

# Patriot IoT Project

---

Ron Lisle



Photons,  
and Echos,  
and iPhones,  
oh My!

Warning: Geek stuff ahead

# But first, a story

---

How to really impress your spouse  
without much effort



# But first, a story

---

Actually, how to unimpress your spouse with a lot of effort





# IoT Situation pre-iPhone

---

- Stuff had gotten a lot cheaper, but still mostly replacing wall switches with custom IoT programming
- Clever programming could do some very cool things
- Initially, cost was my top priority
  - Arduino Nano + nRF24 radios + custom PC \$5
  - Learned value of radio vs wiring
- But lack of OTA programming was inconvenient

# Along Comes the iPhone

---

- Stuff even cheaper, but still mostly replacing wall switches with iPhone buttons
- iPhone programming is tough for hobbyists
  - Too bad, because iPhones are super capable devices
- Commercial solutions still required some sort of programming, tended to be a bit pricy, problematical
- Situation was disappointing. It hadn't really improved all that much

# Amazon Alexa SDK

---

- That all changed with Alexa
- Forget the iPhone, just tell it what to do



# Amazon Echo

---

- Initially, I was optimistic
- The SDK is free to developers
- But I had low expectations
  - Music? Why?
  - What else can I do with it?
- But for my IoT stuff, awesome!



# Alexa Skills

---



- Alexa programs are called
- Two types (initially):
  - Custom Skills - any dialog you want
  - Smart Home Skills - turn things on or off

# Alexa Custom Skills

---

- Requires an invocation word
  - “Alexa, tell **light minder** to turn on {lamp}”
- Completely pre-defined
- “Slots” allow some programmatic handling



# Alexa Smart Home Skills

---

- No invocation word
  - “Alexa, turn on {lamp}”
- Built-in dialogs
- “Devices” determined via “discovery”
- Harder to build, less common
  - Requires “Account Linking”



# A New Story

---

How to actually impress your spouse



# A New Story

---

“Alexa, turn on lamp”

(I dare you to do it faster)



# Alexa Skills

- Simple to make simple skills
- ... but reach a plateau fairly quickly
- ... but you can go further
- Early on I experimented with:
  - Context tracking: Who's On First?
  - Duets: Who's On First?
  - Clever naming: "the folks", "me how"



# ... and I was wrong about the Echo

---

- With use, it is habit forming
- Ask it anything.
  - Often it will surprise you
- Super convenient.
  - I **do** use it for music
  - I use it for time/weather
  - I use it for movie times
- But it isn't 100% accurate
- and it butts-in



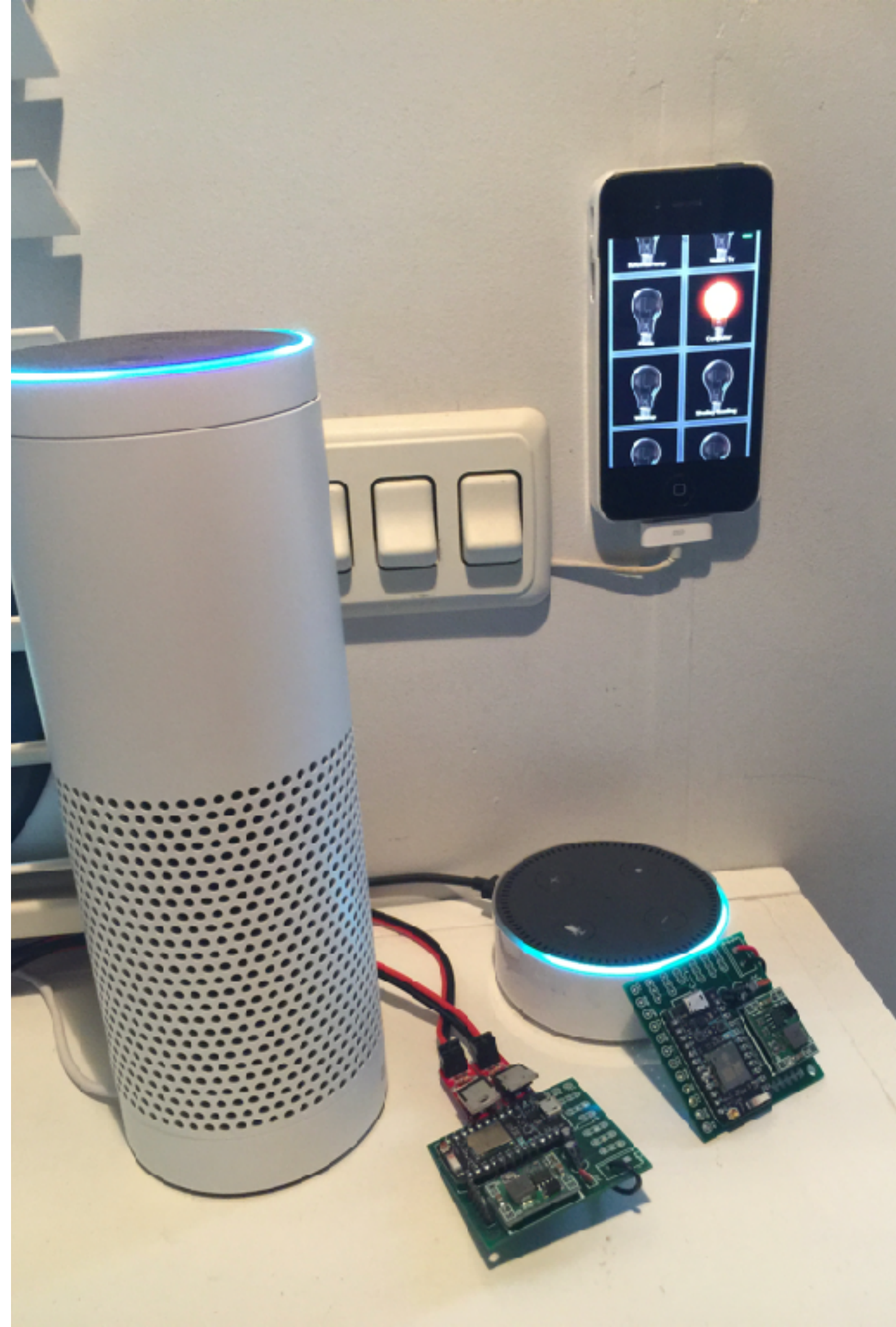
# Back to the Drawing Board

---

So in addition to the existing:

- IoT Switches
- iPhone app
- Photon based controllers

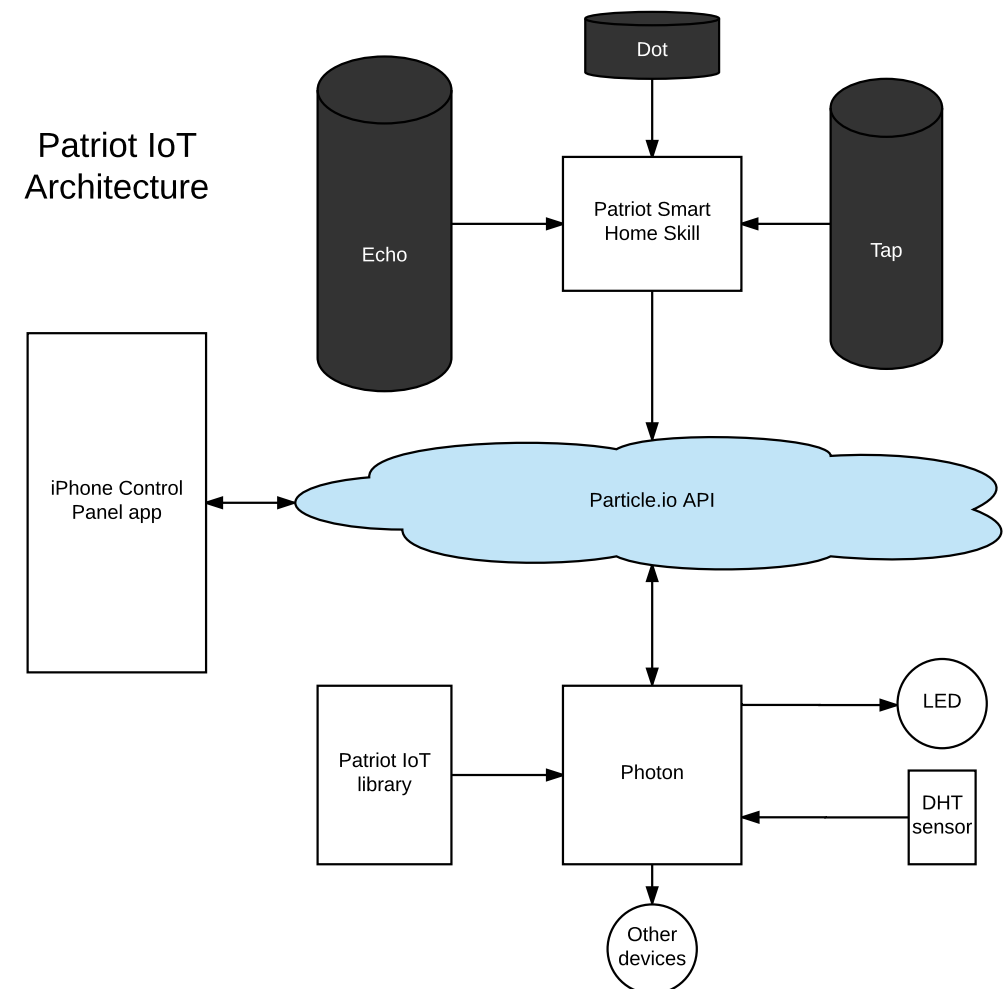
I added Alexa support



# Particle IoT (Patriot)

---

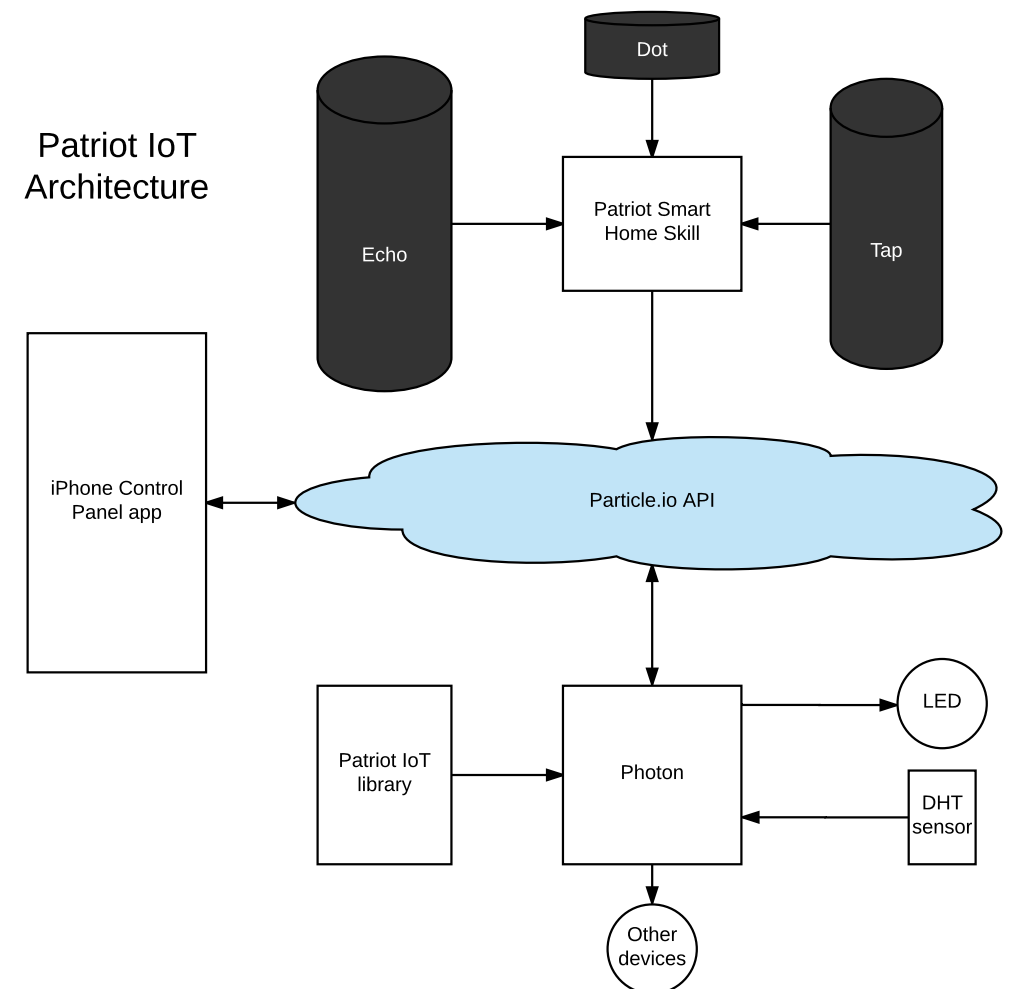
- Very powerful and flexible
- Combination of:
  - Alexa skill (either type)
  - iPhone App(s)
  - Arduino sketches
- Provide incredible capability
- Really flexible for solving home automation problems



# Particle IoT (Patriot)

---

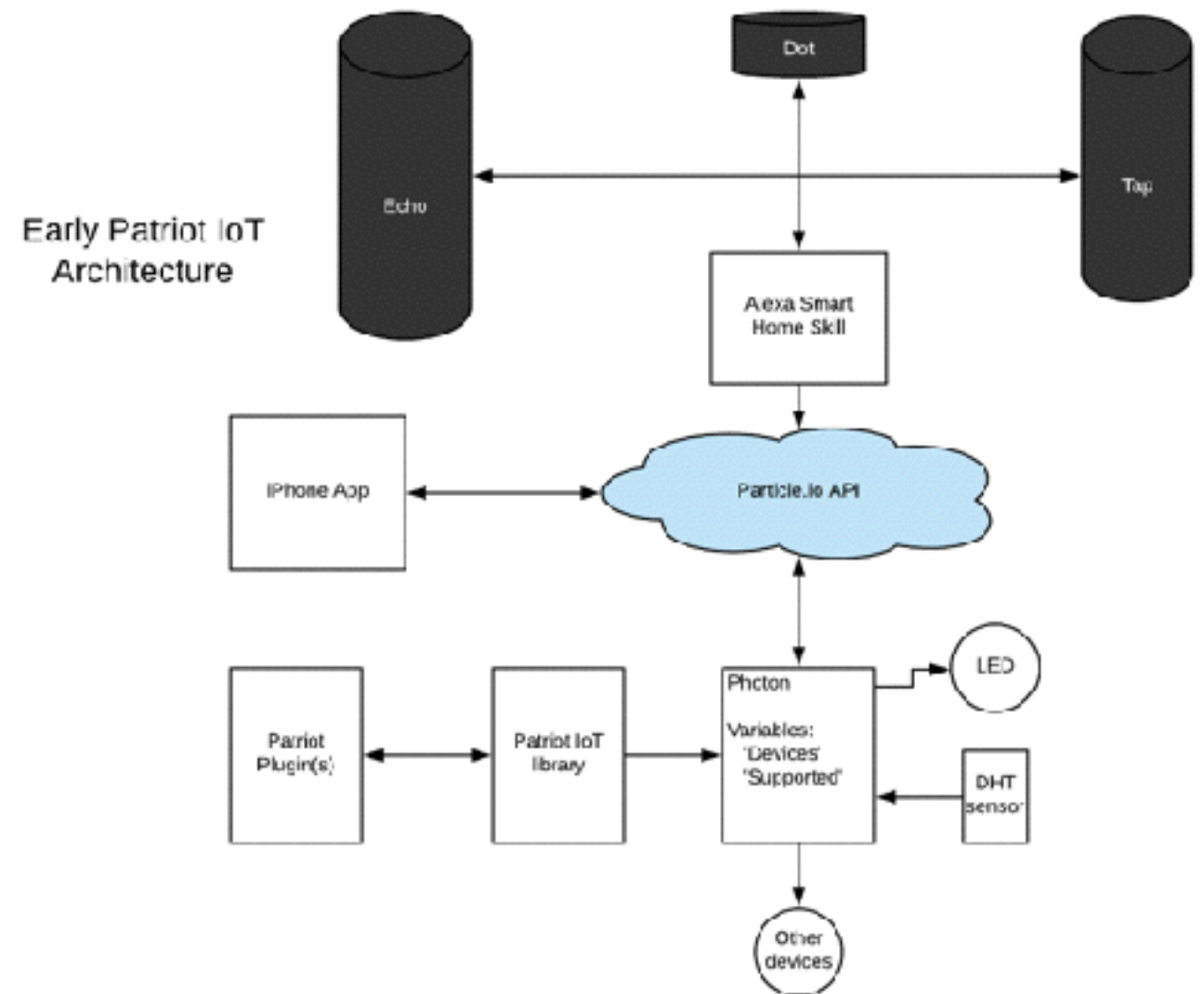
- But creating and updating an:
  - iPhone App
  - and an Alexa Skill
- ... in addition to Arduino sketches is a bit much for any hobbyist (including me)
- I really prefer focusing on just the Photon stuff, but want to use Alexa + iPhone I/O.
- So Particle IoT was born
  - Based on Wifi Photon
  - More expensive, but do the math



# Patriot

---

- Refactored library to plugins
- Expose Photon variables
- Skills and Apps
  - interrogate variables
  - and adapt
- Alexa smart home skill discovery
  - Published
- iPhone app use [particle.io](https://particle.io) to do same
  - TODO: submit (sheesh)



# Example Sketch

---

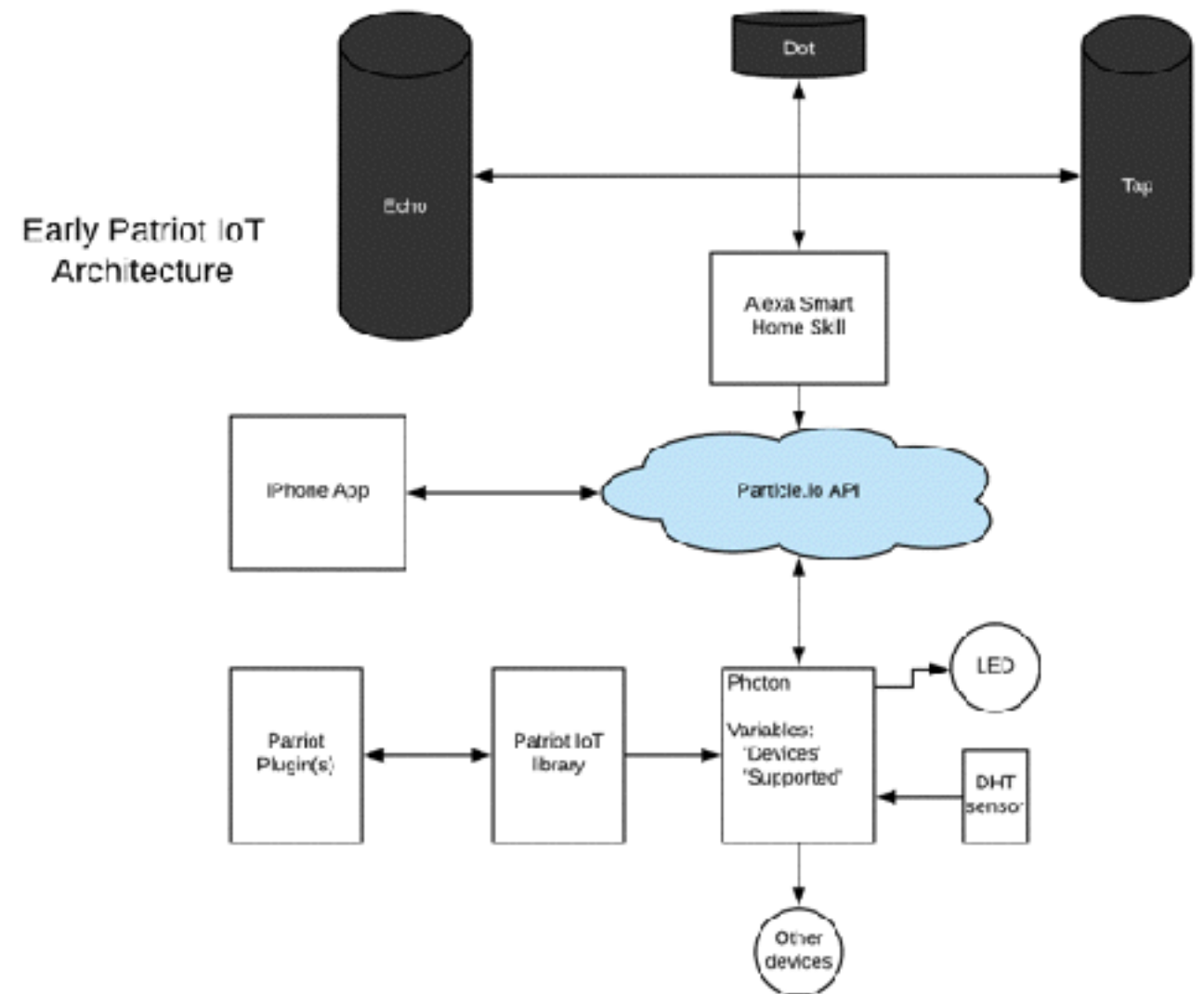
- Nearly all code moved into reusable library
- Boilerplate + 5 lines
- Alexa and iPhone automatically recognize behavior name “photon”
- 

```
18 #include <IoT.h>
19 #include <PatriotLight.h>
20 #include <PatriotSwitch.h>
21
22 IoT *iot;
23
24 void setup() {
25     iot = IoT::getInstance();
26     iot->setControllerName("UnitTest"); // The name of your Photon
27     iot->begin();
28
29     // Create devices
30     //Note: D7 is not a PWM pin, so it can only turn on or off
31     Light *light1 = new Light(D7, "Led");
32     Switch *switch1 = new Switch(D0, "PushMe");
33
34     // Tell IoT about the devices you defined above
35     iot->addDevice(light1);
36     iot->addDevice(switch1);
37
38     // The "behavior" defines the commands that control things.
39     // Alexa will respond to "Alexa, turn photon on" or "Alexa, turn
40     // You can change the word 'photon' to whatever you like, but i
41     // that Alexa can recognize. For now, use a single word.
42     iot->addBehavior(new Behavior(light1, "photon", '>', 0, 100));
43 }
44
45 void loop() {
46     iot->loop();
47 }
```

# Steps To Implement

---

1. Create free [particle.io](https://particle.io) account
2. Install Patriot Smart Home Skill
  - Login with particle
3. Connect Photon to account
4. Use console to create sketch
5. Import Patriot libraries
6. Wait for Alexa discovery
7. Talk to your devices





# Devices

| ID       | Type         | Name          | Last Handshake ⓘ    |     |
|----------|--------------|---------------|---------------------|-----|
| 23003... | Photon       | RearPanel     | 5/21/18 at 10:44am  | ... |
| 29003... | Photon       | FrontPanel    | 5/21/18 at 6:25am   | ... |
| 7ab71... | Raspberry Pi | ronspi        | 5/17/18 at 9:19pm   | ... |
| 7ab7c... | Raspberry Pi | pl            | 11/12/17 at 6:41am  | ... |
| 30004... | Photon       | OfficeCurtain | 10/11/17 at 11:52am | ... |
| 35003... | Photon       | ReferFan      | 7/13/17 at 5:50pm   | ... |
| 2b004... | Photon       | BoothSwitch   | 7/12/17 at 4:36pm   | ... |
| 23003... | Photon       | AboveBooth    | 7/12/17 at 4:36pm   | ... |

Particle Console | Build your cc x

Particle Build

Secure https://build.particle.io/build/59d1712e6c2826cae1000228

Particle Apps

Current App

STARTER

Optional description

Files

starter.ino

Included libraries

IoT (1.1.5)

PatriotLight (1.0.1)

PatriotSwitch (1.0.1)

Code

SHARE THIS REVISION

REMOVE APP

My apps

Type to find

BlinkBlueLED

AboveBooth

Blink an LED

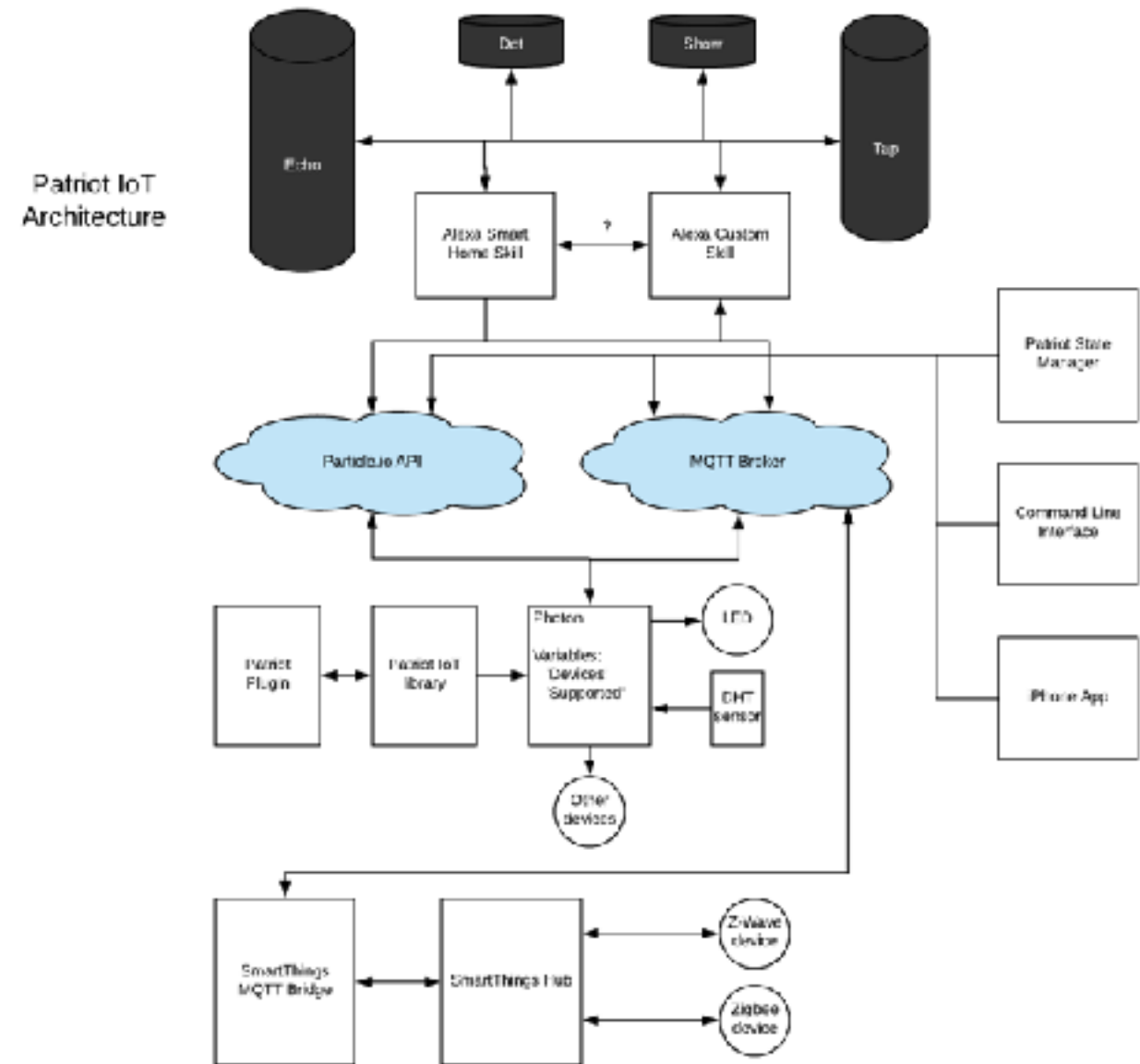
starter.ino

```
13 Changelog:
14 2017-03-28: Use fixed 'patriot' event name.
15 2017-03-24: Initial creation
16 *****/
17
18 #include <IoT.h>
19 #include <PatriotLight.h>
20 #include <PatriotSwitch.h>
21
22 IoT *iot;
23
24 void setup() {
25     iot = IoT::getInstance();
26     iot->setControllerName("UnitTest"); // The name of your Photon device. Can be whatever you named
27     iot->begin();
28
29     // Create devices
30     //Note: D7 is not a PWM pin, so it can only turn on or off
31     Light *light1 = new Light(D7, "Led");
32     Switch *switch1 = new Switch(D0, "PushMe");
33
34     // Tell IoT about the devices you defined above
35     iot->addDevice(light1);
36     iot->addDevice(switch1);
37
38     // The "behavior" defines the commands that control things.
39     // Alexa will respond to "Alexa, turn photon on" or "Alexa, turn off photon"
40     // You can change the word 'photon' to whatever you like, but it needs to be something
41     // that Alexa can recognize. For now, use a single word.
42     iot->addBehavior(new Behavior(light1, "photon", '>', 0, 100));
43 }
44
45 void loop() {
46     iot->loop();
47 }
48
```

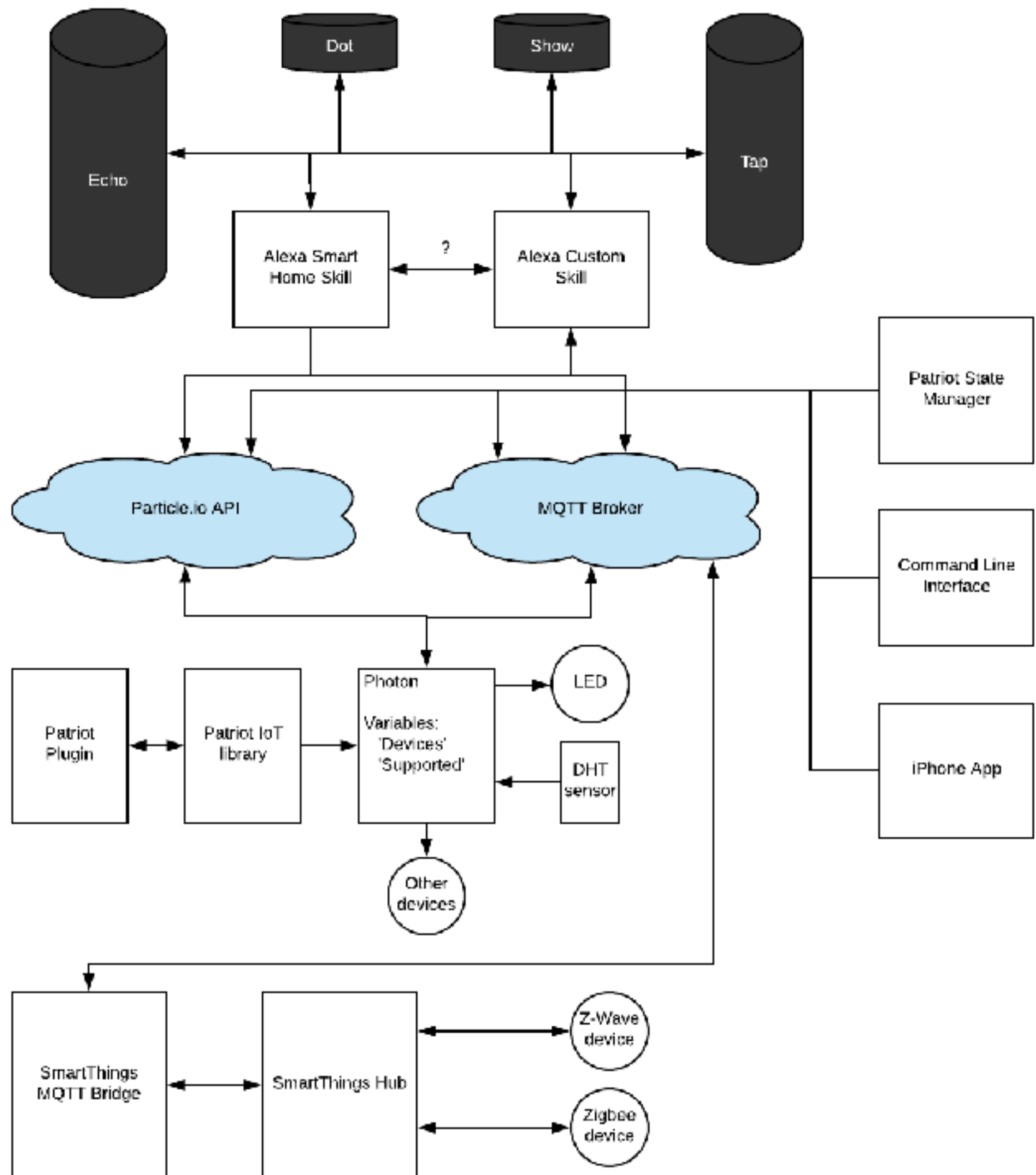
Ready.

# Current Patriot Architecture

- Separate Alexa Skills
- Added Patriot-CLI
- MQTT
  - Works when disconnected
  - Is very fast!
- Integrated with SmartThings
  - Yikes! This is complicated
- Future State Manager



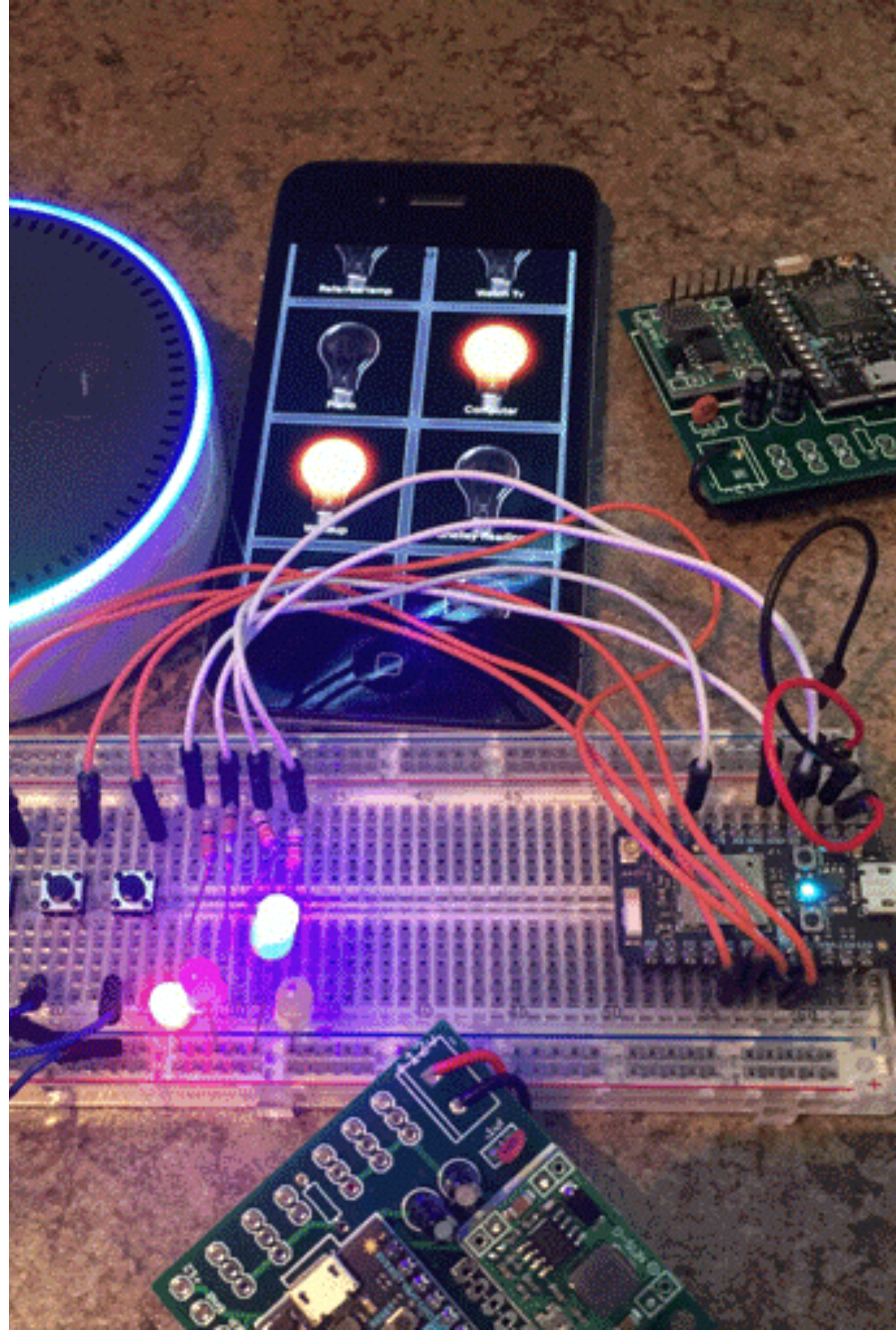
## Patriot IoT Architecture



# What's Next?

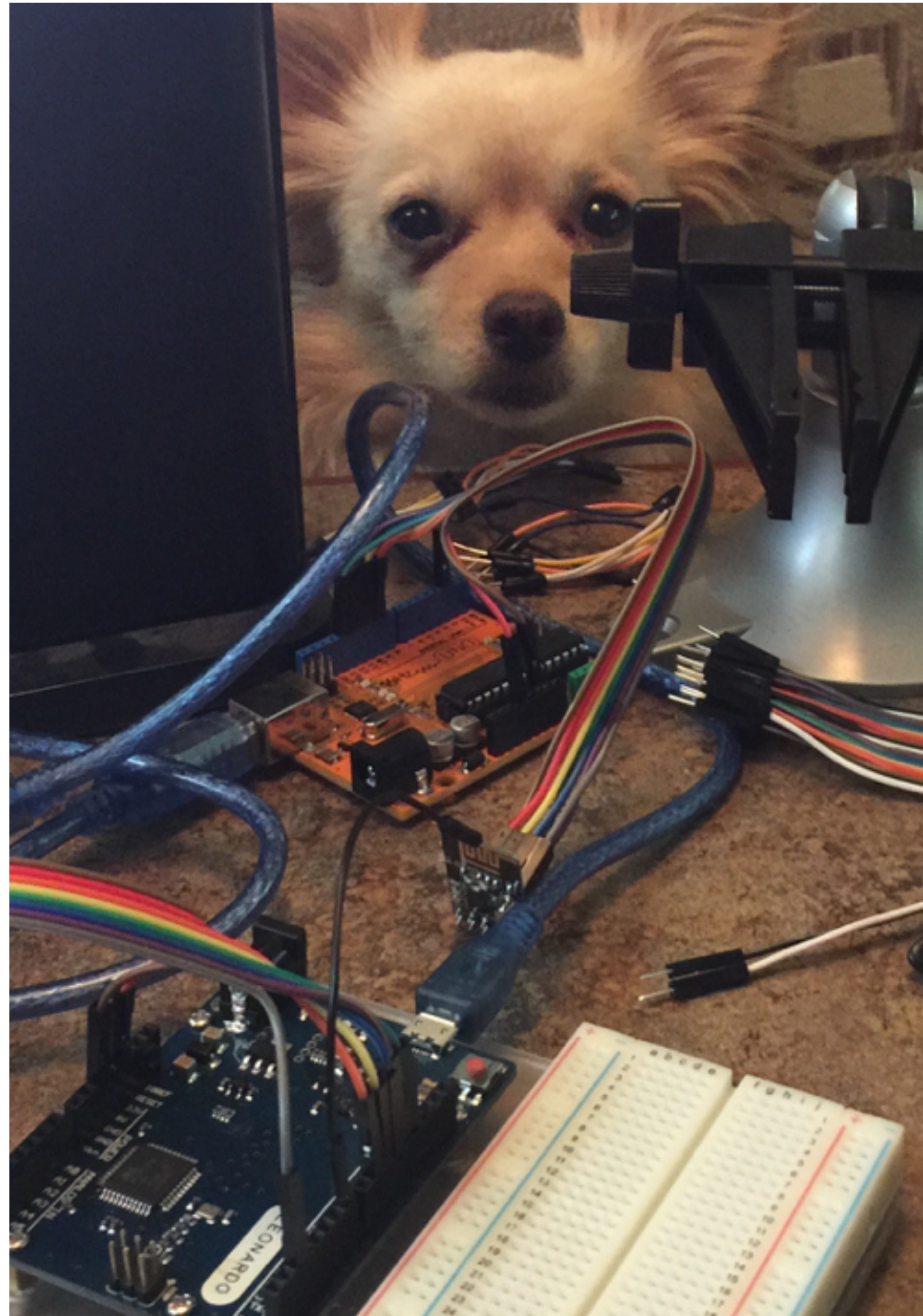
---

- Coordinated Alexa Skills
- MQTT as primary
- Blinds, Awnings, Fans
- Motion/Proximity inputs
- Offline Alexa backup?
- ESP8266, etc.
- OpenCV



Questions?

---



# Links

---

- [Github.com/rlisle](https://github.com/rlisle)
  - Patriot, Patriot-iOS, Patriot-Alexa-Skill, PCBs
- [Hackster.io/rlisle](https://hackster.io/rlisle)
  - 4 Patriot projects including plugin & tutorial projects
- I'm also active on Stack Overflow
  - Mostly Alexa questions