

1. Kinesis Tech

**Sensor Reading use case with
Kinesis, ElasticCache and Real-time Visualization**

2. Replace part of it with AWS IOT

3. Difference between Kinesis Tech and AWS IOT Tech

TI SimpleLink™ Multi-Standard
SensorTag Development Kit
Featuring the CC2650 wireless MCU

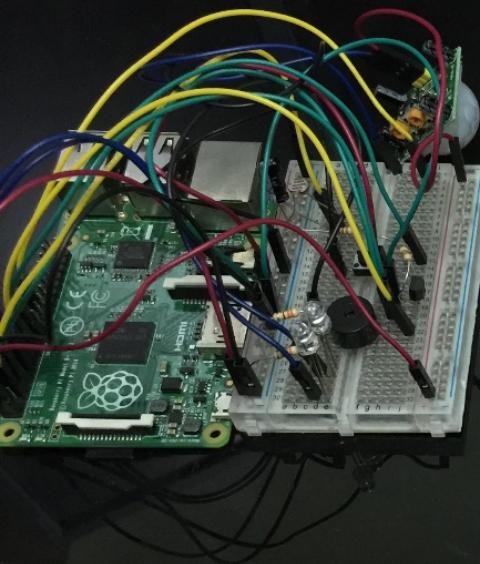
CC2650STK



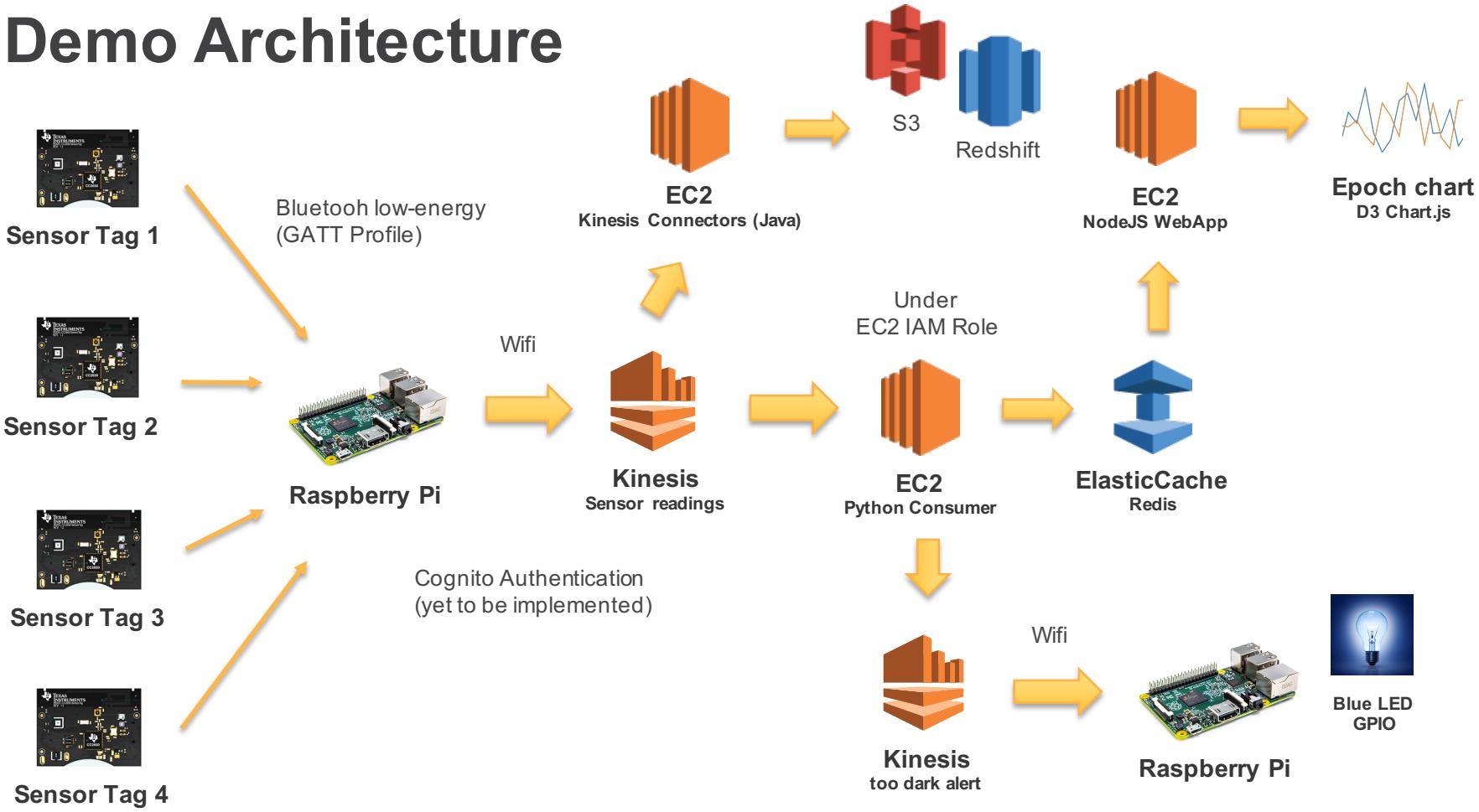
TEXAS INSTRUMENTS

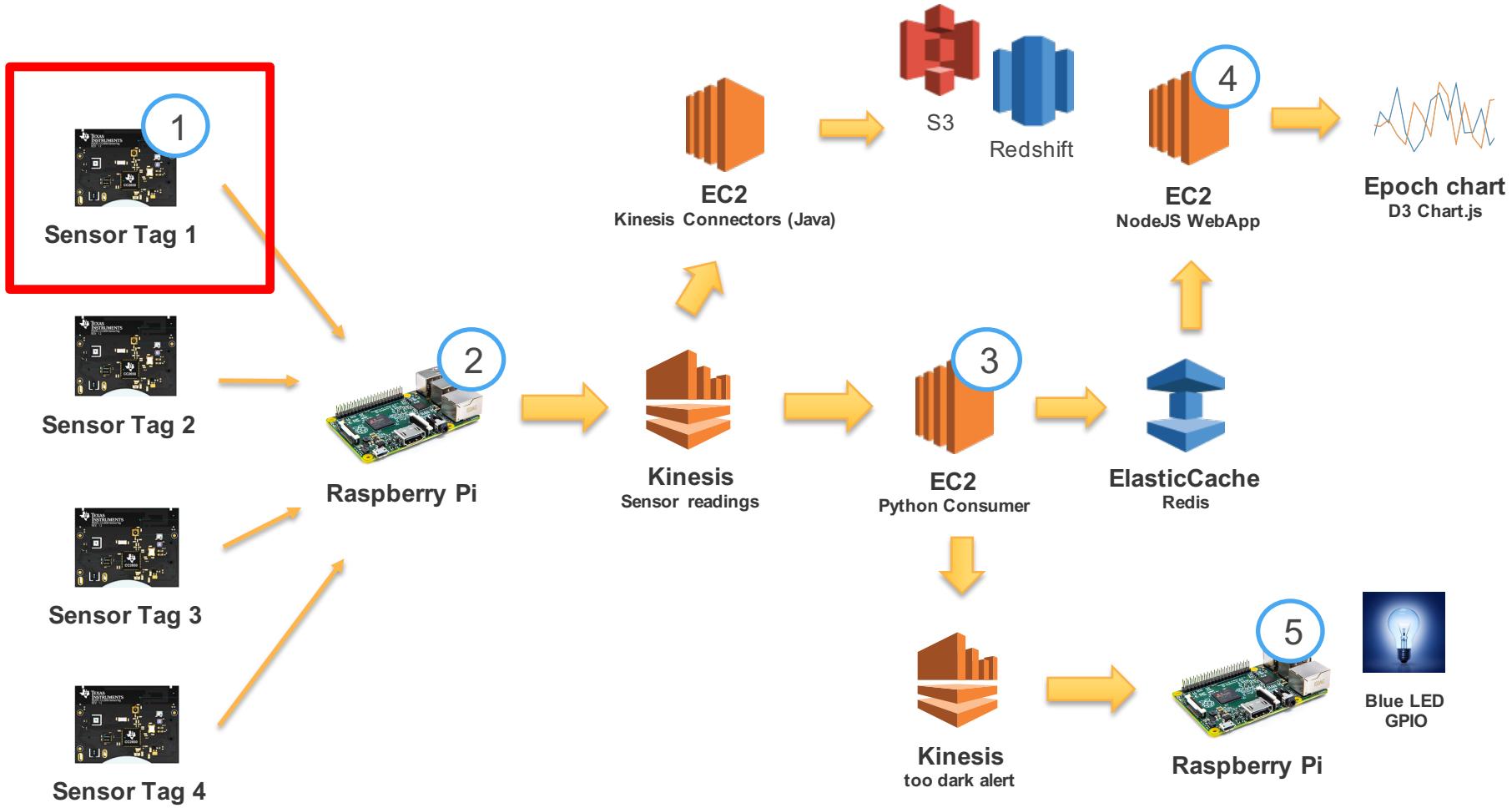
TI SimpleLink™ Multi-Standard
SensorTag Development Kit
Featuring the CC2650 wireless MCU

CC2650STK

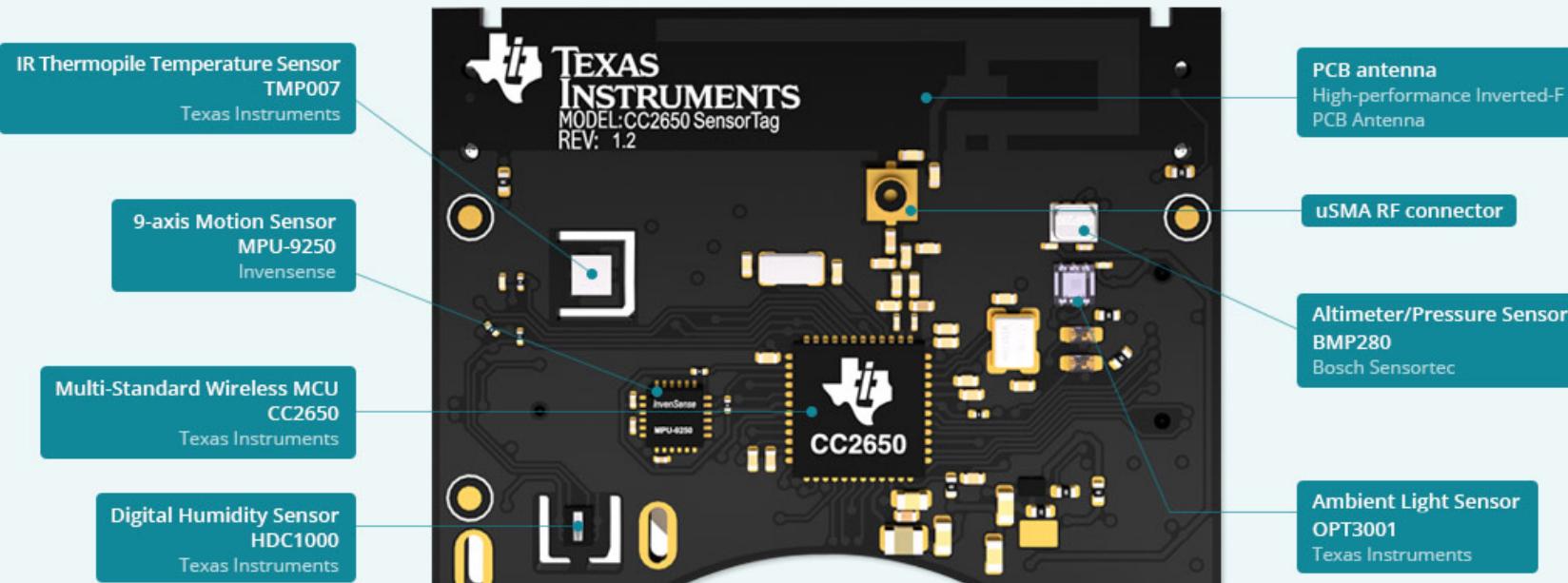


Demo Architecture

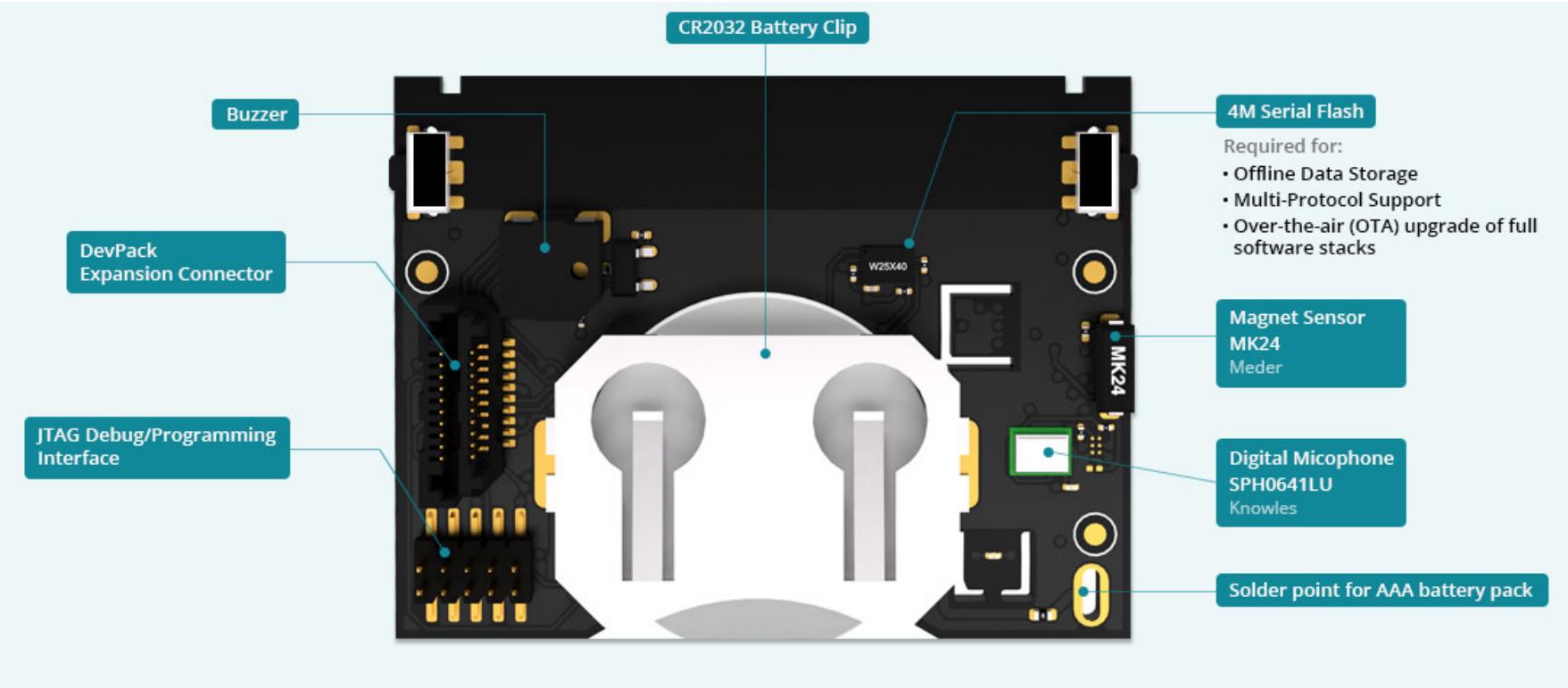




TI SensorTag CC2650



TI SensorTag CC2650





Movement

Accelerometer

X:-0.0G, Y: 0.0G, Z:-1.0G



Magnetometer

X:-99.54uT, Y:-35.08uT, Z:->



Gyroscope

X: 0.6°/S, Y: 0.1°/S, Z: 0.3°/S

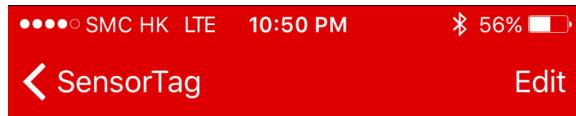


Light Sensor



Interval : 1.0s

TEXAS INSTRUMENTS



Ambient Temperature



28.4°C >

Interval : 1.0s

IR Temperature



23.1°C >

Interval : 1.0s

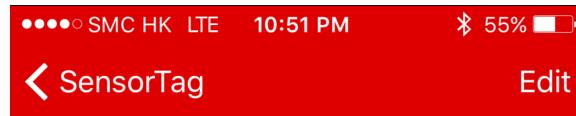
Humidity



114.9%rH >

Interval : 1.0s

TEXAS INSTRUMENTS



30.00ms



20.00
2000...
20.00

Slave latency

0

Automatic

Supervision Timeout

720.00ms

Device Information

System ID : f4:68:c9:0b:00:00:17:c7:0b

Model Number : CC2650 SensorTag

Serial Number : N.A.

FW rev. : 1.20 (Jul 28 2015)

HW rev. : PCB 1.2/1.3

SW rev. : N.A.

Manufacturer Name : Texas Instruments

IEEE 11073-20601 reg. : p

PnP ID :



Notify

RSSI

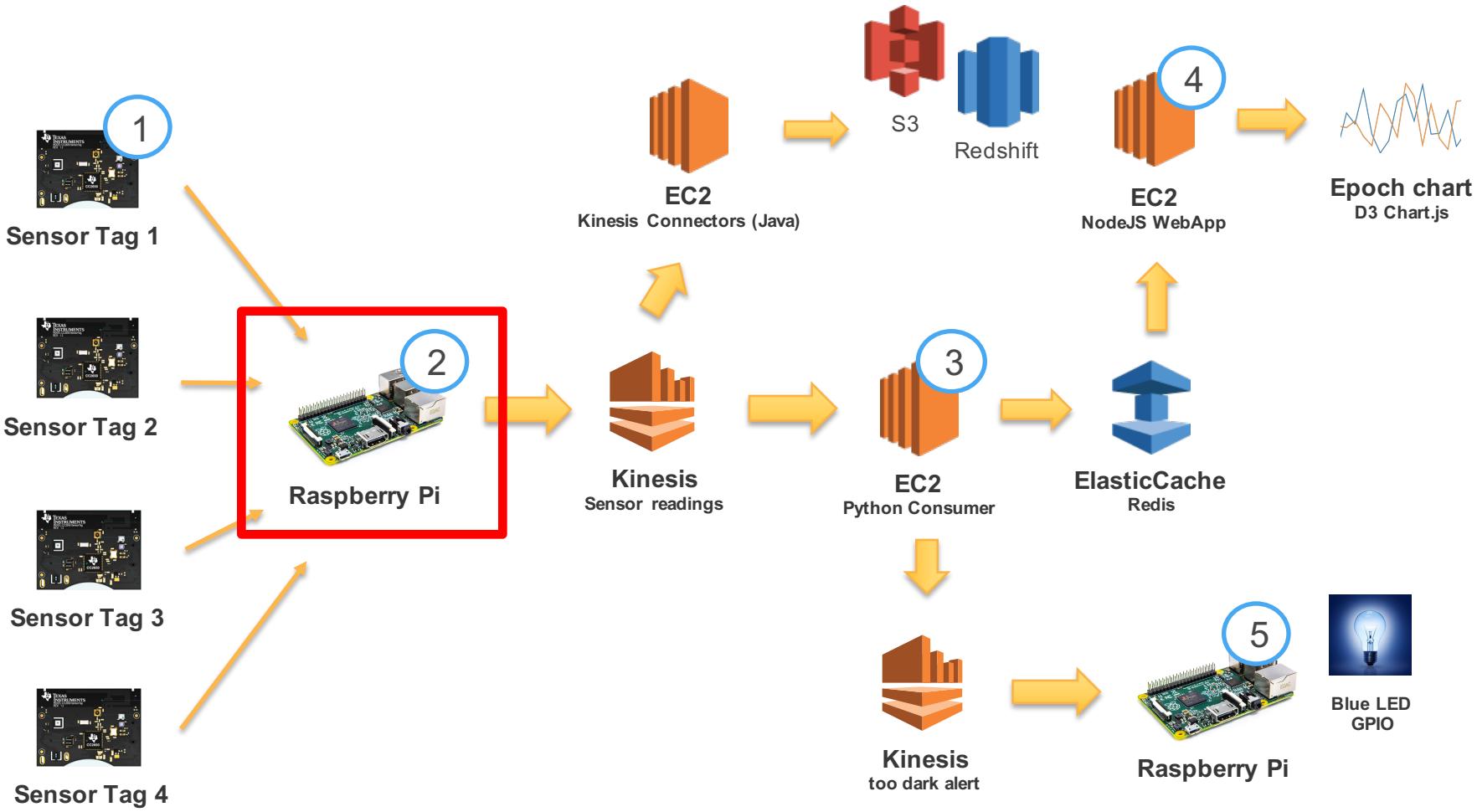


-71dBm

Interval 1.0s

TEXAS INSTRUMENTS

Demonstration





This repository Search

Pull requests Issues Gist



IanHarvey / bluepy

Watch 43

Star 209

Fork 85

Python interface to Bluetooth LE on Linux

101 commits

3 branches

4 releases

18 contributors



Branch: master

bluepy / +



IanHarvey Merge pull request #76 from louiscaron/master

Latest commit 717f2d3 8 days ago



Merge pull request #76 from louiscaron/master

8 days ago



testing/bluez-5.29: build fixes following merge

2 months ago



docs: update documentation to describe notifications

8 months ago



.gitignore: Copy uuids.json during setup; .gitignore various build artifacts

3 months ago



README.md: Update README with summary of recent changes

2 months ago



__init__.py: added __init__.py in order to turn bluepy into a python package

a year ago



setup.py: Copy uuids.json during setup; .gitignore various build artifacts

3 months ago



README.md

Code

Issues 19

Pull requests 0

Wiki

Pulse

Graphs

HTTPS clone URL

<https://github.com>

You can clone with [HTTPS](#),
[SSH](#), or [Subversion](#).

Clone in Desktop

Download ZIP

bluepy



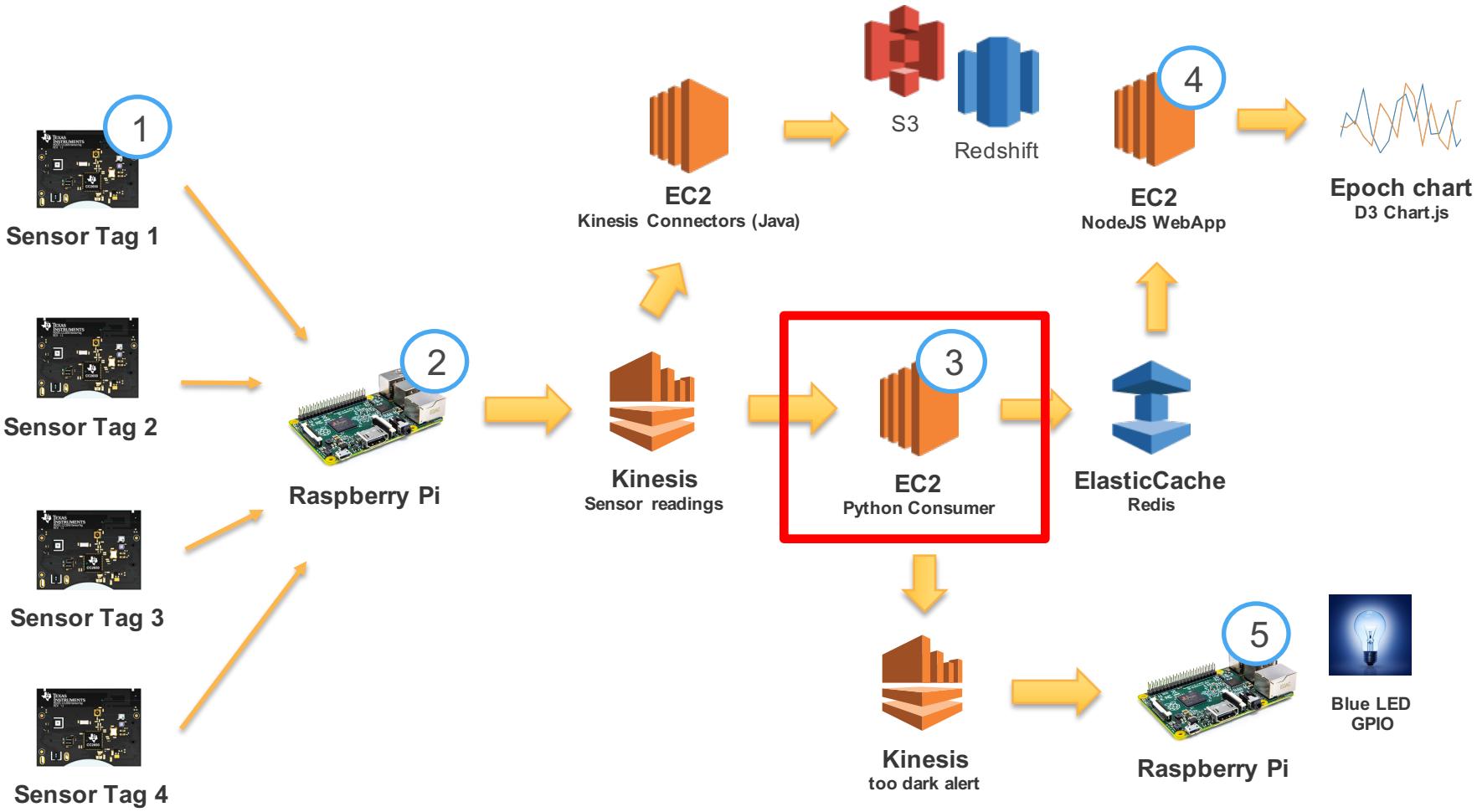
EXPLORE

- ▶ WORKING FILES
- ◀ RASPBERRY-SENSORTAG
 - ▲ kinesis-redis-worker
 - ▶ reference
 - long-worker.sh
 - worker.py
 - ▶ raspberry-alert-worker
 - ▲ raspberry-readings-worker
 - __init__.py
 - bluepy-helper
 - bluepy-helper.c
 - btle.py
 - btle.pyc
 - Makefile
 - push_kinesis_1.sh
 - push_kinesis_2.sh
 - push_kinesis_3.sh
 - push_kinesis_4.sh
 - push_kinesis_5.sh
 - push_kinesis_6.sh
 - push_kinesis_all.sh
 - sensortag.py
 - ▶ sensortag-visual
 - ▶ Solution_SensorTag

sensortag.py raspberry-readings-worker



```
250
251     arg = parser.parse_args(sys.argv[1:])
252
253     while True:
254         tag = None
255         try:
256             #boto.set_stream_logger('info')
257             kinesis = boto.kinesis.connect_to_region("ap-southeast-1")
258
259             print('Connecting to ' + arg.host)
260             tag = SensorTag(arg.host)
261
262             # Enabling selected sensors
263             tag.IRtemperature.enable()
264             tag.humidity.enable()
265             tag.accelerometer.enable()
266             tag.luxometer.enable()
267             tag.keypress.enable()
268             tag.setDelegate(KeypressDelegate())
269
270             # Some sensors (e.g., temperature, accelerometer) need some time for initialization.
271             # Not waiting here after enabling a sensor, the first read value might be empty or incor-
272             time.sleep(1)
273
274             counter=1
275             while True:
276                 data = record()
277                 data.DeviceID = arg.host
278                 data.DeviceName = tag.getDeviceName(arg.host)
279
280                 values = tag.IRtemperature.read()
281                 data.IRTemp, data.AmbientTemp = values[1], values[0]
282
283                 values = tag.humidity.read()
284                 data.Humidity = values[1]
285
```





EXPLORE

WORKING FILES

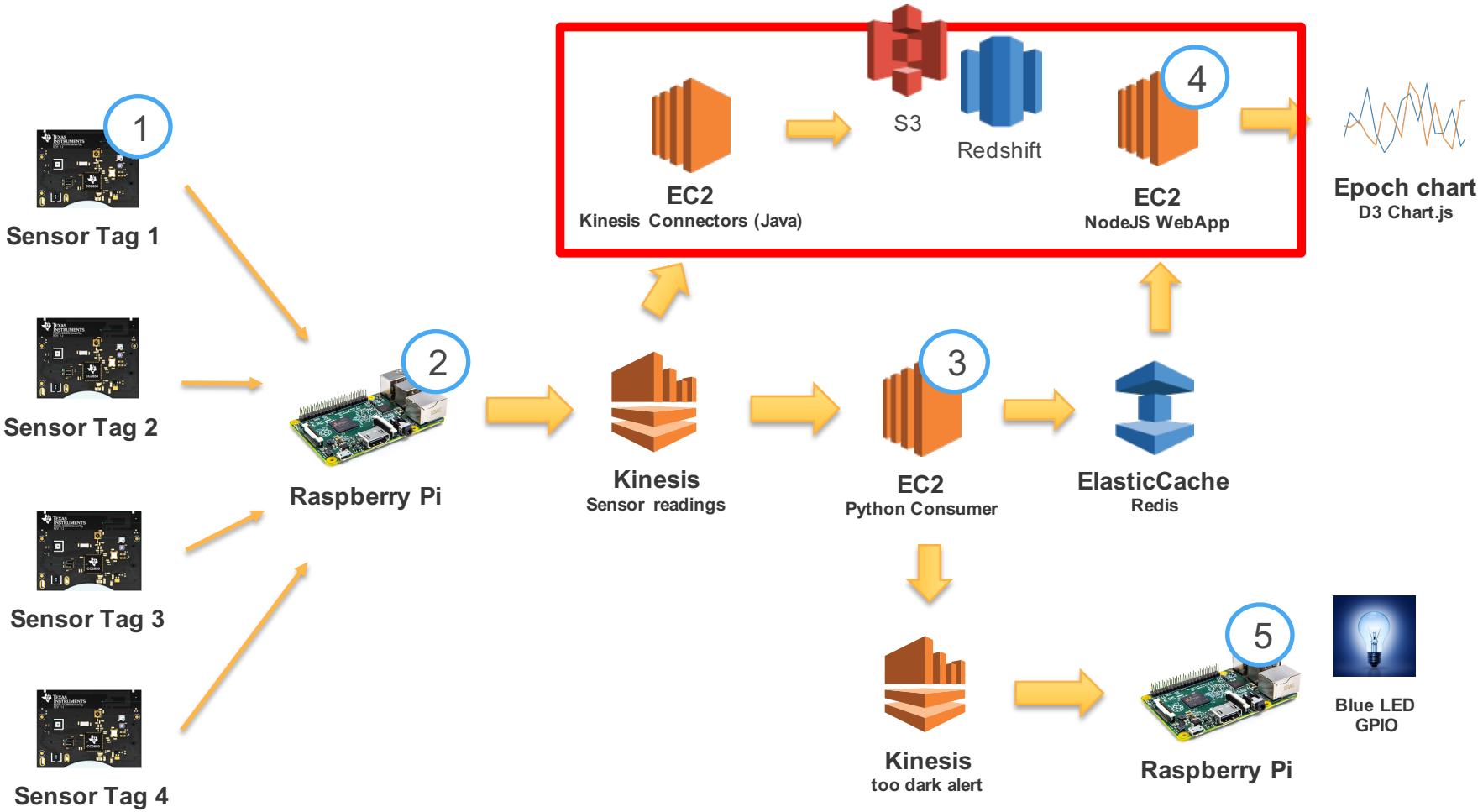
RASPBERRY-SENSORTAG

- kinesis-redis-worker
 - reference
 - long-worker.sh
 - worker.py
- raspberry-alert-worker
- raspberry-readings-worker
- sensortag-visual
- Solution_SensorTag

worker.py kinesis-redis-worker

```
43 #!/usr/bin/python -u
44 r = redis.StrictRedis(host='replicationgroup.nx1t0s.ng.0001.apse1.cache.amazonaws.com', port=6379, db=0)
45
46 class DataRecord:
47     def to_JSON(self):
48         return json.dumps(self, default=lambda o: o.__dict__,
49                           sort_keys=True, indent=4)
50
51 def process_to_redis(records):
52     for record in records:
53         jsonstring = record['Data'].lower()
54         json_data = json.loads(jsonstring)
55
56         newdata = {"time":json_data["time"], "devicename":json_data["devicename"], "irtemp":json_data["irtemp"]}
57         newdatastring = json.dumps(newdata)
58
59         print (newdatastring)
60         r.publish('pubsubCounters', newdatastring)
61
62         if json_data["lux"] < 5:
63             kinesis.put_record(
64                 stream_name="LightAlert",
65                 data=newdatastring, partition_key=json_data["deviceid"])
66             print ('Lights up')
67
68         if json_data["irtemp"] > 29:
69             kinesis.put_record(
70                 stream_name="TemperatureAlert",
71                 data=newdatastring, partition_key=json_data["deviceid"])
72             print ('Overheat')
73
74
75 class KinesisWorker(threading.Thread):
76     """The Worker thread that repeatedly gets records from a given Kinesis
77     stream."""
78     def __init__(self, stream_name, shard_id, iterator_type, sleep_interval=0.5,
```

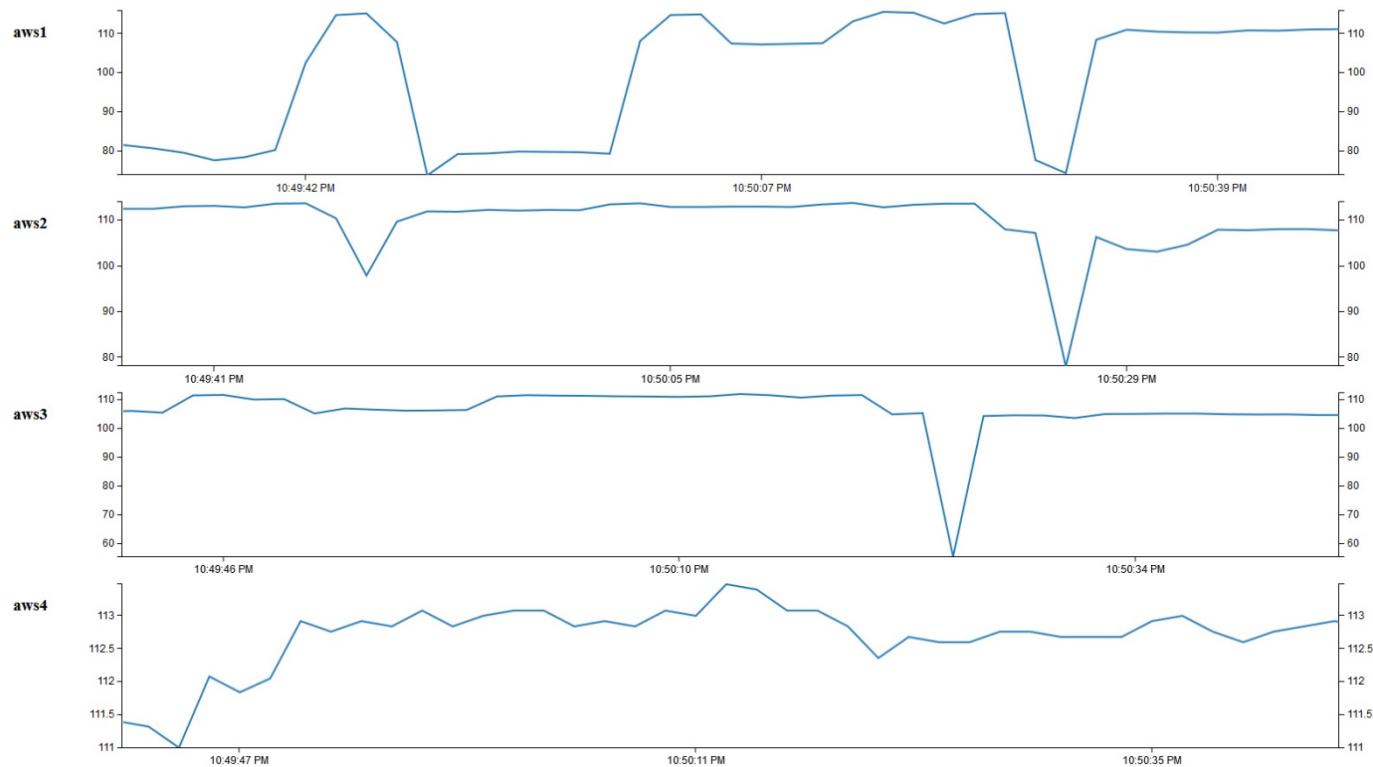
Ln 60, Col 1 (50 selected) UTF-8 LF Python ☺





Hong Kong

Kinesis - Light Sensor





EXPLORE

- ▶ WORKING FILES
- ◀ RASPBERRY-SENSORTAG
 - bluepy-helper.c
 - btle.py
 - btle.pyc
 - Makefile
 - push_kinesis_1.sh
 - push_kinesis_2.sh
 - push_kinesis_3.sh
 - push_kinesis_4.sh
 - push_kinesis_5.sh
 - push_kinesis_6.sh
 - push_kinesis_all.sh
 - sensortag.py
- ◀ sensortag-visual
 - AWS_Booth_App_Header...
 - AWS_Booth_Kinesis_Ligh...
 - AWS_Booth_Kinesis_Tem...
 - epoch.min.css
 - epoch.min.js
 - index.html
 - index2.html
 - my-colors.css
 - my-colors.scss
 - node-app.js
 - sensortag-visual.sh
- ▷ Solution_SensorTag

node-app.js sensortag-visual

```
15
16 var http    = require('http')
17 , connect = require('connect')
18 , redis = require('redis')
19 , serveStatic = require('serve-static');
20
21 function ticker(req,res) {
22   req.socket.setTimeout(Infinity);
23
24   var subscriber = redis.createClient(6379,process.argv[2]);
25
26   subscriber.subscribe("pubsubCounters");
27
28   // When we receive a message from the redis connection
29   subscriber.on("message", function(channel, message) {
30     res.json(message);
31   });
32
33   //send headers for event-stream connection
34   res.writeHead(200, {
35     'Content-Type': 'text/event-stream',
36     'Cache-Control': 'no-cache',
37     'Connection': 'keep-alive'
38   });
39
40   res.json = function(obj) { res.write("data: "+obj+"\n\n"); }
41   res.json(JSON.stringify({}));
42
43   // The 'close' event is fired when a user closes their browser window.
44   req.on("close", function() {
45     subscriber.unsubscribe();
46     subscriber.quit();
47   });
48 }
49
50 connect()
```





EXPLORE

▶ WORKING FILES
 ◀ RASPBERRY-SENSORTAG
 bluepy-helper.c
 btle.py
 btle.pyc
 Makefile
 push_kinesis_1.sh
 push_kinesis_2.sh
 push_kinesis_3.sh
 push_kinesis_4.sh
 push_kinesis_5.sh
 push_kinesis_6.sh
 push_kinesis_all.sh
 sensortag.py
 ◀ sensortag-visual
 AWS_Booth_App_Header...
 AWS_Booth_Kinesis_Ligh...
 AWS_Booth_Kinesis_Tem...
 epoch.min.css
 epoch.min.js
 index.html
 index2.html
 my-colors.css
 my-colors.scss
 node-app.js
 sensortag-visual.sh
 ▶ Solution_SensorTag

index2.html sensortag-visual

```
32  var i = 0, i ~ arrayLength, i++ ) {  
33      var containerDiv = document.createElement('div');  
34      containerDiv.style.width = '100%';  
35      containerDiv.style.height = '21%';  
36  
37      var labelDiv = document.createElement('div');  
38      labelDiv.style.width = '7%';  
39      labelDiv.style.height = '100%';  
40      labelDiv.style.position = 'relative';  
41      labelDiv.style.float = 'left';  
42      labelDiv.style.textAlign = 'center';  
43      labelDiv.innerHTML = "<h3>" + siteArray[i] + "</h3>";  
44  
45      var chartDiv = document.createElement('div');  
46      chartDiv.id = siteArray[i] + 'Chart';  
47      chartDiv.className = "epoch category20 ";  
48      chartDiv.style.width = '93%';  
49      chartDiv.style.height = '100%';  
50      chartDiv.style.position = 'relative';  
51      chartDiv.style.float = 'left';  
52  
53      containerDiv.appendChild(labelDiv);  
54      containerDiv.appendChild(chartDiv);  
55      document.getElementById('mainDiv').appendChild(containerDiv);  
56  
57      eval("var " + siteArray[i] + "Data = [{label:'" + siteArray[i] + "\",values:[{time:cur  
58      eval("var " + siteArray[i] + " = $('#" + siteArray[i] + "Chart').epoch({type:'time.line  
59    }  
60  
61    function tick(e) {  
62      if (e) {  
63        var eventData = JSON.parse(e.data);  
64        window[eventData.devicename].push([ { time: ((new Date).getTime() / 1000) - 10, y: ev  
65      }  
66    }  
67  }
```



Amazon Kinesis Connector Library

The **Amazon Kinesis Connector Library** helps Java developers integrate **Amazon Kinesis** with other AWS and non-AWS services. The current version of the library provides connectors for **Amazon DynamoDB**, **Amazon Redshift**, **Amazon S3**, **Elasticsearch**. The library also includes [sample connectors](#) of each type, plus Apache Ant build files for running the samples.

☞ Requirements

- **Amazon Kinesis Client Library:** In order to use the Amazon Kinesis Connector Library, you'll also need the [Amazon Kinesis Client Library](#).
- **Java 1.7:** The Amazon Kinesis Client Library requires [Java 1.7 \(Java SE 7\)](#) or later.
- **Elasticsearch 1.2.1:** The Elasticsearch connector depends on [Elasticsearch 1.2.1](#).
- **SQL driver** (Amazon Redshift only): If you're using an Amazon Redshift connector, you'll need a driver that will allow your SQL client to connect to your Amazon Redshift cluster. For more information, see [Download the Client Tools and the Drivers](#) in the Amazon Redshift Getting Started Guide.

Overview

Each Amazon Kinesis connector application is a pipeline that determines how records from an Amazon Kinesis stream will be handled. Records are retrieved from the stream, transformed according to a user-defined data model, buffered for batch processing, and then emitted to the appropriate AWS service.

[Upload](#) [Create Folder](#) [Actions ▾](#)

[None](#) [Properties](#) [Transfers](#)

All Buckets / [sensortag-redshift-bucket](#)

	Name	Storage Class	Size	Last Modified
	49554520425067593469152210341886919313251724122254213122-495545204250675934691522103474878726...	Standard	859.9 KB	Wed Nov 04 00:09:18 GMT+800
	49554520425089894214350746683652571958291268861584998418-495545204250898942143507466954311362...	Standard	1.7 MB	Wed Nov 04 00:09:18 GMT+800
	49554520425089894214350746683652571958291268861584998418-495545204250898942143507466954637772...	Standard	1.7 MB	Wed Nov 04 00:16:07 GMT+800
	49554520425089894214350746683652571958291268861584998418-495545204250898942143507466957394123...	Standard	1.7 MB	Wed Nov 04 00:05:40 GMT+800



SensorTag_RedshiftQuery.sql 1

```

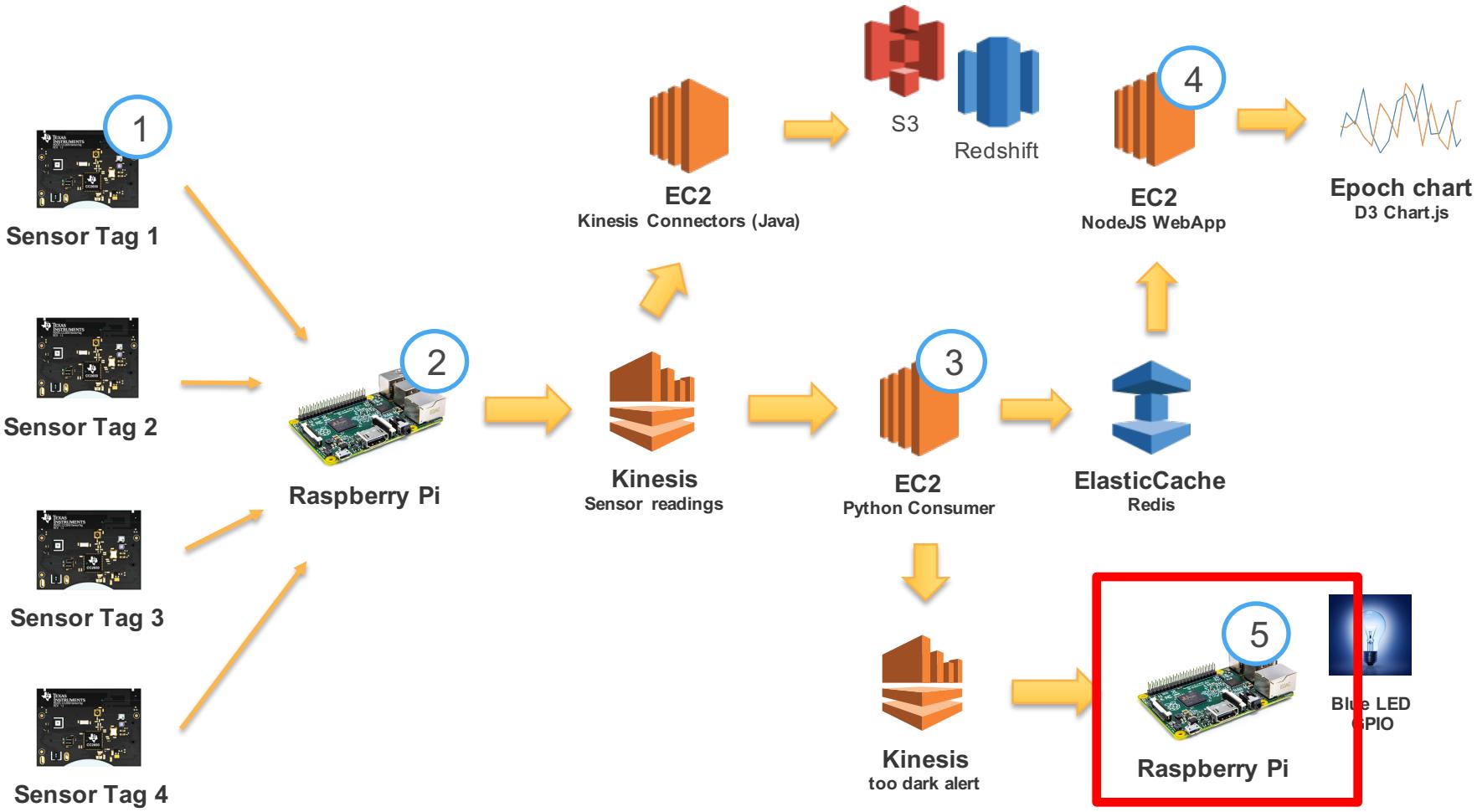
1 SELECT * From kinesisbasictable
2 ORDER BY Time DESC
3

```

Result 1

Messages

deviceid	devicename	ambienttemp	gyrox	gyroy	gyroz	humidity	irtemp	lux	magx
68:C9:0B:06:44:09	aws1	25.75	-0.614776611328125	0.054473876953125	1.657562255859375	118.99237060546875	21.9375	64.41	-3.897461
C4:BE:84:70:91:0E	aws4	25.65625	-0.287933349609375	0.677032470703125	0.66925048828125	86.1630859375	22.53125	71.35	-46.16991
C4:BE:84:71:32:0A	aws3	26.125	0.443572998046875	1.906585693359375	0.38909912109375	86.3309326171875	21.9375	63.41	-25.48331
68:C9:0B:06:44:09	aws1	25.75	-0.8404541015625	-0.070037841796875	1.54083251953125	118.99237060546875	21.84375	64.41	-0.149901
C4:BE:84:70:91:0E	aws4	25.65625	0.1556396484375	0.739288330078125	0.8404541015625	86.1630859375	22.4375	71.19	-48.41841
68:C9:0B:06:44:09	aws1	25.75	-0.630340576171875	0.007781982421875	1.929931640625	118.99237060546875	21.90625	64.32	-1.199211
C4:BE:84:71:32:0A	aws3	26.125	0.54473876953125	2.186737060546875	0.26458740234375	86.3309326171875	21.78125	63.37	-25.33341
C4:BE:84:70:91:0E	aws4	25.65625	-0.490264892578125	0.350189208984375	0.801544189453125	86.1630859375	22.3125	71.27	-47.06931
68:C9:0B:06:44:09	aws1	25.75	-0.98052978515625	0.023345947265625	1.828765869140625	118.99237060546875	21.875	64.28	-0.599601
C4:BE:84:71:32:0A	aws3	26.125	0.474700927734375	1.78985595703125	0.2490234375	86.3309326171875	21.875	63.33	-24.73381
C4:BE:84:70:91:0E	aws4	25.65625	0.062255859375	0.7159423828125	0.73150634765625	86.1630859375	22.375	71.11	-46.91941
C4:BE:84:71:32:0A	aws3	26.125	0.505828857421875	1.906585693359375	0.1556396484375	86.3309326171875	22.0	63.28	-24.43401
68:C9:0B:06:44:09	aws1	25.75	-0.832672119140625	0.10894775390625	2.03887939453125	118.99237060546875	21.875	64.13	-5.396481
C4:BE:84:70:91:0E	aws4	25.65625	0.038909912109375	0.7159423828125	0.87158203125	86.1630859375	22.4375	71.11	-46.76951
68:C9:0B:06:44:09	aws1	25.75	-0.770416259765625	0.163421630859375	1.922149658203125	118.99237060546875	21.8125	64.25	-0.899411
C4:BE:84:71:32:0A	aws3	26.125	0.412445068359375	1.61865234375	0.12451171875	86.3309326171875	21.875	63.29	-25.63331
C4:BE:84:70:D1:00	aws2	25.75	-2.544708251953125	0.988311767578125	1.081695556640625	85.72821044921875	22.34375	64.83	-35.82661
C4:BE:84:71:32:0A	aws3	26.125	0.350189208984375	2.217864990234375	0.07781982421875	86.3309326171875	22.03125	63.29	-24.88371
C4:BE:84:70:91:0E	aws4	25.65625	-0.350189208984375	0.4046630859375	0.88714599609375	86.1630859375	22.40625	71.35	-47.21921
68:C9:0B:06:44:09	aws1	25.75	-0.87158203125	-0.023345947265625	1.688690185546875	118.99237060546875	21.8125	64.41	-1.34912
C4:BE:84:70:D1:00	aws2	25.75	-2.248992919921875	1.19842529296875	1.07391357421875	85.72821044921875	22.09375	64.87	-35.97651
68:C9:0B:06:44:09	aws1	25.75	-0.7159423828125	-0.10894775390625	1.891021728515625	118.99237060546875	21.90625	64.48	-1.798821





EXPLORE

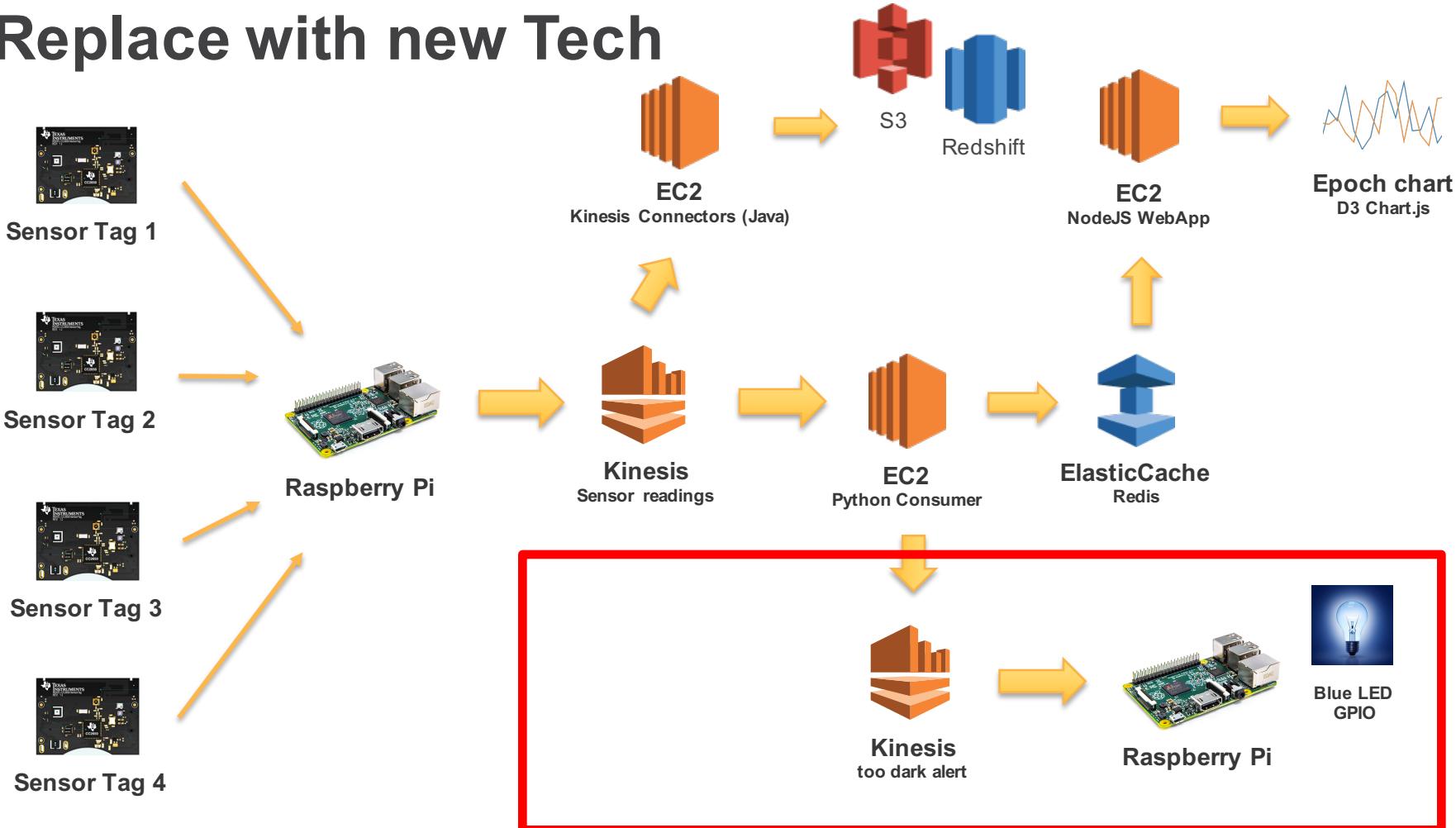
► WORKING FILES
◄ RASPBERRY-SENSORTAG
 ► kinesis-redis-worker
 ◄ raspberry-alert-worker
 ► reference
 long-trigger-all.sh
 long-trigger-light.sh
 long-trigger-temperature.sh
 trigger-light.py
 trigger-temperature.py
 ► raspberry-readings-worker
 ► sensortag-visual
 ► Solution_SensorTag

trigger-light.py raspberry-alert-worker

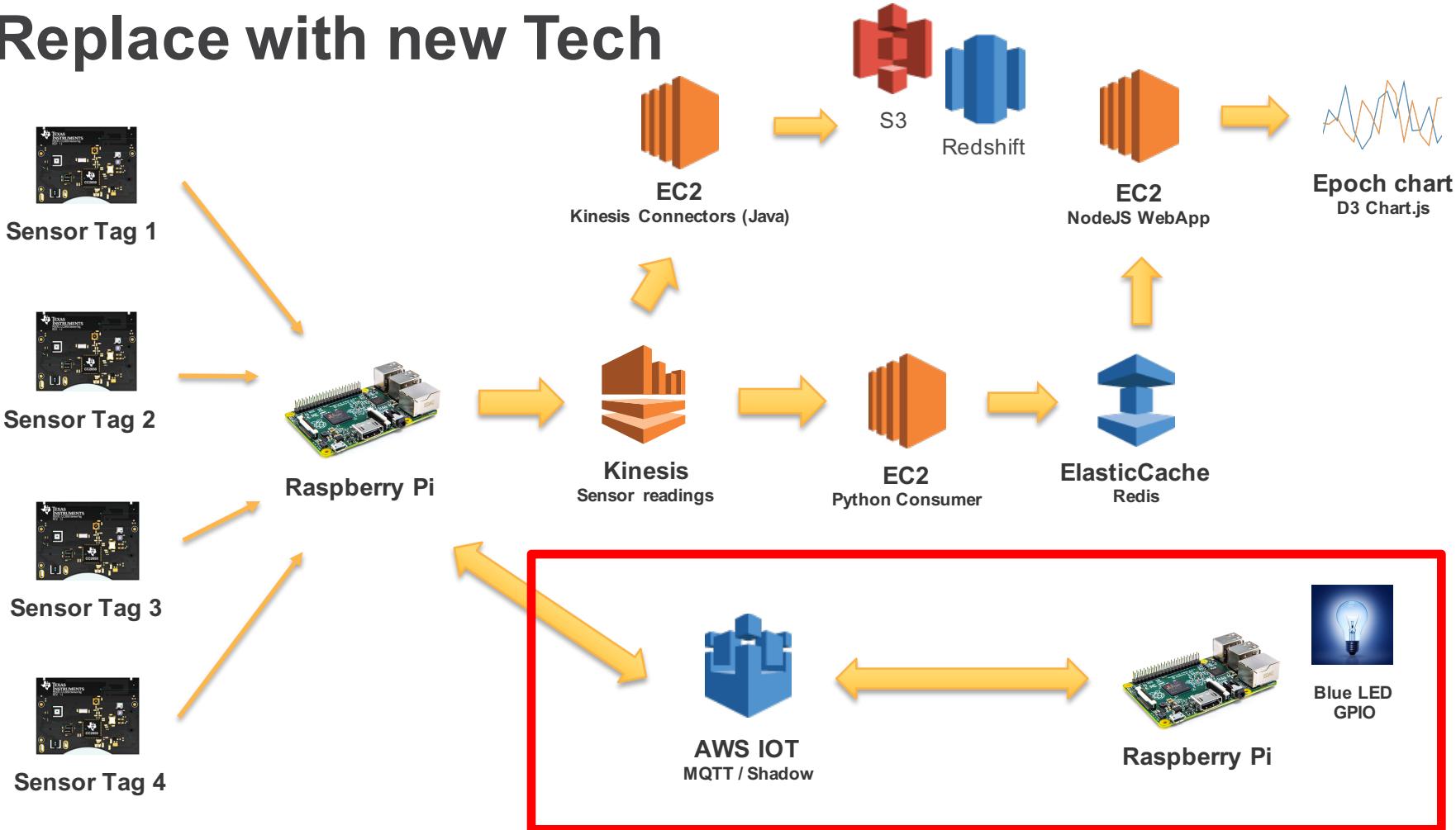
```
/o
71     counter = 0
72     isLightOn = False
73     while True:
74         try:
75             response = kinesis.get_records(next_iterator, limit=25)
76             self.total_records += len(response['Records'])
77             counter = counter + 1
78             print ("Counter:", str(counter) + " and " + str(isLightOn))
79
80             if len(response['Records']) > 0:
81                 print ('\n++> {1} Got {0} Worker Records'.format(len(response['Records']), my_na
82                 process_to_generate_light(response['Records']))
83
84                 if isLightOn is False:
85                     print ("Lights up")
86                     GPIO.setup(17,GPIO.OUT)
87                     GPIO.output(17,GPIO.HIGH)
88                     isLightOn = True
89
90                 counter = 0
91             else:
92                 if counter >= 3 and isLightOn is True:
93                     print ("Lights off")
94                     GPIO.setup(17,GPIO.OUT)
95                     GPIO.output(17,GPIO.LOW)
96                     isLightOn = False
97
98                 next_iterator = response['NextShardIterator']
99                 time.sleep(0.5)
100
101             except ProvisionedThroughputExceededException as ptee:
102                 print (ptee.message)
103                 time.sleep(5)
104
105 if __name__ == '__main__':
```



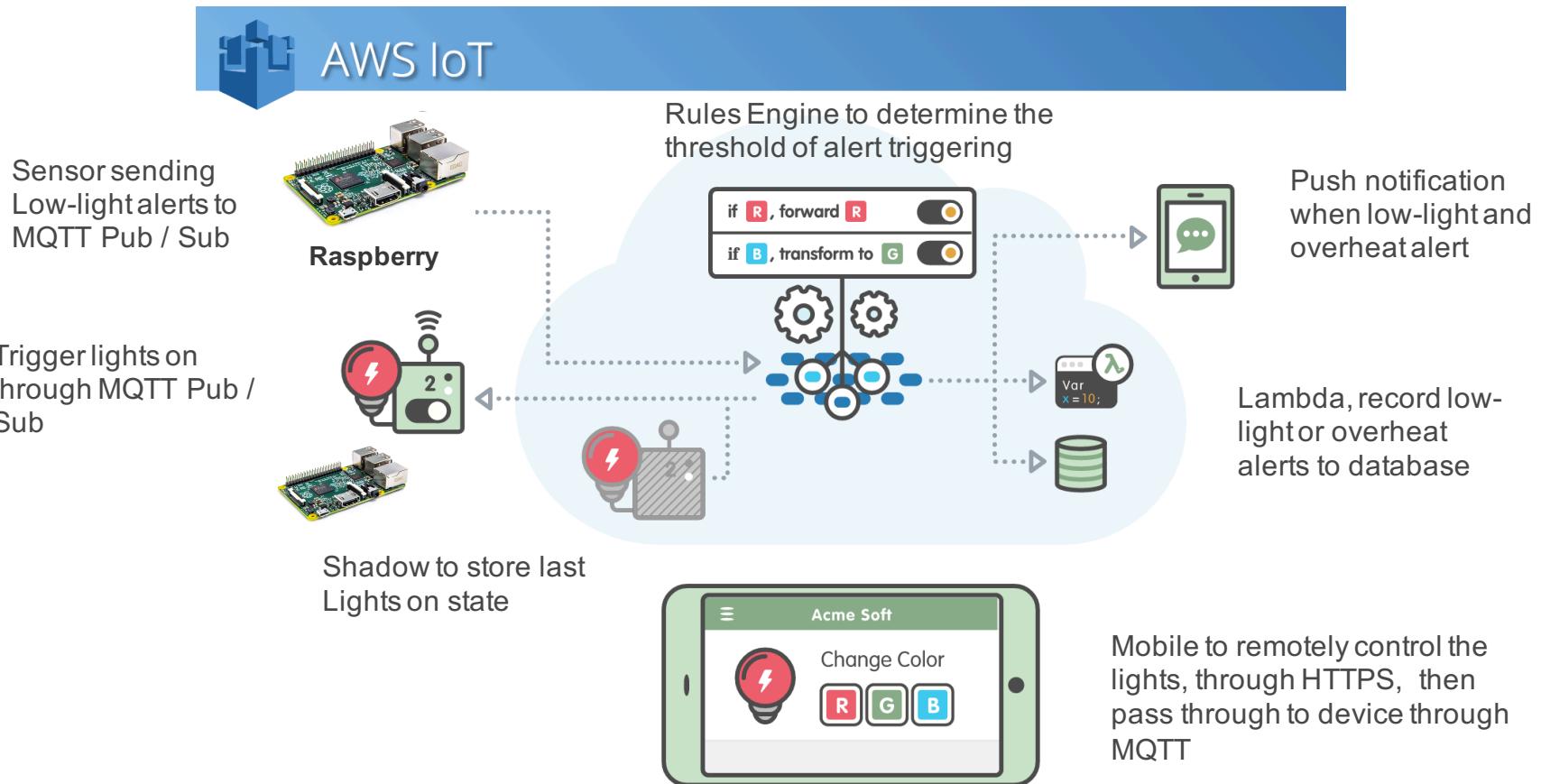
Replace with new Tech



Replace with new Tech

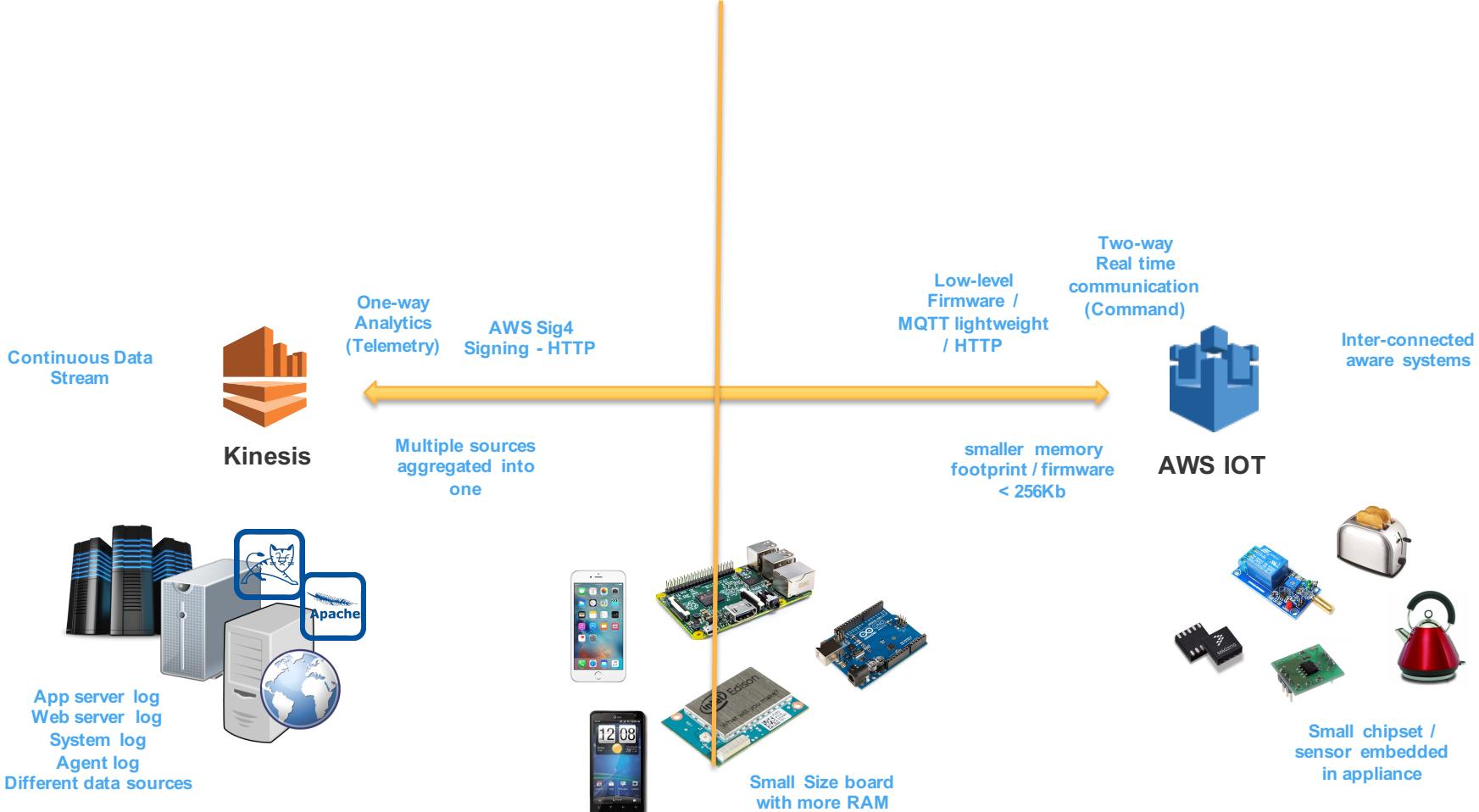


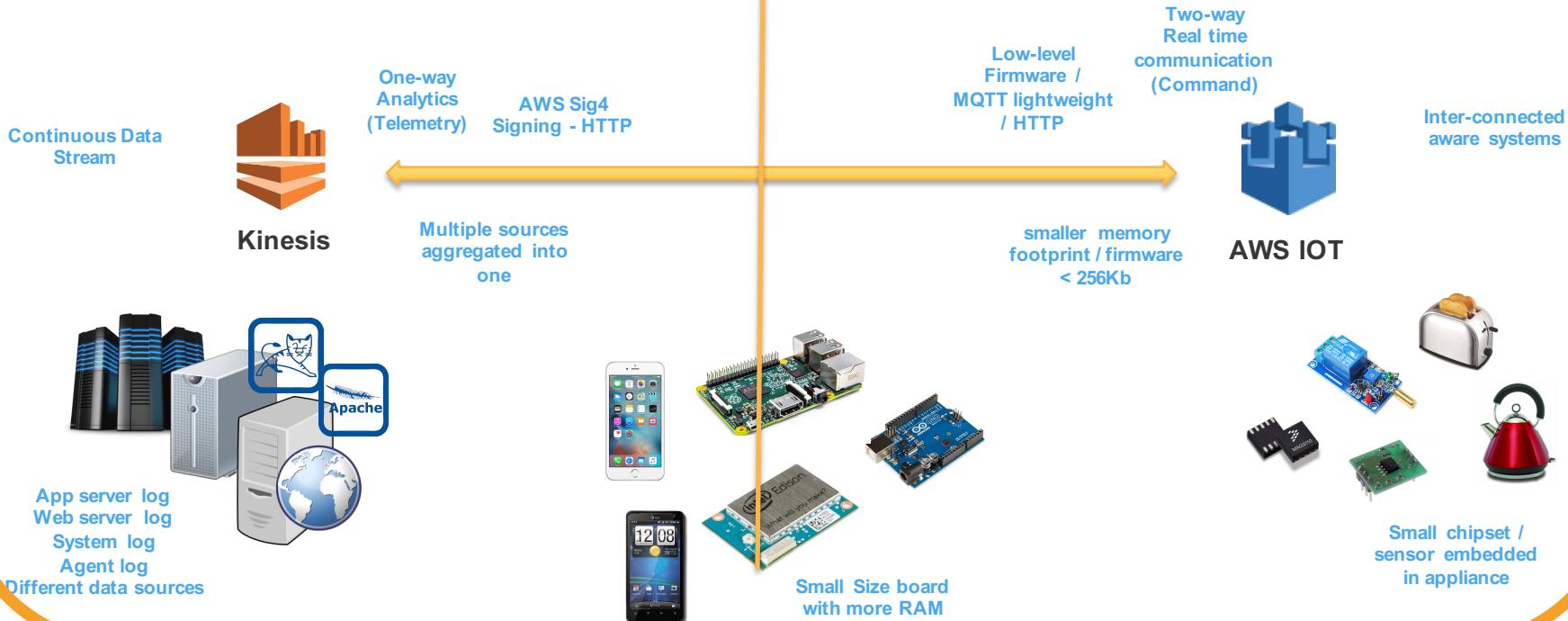
Potential to replace with new Tech



Quick Difference

Different use case with Kinesis and AWS IOT





Kinesis with AWS IOT

