INTERNET of THINGS
with **PYTHON** &
SERVERLESS…

Sonal Raj

PYCON MY 2019

Kuala Lumpur, Malaysia

Hi, I'm Sonal Raj...

.. Open Source evangelist

.. graph databases tinkerer, real time processing enthusiast

.. Python and I share 10+ years of friendship

.. Pursue research in education technology of late

.. Part time author and blogger

LABS

... and here is where you can find me

github.com/sonal-raj

twitter.com/@sonalraj_

www.sonalraj.com

The SCOPE

Internet of Things

1

Serverless Computing

2

Python

3

Creating Applications

4

Ideas

5

Int.of.things

COMPONENTS

Senses

Sensors

Proximity to a surface

Can listen to what you say

See things in front and behind it.

Transfer Info over a wireless network

Your location

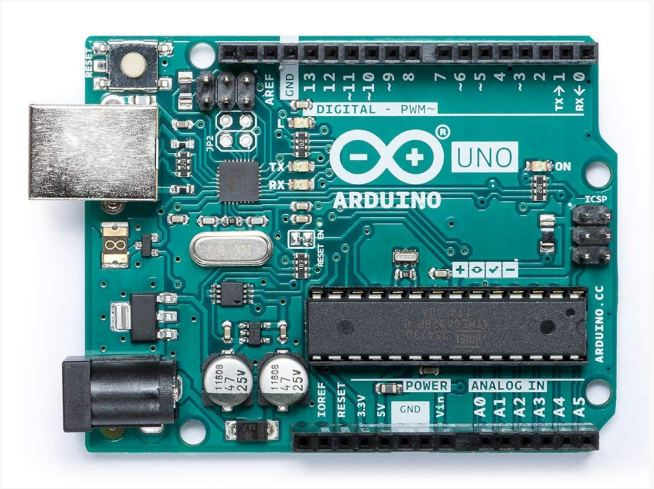
Your state of motion and direction

Orientation of the device

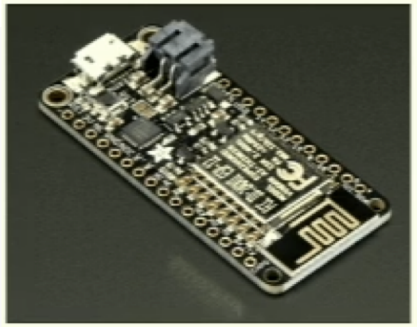
Ambience or light in the room

Controllers
& SOCs

Raspberry Pi 3B+



Arduino

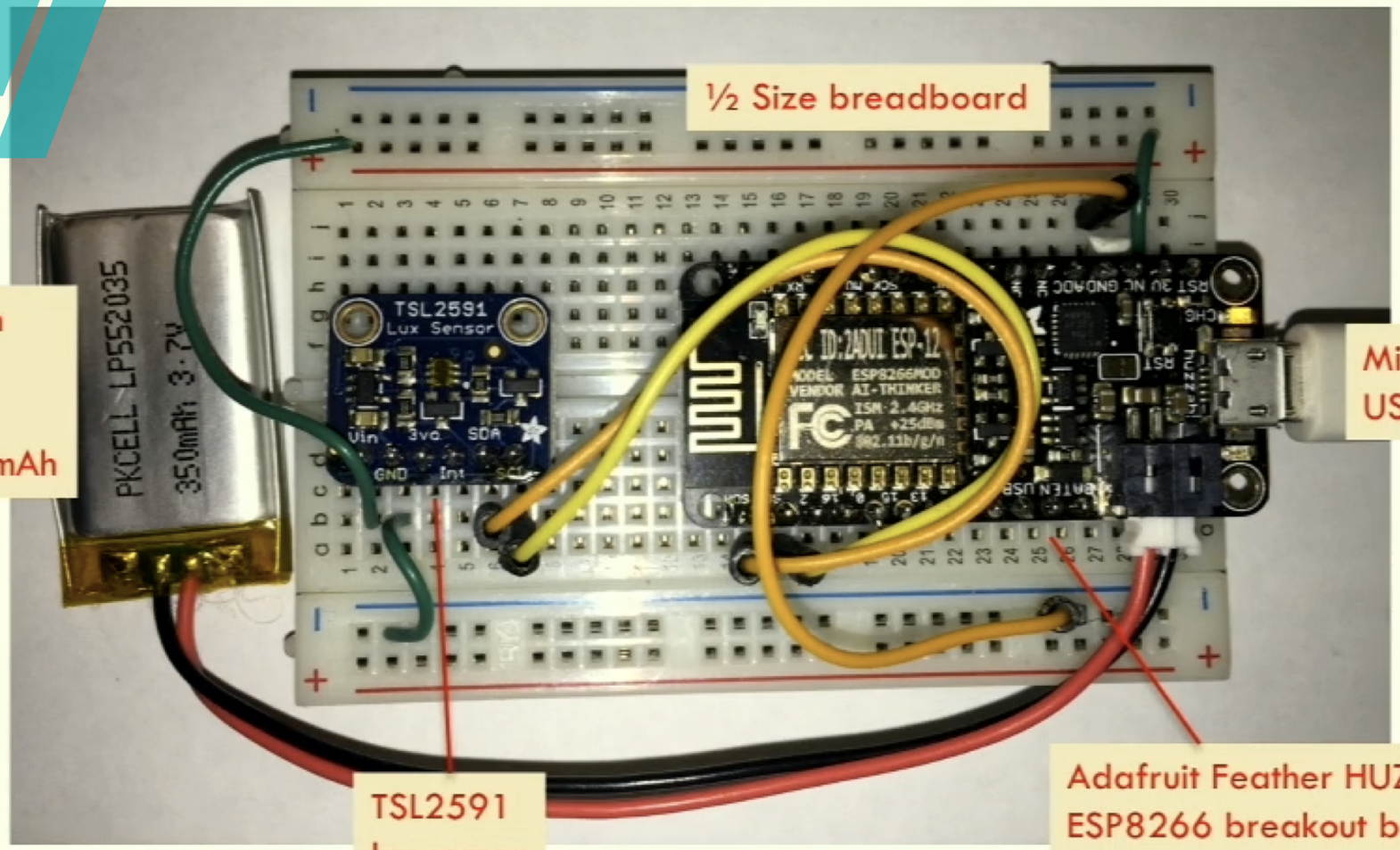


ESP8266

- System-on-a-chip with 32-bit CPU, WiFi, I/O
- Low power consumption
- Only 96K data memory!
- MicroPython to the rescure

PLC / PAC

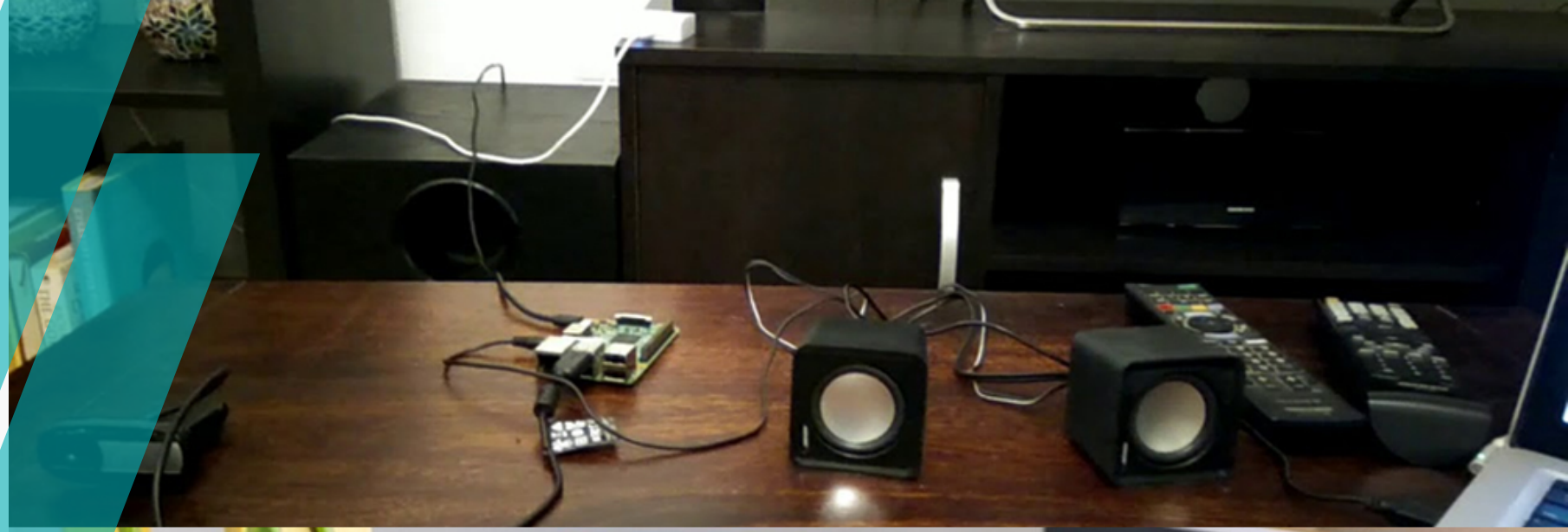
Controllers
& SOCs



WHAT DOES A BASIC IOT SYSTEM IN ACTION LOOK LIKE?

AMBIENT LIGHTING

AUDIO
SYSTEMS



DIGITAL DEVICES

DISCONNECTED DEVICES

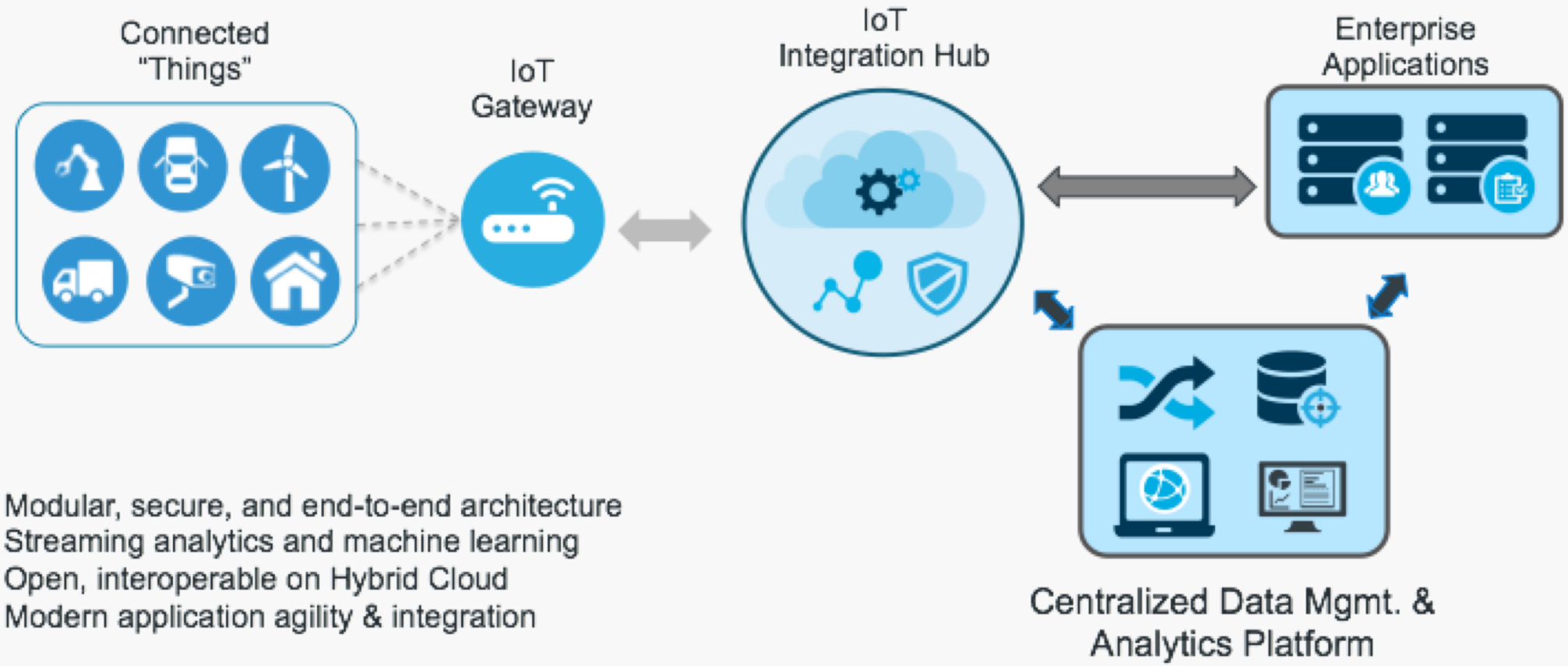
Internet Enabled

Support for Digital Assistants

Smart phone Controlled

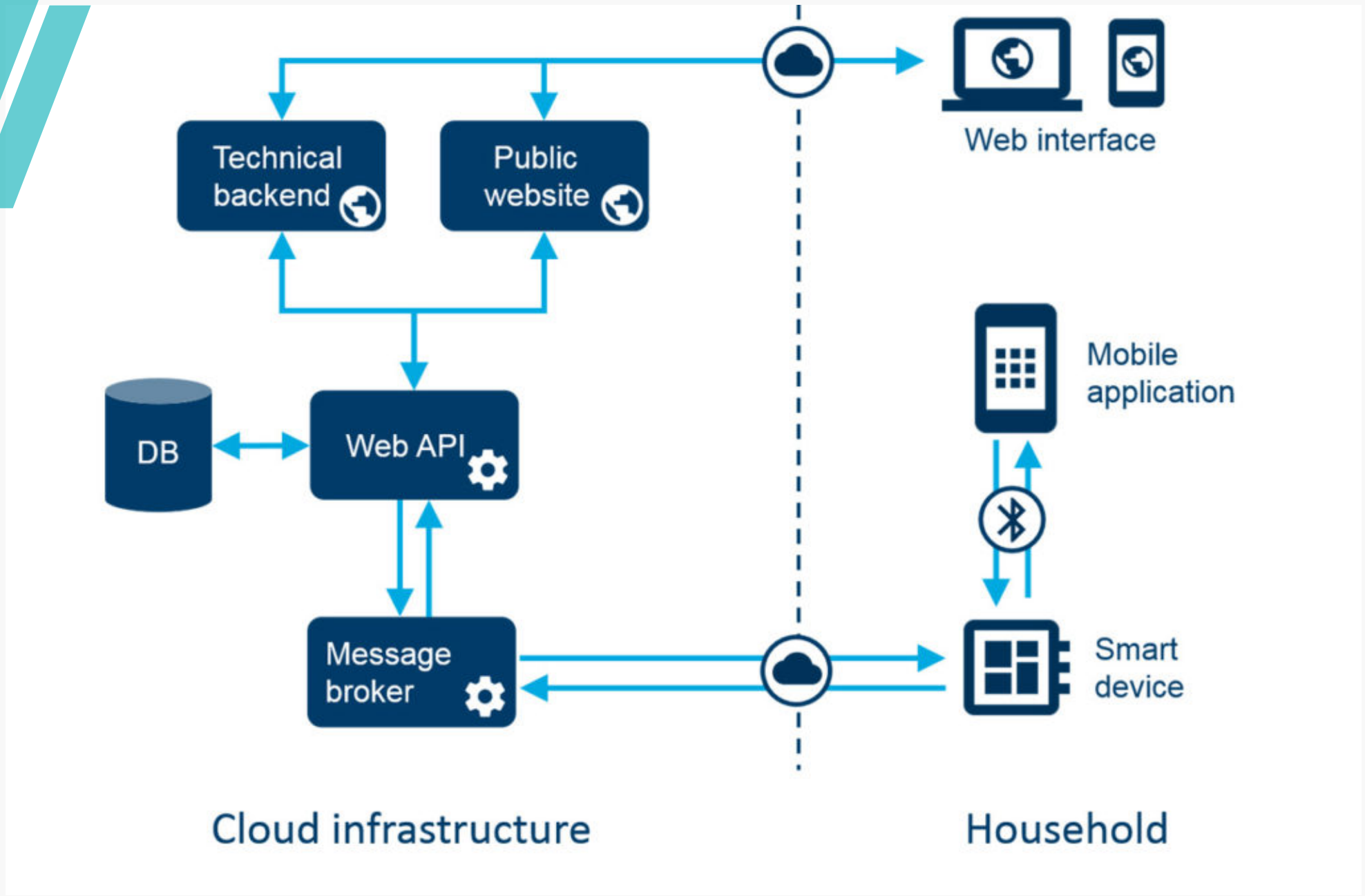
Programmable APIs

IOT
ARCHITECTURE

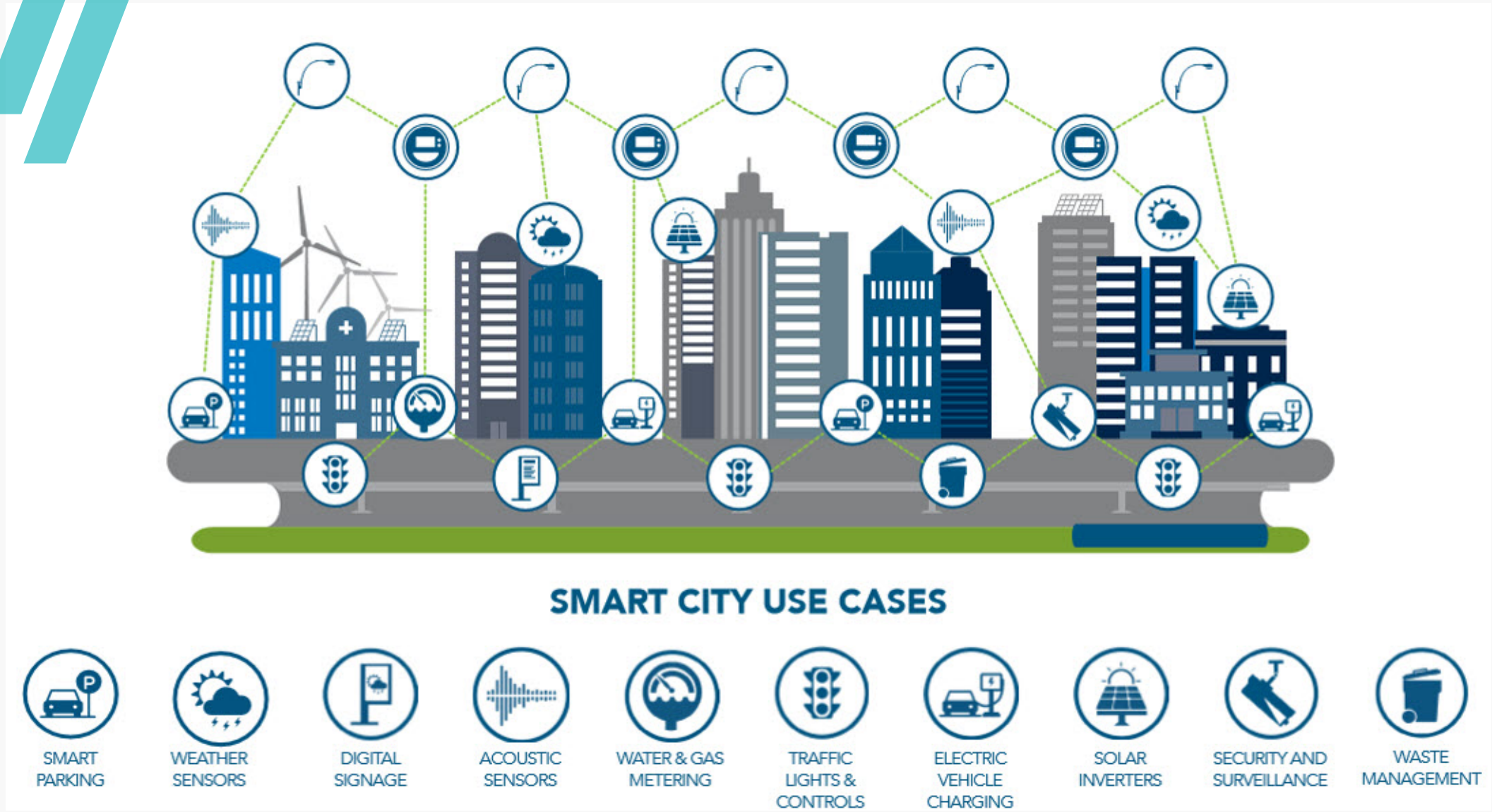


- Modular, secure, and end-to-end architecture
- Streaming analytics and machine learning
- Open, interoperable on Hybrid Cloud
- Modern application agility & integration

Smart HOMES


Technical backend
Public website
Web interface
DB
Web API
Mobile application
Message broker
Smart device
Cloud infrastructure
Household

Smart
CITIES



The SCOPE

Internet of Things

1

Serverless Computing

2

Python

3

Creating Applications

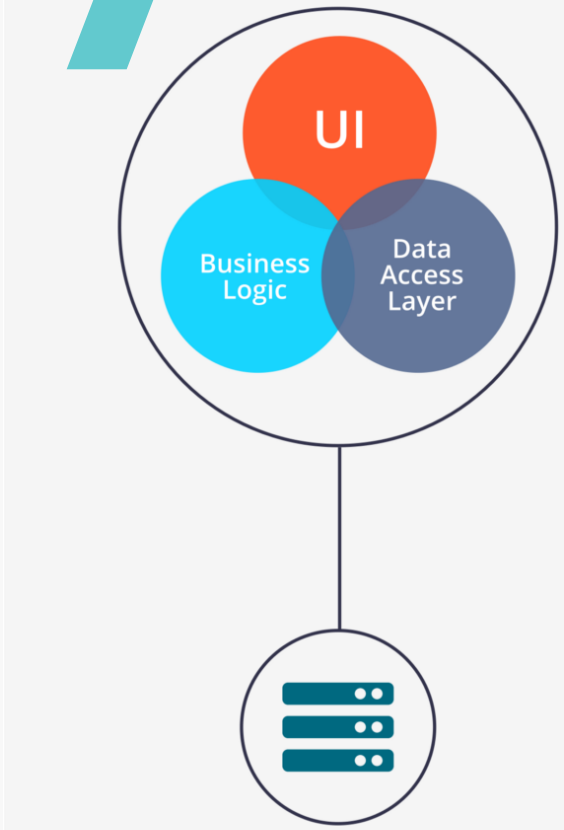
4

Ideas

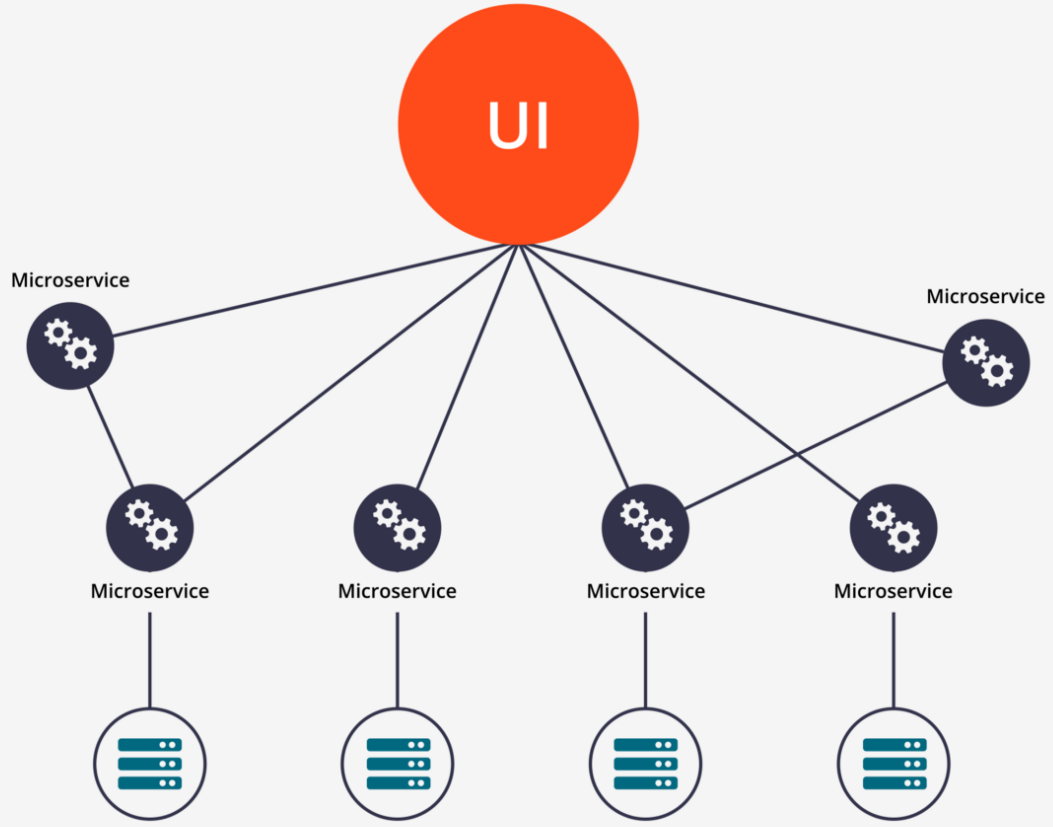
5

Serverless

COMPUTE

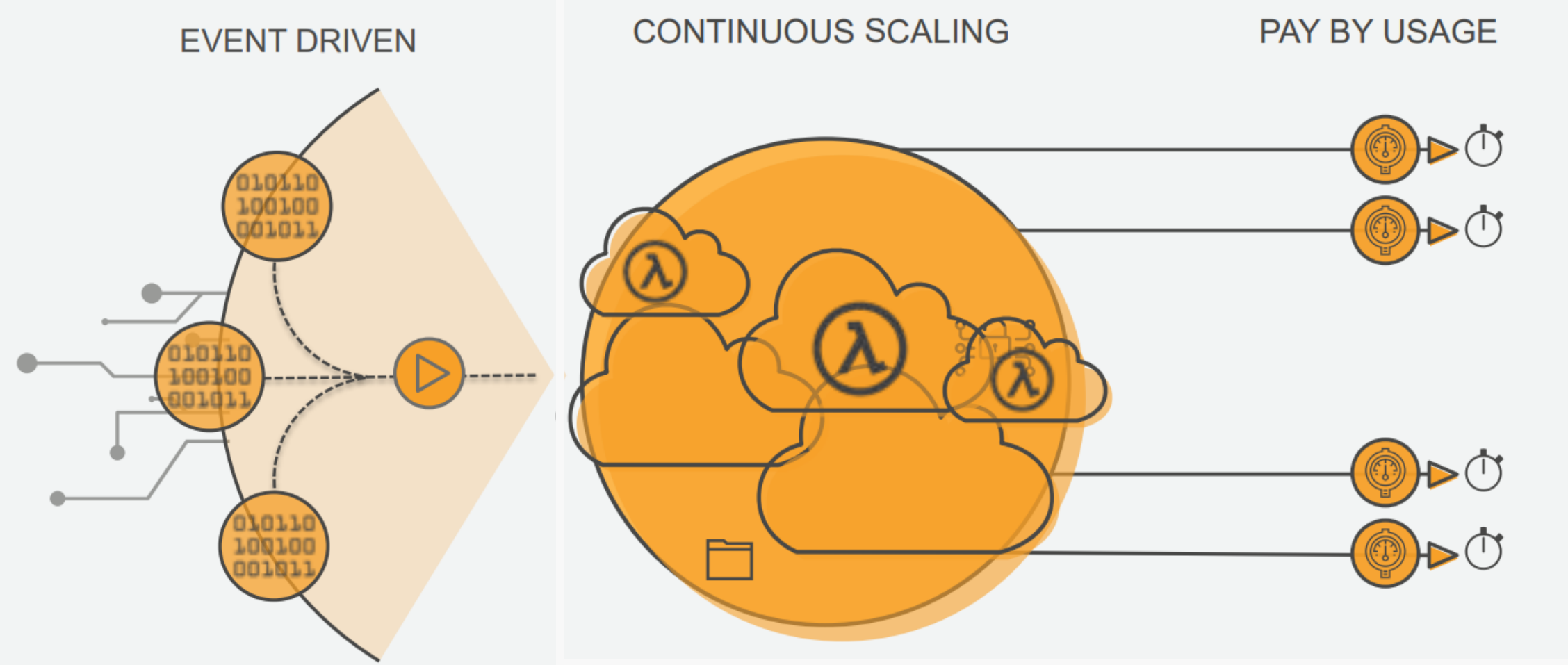


Monolithic Architecture

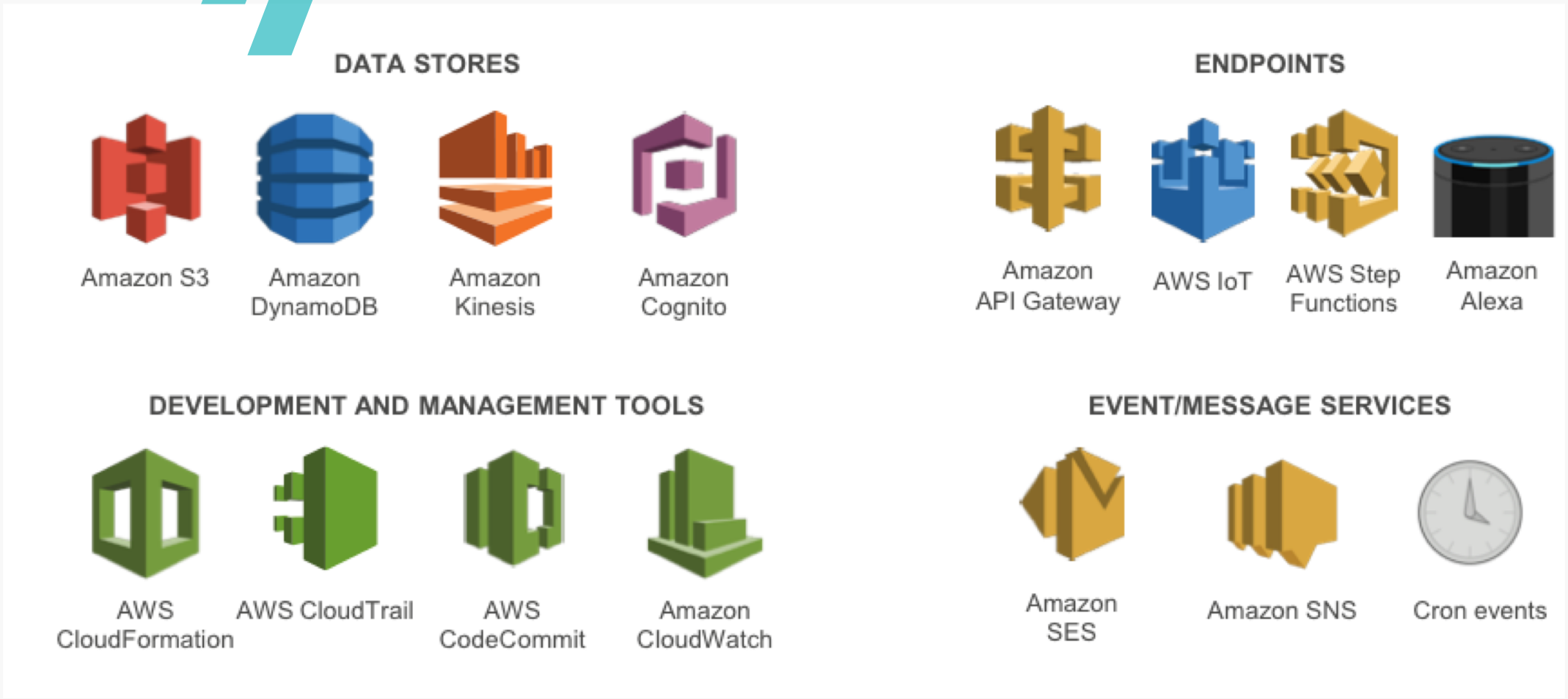


Microservice Architecture

Serverless
COMPUTE



Event
TRIGGERS



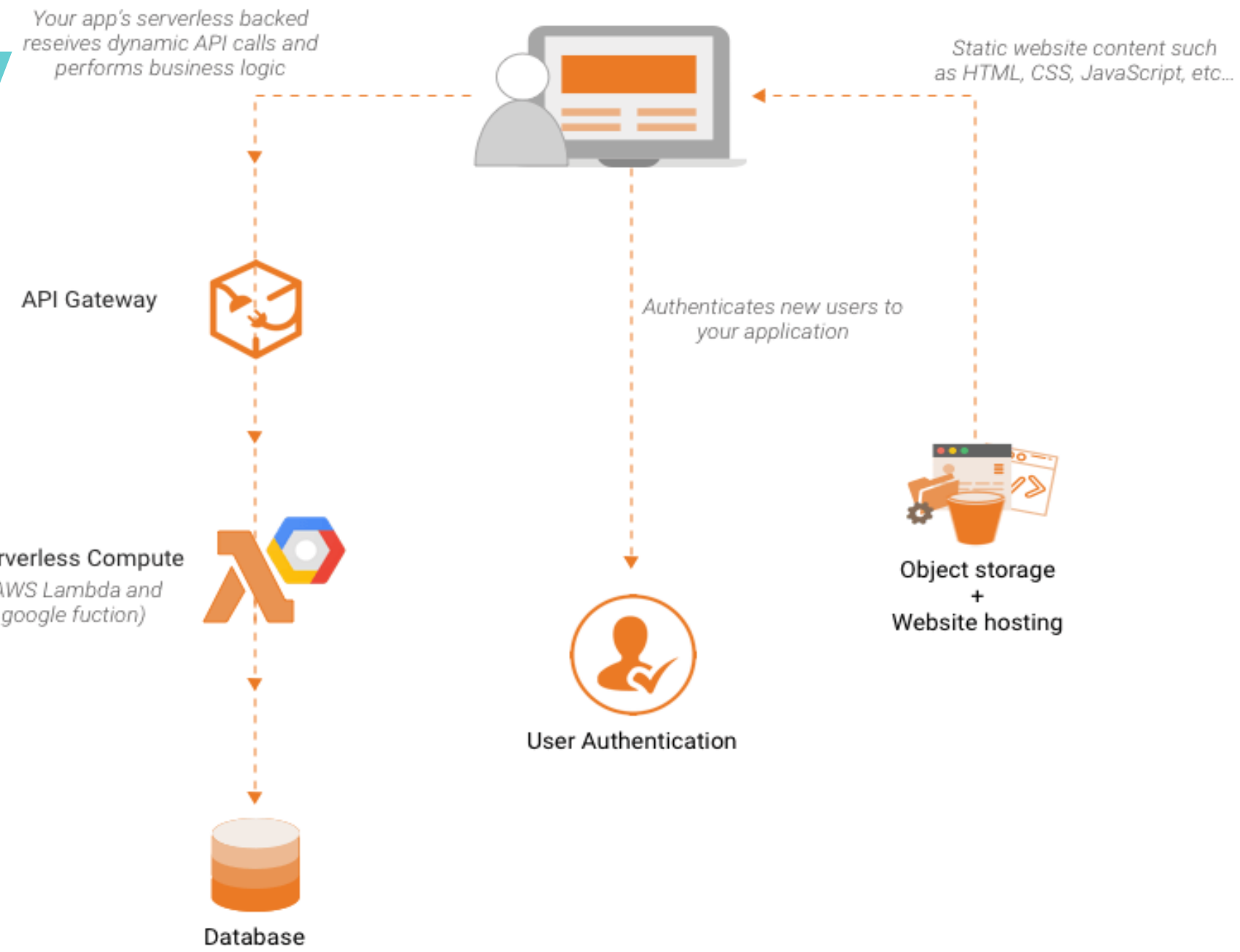
Why
SERVERLESS

- No provisioning or management of infrastructure
- Automatic Scaling
- Pay per use / No Idle Capacity
- Automatic Implied Availability and Scalability

Serverless

PROVIDERS

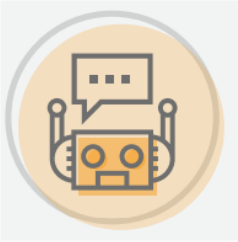
How it all
WORKS



Serverless

APPLICATIONS

Static Websites, Web Apps, Flask Packages

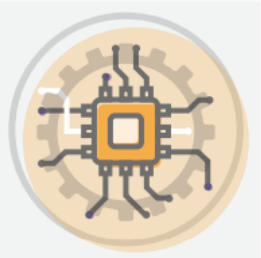


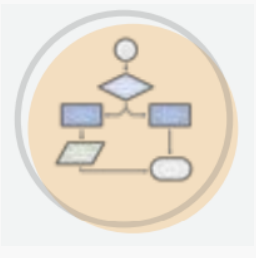
Creating logic for chatbots

Backend Apps and Services, Mobile and IOT

Voice enabled apps and skills in Alexa assistants

Data Processing in Real Time, Batch or MapReduce



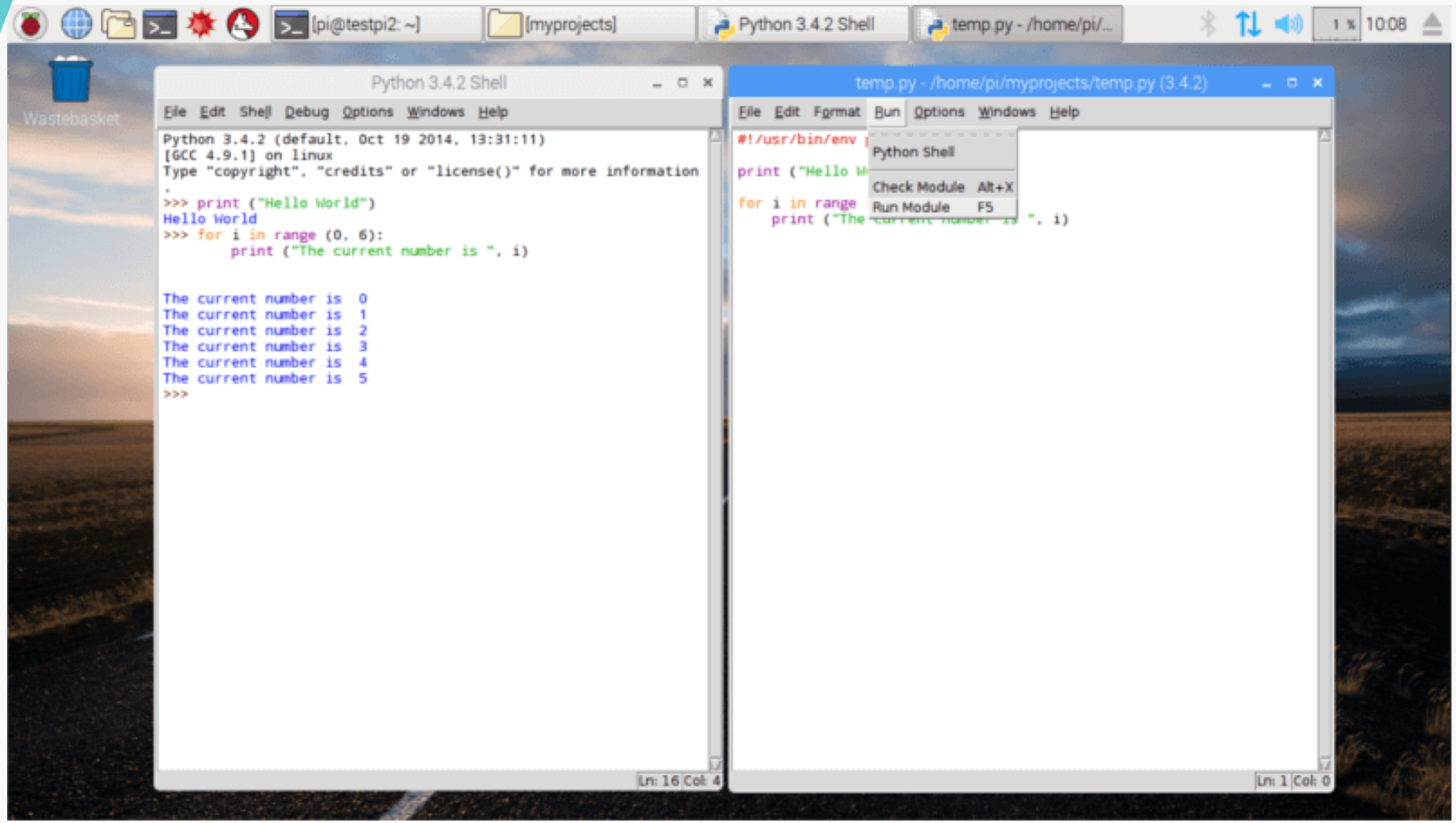


Automation of Services and Infrastructure management

The SCOPE


Internet of Things
Serverless Computing
Python
Creating Applications
Ideas
1
2
3
4
5

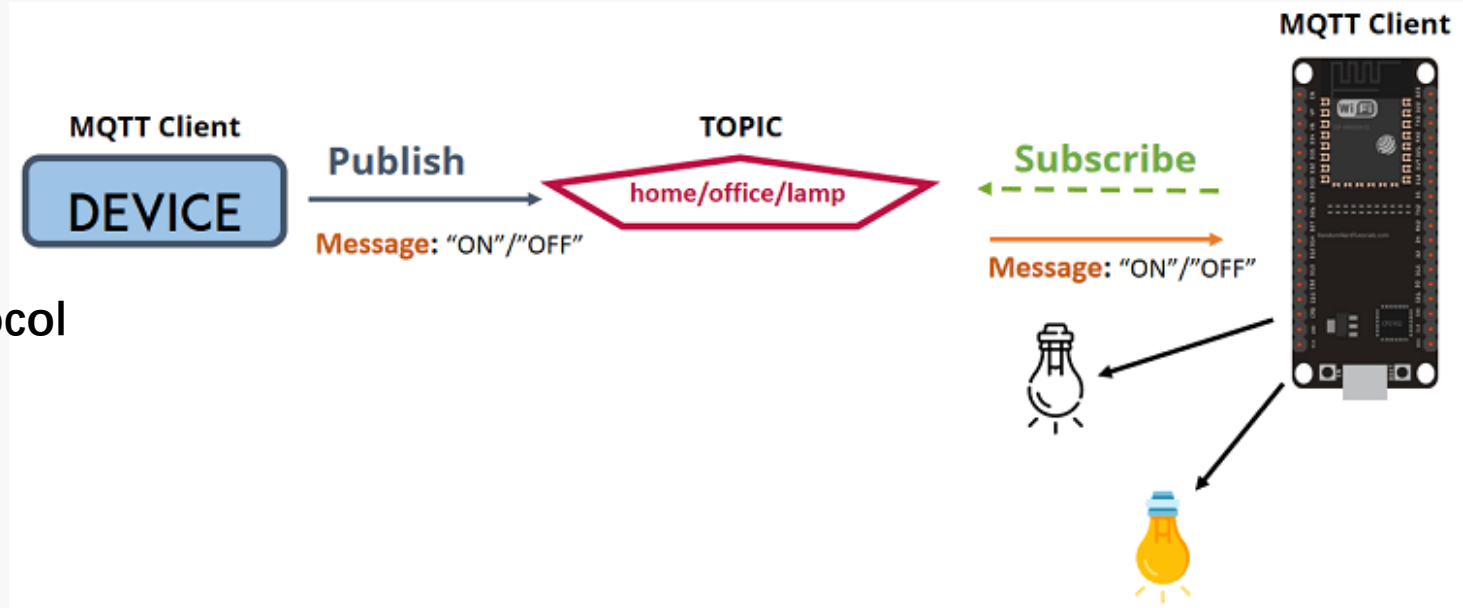
Raspberry
Pi 3B+



Message
BROKERS

MESSAGE QUEUEING
TELEMETRY TRANSPORT

Setting
UP



- Machine to Machine connectivity protocol
- Lightweight Publish / Subscribe
- Small Footprint and low bandwidth
- The MQTT protocol is based on TCP/IP.

Topics - series of levels separated by slashes

```
house/alarm/status
house/alarm/zone

house/camera/capture                    house/camera/#
house/camera/newpicture                 house/camera/+
```

Setting UP

Messages

```
FFFFF11000 # Float device identifier
2-1235489698.jpg
```

Clients

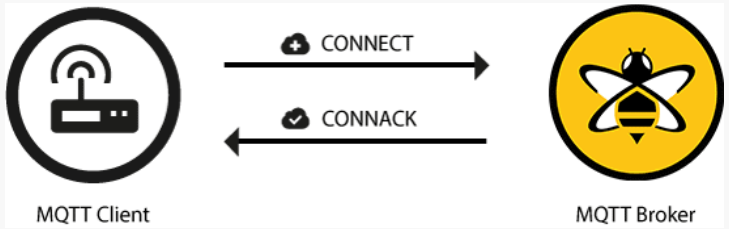
```
Paho-mqtt (https://pypi.org/project/paho-mqtt/)
```

Quality of Service

```
0 – best effort
1 – deliver at least once
2 – exactly once delivery
```

Brokers

```
Mosquitto (Python)
Mosca (NodeJS)
```



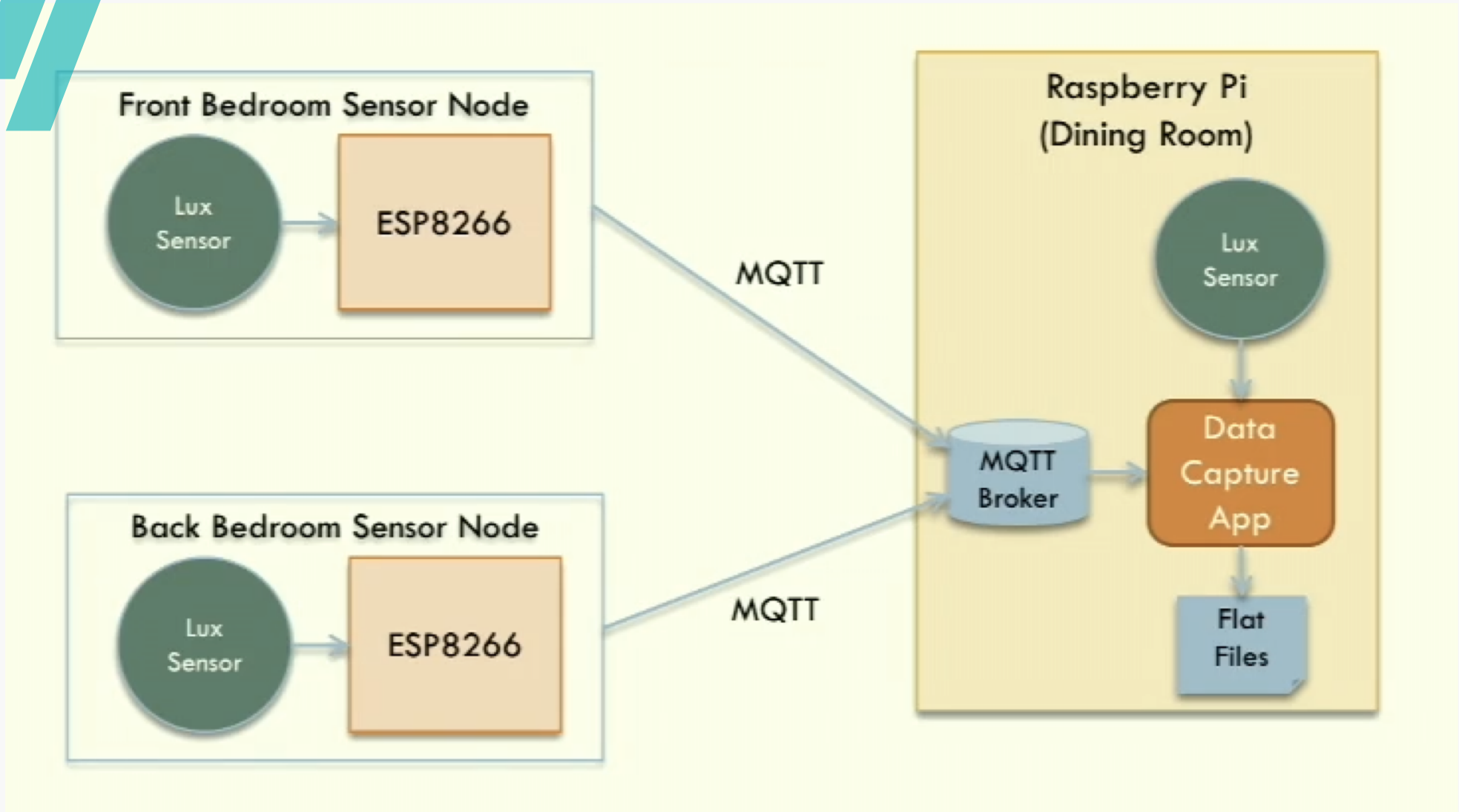
Security

```
Userid / deviceid / passwords
TLS / Certificate
```

IOT Services

```
IBM bluemix
AWS-IOT
AZURE-IOT
```

Setting
UP


Front Bedroom Sensor Node
Lux Sensor
ESP8266
Back Bedroom Sensor Node
Lux Sensor
ESP8266
MQTT
MQTT
Raspberry Pi (Dining Room)
Lux Sensor
MQTT Broker
Data Capture App
Flat Files

Comm

UNIX

libmraa - Low Level Skeleton Library for Communication on GNU/Linux platforms

Librmaa

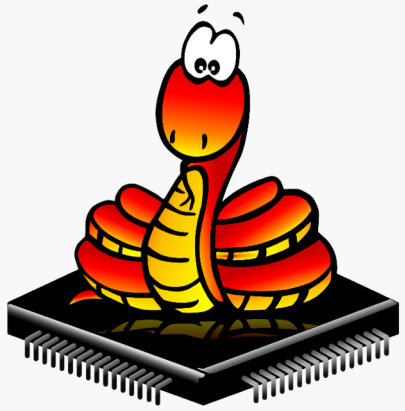
PYTHON

- Easy Portable Interface for working with sensors and actuators
- Low level Interface
- Access and manipulate the basic I/O capabilities of a platform
- Translation layer on top of the Linux GPIO
- Platform Agnostic APIs
- Written in C, but has binding for many languages including Python!

LABS

Python
LIBRARIES

librosa



Micropython

```
(defn neighbours [[x y]]
  (for [dx [-1 0 1]
        dy (if (zero? dx)
             [-1 1]
             [-1 0 1])]
    [(+ dx x) (+ dy y)])) #'user/neighbours

(defn step [cells]
  (set (for [[loc n] (frequencies (mapcat neighbours cells))
             :when (or (= n 3)
                       (and (= n 2) (cells loc)))]
         loc))) #'user/step

(assoc [])

clojure.lang.ArityException: Wrong number of args (1) passed to: core$assoc
            AFn.java:437 clojure.lang.AFn.throwArity
          RestFn.java:412 clojure.lang.RestFn.invoke
        NO_SOURCE_FILE:19 user/eval5559


(def board #{[1 0] [1 1] [1 2]})
```

OpenCV

python™

Flask-Ask

Rapid Alexa Skills Kit Development
for Amazon Echo Devices

Python
LIBRARIES

The SCOPE


Internet of Things
Serverless Computing
Python
Creating Applications
Ideas
1
2
3
4
5

App
DEMO

Rpi-3B+ based Application

Writing SKILLS

Slot

A variable within an utterance. Variable possibilities must be defined.

“Alexa, ask Zomato for the nearest [restaurantType]”

Invocation Name

What a user must say to “start” the skill. Each skill has a unique Invocation Name

Utterance

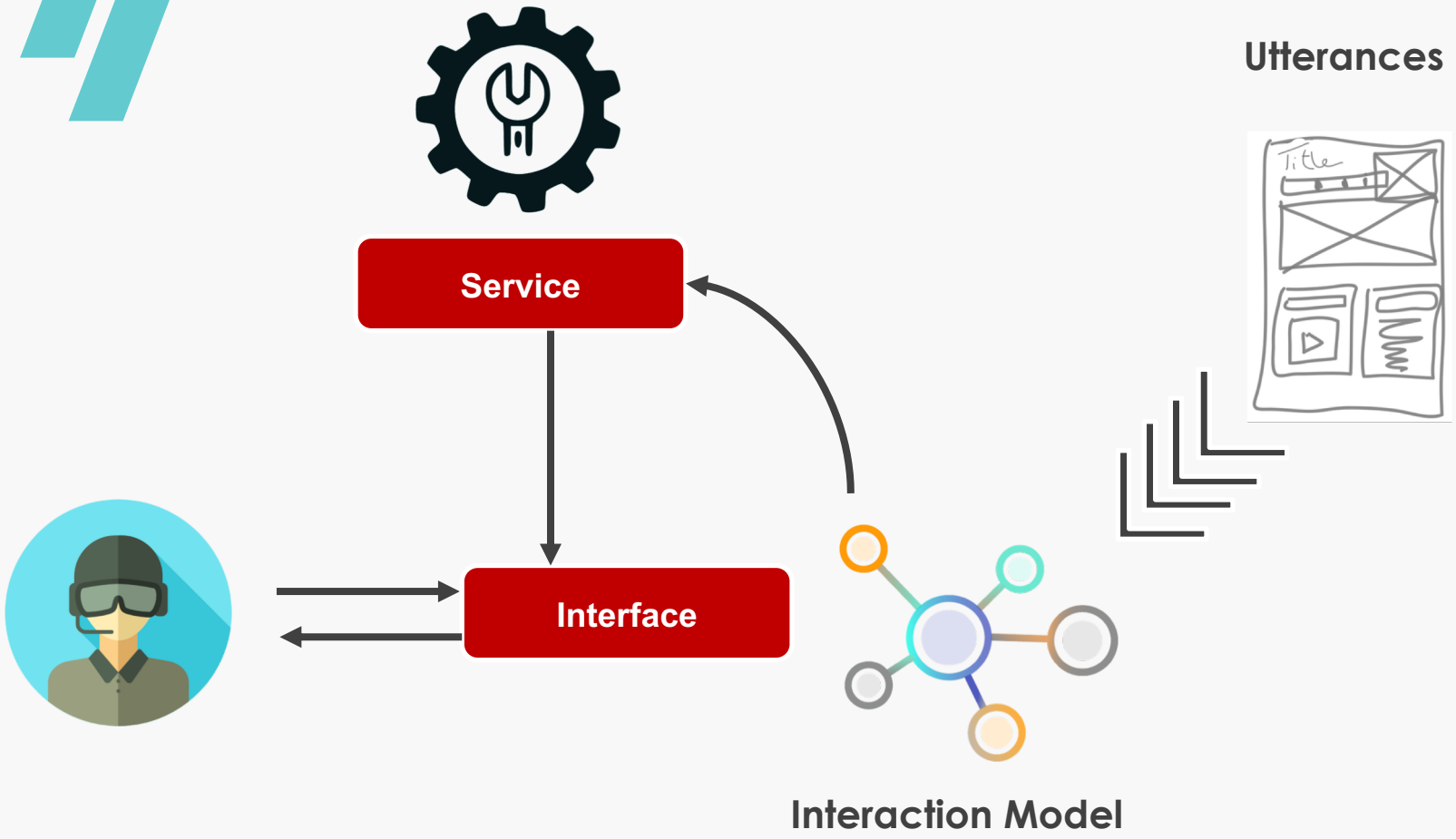
A spoken phrase that maps to an intent. A skill can have many utterances.

Intent

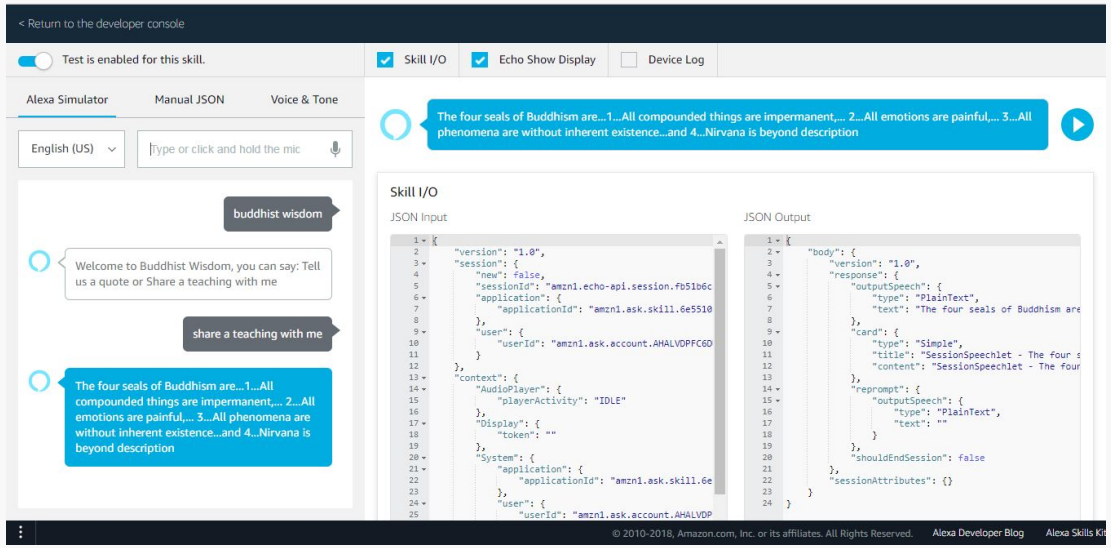
The spoken message is mapped to an intent which decides what code to run within your skill.

LABS

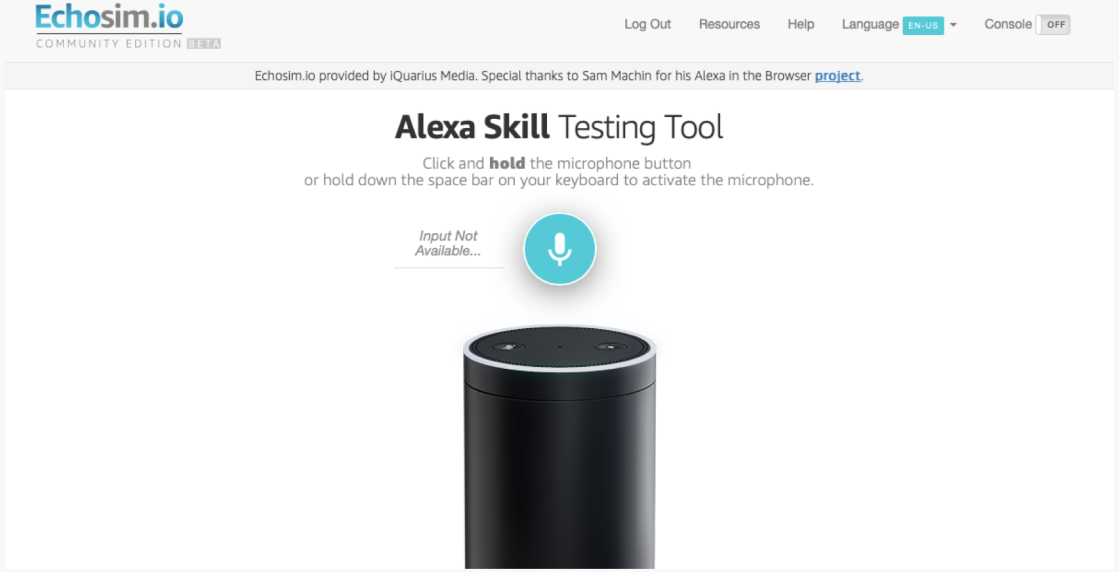
Working
SKILLS



Testing
SKILLS



Simulator



Echosim.io

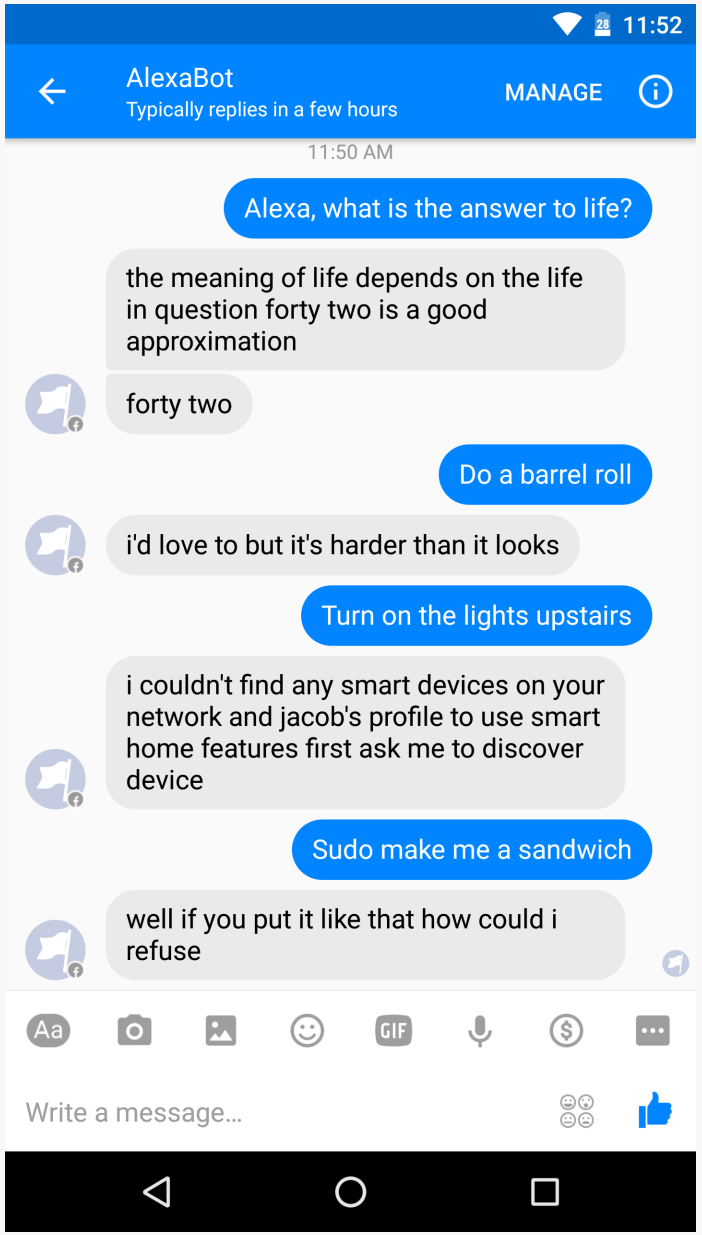
The SCOPE


Internet of Things
Serverless Computing
Python
Creating Applications
Ideas
1
2
3
4
5

What's NEXT

AlexaBot

https://github.com/jacobajit/AlexaBot



What's NEXT

Google Assistant on Your Amazon Echo

https://goo.gl/UEzRGo

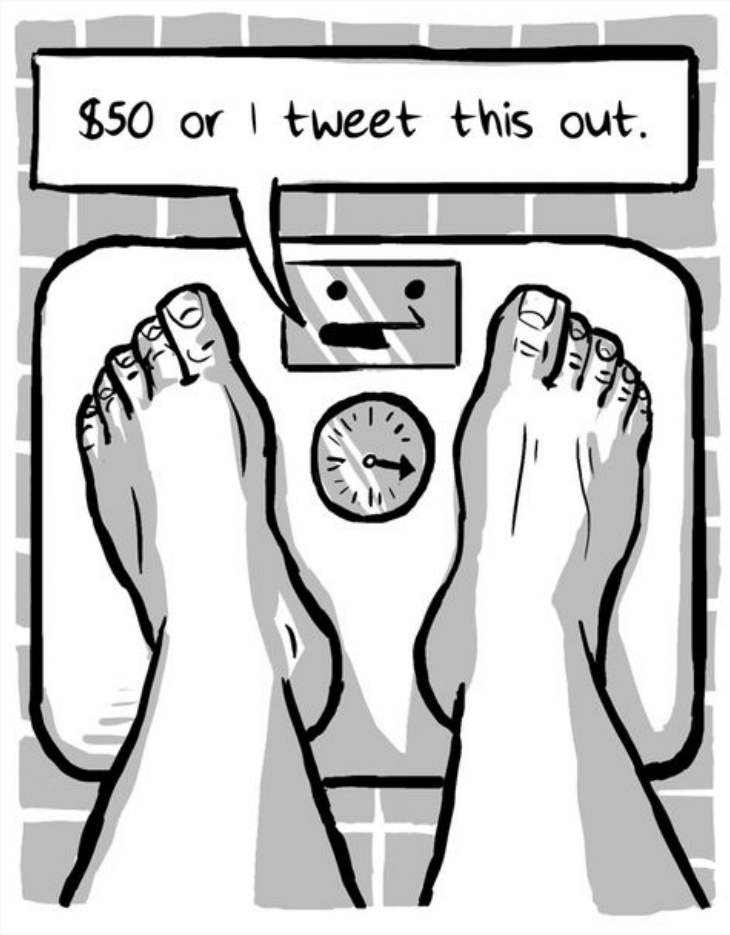
What's NEXT

Using data from fitness trackers . . .

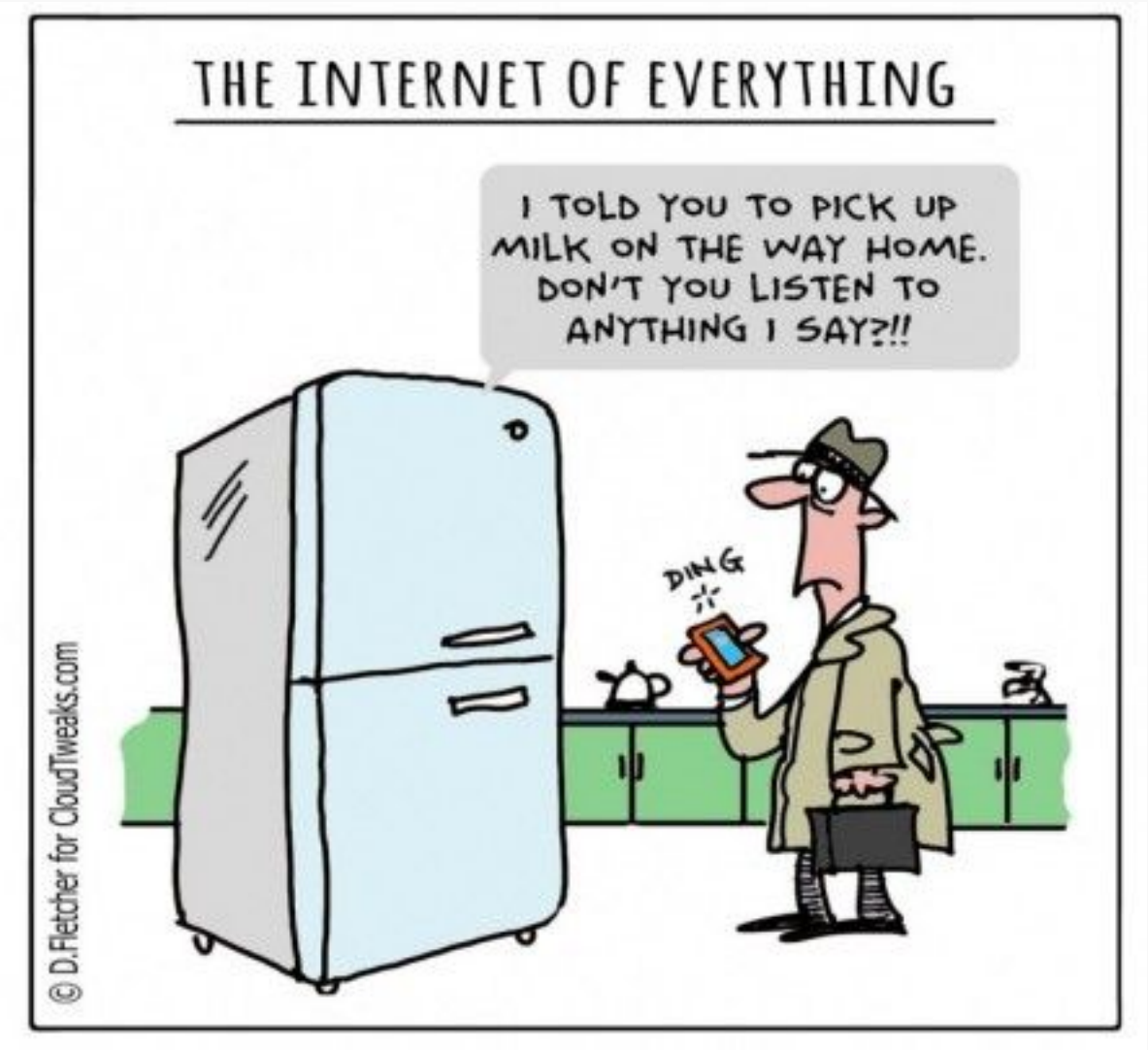
- Continuous HR data – detect spikes, stroke patterns, anxiety
- Calorie estimation from step count, and alert when you need to eat.
- Alerting triggers based on thresholds
- Create APIs to connect other devices like glucometers, and Blood Pressure monitors.
- Publish data to a central store, with web visualization support.

The FUTURE

- Interactive Social Networks
- Interactive Idea boards
- Voice Gaming
- Crisis Alerting
- Office Integration – Meeting room bookings, etc.
- Always aware state!! (Creepy!!)



Smart
PROBLEMS



WHERE WE
ARE HEADED...


Starting another side-project, are we?
Yeah... Well no actually, I've nearly finished the other one so I just wanted to test out this other thing

THAT'S ALL FOLKS

Questions??

You can find the presentation, code and any other tutorials at *https://github.com/sonal-raj/pycon-my-2019*

LABS