Introduction to Azure IoT hub

Loy Vanich (laploy@gmail.com 084 007 5544)

Azure IoT Suite: IoT Hub

Designed for IoT

Connect millions of devices to a partitioned application back-end

Service assisted communications

Devices are not servers

Use IoT Hub to enable secure bi-directional comms

Cloud-scale messaging

Device-to-cloud and Cloud-to-device

Durable messages (*at least once* semantics)

Cloud-facing feedback

Delivery receipts, expired messages

Device communication errors

Per-device authentication

Individual device identities and credentials

Connection multiplexing

Single device-cloud connection for all communications (C2D, D2C)

Multi-protocol

Natively supports AMQP, HTTP

Designed for extensibility to custom protocols

Multi-platform

Device SDKs available for multiple platforms (e.g. RTOS, Linux, Windows)

Multi-platform Service SDK.

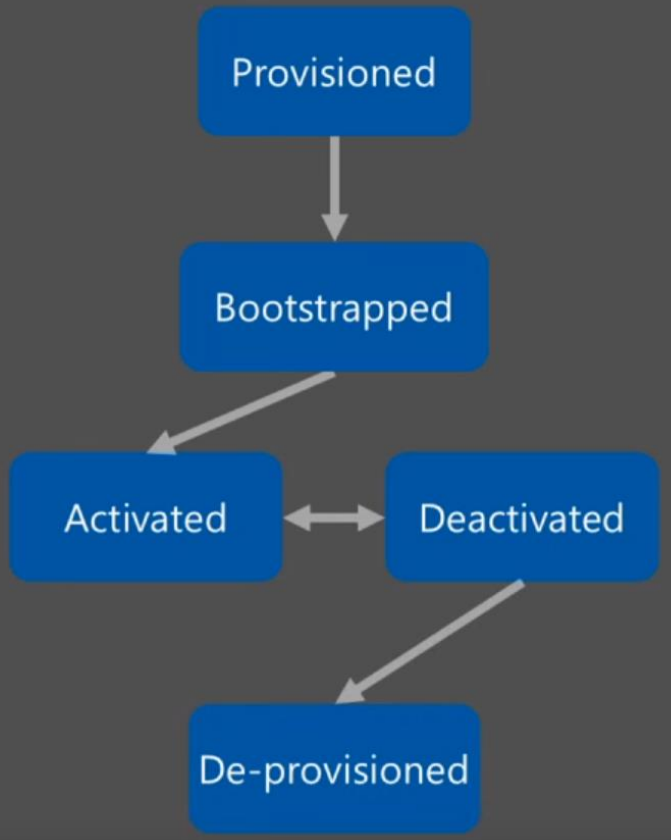
Device provisioning

Making devices known to your system

- Many systems involved
  (IoT Hub, device registry, ERPs, …)
- Device identity
  (composite devices, many concerns)

Sample provisioning

1. Device **provisioned** at manufacturing into system
2. Device connects for the first time and gets associated to its regional data center (**bootstrapped**)
3. As a result of customer interactions the device is **activated**
4. Devices can be **deactivated** for security and other reasons
5. A device can also be **de-provisioned** at end-of-life or decommission.



Device-to-cloud messages

Interface

AMQP and HTTPS device-side endpoint
AMQP service-side endpoint
Device and service SDKs

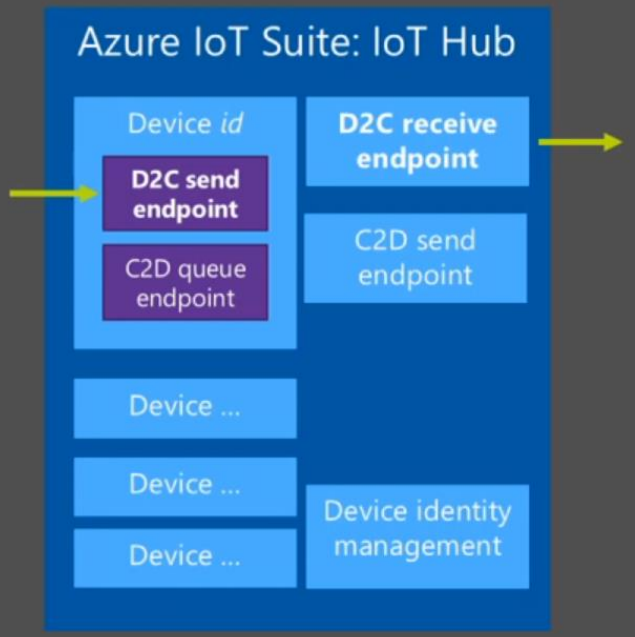
Compatible with Event Hubs

Partitioned receiver, client check-pointing
Integrations with Azure Stream Analytics, Storm, …

IoT Hub services for D2C

Millions of simultaneously connected devices
Per-device authentication
Connection-multiplexing:

- C2D and D2C traffic
- Across multiple devices for gateway scenarios



Cloud-to-device messages

Interface

AMQP and HTTPS device-side endpoint
AMQP service-side endpoint

At-least-once semantics

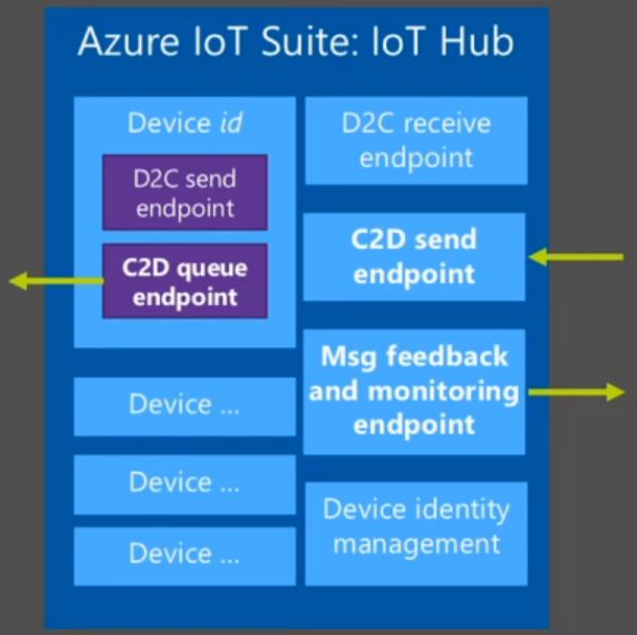
Durable messages
Device acknowledges receipt
(Send - Receive - Abandon OR Complete)

TTL and receipts

Per-message TTL
Per-message positive and negative receipts

Command lifecycle pattern

Use correlated D2C for responses
Use feedback information to retry
Store command state in command registry



\

Monitoring device connectivity

Feedbacks

Device connection/disconnection events

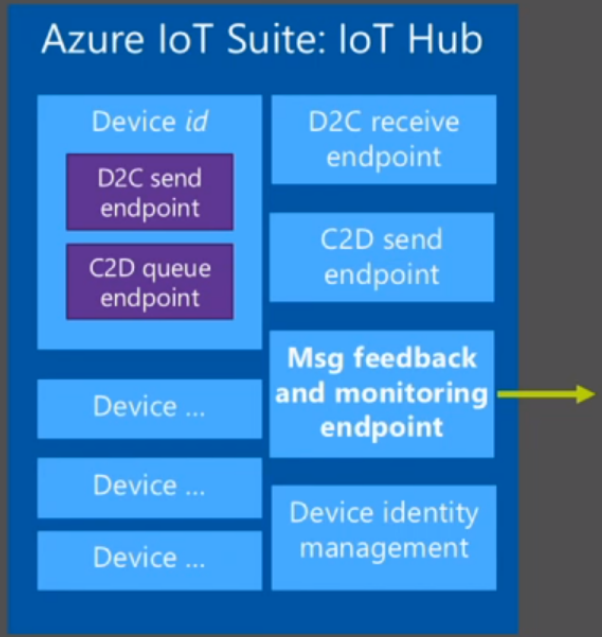
Device error reporting

Event Hub-compatible endpoint

Example

Complex device blocking logic

- Stream Analytics job evaluates:
  number of failed connection attempts per device
- As a result device can be disabled in IoT Hub



What's next?