IoT in Nutshell

Rui Zhang
https://www.linkedin.com/in/ruixzhang

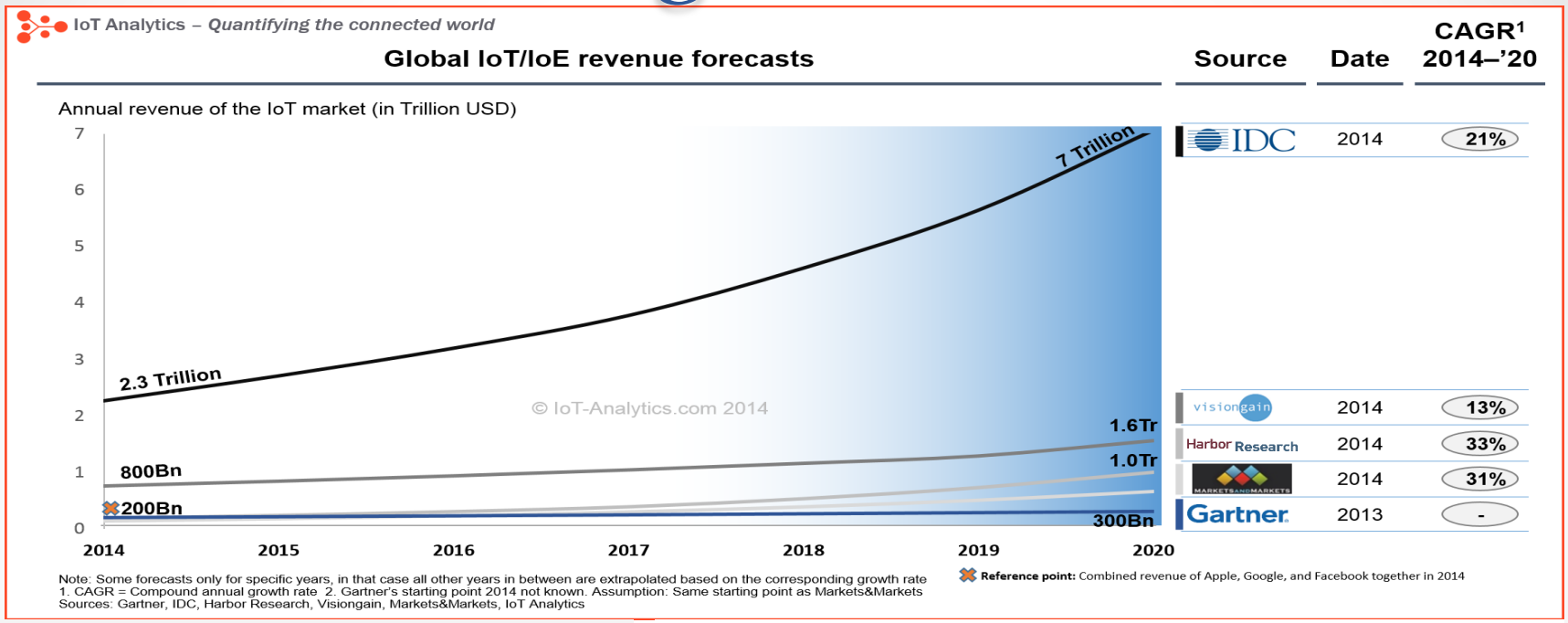
Disclaimer

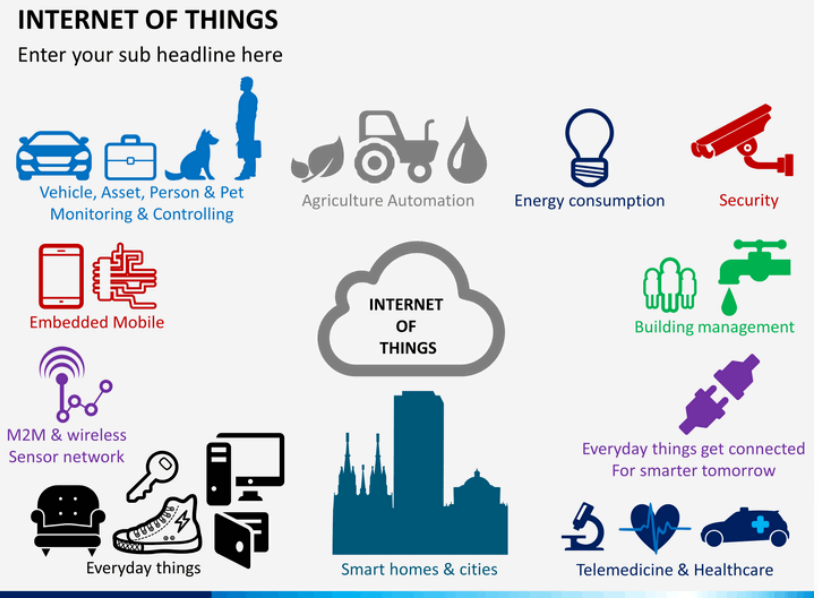
- *"The opinions expressed in this presentation and on the following slides are solely those of the presenter and not representing any company."*

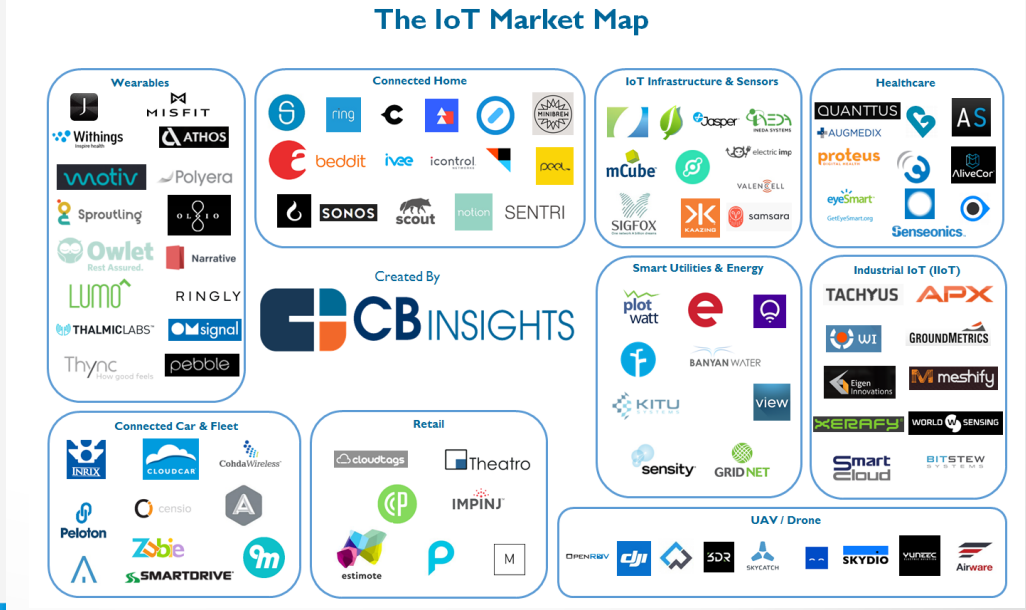
About me

- Background :
  - EE from B.S to Ph.D
  - 10+ years' experience on system research and development
  - PhD research: Wireless networking system, machine learning/information analysis on IoT system, IEEE standard contribution
  - Work experience : Intel CareInnovation, Broadcom
- IoT experience
  - worked on different layers in IoT system: device, OS stack, networking, backend.
  - Delivered large scale IoT product deployed in U.S and E.U
  - Advocate and enthusiast of IoT.

Big Picture

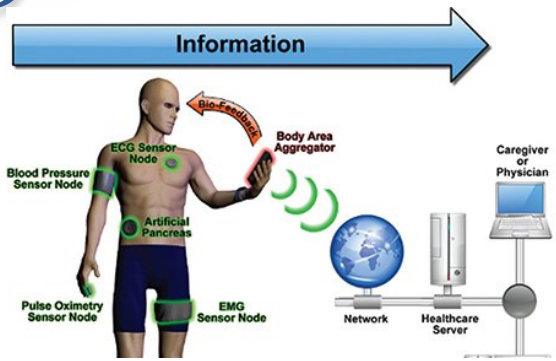

IoT Analytics – Quantifying the connected world
Global IoT/IoE revenue forecasts
Source
Date
CAGR¹ 2014–'20
Annual revenue of the IoT market (in Trillion USD)
7 Trillion
2.3 Trillion
1.6Tr
1.0Tr
800Bn
200Bn
300Bn
© IoT-Analytics.com 2014
IDC 2014 21%
visiongain 2014 13%
Harbor Research 2014 33%
MARKETSANDMARKETS 2014 31%
Gartner 2013 -
Note: Some forecasts only for specific years, in that case all other years in between are extrapolated based on the corresponding growth rate
1. CAGR = Compound annual growth rate 2. Gartner's starting point 2014 not known. Assumption: Same starting point as Markets&Markets
Sources: Gartner, IDC, Harbor Research, Visiongain, Markets&Markets, IoT Analytics
Reference point: Combined revenue of Apple, Google, and Facebook together in 2014

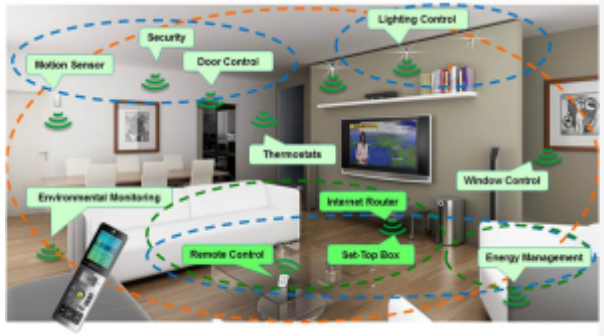

INTERNET OF THINGS
Enter your sub headline here
Vehicle, Asset, Person & Pet Monitoring & Controlling
Agriculture Automation
Energy consumption
Security
Embedded Mobile
INTERNET OF THINGS
Building management
M2M & wireless Sensor network
Everyday things get connected For smarter tomorrow
Everyday things
Smart homes & cities
Telemedicine & Healthcare

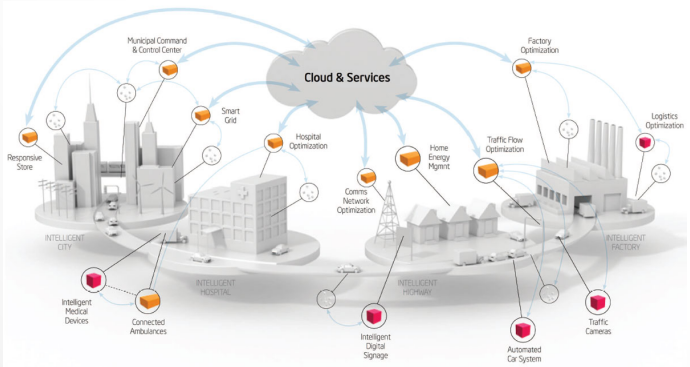

The IoT Market Map
Wearables
Connected Home
IoT Infrastructure & Sensors
Healthcare
Created By
CB INSIGHTS
Smart Utilities & Energy
Industrial IoT (IIoT)
Connected Car & Fleet
Retail
UAV / Drone

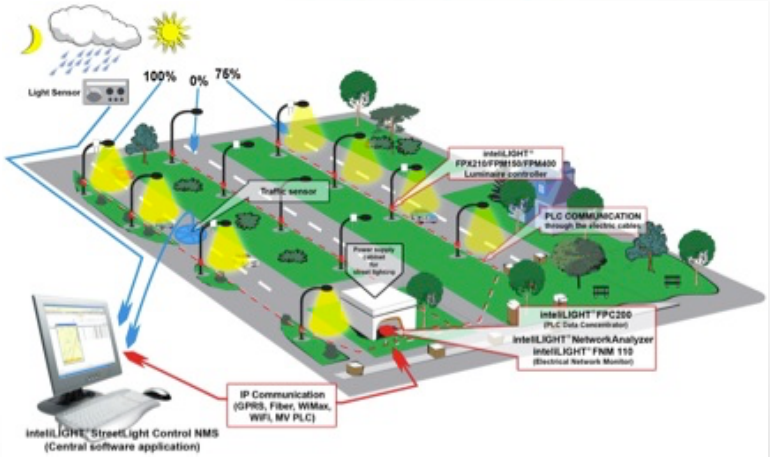
Fragmented Market

- Personal
  - Health/fitness
  - Energy
  - Security
  - Smart home
  - Wearable
- Enterprise:
  - Lighting
  - Smart grid
  - Smart City



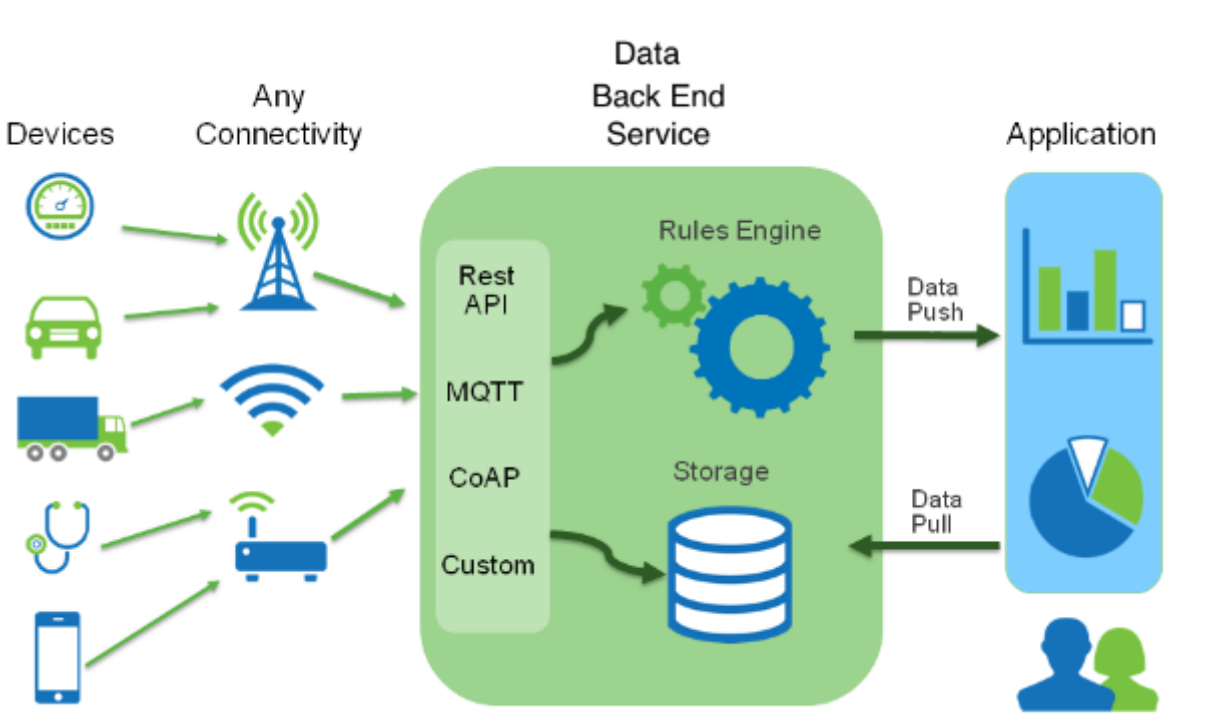






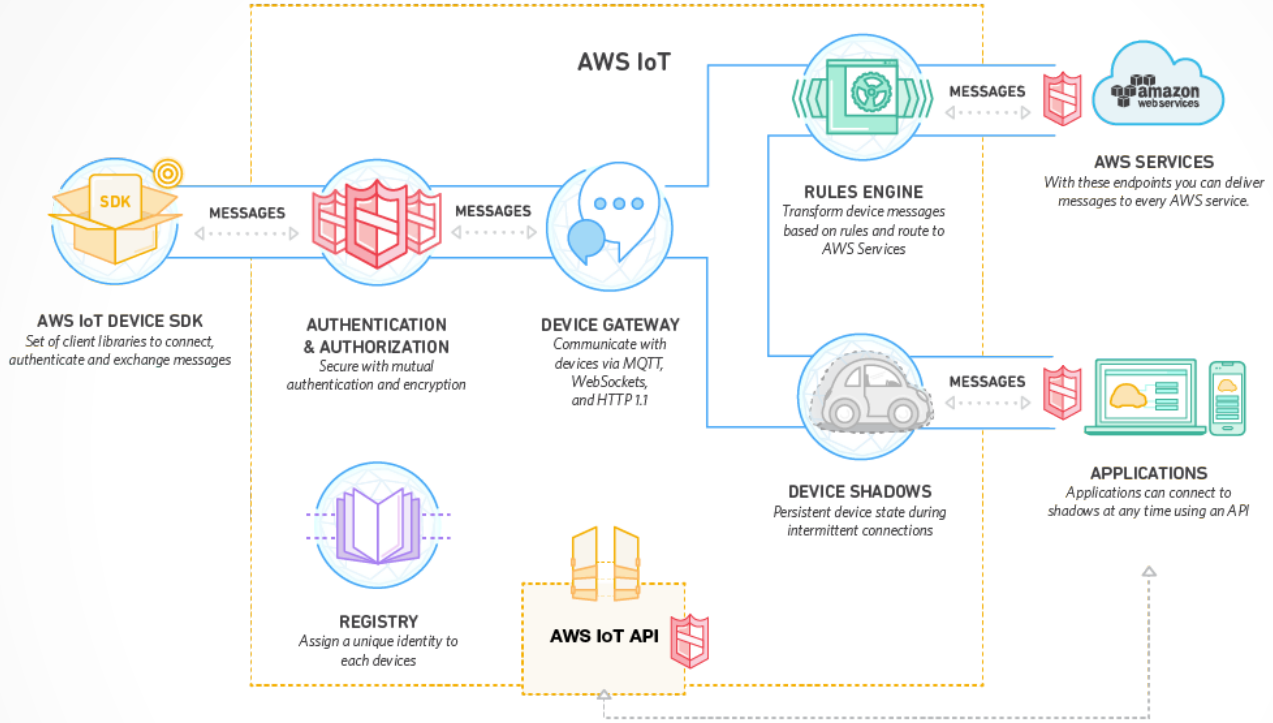
Vertical: Provide integrated service

- HW:
  - Sensor/monitor/…
  - Wireless
  - Processor
- Back end
  - Data storage
  - process/analysis
  - REST API
  - Analysis
- Application
  - Web
  - App
- Tailed to specific market. Total solution

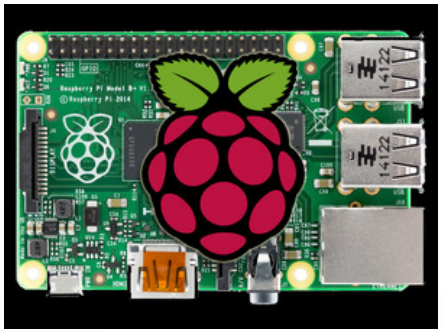


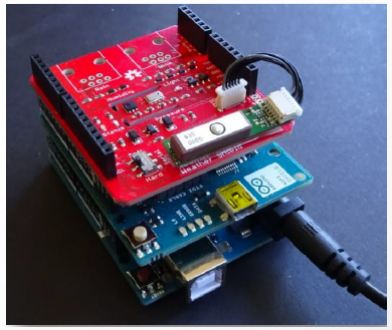
Horizontal: Common Platform

- Backend service/application platform



- HW platform
  - Raspberry Pi
  - Anduino





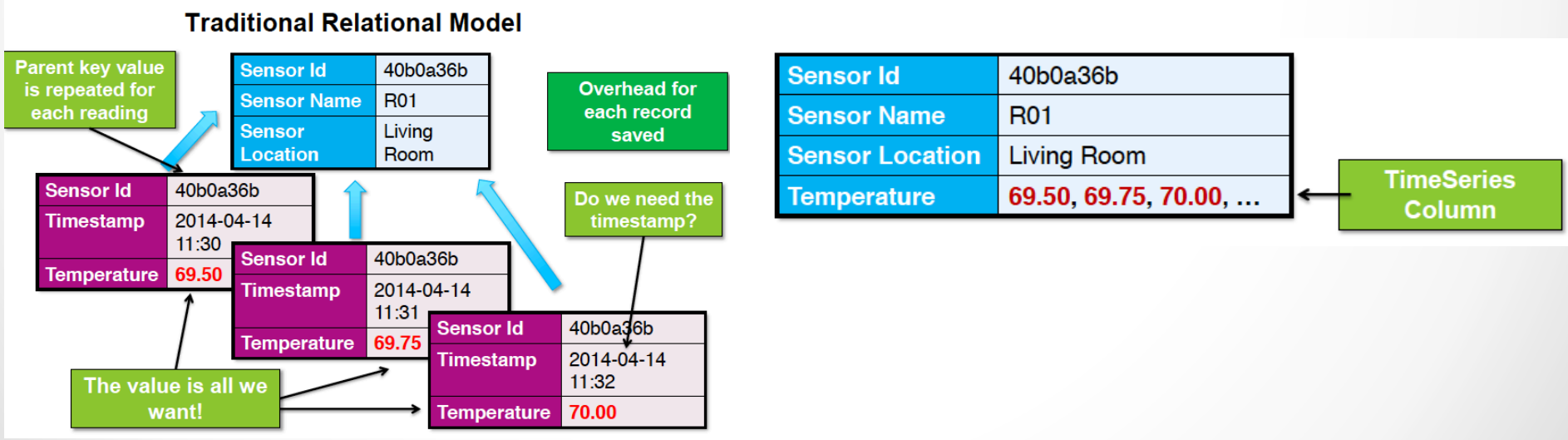
Build Your Own IoT System

Build your own IoT system: Backend

- Similar as other web back end
- IoT specific scenarios
  - Most application is not high concurrent
  - Event driven back end more popular (Node.JS, Python coroutine/Twisted)
  - NoSQL/JSON is more popular(log style data)
  - Serverless: AWS lamda
- Customized for general IOT application
  - Rule engine(IFFFT)
  - Machine learning
- Web service is not necessary needed
  - Rest API still used for extension and potential 3rd party access

Build your own IoT system: Database

- SQL+NoSQL
  - SQL for things, NoSQL for data/stream/log
- Informix:
  - Extended SQL with TIMESERIES
  - sophisticated support for managing time series (time-stamped) data



Build your own IoT system: DataBase

- Informix Usage (http://tinyurl.com/hd6voau)

```
create table sensor(
    sensor_id         char(8),
    xbee_name         char(3),
    sensor_location   char(20),
    sensor_data TimeSeries(sensor_reading)
);
```

```
update sensor
set sensor_data =
   PutElem(sensor_data,
           row("2015-01-18 18:30:00.00000",
               78.1, 45.5, 70)::sensor_reading)
where xbee_name = "R03";
```

```
select *
from sensor
where xbee_name matches "R03";
```

```
insert into sensor values (
   "40b79f8e",
   "R03",
   "Arduino Sensor 2",
   TSCreate(
      'ts_1min',
      '2015-01-18 18:30:00.00000',
      0, 0, 0,
      'sensor_container'));
```

Predefined calendar – new value each minute

```
sensor_id       40b79f8e
xbee_name       R03
sensor_location Arduino Sensor 2
sensor_data     origin(2015-01-18 18:30:00.00000), calendar(ts_1min), containe
                r(sensor_container), threshold(0), regular, [NULL, NULL, NULL,
                 NULL, NULL, NULL, NULL, NULL, NULL, (66.6  ,38.7  ,67    ), (
                65.3  ,40.2  ,67    ), (63.9  ,42.2  ,67    ), (63.3  ,42.7  ,
                66    ), (62.6  ,43.7  ,67    ), (62.1  ,45.0  ,66    ), (61.5
                  ,45.6  ,66    ), (61.2  ,46.6  ,67    ), (60.8  ,47.1  ,67
                  ), (60.3  ,48.1  ,67    ), (60.1  ,48.1  ,67    ), (59.7  ,4
                8.2  ,67    ), (59.5  ,49.2  ,67    ), (59.4  ,48.7  ,67    ),
                ...
```

```
select
   sensor_id,
   xbee_name,
   GetElem(sensor_data, '2015-01-18 18:40:00') as readings
from sensor
where xbee_name matches "R03";
```

```
sensor_id  40b79f8e
xbee_name  R03
readings   ROW('2015-01-18 18:40:00.00000',65.3  ,40.2
,67    )
```

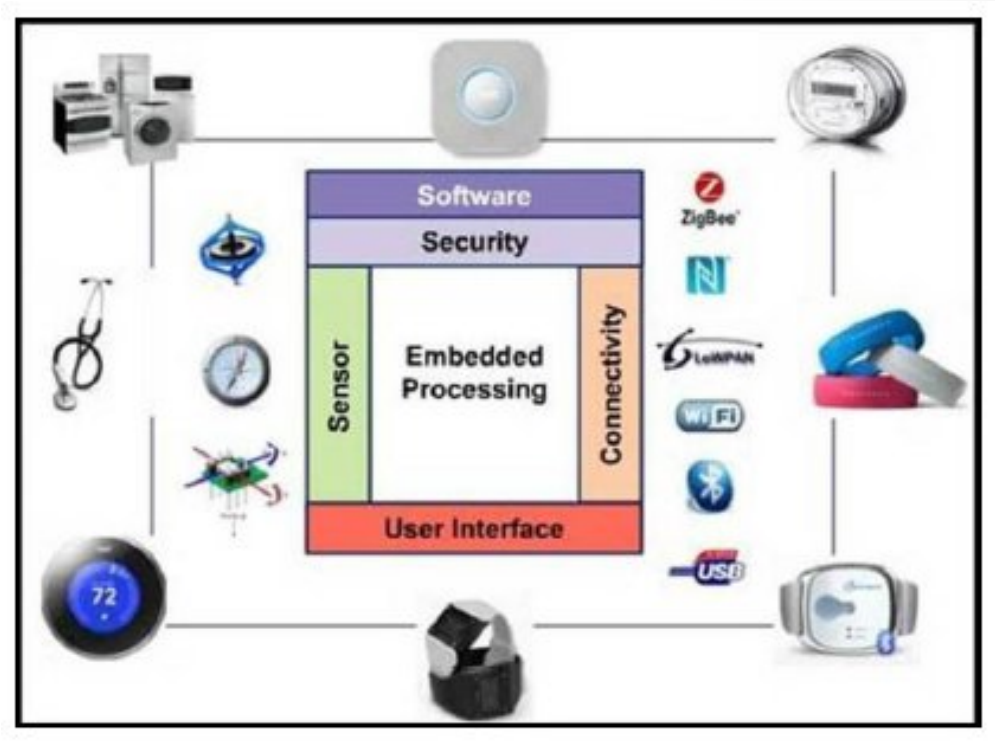
Build your own IoT system: Database

- Postgresql + JSON
  - Postgresql 9.2+
  - MongoDB
- MongoDB is easy to query IoT data
  - Ex: suppose we have device data stream
  - Each data unit has the format of {time:t, value:v}
  - Pymongo

```
from pymongo import MongoClient
from random import randint
from datetime import datetime

try:
    client = MongoClient()
    db = client.test_database
    collection = db.test_collection
    sensor_data = db.sensor_data
    #fake data generation
    sensor_data.insert(dict(date=datetime.now(), value=randint(1, 100)))
    # query mongo DB with data between start and end time, and value smaller than certain threhold
    now = datetime.now()
    start = now.replace(hour=8, minute=0, second=0, microsecond=0)
    end = datetime.strptime("14/07/16 16:30", "%d/%m/%y %H:%M")

    for data_reading in sensor_data.find({"$and": [{'date':{"$gt":start,'$lt':now}},{"value": {"$lt": 50}}]}).sort("value"):
        print data_reading


except pymongo.errors.ConnectionFailure, e:
print "Could not connect to MongoDB: %s" % e
```

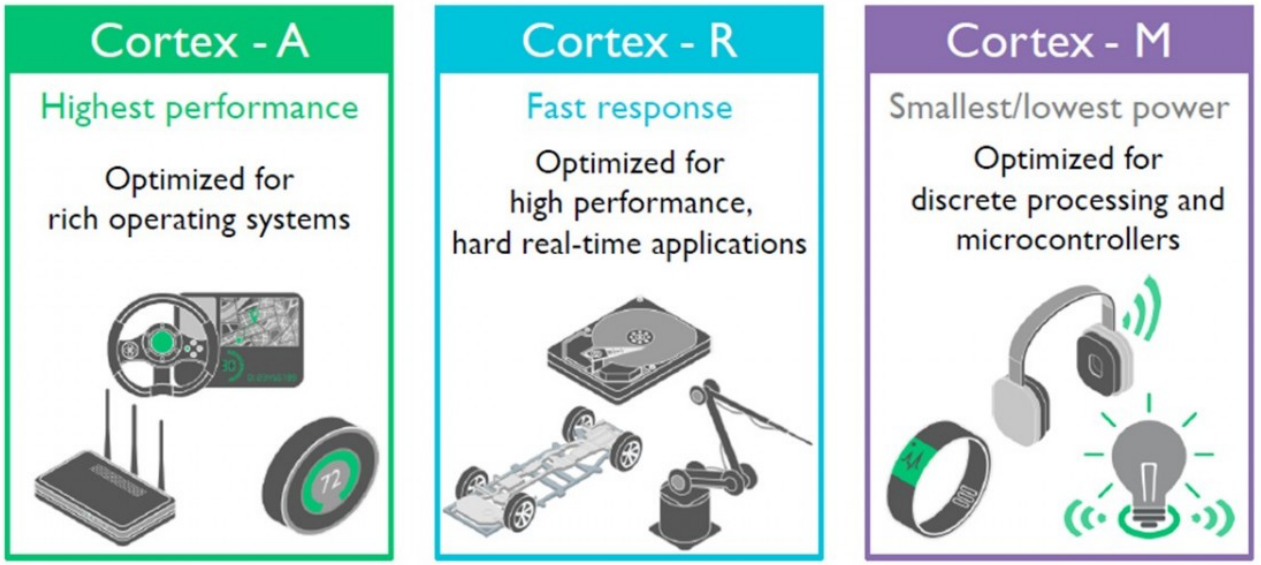
Build your own IoT system: Device

- SOC solution
- Key factors
  - Core Processor
  - Connectivity/Wireless
  - OS
  - Security
- Vendor
  - Reliable
  - Support
  - Tech Doc/Community
  - Channel



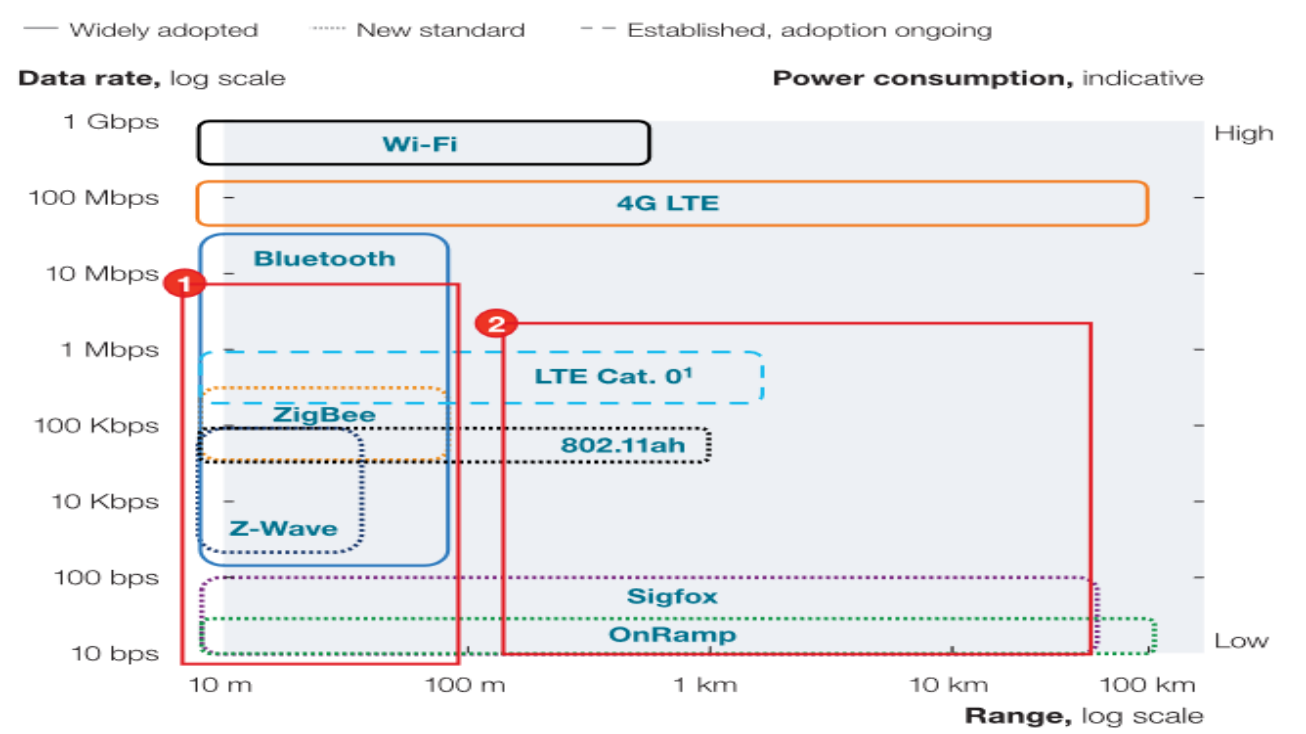
Build your own IoT system: Processor

- ARM processor across all IoT markets
  - Processing power:
  - OS support
  - RTOS requirements
  - Footprint/memory
  - Development resource
- MCU



-

Build your own IoT system: Wireless

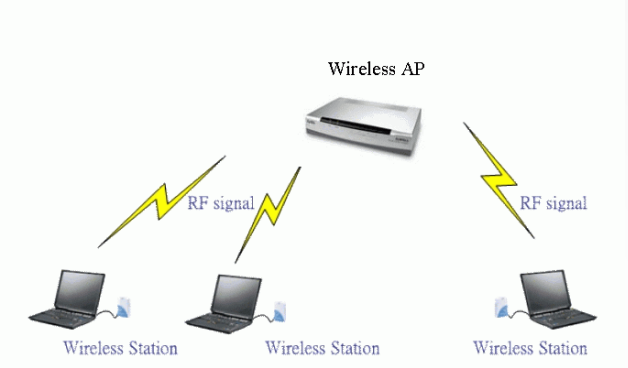


- ISM bands(Free, available world wide)
  - 2.4G is most widely used ISM band
  - Power/penetration/solution available

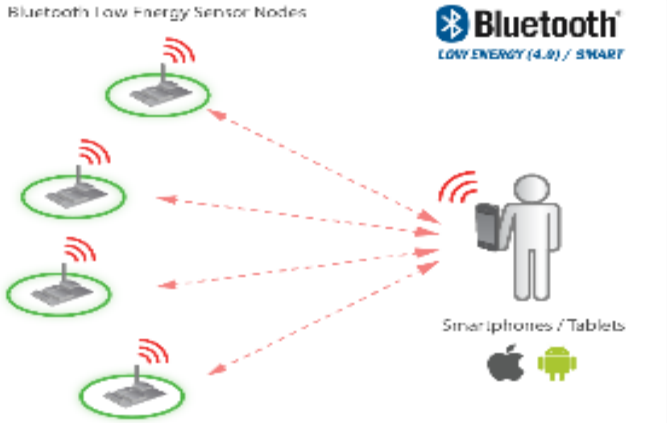
Wi Fi™ Bluetooth® ZigBee®

Build your own IoT system: Wireless

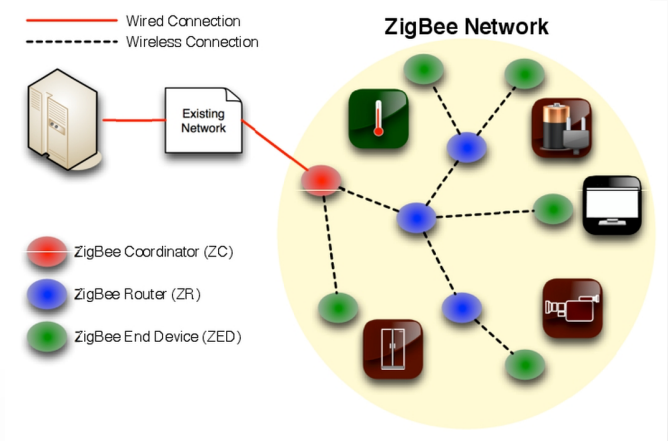
- Wifi
  - AP/station,P2P(One Hop)
  - Pros: High Data rate,
    easy internet access,
    most used and flexible
  - Cons: High power/Solution complexity



- BLE
  - Mater/Slave(One hop)
  - Pros: Low power,
    data rate supporting audio and
    most IoT application.
  - Cons: need to pair with hub/phone



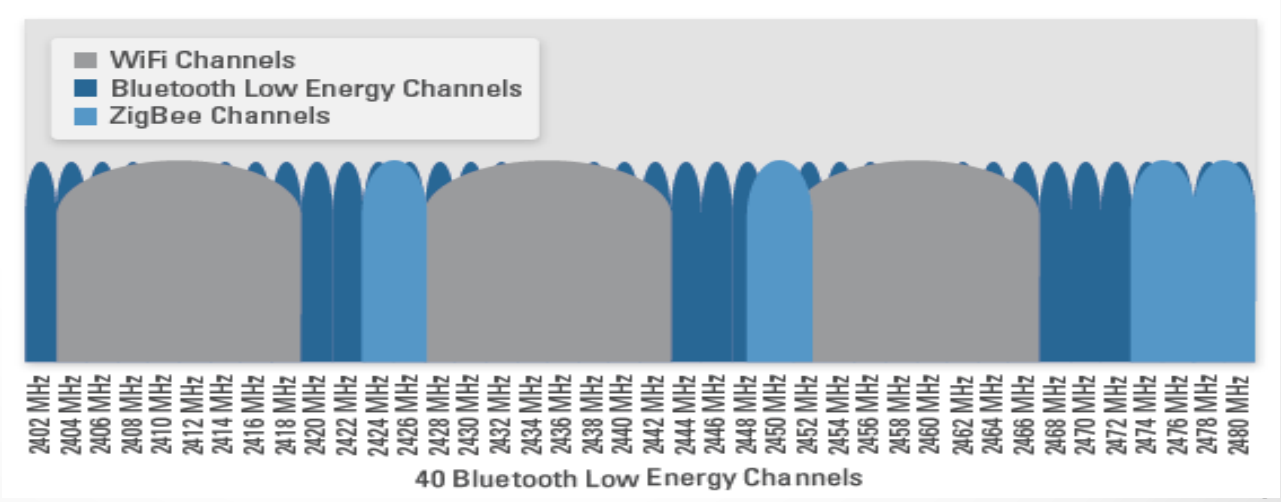
- Zigbee
  - P2P, star, Mesh(multi hop)
  - Pros: Lowest power,
    Self Organized mesh network.
  - Cons: low data rate(sensor application),
    no internet access.



Build your own IoT system: Wireless

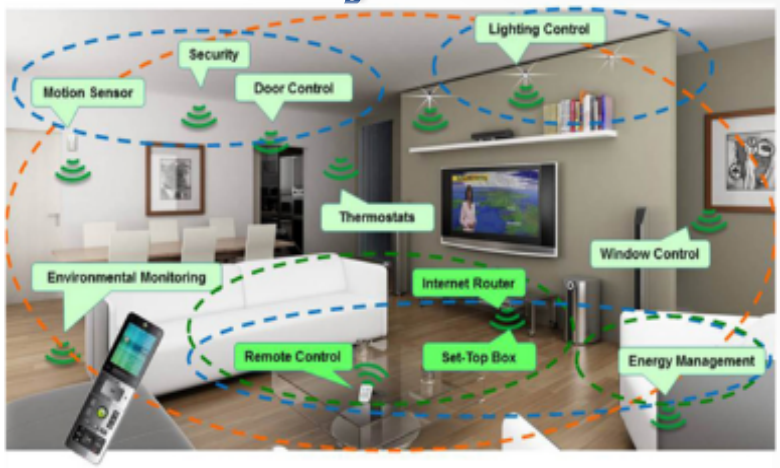
- Rule of thumbs:
  - Mesh capability, Self-organized network: Zigbee
  - Wearable/Audio/directly connected to smartphone): BT/BLE.
  - Internet access (Wifi,LTE),
    - If you need voice capability, LTE
    - Otherwise Wifi

- Interference
  - 2.4 ISM band contention.
    - BLE:AFH(Adaptive Frequency hopping)
    - Zigbee: 4 golden channel.
    - Wifi:5G



Build your own IoT system: OS

- Bare bone
  - Simple and widely used.
    - No Traditional OS,
    - No user/kernel
  - Single thread:
    - ISR and event driven
    - Msg for Inter-Task



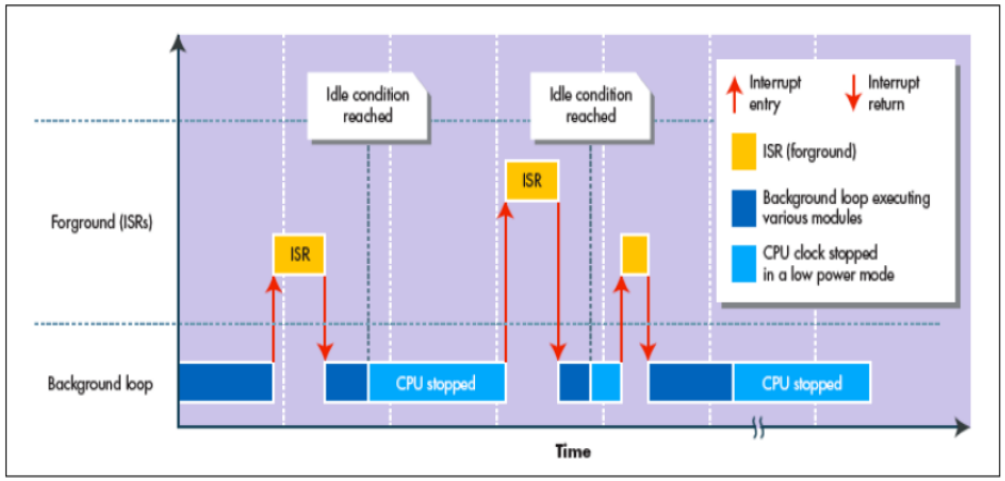


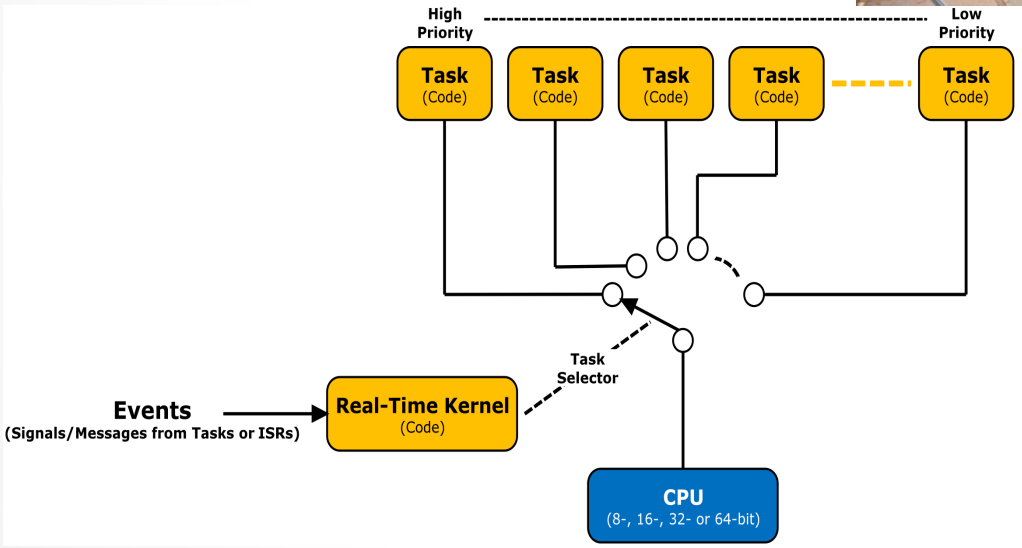
Figure 1: Foreground/background system with a low-power sleep mode.

```
while (true)
{
HAL_ISR();
task_1(task1_event_bitvector);
...
task_n(taskn_event_bitvector);
sleep();
}
```

  - Memory management
    - No MMU, manual management(write your own malloc).

Build your own IoT system: OS

- RTOS
  - Time Critical system.
  - Multi thread support
    - Time guaranteed task
    - Priority based schedule



  - Memory management
    - Minimum dynamic management/Mainly Static allocation
  - GreenHill, Threadx, Vxworks, FreeRTOS

Build your own IoT system: OS

- *nix like (iOS/Android tailed)

  - Need to support rich feature application, 3rd party applications
  - Act as hub to connecting other IOT devices
  - Modern OS support
    - User/kernel
    - Multi thread
    - MMU
    - Driver
  - Power requirement is high
  - More development resource

IoT system problem

High data error rate in network?

Wireless

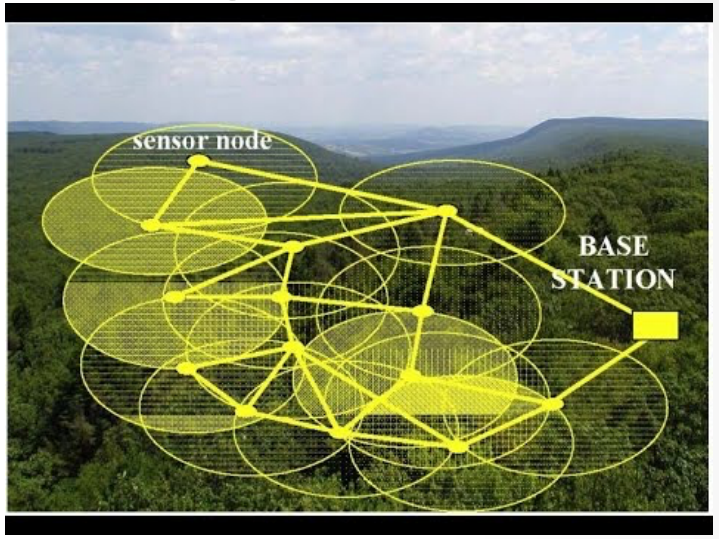
Interference

Link Loss

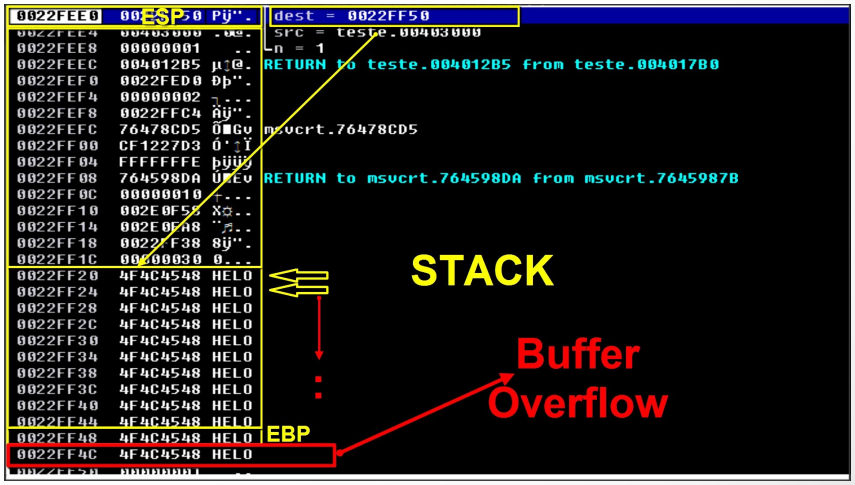
Stack/OS

Packet Overflow

Routing error/timeout

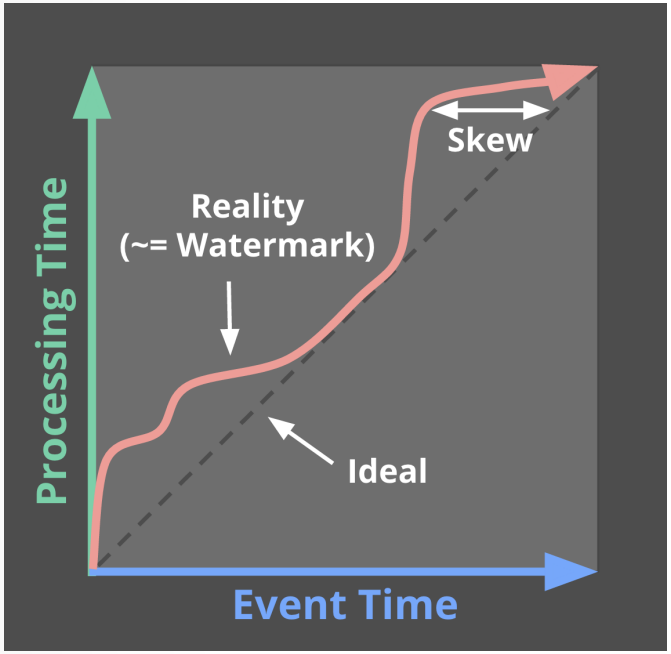


Memory management Bug

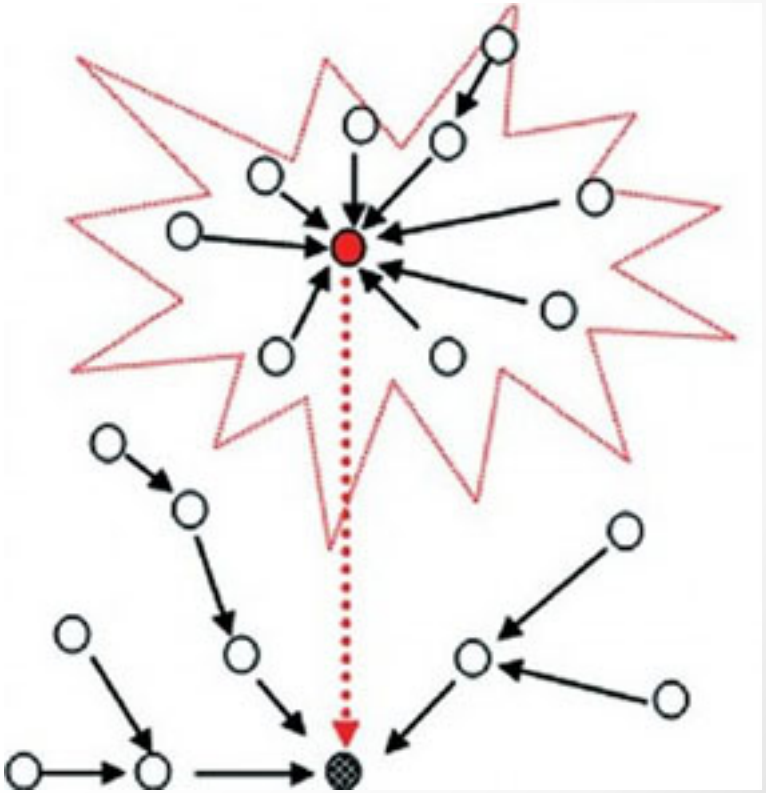


Application

Event based processing delay/discard



Unbalanced traffic



Conclusion

- IoT markets are fragmented :
  - Diversified application
  - No one size fits all solution
- Pick up your Legos for your Legoland!
  - Back End service
    - Battle test solution on cloud
    - Event driven/serverless
    - SQL and NoSQL
  - Device side
    - Processor and OS
      - ARM core/MCU
      - OS choices:
    - Wireless: wifi, bluetooth and zigbee
- Solve the real problem, not just connecting devices