

End-to-end IoT development with Zephyr



Alvaro Viebrantz

Google Developer Expert for IoT and Founding Engineer at golioth.io

aviebrantz.com

 [@alvaroviebrantz](https://twitter.com/alvaroviebrantz)

Why this talk

Why is important to understand the big picture

IoT projects involves multiple areas

- High level layers
 - The Thing - Hardware
 - Connectivity
 - Cloud / Infrastructure / Backend
 - User Facing Apps / Frontend

Application Development

Application Modeling

Usually your device is associated to other real world objects

- Most of the time, application refer to those real world objects, not the the IoT device itself.
- Actions and queries are issued against those objects.
- Is important to think how end users will interact with your (hidden) end device.

Home Automation

A **House** can have multiple **Users**

House have multiple **Rooms**

A **Room** can have multiple **Sensors**

A **Room** can have multiple **Lights**

- Turn on all **Lights** in a given **Room**.
- Turn off all **Lights** on the the House.
- There was any **Movement** in the **House** today ?



Asset Tracking

Device/Tracker is associated to **Vehicle**

Vehicle is part of a **Fleet**

A **Fleet** is owned by an **User**

- Where is a given **Vehicle** ?
- Is this **Vehicle** in movement ?
- Does the user have any **Vehicle** with low battery ?



Indoor Asset Tracking

Device/Tracker is associated to **Medical Equipment**

Scanners/Gateways are associated to a **Room**

Room is associated to a building **Floor**

Floor is part of a **Building**



- Where is a given **Medical Equipment** ?
- There is a **Medical Equipment X** on Building **Y** ?

Remote Sensing

Sensor is tied to a **Silo/Container**

Silo/Container is associated with an **Animal Type**

Farm has **Animal Types**

Farm has multiple **Silos/Containers**



- Does **Animal X** has enough **Food/Water** ?
- What is the level on **Silo/Container X** ?
- What is the trend of **Food consumption** ?

User Management

Authenticate and authorize users

- Authentication (AuthN)
 - Identify users and account management
- Authorization (AuthZ)
 - Permissions and access
- Third Party providers
 - Saves time with common need
 - Security and compliance
- Examples:
 - Auth0, Firebase Auth, AWS Cognito, etc
 - Keycloak, Ory, etc



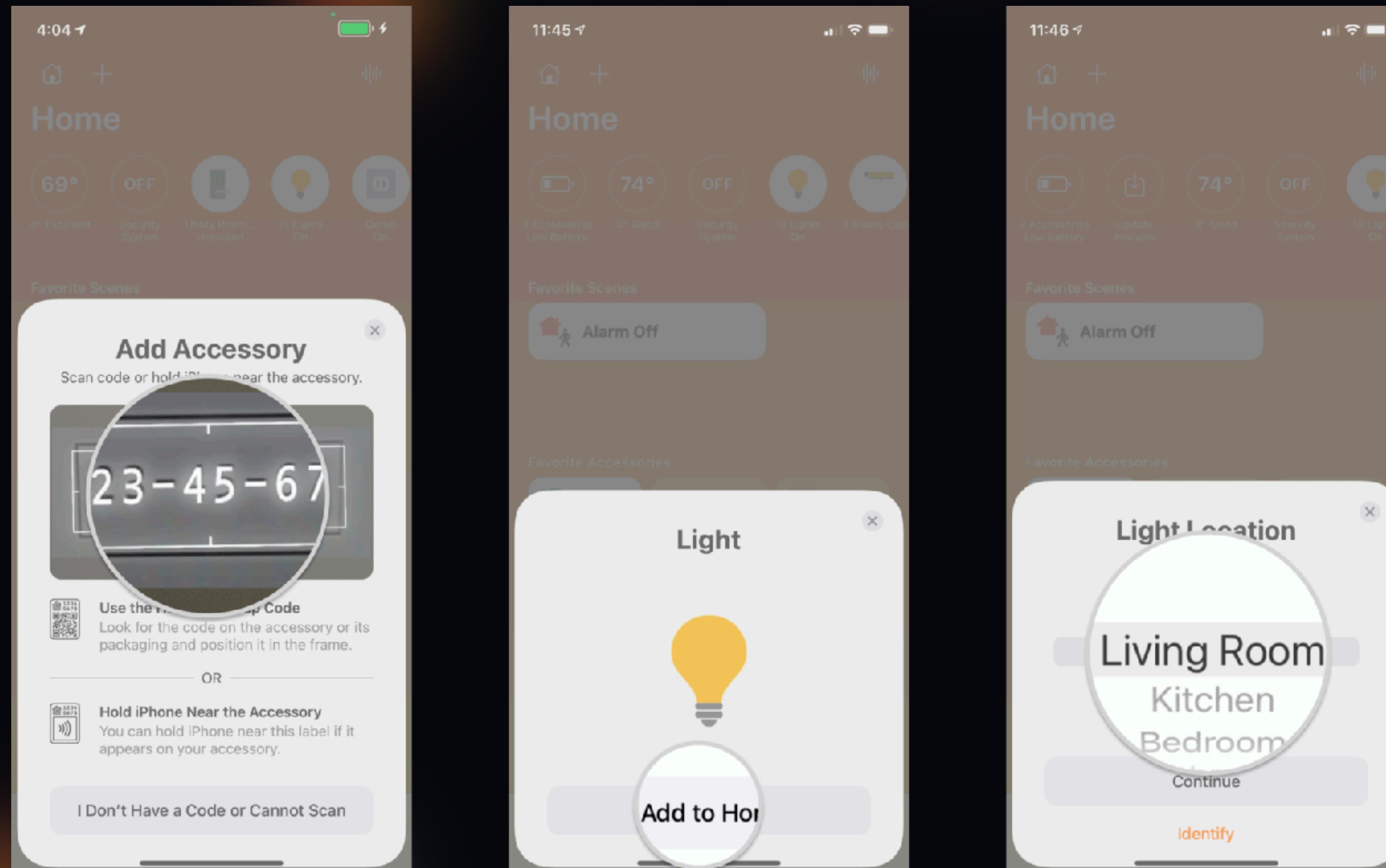
KEYCLOAK



AWS Cognito

Device Association

Link a device to an object and/or user



Device Identification

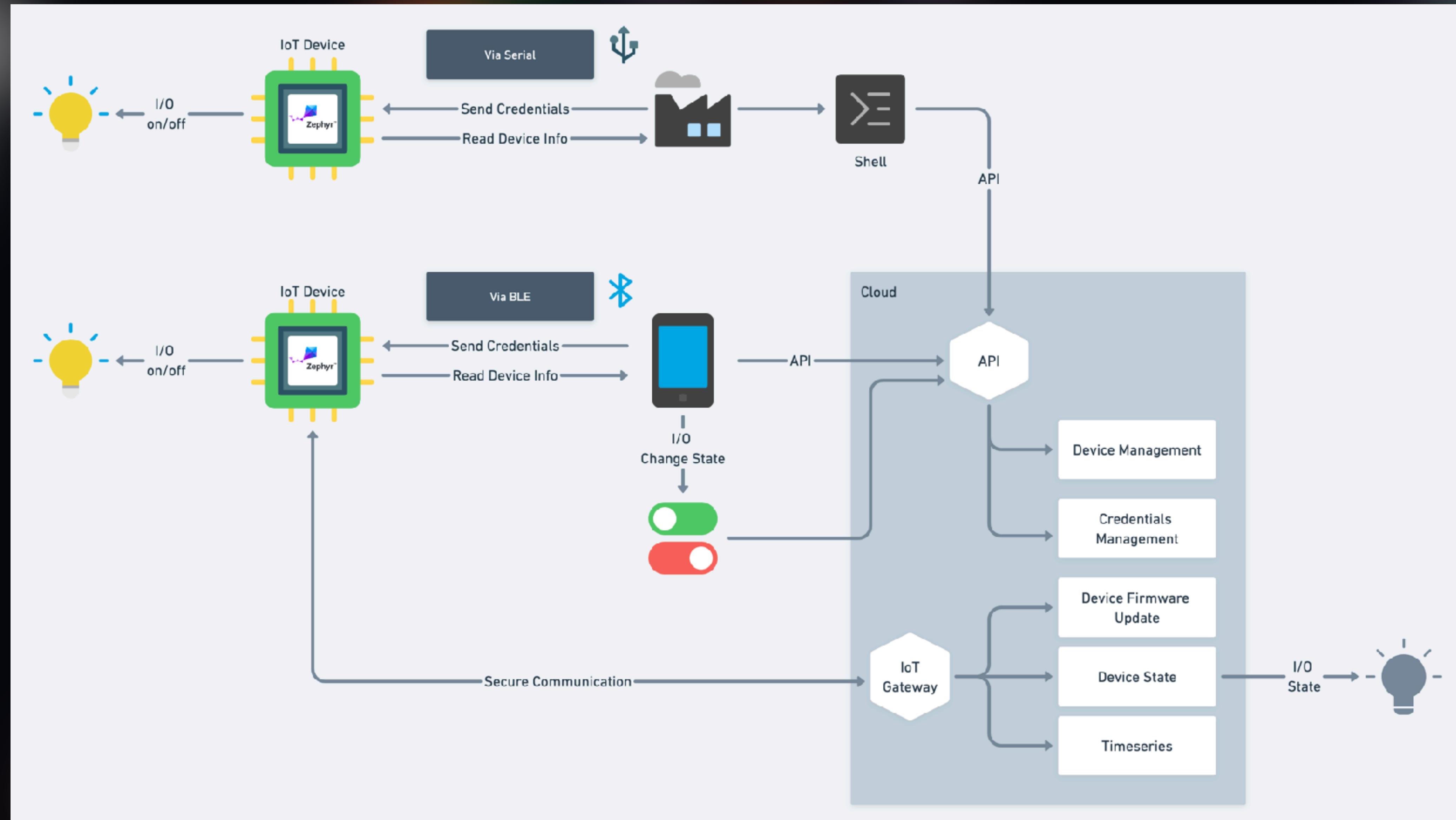
How to map your physical device to the cloud

- Defining a device identification is really important and projects tends to think about it late in the process.
- Examples
 - MAC Address, barcode/QR Codes, microcontroller serial number, etc
- Helps with device provisioning and/or onboarding
- Zephyr provides Hardware Info API

```
uart:~$ hwinfo
hwinfo - HWINFO commands
Subcommands:
    devid      :Show device id
    reset_cause :Reset cause commands
uart:~$ hwinfo devid
Length: 8
ID: 0x0b78d0d7e711ccd5
uart:~$
```

Device Provisioning

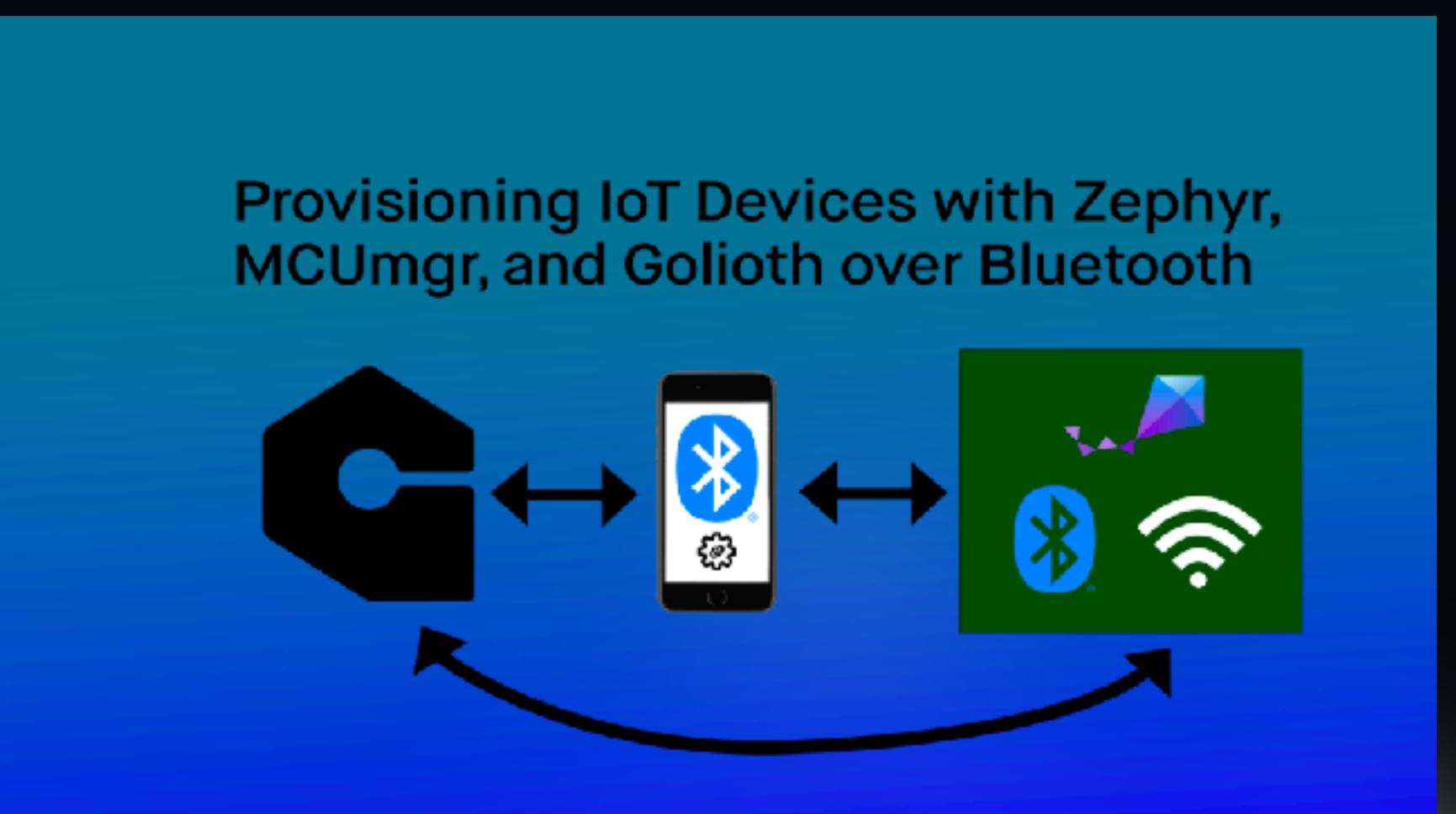
Give credentials and information for the device to connect to the cloud



Content on device provisioning

Link a device to an object and/or user

- Updating your credentials from the Zephyr device shell and more!
 - Zephyr APIs: Settings, Wifi Management, Settings, Shell
 - <https://blog.golioth.io/new-feature-updating-zephyr-settings-from-the-device-shell-and-more/>
- Provisioning an nRF52 + ESP8266 device using a Bluetooth application and Golioth
 - Zephyr APIs: MCU Manager, Wifi Management, Settings, Bluetooth
 - <https://blog.golioth.io/provisioning-iot-devices-with-zephyr-mcumgr-and-golioth/>



Connect to
the Cloud

“I just need to send data to my MQTT broker in the Cloud”

Naive IoT Developer

Why that is not the full history

Having the broker is just the transport/middleware

- Device credential management needs to be created on top of the broker
- How to show data from devices ?
 - Connect the UI directly to the broker ? Not super secure 
 - Ideally you should create an API layer to access the data on the database/broker
- How to get data out of the broker ?
 - Set up some scripts subscribing to a wildcard topic and send things to the database
 - What if that script stop working ?
- Which database to use ? How to scale it to a increasing amount of devices and data coming ?

Common needs for IoT projects

Device Management

Keep track of your connected devices

- Control which devices are allowed to connect to the cloud
- Device Credential management
- Monitoring
 - Device Health
 - Online/Offline
 - Last connection
 - Logs
 - Firmware Updates
 - Deployment Tracks

The screenshot shows the Golioth Device Management interface. On the left, there's a sidebar with navigation links: Dashboard, Projects, Management, Monitor, Firmware Update, Artifacts, Releases (which is highlighted in green), API Keys, and Output Streams. The main area is titled 'Releases' and shows a table of deployment records. The table has columns for Seq. Number, Release Flags, Artifacts, Blueprint, Device Tags, Created, Suit Manifest, and Rollout. There are four rows of data:

Seq. Number	Release Flags	Artifacts	Blueprint	Device Tags	Created	Suit Manifest	Rollout
16341512205...	1.0.1	main - 1.0.1	-	desk	4 months ago	<button>...</button>	<button>ON</button>
16341409133...	1.0.3	main - 1.0.3	-	desk	4 months ago	<button>...</button>	<button>ON</button>
16341390076...	1.0.2	main - 1.0.2	-	desk	4 months ago	<button>...</button>	<button>ON</button>
16339798392...	-	main - 1.0.1	-	desk	4 months ago	<button>...</button>	<button>ON</button>

At the top of the main area, there's a 'Tag Filter' section with several tags listed: development_device, field_troubleshooting, field_data_gather, 1st_deployment_customer_site_delta, 3rd_deployment_customer_site_delta, 2nd_deployment_customer_site_delta, hardware_lab_hq, and desk. There are also buttons for 'Clear Filters', 'Export .csv', and 'Create'.

Device State and Configuration

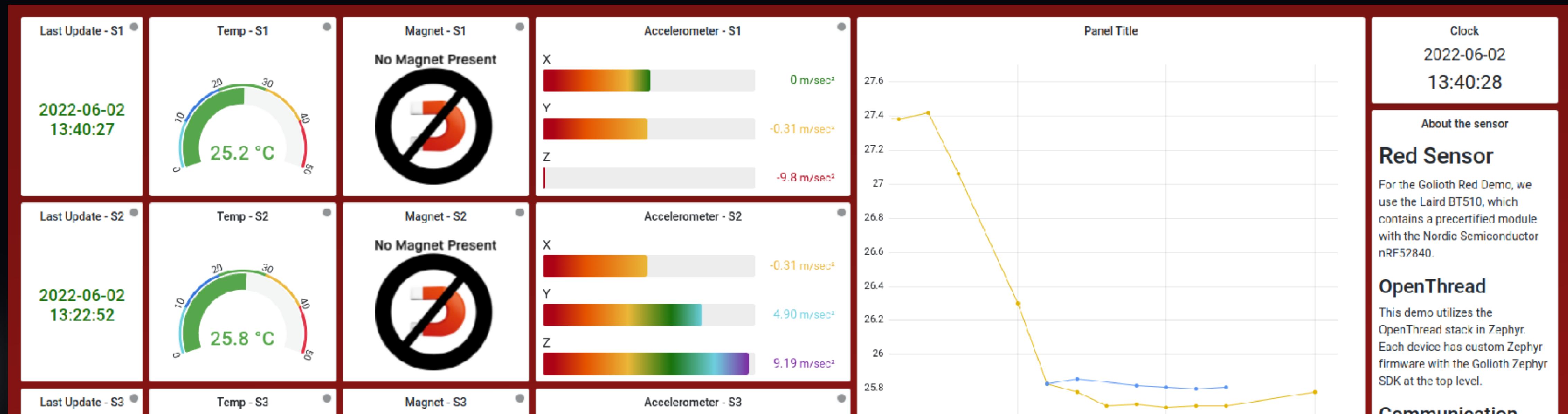
Cloud representation of device state

- IoT Device are not always online
- Latest information about the device
- Change the state in the Cloud side and expect the device to reach that desired state
- Local changes can be queue up using Zephyr concurrency primitives
 - Mutex and Semaphores
 - Message Queues

Device State and Configuration

Cloud representation of device state

- Example of device state/config
 - Light is on/off
 - Alert threshold (Send alert when temperature is above X)
 - Which firmware version the device running version
 - Last sensor report



Device State and Configuration

Cloud implementation examples

LightDB Data - JaredwomWalkthrough

Latest Refresh: 12/16/2021 4:17:00 PM

Refresh 

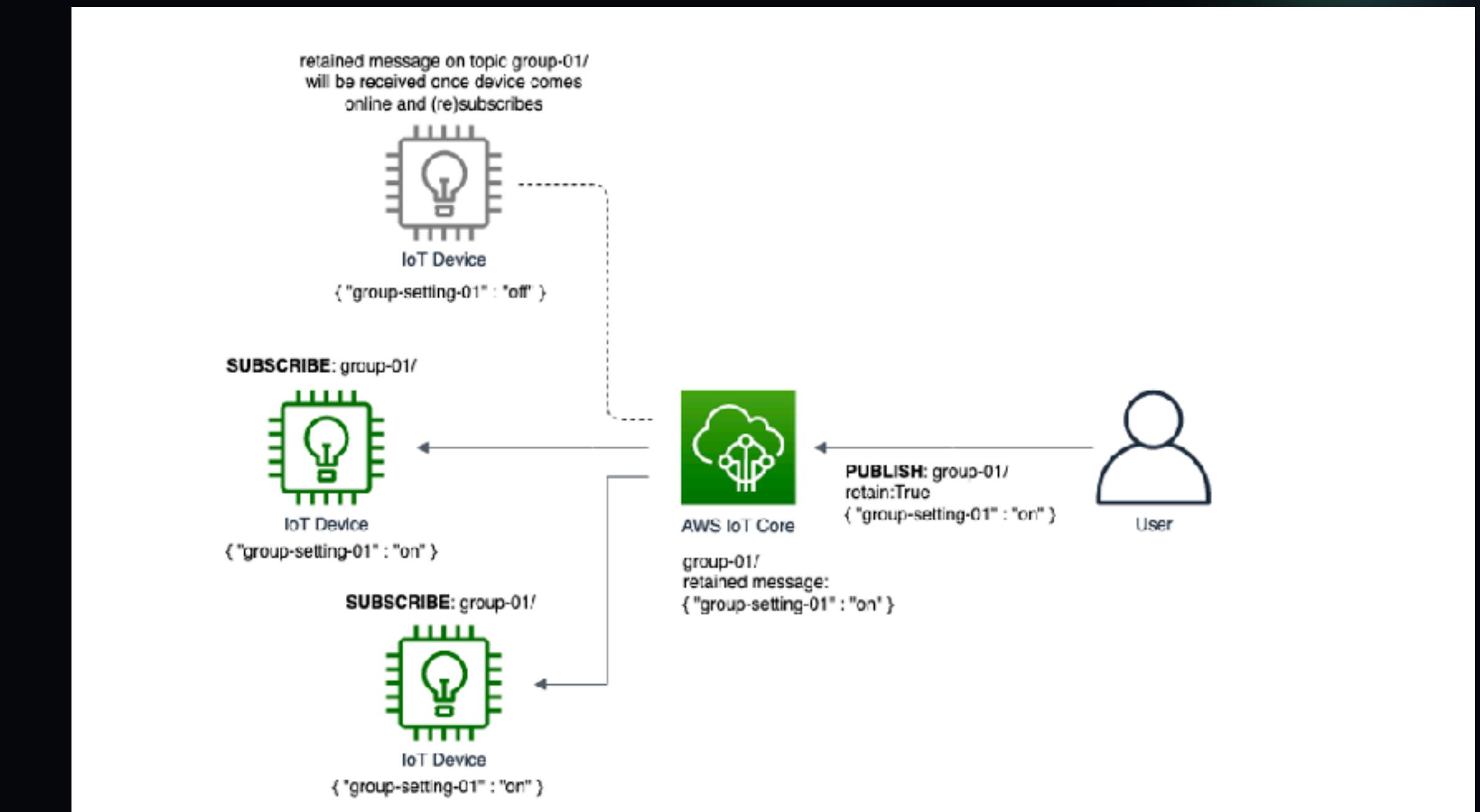
Auto Refresh

1s
3s
5s
10s

{ 2 items  

- "boot": { 1 item
 - "ts": 1639174060817
- "gps": { 3 items
 - "lat": 35.1
 - "lng": -78.
 - "ts": 1639175740000

Golioth LightDB State



AWS IoT Shadow

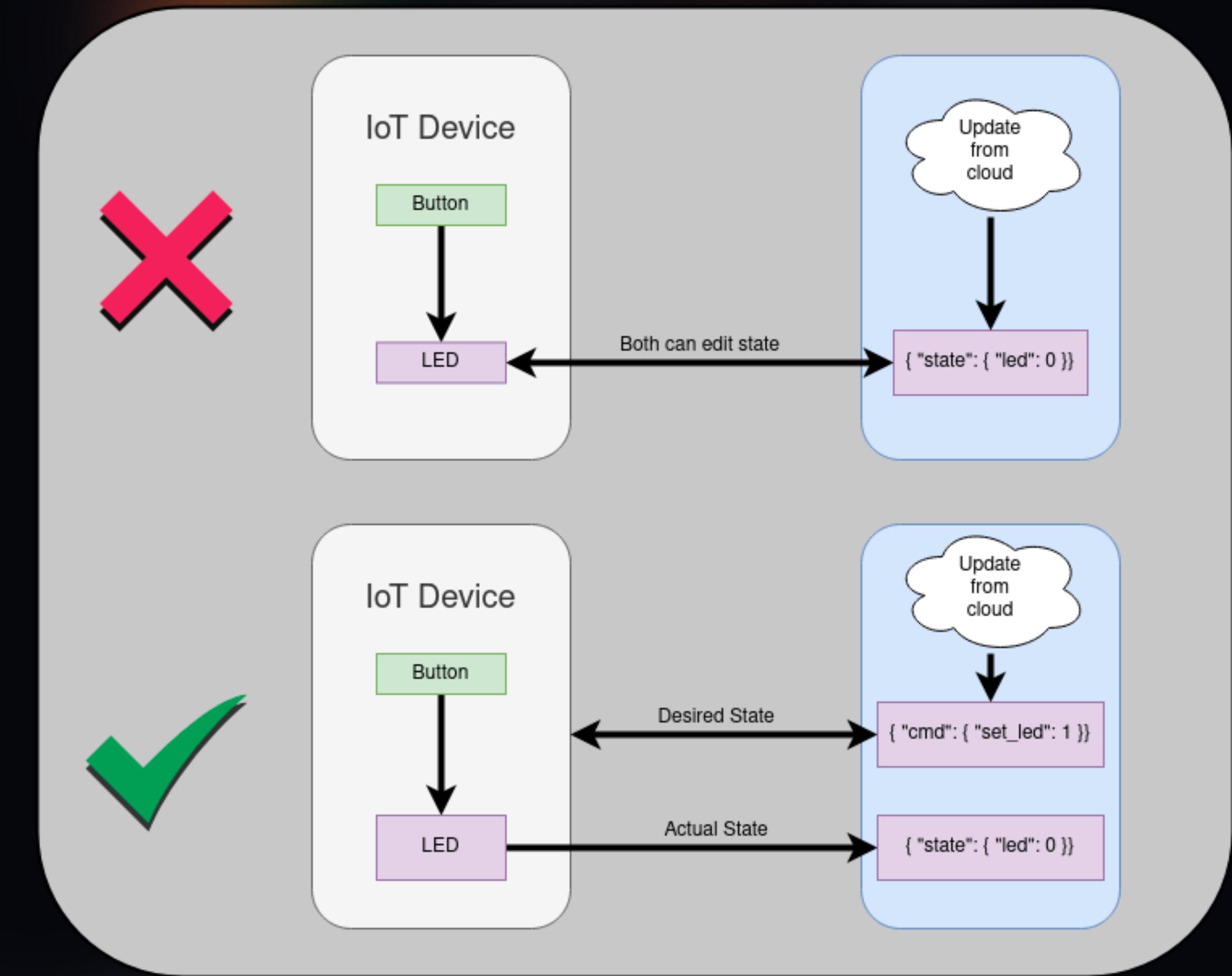
Bidirectional Communication and Commands

Send data down from Cloud → Device

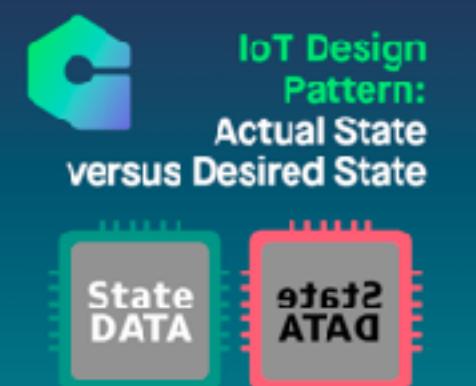
- Usually you need a way to control the device remotely
- Examples
 - Change its configuration after it's out in the field
 - Notify of a new firmware version
 - Trigger action on device in realtime - Turn on/off LED

Desired State vs Actual State

Interactions with an IoT device are usually async



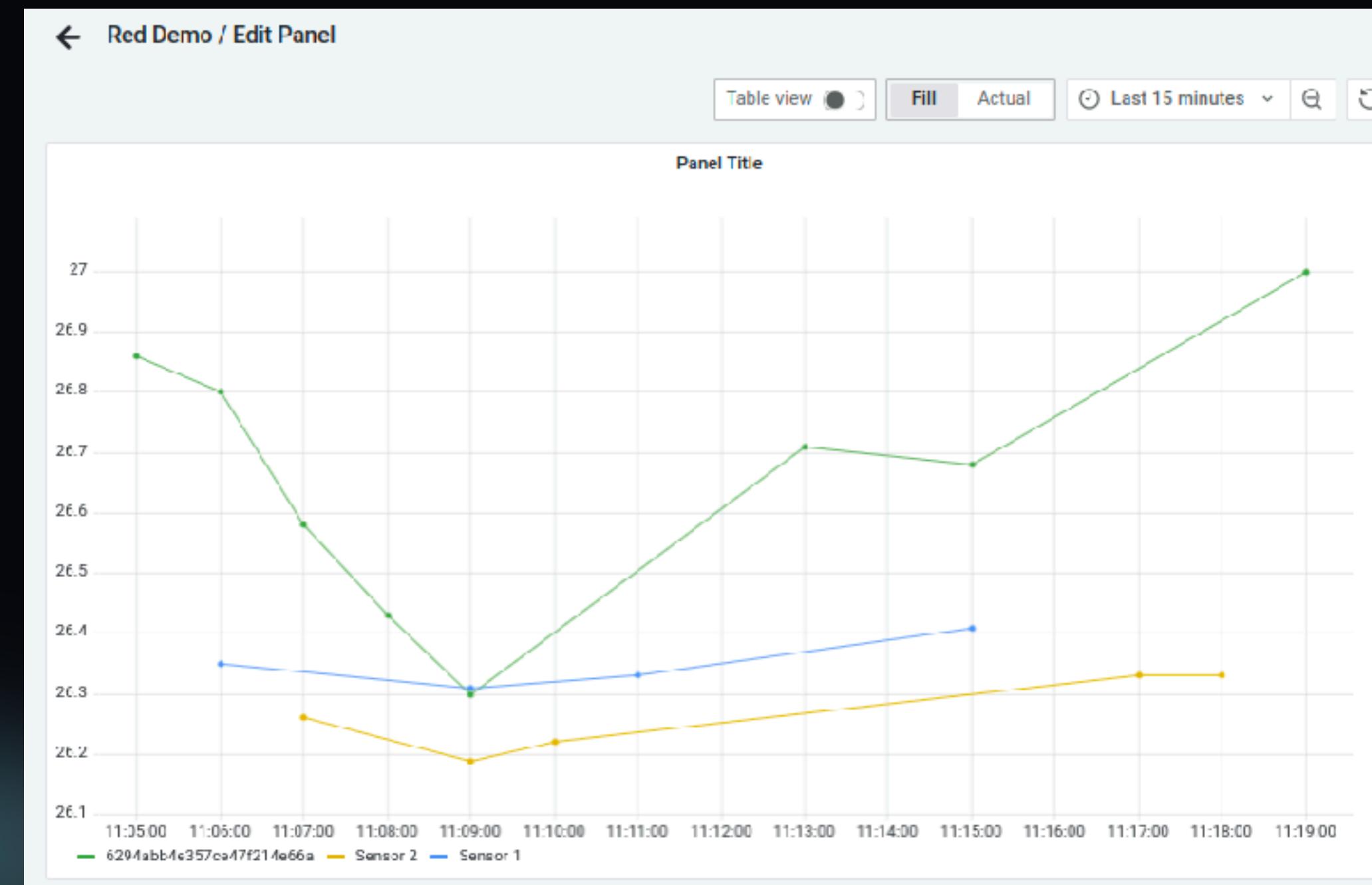
- <https://blog.golioth.io/better-iot-design-patterns-desired-state-vs-actual-state/>



Time-series Data and Events

Massive amount of data send by your fleet of devices

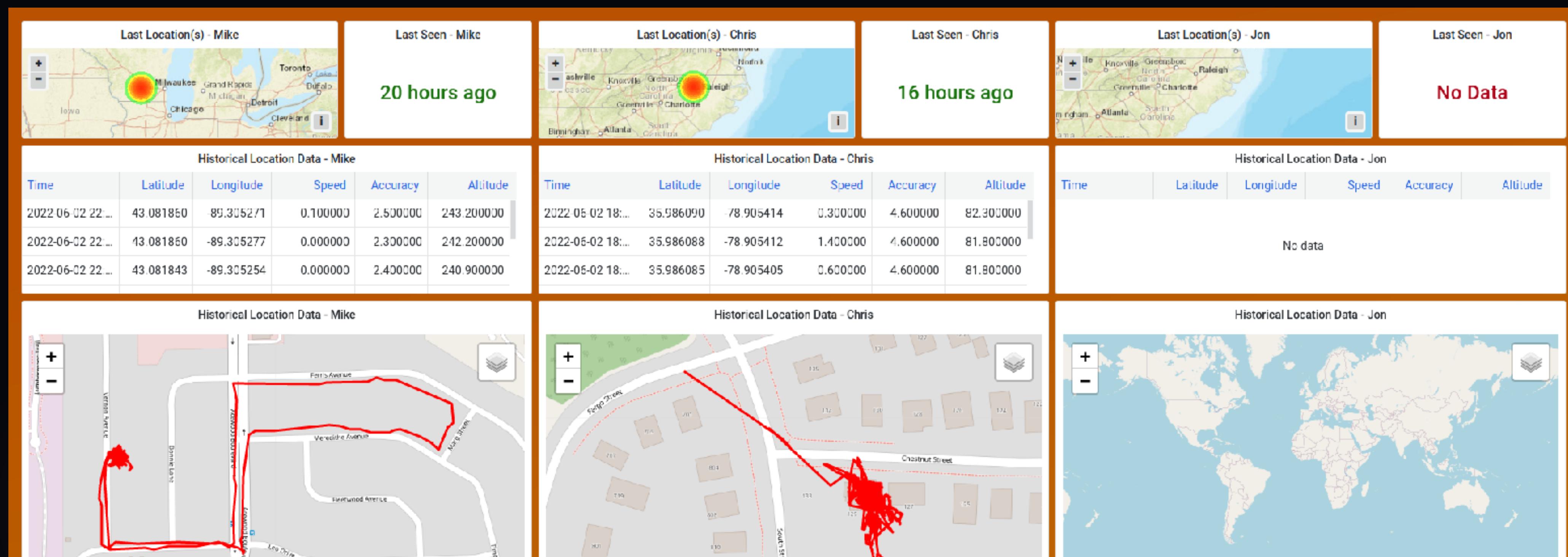
- Most IoT applications generate time series data
- Query data in a given period, window and aggregate data
- Time of capture vs Time of send
 - Device can collect data offline, add timestamps and send later in batch



Time-series Data and Events

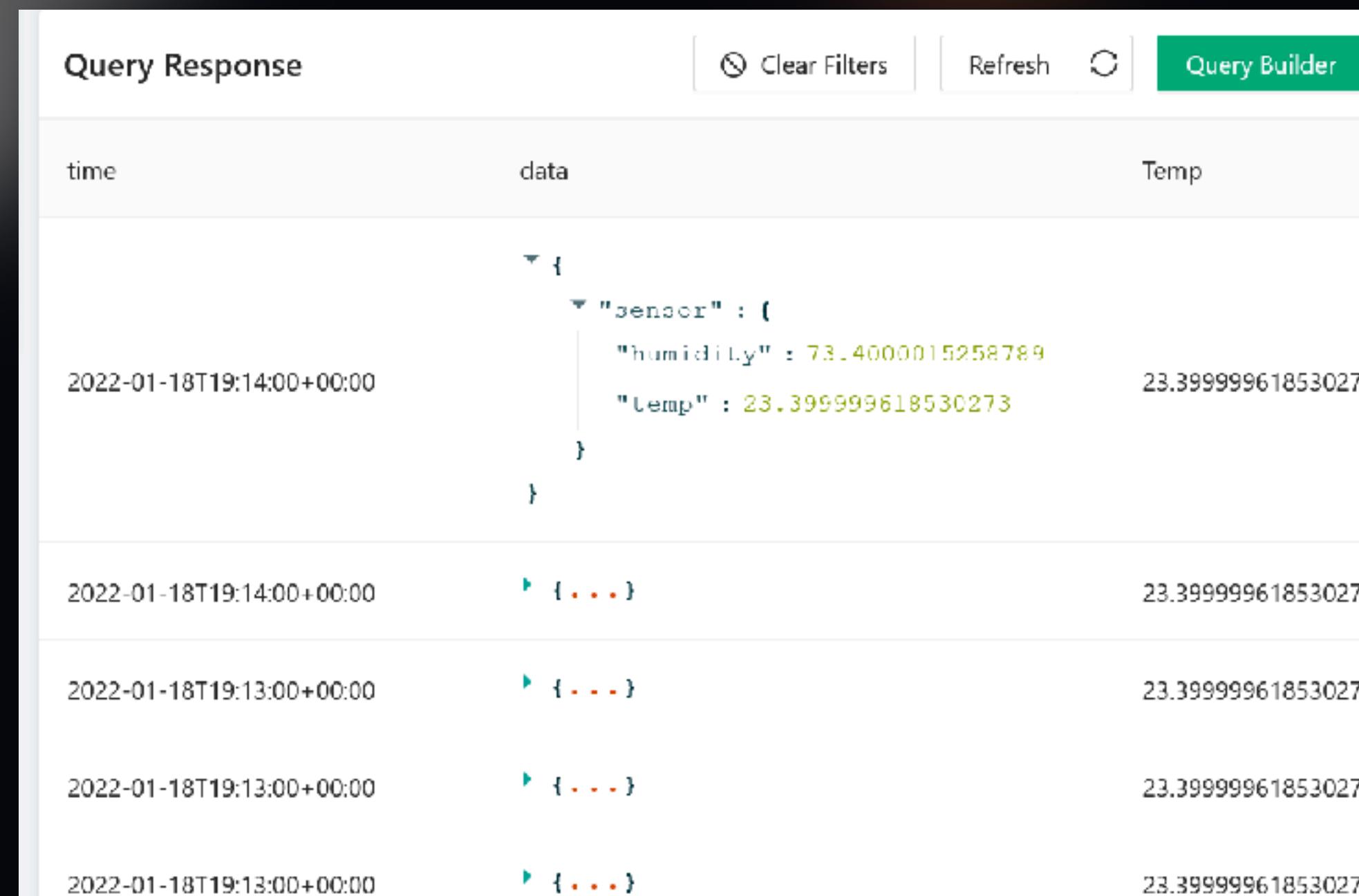
Examples for time-series data

- Any data that you want to chart
- Average temperature of a given fridge in the last 7 days in 15 min window
- Track history of a vehicle in the last 4 hours
- When the humidity was below X in the past 7 days ?

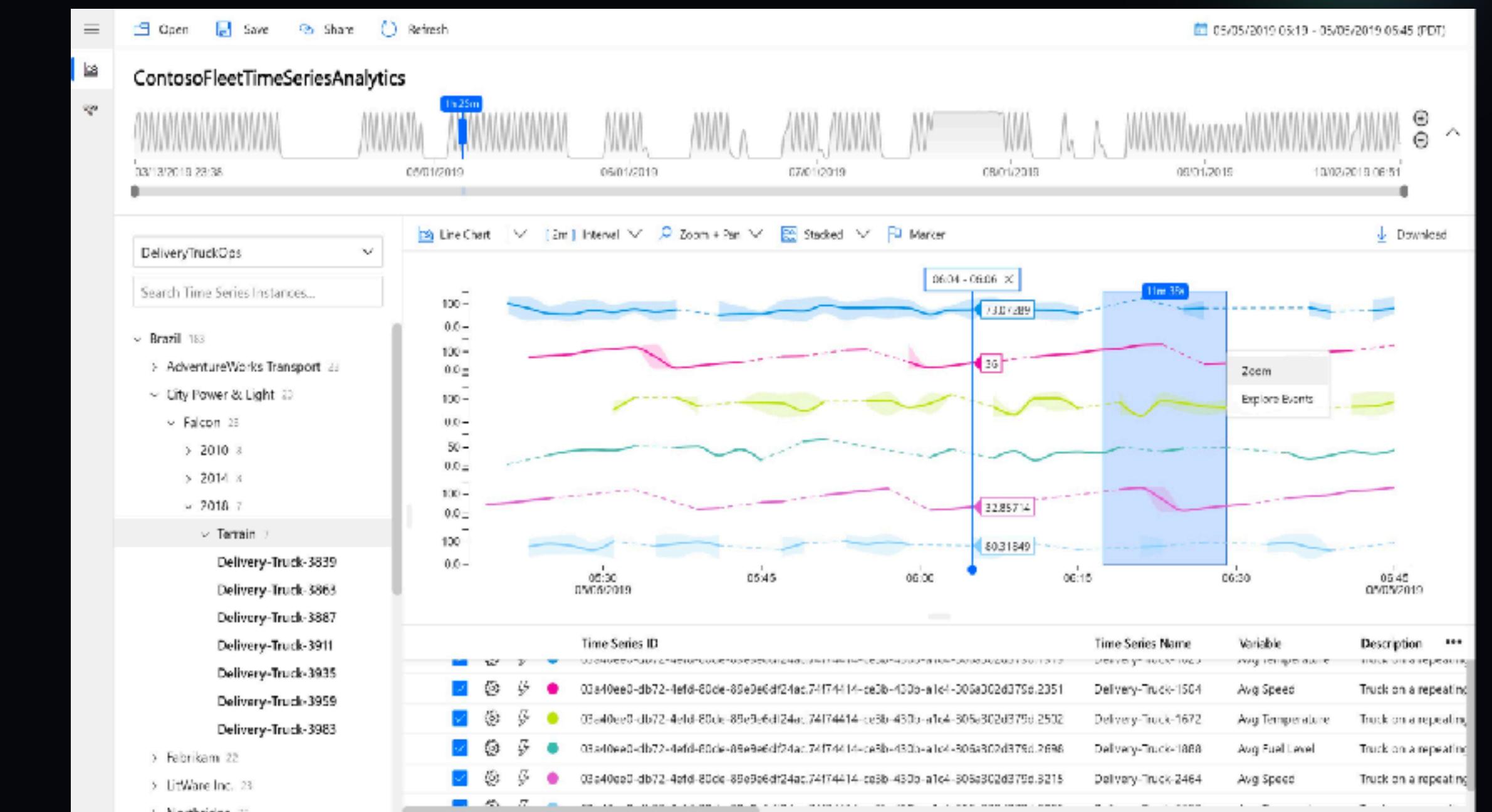


Time-series Data and Events

Cloud implementation examples



Golioth LightDB Stream



Azure IoT Time Series Insights

Connectivity / Security

How to connect my device ?

Protocols

Different ways for the device to talk to the cloud

- IP Protocols
 - TCP
 - HTTP, Websocket and MQTT
 - UDP
 - CoAP and LWM2M
 - Other protocols like DNS, mDNS, SNTP
 - Secure communication
 - Good thing that with Zephyr, security protocols works regardless of the data link layer (L2)
 - So you can use TLS (for TCP) and DTLS (for UDP), both with certificates and with Pre shared keys

Data Encoding

How to format data to be send to the cloud.

- Supported by Zephyr
 - JSON and CBOR
 - More flexible and schema-less
 - CBOR is a binary protocol
 - JSON is human readable
 - Protocol Buffer
 - Strict schema
 - Binary protocol
 - Custom protocol (DIY)

```
QCBORDecode_EnterMap(&decode_ctx, NULL);
qerr = QCBORDecode_GetError(&decode_ctx);
if (qerr != QCBOR_SUCCESS) {
    LOG_WRN("Did not enter CBOR map correctly");
    return -EBADMSG;
}

/* Iterate through all entries in map */
while (true) {
    QCBORDecode_VGetNext(&decode_ctx, &decoded_item);

    qerr = QCBORDecode_GetError(&decode_ctx);
    if (qerr == QCBOR_ERR_NO_MORE_ITEMS) {
        break;
    }

    if (qerr != QCBOR_SUCCESS) {
        LOG_DBG("QCBORDecode_GetError: %d", qerr);
        break;
    }

    if (decoded_item.uLabelType != QCBOR_TYPE_TEXT_STRING) {
        LOG_WRN("Label type should be text string");
        continue;
    }

    if (decoded_item.uDataType != QCBOR_TYPE_FALSE &&
        decoded_item.uDataType != QCBOR_TYPE_TRUE) {
        LOG_WRN("Data type should be boolean");
        continue;
    }

    if (decoded_item.label.string.len > sizeof(name) - 1) {
        LOG_HEXDUMP_WRN(decoded_item.label.string.ptr,
                        decoded_item.label.string.len,
                        "Too long label");
        continue;
    }

    /* Copy label to NULL-terminated string */
```

Cloud interoperability

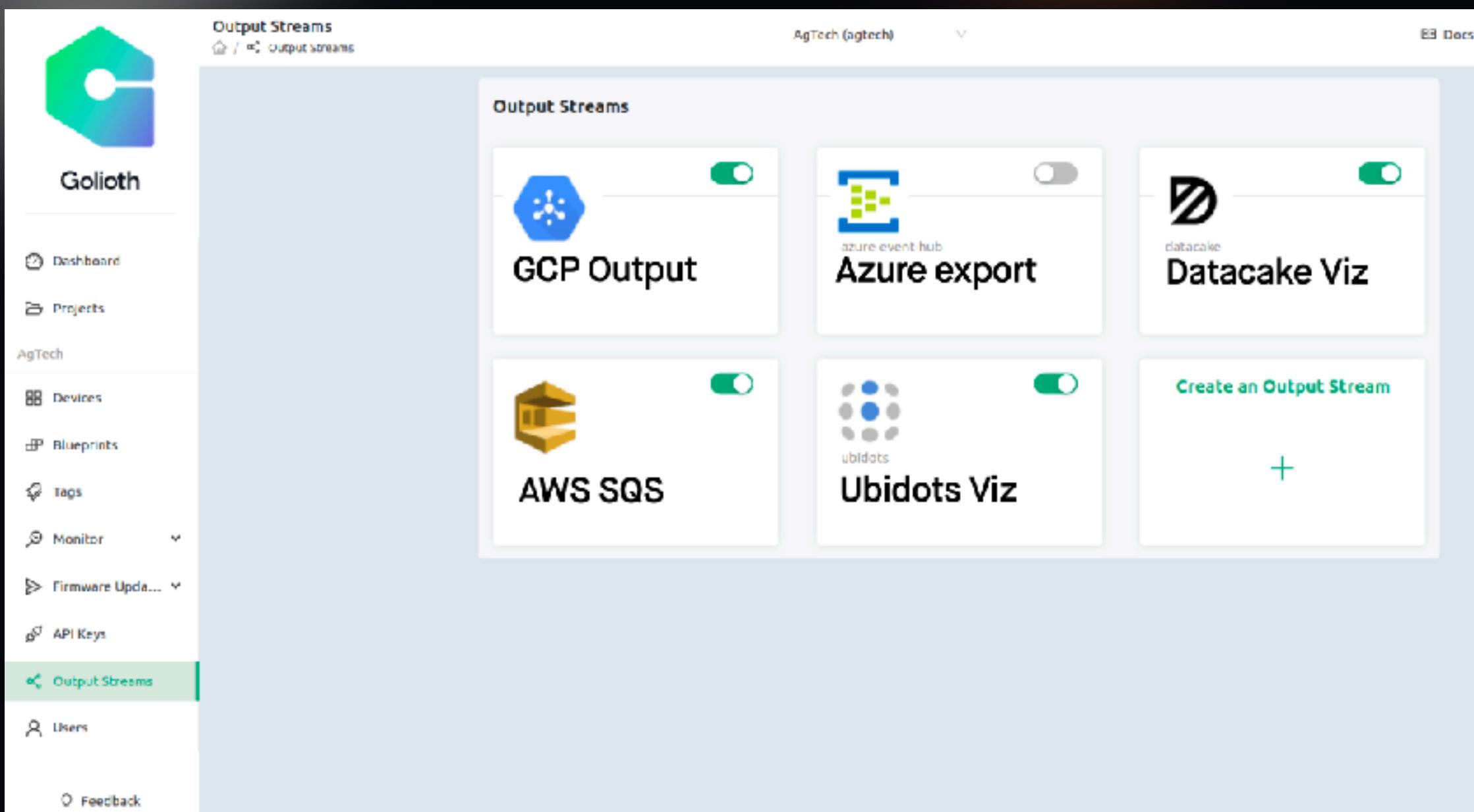
Cloud interoperability

Move burden of processing and routing to the cloud

- Cloud ingests data from constrained devices and have more power to retransmit that data to other systems
- Creating cloud to cloud integrations is easier and more flexible than putting all the logic on the device
- Most public cloud APIs talks JSON over HTTP
 - Your device can send binary data, but the cloud has to convert to a more common format
- Examples
 - Send alerts via Emails/SMS
 - Save data to different databases, each optimized for each use case
 - All data modeling that we discussed before

Cloud interoperability

Examples

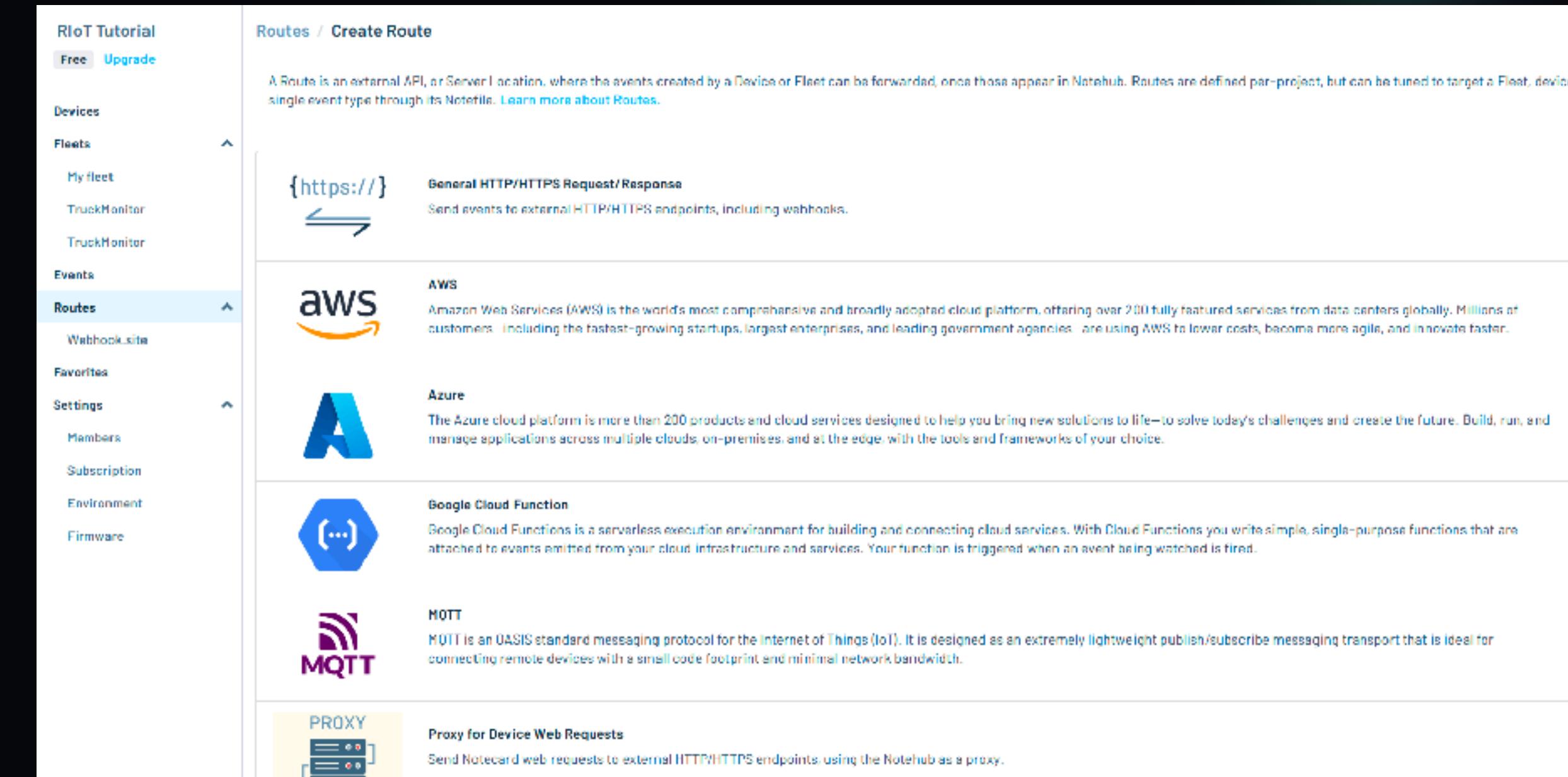


The screenshot shows the Golioth platform's "Output Streams" section. On the left is a sidebar with navigation links: Dashboard, Projects, AgTech, Devices, Blueprints, Tags, Monitor, Firmware Update, API Keys, Output Streams (which is selected and highlighted in green), Users, and Feedback. The main area displays a grid of six output stream configurations:

- GCP Output
- Azure export
- Datacake Viz
- AWS SQS
- Ubidots Viz
- Create an Output Stream

Each configuration includes a small icon, a title, and a toggle switch.

Golioth Output Streams



The screenshot shows the Blues Wireless Notehub "Routes / Create Route" interface. On the left is a sidebar with navigation links: RioT Tutorial, Free, Upgrade, Devices, Fleets, My fleet, TruckMonitor, Events, Routes (which is selected and highlighted in blue), Webhook.site, Favorites, Settings, Members, Subscription, Environment, and Firmware. The main area displays a list of route types:

- {https://} General HTTP/HTTPS Request/Response: Send events to external HTTP/HTTPS endpoints, including webhooks.
- aws Amazon Web Services (AWS): Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.
- Azure The Azure cloud platform is more than 200 products and cloud services designed to help you bring new solutions to life—to solve today's challenges and create the future. Build, run, and manage applications across multiple clouds, on-premises, and at the edge, with the tools and frameworks of your choice.
- Google Cloud Function Google Cloud Functions is a serverless execution environment for building and connecting cloud services. With Cloud Functions you write simple, single-purpose functions that are attached to events emitted from your cloud infrastructure and services. Your function is triggered when an event being watched is fired.
- MQTT MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.
- PROXY Proxy for Device Web Requests: Send Notecard web requests to external HTTP/HTTPS endpoints, using the Notehub as a proxy.

Blues Wireless Notehub Routes

The Thing

Hardware part

Hardware

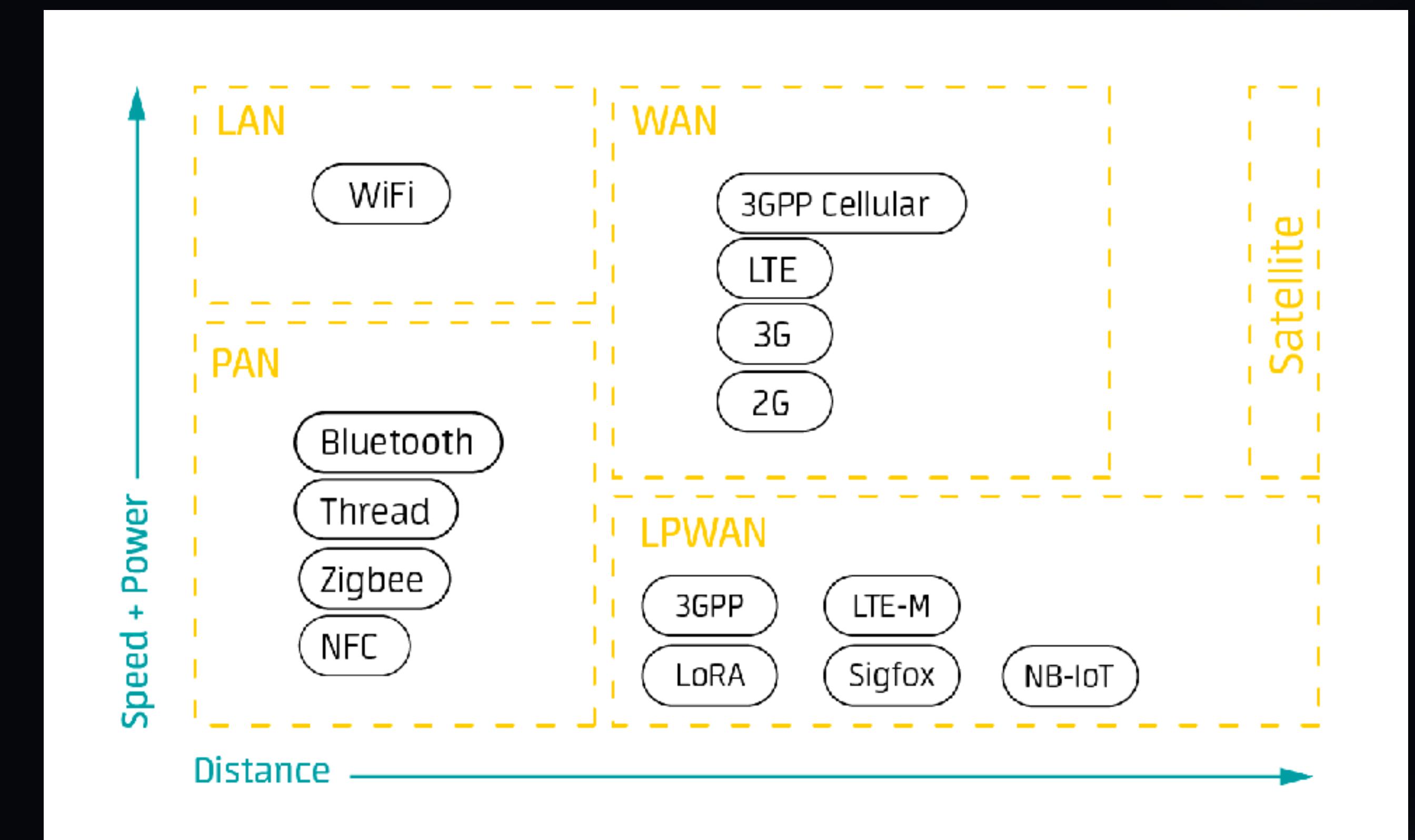
The physical interface to sense and actuate in the world

- Zephyr got you covered for Inputs and Outputs for your project
- Standard hardware protocols
 - I2S, I2C, SPI, UART, CAN BUS, MODBUS, etc
 - Sensor Framework
 - Easy to swap between sensors with the same “type”
 - Ex: Different temperature sensors are standardized to provide the same format
 - 100+ sensor drivers on Zephyr mainline
 - github.com/zephyrproject-rtos/zephyr/tree/main/drivers/sensor

Network Connectivity

How the physical device will talk to the outside world

- Most things are available as an L2 Layer on Zephyr
 - Ethernet
 - Modems
 - Wifi Offload
 - Cellular
 - Thread/OpenTread
- But you have non IP Layers
 - Lora/LoRaWan
 - Bluetooth

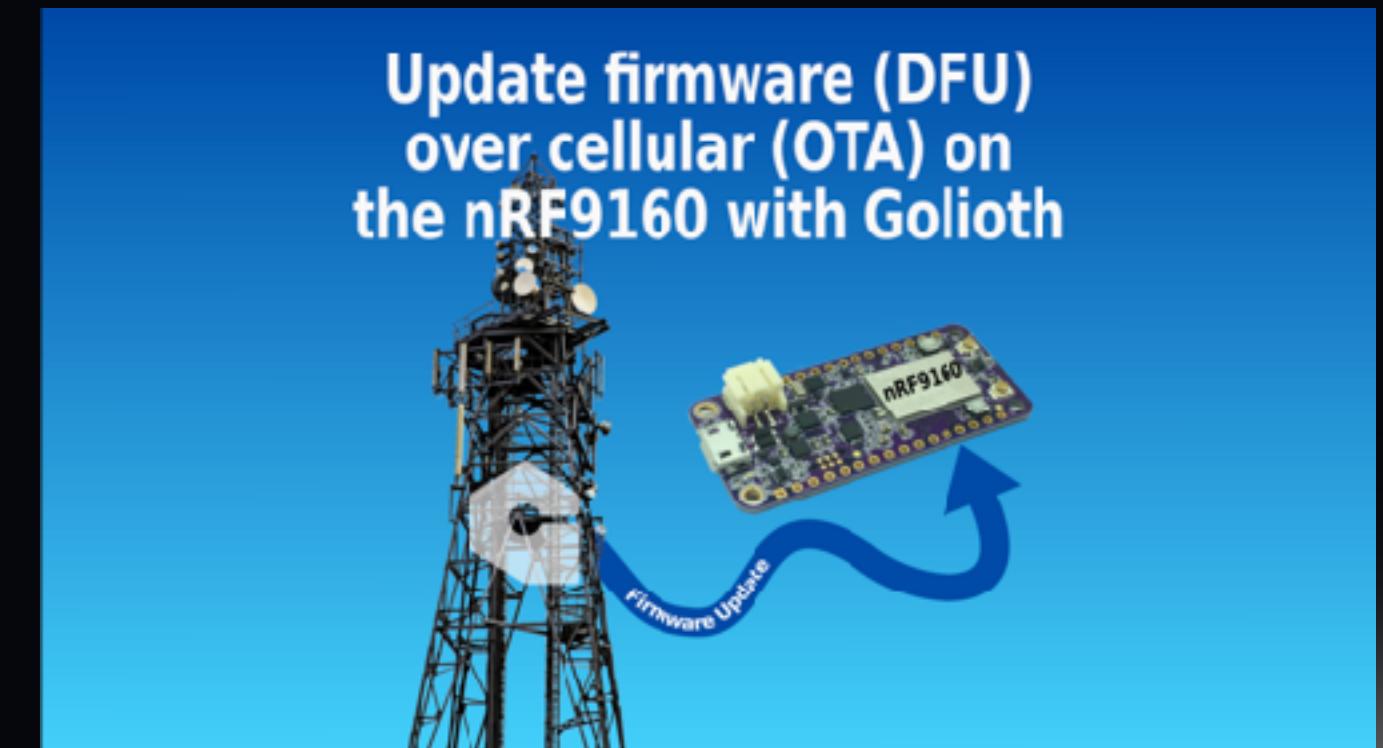


Source: Article on AVSystem blog:
IoT Connectivity – How to Choose the Right Connectivity Option for IoT

Content on network connectivity

Different ways to connect your device to the cloud

- Using Ethernet with Zephyr
 - <https://blog.golioth.io/zephyr-does-ethernet-too/>
- ESP32 AT as Wifi Modem on Zephyr
 - <https://blog.golioth.io/esp32-esp-at-enables-connectivity-on-any-zephyr-project/>
- DFU over Cellular
 - <https://blog.golioth.io/update-firmware-dfu-over-cellular-ota-on-the-nrf9160-with-golioth/>



Wrap up

Wrap up

What to keep in mind when developing IoT projects

- Remember: Most of the time, application refer to those real world objects, not the hardware/device itself.
- Is important to think how end users will interact with your (hidden) end device.
- Your device is not always online
 - Think about what to report to the cloud to make it look like is always online.
 - Taking into account async interactions
- Think on ways that the cloud can enhance your project and reduce burden on your constrained device.

References

References

- <https://blog.golioth.io/better-iot-design-patterns-desired-state-vs-actual-state/>
- <https://blog.golioth.io/provisioning-iot-devices-with-zephyr-mcumgr-and-golioth/>
- <https://blog.golioth.io/new-feature-updating-zephyr-settings-from-the-device-shell-and-more/>
- <https://blog.golioth.io/esp32-esp-at-enables-connectivity-on-any-zephyr-project/>
- <https://blog.golioth.io/zephyr-does-ethernet-too/>

github.com/golioth/samples

Thanks!

aviebrantz.com



Alvaro Viebrantz

Google Developer Expert for IoT and Founding Engineer at golioth.io

aviebrantz.com

 [@alvaroviebrantz](https://twitter.com/alvaroviebrantz)