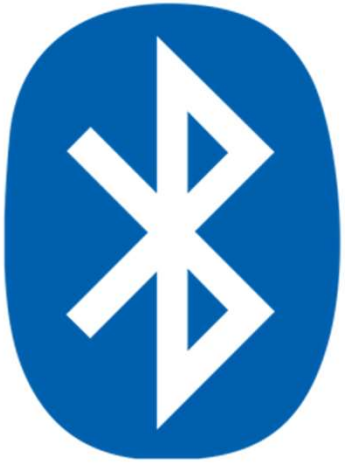


Bluetooth Low Energy

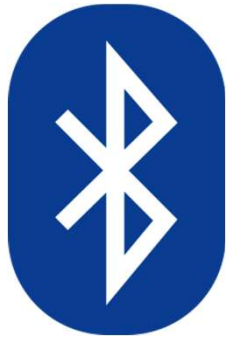
Crash course in the Bluetooth Low Energy protocol

Andreas Haugland

NTNU

February 2024

Bluetooth Low Energy



- Wireless Personal Area Network technology
- Operates in the 2.4GHz ISM (Industrial, Scientific and Medical) band
- Specification defined by the Bluetooth SIG (Special Interest Group)
- Optimized for low-power consumption
  - Racing to idle
  - Low range*
  - Low bandwidth*

Bluetooth modes

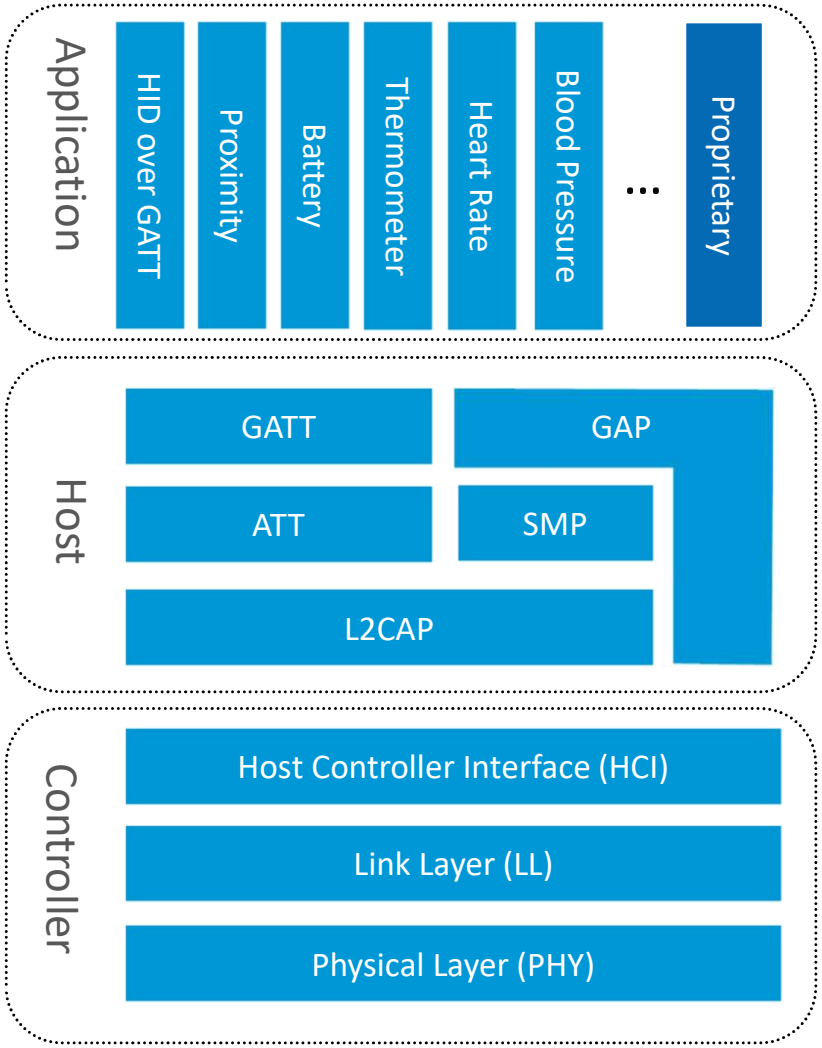
Term	Introduced	Means
BR	1.1 (2002)	Basic Rate (1 Mbit/s)
EDR	2.0 (2004)	Enhanced Data Rate (2 and 3 Mbit/s)
HS	3.0 (2009)	High Speed (up to 24 Mbit/s)
LE	**4.0 (2010)**	Low Energy (1 Mbit/s ultra low power)

Bluetooth modes

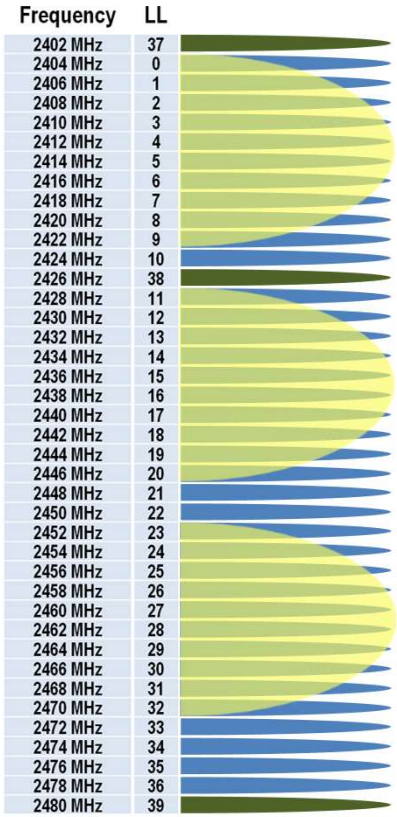
Term	Introduced	Means
BR	1.1 (2002)	Basic Rate (1 Mbit/s)
EDR	2.0 (2004)	Enhanced Data Rate (2 and 3 Mbit/s)
HS	3.0 (2009)	High Speed (up to 24 Mbit/s)
LE	**4.0 (2010)**	Low Energy (1 Mbit/s ultra low power)
	4.1 (2013)	L2CAP CoC, Low Duty Cycle directed adv., multi role, Privacy 1.1, ++
	4.2 (2014)	LESC, DLE, Privacy 1.2, ++
	5.0 (2016)	High Speed(2 Mbit/s), Long Range, LE Adv. Extensions, LE Channel Selection Algorithm #2, Increased max power, ++

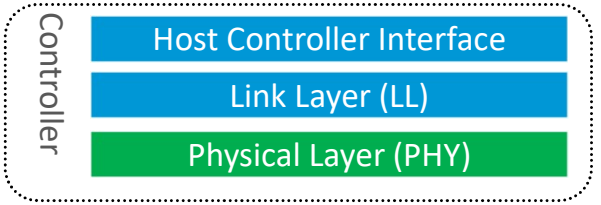
Bluetooth LE Architecture

- Split into three main building blocks
  - Application
    - User application interfacing with the Bluetooth protocol stack
  - Host
    - Upper layers of the Bluetooth protocol stack
  - Controller
    - Low layers of the Bluetooth protocol stack, including the radio



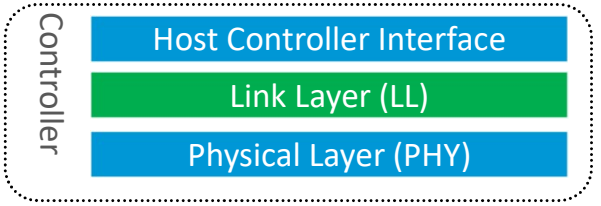
Physical Layer(PHY)





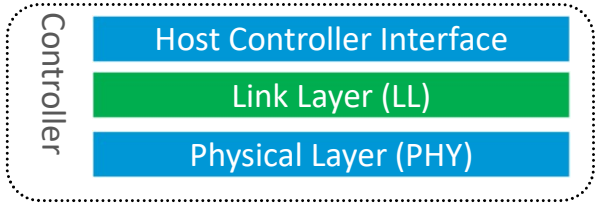
- 2.4 GHz ISM band
- Divided into 40 channels from 2.400GHz to 2.4835GHz
- Frequency Hopping Spread Spectrum(FHSS)
- Channel 37, 38 and 39 are used for advertising
- Remaining channels used during connections

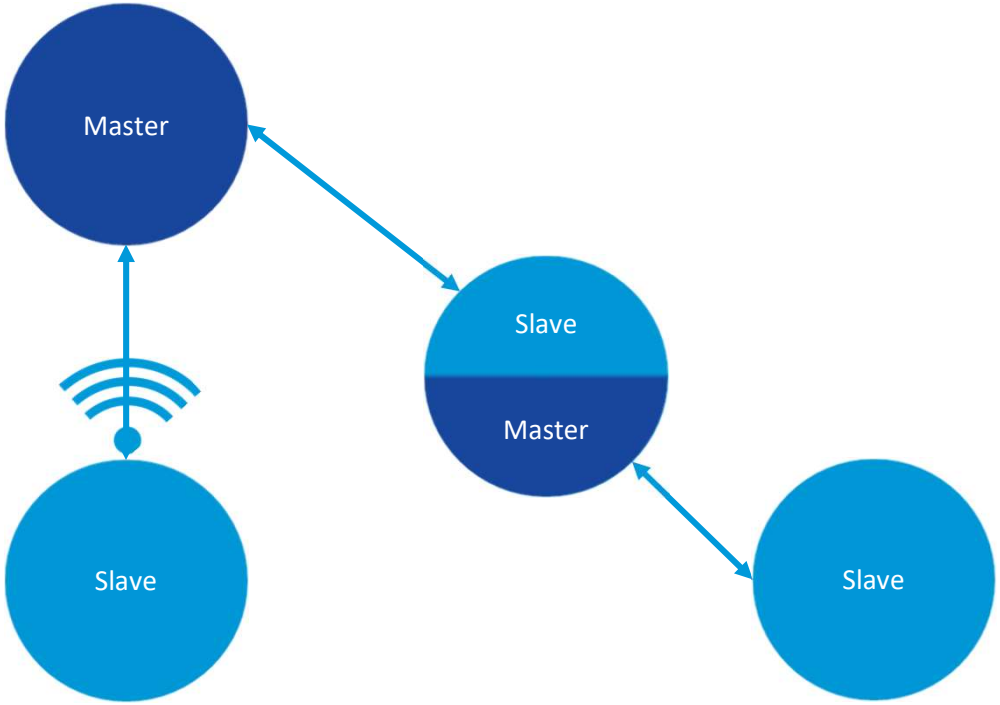
Link Layer (LL)

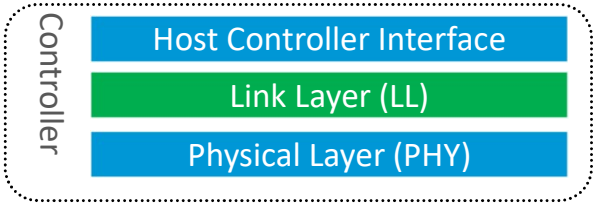


- Combination of hardware and software.
- Interfaces directly with the PHY layer and responsible for:
  - Framing the data from the upper layers in a link layer packet(BLE packet).
  - Meeting the timing requirements of the BLE specification.
  - Encryption of the link
  - CRC generation and verification
- Defines the following roles:
  - **Advertiser**: A device sending advertisement packets
  - **Scanner**: A device scanning for advertisement packets
  - **Master**: A device that initiates a connection and manages it later
  - **Slave**: A device that accepts a connection request and follows the master’s timing

Network topology



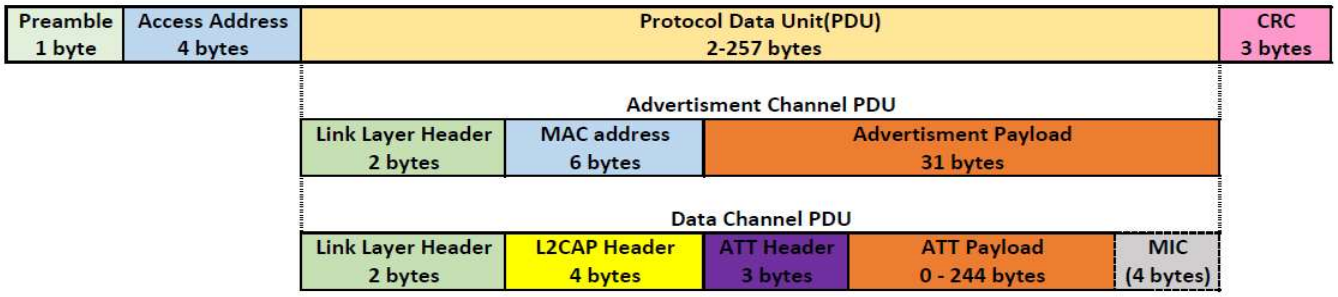




BLE Packet

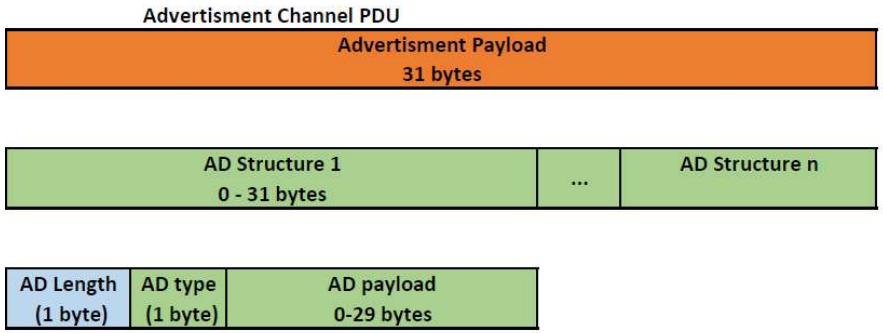
BLE has one packet structure and two packet types:

- Advertisement packets
  - BT 4.0: (31 byte payload + optional 31 byte in scan response packet)
  - BT 5.0: (255 byte payload with advertising extensions)
- Data packets
  - BT 4.0: (27 byte payload, only 20 byte available for user data)
  - BT 4.2: (251 byte payload, 244 byte available for user data)

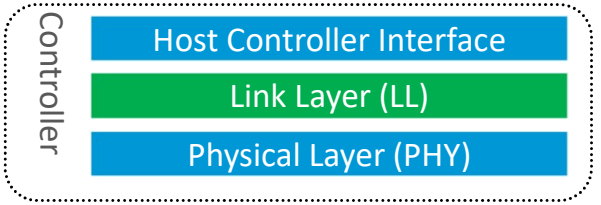


Advertisement Packet

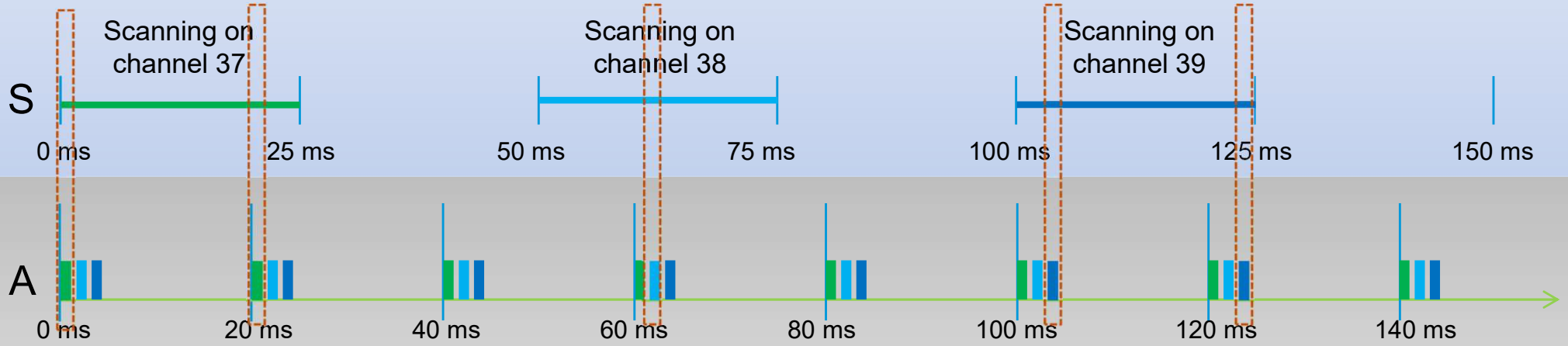
- Advertisement payload contains one or more advertisement structures.
- Each structure consists of:
  - AD Length: AD type + AD Payload in bytes
  - AD Type: The type of the payload data.
  - AD Payload: The payload data.
- Commonly used AD types
  - 0x01 - Flags
  - 0x02 - Incomplete List of 16-bit Service Class UUID
  - 0x06 - Incomplete List of 128-bit Service Class UUIDs
  - 0x08 - Shortened Local Name
  - 0x09 - Complete Local Name
  - 0xFF - Manufacturer Specific Data



Advertising & Scanning

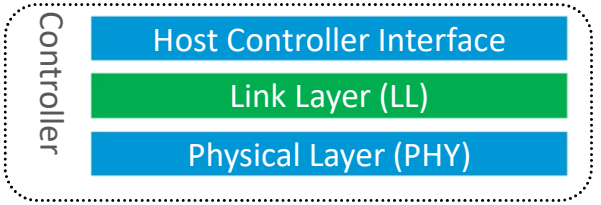


Scanner scan interval = 50 ms
Scanner scan window = 25 ms



Advertising on **37**, **38** and **39**

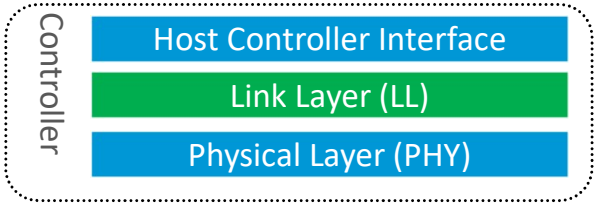
Advertiser advertising, interval = 20 ms



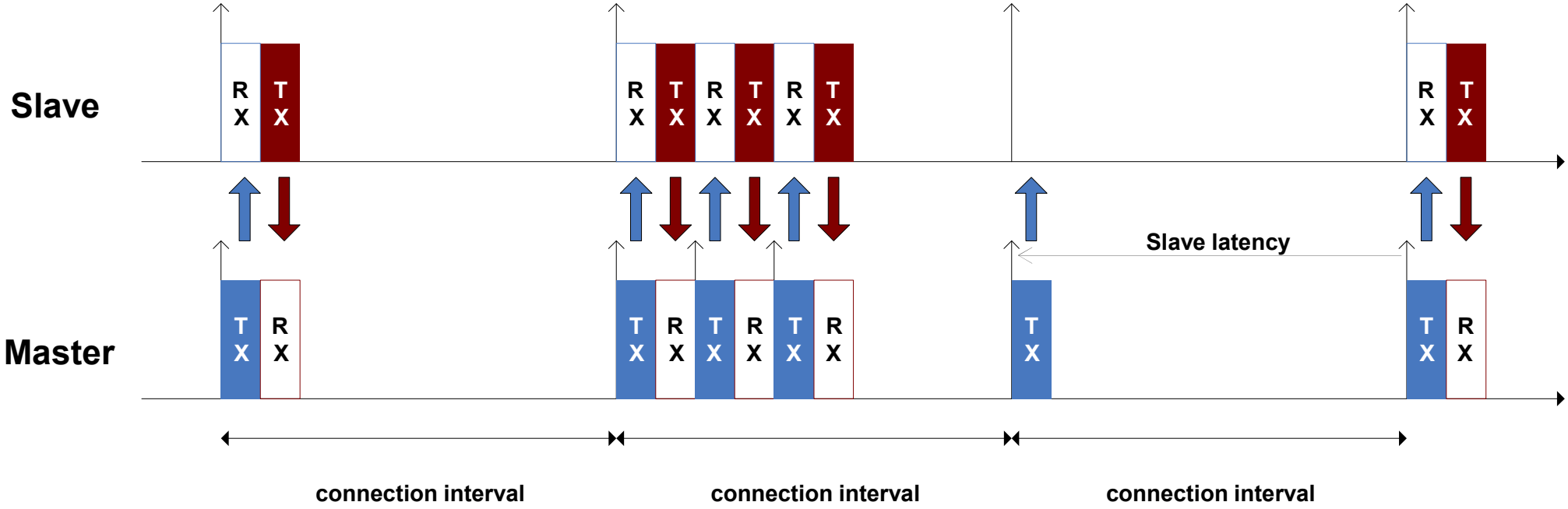
Connection Establishment

- Master starts scanning for advertising devices that are connectable
- Master initiates the connection process based on data in advertisment packet
- When a suitable advertising device is found, the master sends a connection request packet containing the following:
  - **Frequency hop increment**: *Which channel that should be used for the next connection event*
  - **Connection interval:** *The time between two consecutive connection events*
  - **Slave latency:** *The number of connection events that a slave can choose to skip with out risking disconnecting*
  - **Connection supervision timeout:** *Length of time the master will wait for a response from the slave before the connection is terminated.*

Connection

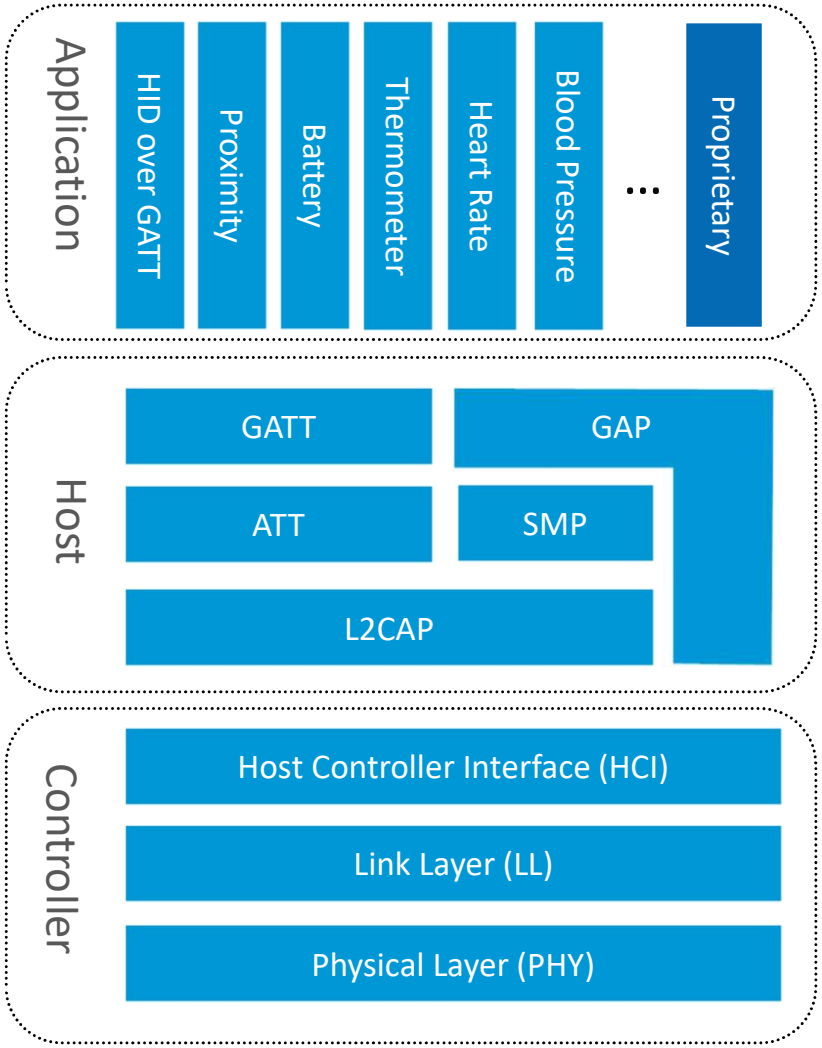


- Master sends first, slave responds
- Multiple data packets can be sent per connection event occurring at each connection interval
- Connection interval can be from 7.5 ms to 4 seconds

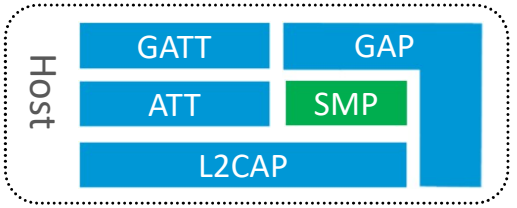


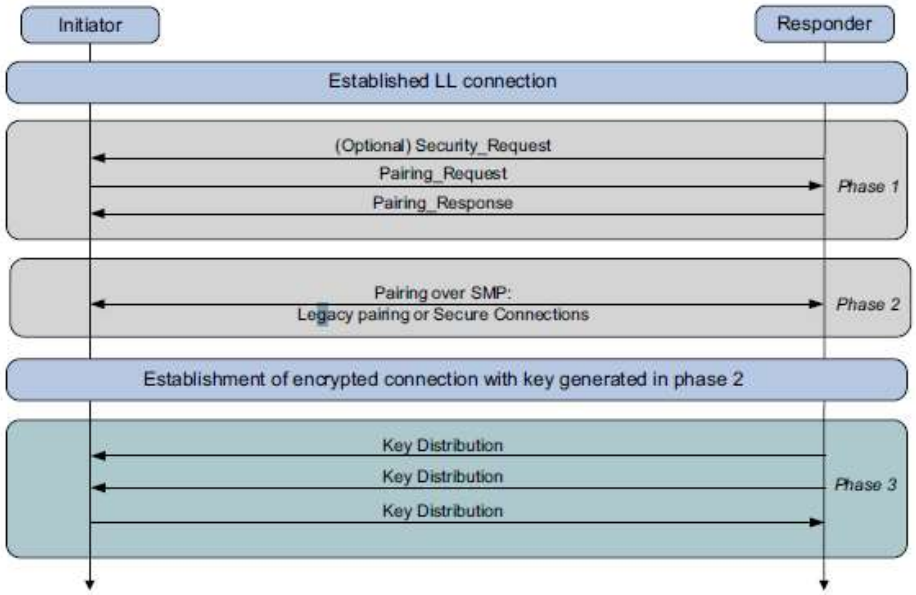
Bluetooth LE Architecture

- Split into three main building blocks
  - Application
    - User application interfacing with the Bluetooth protocol stack
  - Host
    - Upper layers of the Bluetooth protocol stack
  - Controller
    - Low layers of the Bluetooth protocol stack, including the radio



Bluetooth Paring and Bonding





Pairing (Phase 1 & 2)

- Temporary security encryption key, the Short Term Key(STK) is generated and used to encrypt the link

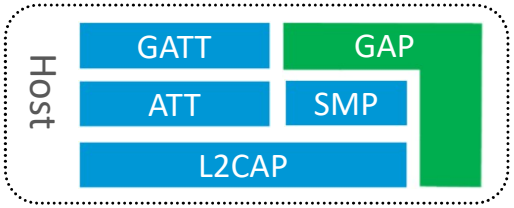
Bonding (Phase 1 & 2 + Phase 3)

- A sequence of pairing followed by the generation and exchange of a Long Term Key(LTK)

Encryption re-establishment

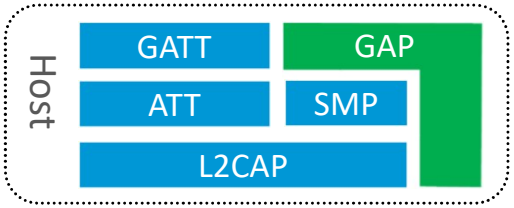
- Long Term Key(LTK) used to re-encrypt connection at a later time

Generic Access Profile(GAP)



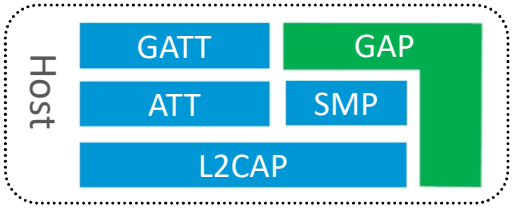
- Defines procedures on how devices discover, connect and present information to each other.
- Roles
  - **Peripheral** (Link Layer Slave)
  - **Central** (Link Layer Master)
  - **Broadcaster** (Link Layer advertiser)
  - **Observer** (Link Layer scanner)
- Security
  - Pairing and creating bonds with peer devices
  - Attribute access security requirements
  - Privacy and address control

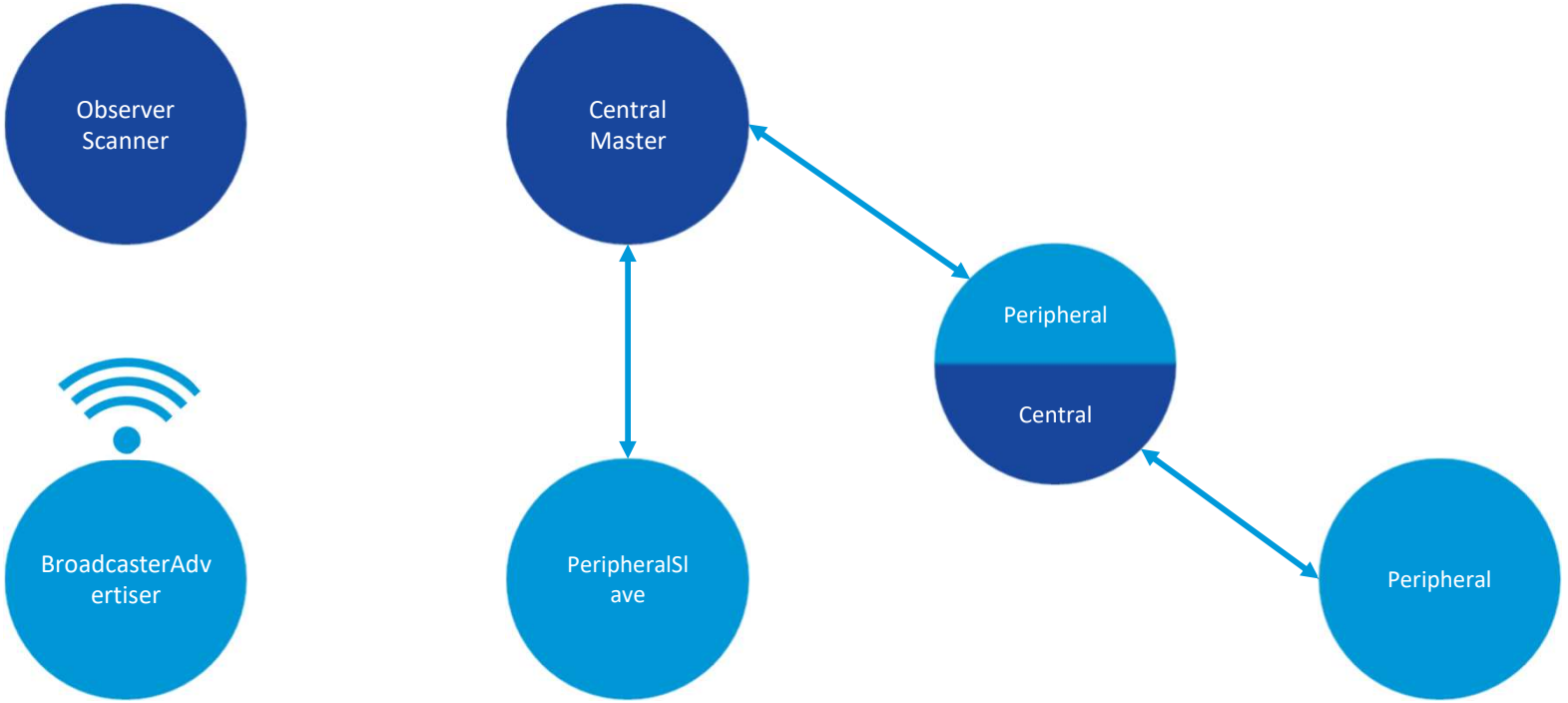
GAP Operations



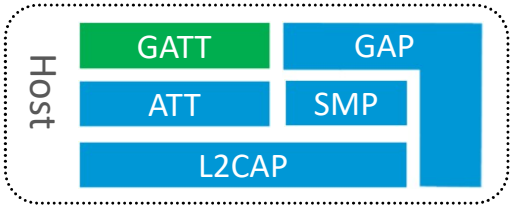
- Set advertisement data
- Start/stop advertising
- Start/stop scanning
- Connect/disconnect to a device
- Update Connection Parameters
- Encrypt link

Network topology





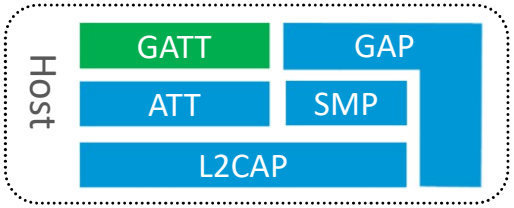
Generic Attribute Profile (GATT)

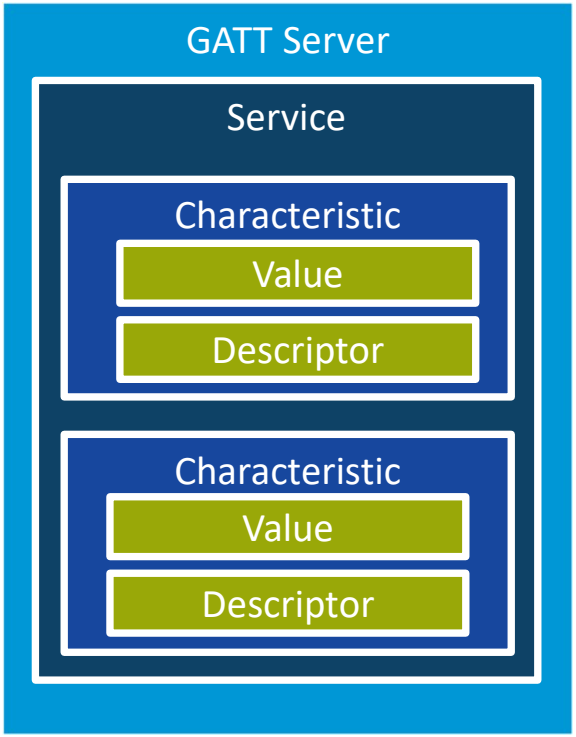


- Defines procedures on how to exchange data over a BLE connection
- Uses the Attribute Protocol(ATT) as its transport layer
- Uses a client-server architecture, like the Attribute Protocol(ATT)
- Attributes organized hierarchically
  - Services
  - Characteristics
  - Descriptors

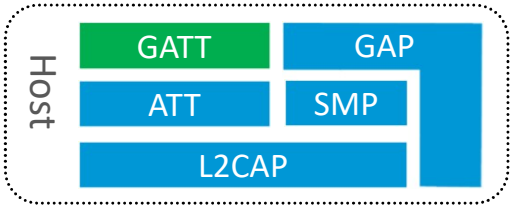
Services, Characteristics and Descriptors

- Services
  - Group of characteristics
- Characteristic
  - Data containers, e.g. temperature, battery voltage
- Descriptor
  - Additional meta-data of the characteristic, e.g. notifications and indications





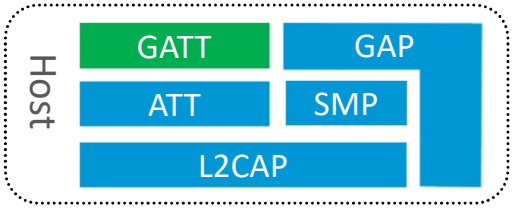
Services, Characteristics and Descriptors cont'd



Services, characteristics and descriptors all have an UUID

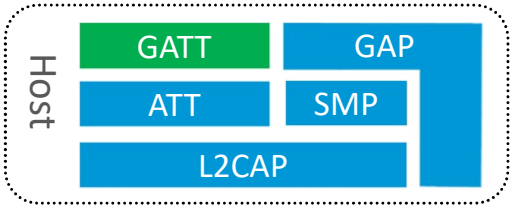
- UUID – Universal Unique Identifier
  - 16-bit short UUID using Bluetooth Base 0000XXXX-0000-1000-8000-00805F9B34FB
  - 128-bit UUID
- In addition Characteristics and Descriptors have
  - Permissions – Read, Write.
  - Value – data that can be read/written by client

GATT Operations

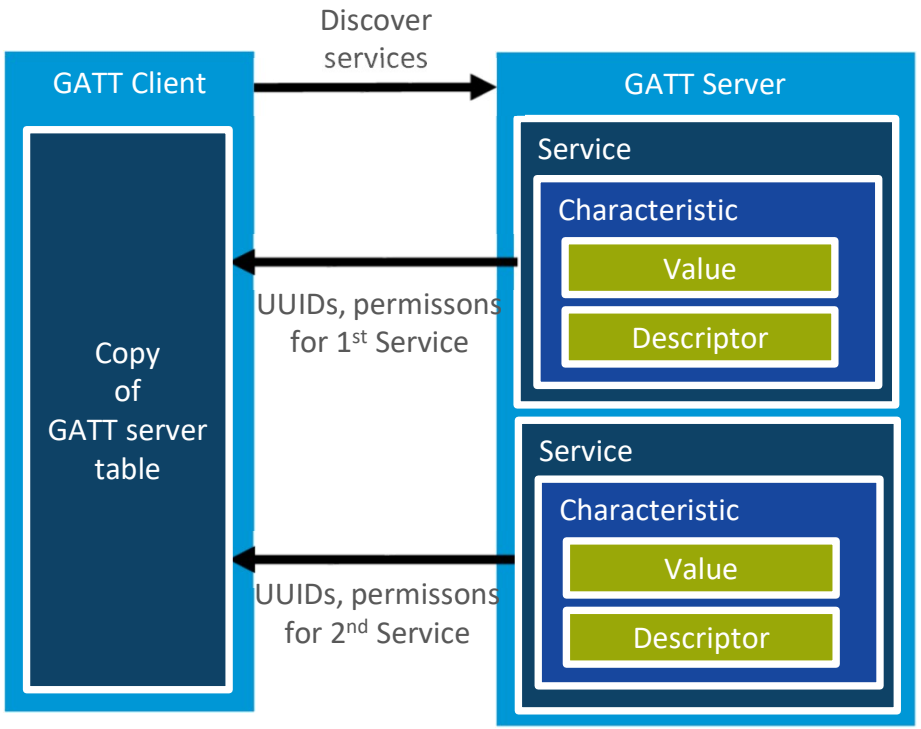


- Service and Characteristic Discovery
- Writing to Characteristics and Descriptors
- Reading from Characteristics and Descriptors
- Server initiated updates
  - Characteristic Value Notification:
  - Characteristic Value Indication:

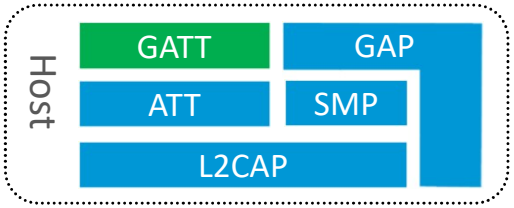
Service Discovery

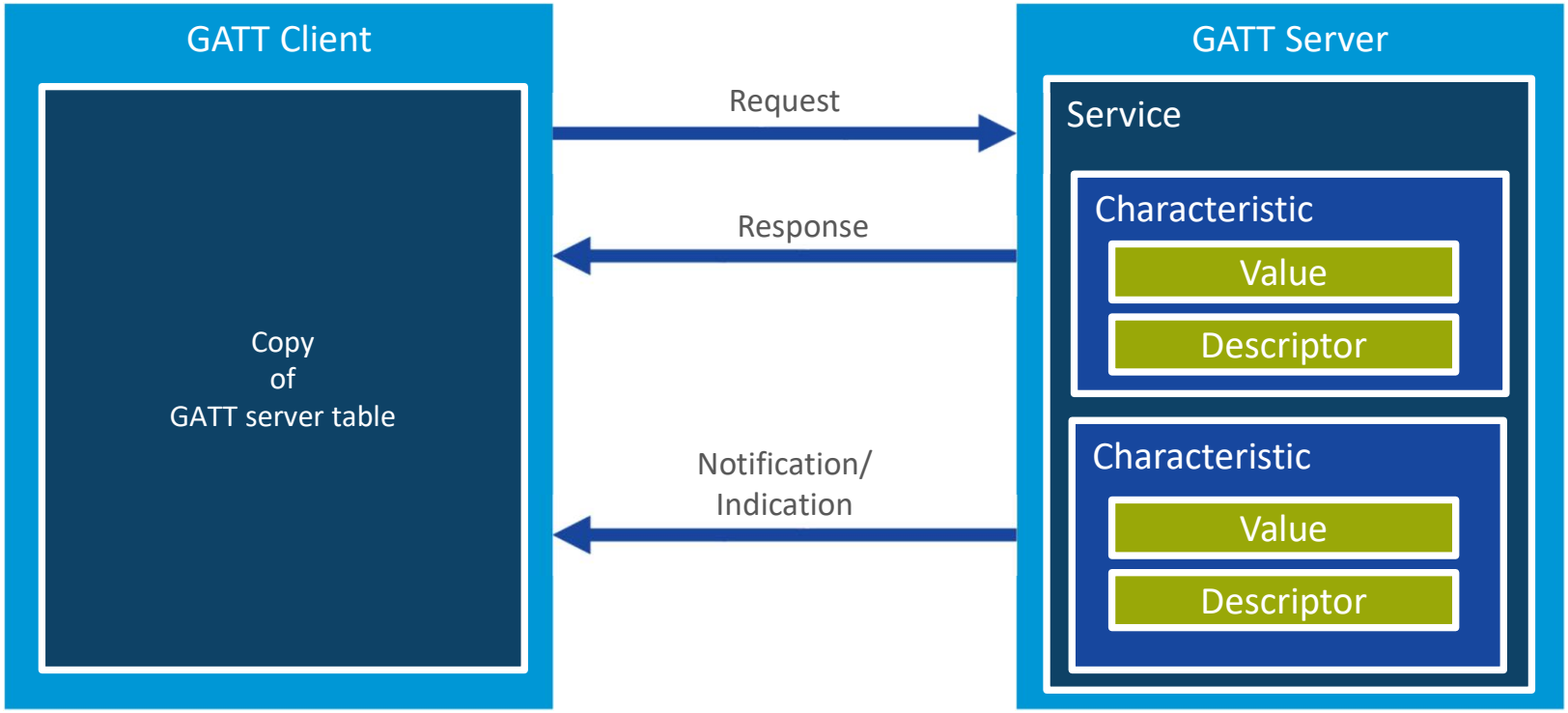


- Client does not know which services/characteristics/descriptors the server has implemented.
- Performs Service discovery and stores copy of the GATT server structure.
- Uses copy as a lookup table when reading/writing to characteristics or descriptors

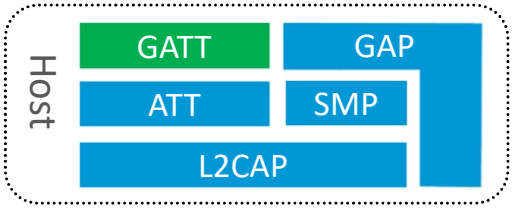


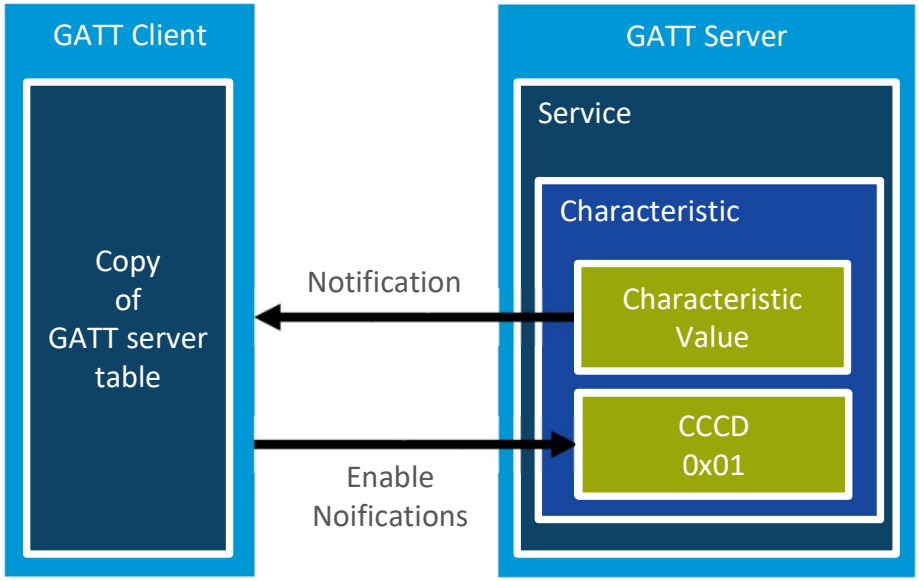
GATT Transaction





Client Characteristic Configuration Descriptor (CCCD)

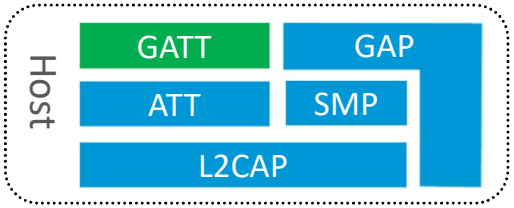




- Notification and indication switch
- Two-bit field value
  - 0x01: Enables Notifications
  - 0x02: Enables Indications
- Client can turn on/off

Notifications/Indications by writing to the CCCD.

GATT table



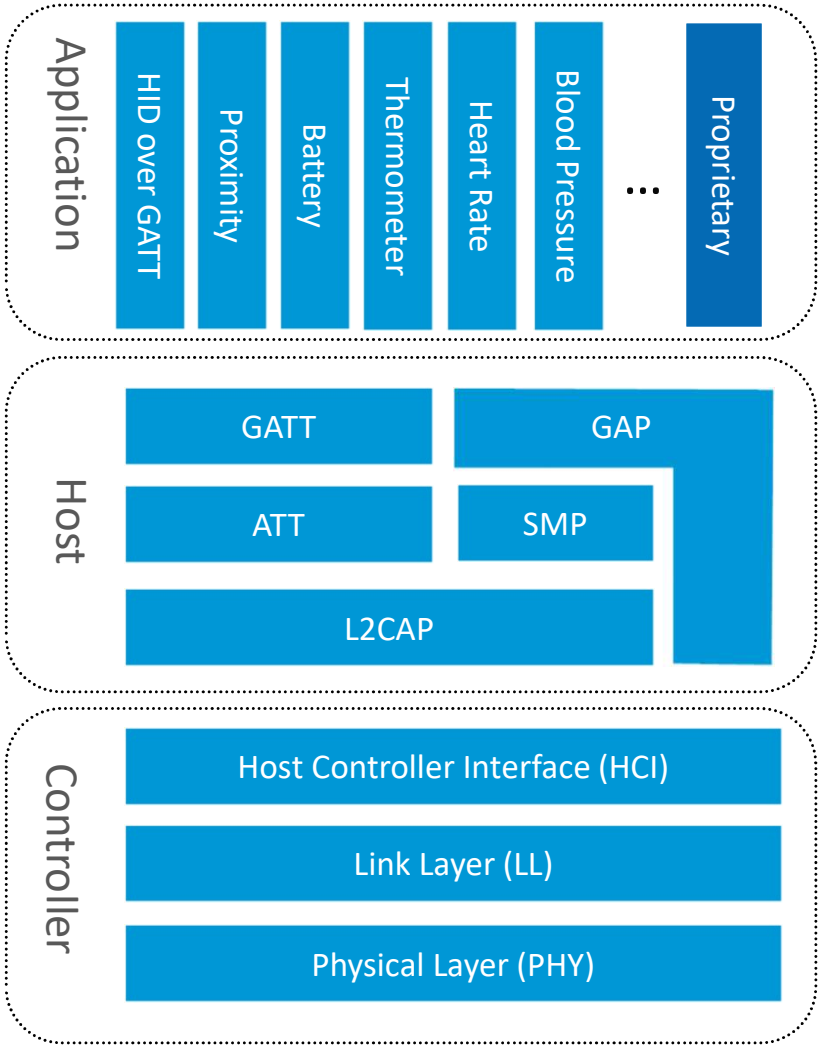
Handle	UUID (Type)	Value (Type)
0x0001	0x2800 (Service)	0x1800 (GAP Service)
0x0002	0x2803 (Characteristic)	{0x0A, 0x0003, 0x2A00}
0x0003	0x2A00 (Device Name)	“Example Device”
0x0010	0x2800 (Service)	0x1801 (GATT Service)
0x0100	0x2800 (Service)	0x180A (Battery State Service)
0x0101	0x2803 (Characteristic)	{0x02, 0x0102, 0x2A19}
0x0102	0x2A19 (Battery Level)	0x04
0x0200	0x2800 (Service)	0x5AB20001-B355-4D8A-96EF-2963812DD0B8
0x0201	0x2803 (Characteristic)	{0x12, 0x0202, 0x5AB2FF01-B355-4D8A-96EF-2963812DD0B8}
0x0202	0x5AB2FF01-B355-4D8A-96EF-2963812DD0B8 (Proprietary Temperature Characteristic)	0x028A
0x0203	0x2904 (Characteristic Format)	{0x0E, 0xFE, «Celsius», «Outside»}
0x0204	0x2901 (Characteristic User Description)	“Outside Temperature”
0x0205	0x2902 (Client Characteristic Configuration Descriptor)	0x0000

Battery State Service

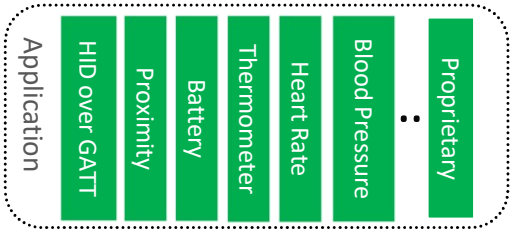
Proprietary Thermometer Service

Bluetooth LE Architecture

- Split into three main building blocks
  - Application
    - User application interfacing with the Bluetooth protocol stack
  - Host
    - Upper layers of the Bluetooth protocol stack
  - Controller
    - Low layers of the Bluetooth protocol stack, including the radio

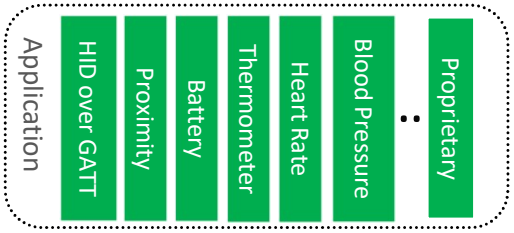


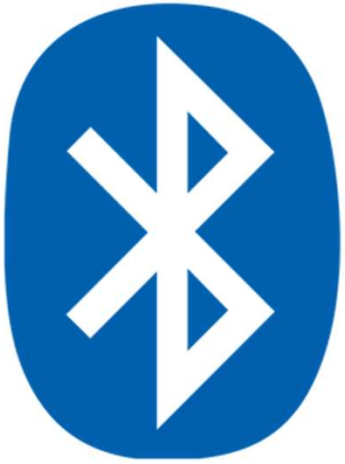
Application



- Profile:
  - Collection of services
  - Selects required features from GAP and GATT
- Use-case specific profiles
  - Bluetooth SIG defined, e.g. Heart Rate Profile (Heart Rate Service + Device Information Service)
  - Vendor-specific (proprietary), Apple iBeacon, Google Eddystone
- Key to interoperability

Application





Bluetooth Low Energy

Crash course in the Bluetooth Low Energy protocol

Andreas Haugland

NTNU

February 2024