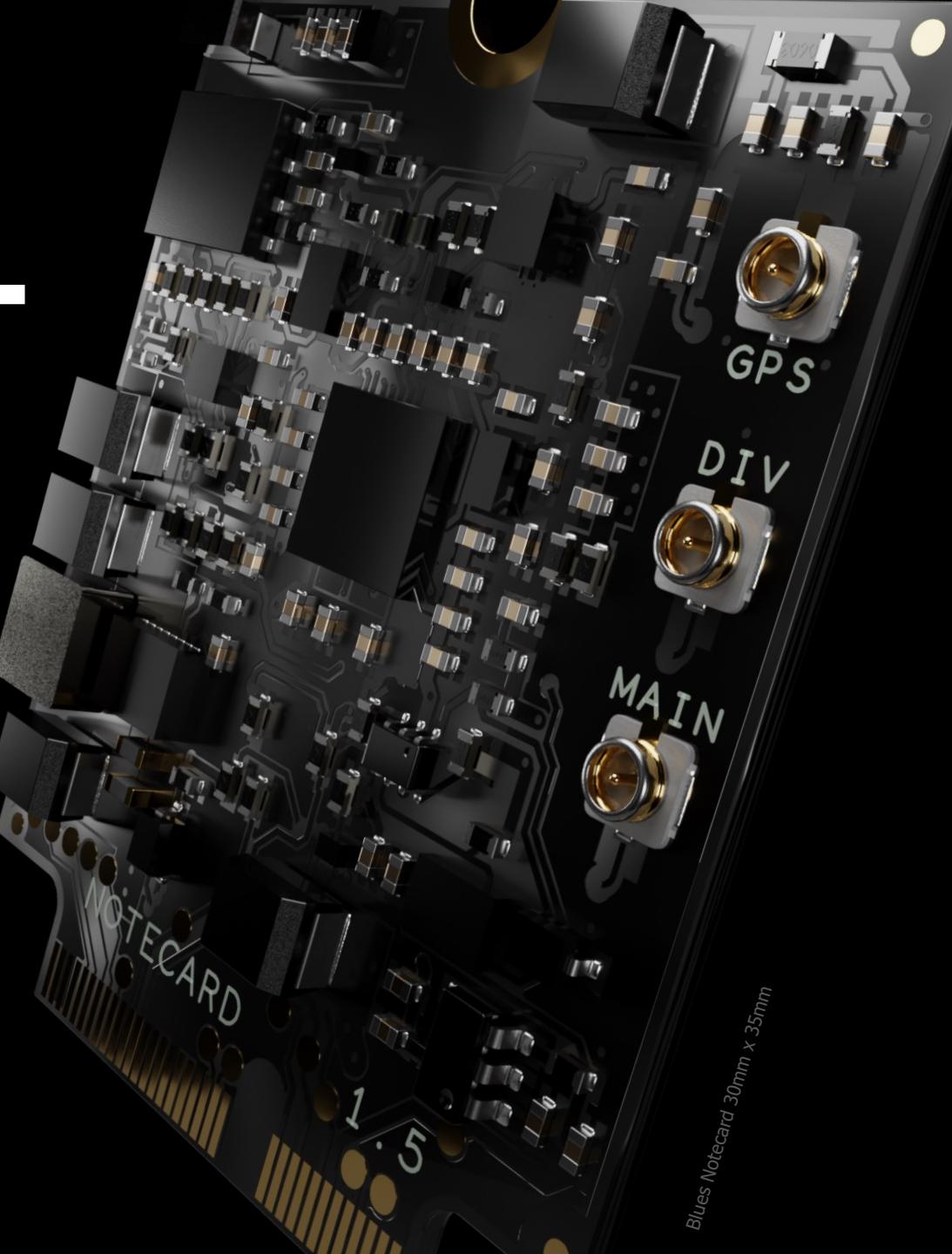


Getting Started with Wireless IoT

Rob Lauer & TJ VanToll

Developer Relations



Blues NoteCard 30mm x 35mm



Rob Lauer

Director of Developer Relations

[@RobLauer](#)



TJ VanToll

Principal Developer Advocate

[@TJVanToll](#)

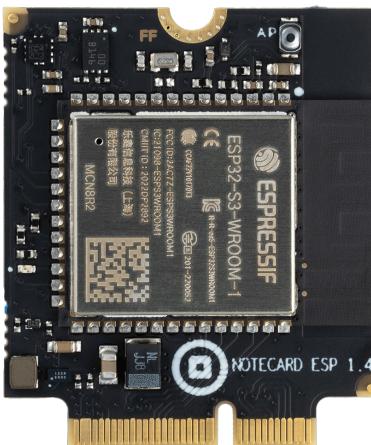
“

Complexity kills. It sucks the life out of developers, it makes products difficult to plan, build, and test.

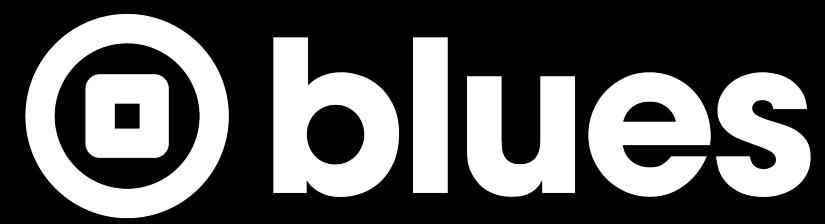
Ray Ozzie - Blues CEO

Today's Agenda

- Intro to the Blues Ecosystem
- Hands-on Demonstration of Wireless IoT

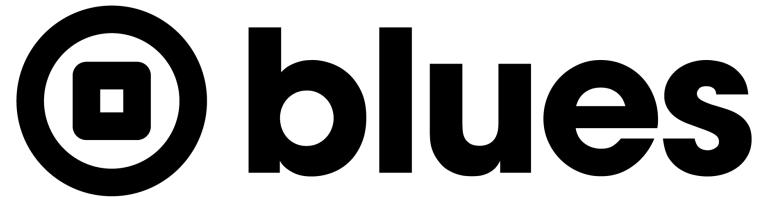








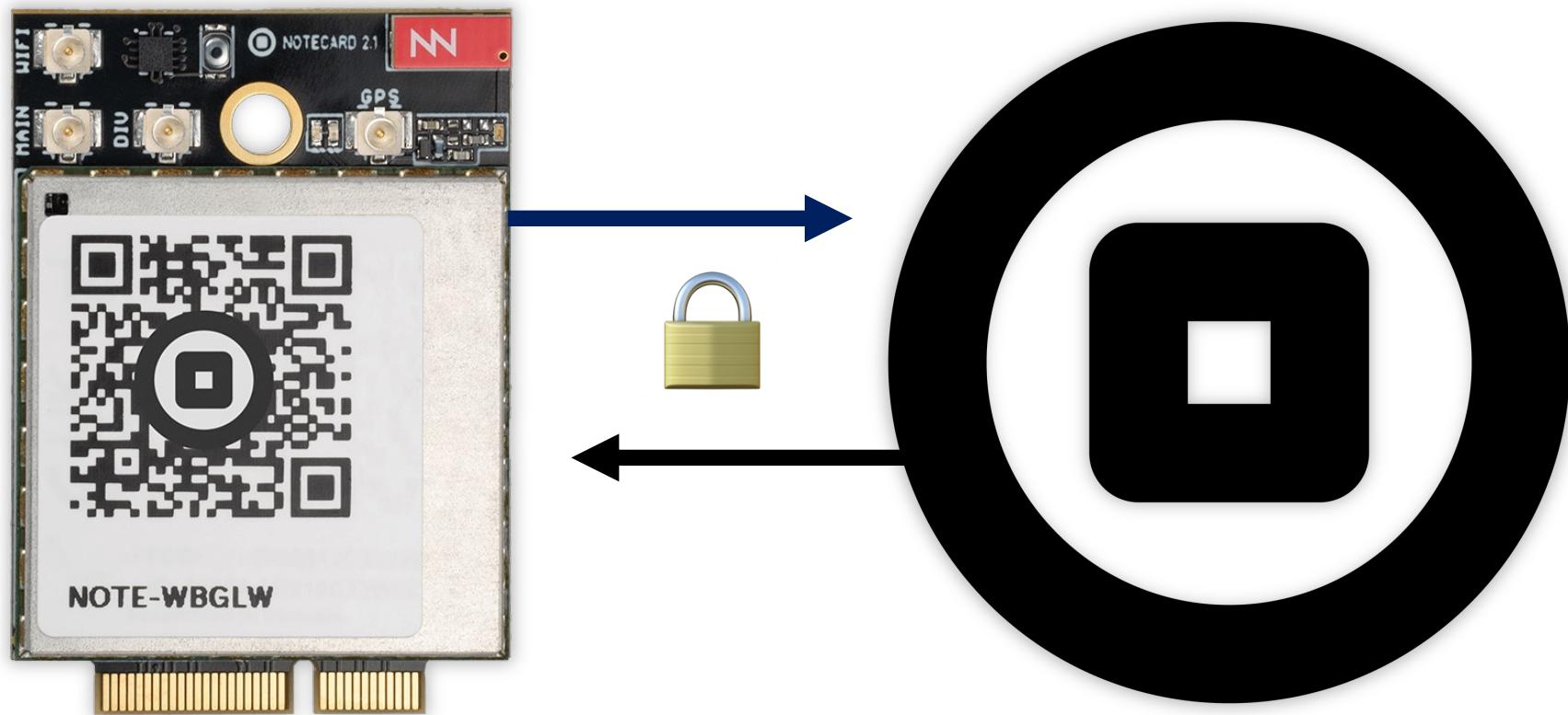
**“Making wireless connectivity easier for
developers and more affordable for all”**



Easy for developers and **affordable** for all.

- 🔒 Securing your data from device to cloud
- 🔋 Building zero-config low-power hardware
- 💻 Providing an unmatched developer experience
- ♻️ Harmonizing the IoT with interchangeable RATs

“Device-to-Cloud Data Pump”



The Notecard Options, Simplified



Cellular Only



LoRa Only



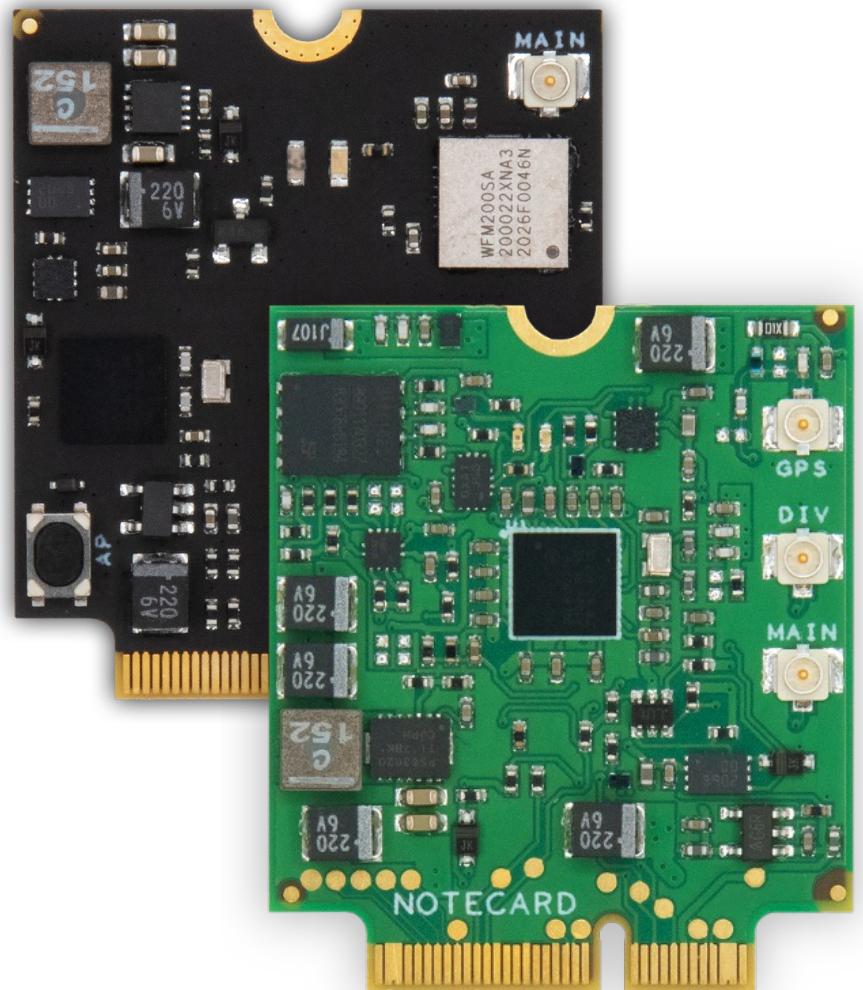
Wi-Fi Only



Cellular and Wi-Fi

Notecard

- Low-power system-on-module
- Global cellular/GPS or Wi-Fi or LoRa
- 500MB cell data + 10 years service
- Dual SIM support
- JSON-based API
- Python, Go, Arduino, C/C++, Zephyr
- NB-IoT, LTE-M, LTE Cat-1



Example: *card.location* API

Request

```
{ "req": "card.location" }
```

Response

```
{
  "status": "GPS updated (58 sec, 41dB SNR, 9 sats),
  "mode":    "periodic",
  "lat":     42.577600,
  "lon":     -70.871340,
  "time":    1598554399
}
```

C Example: *card.location* API

Request

```
J *req = NoteNewRequest("card.location");
```

Response

```
J *rsp = NoteRequestResponse(req);
J *status = JGetString(rsp, "status");
J *mode = JGetString(rsp, "mode");
...etc...
```

What don't you need with Blues?

- SIM or Separate Mobile Plan
- AT Commands or Cellular Radio Management
- Custom Security Implementation
- Roll-Your-Own OTA Firmware Updates
- Power Management
- Custom Cloud Integration
- Certifications



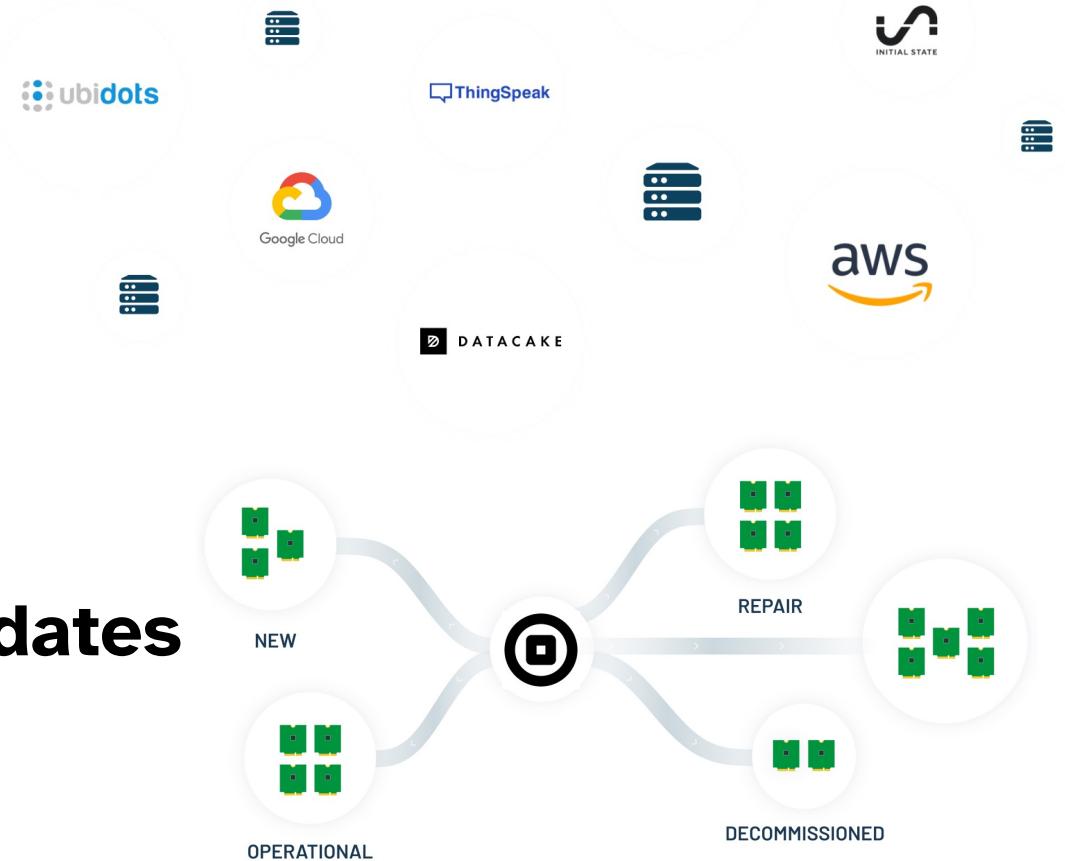
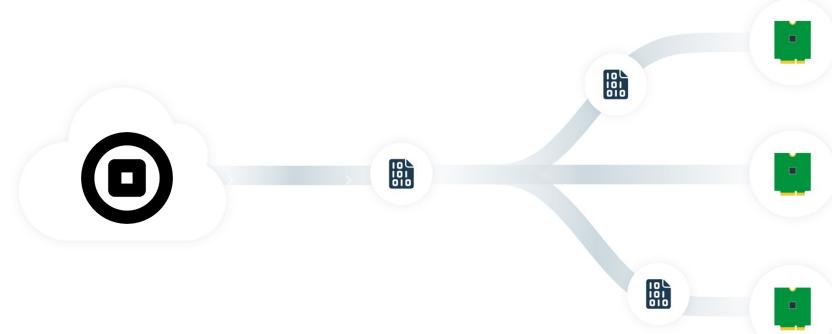
Notecarrier

- Carrier boards for easy prototyping
- Notecarrier for every scenario:
 - **F** - Feather-compatible socket
 - **A** - Any MCU, onboard antennas
 - **B** - Small form factor
 - **Pi** - Raspberry Pi SBC
 - SparkFun MicroMod Cellular Function
 - Blues.ONE from RAKwireless



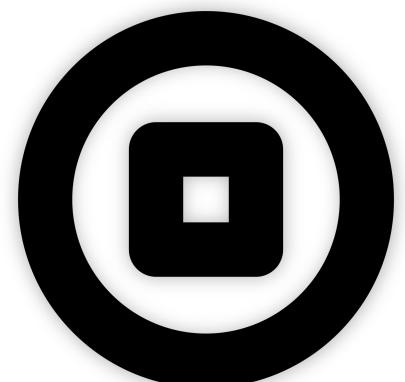
Notehub

- Route data to **any cloud app**
- Manage **fleets** of devices
- OTA MCU/Notecard **firmware updates**
- **Secure** communications



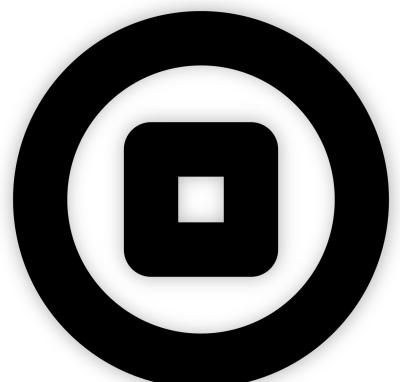
Notehub “Consumption Credits”

- Only Pay for What You Use!
- Billing Account “Topped Up” to 5,000 CCs Monthly
- Notecard Purchase → 5,000 CCs



Notehub “Consumption Credits”

- Send an Event to Notehub? **FREE**
- Route an Event from Notehub to Cloud? **1 CC**
- Pull an Event via API? **1 CC**
- All other API requests? **0.001 CC**



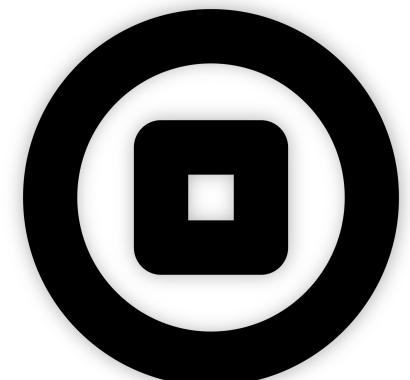
Notehub “Consumption Credits”

Volume Discounting

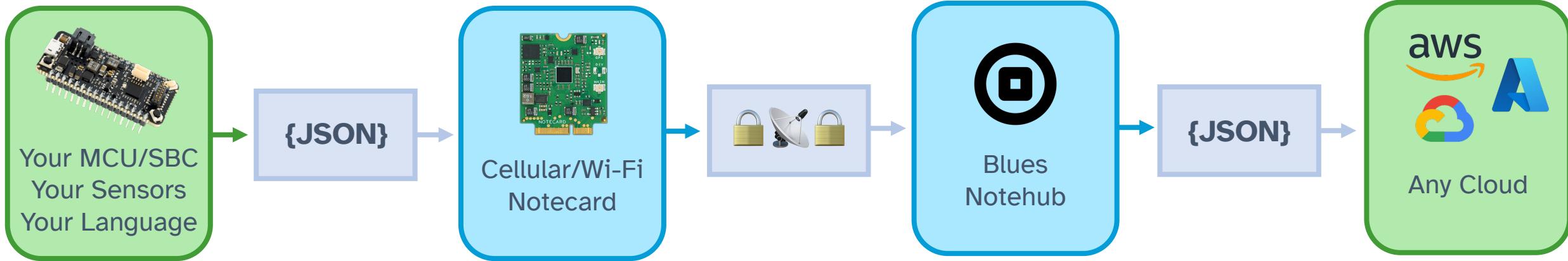
If you require more Consumption Credits, you can purchase more with volume discounts.

Base Unit Price \$0.000750

Quantities Starting From	Cost	Discount	Unit Price
15,000	\$11	0%	\$0.000750
1,000,000	\$563	25%	\$0.000563
10,000,000	\$3,750	50%	\$0.000375
20,000,000	\$4,500	70%	\$0.000225
50,000,000	\$9,375	75%	\$0.000188
100,000,000	\$15,000	80%	\$0.000150



Outbound Communication (from Host to Cloud)



Arduino

C/C++

Python

```
J *req = NoteNewRequest("note.add");
JAddStringToObject(req, "file", "data.qo");

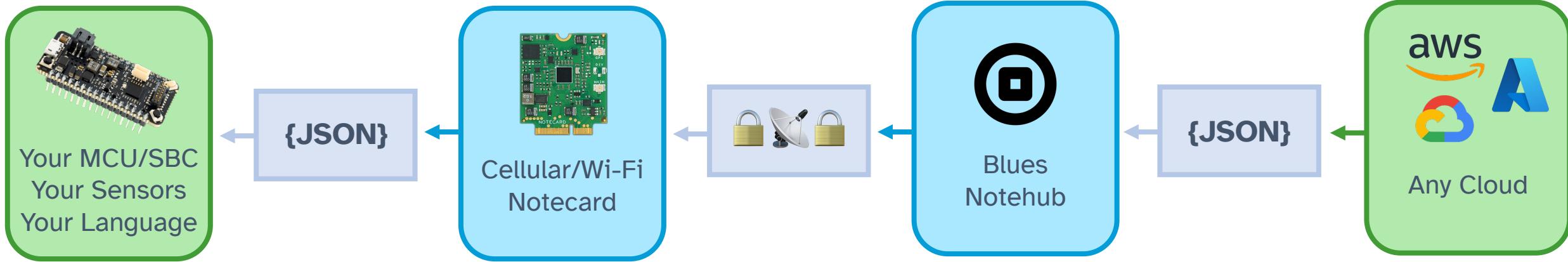
J *body = JCreateObject();
JAddNumberToObject(body, "temp", 27.3);
JAddItemToObject(req, "body", body);

NoteRequest(req);
```

JSON

```
{
  "file": "data.qo",
  "temp": 27.3,
  "when": 1644268443,
  "lat": 42.11,
  "lon": -88.32
  "device": "dev:89347"
}
```

Inbound Communication (from Cloud to Host)



Arduino C/C++ Python

```
J *req = NoteNewRequest("note.get");
JAddStringToObject(req, "file", "data.qi");
JAddBoolToObject(req, "delete", true);

NoteRequest(req);
```

JSON

```
{
  "file": "data.qi",
  "sample_freq": 5,
  "notify": true
}
```



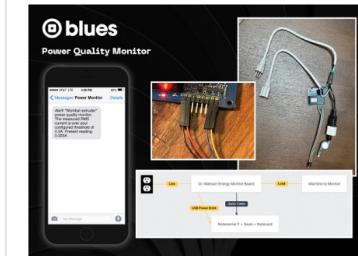
TJ VanToll

Principal Developer Advocate

@TJVanToll

Blues Accelerators

- 50 Purpose-Built IoT Apps, Free and Open Source



Power Quality Monitoring



Valve Monitor

Monitor and control the



CAN bus Vehicle Monitor

blues.dev/accelerators

Energy Monitoring

SMS Notifications Swan

Wiring Required

Web App Provided

Wiring Required

Zephyr Firmware

Soldering Required

Wiring Required

Thanks! (Time for Q&A)

- 🤖 **blues.dev** for Blues developer resources
- 🏁 Get a Blues Starter Kit @ **shop.blues.io**
- 🚶 Clone a free and open-source Blues Accelerator @ **blues.dev/accelerators**



Rob Lauer
Director of Developer Relations
@RobLauer



TJ VanToll
Principal Developer Advocate
@TJVanToll

