

# User and IoT-Oriented Network Traffic Monitoring

Luca Deri <deri@ntop.org>  
@lucaderi

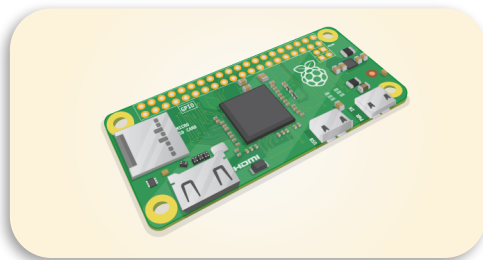
# About Me

- Lecturer at the University of Pisa, CS Department, research grant from CNR Pisa, Italy.
- Founder of the ntop project that develops open source network traffic monitoring applications.
- ntop (circa 1998) is the first app we released and it is a web-based network monitoring application.
- Today our products range from traffic monitoring, high-speed packet processing, deep-packet inspection (DPI), IDS/IPS acceleration, and DDoS Mitigation.
- See <http://github.com/ntop/>



# It all Started with a 5\$ Computer...

- Building low-cost devices able to run full fledged OSs (e.g. Linux) enabled computing to become really pervasive.
- No more excuses for not automating tasks, or rethinking existing processes in a more intelligent fashion.

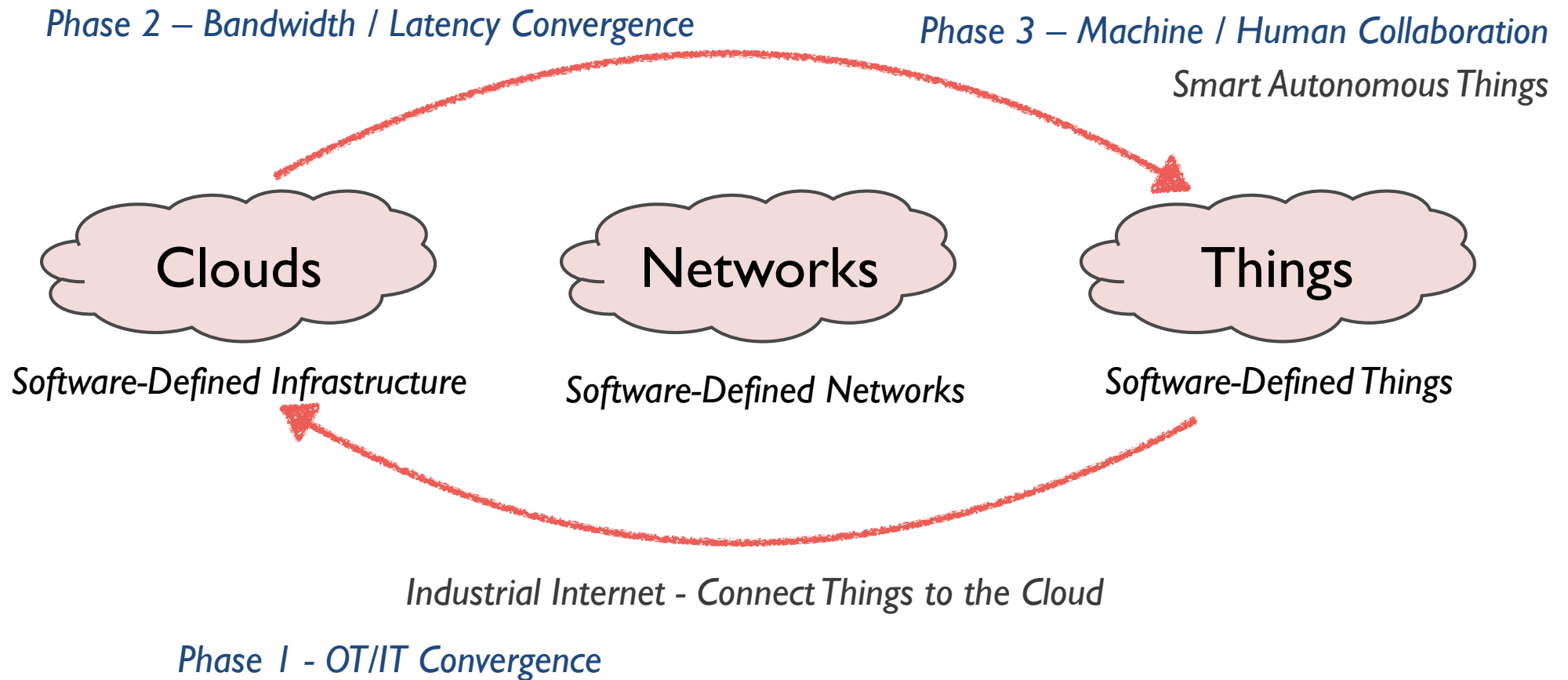


- 1 Ghz, Single-core CPU
- 512MB RAM
- Mini HDMI and USB On-The-Go ports
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers

Raspberry Pi zero (US\$ 5)

Pine64 PADI IoT (US\$ 1.99)

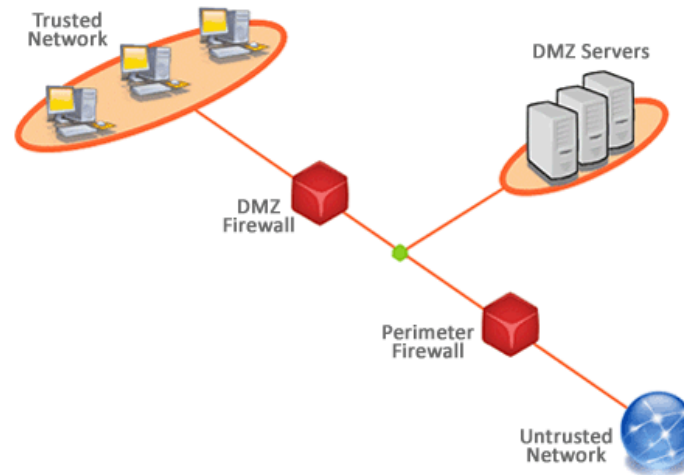
# IoT Transformation



# A Broken Security Model [1/3]

*“Every program and every privileged user of the system should operate using the least amount of privilege necessary to complete the job.”*

Jerome Saltzer



• Procedural Security

• Logical Security

• Physical Security

Denning's Least Privilege Principle

# A Broken Security Model [2/3]

- Low-voltage Environment:
  - Wide-spread use of IoT devices.
  - Increasing interconnection between edge devices and corporate networks:  
an edge device has important topological privileges.
  - Edge devices lack built-in security features: too simple, yes easy to attack or replace with “trojan” devices.
  - Physical location renders networks vulnerable to external attack – even without Internet connection

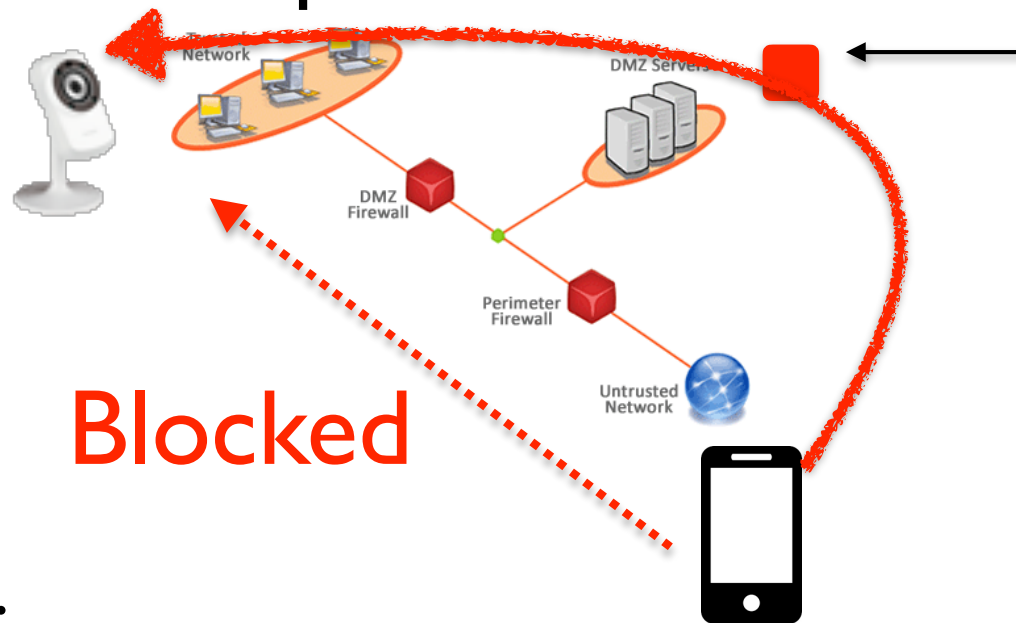


# A Broken Security Model [3/3]

- Unsecured low-voltage devices:
  - Access control
    - Unauthorised opening of gates/doors, false attendance information.
  - Video surveillance cameras
    - Manipulation of video camera streams, unauthorised viewing or disabling video edge-device elements.
  - Building-management/Fire-alarm systems
    - False readings, disabling or blinding.
  - Perimeter IP-based sensors
    - False readings, disabling or blinding.
  - DDoS (Distributed Denial of Service) attacks, can disrupt network operations and thus break a complex system/factory.

# Cloud: Easy vs Safe [1/3]

- When the Internet was created, the distinction between private and local network was clear



This is where the camera was supposed to be *ideally* located:

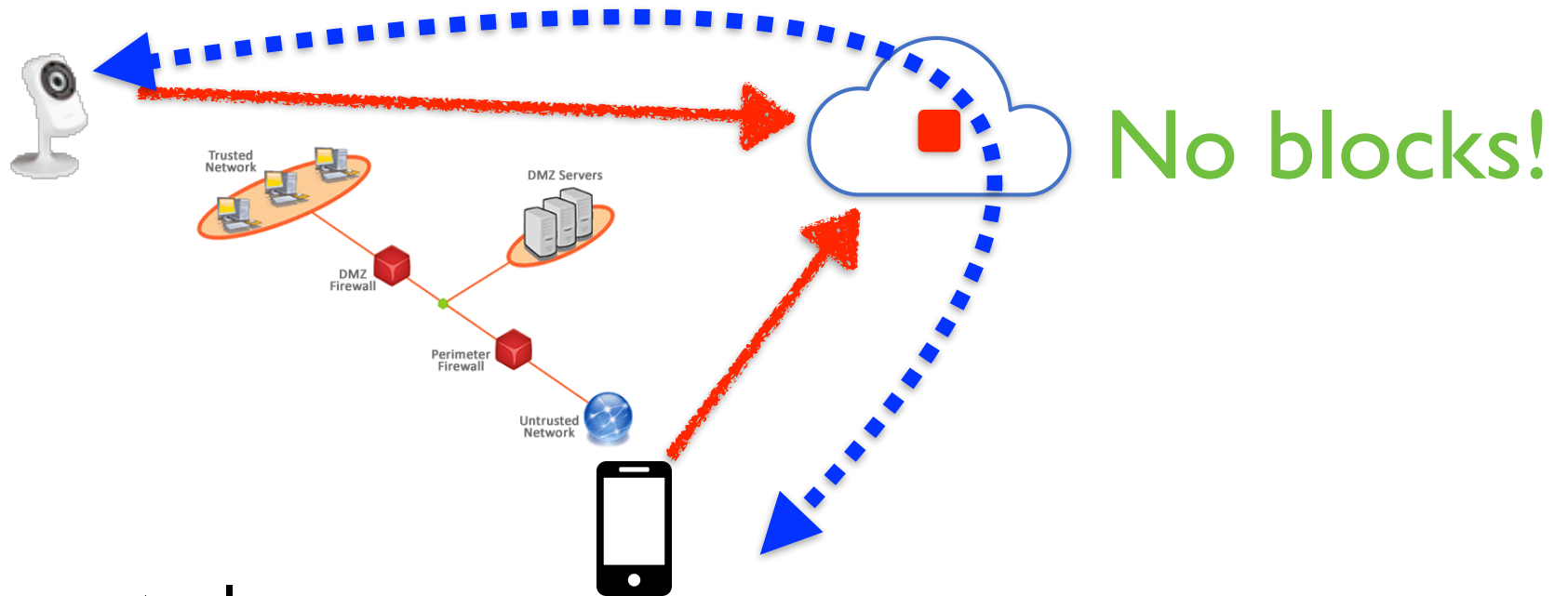
- Open a fixed TCP port
- Use it as a pivot to reach the Internal network

But:

- Most home networks have no DMZ nor static IP
- People do not like to configure anything, just unbox the camera and plug it to electricity



# Cloud: Easy vs Safe [2/3]

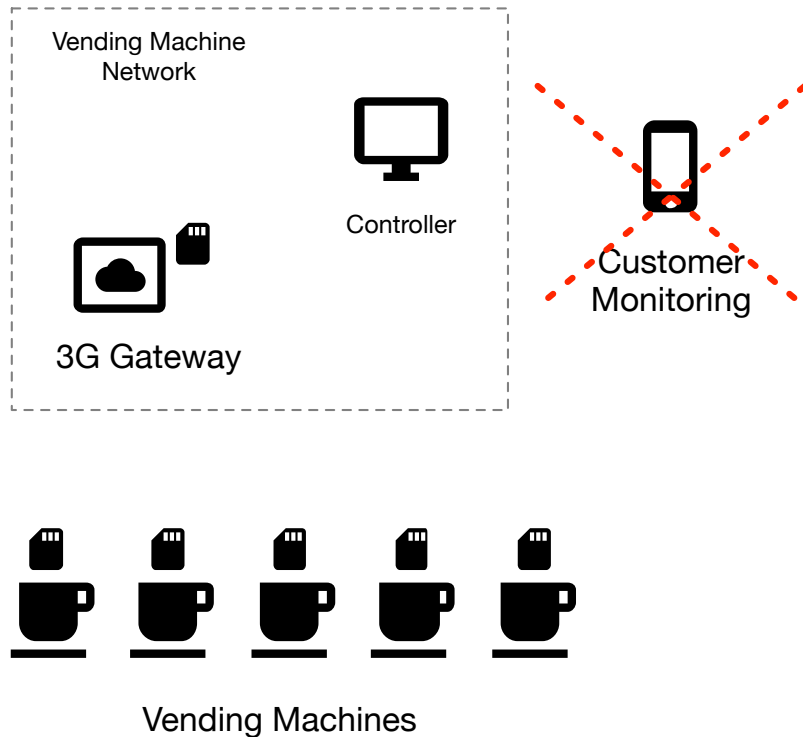


## Caveats

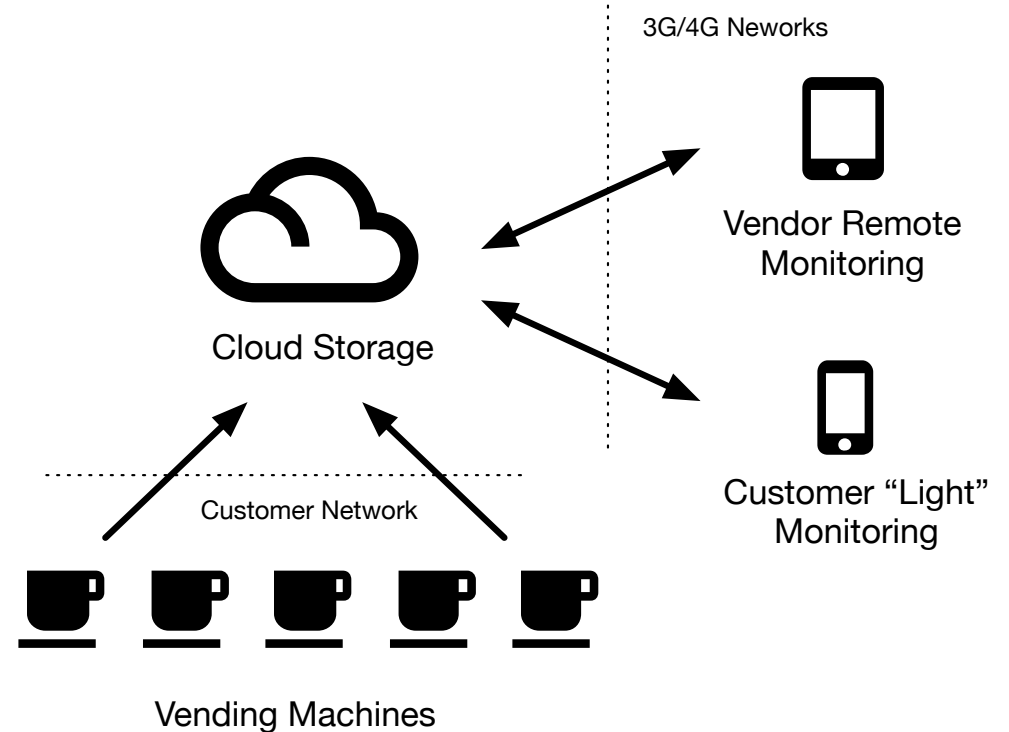
- Access control is managed by the device manufacturer.
- The camera can become a trojan horse if not properly protected.

# Cloud: Easy vs Safe [3/3]

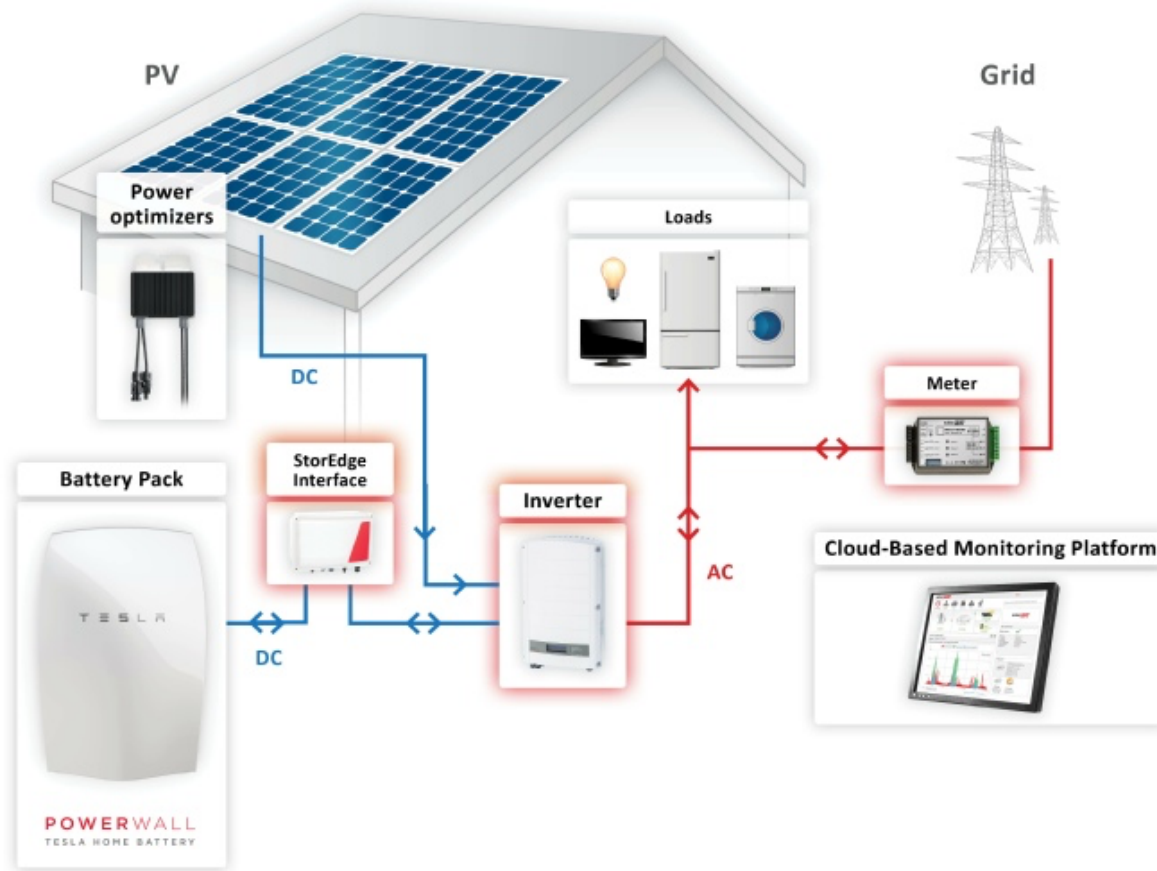
## Before the Cloud



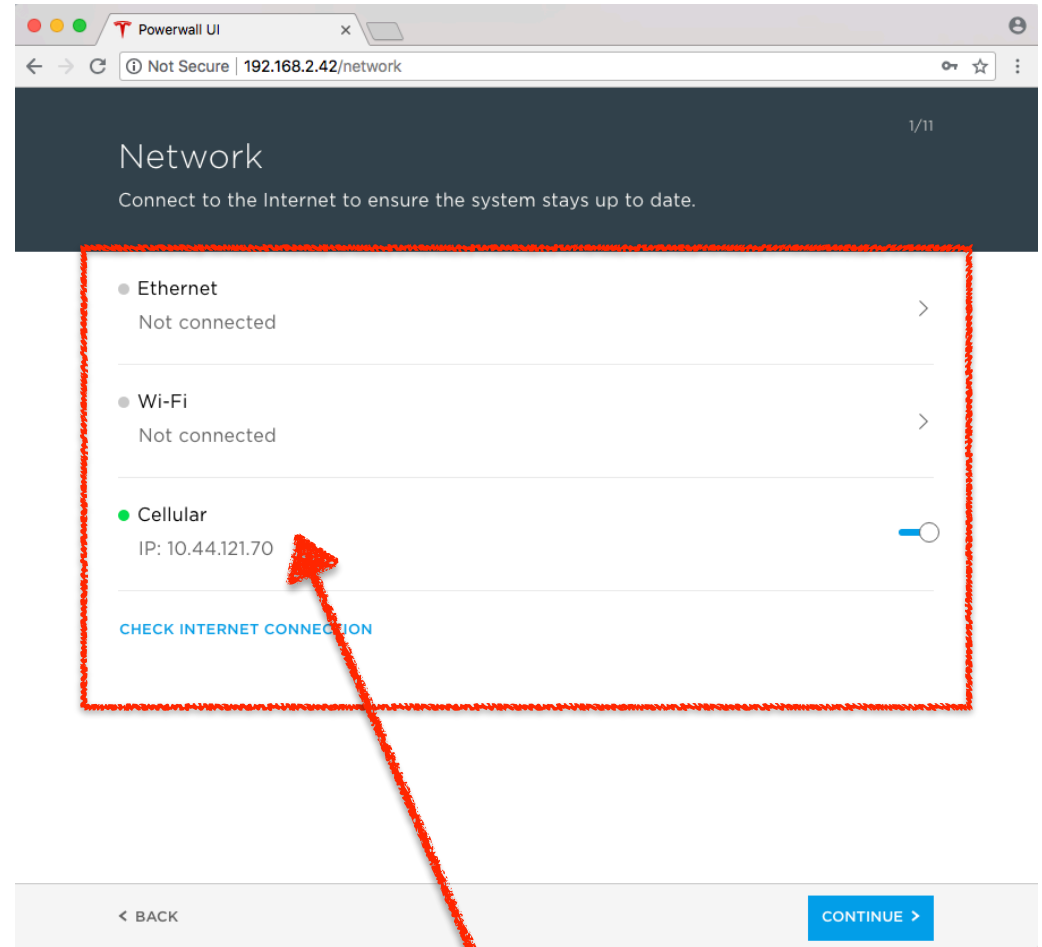
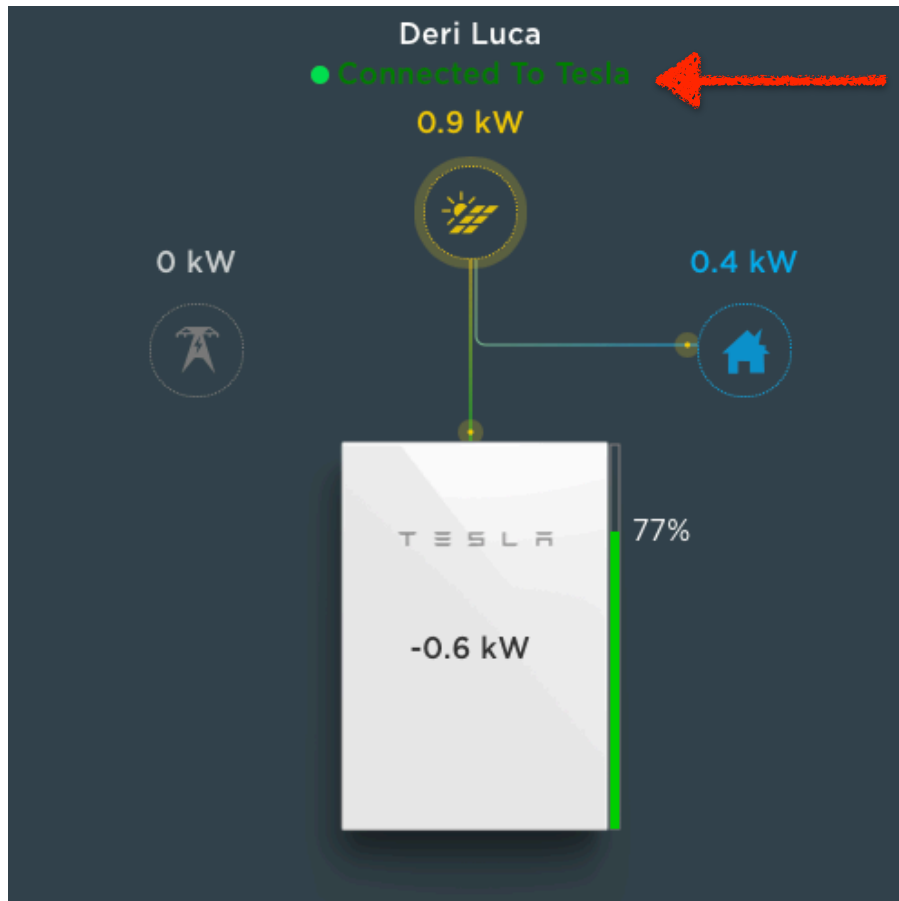
## Today



# IoT Devices in Cloud [1/5]



# IoT Devices in Cloud [2/5]



NOTE Cellular bypasses my home network security devices.

# IoT Devices in Cloud [3/5]



Sonos  
(192.168.177.122)



Alexa Echo Dot  
(192.168.179.172)



Weather Station  
(192.168.179.169)



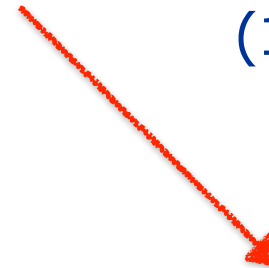
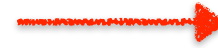
(non-smart) TV



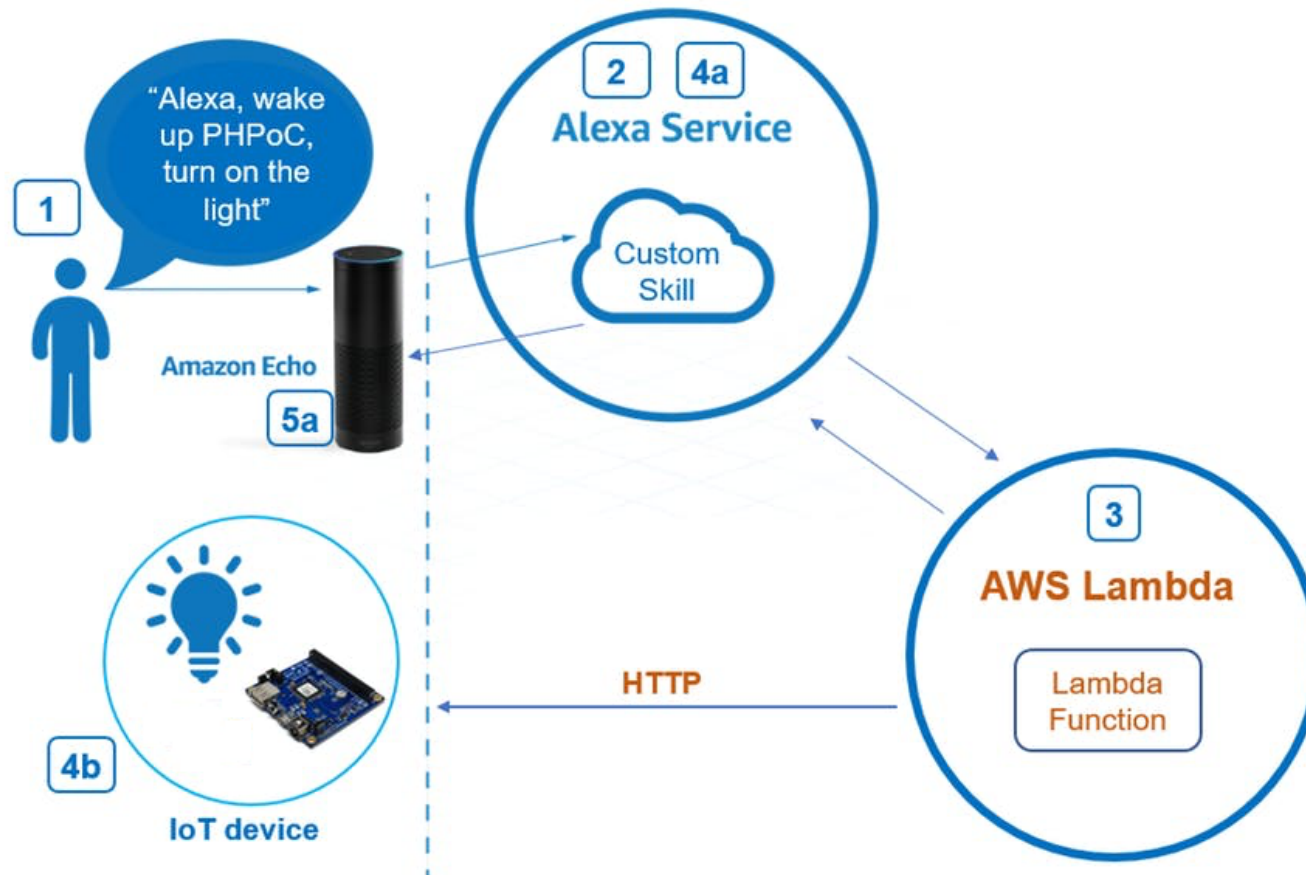
Logitech Harmony  
(192.168.182.11)



Roomba  
(192.168.183.129)



# IoT Devices in Cloud [4/5]



# IoT Devices in Cloud [5/5]

- In essence
  - Direct device communications are no longer the standard communication paradigm. Example:
    - Before: Computer A talks with Printer B
    - Today: Computer A talks with Google Cloud Print, then Google Cloud Print talks with Printer B.
  - Communications are encrypted over proprietary protocols (bye bye RFCs).
  - Security is delegated to the cloud provider that decides who's talking to who based on customer preferences.



# Traditional Network Monitoring Is Becoming Outdated...

- Popular metrics such as bytes, packets, best-match routing are being revisited since users care about latency and application service time.
- Polling-based protocols (e.g. SNMP) are being replaced by push-oriented approaches (e.g. Cisco Telemetry).
- Binary/custom protocols (e.g. NetFlow/IPFIX) are being replaced by (less efficient yet more open) JSON-based data sources so that data can be shared across components.



# Basically We Need to Monitor...

- Dynamic network topologies and moving components.
- Identify IoT devices and threat them differently from “generic” computers (e.g. laptops or tablets)
- Tag network traffic with application protocol and monitor it continuously overtime looking at specialised metrics (e.g. HTTP return code) in addition to generic ones (e.g. jitter and bandwidth).
- As IoT devices are not installed in “controlled environments” (e.g. a rack on a datacenter vs on a corridor) physical security needs also to be monitored.

# IoT Monitoring: Device Profile

- A device profile is a pair

< < Mac, IP, Port >, < Service IN, Service OUT > >



ARP Monitoring



SNMP Device/Bridge Monitoring



L7 services provided by a device  
(e.g. RTP streaming for a camera)



L7 services used by a device  
(e.g. SMTP for sending notifications)

# IoT Monitoring: Traffic Profile

- A traffic profile is a pair  
< < Device, Service, Latency, < Thpt<sub>UP</sub>, Thpt<sub>DOWN</sub> >, Protocol Metadata > >
- Device: subject of the communication.
- Service: Layer 7 (DPI) protocol identification.
- Latency: service time (slow response is a problem for devices such as burglar alarms).
- Throughput: create baseline (e.g. low throughput for a camera is an indication of a problem/attack).
- Metadata: used to pinpoint a problem.

# Monitoring IoT (Security) [1/2]

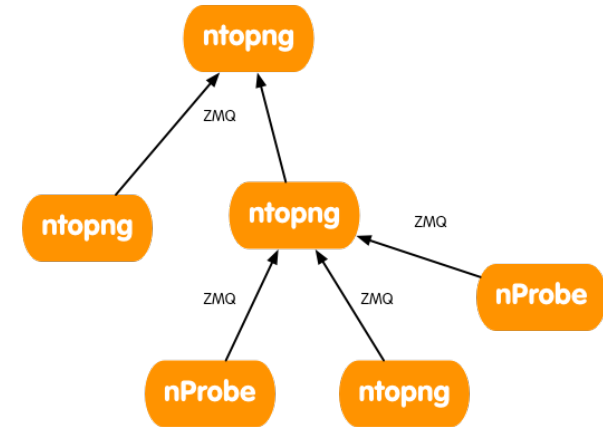
- Learning
  - Identify network elements (discovery), assign them a role (e.g. a printer).
- Profiling
  - Bind a device to a profile (e.g. a printer cannot Skype or share files using BitTorrent) and enforce it via alarms or traffic policy enforcement.
- Continuous Monitoring
  - Physical constraints (e.g. MAC/IP binding and switch port location), traffic constraints (e.g. a new protocol serviced by a device or throughput above/under its historical baseline can be an indication of a problem).

# Monitoring IoT (Security) [2/2]

- In IoT monitoring traffic patterns are rather static and thus once a model is created it must be observed regularly overtime, if not alert.
- Triggers notifications if devices fail due to electrical, software, mechanical or other faults: active monitoring/polling is compulsory.
- Threats
  - External: monitor/detect breaches in the low-voltage network
  - Internal: monitor/detect network threats through unauthorised use (e.g. HTTP access to a device from a client that never did that before).

# Solution Overview [1/3]

- Software-only, low-cost sensors that can be embedded in devices or deployed at the network edge, to create a collaborative monitoring infrastructure.
- Tag devices, traffic, and users.



<https://github.com/ntop/ntopng>

SIP 🛡️ ⚠️ 🔄	0 B	7.06 KB	Rcvd	7.06 KB	0 %
SNMP 🛡️ 🔄	4.5 MB	4.18 MB	Sent Rcvd	8.67 MB	1.54 %
SSH 🛡️ 🔄	593.4 KB	4.11 MB	Sent Rcvd	4.69 MB	0.83 %
SSL 🛡️ 🔄	3.26 MB	4.17 MB	Sent Rcvd	7.43 MB	1.32 %
Skype 🛡️ ⚠️ 🔄	0 B	11.54 KB	Rcvd	11.54 KB	0 %
Tor 🛡️ 🔄	37.34 KB	58.7 KB	Sent Rcvd	96.04 KB	0.02 %

What do we need to hide here?

Ingress but no egress traffic: service scan?

# Solution Overview [2/3]

## All Layer 2 Devices

10 ▾ Filter MACs▾ Manufacturer▾

MAC Address	Manufacturer	Hosts	ARP Sent ▾	ARP Received	Seen Since	Breakdown	Throughput	Traffic
80:2A:A8:8D:69:2C	Ubiquiti Networks Inc.	269	38	8	4 min, 32 sec	Sent Rcvd	9.1 Kbit	4.36 MB
C4:2C:03:06:49:FE	Apple, Inc.	1	10	8	4 min, 32 sec	Se Rcvd	8.75 Kbit	4.37 MB
CC:2D:8C:F6:C7:39	LG ELECTRONICS INC	1	5	2	4 min, 30 sec	Sent R	95.88 bps	14.62 KB
54:4E:90:BA:EC:84	Apple, Inc.	2	5	0	2 min, 16 sec	Sent	361.17 bps	10.22 KB
AC:87:A3:16:3E:30	Apple, Inc.	1	0	0	4 min, 6 sec	Sent	0 bps	2.61 KB
80:2A:A8:8D:2B:EE	Ubiquiti Networks Inc.	1	0	0	3 min, 30 sec	Sent	0 bps	228 B
26:A4:3C:FF:4C:D7	n/a	0	0	0	2 min, 24 sec	Sent	0 bps	468 B
28:57:BE:E3:D7:CF	Hangzhou Hikvision Digital Technology Co.,Ltd.	1	0	0	4 min, 31 sec	Sent	0 bps	13.6 KB
24:A4:3C:FE:4C:D7	Ubiquiti Networks Inc.	1	0	0	2 min, 22 sec	Sent	0 bps	1.45 KB

ARP Stats

Showing 1 to 9 of 9 rows

Hosts Monitoring

Physical Location

Mac: 80:2A:A8:8D:69:2C

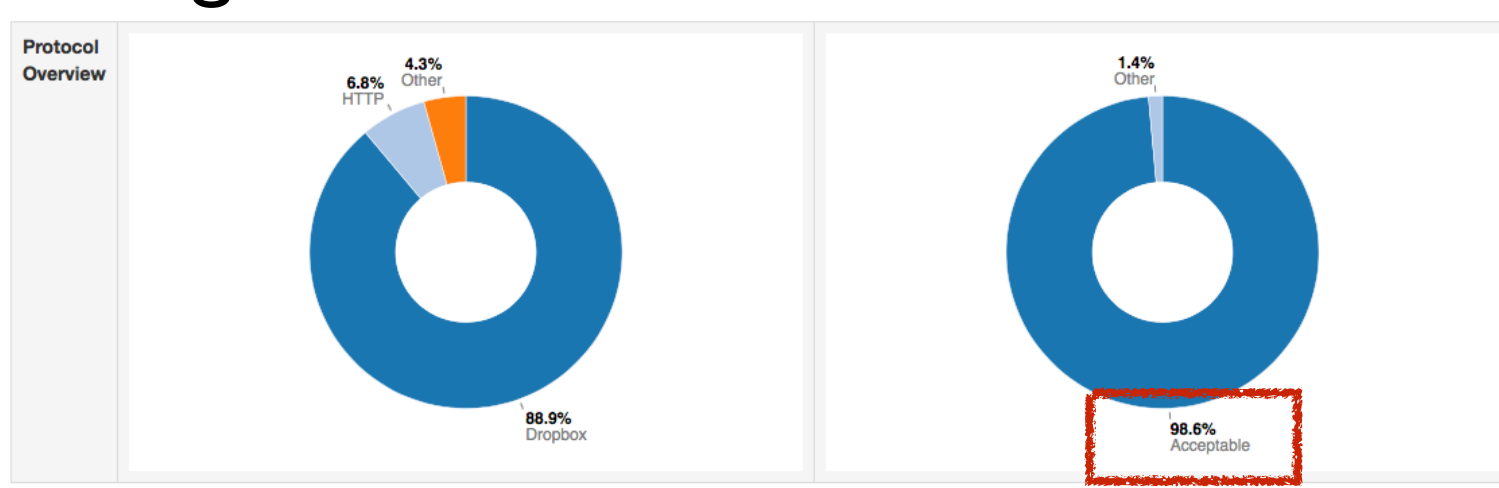
MAC Address	80:2A:A8:8D:69:2C (Ubiquiti_8D:69:2C) [ Show Hosts ]	80:2A:A8:8D:69:2C	<input type="text"/> Save
First / Last Seen	02/04/2017 19:28:54 [4 min, 35 sec ago]	02/04/2017 19:33:26 [3 sec ago]	
Sent vs Received Traffic Breakdown			
Traffic Sent / Received	5,111 Pkts / 3.71 MB	4,558 Pkts / 666.24 KB	
Address Resolution Protocol	ARP Requests	ARP Replies	
	38 Sent / 0 Received	0 Sent / 8 Received	

Device Port

600	ge-0/1/0	trunk
324	ge-0/1/0	trunk
572	ge-0/0/35	

# Solution Overview [3/3]

- Baselineing



- Alerting

Interface: eth0   Home   Packets   Protocols   [Line Graph]   [Print]   [Alert]   [List]   [Settings]   [Group]   SNMP   [Refresh]

⚙️ General Settings   ⚙️ Every Minute   ⚙️ Every 5 Minutes   ⚙️ Hourly   ⚙️ Daily

<b>Interface Alerts</b>	<input checked="" type="checkbox"/> ⚠️ Trigger alerts for Interface eth0
<b>Rearm minutes</b>	<input type="text" value="1"/> [Dropdown]   Save

The rearm is the dead time between one alert generation and the potential generation of the next alert of the same kind.

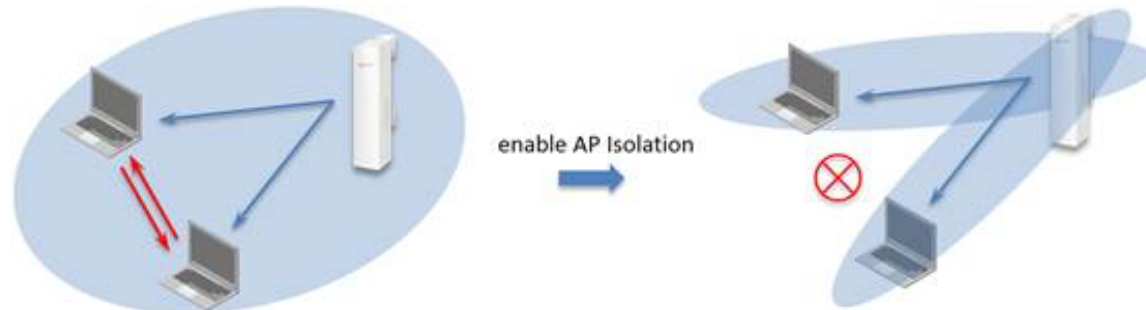


# Next Step: Mitigation and Prevention

- Monitoring is nice to have. However it cannot be used to block threats, just to spot them.
- New efforts such as Manufacturing Usage Description (MUD) will help in the future but they are just a hint from the manufactured, thus untrusted.
- What to do in the meantime?
  - Prevent devices at the edge from doing unwanted communications.
  - Limit and cleanup east-west traffic.
  - What about mobility? Are cloud services the right answer?

# Jailing Devices with Overlays and DPI [1/4]

- Jail devices and prevent them from doing unwanted traffic (i.e. micro segmentation).



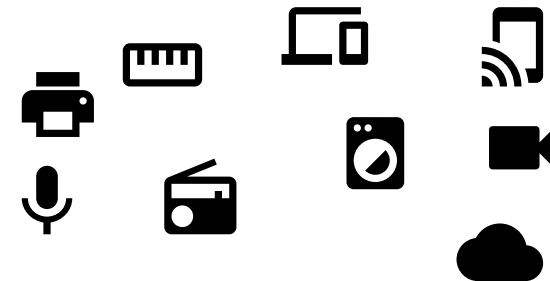
- Easy to do in wireless, but not on wired.
- How to implement layer-7 device micro-segmentation on wired and non-local devices?

# Jailing Devices with Overlays and DPI [2/4]

- Lisa is sick: she needs to keep connected her health care device from the home network with the hospital.



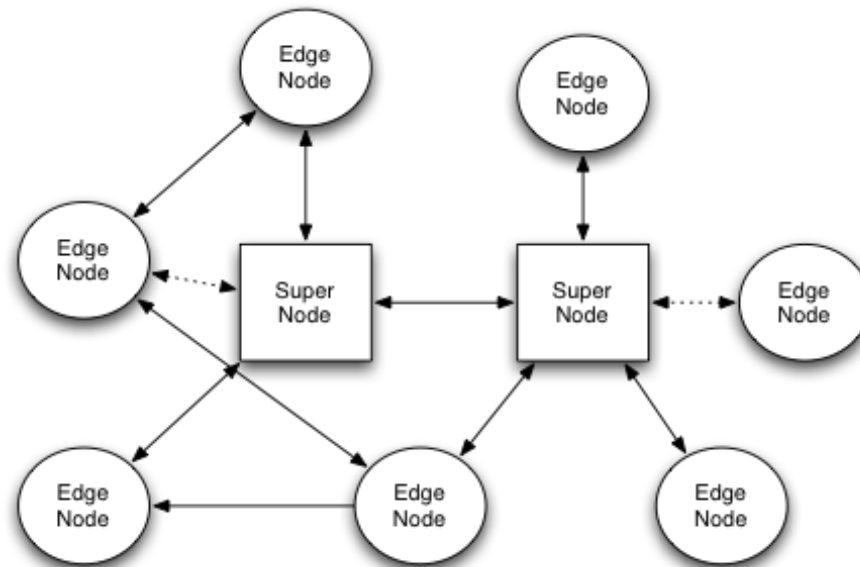
- John manages a fleet of trucks for food delivery. Lisa is John's secretary: from home she carries on her work.




- Some Challenges:
  - Lisa home devices should not be mixed with John devices.
  - A security flaw should not affect both networks.
  - How to contact mobile devices with a non persistent IP address ?

# Jailing Devices with Overlays and DPI [3/4]

- <https://github.com/ntop/n2n> (Linux, Windows, MacOS, Android) implements a peer-to-peer overlay for interconnecting devices on a secure fashion.

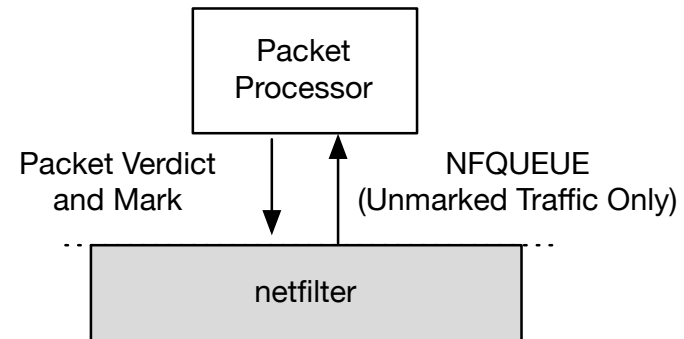


# Jailing Devices with Overlays and DPI [4/4]

- <https://github.com/ntop/ndpi> is a GPL DPI toolkit in order to build an open DPI layer able to dissect ~240 protocols. 
- Idea:
  - Use nDPI in the n2n edge to allow only permitted communication protocols. Enable routing across overlays only for the permitted flows.
  - The n2n supernode enforces communications policies across edge peers and implements device isolation, either local/remote wired/wireless.

# Putting Pieces Together [1/2]

- Low cost Linux-based routers have an embedded switch that could be used to analyse the traffic across ports (software bridge).
- Leveraging on nDPI and iptables it is possible to analyse only the first we connection packets to enforce verdicts (> 300 Mbit on EdgeRouterX).



# Putting Pieces Together [2/2]

- IoT devices that can run n2n natively will be protected by the local edge component that will enable connectivity in compliance with the network policy.
- “Closed” IoT devices are policed by nDPI-powered switches that will permit only selected communication flows.
- In summary n2n+nDPI implement persistent and secure network overlays using open source software on Linux-powered low-cost hardware.

# Final Remarks

- IoT and cloud computing create new monitoring challenges and require an *integrated monitoring* approach: element + periodic active scans + permanent passive traffic monitoring.
- Monitoring hundred/thousand devices require *scalability* and *intelligence* in the monitoring platform (analytics and big data is not enough, platform must be reactive, distributed, multi-tenant).
- Combining network overlays with DPI it is possible to enforce traffic policies and implement a persistent and micro-segmented layer for IoT and cloud communications.