

# THE INTERNET OF THINGS: GETTING TO SELF-MANAGED NETWORKS

---

Henning Schulzrinne

(+ Jan Janak, Kyung-Hwa Kim, Andy Xu & other CUCS IRT contributors)

NetSys 2017, Göttingen



This material is based upon work supported by the National Science Foundation under Grant No. (CNS-1218977). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

*The views and opinions expressed in this presentation are those of the author and do not necessarily reflect the official policy or position of any agency of the U.S. government. Any resemblance to actual policies, living or dead, or actual events is purely coincidental.*

# Natural evolution





# IoT is not exactly new (1978)



X10 HOME AUTOMATION ▾

X10 PRO ▾

HOME SECURITY

CAMERAS

X10 B

ome → X10 Home Automation

## X10 Home Automation



SWITCHES



MODULES



RECEPTACLES



CONTROLLERS

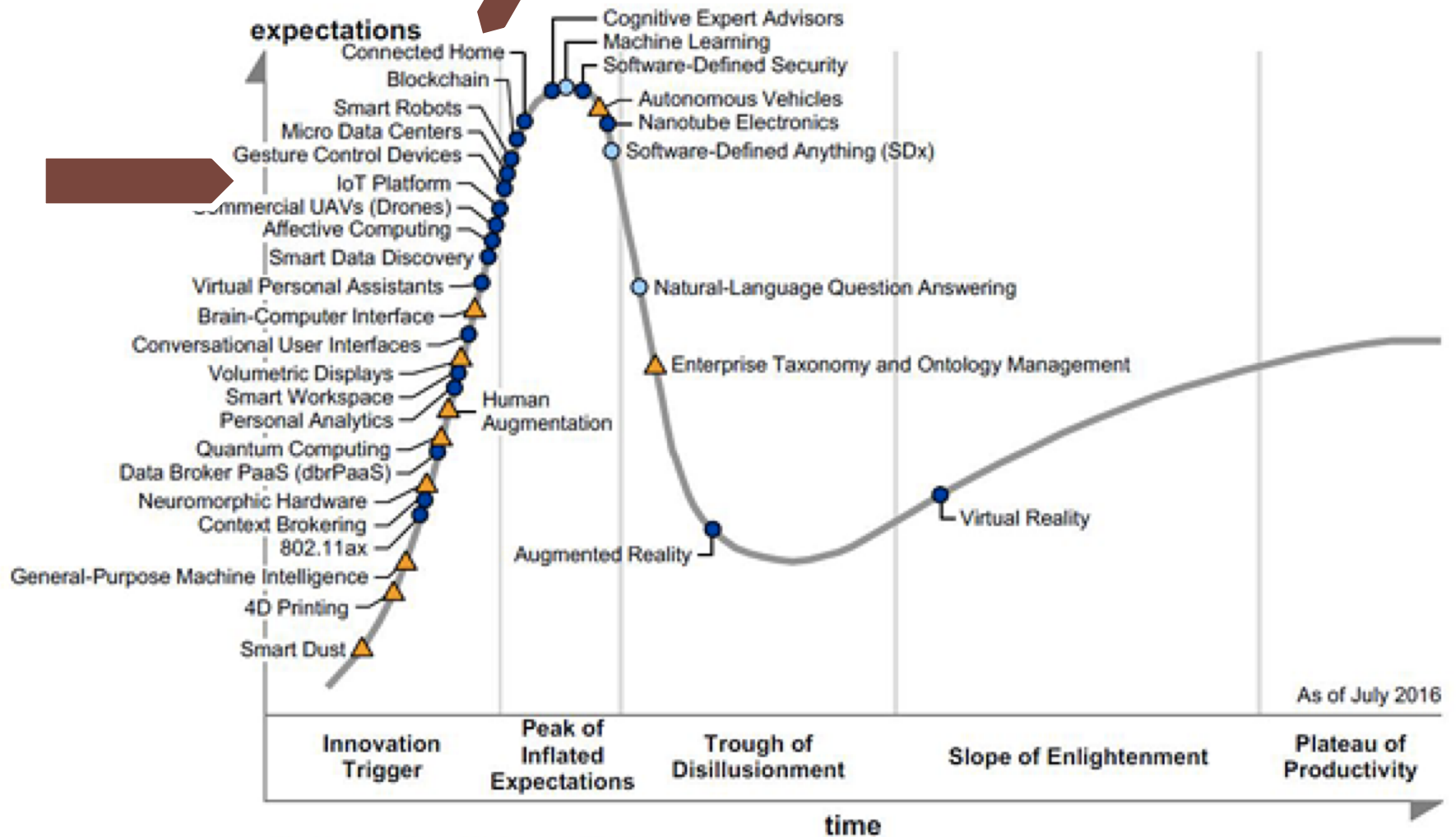
# IoT – an idea older than the web (1985)

Peter Lewis (panel discussion 1985)

*By connecting devices such as traffic signal control boxes, underground gas station tanks and home refrigerators to supervisory control systems, modems, auto-dialers and cellular phones, we can transmit status of these devices to cell sites, then pipe that data through the Internet and address it to people near and far that need that information. I predict that not only humans, but machines and other things will interactively communicate via the Internet. **The Internet of Things, or IoT, is the integration of people, processes and technology with connectable devices and sensors to enable remote monitoring, status, manipulation and evaluation of trends of such devices.** When all these technologies and voluminous amounts of Things are interfaced together -- namely, devices/machines, supervisory controllers, cellular and the Internet, there is nothing we cannot connect to and communicate with. What I am calling the Internet of Things will be far reaching.*



From Chetan Sharma Consulting 2016



# Kids, don't do this at home



# HUGGIES™ Tweet Pee

The first diaper that tells mommy when it's time to change.

The comfort of Huggies was our number one priority.

Appropriate with full social features.

Even the most special babies that make us diapers.

**Actual size**

**Situation**  
Mummy is busy or being a mommy so she doesn't always know when I need a change. And I can't talk yet so it's hard to tell her when I need one.

**Idea**  
TweetPee is a diaper gadget that sends instant "diaper change" notifications, saves money by preventing unnecessary changes, and directs mommy to buy diapers on-line.

**Design**  
Huggies created a durable, cute and functional device. It's tough enough to survive my diapers and it is portable for me to take it off and play with it. Believe me, I tried! Besides the little beak, they were able to combine a "social media" interface that sends and reads tweets that last 3 days with their own recharge.

**Results**  
Huggies is proud that diaper innovation can go beyond just comfort and absorption for babies. There are so many of the possibilities that our tiny, tiny diaper will be able to speak for you! And Babies like me use this and TweetPee!



# Towel dispensers

## Power over ethernet powered paper towel dispensers

WO 2014028808 A1

### ABSTRACT

A system for providing power to a plurality of paper towel dispensers (10) through a power over ethernet (PoE) network (14) and for sensing various operational parameters of the dispensers (10) and communicating those parameters through the network to a central computing device (16). The system includes a Data/Power controller (12) associated with each of the dispensers (10) for providing power (26) to the dispensers (10) and for sending and receiving data (24) between one or more sensors in the dispensers (10) and a central computer device (16).



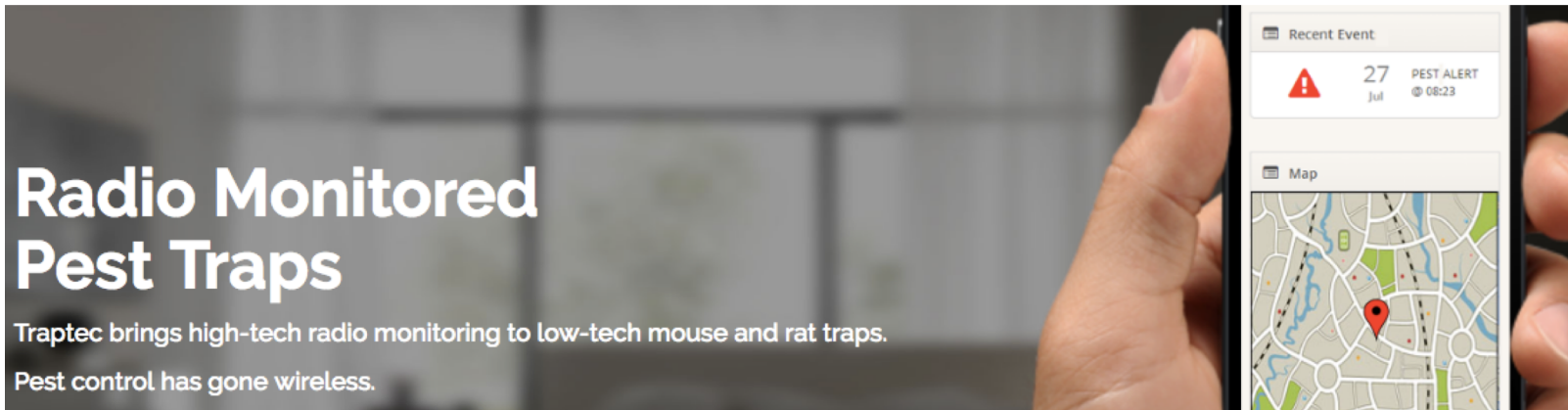
The IoT has already been used for a range of use cases in facilities management. For example, Coor has worked with a paper towel manufacturer in Sweden to implement automated monitoring of dispensers. Sensors fitted to each dispenser monitor its fill level, and send an alert to the building manager, who can make sure it is refilled before it becomes empty.

# The IoT killer app

## Radio Monitored Pest Traps

Traptec brings high-tech radio monitoring to low-tech mouse and rat traps.

Pest control has gone wireless.



<http://www.traptec.eu/>

# link.nyc & smart trash cans



GPRS or CDMA  
GPS location service



## But controlling light switches is still not the best use

Want to turn on the bedroom light? Sure, just pick up your smartphone, enter the unlock code, hit your home screen, find the Hue app, and flick the virtual switch. Suddenly, the smart home has turned a one-push task into a five-click endeavor, leaving Philips in the amusing position of launching a new product, [Tap](#), to effectively replicate the wall switches we always had.

# Where does IoT make sense?

- Probably

- home security
- residential & commercial locks
- home medical (recording)
- housekeeping (restroom supplies)
- outdoor lighting
- parking meters
- vending machines

- Not so much

- light switches
- most household appliances
- clothing
- smoke detectors?

not cost-effective, not just useless

# Two kinds of IoT devices

< \$20

- BlueTooth, ZigBee, proprietary L2
- connected only via gateway
- fixed-function: sense or activate
- single chip transceiver + MPU
- only use L2 security
- similar to peripherals

> \$50

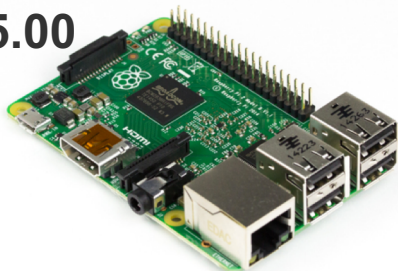
- Wi-Fi, LTE-M, LoRa, SIGFOX
- direct connection to Internet possible
- SOC + network module
- run (small) Linux stack
- programmable
- TLS and kin easy

# Sensor networks may be (tiny) niche

- Most IoT systems will be near power since they'll interact with energy-based systems (lights, motors, vehicles)
- Most IoT systems will **not** be running TinyOS (or similar)
- Protocol processing overhead is unlikely to matter
- Low message volume → cryptography overhead is unlikely to matter
  - exceptions: light switches and similar 1-function I/O devices → BT/Zigbee fixed-function devices

In particular, according to the indexes, a Raspberry Pi is about **seven** times as fast as a baseline SPARCstation 20 model 61 — and has substantially more RAM and storage, too. And the Raspberry Pi 2 is **sixteen** times as fast at single-threaded tasks, and on tasks where all cores can be put to use it's **forty one** times faster.

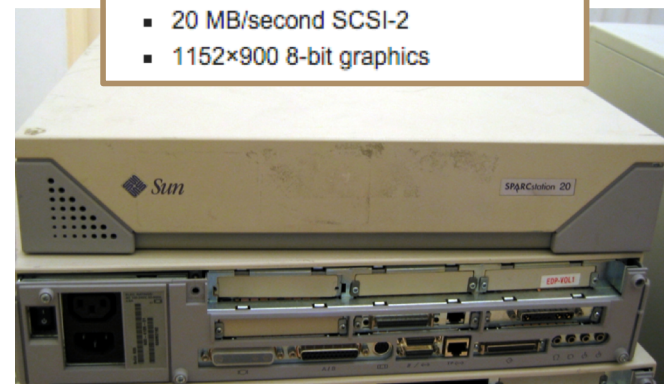
\$35.00



- A 900MHz quad-core ARM Cortex-A7
- 1 GB RAM

16-41x

- One 60 MHz SuperSPARC CPU
- 1 MB of cache
- 32MB RAM (expandable to 512MB)
- 20 MB/second SCSI-2
- 1152×900 8-bit graphics



# SCALING IOT UP

---

# Scaling IoT up



one  
device

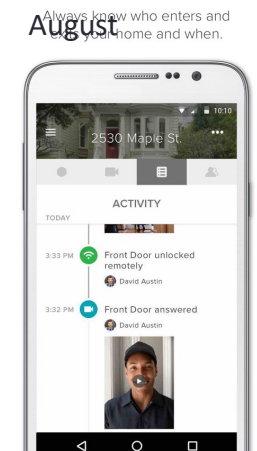
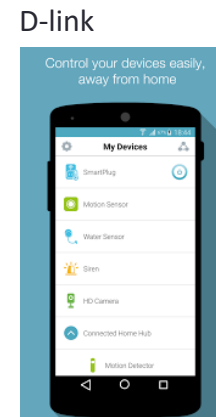
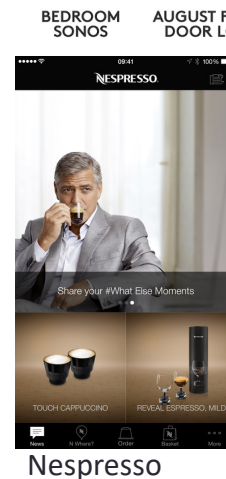
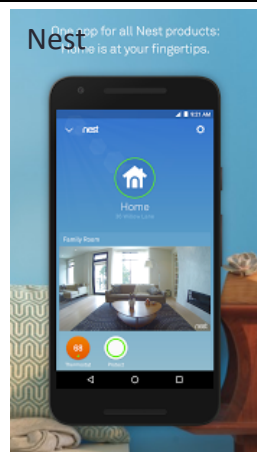
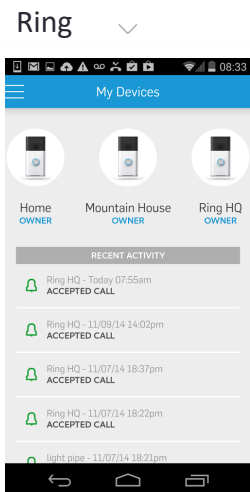
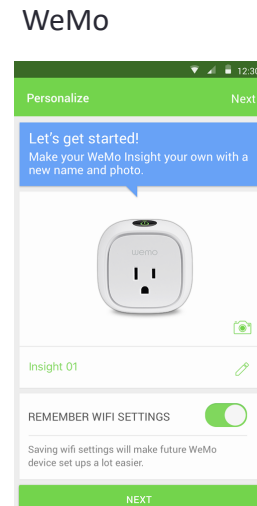
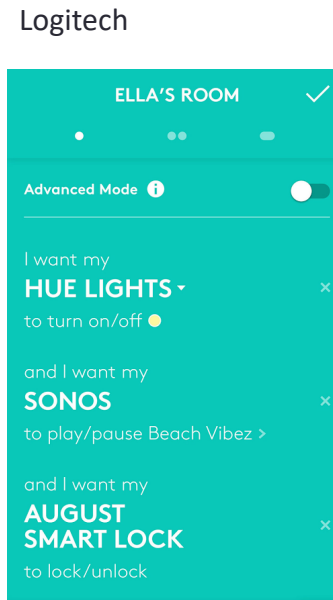
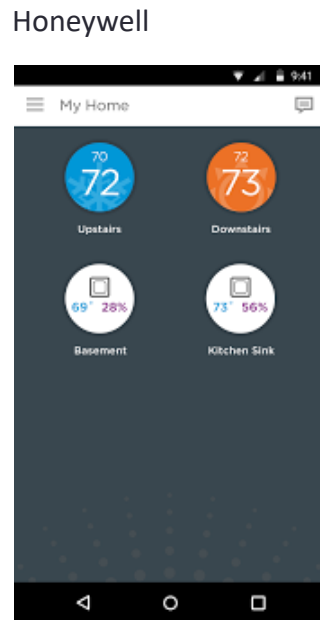
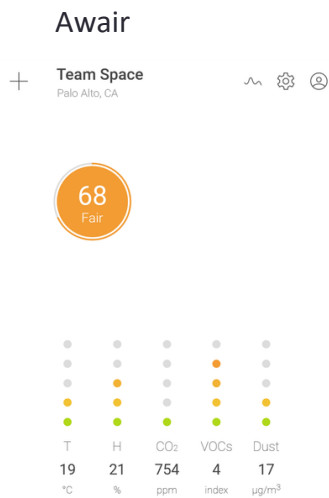
apartment  
building  
( $10^2 - 10^4$ )

city+

( $10^6 - 10^8$ )



# One Thing, one app





# Does not scale well to real-world sizes



Le Lignon, Switzerland (0.7 mi long): 2,780 apartments



Camden NoMa (DC): 405 apartments

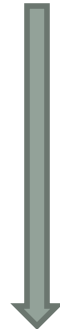


# IoT = Internet at scale

- *Security* at scale
  - still largely “add password to configuration file”
  - identify by IP address
- *Management* at scale
  - device-focused
  - SNMP, at best
  - CLI, at worst
  - no performance diagnostics capabilities (“why is this so slow?”)
- *Naming* at scale
  - identify by node name
- *Programming* at scale



system  
& rack

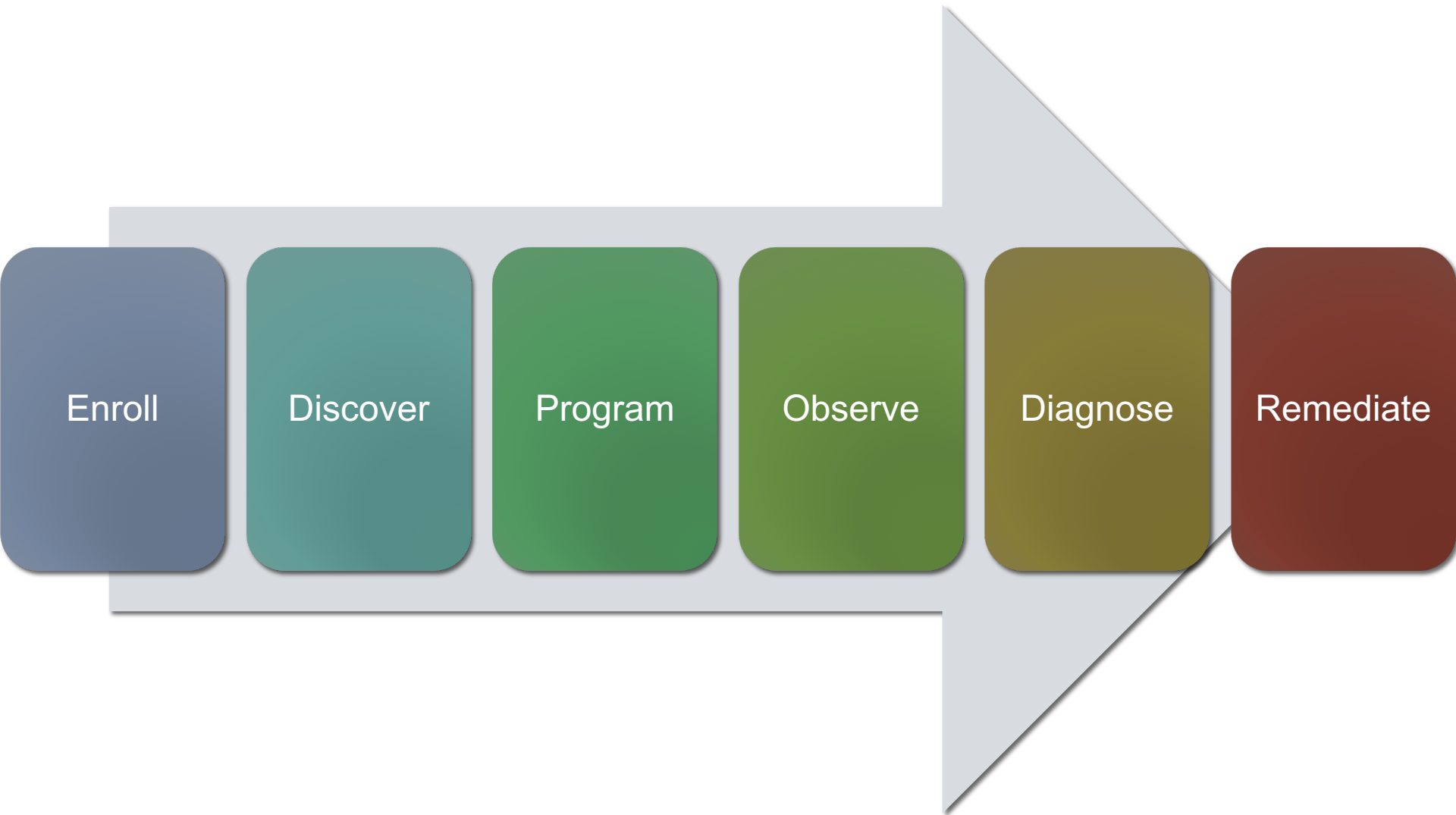


data center

# Managing networks at scale

- Change Wi-Fi WPA password → re-do initialization for dozens of devices
- Change LTE-M service provider → swap thousands of SIMs?
- Who should have access to the device?
  - co-resident family members
  - children of elderly parents
  - emergency service providers
  - landlords (e.g., for air quality monitors)
- Which device was just compromised and is attacking web services?

# IoT lifecycle vision



# SECURITY

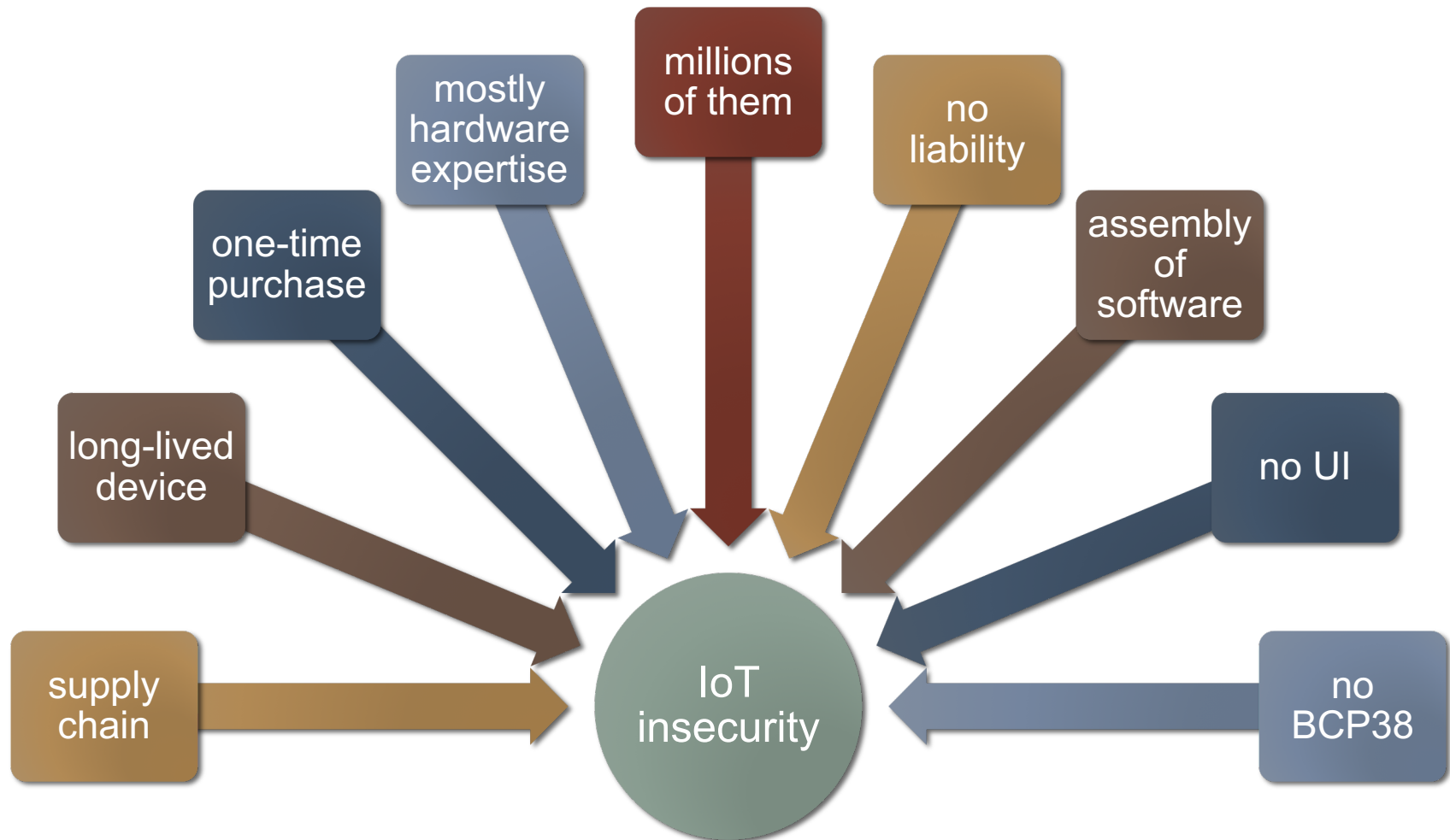
---

# NIST cybersecurity framework

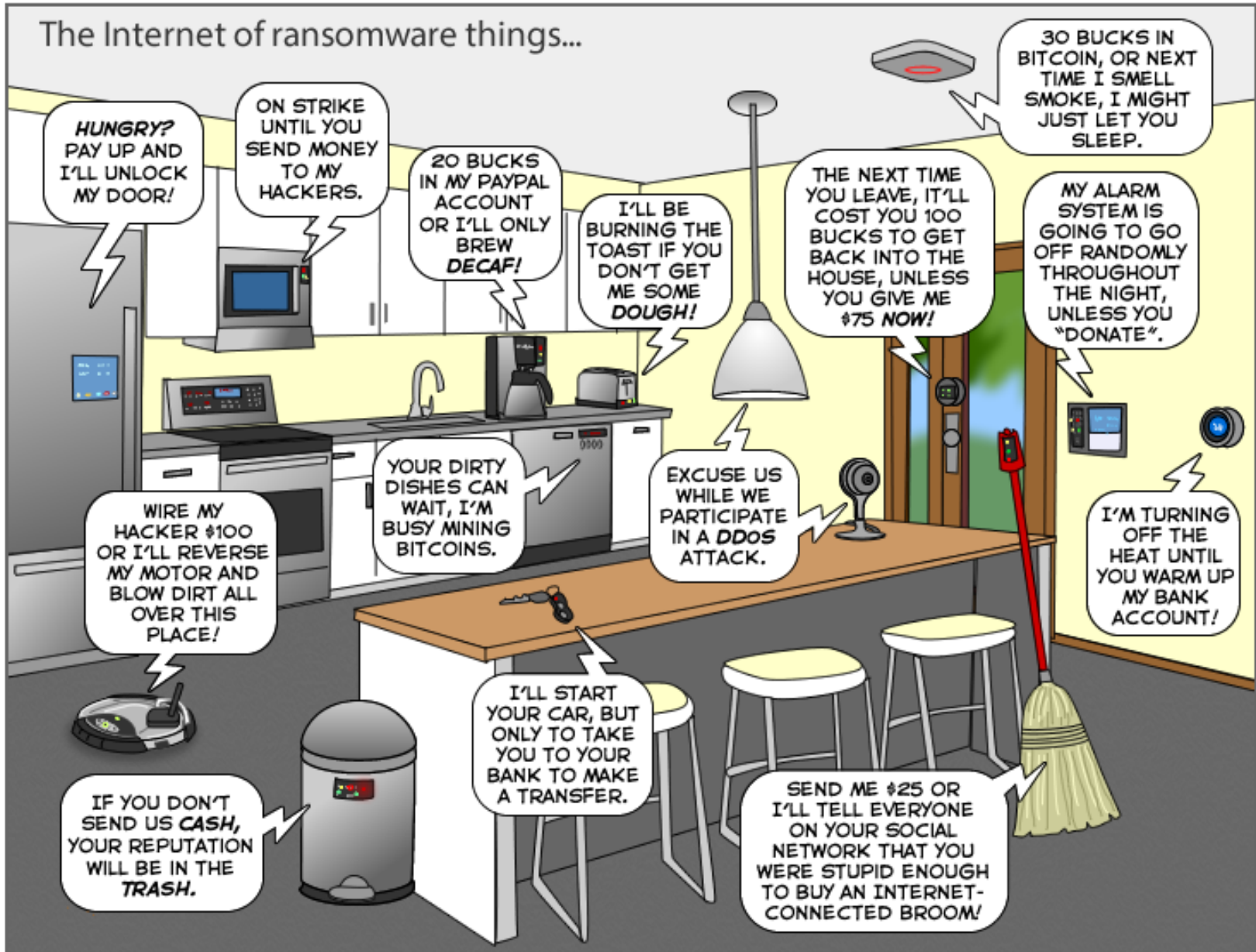
Function Unique Identifier	Function	Category Unique Identifier	Category
ID	Identify	ID.AM	Asset Management
		ID.BE	Business Environment
		ID.GV	Governance
		ID.RA	Risk Assessment
		ID.RM	Risk Management Strategy
PR	Protect	PR.AC	Access Control
		PR.AT	Awareness and Training
		PR.DS	Data Security
		PR.IP	Information Protection Processes and Procedures
		PR.MA	Maintenance
		PR.PT	Protective Technology
DE	Detect	DE.AE	Anomalies and Events
		DE.CM	Security Continuous Monitoring
		DE.DP	Detection Processes
RS	Respond	RS.RP	Response Planning
		RS.CO	Communications
		RS.AN	Analysis
		RS.MI	Mitigation
		RS.IM	Improvements
RC	Recover	RC.RP	Recovery Planning
		RC.IM	Improvements
		RC.CO	Communications



# IoT security confluence

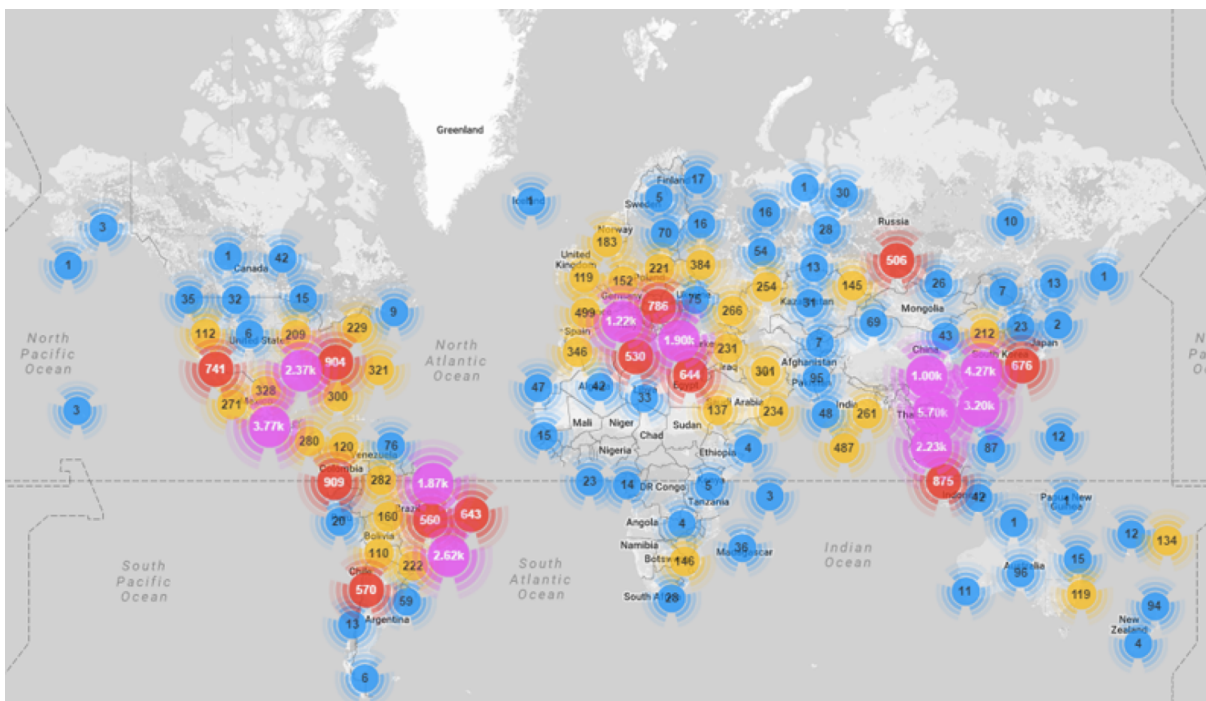


# The Internet of ransomware things...



# DDOS via IoT

- Krebs DDOS, 9/2016: **620 Gb/s, total of > 1.5 Tb/s**
- GRE, SYN, HTTP GET, POST
- MiraiNet: “380k bots from telnet alone”
- Enabled by UPnP → bypass NATs

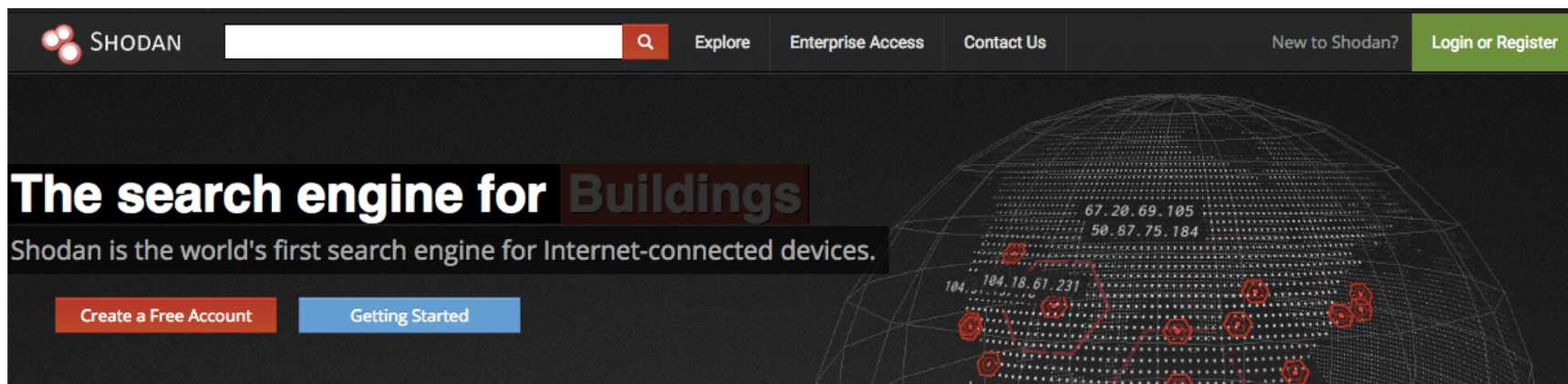


```
xc3511 vizxv  
admin 888888  
xmhdipc  
default  
123456 54321  
support
```



# Mirai botnet

- Chinese manufacturer, used by lots of OEMs
- BusyBox Linux
- Brute-force ssh and telnet
- Web reset doesn't change ssh or telnet



SHODAN  [Explore](#) [Enterprise Access](#) [Contact Us](#) [New to Shodan?](#) [Login or Register](#)

## The search engine for Buildings

Shodan is the world's first search engine for Internet-connected devices.

[Create a Free Account](#) [Getting Started](#)



### Explore the Internet of Things

Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them.



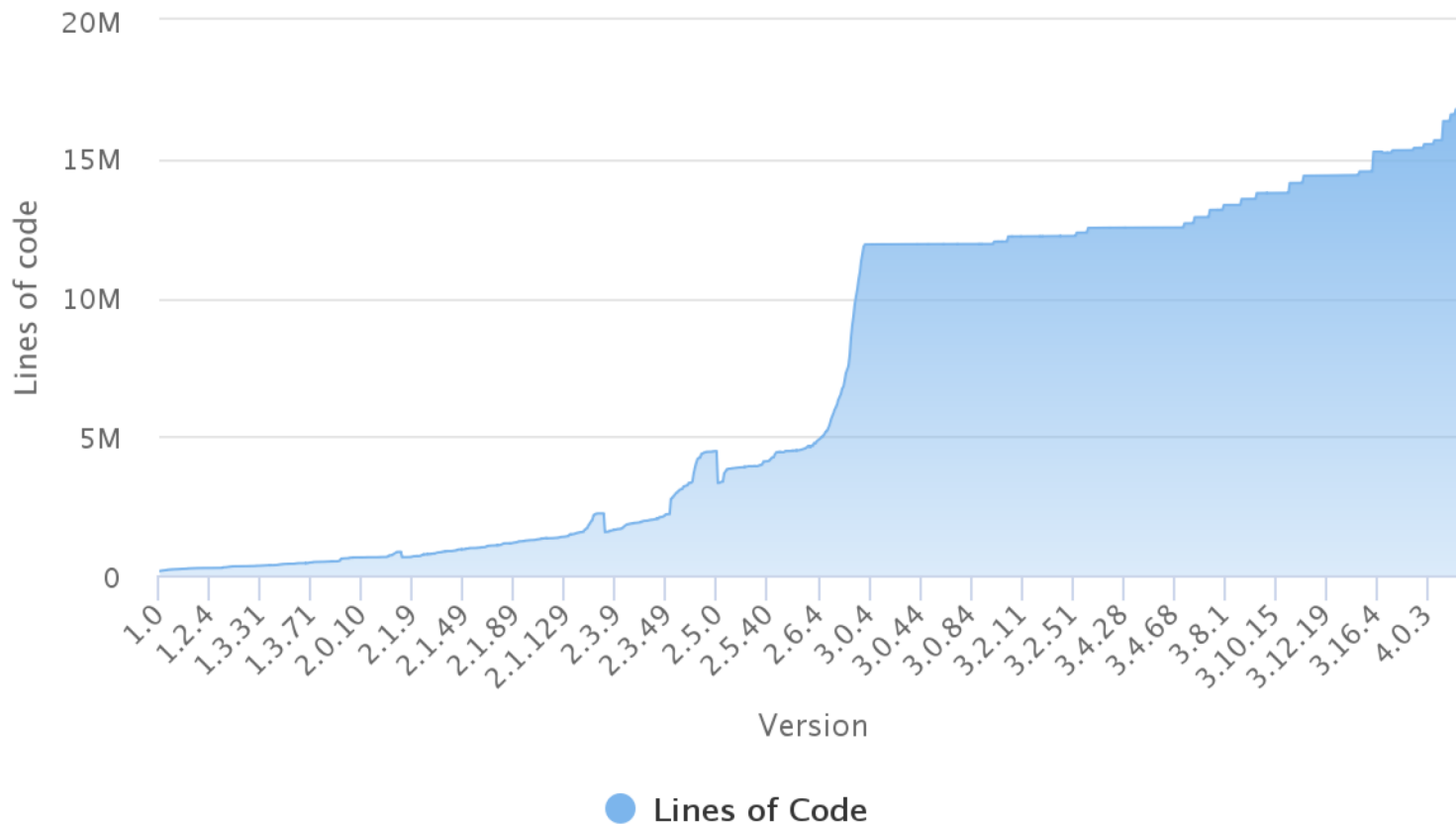
### See the Big Picture

Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan!

# Linux kernel lines of code

## Lines of code per Kernel version

Click and drag in the plot area to zoom in



BusyBox:  
177,650 SLOC

# You cannot hide

Hackers worldwide currently probe IoT devices for vulnerabilities after they have been connected to the internet for six minutes. Each hour these devices are tested for vulnerabilities - at least 800 times per hour - with an average of 400 login attempts occurring daily. On average, hackers try to access one IoT device every five minutes and a total of 66 per cent of their attempts end up being successful.

<http://www.itproportal.com/news/the-average-iot-device-is-compromised-after-being-online-for-6-minutes/>

# IoT DDOS economics

- DDOS as externality
  - device owners don't care:
    - barely slows down their Internet service
    - device still functions normally
    - don't know victims, generally
  - vendors don't care (enough)
    - not liable for damage (right now) – public nuisance?
    - only marginally affects their business reputation
  - ISP don't care (much)
    - individually, not much load – in lightly-loaded direction (outbound)
    - hard to combat
    - haven't adopted BCP38 (egress address filtering)



Schneier  
Oct. 2016  
Cohan  
Apr. 2013

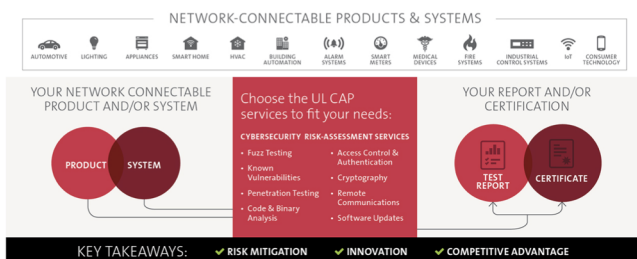
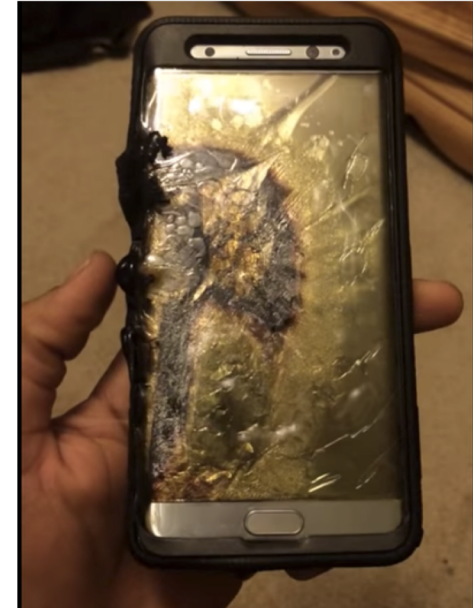
# IoT lemons

- “*The Market for Lemons: Quality Uncertainty and the Market Mechanism*” (Akerlof, 1970)
- Information asymmetry
  - purchaser cannot judge invisible qualities
  - pays only average price
  - → above-average-quality goods not marketed
- “defect four or more times and the problem is still occurring, the car may be deemed to be a lemon” → get purchase price back
  - more than four patches?



# Fixes for externalities and lemons

- Liability
  - slow, one-by-one, uncertain standards of care
  - what is “negligent”?
- Certification
- Insurance liability
  - homeowner’s insurance
- Regulation
  - adherence to minimum performance standards

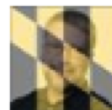


## 1894 The Birth of UL

Founder William Henry Merrill opens Underwriters' Electrical Bureau, the Electrical Bureau of the National Board of Fire Underwriters. The Bureau's first test is conducted on March 24, 1894, on non combustible insulation material for "Mr. Shields."

# This is not **that** hard!

- No factory-default passwords
  - long-term, no human-settable passwords at all → client certs
- No telnet, ssh, SNMP (typically)
- Only configure from local subset
- Automated, signed updates
- Web interfaces use non-root accounts
- Automated testing for XSS and SQL injection



**David Troy**

18 hrs · Baltimore, MD · 🌐

Many of the jobs of the future will not be about making things or creating value: they will involve keeping our increasingly complex and brittle infrastructure from collapsing on itself. For example, "cybersecurity" is a compounding tax on the deferred externalities of lazy design.

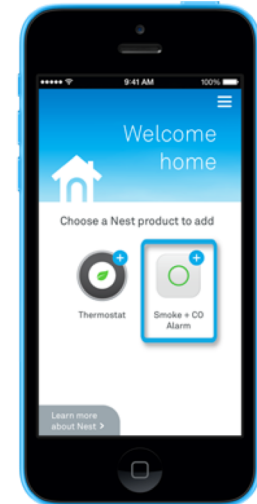
# ENROLL

---

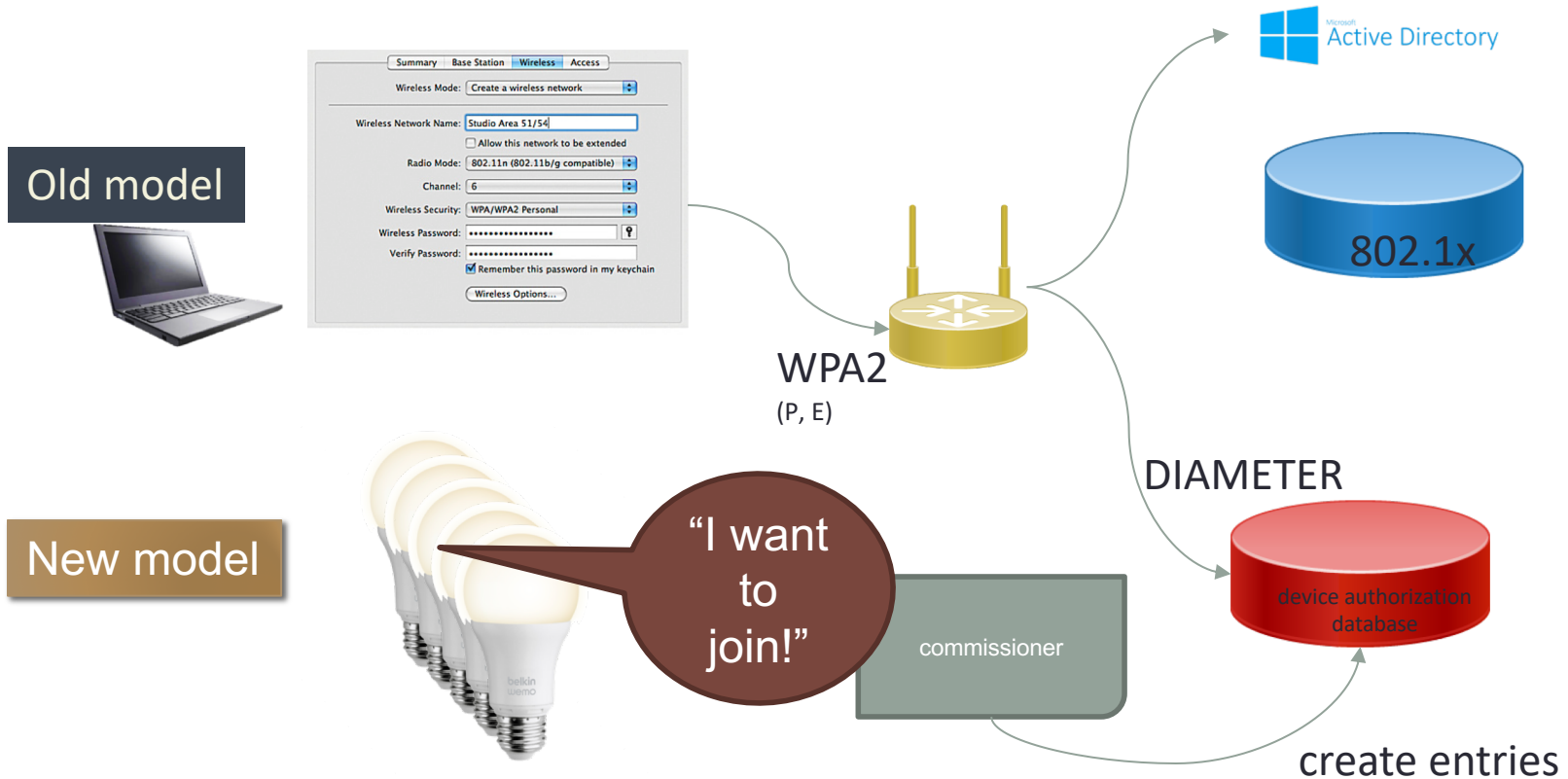


# Challenge: enrollment

- Commercial buildings → enroll 1,000s of devices at once
- Home → enroll one device at a time
  - current model: one app per device (class)
  - re-do if Wi-Fi password changes
  - common options:
    - QR code
    - P2P Wi-Fi (Wi-Fi Direct)
  - possibilities
    - “hi, I’m a Philips light bulb – add me!” (PKI)



# How should we secure things?



# Our Device Enrollment Protocol



WiFi Direct + ZeroConf + TLS



Join P2P network



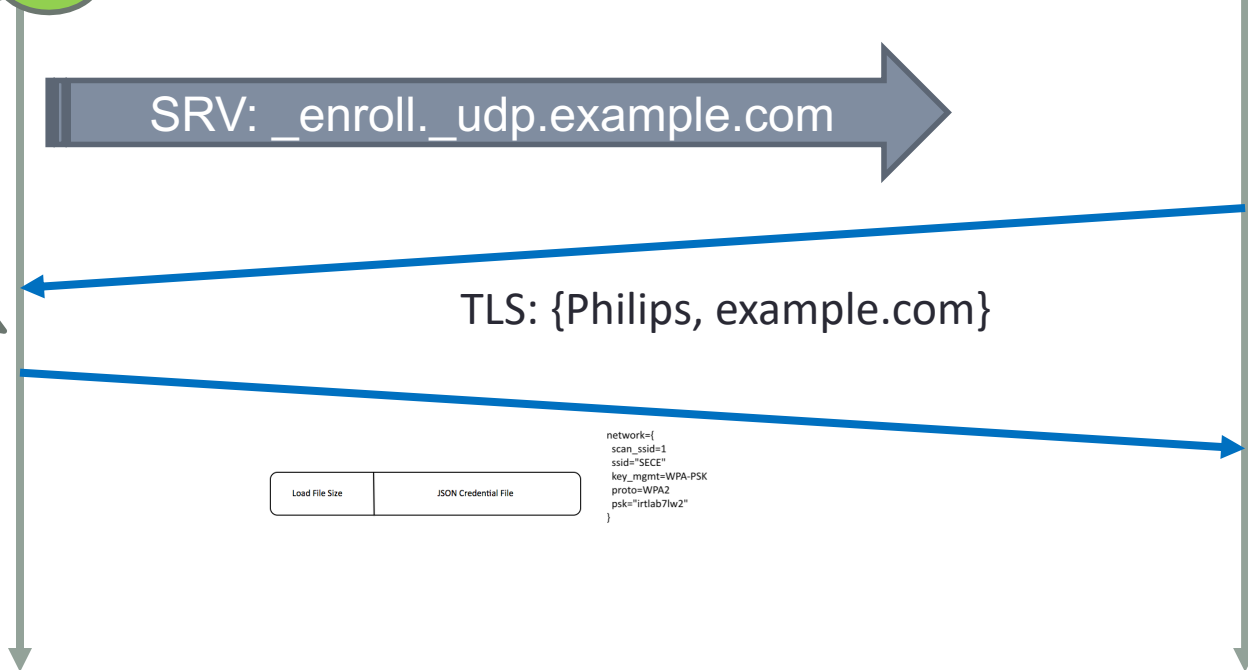
Admit Philips He?

TLS: {Philips, example.com}

Load File Size	JSON Credential File
----------------	----------------------

```
network={
  scan_ssid=1
  ssid="SECE"
  key_mgmt=WPA-PSK
  proto=WPA2
  psk="irrtlab7lw2"
}
```

Join WPA network



# AllJoyn is doing something similar

## 1. Onboarder broadcasts its SSID

When an Onboarder device is first plugged in, it will advertise its SSID over Wi-Fi. The SSID is either prefixed with "Aj\_" or postfixed with "\_Aj" to help indicate that this device supports the AllJoyn™ Onboarding service.

## 2. Onboarder connects to Onboarder

The Onboarder will scan for unconfigured AllJoyn devices by looking for SSID names with "Aj\_" or "\_Aj". A user can then choose to onboard a specific Onboarder device. The first step is to connect to the Onboarder device's SSID. Depending on the Onboarder platform, this may be done automatically by the application.

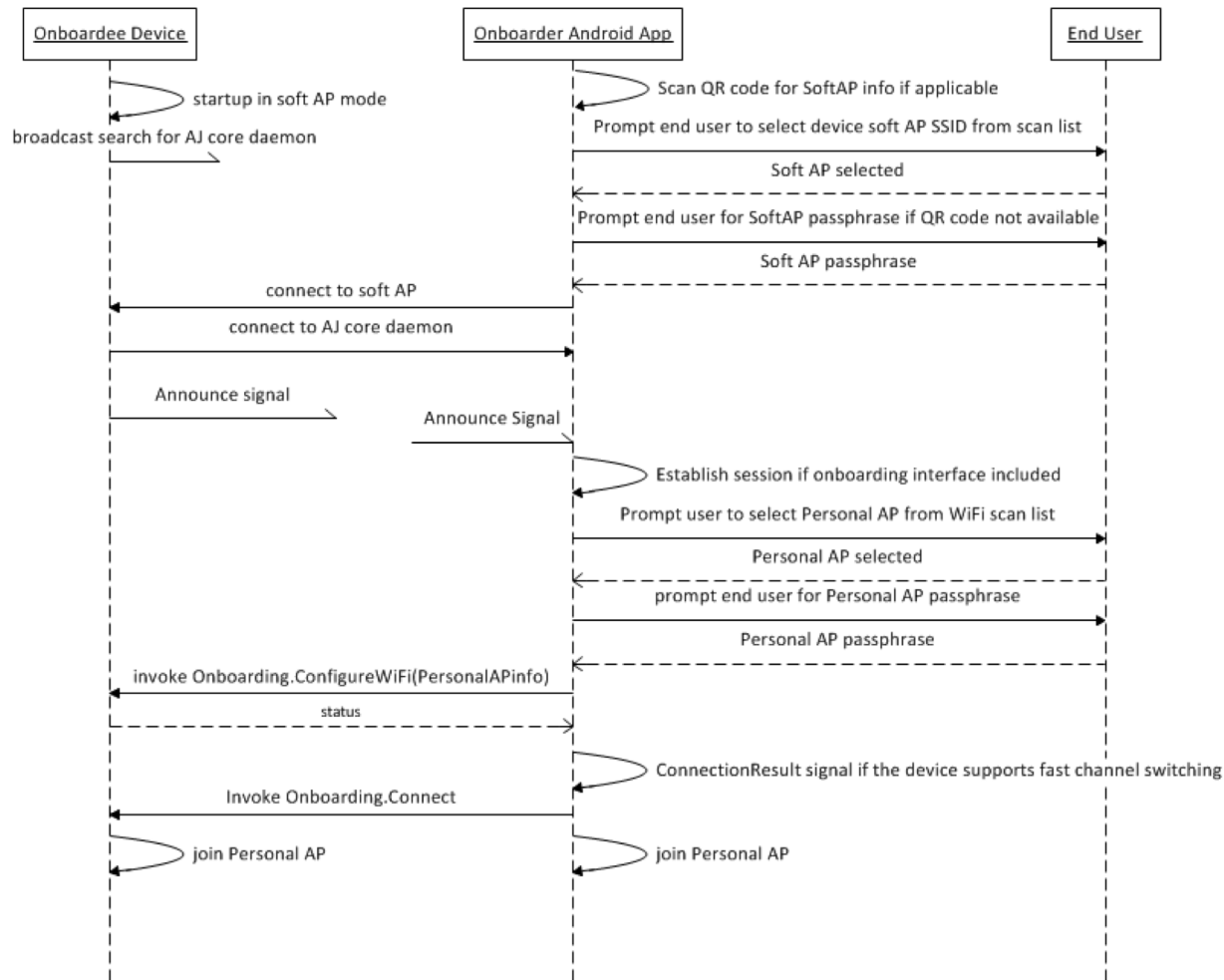
## 3. Onboarder sends Wi-Fi credentials

After connecting to the Onboarder's SSID, the Onboarder will listen for [AllJoyn About announcements](#). Then, the Onboarder will use the Onboarding service interfaces to send the target Wi-Fi network credentials to the Onboarder device.

## 4. Switch to target Wi-Fi network

Both devices will then switch to the target Wi-Fi network.

# AllJoyn



# DISCOVER & PROGRAM

---



“Remember when, on the Internet, nobody knew who you were?”



# Don't depend on one cloud



**Brian**

@Hamster\_Brian

Follow

Joys of the @internetofshit - AWS goes down. So does my TV remote, my light controller, even my front gate. Yay for 2017.

RETWEETS

552

LIKES

583



11:58 AM - 28 Feb 2017



**Ashley Mayer**

@ashleymayer

Follow

Um is Alexa striking for A Day Without a Woman?

#alexadown means I have to figure out how to turn the lights off the old fashioned way.

7:14 AM - 8 Mar 2017

Amazon Glacier (N. Virginia)	
Amazon Glacier (Ohio)	
Amazon Glacier (Oregon)	
Amazon Inspector (N. Virginia)	
Amazon Inspector (Oregon)	
Amazon Kinesis (Montreal)	
Amazon Kinesis (N. California)	
Amazon Kinesis (N. Virginia)	
Amazon Kinesis (Ohio)	
Amazon Kinesis (Oregon)	
Amazon Kinesis Analytics (N. Virginia)	
Amazon Kinesis Analytics (Oregon)	
Amazon Kinesis Firehose (N. Virginia)	
Amazon Kinesis Firehose (Oregon)	
Amazon Lightsail (N. Virginia)	
Amazon Machine Learning (N. Virginia)	
Amazon Mobile Analytics (N. Virginia)	

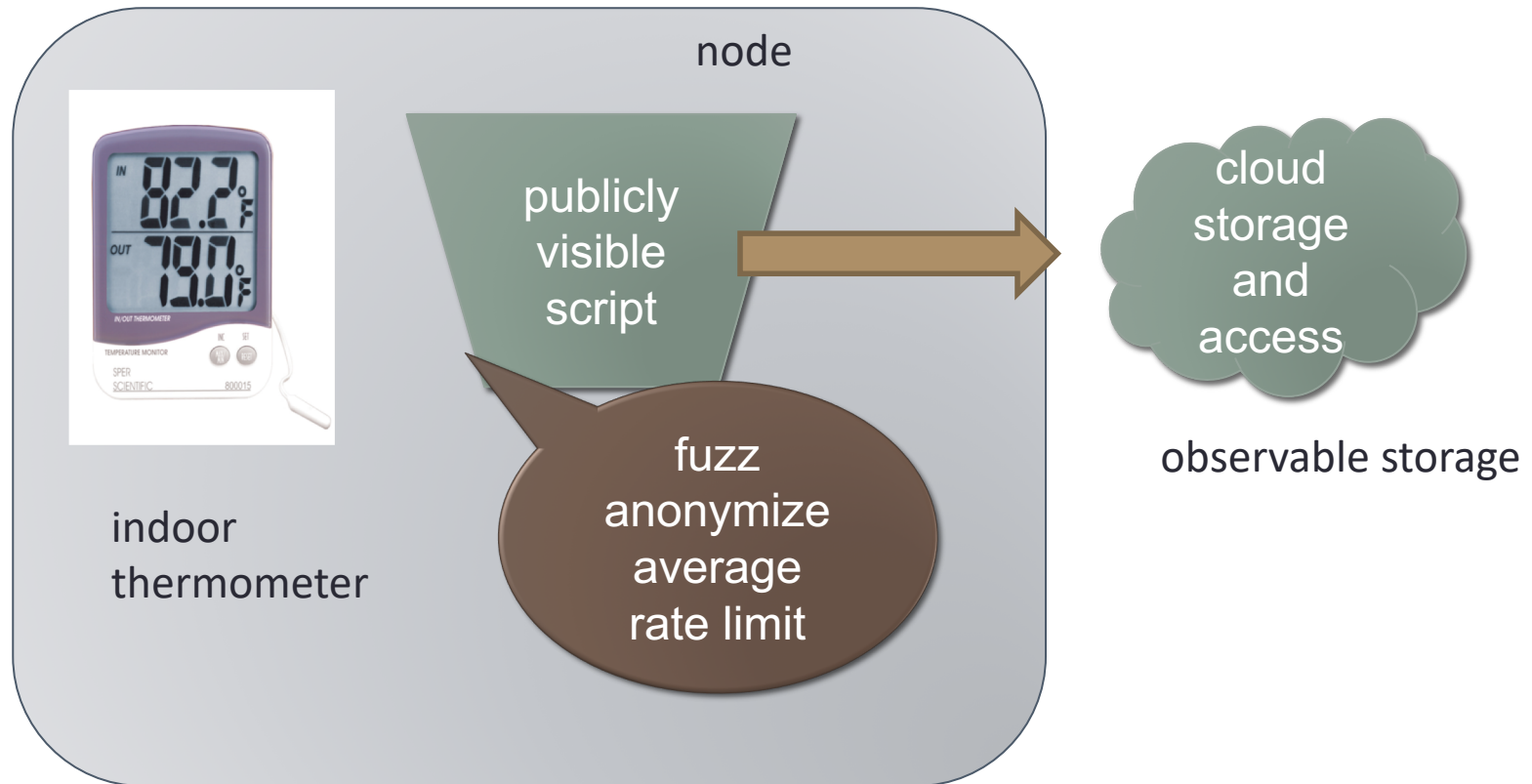
## Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region

We'd like to give you some additional information about the service disruption that occurred in the Northern Virginia (US-EAST-1) Region on the morning of February 28th. The Amazon Simple Storage Service (S3) team was debugging an issue causing the S3 billing system to progress more slowly than expected. At 9:37AM PST, an authorized S3 team member using an established playbook executed a command which was intended to remove a small number of servers for one of the S3 subsystems that is used by the S3 billing process. Unfortunately, one of the inputs to the command was entered incorrectly and a larger set of servers was removed than intended. The servers that were inadvertently removed supported two other S3

# Discover & program

- Model: local computation (“fog”, “edge computing”) + cloud
  - → if owned by user, ensures (some) privacy
  - → basic functionality independent of network connectivity
  - home router or building infrastructure
- Local gateway needs to discover new devices
  - template-based programming
  - “for thermostat in room X, set valve in room X”
  - “for PIR sensor in room Y, turn on light in room Y”

# Local processing for ~~efficiency~~ privacy



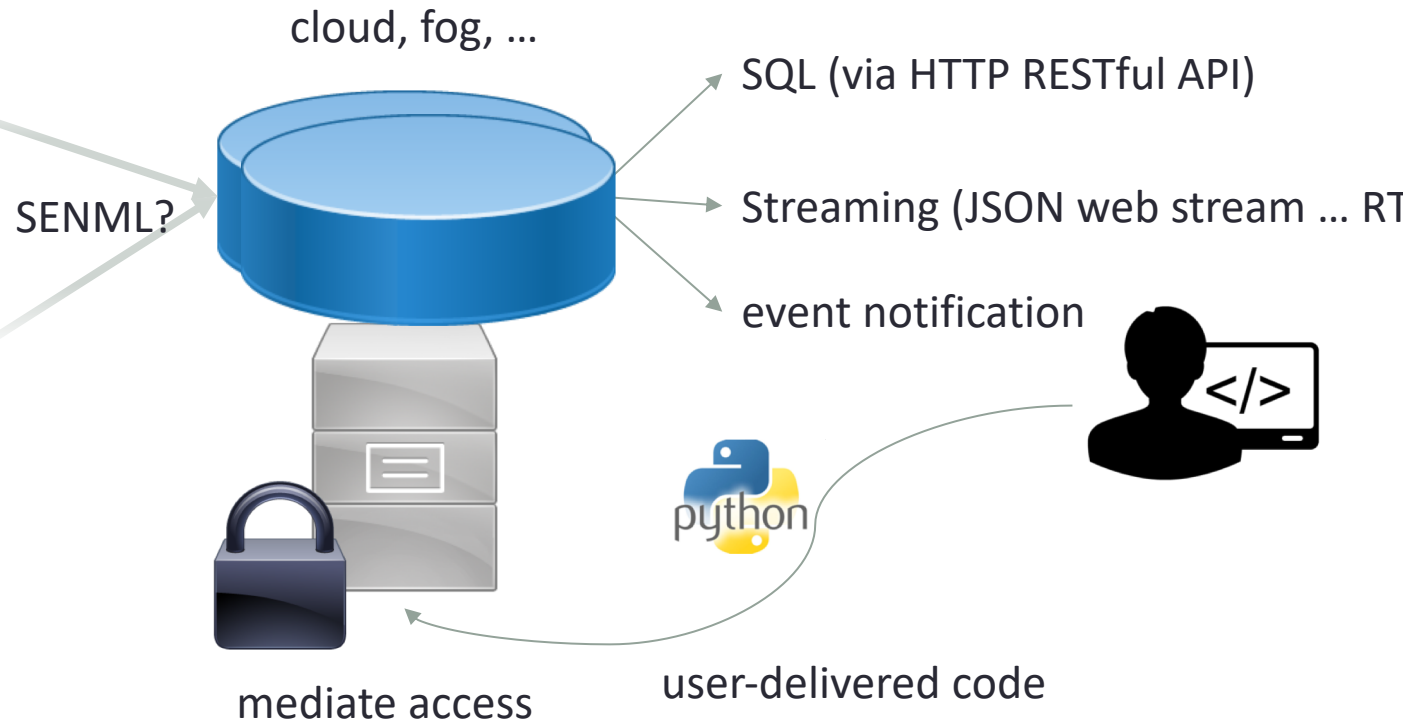
fog computing model

# Protocols matter, but programmability matters more

- Nobody wants to program raw protocols
- Most significant network application creation advances:
  - 1983: socket API → abstract data stream or datagram
  - 1998: Java network API → mostly names, HTTP, threads
  - 1998: PHP → network input as script variables
  - 2005: Ruby on Rails → simplify common patterns
- Many fine protocols and frameworks failed the programmer hate test
  - e.g., JAIN for VoIP, SOAP for RPC
- Most IoT programmers will not be computer scientists

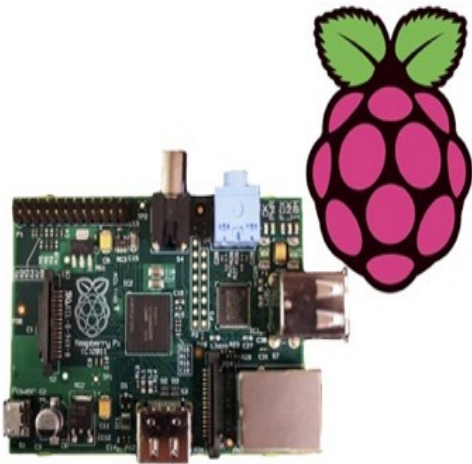
# What is the best generic (simple) architecture?

MQ135 Air Pollution sensor

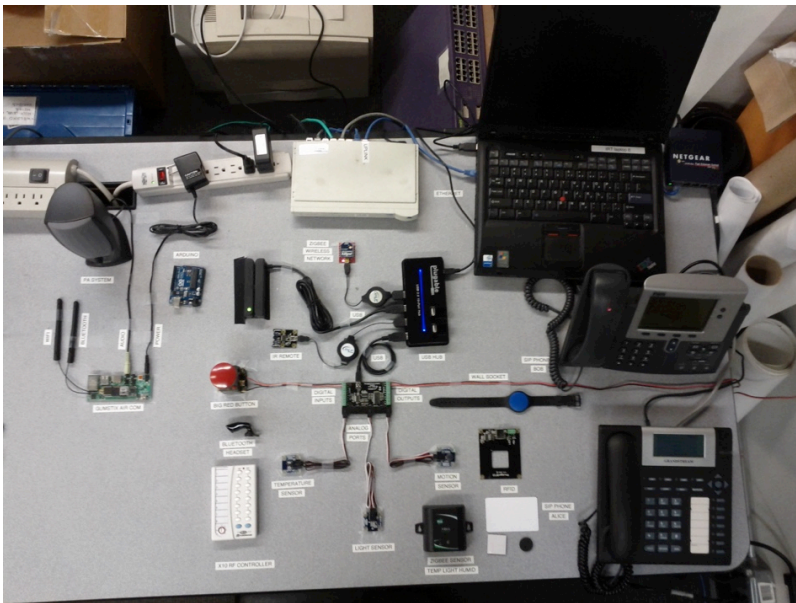


# Challenge: integrate embedded, mobile & virtual

magnetometer  
accelerometer  
location  
gyroscope



# Multi-network IoT, with differential visibility



public sensors &  
actuators



semi-private



private



# Some of IoT is streaming



## Protocols

TCP/IP, DHCP, SMTP, DNS, RTSP,  
RTCP, RTP, HTTP, TCP, UDP, STUN,  
TURN, XMPP, uPNP, SNTP, IPv4,  
ICMP, Bonjour, SUNAPI



**Honeywell**

update rate of 10 to 250 Hz

# SECE (Sense Everything, Control Everything)

monitor real & simulate devices

IRT Lab

Fullscreen
  Go back
  Reload panel

The dashboard displays several monitoring components:

- Sensors:** Swipe Activity (blue bar), Keypad Activity (green bar), Sensor Activity (yellow bar), Door Unlocked (dark red triangle), Door Open (orange circle), Profile (text label).
- Gauge:** A circular gauge with a needle pointing to 0, labeled 'Gauge'.
- Lamp Status:** Lamp C1 (dark red square), Lamp B1 (dark red square), Lamp A4 (dark red square).
- Video Feeds:**
  - IRT Lab Feed 2: A small video window showing a fire alarm sign.
  - IRT Lab Feed 1: A large video window showing a computer workstation in a control room.
  - CEPSR 7: A video window showing a close-up of a hand holding a device.
- Controls:** A slider control and a play button are visible at the bottom.

# We could do better

- Somewhat unsatisfactory
  - AllJoyn model only for LAN operations
  - CoAP & HTTP better for get/set operations
  - MQTT simpler for publish/subscribe
  - SIP (or RTSP) better for media streaming
- Lots of proprietary network protocols
  - BAC for building automation
- Same device or source, multiple identifiers
  - HTTP URL or SIP URL or MQTT IP address/domain name
  - none are particularly useful or semantically meaningful
    - e.g., likely change if device is replaced

# The age of application-specific {sensors, spectrum, OS, protocol ...} is over

- *Computing system*: dedicated function → OS
  - → abstract into generic components
  - e.g., USB human interface device (HID)
  - e.g., HTTP + JSON for web interfaces
- What are the equivalent sensor and actuator classes?
  - see SenML
- *Networks*: generic app protocols
  - request/response → HTTP
  - event notification → SMTP, SIP, XMPP?



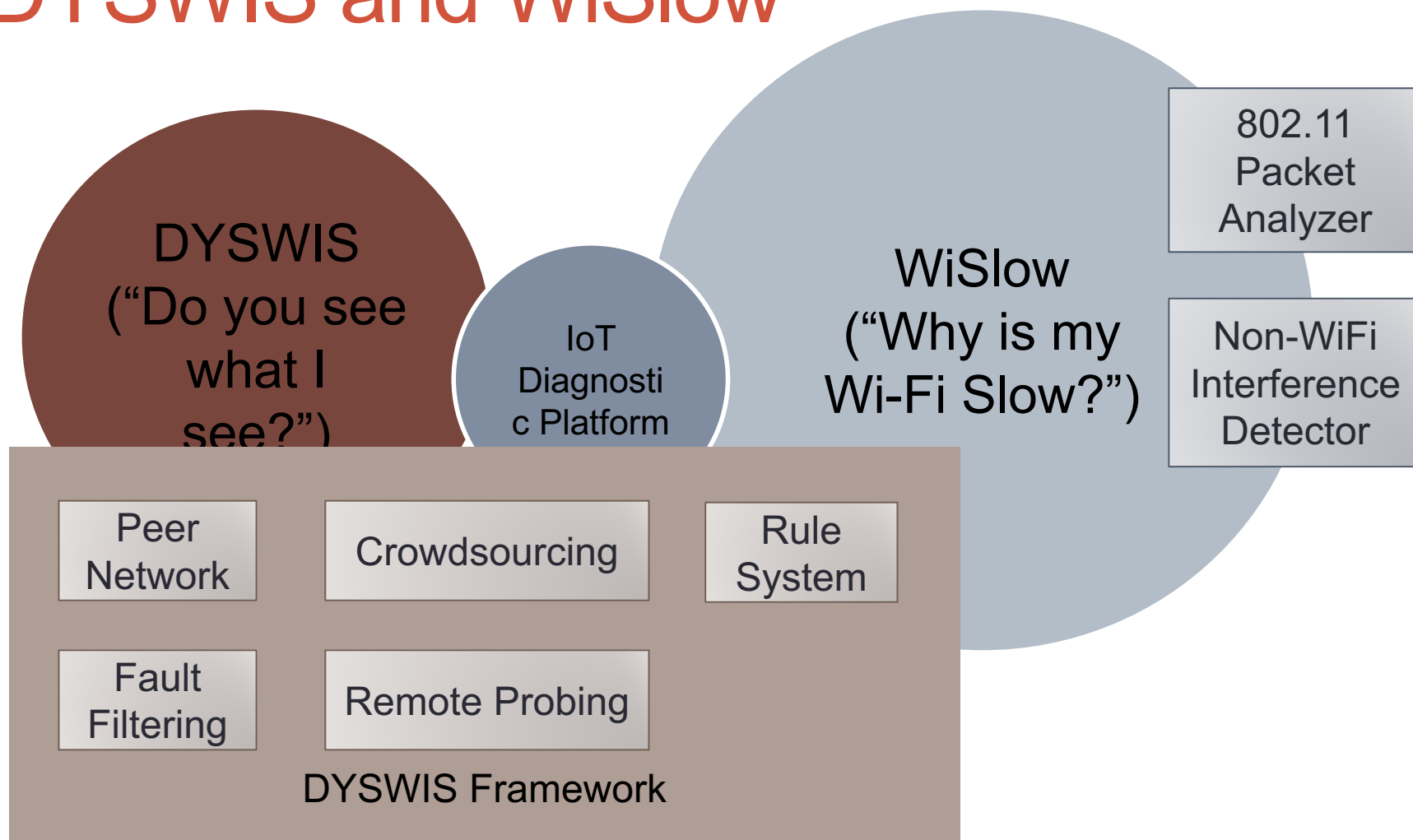
# Open issues

- Naming
  - network-independent → logical addressing
  - what stays constant when {location, manufacturer, network} change?
  - do we need dynamic group addresses?
- Access control
  - who can do what?
  - same device may have public (read), semi-private (set within range) and private (set across range) access
  - who protects the devices?
  - how do we reason about who can do what?
- Simulation to deployment
  - how can we test very large systems without building them?
  - how can we test impact of faults and failures?
  - mixed programming models: graphical, Python or Lua, IFFFT-style

# DIAGNOSE & REMEDIATE

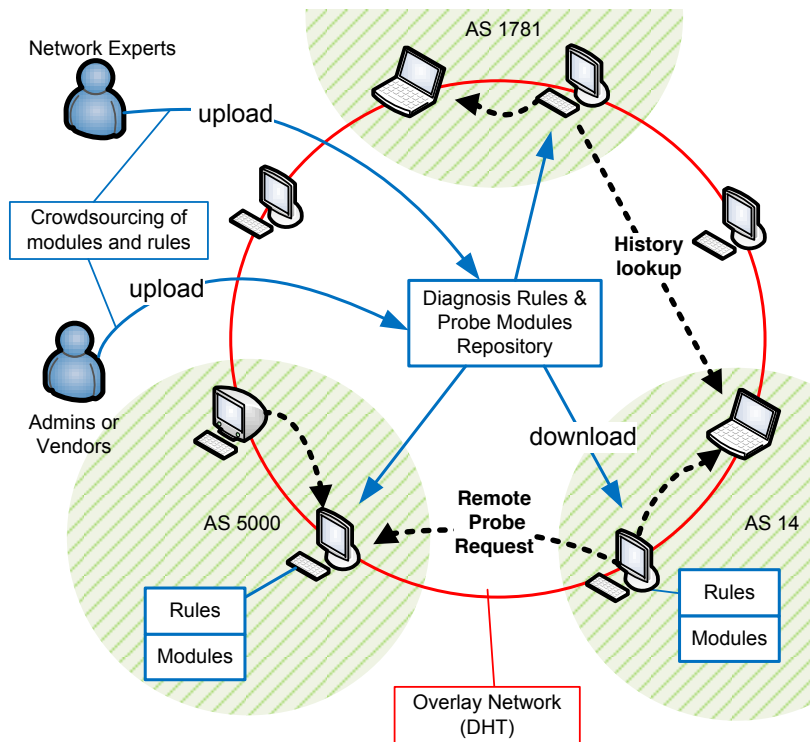
---

# DYSWIS and WiSlow





# DYSWIS Framework



- Collaboration of peers
  - Peer-to-peer network
  - Request probes
  - More clues when probe results from various environments are obtained
  - Overcome limitations of single-user investigation
  - Identify via Facebook friends
- Automatic detection
- Filtering mechanism
  - Distinguishes meaningful failures
- Crowdsourced rules
  - Small and independent
  - Cooperate to build probe modules and diagnostic rules.
- Distributed probes
  - Parallel and systemic approach

Problem detected → Ask others:

TCP  
connection  
?

DHCP?

DNS?

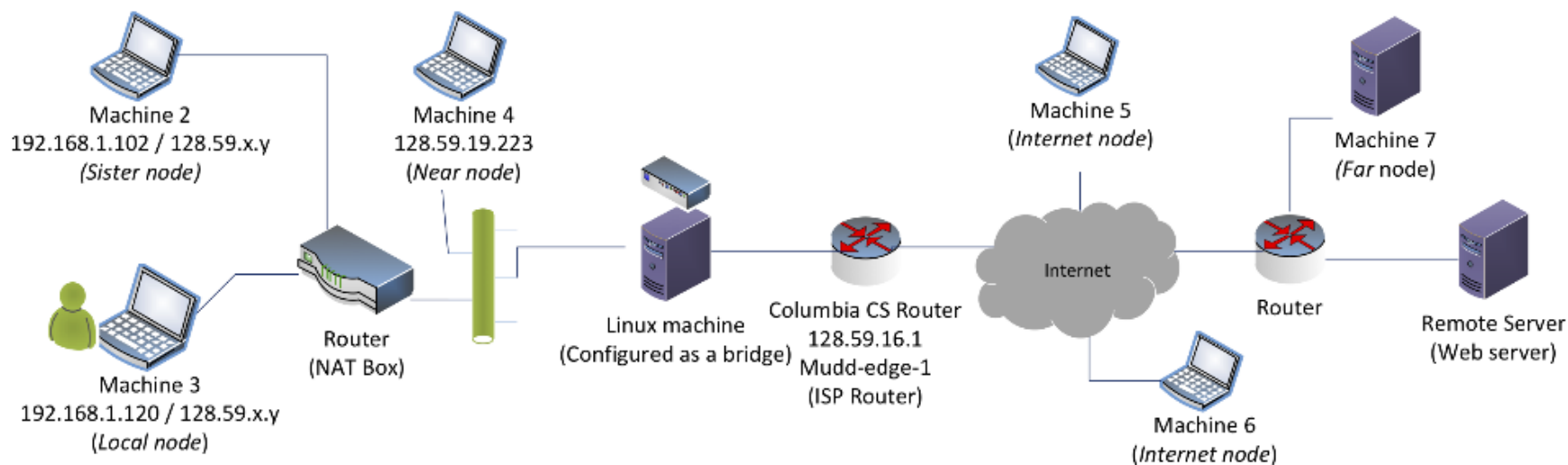
Traceroute

Ping

HTTP?

Port  
blocking

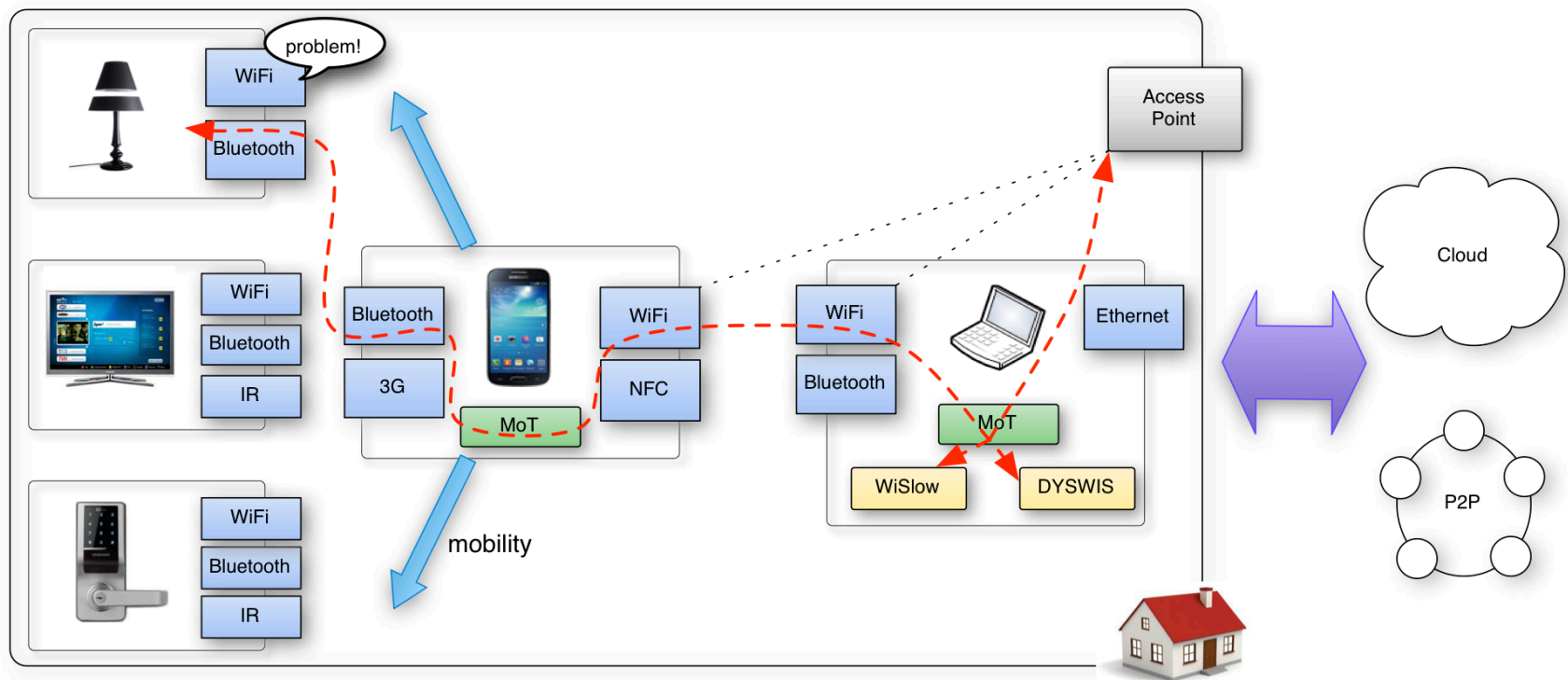
# Distributed Probing



Problem ID	Description
C1	Misconfiguration on the user's computer
C2	A problem on the link to a router
C3	Misbehavior of the local router
C4	ISP outage
C5	Link between the ISP and the Internet
C6	Remote service provider network outage
C7	Remote server down
C8	The service provider blocks your ISP
C9	The server blocks your ISP
C10	The service provider blocks your IP address
C11	The server blocks your IP address

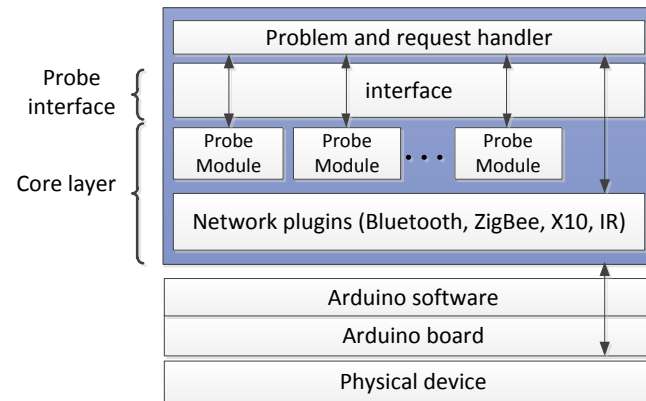
Rule ID	Requesting probing to:	Probing module	If response is:	Likely cause	Unlikely cause
R1.1	Sister node	TCP connection	Yes	C1	C2-C11
R1.2	Sister node	TCP connection	No	C3-C11	C1, C2
R1.3	Sister node	TCP connection	No response	C2	-
R1.4	Near node	TCP connection	Yes	C10, C11	C1-C9
R1.5	Near node	TCP connection	No	C5, C7-C9	C1-C3
R1.6	Near node	TCP connection	No response	C2-C4	-
R1.7	Internet node	TCP connection	Yes	C8-C11	C1-C7
R1.8	Internet node	TCP connection	No	C6, C7	C1-C5
R1.9	Internet node	TCP connection	No response	C1-C5	-
R1.10	Far node	TCP connection	Yes	C11	C1-C8, C10
R1.11	Far node	TCP connection	No	C7	C1-C6, C8-C11

# DYSWIS for IoT

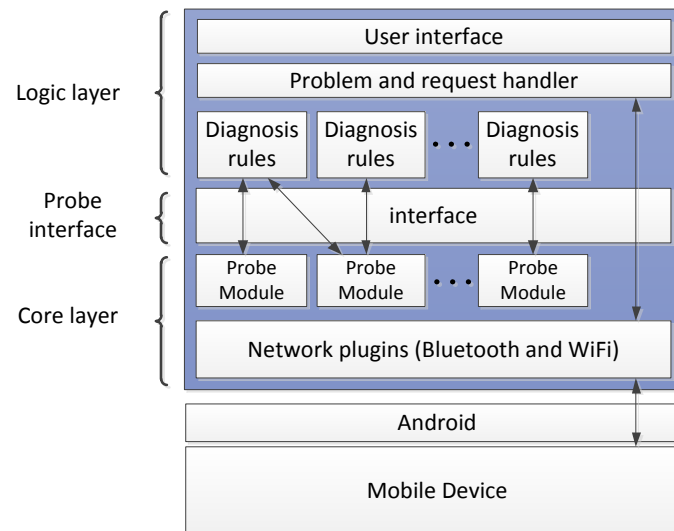


\* MoT: A Collaborative Network Troubleshooting Platform for the Internet of Things, Kyung-Hwa Kim, Hyunwoo Nam, Jin-Hyung Park, and Henning Schulzrinne, IEEE WCNC, April 2014

# DYSWIS for IoT – node architecture



(a) MoT Client for Arduino (microcontroller)



(b) MoT Client for Android

# WiSlow: Why is my Wi-Fi slow?

Nearby  
Networks



Channel  
Contention



WiFi Collisions



Non-WiFi Interference at 2.4GHz

Why  
Slow?

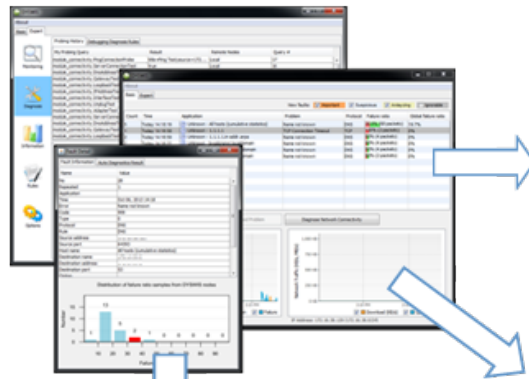


WiSlow?  
Because  
there is ...

Application

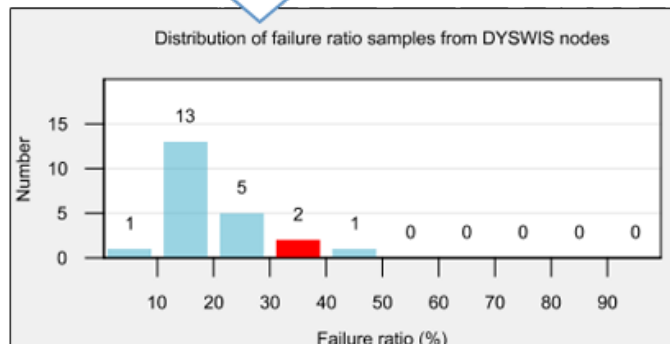
No specialized hardware!

# DYSWIS UI



View faults:  Important  Suspicious  Analyzing  Ignorable

Problem	Protocol	Failure ratio	Global failure ratio
Name not known	DNS	30.8% (68 packets)	19.7%
TCP Connection Timeout	TCP	100% (2 packets)	0%
Name not known	DNS	50% (4 packets)	0%
Name not known	DNS	50% (4 packets)	0%
Name not known	DNS	50% (4 packets)	0%
Name not known	DNS	50% (2 packets)	0%



**Probing**

- ✓ You have a proper IP address: 172.16.38.129
- Loopback Test
  - ✓ No problem.
- Gateway
  - ✓ Successfully connected to your gateway router: 172.16.38.2
- DNS Address
  - ✓ You have a proper DNS server address
- Server Connection Test (Well-known ports)
  - ✓ Successfully connected to www.google.com, port: 80 (Web)
  - ✓ Successfully connected to www.yahoo.com, port: 80 (Web)
  - ✓ Successfully connected to clic.cs.columbia.edu, port: 22 (SSH)
  - ✓ Successfully connected to imap.gmail.com, port: 993 (IMAP)
  - ✓ Successfully connected to smtp.gmail.com, port: 465 (SMTP)

**Diagnosis Result**

CONNECTIVITY  
 \* Disconnected network adapters have been detected. Please ensure that the Ethernet card is properly installed and you are using a laptop, then check whether the wireless card is enabled

Injected Problem	Distance from the AP	Accuracy	False Positive
No interference	-	100.0 %	14.1 %
Channel contention	-	92.2 %	1.5 %
Non-Wi-Fi interference (baby monitor, cordless phone, and microwave oven)	0.0 m	100.0 %	3.9 %
	0.5 m	97.8 %	
	1.0 m	82.2 %	
	1.5 m	82.2 %	
	2.0 m	73.3 %	
	2.5 m	68.9 %	

Non-Wi-Fi Interference	Distance from the AP	Avg. Throughput	Diagnostic Accuracy	False Positive
Microwave oven	0.0 m	7.54 Mb/s	100 %	0.4 %
	0.5 m	8.52 Mb/s	100 %	
	1.0 m	8.96 Mb/s	100 %	
	1.5 m	9.33 Mb/s	100 %	
	2.0 m	9.30 Mb/s	100 %	
	2.5 m	8.91 Mb/s	93.3 %	
Baby monitor	0.0 m	0.51 Mb/s	100 %	1.1 %
	0.5 m	3.16 Mb/s	100 %	
	1.0 m	4.79 Mb/s	100 %	
	1.5 m	4.49 Mb/s	100 %	
	2.0 m	4.81 Mb/s	100 %	
	2.5 m	5.17 Mb/s	100 %	
FHSS Cordless phone	0.0 m	6.76 Mb/s	100 %	24.8 %
	0.5 m	9.65 Mb/s	100 %	
	1.0 m	10.02 Mb/s	100 %	
	1.5 m	10.05 Mb/s	66.7 %	
	2.0 m	12.44 Mb/s	26.7 %	
	2.5 m	13.28 Mb/s	6.7 %	

- The accuracy of distinguishing problem sources:
  - Channel contention
  - Non-Wi-Fi interference
  - No interference
- The accuracy of detecting the type of non-Wi-Fi interference source
  - More than 90% when the problem source is close to the Wi-Fi devices



# Conclusion

- IoT is finding lots of boring niches
- But IoT security is exposing almost all the security deficiencies of the Internet eco system
  - “thoughts and prayers” approach
  - continuing to do the same thing for the next 5 years and hoping for better results is not a strategy
- Start thinking beyond stove pipes of applications and home automation
- → engineering large scale systems x 10