Bluetooth Low Energy
(BLE)

Bluetooth “classic”

- The “conventional” Bluetooth
- 2.4GHz
- Range: 1m - 100m (10m typical)
- Connection-oriented: audio, file transfer, networking
- Reasonably fast data rate: **2.1** Mbps
- Power consumption:
  - < Wifi < 3G

Bluetooth Low Energy

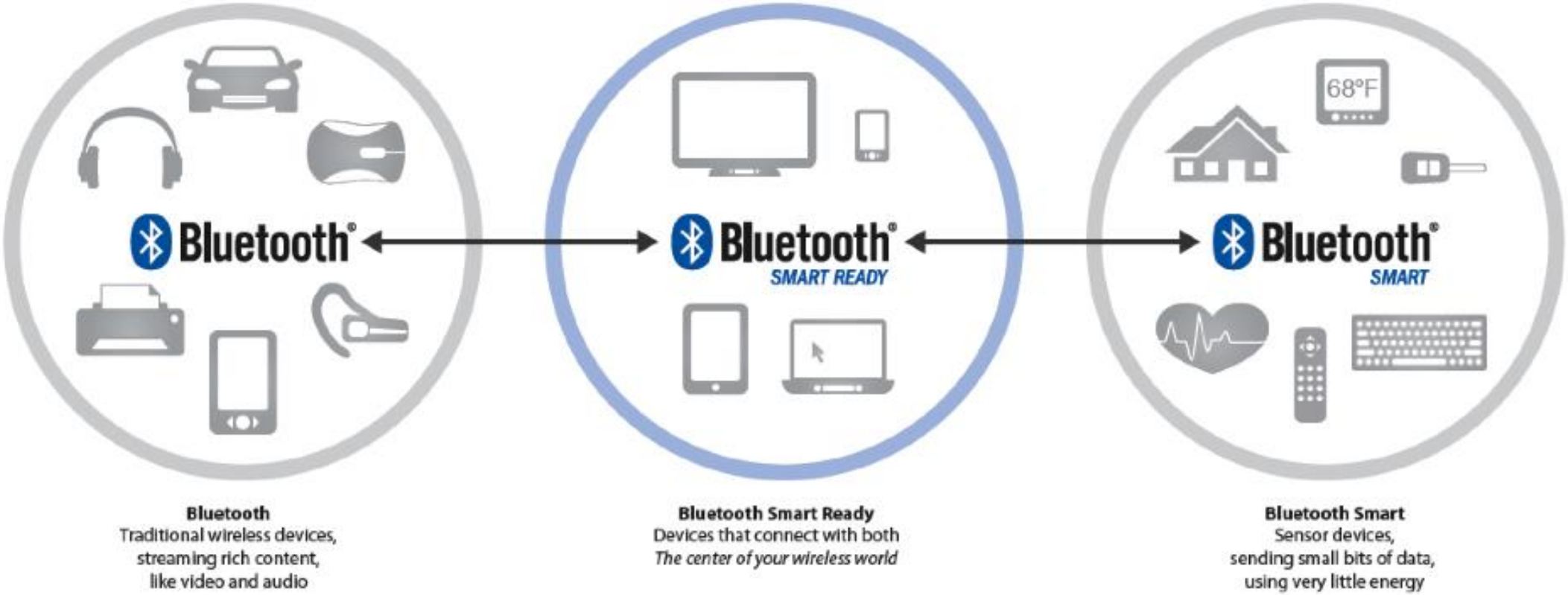
- "Bluetooth Smart“
- Light-weight subset of classic Bluetooth
  - Operates in same freq. Range
  - introduced as part of the Bluetooth 4.0 core specification
- Started by Nokia as an in-house project called “Wibree”
- Target Apps:
  - Wireless battery-powered sensors eg. heart rate, thermometer, wearables
  - Low bandwith
  - No always on, constrained devices

BLE Applications

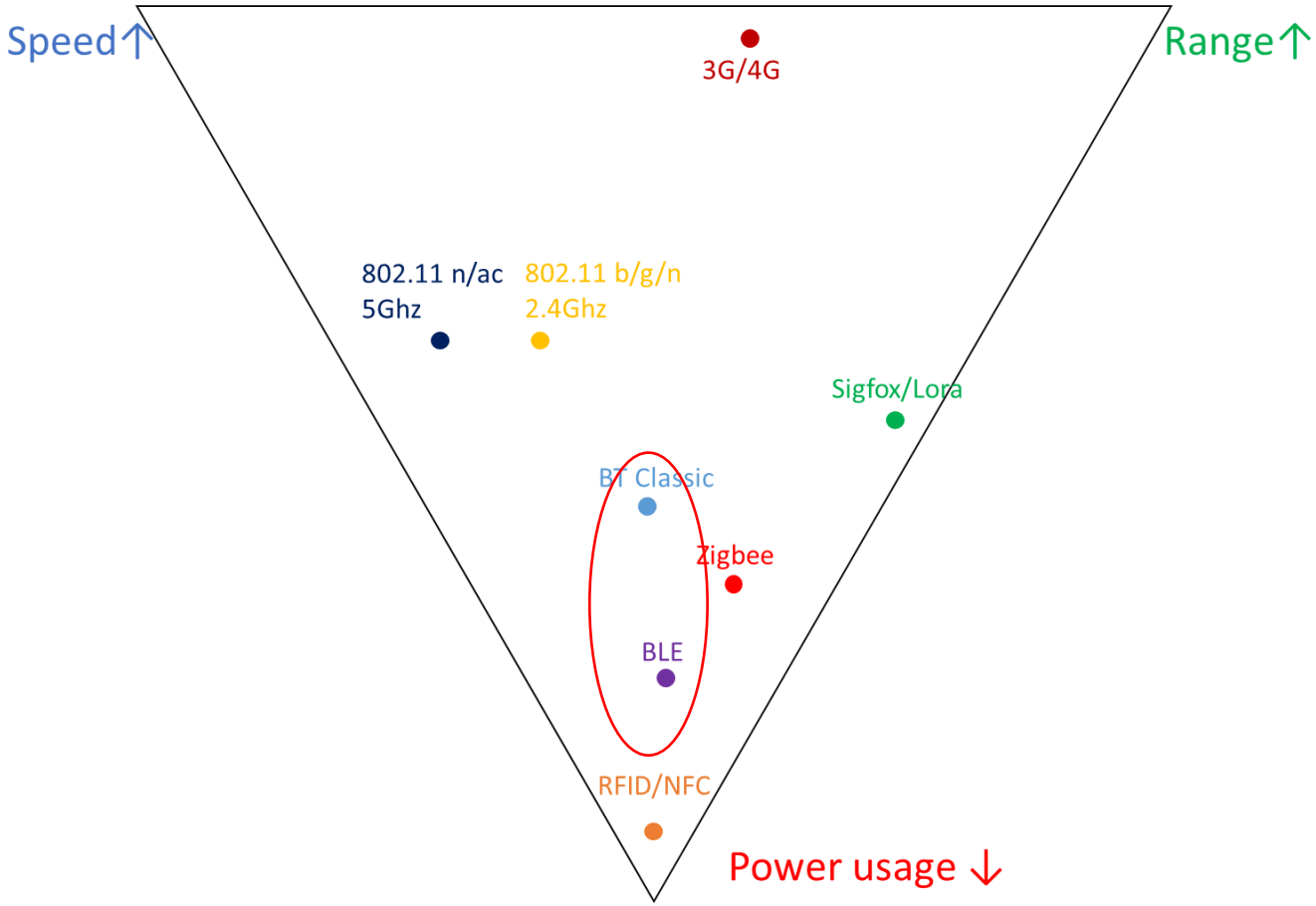
Bluetooth Classifications

- **Bluetooth Classic**
  - high throughput, e.g. wireless audio and file transmission.
- **Bluetooth Smart**
  - State information
  - Devices with low-duty cycles
- **Bluetooth SmartReady**
  - These devices are essentially the “hub” devices such as computers, smartphones, etc. They support both the “classic” and “smart” devices, just as our smartphones can connect to a Bluetooth speaker to transmit audio and also communicate to a fitness tracker.

Bluetooth family



Wireless Technologies Comparison



BLE vs Classic

- Bluetooth and Bluetooth Low Energy are used for different purposes
- Bluetooth Classic
  - can handle a lot of data
  - consumes battery quickly
- BLE
  - used for applications that do not need to exchange large amounts of data
  - cheap
  - Marginally further range

BLE Platform Support

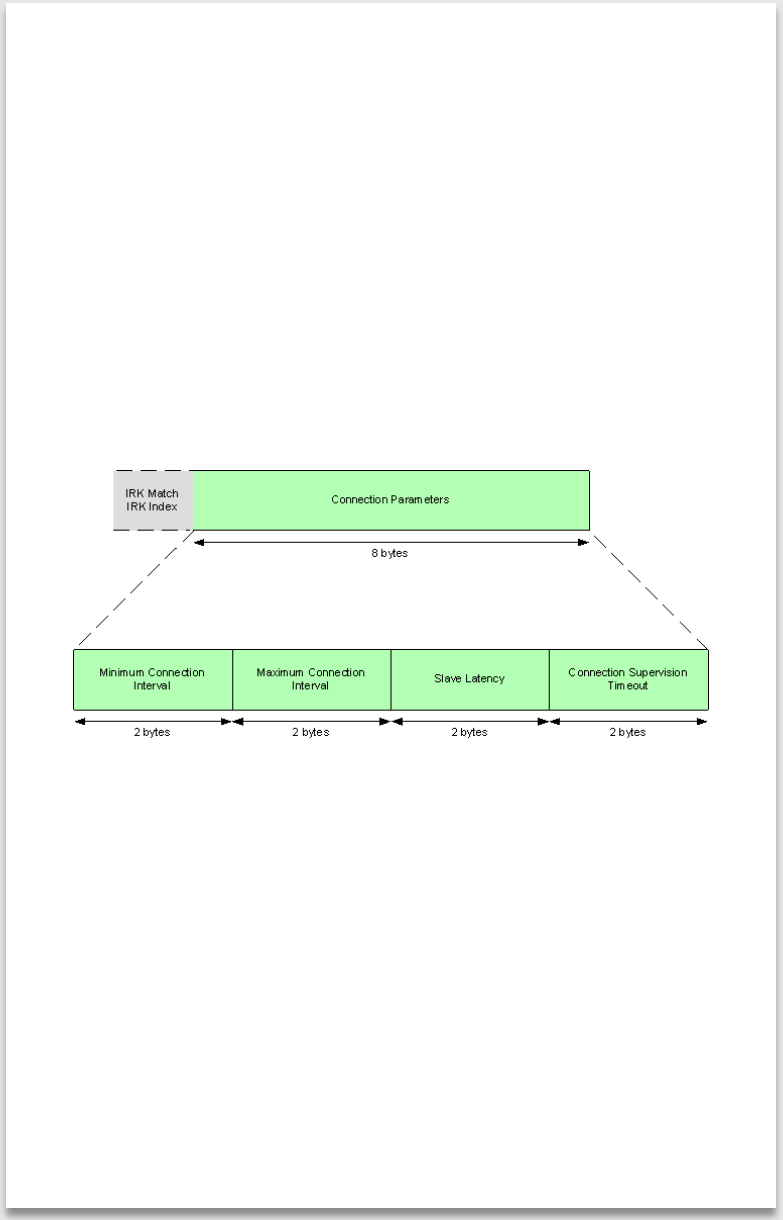
- Bluetooth **4.0** + (and Bluetooth Low Energy) is available on most major platforms:
  - iOS5+
  - Android 4.3+
  - Apple OS X 10.6+
  - Windows 8 +
  - GNU/Linux Vanilla BlueZ 4.93+

GAP and GAT for BLE

- Generic Access Profile (GAP) or Advertising
  - Information advertised to central before connection
  - Name of peripheral
  - Is it connectable?
  - Supported features (services)
- Generic Attribute Profile (GATT)
  - How to exchange data once connected
  - Identifies Services, Characteristics and Descriptors

Generic Access Profile (GAP)

- GAP responsible for “device visibility.”
- Determines how two devices can (or can't) interact with each other.



GAP Device Roles

- GAP defines 2 roles for devices:
  - Central
  - Peripheral
- **Peripheral** devices are small, low power, resource constrained devices
  - Connect to a powerful central device.
- **Central** devices are usually far more processing power and memory.
  - Tablet, Mobile phone, laptop

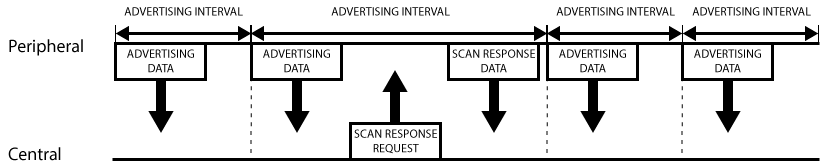
GAP - Advertising and Scan Response Data

- 2 ways for a device to advertise with GAP
  - *Advertising Data* payload
  - *Scan Response* payload.
- Advertising data payload is mandatory
- Scan response payload is optional
  - allows device designers to fit more information in the advertising payload such a strings for a device name, etc.

- Advertising Process

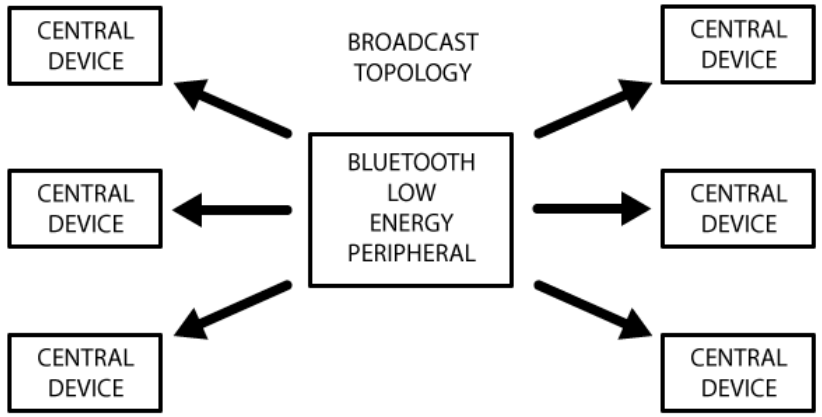
Advertising process

- Peripheral sets a specific advertising interval and transmit advertising packet
  - longer delays saves power but less responsive
- A listening device interested in the scan response payload can optionally request the scan response payload, and the peripheral will respond with the additional data.



Broadcast Network Topology

- Some devices/apps only require advertise data.
  - E.g. app requires peripheral to send data to more than one device at a time.
- Can include small amount of custom data in **31** byte advertising or scan response payloads.
- In this way, BLE peripheral can send data one-way to any devices in listening range



Bluetooth Connection

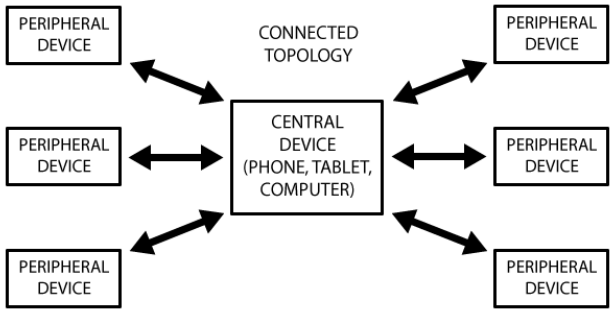
- Establishing a connection between a peripheral and a central device results in 1 to 1 communiction
  - the advertising process will stop
  - no longer be able to send advertising packets
- Communiction in both directions
- Must use GATT services and characteristics to communicate

Generic Attribute Profile - GATT

- Defines the way that two Bluetooth Low Energy devices transfer data back and forth
- Uses **Services** and **Characteristics**.
- With GATT, a BLE peripheral can only be connected to one central device (e.g. a mobile phone, etc.)

BLE Network Topology

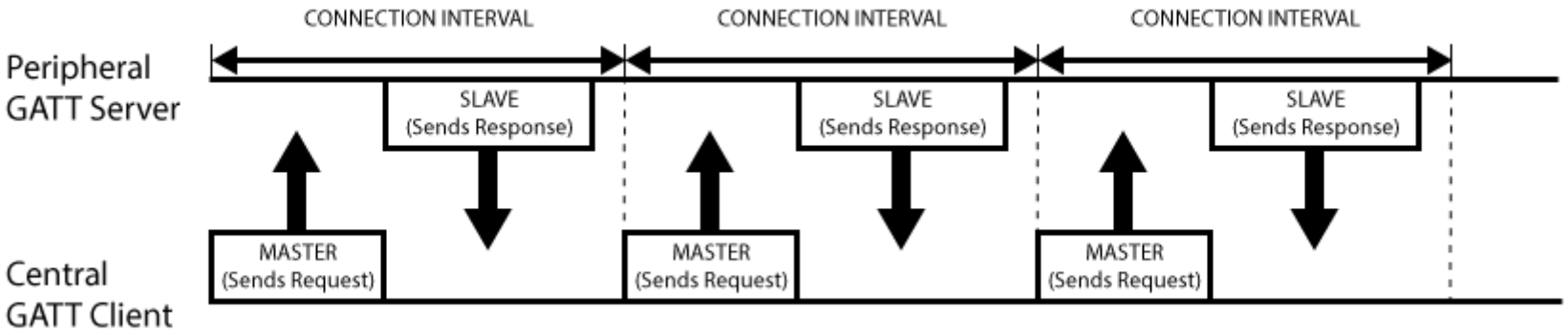
- A peripheral can only be connected to one central device
- Ccommunication is **2** -way
- Central device can be connected to multiple peripherals.
- If data needs to be exchanged between two peripherals, a custom messaging system will need to be implemented
  - all messages pass through the central device.



GATT transactions

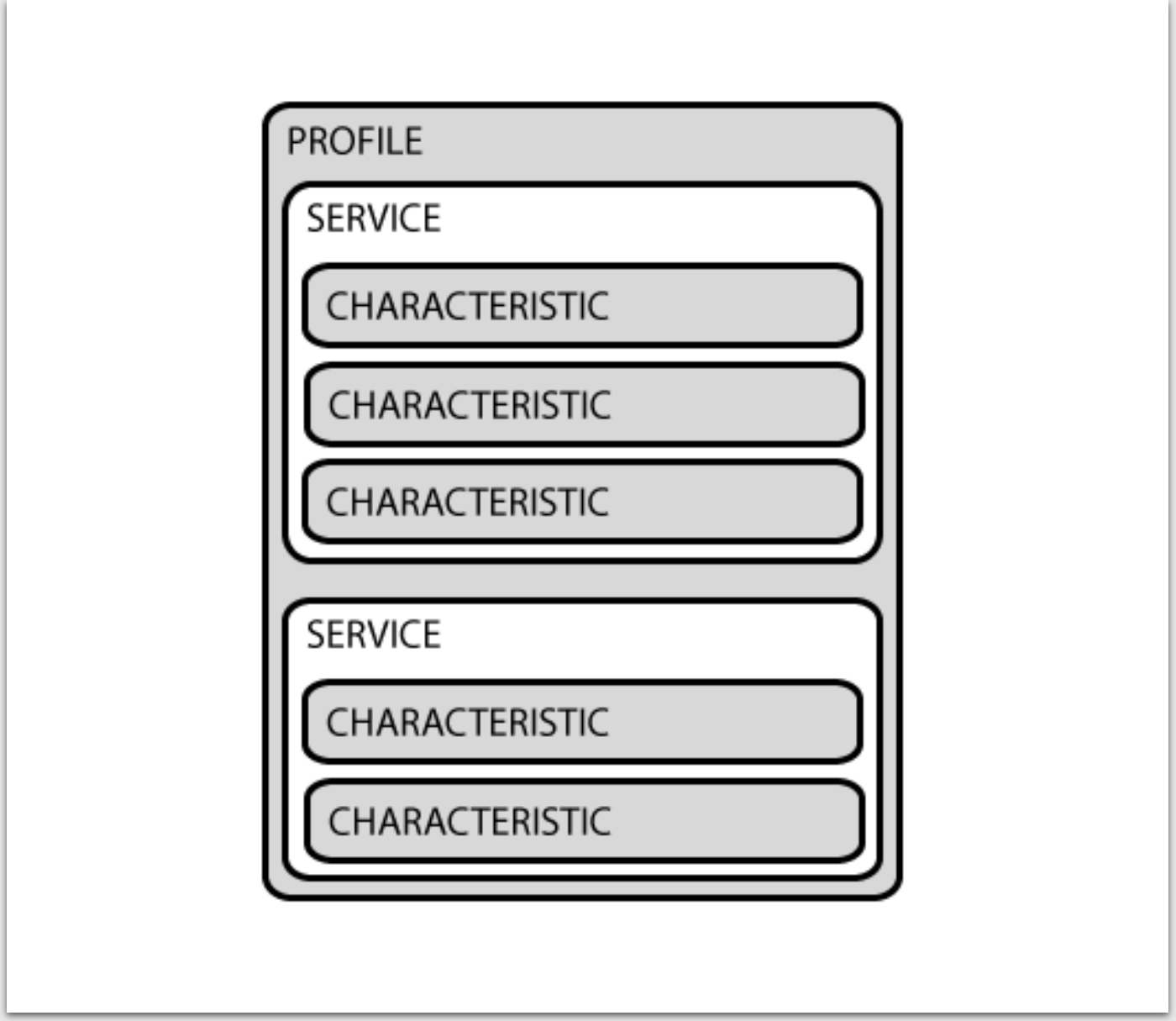
- Follows a (strange) client server relationship
- Peripheral is known as the **GATT Server**
- The **GATT Client** (the phone/tablet), which sends requests to this server.
- Follows master/slave approach
  - All transactions are started by the master device, the **GATT Client**
  - GATT client receives response from the slave device, **the GATT Server**.
- 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,

GATT Transactions



GATT – Services and Characteristics

- GATT transactions are based on Profiles, Services and Characteristics



GATT Profile

- A pre-defined collection of Services that has been compiled by either the Bluetooth Special Interest Group or by the peripheral designers.
  - E.g. The Heart Rate Profile
  - combines the Heart Rate Service and the Device Information Service.
- Complete list of GATT-based profiles can be found here: Profiles Overview

GATT Services

- Breaks data up into logic entities
- Contain one or more **characteristics**
- Each service distinguished by unique numeric ID called a UUID
  - 16 bit
- Set of officially adopted BLE services can be seen on the Services
- E.g. official Heart rate service
  - service has a 16-bit UUID of 0x180D
  - contains up to 3 characteristic
    - ***Heart Rate Measurement***, ***Body Sensor Location*** and ***Heart Rate Control Point***.

GATT Characteristics

- Represents a single data point
- Similarly to Services, each Characteristic distinguishes itself via a pre-defined UUID
  - Also use standard characteristics defined by the Bluetooth SIG
- E.G Heart Rate:
  - the Heart Rate Measurement characteristic is mandatory for the Heart Rate Service
  - Heart rate measurement has UUID of 0x2A37
- If you write apps that use BLE, **characteristics** are what you will be after with your BLE peripheral

Sources

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