



Help, I Have Too Many Raspberry Pis!

Tim Spann

NiFi Expert, Easily Distracted by New TensorFlow Models

21 March 2019 Dataworks Summit Barcelona





The First Mile – Edge and IoT Data Collection with Apache NiFi and MiNiFi

Andy LoPresto | @yolopey

DIM Security Engineering Manager at Cloudera, Apache NiFi PMC & Committer

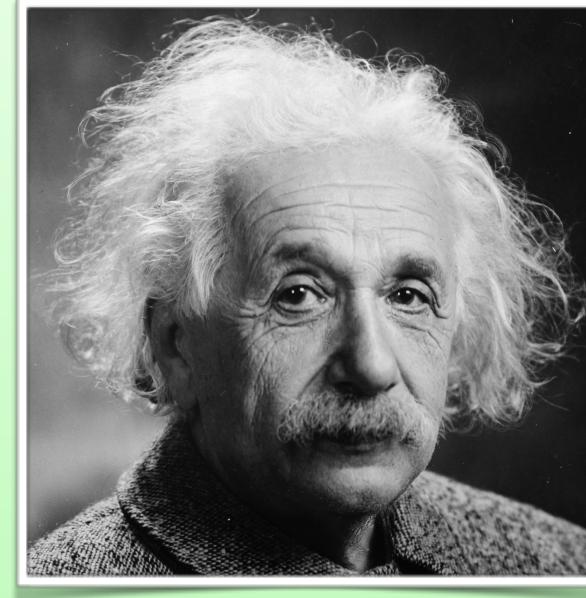
21 March 2019 Dataworks Summit Barcelona

Gauging Audience Familiarity With NiFi



“What’s a NeeFee?”

No experience with dataflow
No experience with NiFi



“I can pick this up pretty quickly”

Some experience with dataflow
Some experience with NiFi



“I refactored the Ambari integration endpoint to allow for mutual authentication TLS during my coffee break”

Forgotten more about NiFi
than most of us will ever
know

Agenda

- *What is dataflow and what are the challenges?*
- *Apache NiFi*
- IoT Challenges
- Apache MiNiFi
- Exploration
- Community
- *Roadmap*
- Q&A



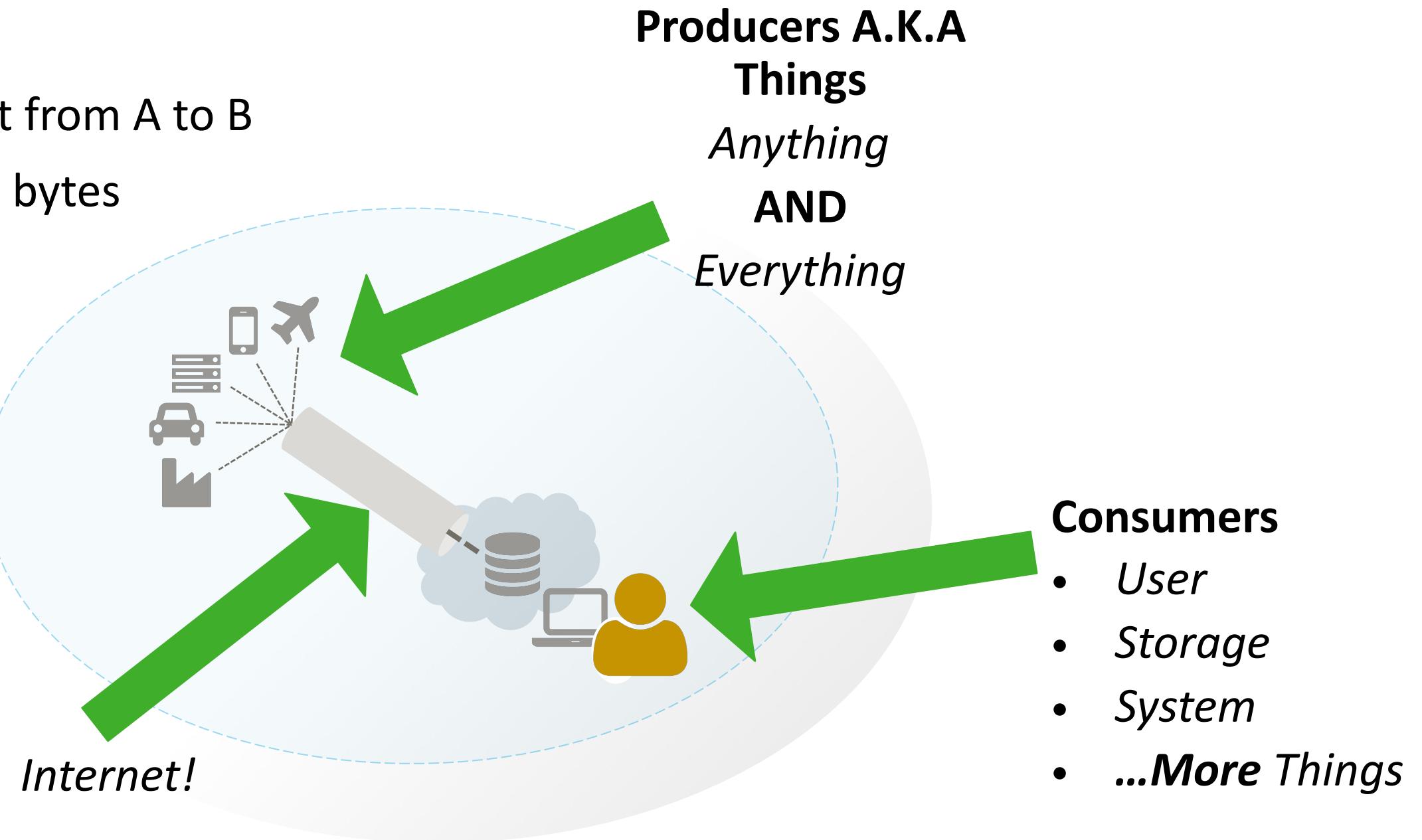
All slides provided online, no need to transcribe



What is dataflow?

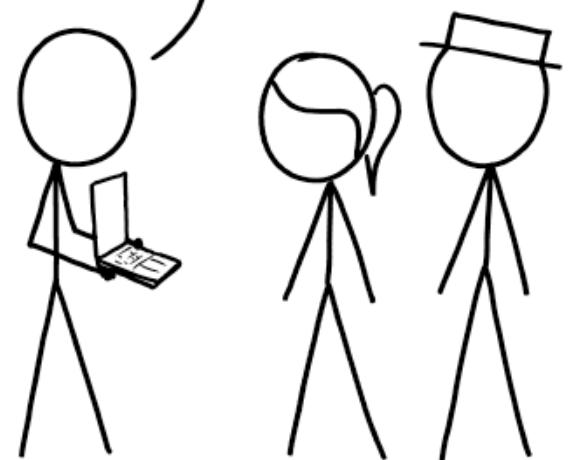
What is dataflow?

- Moving some content from A to B
- Content could be any bytes
 - Logs
 - HTTP
 - XML
 - CSV
 - Images
 - Video
 - Telemetry

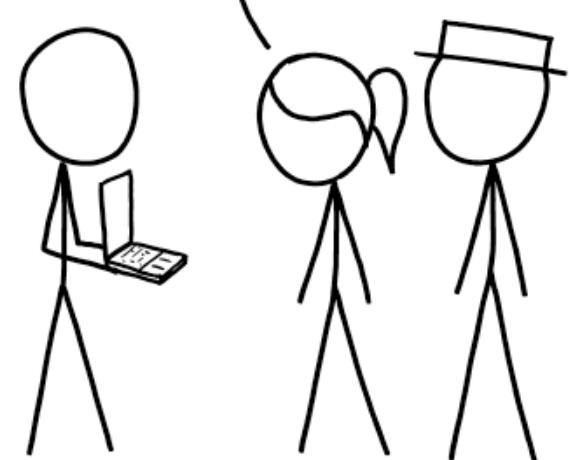


Moving data *effectively* is hard

CHECK IT OUT—I MADE A FULLY AUTOMATED DATA PIPELINE THAT COLLECTS AND PROCESSES ALL THE INFORMATION WE NEED.



IS IT A GIANT HOUSE OF CARDS BUILT FROM RANDOM SCRIPTS THAT WILL ALL COMPLETELY COLLAPSE THE MOMENT ANY INPUT DOES ANYTHING WEIRD?



IT... MIGHT NOT BE.

| I GUESS THAT'S SOMETHING.
WHOOPS, JUST COLLAPSED. HANG ON, I CAN PATCH IT.



“Data Pipeline” <https://xkcd.com/2054/>

Dataflow Challenges In 3 Categories

Data

- Standards
- **Formats**
- Protocols
- Veracity
- Validity
- Schemas
- Partitioning/
Bundling

Infrastructure

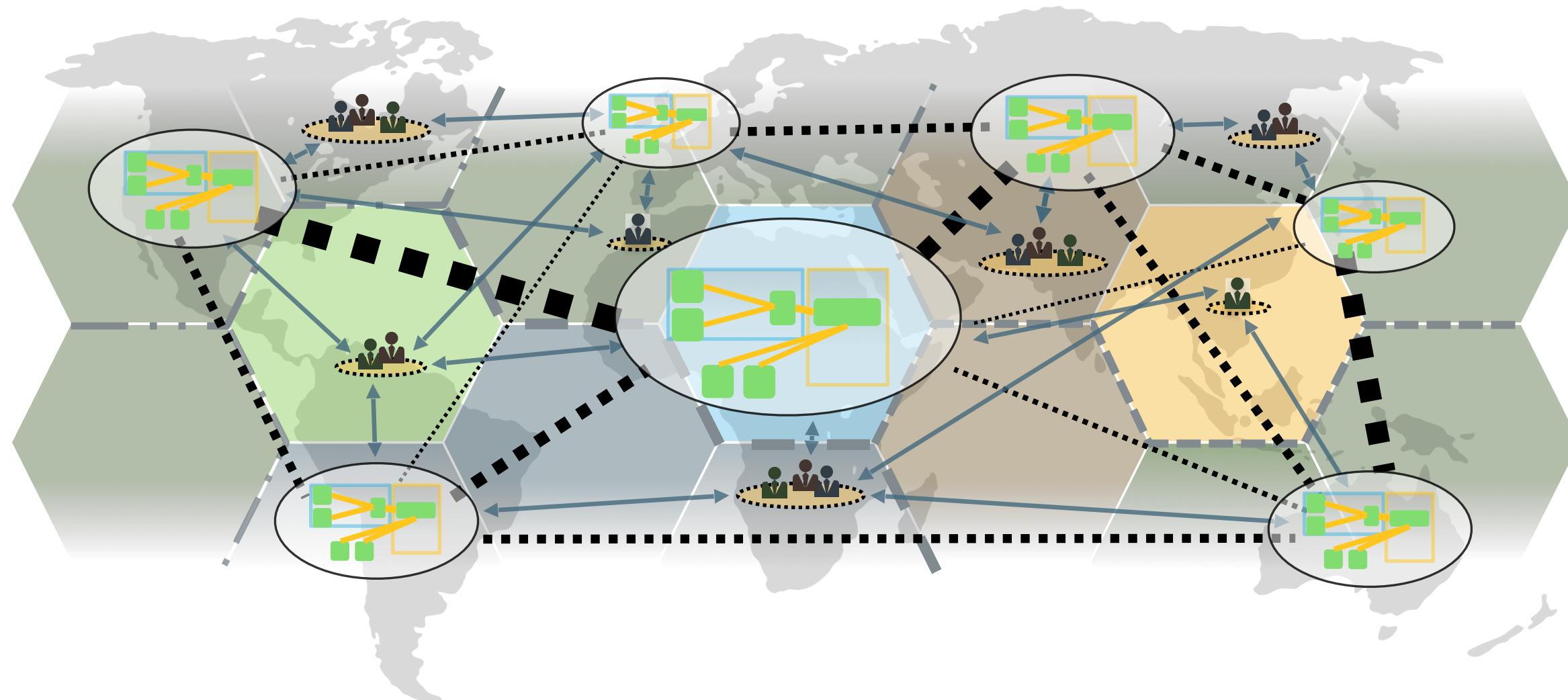
- “Exactly Once”
Delivery
- Ensuring
Security
- **Overcoming**
Security
- Credential
Management
- Network

People

- Compliance
- “**That** [person |
team | group]”
- **Consumers
Change**
- **Requirements
Change**
- “Exactly Once”
Delivery

Let's Connect Lots of As to Bs to As to Cs to Bs to Δ s to Cs to φ s

Raise your hand if you want to maintain Python scripts for the rest of your life



What is Apache NiFi?

One Minute Intro to the NiFi Ecosystem



NiFi

- Server/DC class
- GUI & REST API
- Interacts with hundreds of services



MiNiFi (JVM/C++)

- Agent class
- Runs on minimal hardware
- Interacts with NiFi
- Simple event processing

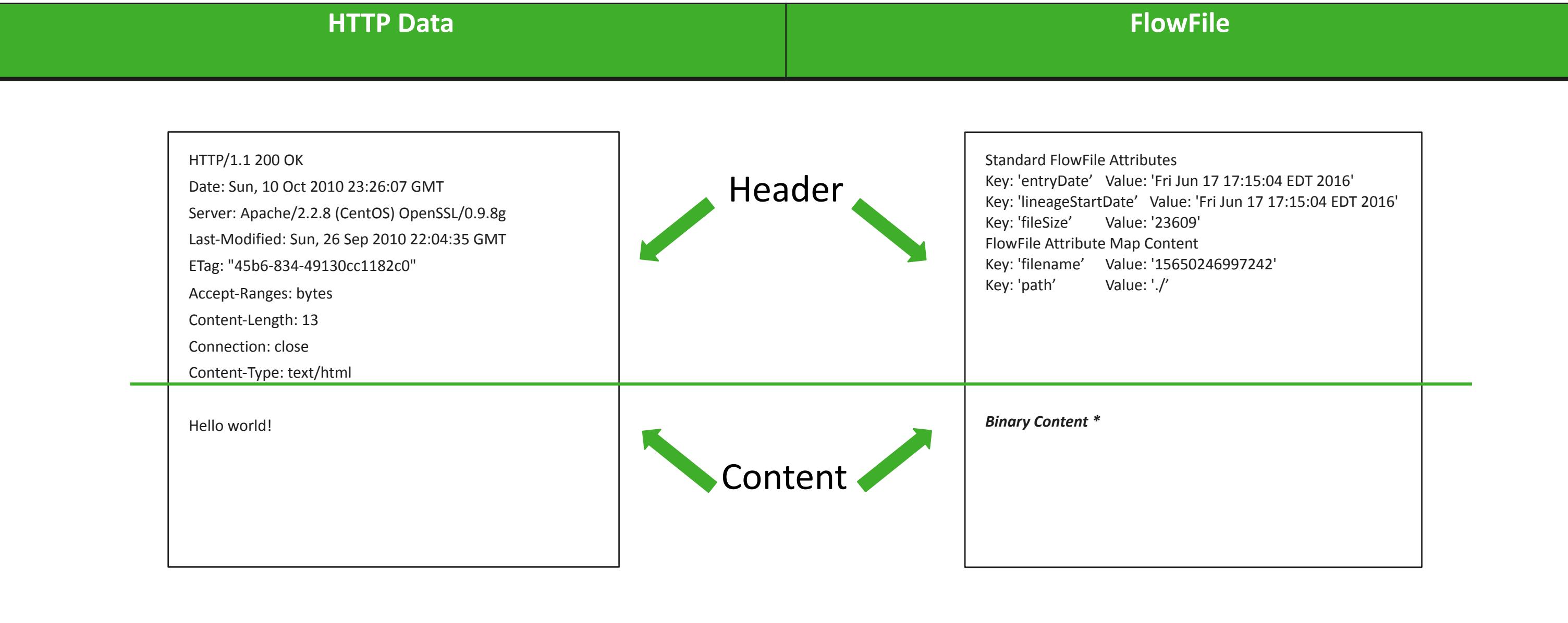


NiFi Registry

- Standalone application
- Handles asset management
- Flow versioning & deployment
- *Extension registry*



Flowfiles Are Like HTTP Data

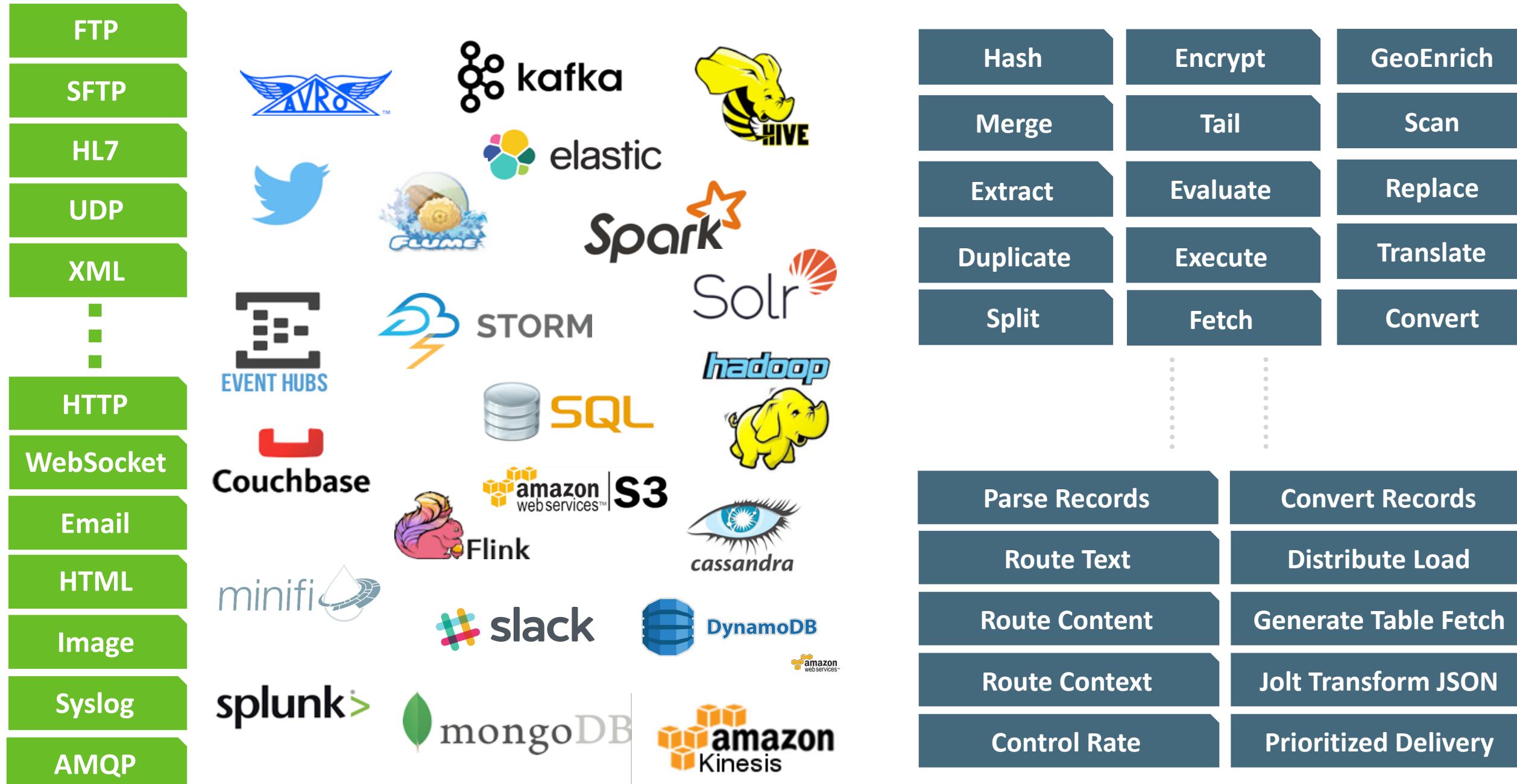


User Interface

Less of this..... more of this



Deeper Ecosystem Integration: 286+ Processors, 61 Controller Services



People want to know about their data



[Colin the Chicken | Portlandia | IFC](#)

“This is local? I’m going to ask you this *one* more time — this is local?”



[Colin the Chicken | Portlandia | IFC](#)

“Is that USDA organic, or Oregon organic, or Portland organic?”



[Colin the Chicken | Portlandia | IFC](#)

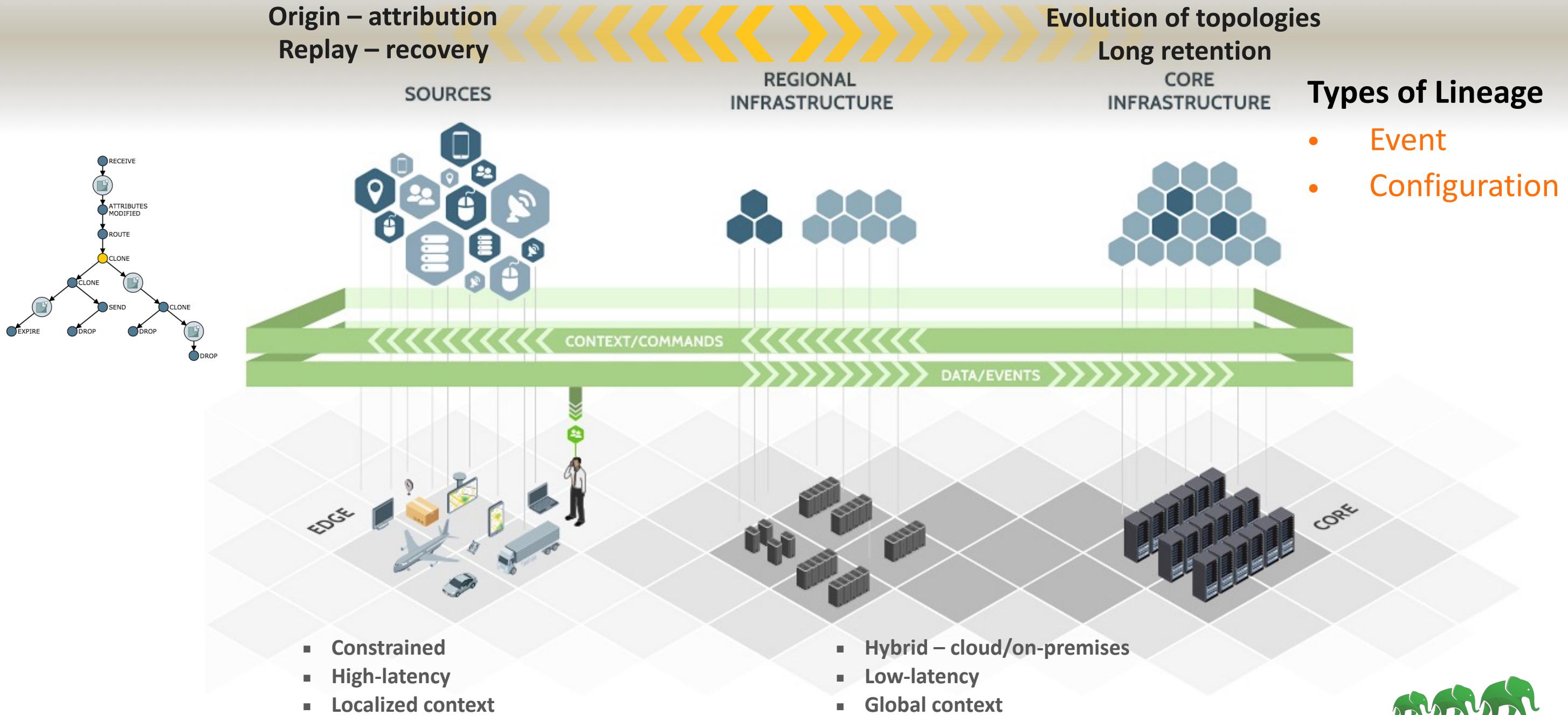
“Oh you have this information? This is fantastic!”



[Colin the Chicken | Portlandia | IFC](#)

Data Provenance

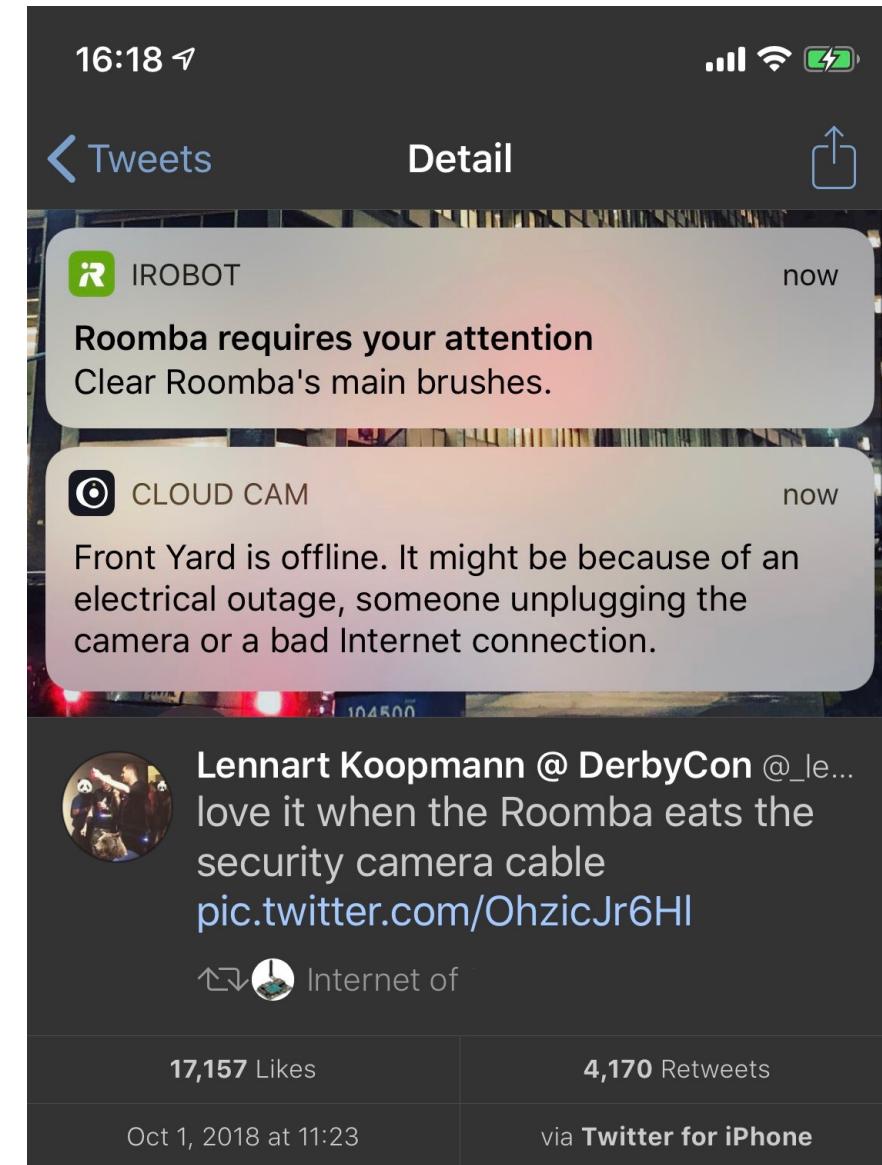
Origin – attribution
Replay – recovery



What are the IoT challenges?

IoT Challenges

- Limited computing capability
- Limited power/network
- **Restricted software library/platform availability**
- **No UI**
- Physically inaccessible
- Not frequently updated
- **Competing standards/protocols**
- Scalability
- **Privacy & Security**



@_lennart

Recent Examples

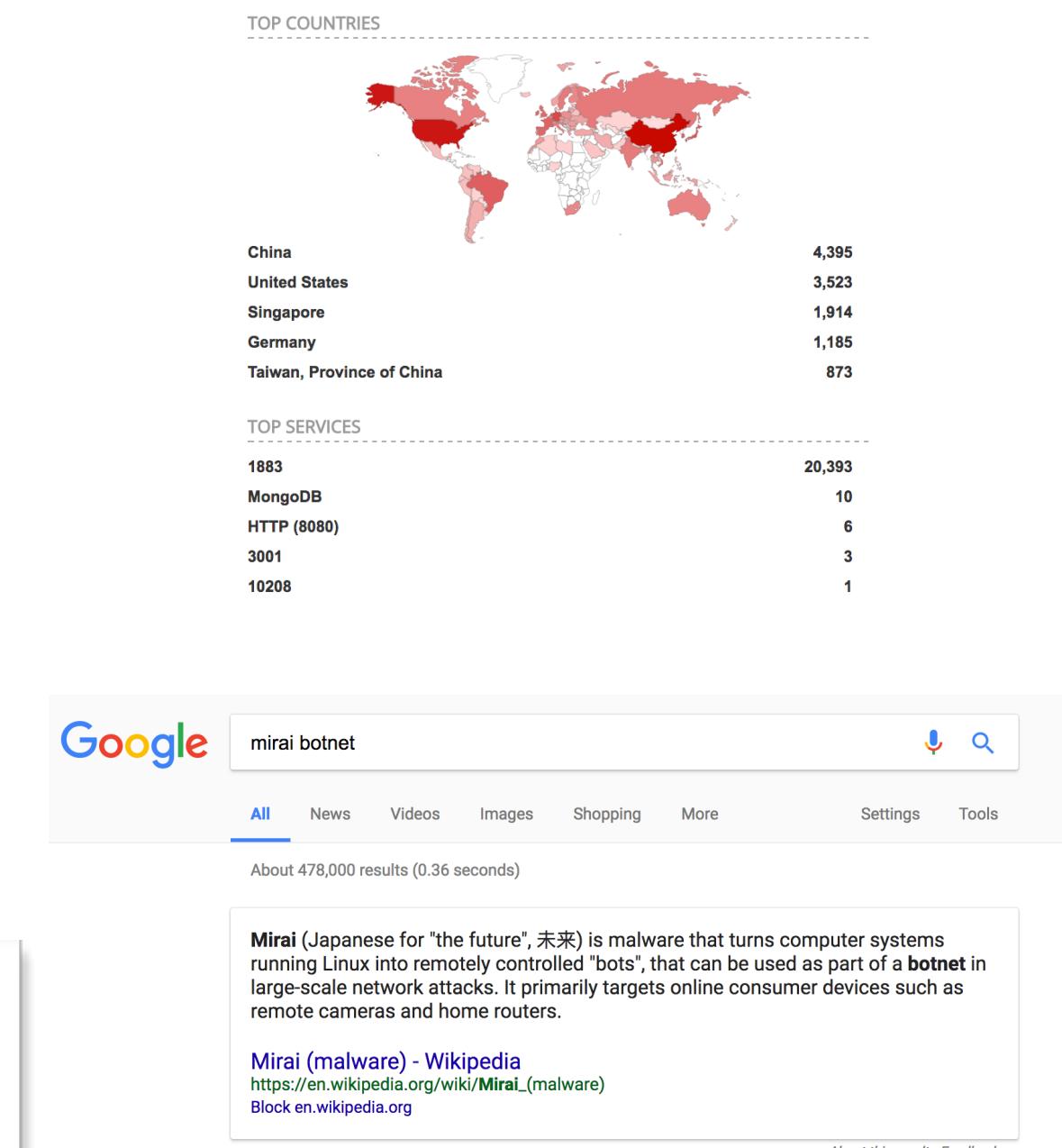
- When the Mirai attack has its own Wikipedia page, that's not good
- Hackers stole high-roller database from casino via aquarium thermometer connected to internet (04/2018)



IoTPOT: Analysing the Rise of IoT Compromises

Yin Minn Pa Pa^{†1}, Shogo Suzuki^{†1}, Katsunari Yoshioka^{†1}, Tsutomu Matsumoto^{†1},
 Takahiro Kasama^{†2}, Christian Rossow^{†3}

^{†1}Graduate School of Environment and Information Sciences/Institute of Advanced Sciences
^{†1}Yokohama National University, Japan
^{†2}National Institute of Information and Communications Technology, Japan
^{†3}Institute of Advanced Sciences, Yokohama National University, Japan and
^{†3}Cluster of Excellence, MMCI, Saarland University, Germany





“The S in IoT stands
for security”

Oleg Šelajev, @shelajev

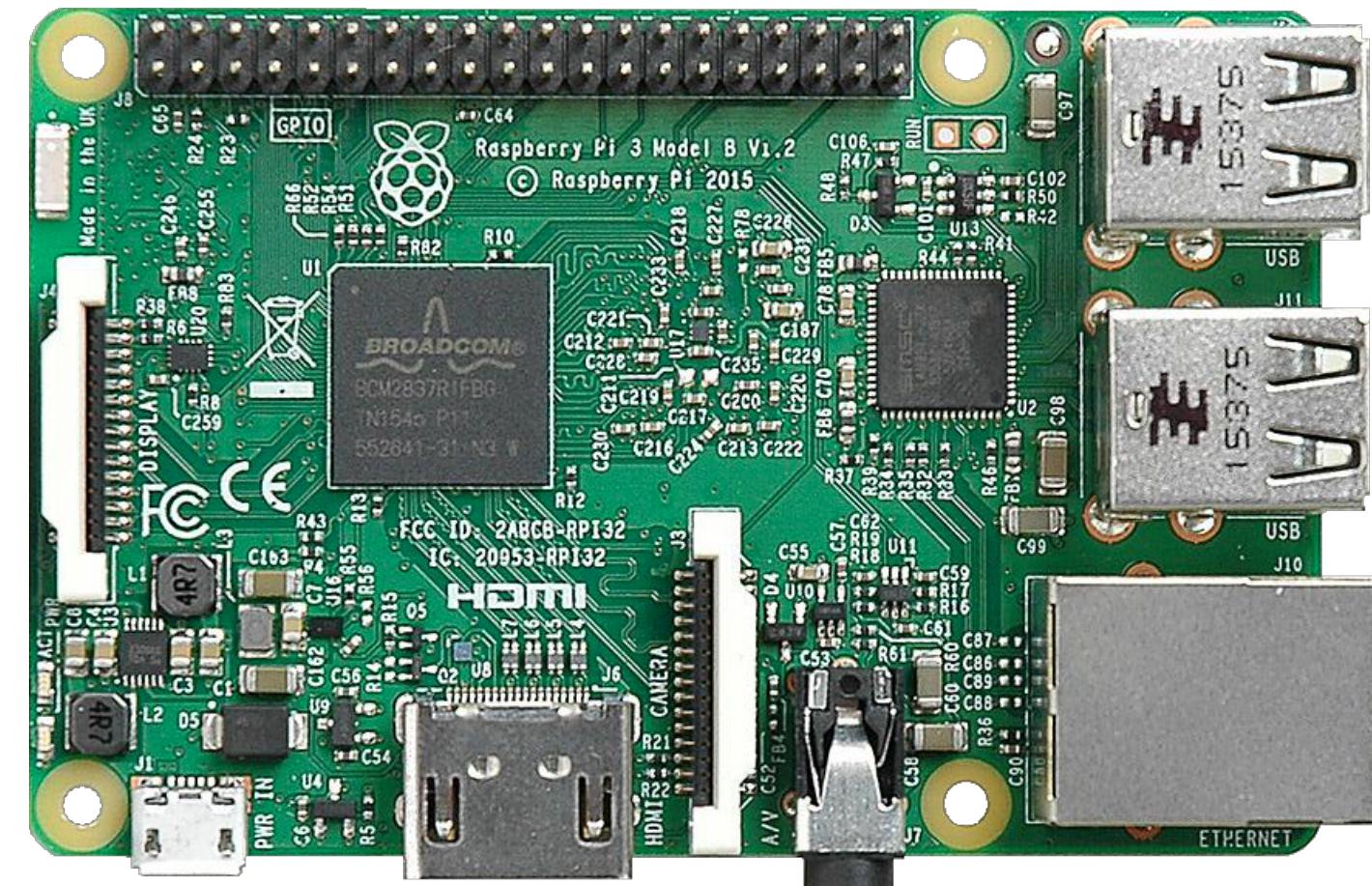
NiFi Solves Everything*

- Runs on JVM
- Provides UI for flow design & monitoring
- Security built-in
 - TLS, authentication/authorization, encrypted data
- Handles practically any format/protocol

NiFi for IoT

- NiFi supports AMQP, MQTT, UDP, TCP, HTTP(S), CEF, JMS, (S)FTP, *AWSIoT*
- With a little pruning, NiFi can run on a Raspberry Pi

```
bootstrap
jcl-over-slf4j-1.7.12.jar
jul-to-slf4j-1.7.12.jar
log4j-over-slf4j-1.7.12.jar
logback-classic-1.1.3.jar
logback-core-1.1.3.jar
nifi-api-0.6.1.jar
nifi-documentation-0.6.1.jar
nifi-framework-nar-0.6.1.nar
nifi-html-nar-0.6.1.nar
nifi-http-context-map-nar-0.6.1.nar
nifi-jetty-bundle-0.6.1.nar
nifi-kerberos-iaa-providers-nar-0.6.1.nar
nifi-ldap-iaa-providers-nar-0.6.1.nar
nifi-nar-utils-0.6.1.jar
nifi-properties-0.6.1.jar
nifi-provenance-repository-nar-0.6.1.nar
nifi-runtime-0.6.1.jar
nifi-scripting-nar-0.6.1.nar
nifi-ssl-context-service-nar-0.6.1.nar
nifi-standard-nar-0.6.1.nar
nifi-standard-services-api-nar-0.6.1.nar
nifi-update-attribute-nar-0.6.1.nar
slf4j-api-1.7.12.jar
```



So Why Do We Need A Different Solution?

- NiFi is designed to “own the box”
- NiFi 0.7.x started up in about 10-15 minutes on RP3 (593 MB)
- NiFi 1.x started up in about 30 minutes on RP3 (760 MB)
 - 33 new processors
 - Rewrite for multi tenant authorization
 - Complete UI overhaul

```
▶hw12203:/Users/alopresto/Workspace/scratch/rp3b-demo (master) alopresto
 113s @ 17:09:05 $ ssh pi@my-raspberry-pi
 ^C
▶hw12203:/Users/alopresto/Workspace/scratch/rp3b-demo (master) alopresto
 145s @ 17:09:37 $
```

Enter Apache MiNiFi

Apache NiFi Subproject: MiNiFi

- Get the key parts of NiFi close to where data begins and provide bidirectional communication
- NiFi lives in the data center — give it an enterprise server or a cluster of them
- MiNiFi lives as close to where data is born and is a guest on that device or system
 - IoT
 - Connected car
 - Legacy hardware

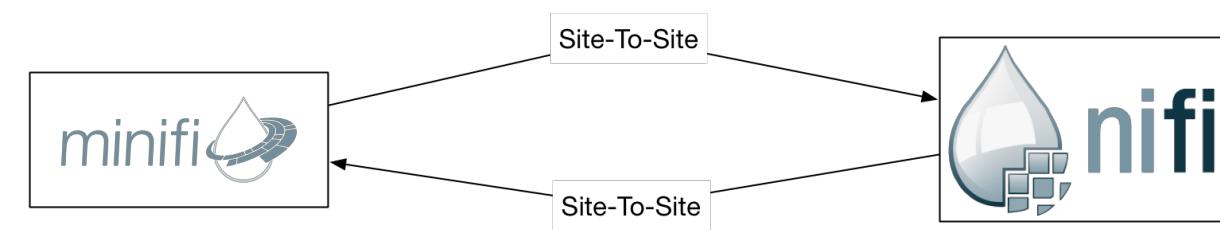
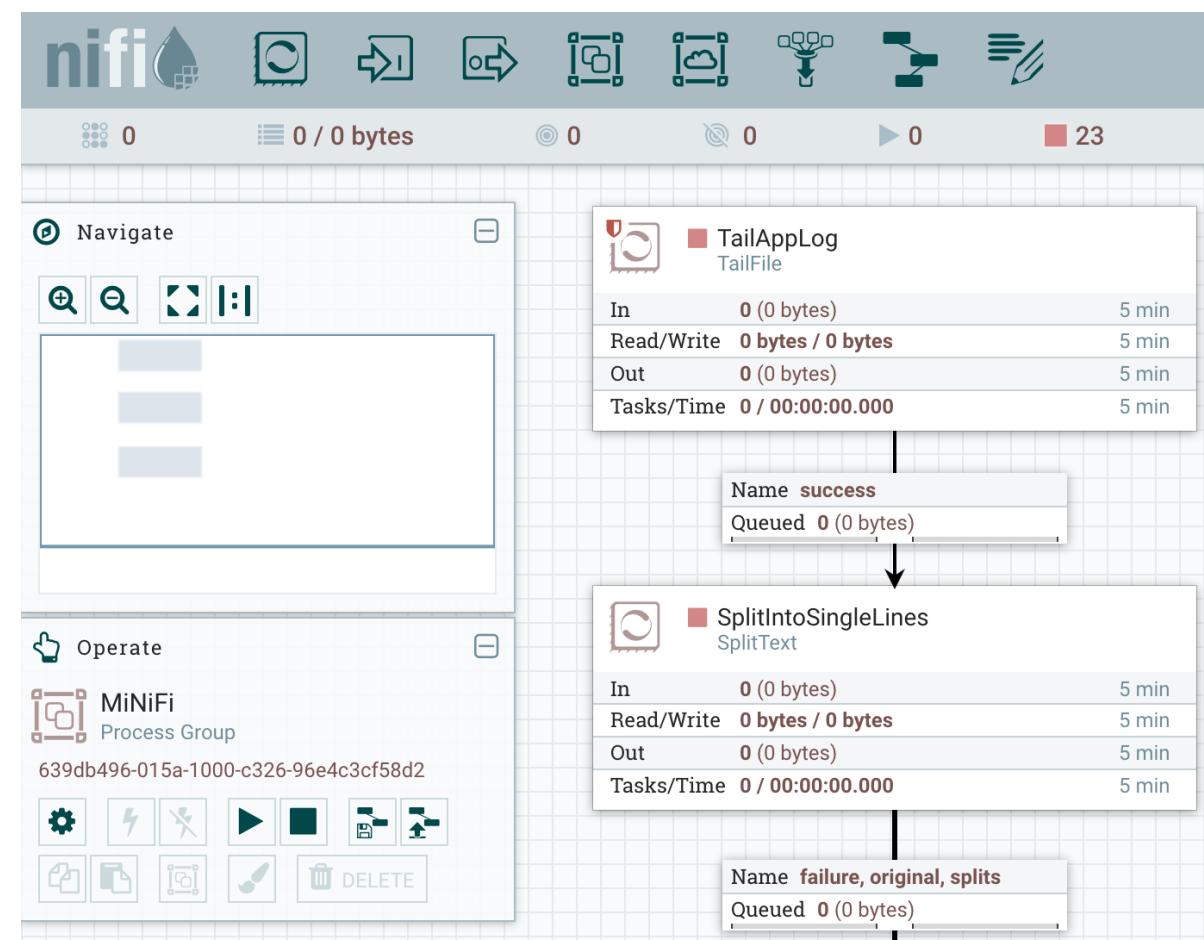


Why build MiNiFi?

- NiFi is big
 - 1.9.1 release is 1.2 GB compressed
 - Can be modified to run in restricted environments, but requires manual surgery
 - Provides UI, provenance query, etc.
 - Runs on dedicated machines/clusters — “owns the box”
- MiNiFi lives at the edge
 - No UI
 - 0.5.0 Java release is 67 MB, C++ release is 6.1 MB (**0.2.0 fits on a floppy disk**)
 - “Good guest”

Flavors of MiNiFi

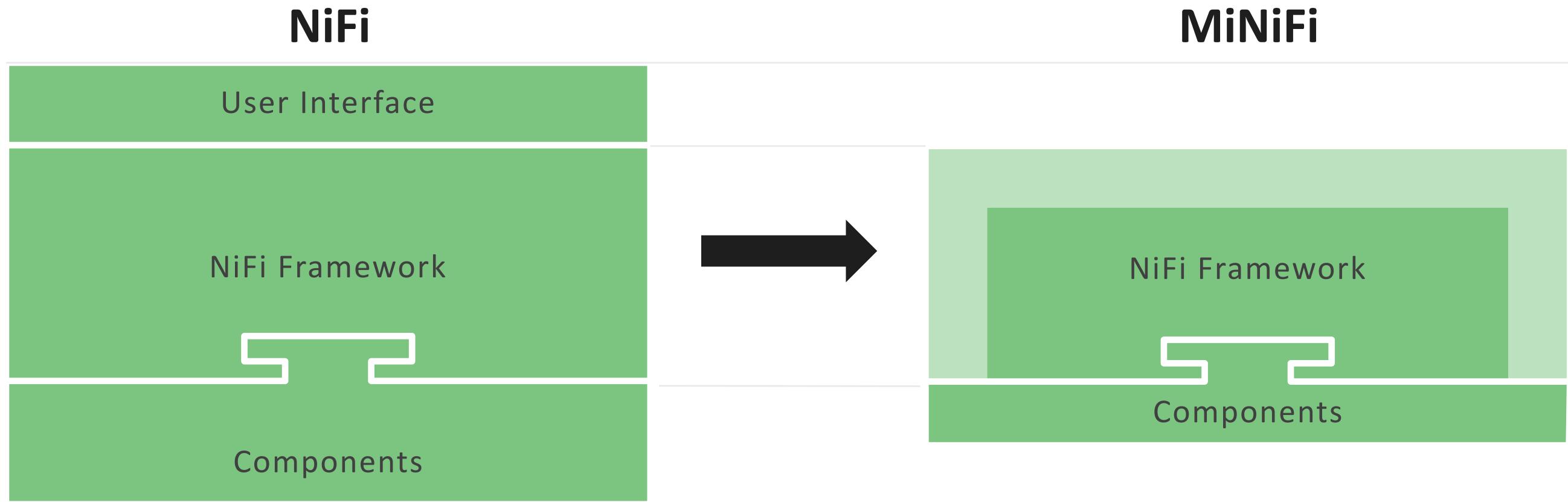
- MiNiFi Java (v0.5.0)
 - Modified version of NiFi
 - No UI
 - YAML configuration
 - Reduced processor count
 - 63+ by default, more available with additional NARs
- MiNiFi C++ (v0.5.0)
 - Written from scratch
 - 33 processors by default
 - Bi-directional site-to-site & provenance data



```
Security Properties:
  keystore: /tmp/ssl/localhost-ks.jks
  keystore type: JKS
  keystore password: localtest
  key password: localtest
  truststore: /tmp/ssl/localhost-ts.jks
  truststore type: JKS
  truststore password: localtest
  ssl protocol: TLS
  Sensitive Props:
    key:
      algorithm: PBWEWITHMD5AND256BITAES-CBC-OPENSSL
      provider: BC

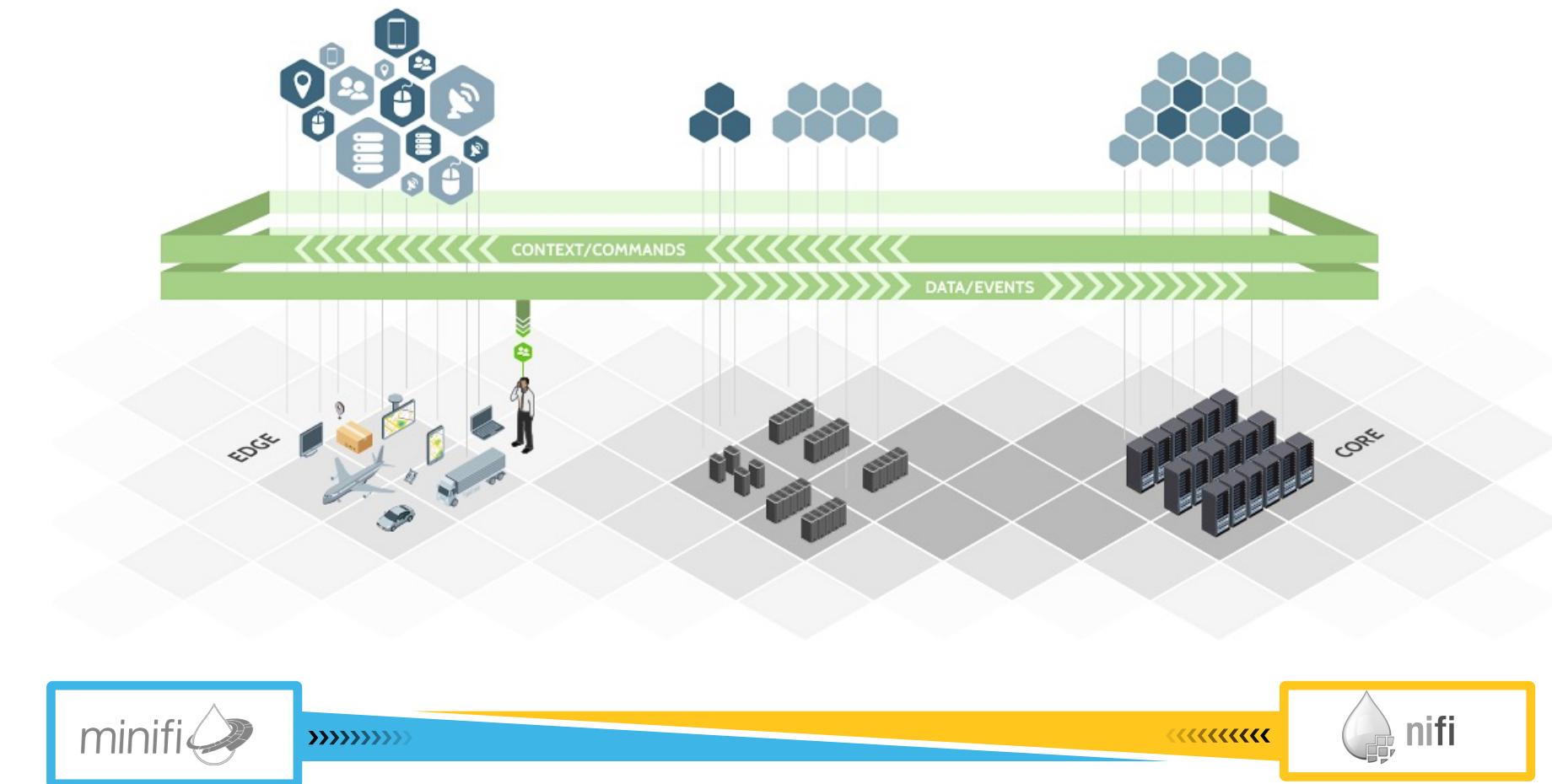
Processors:
  - name: TailAppLog
    class: org.apache.nifi.processors.standard.TailFile
    max concurrent tasks: 1
    scheduling strategy: TIMER_DRIVEN
    scheduling period: 10 sec
    penalization period: 30 sec
    yield period: 1 sec
    run duration nanos: 0
    auto-terminated relationships list:
      Properties:
        File to Tail: logs/minifi-app.log
        Rolling Filename Pattern: minifi-app*
        Initial Start Position: Beginning of File
  - name: SplitIntoSingleLines
    class: org.apache.nifi.processors.standard.SplitText
    max concurrent tasks: 1
    scheduling strategy: TIMER_DRIVEN
    scheduling period: 0 sec
    penalization period: 30 sec
    yield period: 1 sec
    run duration nanos: 0
    auto-terminated relationships list:
      - failure
      - original
    Properties:
      Line Split Count: 1
      Header Line Count: 0
      Remove Trailing Newlines: true
```

NiFi vs MiNiFi Java Processes



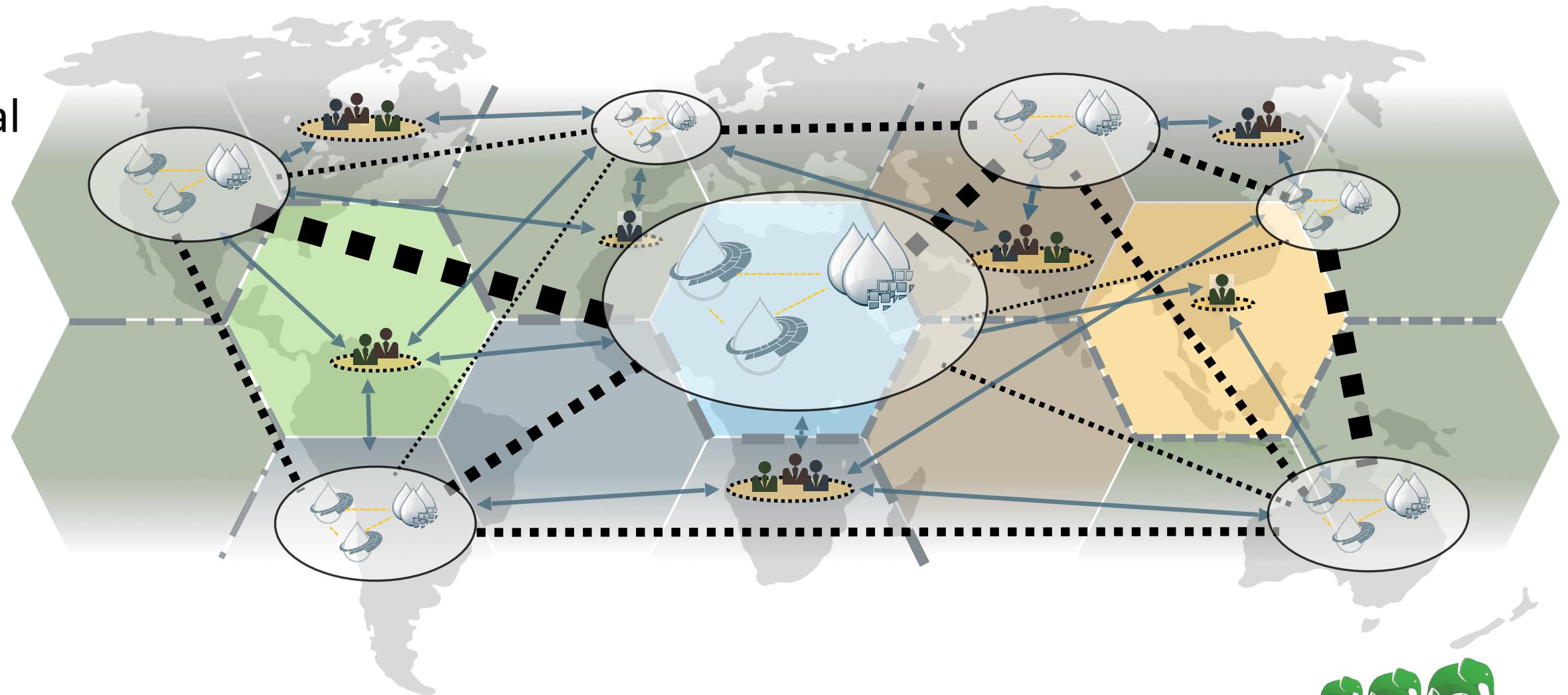
How Does MiNiFi Interact With NiFi?

- NiFi
 - Design flows
 - Aggregate data from many sources
 - Perform routing/analysis/SEP
- MiNiFi
 - Receive flows
 - Collect data
 - Send for processing



Let's Add Dimensionality

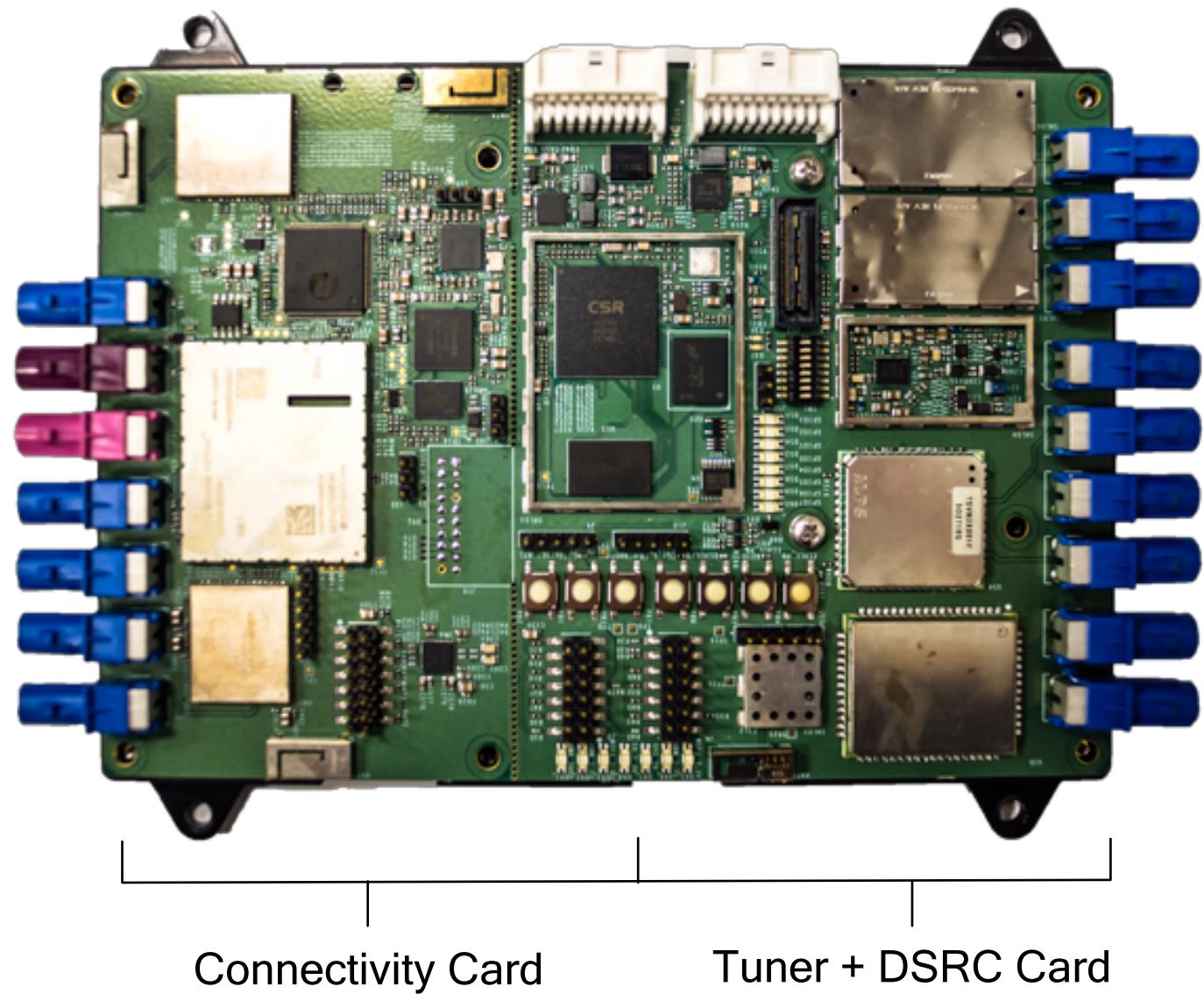
- We've been imagining EDGE to CORE as a bi-directional linear system
- Let's expand that to the real world



What does MiNiFi provide?

- Data tagging/provenance
- Governance from edge (geopolitical restrictions)
- Security (encryption, certificate-based authentication)
- Low latency (immediate reactions & decision-making)

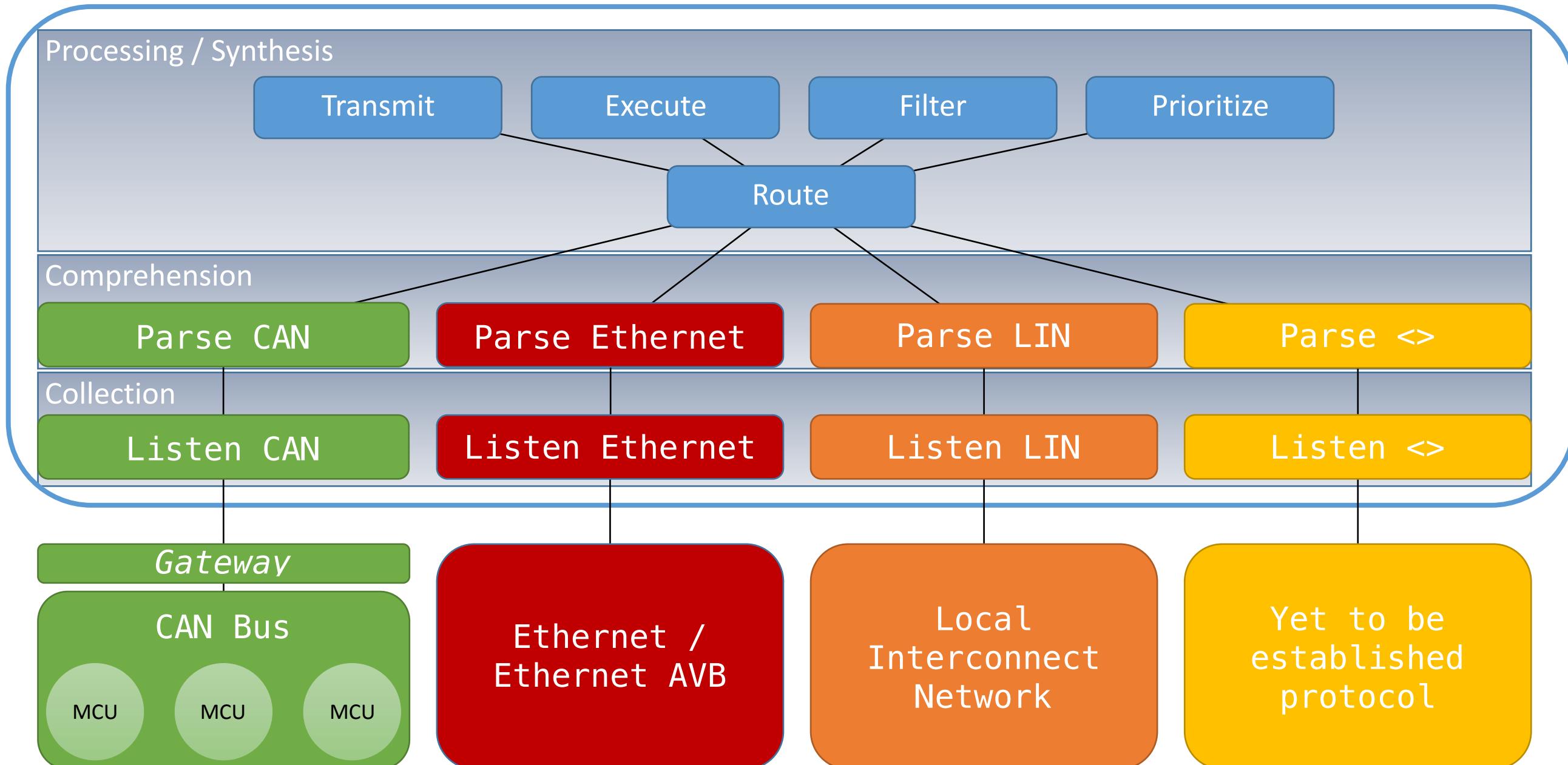
Connected Car Reference Platform Box



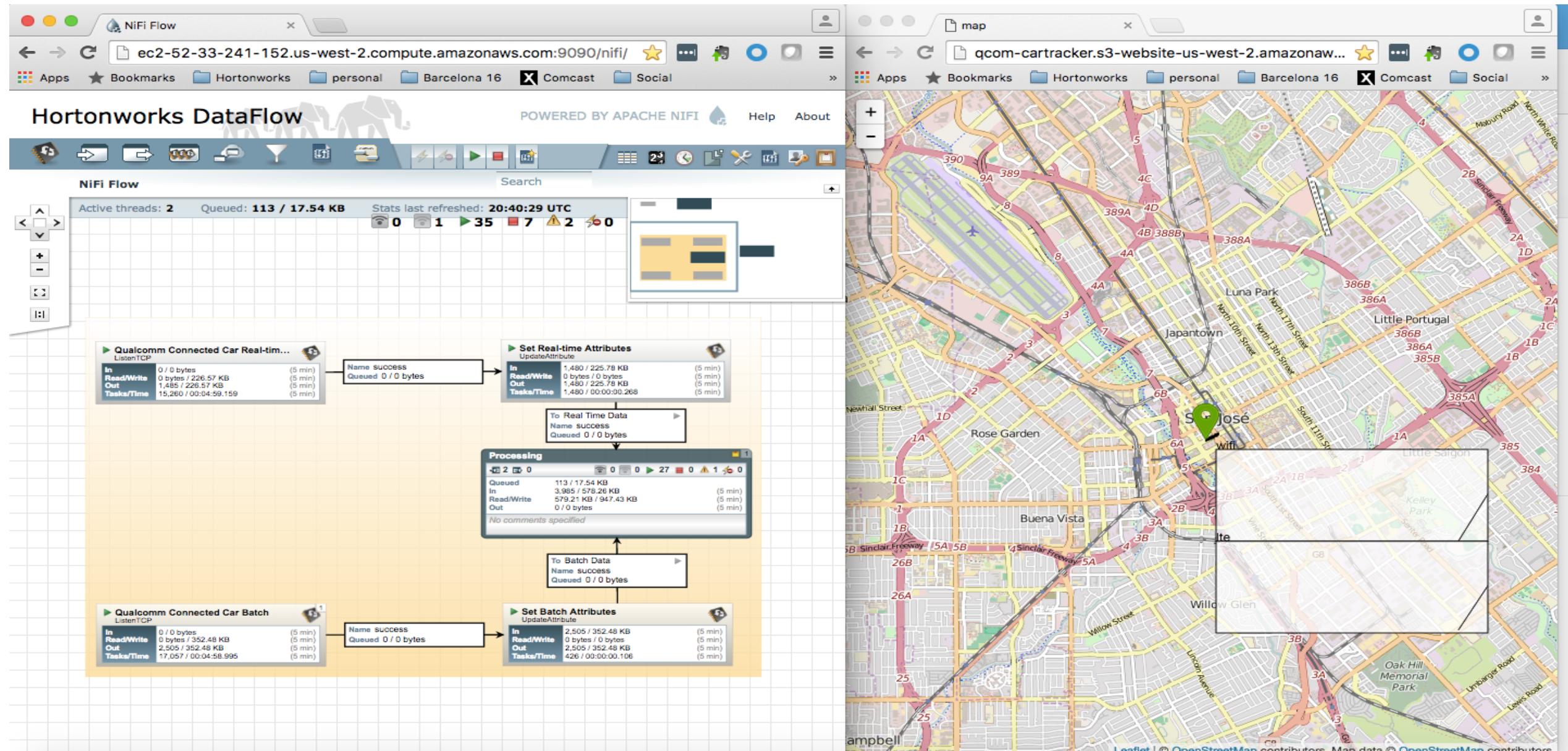
Connectivity Card

Tuner + DSRC Card

MiNiFi on a Connected Car



MiNiFi on a Connected Car



MiNiFi Exfil

- Site-to-Site
 - NiFi protocol
 - Two implementations
 - Raw socket
 - HTTP(S)
- Secured with mutual authentication TLS
 - HTTP(S), (S)FTP, JMS, Syslog, File, Email, Process

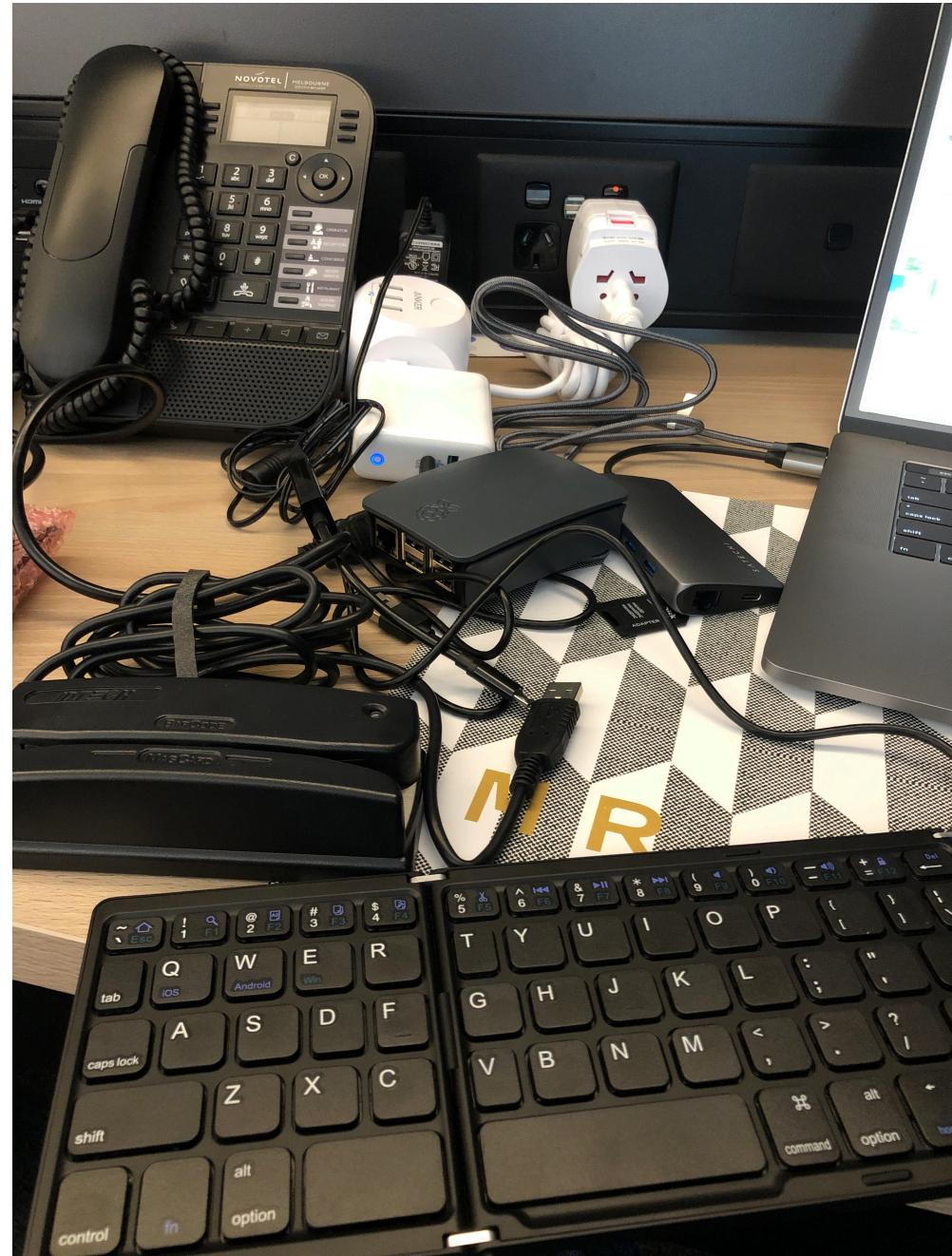
Edge Data Exploration



Credit Card “Processing”

Collect Card Contents

- User swipes magnetic stripe card
- Raspberry Pi collects data
- Sends back to laptop



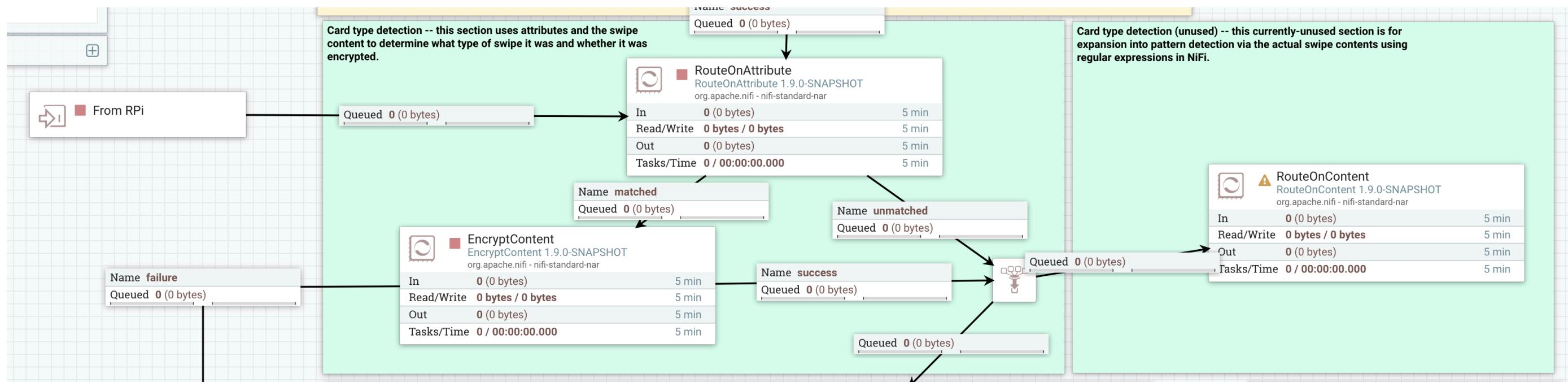
Component List

- IDTECH WCR3237-733U
- Raspberry Pi 3 B+
- Apache MiNiFi 0.6.0-SNAPSHOT
- Apache NiFi 1.9.0-SNAPSHOT
- Apache NiFi Registry 0.4.0-SNAPSHOT

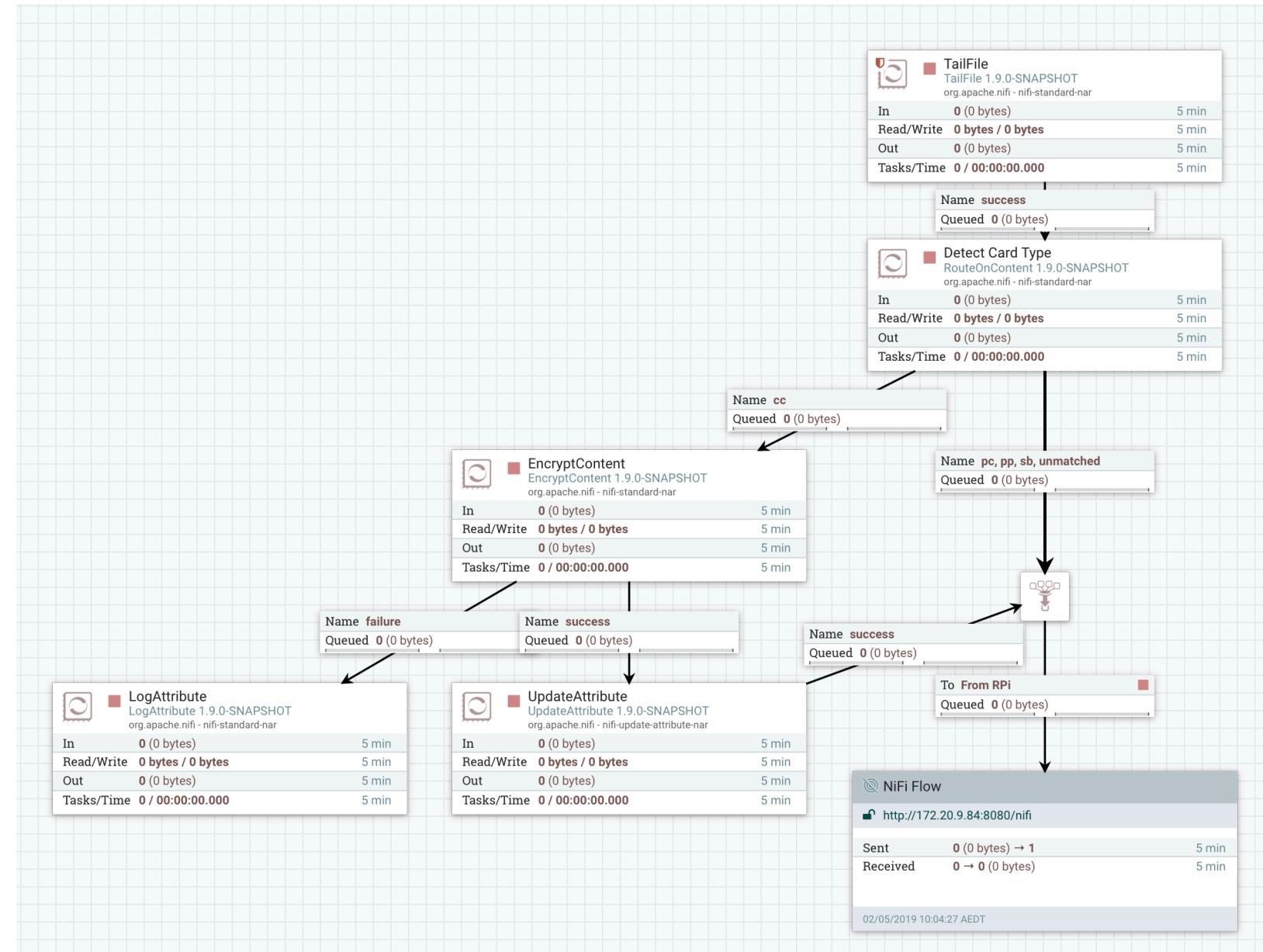
Flow Design

- NiFi Flow
 - Receive data via S2S
 - Route encrypted CC data to decrypt
 - Route all plaintext swipes through parser
 - Convert to JSON
 - Batch & write to file
- MiNiFi Flow
 - Tail file containing swipe data
 - Detect CC swipes & encrypt
 - Transmit all swipes to NiFi via S2S

NiFi Flow



MiNiFi Flow



Custom MSR Parsing Logic

```
1 import groovy.json.JsonBuilder
2 import groovy.json.JsonOutput
3 import groovy.json.JsonSlurper
4 import org.apache.commons.io.IOUtils
5 import org.apache.nifi.processor.io.StreamCallback
6
7 import java.nio.charset.StandardCharsets
8
9 def flowFile = session.get()
10 if (flowFile == null) {
11     return
12 }
13
14 final String CC_PATTERN = /%B\d{16}\^A-Za-z+/
15 final String PC_PATTERN = /%1w+/
16 final String PP_PATTERN = /%PP\//\w+/
17 final String SBUX_PATTERN = /%B601.*SERVICERECOVERY/
18
19 String determinedType = "unknown"
20
21 flowFile = session.write(flowFile, { inputStream, outputStream ->
22     def content = IOUtils.toString(inputStream, StandardCharsets.UTF_8)
23     Map parsedObject
24
25     // Logic to do various parsing
26     if (content =~ CC_PATTERN) {
27         log.info("Detected CC format")
28         parsedObject = parse(content, detectedType: "CC")
29     } else if (content =~ PP_PATTERN) {
30         log.info("Detected Priority Pass format")
31         parsedObject = parse(content, detectedType: "PP")
32     } else if (content =~ PC_PATTERN) {
33         log.info("Detected Players Club format")
34         parsedObject = parse(content, detectedType: "PC")
35     } else if (content =~ SBUX_PATTERN) {
36         log.info("Detected Starbucks Gift Card format")
37         parsedObject = parse(content, detectedType: "SB")
38     } else {
39         log.info("Could not determine MSR type; using generic parsing heuristics")
40         parsedObject = parse(content, detectedType: "generic")
41     }
42
43     // Write the determined type to an attribute
44     determinedType = parsedObject["type"]
45
46     // Write the timing info to the parsed object
47     parsedObject["time_parsed"] = new Date().format(format: 'YYYY-MM-dd_HH-mm-ss.SSS Z')
48     parsedObject["id"] = flowFile.uuid
49     parsedObject["source"] = flowFile.filename
50
51     // Convert the object into JSON
52     def jsonString = JsonOutput.toJson(parsedObject)
53     log.info("Formed JSON: ${jsonString}")
54
55     outputStream.write(jsonString.getBytes(StandardCharsets.UTF_8))
56
57 } as StreamCallback)
58 flowFile.determinedType = determinedType
```

```
59
60     private Map parse(String content, String detectedType) {
61         Map parsedObject = [:]
62         parsedObject["type"] = detectedType
63
64         def tracks = content.split( regex: /\?\n/ )
65         log.info("Tracks ${tracks.size()}: ${tracks.collect { "\n\t${it}" }}")
66
67         // Put the track 1 data directly
68         parsedObject["track1"] = tracks[0]
69
70         def track1Elements = tracks[0].split( regex: /\^/ )
71         log.info("Track 1 Elements ${track1Elements.size()}: ${track1Elements.collect { "\n\t${it}" }}")
72
73         // The PAN and Name are common elements
74         parsedObject["pan"] = track1Elements[0][2..-1] // Skip the start sentinel and format code (%B)
75         if (track1Elements.size() > 1) {
76             parsedObject["name"] = track1Elements[1].trim()
77         }
78         if (track1Elements.size() > 2) {
79             // Put the remaining track 1 data in a generic field
80             parsedObject["additionalTrack1"] = track1Elements[2..-1].join("^")
81         }
82
83         // Put track 2 data directly
84         parsedObject["track2"] = tracks[1]
85
86         // Return the parsed object
87         parsedObject
88
89
90     session.transfer(flowFile, REL_SUCCESS)
```

Running on Raspberry Pi

LXTerminal

File Edit Tabs Help

```
pi@gobias:/opt/iot$ ./prompt.sh
Swipe your card : : pi@gobias:/opt/iot$ more swipes.txt
%PP//ANDREW/D/LOPREST0?\n;7357450001927901=102018?
pi@gobias:/opt/iot$
```

0 items

LXTerminal

File Edit Tabs Help

```
459-510.txt,size=51]
-----[Transmitting data]-----
Standard FlowFile Attributes
Key: 'entryDate'
    Value: 'Tue Feb 05 14:43:32 AEDT 2019'
Key: 'lineageStartDate'
    Value: 'Tue Feb 05 14:43:32 AEDT 2019'
Key: 'fileSize'
    Value: '51'
FlowFile Attribute Map Content
Key: 'RouteOnContent.Route'
    Value: 'pp'
Key: 'filename'
    Value: 'swipes.459-510.txt'
Key: 'mime.type'
    Value: 'text/plain'
Key: 'path'
    Value: './'
Key: 'tailfile.original.path'
    Value: '/opt/iot/swipes.txt'
Key: 'uuid'
    Value: '3dc63eb3-a3fd-4e47-bfce-46dc7792d4a3'
-----[Transmitting data]-----
%PP//ANDREW/D/LOPREST0?\n;7357450001927901=102018?
```

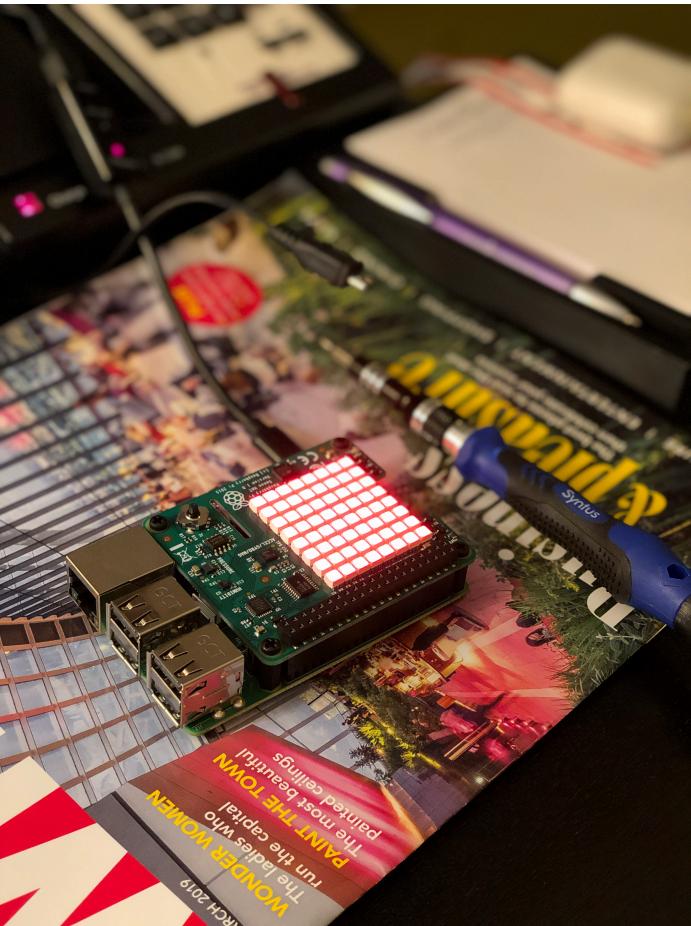
Free spa

Running on MacBook Pro

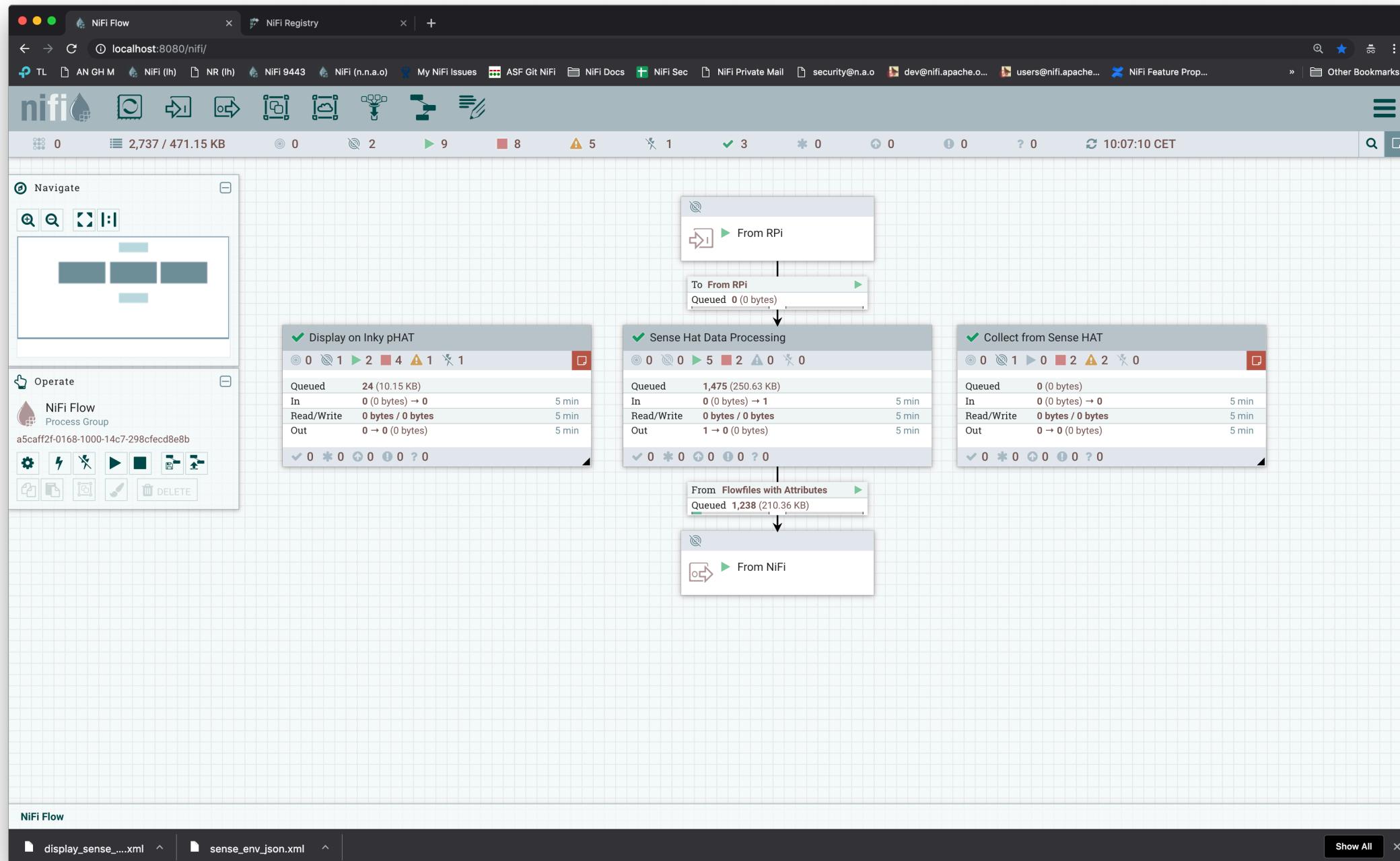
```
× 2 scratch/iot_ms (more)
drwxr-xr-x 8 alopreno staff 256B Feb 5 14:47 .
drwxr-xr-x 180 alopreno staff 5.6K Feb 3 14:17 ..
-rw-r--r-- 1 alopreno staff 2.9K Feb 3 14:19 2019-02-03_14-19-24_824_+1100.json
-rw-r--r-- 1 alopreno staff 2.9K Feb 3 14:19 2019-02-03_14-19-29_828_+1100.json
-rw-r--r-- 1 alopreno staff 3.5K Feb 3 14:19 2019-02-03_14-19-34_836_+1100.json
-rw-r--r-- 1 alopreno staff 3.4K Feb 3 14:19 2019-02-03_14-19-40_837_+1100.json
-rw-r--r-- 1 alopreno staff 3.7K Feb 3 14:19 2019-02-03_14-19-46_849_+1100.json
-rw-r--r-- 1 alopreno staff 3.4K Feb 3 14:19 2019-02-03_14-19-52_853_+1100.json
> . . . sers/alopreno/Workspace/scratch/iot_ms (master) 😊
os @ 14:49:01 $ more 2019-02-03_14-19-52_853_+1100.json
{"type": "PC", "track1": "%1AANDREW. 01037983102300", "pan": "AANDREW. 01037983102300", "track2": "01037983102300000?", "time_parsed": "2019-02-03_14-19-46.884 +1100", "id": "47fc007e-aa24-47b5-94aa-7793acb8cbe4", "source": "564a9f2f-74b8-4a47-9cad-df8742e4a02a"}  
{"type": "PP", "track1": "%PP//ANDREW/D/LOPREST0", "pan": "P//ANDREW/D/LOPREST0", "track2": "7357450001927901=102018?", "time_parsed": "2019-02-03_14-19-47.862 +1100", "id": "ffea2add-03e6-49d8-b0ea-570ff7485d0c", "source": "739599e1-2286-440a-8635-623206f589cf"}  
{"type": "PC", "track1": "%1AANDREW. 01037983102300", "pan": "AANDREW. 01037983102300", "track2": "01037983102300000?", "time_parsed": "2019-02-03_14-19-47.863 +1100", "id": "948ba9d7-aada-462f-8e37-fb9ab86515c9", "source": "70c3b55e-ce4e-49b6-b818-e3308eee6f80"}  
{"type": "PC", "track1": "%1AANDREW. 01037983102300", "pan": "AANDREW. 01037983102300", "track2": "01037983102300000?", "time_parsed": "2019-02-03_14-19-48.877 +1100", "id": "5bb6bead-a1c2-4b90-883d-532873d9204d", "source": "323b7c10-21d1-439f-9ce6-20b7afa233c1"}  
{"type": "CC", "track1": "%B4147098571707173^LOPREST0/ANDREW ^170510100000000014051514000000", "pan": "4147098571707173", "name": "LOPREST0/ANDREW", "additionalTrack1": "170510100000000014051514000000", "track2": "4147098571707173=170510114051514?", "time_parsed": "2019-02-03_14-19-48.899 +1100", "id": "2b30ccfe-14d1-4a34-bd8c-72200d013ff0", "source": "9ceb7d41-d863-4187-87d4-64607471539c"}  
{"type": "PC", "track1": "%1AANDREW. 01037983102300", "pan": "AANDREW. 01037983102300", "track2": "01037983102300000?", "time_parsed": "2019-02-03_14-19-49.867 +1100", "id": "5738eab2-8042-4d15-bc4f-fa23171f9f87", "source": "963d9b68-8f8a-4cc2-a51f-d4c8aeb63250"}  
{"type": "PP", "track1": "%PP//ANDREW/D/LOPREST0", "pan": "P//ANDREW/D/LOPREST0", "track2": "7357450001927901=102018?", "time_parsed": "2019-02-03_14-19-49.868 +1100", "id": "dbfa5730-c689-47a3-8583-bd90b6adb0da", "source": "a29256ed-a299-4bb2-8f48-0e1be944a16f"}  
{"type": "SB", "track1": "%B6010569709984040^0254/SERVICERECOVERYUSD^00010004000060144271. ", "pan": "6010569709984040", "name": "0254/SERVICERECOVERYUSD", "additionalTrack1": "00010004000060144271. ", "track2": "6010569709984040=00010004000060144271?", "time_parsed": "2019-02-03_14-19-49.866 +1100", "id": "96e37f83-af28-4b53-a86e-fcb8ba3184ab", "source": "0cd54ca1-7efb-4e42-b600-979c56d0b6b3"}  
{"type": "PC", "track1": "%1AANDREW. 01037983102300", "pan": "AANDREW. 01037983102300", "track2": "01037983102300000?", "time_parsed": "2019-02-03_14-19-50.879 +1100", "id": "91bb2200-3b65-4659-a8ad-33943a22f335", "source": "6730a23a-5106-4ebb-b3d9-bae3b896f4c1"}  
{"type": "CC", "track1": "%B4147098571707173^LOPREST0/ANDREW ^170510100000000014051514000000", "pan": "4147098571707173", "name": "LOPREST0/ANDREW", "additionalTrack1": "170510100000000014051514000000", "track2": "4147098571707173=170510114051514?", "time_parsed": "2019-02-03_14-19-51.868 +1100", "id": "882a0255-81c4-41ed-9b34-9856557509ec", "source": "d3bd20ee-f74a-4a01-9676-9dd0384fca86"}  
{"type": "PC", "track1": "%1AANDREW. 01037983102300", "pan": "AANDREW. 01037983102300", "track2": "01037983102300000?", "time_parsed": "2019-02-03_14-19-51.867 +1100", "id": "1dacea93-59e1-4205-b77a-6872907c0013", "source": "dc1e8196-16a4-421f-870a-b369b4bbe099"}  
{"type": "PP", "track1": "%PP//ANDREW/D/LOPREST0", "pan": "P//ANDREW/D/LOPREST0", "track2": "7357450001927901=102018?", "time_parsed": "2019-02-03_14-19-51.869 +1100", "id": "18c23941-7665-49f5-a99a-d157de6d828f", "source": "6b1de415-1578-4be9-b872-ce6216d86cc9"}  
2019-02-03_14-19-52_853_+1100.json (END)
```

Sense HAT & Inky pHAT

Sense Hat & Inky pHAT

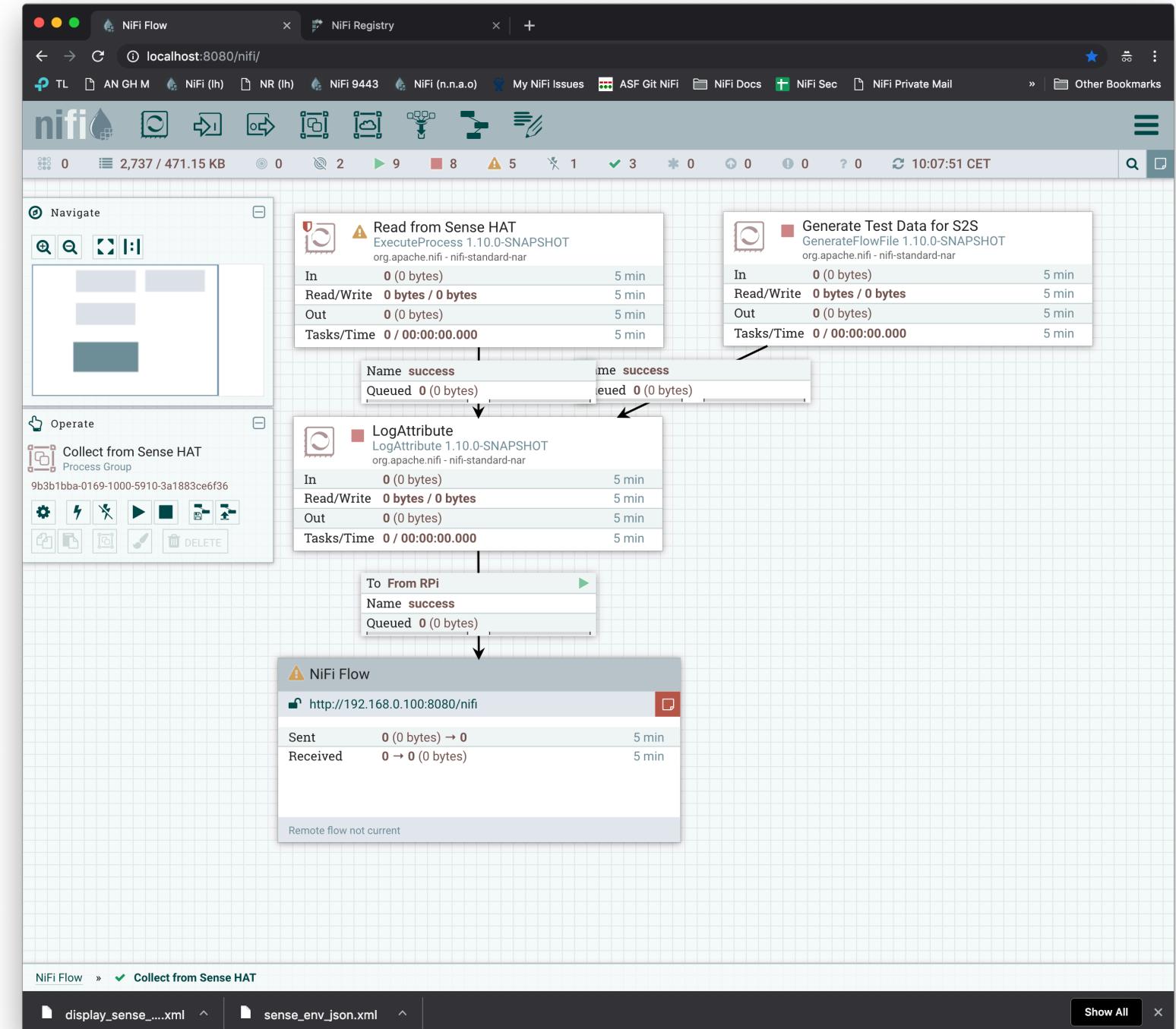


Flow Design



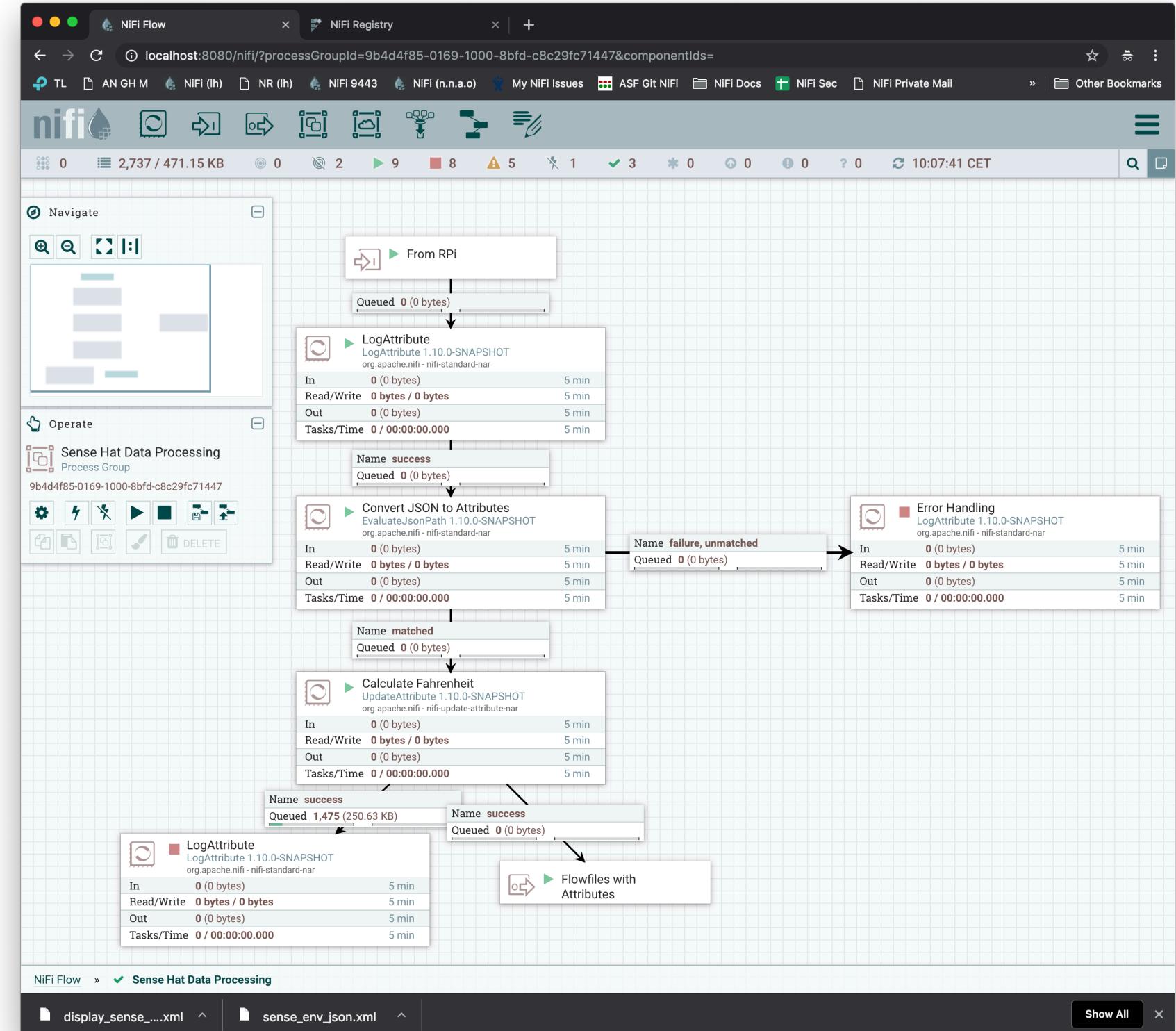
Reading from Sense HAT

- Read JSON output from Python script
- Send to NiFi via Site-to-Site



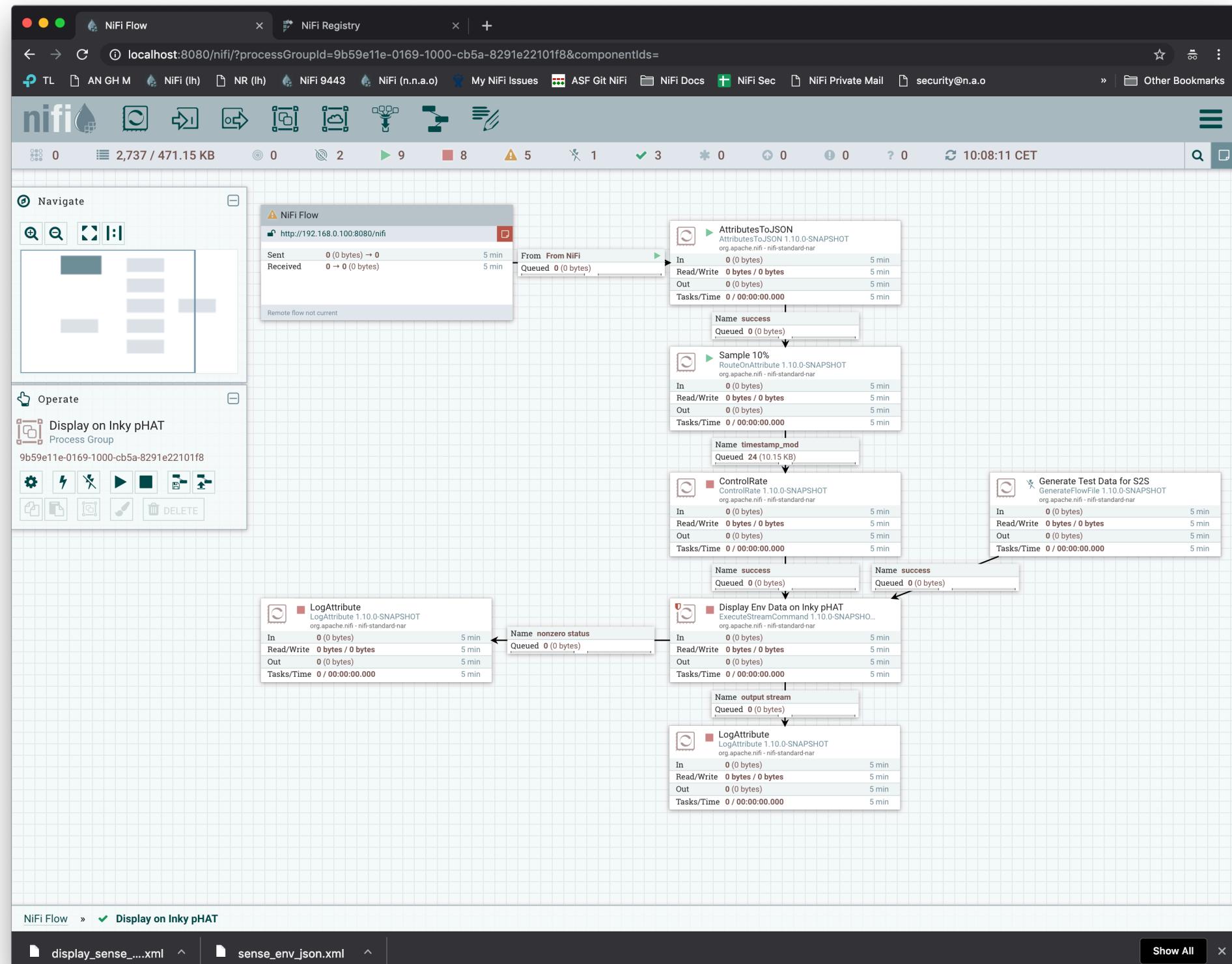
Processing Data

- Ingest JSON from Site-to-Site
- Extract JSON elements to attributes
- Perform conversion

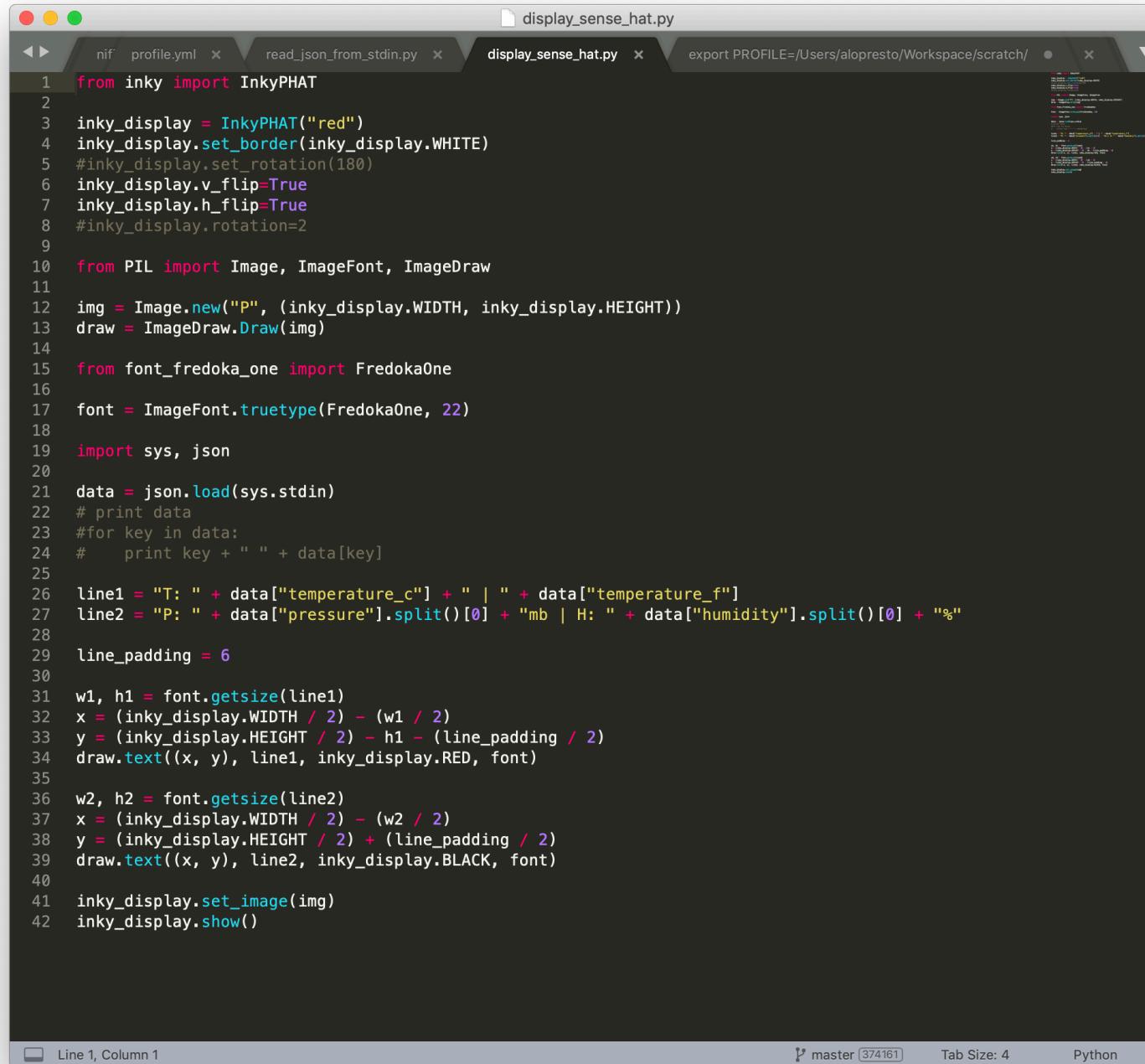


Displaying Data

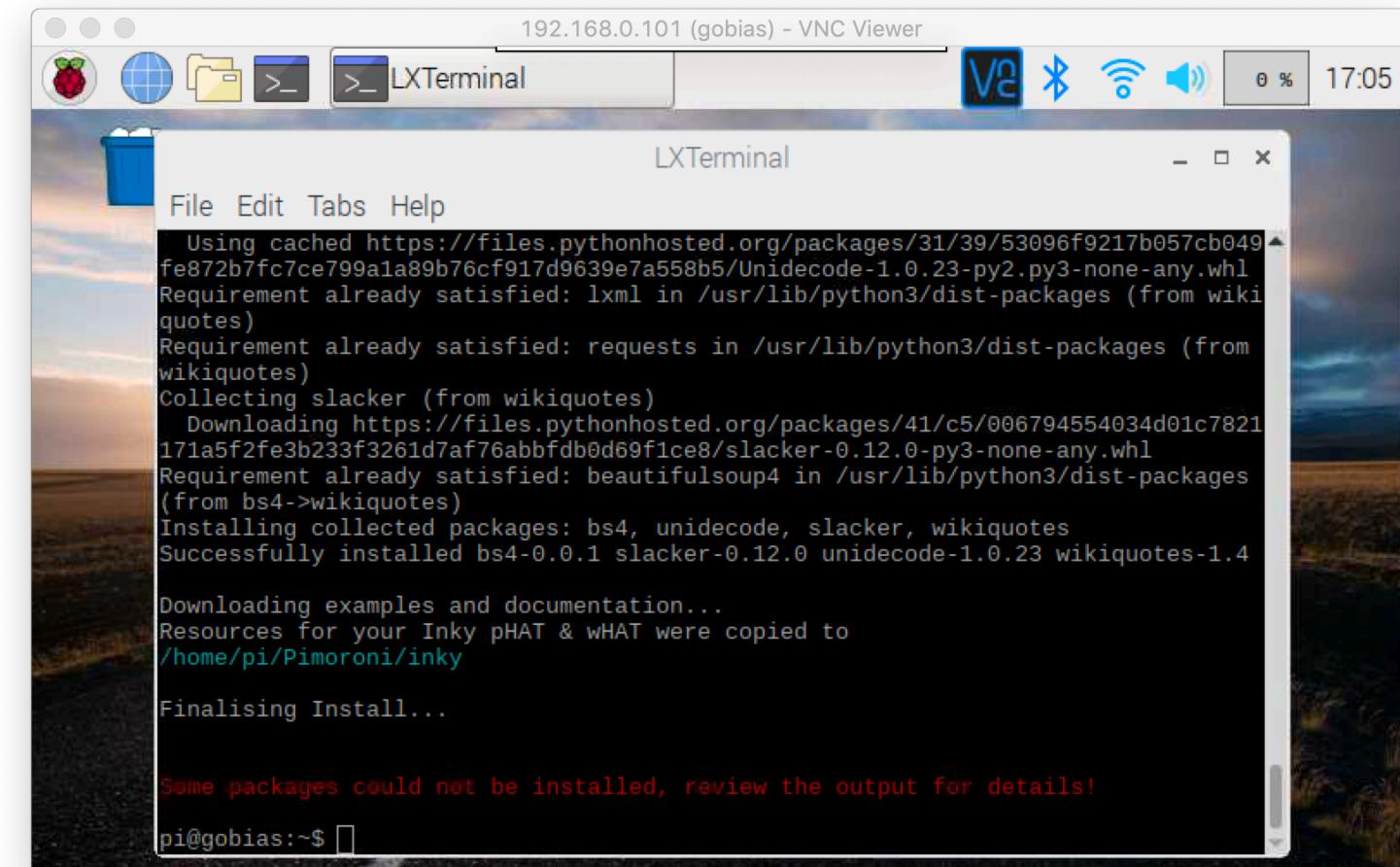
- Receive flowfiles from “Data Center” via Site-to-Site
- Sample 10% (down throttling)
- Safety valve
- Send data to display



Inky pHAT Python Code & VNC



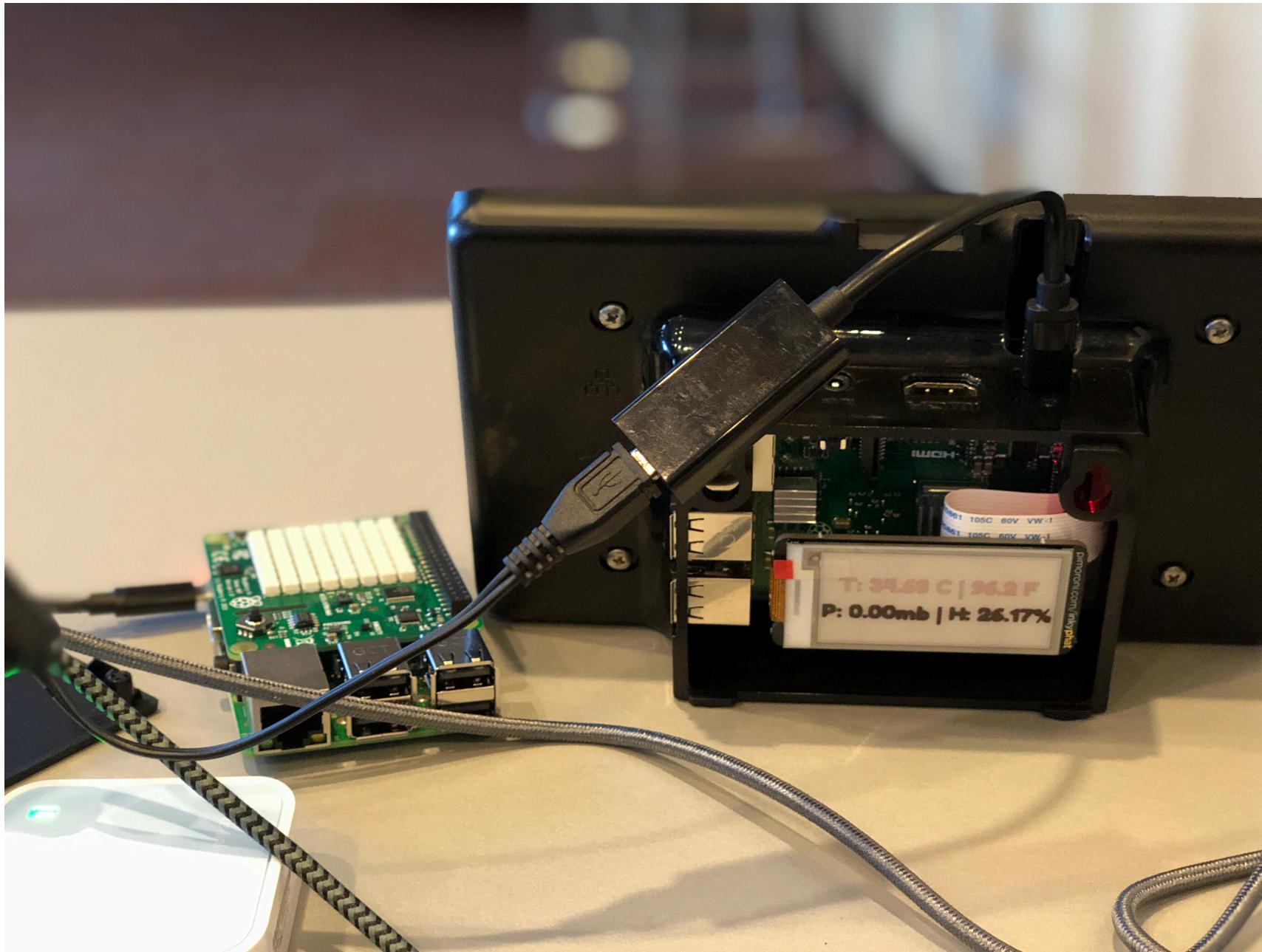
```
display_sense_hat.py
1  from inky import InkyPHAT
2
3  inky_display = InkyPHAT("red")
4  inky_display.set_border(inky_display.WHITE)
5  #inky_display.set_rotation(180)
6  inky_display.v_flip=True
7  inky_display.h_flip=True
8  #inky_display.rotation=2
9
10 from PIL import Image, ImageFont, ImageDraw
11
12 img = Image.new("P", (inky_display.WIDTH, inky_display.HEIGHT))
13 draw = ImageDraw.Draw(img)
14
15 from font_fredoka_one import FredokaOne
16
17 font = ImageFont.truetype(FredokaOne, 22)
18
19 import sys, json
20
21 data = json.load(sys.stdin)
22 # print data
23 #for key in data:
24 #    print key + " " + data[key]
25
26 line1 = "T: " + data["temperature_c"] + " | " + data["temperature_f"]
27 line2 = "P: " + data["pressure"].split()[0] + "mb | H: " + data["humidity"].split()[0] + "%"
28
29 line_padding = 6
30
31 w1, h1 = font.getsize(line1)
32 x = (inky_display.WIDTH / 2) - (w1 / 2)
33 y = (inky_display.HEIGHT / 2) - h1 - (line_padding / 2)
34 draw.text((x, y), line1, inky_display.RED, font)
35
36 w2, h2 = font.getsize(line2)
37 x = (inky_display.WIDTH / 2) - (w2 / 2)
38 y = (inky_display.HEIGHT / 2) + (line_padding / 2)
39 draw.text((x, y), line2, inky_display.BLACK, font)
40
41 inky_display.set_image(img)
42 inky_display.show()
```



```
192.168.0.101 (gobias) - VNC Viewer
LXTerminal
File Edit Tabs Help
Using cached https://files.pythonhosted.org/packages/31/39/53096f9217b057cb049
fe872b7fc7ce799a1a89b76cf917d9639e7a558b5/Unidecode-1.0.23-py3-none-any.whl
Requirement already satisfied: lxml in /usr/lib/python3/dist-packages (from wiki
quotes)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from
wikiquotes)
Collecting slacker (from wikiquotes)
  Downloading https://files.pythonhosted.org/packages/41/c5/006794554034d01c7821
171a5f2fe3b233f3261d7af76abbfdb0d69f1ce8/slacker-0.12.0-py3-none-any.whl
Requirement already satisfied: beautifulsoup4 in /usr/lib/python3/dist-packages
(from bs4->wikiquotes)
Installing collected packages: bs4, unidecode, slacker, wikiquotes
Successfully installed bs4-0.0.1 slacker-0.12.0 unidecode-1.0.23 wikiquotes-1.4
Downloading examples and documentation...
Resources for your Inky pHAT & wHAT were copied to
/home/pi/Pimoroni/inky
Finalising Install...

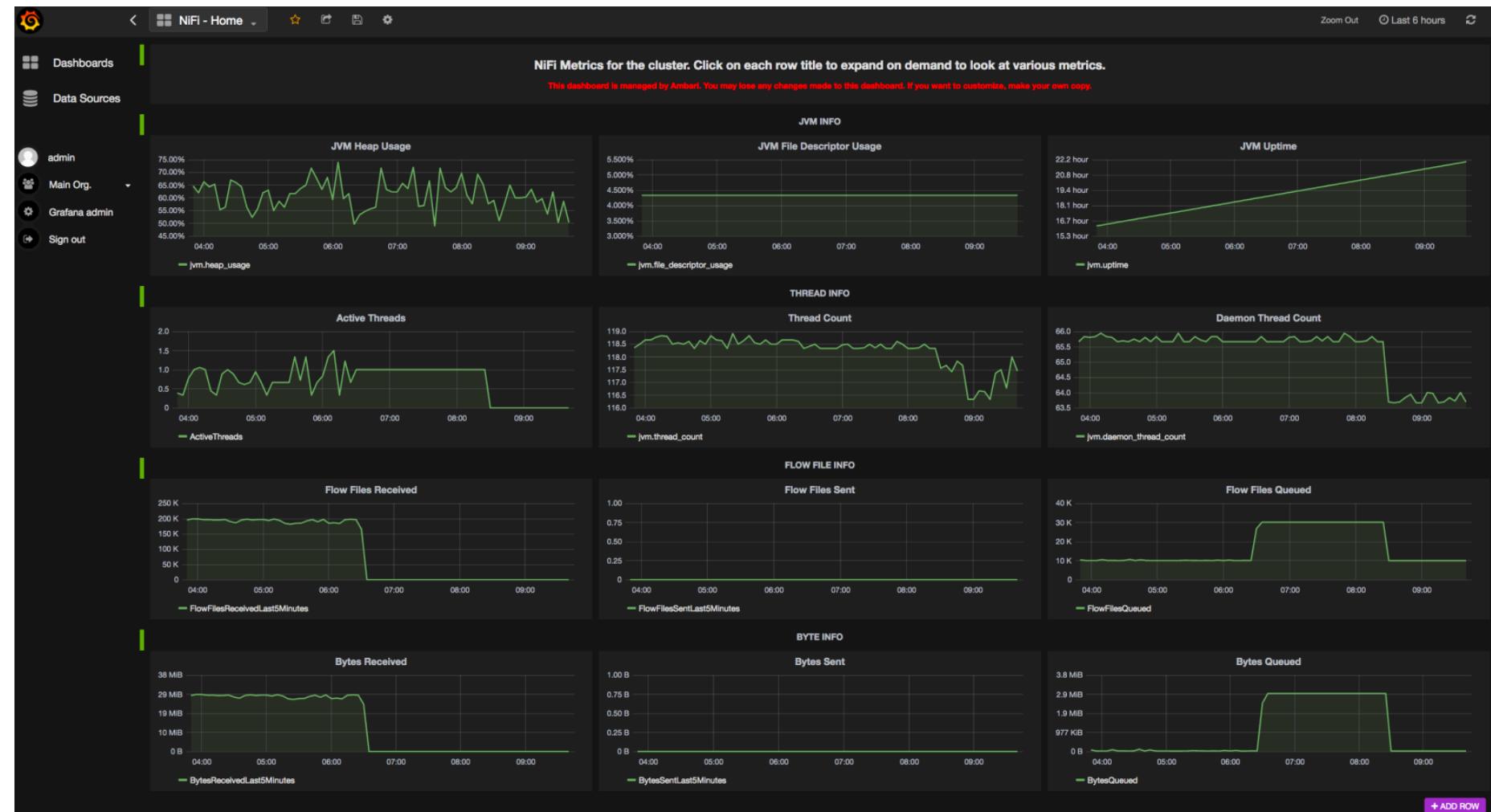
Some packages could not be installed, review the output for details!
pi@gobias:~$
```

Sense Hat & Inky pHAT



Next Steps

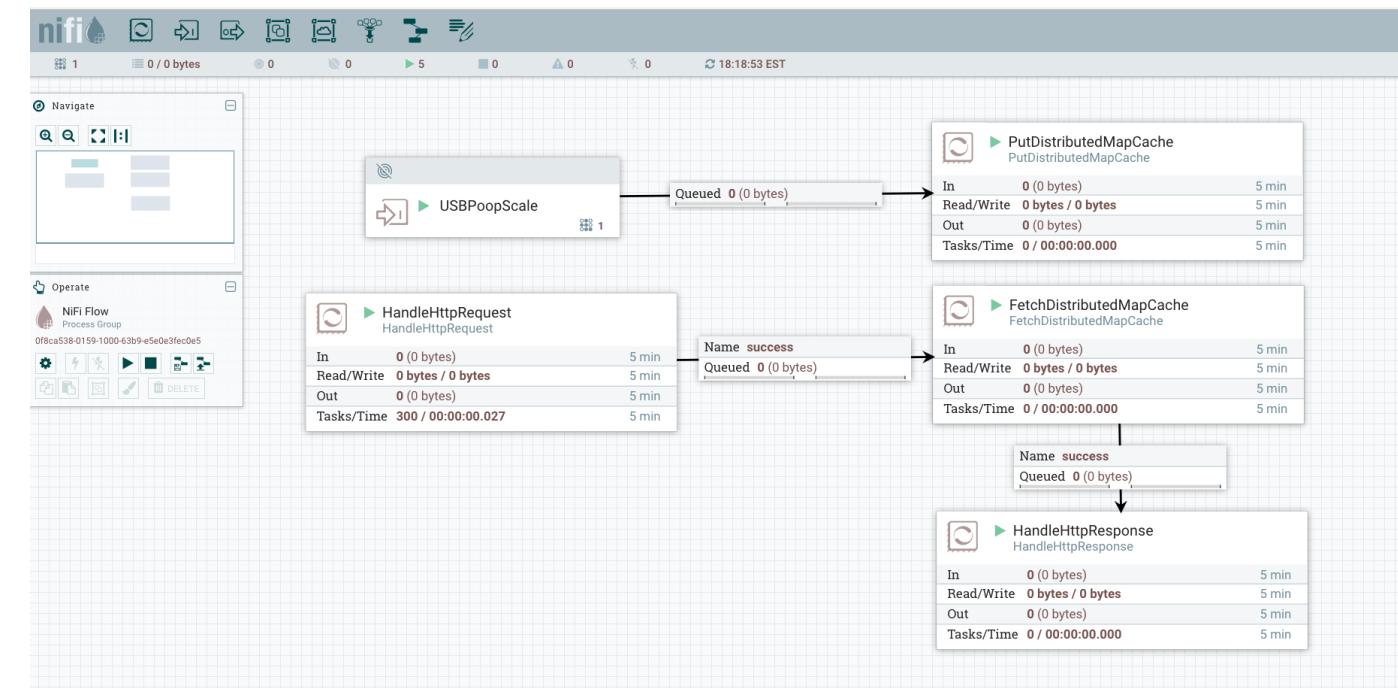
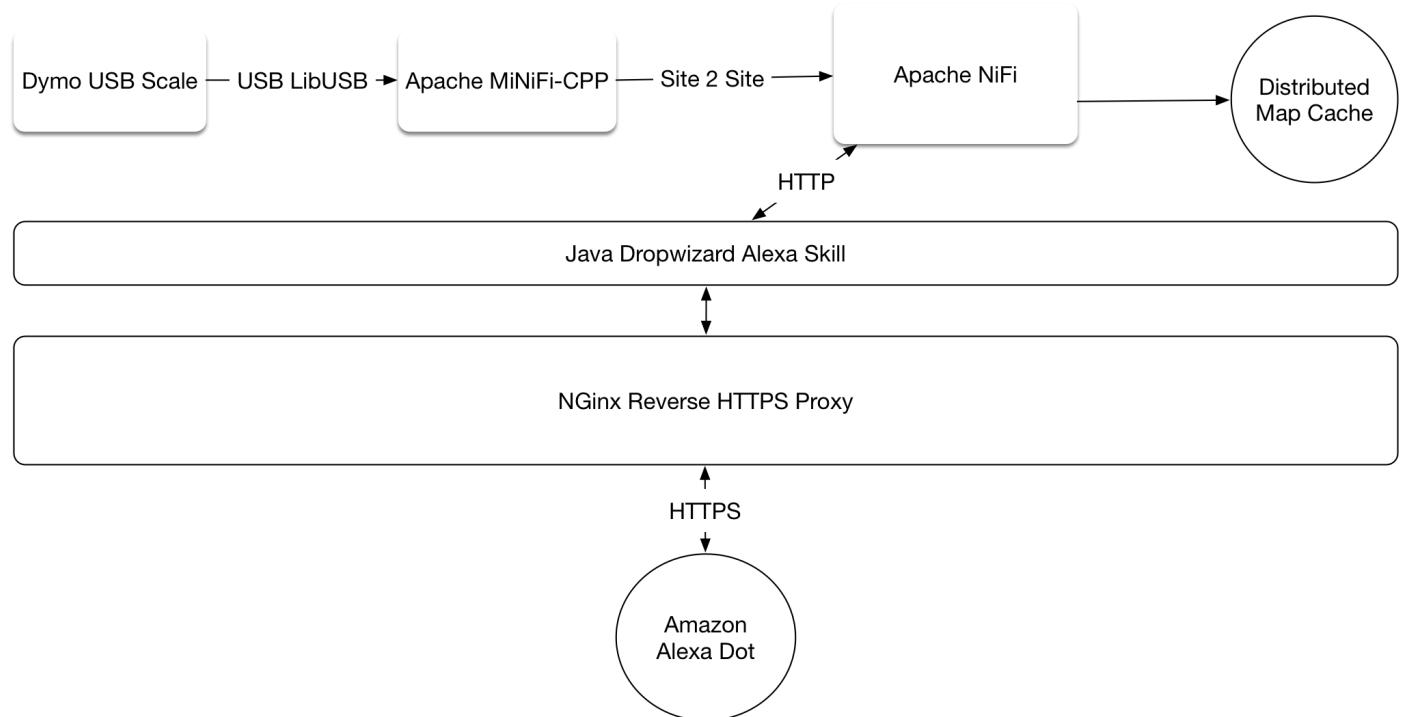
- Add icons & dynamic coloring to elnk display
- Signal traffic via LED output on Sense HAT
- Build simple UI for touchscreen to enable/disable
- Send traffic to Grafana dashboard
- Train machine learning model on optimal frequency for each data point sampling



Community

Community Example

- Jeremy Dyer
 - Alexa + MiNiFi + Dyer 2.0





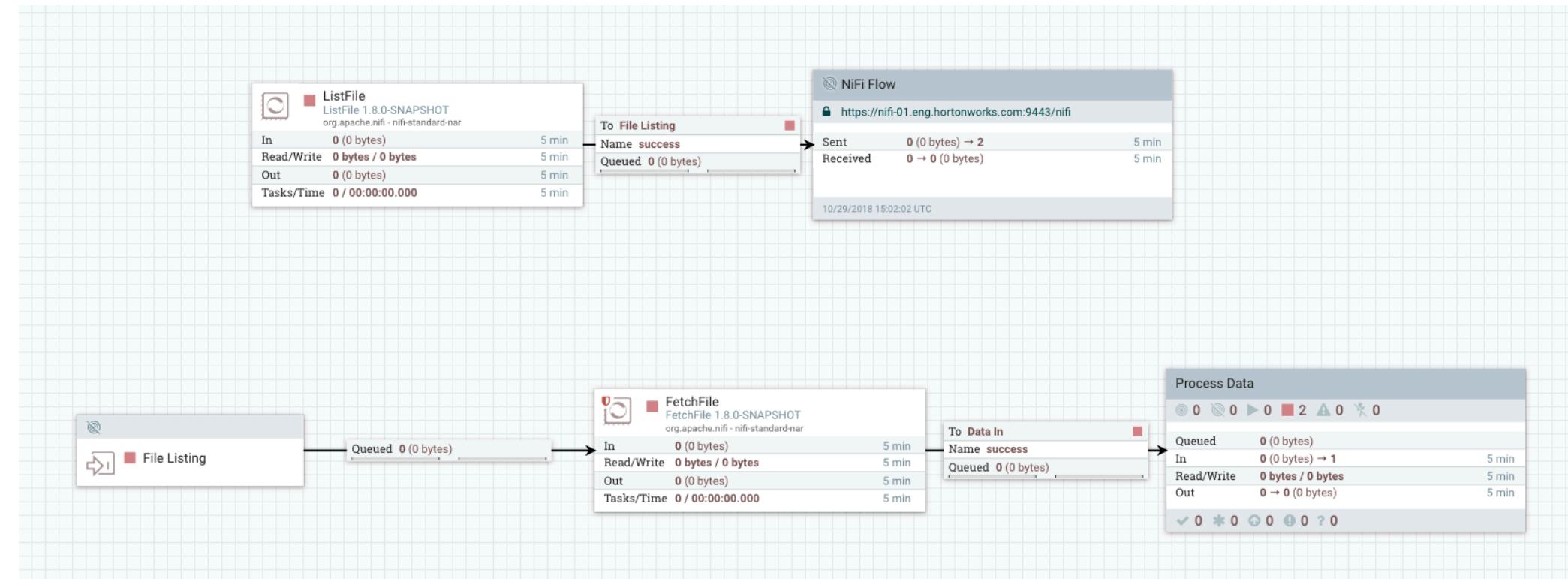
What's Next?

New Features in 1.8.0 - 1.9.1

- Cluster Data Management
 - **Load-Balanced Connections**
 - Node Decommissioning
- SQL results to record format
- Hot-loading NARs
- Auto-infer schema for record processors
- Docker improvements
- TLS Toolkit signing w/ external CA (standalone only)

Previously, on *NiFi Clusters...*

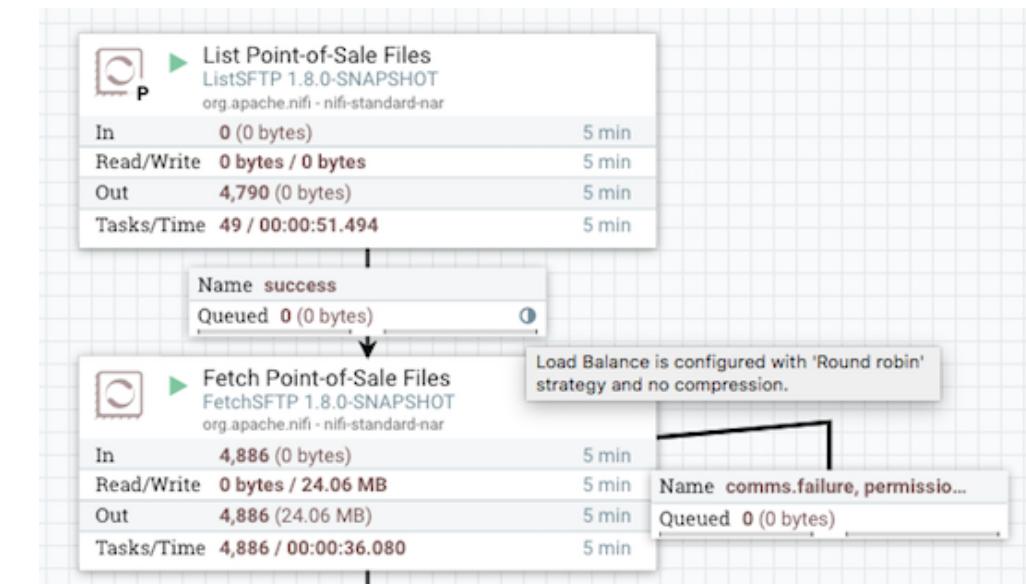
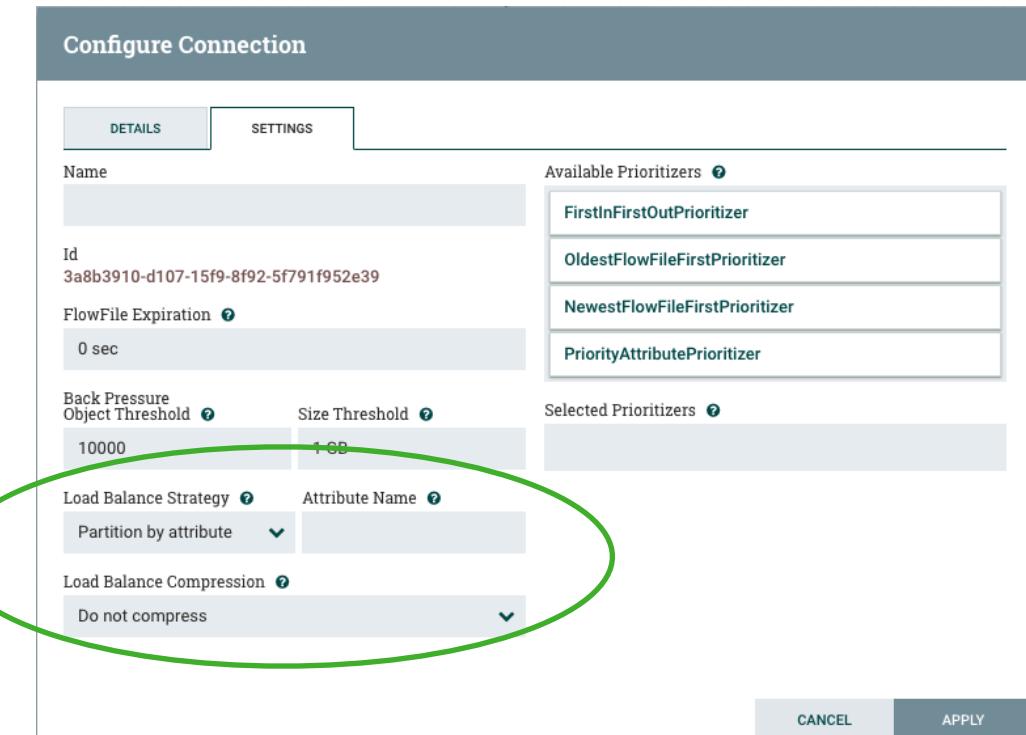
- Ingest activity (FTP, file read, etc.) performed on single node
- Routed via RPG to Input Port in cluster
- Flowfiles distributed across all nodes



[Mark Payne, Load Balancing Data Across a NiFi Cluster](#)

Load-Balanced Connections

- Individual connections can be load-balanced
 - None
 - Round Robin
 - Attribute node affinity
 - Single node

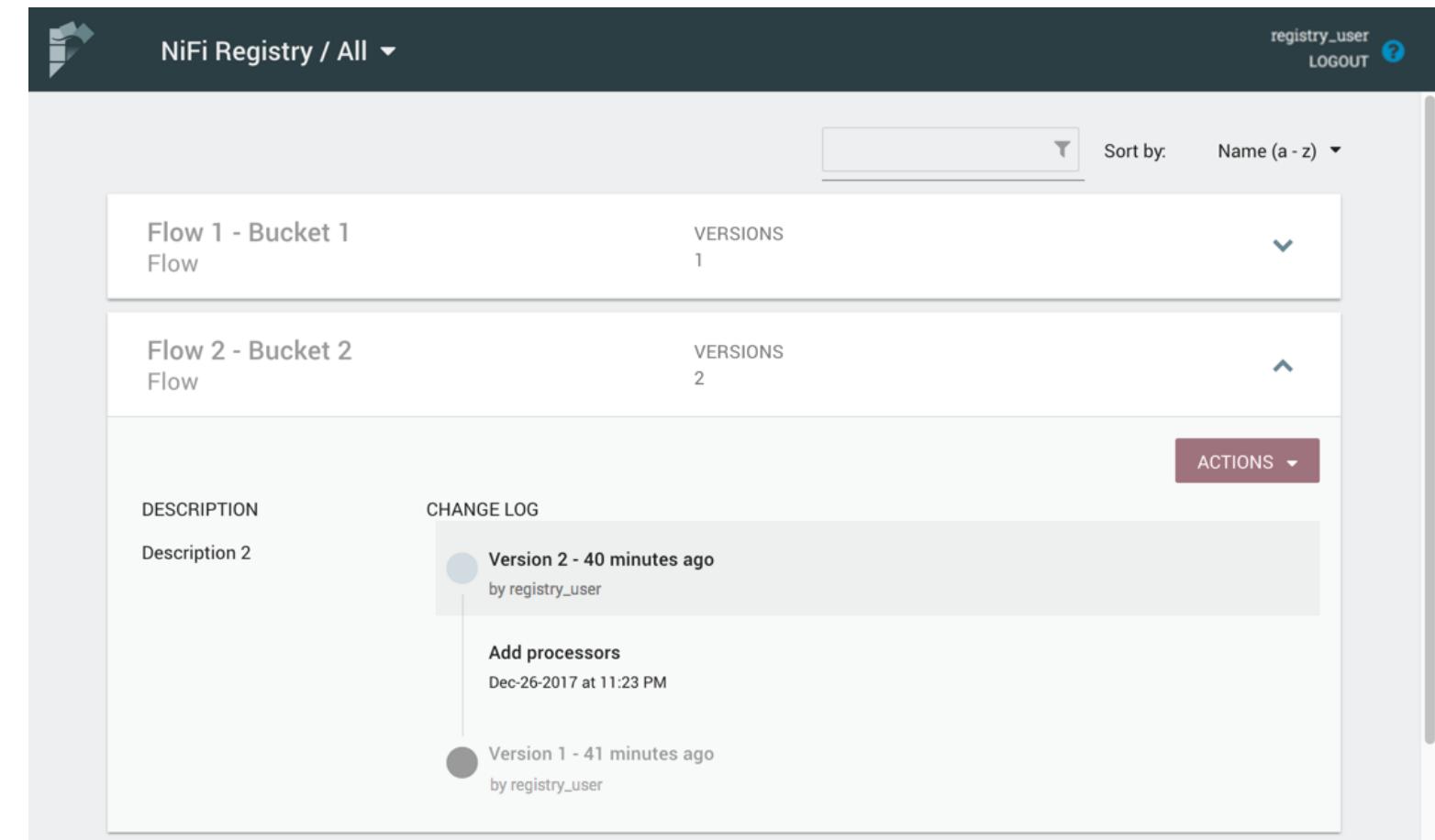


[Mark Payne, Load Balancing Data Across a NiFi Cluster](#)

NiFi Registry for Dataflows

Introducing Apache NiFi Registry 0.3.0

- Previously, flows were exported via XML templates
 - Didn't contain sensitive values
 - Couldn't be updated in-place
 - No tracking system
- NiFi Registry brings asset management as first-class citizen to NiFi
- Flows can be versioned
- Flows can be promoted between environments



NiFi Registry / All ▾

registry_user LOGOUT ?

Sort by: Name (a - z) ▾

Flow 1 - Bucket 1
Flow

VERSIONS 1

Flow 2 - Bucket 2
Flow

VERSIONS 2

DESCRIPTION Description 2

CHANGE LOG

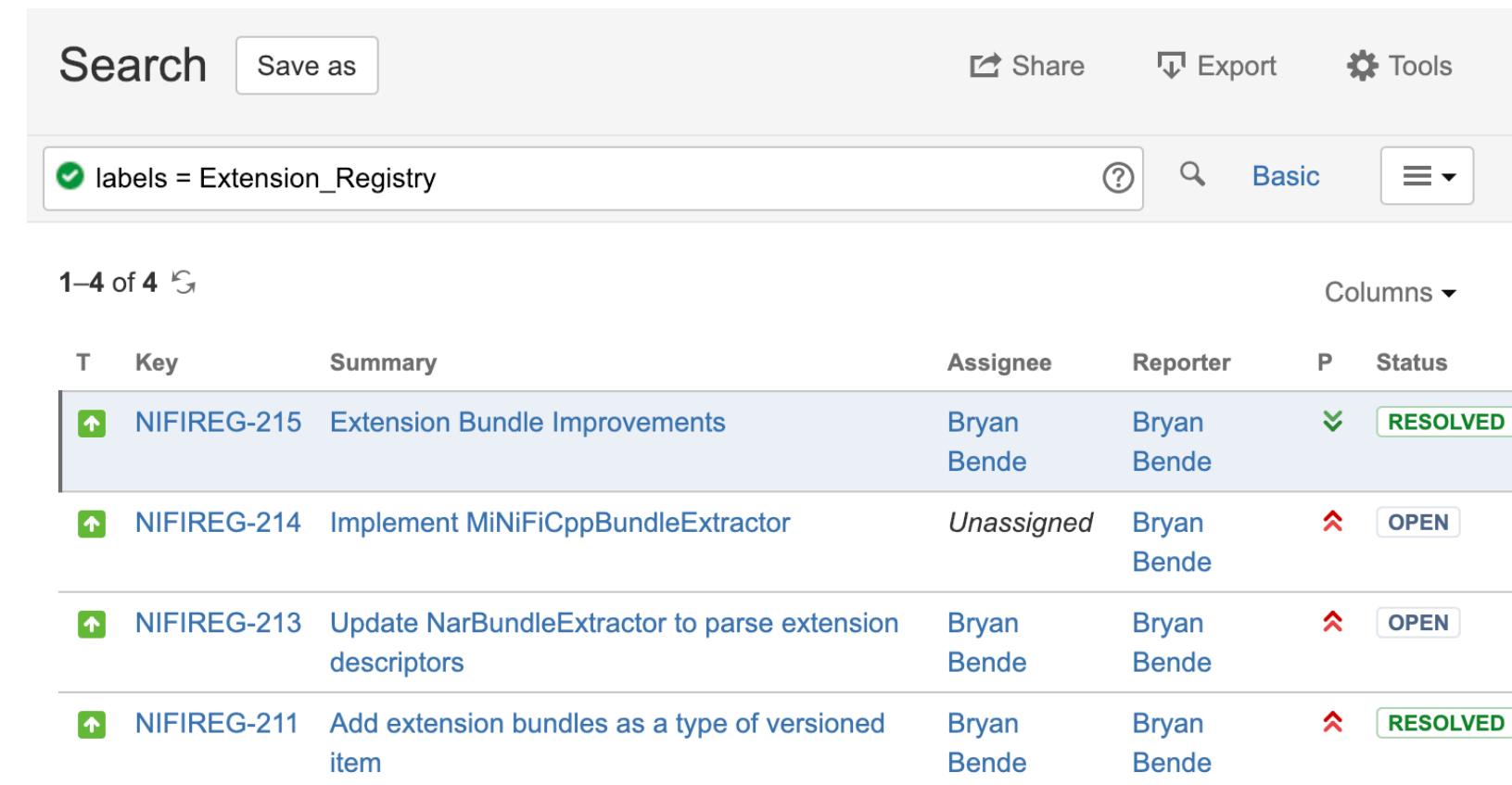
- Version 2 - 40 minutes ago by registry_user
- Add processors Dec-26-2017 at 11:23 PM
- Version 1 - 41 minutes ago by registry_user

ACTIONS ▾

Roadmap

Roadmap

- Command & Control Architecture
- Extension Registry



The screenshot shows a Jira search results page with the following details:

Search: labels = Extension_Registry

Results: 1–4 of 4

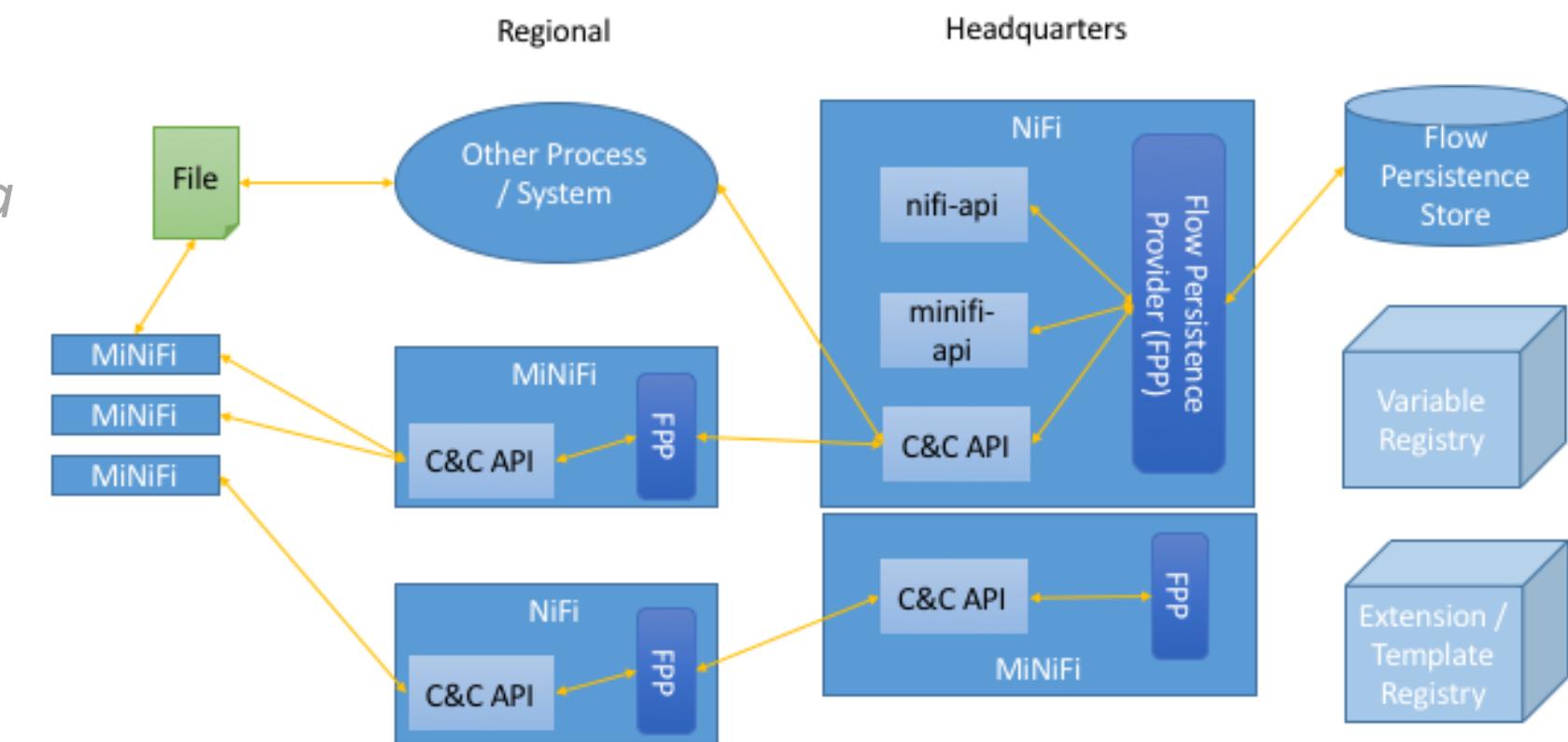
Table Headers: T, Key, Summary, Assignee, Reporter, P, Status

Table Data:

T	Key	Summary	Assignee	Reporter	P	Status
↑	NIFIREG-215	Extension Bundle Improvements	Bryan Bende	Bryan Bende	✓	RESOLVED
↑	NIFIREG-214	Implement MiNiFiCppBundleExtractor	Unassigned	Bryan Bende	✗	OPEN
↑	NIFIREG-213	Update NarBundleExtractor to parse extension descriptors	Bryan Bende	Bryan Bende	✗	OPEN
↑	NIFIREG-211	Add extension bundles as a type of versioned item	Bryan Bende	Bryan Bende	✗	RESOLVED

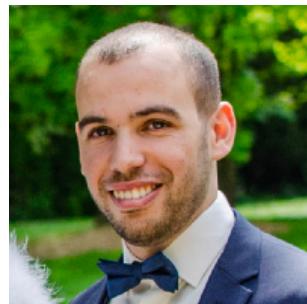
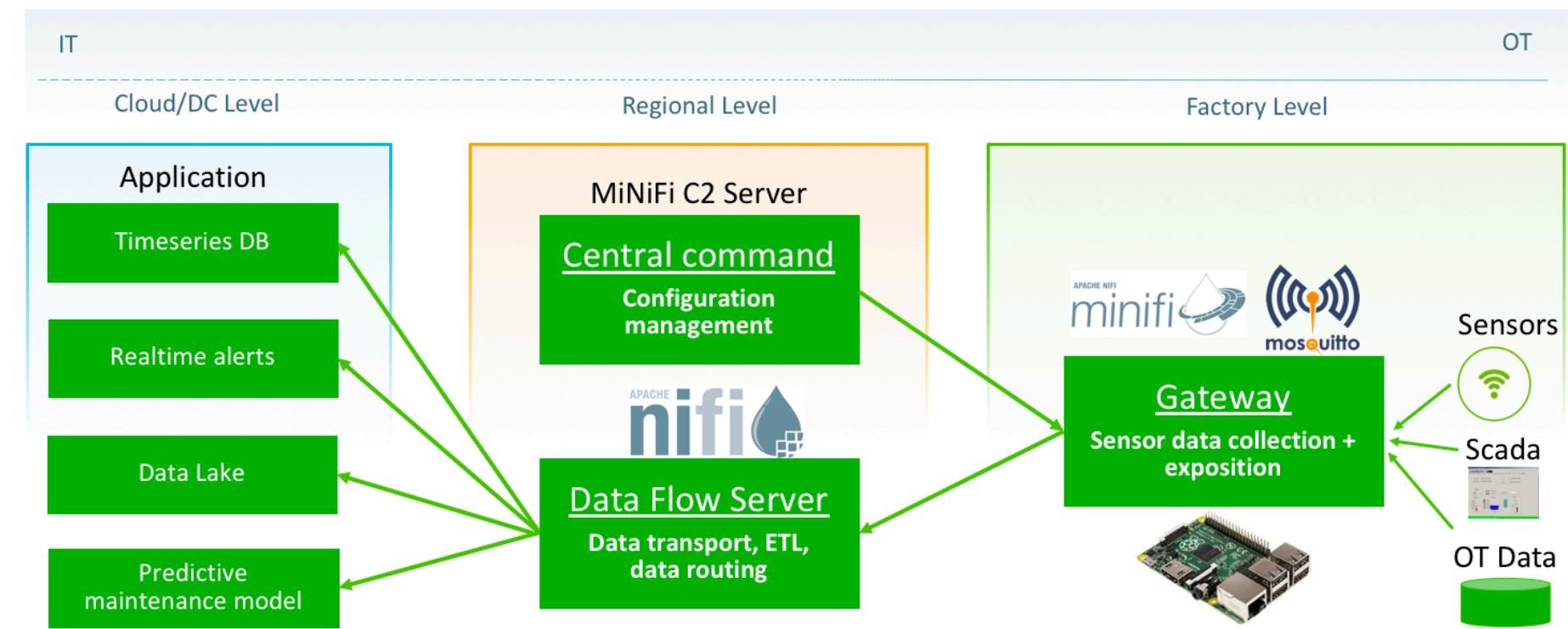
MiNiFi Command & Control Architecture

- Flow Versioning
 - Develop flows for class of MiNiFi instances
- Command & Control (C2) API (*in Java master*)
 - FileChangeIngestor
 - RestAPIIngestor
 - PullHTTPIngestor



MiNiFi C2 + Raspberry Pi

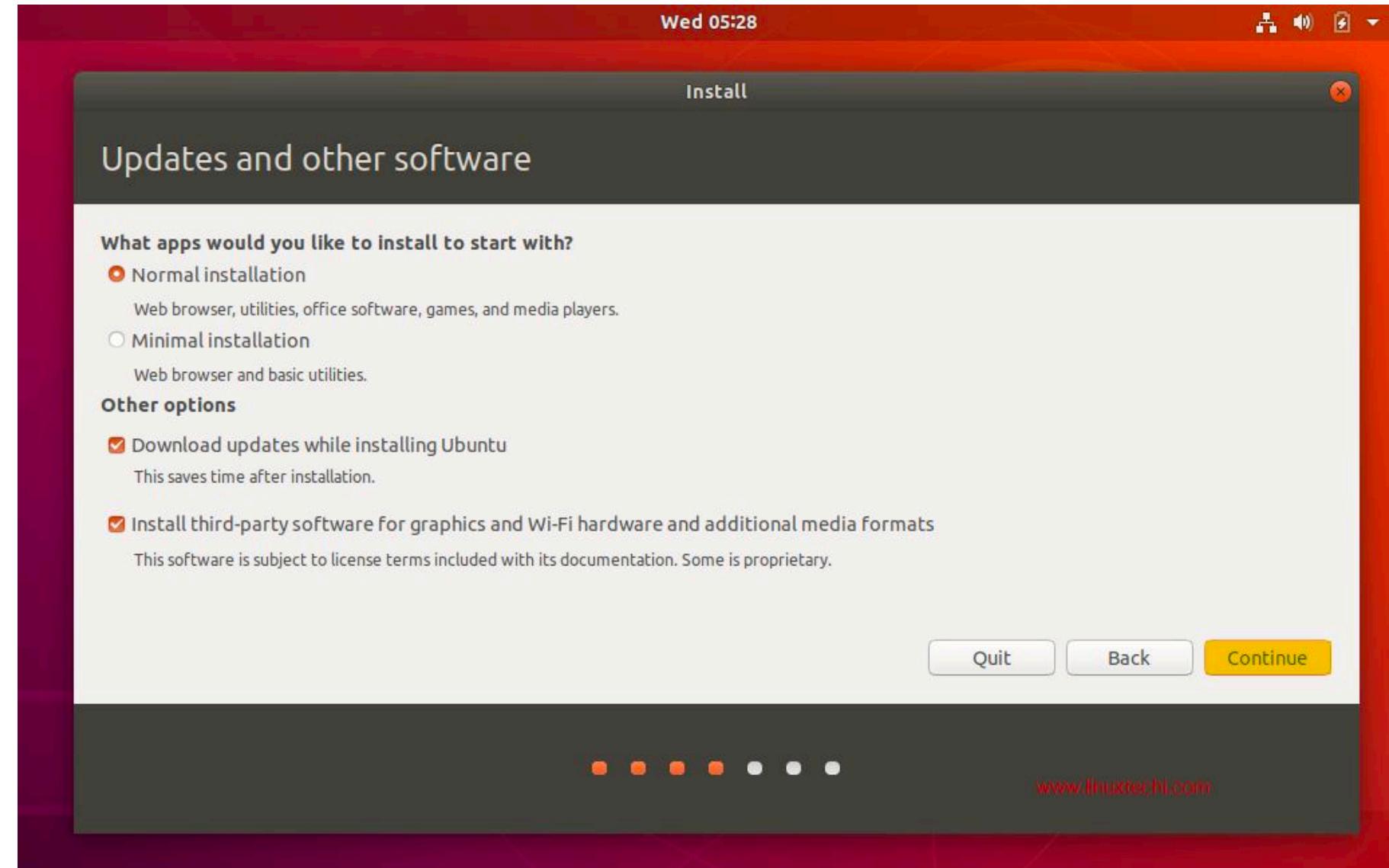
- Abdelkrim Hadidj
- Step-by-step walkthrough
 - deploying services
 - configuring endpoints
 - developing flows



[How to build an IIoT system using Apache NiFi,
MiNiFi, C2 Server, MQTT and Raspberry Pi](#)

Extension Registry

- Uses NiFi Registry to manage extensions (“narballs”)
- Allows faster download, build, and deployment
 - *Think Ubuntu Server vs. Desktop*
- Create, upload, and download functional bundles



LinuxTechi.com

- [NiFi Wiki - Extension Registry Feature Proposal](#)
- [NiFi Dev Mailing List - Discussion of implementation](#)

New Announcements

- NiFi 1.9.1 – 16 Mar 2019 (167+ Jiras)
 - New processors (Kudu, Slack)
 - Hot-loading NARs
 - Security HTTP headers
 - Record processors automatically infer schema
- MiNiFi C++ 0.6.0 – Voting now...
- MiNiFi Java 0.5.0 – 7 July 2018
- NiFi Registry 0.3.0 – 25 Sept 2018



Learn more and join us

Apache NiFi site

<https://nifi.apache.org>

Subproject MiNiFi site

<https://nifi.apache.org/minifi/>

Subscribe to and collaborate at

dev@nifi.apache.org

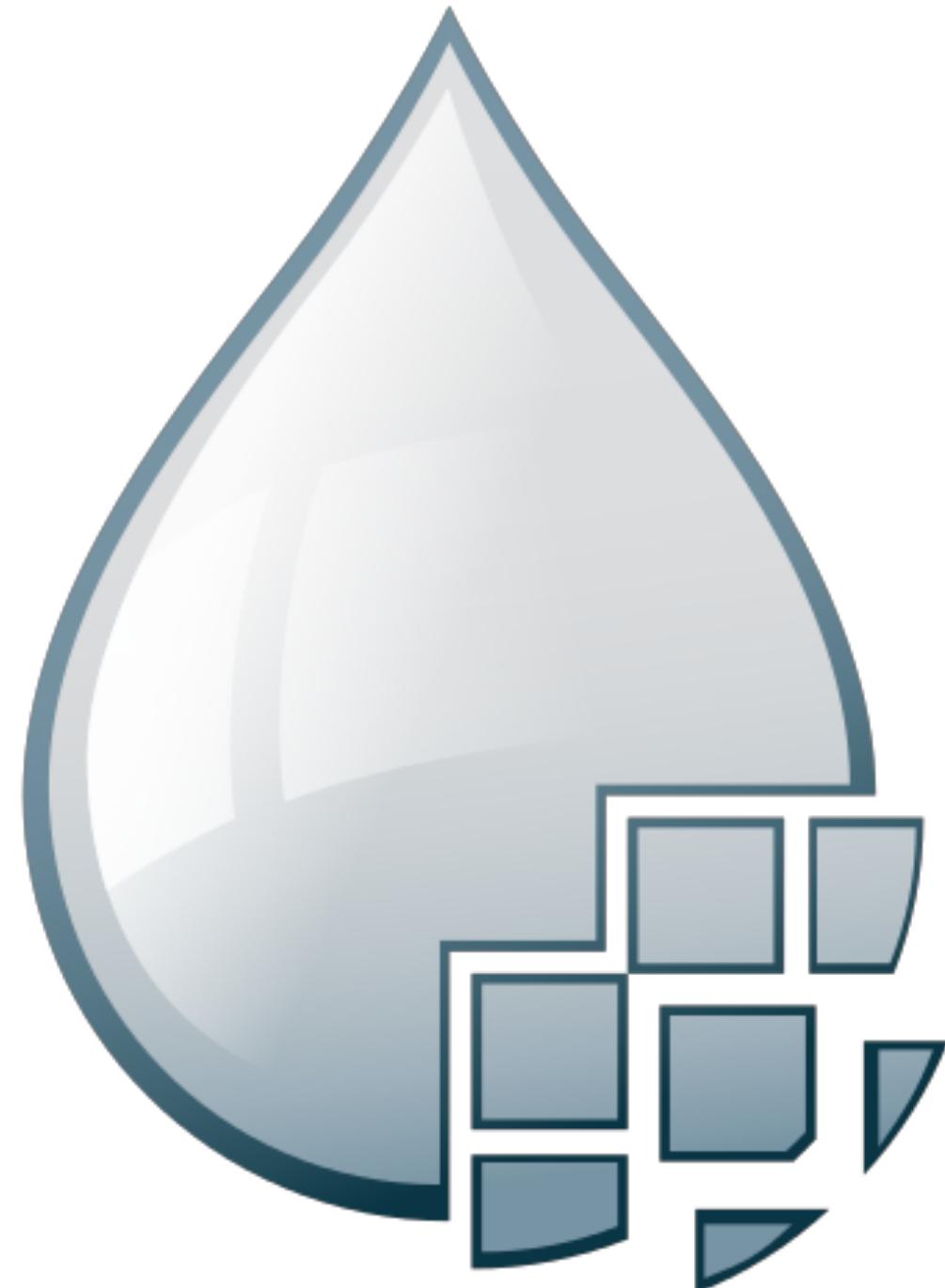
users@nifi.apache.org

Submit Ideas or Issues

<https://issues.apache.org/jira/browse/NIFI>

Follow us on Twitter

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



More NiFi This Week

Title	Date		Speakers	Room
Cloudera Data in Motion Meetup - Future of Data Barcelona (free)	Tuesday 3/19	1800 - 2000	Tim Spann, Dan Chaffelson, Andy LoPresto	127-128
Apache NiFi Crash Course		1400 - 1600	Nathan Gough, Andy LoPresto	111
Data Acquisition Automation for NiFi in a Hybrid Cloud environment – the Path towards DataOps	Wednesday 3/20	1450 - 1530	Arda Basar, Ivan Georgiev	129-130
IoT, Streaming, and Dataflow Birds of a Feather		1600 - 1730	Tim Spann, Dan Chaffelson, Purnima Reddy Kuchikulla, Andy LoPresto	129-130
Addressing Challenges with IoT Edge Management		1100 - 1140	Dinesh Chandrasekhar	127-128
Intelligently Collecting Data at the Edge – Intro to Apache MiNiFi		1150 - 1230	Andy LoPresto	127-128
Edge to AI: Analytics from Edge to Cloud with Efficient Movement of Machine Data	Thursday 3/21	1400 - 1440	Tim Spann	129-130
BYOP: Custom Processor Development with Apache NiFi		1600 - 1640	Andy LoPresto	131-132
Apache Deep Learning 201		1650 - 1730	Tim Spann	122-123
Platform for the Research and Analysis of Cybernetic Threats		1650 - 1730	Monica Franceschini	124-125



Thank you

alopresto@hortonworks.com | alopresto@apache.org | [@yolopey](https://twitter.com/yolopey)
github.com/alopresto/slides