works with
BY SILICON LABS
VIRTUAL CONFERENCE

GW-201: Gateway Software Architecture – Hosting Diverse Protocols and Apps

SILICON LABS

firia

David Ewing
President

Donnie Pitts
Design Engineer

Meet Your Presenters

David Ewing

President, Firia

Donnie Pitts

Design Engineer, Firia

Connecting to the Cloud

Building the
Whole Product

- Design choices
- Architecture
- Recurring Costs
- Maintenance
- Performance
- User Experience

What's the matter?

- It's all about IP
- That's Internet Protocol, not Intellectual Property!
- You've got an IP connection to your product…
- Now what?
  - Which cloud provider to use?
  - How much will it cost to run this system?
  - Will I need DevOps staff?
  - Am I going to be hacked?

Connectivity First

- **How you’re connecting impacts back-end decisions too!**
- **Embedded Wi-Fi?**
  - Consider impact of constrained CPU, especially on security requirements
  - Can your device support JWT (JSON Web Tokens)?
- **Gateway or Hub**
  - Can you deploy application services here?
  - Software libraries available for all the Clouds?
- **Dealing with multiple wireless PHYs!**

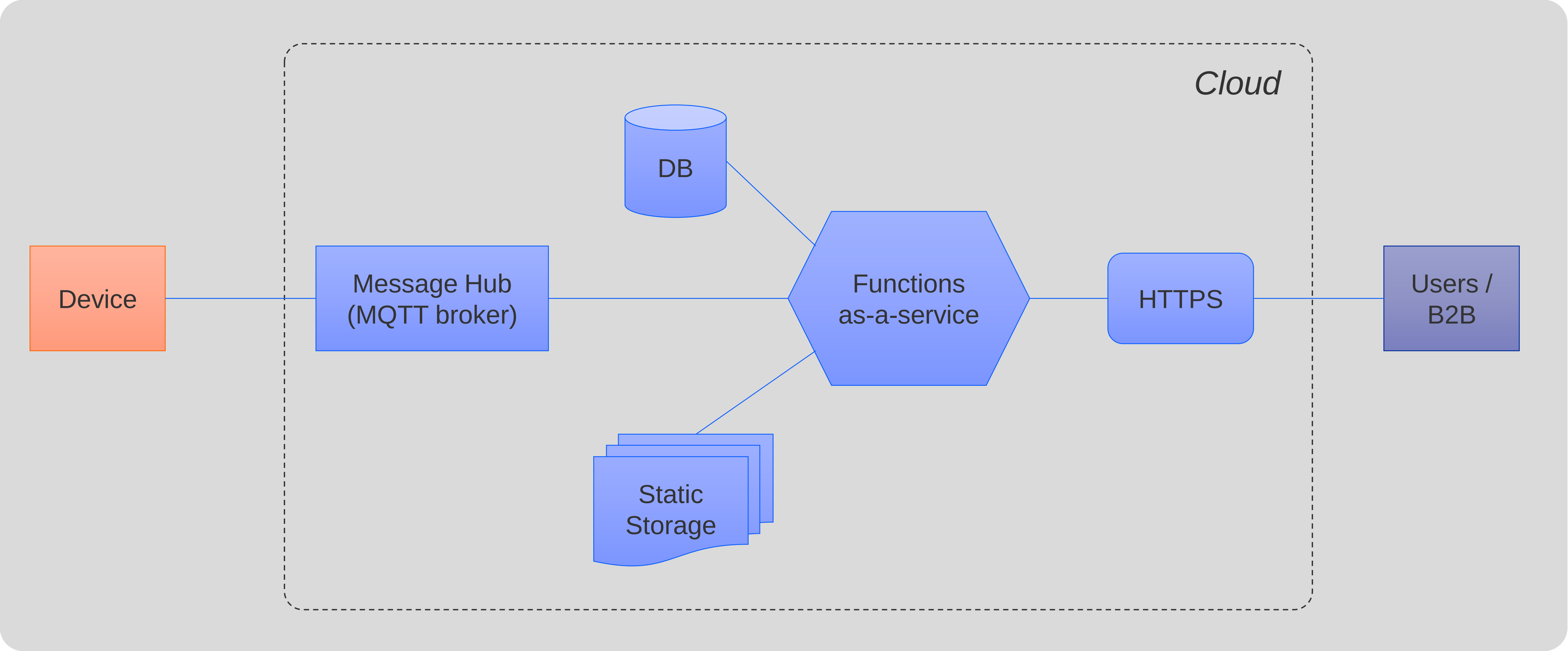
Cloud IoT Architecture

Common patterns for modern IoT Cloud device connectivity, and their demands on your Gateway.

Cloud Providers

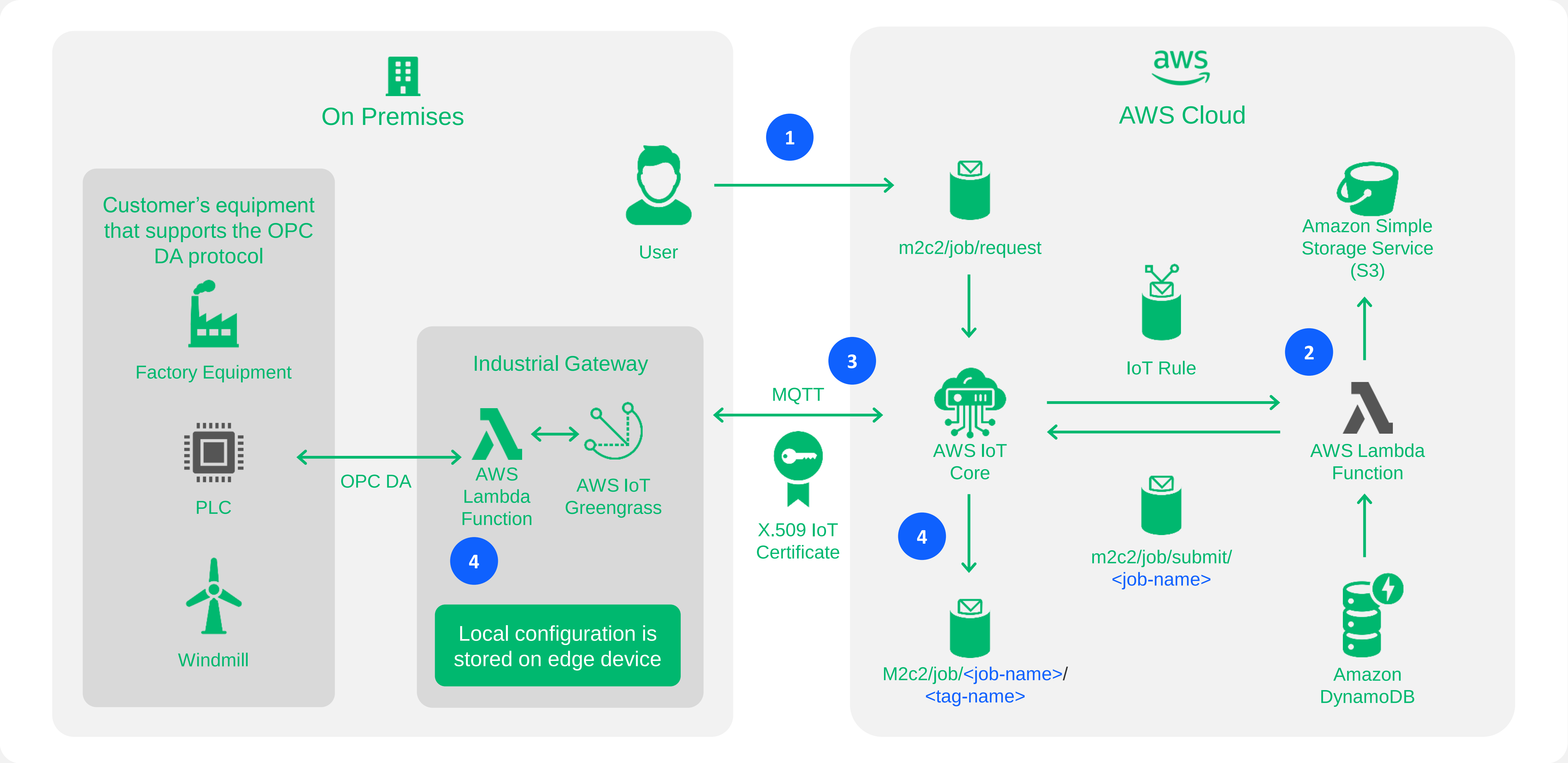
- **The big three:**
  - **AWS** – Amazon Web Services
  - **Azure** – Microsoft Cloud Platform
  - **GCP** – Google Cloud Platform
- **On-Premises options?**
  - Also, “Hybrid-Cloud”
- **Multi-Cloud**
  - A myth to justify haphazard corporate rollouts?
  - Containers and microservices.
  - What cost to avoid vendor lock-in?

A Generic IoT Cloud Architecture

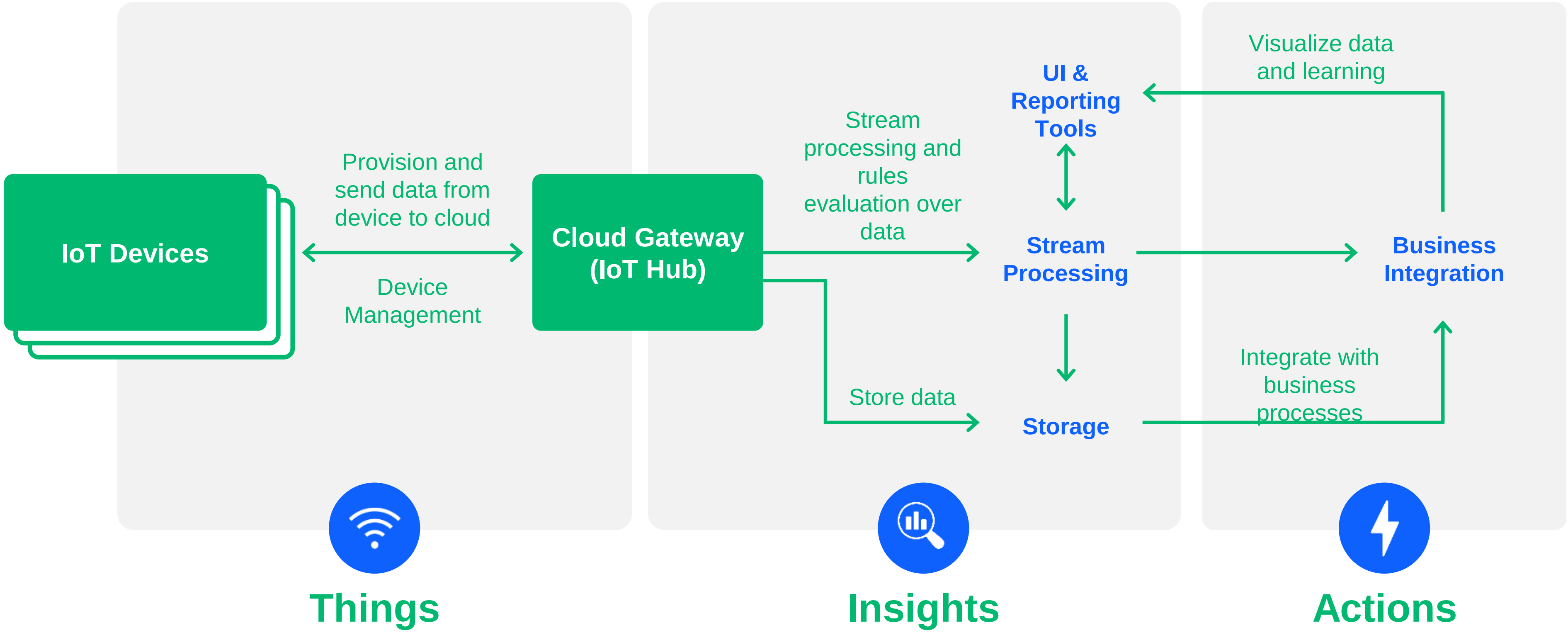


Now let's compare this to the Big Three…

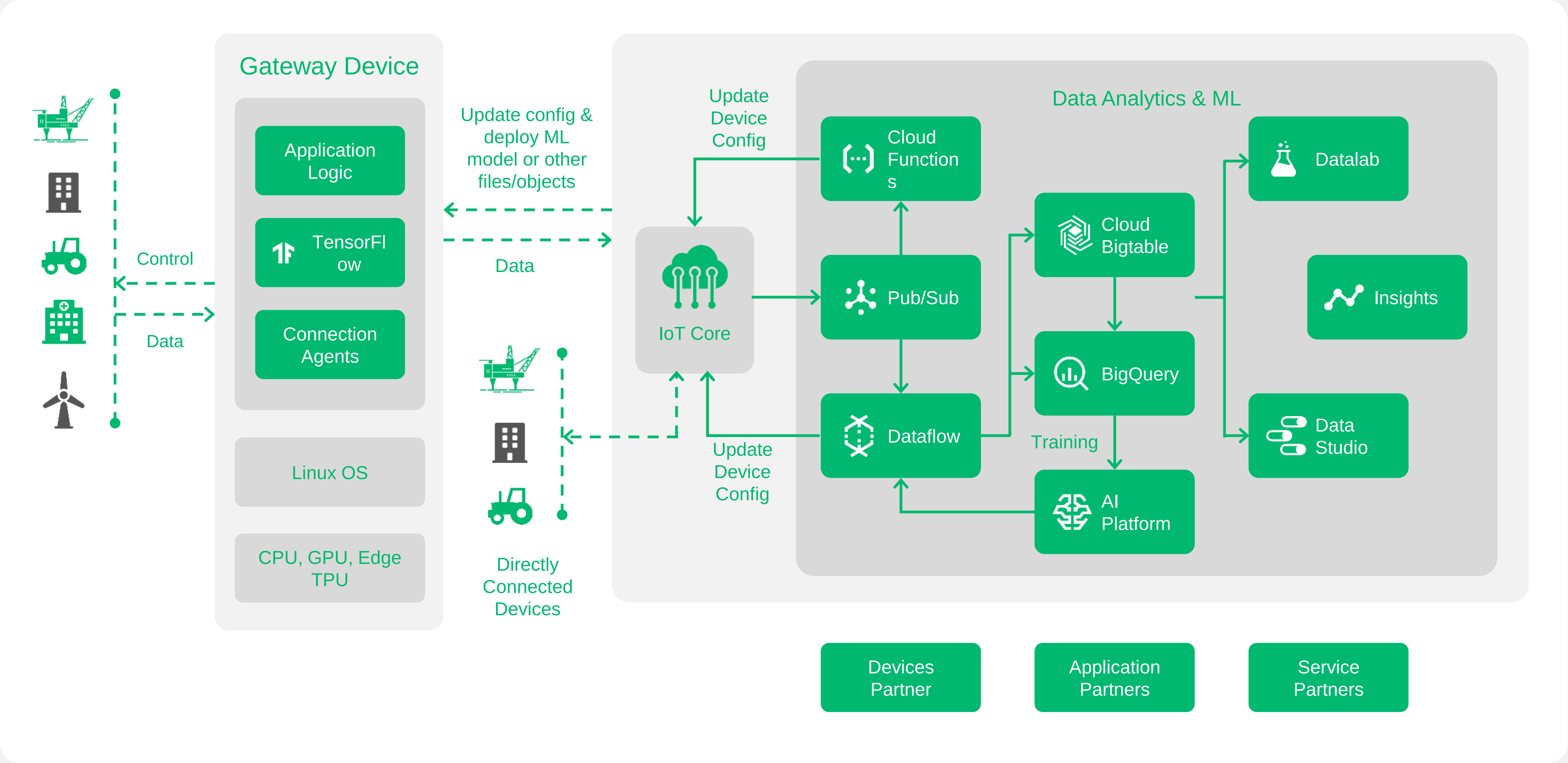
AWS IoT Core


On Premises
Customer's equipment that supports the OPC DA protocol
Factory Equipment
PLC
Windmill
OPC DA
Industrial Gateway
AWS Lambda Function
AWS IoT Greengrass
4
Local configuration is stored on edge device
User
1
3
MQTT
X.509 IoT Certificate
AWS Cloud
m2c2/job/request
AWS IoT Core
4
M2c2/job/<job-name>/<tag-name>
IoT Rule
m2c2/job/submit/<job-name>
Amazon Simple Storage Service (S3)
2
AWS Lambda Function
Amazon DynamoDB

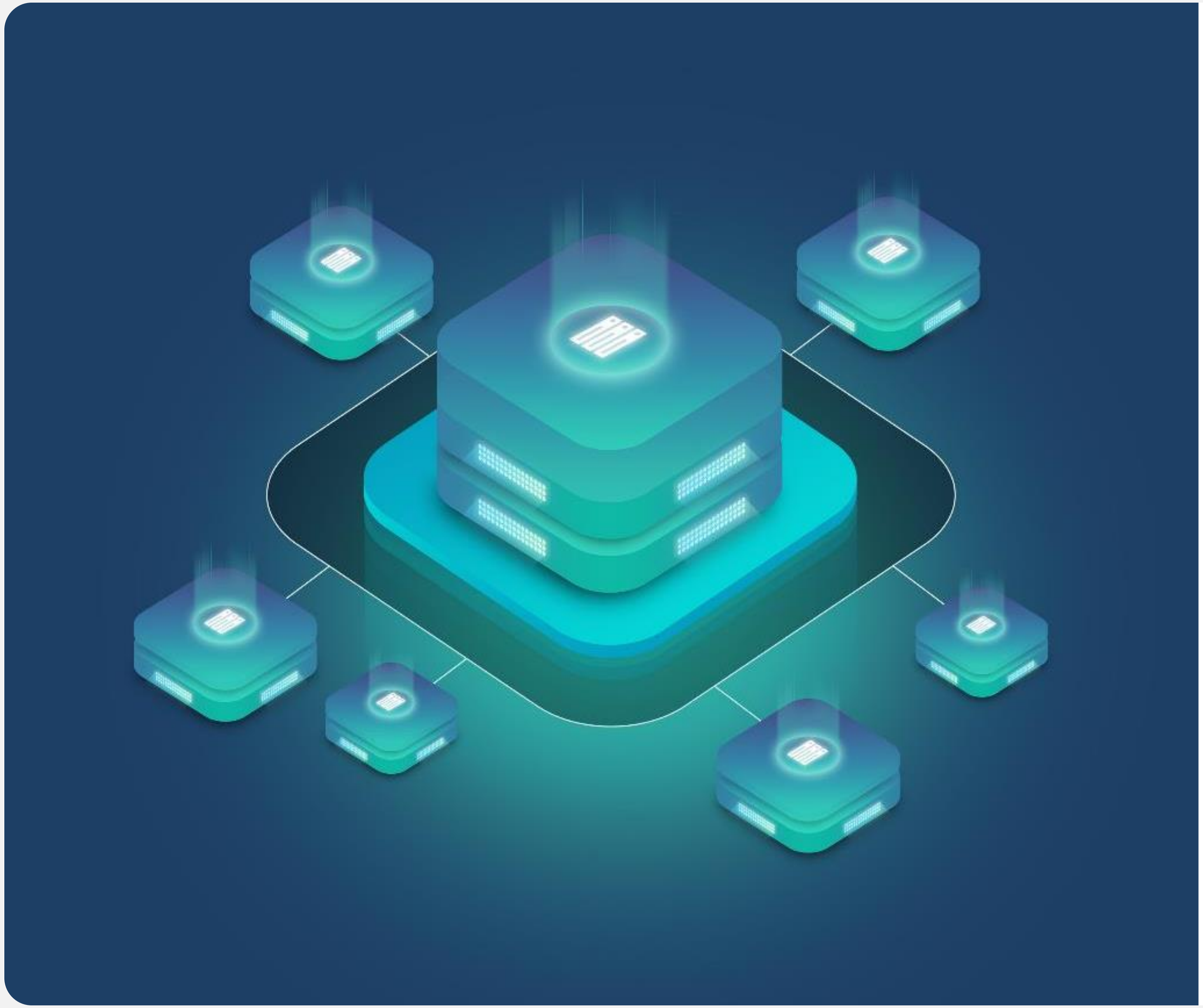
Azure IoT Hub


IoT Devices
Provision and send data from device to cloud
Device Management
Cloud Gateway (IoT Hub)
Stream processing and rules evaluation over data
Store data
UI & Reporting Tools
Stream Processing
Storage
Visualize data and learning
Business Integration
Integrate with business processes
Things
Insights
Actions

Google Cloud IoT Core



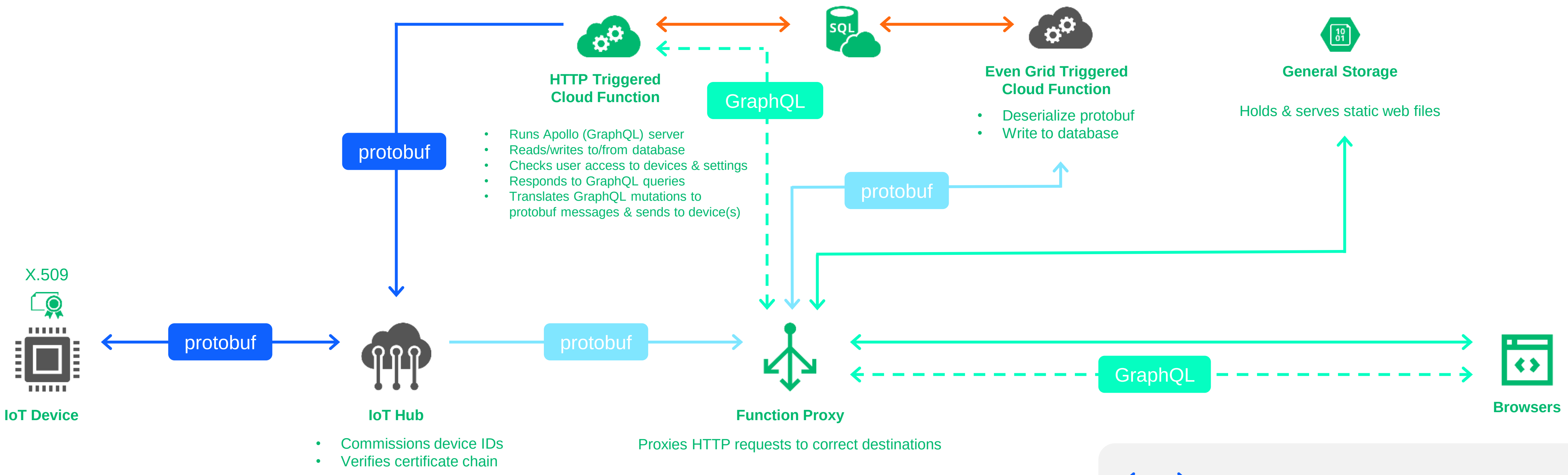
Serverless vs *[containers + microservices, etc.]*



- **Cost and Scale advantages lean toward “serverless” Cloud**
  - Let’s talk cost $$
- **Reasons you may opt-out of serverless?**
  - Need fully private or on-premises operation
  - Portability between cloud platforms (no vendor lock-in)
  - Technology constraints – ex: legacy code or databases
  - …often a hybrid approach can be used
- **DevOps – who’s going to maintain this?**

Full Stack Serverless Architecture: *Device – Cloud – User*

UNDERSTANDING THE DATA FLOW: AZURE EXAMPLE



HTTP Triggered Cloud Function

- Runs Apollo (GraphQL) server
- Reads/writes to/from database
- Checks user access to devices & settings
- Responds to GraphQL queries
- Translates GraphQL mutations to protobuf messages & sends to device(s)

Even Grid Triggered Cloud Function

- Deserialize protobuf
- Write to database

General Storage

Holds & serves static web files

X.509

IoT Device

IoT Hub

- Commissions device IDs
- Verifies certificate chain

Function Proxy

Proxies HTTP requests to correct destinations

Browsers

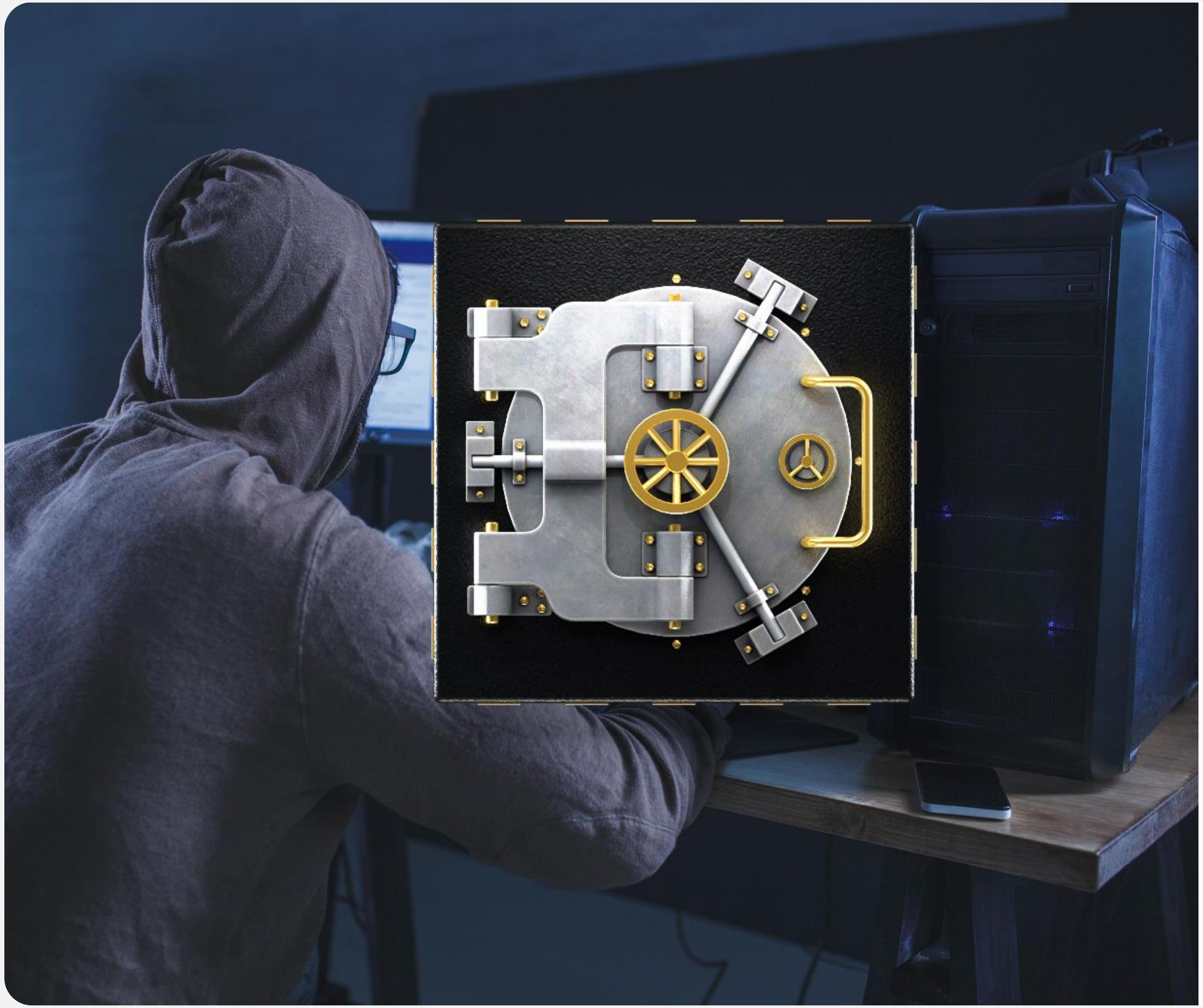
firia cloud

- ⟷ MQTT with TLS encryption
- ⟷ Event Grid Schema (authentication handled by Azure)
- ⟷ HTTPS
- ⇠⇢ HTTPS with OAuth user session
- ⟷ SQL with TLS encryption

Gateway Software Architecture

Design choices for security, wireless connectivity, commissioning, offline scenarios, and remote software updates.

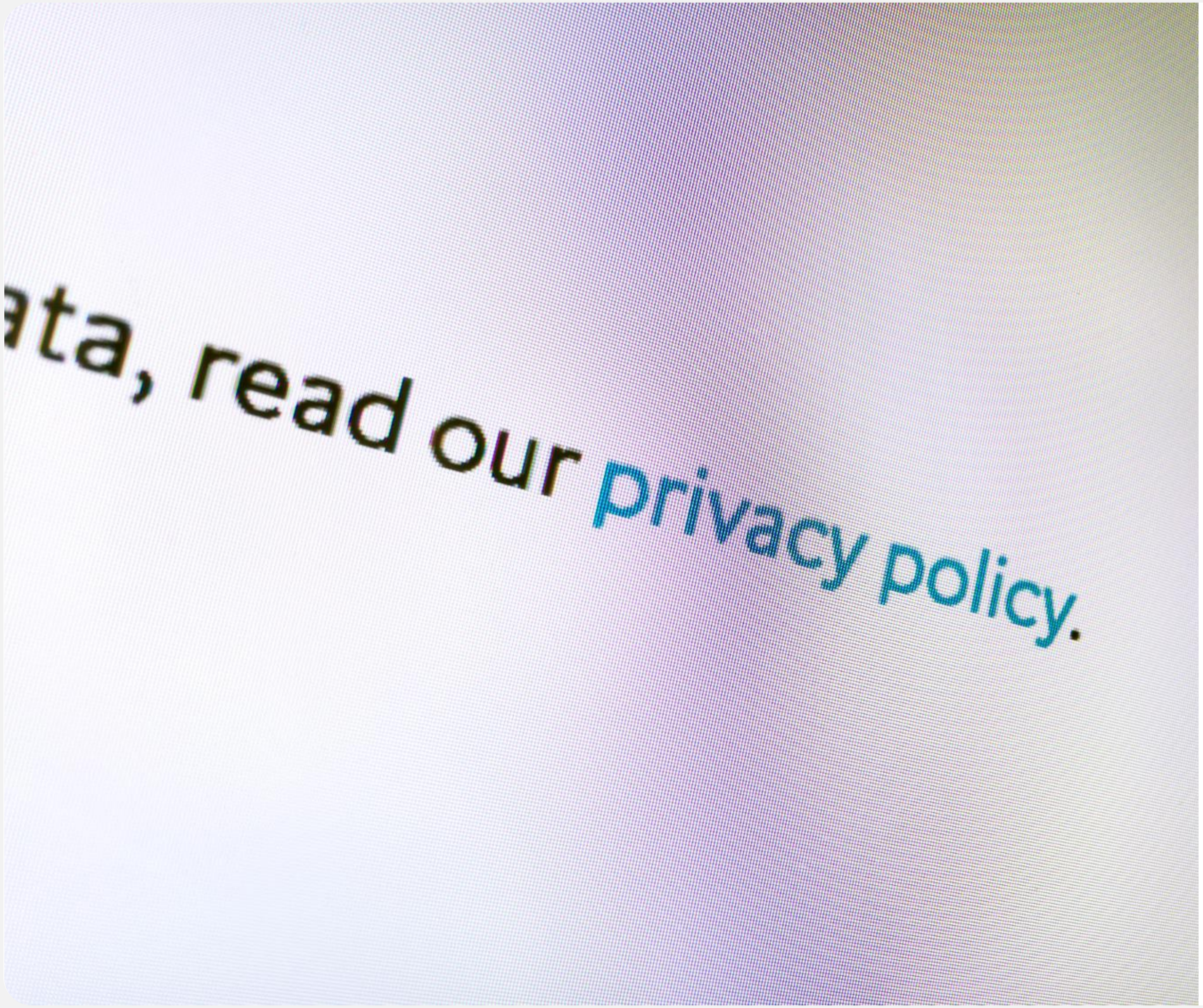
Security



- **Provisioning certificates**
  - Cloud impact during factory programming / test
- **Managing keys**
  - End-user impact of security
- **Local connectivity concerns**
  - The Wi-Fi password changed!?
  - Real-world scenarios

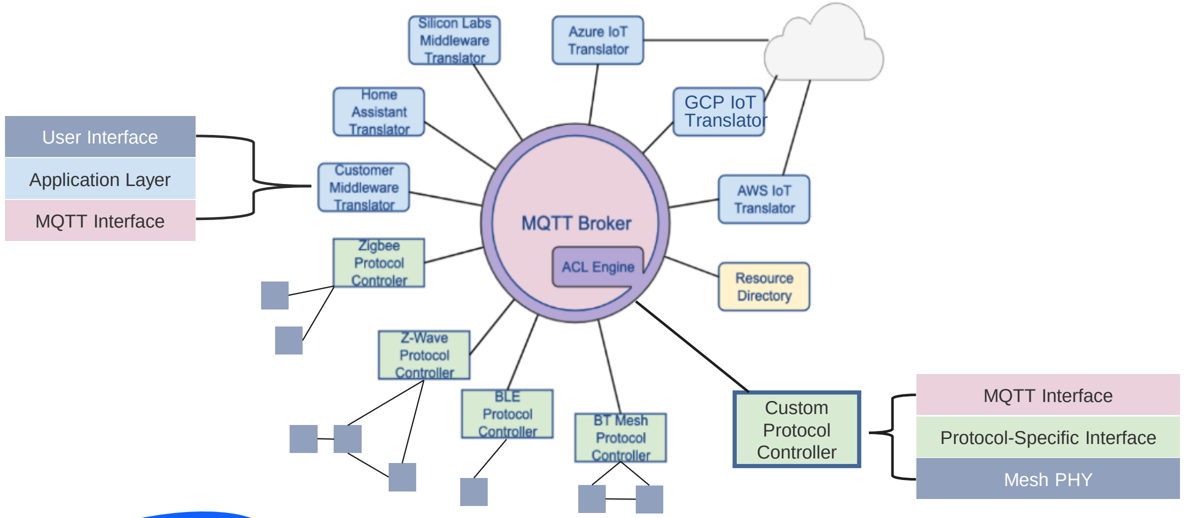
The Fine Print

Cloud Service Agreement

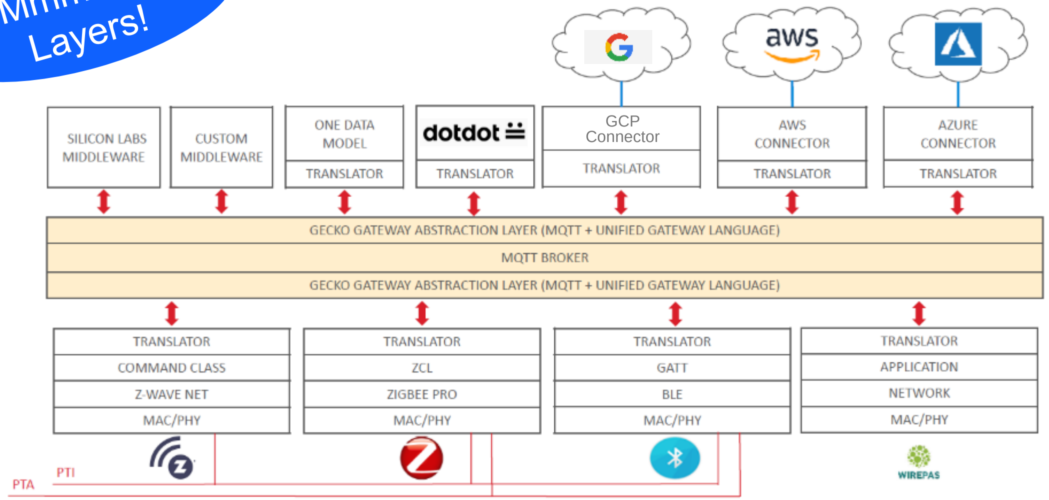


- **Offering a cloud-based service to your customers?**
- **New legislation mandating IoT security**
  - California SB-327 effective Jan 1, 2020 (*other States following*)
  - **NIST** *"Cybersecurity Feature Baseline for Securable IoT Devices"*
  - **ETSI** *EN 303 645*
- **Terms of Service Agreements (TOS)**
  - Users don't get a "license" to the software… this is a *service.*
- **Service Level Agreements (SLA)**
  - Check your cloud provider – service level rolls downhill ;-)
- **Acceptable Use Policy (AUP)**
- **Privacy Policy – consider GDPR**

Wireless Connectivity


User Interface
Application Layer
MQTT Interface
Silicon Labs Middleware Translator
Azure IoT Translator
Home Assistant Translator
GCP IoT Translator
Customer Middleware Translator
MQTT Broker
ACL Engine
AWS IoT Translator
Zigbee Protocol Controler
Resource Directory
Z-Wave Protocol Controller
BLE Protocol Controller
BT Mesh Protocol Controller
Custom Protocol Controller
MQTT Interface
Protocol-Specific Interface
Mesh PHY

Mmmm… Layers!



- **Multiprotocol support**
  - Traditional custom software approach is brittle…
  - Can it work more like TCP/IP?
    - Requires OS support for Network Interfaces
- **Message Bus “Pub/Sub” – MQTT**
  - Decouples concerns… and processes.

Unified IoT Controller

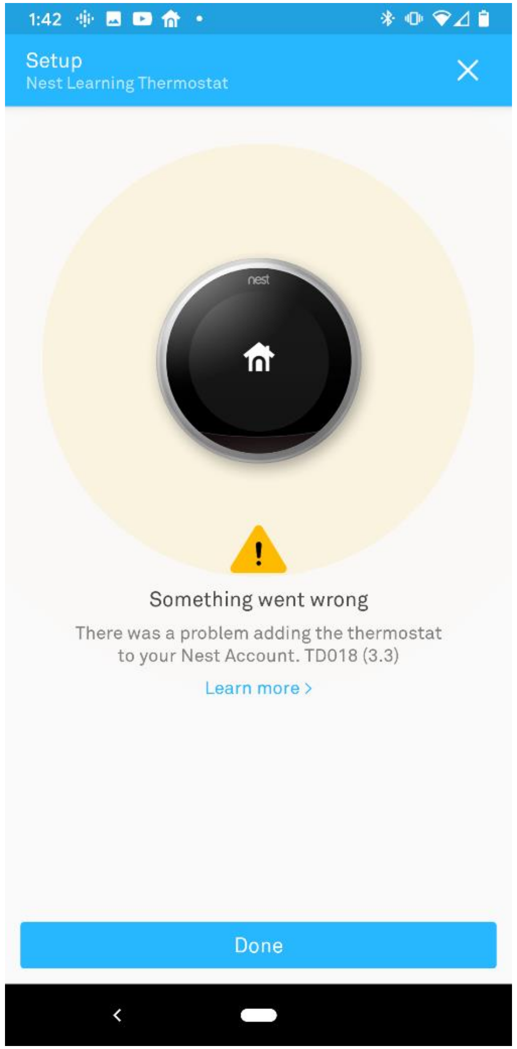
NEW

- A unified IoT software abstraction layer.
- “Device Driver” for IoT.
- Built on MQTT, provides JSON payload definitions for all the layers.

Commissioning

- **Process for onboarding new devices**
- **Case Studies – State of the Art commissioning experiences**
  - Nest Camera
  - Philips Hue
- **Associating unique Devices with User Accounts**
  - Cloud database concerns

Case Study: Commissioning is Hard!



1:42

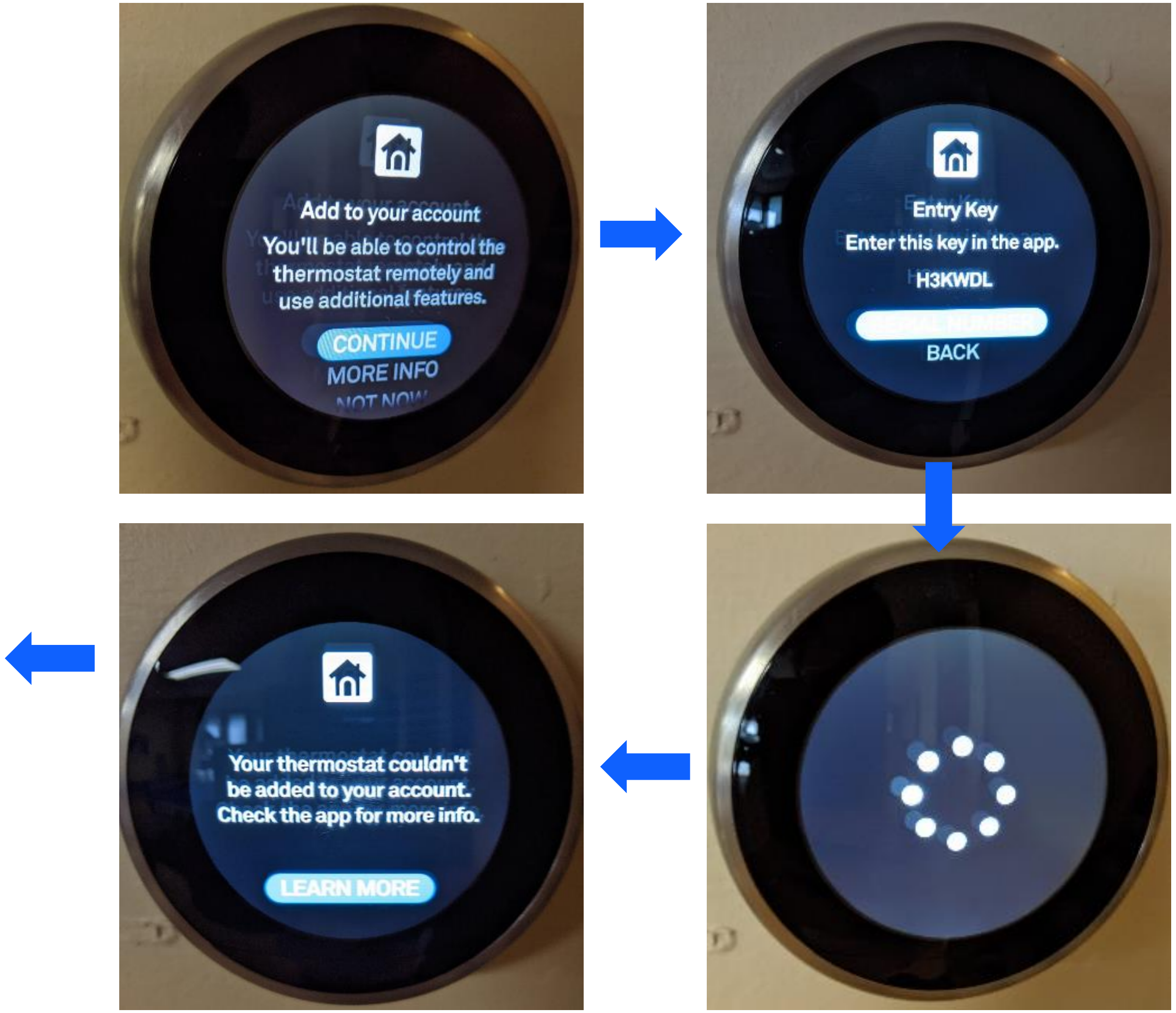
Setup
Nest Learning Thermostat

Something went wrong

There was a problem adding the thermostat to your Nest Account. TD018 (3.3)

Learn more >

Done



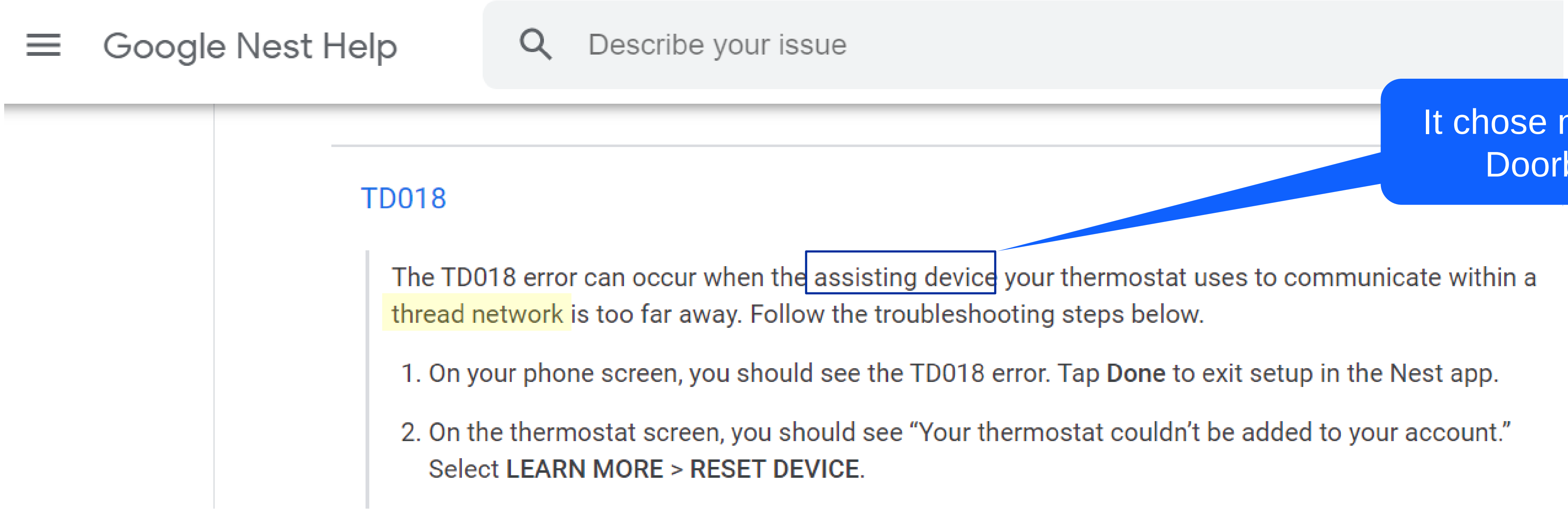
Add to your account
You'll be able to control the thermostat remotely and use additional features.

Entry Key
Enter this key in the app.
H3KWDL

Your thermostat couldn't be added to your account. Check the app for more info.

Oh! It’s using Thread - IEEE 802.15.4

- **Designers could’ve opted for Wi-Fi soft AP mode, or BLE direct…**



Google Nest Help

Describe your issue

TD018

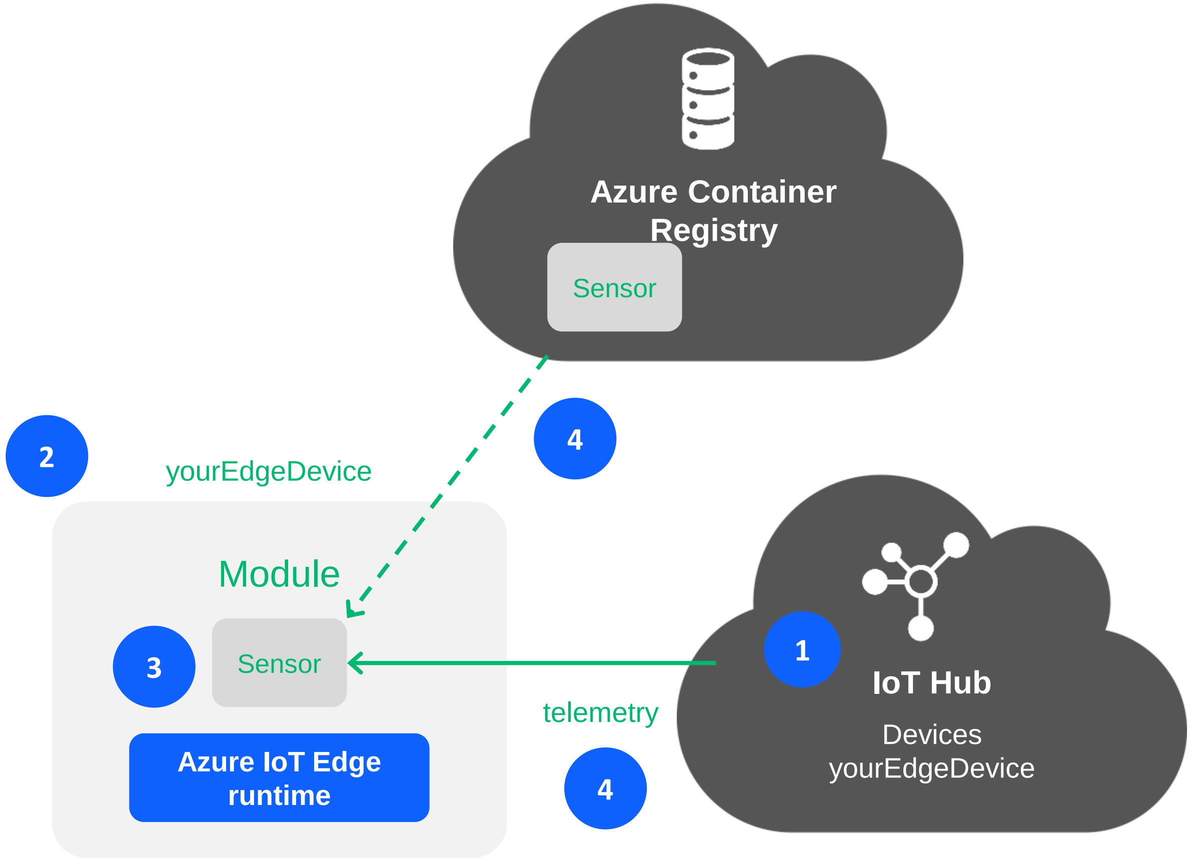
The TD018 error can occur when the assisting device your thermostat uses to communicate within a thread network is too far away. Follow the troubleshooting steps below.

1. On your phone screen, you should see the TD018 error. Tap **Done** to exit setup in the Nest app.
2. On the thermostat screen, you should see “Your thermostat couldn’t be added to your account.” Select **LEARN MORE** > **RESET DEVICE**.

It chose my Front Doorbell !

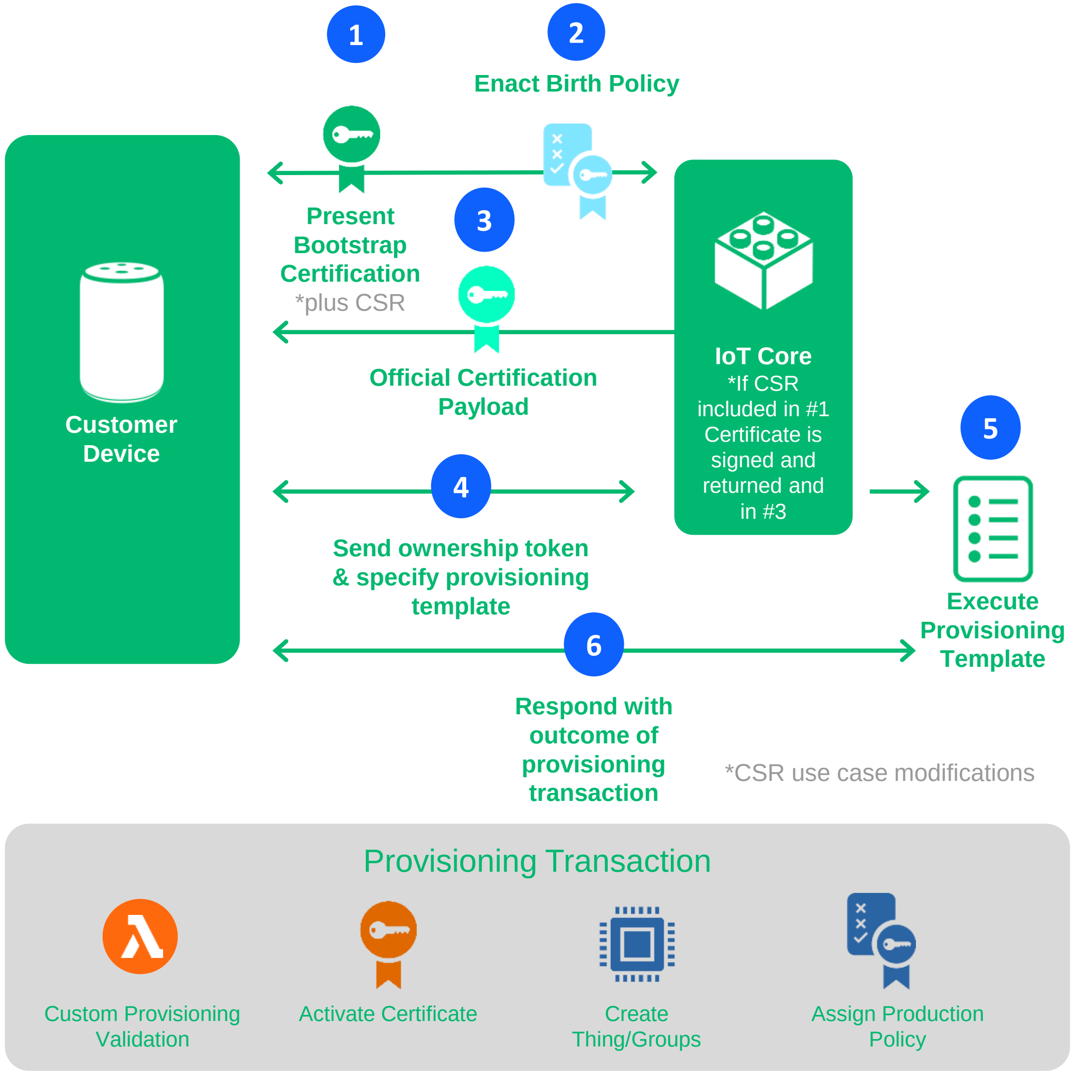
… Goes on through Step 11!

Offline Scenarios and Edge Computing



- **Gateway or Device based caching / spooling**
- **Digital Twins on the Cloud**
- **On-premises Cloud Functions**
  - Containerized Edge Computing
  - AWS GreenGrass
  - Azure IoT Edge

Software Updates


1
2
Enact Birth Policy
Present Bootstrap Certification
*plus CSR
3
Official Certification Payload
Customer Device
IoT Core
*If CSR included in #1 Certificate is signed and returned and in #3
4
Send ownership token & specify provisioning template
5
Execute Provisioning Template
6
Respond with outcome of provisioning transaction
*CSR use case modifications
Provisioning Transaction
Custom Provisioning Validation
Activate Certificate
Create Thing/Groups
Assign Production Policy

- **Security patches make it risky to opt-out**
- **Roll your own update option?**
- **Cloud Provider service offerings**
  - Azure IoT Hub Device Provisioning Service
  - AWS IoT Core Fleet Provisioning
- **3rd Party Services**
  - https://www.upswift.io/

UPSWIFT

Containerization on the Gateway

Watchtower

- **Technology often used on the Cloud**
  - Micro-Services!
  - Deploy applications independently.
  - Future proof.

- **Free your Gateway Applications from Dependency Hell™**🔥
- **…But they may have to break out of Container Jail™**
- **Service Containers – like *Watchtower* can provide remote updates, and more.**
  - Use 3rd party micro-services or roll your own.

works with

BY SILICON LABS

VIRTUAL CONFERENCE