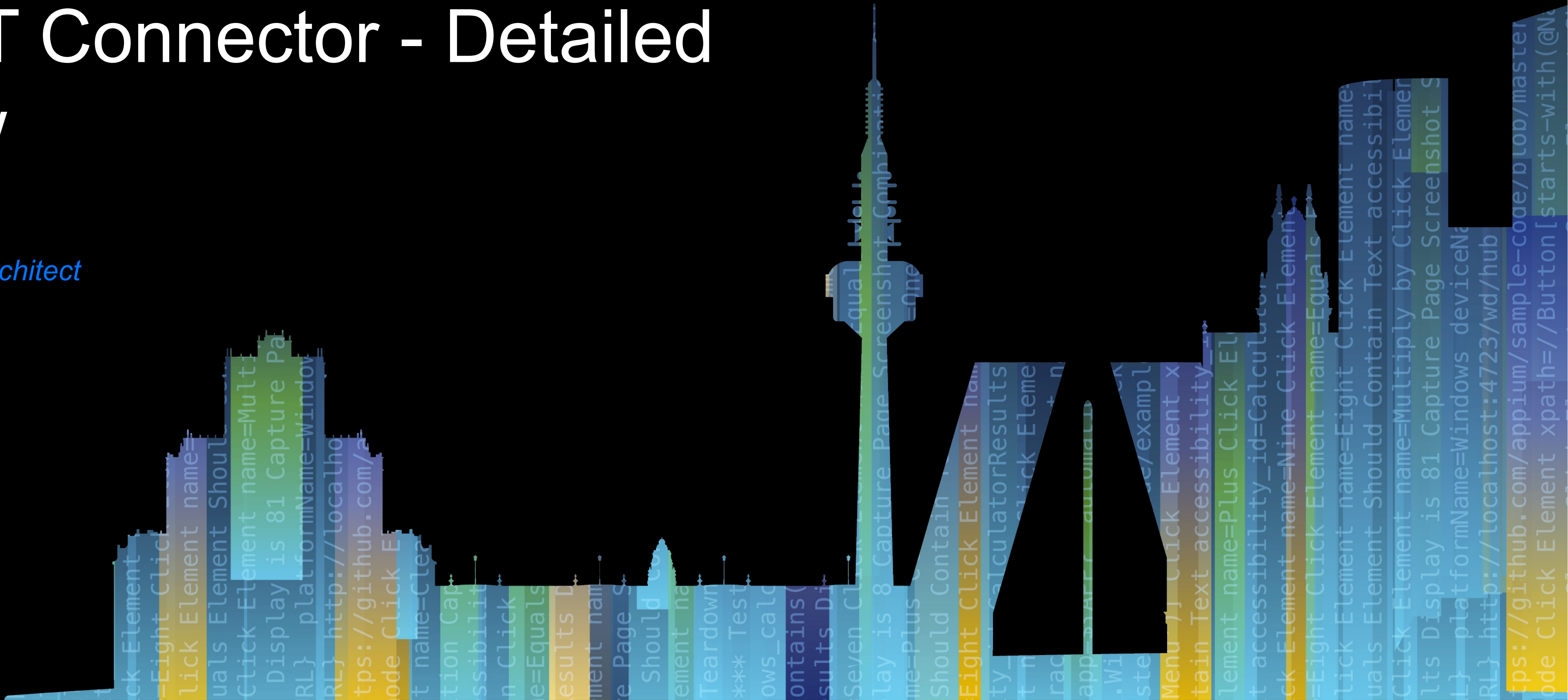
Zebra DevCon 2023

ZEBRA

Zebra IoT Connector - Detailed Overview

Alex Lavie
Zebra Sales Enginer/Architect

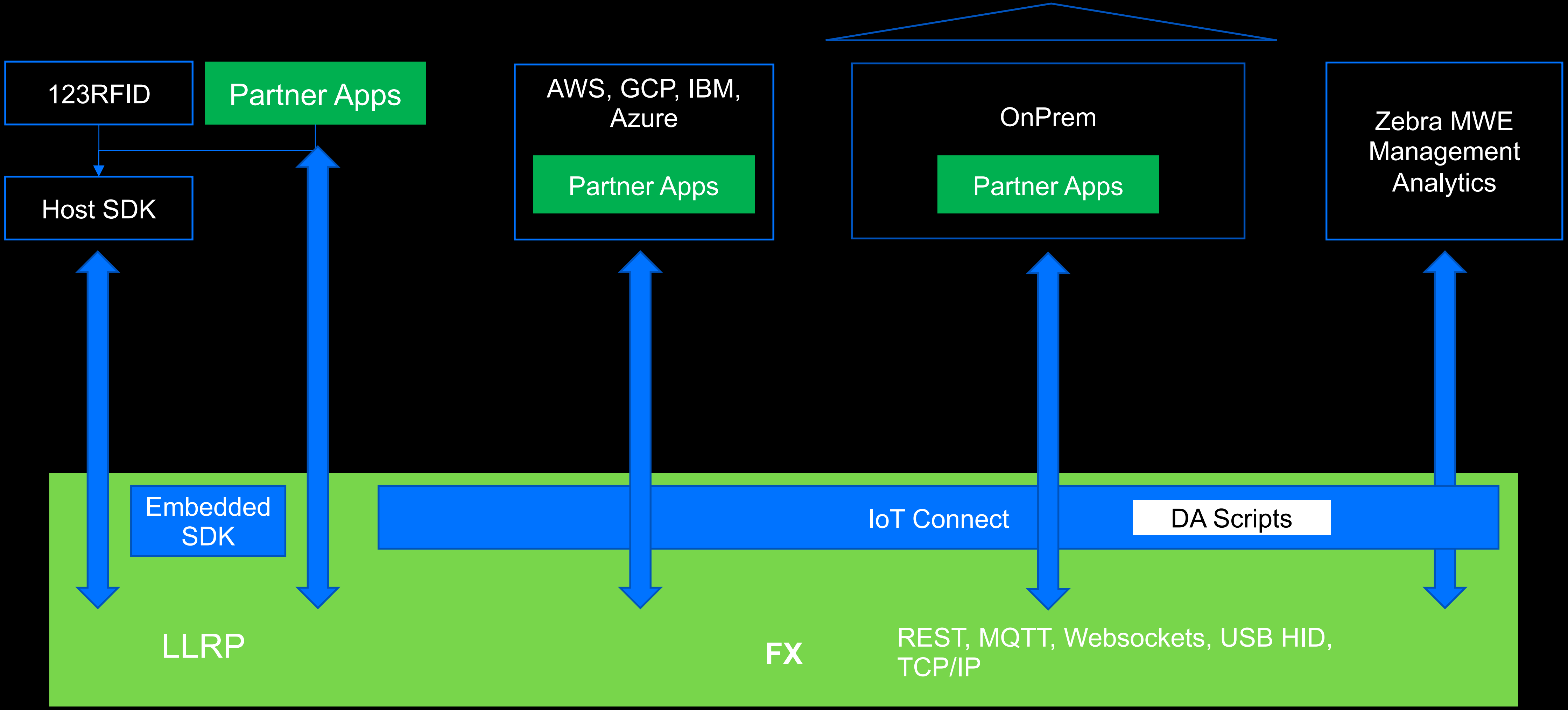


Zebra IoT Connector

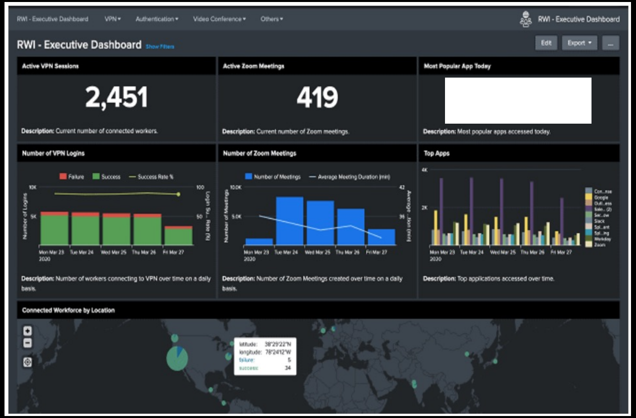
- Free of charge standard feature, built-in reader feature set that replaces FX Connect and on-device CloudConnect
- Fully automated, real-time enterprise data collection tool using modern IoT protocols such as MQTT, WebSockets and HTTPS
- Routes data from Zebra devices into your preferred IoT endpoint, whether it's a data lake in the cloud or your on-premises web server
- Accesses vital information from your fleet of Zebra devices, such as health alerts, with date and time stamps
- Manages and controls readers using MQTT or REST APIs
- Simple to configure, off-the-shelf tool—no coding required
- Allows script-based app development using Python or NodeJS to do more sophisticated analytics on device, enabling users to make real-time decisions at the edge

General FX readers Application Architecture

Zebra DevCon 2023



Zebra
DevCon 2023


RWI - Executive Dashboard

Active VPN Sessions

2,451

Description: Current number of connected workers.

Active Zoom Meetings

419

Description: Current number of Zoom meetings.

Most Popular App Today

Description: Most popular apps accessed today.

Number of VPN Logins

Description: Number of workers connecting to VPN over time on a daily basis.

Number of Zoom Meetings

Description: Number of Zoom Meetings created over time on a daily basis.

Top Apps

Description: Top applications accessed over time.

Connected Workforce by Location

Customer Dashboard/Application
in Cloud or On-prem
- Where is my asset?
- How are my devices doing?

Scenario - Customer leverages analytics services offered by 3rd party cloud or Zebra cloud to create dashboard

Scenario - Customer dashboard (customer implements its own analytics)

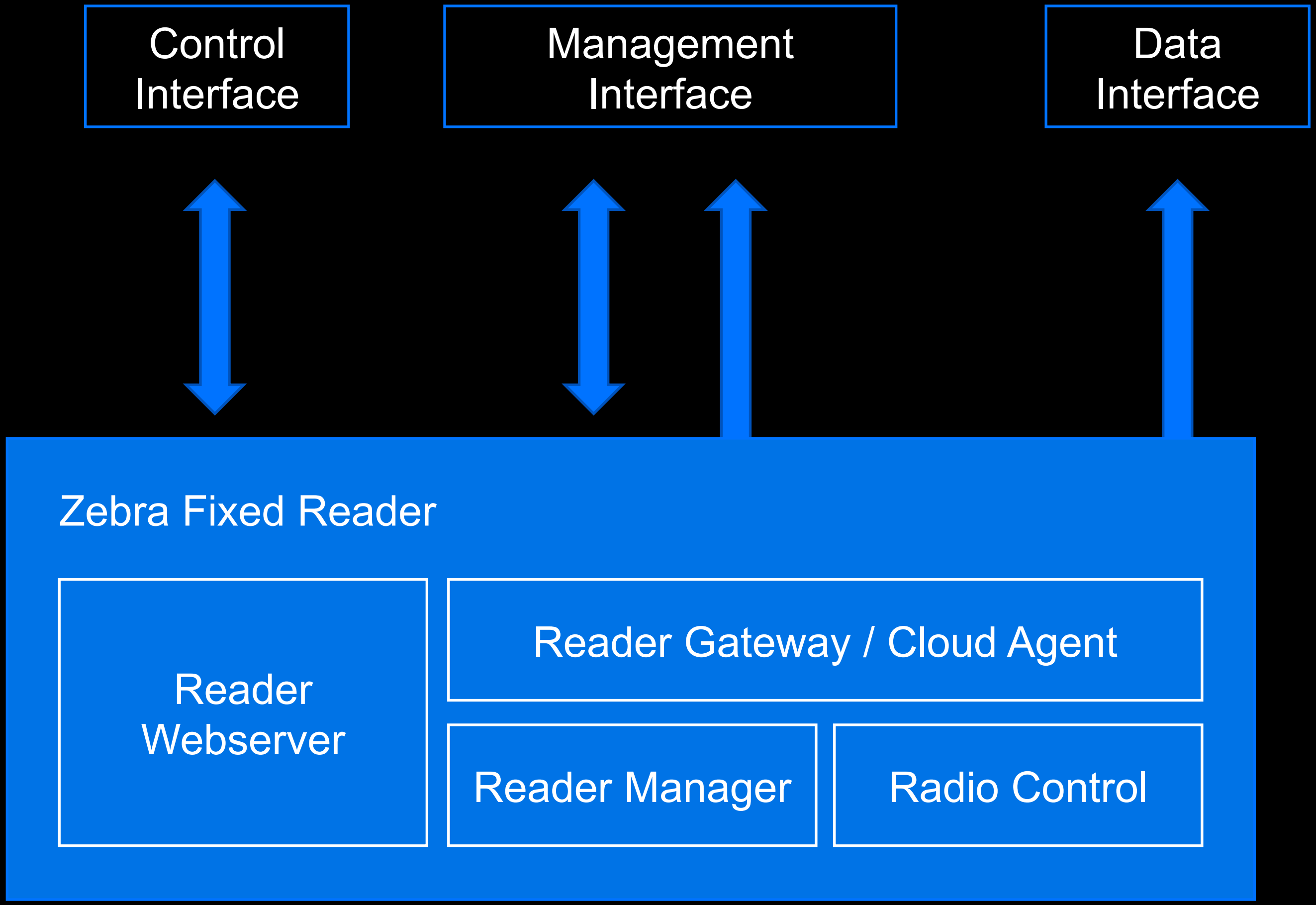
AWS, GCP, IBM, Azure

Health Events

Tag Data

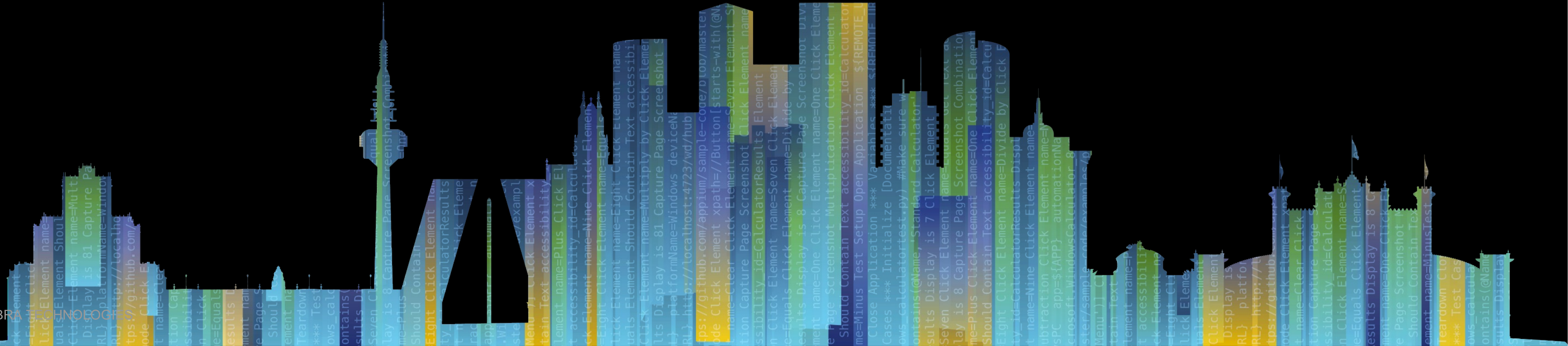
Tag Data

Health Events


Control Interface
Management Interface
Data Interface
Zebra Fixed Reader
Reader Webserver
Reader Gateway / Cloud Agent
Reader Manager
Radio Control

Zebra
DevCon 2023

Reader Operations with IoTConnect



IoTConnect vs. Host or Embedded SDKs

Zebra DevCon 2023

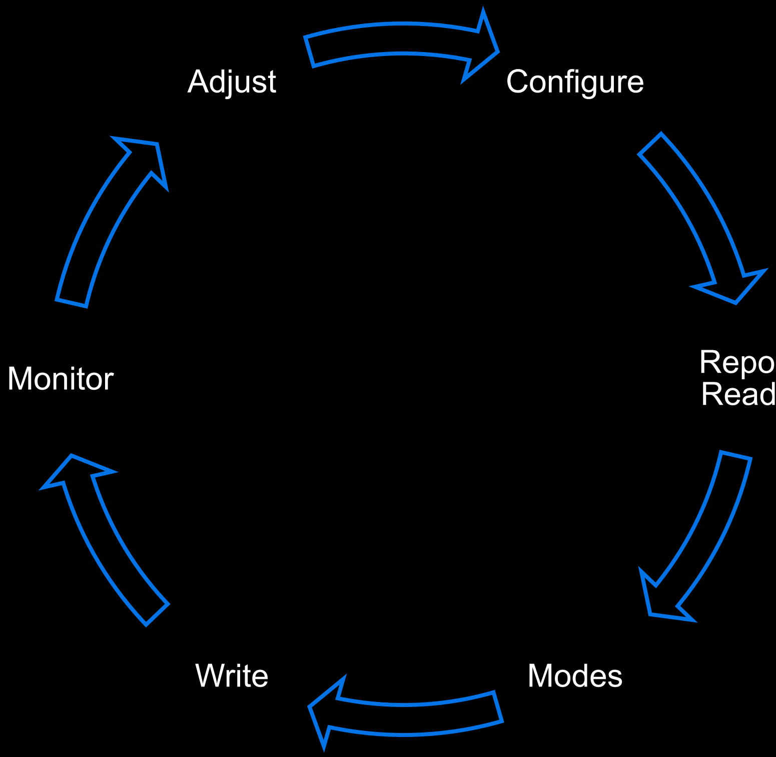
- Configuration vs. Coding
- Less complexity
- Easier deployment and prototyping
- Faster integration
- On reader intelligence

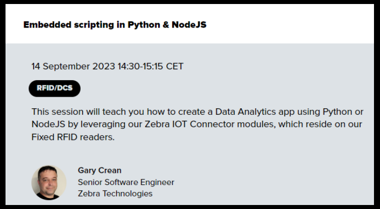
Main Activities in Action

Zebra DevCon 2023

- All these activities can be done
- FX Reader WebConsole
- Using Rest APIs with Tools like Postman
- Mqtt based tools like Node-Red
- Use Zebra RFID Reader Management Console

- Today we will not have time to cover:
- Deployment of Certificates
- Deployment of Functions and applications for on Reader Intelligence (*covered in a session tomorrow*)




Embedded scripting in Python & NodeJS

14 September 2023 14:30-15:15 CET

RFID/DCS

This session will teach you how to create a Data Analytics app using Python or NodeJS by leveraging our Zebra IOT Connector modules, which reside on our Fixed RFID readers.

Gary Crean
Senior Software Engineer
Zebra Technologies

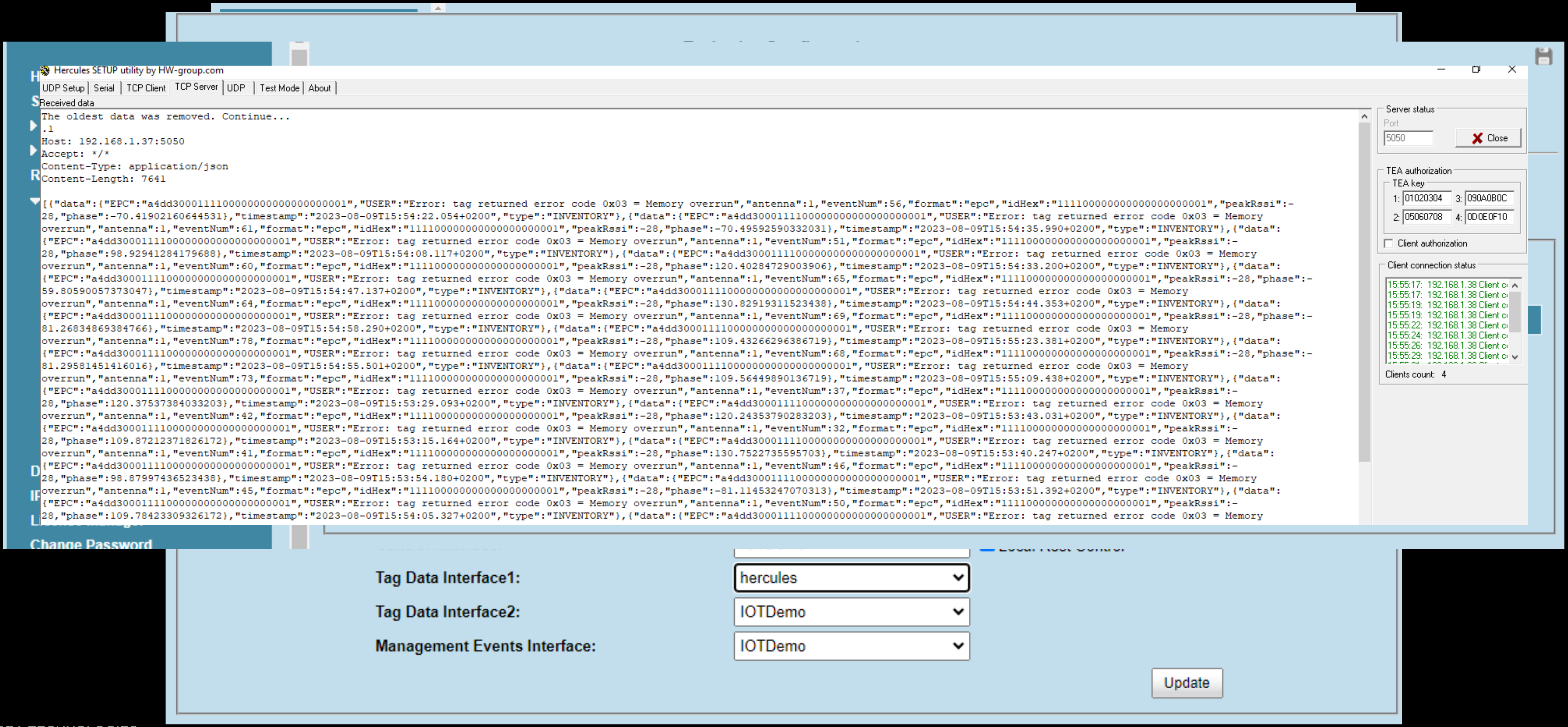
Configure Reader for IOTConnect

Demo

Zebra DevCon 2023

- Define Endpoints for Management and Data
- Set Security for Data flows
- Define rules for data Retention and optimise Network utilisation
  - Start IoTConnector
  - Start the Reading process
- Export Configuration to use for other readers

Using WebConsole


Hercules SETUP utility by HW-group.com
UDP Setup
Serial
TCP Client
TCP Server
UDP
Test Mode
About
Received data
Server status
Port
5050
Close
TEA authorization
TEA key
1: 01020304
3: 090A0B0C
2: 05060708
4: 0D0E0F10
Client authorization
Client connection status
15:55:17: 192.168.1.38 Client c
15:55:17: 192.168.1.38 Client c
15:55:19: 192.168.1.38 Client c
15:55:19: 192.168.1.38 Client c
15:55:22: 192.168.1.38 Client c
15:55:24: 192.168.1.38 Client c
15:55:26: 192.168.1.38 Client c
15:55:29: 192.168.1.38 Client c
Clients count: 4
Change Password
Update

```
The oldest data was removed. Continue...
.1
Host: 192.168.1.37:5050
Accept: */*
Content-Type: application/json
Content-Length: 7641
```

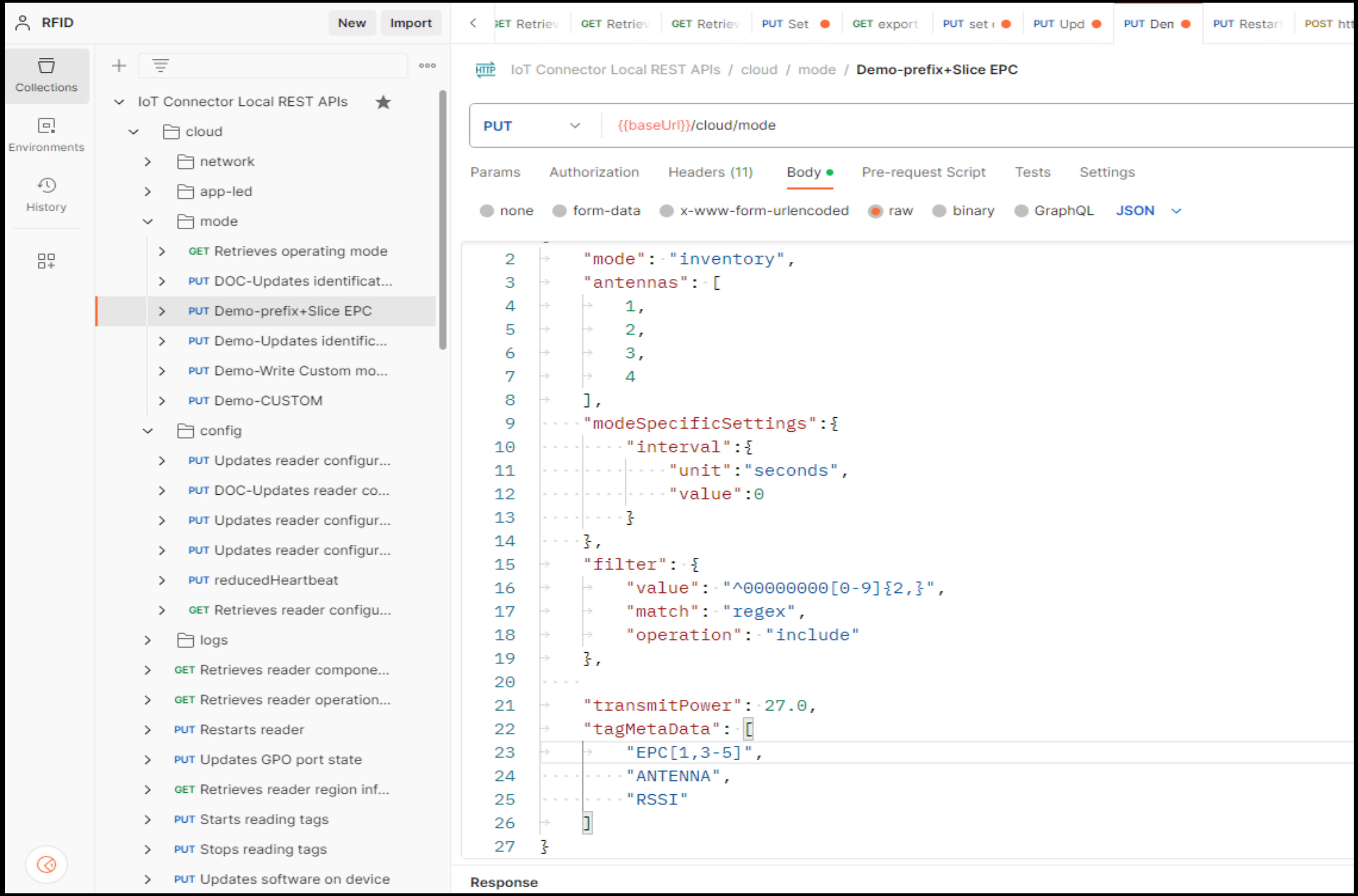
```
[{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":56,"format":"epc","idHex":"111100000000000000000001","peakRssi":-
28,"phase":-70.41902160644531},"timestamp":"2023-08-09T15:54:22.054+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
overrun","antenna":1,"eventNum":61,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":-70.49592590332031},"timestamp":"2023-08-09T15:54:35.990+0200","type":"INVENTORY"},{"data":
{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":51,"format":"epc","idHex":"111100000000000000000001","peakRssi":-
28,"phase":98.92941284179688},"timestamp":"2023-08-09T15:54:08.117+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
overrun","antenna":1,"eventNum":60,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":120.40284729003906},"timestamp":"2023-08-09T15:54:33.200+0200","type":"INVENTORY"},{"data":
{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":65,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":-
59.80590057373047},"timestamp":"2023-08-09T15:54:47.137+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
overrun","antenna":1,"eventNum":64,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":130.82919311523438},"timestamp":"2023-08-09T15:54:44.353+0200","type":"INVENTORY"},{"data":
{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":69,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":-
81.26834869384766},"timestamp":"2023-08-09T15:54:58.290+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
overrun","antenna":1,"eventNum":78,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":109.43266296386719},"timestamp":"2023-08-09T15:55:23.381+0200","type":"INVENTORY"},{"data":
{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":68,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":-
81.29581451416016},"timestamp":"2023-08-09T15:54:55.501+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
overrun","antenna":1,"eventNum":73,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":109.56449890136719},"timestamp":"2023-08-09T15:55:09.438+0200","type":"INVENTORY"},{"data":
{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":37,"format":"epc","idHex":"111100000000000000000001","peakRssi":-
28,"phase":120.37537384033203},"timestamp":"2023-08-09T15:53:29.093+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
overrun","antenna":1,"eventNum":42,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":120.24353790283203},"timestamp":"2023-08-09T15:53:43.031+0200","type":"INVENTORY"},{"data":
{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":32,"format":"epc","idHex":"111100000000000000000001","peakRssi":-
28,"phase":109.87212371826172},"timestamp":"2023-08-09T15:53:15.164+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
overrun","antenna":1,"eventNum":41,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":130.7522735595703},"timestamp":"2023-08-09T15:53:40.247+0200","type":"INVENTORY"},{"data":
{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":46,"format":"epc","idHex":"111100000000000000000001","peakRssi":-
28,"phase":98.87997436523438},"timestamp":"2023-08-09T15:53:54.180+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
overrun","antenna":1,"eventNum":45,"format":"epc","idHex":"111100000000000000000001","peakRssi":-28,"phase":-81.11453247070313},"timestamp":"2023-08-09T15:53:51.392+0200","type":"INVENTORY"},{"data":
{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory overrun","antenna":1,"eventNum":50,"format":"epc","idHex":"111100000000000000000001","peakRssi":-
28,"phase":109.78423309326172},"timestamp":"2023-08-09T15:54:05.327+0200","type":"INVENTORY"},{"data":{"EPC":"a4dd3000111100000000000000000001","USER":"Error: tag returned error code 0x03 = Memory
```

Tag Data Interface1: hercules

Tag Data Interface2: IOTDemo

Management Events Interface: IOTDemo

Using APIs


RFID
New
Import
Collections
Environments
History
IoT Connector Local REST APIs
cloud
network
app-led
mode
GET Retrieves operating mode
PUT DOC-Updates identificat...
PUT Demo-prefix+Slice EPC
PUT Demo-Updates identific...
PUT Demo-Write Custom mo...
PUT Demo-CUSTOM
config
PUT Updates reader configur...
PUT DOC-Updates reader co...
PUT Updates reader configur...
PUT Updates reader configur...
PUT reducedHeartbeat
GET Retrieves reader configu...
logs
GET Retrieves reader compone...
GET Retrieves reader operation...
PUT Restarts reader
PUT Updates GPO port state
GET Retrieves reader region inf...
PUT Starts reading tags
PUT Stops reading tags
PUT Updates software on device
IoT Connector Local REST APIs / cloud / mode / Demo-prefix+Slice EPC
Params
Authorization
Headers (11)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Response

```
PUT {{baseUrl}}/cloud/mode
```

```
    "mode": "inventory",
    "antennas": [
        1,
        2,
        3,
        4
    ],
    "modeSpecificSettings":{
        "interval":{
            "unit":"seconds",
            "value":0
        }
    },
    "filter": {
        "value": "^00000000[0-9]{2,}",
        "match": "regex",
        "operation": "include"
    },

    "transmitPower": 27.0,
    "tagMetaData": [
        "EPC[1,3-5]",
        "ANTENNA",
        "RSSI"
    ]
}
```

Report Reads

Zebra DevCon 2023

- Tag read will be reported in what ever interface that is available to the Endpoint you selected
- You can send data to 2 Destinations at the same time
- Data format will be Json
- To customise the data message you can use Operation Modes for the data reported; and IOTConnect embedded/DA applications to adapt format for easier mapping/integration

Operating Modes

Zebra DevCon 2023

SIMPLE

The radio attempts to read tags on all antennas. This can be adjusted using the Antennas object when setting the mode.

The radio reports all unique tags once..

INVENTORY

The radio attempts to read tags on all antennas. The radio reports all unique tags once. The radio reports tags every defined interval.

PORTAL

The radio waits for a LOW signal on GPI The radio attempts to read tags on all antennas.

The radio reports all unique tags once. This can be adjusted using the Filter object when setting the mode.

CONVEYER

The radio attempts to read tags on all antennas.

The radio reports all unique tags once. This can be adjusted using the Filter object when setting the mode.

Custom Mode

Black Belt Level ;-)

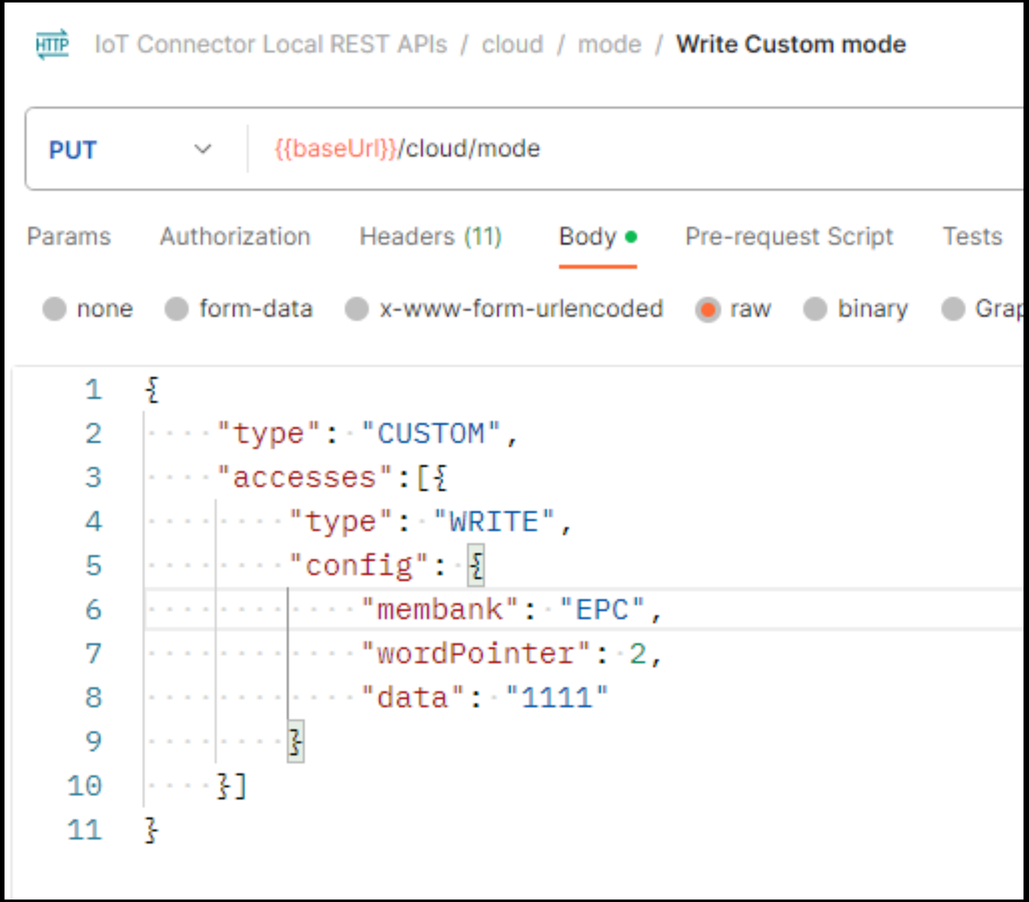
Zebra DevCon 2023

```
{
  "type": "CUSTOM",
  "antennas": [
    1,
    2,
    3,
    4
  ],
  "filter": {
    "value": "[a-zA-Z0-9]{2,}",
    "match": "regex",
    "operation": "include"
  },
  "environment": "AUTO_DETECT",
  "transmitPower": 27,
    "query": {
    "tagPopulation": 100,
    "sel": "SL",
    "session": "S3",
    "target": "B"
  },
  "tagMetaData": [
    "RSSI",
    "ANTENNA"
  ],
  "radioStartConditions": {
    "type": "GPI",
    "gpis": [
      {
        "port": 1,
        "signal": "HIGH",
        "debounceTime": 0
      }
    ]
  },
  "radioStopConditions": {
    "gpis": [
      {
        "port": 1,
        "signal": "HIGH",
        "debounceTime": 0
      }
    ]
  }
}
```

Writing data to Tags with IOTConnector

New in latest release – Brings IOT Connector to almost full coverage of SDK features

- Use Custom Mode on the fly to perform Write Operation to tag in the field


IoT Connector Local REST APIs / cloud / mode / **Write Custom mode**

PUT {{baseUrl}}/cloud/mode

```
{
    "type": "CUSTOM",
    "accesses":[{
        "type": "WRITE",
        "config": {
            "membank": "EPC",
            "wordPointer": 2,
            "data": "1111"
        }
    }]
}
```

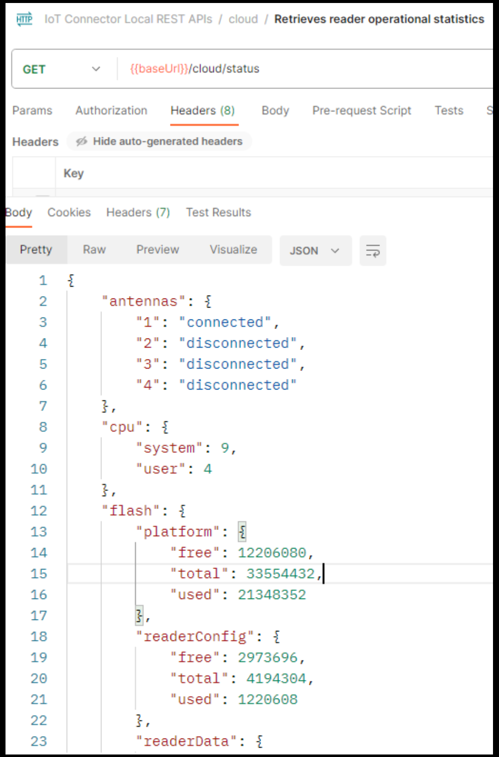
```
import http.client
import json

conn = http.client.HTTPSConnection("192.168.1.38")
payload = json.dumps({
  "type": "CUSTOM",
  "accesses": [
    {
      "type": "WRITE",
      "config": {
        "membank": "EPC",
        "wordPointer": 2,
        "data": "1111"
      }
    }
  ]
})
headers = {
  'Content-Type': 'application/json',
  'Accept': 'text/html',
  'Authorization': 'Bearer
~~eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJaRUJSQSIsImV4cCI6MTY5MTU3ODQ1M30.M~~
~~PEZAwHt2yRT3DKA3ghe4KxlkJDW-~~
~~AOwjWOV_CNA9At78Yn1TZl_jwDTnRtoyE7JXCHgX_Pryoh0pVYF0uwcRA'~~
}
conn.request("PUT", "/cloud/mode", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

```
In case of an error: {"code": 3, "message": "\"set_mode\" Failed to apply. Description: unsuccessful rc response: Failure: \"accesses\" must be either an array of objects or an array of array of objects"}
```

Monitor

- With REST API



IoT Connector Local REST APIs / cloud / **Retrieves reader operational statistics**

GET {{baseUrl}}/cloud/status

Params Authorization Headers (8) Body Pre-request Script Tests

Headers Hide auto-generated headers

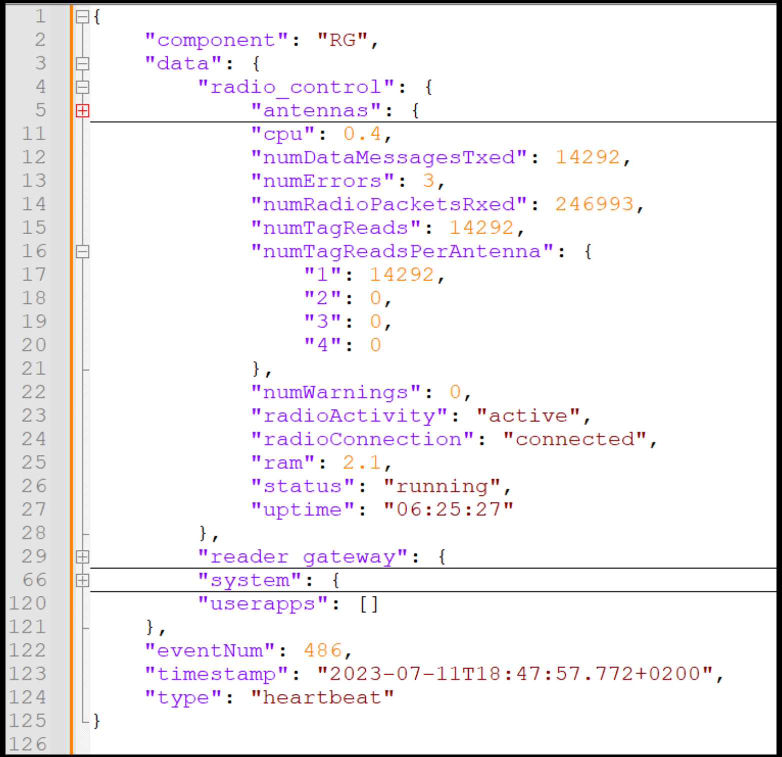
Key

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "antennas": {
        "1": "connected",
        "2": "disconnected",
        "3": "disconnected",
        "4": "disconnected"
    },
    "cpu": {
        "system": 9,
        "user": 4
    },
    "flash": {
        "platform": {
            "free": 12206080,
            "total": 33554432,
            "used": 21348352
        },
        "readerConfig": {
            "free": 2973696,
            "total": 4194304,
            "used": 1220608
        },
        "readerData": {
```

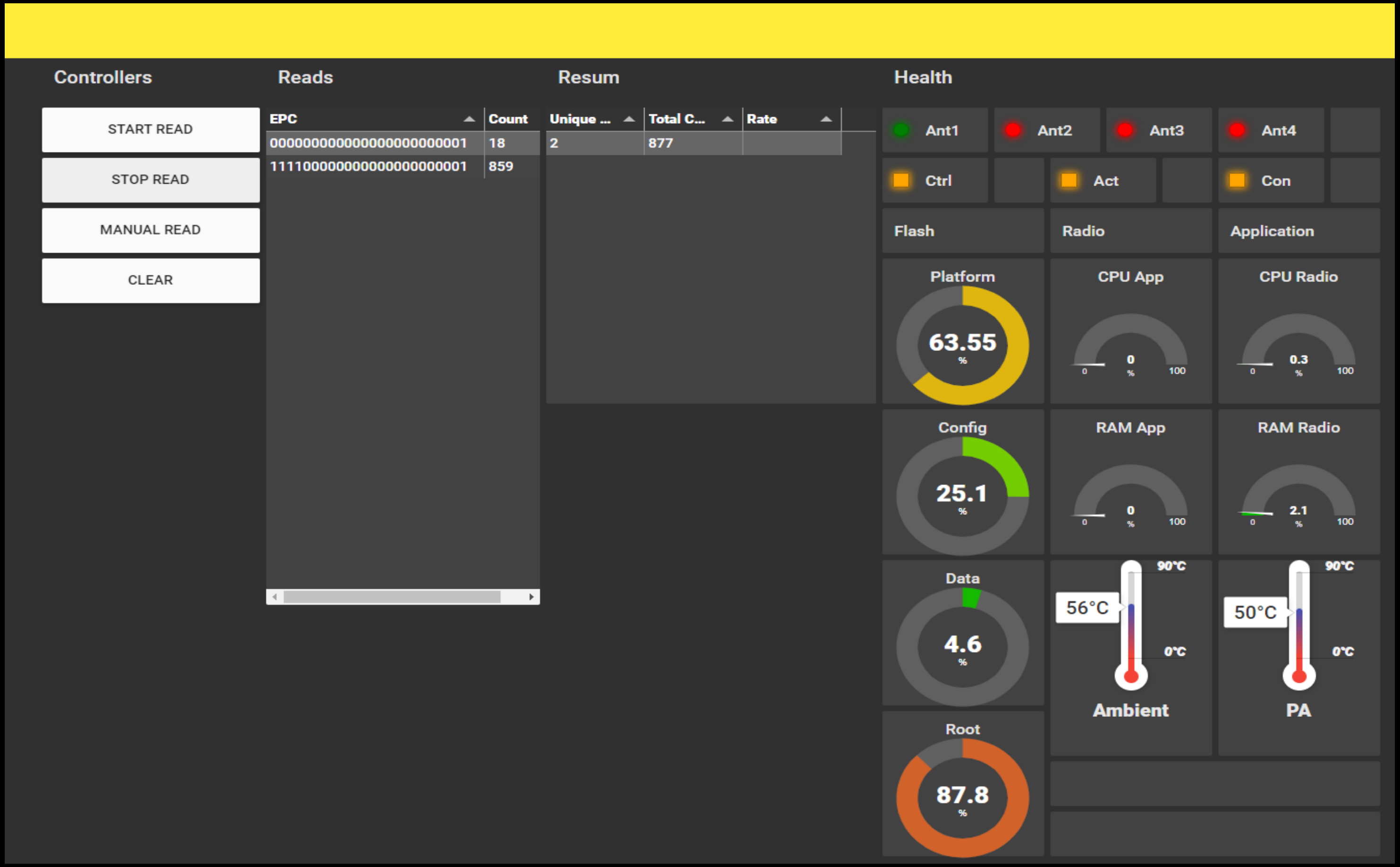
- More dynamic with MQTT using heartbeat with interval



```
{
    "component": "RG",
    "data": {
        "radio_control": {
            "antennas": {
            "cpu": 0.4,
            "numDataMessagesTxed": 14292,
            "numErrors": 3,
            "numRadioPacketsRxed": 246993,
            "numTagReads": 14292,
            "numTagReadsPerAntenna": {
                "1": 14292,
                "2": 0,
                "3": 0,
                "4": 0
            },
            "numWarnings": 0,
            "radioActivity": "active",
            "radioConnection": "connected",
            "ram": 2.1,
            "status": "running",
            "uptime": "06:25:27"
        },
        "reader_gateway": {
        "system": {
        "userapps": []
    },
    "eventNum": 486,
    "timestamp": "2023-07-11T18:47:57.772+0200",
    "type": "heartbeat"
}
```

Sample Dashboard with MQTT Health Messages

Zebra DevCon 2023



EPC	Count
000000000000000000000001	18
111100000000000000000001	859

Unique ...	Total C...	Rate
2	877	

Questions

Resources

Zebra DevCon 2023

- Documentation:
  - https://zebradevs.github.io/rfid-ziotc-docs/index.html
- FXWedge: Android APK to call IOTConnect methods from Android device: https://github.com/ltrudu/ZebraFXWedge/tree/master/ZebraFXWedge
- Windows Dockers Project that demonstrates IOTConnect MQTT and data storage in RDBMS:
- https://github.com/ZebraDevs/RFID-IoTConnector-MQTT-dotnet-DockDoorSample

FAQ

Zebra DevCon 2023

How is IoT Connector different from Cloud Connect? How is it different from FX Connect?	IoT Connector is the evolution of on-Device Cloud Connect. Just like Cloud Connect, it will support both the management and data for fixed readers. In addition, it will have a simple UI to configure reader operating modes, MQTT or HTTP endpoints. IoT Connector can also talk to 3rd party cloud IoT services (AWS IoT Core, GCP IoT Core, IBM Watson IoT Platform, Azure IoT Hub) IoT Connector will replace FXConnect and shall support all the FXConnect capabilities.
Is IoT Connector compatible with applications currently developed for Cloud Connect?	It is compatible but there may be some minor changes to configurations on endpoints if the instance is not already set up. If the applications are currently running Cloud Connect the REST APIs will be compatible. All endpoints configured in Cloud Connect will remain intact migrating to IoT Connector
Is IoT Connector licensable? What will happen to my FX Connect licenses?	IoT Connector is not licensable. If you have an FX Connect license it will cease to exist, but you will have the option to upgrade from FX Connect to IoT Connector w/o interrupting your workflow. No additional cost to the end user.
Is IoT Connector needed to connect FX reader with Zebra Data Services, MWE or any 3rd party cloud?	Yes. This capability is available inside the FX reader firmware. No need to install a separate on-reader app. To connect to ZDS or 3rd party cloud, you will need to enroll the reader to those services first. To connect to MWE, reader enrollment will happen automatically as part of device initialization step.

Zebra
DevCon 2023

ZEBRA

Thank You