FUNDAMENTAL CAPABILITIES OF IOT SYSTEMS

REQUIREMENTS AND TRADEOFFS

Ravi Pappu

Chief Architect, InQTel

FOUR BIG IDEAS

1. SOFTWARE REPRESENTATIONS OF THINGS

> *Anything that can be represented by software will be represented by software…separating the logical content of objects from their physical representation – cleaving the bits from the atoms.*

2. INVISIBLE TECHNOLOGY

> *The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. - Marc Weiser*

3. MEASUREMENT, MEASUREMENT, MEASUREMENT

> *Software is eating the world, but the world can’t eat software. Some of the largest challenges facing our species cannot be solved by code; we cannot program away climate change, water contamination, crowded cities, or hunger.*

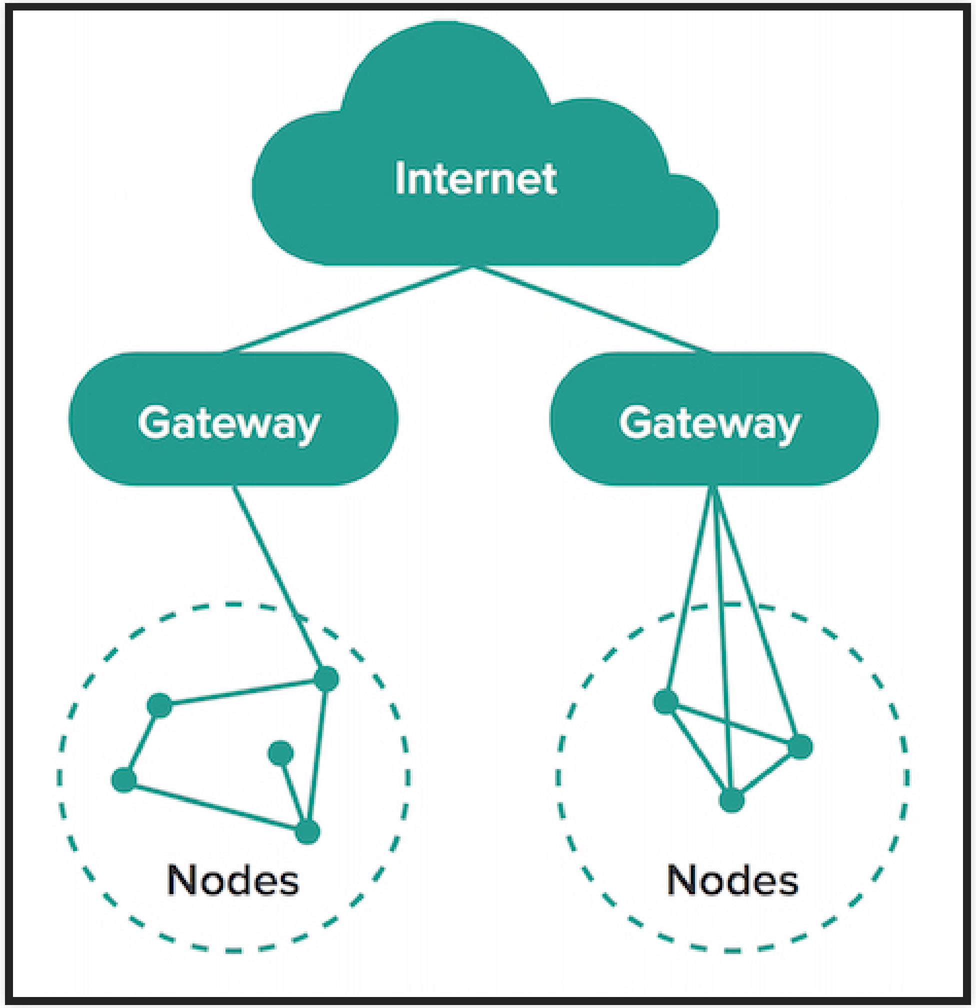
4. RECOMBINANT TECHNOLOGY CAPABILITIES

The more technology artifacts we have, the more we will have, owing to the power of recombination. Engineering relies on the encapsulation of discrete capabilities into modular artifacts that can then be combined to create new artifacts, which can themselves be modularized, and so on.

DEFINITION

The Internet of Things makes the physical world amenable to computation.

DOMINANT PARADIGM



FUNDAMENTAL CAPABILITIES

1. Communication
2. Hardware
3. Software
4. Management
5. Security
6. Analytics

Sensing and Power are also core capabilities. I chose to omit them here because they are very domain-specific.

COMMUNICATION

EXERCISE

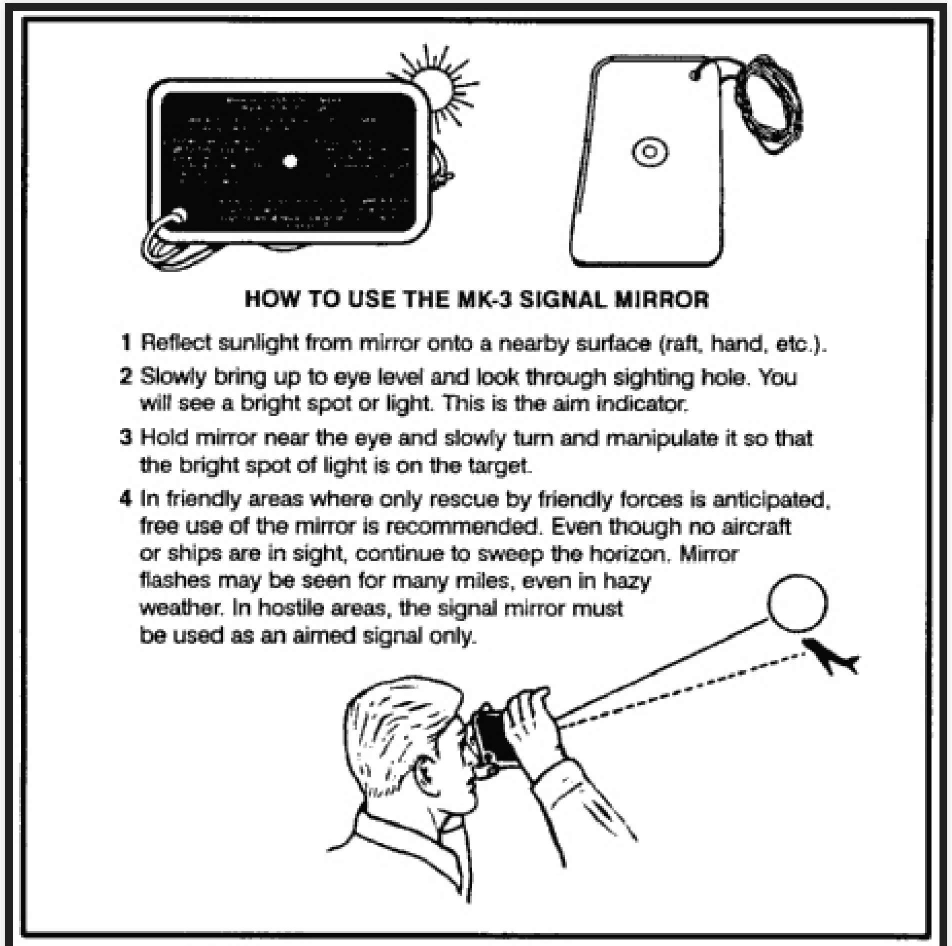


Figure 19-3. Signal mirror.

IOT COMMS REQUIREMENTS

1. Low cost
2. Long range
3. Long operating times i.e., low power consumption
4. High concurrency
5. Efficiency: optimized for short data payloads (IPv6 won't work)
6. Mobility
7. In-building penetration
8. Bi-directional communication
9. Global license-free operation

GOVERNING EQUATIONS

Shannon Capacity

$$C(b/s/Hz) = B\log_2(1 + \frac{S}{N})$$

Friis Propagation

$$P_{rec} = P_{xmit}G_{xmit}G_{rec}\frac{(\lambda)^2}{(4\pi r)^2}$$

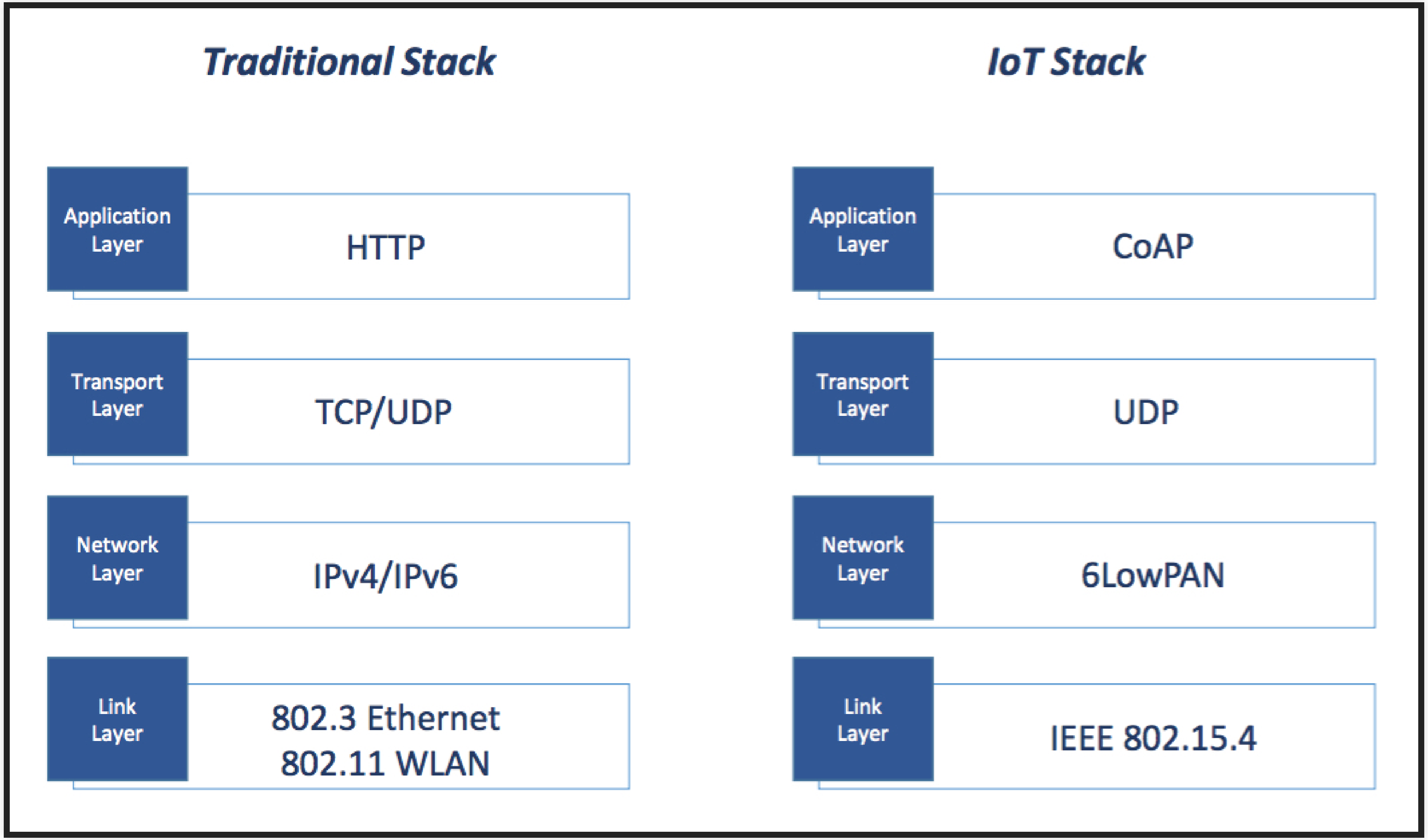
ENERGY OF ELECTRONICS VS. COMMMUNICATION

DATA RATE VS. RANGE

COMMUNICATIONS TRADEOFFS

1. Range vs. Power
2. Directionality vs. Size or Cost
3. Bandwidth vs. Power
4. Data rate vs. Cost
5. Computing vs. Communication

COMMUNICATION STACK



COMMUNICATION LANDSCAPE

HARDWARE

Includes 4 major subsystems: Computing, Communication, Storage, and Analog-to-Digital conversion

HARDWARE REQUIREMENTS

CPU POWER CONSUMPTION

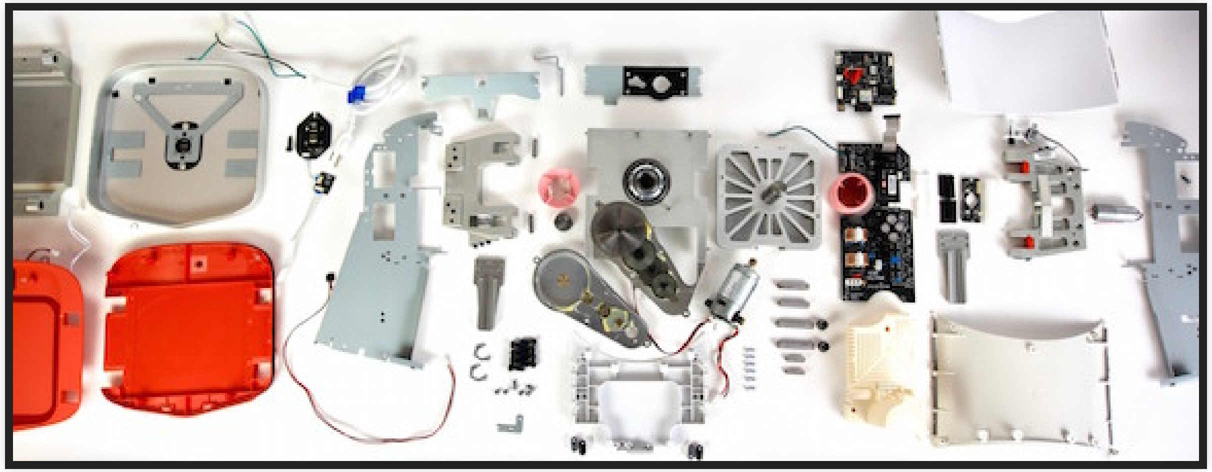
Total Power Consumption

$$P_{total} = P_{dyn.} + P_{sc} + P_{leakage}$$

Dynamic power

$$P_{dyn.} = CV^2 f$$

MAKE VS. BUY



Premature optimization is the root of all evil - Donald Knuth

EARLY CUSTOMERS WON'T PAY FOR...

1. How complex your hardware supply chain is
2. How hard you had to work to screenprint your 3-color logo on the case.
3. The 55 different types of fasteners you have in your assembly.
4. An expensive custom-colored micro-USB cable.
5. 72.5 iterations of your product design from a highly-rated ID firm.

≡

SOFTWARE

OS REQUIREMENTS

OS CHOICES

Two approaches - top-down and bottom-up

IoT-specific OSes - RIOT, Contiki, TinyOS, ROS

General-purpose OSes - Linux, Android, Brillo, etc

Full-stack platforms - mBed, AWS IoT etc.

A NEW APPROACH - PLATFORMIO.ORG

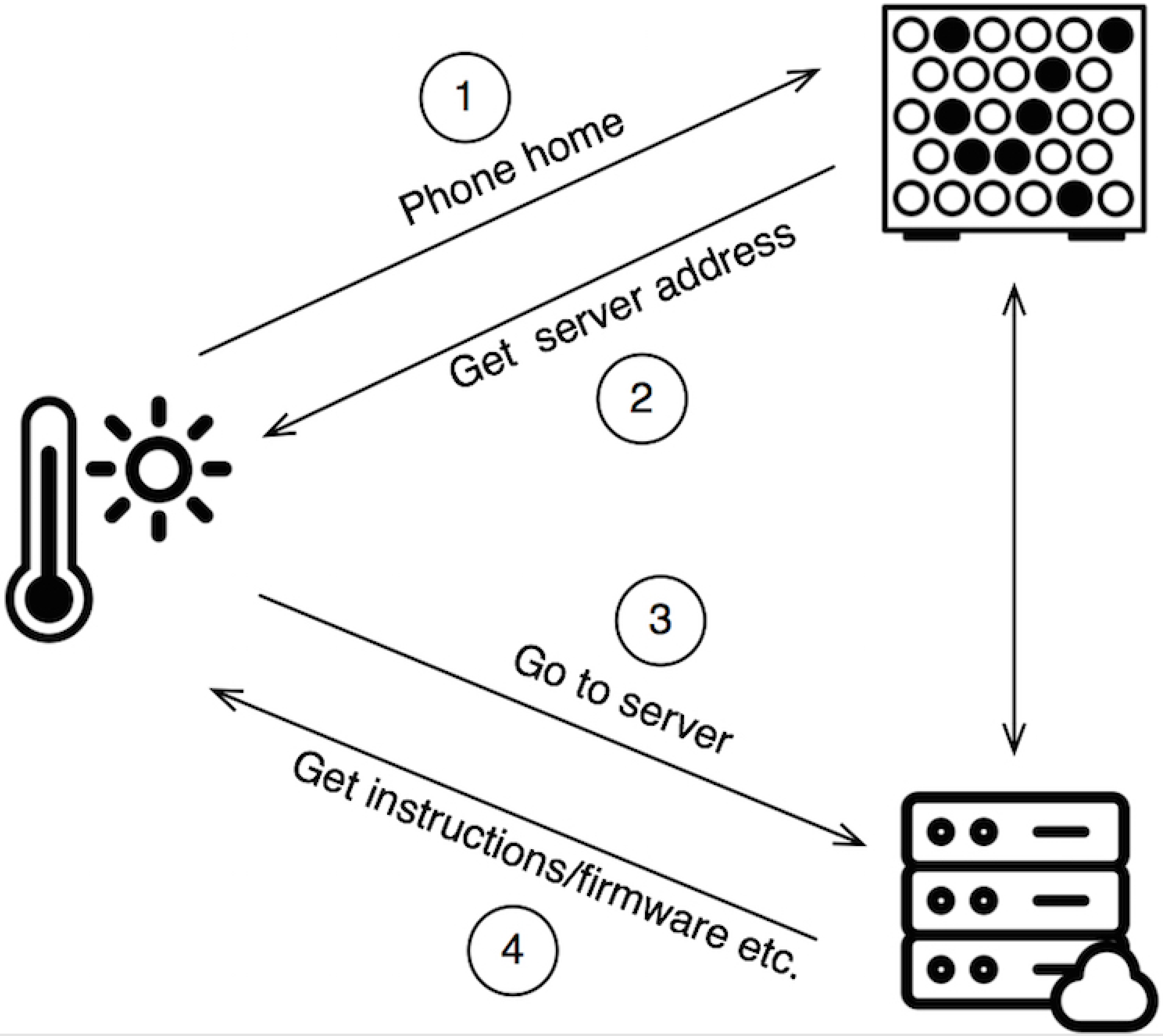
A NEW APPROACH - PLATFORMIO.ORG

MANAGEMENT

> *Managing involves Provisioning, Controlling, Updating, Logging and Monitoring IoT networks*

REQUIREMENTS FOR MANAGING IOT

1. Resources: Constrained Nodes which cannot implement complex management protocols
2. Scale: Support for thousands of nodes
3. Occasionally connected: Nodes can be powered down or go out of range at any time
4. No downtime: Managing activities might need to take place while the device is operational
5. High cost of failure: Failures could cause the Node to be “bricked”, leading to expensive recalls


1
Phone home
Get server address
2
3
Go to server
Get instructions/firmware etc.
4

SECURITY

Securing IoT Nodes comprises:

Data integrity
Authentication/authorization
Confidentiality

CHALLENGES FOR IOT SECURITY

Securing IoT

RFID tag · ASIC · FPGA · uC · Consumer Devices · Smart Phone · Laptop · Server

capability

1 INTEGRITY

Is the system working as designed and is the data from the Nodes trustworthy?

Difficult to achieve physical security and data integrity given resource and cost constraints.

Software tamper resistance is economically feasible in this regime, but side-channel resistance is cost prohibitive.

Economically feasible to incorporate physical security and measures to preserve data, hardware, and software data integrity.

2 AUTHENTICATION

Are Nodes talking to legitimate entities who have privileges to perform relevant actions?

3 ENCRYPTION

Is data from nodes private?

Lightweight Crypto is targeted at these platforms, but is still in its infancy

Existing NIST or ISO standards can and should be implemented on these platforms.

ANALYTICS

Extracting insight from the data collected in IoT systems

- ***Aggregate** device analysis: higher fidelity results from many lower fidelity sensors*
- *Pushing analytics to the **edge***

IOT-SPECIFIC CHALLENGES

1. Big Data: Machine generated data brings increase in data volumes and speed of ingestion
    - Requires rethinking of traditional ingestion systems and methods.
    - Efficient methods to store and retrieve large numbers of small-payload needed.
2. Occasional Connectecness: Data can arrive out of order
3. Privacy: Privacy implications of individually-identifiable data in sensor-rich world

ADVANCES IN ANALYTICS

STREAMING ALGORITHMS

Computing some function of the input data stream when:

- The amount of memory available is much smaller than the size of the entire stream
- We want to access each element of the stream as few times as possible (preferably once)
- There's only a limited amount of processing time available for each item

SOME POPULAR STREAMING/PROBABILISTIC PROBLEMS

- Average value of a sequence of measurements
- Top-K elements (also called heavy hitters)
- Distinct elements of a data stream
- Event detection
- Set membership
  - returns either "possibly in set" with bounded probability or "definitely not in set"

FINAL THOUGHTS

1. Solve real problems.
2. Get only as much data as necessary - or less if possible.
3. Use schemaless data models so you can evolve your data products.
4. Don't optimize until you have a real problem.
5. Use logs and monitoring to see if you have a real problem.
6. Always know the theoretical limits of your systems.
7. When you see a good move, think of a better one.