Introduction to Bluetooth Low Energy

Qualcomm® Bluetooth® Low Energy

Terminology clarification

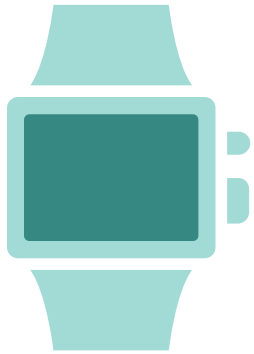
In this document you will notice a number of references are made to **Qualcomm Bluetooth Low Energy SDK.** While this **is** the official name of the SDK moving forward, within the actual SDK you will see it currently referred to as **CSR µEnergy**. In addition, many of the directories and folders may have the **CSR µEnergy SDK** naming convention. Please be assured these are the same thing.

Thank you for your understanding while we take the necessary steps to phase out the use of **CSR µEnergy** across our product lines.

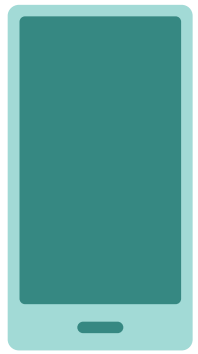
Introduction to Bluetooth Low Energy

Contents

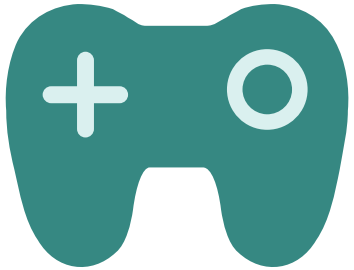
What is Bluetooth Smart?



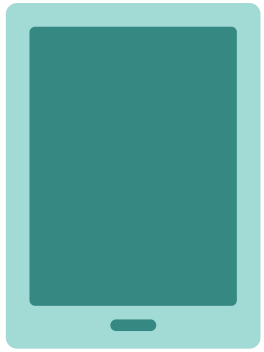
Bluetooth SMART



Bluetooth SMART READY



Bluetooth SMART



Bluetooth SMART READY

Basic Concepts

Dual-mode and Single-mode Radios

Dual-mode Radio

- Can support Bluetooth classic connections and Bluetooth low energy connections
- Typically for Master devices - Phones, PC, Tablet, PMP, TV
- Does not deliver power savings of Bluetooth Smart

Single-mode Radio

- Only supports Bluetooth low energy devices (cannot communicate with Bluetooth classic radios)
- Typically for sensors and peripherals (can act as master to a small number of sensors)
- Much lower power than Bluetooth Classic or dual mode radio

Basic concepts

Optimized for low power consumption

Low data rates

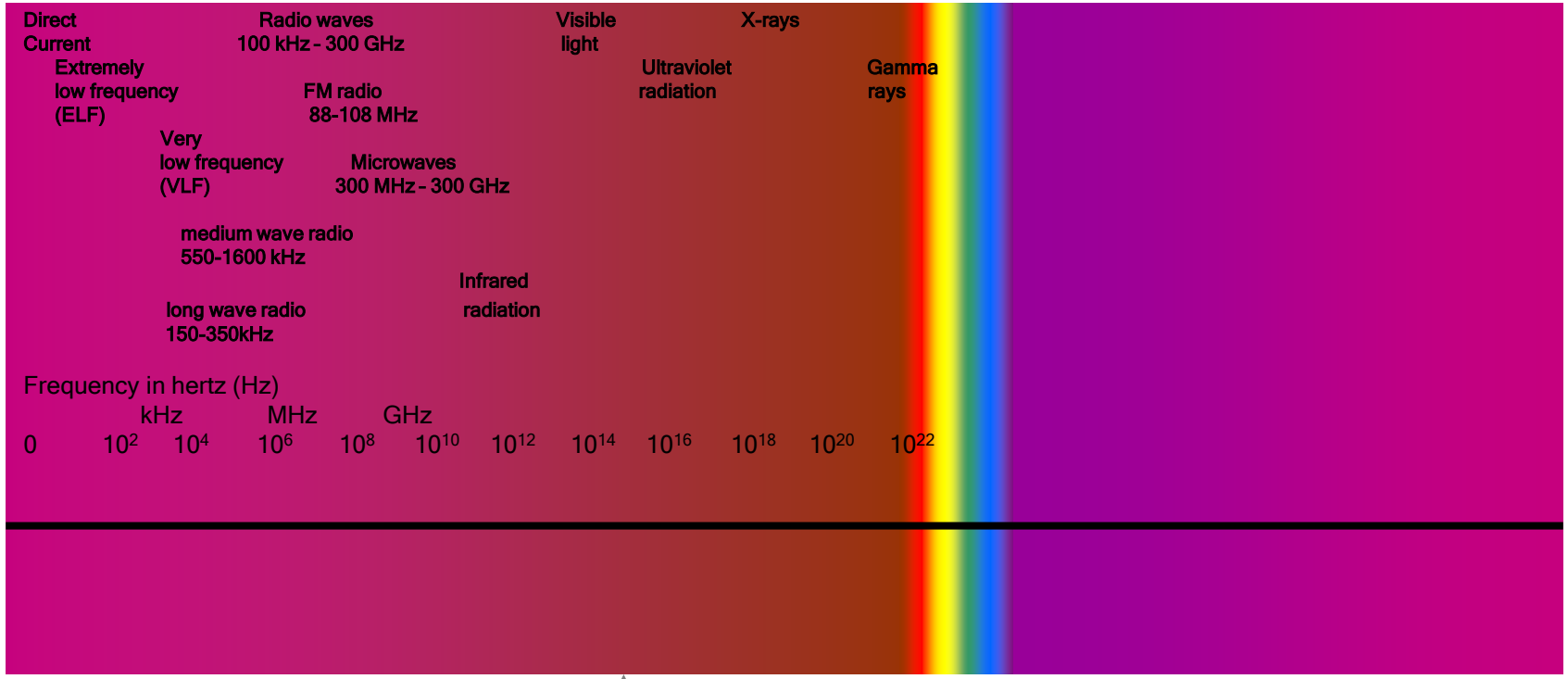
- Optimally less than 64kbps (up to 300kbps theoretical limit)
- Packets are short and the radio is off more

Simple profile structure - attributes

- Peripheral operates like a database
- Host devices can read or write attributes e.g. battery level
- Peripherals can inform host devices of a change e.g. “key pressed”

Spectrum usage

The 2.4GHz ISM band is a free for all for anyone who wants to use it



Bluetooth

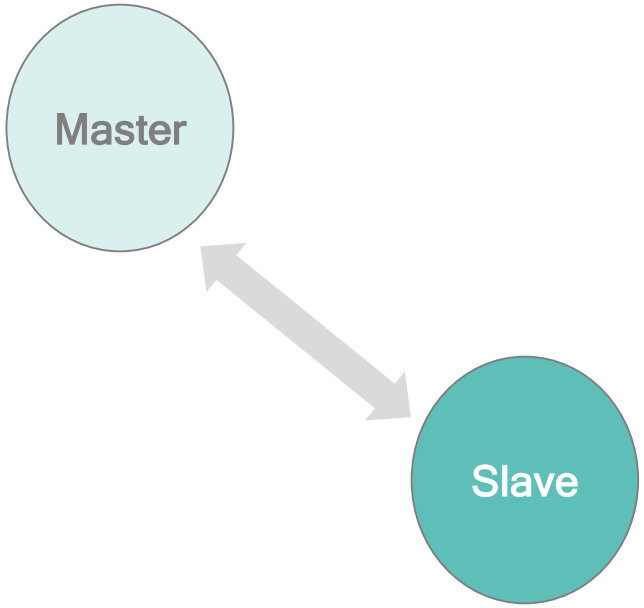
The 2.4GHz ISM Band is also used by:

- Microwave Ovens
- Digital Cordless Phones
- 802.11b/g

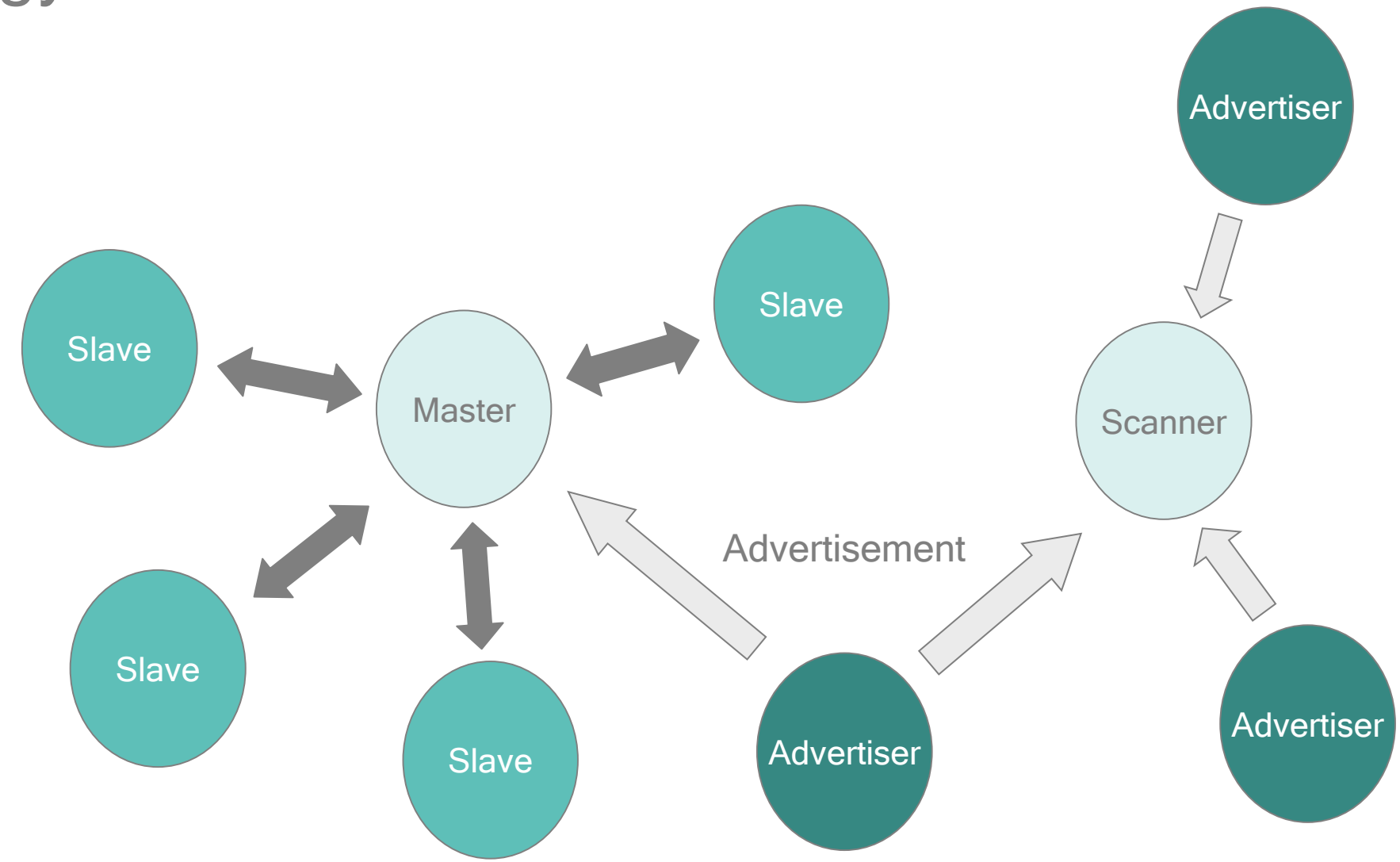
Bluetooth Low Energy

Bluetooth Smart

- Designed to run from Coin Cell Battery
- Optimised for low power, low cost
- Global Standard
- Interoperable
- Robust
- Low Duty Cycle
- Maximum length of packets is very short
  - Radio is off for longer periods
- Efficient Protocols
- Fast / Lazy acknowledgement both possible
- Very low latency from wake-up to communication complete (< 10 ms)
- Low data rates



Topology


Advertiser
Slave
Slave
Master
Scanner
Slave
Advertisement
Slave
Advertiser
Advertiser

Transmitter and Receiver Characteristics

Transmit output power

- -20dBm to +10dBm
- No concept of Class 1 / 2 / 3

Receive sensitivity

- -70dBm (-92dBm is expected performance)

Modulation Index

- 0.5 (compared to 0.25 for Bluetooth BR/EDR)
- Easier to distinguish “1” from “0”
- Frequency hopping not required by regulatory agencies
  - No frequency hopping in advertising/scanning
  - Frequency hopping only in connections
  - Frequency changes every packet (vs. every bit in Bluetooth classic)

Modulation Scheme

Data is transmitted using Gaussian Frequency Shift Keying (GFSK)

FSK uses two different frequencies to transmit a binary ‘1’ or ‘0’

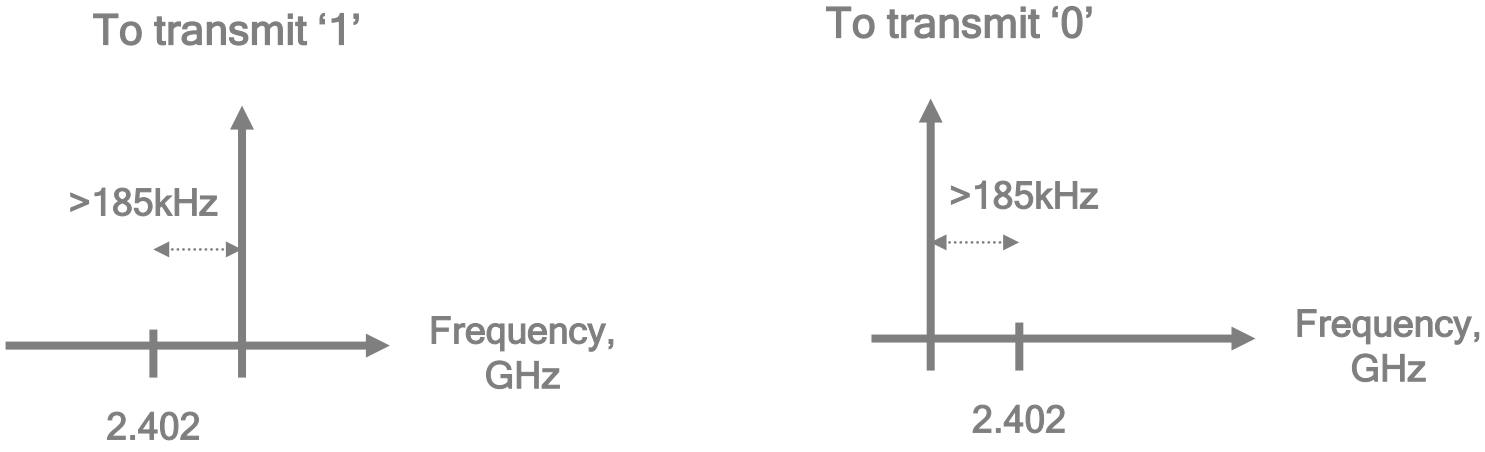
For Bluetooth low energy the two frequencies are:

- $fc + \Delta$ for ‘1’
- $fc - \Delta$ for ‘0’

where fc = frequency of current hop and Δ = >185kHz

Modulation example

For channel 0 (Frequency 2.402GHz)



During one time slot the data can change value every 1µs, so the transmit frequency oscillates back and forth around the centre channel frequency.

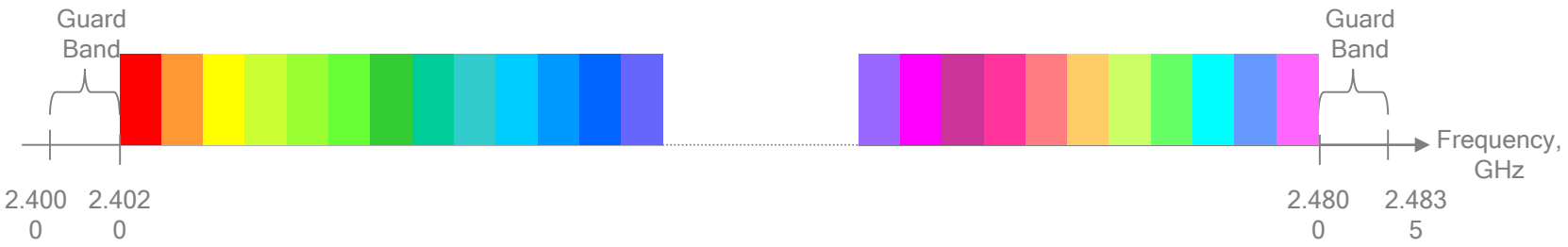
Frequency Hopping Spread Spectrum - FHSS

Frequency hopping occurs while in a connection

- Advertisements are transmitted on 1, 2 or 3 channels at the same time (configurable)

The frequency hops follow a hop-length that is pseudo-random per connection

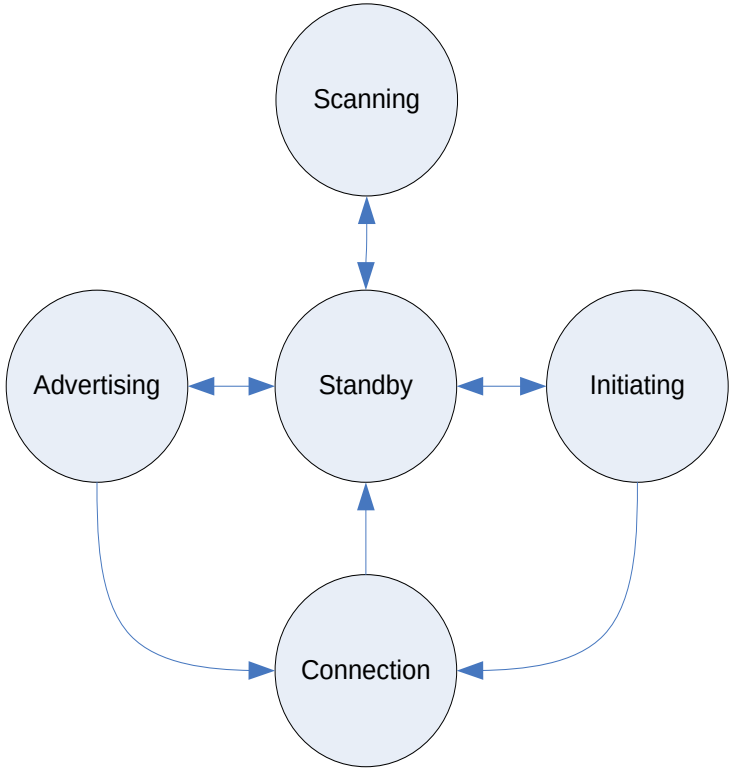
- Communicated in the “Connection Request”
- Provides instant adaptive frequency hopping capability
- Can be updated using a channel update message



Bluetooth Low Energy connections

Operating States and Roles

State		State Description
Standby		Does not transmit or receive packets
Advertising		Broadcasts advertisements in advertising channels
Scanning		Looks for advertisers
Initiating		Initiates connection to advertiser
Connection	Master	Communicates with device in the Slave role, defines timings of transmissions
	Slave	Communicates with device in Master Role



Establishing Bluetooth Low Energy Connection

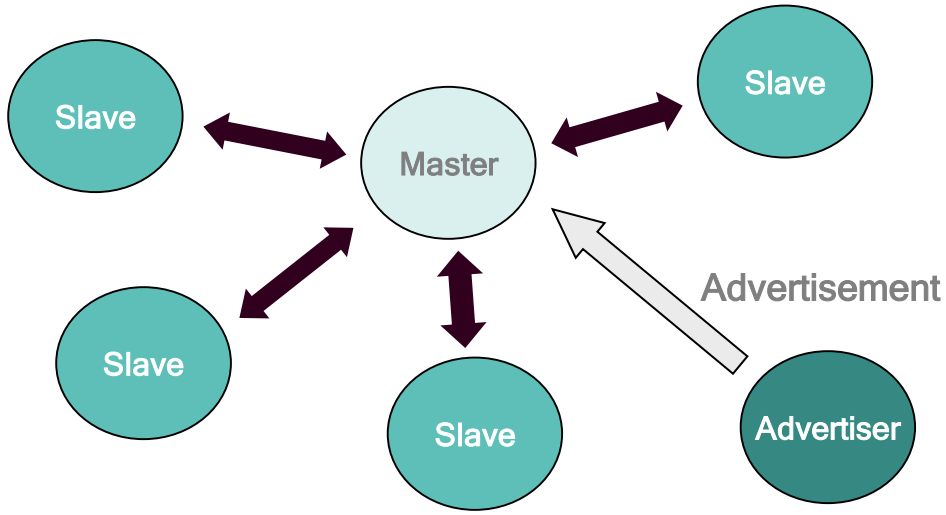
BLE Channels

MHz	Advertising	Data	WiFi
2480	39		
2478		36	
2476		35	
2474		34	
2472		33	
2470		32	11
2468		31	11
2466		30	11
2464		29	11
2462		28	11
2460		27	11
2458		26	11
2456		25	11
2454		24	11
2452		23	
2450		22	
2448		21	
2446		20	6
2444		19	6
2442		18	6
2440		17	6
2438		16	6
2436		15	6
2434		14	6
2432		13	6
2430		12	6
2428		11	
2426	38		
2424		10	
2422		9	
2420		8	1
2418		7	1
2416		6	1
2414		5	1
2412		4	1
2410		3	1
2408		2	1
2406		1	1
2404		0	1
2402	37		

An unconnected device decides it wants to be connected....

This device (e.g. data gatherer) needs to “advertise” its presence

- 3 channels for connectable and discoverable
- compromise for low power and robustness
- Time when connectable & discoverable = 1.3 ms
- (22.5 ms minimum for BR Bluetooth)
- Duty cycle 10x to 20x better than BR Bluetooth



Establishing Bluetooth Low Energy Connection - Advertising

- Slave device broadcasts its presence when it is ready to communicate.
- Advertising is broadcast on between 1, 2 or 3 available channels (user configurable)

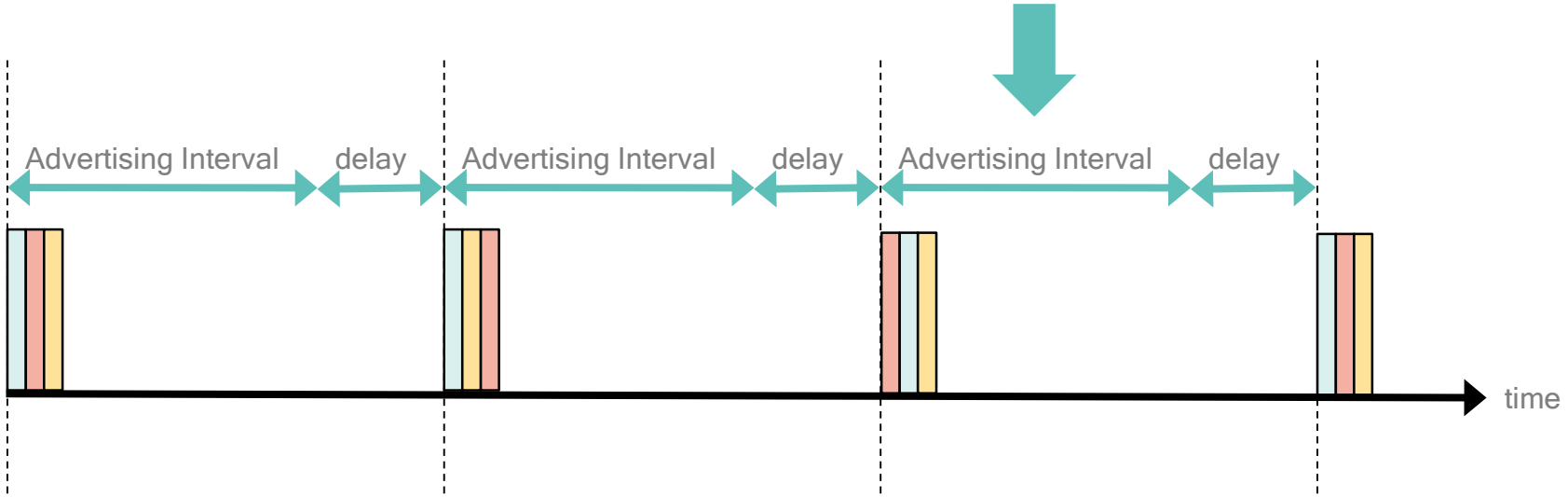
Advertising Event:

Send Advert PDU on channel 37 2402 MHz	Send Advert PDU on channel 38 2426 MHz	Send Advert PDU on channel 39 2480 MHz

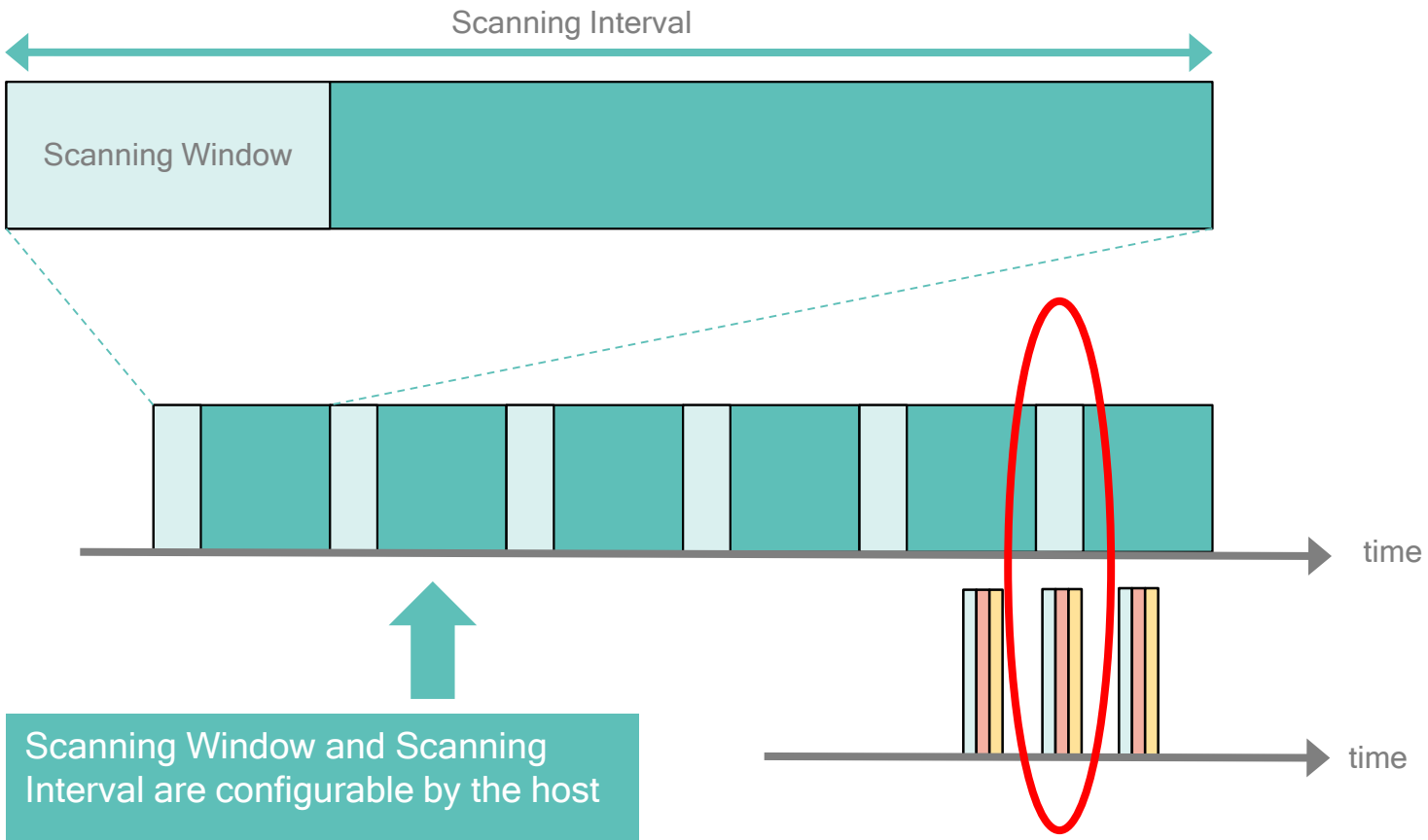
Advertising Interval is configurable by the host.

Min = 20ms
Max = 10.24s

Pseudo-random delay of 10ms added by stack to improve coexistence between devices



Establishing Bluetooth Low Energy Connection - Scanning



Scanning Window and Scanning Interval are configurable by the host

Min = 2.5ms
Max = 10.24s

If both are the same, scanning is continuous

Scanning interval coincides with advertising event. Master is aware that the advertiser is present

Establishing Bluetooth Low Energy Connection

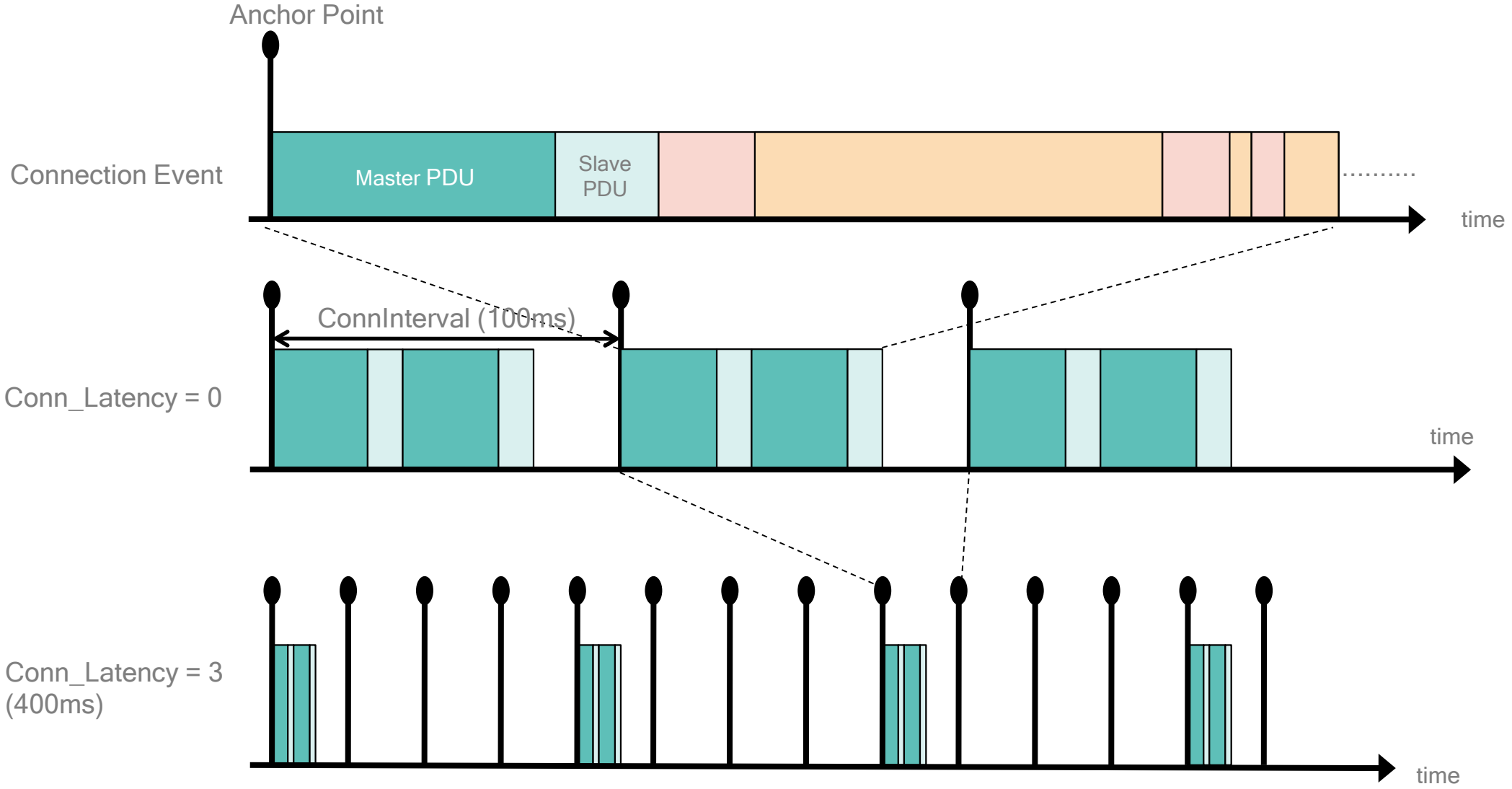
Extract from the Bluetooth Core Specification:

Command	OCF	Command parameters	Return Parameters
HCI_LE_Create _Connection	0x000D	LE_Scan_Interval, LE_Scan_Window, Initiator_Filter_Policy, Peer_Address_Type, Peer_Address, Own_Address_Type, Conn_Interval_Min, Conn_Interval_Max, Conn_Latency, Supervision_Timeout, Minimum_CE_Length, Maximum_CE_Length	

Important Parameters for Low Energy Operation

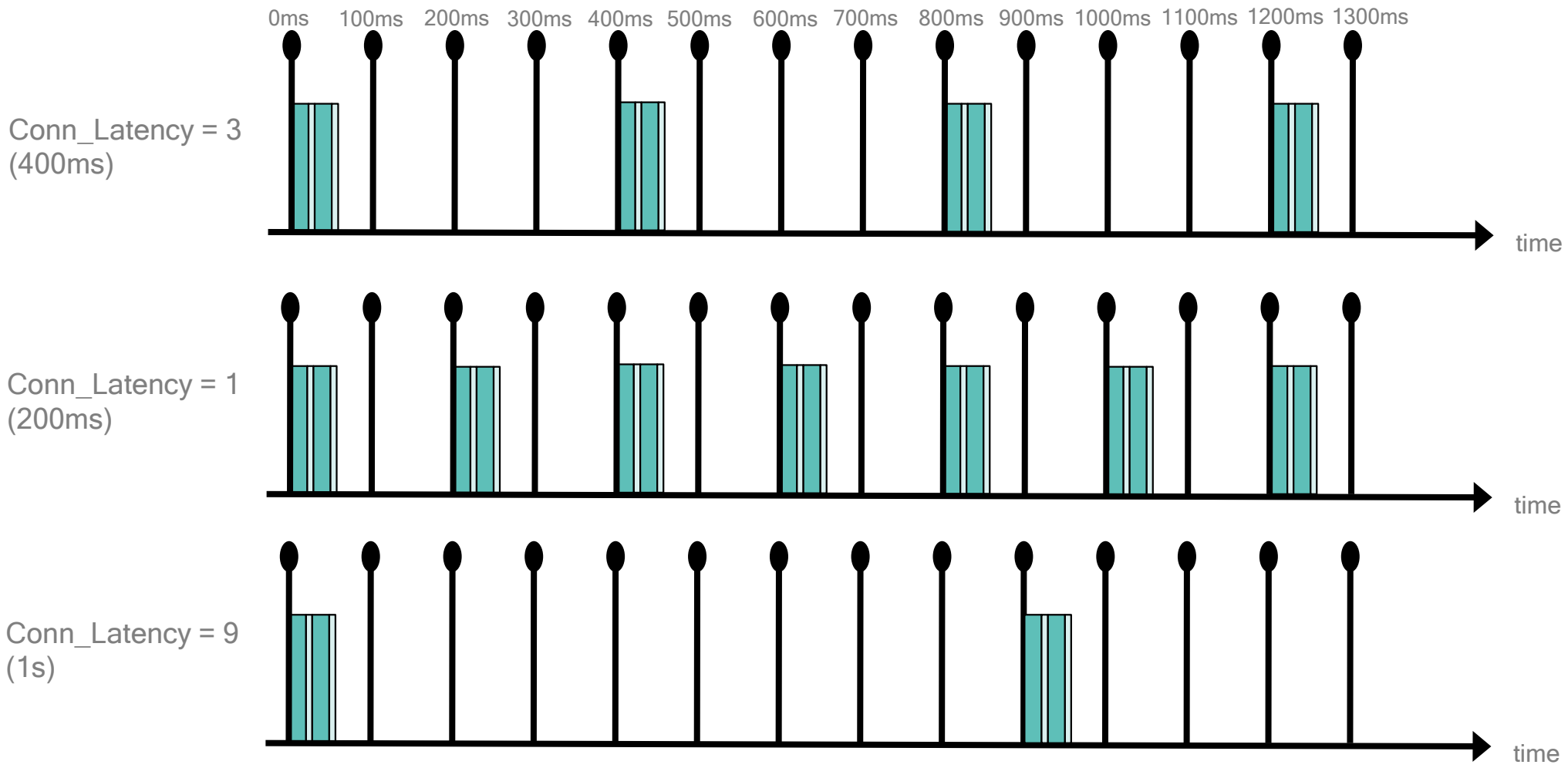
- Connection Interval - How much time elapses between connections
- Connection Latency - How many intervals a slave device may sleep for

Anchor Points and Latency



Anchor Points and Latency

Assuming Connection Interval = 100ms



Frequency Usage

BLE Channel Map

37	0	1	2	3	4	5	6	7	8	9	10	38	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	39

CONNECTION REQUEST PACKET DATA PAYLOAD (from core spec)

LLData									
AA (4 octets)	CRCInit (3 octets)	WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

Channel Map

Bitfield containing information on which channels are available for use

Min number of channels “enabled” per connection = 2.

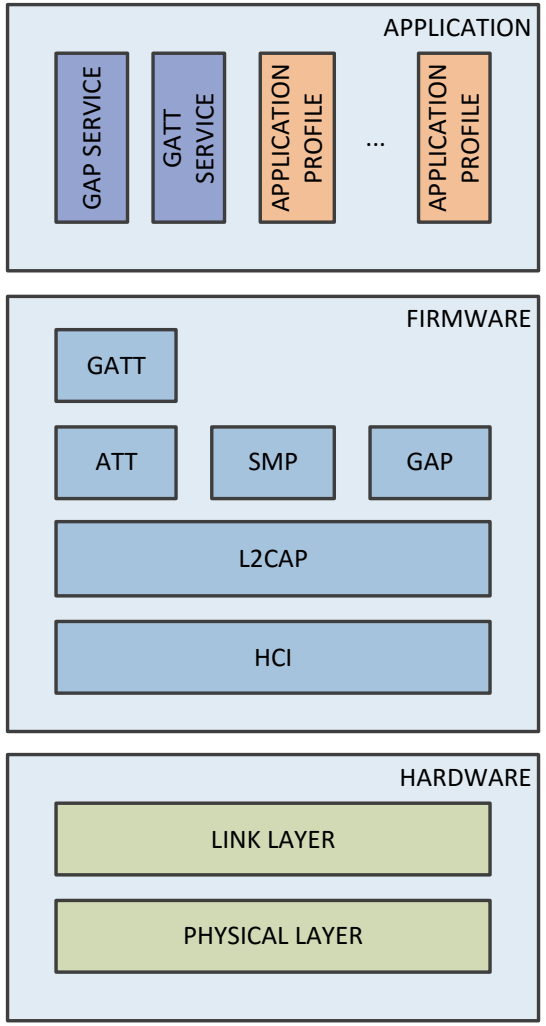
Unused channels are automatically remapped to used channels

Hop Increment

Channel Hop between consecutive connection events.

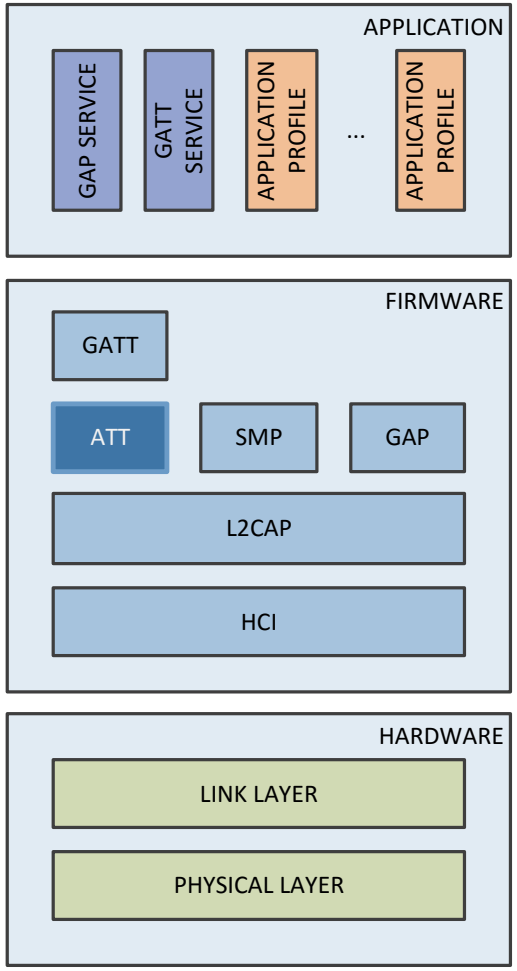
Bluetooth Low Energy protocol

Bluetooth Low Energy stack



Attribute protocol

- All current Bluetooth Low Energy profiles use ATT for transport
- Client-server architecture
  - ATT defines protocol for client to discover and access server data
  - Client and server roles do not depend on the link establishment roles – connection initiator can be ATT server or both client and server at the same time
- Server data is arranged in table of attributes
  - Each row in the server table is indexed by *attribute handle*



Attributes

Data is arranged as table of attributes

HANDLE	
0001	ATTRIBUTE
0002	ATTRIBUTE
...	
FFFE	ATTRIBUTE
FFFF	ATTRIBUTE

Each attribute consists of

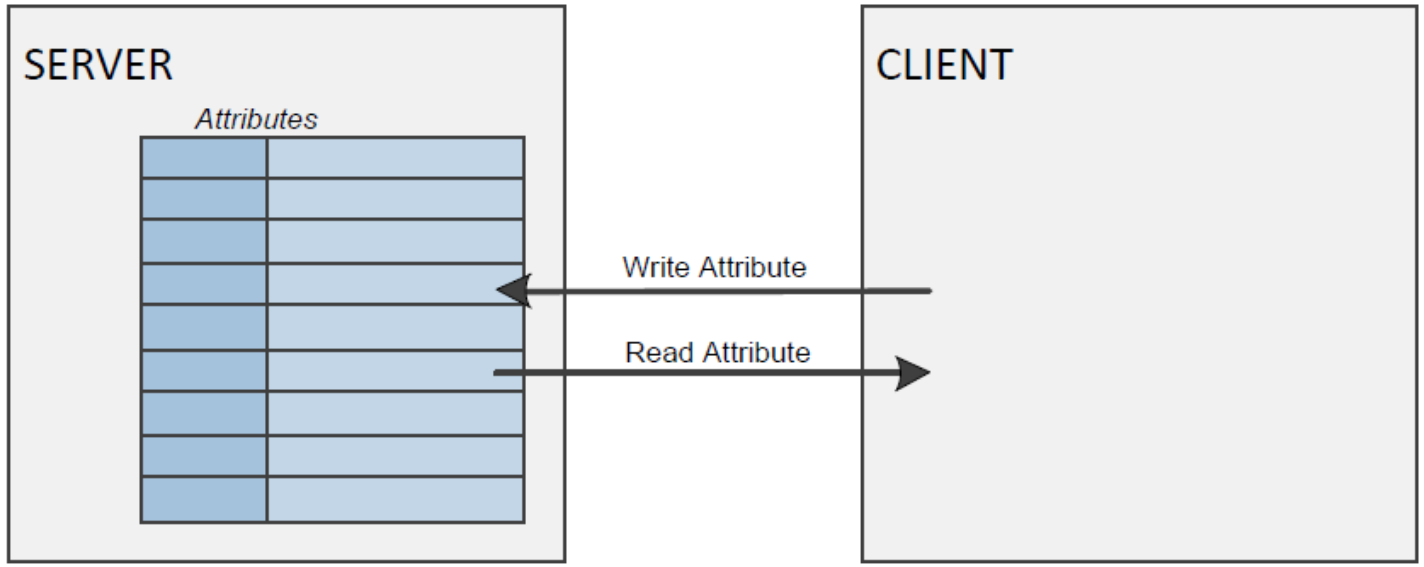
- Handle
- Attribute type (UUID)
- Permissions
- Attribute value

HANDLE	UUID	PERMISSIONS	VALUE

ATT: client-initiated data access

Client-initiated access

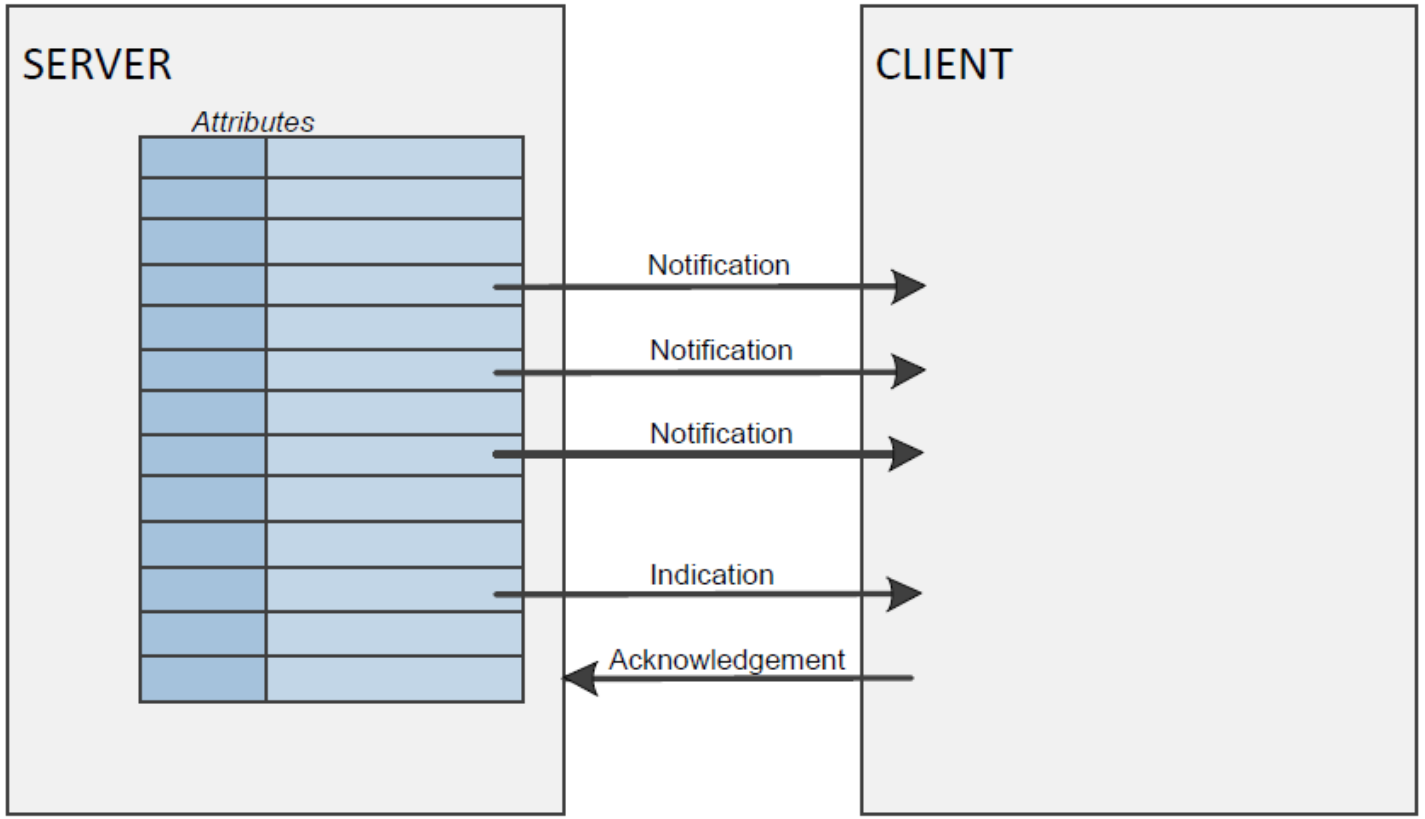
- Find information
- Read attributes
- Write attributes
- Queued attribute write



ATT: server-initiated data access

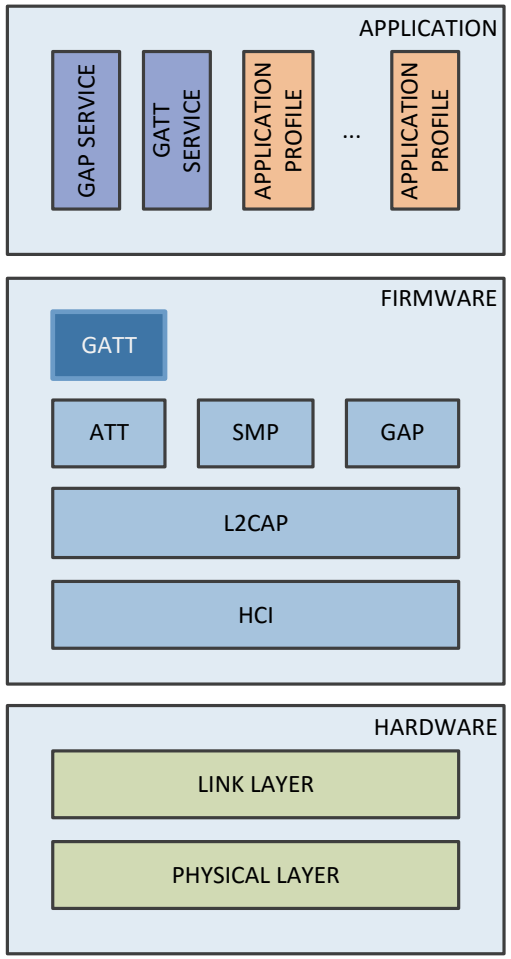
Server-initiated access

- Notifications
- Indications
  (acknowledged by client)



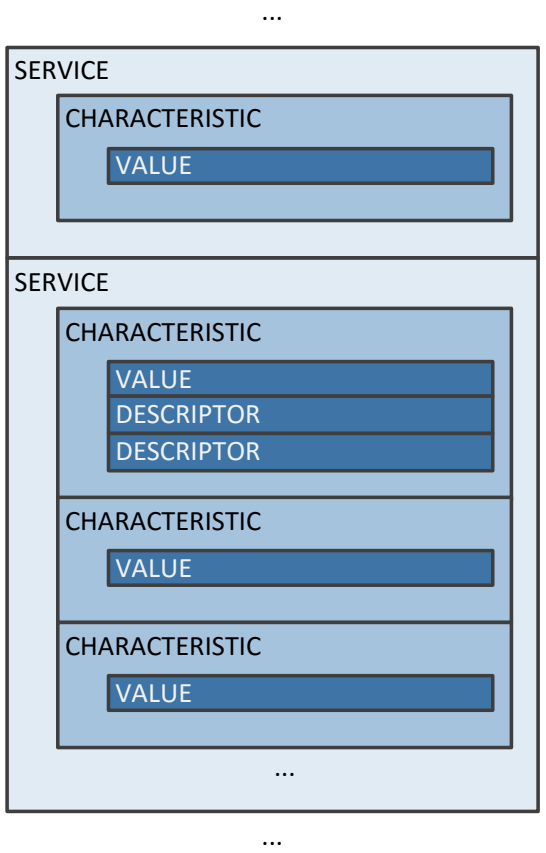
Generic attribute protocol

- All current Bluetooth Low Energy profiles are based on GATT
- Reuses ATT data structure and client-server architecture
  - Similar to ATT, GATT server and client roles do not depend on the link establishment role
- GATT server data is arranged in services with characteristics
- Defines access protocol to discover and access services and their characteristics



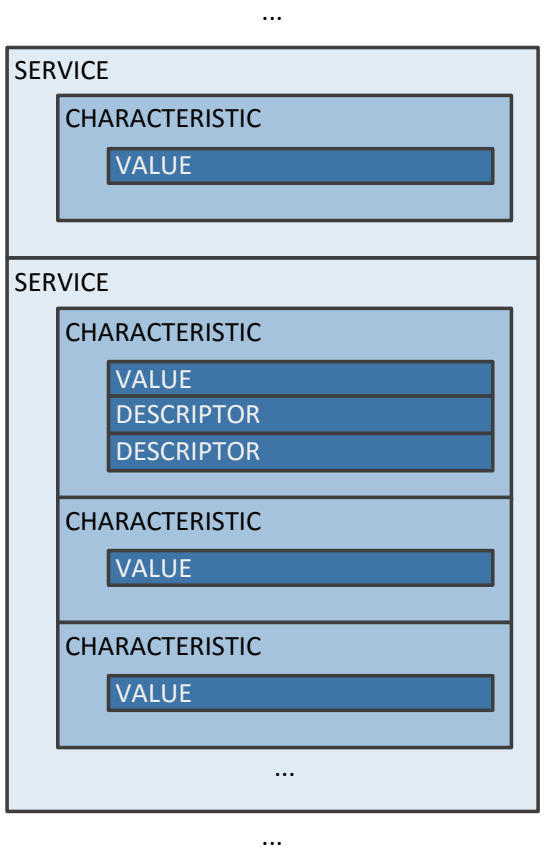
GATT database

- GATT server data is conceptually arranged into *services*, e.g. Battery Service
- The actual data (e.g. battery level in percent or battery state) is represented by *characteristics* of the service and their values, e.g. Battery Level

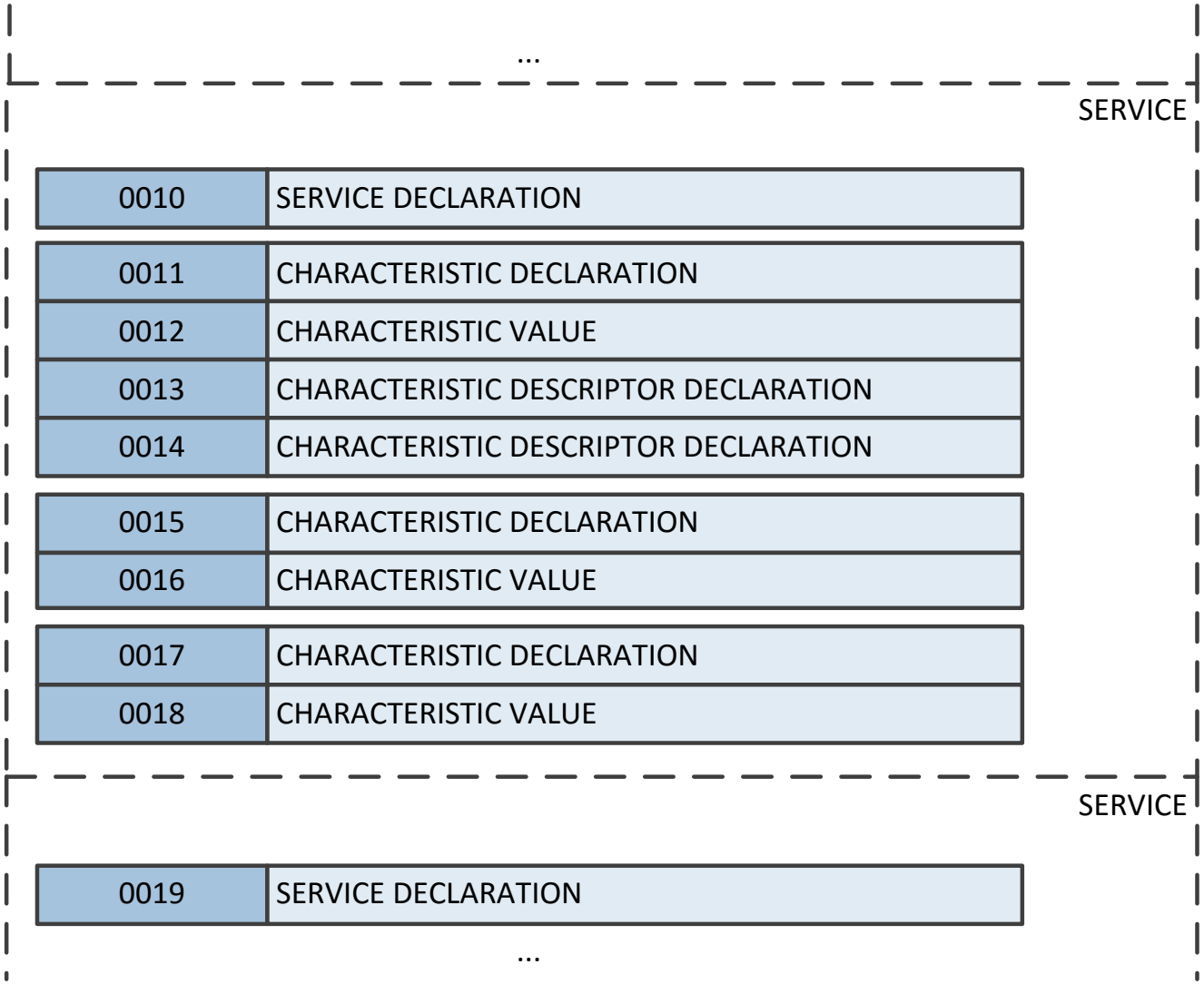


GATT database

- The characteristic may be further described by characteristic descriptor, for example:
  - Appearance
  - Client configuration
- The services, characteristics and descriptors are mapped on top of the ATT table structure
- Services and characteristics are identified by UUID



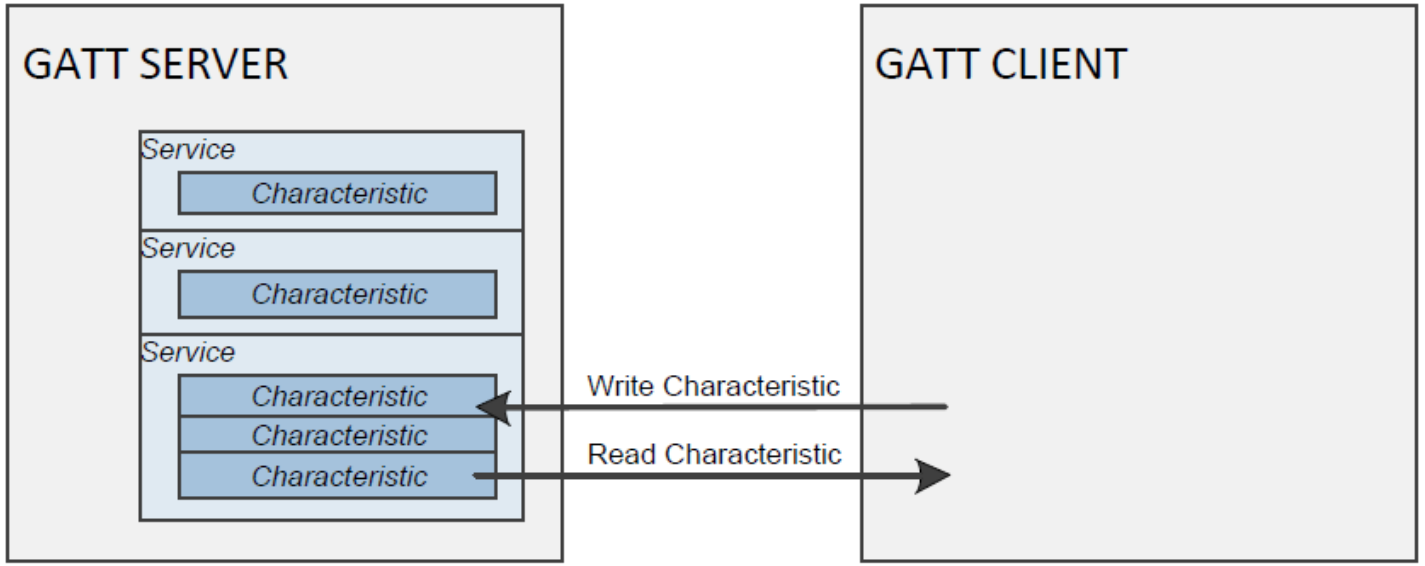
GATT database mapped to the ATT attributes table



GATT: client-initiated data access

Client-initiated access

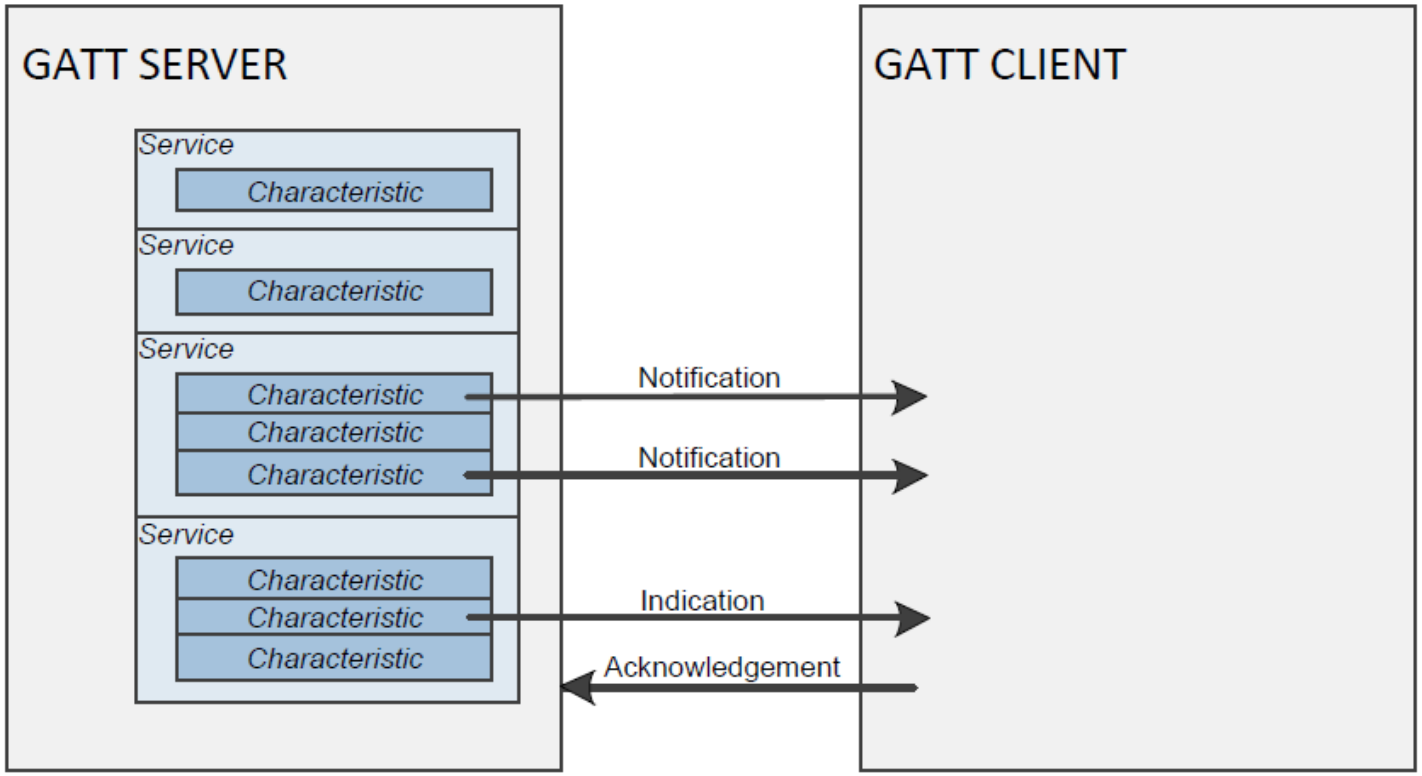
- Services, characteristics and descriptors discovery
- Read characteristic and descriptor value
- Write characteristic and descriptor value



GATT: server-initiated data access

Server-initiated access

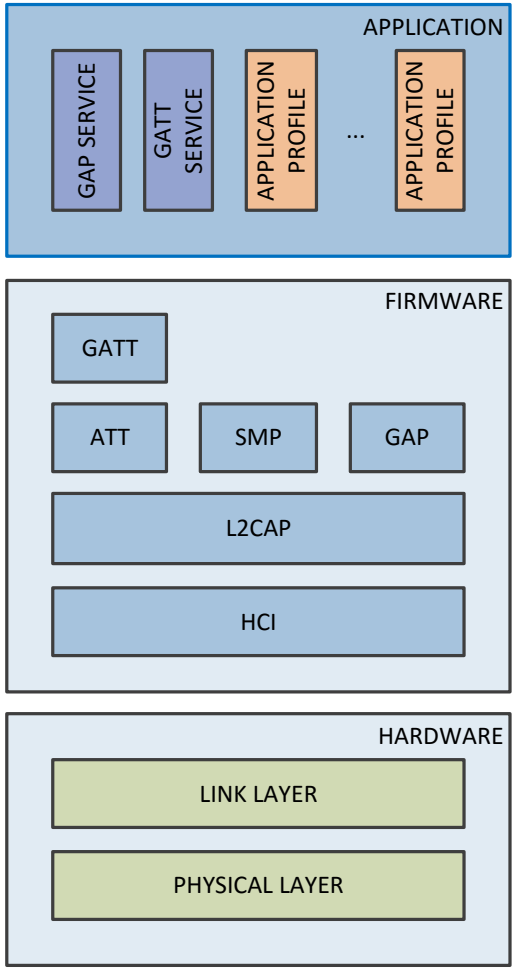
- Characteristic value notification
- Characteristic value indication
- Server has to be configured first by setting the “Client configuration” descriptor of the characteristic



Generic Access Profile - GAP

Defines profile roles

- Broadcaster – optimised for transmitter only applications
- Observer – optimised for receiver only applications
- Peripheral – optimised for low power, slave devices (e.g. sensors, input devices)
- Central – initiates and maintains connections (e.g. smartphone)



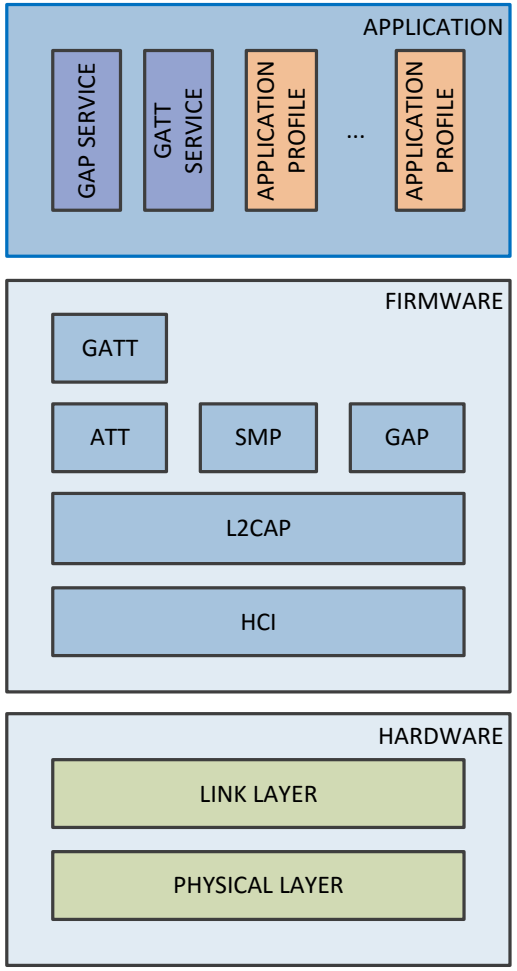
Generic Access Profile - GAP

Defines procedures for:

- Discovering identities, names, and basic capabilities
- Creating bonds
- Exchange of security information
- Establishing connections
- Resolvable Private addresses

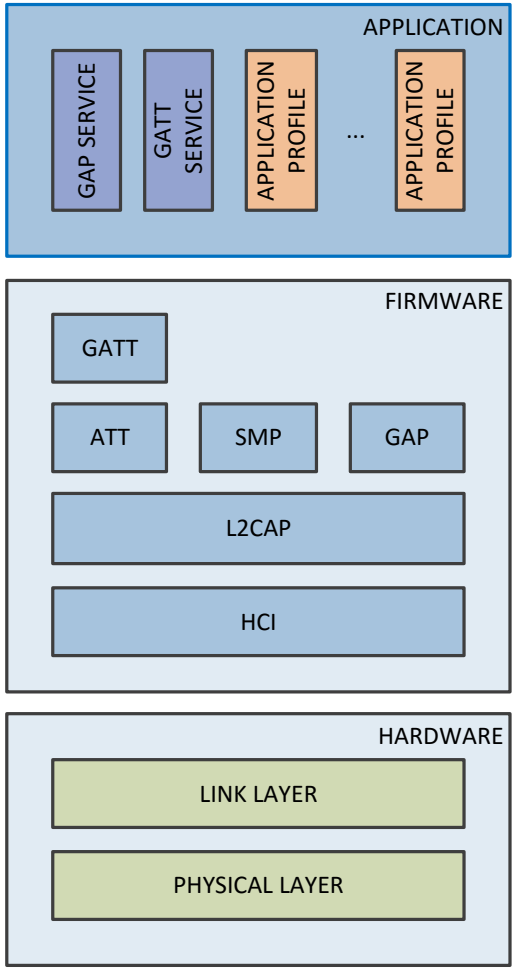
Defines Advertising and Scan Response Data formats

All profiles are built upon GAP



Application layer

- All current Bluetooth Smart application have to use GATT and GAP
  - Mandatory support for GAP service in GAP Central and Peripheral applications
  - Mandatory support for GATT service in GATT Client applications
- Bluetooth Smart profiles define requirements to GATT services, security aspects and GAP roles



Bluetooth Low Energy profiles

Qualcomm® Bluetooth® Low Energy

Terminology clarification

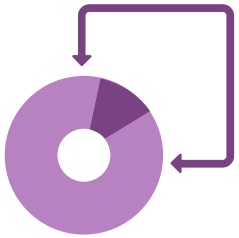
In this document you will notice a number of references are made to **Qualcomm Bluetooth Low Energy SDK.** While this **is** the official name of the SDK moving forward, within the actual SDK you will see it currently referred to as **CSR µEnergy**. In addition, many of the directories and folders may have the **CSR µEnergy SDK** naming convention. Please be assured these are the same thing.

Thank you for your understanding while we take the necessary steps to phase out the use of **CSR µEnergy** across our product lines.

CSR102x SDK: Core and Packages

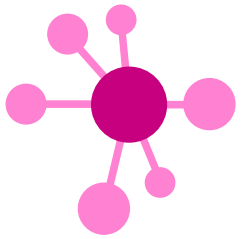
Segment Specific Packages

Available from https://www.csrsupport.com/uEnergy/Software



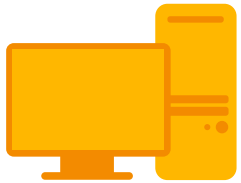
Core SDK

Firmware
Profile Demonstrator
GATT Server/Client
Serial Port Server/Client
Alert Tag
Security Tag
Time Client
Peripherals



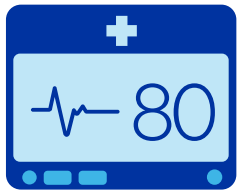
Generic

A4WP PRU
A4WP PTU
Alert Client
ANCS
Beacon
Environment Sensor
Security Tag
Time (Client)



HID

Keyboard
Mouse
Smart Remote



Health and Fitness

Smart Watch
Cycling Speed/Cadence
Running Speed/Cadence
Health Thermometer
Heart Rate
Weight Scale,
Glucose
Blood Pressure

Automotive

Keyless entry system
Multifunction Steering wheel
TPMS

Bluetooth Smart profiles

The Bluetooth SIG publishes a number of documents that describe Profiles and Services which form the basis of industry wide interoperability:

https://www.bluetooth.org/en-us/specification/adopted-specifications

Profiles

- Define required functions and features of each layer
  Specify inter-layer communication formats
- Specify connection, scanning and advertising parameters

Services

- Specify collections and format of required characteristics

Qualcomm® Bluetooth® Low Energy SDK example applications

Core SDK Packages

Core SDK Package (3.0.2)

- GATT Client application specific compliance
- GATT Server application specific compliance
- Serial Port Client profile application specific compliance
- Serial Port Server profile application specific compliance
- Peripheral operation example code

  AIO, Edge_capture, IR, Keyscan, PIO,

  PWM, Quadrature_Decoder, Timers, UART

Qualcomm® Bluetooth® Low Energy SDK example applications

Application Packages

Additional Package modules can be downloaded

Full source code is available with detailed documentation and customisation guidelines

Example applications are qualified and listed on Bluetooth SIG website

- **Generic Applications**
- **Health and Fitness Applications**
- **Automotive Applications**
- **HID Applications**

The Core SDK is a pre-requisite and must be pre-installed

Qualcomm Bluetooth Low Energy SDK example applications

Application Packages

Generic Applications (3.0.2)

- A4WP Receiving Unit
- A4WP Transferring Unit
- Alert Client
- ANCS
- Environmental Sensor (link)
- Security Tag
- Time Client

Qualcomm Bluetooth Low Energy SDK example applications

Application Packages

Health and Fitness Applications (3.0.2)

- Blood Glucose Sensor Application Specific Compliance
- Blood Pressure Sensor Application Specific Compliance
- Cycling Speed and Cadence Application Specific Compliance
- Health Thermometer Application Specific Compliance
- Heart Rate Application Specific Compliance
- Running Speed and Cadence Application Specific Compliance
- Smart Watch Application
- Weight Scale Application Specific Compliance

Qualcomm Bluetooth Low Energy SDK example applications

Application Packages

Automotive Applications (3.0.2)

- Keyless entry system automotive host application specific compliance
- Keyless entry system automotive key fob application specific compliance
- Multi-function steering wheel application specific compliance

Qualcomm Bluetooth Low Energy SDK example applications

Application Packages

HID Applications (3.0.2)

- BLE Remote Windows Demonstrator
- CSR Remote Control Application Generator
- CSR Smart Remote Test Tools
- Keyboard Application
- Mouse Application

Qualcomm Bluetooth Low Energy SDK example applications

Application Packages

Tools (3.0.2)

- Read or write Non-Volatile Memory devices, such as OTP, MTP, and SMEM (SPI-Flash).
- Configure CS keys
- Perform hardware tests
- Use and configure application security features
- Develop production test solutions using a set of libraries

Thank you

Follow us on:

For more information, visit us at:

www.qualcomm.com & www.qualcomm.com/blog

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

©2016 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to “Qualcomm” may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes Qualcomm’s licensing business, QTL, and the vast majority of its patent portfolio. Qualcomm Technologies, Inc., a wholly-owned subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of Qualcomm’s engineering, research and development functions, and substantially all of its product and services businesses, including its semiconductor business, QCT.