UCDAVIS
UNIVERSITY OF CALIFORNIA

Internet of Things

Senior Design Project Course

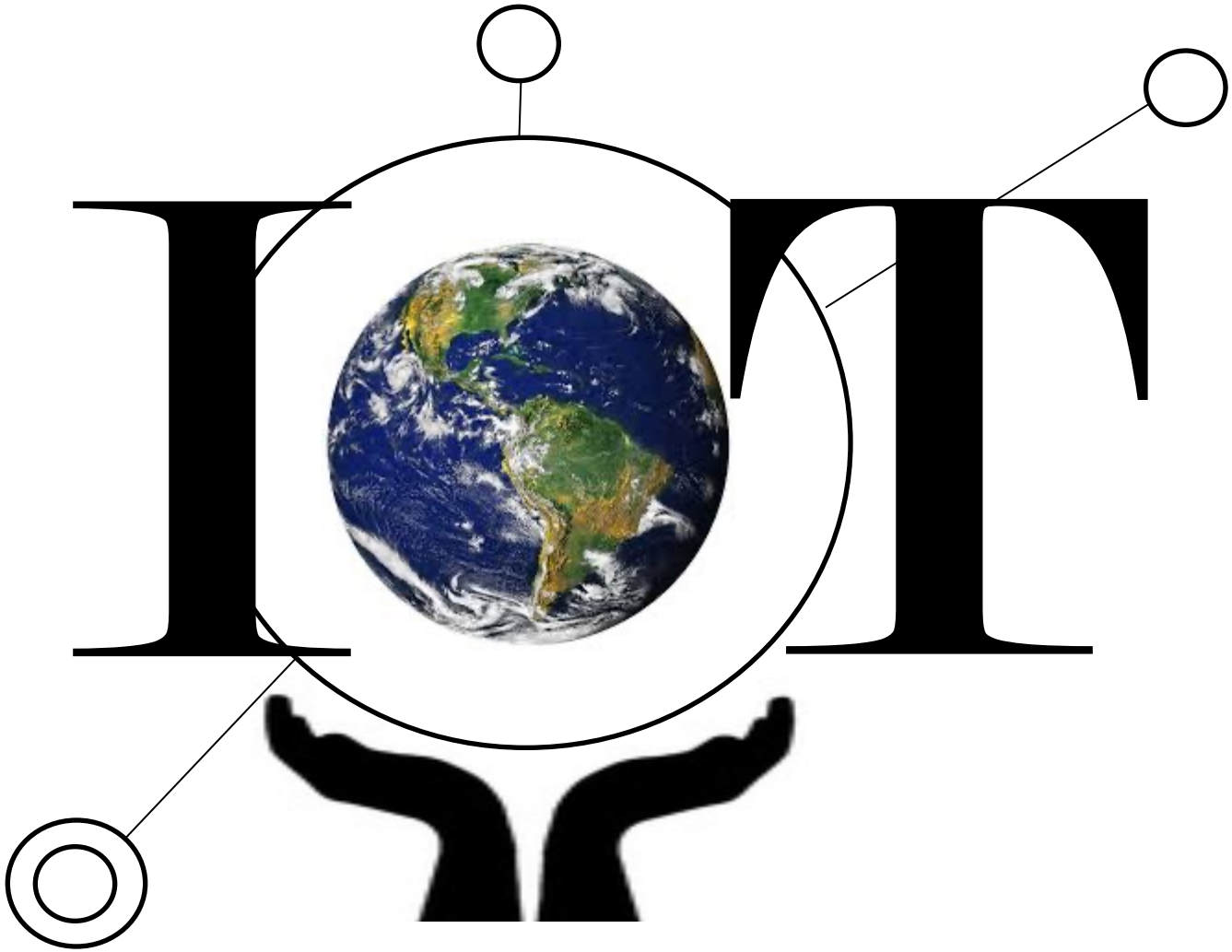
Communication – Part 1

MCU to Gateway

Lecturer: Avesta Sasan

University of California Davis

Lets Get Started:



Focus of Today's Lecture:

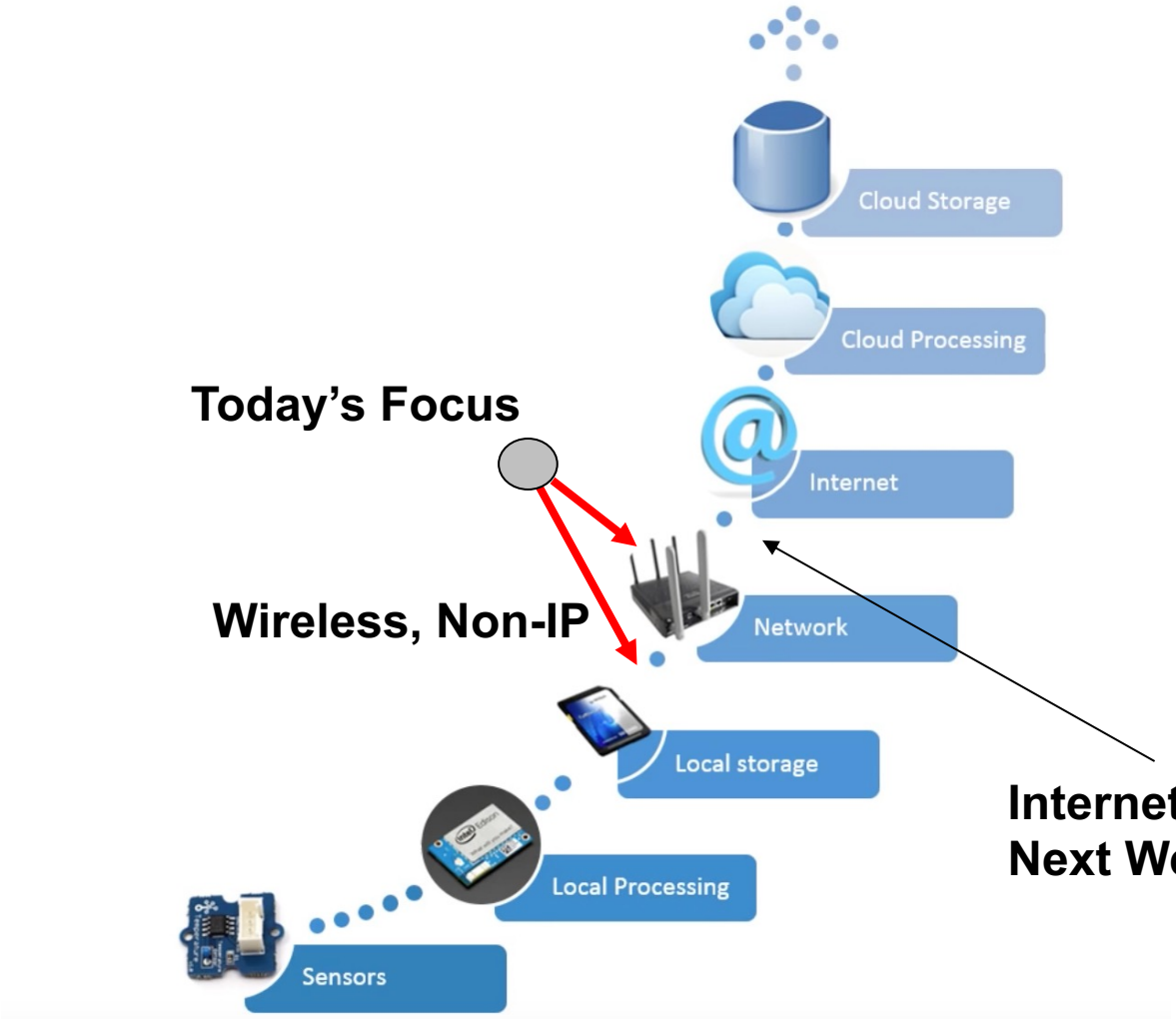
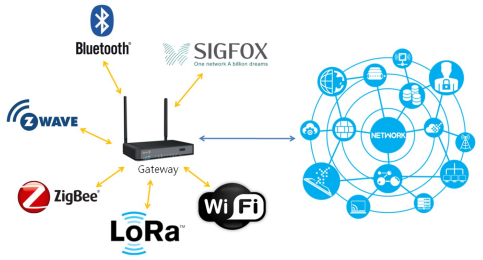


Image source: http://www.cchc.cl/informacion-a-la-comunidad/industria-de-la-construccion/personaje/

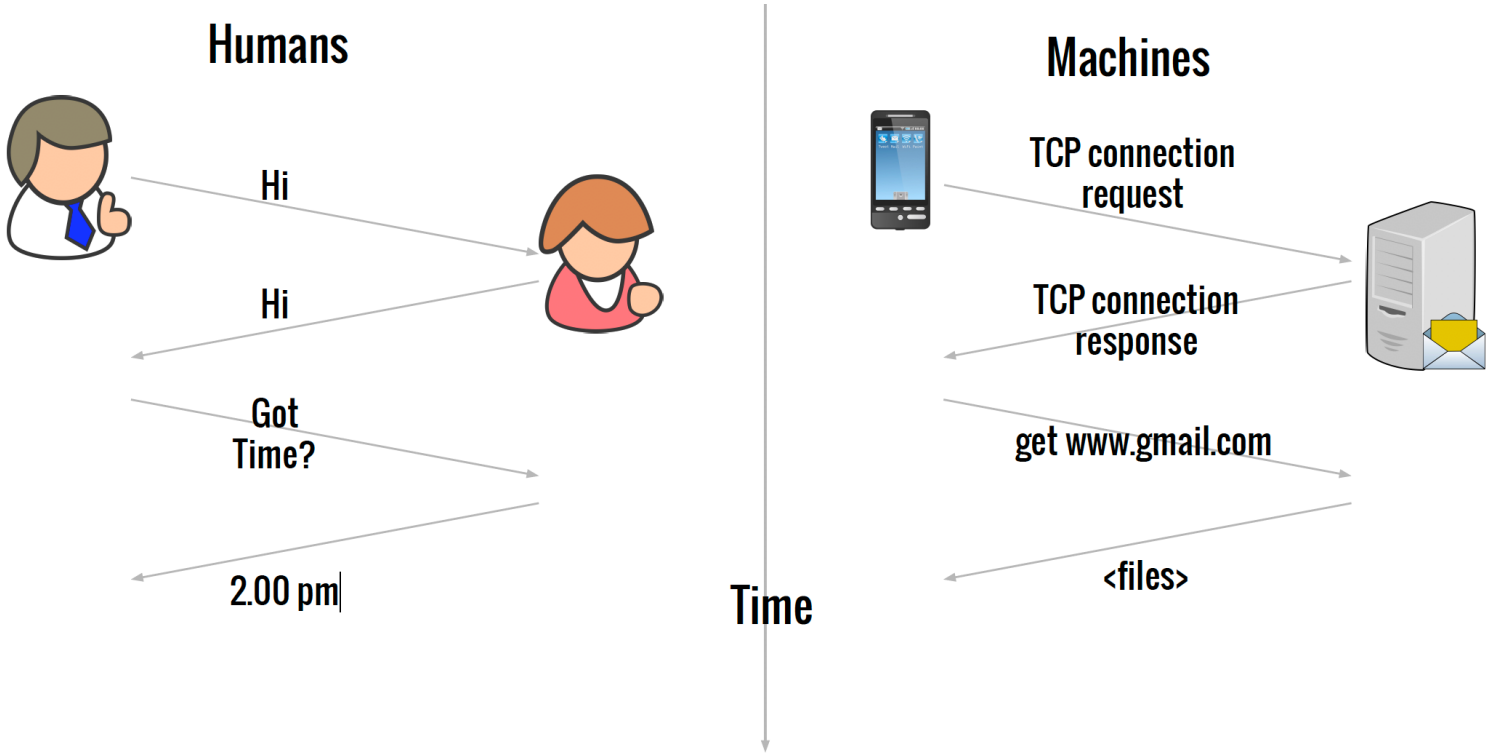
Why Using Gateway?

1. **Lowering Power**
   - Sensor sends the data to a gateway in short range (requires lower power)
   - gate way send the data to cloud.
2. **Supporting varying** to/from sensor communication **protocols**
   - Each sensor may have a different protocol.
   - Gateway translate it to IP



3. **Filtering the data**
   - Usually small fraction of data is usable.
   - Filtering could be done at gateways.
     (more resources than MCU, reduce communication size, reduce cloud computation load)
4. **Reducing latency**
   - Many IoT devices too small to do the processing themselves, and it take too long to wait for cloud. Gateway (an intermediate computation layer) remedy this.
5. **Improving security**
   - Can afford to make data transmission through gateway more secure.
   - Prevent too many lightly secured sensors to be connected to internet.

What is a Protocol



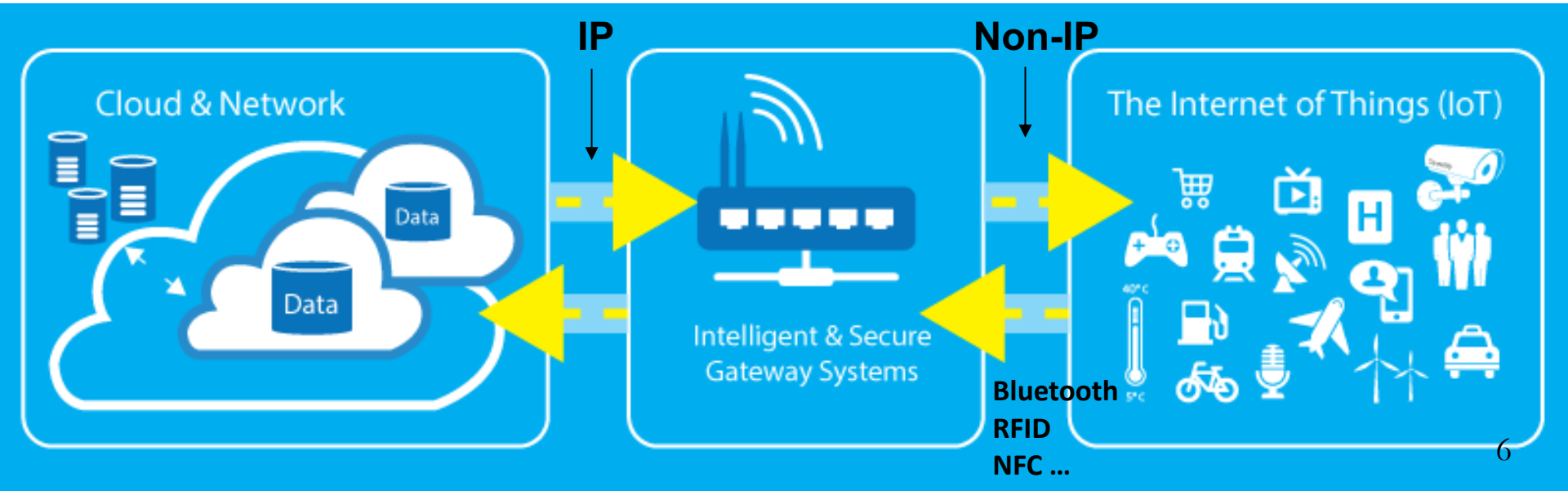
Language, Semantics, Grammar, Loudness, Noise, Reliability (repetition)

Protocols, Interface, Sockets, Signal Strength, Noise, Error Recovery (Re-transmission)

Gateway and Protocol Translation

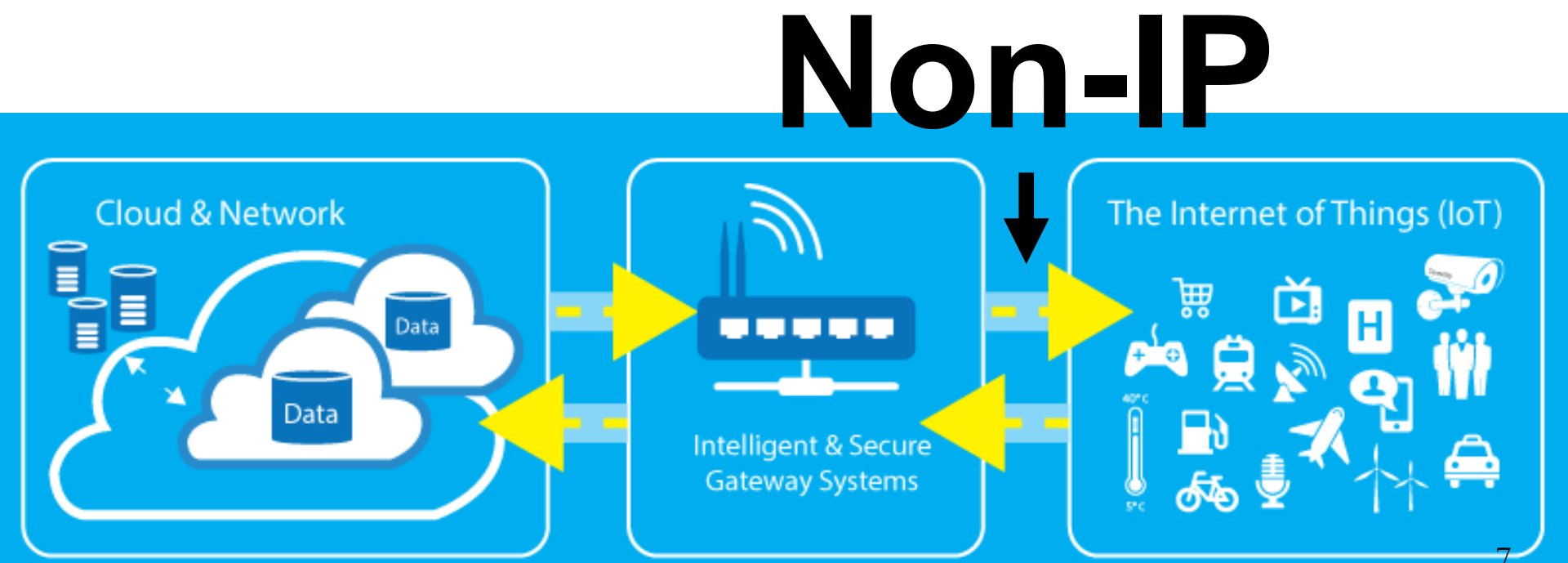
- IoT devices can connect to Internet using **Internet Protocol (IP)**
  Problem:
  - IP stack is very **complex** and demands a large amount of **power** and **memory** from the connecting devices.
  - **Wired**, hence **mobility** is limited!

- **Gateway removes the need for direct connection to internet.**
  - IoT devices can also connect locally through non-IP networks, which consume less power and offer larger mobility, and connect to the Internet via a smart gateway. It also enhances mobility of IoT devices.

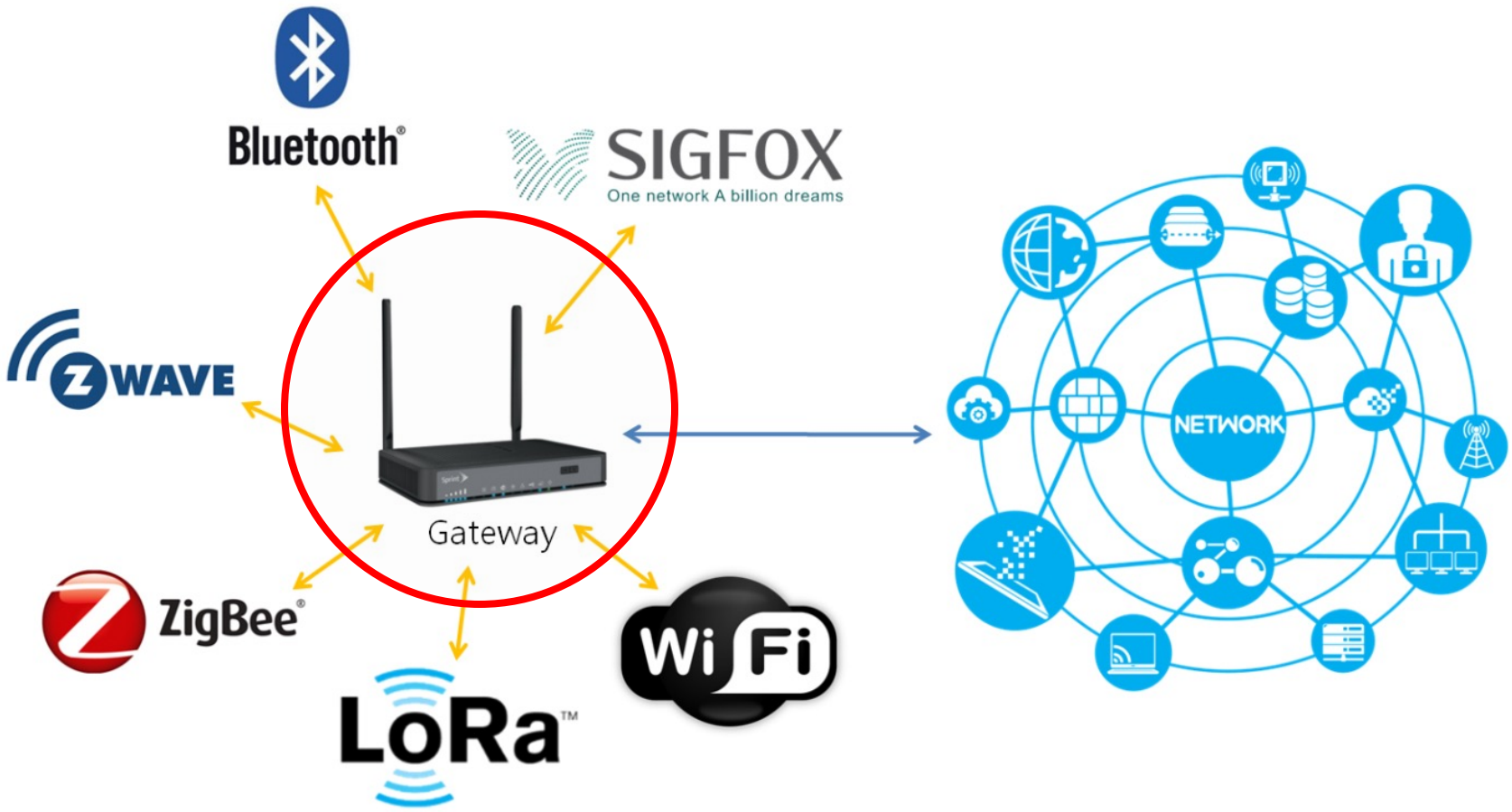


Protocol Interfacing IoT Objects

- Now lets look at the communication between gateway and IoT devices.



Gateway Supporting Various Protocols



Wireless Comm. To/From Gateways

- Wireless communication technologies used for:
  - Connecting the IoT device as local networks
  - Connecting these local networks (or individual IoT devices) to the Internet

- Some of popular wireless technologies are:
  - Near Field Communication (**NFC**)
  - **Bluetooth**
  - **ZigBee**
  - Wireless Fidelity (**WiFi**)
  - **Cellular network**
  - Low Power Wide Area Network (**LPWAN**)
  - …

[1] F. Samie, L. Bauer and J. Henkel, "IoT technologies for embedded computing: A survey," *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Pittsburgh, PA, 2016, pp. 1-10.

NFC

Near Field Communication (NFC)

- **Short-range** (4~20 cm)
- **Low-speed** connection with simple setup
  - supported data rates: 106, 212 or 424 kbit/s.
  - operates at 13.56 MHz on ISO/IEC 18000-3 air interface.
- Can be used **to bootstrap** more capable wireless connections
- Has a tag that can contain small amount of data
  - can be read-only
  - can be re-writable
- Popular current usage:
  - contactless payment systems
  - sharing contacts, photos, videos or files

NFC

Near Field Communication (NFC)

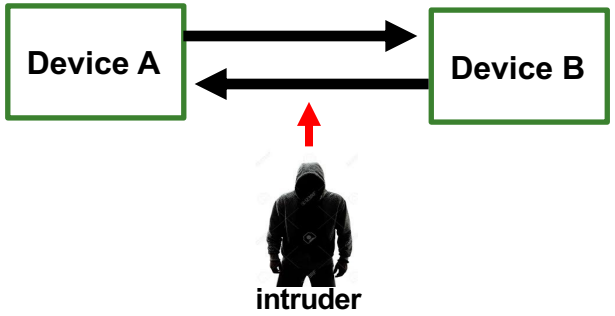
- NFC devices are **Full-Duplex**
  - able to receive and transmit data at the same time.

- **Modes of operation:**
  - **Passive**
    - the initiator device provides a carrier field.
    - the target device answers by modulating the existing field.
    - the target device may draw its operating power from the initiator-provided electromagnetic field, making the target device a transponder.
  - **Active**
    - Both initiator and target device communicate by alternately generating their own fields.
    - A device deactivates its RF field while it is waiting for data.
    - both devices typically have power supplies.

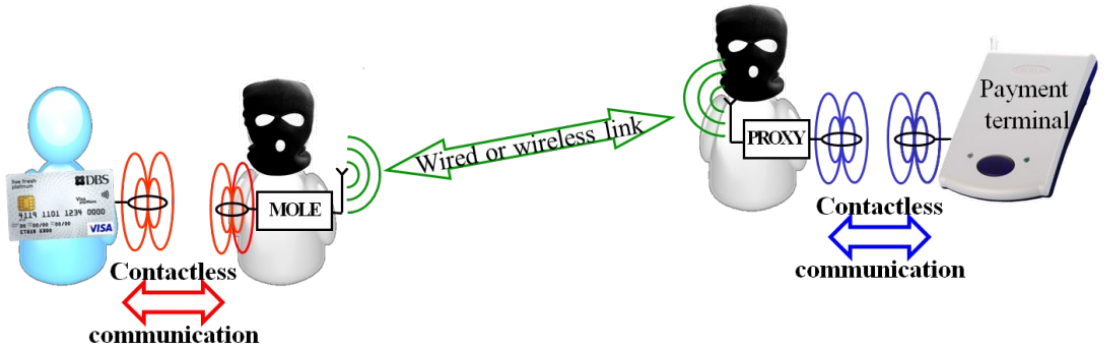
NFC Security Problems!

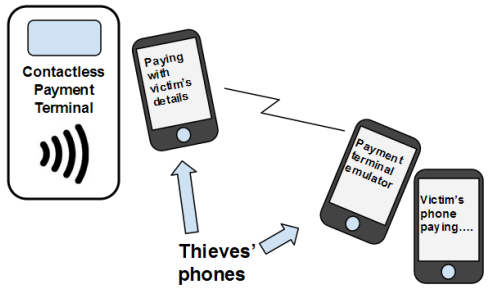
Eavesdrop:
/ˈēvzˌdräp/
To secretly listen to a conversation

- **No protection against eavesdropping**
  - Can be vulnerable to data modifications
  - An attacker can typically eavesdrop within
    - 10 m for active devices
    - 1 m for passive devices
  - Can use higher-layer cryptographic protocols (e.g. SSL) to establish secure channel



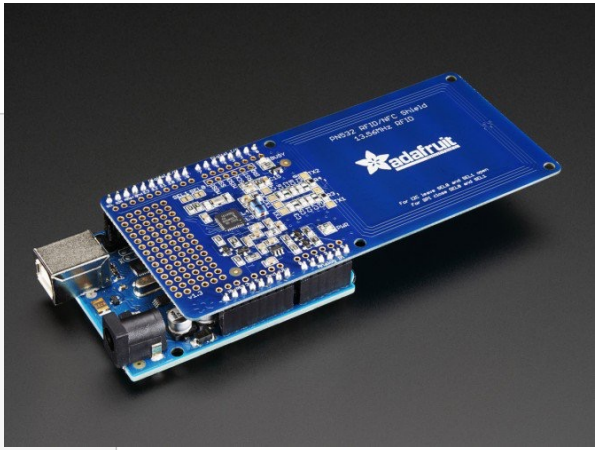
- **Relay attacks are feasible**
  - The adversary forwards the request of the reader to the victim
  - Relays its answer to the reader in real time pretending to be the owner of the victim's smart card





Using NFC with Arduino

- Read and write on NFC tag using Arduino
- You can add NFC capabilities by adding a shield to Arduino:
  - Adafruit PN532 RFID/NFC Shield




```
#include <Wire.h>
#include <PN532_I2C.h>
#include <PN532.h>   // The following files are included in the libraries Installed
#include <NfcAdapter.h>

PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc = NfcAdapter(pn532_i2c);  // Indicates the Shield you are using
```

```
void setup(void) {
  Serial.begin(9600);
  Serial.println("NFC TAG READER"); // Header used when using the serial monitor
  nfc.begin();
}
```

Using NFC with Arduino

```
void loop(void) {
  Serial.println("\nScan your NFC tag on the NFC Shield\n");  // Command so that you an others
will know what to do

  if (nfc.tagPresent())
  {
    NfcTag tag = nfc.read();
    Serial.println(tag.getTagType());
    Serial.print("UID: ");Serial.println(tag.getUidString()); // Retrieves the Unique Identific
ation from your tag

    if (tag.hasNdefMessage()) // If your tag has a message
    {

      NdefMessage message = tag.getNdefMessage();
      Serial.print("\nThis Message in this Tag is ");
      Serial.print(message.getRecordCount());
      Serial.print(" NFC Tag Record");
      if (message.getRecordCount() != 1) {
        Serial.print("s");
      }
      Serial.println(".");

      // If you have more than 1 Message then it wil cycle through them
      int recordCount = message.getRecordCount();
      for (int i = 0; i < recordCount; i++)
      {
        Serial.print("\nNDEF Record ");Serial.println(i+1);
        NdefRecord record = message.getRecord(i);

        int payloadLength = record.getPayloadLength();
        byte payload[payloadLength];
        record.getPayload(payload);


        String payloadAsString = ""; // Processes the message as a string vs as a HEX value
        for (int c = 0; c < payloadLength; c++) {
          payloadAsString += (char)payload[c];
        }
        Serial.print("  Information (as String): ");
        Serial.println(payloadAsString);


        String uid = record.getId();
        if (uid != "") {
          Serial.print("  ID: ");Serial.println(uid); // Prints the Unique Identification of th
e NFC Tag
        }
      }
    }
  }
  delay(10000);
}
```

Bluetooth

Bluetooth

- Technology designed and marketed by the Bluetooth Special Interest Group (SIG)
- A wireless technology standard for exchange of data over **short distances**.
- Uses short-wavelength Ultra High Frequency (**UHF**) radio waves in the industrial, scientific and medical **(ISM)** band from 2.4 to 2.485 GHz.
  - ISM is globally unlicensed (but not unregulated) band.
- Invented by telecom vendor Ericsson in 1994
- managed by the Bluetooth Special Interest Group (SIG)
- Bluetooth is a packet-based protocol.
- Master slave structure
- Multiple flavors:
  - Classic Bluetooth
  - Bluetooth Low Energy (BLE)
  - Bluetooth 5.0 (BT v5)
  - …

Bluetooth Flavors

- Classic **Bluetooth**
  - High throughput and bandwidth
    - Suitable for data stream applications
  - Limitations
    - Limited number of nodes in the network (up to seven slaves)

- **Bluetooth Low Energy (BLE)**
  - A.K.A. Bluetooth smart
  - Originally introduced under the name **Wibree** by Nokia in 2006
  - Short-rage, low bandwidth, and low latency
  - Lower power consumption
  - Lower setup time
  - Supports unlimited number of nodes

Bluetooth Classic vs BLE

- Choose wisely!

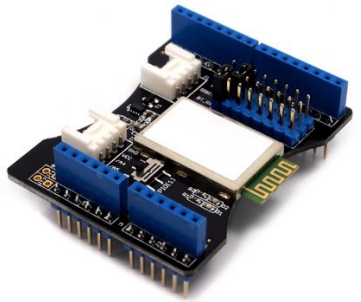
	Bluetooth Classic	Bluetooth Low Energy
Range (theoretical)	100 m	> 100 m
Power consumption	1 W	0.01 to 0.5 W
Peak current consumption	<30 mA	<15 mA
Data rate	1-3 Mbit/s	1 Mbit/s
Radio Frequencies	2.4 GHz	2.4 GHz
Focus	Wireless protocol for short range data exchange	Low power consumption – periodic exchange of small amounts of data
Use Cases	Wireless speakers, headsets	Wearable devices, smart pay systems

Bluetooth Flavors

- **Bluetooth 5.0 (BT v5)**
  - Focus on IoT
  - The Bluetooth SIG officially unveiled Bluetooth 5 on June 16, 2016
    - Iphone 8, Samsung Galaxy S8, Xpheria XZ (sony) first to lunch with BT v5.0
  - Doubles the speed of low energy connections (2MBps)
  - Quadruple the range
    - Forward Error Correction (FEC)
  - Eight-fold increase in data broadcasting
  - Today: BT v5.3 (as of July 2021)

Bluetooth with Arduino

- Three connectivity options:
  - **Arduino BT**: an MCU with embedded Bluetooth capabilities.
  - Add a **Bluetooth shield**.
  - Connect a standalone BLE module
    - Connect using UART, I2C, SPI, …



- (1) Example code (control Arduino with BLE)
- (2) Example code