PADO

Labs

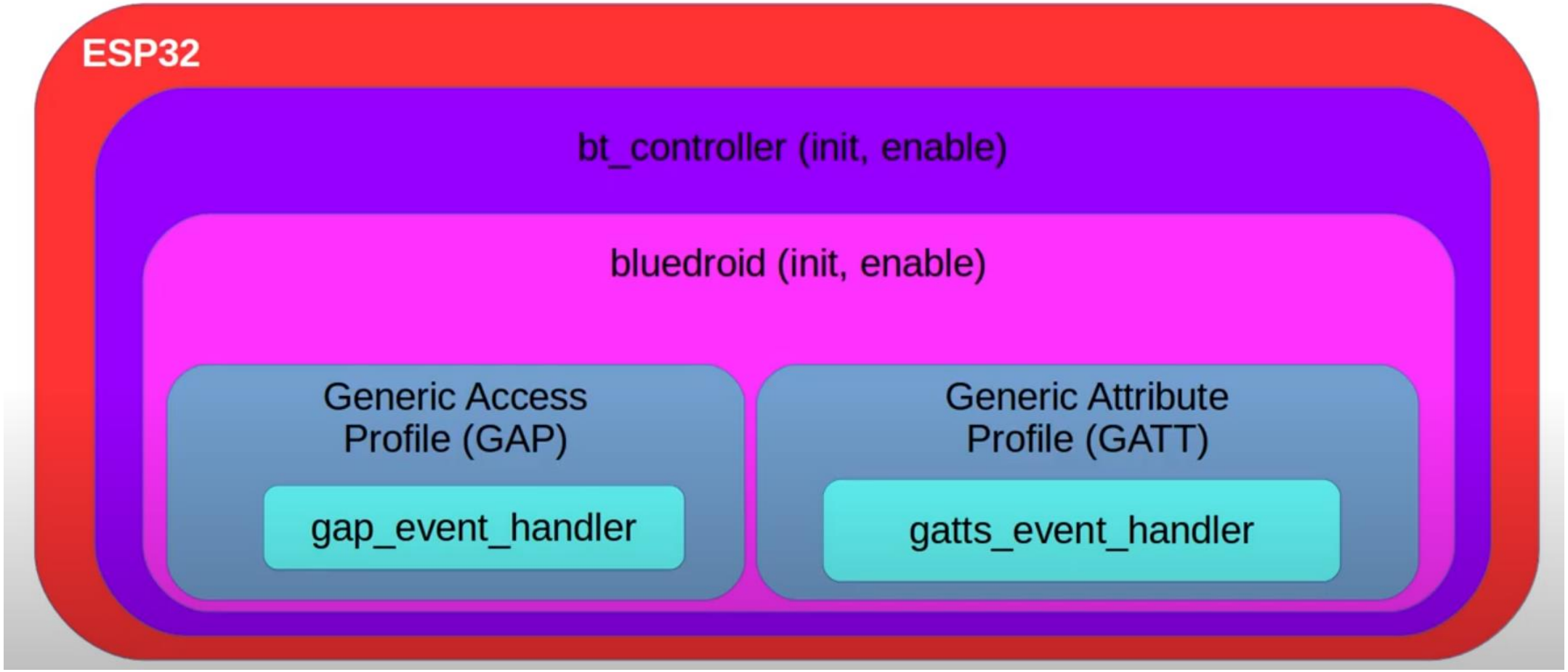
Microcontroladores Aplicados a IoT

Dilson Liukiti Ito

12 – Exercícios com Bluetooth Low Energy (BLE)

Microcontroladores Aplicados a IoT

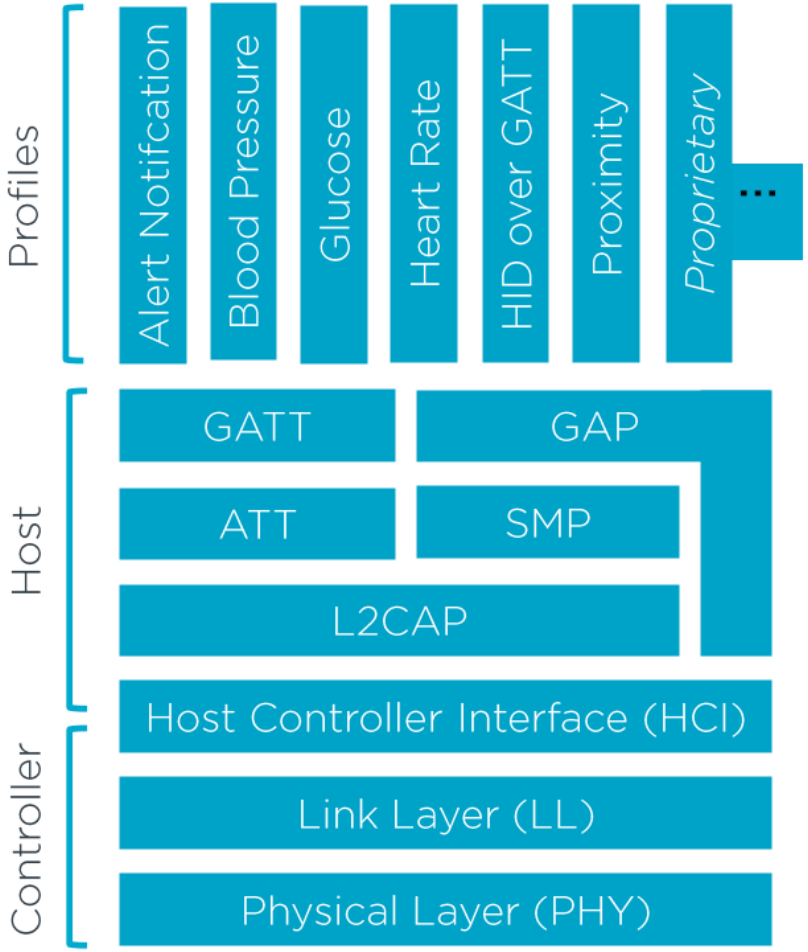
Bluetooth x ESP32



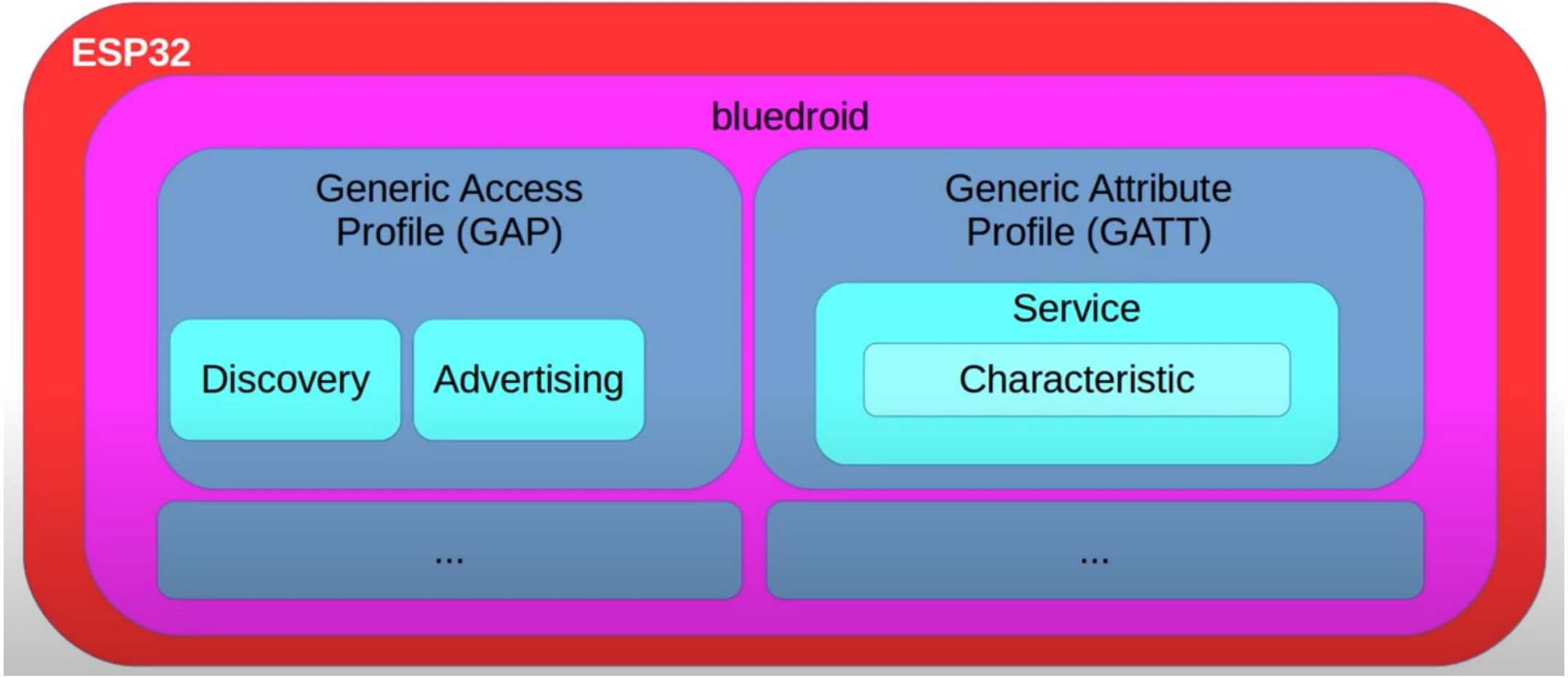
https://www.youtube.com/watch?v=FvIywveUVZw

GAP e GATT

- O **GAP** fornece uma estrutura padrão para **controlar um dispositivo** BLE,
- enquanto o **GATT** fornece uma estrutura padrão para **gerenciar dados** em um dispositivo BLE.

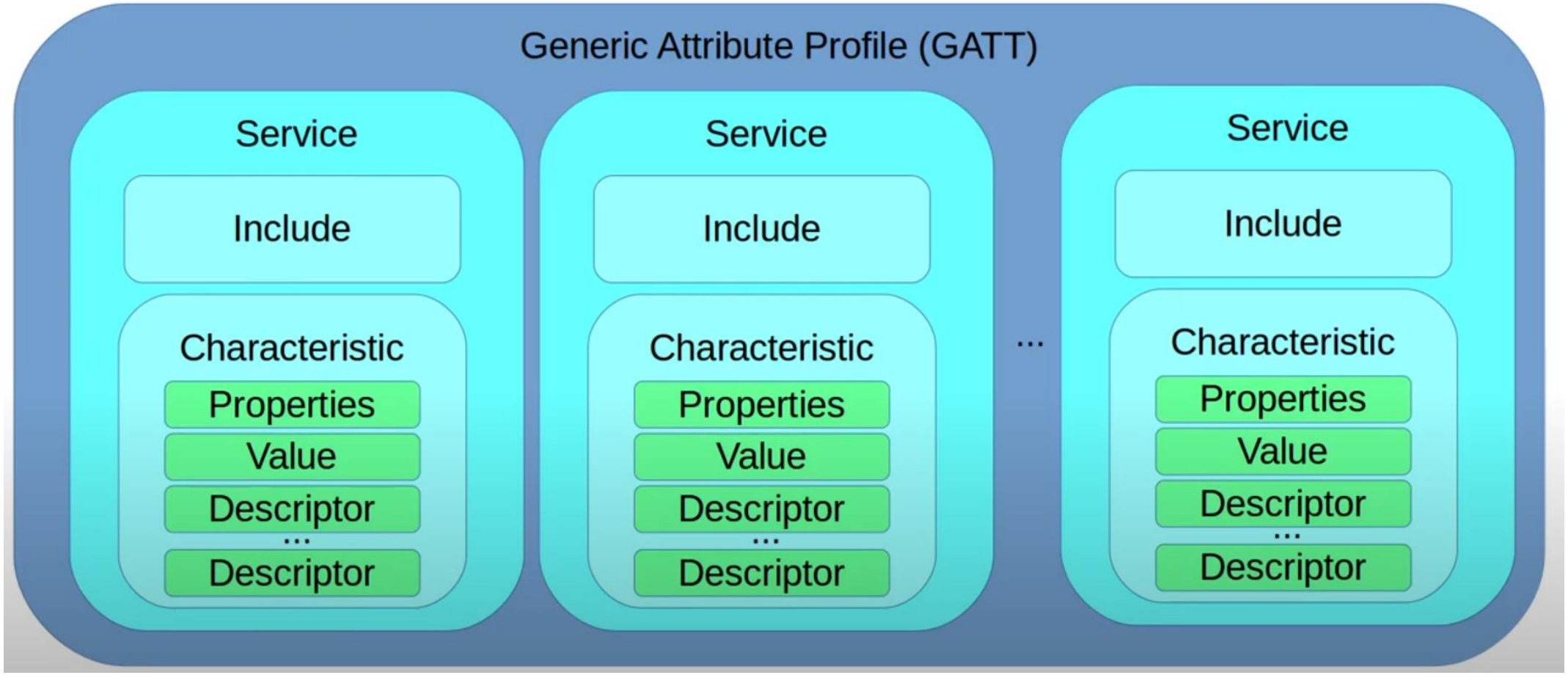


GAP e GATT (cont.)



https://www.youtube.com/watch?v=FvIywveUVZw

GATT



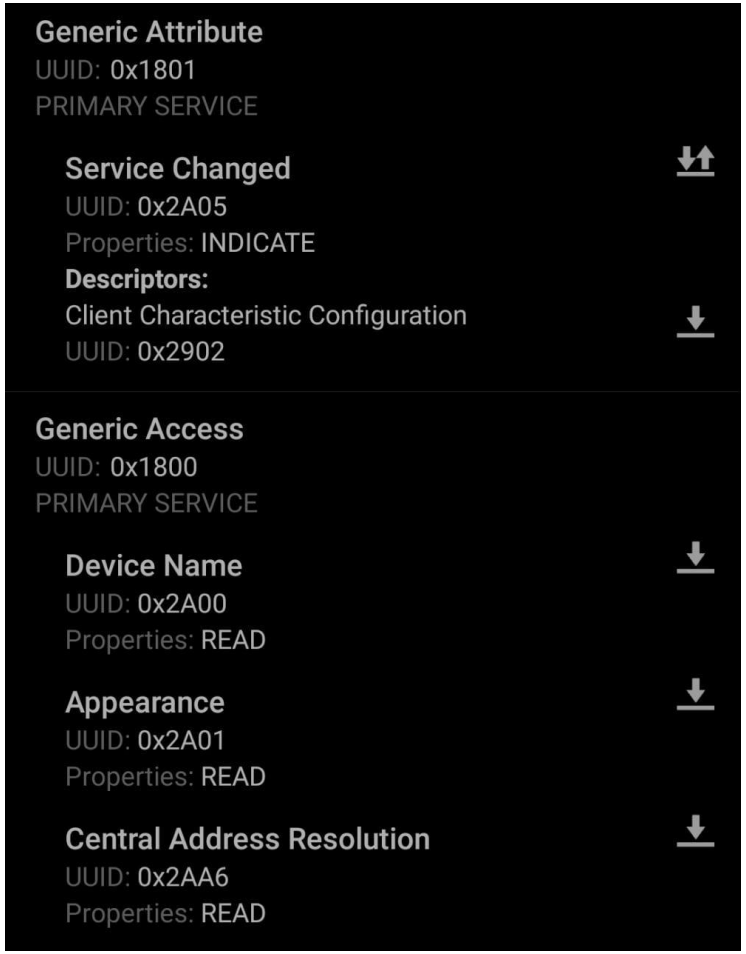
https://www.youtube.com/watch?v=FvIywveUVZw

Descriptor e Notify

- Habilitar ou desabilitar indicate e notify. Este tipo de descriptor é conhecido como CCCD (Client Characteristic Configuration Descriptor) 0x2902

GATT Descriptor	0x2900	Characteristic Extended Properties
GATT Descriptor	0x2901	Characteristic User Description
GATT Descriptor	0x2902	Client Characteristic Configuration
GATT Descriptor	0x2903	Server Characteristic Configuration

Generic Attribute e Generic Access



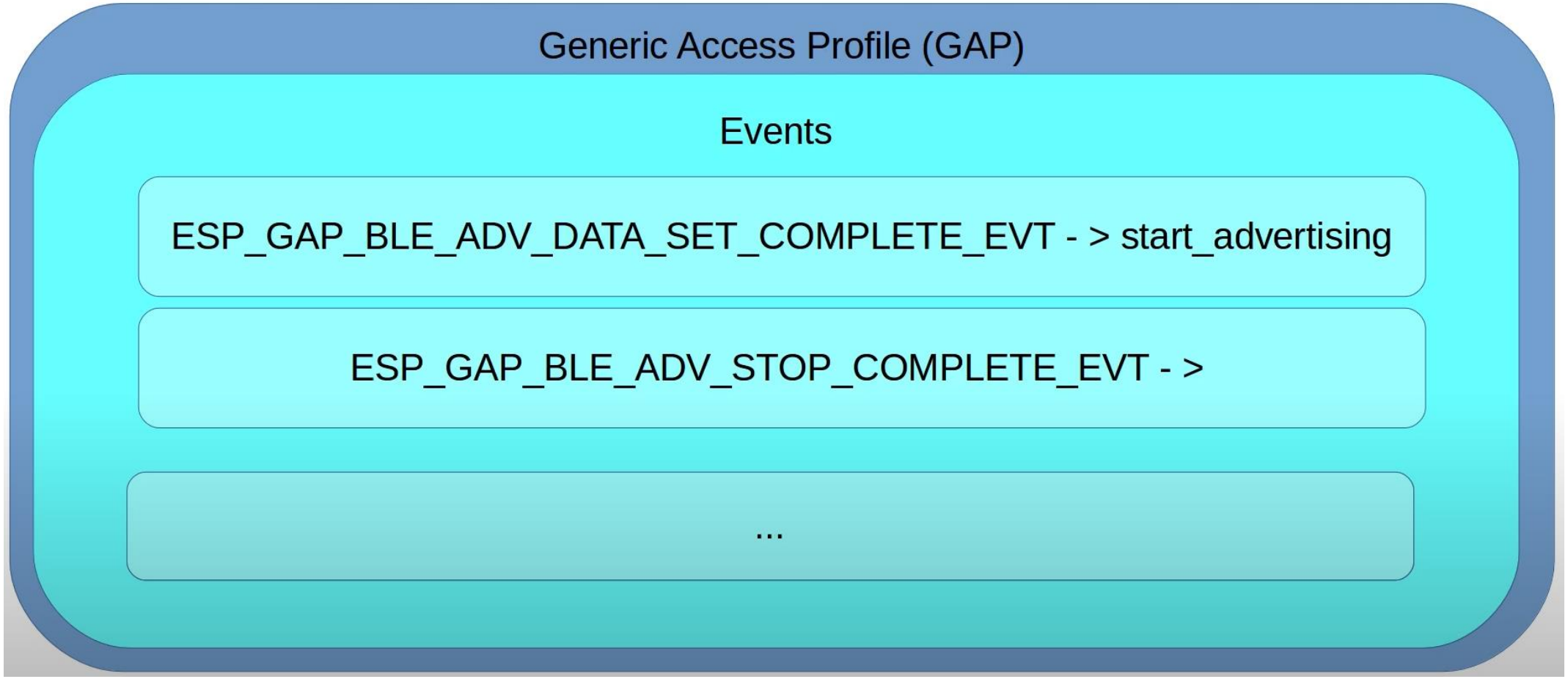
- BLUETOOTH SPECIFICATION Version 4.2 [Vol 3, Part G] page 577
- *Getting Started with Bluetooth Low Energy* by Townsend, Cufí, Akiba and Davidson, chapter 4 *GATT (Services and Characteristics)*, section *GATT Service page 73*

- BLUETOOTH SPECIFICATION Version 4.2 [Vol 3, Part C] page 390
- *Getting Started with Bluetooth Low Energy* by Townsend, Cufí, Akiba and Davidson, chapter 3 *GAP (Advertising and Connections)*, section *GAP Service page 50*

https://www.bluetooth.com/specifications/specs/core-specification-4-2/

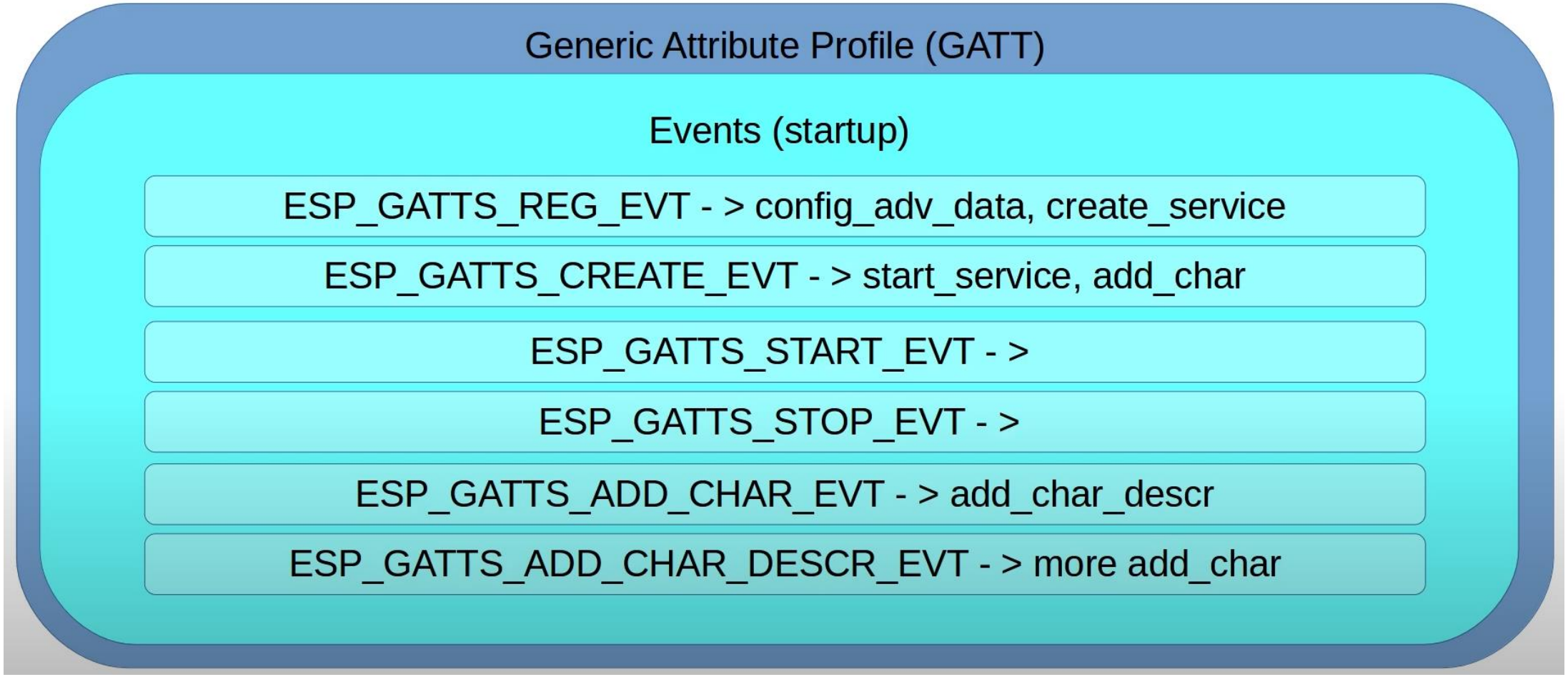
https://github.com/lisider/my_book/blob/master/communicationProtocol/bluetooth/Getting%20Started%20with%20Bluetooth%20Low%20Energy.pdf

GAP: Eventos



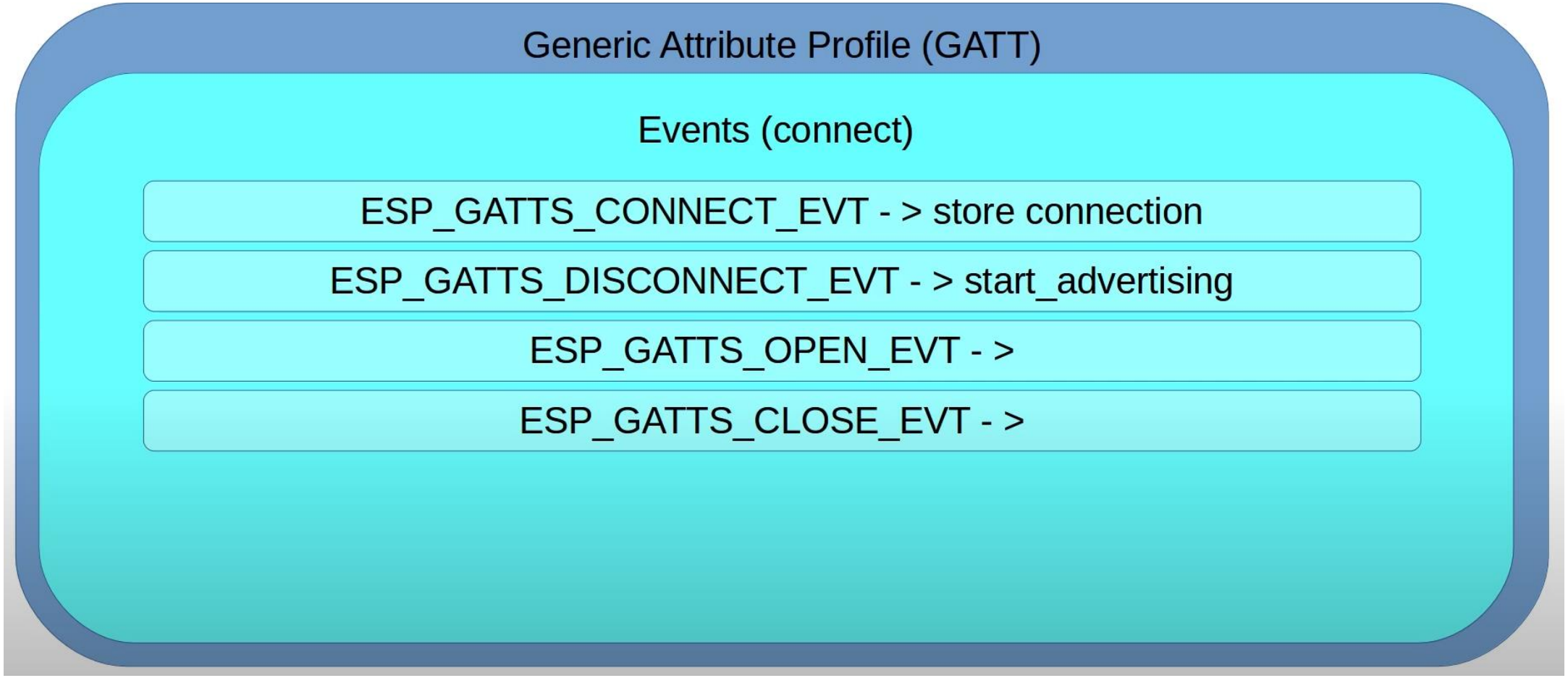
https://www.youtube.com/watch?v=FvIywveUVZw

GATT: Eventos de inicialização



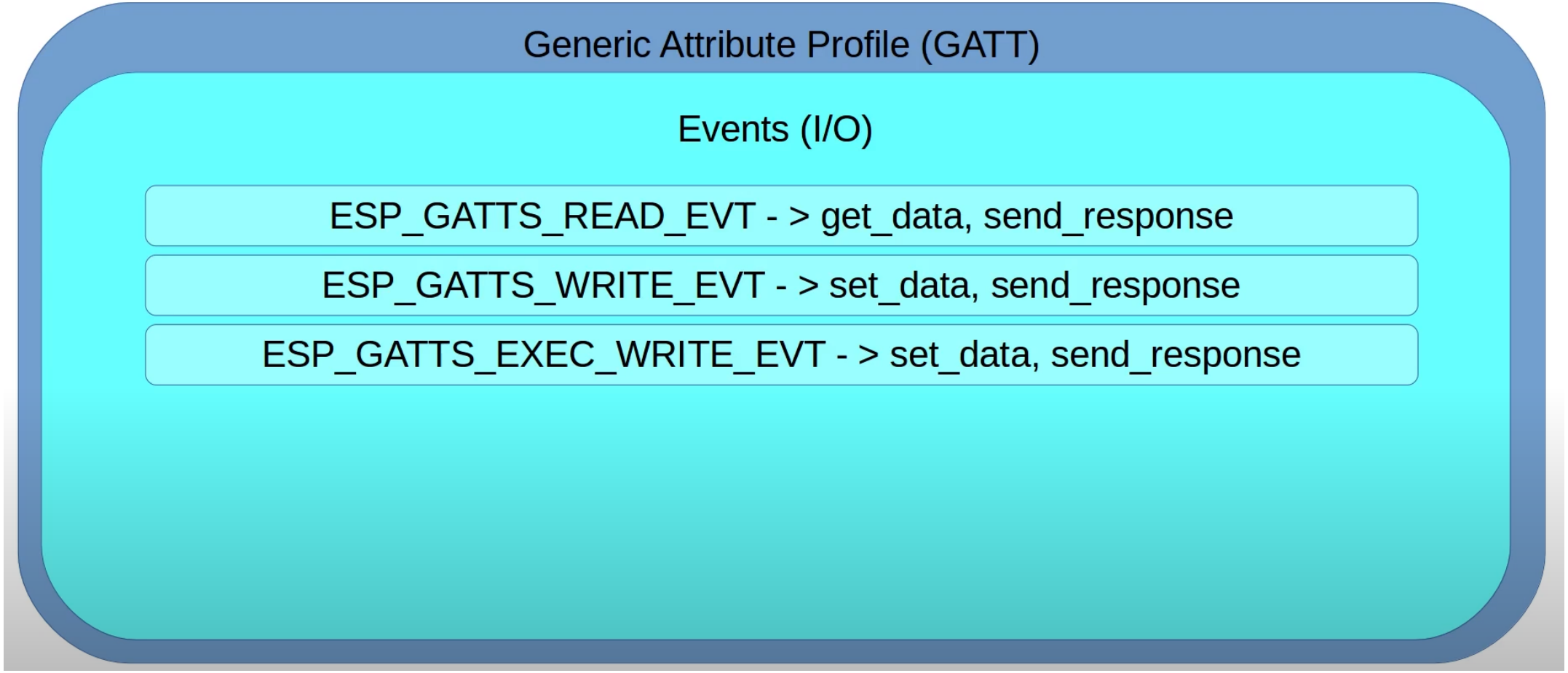
https://www.youtube.com/watch?v=FvIywveUVZw

GATT: Eventos de Conexão



https://www.youtube.com/watch?v=FvIywveUVZw

GATT: Eventos de IO



https://www.youtube.com/watch?v=FvIywveUVZw

Exemplo 1

- Utilize o arquivo ibeacon_scanner.c

Exemplo 2

- Baixe os arquivos da Aula12.zip
- Utilize o arquivo gatt_client.c
- Sequência de eventos mapeado no Figma:
  - https://www.figma.com/file/KLzzT0ebLztwzWtOSBdSkx/gatt_client.c?node-id=8%3A275

Exemplo 2 (cont.)

- Utilizar a TAG para informação mais útil:

```
//https://www.decompile.com/cpp/faq/file_and_line_error_string.htm
#define STRINGIFY(x) #x
#define TOSTRING(x) STRINGIFY(x)
#define GATTC_TAG "line " TOSTRING(__LINE__)
```

- Funções que convertem enums em strings:

```
char* gattc_events_to_name(esp_gattc_cb_event_t event);
char* gap_events_to_name (esp_gap_ble_cb_event_t event);
char* gatt_conn_to_name(esp_gatt_conn_reason_t reason);
```

Exercício 1

Faça os exercícios em **dupla (de dois)**!

1) Scanner de iBeacon

- Um ESP32 deverá ser um iBeacon com major e minor conhecidos. Utilize o exemplo da Aula 11;
- Outro ESP32 (modifique o exemplo desta aula) será um scanner que exibirá por meio de LEDs a distância calculada pelo RSSI do iBeacon do passo anterior, seguindo a tabela:

LED Ligado	Distância
Verde	< 1m
Amarelo	> 1 m & < 3 m
Vermelho	> 3 m

Exercício 1 (cont)

- A distância com o valor de RSSI pode ser calculada segundo a fórmula a seguir:

$$\text{Distância} = 10 \text{ ^ } [(\text{MP} - \text{RSSI}) / 20]$$

MP = Measured Power: RSSI medido a 1 metro de distância

- Faça a calibração do *Measured Power* do iBeacon para corresponder ao valor de RSSI medido quando o iBeacon e o Scanner da dupla estiverem a 1 metro de distância.

https://medium.com/beingcoders/convert-rssi-value-of-the-ble-bluetooth-low-energy-beacons-to-meters-63259f307283

https://iotandelectronics.wordpress.com/2016/10/07/how-to-calculate-distance-from-the-rssi-value-of-the-ble-beacon/

Exercício 2

2) Cliente GATT para Battery Service

- Um ESP32 será o GATT Server (batt.c) como o exemplo da Aula11;
- O outro será um GATT Client (utilize o exemplo desta aula) que irá conter um botão que toda vez que for pressionado deverá receber por notificação 3 valores do serviço de bateria (1 notificação a cada 10 segundos), fazer a média dos 3 valores e exibir o resultado do estado atual da bateria com os LEDs Vermelho, Amarelo e Verde.

Exercício 2 (cont.)

- Lembrando que o valor da bateria recebido pelo BLE estará em porcentagem. Transforme novamente em tensão.
  - 0% = 2,8V e 100% = 4,2V
- apresente os valores com os LEDS da seguinte maneira:

LED Ligado	Faixa de Tensão
Verde	4,2V à 3,75V
Amarelo	3,75 à 3,4V
Vermelho	3,4 à 2,8V

PADO

Labs