

Putting the Internet back into IoT

Or how you should / should not use the cloud

Let us lay some ground works...

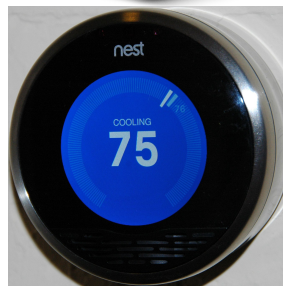
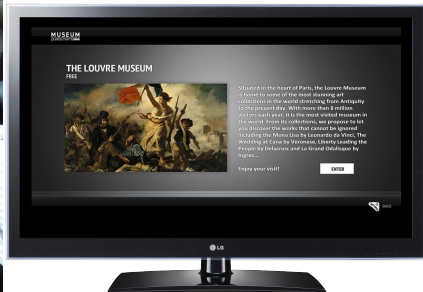
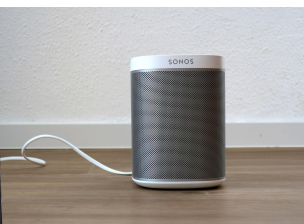
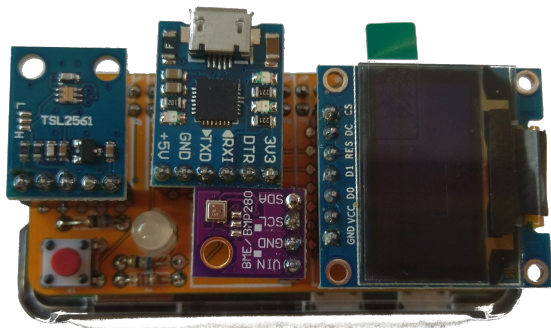
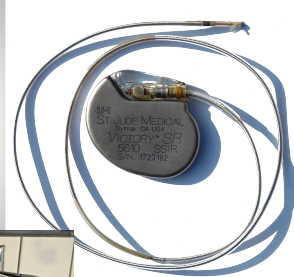
What most “home” networks look like:

Firewall

Main
Network

Wireless
Guest

More Groundwork: IoT devices



Typical ways devices connect to the Internet

- Through a Gateway:
 - Bluetooth
 - Z-wave
 - 802.11.6
 - Zigbee
 - IR
 - Smoke Signals
 - Carrier Pigeons
- Directly:
 - Wifi
 - Ethernet
- Using:
 - IPv4
 - IPv6

Lets come back to this for a minute to talk about IPv4 vs. IPv6

Firewall

Main
Network

Wireless
Guest

Local Access vs. Remote Access

- IPv4 - Local

- Direct Access
- Straight Forward
- Mostly ubiquitous support

- IPv4 - Remote

- NAT traversal
- Punching holes in firewalls
- Port Forwarding
- UPNP
- Cloud reverse proxies

- IPv6 - Local

- Direct Access
- Straight Forward
- Getting more ubiquitous but not there

- IPv6 - Remote

- Direct Access
- Punching holes in firewalls
- UPNP
- Cloud based IP lookup (and/or reverse proxies)

Some general words of caution...

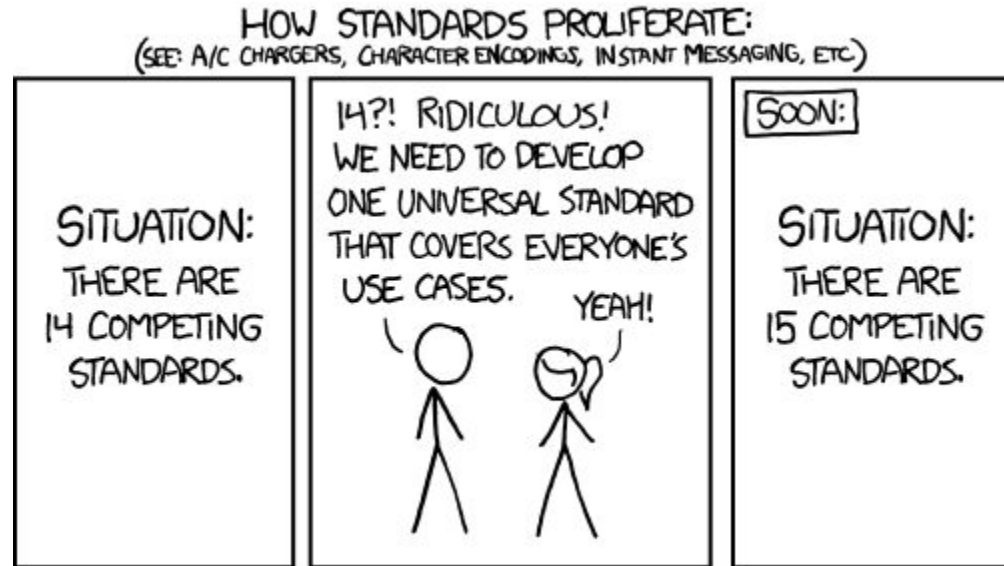
- Think about what you are using the Internet for
- Be mindful of where your services live
- Sometimes UX the user can use may make you less secure
- Always change the default passwords!
- Make it possible to do things without auto-discovery
- Don't always assume you are on the same network as the device
- Upgrade schemes need to be done

Shifting gears & talk about how to talk to the devices

But the real advantage to IoT is the I - Internet!

Lots of good ways to do this...

- MQTT
- Liota
- AMQP
- STOMP
- RabbitMQ
- REST
- WAMP
- ZeroMQ
- Java Message Service (JMS)
- CoAP
- CLOUD!
- XMPP-IOT
- XMPP
- etc.....



Now lets talk about something to try

- MQTT - Mosquitto, MQTT broker, good for local passing of data
- Think of it as a message bus on the network
- Clients Subscribe to Topics that can be hierarchical, and listen to the Topic
 - /myhome/groundfloor/livingroom/temperature for example
 - You can listen at any level of the hierarchy, anything below your level will be filtered to you
 - Wildcards, +, are allowed /myhome/+/temperature
- Devices Publish data to topics
 - The data is freeform, the receiving end is expected to interpret it

Lets just try listening...

On your laptop/VM:

```
yum install mosquitto
```

```
apt-get install mosquitto-clients
```

then

```
mosquitto_sub \  
  -h 10.111.0.5 \  
  -t "pugnose/temp/core0" \  
  -u "ale" \  
  -P "Penguins"
```

Expected output:

+67.0°C

What's running on "pugnose":

```
while [[ 1 ]];do \  
  mosquitto_pub \  
    -h 10.111.0.5 \  
    -t "pugnose/temp/core0" \  
    -m "$( \  
      sensors | \  
      grep "Core 0" | \  
      tr " " "\n" | \  
      grep "°" | \  
      head -n 1 \  
    )" \  
    -u "ale" \  
    -P "Penguins"; \  
  sleep 10;\  
done
```

Next Up, adding the Raspberry Pi

Server Side:

- MQTT server
 - Mosquitto
 - Default config w/ username/password set

Client Side

- apt-get install mosquitto mosquitto-clients
- On your laptop / VM / not the Raspberry Pi
 - ```
mosquitto_sub \
-h 10.111.0.5 \
-t "topic_name" \
-u "ale" \
-P "Penguins"
```
- On the Raspberry Pi
  - ```
while [[ 1 ]];do \  
mosquitto_pub \  
-h 10.111.0.5 \  
-t "topic_name" \  
-m "hello world" \  
-u "ale" \  
-P "Penguins"; \  
sleep 10;\  
done
```

What this is doing:

The MQTT broker is passing published messages (`mosquitto_pub`) to subscribed (`mosquitto_sub`) clients.

What this is doing:

The MQTT broker is passing published messages (mosquitto_pub) to subscribed (mosquitto_sub) clients.

```
mosquitto_sub \ <----- The Command
  -h 10.111.0.5 \ <---- The (-h)ost
  -t "topic_name" \ <-- The (-t)opic
  -u "ale" \ <----- The (-u)sername
  -P "Penguins" <----- The (-P)assword
```

Output:

```
hello world
```

```
mosquitto_pub \ <----- The Command
  -h 10.111.0.5 \ <---- The (-h)ost
  -t "topic_name" \ <-- The (-t)opic
  -m "hello world" \ <- The (-m)essage
  -u "ale" \ <----- The (-u)sername
  -P "Penguins" <----- The (-P)assword
```

Output:

```
<none unless error>
```

So Lets try this:

Laptop:

```
mosquitto_sub \  
  -h 10.111.0.5 \  
  -t "ale/<your name no spaces>/test" \  
  -u "ale" \  
  -P "Penguins"
```

Output:

hello world

Raspberry Pi:

```
mosquitto_pub \  
  -h 10.111.0.5 \  
  -t "ale/<your name no spaces>/test" \  
  -m "hello world" \  
  -u "ale" \  
  -P "Penguins"
```

Output:

<none unless error>