Hardware Technology Impacts on IoT Space

Selected Topics

James Espy

Manager and Senior Principal Software Engineer, Red Hat

Agenda

- Introductions
- Enablers and Drivers
- Architecture and Examples
- Component and System Design Considerations
- Breakout Session
- Q&A
- … Interspersed with open discussion

Who Am I?

- Long career in storage and related product development
- From spinning disk drive servo to large systems design
- 2nd year at Red Hat
- Always drawn to hardware - So why am I at Red Hat??
- Music student (jazz guitar)

Enablers and Drivers

How did this

happen?

Yet Another IoT Point of View...

You've heard many definitions already

I

Interconnect
Internet

Ubiquitous connectivity.
Otherwise just <u>isolated</u> things.

o

T

Things (Usually) =
Tangible i.e. HW*

Sometimes associated with the so-called "EDGE" or M2M

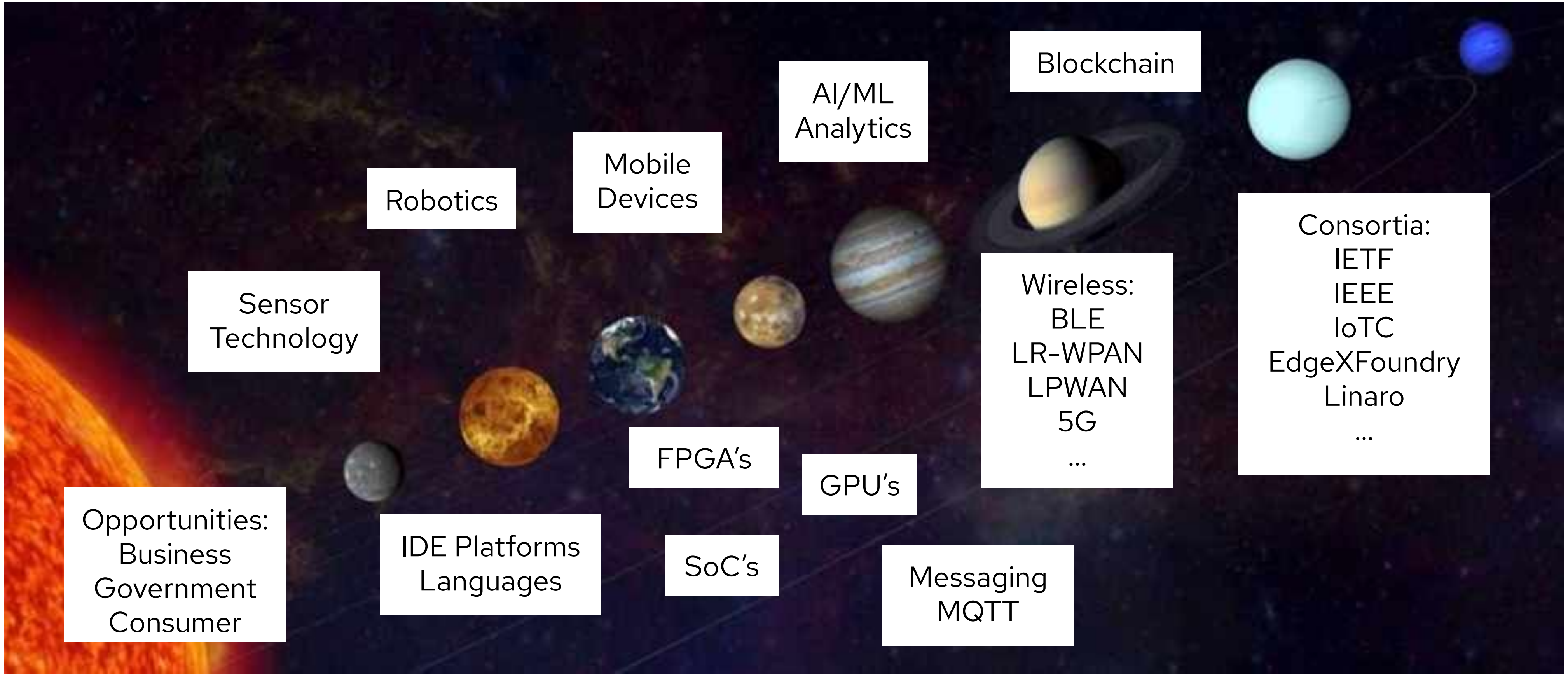
<u>Impact</u>: Interconnectivity aspects enable the collection of huge amounts of data and control of a wide spectrum of devices creating a whole new range of applications and management.

Red Hat

*Everything we'll talk about could also include virtual entities... but we'll stick with hardware for the sake of this discussion.

Ecosystem Alignment

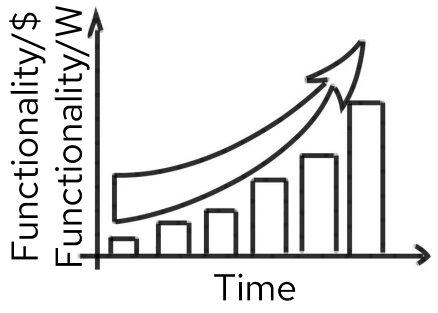
Many coinciding technologies and business trends/needs



Red Hat

Particular Enablers

- Economics - Driven to a great degree by the mobile marketplace
  - Increasing functionality
  - Lower cost and power per function
  - SoCs, GPUs, FPGAs, ...
  - Particularly reduced interconnect cost - Economical sensor attach

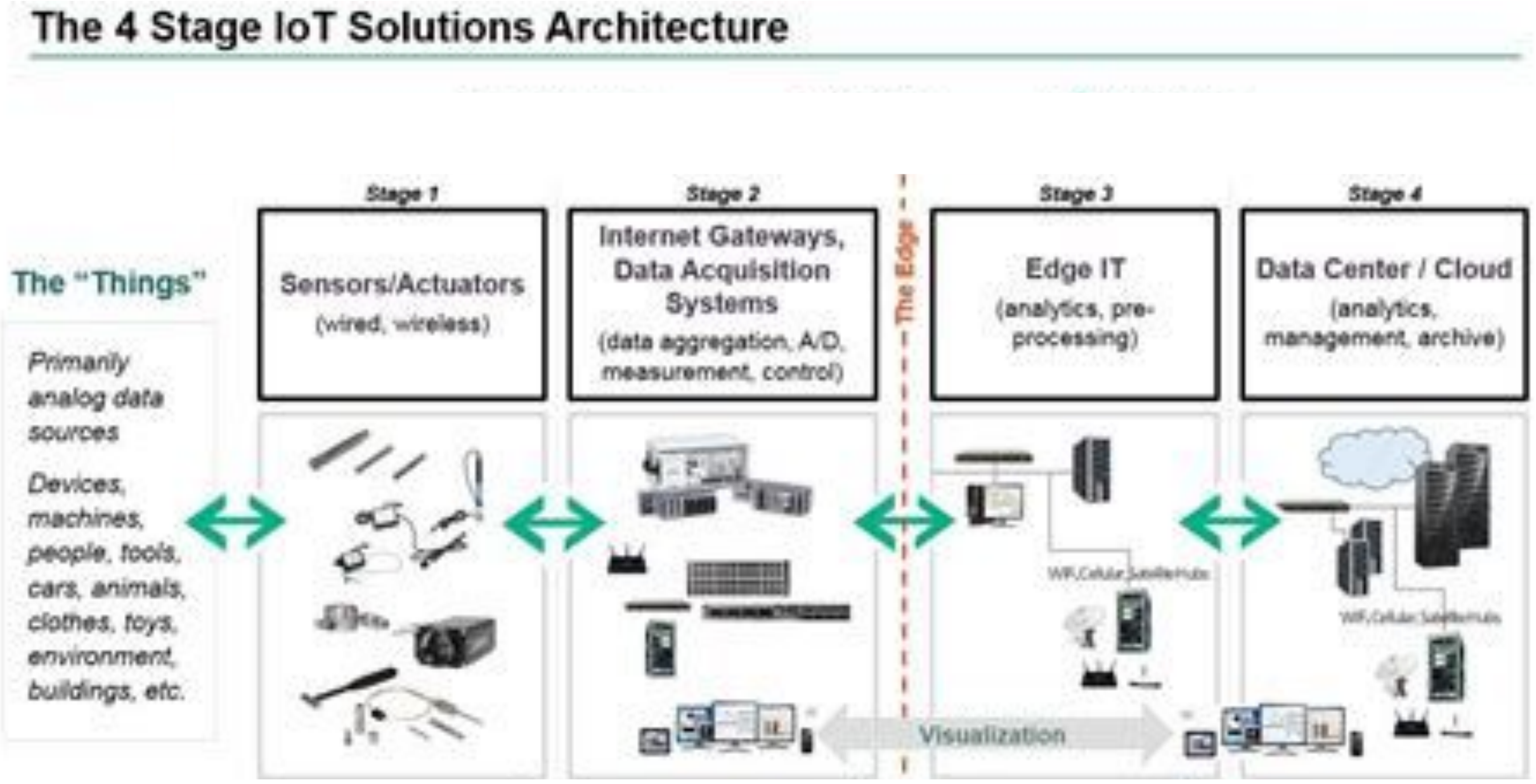


- Supports moving "up the stack"
  - What were (still are) complex systems (e.g. servers, robots)...
  - ... are now components of larger, interconnected systems

Yet More Enablers

- Development platforms (IDEs) - Just a few...
  - Arduino and Raspbian
  - OpenSCADA - Industrial
  - Eclipse Kura - Java
- Consortia - Specifications, standards (SW and HW) ---> OPEN
  - Linux Foundation: EdgeXFoundry, ACRN, Node-RED
  - IETF, IEEE, and ETSI
  - Linaro (ARM SW), ARM MBED (RTOS) & Pelion (full management)
  - IoTC, Industrial Internet Consortium, OpenFog, ...

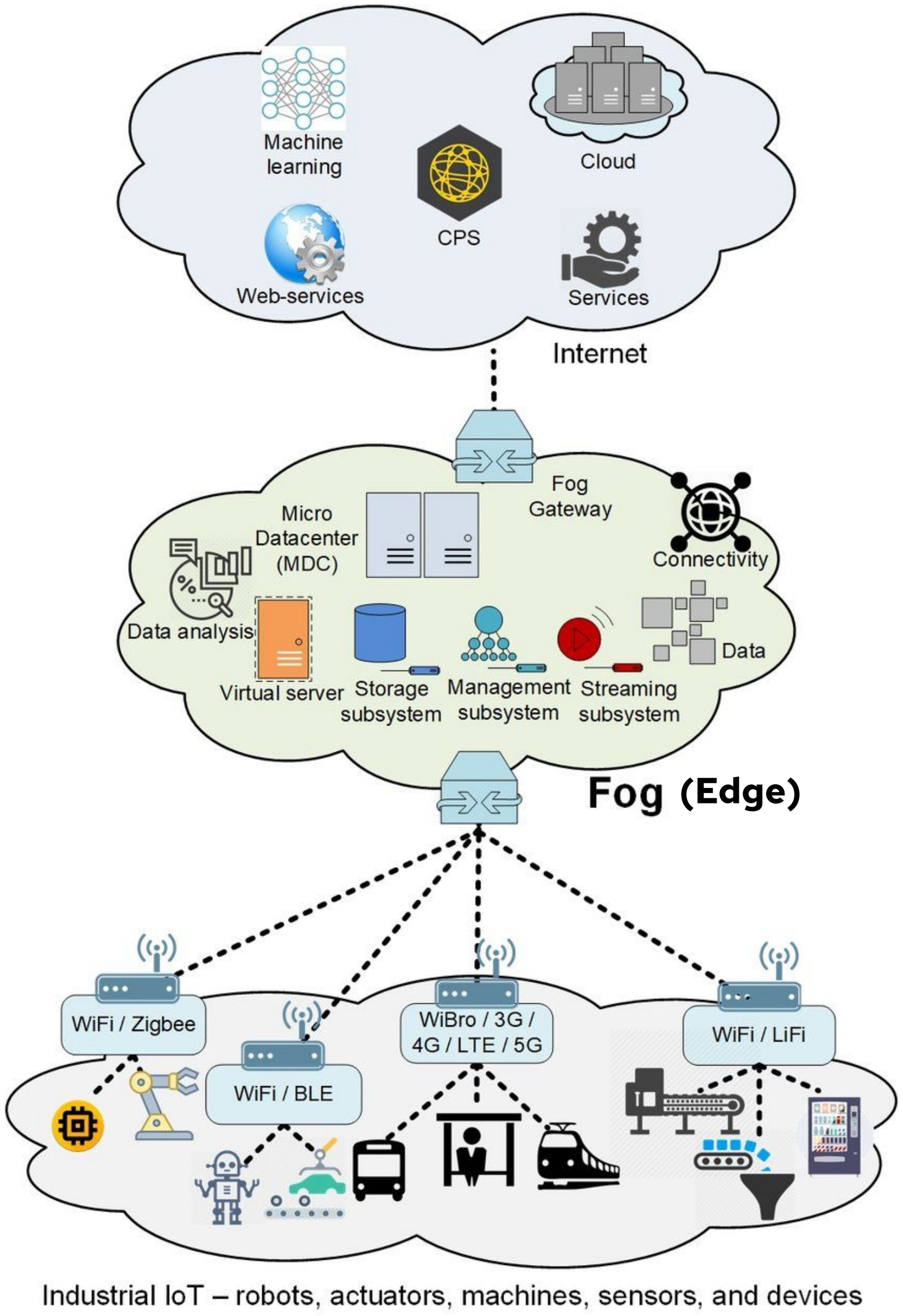
Business – Technology Model



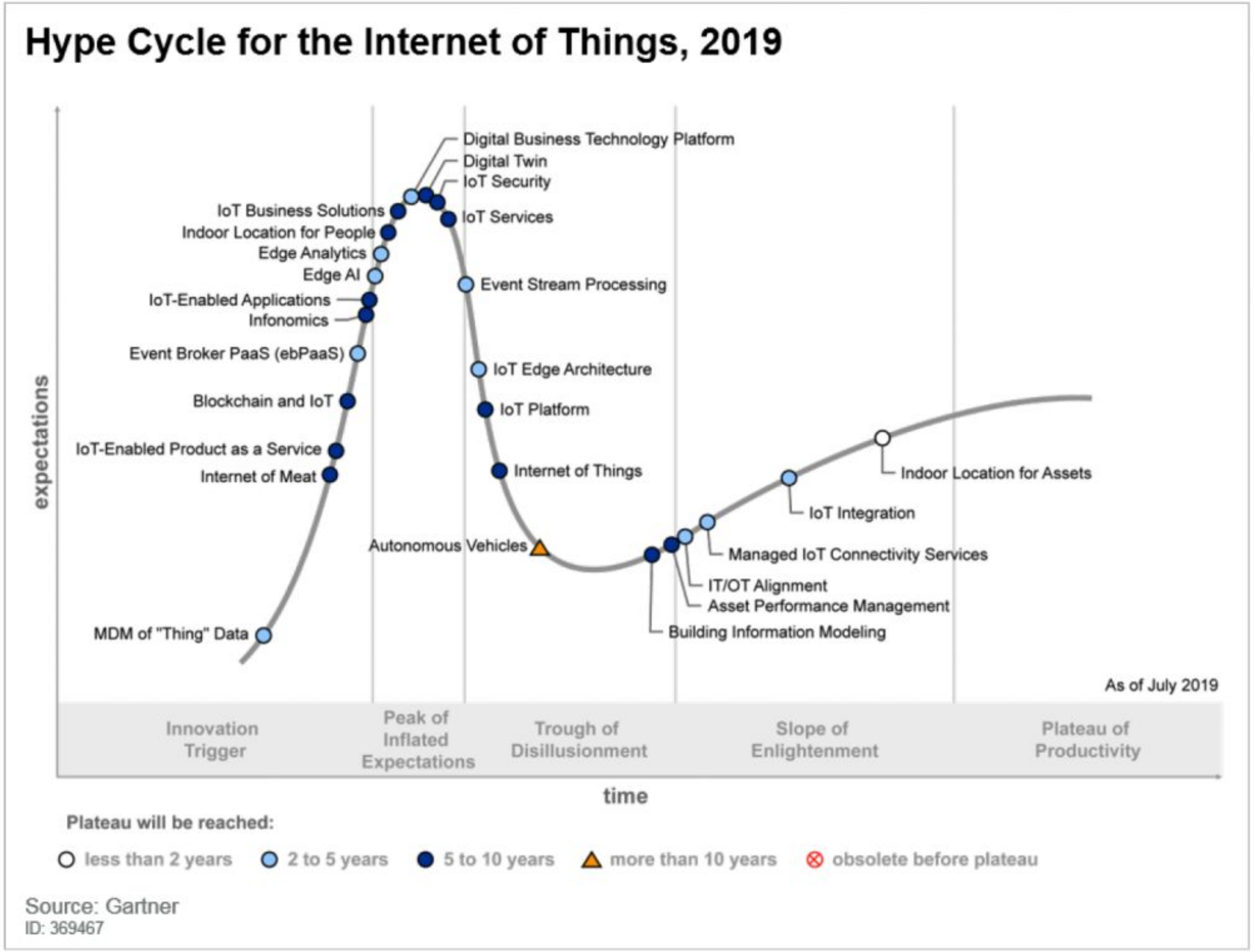
Massive Data

Decisions/Actions

Another Viewpoint



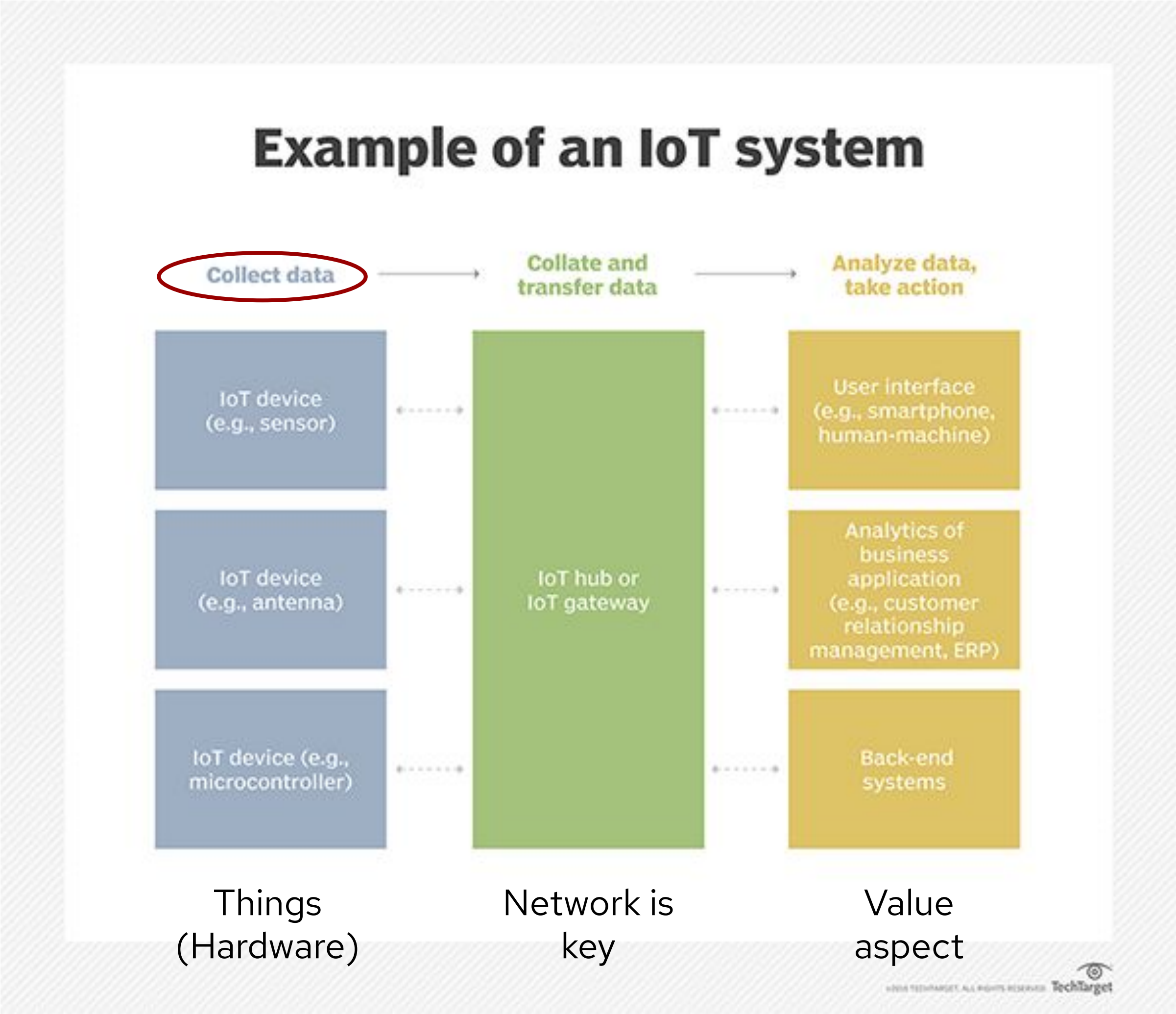
Sobering Aspect – Gartner Hype Cycle



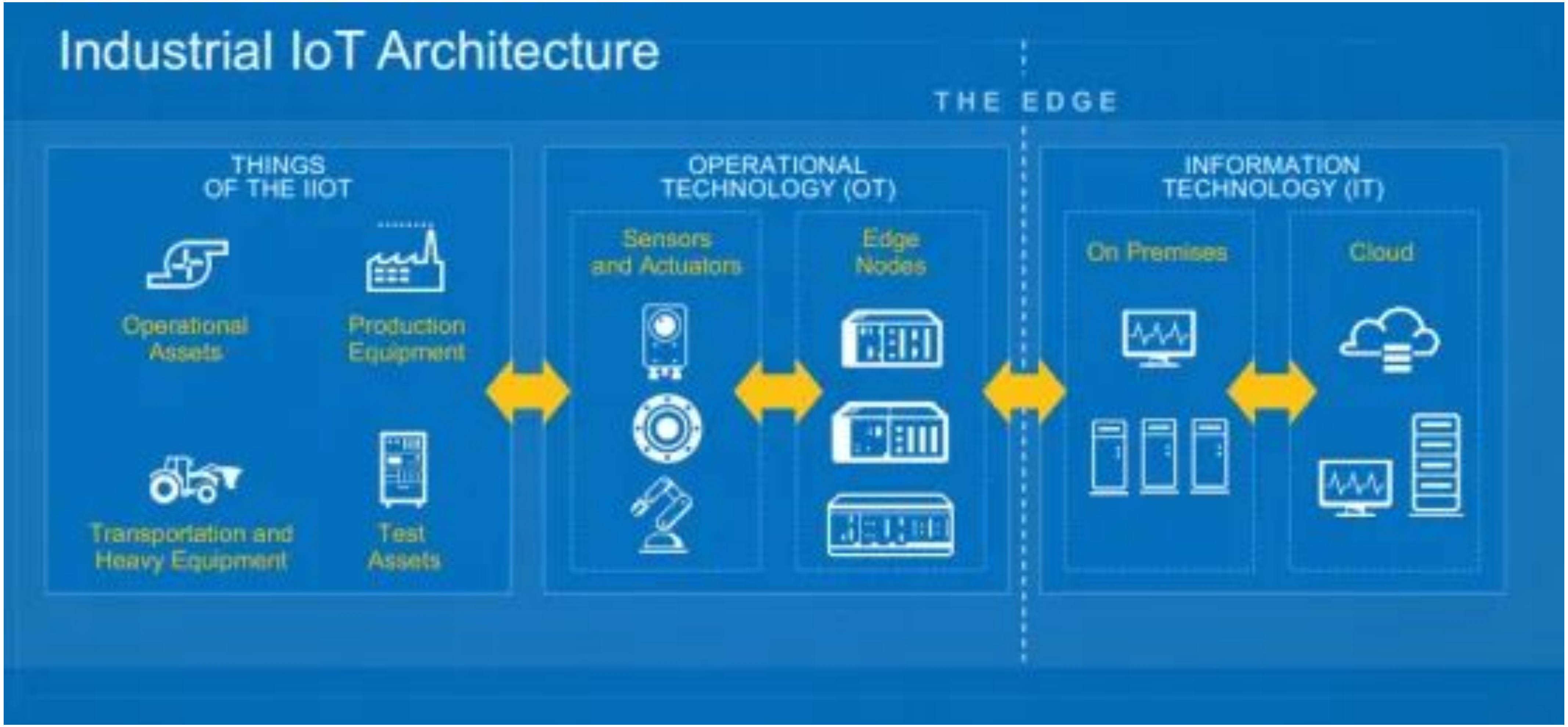
Architecture and Examples

Let’s set some discussion baselines...

A Coarse Architectural Model



IIoT Scale Case

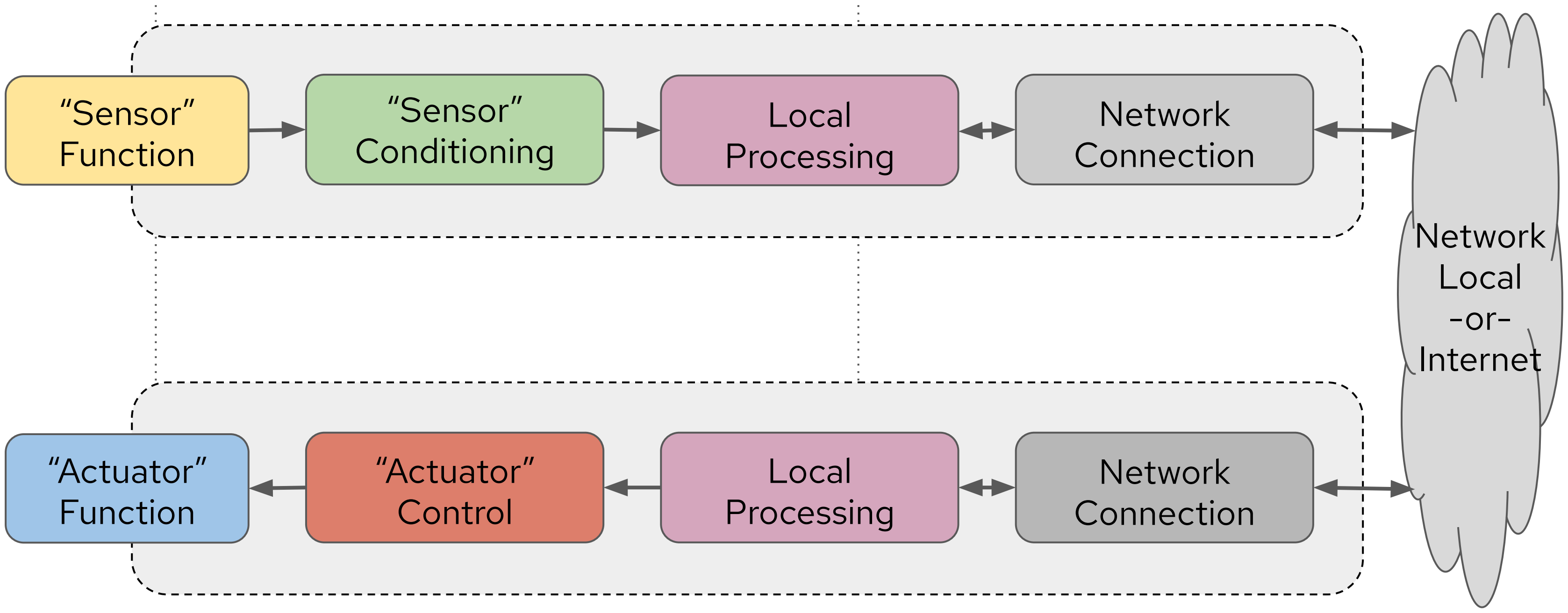


What Do These So-Called Things Actually Do?

Effectively perform one (or both) of two basic functions (always exceptions)

Sensing
Data gathering

Actuation
Actuation and control
Other “actions”
Not always physical



Discrete/Separate (Raw) IoT Functions

What Do These So-Called Things Actually Do?

Effectively perform one (or both) of two basic functions (always exceptions)

Sensing

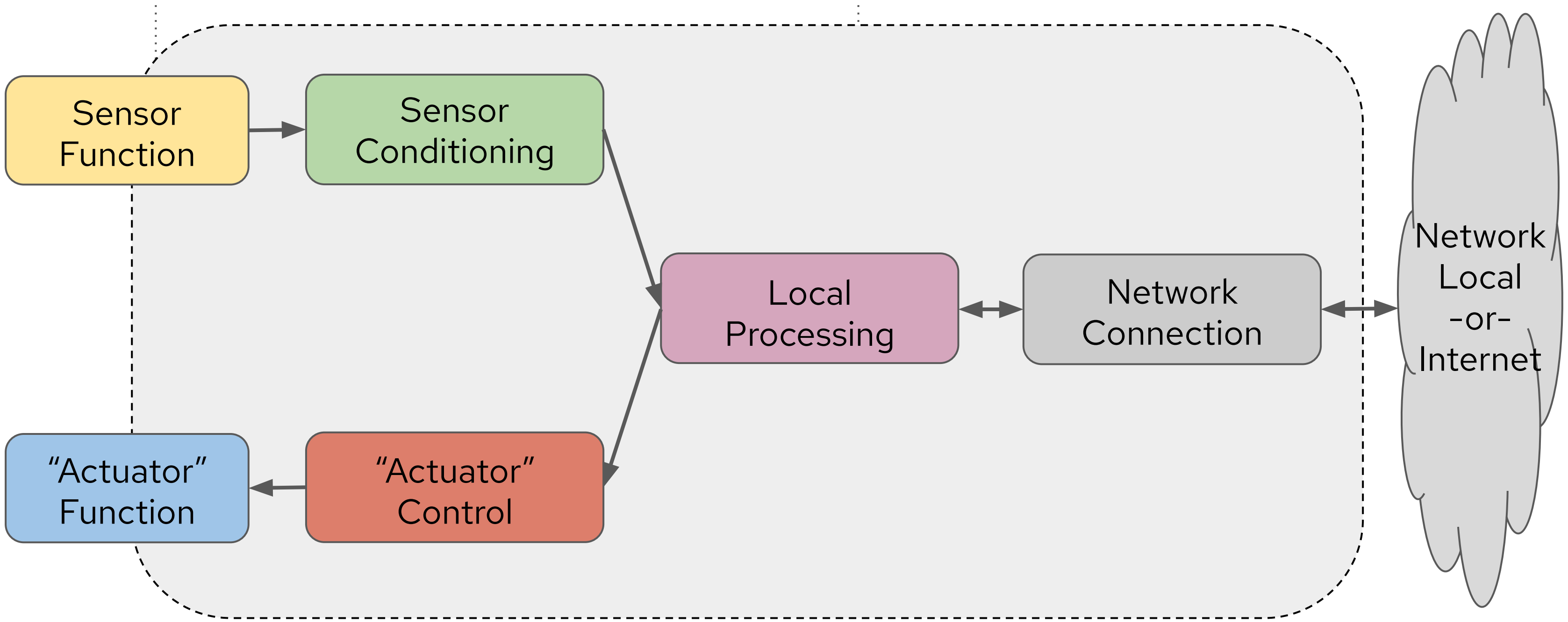
Data gathering

Actuation

Actuation and control

Other "actions"

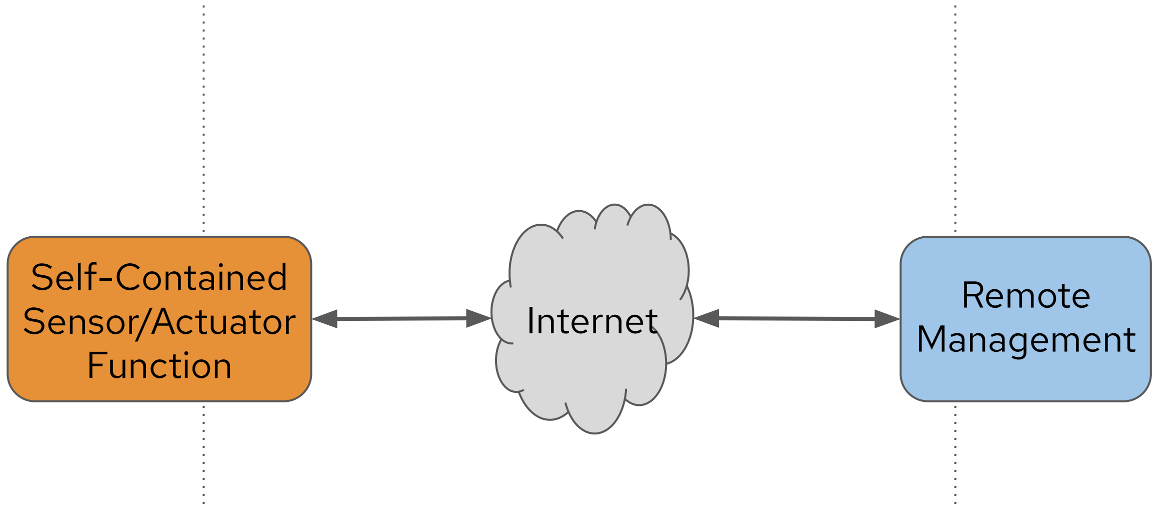
Not always physical



Combined/Self-Contained (Embedded) IoT Function

Scale Impact

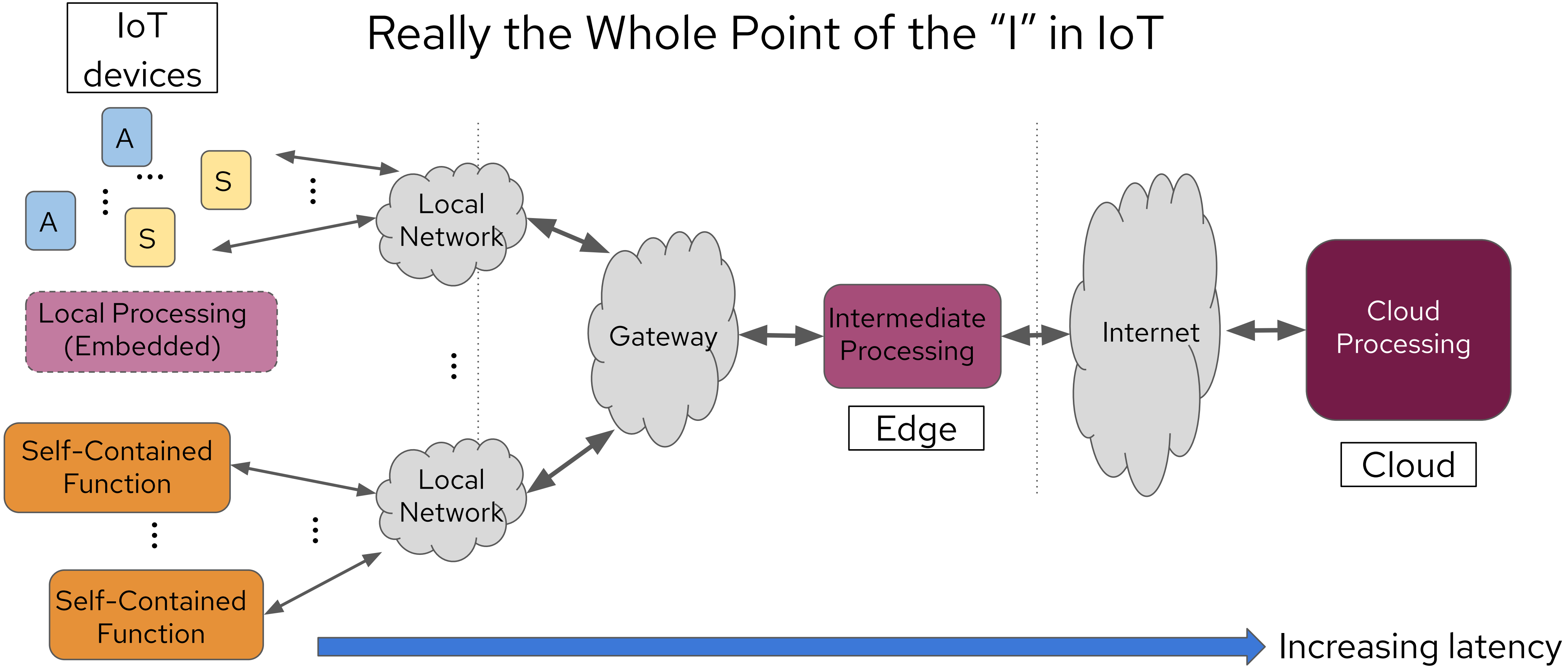
Single Function



Terrarium Example

Scale Impact

Really the Whole Point of the “I” in IoT



City Scale Case

Component and System Design Considerations

Things are part of systems...

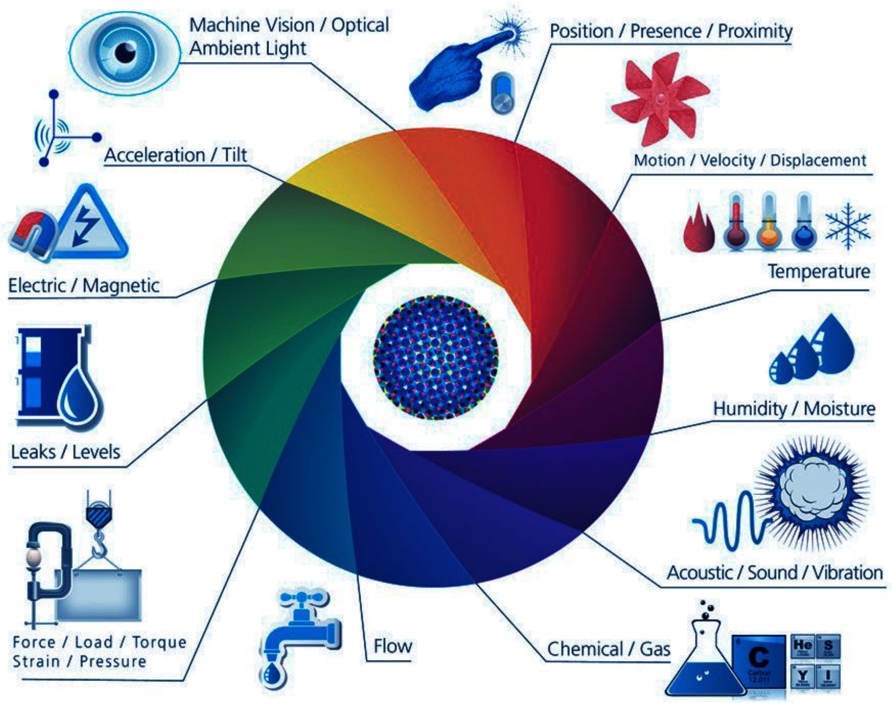
Each w/ unique requirements...

Which call for interesting and sometimes unique implementations.

IoT components represent part of larger scaled solutions to new systems problems and challenges.

Sensors

Dictionary definition:
noun
A device that detects or measures a physical property…



Expanded definition (mine):
A device or process that detects/collects/digitizes activity…

… almost infinitely expands the concept of a "sensor"

Rich sensor set with very powerful local processing

NAV/AV greater need for local processing and storage

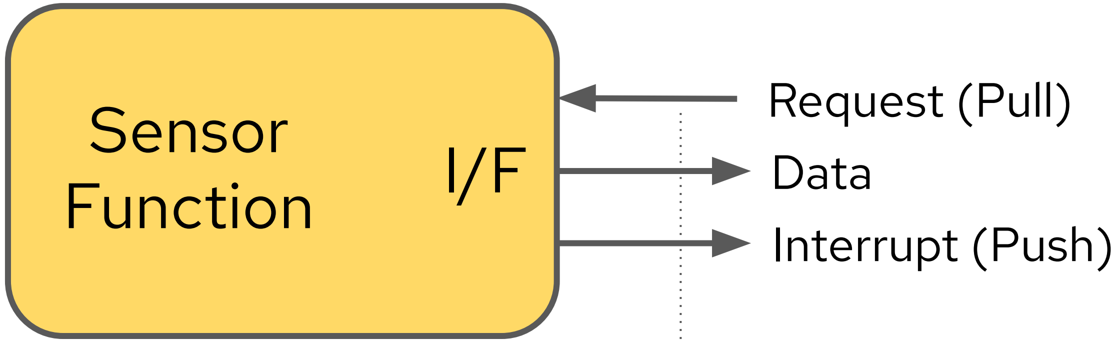
Red Hat

Sensor Functional Taxonomy (Limited)

Function	Output	Example
Threshold	Binary (Y/N)	Threshold crossed, switch activated, ...
Single Value	Sampled Digitized Analogy	Temperature, position, ...
Stream	Time-Based Sampled Values	Continuous position, temperature, ...
Complex	Digitized Capture	Photo storage-analysis (A/I) Collect actions such as search activity
...	...	...

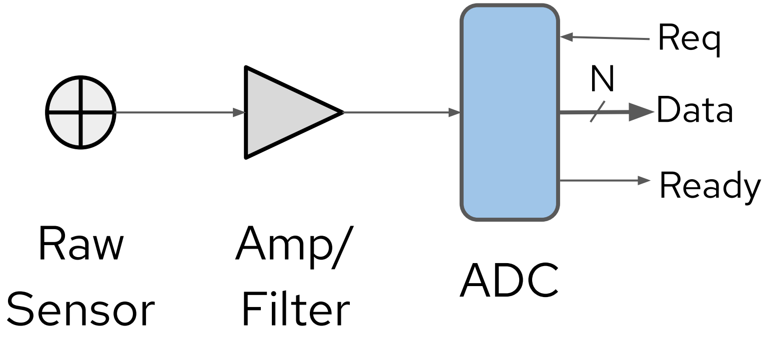
Red Hat

Physical Sensor Aspects



What besides a raw sensor could be in this box?

Conceptual basic example:



Access Behavioral Model:

- Pull (polling)
- Push (event interrupt)

Interface:

- Parallel and serial
- Access timing

Quality Issues/Concerns:

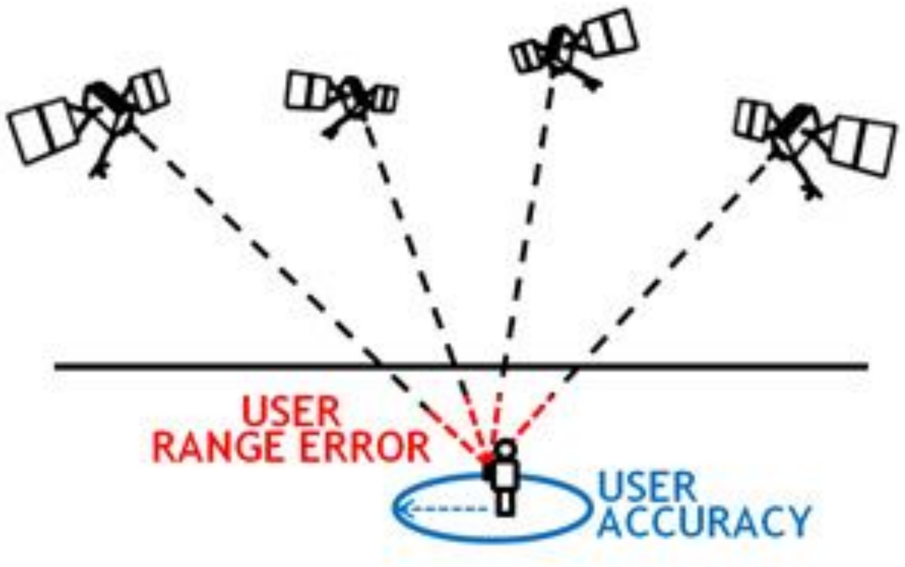
- Noise
- Accuracy and drift
- Sampling rate
- Resolution

Physical Sensor Conditioning

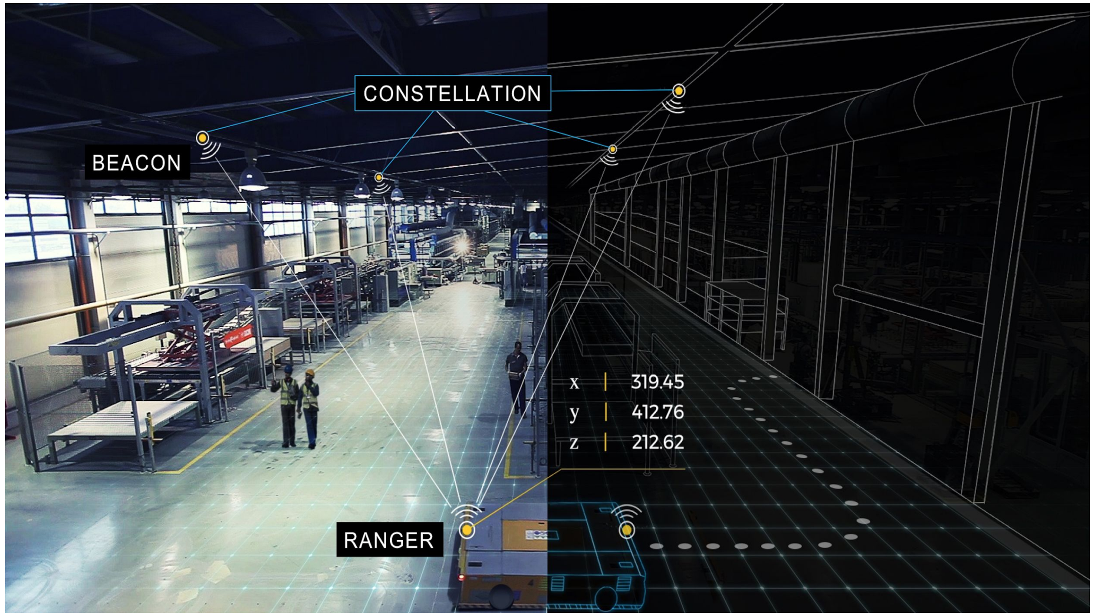
- Why spend a lot of time on sensors?
  - Data quality - Garbage in/Garbage out
  - How and where to avoid/mitigate garbage?
- Sensor itself may be a very raw signal, e.g. ...
  - Voltage or current analog representing a sensed value
  - Requires filtering (noise) and compensation (drift)
  - Requires digitization (ADC)
- Particular considerations
  - Accuracy/Resolution/Sampling rate/Other signal processing

Accuracy

Depends on What's Required



GPS lower accuracy triangulation
Accuracy in meters
iPhone uses GPS (and other sources)
...
Good enough for car mapping apps

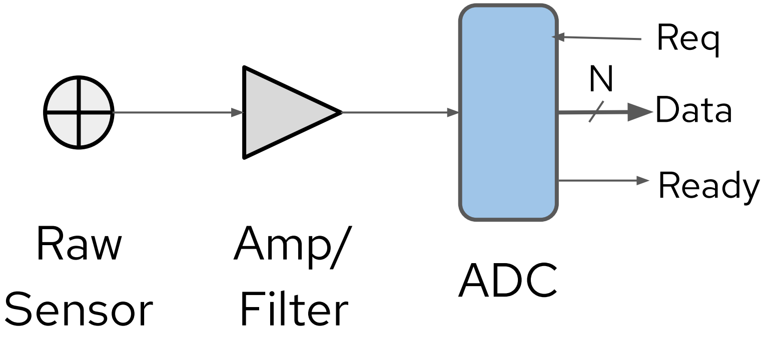


Humatics™ higher accuracy triangulation
Accuracy in cm to mm
https://humatics.com/
...
Pick-place robots need finer grain accuracy

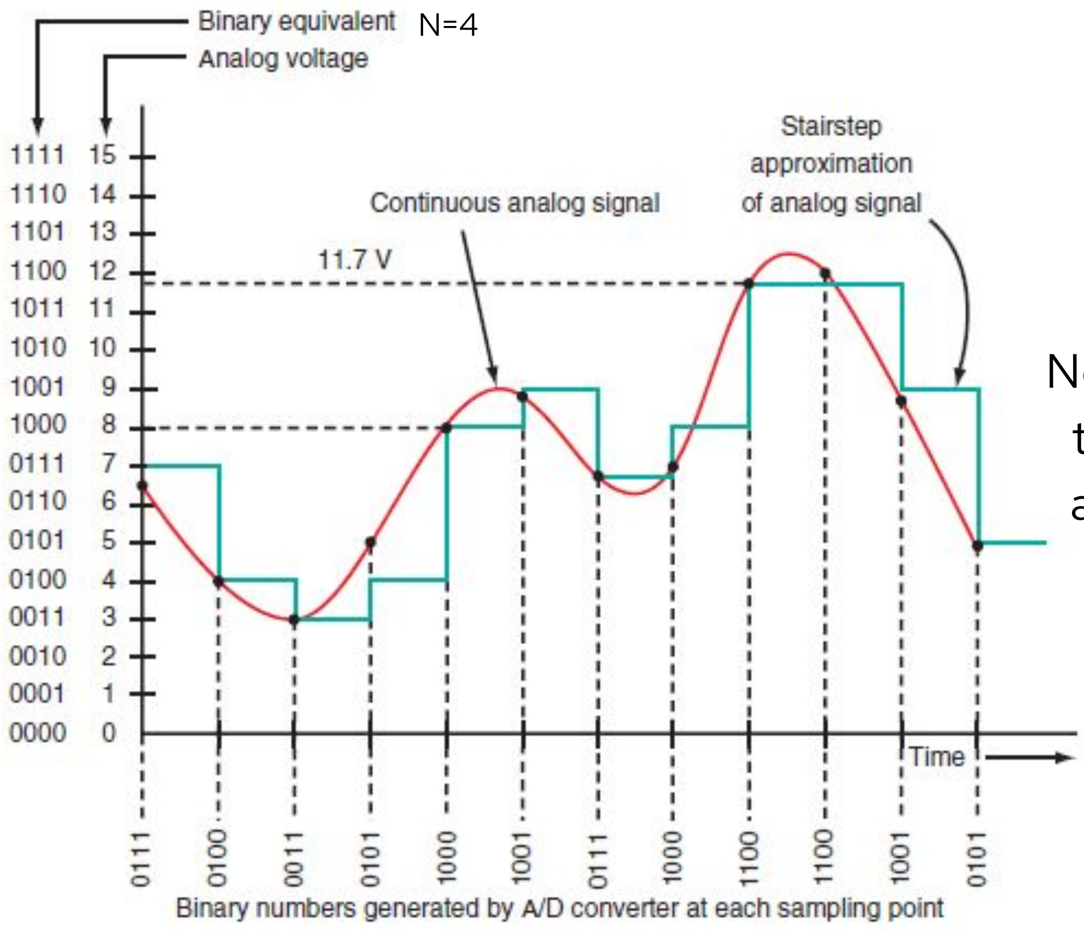
Red Hat

Resolution

Is the Sensed Signal Properly Digitized?



Select the correct value for N

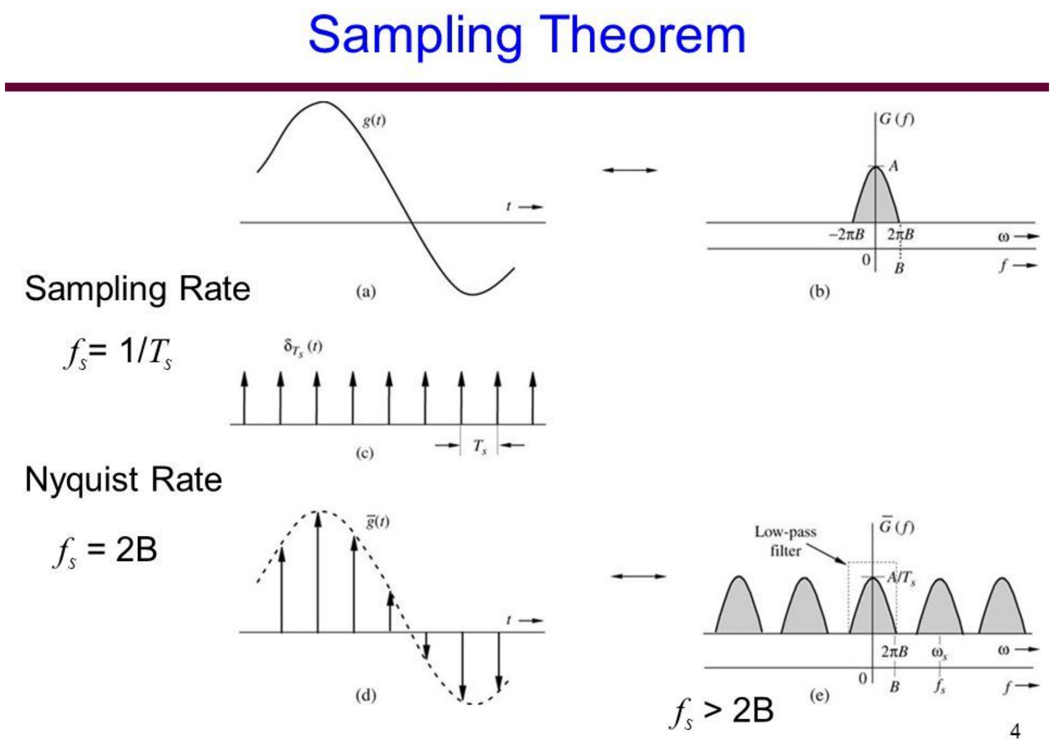


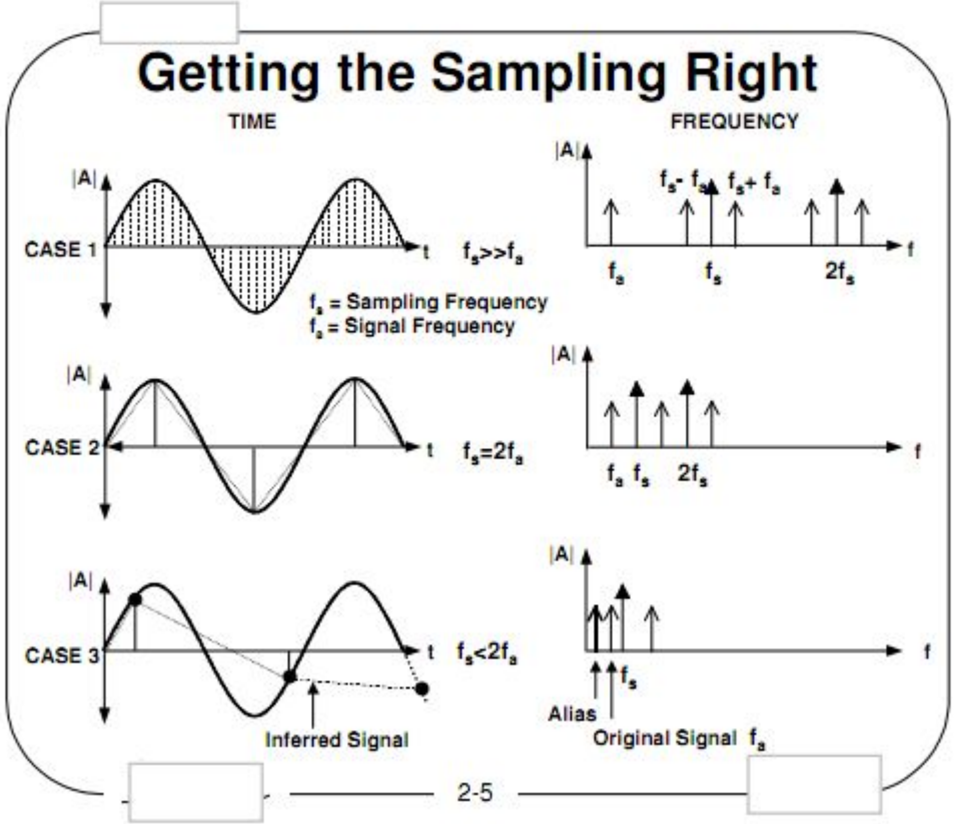
Note the impact of both the quantization error and effective time lag

Red Hat

Sampling/Nyquist Rate

Avoiding aliasing: Sample rate > 2x of the sensed signal bandwidth





Simple Signal Processing Example

Ex: Need to Calculate Energy from a Power Sensor

Raw sensor data is power but need is to provide total energy over an extended time period T

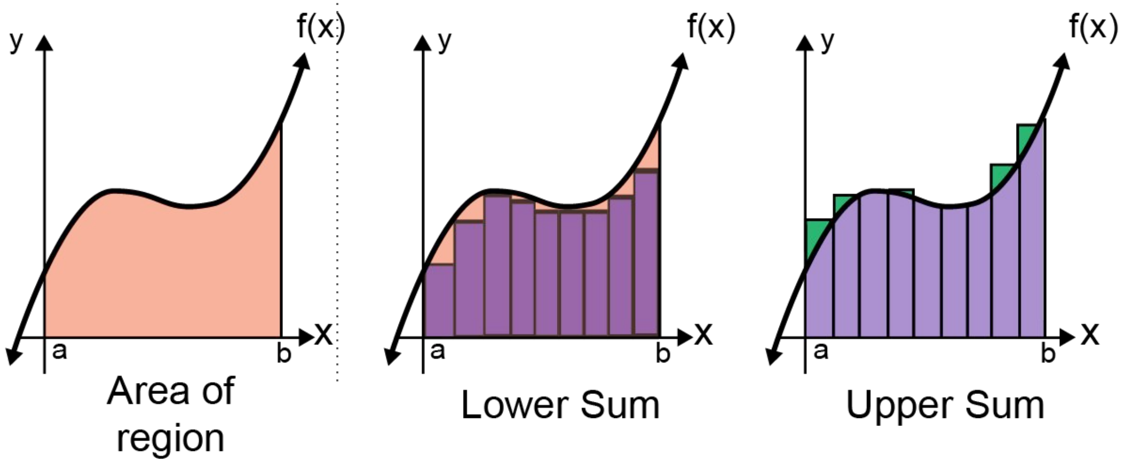
Facilitated by sensor and local processing

Processor utilizes appropriate constant sample period T_S and quantization resolution

Simple Riemann integral

$$T_S * \sum_{k=1}^{N} P(k)$$

Where total time of interest = $N*T_S$

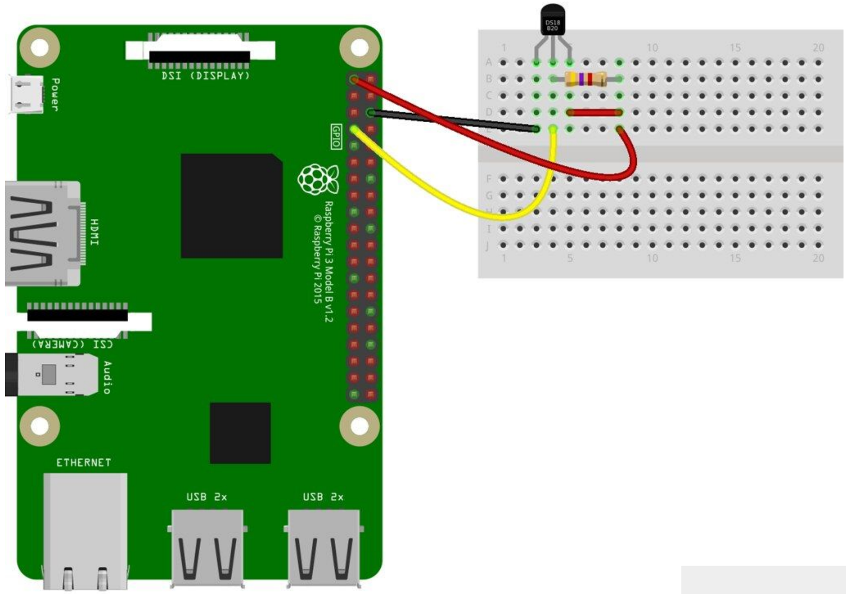


Let x represent time t

Let $f(x)$ represent $P(t)$

Sensor Examples

Demonstrating a Range of Possibilities and Functional Integration



Simple:
Capture thermal
sensor readings with
just a few commands

Complex:
Capture a 4M pixel
image also with just
a few commands

Red Hat

Actuators

Dictionary definition:
Noun
A device that causes a machine or device to operate.

Discrete Devices

Subsystems

Expanded definition (mine):
… a device or process that commands a physical, virtual, or other intangible output/action

… including but not limited to photo display, recommendation, …

Sense

Process

Recommendation:
"Get some exercise"

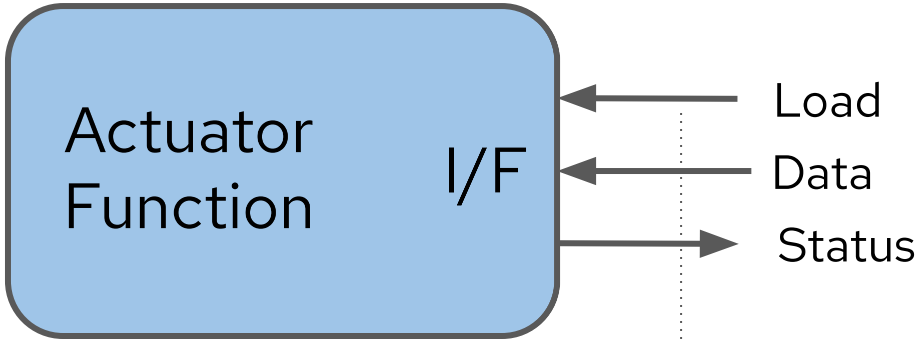
"Actuate"

Red Hat

Actuator Functional Taxonomy (Limited)

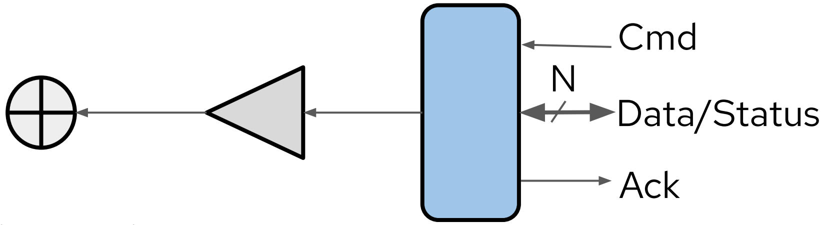
Function	Command	Comment/Example
Switch	Binary (On/Off)	Turn a LED on or off, ...
Simple Movement	Binary Actuation	Activate a solenoid e.g. a lock arm, ...
Complex Movement	Position/Velocity/...	Command a simple servo motor Command an entire robot operation
Recommend	Statement	Provide a course of action, ...
...	...	...

Physical Actuator Aspects



What besides an actuator could be in this box?

Conceptual basic example:



Access Behavioral Model:

- Command
- Response

Interface:

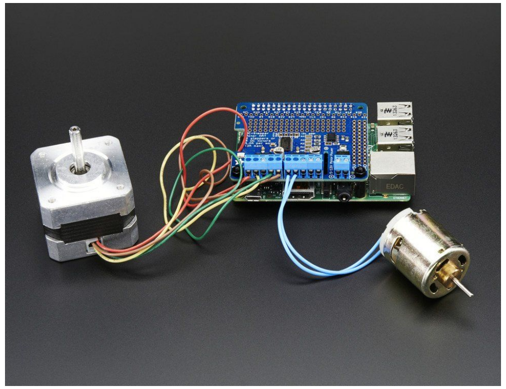
- Parallel and serial (e.g. PWM)
- Access timing

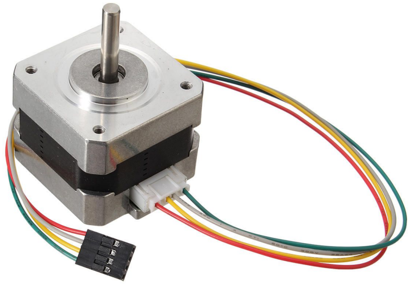
Issues/Concerns:

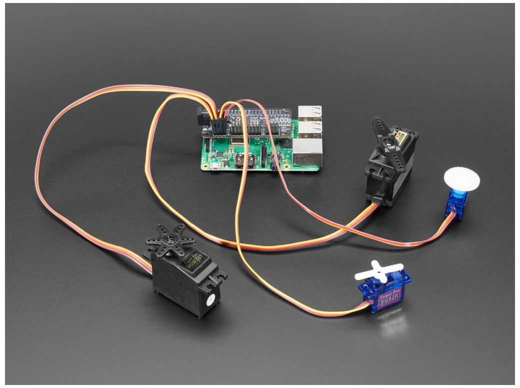
- Drivers - Power requirements
- Load and response
- Completion acknowledgement
- Failure protection/reporting

Red Hat

Mechanical Examples







Permanent Magnet Stepper Motor:

- Open loop position control
    - Send phased pulses for discrete steps
    - Maintains position after pulses stop
    - Inherently stable (mostly)

DC Motor:

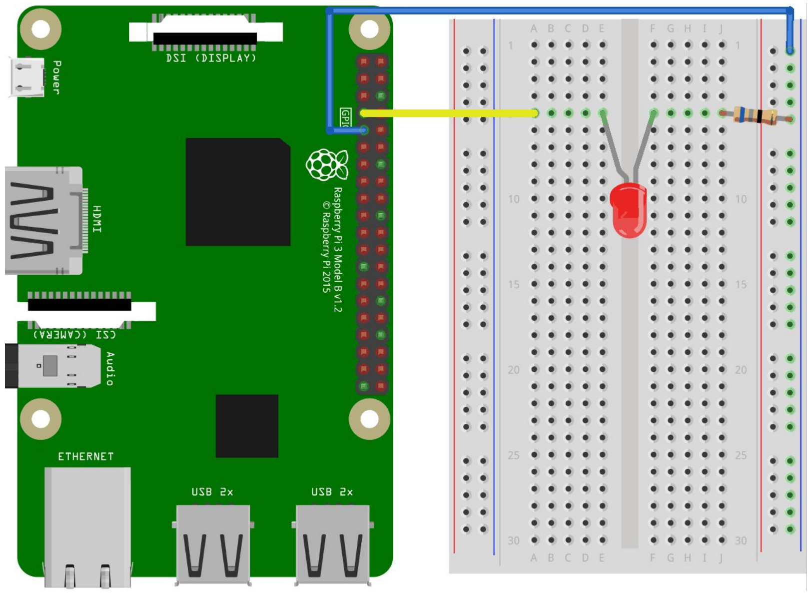
- Open loop velocity control
    - Provide voltage representing desired speed
    - Inherently stable (back EMF)

“Servo” Motor:

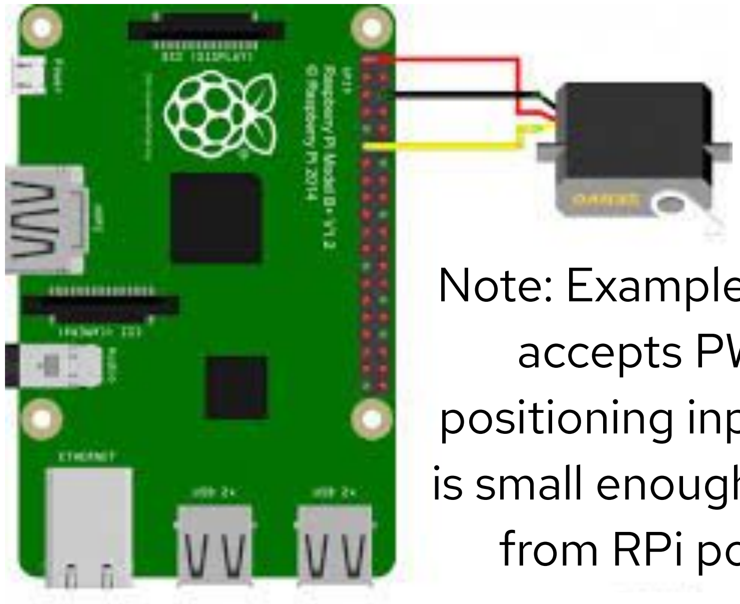
- Open loop position control (apparently)
    - Provide signal representing desired position
    - Stable (internally controlled so you don’t have to)
    - Ex SG90 servo with PWM input – Duty cycle represents desired rotational position

Red Hat

Simple Actuator Examples



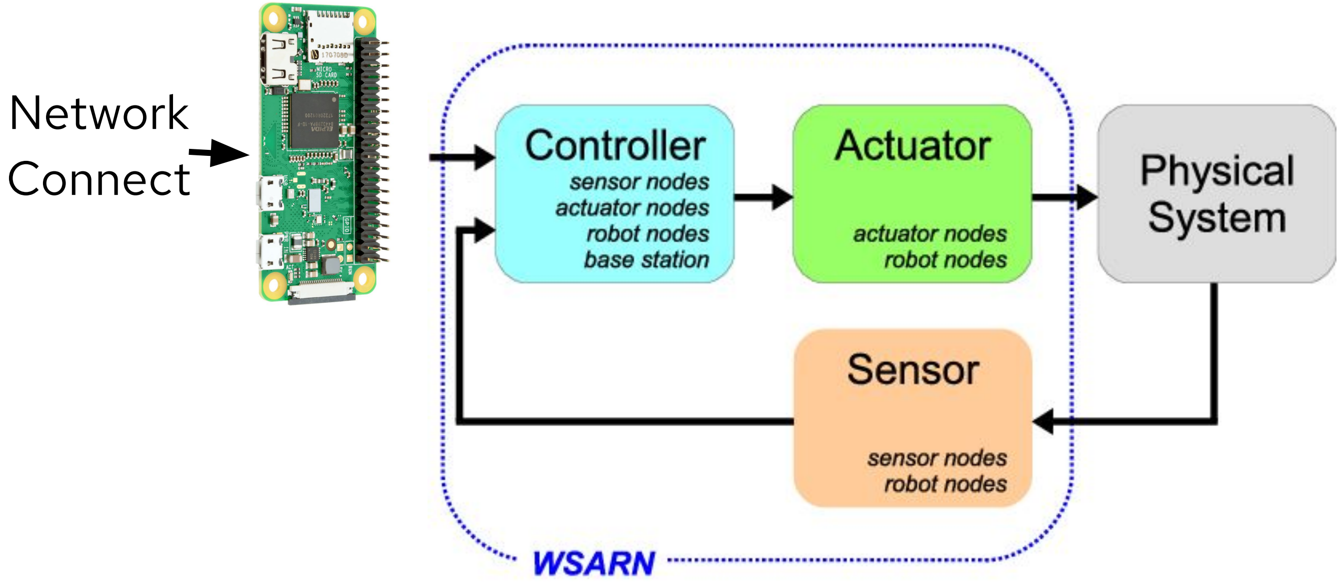
Simple:
"Actuate" a LED with
just a few commands



Note: Example motor accepts PWM positioning input and is small enough to run from RPi power

Also simple:
Position a servo motor
with just a few commands

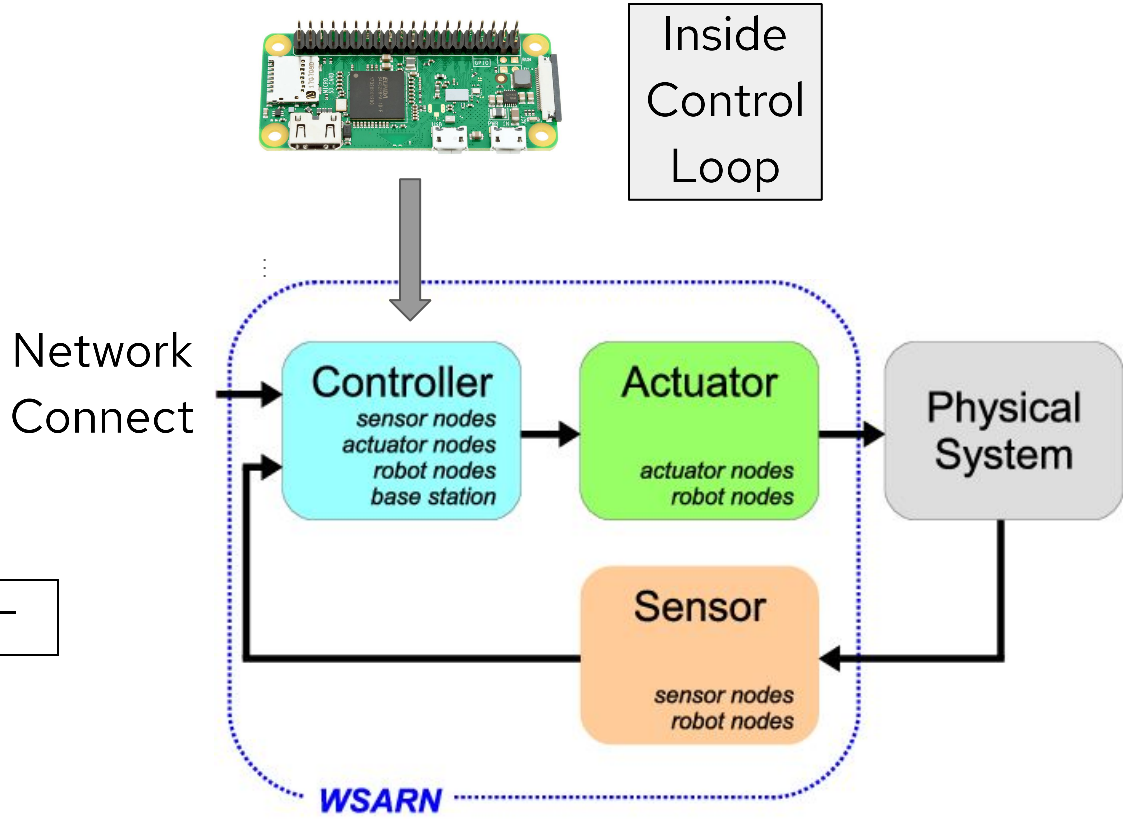
More Complex Example



Outside Control Loop

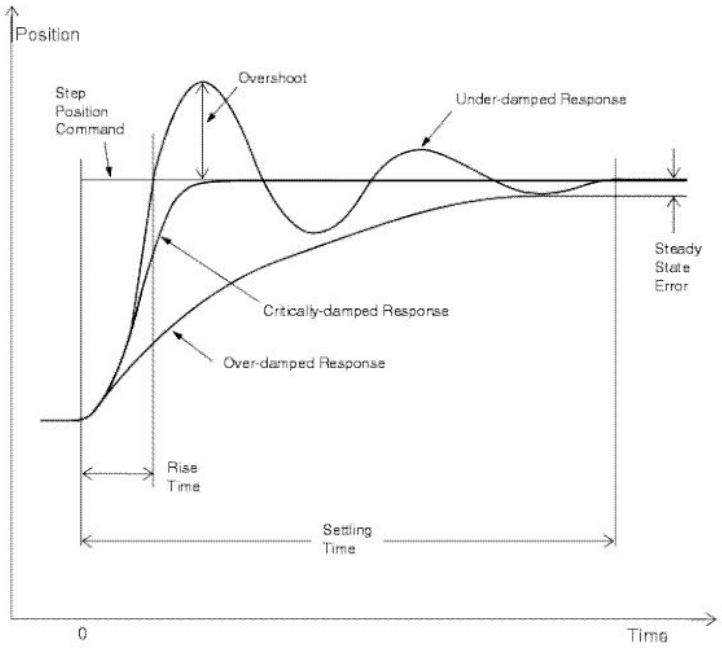
-OR-

Inside Control Loop



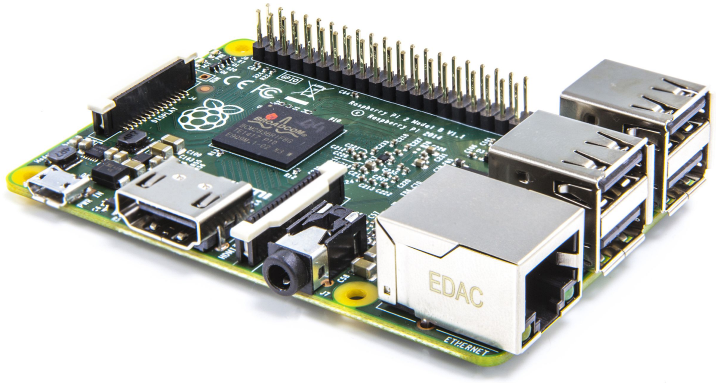
Other Actuator Considerations

- Concept is simple enough…
  - Command an action
  - Expect particular response
- Reality
  - Command completed successfully?
  - Failure detection?
- Functional requirements/specifications
  - Load, power, environment, maintenance/upgrade
  - Ex: Response time (stability), accuracy/stiffness, …



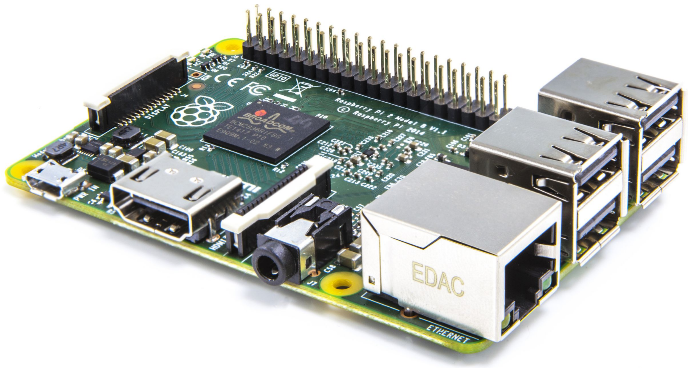
Processing

- Wide spectrum of use cases - Innovation opening
    - From simple sensing/actuation to AI/ML
- Focus on local/embedded for this discussion
    - Could be any processor type
    - ARM, X86, AMD, RISC-V, GPU, ...



- RPi, Arduino popular because
    - Simple, very low cost, and decent I/O - Essentially a server
    - Wide ecosystem - HATs, pHATs, sensors, motors, cameras, ...
    - But is it really field/enterprise ready?

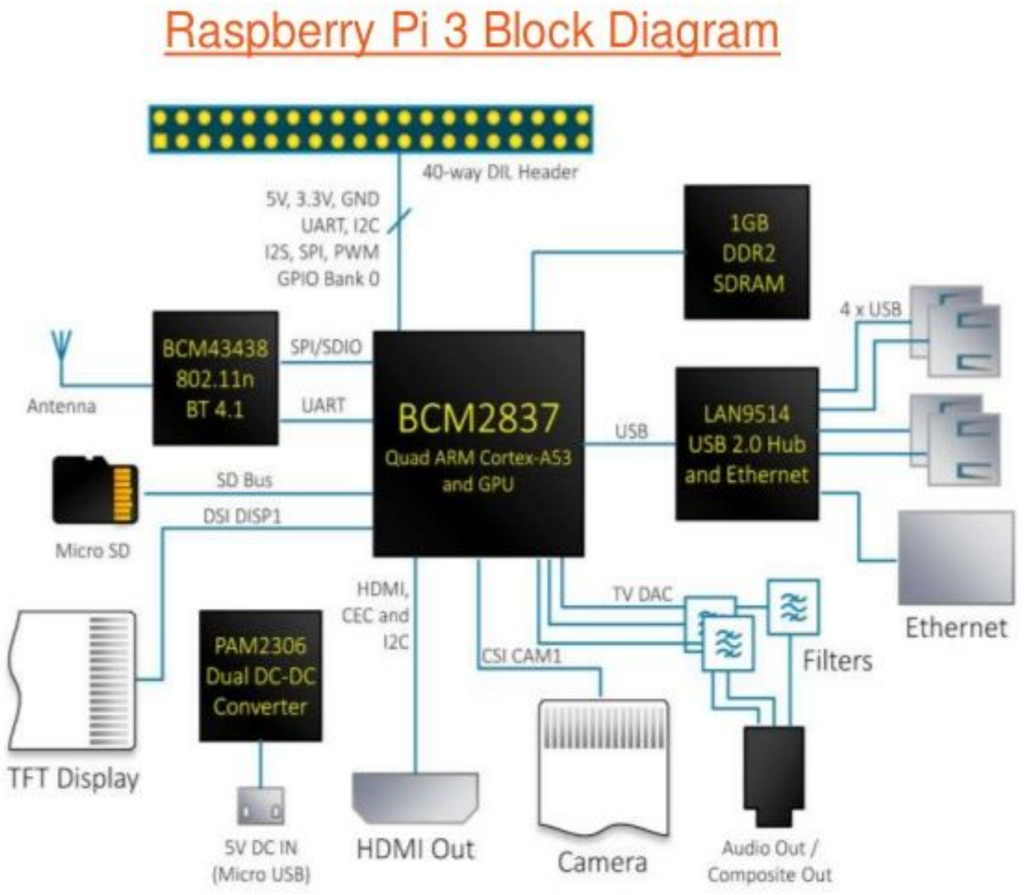
Processing



- Network connectivity and messaging
  - IP based - MQTT support
- Time-sensitive functional issues --> Real Time OS requirement
  - OS context switching can result in RT process interference
  - Real time value is in outcome predictability vs throughput
  - Otherwise time-sensitive functionality may require offloading
  - Event and time-sharing RTOS's - E.g. ARM MBED (embedded OS)

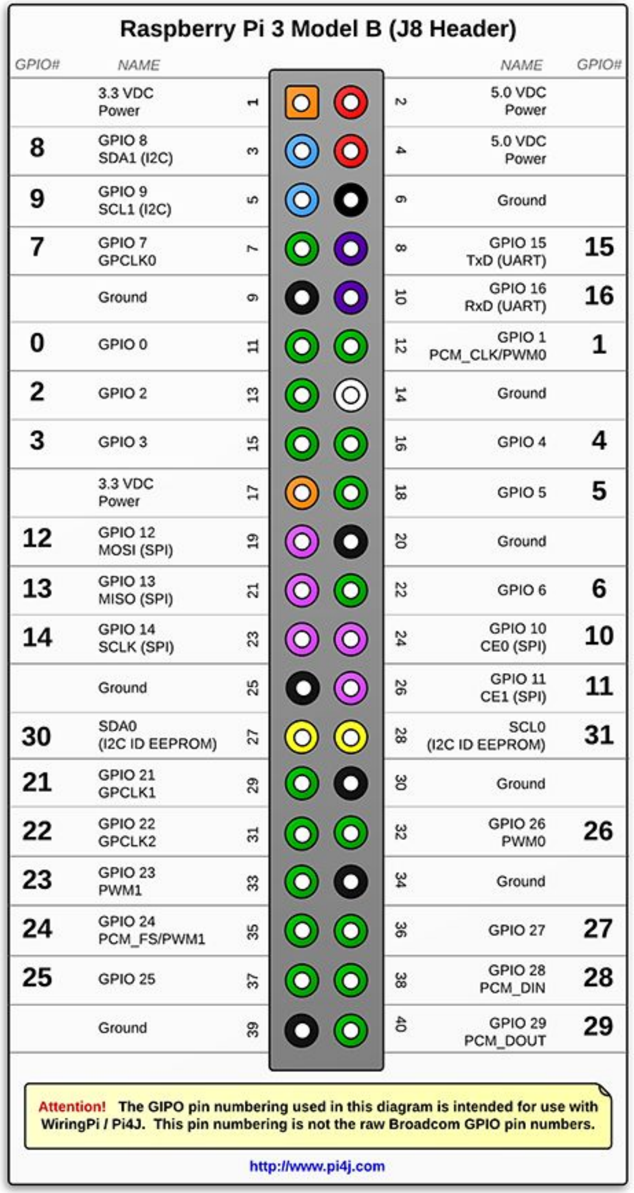
Raspberry Pi Local Processing Example

- Effectively a server
  - Highly functional
  - Flash storage
  - Relatively rich I/O
- Add GPIO HATs, pHATs
  - Power considerations
- Nice programming models



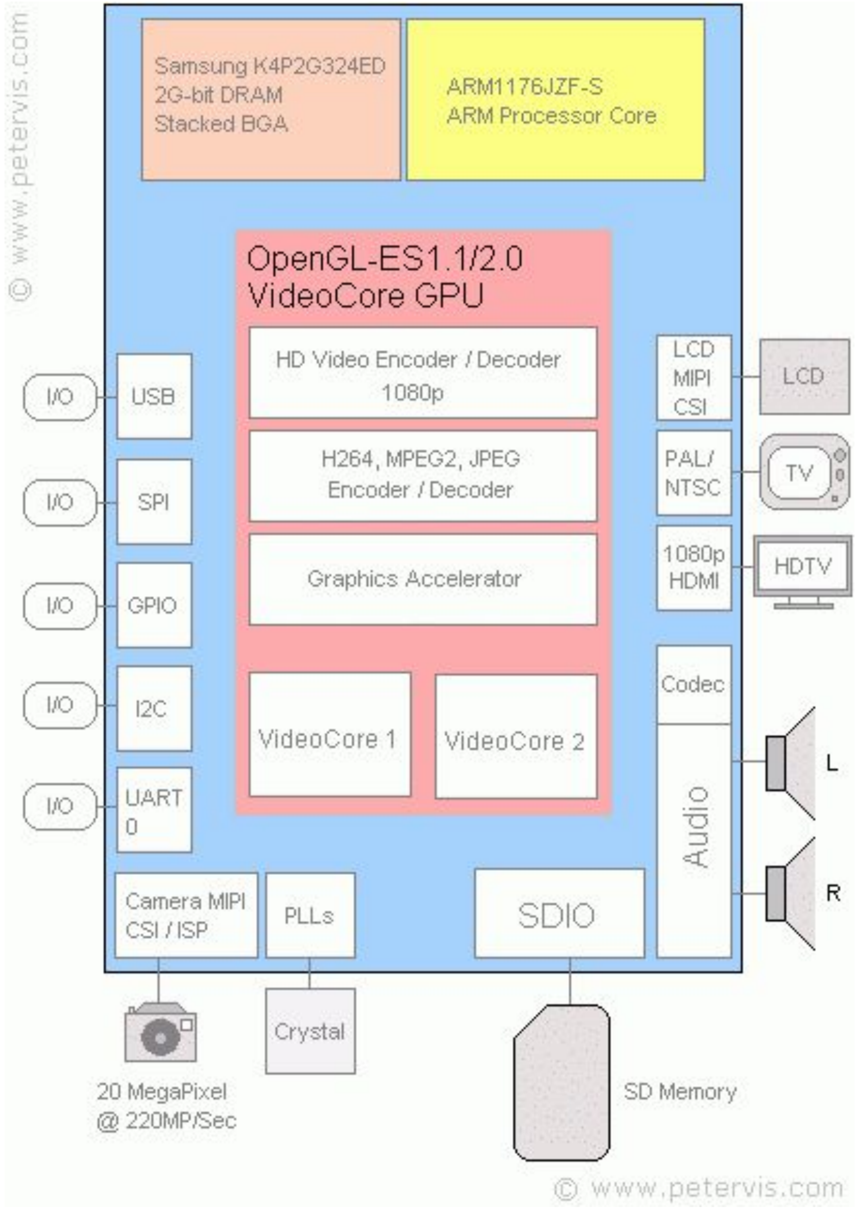
Raspberry Pi Example

- GPIO Header
- Wealth of possibilities
  - Logical control
  - PWM
  - Serial interfaces
  - ...
- HAT connection point



BCM 2837

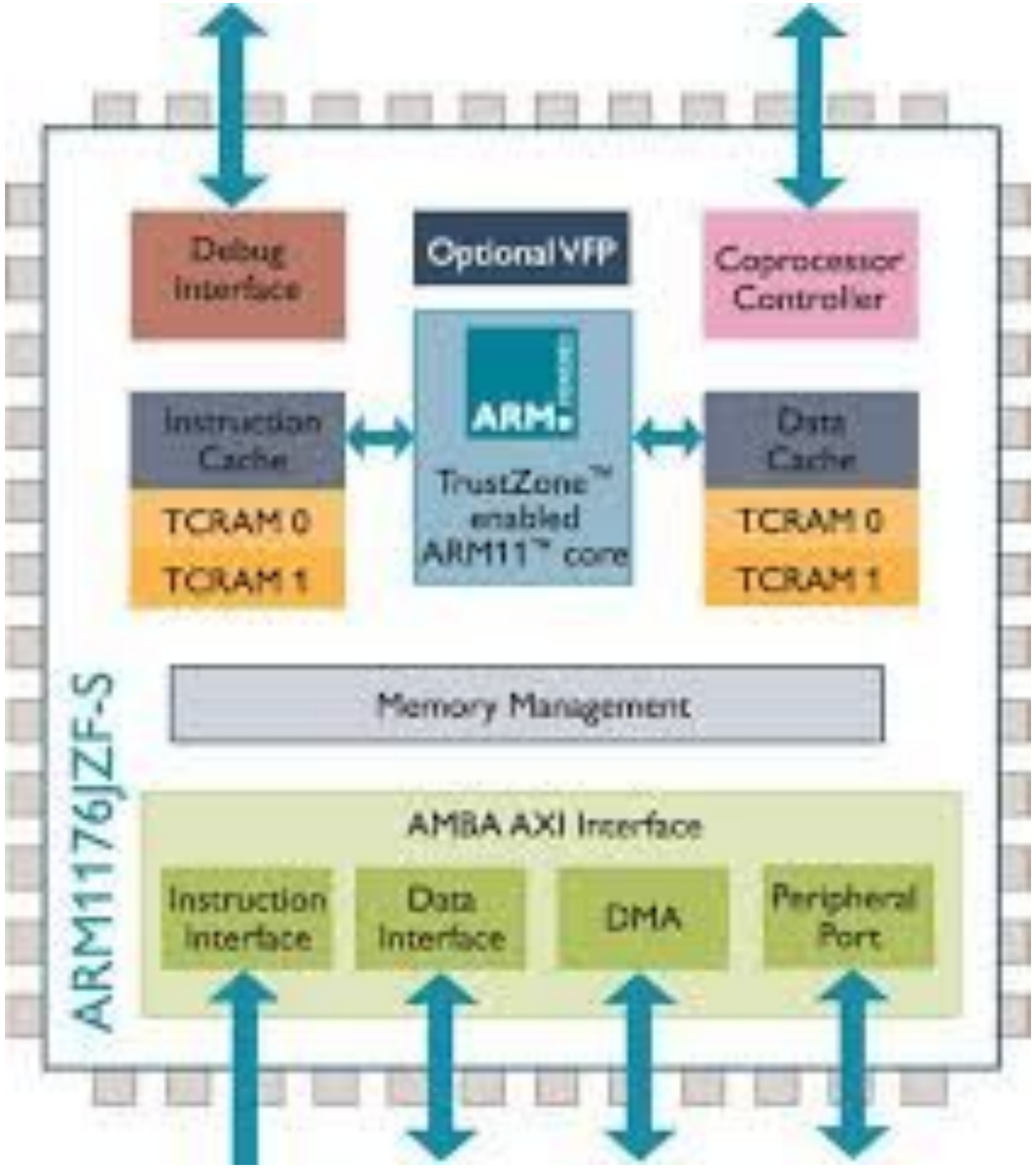
- SoC - High integration
  - Licensed ARM cores
  - GPU
  - I/O
- Increasing functionality
- Plummeting cost



Red Hat

ARM Core

- Ubiquitous!
  - Major mobile share
- Multiple IP licensing options
- RISC architecture
  - vs x86 CISC
- Ecosystem



A Few Network Connection Items

- Wired - Copper and optical
  - Ethernet, USB - Ubiquitous
  - Power delivery options - PoE, USB
  - Many connector types...
- Wireless in particular - Various rates, ranges, and power
  - LR-WPAN: Class of low rate, low power, short range/mesh connect
    - Zigbee, Zwave, BLE, 6LoWPAN (IPv6 - nearly infinite # IP addrs)
  - LPWAN: Class of low power, low rate, wider range
    - NB-IoT, LoRaWAN, Sigfox

A Few Network Connection Items

- Wireless in particular - Various rates, ranges, and power
  - Wifi
  - 3 major 5G use cases...
    - uRLLC: Ultra-reliable and low latency comms (NAV/AVs)
    - mMTC: Massive machine type communications (IoT)
    - eMBB: Enhance mobile broadband (consumer)
- Wireless popular/required for many reasons
- Messaging support - E.g. MQTT

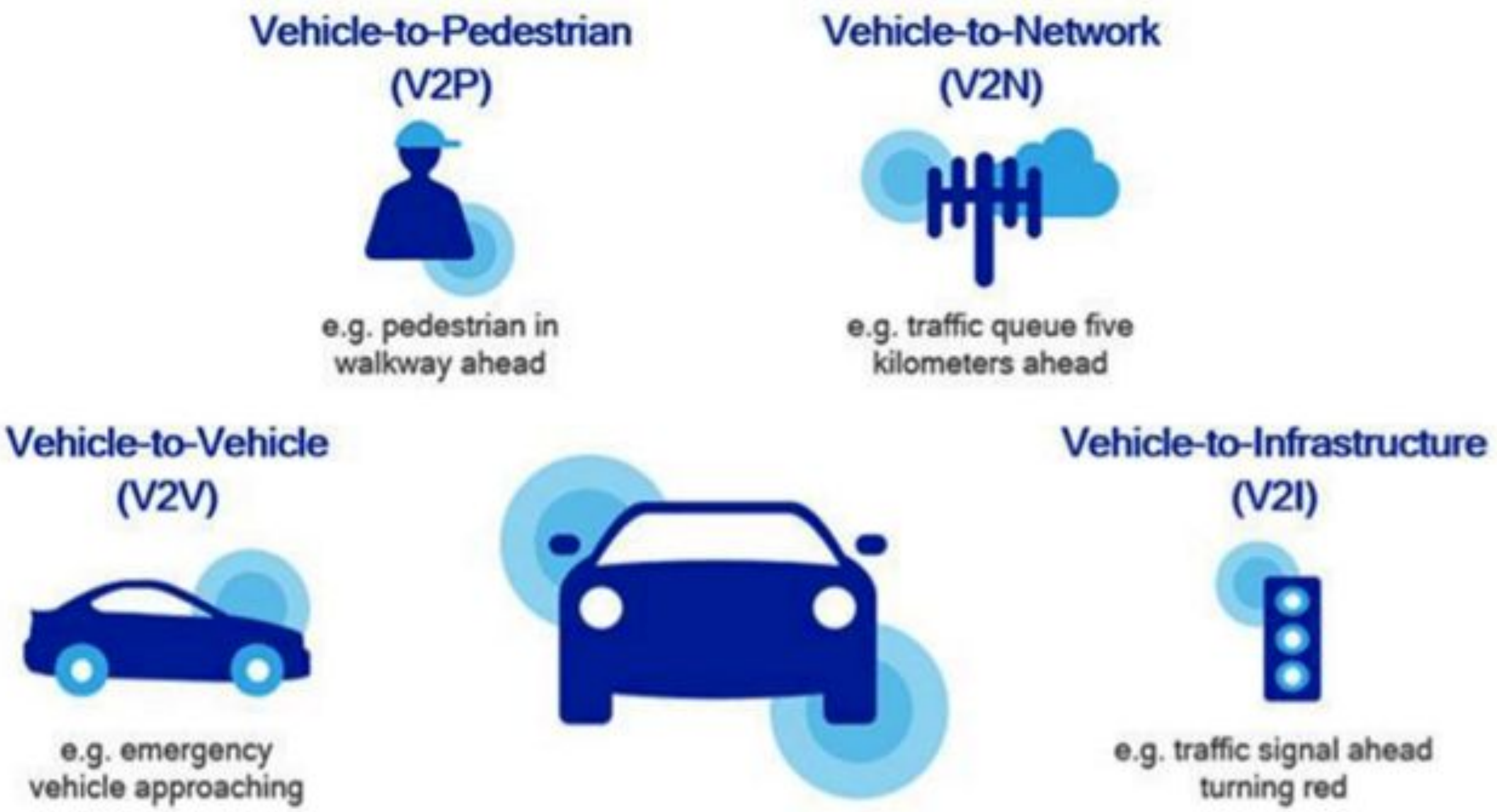
Red Hat

Network Connection Considerations

- Rate vs range vs power vs cost (device and network transmission)
  - Not all IoT devices require high/continuous transmission rates
  - Wireless radio can dominate IoT power - Use sleep modes
  - Efficient data transfer - Compression (lossy & lossless)
- Environment - Benign/harsh, stationary/moving, shock/vibration, ...
  - Reliability - Another reason wireless is popular vs conns/cables
- Latency - Keeps coming up - Determines processing/storage locations
  - 4G can have an average 50ms latency
  - 5G average latency is < 10ms and can be as low as 1ms

Red Hat

V2X - Vehicle to Everything Links



V2X has a 4ms response time and longer transmission distance requirements → 5G required

System Definition - IoT Perspective

- A combination of one or more IoT (embedded) functions providing particular services and/or outcomes
  - Collective (interconnected) operation - Internet of things (plural)
  - Embedded components designed/optimized for particular tasks
  - Potential for massive scale and/or distribution
  - System determines overall outcome/value

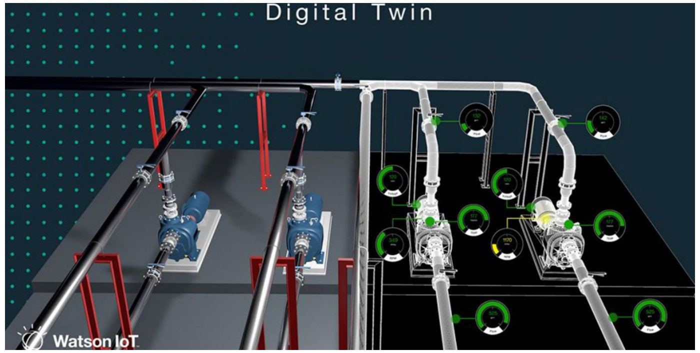
System Considerations - A Sampling

- What is the desired outcome/service?
  - High level business, government, consumer, etc. need
  - Real time requirements - Response time, accuracy, ...
  - Costs and ROI - CAPEX and particularly OPEX
- What architecture provides the outcome/service? HW aspects...
  - Particular sensor and actuator requirements
  - Processing/storage - Local (IoT/edge) or remote (upstream cloud)
    - What are data persistence requirements, caching, etc.?
  - Network topology - Scale/distribution/range/power/security

Red Hat

System Considerations - A Sampling

- Environment/location - Especially impacts HW/network choices
  - Benign or noisy/hazardous - Operational specs, cooling, S/V, ...
  - Physical accessibility - Scattered, remote, power availability
- Management - Operation/health/service (repair, upgrade, scaling)
  - Service cost - Accessibility - Predictive failure/scheduled service
  - Remote FW upgrade on 1000's of devices - Non-disruptively
  - Reliability - Pets and cattle
  - Security - Access, protection
  - Digital Twin concept



Red Hat

A Note on Efficiency

- Efficiency and effectiveness
  - A simple collection of individually *efficient* components...
  - Does not necessarily provide a better outcome than...
  - The overall *effective use* of a collection of components
- Focus at the system outcome level
  - Doesn't imply that components don't themselves need efficiency
  - Just states best effectiveness is achieved at the system level

A Note on AI/ML

- In the IoT context there are arguably two implementations...
  a. IoT system as a whole provides data for business-oriented learning/inference engine applications
     - E.g. drives enhanced/new market opportunities
  b. IoT system uses training/inference engines to enhance/improve infrastructure operational effectiveness
     - E.g. IIoT data used to improve factory operations/service

Red Hat

Breakout Session

Design

exercise.

System Design Problem

- Split into 4 groups each with their own design problem
  - City traffic management
  - Building environmental operations
  - Manufacturing floor operations
  - Remote weather station

System Design Problem

- Collect HW and system design selections per requirements
  - Sensor, actuator, processing, networking function selections
  - Functional, reliability, and/or environmental considerations
  - 15 min allotted
- Regroup and share results - What/why/how?
  - 10-15 min
- It's not about a right or complete answer…
- It concerns how to think about system-level considerations
- Have fun!

Q&A

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat