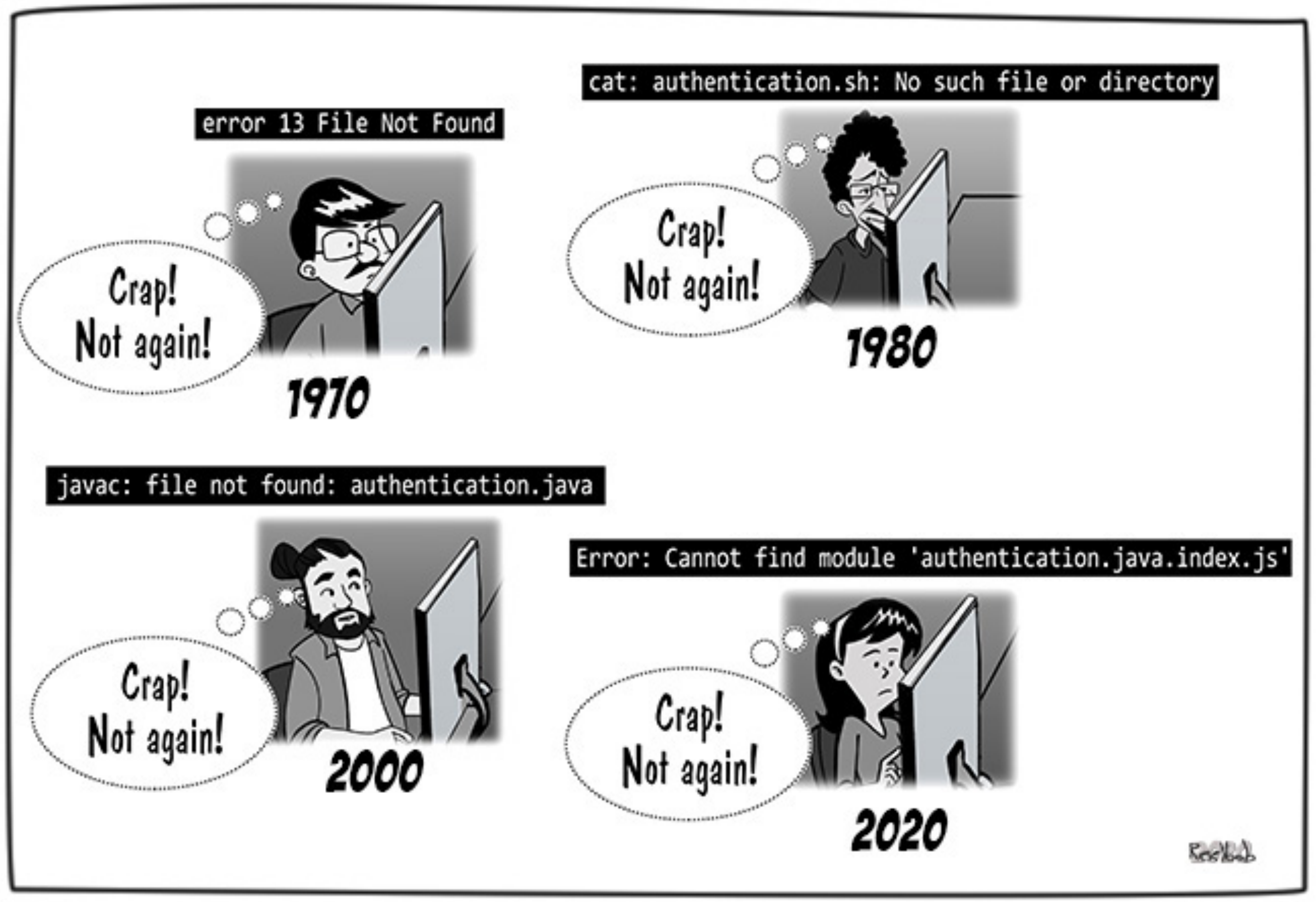
IOT AND RFID - RESTFUL VS. LLRP



```
error 13 File Not Found
cat: authentication.sh: No such file or directory
javac: file not found: authentication.java
Error: Cannot find module 'authentication.java.index.js'
```

BACKGROUND

- How did we get here?
  - Customer requests
  - Lots of software visitors
    - MQTT, Azure, Green Grass, etc…
- Acknowledgements
  - Health Care Logistics → RESTful api
    - Kurt Wolfe, Stephen Stock, Kevin Watts, and Tiffany Timmons
  - Reynolds and Reynolds → LLRP
    - Stephen Hildebrand

WHAT ARE THEY?

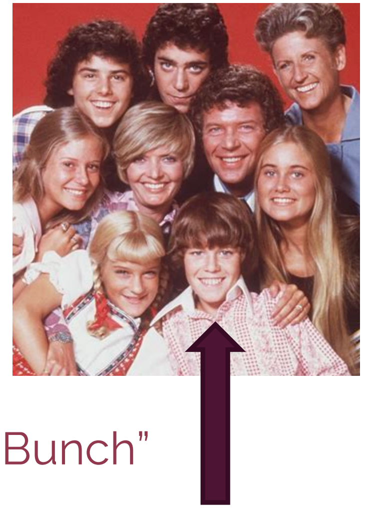
	LLRP	RESTful
What is it	Low Level Reader Protocol The UHF RFID standard.	REpresentational State Transfer It's a software architecture style.
Purpose	Interoperability between RFID readers	Interoperability between any internet device
Introduced in	2007	2000
Introduced by	EPC Global (now GS1)	Roy Fielding, UC Irvine
Evolution	Oct 2010 to V1.1, managed by GS1	Continuous, managed by OpenAPI
Characteristics	• Very low level • Over 100 low-level commands	High level client/server **stateless** web service

COMMUNITY

SensThys

- RESTful API
  - General IoT
  - Wide application
- Google "RESTful API"
  - 34,400,000 results
- Google "RESTful API programmers"
  - 89,400,000 results
- Google "Brad Pitt"
  - 116,000,000 results

- LLRP
  - UHF RFID Reader MFG
  - UHF RFID Integrators
- Google "LLRP"
  - 142,000 results
  - 0.4% of RESTful
- Google "LLRP programmers"
  - 7,100 results
  - 0.0008% of RESTful
- Google "Mike Lookinland"
  - Bobby Brady of the "The Brady Bunch"
  - 84,000 results



HIGH LEVEL PROS AND CONS

	LLRP	RESTful
Advantages	• Current UHF RFID standard • Interoperability between different readers	• IoT & computing devices Interoperability • Standard payloads such as HTML, XML or JSON • Networking capability • Programmer availability
Disadvantages	• Fixed reader only • Most implementations are custom • Very low-level • Lack of skilled programmers	• Limited adoption in the UHF RFID community

"I'M NOT SURE WHICH DEAD LANGUAGE TO TAKE NEXT SEMESTER - LATIN, MIDDLE ENGLISH, OR LLRP."

PROGRAMMER'S VIEW - LLRP

Highlights

- GS1, V1.1 (2010)
- Current RFID popularity

Lowlights

- C, C++, Java, Python
- Programming effort
- RFID-centric

Recommended for….

- Heavy-duty interface to reader
- Mixed-reader environments

NOT recommended for…

- Browser-based apps
- Mobile applications

PROGRAMMER'S VIEW - RESTFUL

SensThys

Highlights

- OpenAPI standards body
- Development environment
- HTML, XML or JSON payloads
- Strong security

Lowlights

- Not optimal for high sensor data flow

Recommended for….

- Web or mobile platforms
- Mixed-sensor deployments
- User-facing applications

NOT recommended for…

- Very high data throughput

CODE EXAMPLE

SET POWER ON ANTENNA 3

- RESTful: An AJAX POST call.

```
url: /stapi/v0/ant/pwr
data: [ {
  “antenna_id”: 3,
  “read_power”: 30
  } ]
response_handler: {
  …
  }
```

- LLRP

```
// Create reader object:
LLRP_Reader *rdr = new LLRP_Reader();

// Create LLRP Configuration object:
LLRP_ConfigMsg *msg = new
   LLRP_ConfigMsg();

// Initialize LLRP_ConfigMsg:
   msg->SetAntennaReadPower(3, 30);

// Send LLRP_ConfigMsg and receive status
// as to whether succeeded or not:
   LLRP_Status status = rdr->send(msg);
```

Analogy: C# vs Assembler

RESTFUL CODE GENERATION

- "Swagger Codegen can simplify your build process by generating server stubs and client SDKs for any API, defined with the OpenAPI (formerly known as Swagger) specification"

- Translation – RESTful takes an API and *generates key code*, allowing developers to focus on customer experience as opposed to the nitty gritty stuff

HEAD-TO-HEAD QUESTIONS…

- Which is better for reader control?
  - No substantive difference
- Which is better for general sensor (IoT) control?
  - REST APIs are more suited to web and mobile apps
- Lines of code – which requires more coding?
  - RESTful wins for simplicity
  - RESTful also generates code for certain tasks
- Which has better resources and documentation?
  - RESTful community is HUGE as compared to LLRP
  - RESTful is a set of URL and data encodings that ride on top of standard HTTP actions
  - LLRP is fully documented & with support libraries

HEAD-TO-HEAD QUESTIONS…

- Is one better for handling a lot of sensors?
  - RESTful wins since the web server software handles scaling problem
  - LLRP better for UHF RFID reader if skilled with highly threaded server

- Is one better for cloud-based sensor control?
  - RESTful is designed to handle the security issues around HTTP traffic

- Why isn't MQTT being discussed here?
  - MQTT is a data format specification
  - LLRP is traditional for RFID
  - RESTful is easier to code

OTHER QUESTIONS

- Is “RESTful” transportable from one device to another?
  - If application written in Javascript → Yes
  - At the level of communicating with different types of sensors → No
  - Custom coding to LLRP capabilities can extend

- Are there costs associated with LLRP? RESTful?
  - No

- Why is LLRP managed by barcode guys?
  - Because RFID was seen as a barcode replacement

RAIN RFID

- RAIN Communication Interface (RCI)
  - Designed to address LLRP shortcomings
    - Still targets UHF RFID
    - NOT low level
    - VERY light
  - Claims:
    - Simple to use and implement
    - Use any serial interface (USB, TCP/IP, Bluetooth, RS-232, etc.)
    - Extendable to allow vendors to add their own features

More information https://rainrfid.org/technology/rain-communication-interface-rci/

SUPPLIERS

	Proprietary	LLRP	RESTful	RAIN RCI
SensThys	✓	✓	✓	✗
Alien	✓	✓	✗	✗
CAEN	✓	✗	✗	✗
Impinj	✓	✓	✓	✗
Jadak	✓	✓	✗	✗
Kathrein	✓	✓	✗	✗
Nordic ID	✓	✓	✗	✗
Zebra	✓	✓	✗	✓

OTHER CONSIDERATIONS - HANDHELD RFID

SensThys

- Monolithic devices migrated from Windows CE to Android
- Sled devices up latest Android and iOS
- LLRP not used for handheld
- RESTful can work fine on handheld
  - CS108 → Java, Objective C, and C#

THANK YOU TO OUR CUSTOMERS

- 2019 – 55% growth
- 2020 – 98% growth
- We welcome your problems, it is how we grow.

NEXT WEBINAR
An epic announcement

Omission to do what is necessary seals a commission to a blank of danger;

And danger, like an ague, subtly taints even then when we sit idly in the sun.

William Shakespeare
(1564-1616)