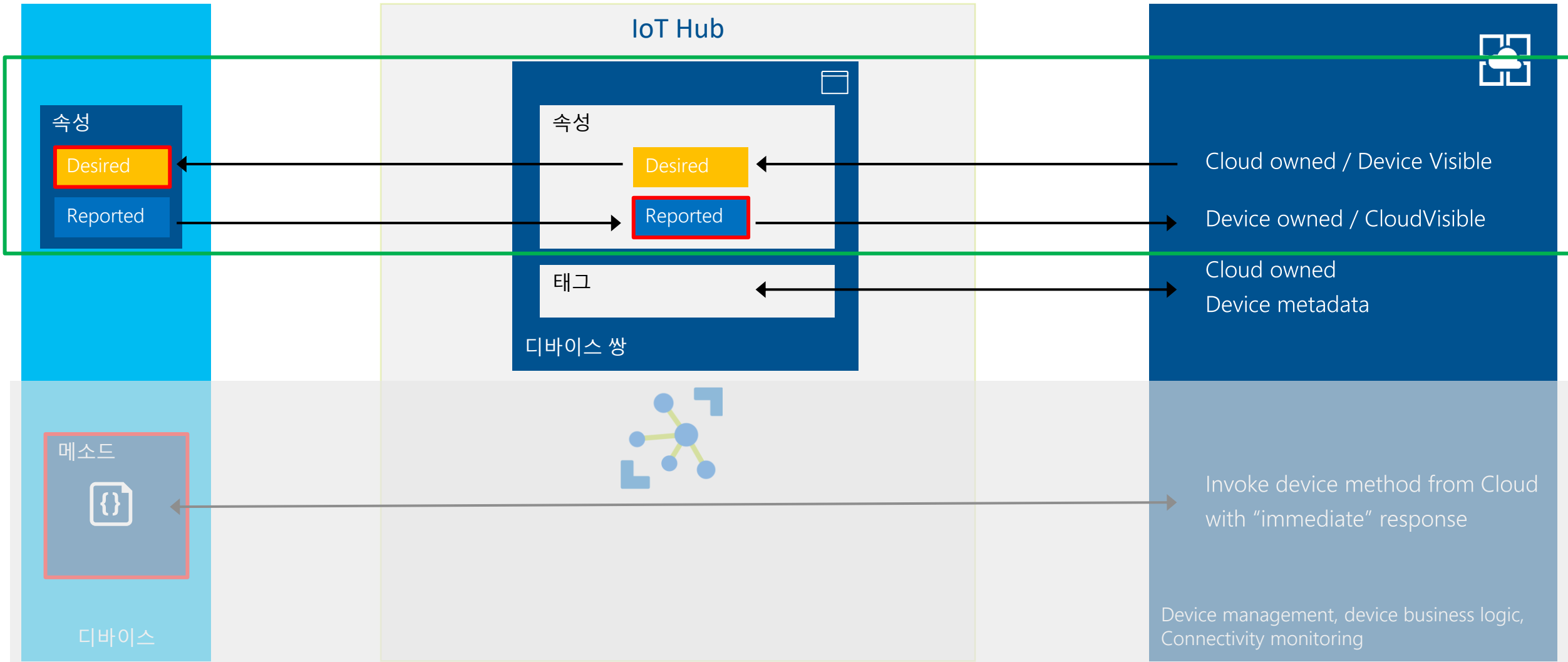
## Episode 4 : Azure IoT 디바이스 쌍 / 직접 메소드로 제어

윤기석  
마이크로소프트

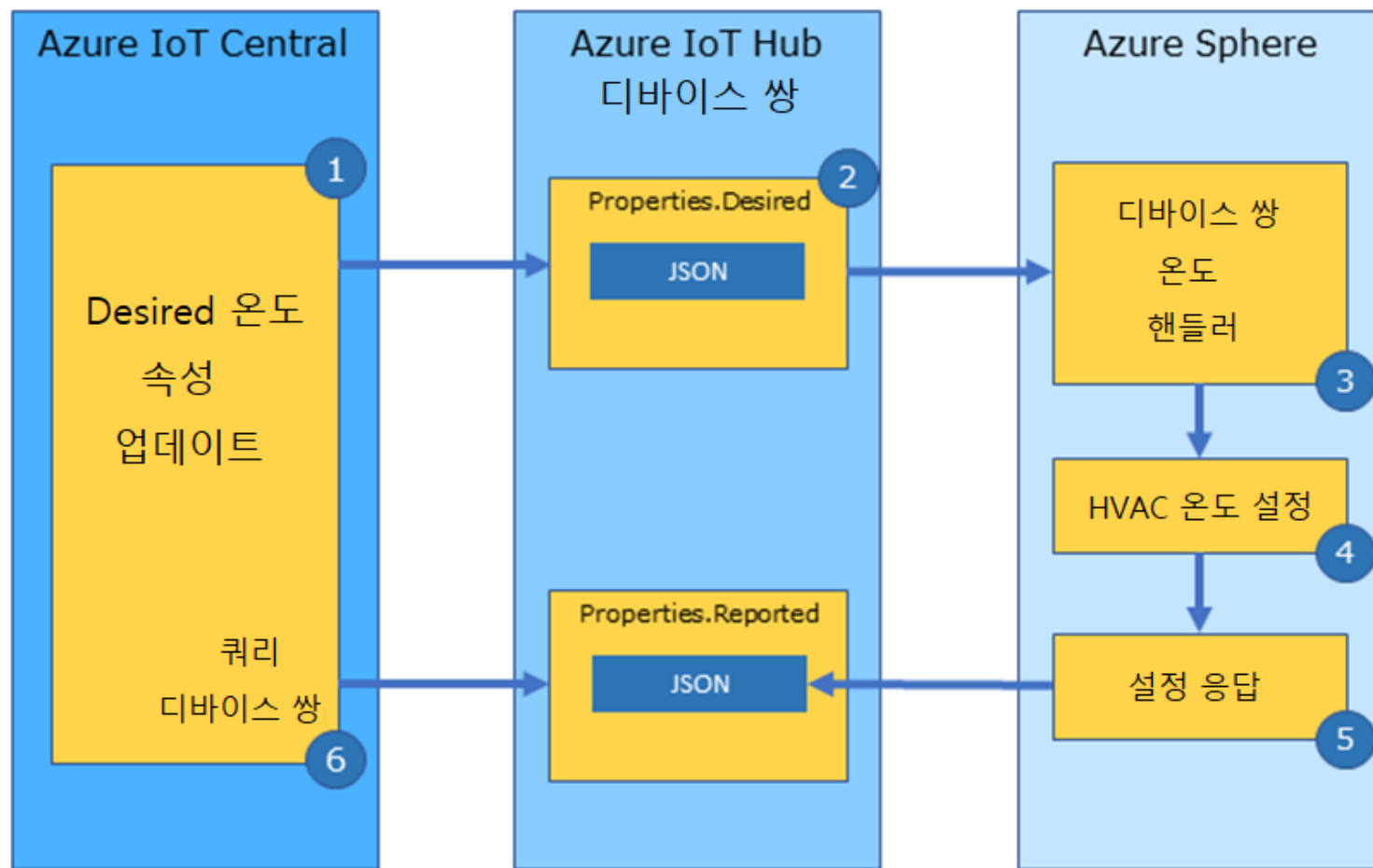
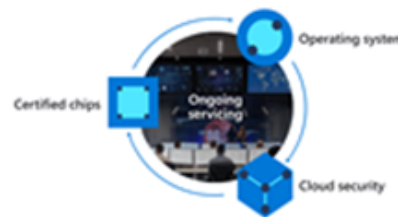
# 지난 시간에 한 것 - 원격 모니터링



# IoT Hub 디바이스 관리



# Azure IoT Central 에서 냉/난방/공조 장치 제어



# 디바이스 쌍 사용 - (Desired)

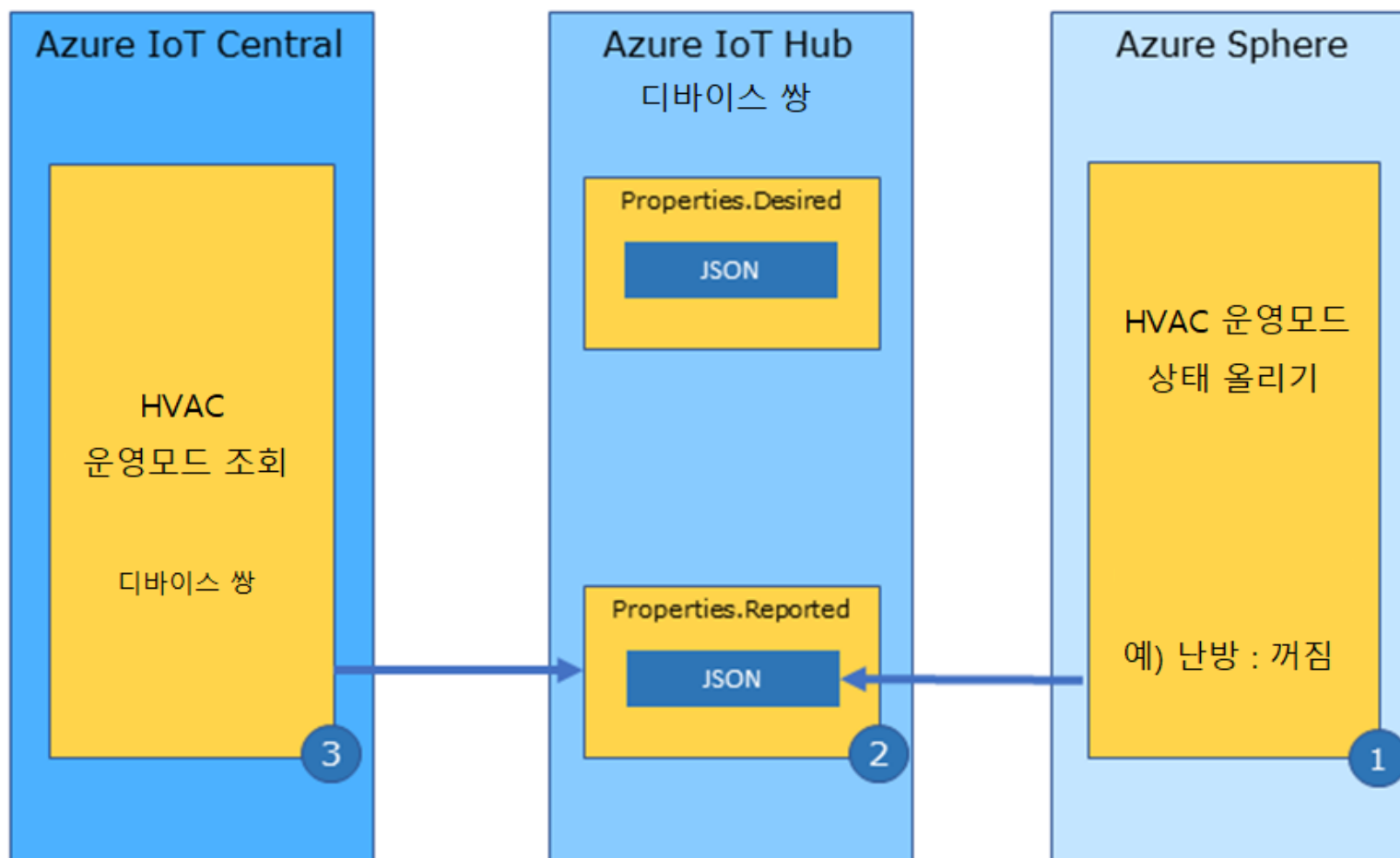
- 디바이스 쌍 속성 바인딩

```
static LP_DEVICE_TWIN_BINDING dt_desiredTemperature = {  
    .twinProperty = "DesiredTemperature",  
    .twinType = LP_TYPE_FLOAT,  
    .handler = DeviceTwinSetTemperatureHandler };
```

- 희망 온도 설정 (DeviceTwinSetTemperatureHandler)

```
/// <summary>  
/// Device Twin Handler to set the desired temperature value  
/// </summary>  
static void DeviceTwinSetTemperatureHandler(LP_DEVICE_TWIN_BINDING* deviceTwinBinding)  
{  
    // validate data is sensible range before applying  
    if (deviceTwinBinding->twinType == LP_TYPE_FLOAT && *(float*)deviceTwinBinding->twinState >= -20.0f && *(float*)deviceTwinBinding->twinState <= 80.0f)  
    {  
        lp_deviceTwinAckDesiredState(deviceTwinBinding, deviceTwinBinding->twinState, LP_DEVICE_TWIN_COMPLETED);  
        SetHvacStatusColour((int)previous_temperature);  
    }  
    else {  
        lp_deviceTwinAckDesiredState(deviceTwinBinding, deviceTwinBinding->twinState, LP_DEVICE_TWIN_ERROR);  
    }  
}
```

# 현재 HVAC 운영 모드 상태 올리기



# 디바이스 쌍 사용 - (Reported)

- 디바이스 쌍 속성 바인딩

```
static LP_DEVICE_TWIN_BINDING dt_reportedHvacState = {  
    .twinProperty = "ReportedHvacState",  
    .twinType = LP_TYPE_STRING };
```

- 디바이스 -> 클라우드 단방향은 핸들러 함수 필요x

- 디바이스 쌍 속성 Report

```
lp_deviceTwinReportState(&dt_reportedHvacState, (void*)hvacState[(int)current_led]);
```

# 디바이스 쌍 사용

- 선언한 디바이스 쌍 바인딩을 핸들러에 매핑

```
LP_DEVICE_TWIN_BINDING* deviceTwinBindingSet[] = { &dt_desiredTemperature, &dt_reportedTemperature, &dt_reportedHvacState };
```

Azure IoT Hub -> 디바이스 디바이스 쌍 메시지 수신 시 deviceTwinBindingSet 배열에 있는 twinProperty 이름에 선언된 핸들러 함수가 호출됨

- 디바이스 쌍 바인딩 세트 열기 (deviceTwinBindingSet[ ])

```
lp_deviceTwinSetOpen(deviceTwinBindingSet, NELEMS(deviceTwinBindingSet));
```

main.c 의 InitPeripheralsAndHandlers 함수에서 호출됨

- 디바이스 쌍 바인딩 세트 닫기 (deviceTwinBindingSet[ ])

```
lp_deviceTwinSetClose();
```

main.c 의 ClosePeripheralsAndHandlers 함수에서 호출됨



# Azure IoT Central 디바이스 속성

The screenshot displays the Azure IoT Central interface for configuring a device template. The left sidebar shows the navigation menu with '디바이스 템플릿' (Device Template) selected. The main area shows the 'Interface' configuration for the 'Azure Sphere' device template. The configuration includes several properties, with the 'DesiredTemperature' property being the focus. The '스키마' (Schema) is set to 'Float', and the '쓰기 가능' (Writable) toggle is turned on. The '단위' (Unit) is set to '°C'. The 'Property' and 'Temperature' fields are also visible.

디바이스 템플릿 – Interface 에서 정의

예) DesiredTemperature 속성 스키마 형식은 Float / 쓰기 가능 (클라우드 -> 디바이스)

속성 이름(DesiredTemperature) 은 디바이스 쌍 바인딩 선언과 같아야 함

# 직접 해보기 - HL\_app 배포 (Cortex-A7)

- Lab\_3\_Device\_Twins 폴더 열기
- app\_manifest 파일에 메모장 내용 붙여넣기
  - AllowedConnections / DeviceAuthentication 항목

대시보드

디바이스

디바이스 그룹

규칙

분석

작업

응용 프로그램 설정

디바이스 템플릿

데이터 내보내기

데이터 내보내기(레...

관리

aeec4f4b24ba7530d0b472e223c54

속성 원격분석 명령 원시 데이터

저장

섹션

마지막으로 보고된 디바이스 Reboot 시간

마지막으로 보고된 HVAC 운영 모드

cooling

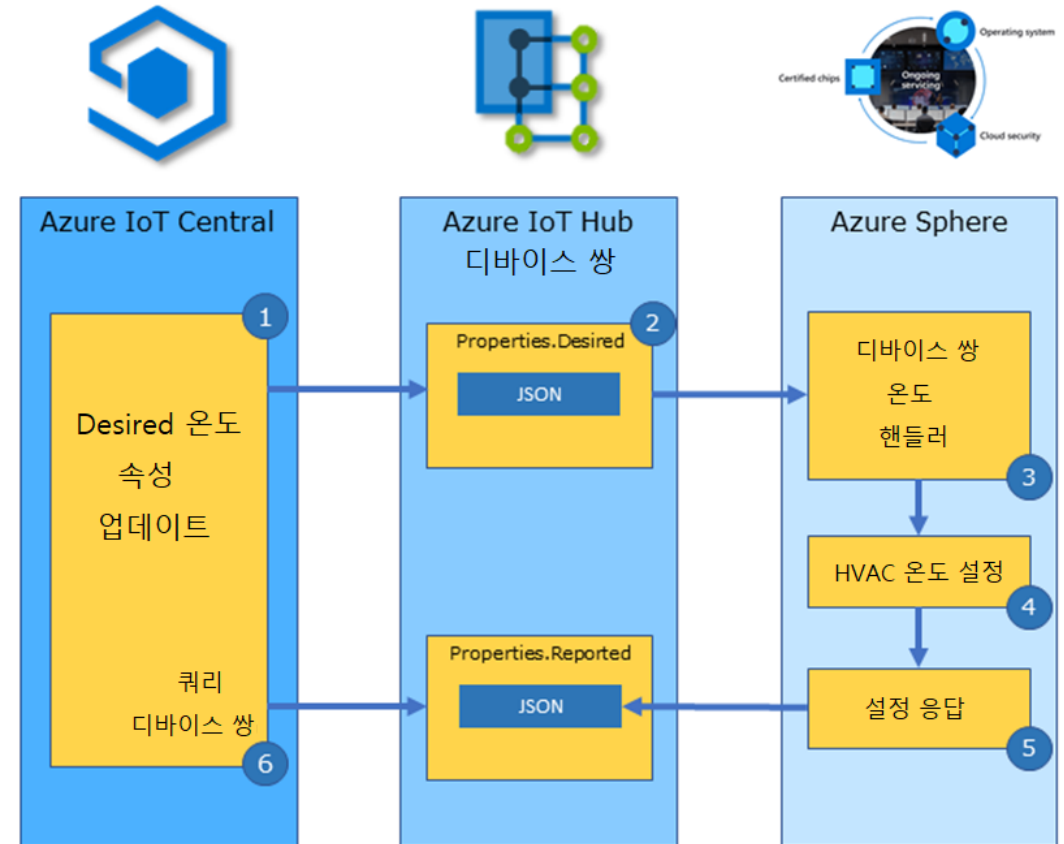
마지막으로 보고된 온도 (°C)

32.226562

희망온도 (°C)

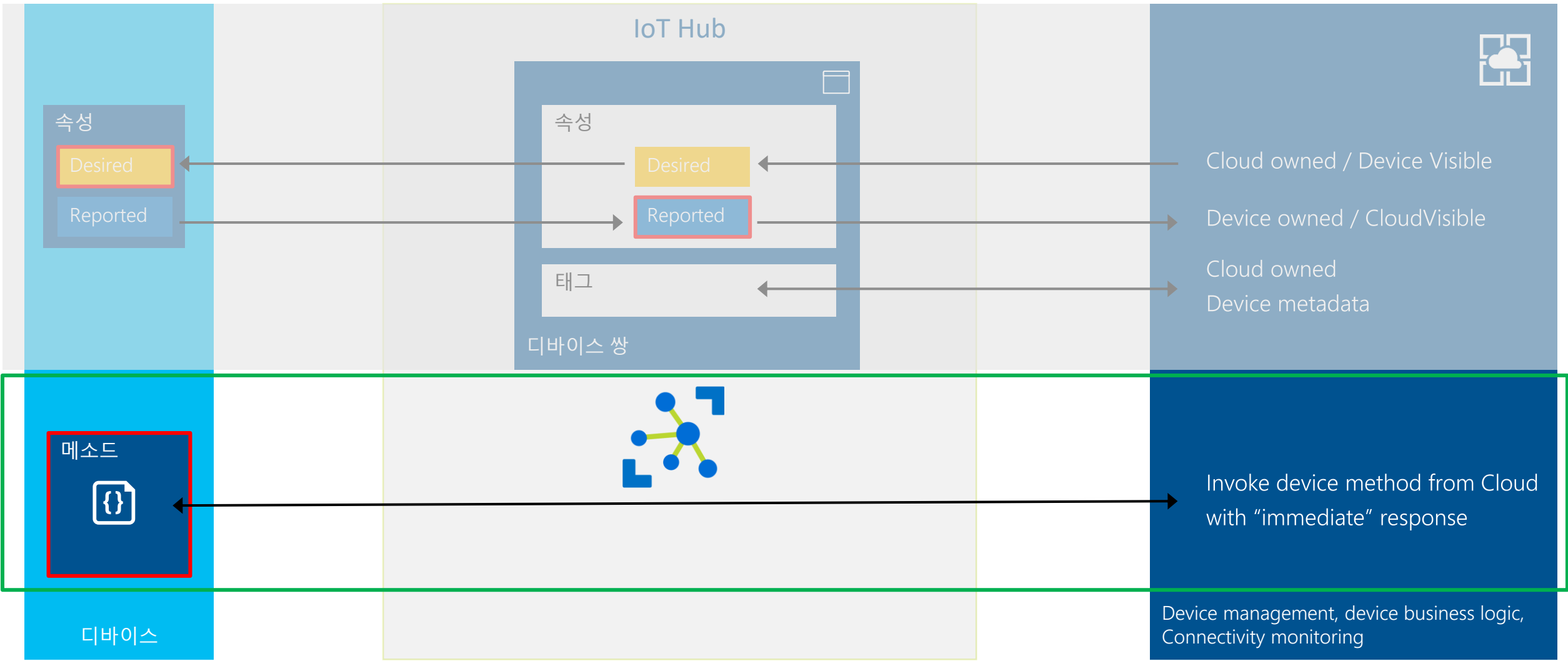
26

수락됨: 지금

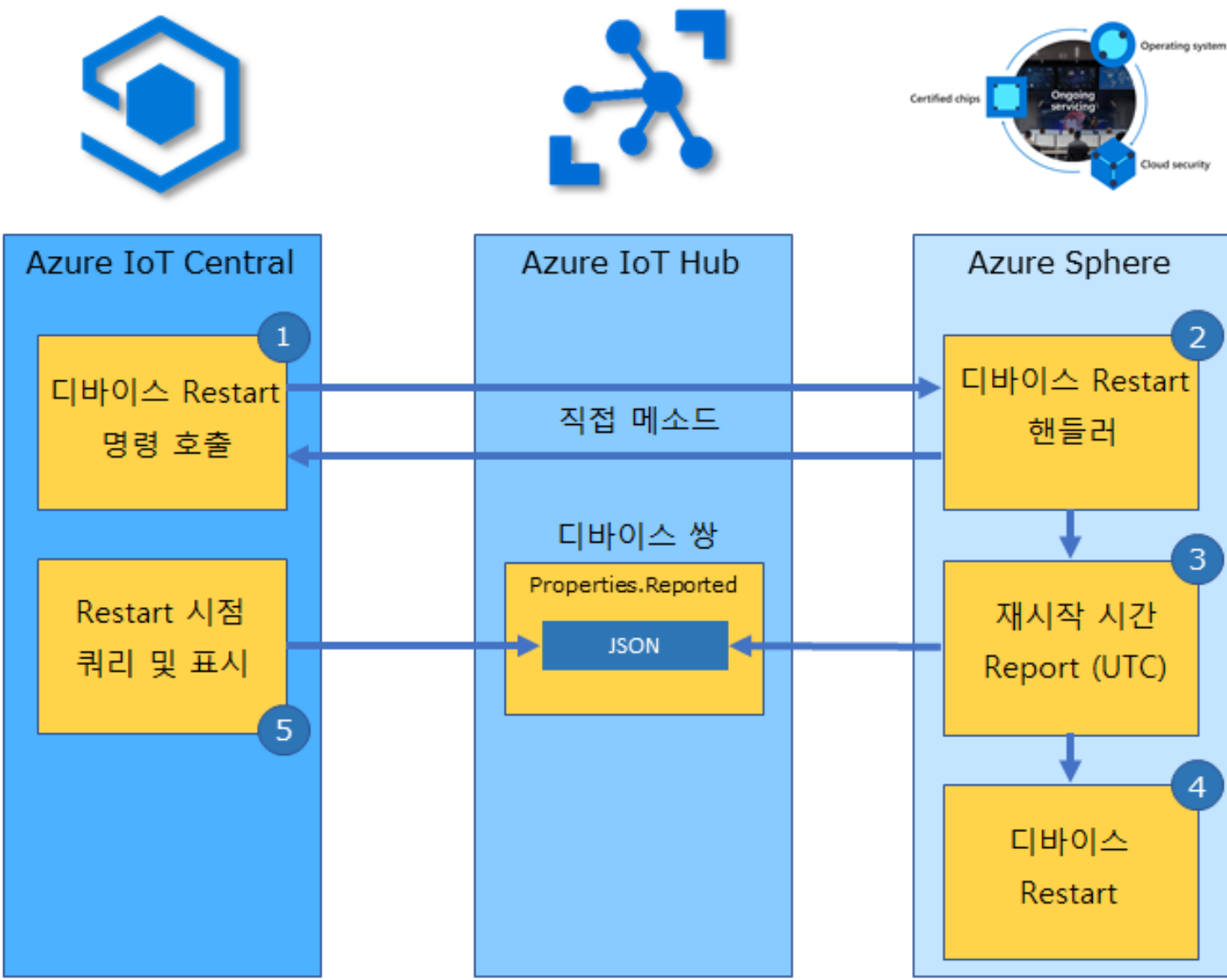


- 보드 상태 LED
  - cooling = 파란색 LED
  - heating = 빨간색 LED
  - off = 녹색 LED

# Azure IoT 직접 메소드로 Azure Sphere 원격제어하기



# Azure Sphere 원격제어 – restart



# 직접 메소드 사용

- 직접 메소드 바인딩

```
static LP_DIRECT_METHOD_BINDING dm_restartDevice = {  
    .methodName = "RestartDevice",  
    .handler = RestartDeviceHandler };
```

- 직접 메소드 핸들러

```
/// Start Device Power Restart Direct Method 'ResetMethod' integer seconds eg 5  
static LP_DIRECT_METHOD_RESPONSE_CODE RestartDeviceHandler(JSON_Value* json, LP_DIRECT_METHOD_BINDING* directMethodBinding, char** responseMsg)  
{  
    const size_t responseLen = 60; // Allocate and initialize a response message buffer. The calling function is responsible for the freeing memory  
    static struct timespec period; *responseMsg = (char*)malloc(responseLen);  
    memset(*responseMsg, 0, responseLen);  
  
    if (json_value_get_type(json) != JSONNumber) { return LP_METHOD_FAILED; }  
  
    int seconds = (int)json_value_get_number(json);  
  
    // leave enough time for the device twin dt_reportedRestartUtc to update before restarting the device  
    if (seconds > 2 && seconds < 10)  
    {  
        // Report Device Restart UTC  
        lp_deviceTwinReportState(&dt_reportedRestartUtc, lp_getCurrentUtc(msgBuffer, sizeof(msgBuffer))); // LP_TYPE_STRING  
  
        // Create Direct Method Response  
        snprintf(*responseMsg, responseLen, "%s called. Restart in %d seconds", directMethodBinding->methodName, seconds);  
  
        // Set One Shot LP_TIMER period = (struct timespec){ .tv_sec = seconds, .tv_nsec = 0 };  
        lp_timerOneShotSet(&restartDeviceOneShotTimer, &period);  
        return LP_METHOD_SUCCEEDED;  
    }  
}
```

# 직접 메소드 사용

- One-shot 타이머

```
static LP_TIMER restartDeviceOneShotTimer = {  
    .period = {0, 0},  
    .name = "restartDeviceOneShotTimer",  
    .handler = DelayRestartDeviceTimerHandler };
```

- Restart 핸들러

```
/// <summary>  
/// Restart the Device  
/// </summary>  
static void DelayRestartDeviceTimerHandler(EventLoopTimer* eventLoopTimer)  
{  
    if (ConsumeEventLoopTimerEvent(eventLoopTimer) != 0)  
    {  
        lp_terminate(ExitCode_ConsumeEventLoopTimeEvent);  
    }  
    else {  
        PowerManagement_ForceSystemReboot();  
    }  
}
```

- app\_manifest.json 에 리소스 권한 추가

```
"PowerControls": [  
    "ForceReboot"  
]
```

# 직접 메소드 사용

- 선언한 직접 메소드 바인딩을 핸들러에 매핑

```
LP_DIRECT_METHOD_BINDING* directMethodBindingSet[] = { &dm_restartDevice };
```

Azure IoT Hub -> 디바이스 직접 메소드 메시지 수신 시 directMethodBindingSet 배열에 있는 methodName 과 연결된 핸들러 함수가 호출됨

- 직접 메소드 바인딩 세트 열기 (directMethodBindingSet[ ])

```
lp_directMethodSetOpen(directMethodBindingSet, NELEMS(directMethodBindingSet));
```

main.c 의 InitPeripheralsAndHandlers 함수에서 호출됨

- 직접 메소드 바인딩 세트 닫기 (directMethodBindingSet[ ])

```
lp_directMethodSetClose();
```

main.c 의 ClosePeripheralsAndHandlers 함수에서 호출됨

# Azure IoT Central 명령

대시보드

디바이스

디바이스 그룹

규칙

분석

작업

응용 프로그램 설정

디바이스 템플릿

데이터 내보내기

데이터 내보내기(레...

관리

버전 테스트 디바이스 관리 게시 이름 바꾸기 삭제

디바이스 템플릿 > Azure Sphere > 모델 > Azure Sphere > Interface

Azure Sphere

애플리케이션이 업데이트됨: 그저께 인터페이스가 게시됨: 그저께

모델

Azure Sphere

Interface

클라우드 속성

사용자 지정

뷰

원격분석

저장 + 기능 추가 ID 보기 버전 내보내기 삭제

희망온도	DesiredTemperature	Property	Temperature
마지막으로 보고된 온도	ReportedTemperature	Property	Temperature
마지막으로 보고된 디...	ReportedRestartUTC	Property	없음
마지막으로 보고된 HV...	ReportedHvacState	Property	없음
디바이스 다시 시작	RestartDevice	Command	

바인딩에 선언된 이름과 같아야 함

디바이스 다시 시작

RestartDevice

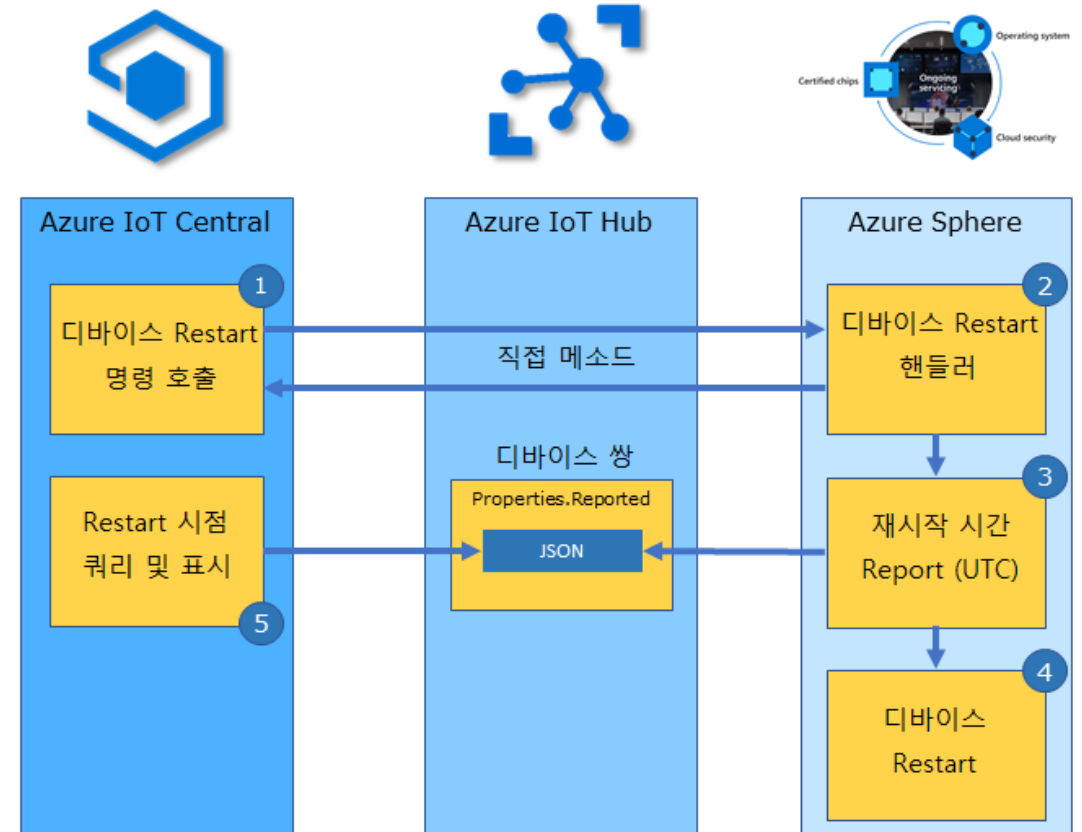
Command



# 직접 해보기 - HL\_app 배포 (Cortex-A7)

- Lab\_4\_Device\_Methods 폴더 열기
- app\_manifest 파일에 메모장 내용 붙여넣기
  - AllowedConnections / DeviceAuthentication 항목

The screenshot shows the Azure IoT Central console interface. On the left, a sidebar menu has '디바이스' (Devices) highlighted with a red box. The main area displays the details for a specific device with ID 'aeec4f4b24ba7530d0b472e223c54d237b4'. Below the device ID, there are tabs for '속성' (Properties), '원격분석' (Remote Analysis), '명령' (Commands), and '원시 데이터' (Raw Data), with '명령' highlighted by a red box. The '명령' tab shows a command titled 'Restart Device (Between 3 and 9 seconds inclusive)'. A text input field contains the number '5', which is also highlighted with a red box. Below the input field is a blue button labeled '실행' (Execute), also highlighted with a red box. At the bottom, a note says '응답을 보려면 명령 기록을 확인하세요.' (Check the command log to see the response).



# 다음 에피소드

- **EP1**
  - Azure Sphere 아키텍처 / 개발환경
  - 가상의 보안 IoT 프로젝트 정의
- **EP2**
  - 하드웨어 및 이벤트 기반 프로그래밍 이해
  - Azure Sphere 설정 방법
- **EP3**
  - Azure IoT Central 에 실내 환경 센서를 연결
  - Azure Sphere 를 보호하는 방법
  - Azure Sphere 에 HL App(고급 애플리케이션) 배포
- **EP4**
  - Azure IoT 디바이스 쌍으로 실내 온도 설정
  - Azure IoT 직접 메시드로 Azure Sphere 원격 제어
- **EP5**
  - Azure RTOS 실시간 센서 앱 배포 / 실내 환경 모니터링
- **EP6**
  - Azure RTOS 실시간 실내 환경 센서 데이터를 IoT Central 에 전송
- **EP7**
  - 간단하게 OTA 업데이트 사용하기