The Internet of Things

The Internet of Things (IoT) is the network of physical objects or "things" embedded[1] with electronics, software, sensors[2] and connectivity to enable it to achieve greater value and service by exchanging data with the manufacturer, operator and/or other connected devices. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.

-- Wikipedia

The Internet of Things

Kevin Ashton

- MIT
- Helped create RFID as an Open Standard
  - ...where everything has an ID
- Started "Internet of Things" concept
- (He's also interesting in other ways...)

If everything has an address...

- *Things* can have and report status
- Status is
  - immediate
  - useful
  - associated with a thing
  - associated with the thing's attributes
- *Things* can accept individual input
  - Control themselves
  - Control other things in the environment

What kind of information?

- Internal state
  - Temperature,Pressure,Speed
  - Location
- Environmental state
  - Weather
  - Traffic
  - Counts
- Financial, economic, network data
- Anything...

Even more things to measure...

- Traffic patterns
- Shopping habits
- Medical readings
- Wearables - health, mental state, locations
- Inventory
- Energy use
- Water quality
- Wildlife habits
- Airline locations

Feedback and control

- Devices can receive commands
  - Control themselves
  - Control directly connected things
  - Send control messages to other things
- Direct feedback and control
  - Microcontrollers, no networked control
- Networked feedback (or not) and control
  - Networking capability is needed (e.g. SOC)
  - Control decisions are made elsewhere

Things to control…

- Thermostats
- Speeds
- Alarms
- Positions
- Dosages
- Feeding
- Trades and economic activity
- Inventory orders
- ...and so forth...

Measurement devices

- Analog devices measure the real world
  - Temperature, light, speed, etc.
  - Usually a chemical, electrical, or magnetic sensor
- Analog to digital converters (ADCs)
  - Convert analog voltage to digital data
  - Which can be handled by a computer
  - Which can be sent via a digital network

Control Devices

- Mechanical analog devices control the real world
    - Driven by voltage levels
        - Presence of voltage
        - Intensity and frequency of voltage
    - Frequently electomagnetic or peizoelectric
- Digital to analog converter (DACs)
    - DACs create analog voltages from digital signals
    - Digital signals can be computed and sent
    - Analog voltages drive analog devices.

Computational Devices

- Requirements
  - small and inexpensive
  - simple to program
  - durable, physically and evironmentally
  - low power draw
  - easy to incorporate into products
- Devices
  - These are basically small computers
  - History and details later

How much of this is there?

- More things now than people on earth
- 20-30 billion things by 2020
- But these are "interesting" things.
- What about _everything_?

The Internet of Objects

- RFID is extremely inexpensive
  - printable circuit costs pennies.
  - doesn't require inherent power.
  - not very smart. Just transmits a number.
  - what about a 'smart' RFID?
- Internet communication for _everything_
  - *Trillions* of things -- soup cans, toothpaste tubes
  - Can you imagine the data?

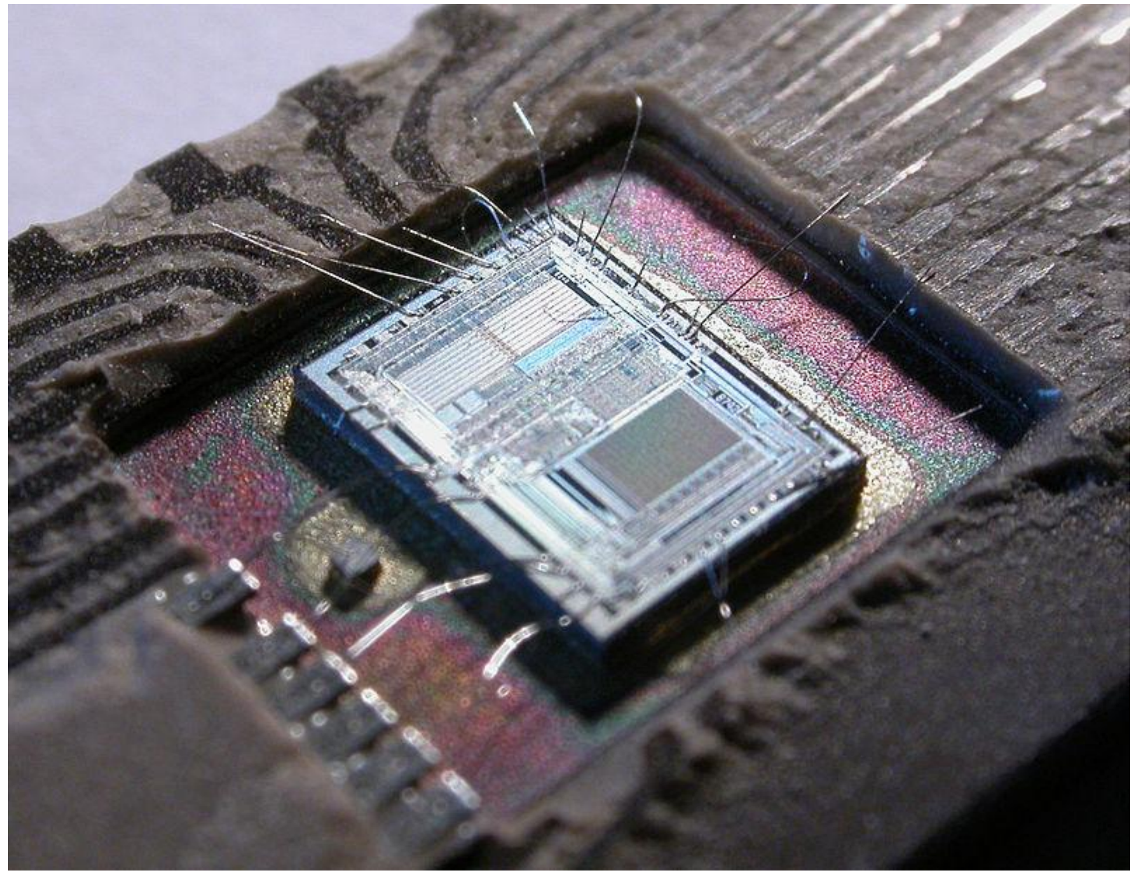
Back to Interesting Things

- Is the data or control worth $20?
- If so, you can afford a small computer
  - Processor
  - Memory
  - Storage
  - Input/output
- Early *microcomputers*
  - used external chips for memory, I/O, storage
  - expensive

Microcontrollers - system on a chip

- Intel 8742
- 12 MHz
- 128 Bytes (yes, bytes) RAM
  - Variables go here
- 2048 Bytes of EPROM
  - Put programs there
- Digital I/O ports

Put assembly program into EEPROM
(electrically erasable ROM)
Power on - starts running
Power off - stops running



Microcontroller programs

If thermometer says too cold
    Turn on heater
If thermometer says too hot
    Turn on air conditioner
Otherwise
    Turn them both off

Microcontroller characteristics

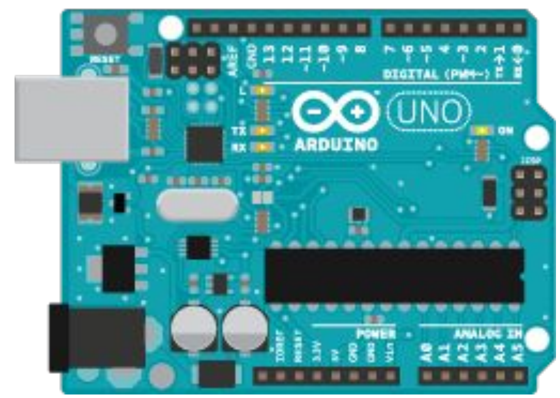
- Local data collection
- Local decisions
- Some LAN communication
  - E.g. cars, fly-by-wire, environmental control
- Little WAN communication
  - Networking is *hard*.
- Hard to program complexity
- Cheap and durable, cheaper than wires

Microcontroller usage

- Tiny computers in everything
- Cheaper than conventional circuitry
  - E.g. cheaper in TVs than analog tuning
  - Sensors and servos cheaper than long wires
- Smaller than other options
  - Ever seen an original iPad nano? A Tamagochi?
- Great for operating devices
- Not so great for Internet networking
  - Often networking is harder than data collection

Current Generation Microcontrollers

- Standardized circuits
- Easier to program
- Development kits available
- We will look at the Arduino
  - Very common for hobby use
  - Industrial versions for real world
  - http://www.arduino.cc/
  - Subject of this week's lab (on a simulator)



Microcontroller advantages

- Versatile
- Common
- Cheap
- Durable
- Low power consumption
- Easy to learn

Microcontroller limitations

- Harder to network
  - (Networking modules are available)
- Require external IDEs
- Nonstandard tools and languages (maybe)
- Relatively slow and not very smart
- Not much local processing power

We want an *Internet* of things

- Microcontrollers are toy computers
- Cell phones need *real* computers
- Cell phones need to communicate
- So what powers cell phones?

System on a Chip (SOC)

A system on a chip or system on chip (SoC or SOC) is an integrated circuit (IC) that integrates all components of a computer or other electronic system into a single chip. It may contain digital, analog, mixed-signal, and often radio-frequency functions—all on a single chip substrate.

-- Wikipedia

System on a Chip (SOC)

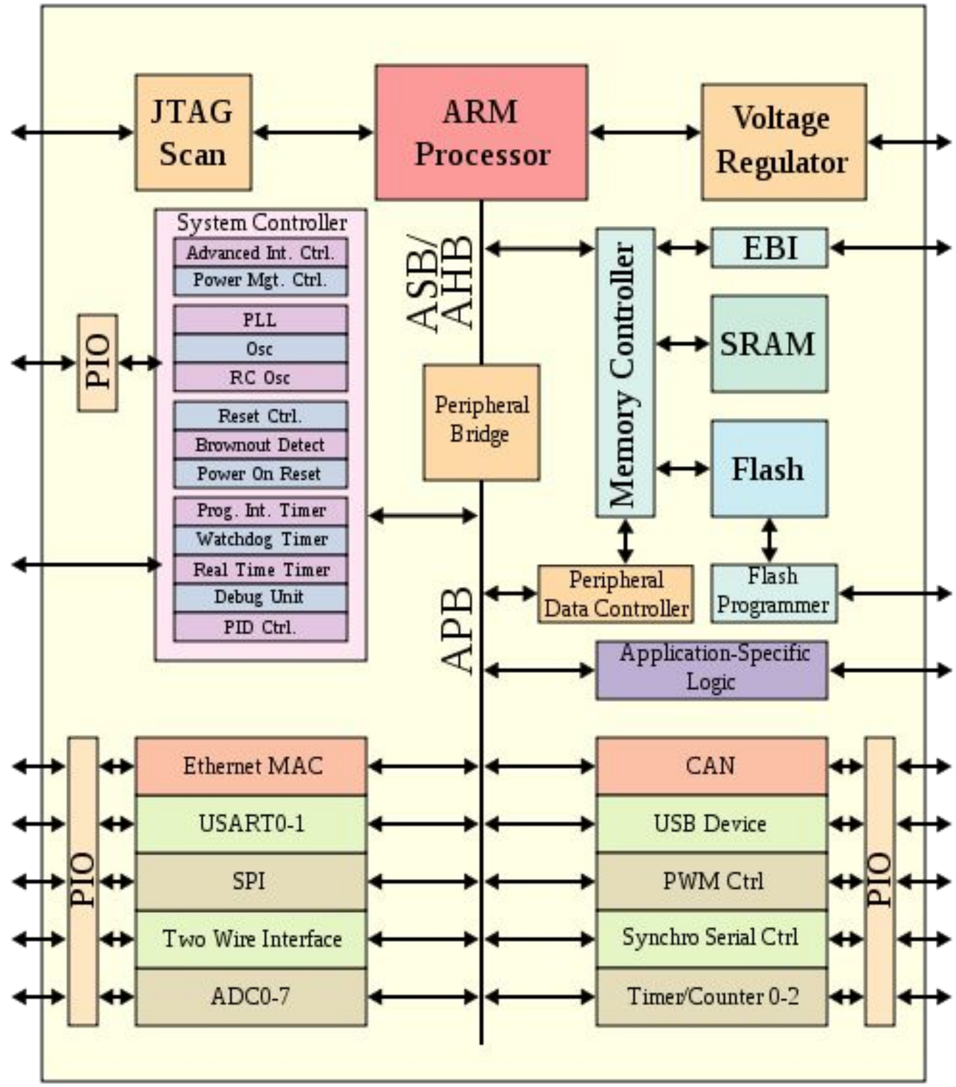
- System on a Chip (SOC)
  - Significant amounts of RAM
  - Local storage as flash-ram
  - Ability to run conventional software
    - Linux
    - Windows
    - Well-known software development stacks
  - Reasonably low power consumption
  - Easy to incorporate into designs

ARM SOC

ARM devices

- RISC
- Low power
- Open design
- Fairly fast

Seems useful...



SOCs become popular

Check this out:

http://en.wikipedia.org/wiki/List_of_system-on-a-chip_suppliers

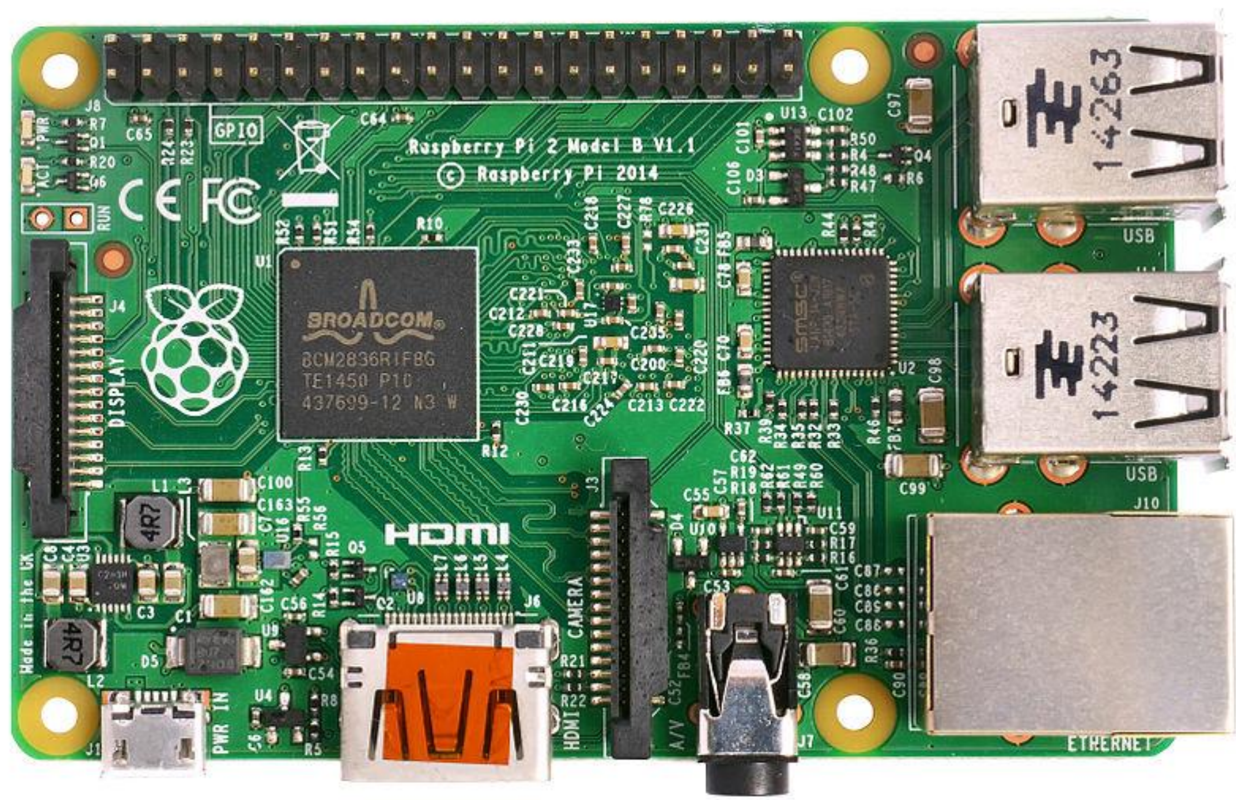
- lots of options
- sold in manufacturing quantities
- a few hobbyist SOCs designs emerge
-

A Teaching Problem...

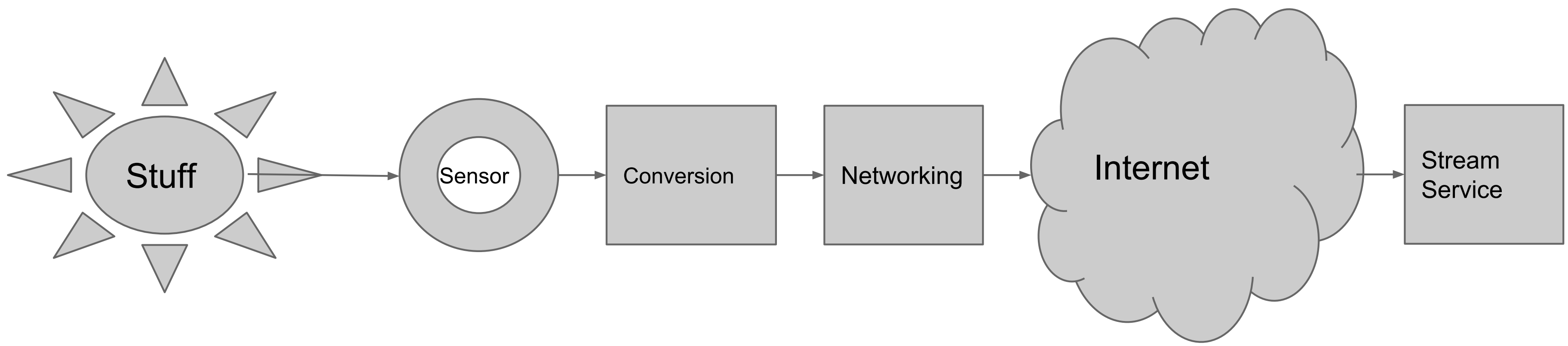
- Students in England getting worse at CS
- Why?
  - Home computers used to be toys
  - Modern home computers are _*not*_ toys.
  - Kids can't tear apart Mom and Dad's MacBook Pro
- Solution
  - Raspberry Pi Foundation formed
  - Design and build a cheap, educational SOC board
  - Make it very flexible and easy to use

The Raspberry Pi

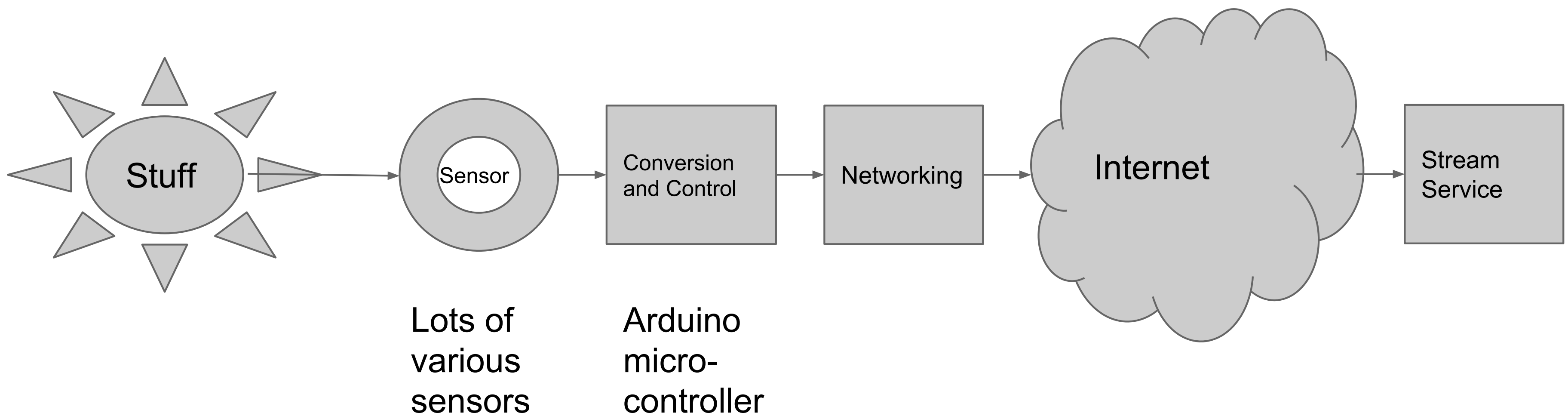
1 GHz
1 Gb
Linux
Ethernet
USB
GPIO



Data Collection Architecture



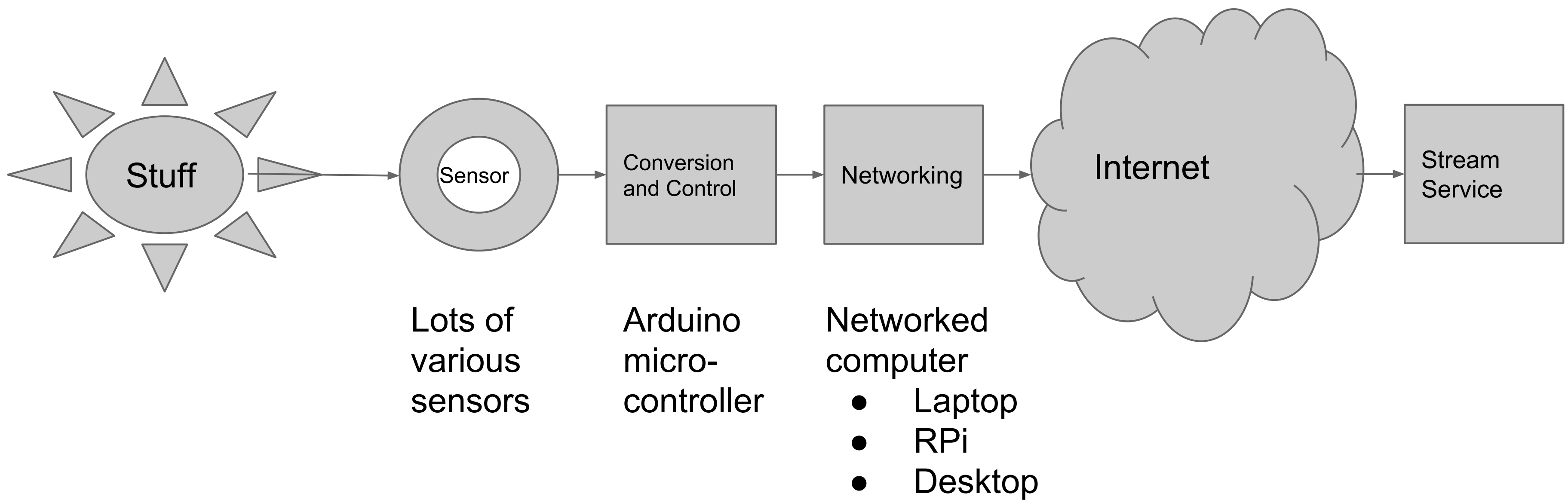
Data Collection Architecture



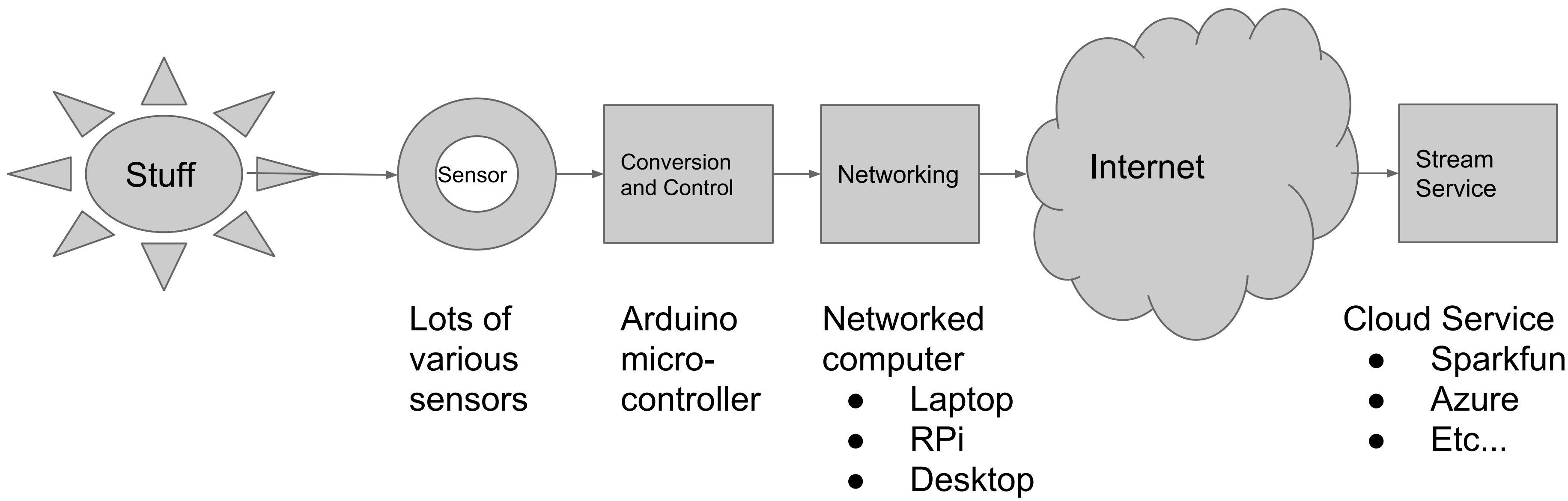
...About the Raspberry Pi

- So it's a great little device for lots of things
- It does not collect data well
  - Lots of power consumption
  - Relatively large
  - No analog ports
- Communicates very well
  - But so does your laptop
  - You're welcome to use an RPi or your laptop for that
  - An RPi is a cheaper device to dedicate to that job

Data Collection Architecture



Data Collection Architecture



Lab Preparation

- Bring your laptop running Chrome
  - Make sure WiFi works for you
- Read a little about the Arduino
  - http://www.arduino.cc/
  - http://en.wikipedia.org/wiki/Arduino
- We will be programming a simulated Arduino

Demo Time

Questions?

See you next time!