Zebra DevCon 2023

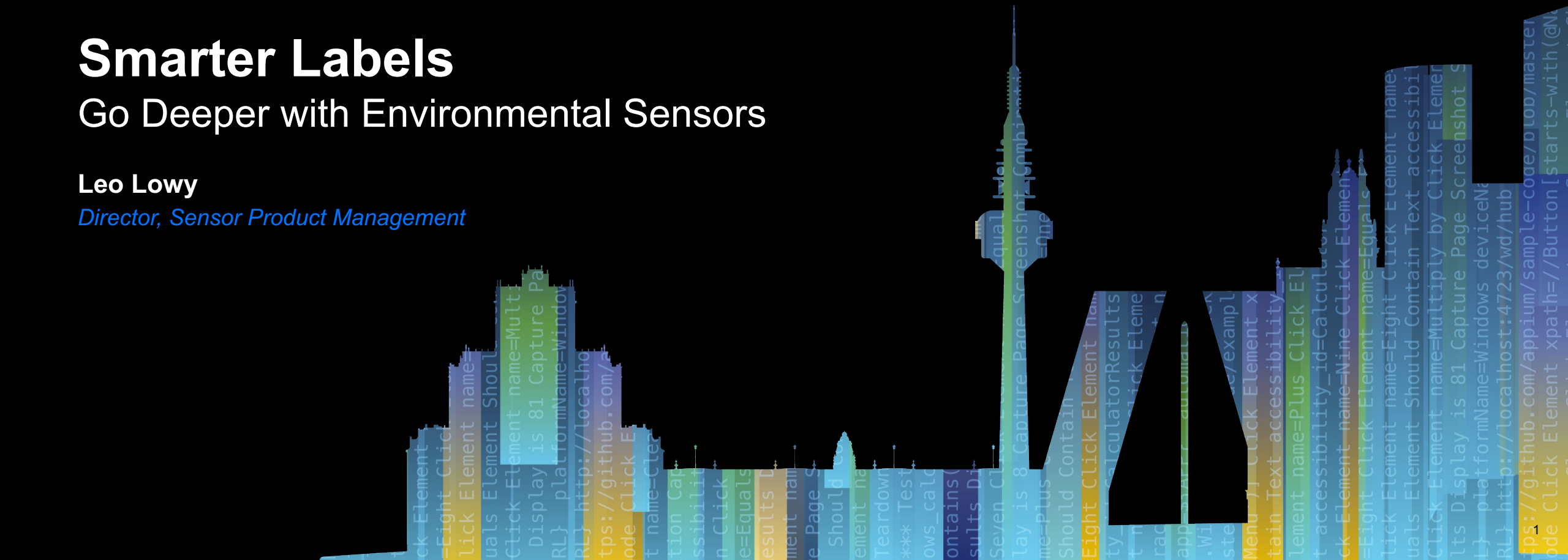
ZEBRA

Smarter Labels

Go Deeper with Environmental Sensors

Leo Lowy

Director, Sensor Product Management



The Critical Role of Cold Chain Logistics

Value drivers exist in multiple industries and applications

Zebra DevCon 2023

35B[1]
Annual pharmaceutical industry loss due to temperature-related issues.

48M[2]
People in the U.S. affected by food-borne illness every year.

22%[3] of the 60M tons of food wasted in the U.S. every year, 22% is lost after production and before consumption.

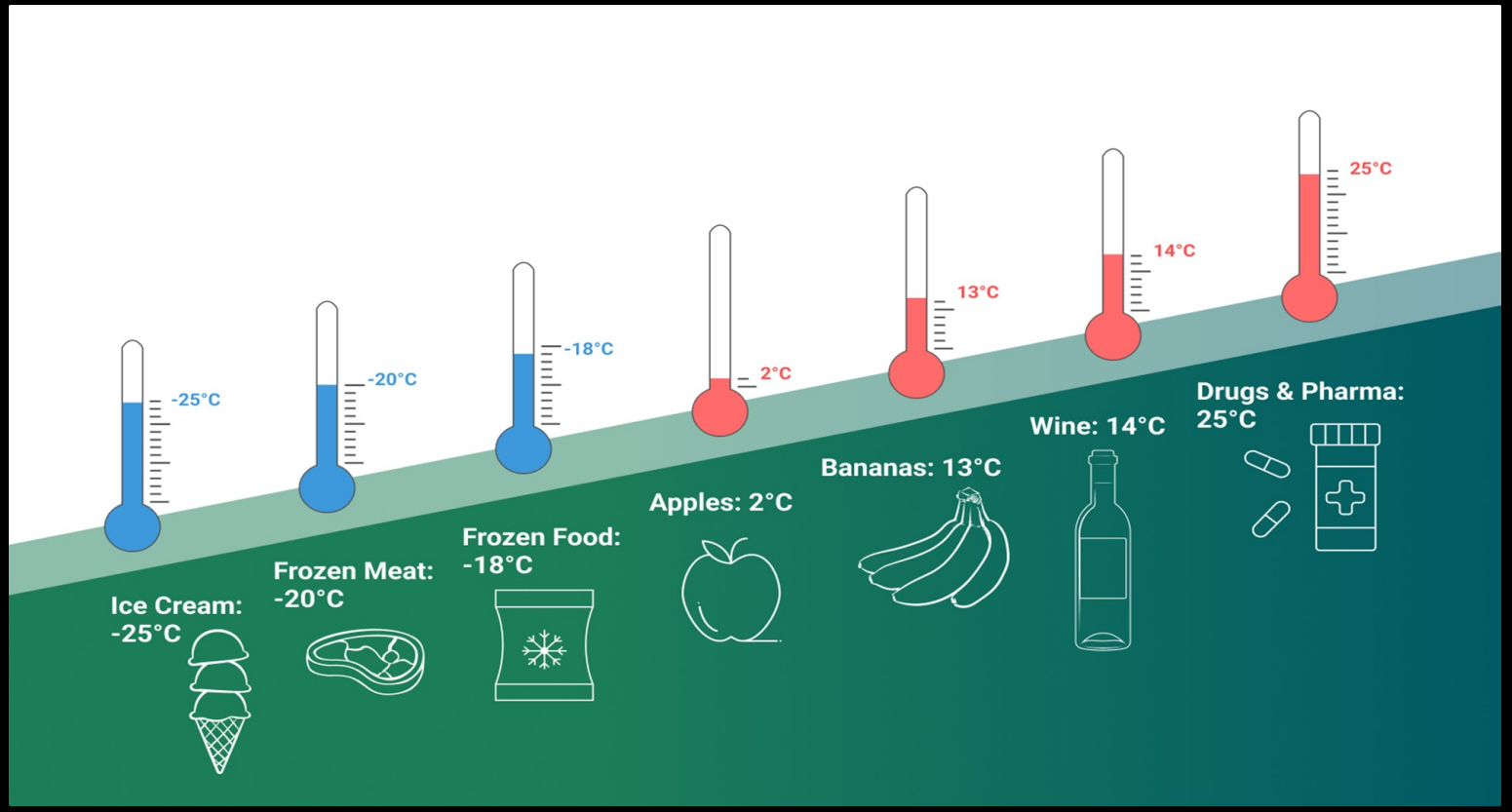
Electronic Sensors

Visual & RFID Sensors

Environmental Sensor Use Cases

Food & Pharma Manufacturing, Transportation and Logistics

Zebra DevCon 2023



Pharma / Biologics

Poultry/Beef

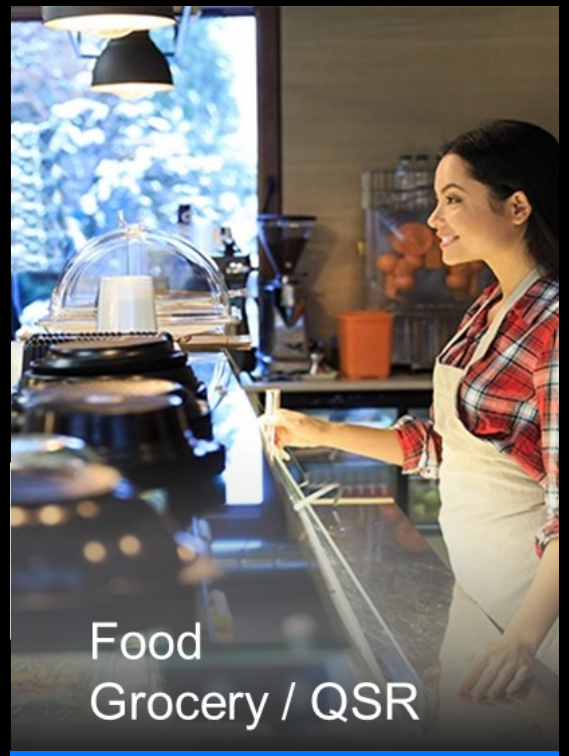
Ice Cream

Zebra Environmental Sensors: Core Markets & Applications

Well-suited to provide value in multiple industries and applications

Zebra DevCon 2023

Temperature sensitive products in transit



Ensure foods are stored in the proper condition

Temperature Monitoring during short- or longer-term storage

Validate high-heat processing

ZS300 Electronic Temperature Sensors

Zebra DevCon 2023

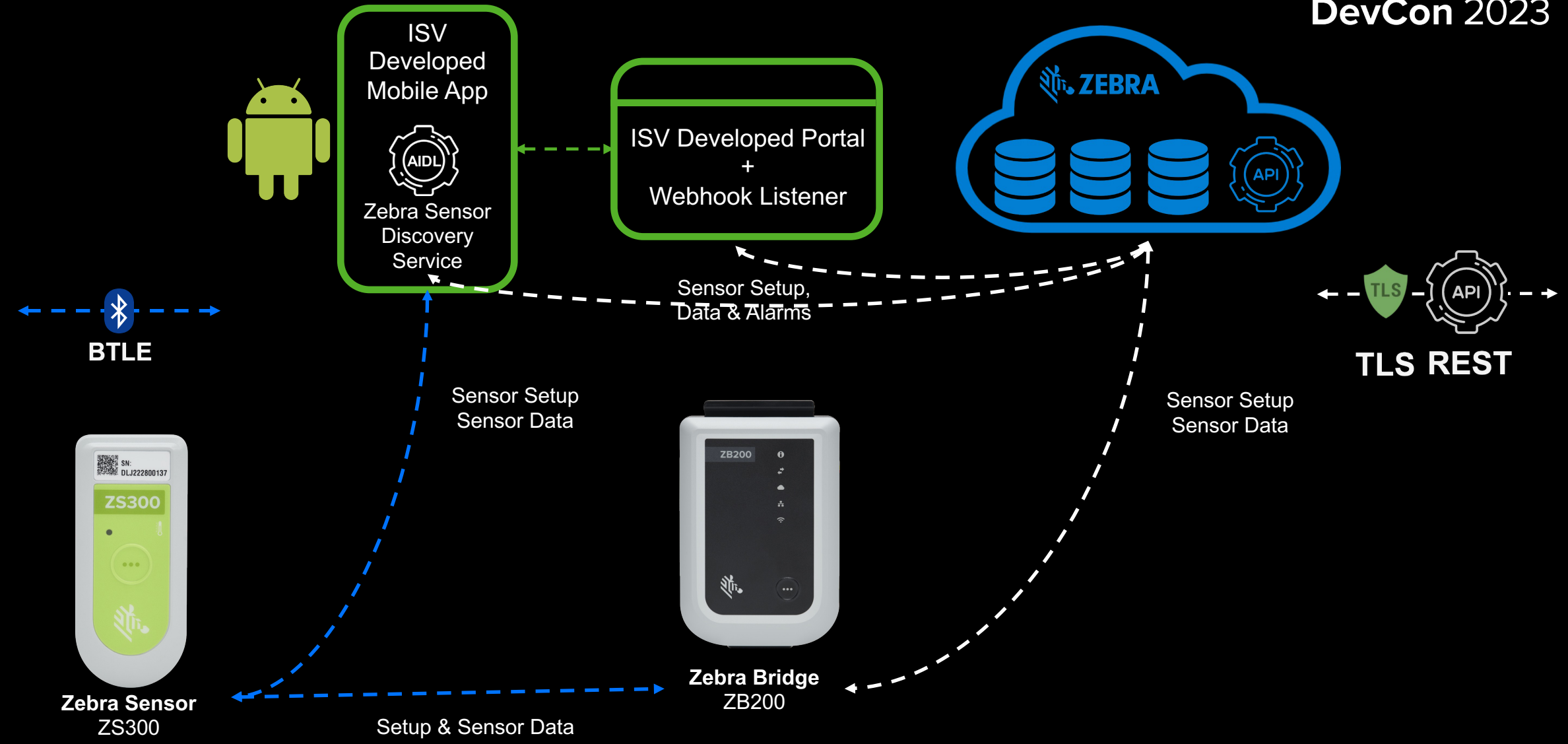
- Environmental Exposure Revealed via BLE and Cloud
- Detects temperatures from -40˚C to +85˚C, transmits data via Zebra Bridge or Android™ to the cloud
- Enables Real-Time responses to Temp events all along the Cold Chain (Transportation or Storage)
- Easily integrates with existing Transportation and Warehouse Management Systems

Ice Cream

Poultry/Beef

Pharma / Biologics

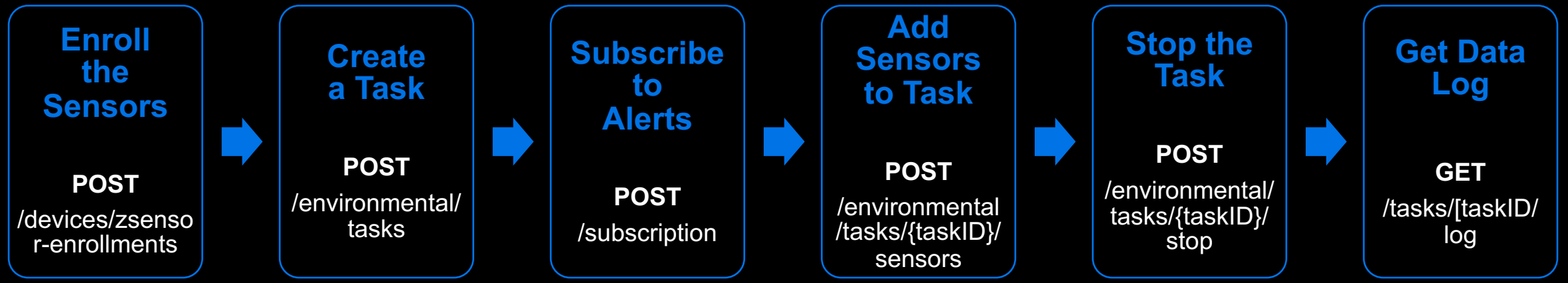
Solution Architecture & Connectivity



Sensor Workflow

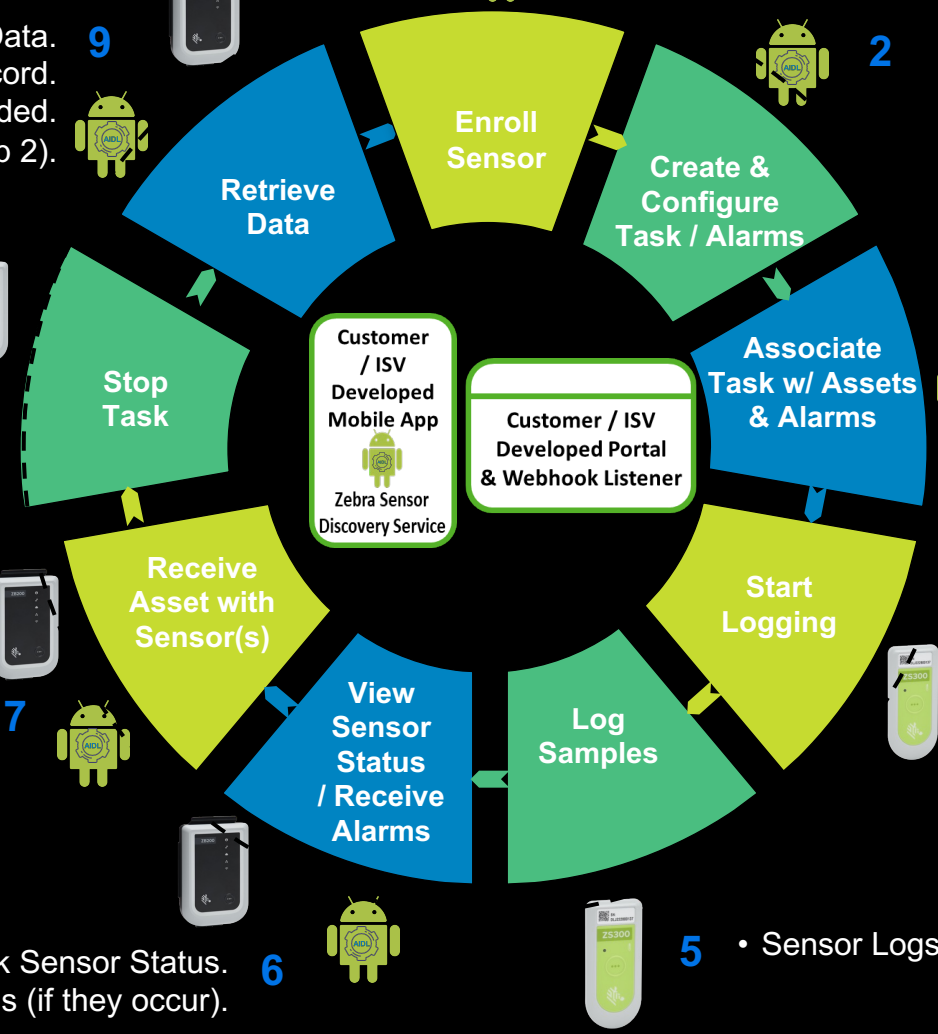
A Summary

Zebra DevCon 2023



Sensor Workflow

In Detail



1

- Press Sensor Button for three seconds to exit Deep Sleep Mode.
- Scan Sensors QR Code to capture Sensor ID.
- Connect to the Sensor.
- Enroll / Register Sensor with your tenant Sensor ID.
- Retrieve Calibration Cert (opt)

2

- Create Task ID (3-64 characters).
- Scan QR Code to capture Sensor ID.
- Press Sensor Button to wake the Sensor from sleep
- Associate Sensor w/ Task ID.
- Set Task Config: Reading Interval (15 sec to 240 Min), Start Mode (Immediate, Delayed), Upper & Lower Alarm Limits (-40 to 85c), Upper & Lower Alarm Delays (0 to 60 min), Data Overwrite (Yes, No).

3

- Associate Task with your self-defined Asset ID (1 to 120 chars). Subscribe to webhooks Alarms (opt).

4

- Place Sensor(s) with Asset(s)
- Start Reading Data, based on chosen Start Mode: Immediate, Start on Button Press, Start when below/above a set temp, Start until chosen time/date, Start after chosen delay). 16,000 records will be recorded.

5

- Sensor Logs Samples throughout . 16,000 records will be recorded.

6

- Check Sensor Status.
- Receive Alarms (if they occur).

7

- Receive Assets.
- Retrieve Sensor(s)

8

- Stop Task.
- Record Battery Status.
- Unsubscribe from Alarms

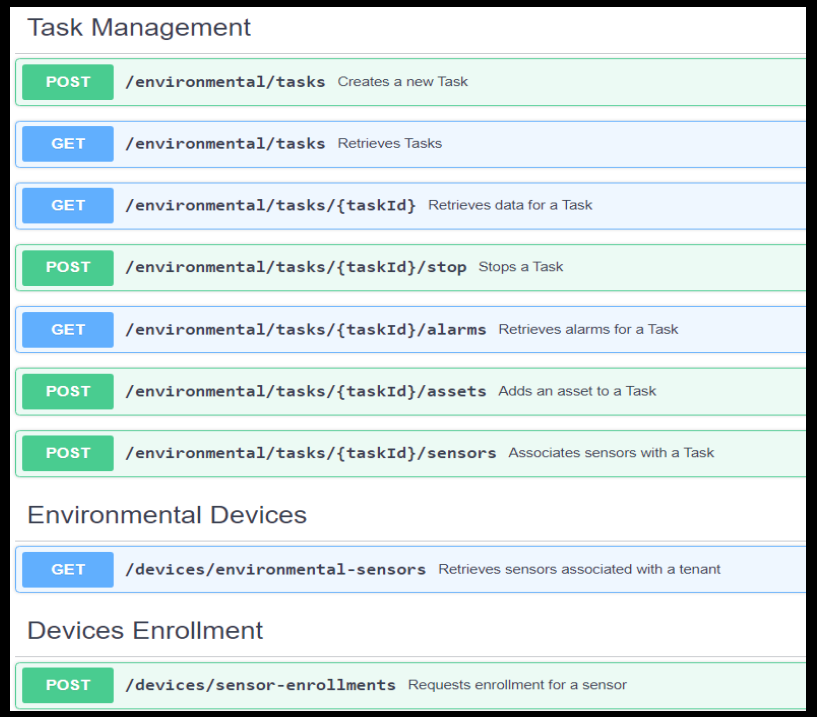
9

- Retrieve Data.
- Transfer Data to System of Record.
- Create Reports, as needed.
- Sensor is ready for another task (Step 2).

Demo Time!

Management APIs

- Included at No Charge
- Enable Management of Sensors & Task Configuration
- Enables retrieving Sensors Certificate of Calibration



Example: Creating a new task →

```
curl -X 'POST' \
  'https://api.zebra.com/v2/environmental/tasks' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "task_from_details": {
    "task_details": {
      "name": “Task Name",
      "interval_minutes": 240,
      "interval_seconds": 59,
      "loop_reads": false,
      "start_immediately": {},
      "start_delayed": {
        "on_button_press": false,
        "delayed_temp_below": 5.5,
        "delayed_temp_above": 5.5,
        "delayed_minutes": 30,
        "delayed_until": "2023-03-08T05:41:43.620792Z"
      },
      "sensor_type": "SENSOR_TYPE_TEMPERATURE",
      "alarm_low_temp": 5.5,
      "alarm_high_temp": 10.5,
      "notes": “These are my notes for this task",
      "low_duration_minutes": 60,
      "low_duration_seconds": 59,
      "high_duration_minutes": 60,
      "high_duration_seconds": 59,
      "required_sensors": 100
    }
  }
}'
```

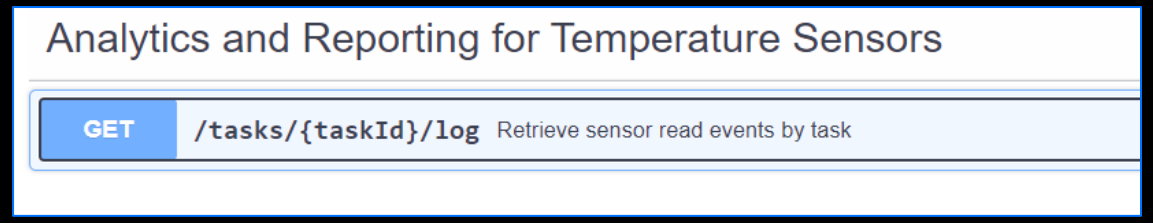
Response:

```
{
  "id": "51f9b386-62d5-4867-9ba4-0e232936227e"
}
```

More Details on Zebra DevPortal

Data APIs

- Included in LogView and EventView subscriptions
- Provides ability to securely query and retrieve data gathered by sensors



Example: Getting a tasks logs:

```
curl -X 'GET' \
'https://api.zebra.com/v2/data/environmental/tasks/MyTaskID/log?startTime=2022-11-01T08%3A00%3A00.006Z&endTime=2022-11-08T16%3A00%3A00.006Z&limit=100' \ -H 'accept: application/json'
```

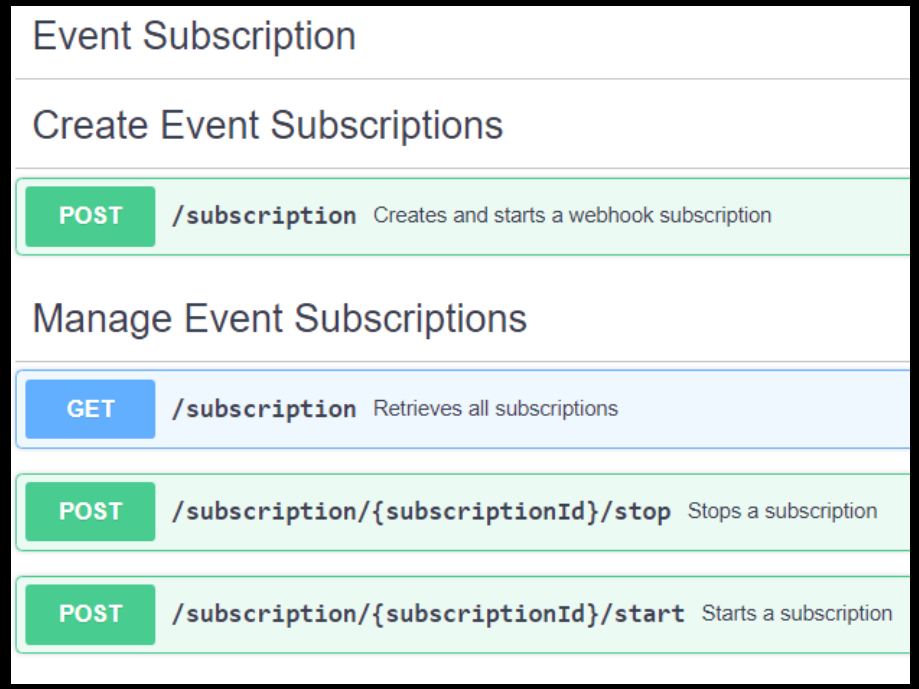
Response:


```
{
 "results": [
  {
   "type": "beacon",
   "event": {
    "id": "6359fcb8-96a0-461c-90b1-07dbb002c063",
    "timestamp": 1633359112806,
    "deviceId": "bridge-id_or_phone-id",
    "data": {
     "format": "beacon",
     "id": "sensormac",
     "value": "-2.06",
     "rssi": -51
    },
    "analytics": {
     "recordedTimestamp": 1633359112806,
     "resourceId": "bridge-id_or_phone-id",
     "tenant": "my_tenant",
     "timestamp": 1633359112806,
     "meta": {
      "data": {
       "taskId": "task id"
      }
     },
     "coordinates": {
      "global": {
       "lat": 0,
       "lng": 0
      }
     }
    },
    "decode": {
     "temperature": {
      "alert": true,
      "deviation": 0.3,
      "format": "celsius",
      "taskId": "task id",
      "sample": -2.06
     }
    }
   }
  }
 ]
}
```


More Details [on Zebra DevPortal]

Alarm APIs

- Included in EventView subscription
  - Allows subscribing to Temp. Excursion Alarms
  - NOTE: webhook listener required



Example:
Creating a subscription:

```
{
   "headers": {
       "apikey": "abc12345",
       "tenant": "12345abc“
   },
   "taskIds": [
       "449d226f3a1a4ad48e5c552831aa9334",
       "629427d184744093a5dd3cd6b19c96b1"
  ],
   "epcis": true,
   "name": "The name of my new webhook subscription",
   "webhookUrl":
  "https://hooks.myorganization.com/services/hooks/myendpoint",
  "webhookVerb": "POST"
}
```

Response:

```
{
   "subscription": {
       "id": "B5JD07WPJ3BDVXRR"
   }
}
```

More Details on Zebra DevPortal

Android Sensor Discover App: “ZSFinder”

Zebra DevCon 2023

Features
• Communicates with ZS300 Sensor via Bluetooth® Low Energy (v5.2).
• Discovers Sensors and enables bidirectional communication with them.
• Connects to Zebra Cloud Platform to upload sensor data.
• Offers an Android™ Interface Definition Language interface, allowing data to be retrieved from sensor for transmission to a user chosen backend system. • ***NOTE**: The AIDL interface is for use with the SelfView Subscription. Data is not signed during transport.*
• For use on Android™ v8.1 and above. • Access via port 80, 443 to scv.zpc.zebra.com and acs.zebra.com for certificate exchange and data transfer to the Zebra Cloud Platform. • Requires Bluetooth® Low Energy v4.1 or later.
• Authenticated via the Android™ Services Authentication API
• ZSFinder is available on Google Play

Android™ Software
Sensor Discovery Service
“ZSFinder”

Sensor BTLE Advertising Packet

The Advertising Packet is a structured set of information the ZS300 sensor sends every 1.5 seconds to let Bluetooth Low Energy (BTLE) hardware and software-based receivers “find” the sensor.

It contains information on the sensors Unique ID, battery status, last temperature read and if the sensor is in an alarm condition, plus many other attributes.

The Advertising Packet also contains forward-looking data fields for potential future products.

The Advertising Packet, along with the Sensor Discovery Service app (ZSFinder) are key assets for developers using the SelfView Subscription.

Parameter	Length	Value and/or Description
Flag Length	1 byte	0x02: Length of flag data is 2 bytes
Flag Type	1 byte	0x01: Type set to 1 to indicate ‘various’ flags
Flag Value	1 byte	0x06: Set bits 1 and 2 to indicate LE general discoverable mode, BR/EDR not supported (see table below for details)
Manufacturer ID Length	1 byte	0x1B: Total manufacturer data length is 27 bytes
Manufacturer ID Type	1 byte	0xFF: Type set to 0xFF to indicate mfg-specific data
Manufacturer ID Value	2 bytes	0x01F1: Indicates Zebra Bluetooth code
Datalogger Type	1 byte	0x01 = Datalogger #1 with temperature sensor
Battery Status	1 byte	0 – 100 %
System ID	6 bytes	Unique identifier assigned to device
Reserved for future use	1 byte	Zero pad for byte-alignment purposes
Sensor Status	2 bytes	Bits 02:00 – temperature sensor Bits 05:03 – reserved for future use Bits 08:06 – reserved for future use Bits 11:09 – reserved for future use Bits 14:12 – reserved for future use Bit 15 – alarm
Temperature Samples	2 bytes	Number of samples stored (up to 16,000)
reserved for future use	2 bytes	Number of samples stored (up to 16,000)
reserved for future use	2 bytes	Number of samples stored (up to 16,000)
reserved for future use	2 bytes	Number of samples stored (up to 16,000)
reserved for future use	2 bytes	Number of samples stored (up to 16,000)
Last Temperature Sampled	2 bytes	Last temperature value sampled
Reserved for future use	1 byte	Zero pad to fill to 31 bytes total
Total Length	**31 bytes**	

Dev Tools Best Practices

Sensor Discovery App (ZSFinder)

- Use Android v8.1 or later to ensure BLTE functionality.
- Ensure that Bluetooth® is enabled on the Android™ device, and prompt user to enable if not enabled.
- If using the “SelfView” subscription, ensure that data is both retrieved from the Sensor and securely transmitted to your backend system(s).
- If using the “SelfView” subscription, ensure data is retained for up to 7 years, to support future product handling process audits.
- If using the “EventView” subscription, set up a webhook listener to capture Alarms.
- If using the Android™ Interface Definition Language feature to retrieve data from the Sensor, review the Google AIDL documentation.
- Provide feedback to end user to ensure they stay within range of the sensor when collecting data. Most Android™ devices will support a 30-foot range. Provide feedback when data collection is complete.

Zebra Cloud Platform (APIs)

- Review the Developer Guide, available here
- Keep your API keys and Tokens private and secure.
- Handle any errors that may occur during API communication in a graceful manner. This could include retrying failed requests or displaying an appropriate error message to the user.
- Use pagination for large data sets: If an API returns a large data set, consider using pagination to retrieve the data in smaller chunks. This can help to improve the performance and scalability of your application.
- Know and handle rate limits, providing users feedback where needed. The daily rate limit is 5000 calls per day per API key, with a spike rate limit of 7 calls per second per API Key.
- Keep current on API changes over time, updating your applications as needed to keep them up-to-date and fully functional. Update information is available here.
- Use the correct method for the intended action (e.g., GET, POST, PUT, DELETE). Example, use GET to retrieve data and POST to create new resources.

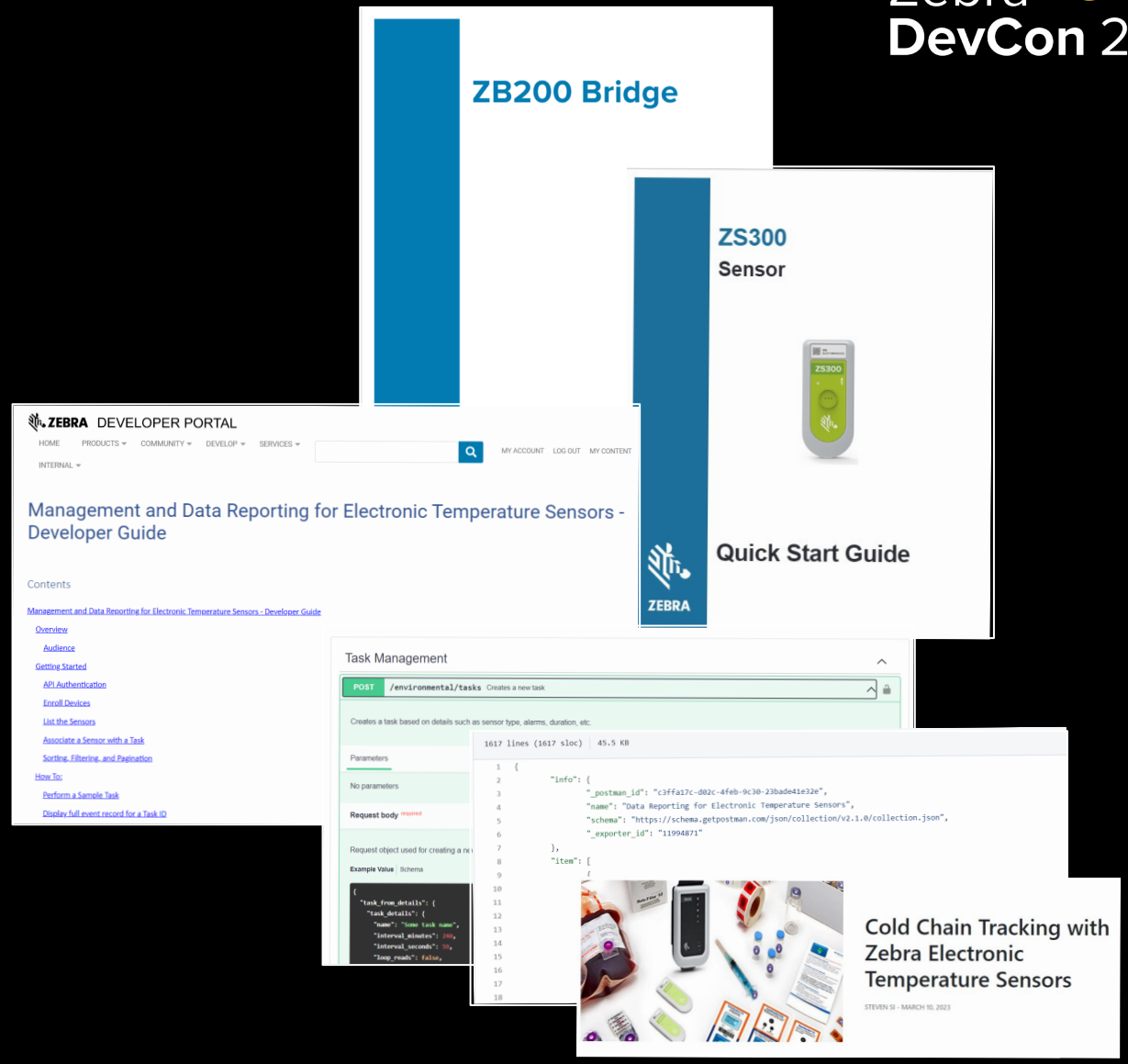
Documentation

Software

- Developer Portal
  - “Creating an ZS300 app” blog post
  - Developer Getting Started Guide
  - Postman Collection
  - OpenAPI Spec including
    - API Description
    - Example Use Case
    - Sample Code
    - Error Responses
    - Postman Collection
  - Android™ AIDL Documentation

Hardware

- User Guides for ZS300 and ZB200
- Quick Start Guides for ZS300 and ZB200
  - Setup Utility App will be included with ZB200



API Subscriptions

Zebra DevCon 2023

Subscription Types	Features	Term
Developer Subscription Enables Developers to create their apps	Unlimited API calls Access to Android™ Sensor Discovery Service Access to Sensors Management APIs Access to Data APIs Access to Alerts	90 Days (Renewable for up to 6 months
SelfView Subscription Access Data from your Cloud	Unlimited API calls Access to Android™ Sensor Discovery Service Access to Sensors Management APIs	1 year 3 years
LogView Subscription Enables apps to operate from Zebra's cloud	Unlimited API calls Access to Android™ Sensor Discovery Service Access to Sensors Management APIs Access to Data APIs	1 year 3 years
EventView Subscription Enables apps to receive real-time Alerts from Zebra's cloud platform	Unlimited API calls Access to Android™ Sensor Discovery Service Access to Sensors Management APIs Access to Data APIs Access to Alerts	1 year 3 years

Visual & RFID Sensors

Zebra DevCon 2023

Visual Sensors

RFID Sensors

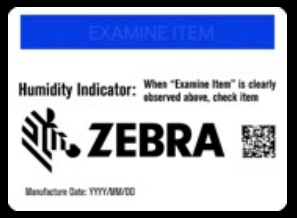
ZeOn-Demand Printable Indicators

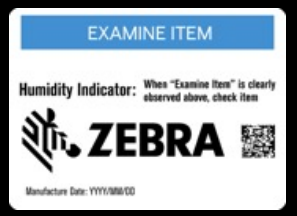
Zebra DevCon 2023

- Environmental Exposure Revealed with a Simple Machine Readable Visual
- Combines environmental indicator with on-demand thermal printing
- Reveals Insights into a sensitive assets
- Supports quality processes and workflows across the spectrum of label materials
- Come Build with Us!

Temperature

Humidity/Moisture



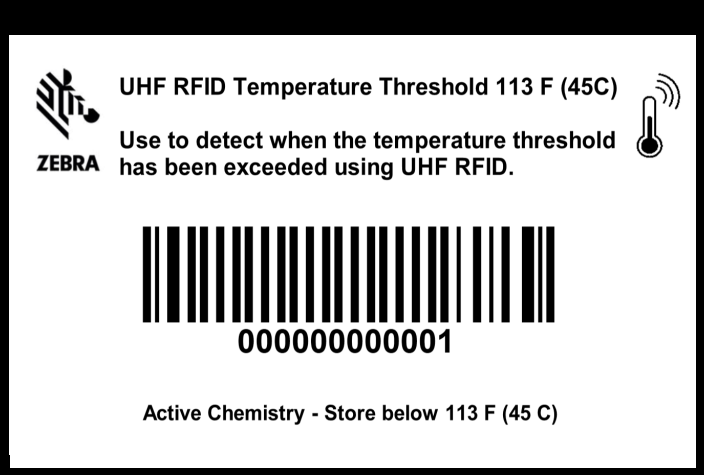


Sterilization

ZeOn-Demand High Temp RFID Labels (Q4-2023)

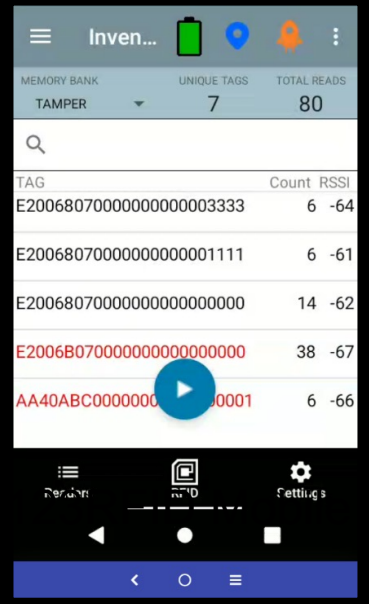
Zebra DevCon 2023

- Environmental Exposure Revealed Via existing RFID Infrastructure
- Combines environmental indicator with on-demand thermal printing
- Supports “High Temp” detection: 113F (45c)
- Currently in “Market Trial”: Seeking ISV’s and customers to build out software ecosystem



Product Kept Below Temp Threshold

Product Exceeded Temp Threshold



…and, reach out to us at sensorpm@zebra.com

Questions

Zebra DevCon 2023

ZEBRA

Thank You