
# 蓝牙4.0核心规范培训

**Dan Tian**

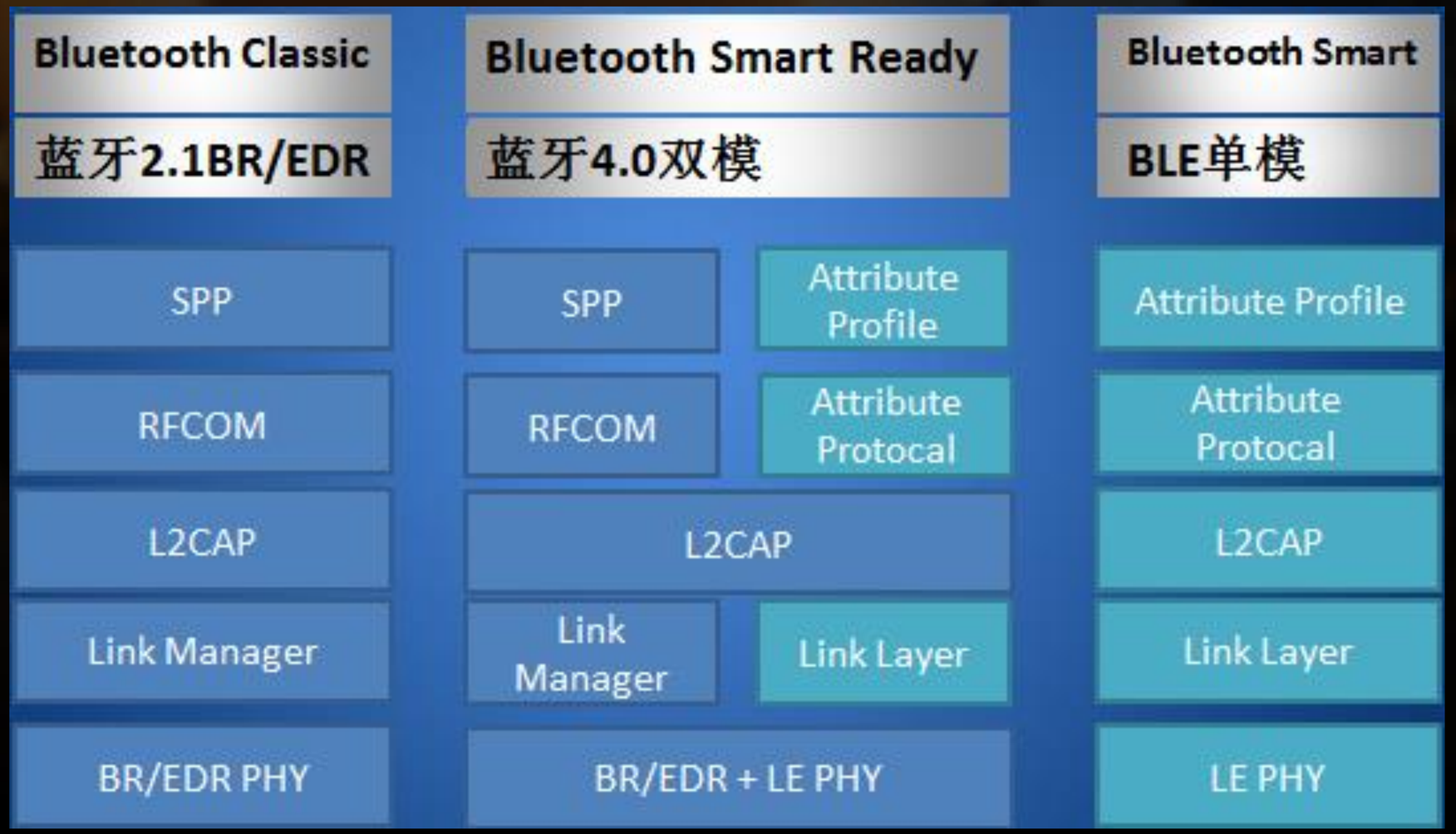
**JUMA Technology**

www.bcsphere.org

# Bluetooth

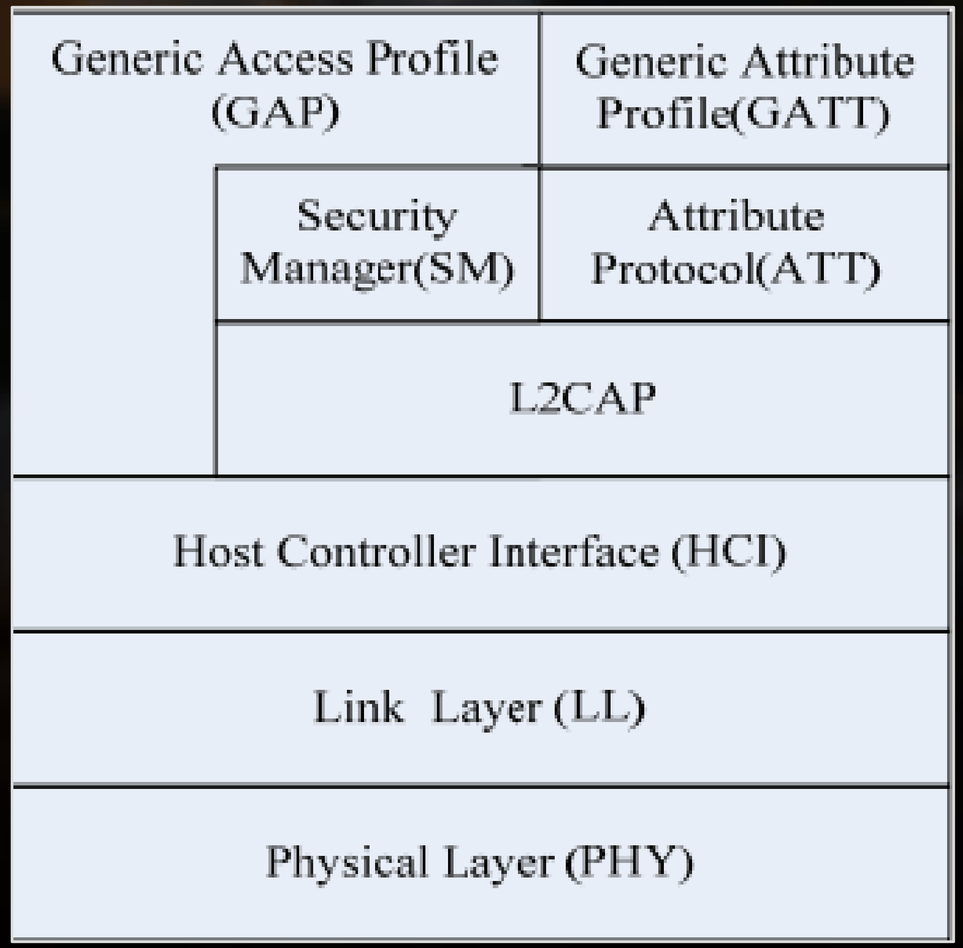
- 蓝牙4.0协议版本是在蓝牙3.0高速版本基础上增加了低能耗协议部分
- 具有低成本、跨厂商互操作性、3毫秒低延迟、AES-128加密等特性
- 可以广泛应用于计步器、心律监视器、智能仪表、传感器物联网等众多领域
- 预计在2016年将有十亿的设备出货量
- 就单模而言和经典蓝牙设备不兼容

# Bluetooth


Bluetooth Classic
Bluetooth Smart Ready
Bluetooth Smart
蓝牙2.1BR/EDR
蓝牙4.0双模
BLE单模
SPP
SPP
Attribute Profile
Attribute Profile
RFCOM
RFCOM
Attribute Protocal
Attribute Protocal
L2CAP
L2CAP
L2CAP
Link Manager
Link Manager
Link Layer
Link Layer
BR/EDR PHY
BR/EDR + LE PHY
LE PHY

# BLE协议原理

LE协议的底层与基础蓝牙协议底层基本相似，但在主机端，针对传感器网络应用推出了属性协议ATT以及通用属性剖面GATT，具体协议分层结构，如图所示：其中，基于逻辑链路与适配协议即L2CAP以上的部分可在主机端实现，这一部分可称为主机端部分，HCI层以下部分可称为芯片控制器层也可简称底层协议。



# Bluetooth

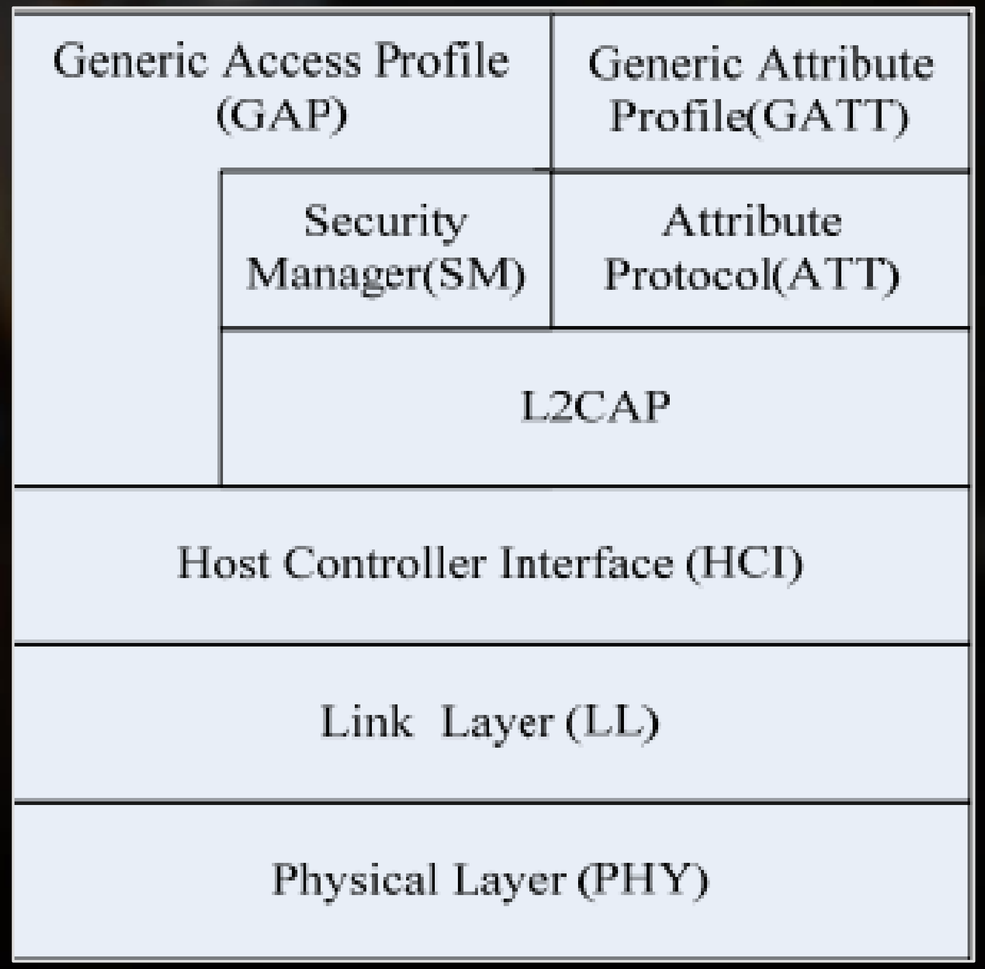
采用调频技术减少干扰与信号衰减，从2.402～2.480 GHz均匀分为40个信道，每个信道宽2MHz

GFSK调制方式（高斯频移键控）

输出功率为：0.01～10mw，传输速率1 Mb/s

提供3个固定的广播信道，广播数据用于建立连接以及发现设备，这样使得建立连接的时间可以压缩到3ms左右，大大提高了设备建立连接的效率

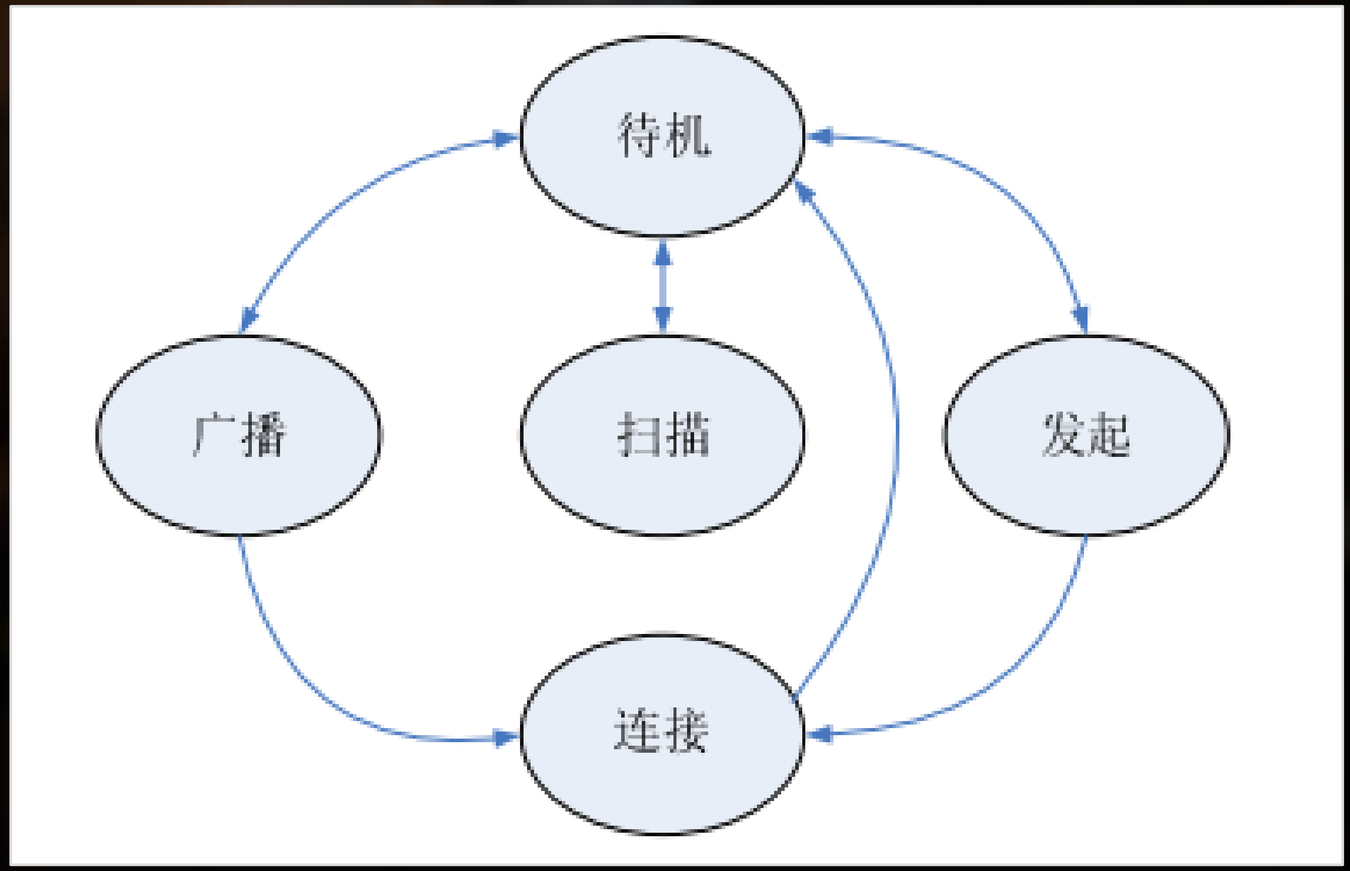
提供37个数据信道采用自适应调频技术发送数据。



# 链路层（Link Layer）

- 链路层功能是执行一些基带协议底层数据包管理协议。

- 链路层设备主要有待机、发起、扫描、连接、广播5种工作状态，状态转换图，如图所示：



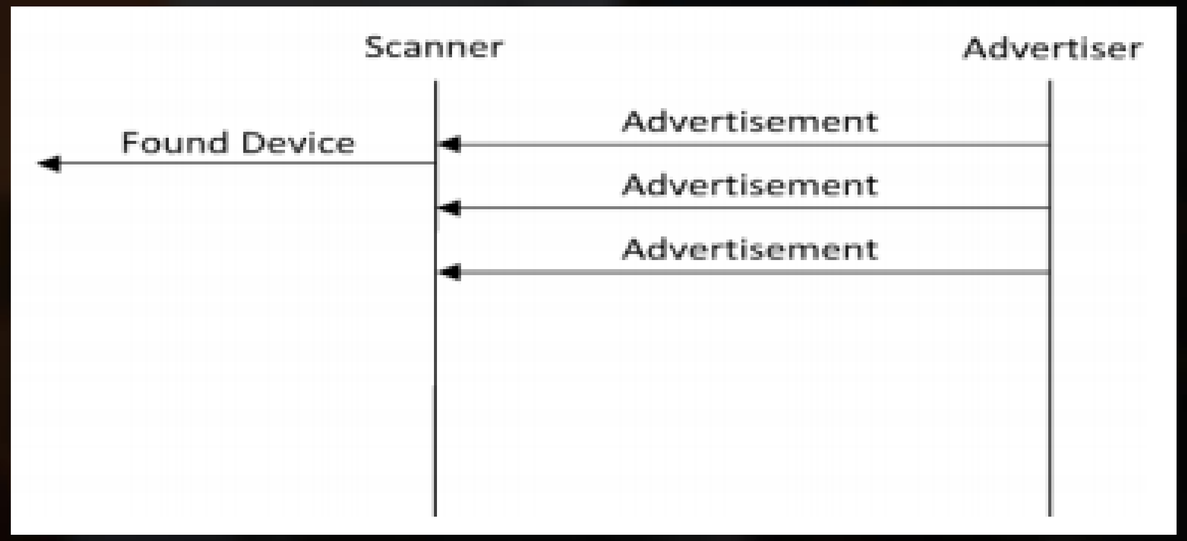
# 链路层设备五种工作状态

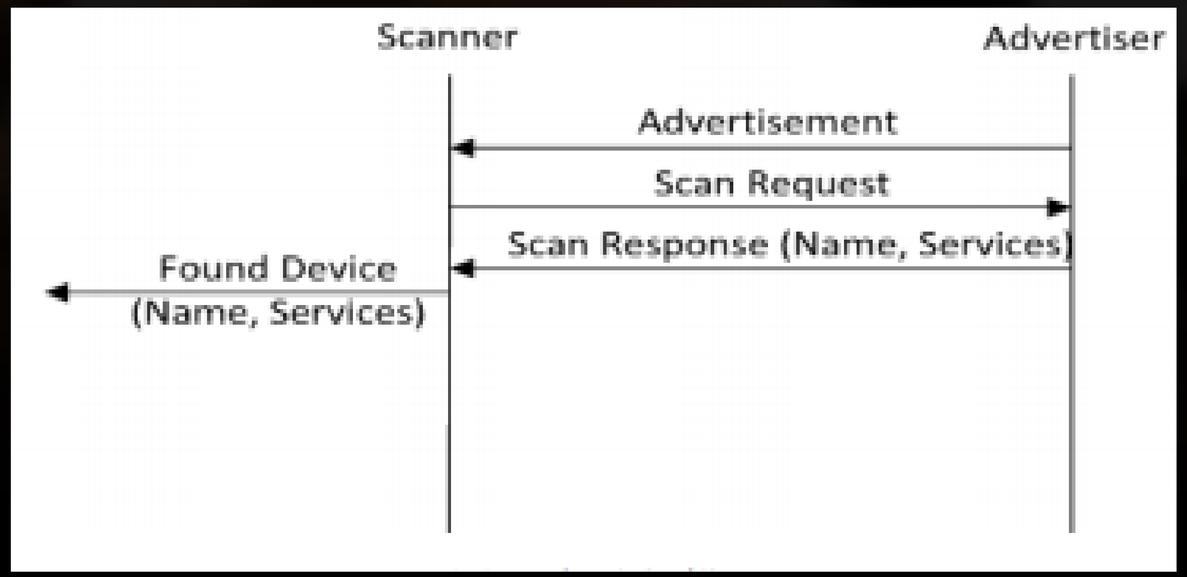
- 待机状态不发送和不接受任何包，任何状态都可以进入待机状态。
- 广播状态在广播信道发送广播包并且监听可能的响应包，广播状态可以由待机状态进入。
- 扫描状态将会监听广播信道包，扫描状态可以从待机状态进入。
- 发起状态将会监听从特定设备发出的广播包并且发起链接请求作为响应，发起状态可从待机状态进入。
- 连接状态可以从发起状态或者广播状态进入，在连接状态下有主从两种角色。
- 当从发起状态进入连接状态时，发起连接请求，将会是主设备，当从广播状态进入连接状态时，将会是从设备。

# 链路层事件操作

- 链路层主要有两种重要的事件操作：扫描与建立链接。

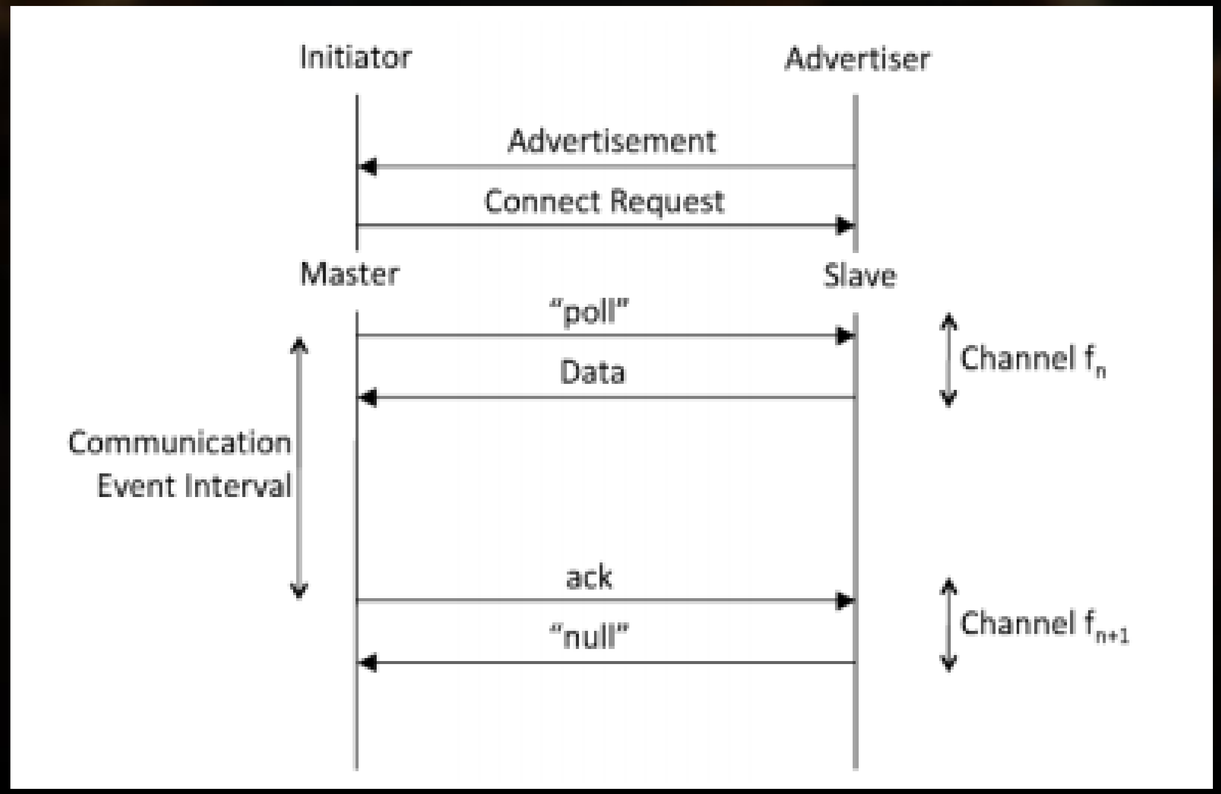
- 设备扫描有被动扫描和主动扫描：被动扫描是通过被动接收广播包得到设备信息；主动扫描是通过发送扫描请求得到扫描回应得到设备信息。



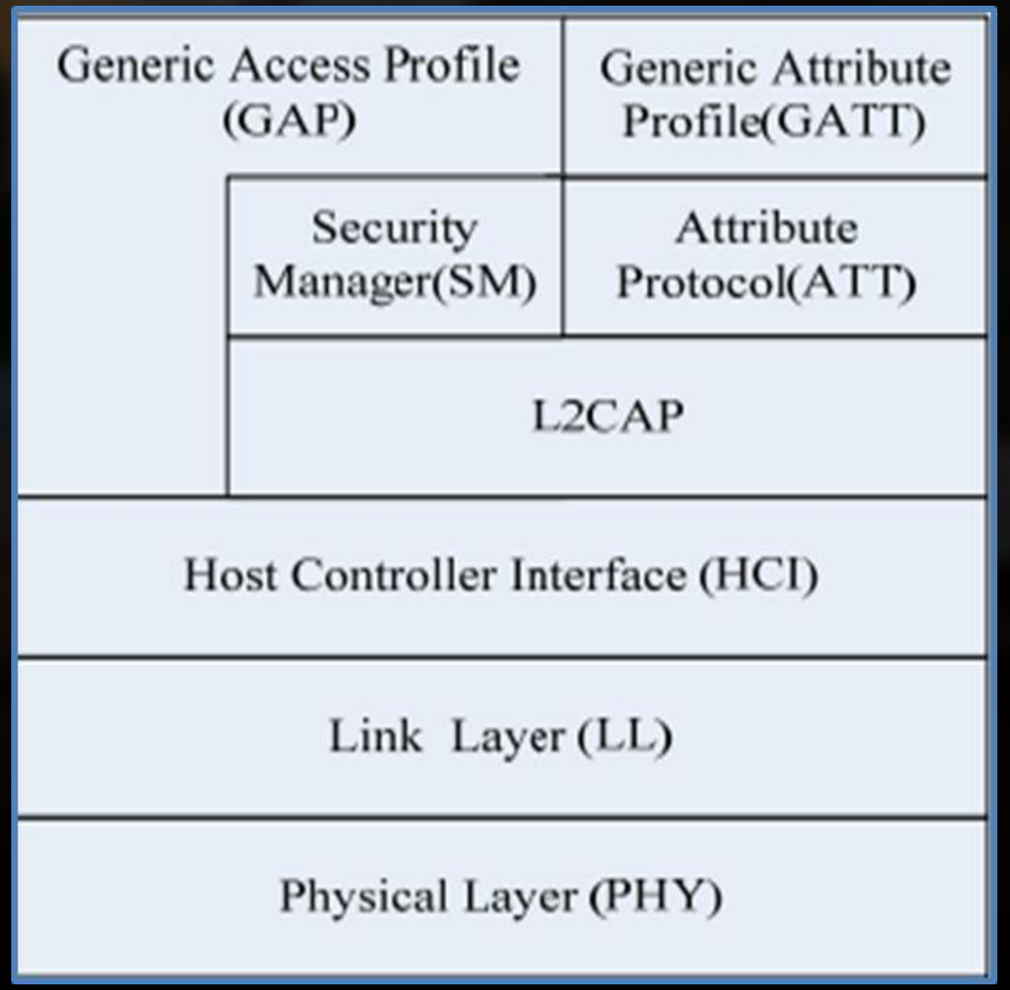
# 链路层事件操作

建立链接过程是通过发送链接请求包来建立设备链接



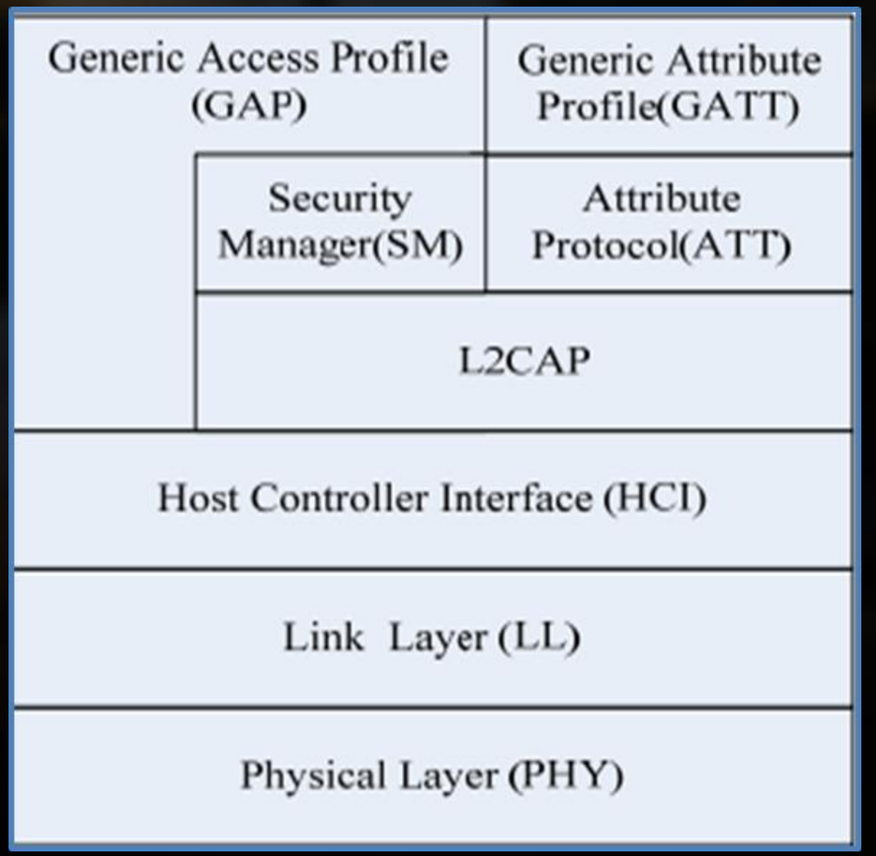
# 主机控制接口层(HCILayer)

主机控制接口与标准蓝牙技术相同，提供了主机与控制器层的通信方式与命令事件格式，重用标准蓝牙传输层接口如UART,USB等。



# 逻辑链路与适配协议层(L2CAPLayer)

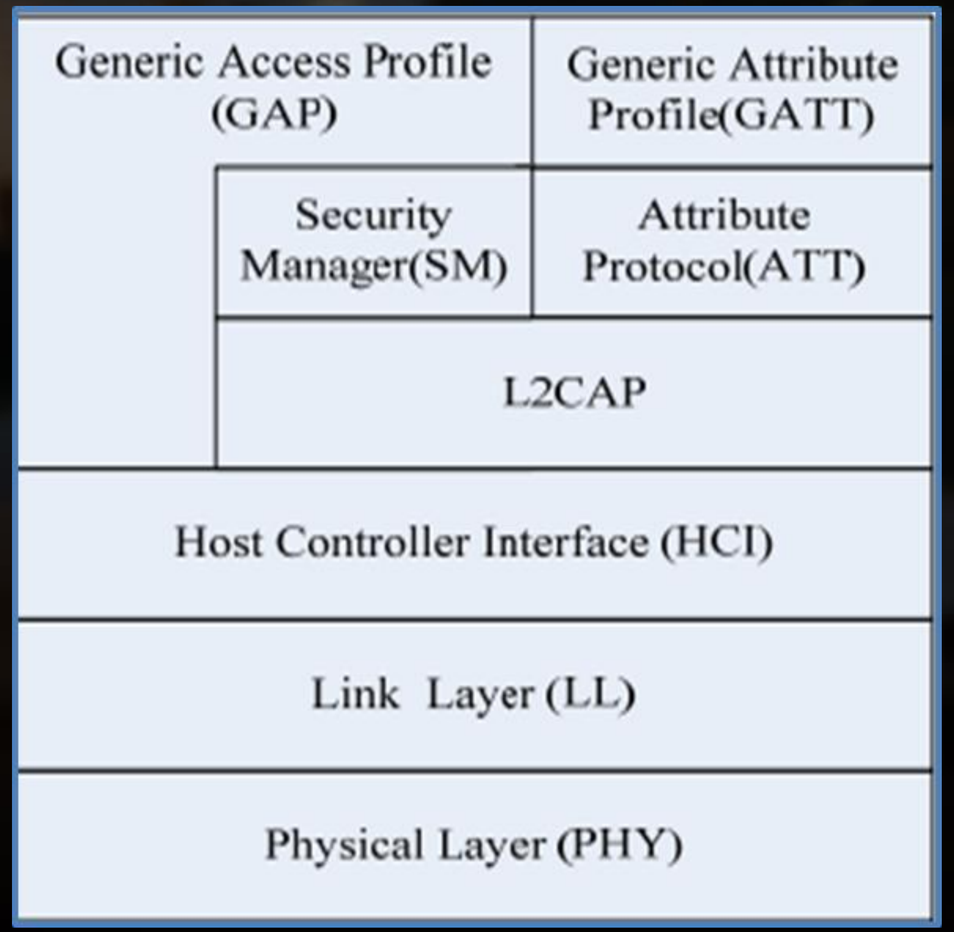
与标准蓝牙技术相同，为上层提供了数据封装业务，提供端到端的逻辑数据通信。



# 安全管理层(Security Manager Layer)

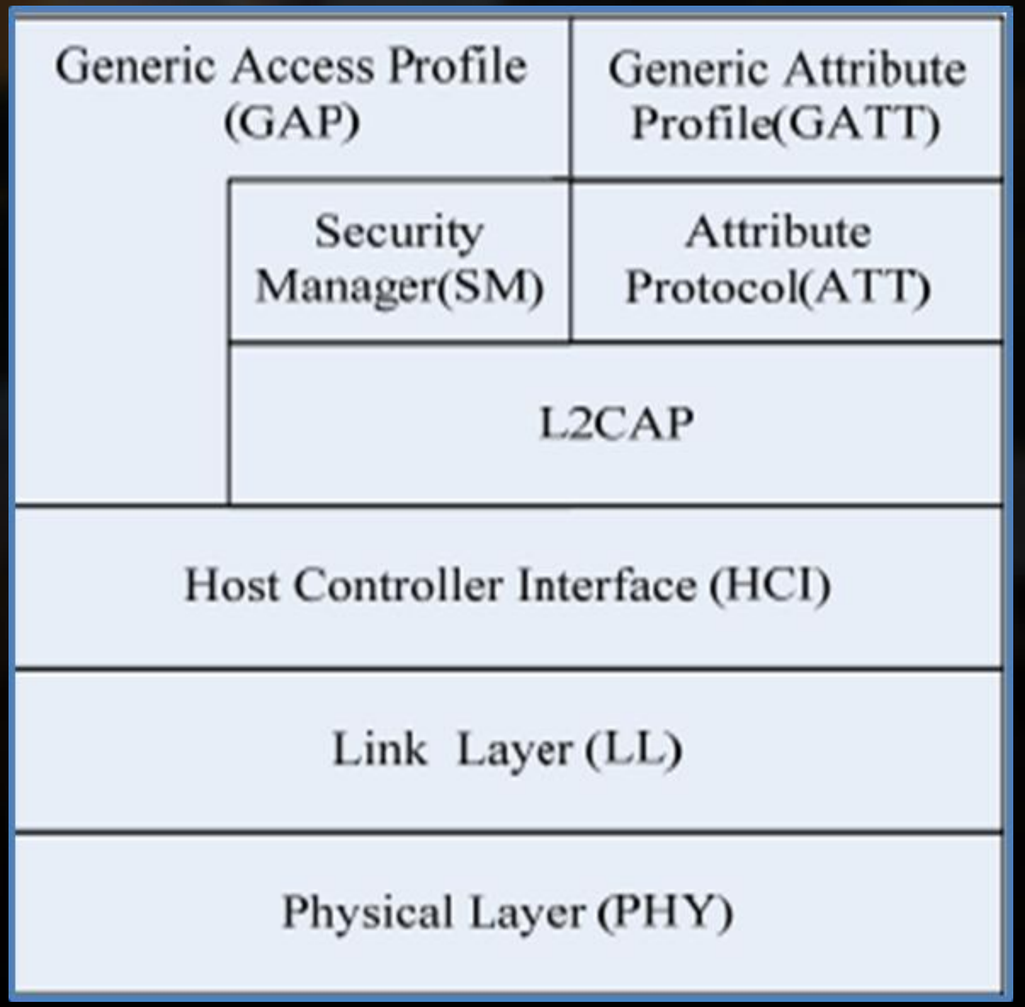
- 定义了配对和密钥分发方法，提供其它层协议接口来安全的建立连接以及交换数据。

- 安全管理层不涉及具体的BLE安全算法，只是提供一些接口，为节省功耗以及降低复杂性，具体安全算法可以通过在底层硬件实现。



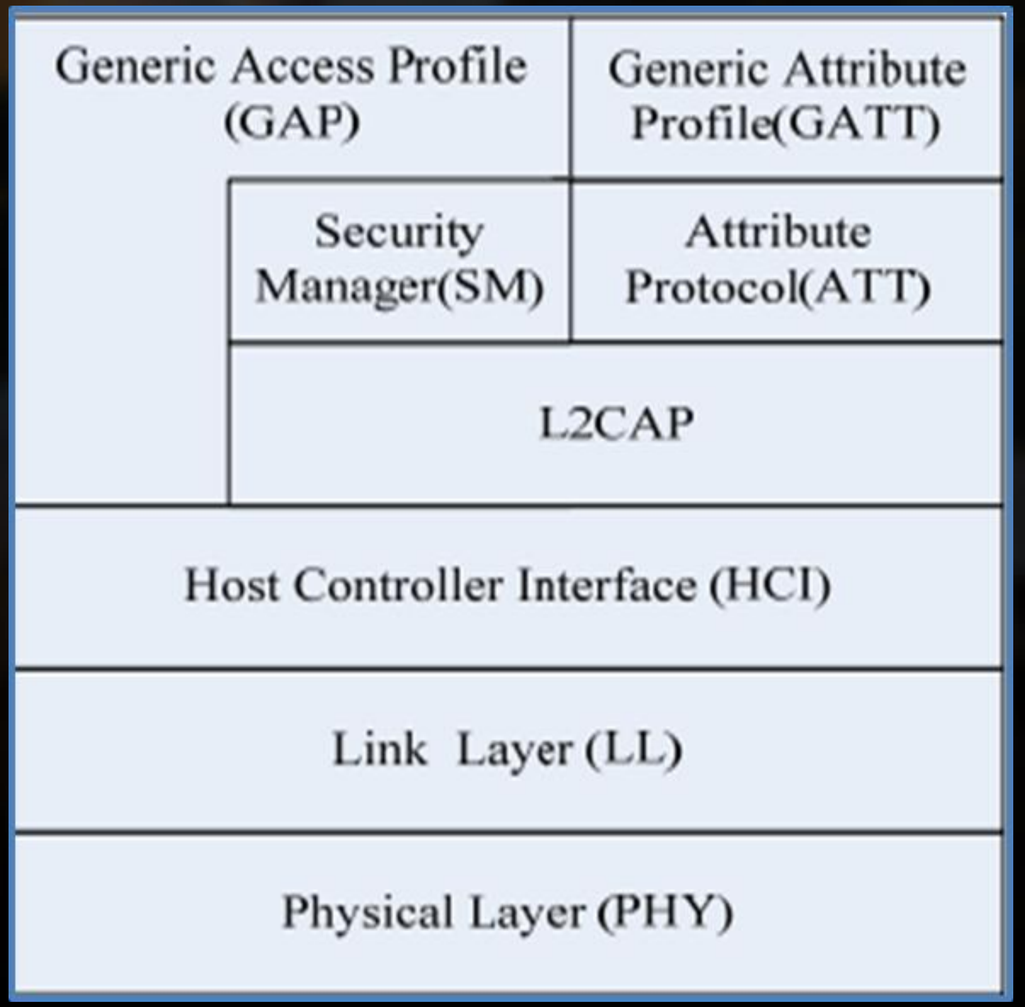
# 通用接入(GAP)

定义了通用的接口，供应用层调用底层模块，比如设备发现，建立连接相关的业务，同时封装了安全设置相关的API。



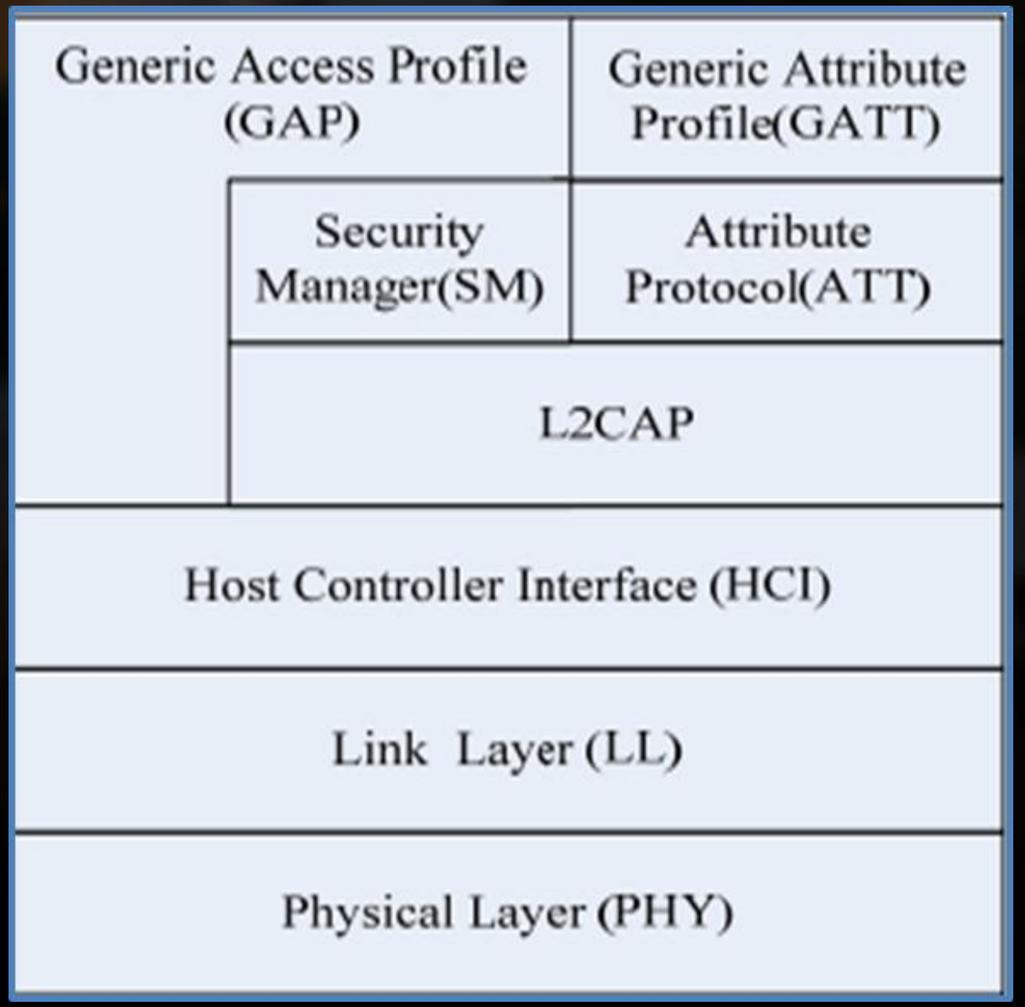
# 属性协议层(ATT Layer)

属性协议允许设备以“属性”的形式向另外的设备暴露他的某些数据。在ATT协议里，暴露属性的称为Server 端，另外一端称为Client。



# 通用属性(GATT Profile)

GATT 层是一种具体使用属性协议的应用框架。GATT 定义了属性协议应用的架构。在BLE协议中，应用中数据片段被称为“特征”， 而BLE中两个设备之间的数据通信就是通过GATT 子过程来处理的。



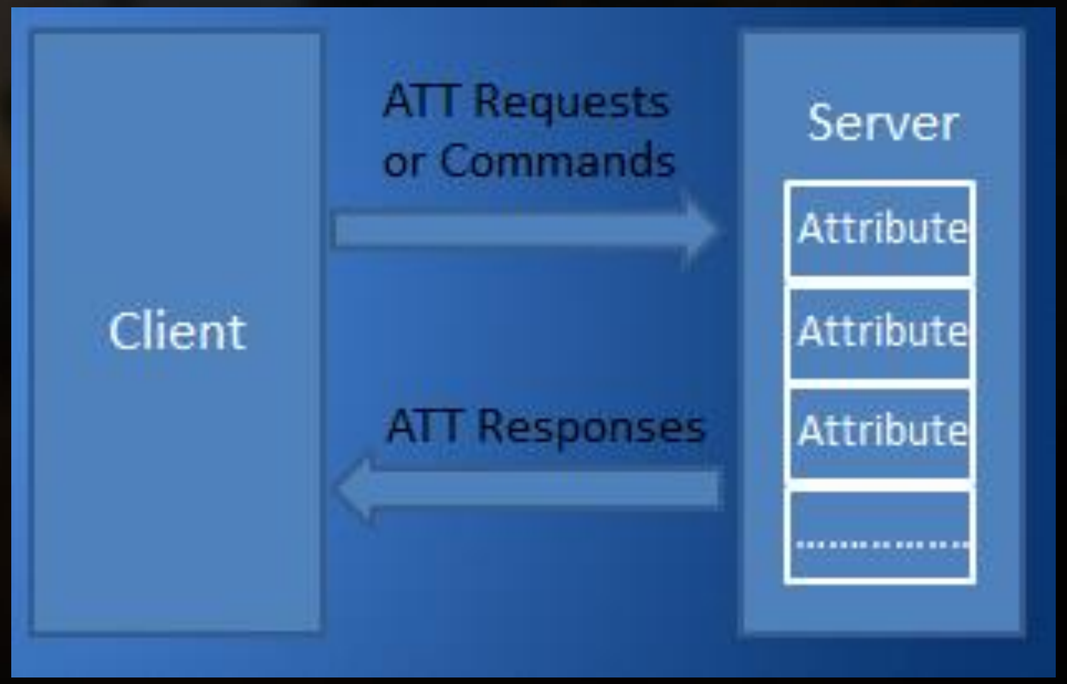
# BLE拓扑结构和设备状态

BLE 是一种星形拓扑结构:
主设备管理着连接，并且可以连接多个从设备
一个从设备只能连接一个主设备


做为一个BLE设备，有六种可能的状态:
待机状态(Standby)：设备没有传输和发送数据，并且没有连接到任何设备
广播状态(Advertiser)：周期性广播状态
扫描状态(Scanner)：主动地寻找正在广播的设备
发起连接状态(Initiator): 主动向某个设备发起连接
主设备(Master)：作为主设备连接到其他设备
从设备(Slave) ：作为从设备连接到其他设备

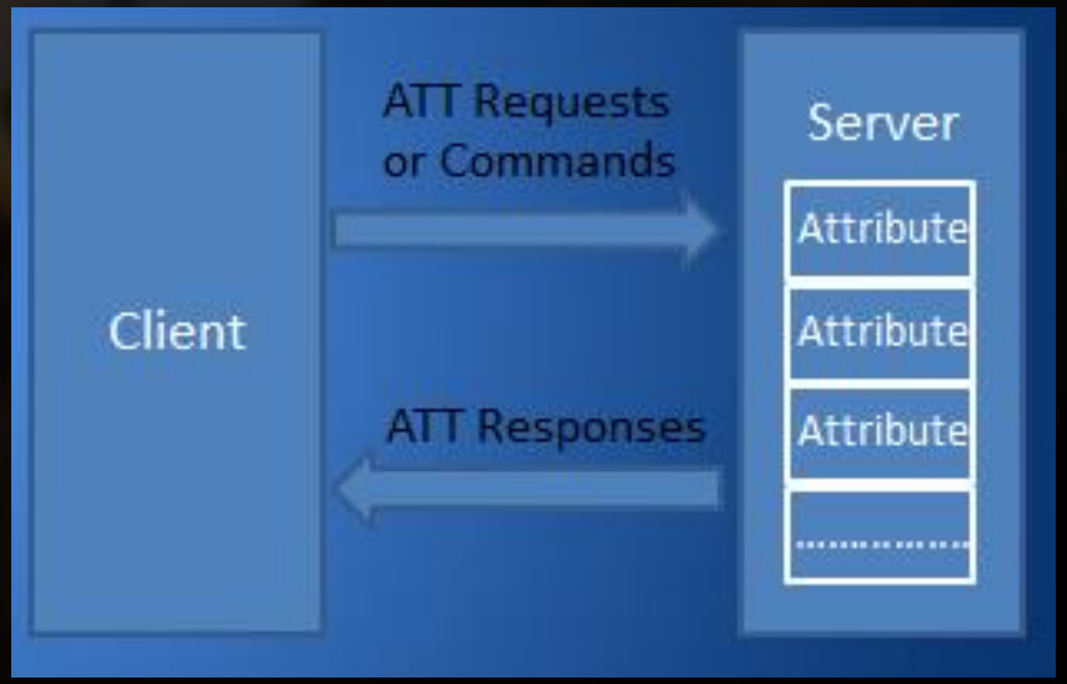
# BLE (ATT): Client / Server架构

- 服务设备提供数据，客户端使用这些数据
- 服务端通过操作属性的方式，提供数据访问服务
- 设备的服务/客户角色，不依赖于GAP层中心设备/外围设备角色，和LL层master/slave角色定义
- 一个设备可能同时做为一个客户端和服务端，而两个设备上的属性不会相互影响。



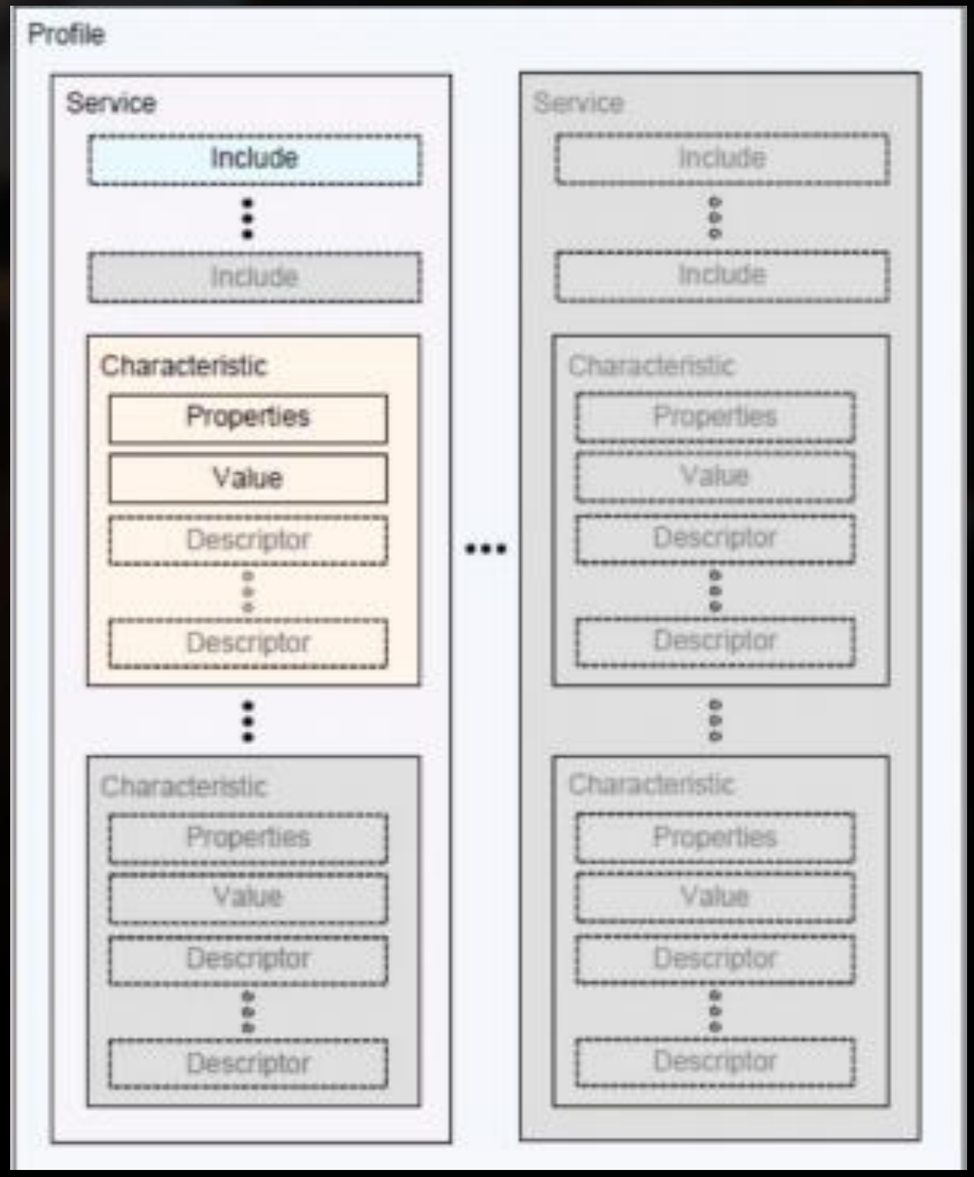
# BLE (ATT): Client / Server架构

- GATT指定了profile数据交换所在的结构
- 除了数据的封装方式不同，client server和Attribute 协议结构相同，数据封装在“Services”里，用“Characteristic”表示。



# BLE (GATT):Profile 层次结构

- 为了实现用户的应用，profile 通常是由一个或者多个 “services” 组成
- 一个service 或许包含某个特征值 “characteristic values”(例如：在一个温度采集设备中，通常会包含一个温度的特征值)
- 每一个特征值必须有占用一个特征申明结构，其中包括它的其他特性，它是服务端和客户端共享的读写空间
- 这个特征值可以包含一个可选的描述（descriptor字串），来指示这个特征值的含义



# BLE (GATT): Service Example

- service 起始于 handle 39 ，用 0x2800来指示这个起始位置，这个0x2800是Bluetooth SIG 的相关数据手册定义的，做为 GATT Service的UUID
- handle 39 所指向的属性值为 0xFFE0, 这个是用户自定义 profile中的按键服务UUID (这里仅仅是一个例子，0xFFE0 可能已经被Bluetooth SIG使用)
- 这个按键服务包含了所有后面的属性，直到下一个服务 UUID定义处，或者到服务列表的结尾处。在这个例子中，按键服务的最后的属性所在地址为handle 42，因此新的服务开始位置将会是 handle 43

| Handle | Type | Permissions | Value |
|---|---|---|---|
| 39 | 0x2800 (GATT Primary Service UUID) | Read | E0:FF (2 bytes)<br>(0xFFE0 = Simple Keys Service custom UUID) |
| 40 | 0x2803 (GATT Characteristic Declaration UUID) | Read | 10:29:00:E1:FF (5 bytes)<br>(0xFFE1 = Simple Keys Value custom UUID)<br>(0x0029 = handle 41)<br>(0x10 = characteristic properties: notify only) |
| 41 | 0xFFE1 (Simple Keys state) | (none) | 00 (1 byte)<br>(value indicates state of keys) |
| 42 | 0x2902 (GATT Client Characteristic Configuration UUID) | Read and Write | 00:00 (2 bytes)<br>(value indicates whether notifications or indications are enabled) |
| 43 | 0x2800 (GATT Primary Service UUID) | Read | A1:DD (2 bytes)<br>(0xDDA1 = Other Service custom UUID) |

# BLE (GATT): Characteristic Declaration

- Handle 40 是一个特征值的声明，用0x2803来指示，这个0x2803 同样也是Bluetooth SIG 的相关数据手册定义的，做为GATT Characteristic Declaration的UUID
  特征值的属性值包含5个字节的长度10:29:00:E1:FF :

- (0xFFE1) – 表明特征值的属性类型 (0xffe1:客户自定义特征值的UUID)
  (0x0029) – 是这个值所保存的位置handle (0x0029 = 41)
  (0x10) – 表明对这个特征值的操作权限 (0x10 : notify only)

| Handle | Type | Permissions | Value |
|---|---|---|---|
| 39 | 0x2800 (GATT Primary Service UUID) | Read | E0:FF (2 bytes)<br>(0xFFE0 = Simple Keys Service custom UUID) |
| 40 | 0x2803 (GATT Characteristic Declaration UUID) | Read | 10:29:00:E1:FF (5 bytes)<br>(0xFFE1 = Simple Keys Value custom UUID)<br>(0x0029 = handle 41)<br>(0x10 = characteristic properties: notify only) |
| 41 | 0xFFE1 (Simple Keys state) | (none) | 00 (1 byte)<br>(value indicates state of keys) |
| 42 | 0x2902 (GATT Client Characteristic Configuration UUID) | Read and Write | 00:00 (2 bytes)<br>(value indicates whether notifications or indications are enabled) |
| 43 | 0x2800 (GATT Primary Service UUID) | Read | A1:DD (2 bytes)<br>(0xDDA1 = Other Service custom UUID) |

# BLE (GATT):Characteristic Configuration

- 另外做为特征值声明，可以有一个可选的描述信息
  这个例子中，handle 42 包含了特征值的配置信息，0x2902，这个值同样也是Bluetooth SIG 的相关数据手册定义的，做为GATT Client Characteristic Configuration的UUID

  这个配置值有读写权限，意味着GATT客户端可以改变这个值
  如果把这个值(通知开关使能)从0x0000 (notifications off) 改为0x0001 (notifications on), GATT服务端将开始发送这个特征值的通知到GATT客户端

| Handle | Type | Permissions | Value |
|---|---|---|---|
| 39 | 0x2800 (GATT Primary Service UUID) | Read | E0:FF (2 bytes)<br>(0xFFE0 = Simple Keys Service custom UUID) |
| 40 | 0x2803 (GATT Characteristic Declaration UUID) | Read | 10:29:00:E1:FF (5 bytes)<br>(0xFFE1 = Simple Keys Value custom UUID)<br>(0x0029 = handle 41)<br>(0x10 = characteristic properties: notify only) |
| 41 | 0xFFE1 (Simple Keys state) | (none) | 00 (1 byte)<br>(value indicates state of keys) |
| 42 | 0x2902 (GATT Client Characteristic Configuration UUID) | Read and Write | 00:00 (2 bytes)<br>(value indicates whether notifications or indications are enabled) |
| 43 | 0x2800 (GATT Primary Service UUID) | Read | A1:DD (2 bytes)<br>(0xDDA1 = Other Service custom UUID) |

# BLE (GATT):Client Commands

- 当两个BLE设备处于连接状态，客户端和服务端设备的通讯方式：

  - Discover Characteristic by UUID – 搜索服务端设备所能提供的所有匹配UUID规范的属性
  - Read Characteristic Value –使用指定的handle 读特征值
  - Write Characteristic Value –使用指定的handle 写特征值

- 除此之外，如果通知被使能，服务设备会自动向客户端设备发出下列信息：

  - Notification – 某个特征值被发送到客户端设备，而没有被读请求，并且不需要应答。
  - Indication – 某个特征值被发送到客户端，没有被读请求的情况下，但是在其他数据被发送之前必 须被确认。


We BCSphere™ Connect

**JUMA Technology**

www.bcsphere.org

bc·sphere

# Welcome to Join in

- www.bcsphere.org
- www.github.com/bcsphere
- www.bcstack.org
- www.github.com/bcstack
- BC QQ group: 303236541


聚码助力您的成功！

**JUMA Technology**

www.bcsphere.org