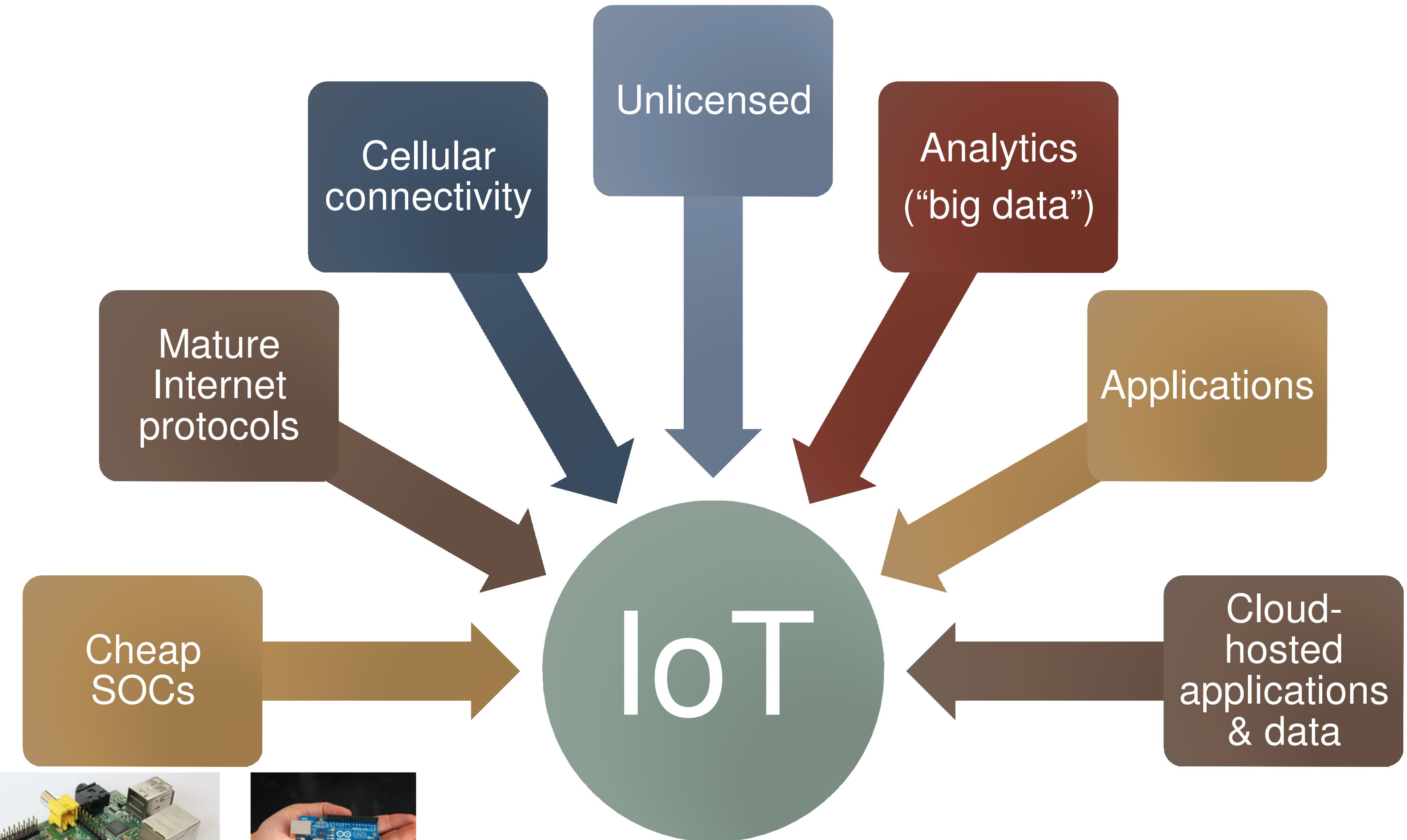
THE INTERNET OF THINGS AND OTHER CHALLENGES TO THE INTERNET AS WE KNOW IT

Henning Schulzrinne

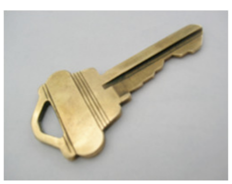
(+ Jan Janak & other CUCS IRT contributors)

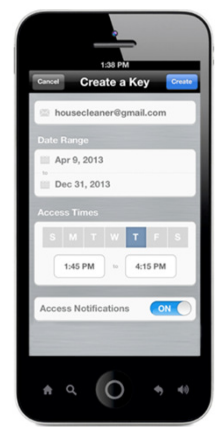
ICC 2016

Key enablers



Natural evolution





Internet of Things

- Mostly about devices, not the Internet
- Network part not really new or exciting
- Software-controlled networked devices
- Challenges:
  - lack of UI → usability
  - lack of UI → usable security
  - integration (service & APIs)
  - programming – beyond a single device

M2M/IoT/CPS is not…

- isn't just about fancy thermostats and $199 door bells
- doesn't always uses cellular networks
- is not always energy-constrained
- is not always cost-constrained
- doesn't always use puny microcontrollers
- is not always run by large organizations
  - many small & mid-sized providers
  - usually embedded into other products

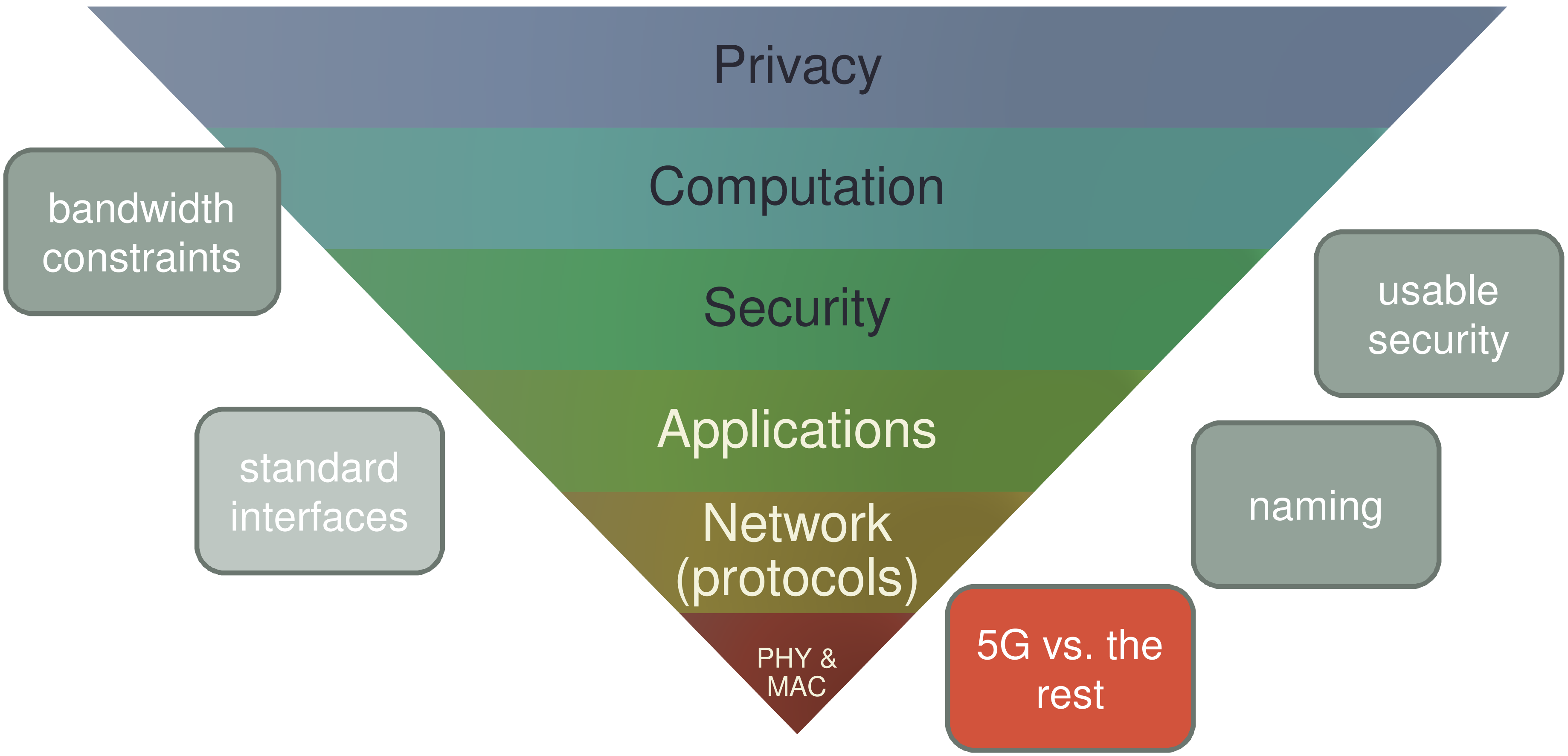
Where does IoT make sense?

- Automate manual data extraction
  - health, car, electric/gas meter, …
- Remote maintenance
  - vending machines, appliances, cars & trucks, trains, pumps, …
- Incorporate additional information
  - thermostats, light switches, traffic lights, parking meters, …
- Software-Defined Mechanics
  - locks, light switches
- But where does it solve more than 1st world problems?
  - commercial maintenance savings?
  - in-home customizable assistive technology

The killer app

with energy-harvesting

What's different?



Lessons from Internet experience

- The Internet is about more than the Internet protocol
- Reliability multiplies, costs add
- Quality is no substitute for quantity
- Data links layers come & go, IP stays
- The age of application-specific {sensors, spectrum, OS, protocol …} is over
- Protocols matter, but programmability matters more

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

Lessons from early IoT (and cousins)

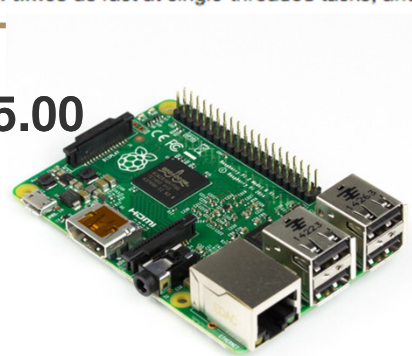
ATC	proprietary network architecture	"Ongoing problems continue to threaten NextGen's costs and timeline."
PTC	220 MHz dedicated network	"[NTSB] has advocated for some form of positive train control for more than 45 years."
ITS	5.9 GHz	allocated in 1999

Lesson: sensor networks may be (tiny) niche

- Most IoT systems will be near power since they'll interact with energy-based systems (li
- 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

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.

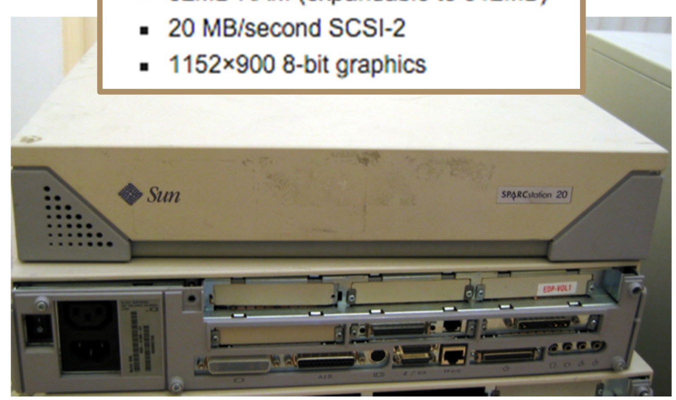
$35.00



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

http://eschatologist.net/blog/?p=266

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

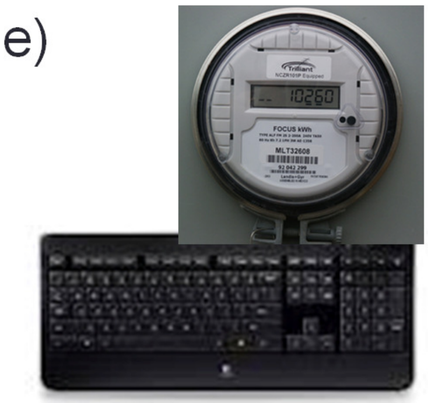


The age of application-specific {sensors, spectrum, OS, protocol …} is over

- *Computing system*: dedicated function → OS
  - → abstract into generic components
  - e.g., USB human interface device (HID)
- What are the equivalent sensor and actuator classes?
- *Networks*: generic app protocols
  - request/response → HTTP
  - event notification → SMTP, SIP, XMPP
- *Spectrum*: from new application = new spectrum to generic data transport

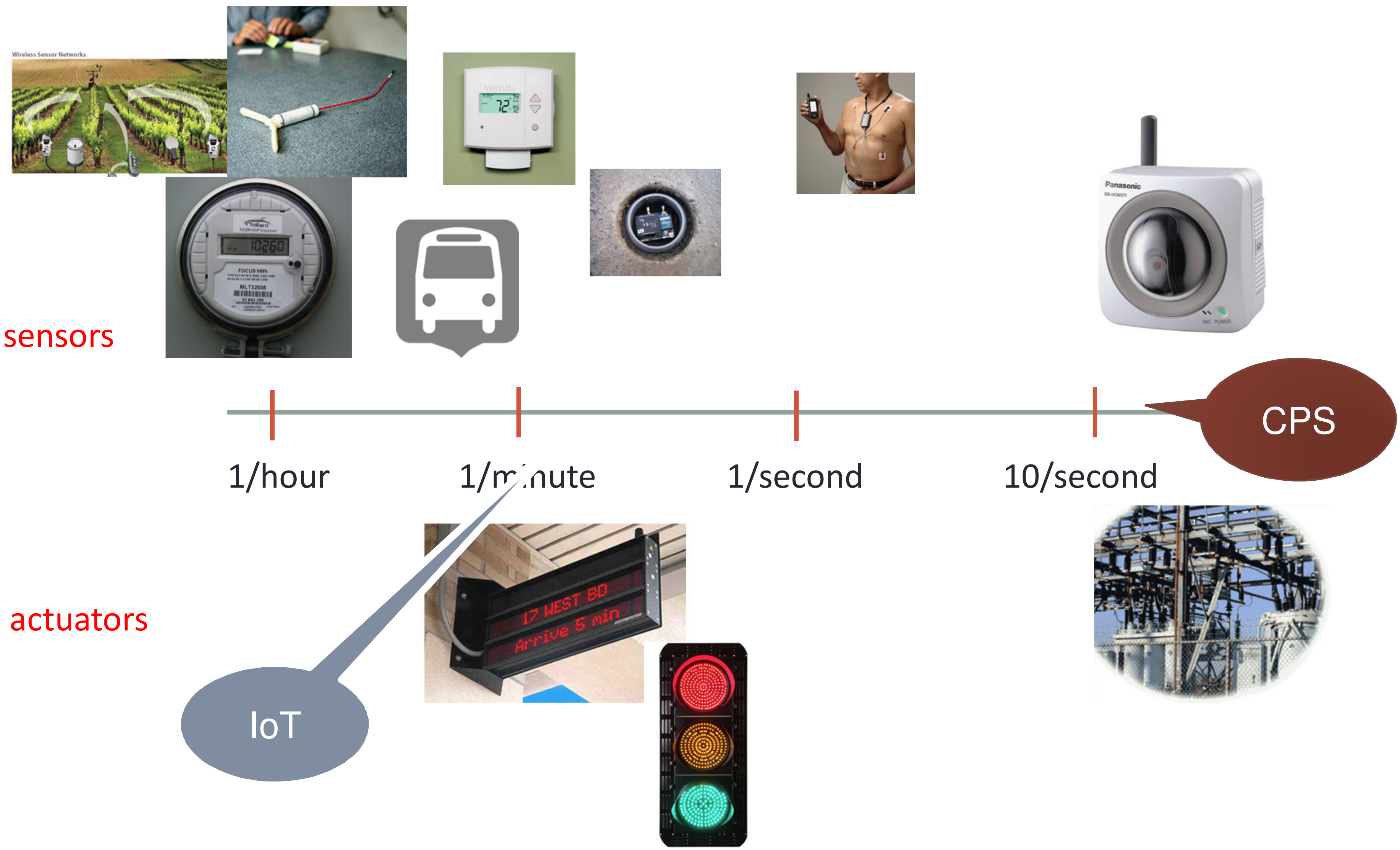
NETWORKS – PHY, MAC, LAYER 3

Network challenges

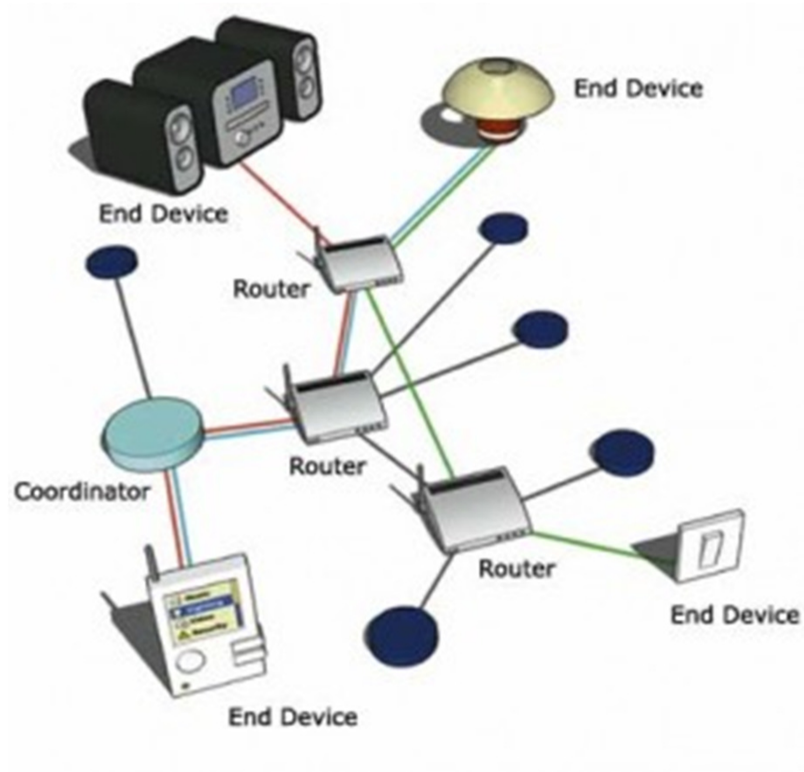


- Unlicensed
  - How do I attach and authenticate a device to a (home) network?
  - Credentials?
- Licensed
  - Reliability → multiple *simultaneous* providers
  - Mobility → different providers in different regions
  - Charging → often low, intermittent usage, sometimes deferrable (“Whispernet”)
    - From $50/device/month → < $1/month?
- Authentication
  - Which devices can be used by whom and how?
    - “Any employee can monitor the room temperature in any public space, but only Facilities staff can change it”

IoT varies in communication needs



Not just cellular *or* unlicensed



5G is not the only option

indoor
unmanaged

indoor
ext. managed

outdoor
urban

outdoor
rural

outdoor
remote

ZigBee

Bluetooth

Lte

LoRa

Lte

SIGFOX

Wi Fi

Niche networks

short range

ZigBee®

low energy; mesh

Bluetooth®

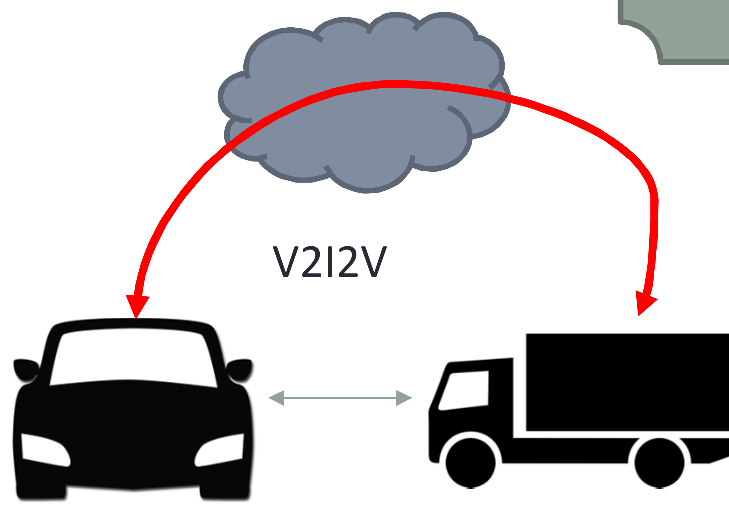
ubiquity; low cost

Wi Fi™

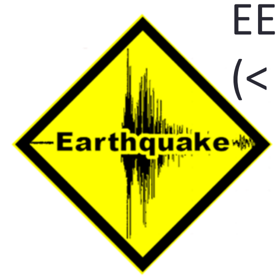
speed; public APs

5G = low latency + mmW + …

one-to-many!

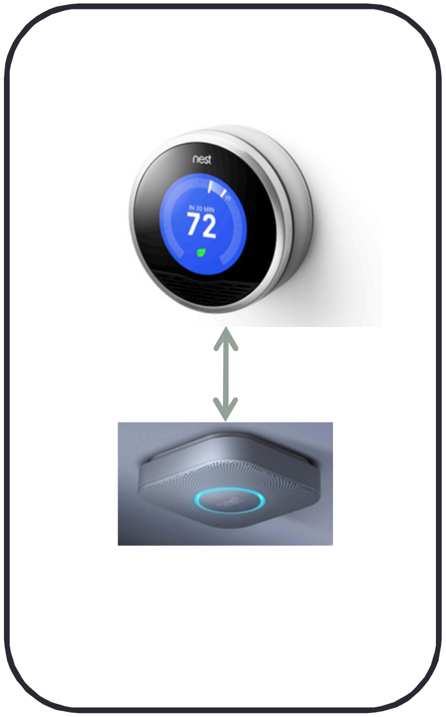


EEW
(< 5 s)



NETWORKS – APPLICATION PROTOCOLS

IoT islands vs. IoT eco system



Challenge: enabling discovery & access control

- Devices should be discoverable & re-usable
  - e.g., provide audio interface to bus display
  - environmental probes (temperature, noise, rain, …)
  - location (iBeacon) → 911
- Layers of functionality
  - anybody in vicinity can read
  - anyone in *family* can change
  - parents can re-program
- Allow delegation
  - grant temporary access to somebody or something else
  - by message or physical proximity
- Currently, all one-off solutions
  - OAuth? NFC?

Technology comes & goes, interfaces are forever

1904

1908

1947

1956

1993

INTERNET PROTOCOL

DARPA INTERNET PROGRAM
PROTOCOL SPECIFICATION

September 1981

1878

1982

1970s

fuel nozzle
1885?

1974

JPEG

1992

How should we name things?

network interface

EUI-64

IPv6

device
(independent of network)

domain name? → portability?
phone number?

device by function & location

"ceiling lamp in kitchen" (used in programs)

Communication identifiers

Property	URL owned	URL provider	E.164	Service-specific
Example	alice@smith.name sip:alice@smith.name	alice@gmail.com sip:alice@ilec.com	+1 202 555 1010	www.facebook.com/alice.example
Protocol-independent	no	no	yes	yes
Multimedia	yes	yes	maybe (VRS)	maybe
Portable	yes	no	somewhat	no
Groups	yes	yes	bridge number	not generally
Trademark issues	yes	unlikely	unlikely	possible
Privacy	Depends on name chosen (pseudonym)	Depends on naming scheme	mostly	Depends on provider "real name" policy

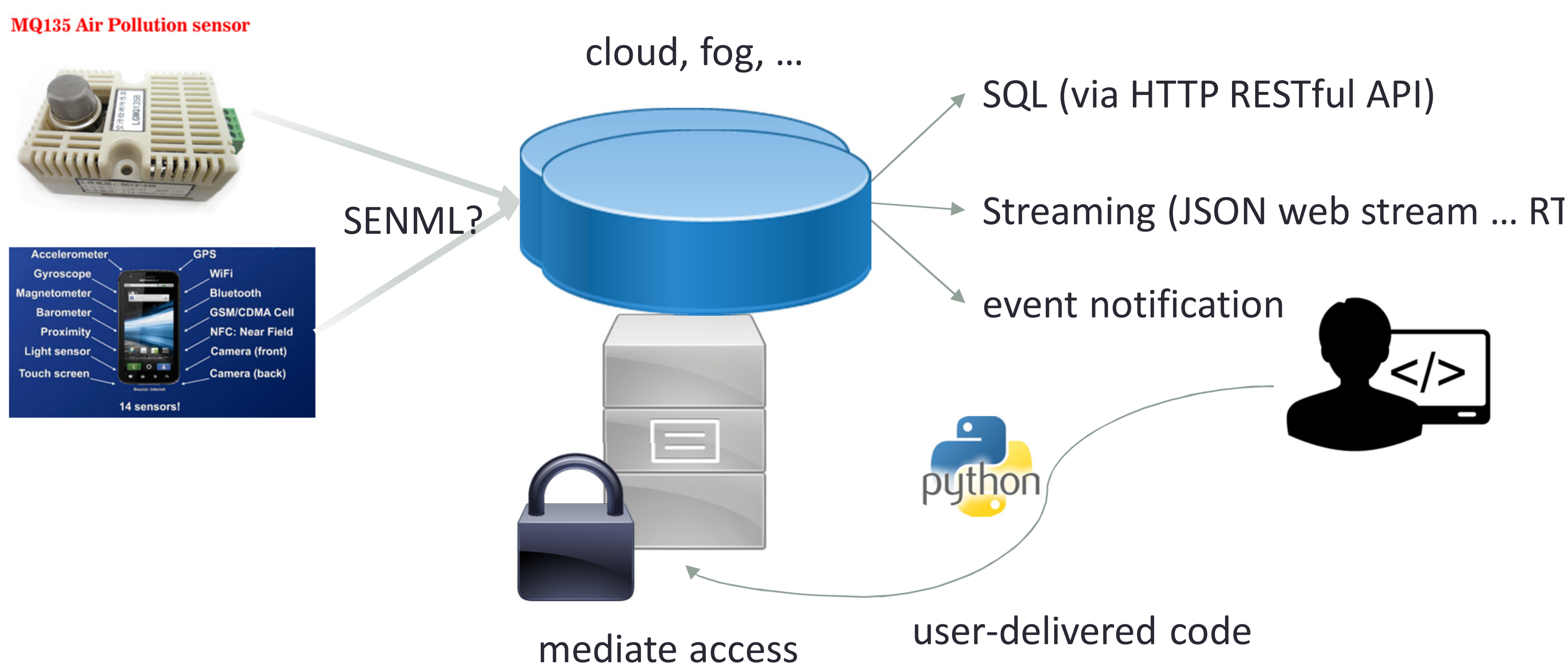
→ IoT will likely be assigned local IP address space and owner-based names (meter17.pseg.com) [if any]

COMPUTATION & SERVICES

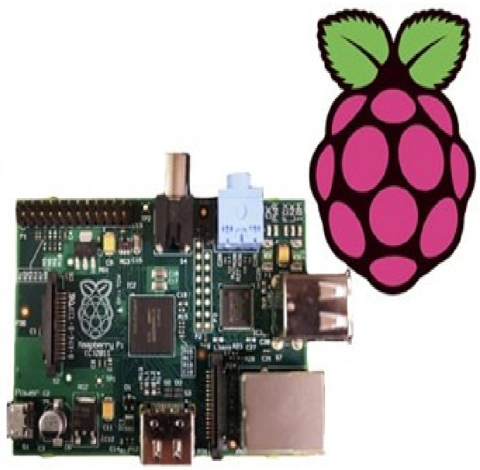
Protocols matter, but programmability matters more

- 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?



Challenge: integrate embedded, mobile & virtual



magnetometer
accelerometer
location
gyroscope

LIFECYCLE

Windows XP, Corolla & Revolv

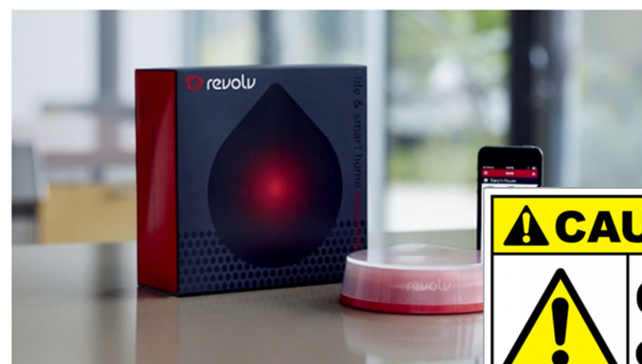
13 years

available 12/2001	end of sales 6/2008	end support 4/2009	end install 10/2010	end ext. support 4/2014

1996 Corolla
- still can get parts

$2,359

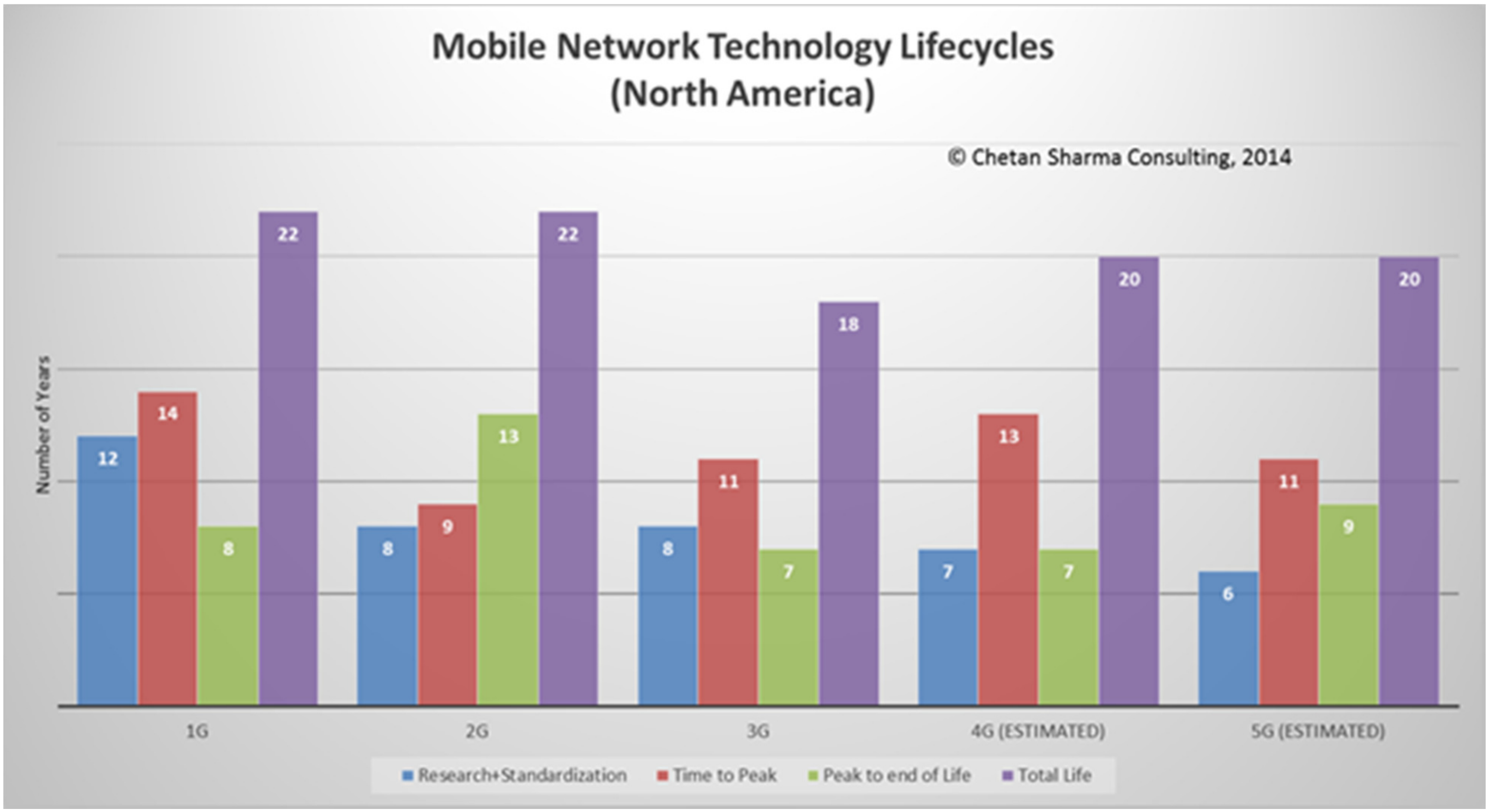
NEST'S HUB SHUTDOWN PROVES YOU'RE CRAZY TO BUY INTO THE INTERNET OF THINGS



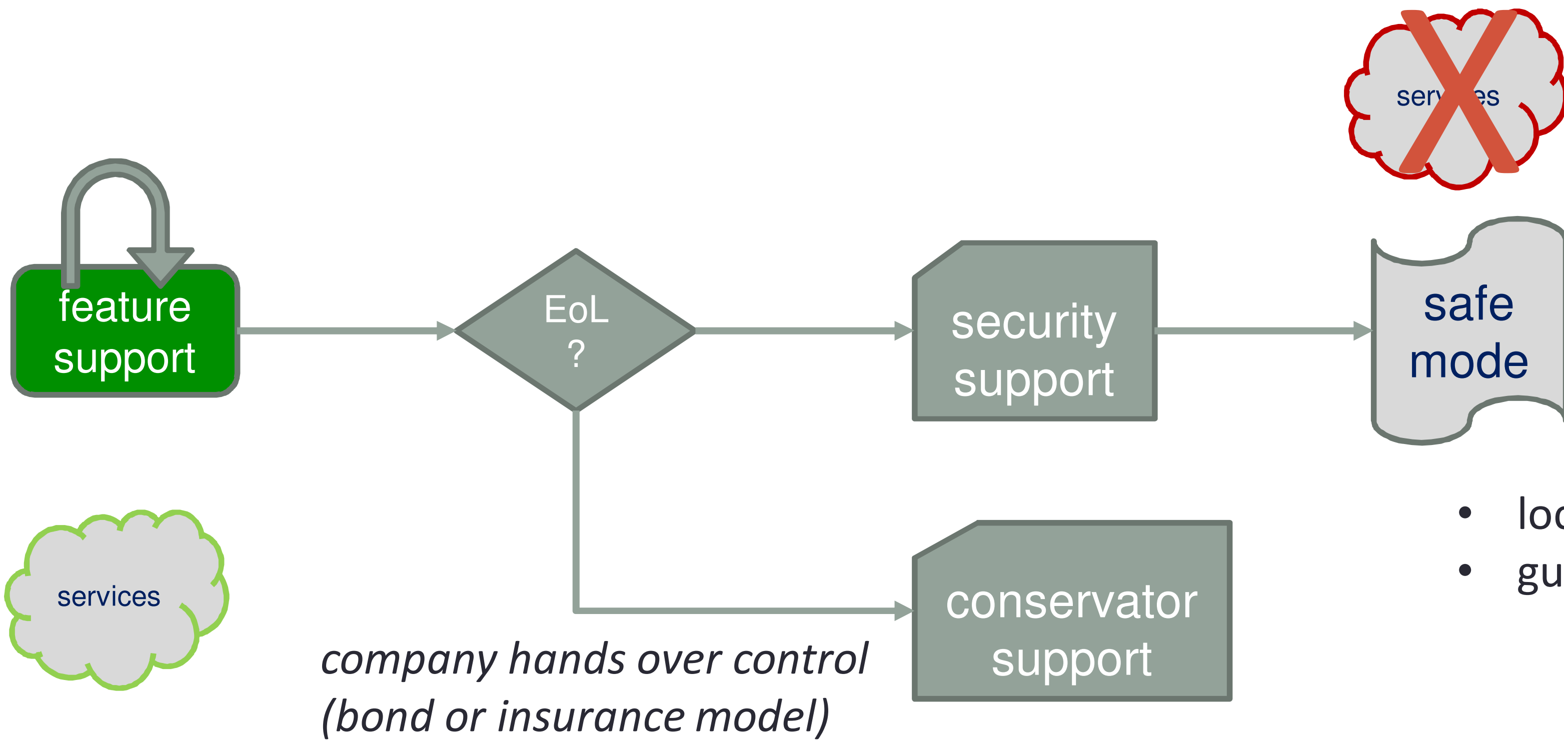
founded 2012
acquired by Nest 2014
shut down May 2016

IF YOU WERE one of the people who shelled out $300 for Revolv's smart home hub, you've probably already heard the bad news: the web service that powers the little gadget is shutting down next month, which will render the thing effectively useless.

Design for 20 years



IoT needs a life cycle model



- local only
- guardian

IoT needs an economic model

- Do you own or rent a device?
  - and do you know what rights you have (transfer, sale, …)?
  - and for how long?
- What is expected lifetime?
  - in what mode?
  - with what enhancements?
- Who pays for computation and storage?
  - printer & ink? stove & electricity?
  - subscription model → doesn’t scale except with aggregator
  - advertising model → creepiness-factor, no direct interaction
  - third party model: health or fire insurance, research (“your data for science”), electric utility

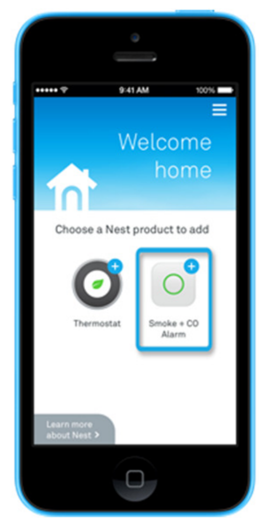
SECURITY

ShellShock for light switches

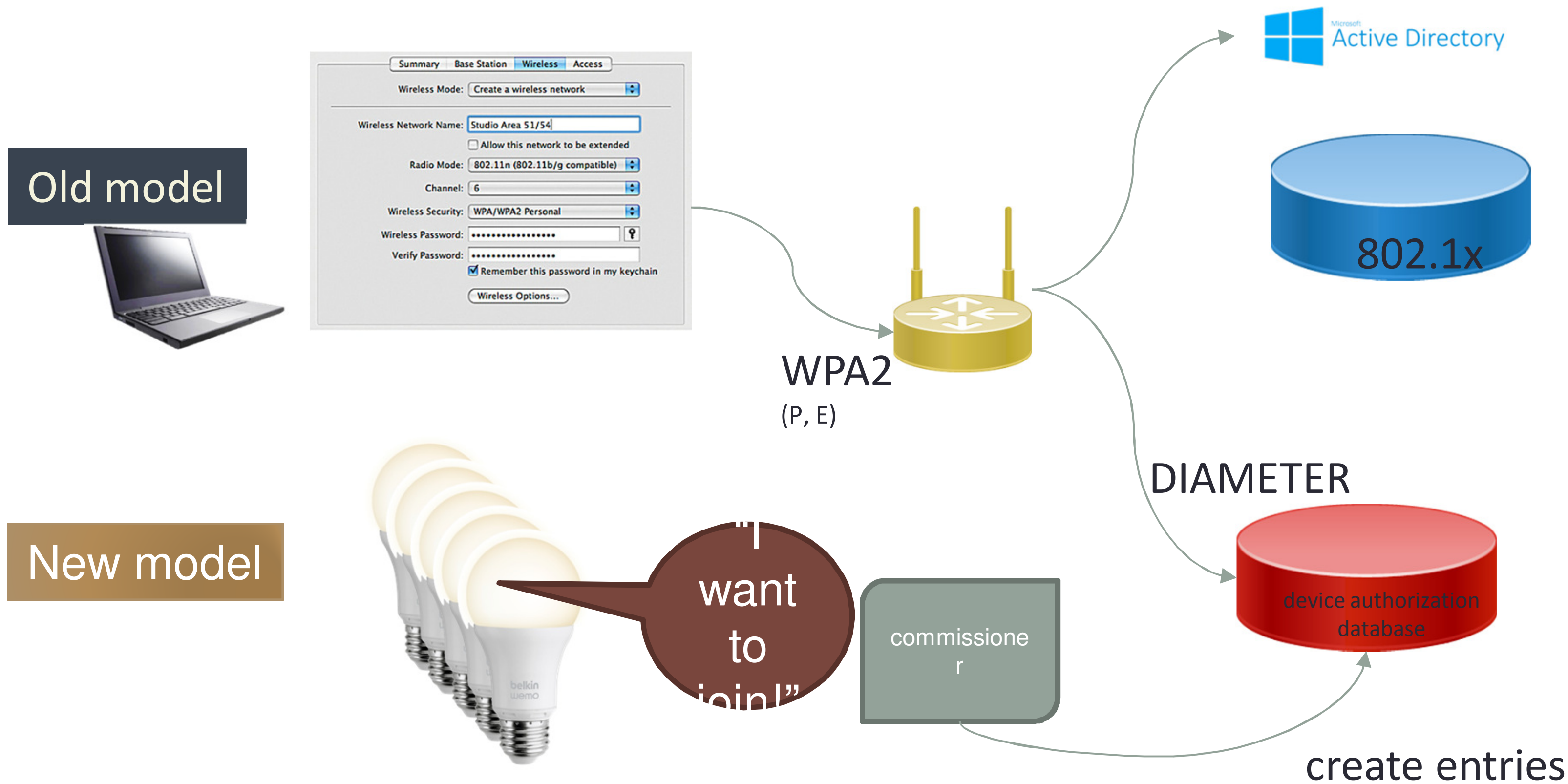
- IoT risks: privacy, DDOS, extortion (ransomware for your freezer), …
- *Securely* field updateable or no connection to Internet
  - still vulnerable if malware on home network
- Lifetime of devices > lifetime of company
- Insurance model:
  - source code escrow + maintenance for N years
- UL listing

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?



PRIVACY

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

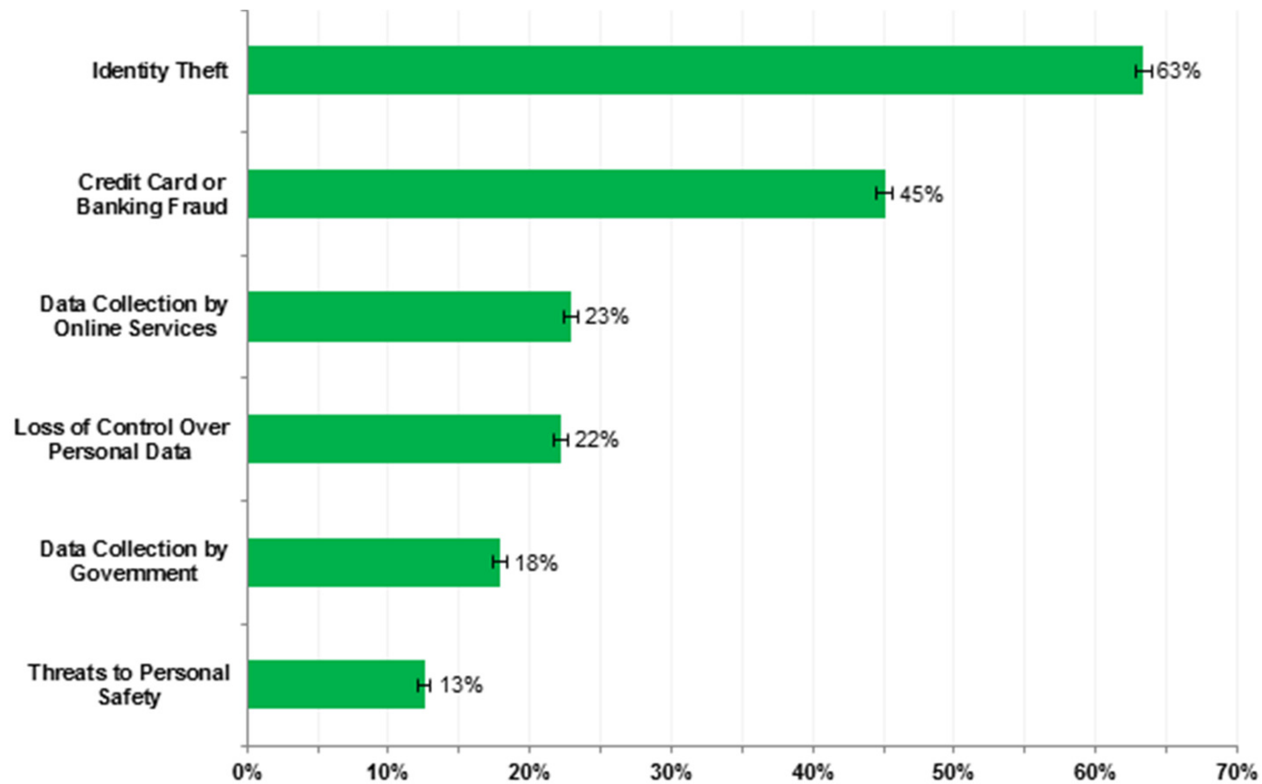
IoT: more than programmable light bulbs

public sensors & actuators

semi-private

private

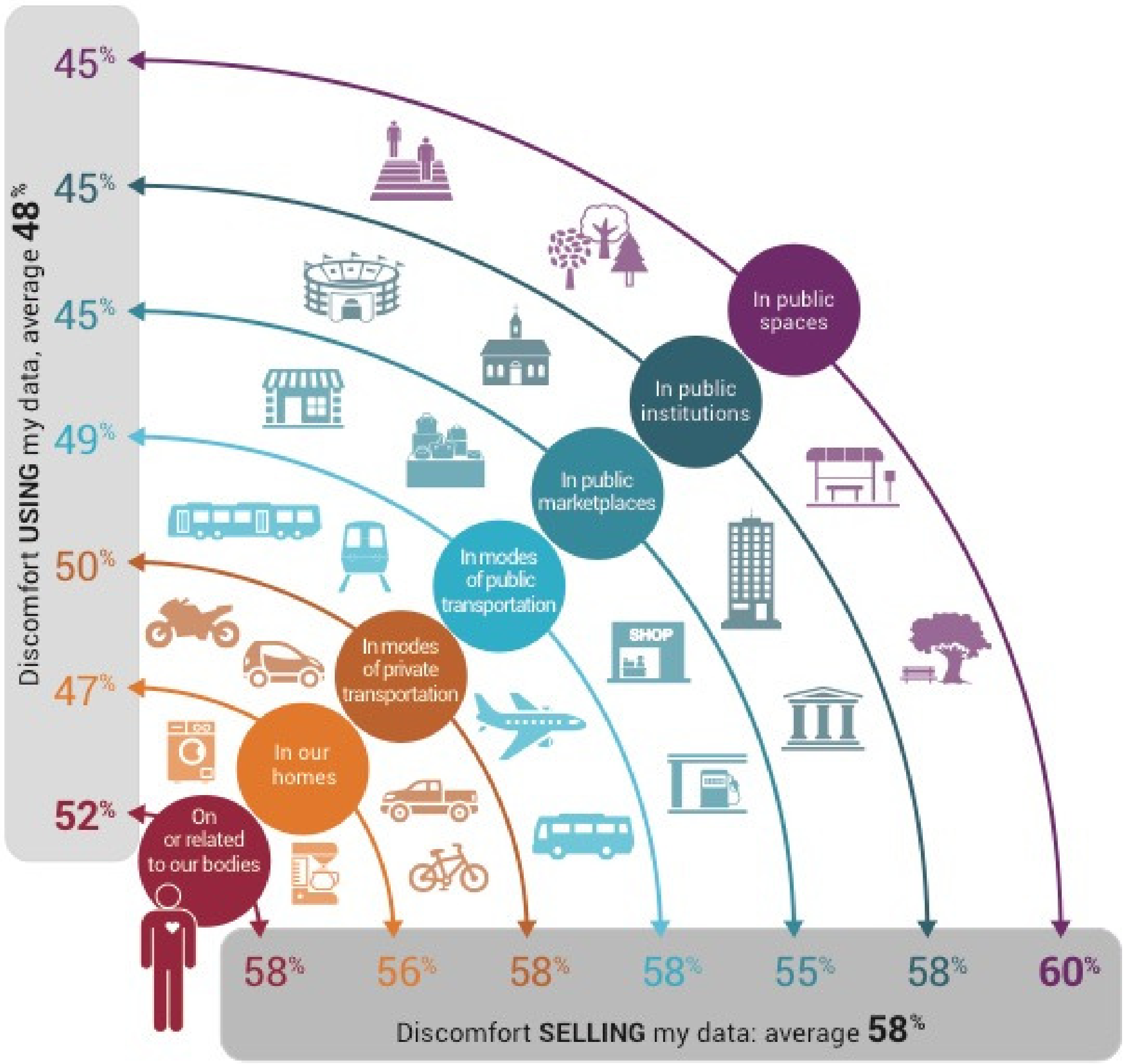
Privacy fears deter usage



NTIA
May 2016

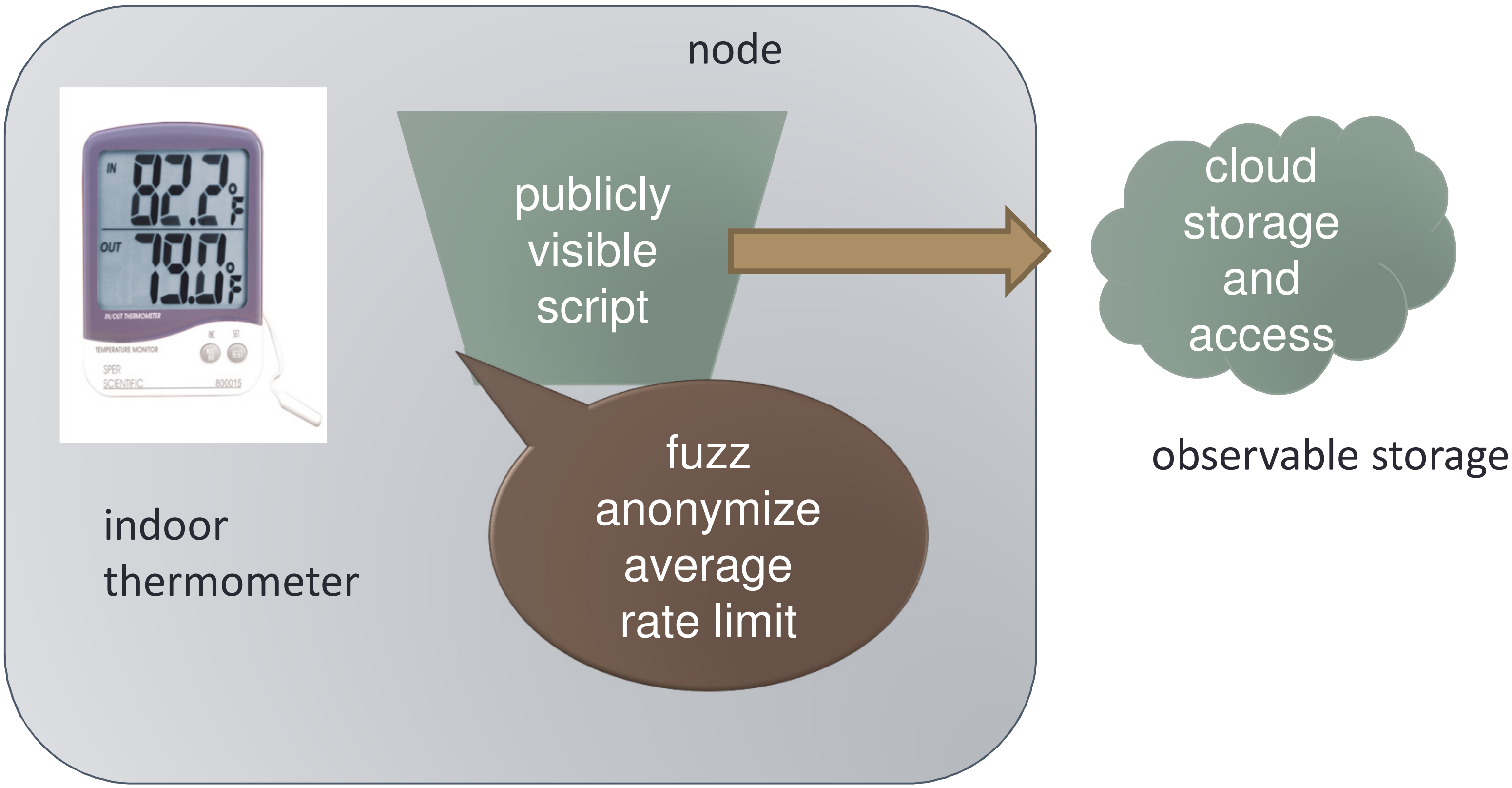
Major Concerns Related to Online Privacy and Security Risks, Percent of Households with Internet Users, 2015

Roughly half of consumers uncomfortable



Altimeter Group
June 2015

Local processing for ~~efficiency~~ privacy



fog computing model

Conclusion

- Design for simplicity and generality, not performance
- Design for surprises
- Design for developers – what do they need and want?
- Design for L2 evolution and co-existence