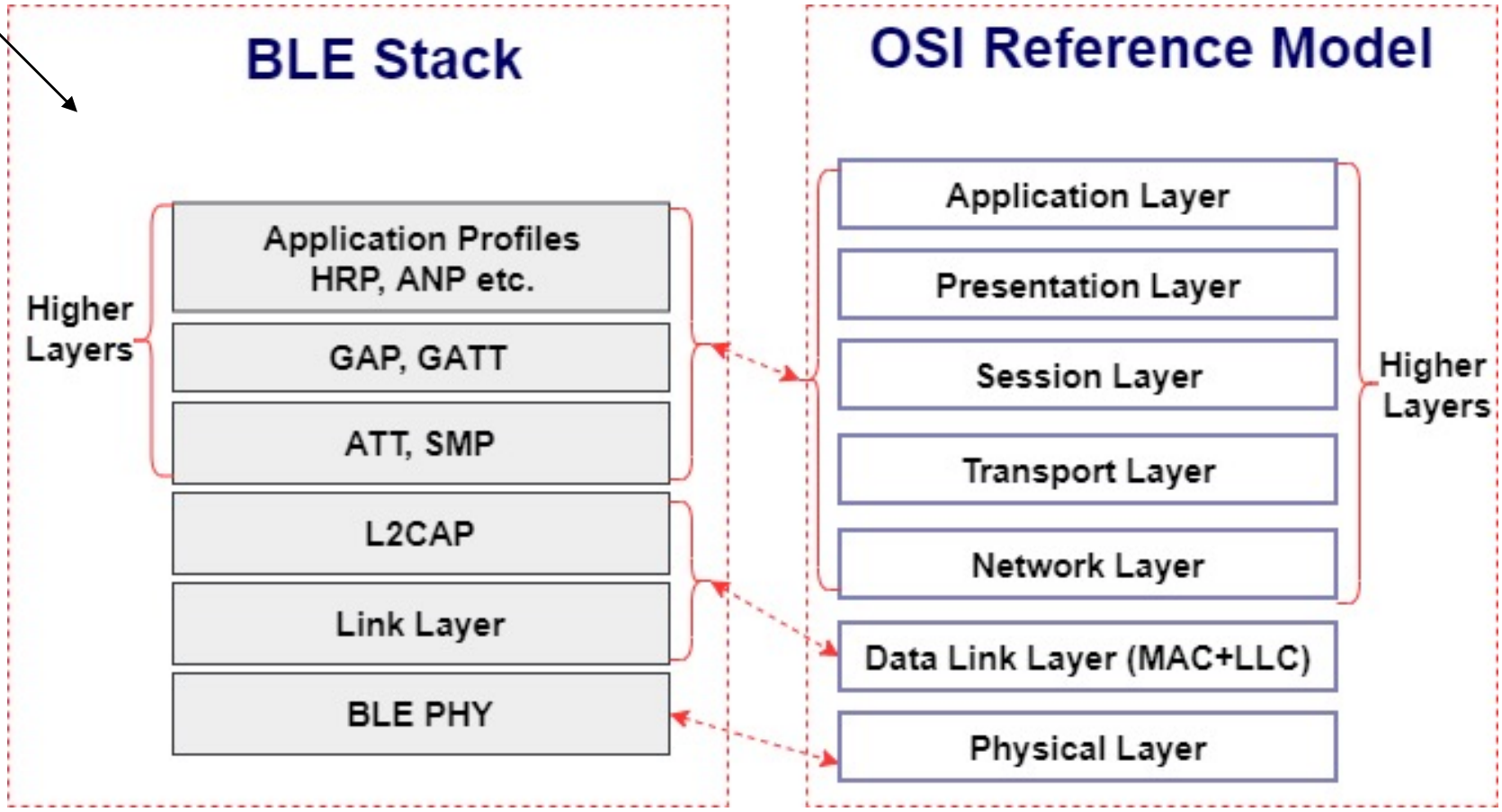
Carnegie Mellon University

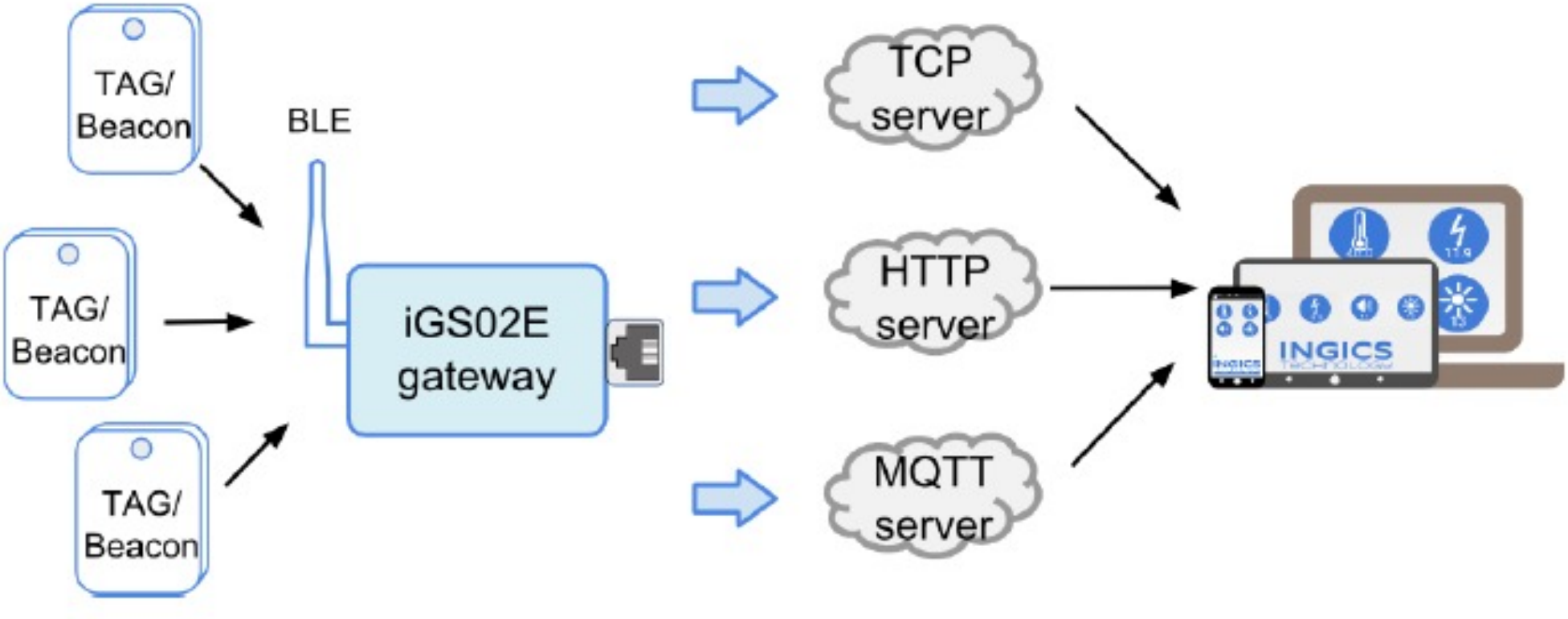
95-733 Internet of Things
Bluetooth LE

Where are we?

We are here.



HRP = Heart Rate Profile, ANP = Alert Notification Profile.
We are not in the TCP/IP world. The stack on the left and the TCP/IP stack would both run on the gateway in the next slide.



Why not have the tags communicate directly with MQTT?
More bits means more energy and the tags may be battery powered.

Bluetooth Classic and LE

- Bluetooth is like a wireless version of serial communication.
- Bluetooth classic handles a lot of data but consumes lots of power.
- Bluetooth Low Energy is typically used for small amounts of data and may run for years on battery power.
- Bluetooth works over a short range (< 100 m). Normally much less.
- Both operate at 2.4GHz in the unlicensed Industrial, Scientific, and Medical (ISM) frequency band (like ZigBee and WiFi). Managed by Bluetooth SIG.
- Bluetooth networks are called piconets.
- A device may support multiple services. For example, most heart rate monitors include both the Heart Rate Measurement Service and Battery Service.

Bluetooth LE

- A central device coordinates communication within the pico net.
- Peripheral devices may transmit data to or receive data from only their central device.
- Each device has a unique 48 bit address – abbreviated BD_ADDR
- 48 bits (24 assigned to manufacturer and 24 assigned by the organization)
- For example MAC NO. 000666422152
- A peripheral device acts as a client while advertising its presence. “Anyone want to talk? Anyone want to talk?...”
- A central device may be scanning and agree to talk and becomes the client of the service offered by the peripheral.
- An advertising packet may include the peripherals service UUID’s.

Bluetooth LE

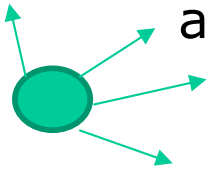
- Creating a connection:
  - One device transmits an inquiry request to any listener (discovery).
  - This request includes the sender's address.
  - Another device responds with its address.
- Each device knows the address of the other.
- For two devices to interoperate, they both must use the same profile (perhaps HRP or ANP, more in a moment).
- The profile determines what application its geared toward.
- Example: A heartrate peripheral communicates with an Argon working as a gateway to the internet over Wi-fi.
- Example: An iBeacon transmits a URL in its BLE advertisement packet.

Bluetooth comparisons

Name	Bluetooth Classic	Bluetooth 4.0 Low Energy (BLE)	ZigBee	WiFi
IEEE Standard	802.15.1	802.15.1	802.15.4	802.11 (a, b, g, n)
Frequency (GHz)	2.4	2.4	0.868, 0.915, 2.4	2.4 and 5
Maximum raw bit rate (Mbps)	1-3	1	0.250	11 (b), 54 (g), 600 (n)
Typical data throughput (Mbps)	0.7-2.1	0.27	0.2	7 (b), 25 (g), 150 (n)
Maximum (Outdoor) Range (Meters)	10 (class 2), 100 (class 1)	50	10-100	100-250
Relative Power Consumption	Medium	Very low	Very low	High
Example Battery Life	Days	Months to years	Months to years	Hours
Network Size	7	Undefined	64,000+	255

GAP Basic Operation Profile

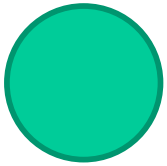
- GAP is an acronym for the Generic Access Profile.
- Controls advertising and connection establishment.
- Determines how two devices can (or can't) interact with each other.
- Two roles are defined: Peripheral (constrained device) and Central (more powerful device).
- After a connection, the roles of client and server are reversed.



advertisements

Peripheral:

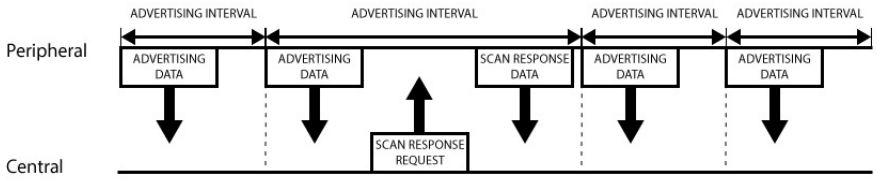
Every 20ms, send 31 bytes.
Or, every 2 seconds to tradeoff responsiveness for battery life.

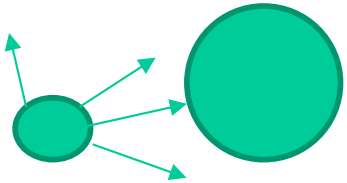


Central device

GAP Governs the Advertising

- When within range, the Central device receives the 31 bytes and may ask for an additional 31.
- The first 31 bytes are the Advertising Data and the second are the Scan Response payload.
- The first 31 are required and the second are optional. May be used to add additional information to the advertisement.



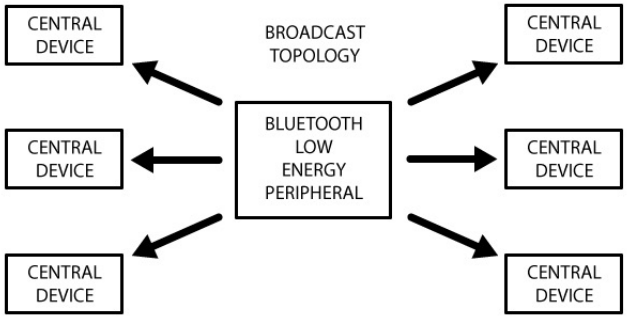


Central device

Peripheral device

https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

Just Broadcast a URI with GAP (e.g. iBeacon)



Use the advertisement data to transmit a URI

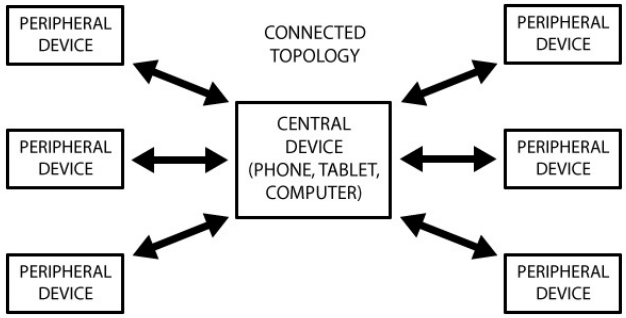
https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

GATT Basic Operation Profile

- GATT is an acronym for the Generic Attribute Profile.
- Used after the advertising by GAP. Query GATT for services available.
- Advertising stops and GATT services are used to transmit data in both directions.
- Since there is no more Advertising, a peripheral may only connect to one central device at a time.
- A central device may be connected to many peripherals at a time.
- If two peripherals need to talk, they would do so through a mailbox on the central device.
- GATT defines the way that two Bluetooth Low Energy devices transfer data back and forth using concepts called Services and Characteristics.
- It makes use of a generic data protocol called the Attribute Protocol (ATT), which is used to store Services.
- Characteristics and related data are stored in a simple lookup table using 16-bit IDs for each entry in the table.
- Establishing a connection is also the only way to allow two-way communication.

https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

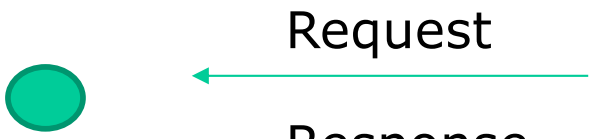
After a connection is established…

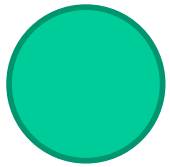


Both sides must agree on what the conversation is about.
The peripheral device is now the server. The central device is the client – initiating requests.

https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

GATT Transactions





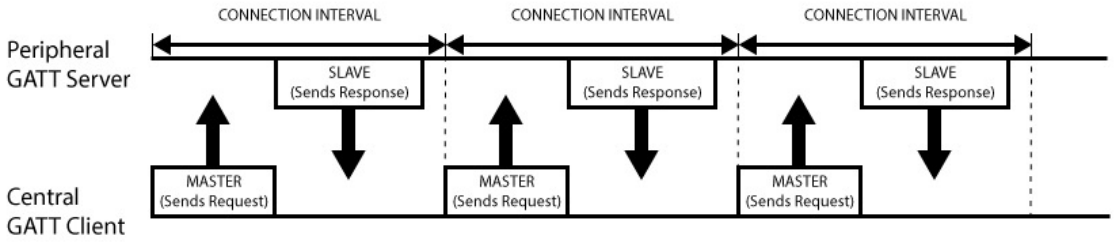
Central device
Client

Peripheral device
Server
Holds the ATT lookup data and service and characteristic definitions

When establishing a connection, the peripheral will suggest a 'Connection Interval' to the central device, and the central device will try to reconnect every connection interval to see if any new data is available.

https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

GATT Transactions



https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

GATT Transactions

The GATT client has no advanced knowledge of the server's attributes.

So, it must first inquire about the presence and nature of those attributes by performing service discovery.

After completing service discovery, it can then start reading and writing attributes found in the server, as well as receiving server-initiated updates.

https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch04.html

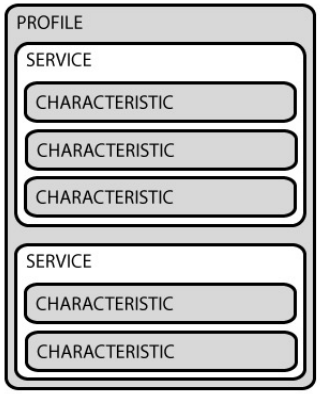
Services and Characteristics

A BLE service is a collection of characteristics.
Each service has a UUID (Universally Unique Identifier).
Many standard UUID's are established by Bluetooth SIG.
For example, a light bulb service has UUID of 0XFF10.
16-bit UUID's are part of the standard. These are reserved aliases for full 128-bit UUID's.
If you build a proprietary service, use 128 bits for your UUID.
For another example, the lock service has UUID 0xD270.
A BLE characteristic is a simple data value, e.g., heart rate.
Characteristics also have UUID's.
For example, the light bulb service has a switch characteristic with UUID 0xFF11 and a dim characteristic with UUID 0xFF12.
A characteristic may be Read, Write, Read and Write, Notify, or Indicate.
The last two are pub/sub and require a subscription step by the client.
Indicate requires an acknowledgement from the subscriber.

https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

Profiles

Transactions in BLE are based on high-level, nested objects called **Profiles - which include Services** and **Characteristics.**

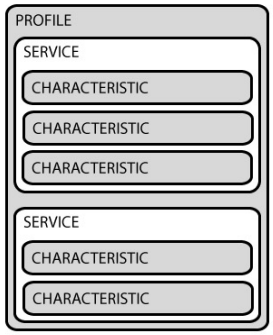


A profile is a well-defined collection of services defined by the SIG or device maker.
The heart rate profile, for example, combines the heart rate service and the device information service.
These services have characteristics.

https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

Services and Characteristics

Many profiles exist.
See https://www.bluetooth.com/specifications/gatt/



https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction

Health Thermometer Firmware

```
// The "Health Thermometer" service is 0x1809.
BleUuid healthThermometerService(0x1809);
// We're using a well-known characteristics UUID.
// The temperature-measurement is 16-bit UUID 0x2A1C
BleCharacteristic temperatureMeasurementCharacteristic("temp",
                    BleCharacteristicProperty::NOTIFY, BleUuid(0x2A1C),
                    healthThermometerService);

// The battery level service allows the battery level to be monitored
BleUuid batteryLevelService(0x180f);
// The battery_level characteristic shows the battery level
BleCharacteristic batteryLevelCharacteristic("bat",
                    BleCharacteristicProperty::NOTIFY,
                    BleUuid(0x2A19),
                    batteryLevelService);
```

From: https://docs.particle.io/tutorials/device-os/bluetooth-le/

BLE Security

- A Popular Electric Scooter Can Be Hacked to Speed Up or Stop
- A hacker can accelerate Xiaomi M365 scooter—or hit the breaks—while a rider is on it. - Wired 2019

https://youtu.be/ASygXa8UVYk

Rani Idan, Zimperium's director of software research, says he found and was able to exploit the flaw within hours of assessing the M365's security. His analysis found that the scooters contain three software components: battery management, firmware that coordinates between hardware and software, and a Bluetooth module that lets users communicate with their scooter via a smartphone app. The latter leaves the devices woefully exposed. (Wired)

Scooter Hack

- Idan quickly found that he could connect to the scooter via Bluetooth without being asked to enter a password or otherwise authenticate. From there, he could go a step further and install firmware on the scooter without the system checking that this new software was an official, trusted Xiaomi update. This means that an attacker could easily put malware on a scooter, giving herself full command over it.
- "An attacker could brake suddenly, or accelerate a person into traffic, or whatever the worst-case scenario you can imagine."
- When the company contacted Xiaomi to disclose the bugs, the scooter maker said it is aware of the problem and doesn't have the ability to fix it on its own. This is apparently because Xiaomi sources its Bluetooth implementation module from a third-party developer rather than coding it in-house. (Wired 2019)
- Researchers say that this vulnerability exists because the password used to log in to the mobile app is not verified by the scooter. https://lembergsolutions.com/blog/how-secure-ble-communication-standard

BLE Security

- BLE 4.2 is capable, during its pairing step, of using Elliptic Curve Cryptography and Diffie-Hellman key exchange to establish encrypted and authenticated communications.
- During a "bonding" step, the keys can be persisted for subsequent use – to skip the pairing step.
- The battle between the good players and the bad players continues.
- See September 2021 research: https://news.fit.edu/academics-research/apps-for-popular-smart-home-devices-contain-security-flaws-new-research-finds/