

# Bluetooth LE

Not your father's Bluetooth



Antoni Kędracki

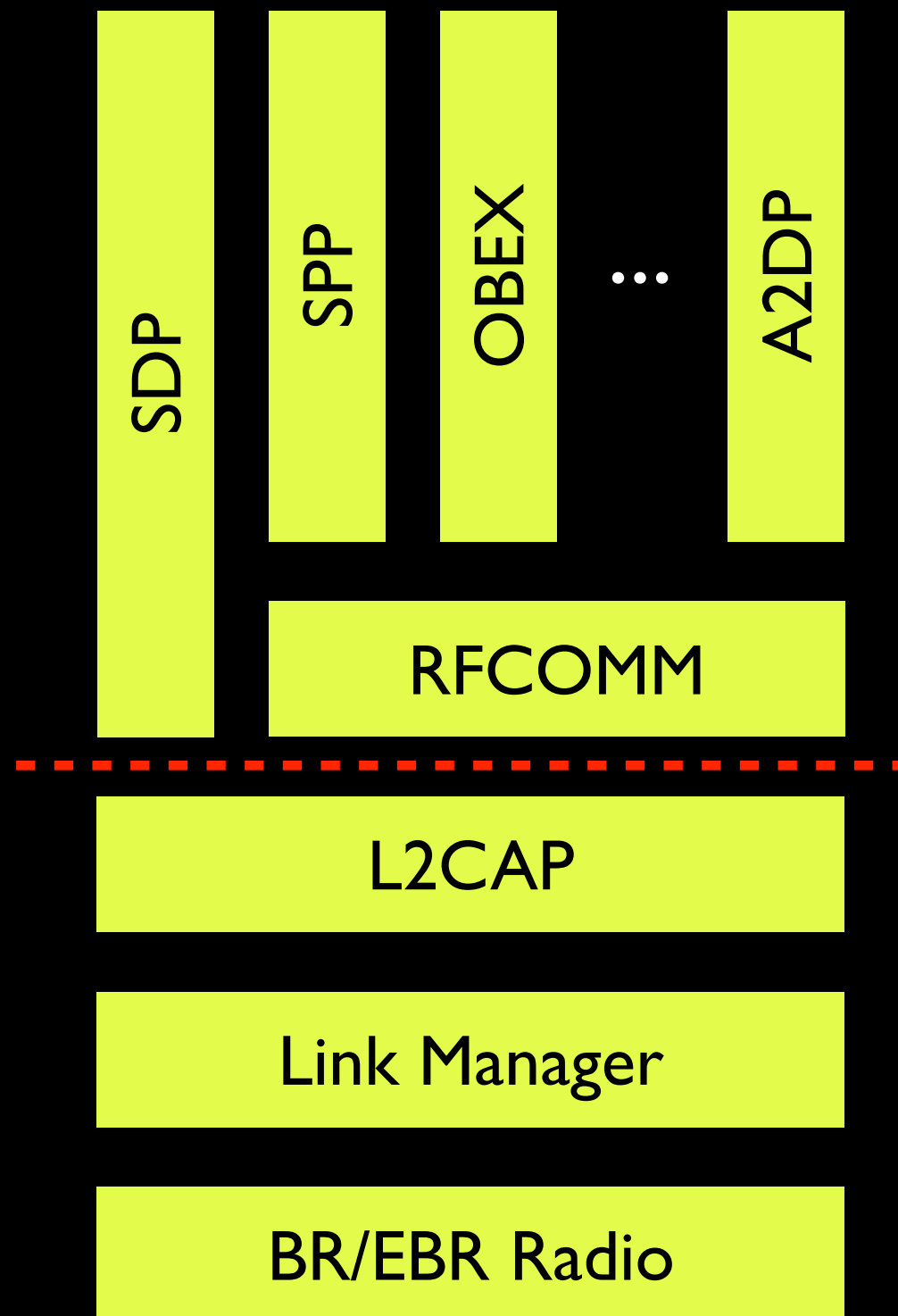
[@spherefoundry](#)

- Czym jest Bluetooth 4.0?
- Dlaczego powstał?
- Jak to działa?
- A ile urządzeń to wspiera?

# Trochę Historii

- Stworzone w 1994
- Sieć PAN
- Wspólne radio
- Wiele profili (RFCOMM)
- Parowanie?!

# “Klasyczny” Bluetooth



# Bluetooth 4.0

- Dwie konfiguracje
  - Basic/Extended Bit Rate - BR/EBR
  - Low Energy - LE
- Cel:
  - Obniżenie wymagań prądowych
  - Ułatwienie implementacji
- Rozwiązanie
  - Uproszczone radio
  - Nowy sposób “rozgłaszania”
  - Pojedynczy profil (GATT)

*“Penny, everything is better with bluetooth!”*

Sheldon Cooper, The Big Bang Theory

# BLE w liczbach

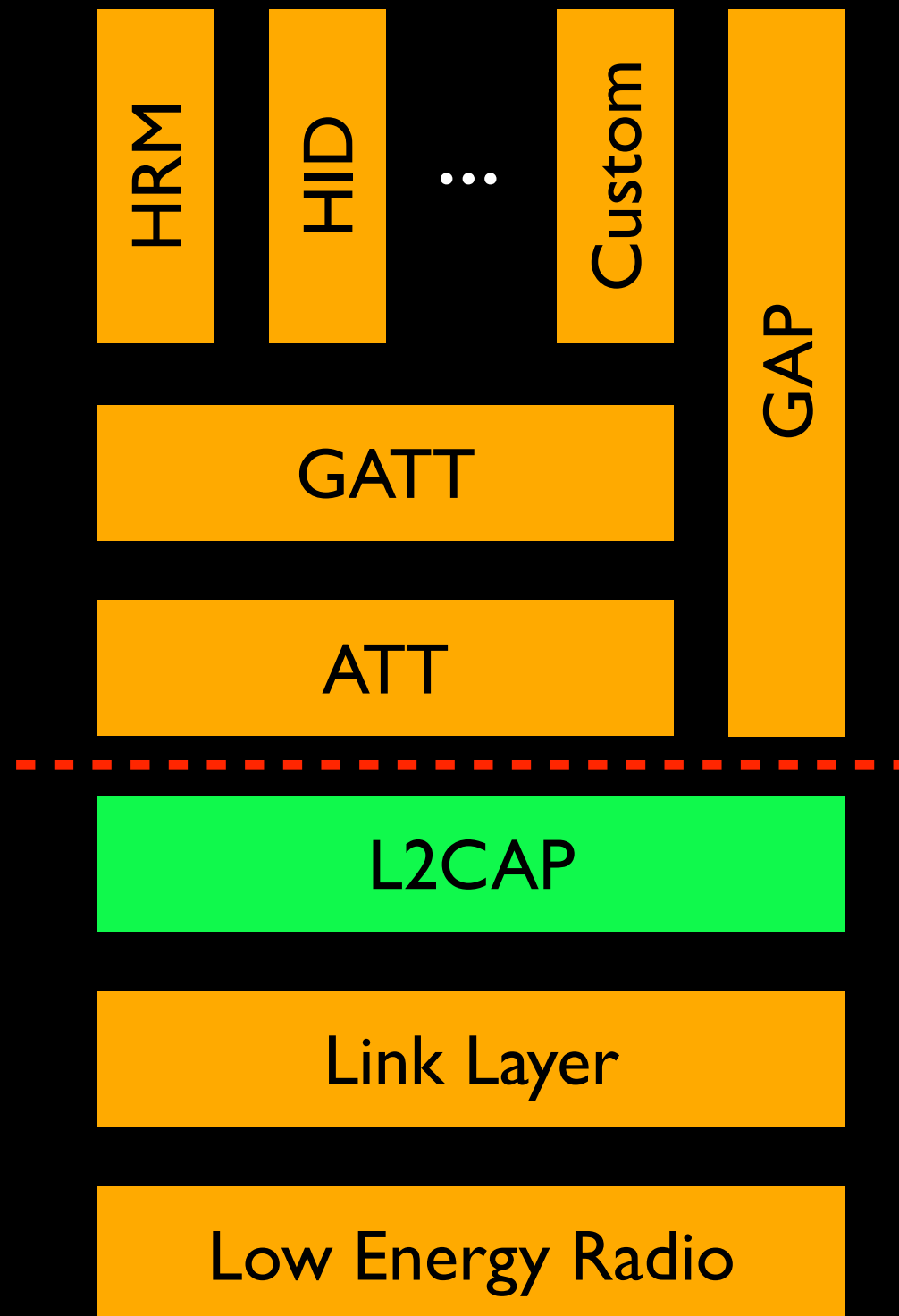
	rozmiar	waga	moc/praca
BLE112	18mm x 12mm x 2.3mm	~2g	TX: 27mA sleep: 0.5uA
CR2032	19,9mm	3g	225mAh

# Na czym to chodzi?

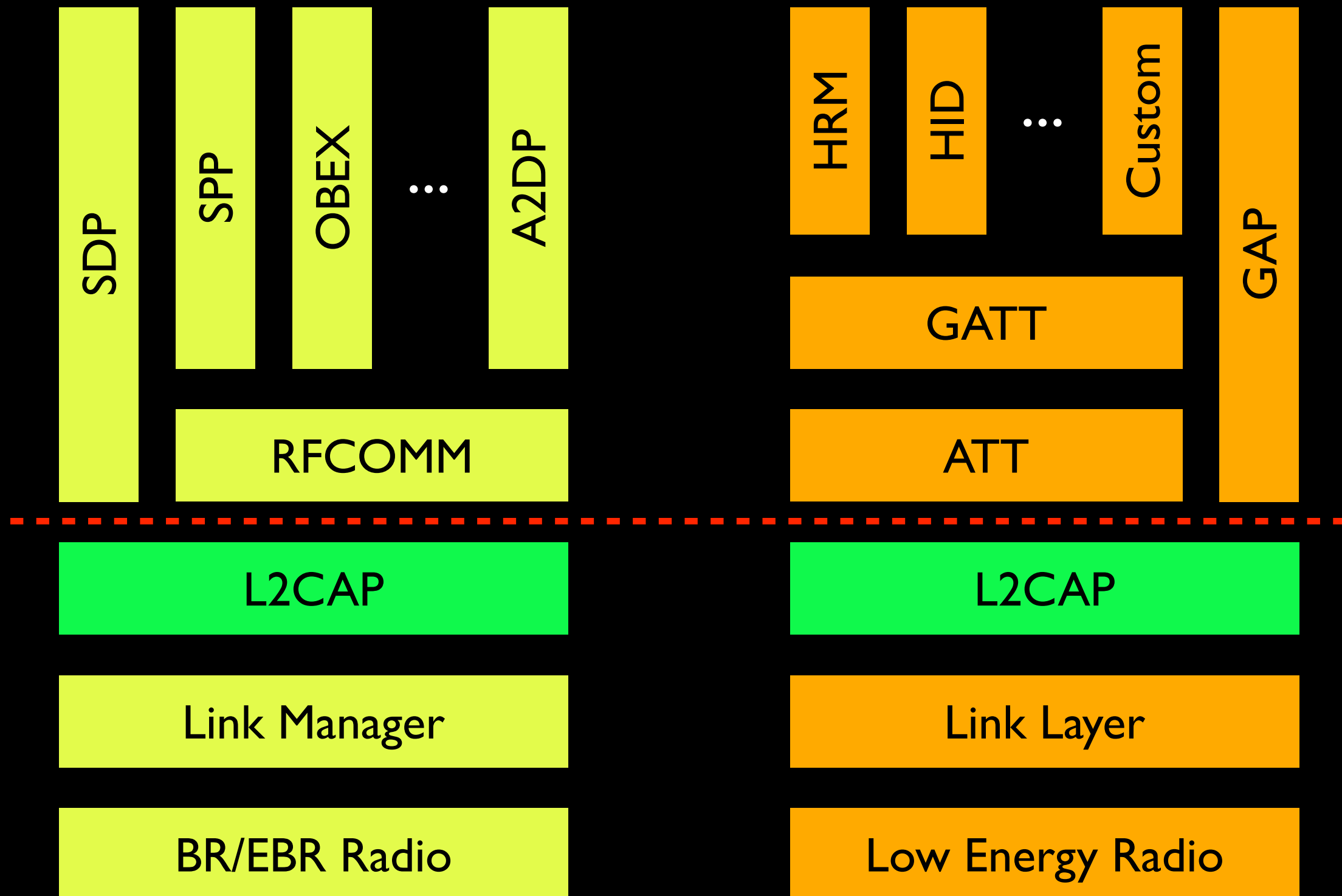
- Apple (iOS5+, OS X 10.7+)
  - iPhone 4S+, iPad 3+, iPad Mini, iPod Touch 5Gen
  - MacBook Air i Mac Mini mid2011+, reszta mid2012+
- Android (lipiec 2013 - 4.3+)
  - Google Nexus 4, Nexus 7 (2013)
  - Samsung Galaxy S3, S4, S4 Mini, Note 2
  - Sony 2013+
  - HTC 2012+
  - Motorola 2012+



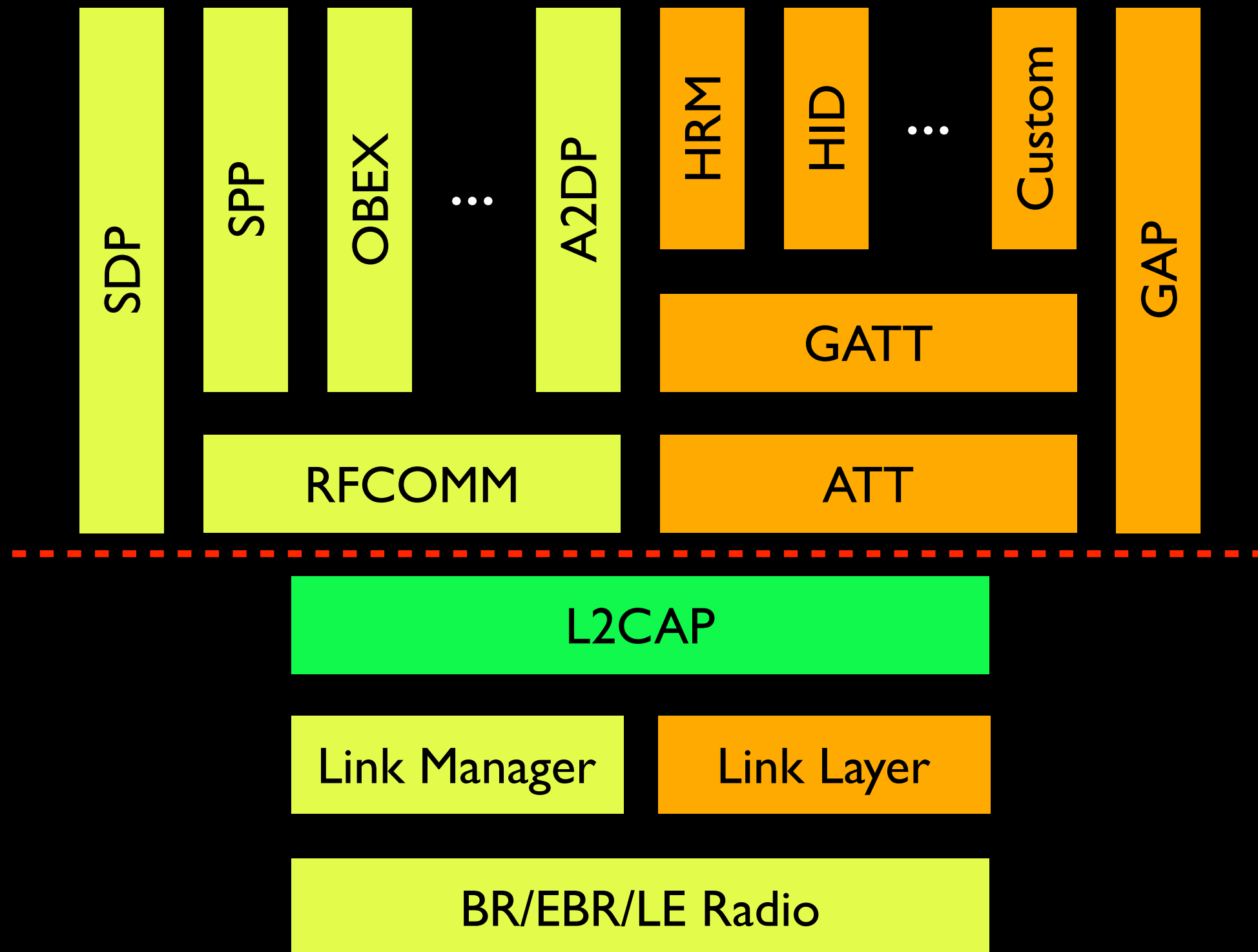
# Bluetooth “Smart”



# “Smart” vs. “Klasyczny”



# Bluetooth “Smart Ready”



# GATT

“One protocol to rule them all”

# GATT

- Zdefiniowane role:
  - Serwer
    - wystawia model
    - informuje o schemie tego modelu
    - przykładowo: czujnik tętna, termometr, myszka
  - Klient
    - konsumuje model
    - przykładowo: telefon, samochód

# GATT

- Key/Value Store na sterydach
  - Klucz + wartość == charakterystyka
  - Grupa charakterystyk == serwis
  - Metadane charakterystyki == deskryptor
- Wszystko ma identyfikator UUID (16/128 bit)

# Serwis pomiaru tętna

- Heart Rate (0x180D)
  - Body Sensor Location (0x2A38) [read]
  - Heart Rate Control Point (0x2A39) [write]
  - Heart Rate Measurement (0x2A37) [notify]



subscribe 0x2A37



notify 0x2A37



notify 0x2A37



notify 0x2A37





DEMO

# Charakterystyka pomiaru tętna

- Heart Rate Measurement (0x2A37)

<b>Flags</b>	8bit	1 - format 2,3 - contact status 4 - energy expanded 5 - RR interval 6,7,8 - reserved
<b>HR Value</b>	uint8/uint16	bpm
<b>Energy Expended</b>	uint16	kJ
<b>RR Interval</b>	uint16	1/1024s

Bluetooth 4.0 LE nie wspiera SPP?!

# Radio

“ET phone’s home”

# Problem wielodostępu

- Bluetooth operuje na paśmie ISM 2.4GHz
- Tak samo jak WiFi, ZigBee, i wiele innych
- Wszystko to się zakłuca

# Frequency Hopping w LE

- Oparty o inkrementacje o stałą wartość i operacje modulo
- Mapa używanych kanałów
- Algorytm
  - $unmappedChannel = (lastUnmappedChannel + hopIncrement) \bmod 37$
  - użyj  $unmappedChannel$  jeśli jest w mapie używanych kanałów
  - w przeciwnym razie użyj kanału pod pozycją  $remappingIndex = unmappedChannel \bmod numUsedChannels$

# Rozgłaszanie

“Ja Brzoza, ja Brzoza, Grab, jak mnie słyszysz?”

# Rozgłaszanie

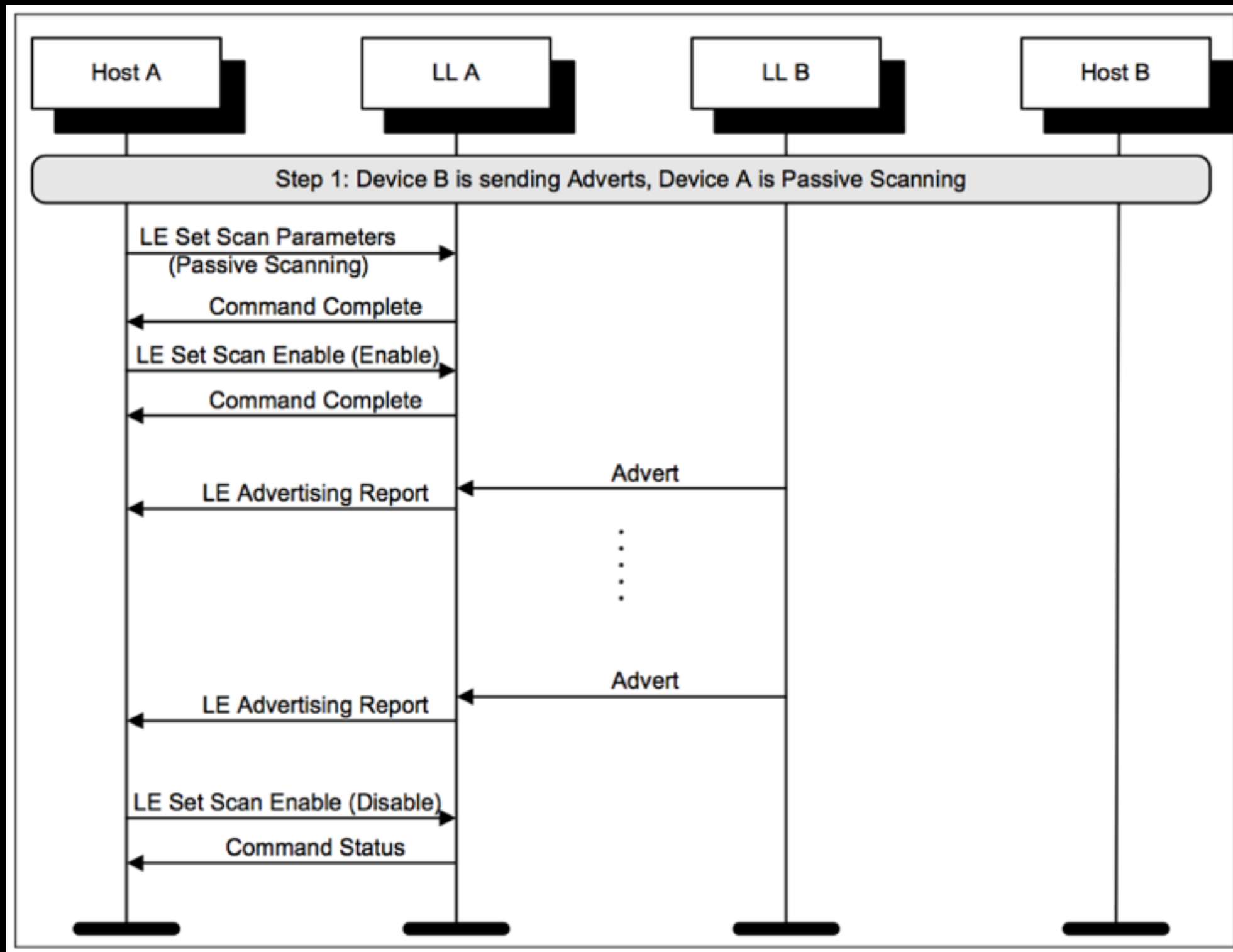
- Jak wszystko w LE zoptymalizowane pod kątem zużycia energii
- Pasywne i aktywne skanowanie
- Możliwość umieszczenia własnych danych w pakietach rozgłoszeniowych



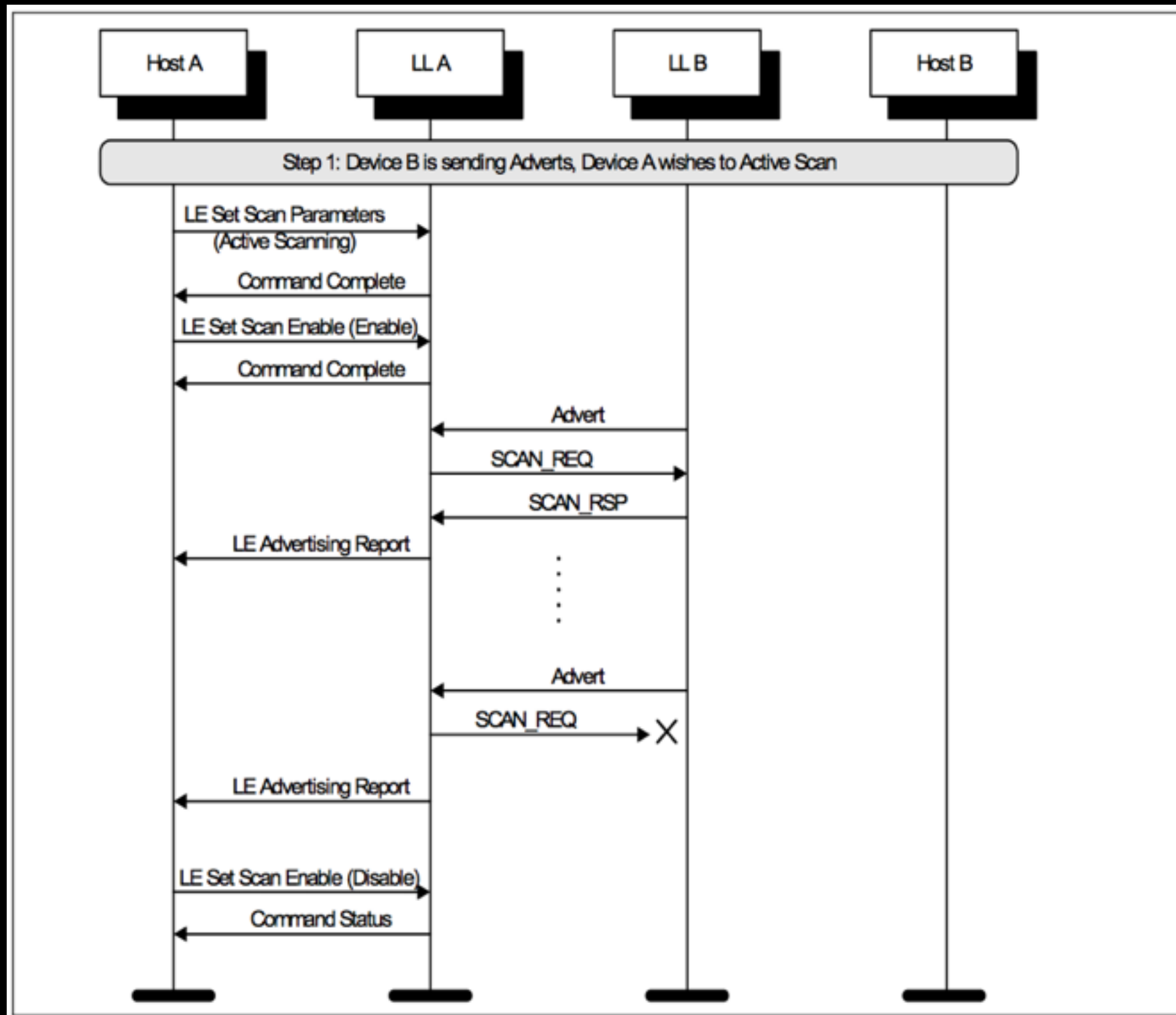
# Kanały rozgłaszające

- 3 (z 40) kanałów przeznaczone wyłącznie do rozgłaszania
- algorytm:
  - posiedź na kanale A przez x
  - posiedź na kanale B przez x
  - posiedź na kanale C przez x
  - i od nowa!

# Pasywne rozgłaszanie

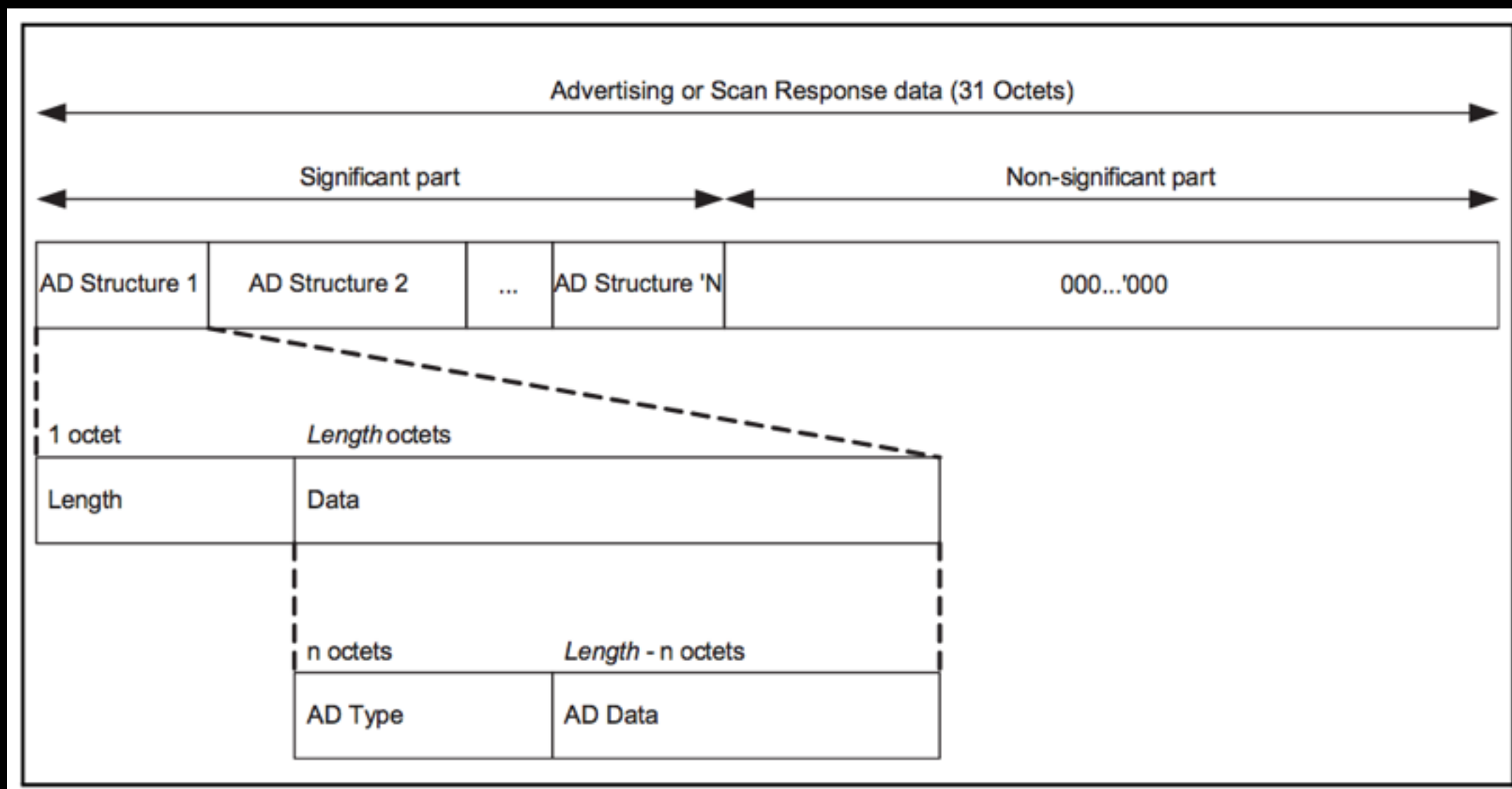


# Aktywne rozgłaszanie



# Pakiety AD

- Do 31 bajtów danych
- Elastyczny format



# GAP + GATT

- [GATT] serwis Generic Access(0x1800)
  - Device Name (0x2A00)
- [GAP] AD flags
  - typ 0x02-0x07 - UUID dostępnych serwisów GATT
  - typ 0x08-0x09 - nazwa

# Rozgłaszanie serwisów GATT

length	type	flags
0x02	0x01	0x1A
2	FLAGS	

length	type	UUID 1	UUID2
0x09	0x03	0x1800	0x180D
9	complete list of 16bit UUID	Generic Access	HRM

length	type	name
0x07	0x09	0x41 6E 74 6F 6E 69
7	local name	“Antoni”

# iBeacon

- Cały mechanizm opiera się na rozgłaszaniu
- Odległość liczona na podstawie RSSI odbioru i wartości referencyjnej

length	type	flags
0x02	0x01	0x1A
2	FLAGS	

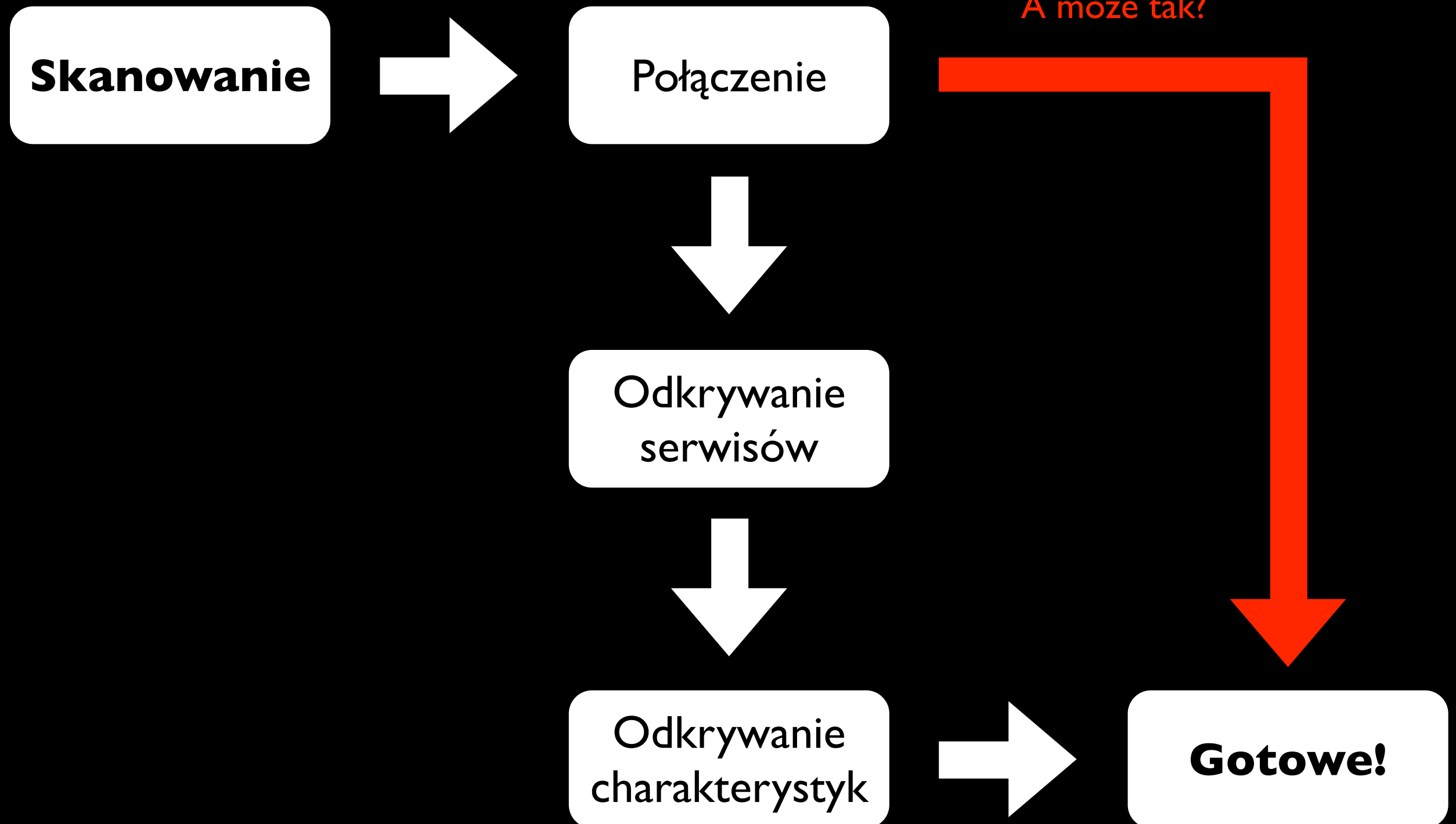
length	type	vendor	?	beacon UUID	major	minor	?
0x1A	0xFF	0x4C00	0x0215	0xA1A2A3A4 B1B2 C1C2 D1D2 E1E2E3E4E5E6	0x000E	0x000F	0xC5
26	CUSTOM	Apple					

# Nawiązanie połączenia

“You talkin' to me?”



# “Przygotowanie” do interakcji



# Odkrywanie modelu

- zamiana UUID na uchwyt
- doprecyzowanie które z opcjonalnych charakterystyk jest wspierane

# Bezpieczeństwo



# “z czy bez?”

- Domyślnie urządzenia w BLE komunikują bez szyfrowania i autentykacji
- charakterystryki mogą zadeklarować że operacje na nich mogą przebiegać wyłącznie po zabezpieczonym połączeniu

# Parowanie

- Polega na autentykacji i wymianie kluczy szyfrujących (AES-128)
- procesem rządzi “master”
- “slave” może zgłosić chęć parowania
- trzy metody autentykacji:
  - just works
  - passkey
  - OOB

# Czy to oby wystarczy?

- process ustalania/wymiany kluczy jest podatny na podsłuch
- postronne urządzenie może wymusić ponowne jego przeprowadzenie
- passkey ma 6 cyfr?!
- cała nadzieja w OOB lub szyfrowaniu w warstwie aplikacji

# Producenci

“You can have it in every color! As long as it's black”

# Master

- Apple (iOS5+, OS X 10.7+)
  - iPhone 4S+, iPad 3+, iPad Mini, iPod Touch 5Gen
  - MacBook Air i Mac Mini mid2011+, reszta mid2012+
- Android (lipiec 2013 - 4.3+)
  - Google Nexus 4, Nexus 7 (2013)
  - Samsung Galaxy S3, S4, S4 Mini, Note 2
  - Sony 2013+
  - HTC 2012+
  - Motorola 2012+

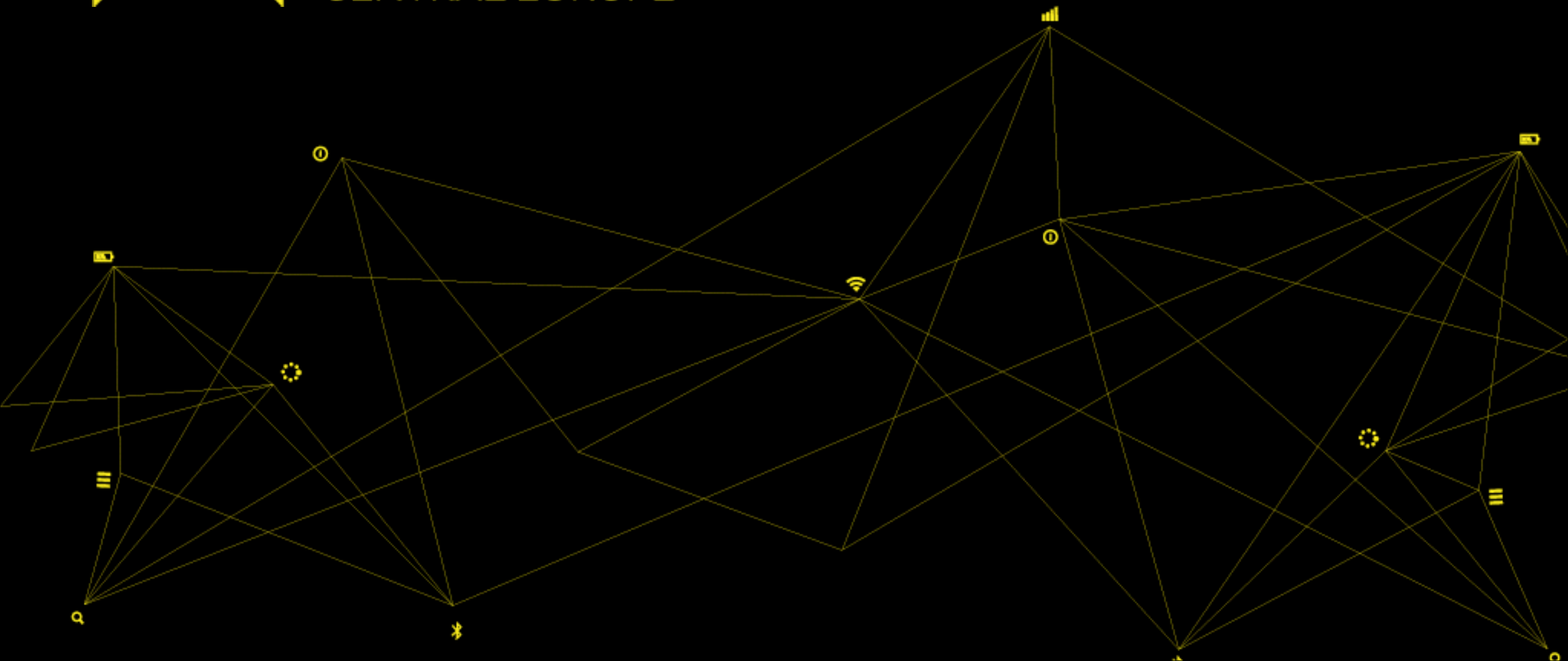


- Wsparcie po stronie iOS spowodowało eksplozję ilości urządzeń
  - Fitbit, Nike+ Fuelband 2.0, etc
  - Dice+
  - HRM
  - tokeny
- SmartWatch?

# Podziękowania i Źródła

- Bluetooth Core Specification 4.0 Volumes 1-6
- Smartphone designed by George Agpoon from The Noun Project

# Q&A





20% zniżki dla wszystkich!!

MobileWarsawWroclaw