# 6.8 The Bluetooth Protocol Stack

The Bluetooth protocol stack is a software framework that defines how Bluetooth devices communicate. It is structured in layers, where each layer provides services to the layer above it and uses services from the layer below it. The stack is divided into two main components: the **Controller** and the **Host**.

## The Controller

The Controller is responsible for the low-level radio operations. It handles the transmission and reception of radio signals and manages the physical connection between devices. It is often implemented as a dedicated chip (a "System-on-a-Chip" or SoC) that includes the radio hardware.

- **Physical Layer (PHY):** This is the actual radio hardware that transmits and receives signals in the 2.4 GHz band. Bluetooth 5 introduced multiple PHY options for BLE:
  - **LE 1M PHY:** The original 1 Mbps PHY, providing a balance of range and speed.
  - **LE 2M PHY:** A 2 Mbps PHY that doubles the speed at the cost of slightly reduced range.
  - **LE Coded PHY:** A long-range PHY that uses error correction to significantly increase range (up to 4x), but with lower data rates.
- **Link Layer (LL):** This is the core of the Controller. It manages the state of the radio (advertising, scanning, initiating, connected) and defines the fundamental device roles in BLE:
  - **Advertiser/Broadcaster:** A device sending out advertising packets.
  - **Scanner/Observer:** A device listening for advertising packets.
  - **Master/Central:** A device that initiates and manages a connection.
  - **Slave/Peripheral:** A device that accepts a connection request.

## The Host

The Host is responsible for the high-level logic, data organization, and application functionality. It typically runs on the main processor of a device (e.g., in your ESP32 code).

- **Host-Controller Interface (HCI):** A standardized protocol that allows the Host and Controller to communicate. This standard interface means a Host from one manufacturer can work with a Controller from another.
- **Logical Link Control and Adaptation Protocol (L2CAP):** This layer acts as a multiplexer. It takes data from the upper layers and prepares it for transmission by the Link Layer.
- **Security Manager (SM):** Manages the entire security process, including pairing, key distribution, and encryption.

- **Attribute Protocol (ATT):** Defines a simple client-server protocol for data exchange. The server holds a set of data called "attributes," and the client can read or write these attributes.
- **Generic Attribute Profile (GATT):** This is the most critical layer for application developers. GATT provides a structured way to organize and exchange data based on the ATT protocol. It defines the hierarchy of data:
  - **Profile:** A collection of services for a specific use case (e.g., a "Heart Rate Profile").
  - **Service:** A collection of related data points, identified by a unique number called a **UUID**. A service can be official (e.g., "Heart Rate Service") or custom.
  - **Characteristic:** A single data point or value, also identified by a UUID (e.g., "Heart Rate Measurement"). This is what your application will read from or write to.
  - **Descriptor:** Provides additional information about a characteristic.
- **Generic Access Profile (GAP):** This profile defines how devices interact with the outside world. GAP is responsible for:
  - **Device Discovery:** How a device makes itself known (advertising) and finds other devices (scanning).
  - **Connection Management:** How connections are established and terminated.
  - **Security:** Defining the security model for a device.

---

Revision #1
Created 2025-08-28 12:19:41 UTC by GI
Updated 2025-08-28 12:19:56 UTC by GI