

Blue2thprinting: {blue-[tooth}-printing]

Answering the question "WTF am I even looking at?!"

Xeno Kovah

OpenSecurityTraining2 (ost2.fyi)

& Dark Mentor LLC (darkmentor.com)



About Me

- 75% of my time is spent making free (as in beer), open access, and *open source* (CreativeCommons licensed) classes for a non-profit I started, **OpenSecurityTraining2 (ost2.fyi)**





About Me

- 75% of my time is spent making free (as in beer), open access, and *open source* (CreativeCommons licensed) classes for a non-profit I started, **OpenSecurityTraining2 (ost2.fyi)**
- 25% of my time doing consulting and research for **Dark Mentor LLC**
 - The research is for fun, but is *also a trojan horse* to get me into conferences to tell you about OST2 ;)



DARK MENTOR





OST2 Crew

In order of appearance



Gal Zaban

RE3011 ~6 hours



Piotr Król

Arch4021 ~6 hours
Arch4031 ~6 hours



Kc Udonsi

Vulns1001 ~15 hours



Michał Żygowski

Arch4021 ~6 hours

Xeno Kovah

Arch1001 ~28 hours, Arch2001 ~27 hours
Arch4001 ~14 hours, HW1101 ~6 hours
Vulns1001 ~15 hours, Vulns1002 ~23 hours



Thaís Moreira Hamasaki

RE3201 ~6 hours



Cedric Halbronn

Dbg3011 ~6 hours
Arch2821 ~5 hours
Exp4011 ~33(!) hours



Sina Karvandi

Dbg3301 ~16 hours



What I Want To Know:

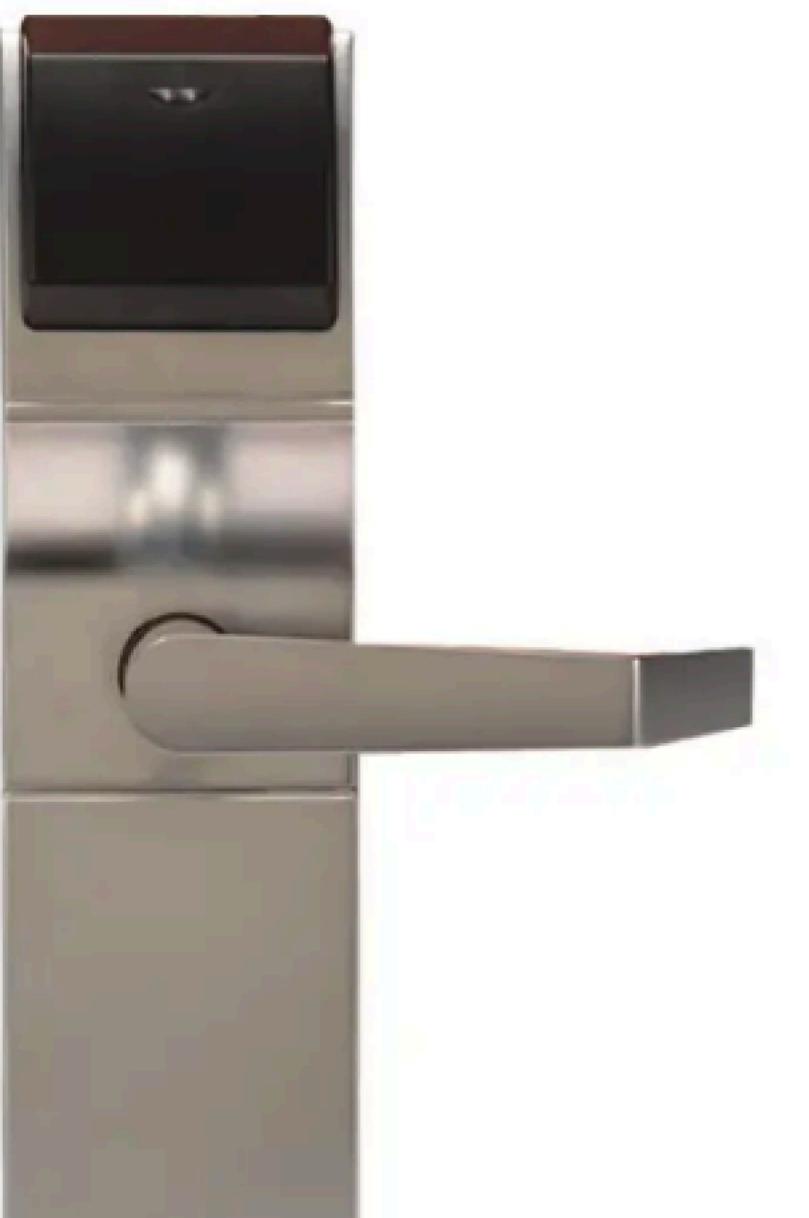
What Bluetooth Chip Is Inside Any Device



DST2
.FYI



DST2
.FYI





DST2
.FYI



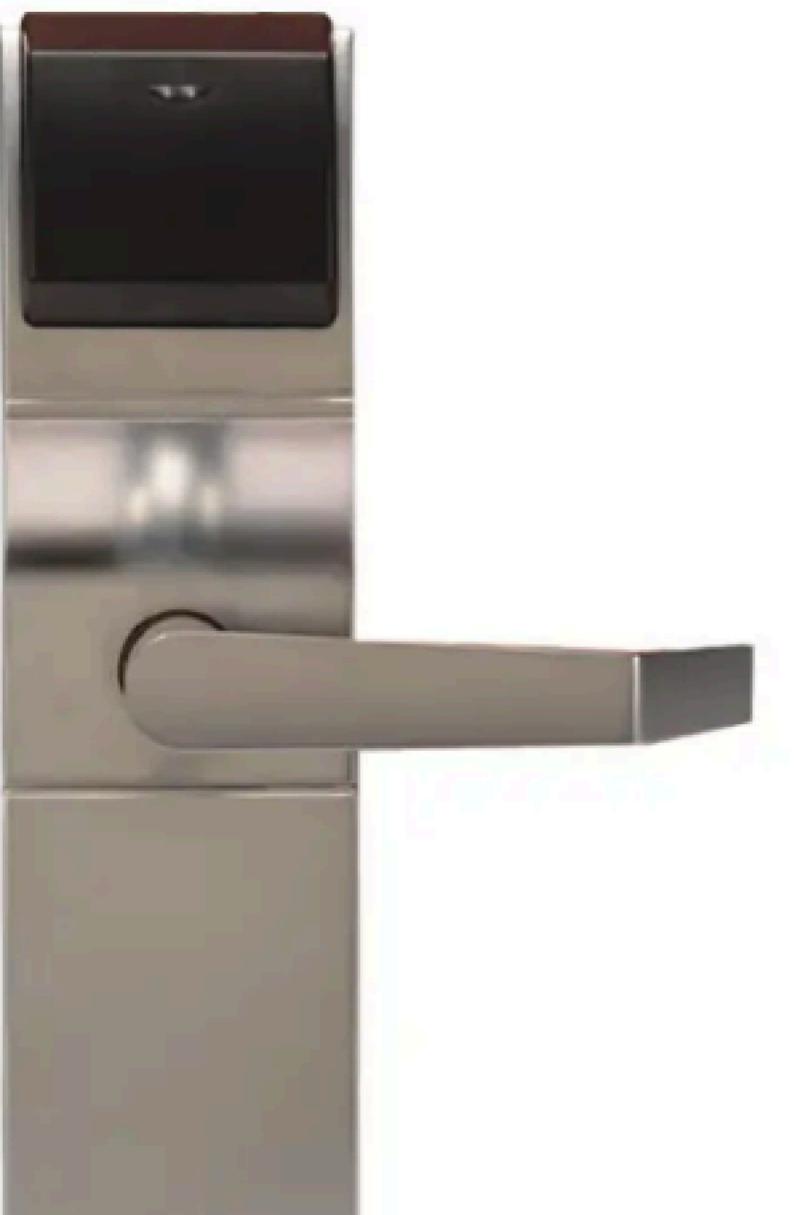
 TEXAS
INSTRUMENTS



 BROADCOM®



 SILICON LABS



?



Why I Want To Know It:

So I Know if it's Vulnerable To a Firmware-Level Exploit



DST2
.FYI









DST2
.FBI



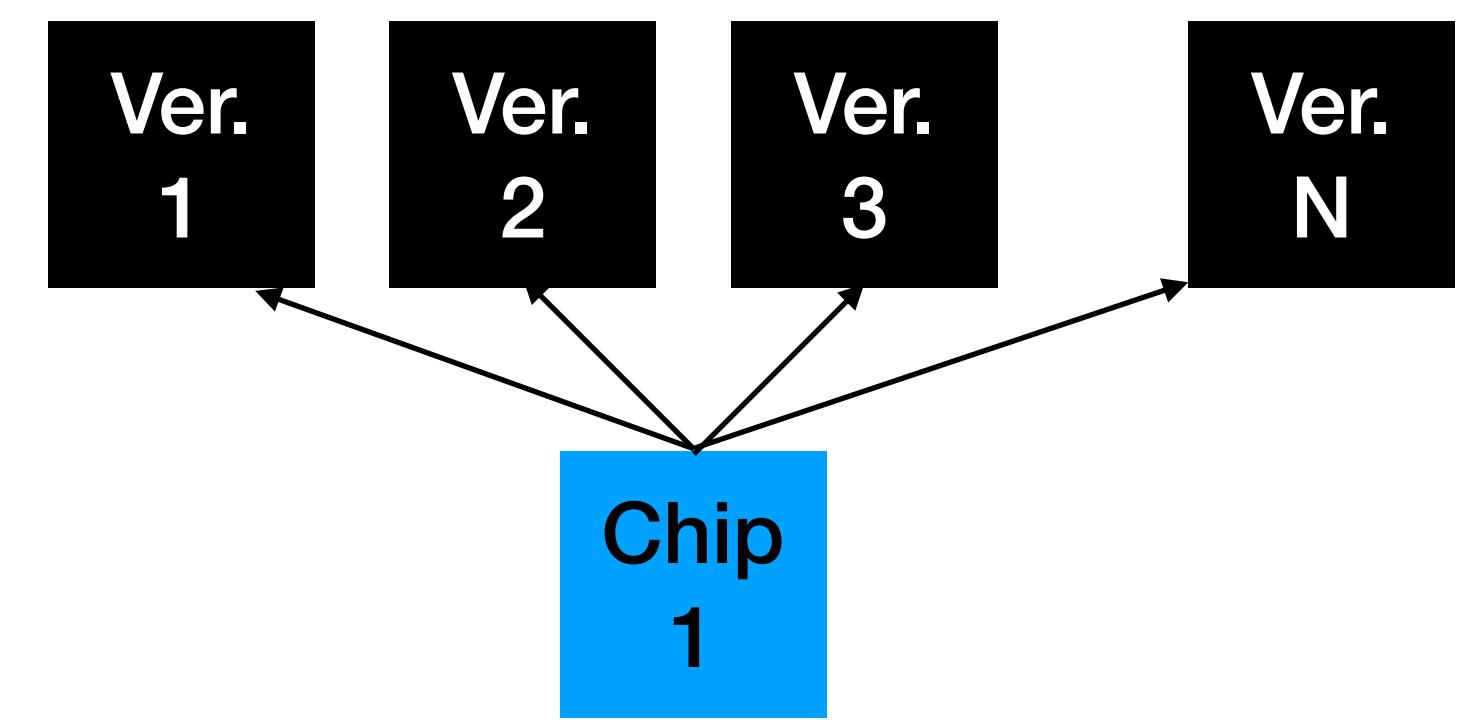


DST2
.FBI





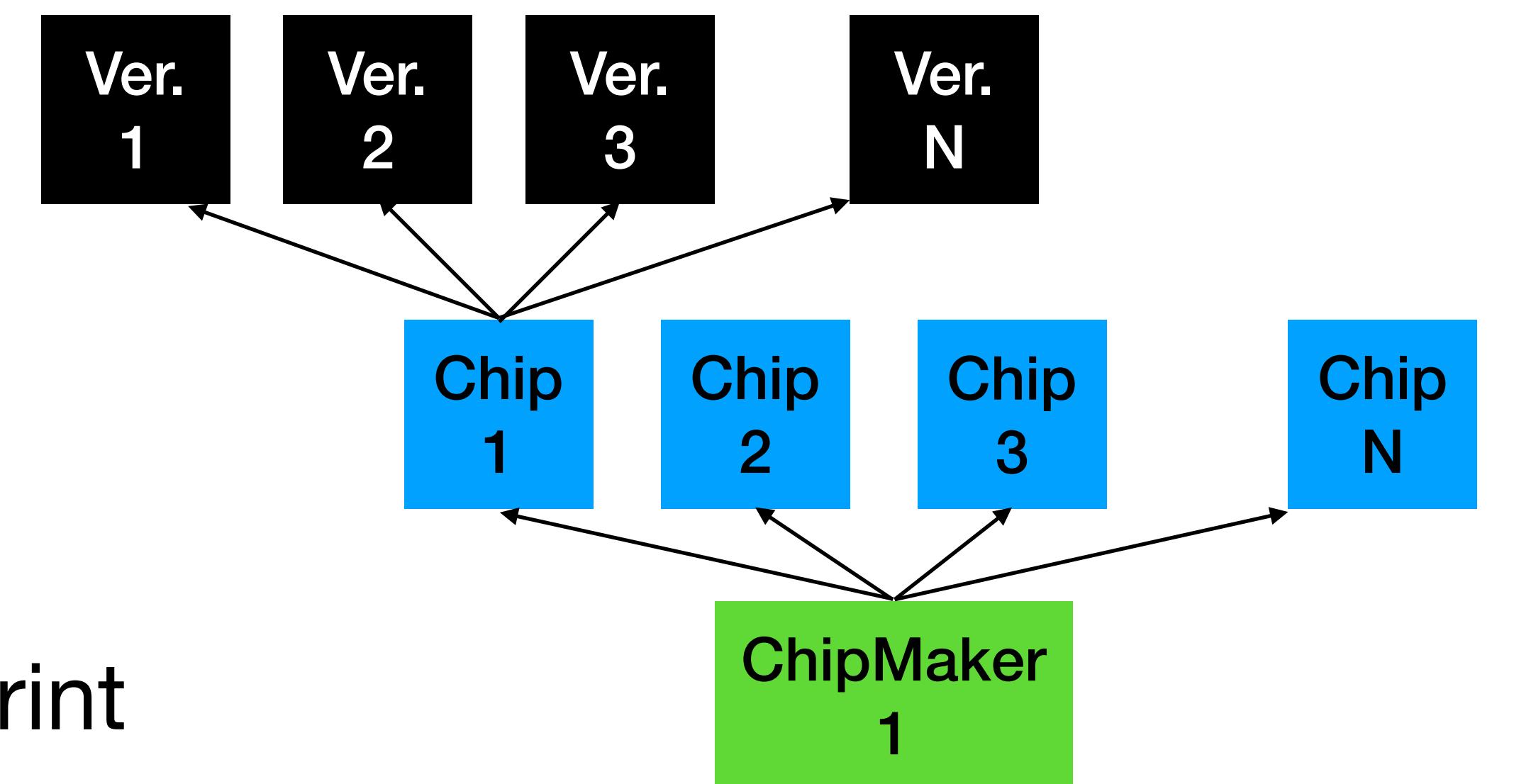
VersionPrint



ChipPrint



VersionPrint

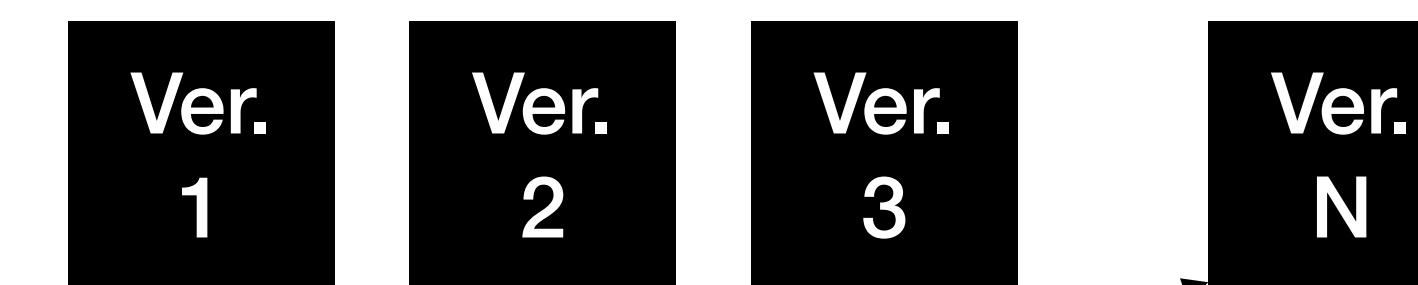


ChipPrint

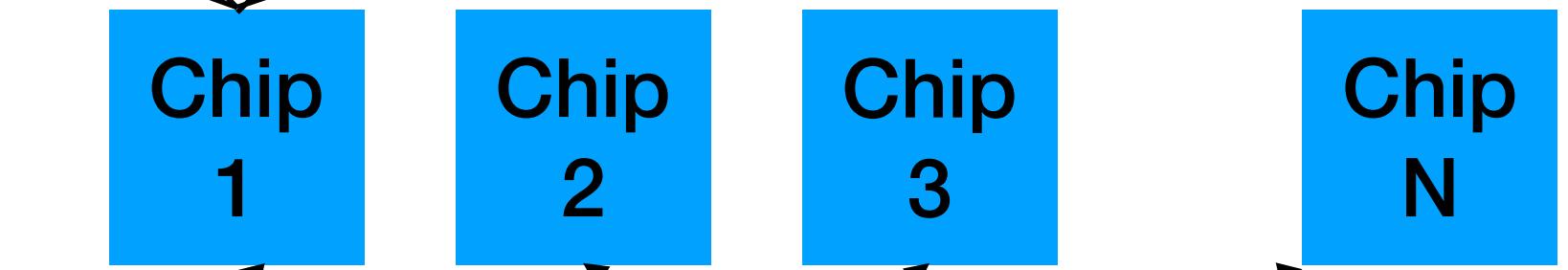
ChipMakerPrint



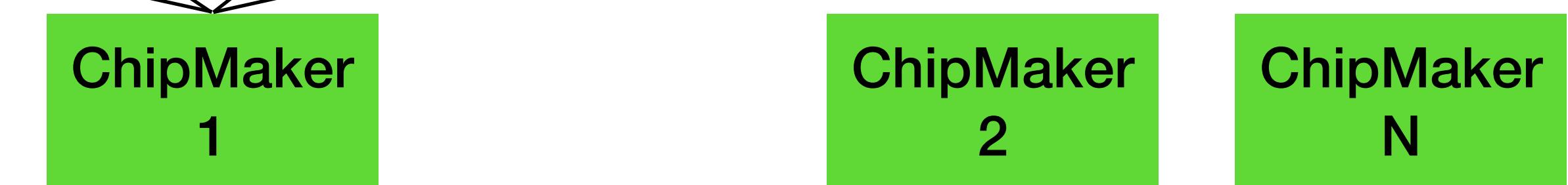
VersionPrint



ChipPrint

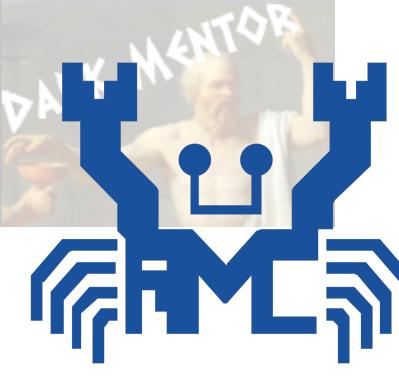


ChipMakerPrint



ModuleMakerPrint

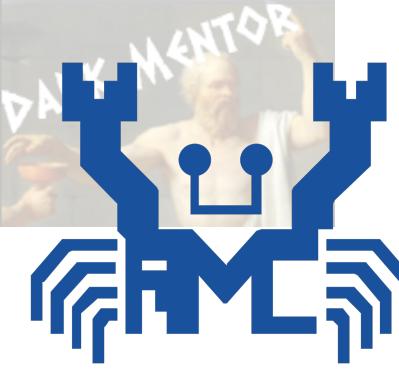




REALTEK

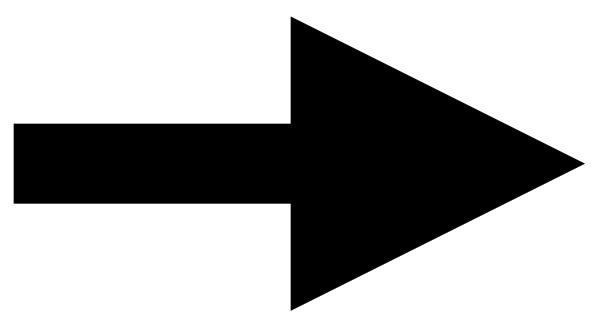
ChipMaker
1

DST2
.FYI



REALTEK

ChipMaker
1



Ai-Thinker Technology

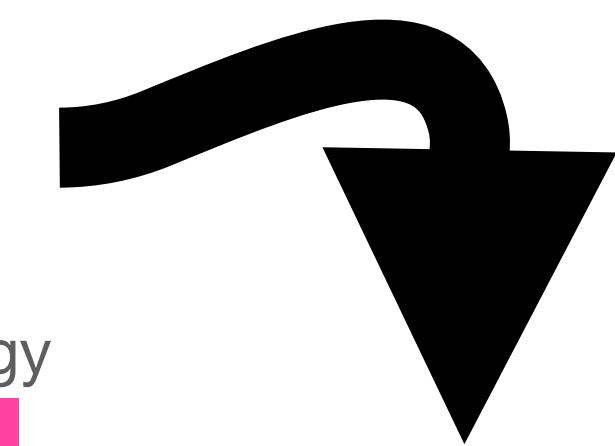
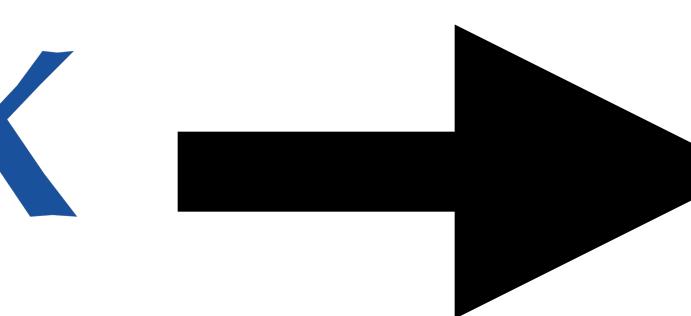
ModuleMaker
1

OST2
.FBI



REALTEK

ChipMaker
1



Ai-Thinker Technology

ModuleMaker
1

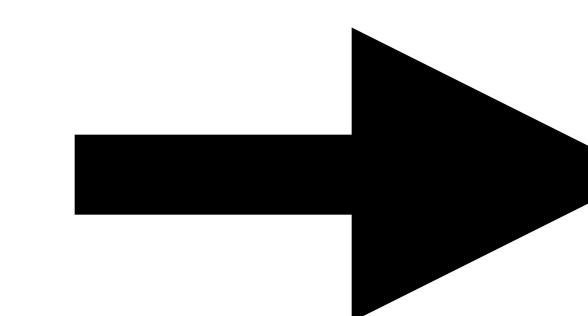
DST2
.FYI

Model	BW12		BW15	BW16 (hot)
Picture				
Chip	RTL8710BX	RTL8710BX	RTL8720CF	RTL8720DN
Package	SMT-16	/	SMD-16	SMD-16
Size	24 x16x 3mm (LxWxH) ±0.2mm	50.5*29.2*3.3 (±0.2) mm	24*16*3(±0.2)MM	24*16*3(±0.2)MM
Antenna	on-board PCB/ IPEX antenna	on-board PCB/ IPEX antenna	on-board PCB/ IPEX antenna	on-board PCB/ IPEX antenna
Frequency range	2.4 Module Hz	2.40 Module Hz	2.40 Module GHz	2400-248 Module 180-5825MHz
Bluetooth	1	2	3	4
Operating temperature	-20~+85° C	-20 °C~70°C	-40 °C ~ 85 °C	-20 °C ~ 70 °C
Storage temperature	-40 ~125°C	-40°C~125°C	-40 °C ~ 125 °C , < 90%RH	-40 °C ~ 125 °C , < 90%RH
Power supply	3.3±10%V	5V ~ 12V	Voltage 3.0 V ~3.6 V, current >500 mA	Voltage 3.0V ~ 3.6V, Typical 3.3V, Current >450mA
Interface	UART,I2C, SPI, GPIO, SWD, PWM	UART	UART/GPIO/ADC/PWM /IIC /SPI	UART/GPIO/ADC/PWM/IIC/SPI/SWD

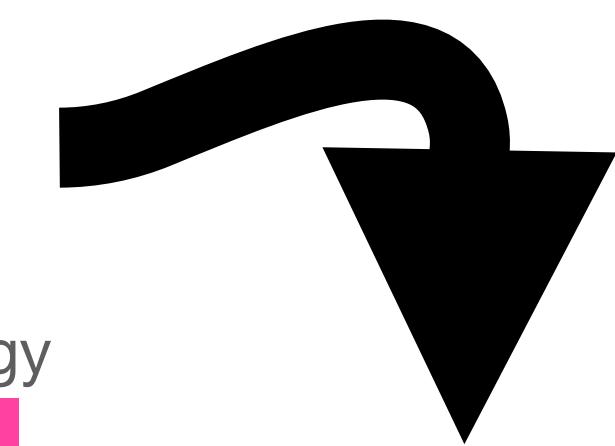


REALTEK

ChipMaker
1



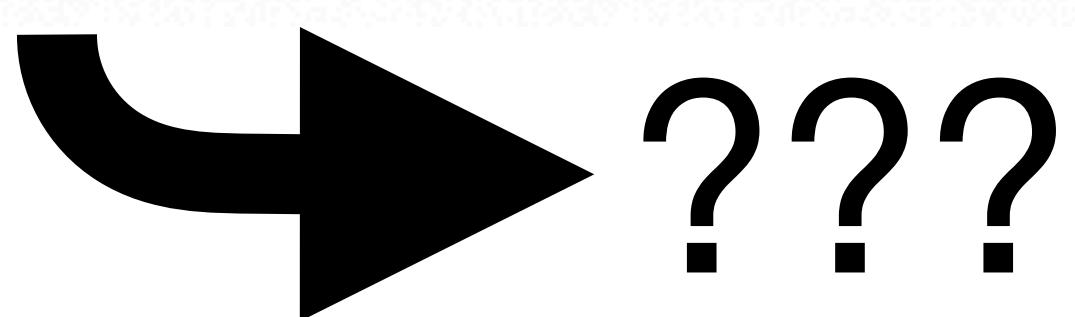
Ai-Thinker Technology



ModuleMaker
1

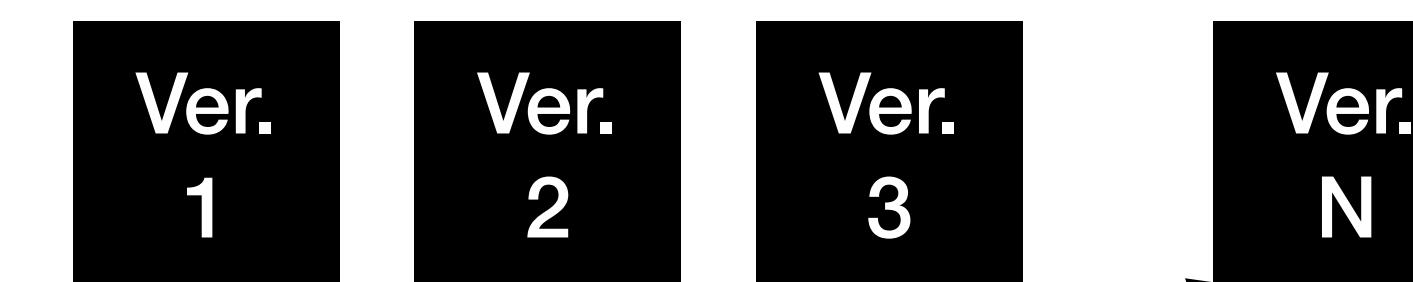
DST2
.FYI

Model	BW12		BW15	BW16 (hot)
Picture				
Chip	RTL8710BX	RTL8710BX	RTL8720CF	RTL8720DN
Package	SMT-16	/	SMD-16	SMD-16
Size	24 x16x 3mm (LxWxH) ±0.2mm	50.5*29.2*3.3 (±0.2) mm	24*16*3(±0.2)MM	24*16*3(±0.2)MM
Antenna	on-board PCB/ IPEX antenna	on-board PCB/ IPEX antenna	on-board PCB/ IPEX antenna	on-board PCB/ IPEX antenna
Frequency range	2.4 Module Hz	2.4G Module Hz	2.4G Module GHz	2400-2484 Module 180-5825MHz
Bluetooth	1	2	3	4
Operating temperature	-20~+85° C	-20 °C~70°C	-40 °C ~ 85 °C	-20 °C ~ 70 °C
Storage temperature	-40 ~125°C	-40°C~125°C	-40 °C ~ 125 °C , < 90%RH	-40 °C ~ 125 °C , < 90%RH
Power supply	3.3±10%V	5V ~ 12V	Voltage 3.0 V ~3.6 V, current >500 mA	Voltage 3.0V ~ 3.6V, Typical 3.3V, Current >450mA
Interface	UART,I2C, SPI, GPIO, SWD, PWM	UART	UART/GPIO/ADC/PWM /IIC /SPI	UART/GPIO/ADC/PWM/IIC/SPI/SWD

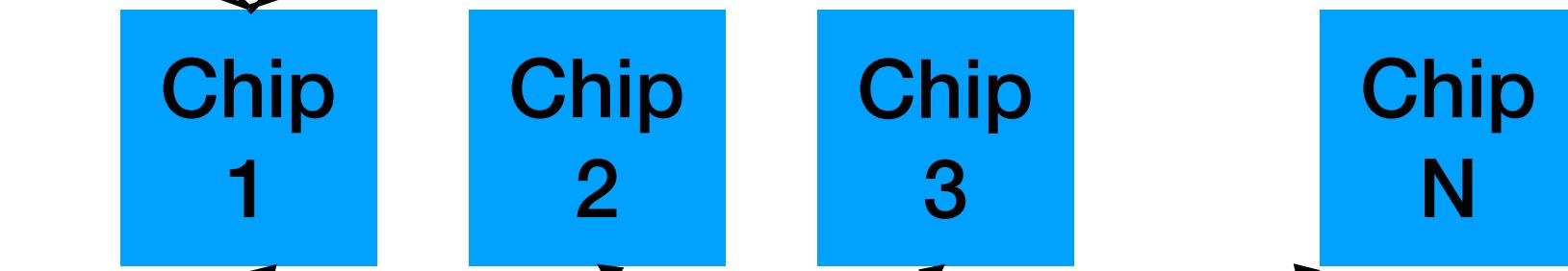




VersionPrint



ChipPrint



ChipMakerPrint



ModuleMakerPrint



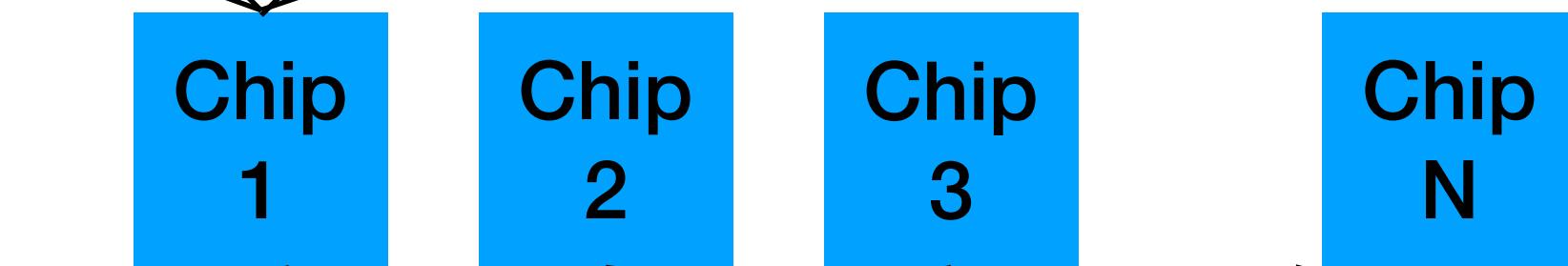
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



VersionPrint



ChipPrint



ChipMakerPrint



ModuleMakerPrint



ProductMakerPrint



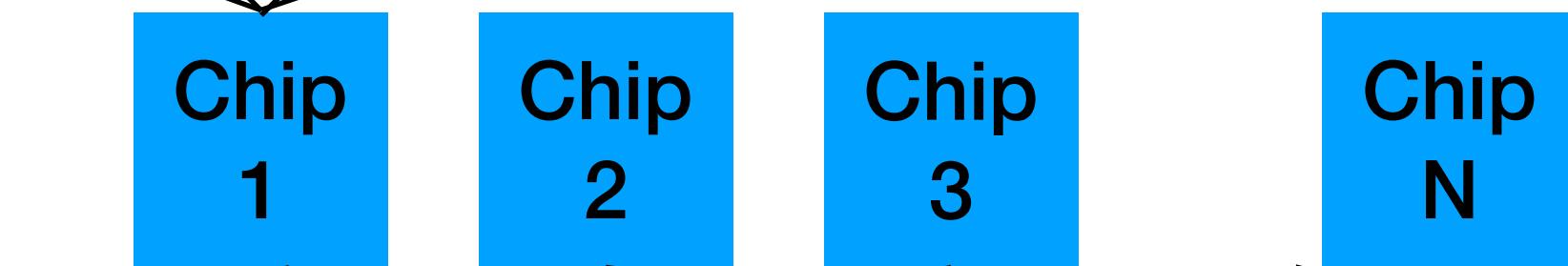
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



VersionPrint



ChipPrint



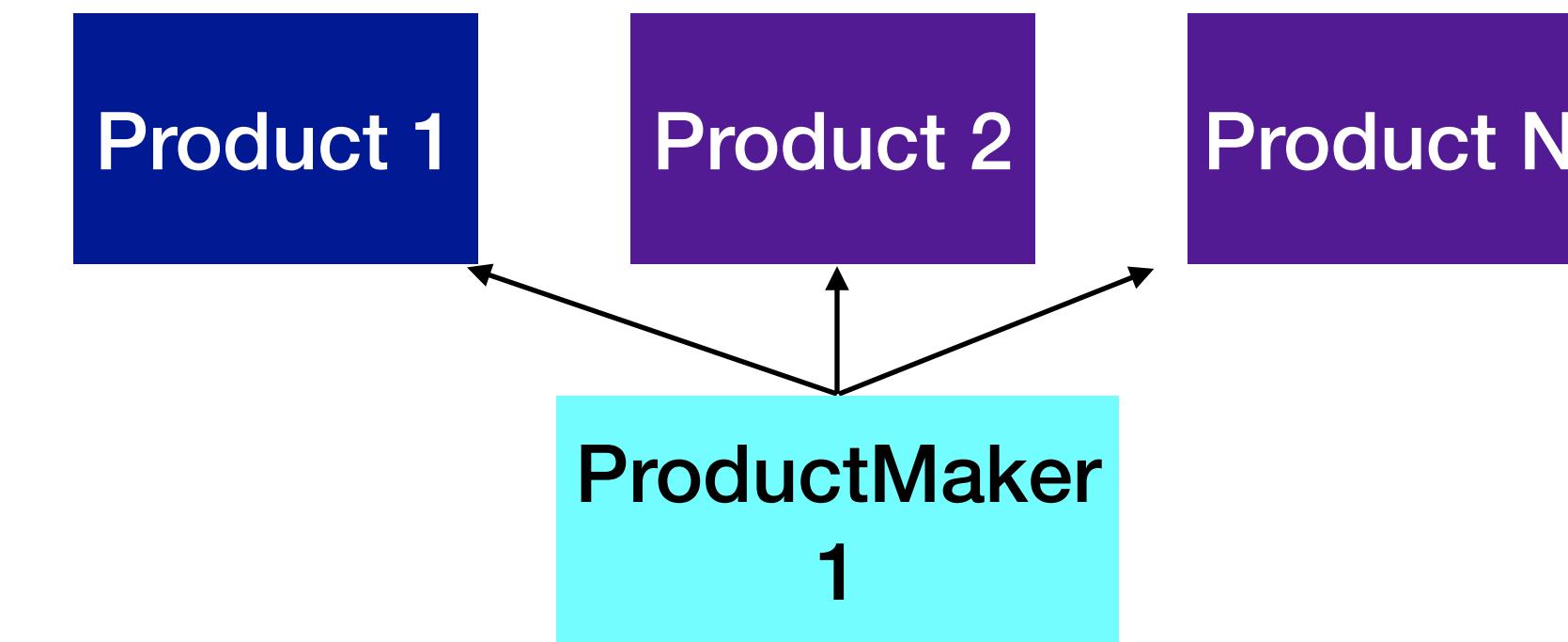
ChipMakerPrint



ModuleMakerPrint



ProductPrint



ProductMakerPrint



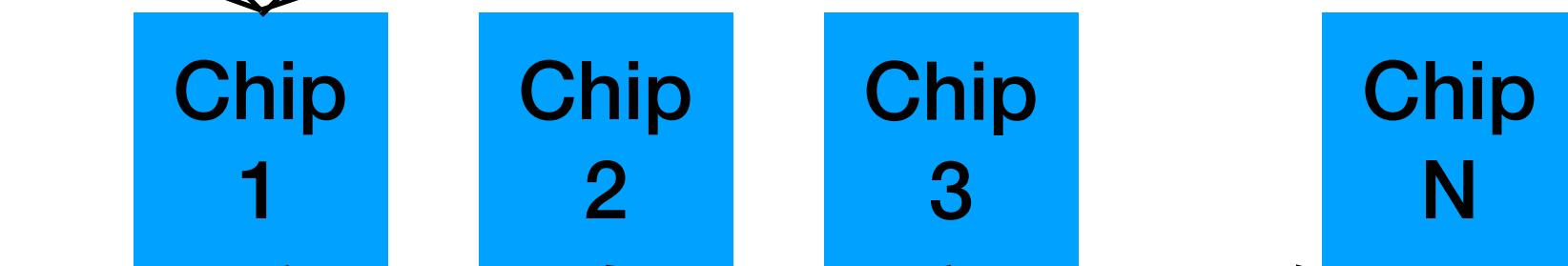
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



VersionPrint



ChipPrint



ChipMakerPrint



ModuleMakerPrint



ProductPrint



ProductMakerPrint



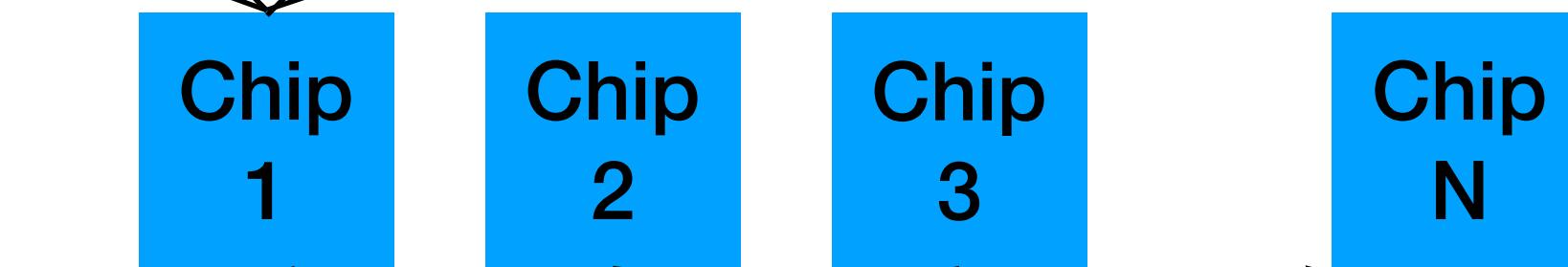
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



VersionPrint



ChipPrint



ChipMakerPrint



ModuleMakerPrint



ProductPrint



ProductMakerPrint



3330 Product Makers registered with Bluetooth SIG as of the time of writing!



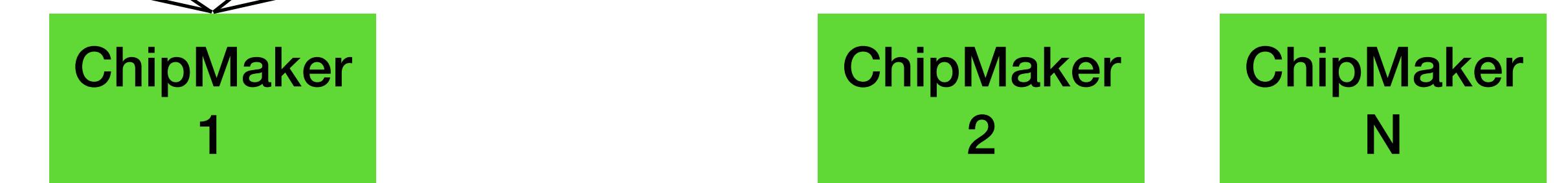
VersionPrint



ChipPrint



ChipMakerPrint



ModuleMakerPrint



ProductPrint



ProductMakerPrint



There is almost always a
1:1 relationship from
Products to Chips
We need to discover it

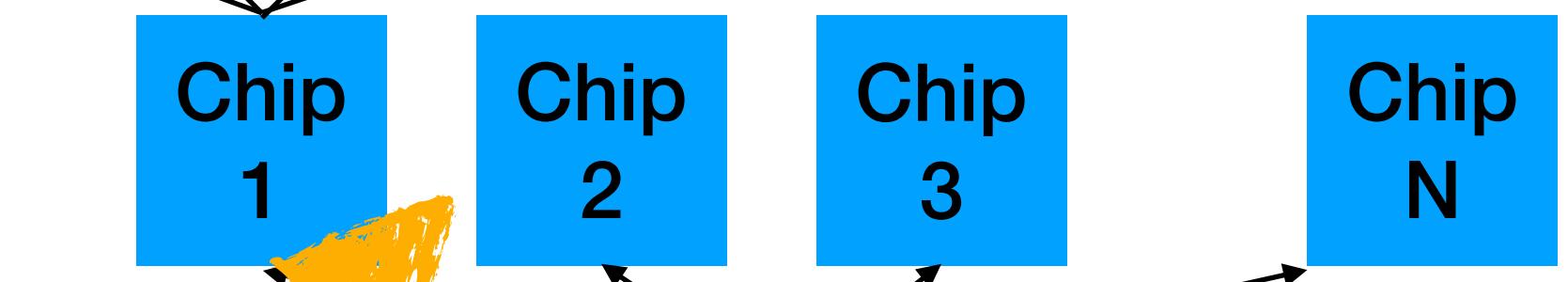
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



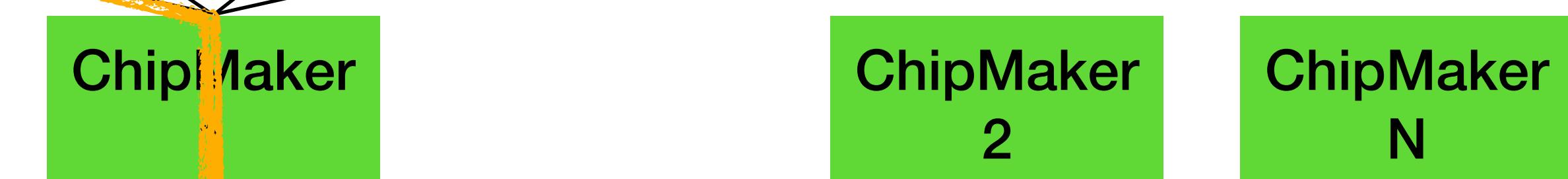
VersionPrint



ChipPrint



ChipMakerPrint



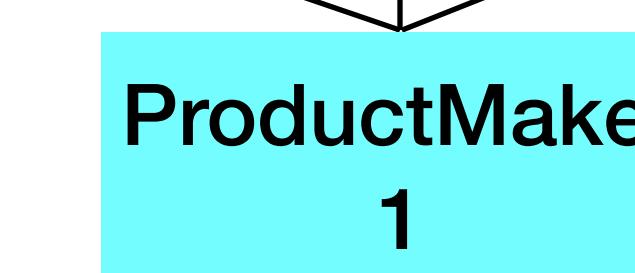
ModuleMakerPrint



ProductPrint



ProductMakerPrint



There is almost always a
1:1 relationship from
Products to Chips
We need to discover it

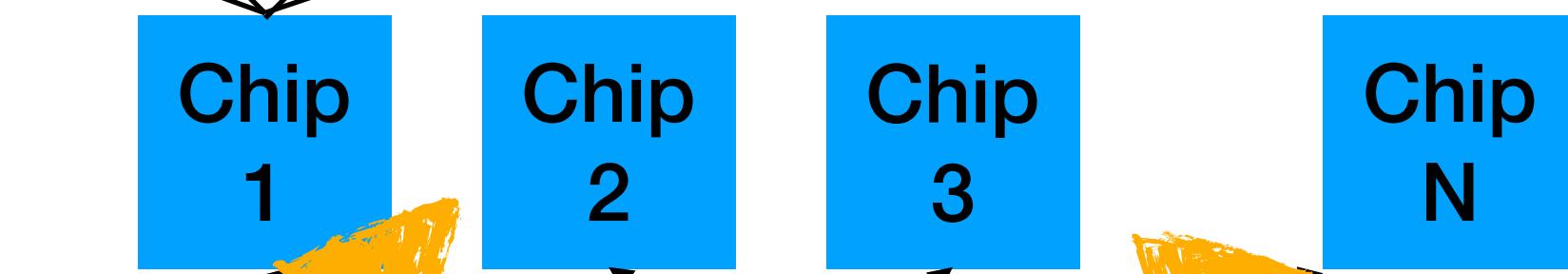
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



VersionPrint



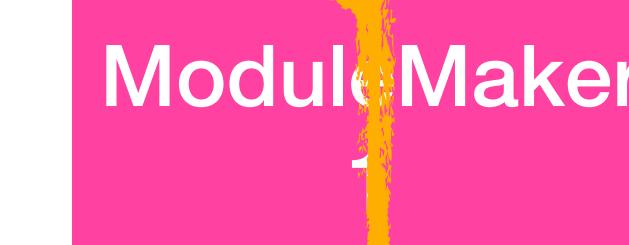
ChipPrint



ChipMakerPrint



ModuleMakerPrint



ProductPrint



ProductMakerPrint

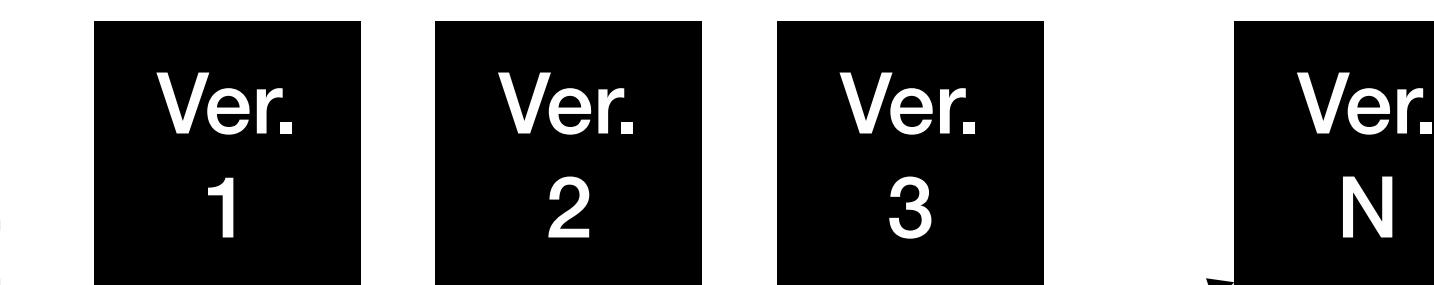


There is almost always a
1:1 relationship from
Products to Chips
We need to discover it

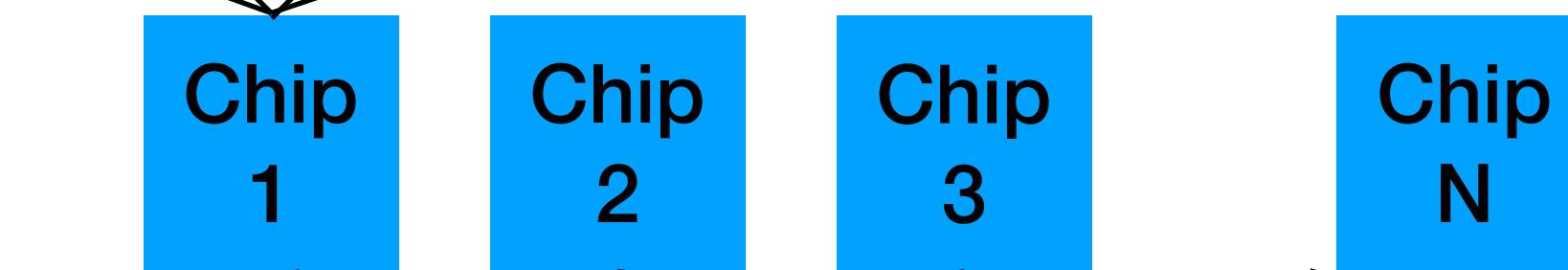
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



VersionPrint



ChipPrint



ChipMakerPrint



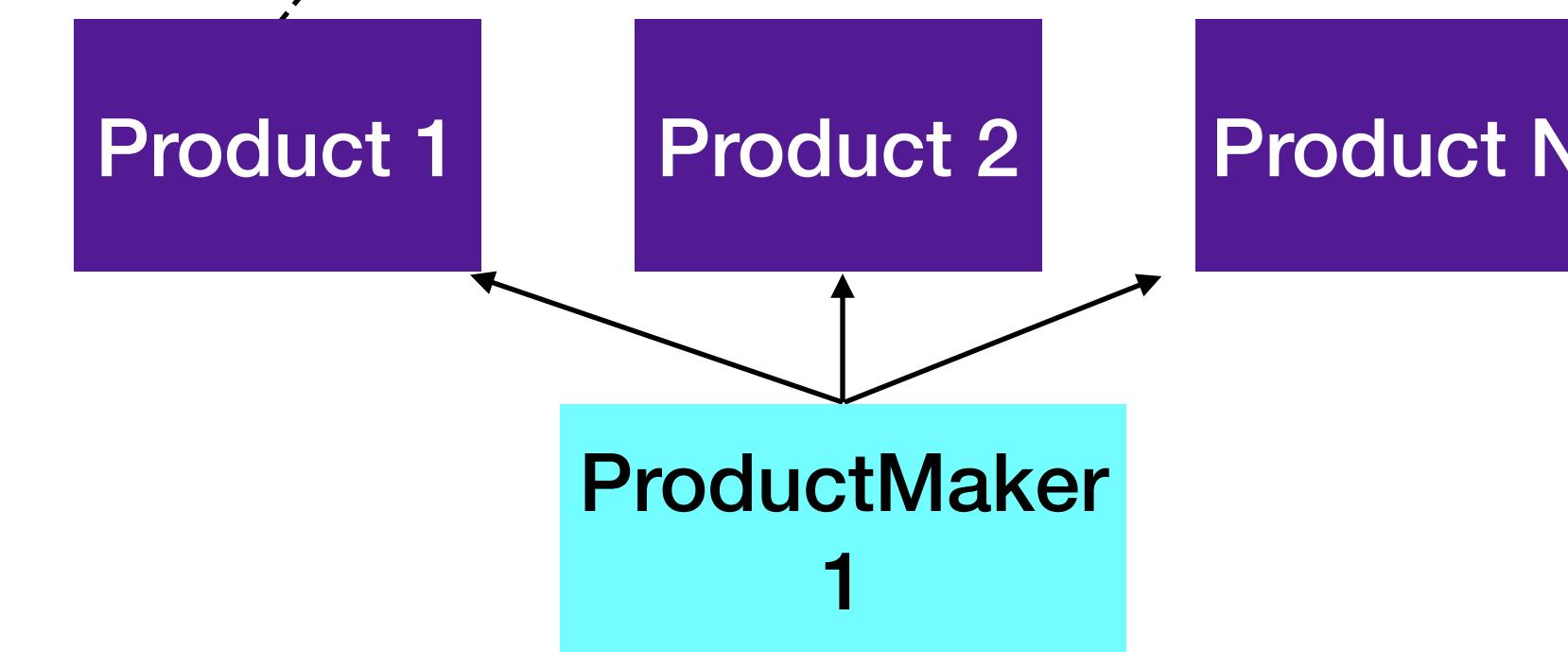
ModulePrint



ModuleMakerPrint



ProductPrint



ProductMakerPrint

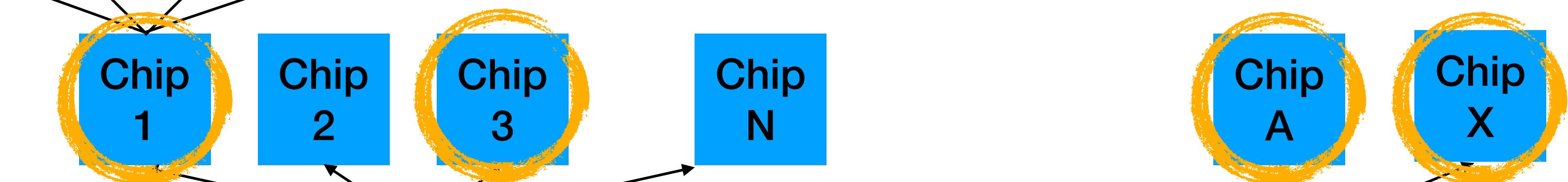
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



VersionPrint



ChipPrint



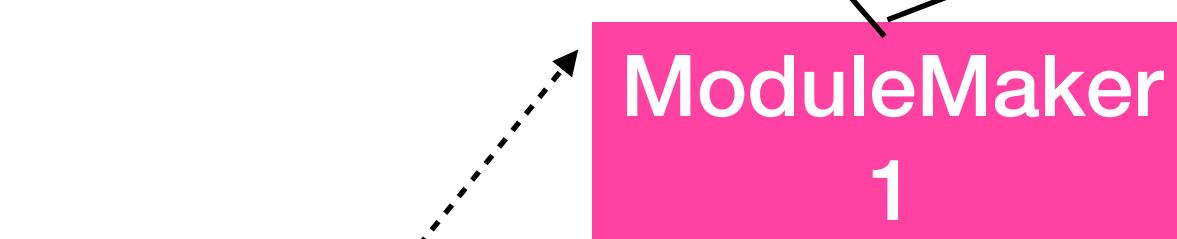
ChipMakerPrint



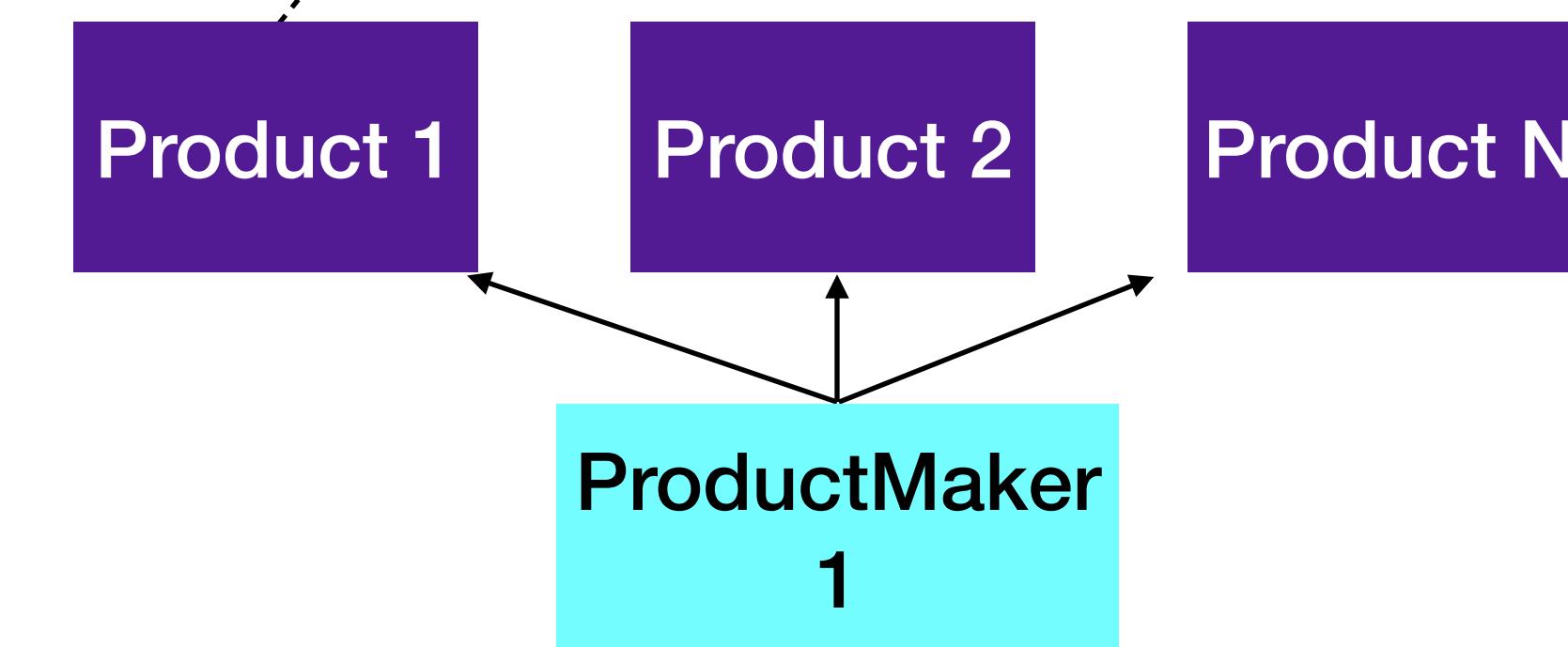
ModulePrint



ModuleMakerPrint



ProductPrint



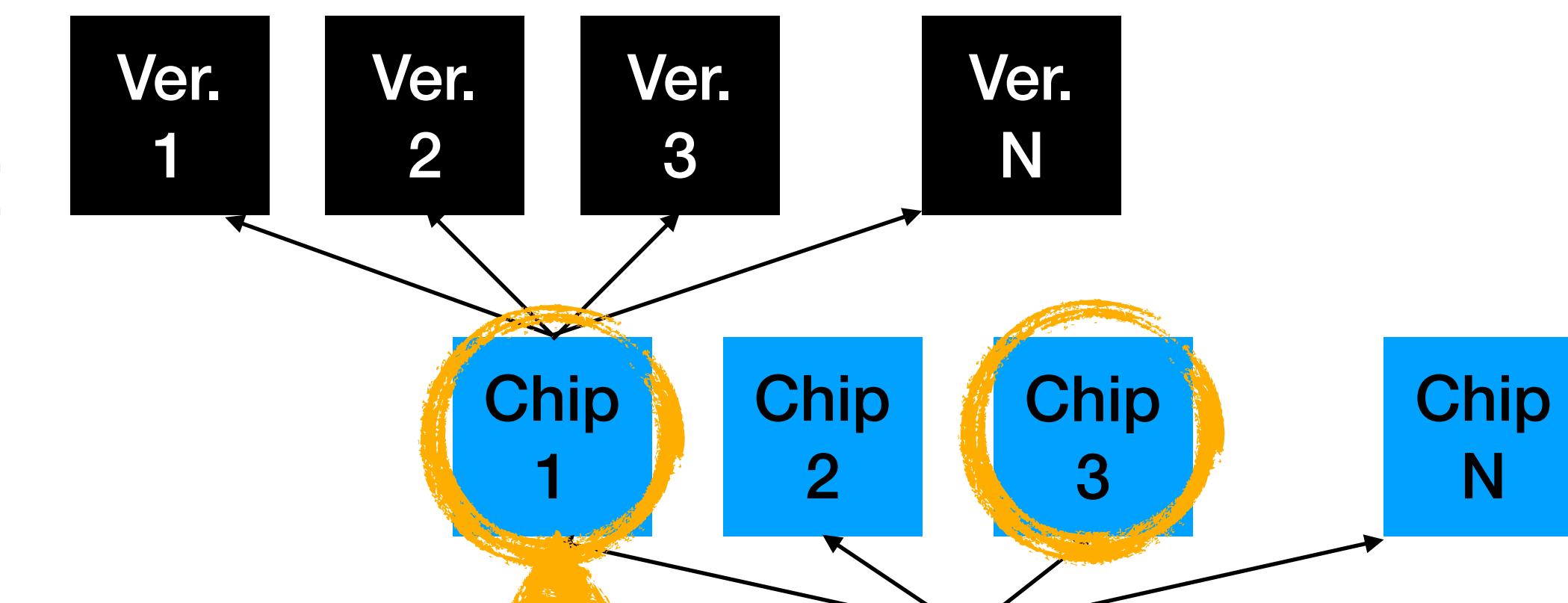
ProductMakerPrint

Module Makers only use certain chips
Mapping that, reduces the possible Chip space

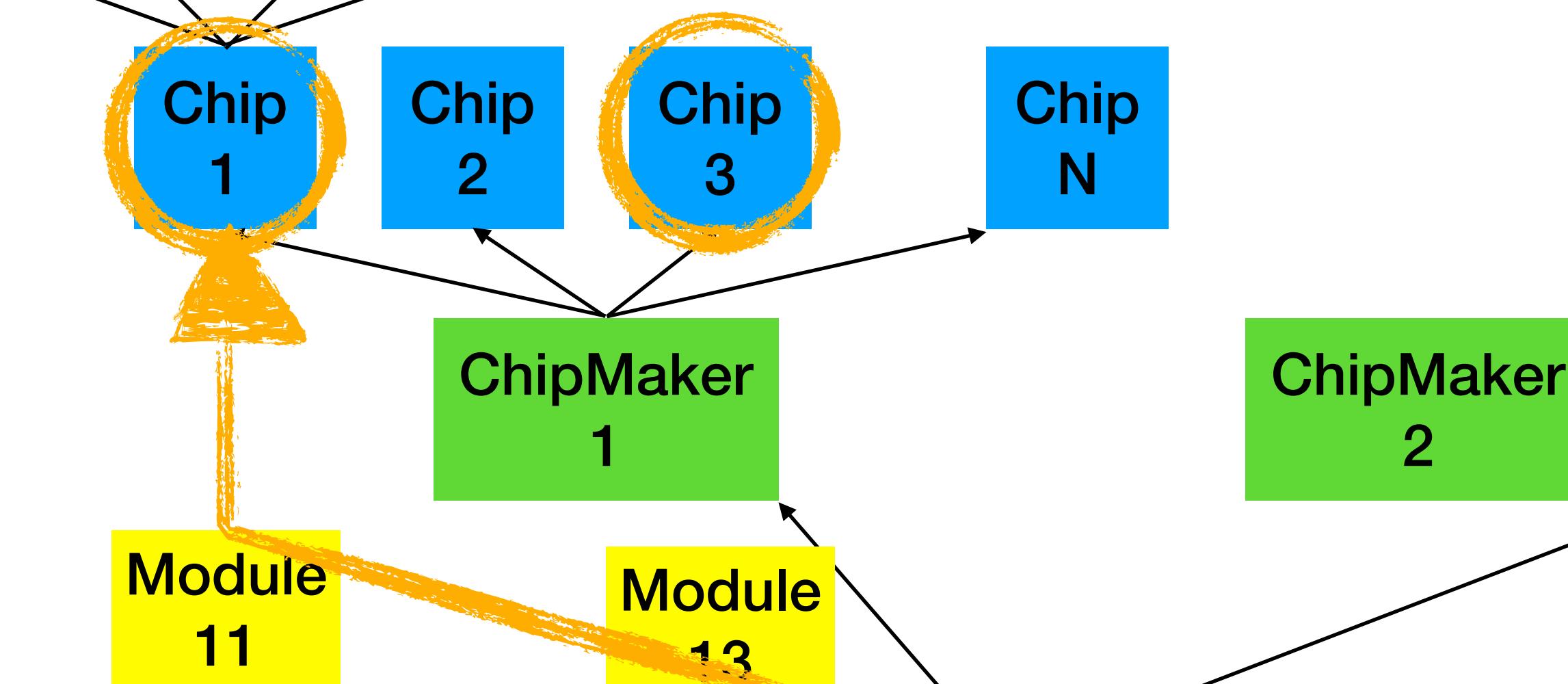
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



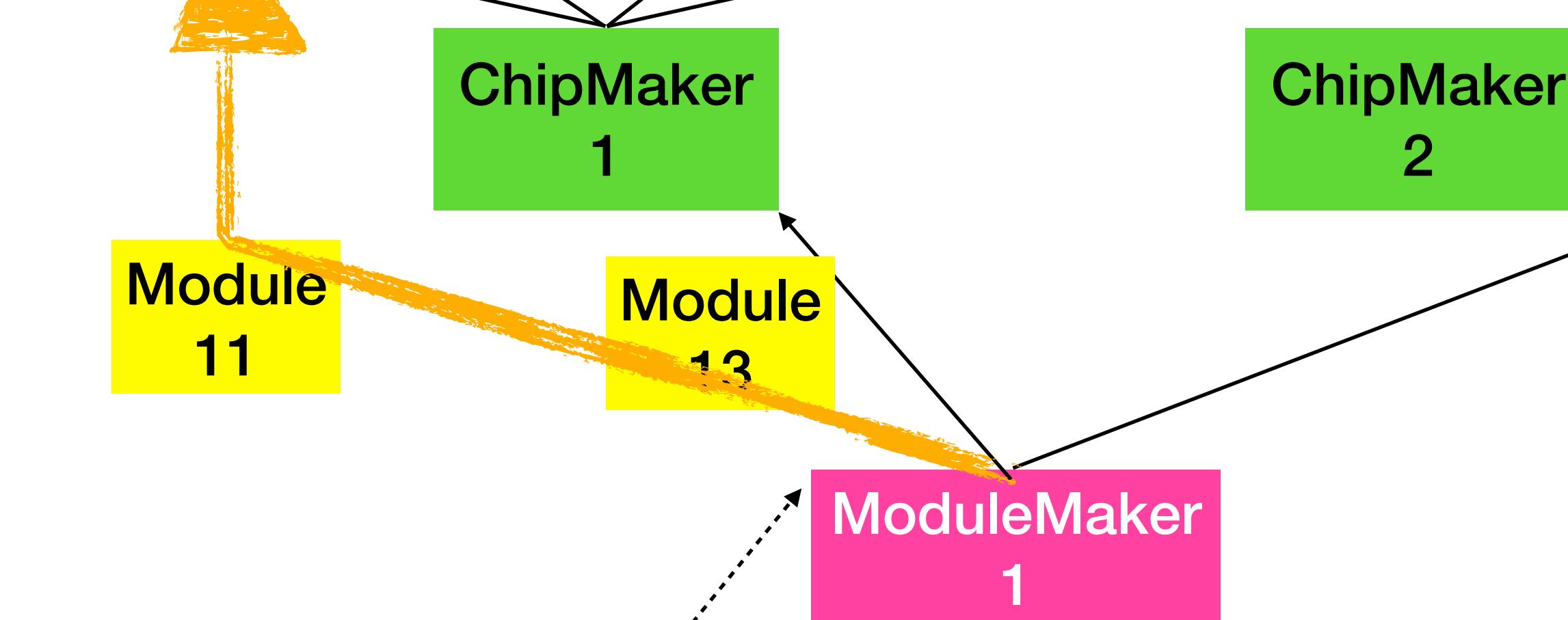
VersionPrint



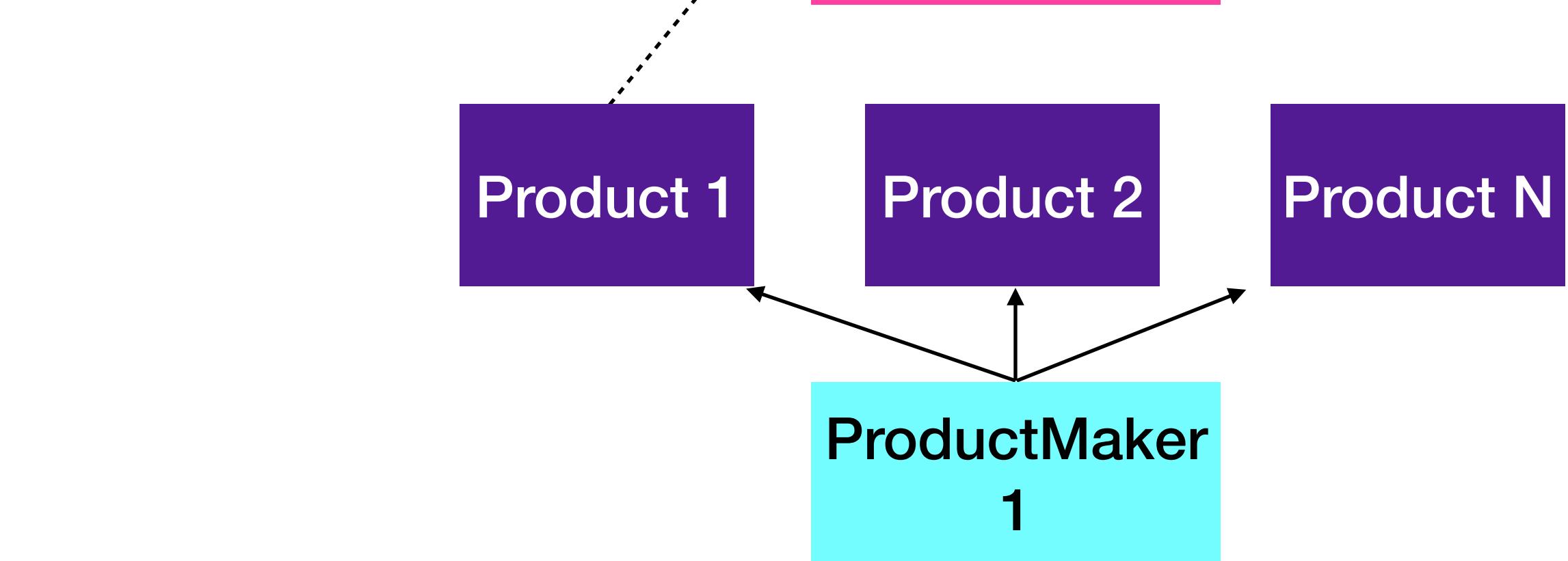
ChipPrint



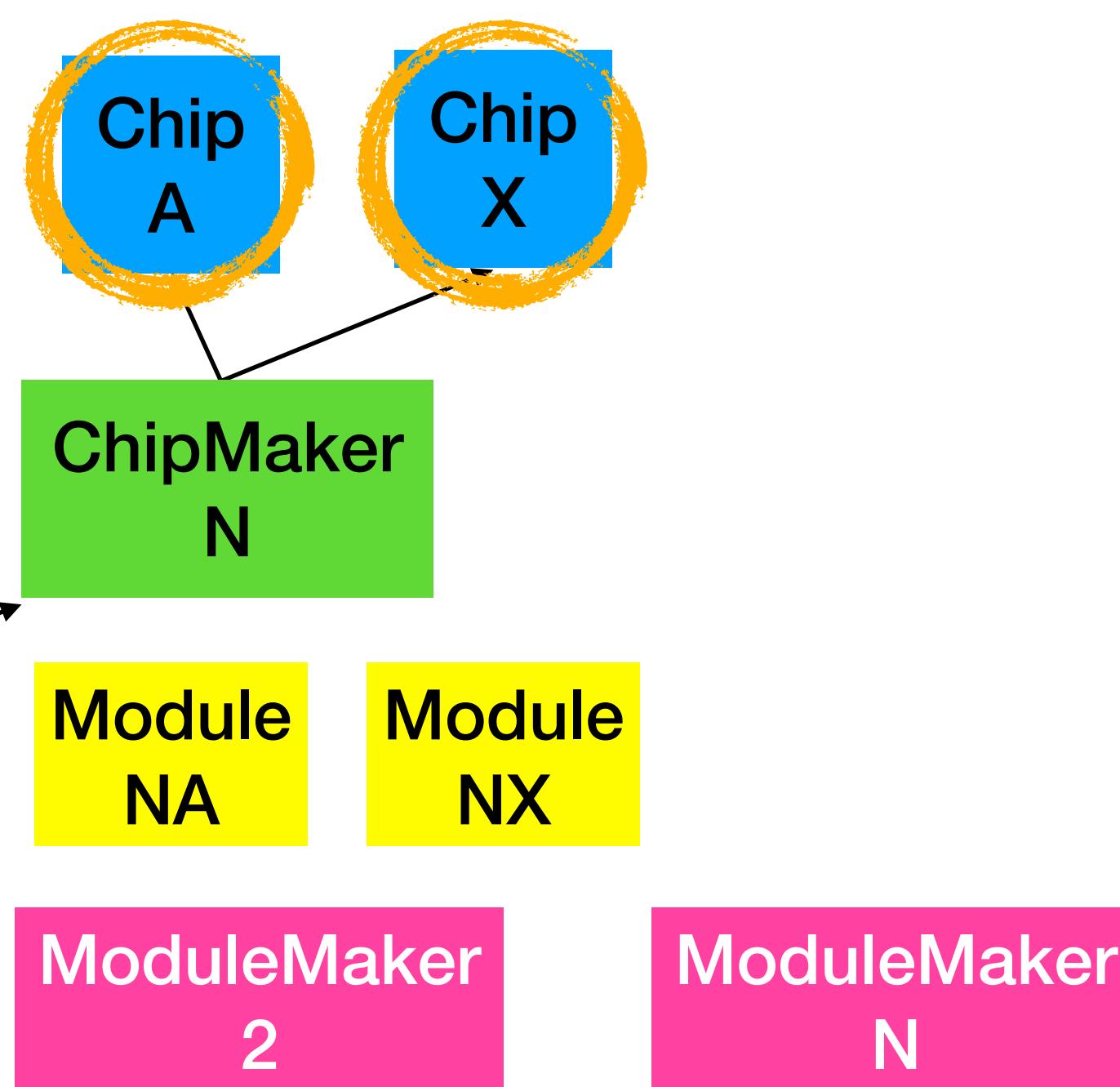
ChipMakerPrint



ModulePrint



ModuleMakerPrint



ProductPrint

Module Makers only use certain chips
Mapping that, reduces the possible Chip space

ProductMakerPrint

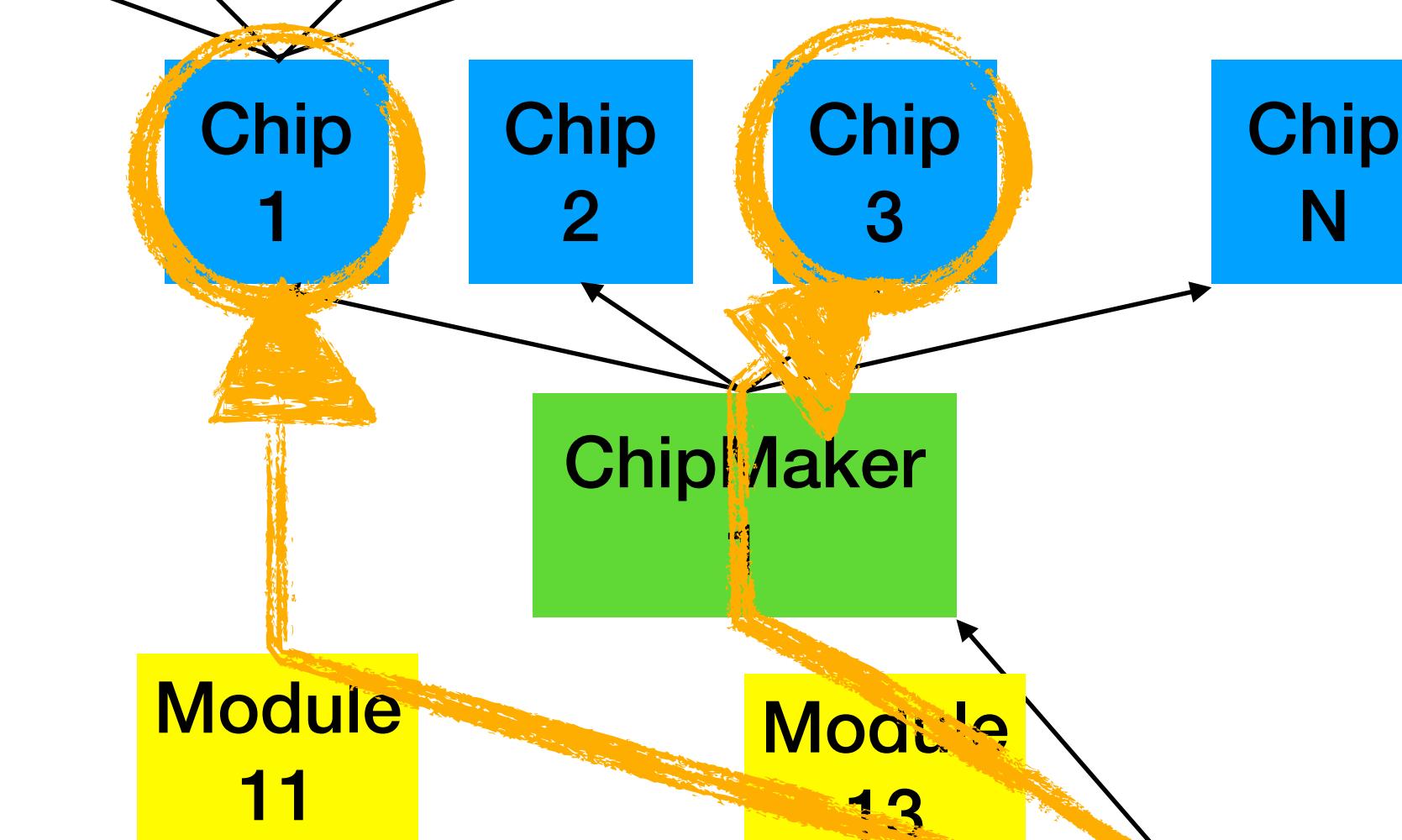
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



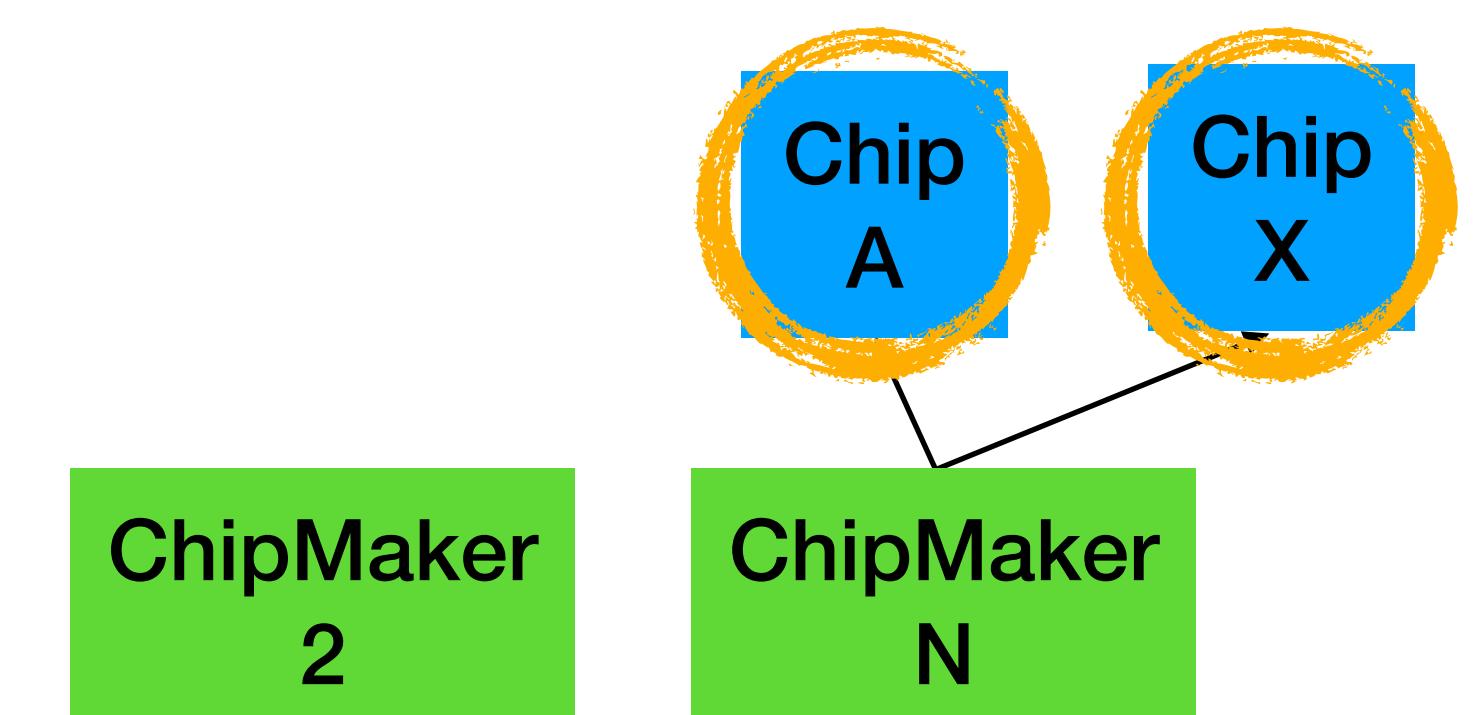
VersionPrint



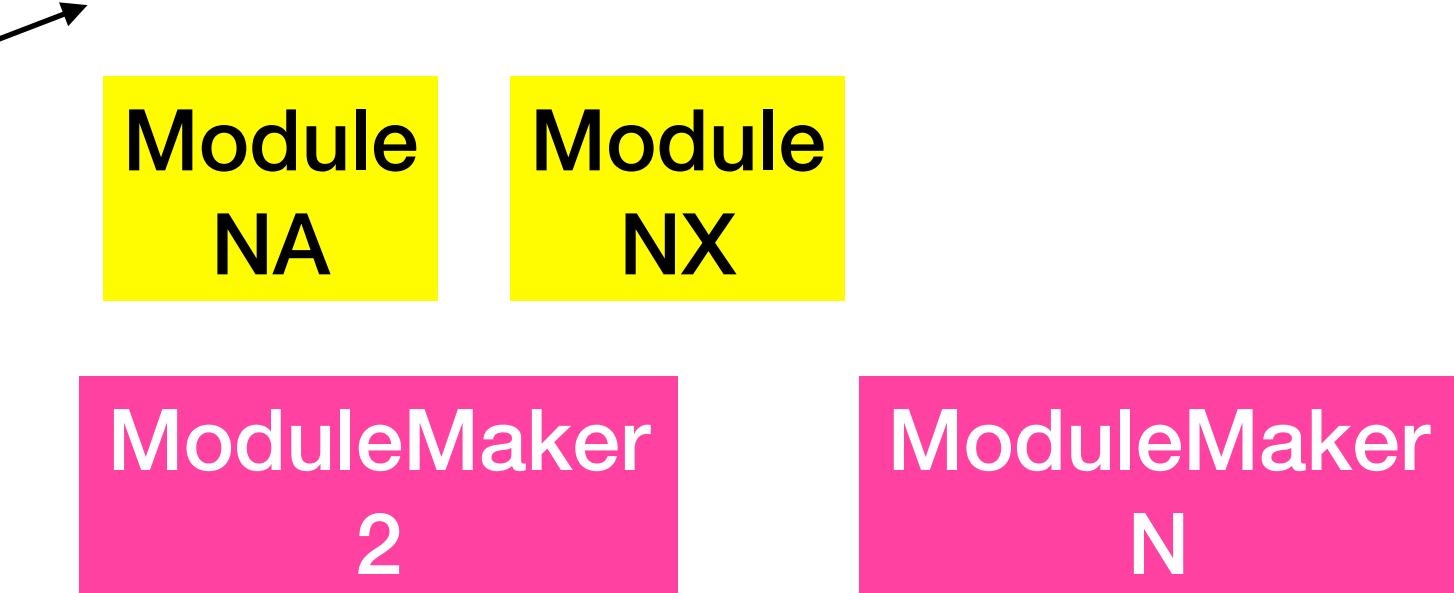
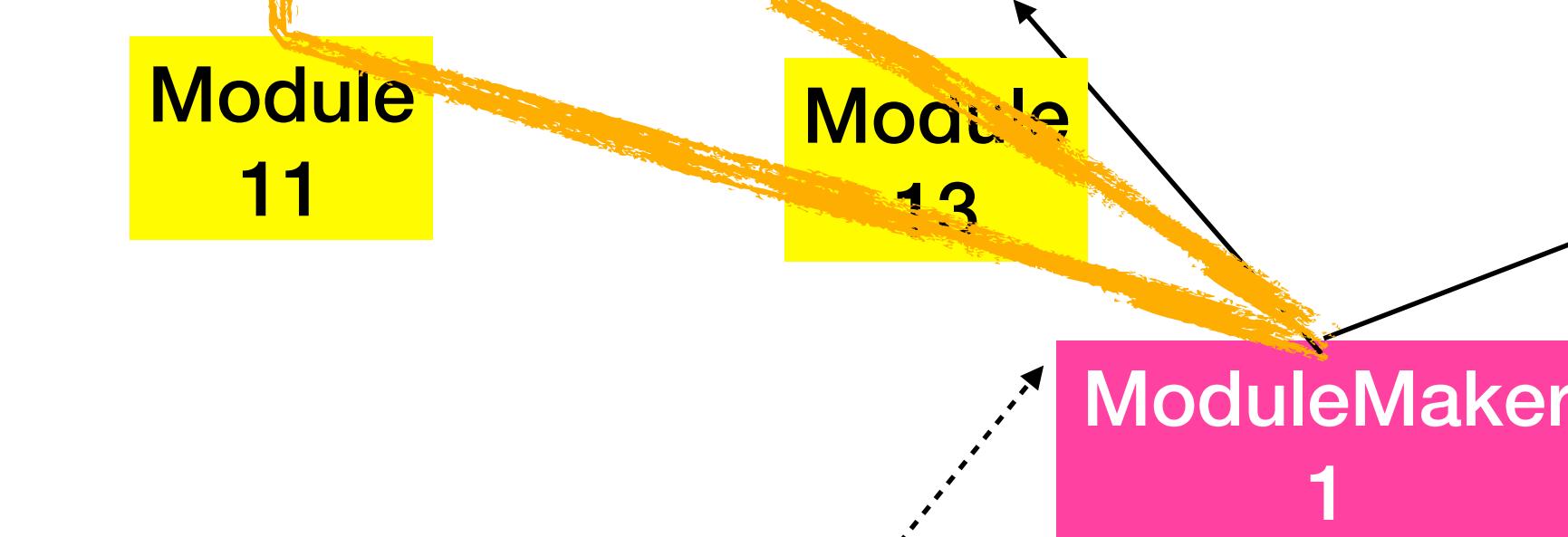
ChipPrint



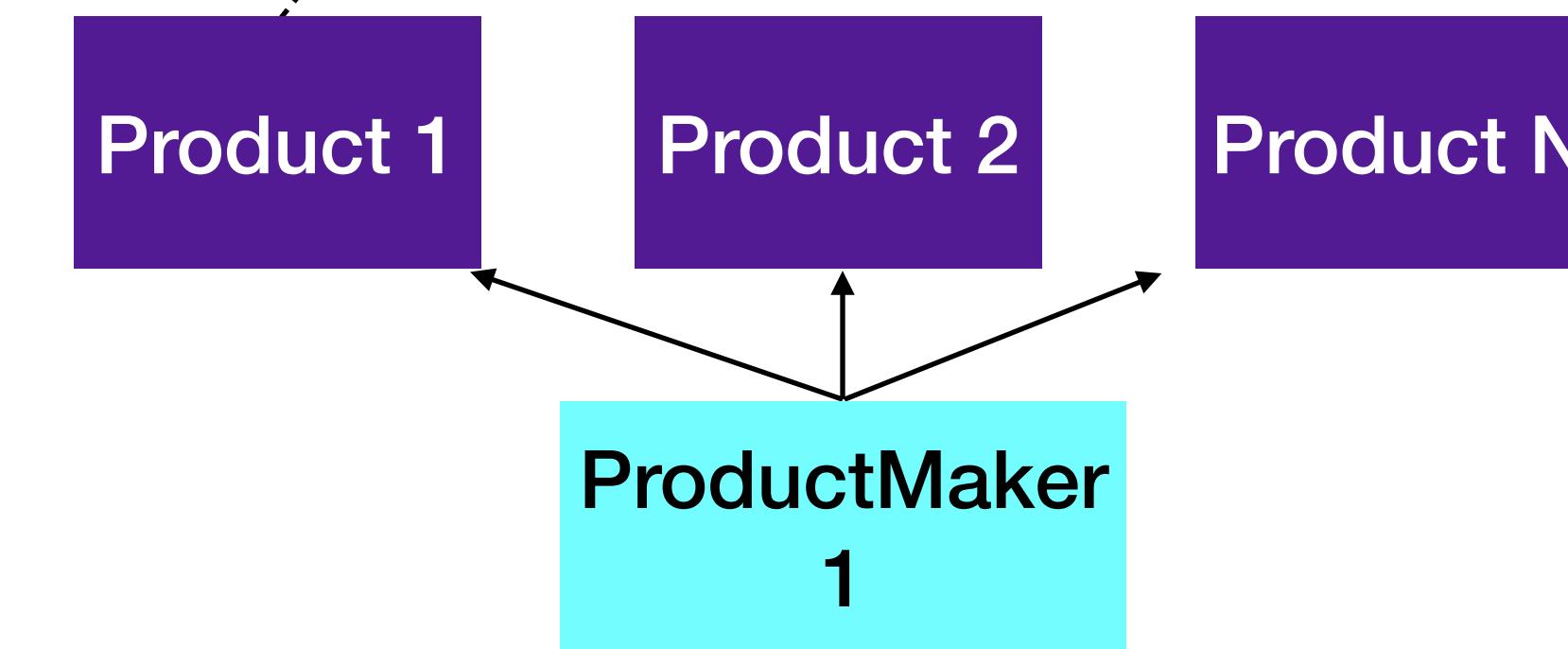
ChipMakerPrint



ModulePrint



ModuleMakerPrint



ProductPrint

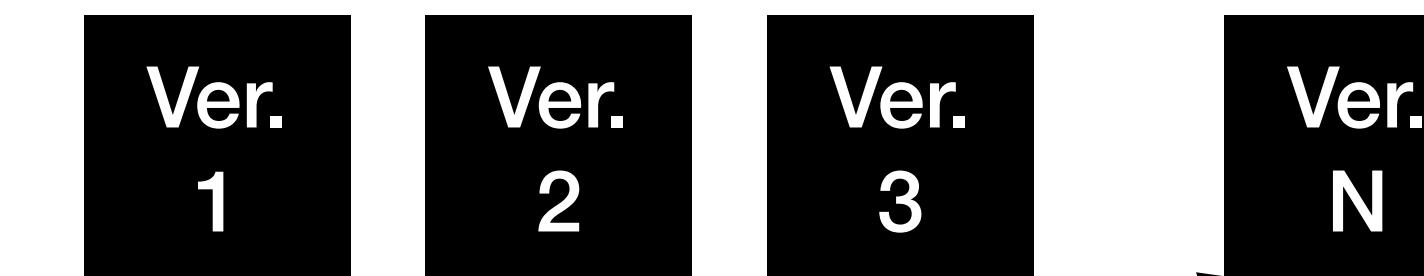
Module Makers only use certain chips
Mapping that, reduces the possible Chip space

ProductMakerPrint

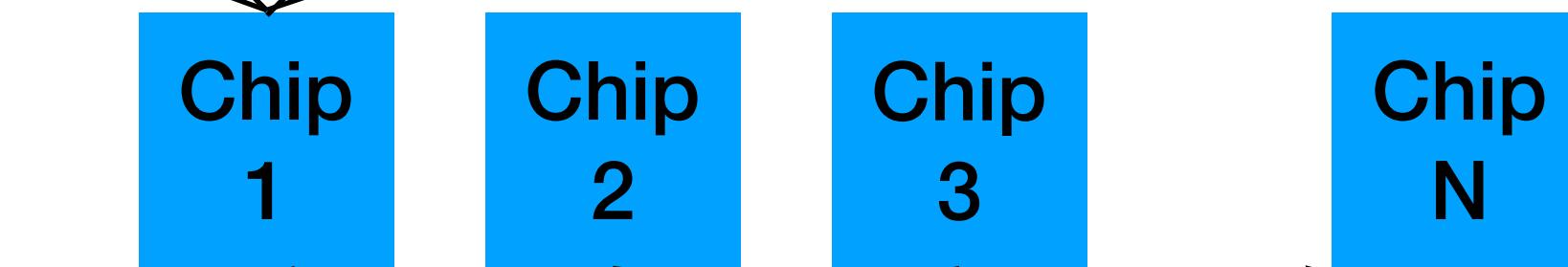
3330 Product Makers registered with Bluetooth SIG as of the time of writing!



VersionPrint



ChipPrint



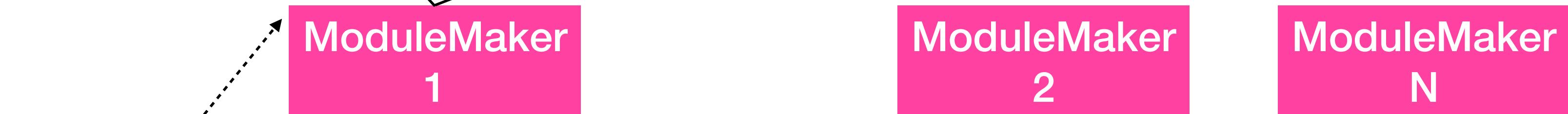
ChipMakerPrint



ModulePrint



ModuleMakerPrint



ProductPrint

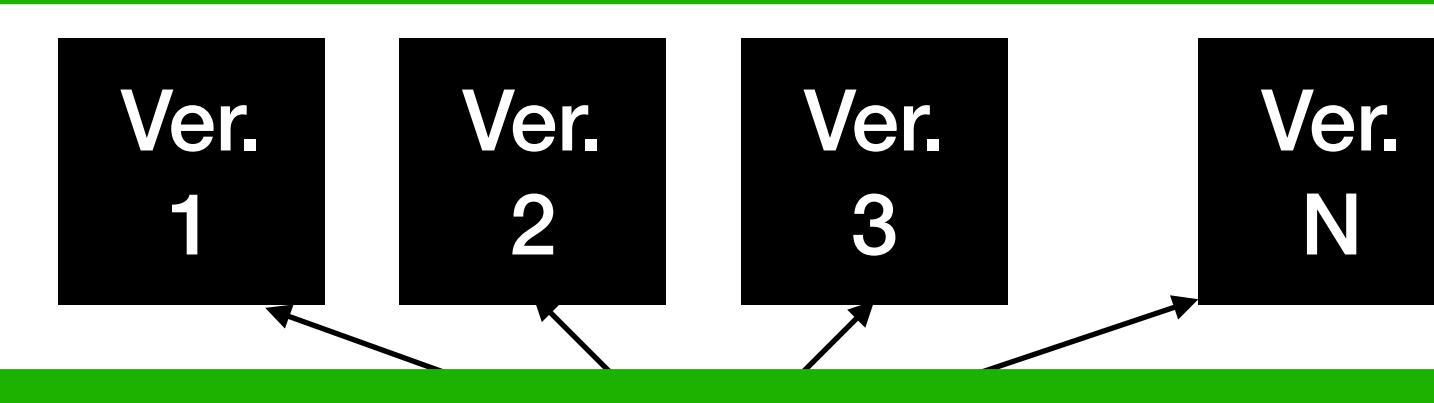


ProductMakerPrint

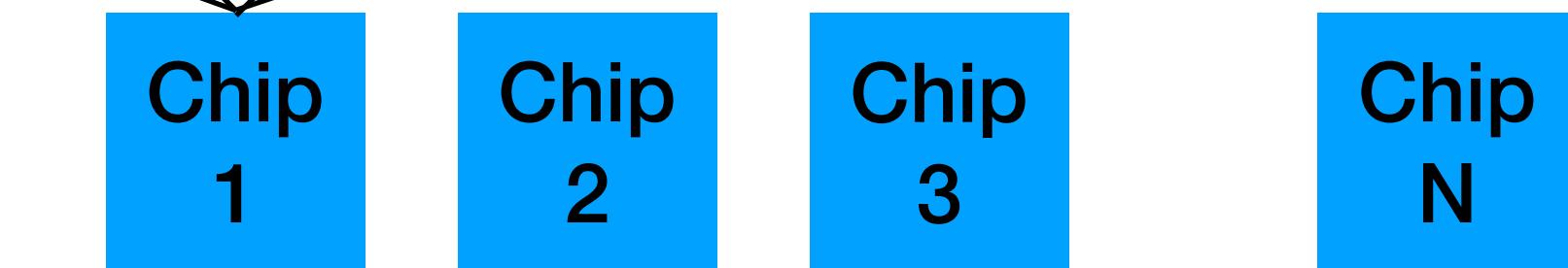




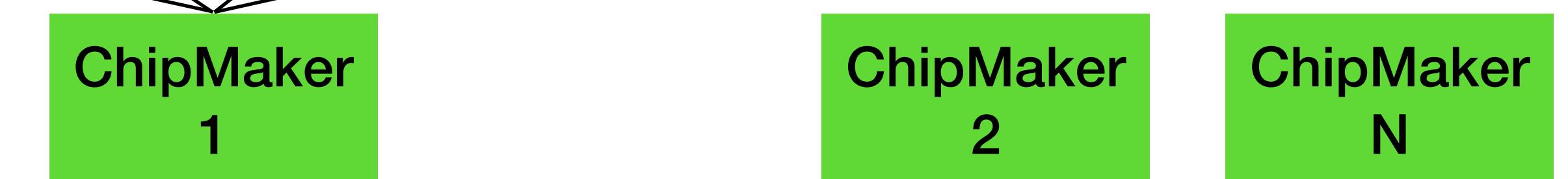
/ersionPrint



ChipPrint



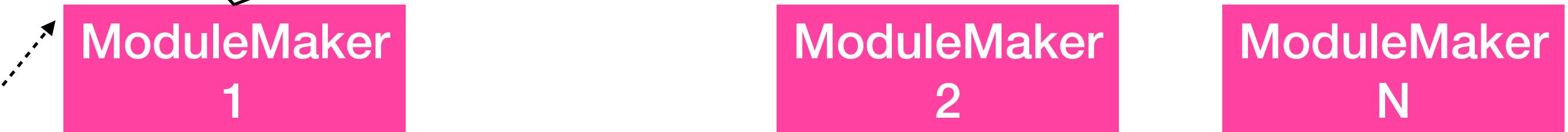
ChipMakerPrint



ModulePrint



ModuleMakerPrint



ProductPrint

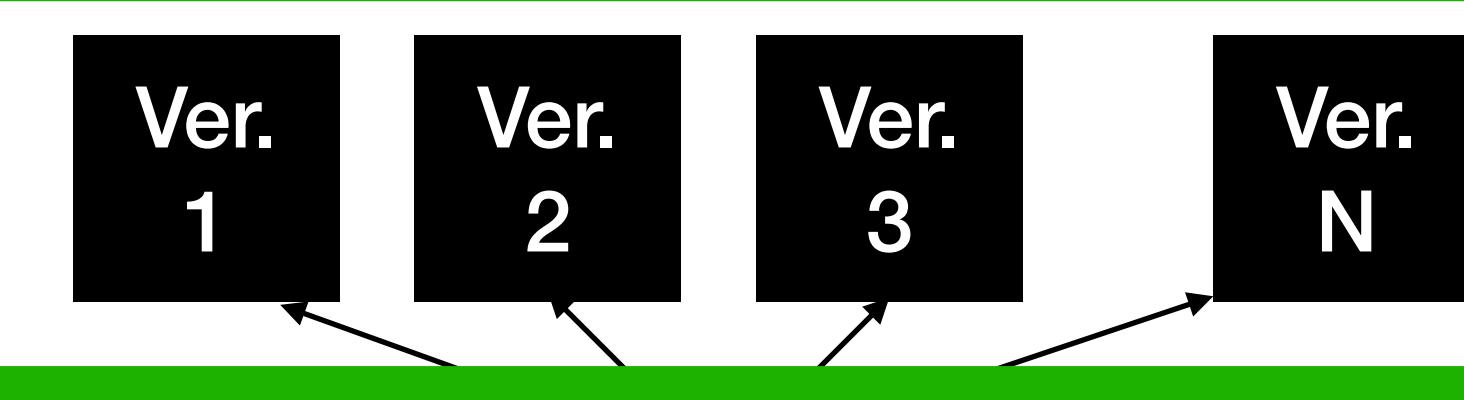


ProductMakerPrint



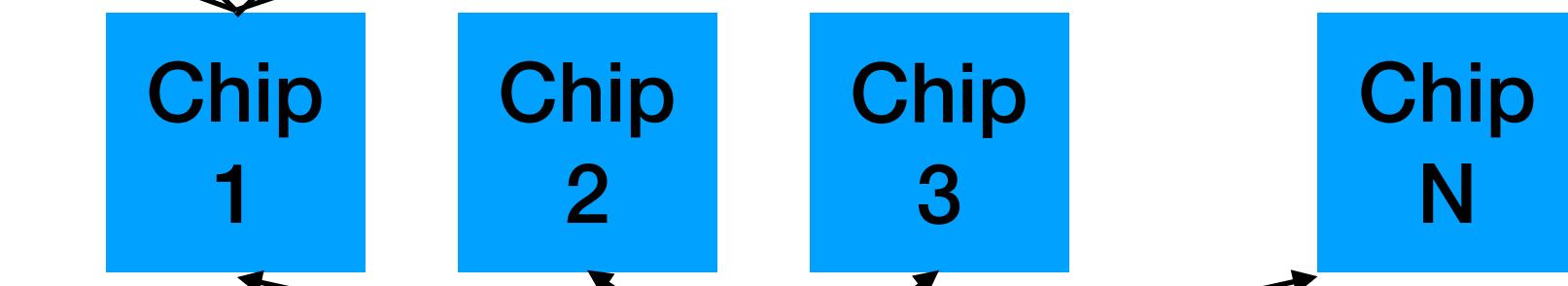


/ersiOnPrint



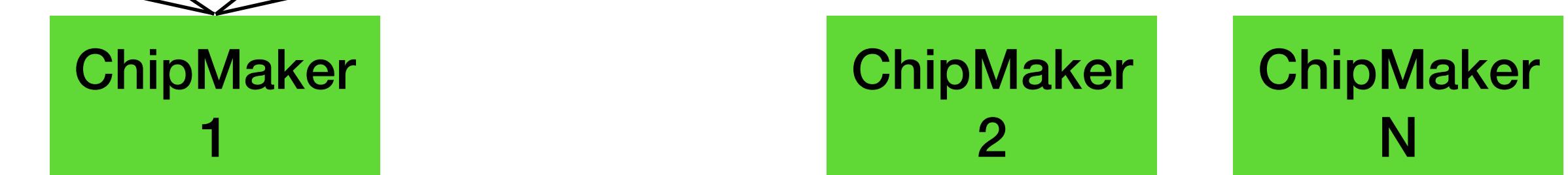
WHAT I WANT!

ChipPrint



WHAT I MOSTLY GET 😔

ChipMakerPrint



ModulePrint



ModuleMakerPrint



ProductPrint



ProductMakerPrint

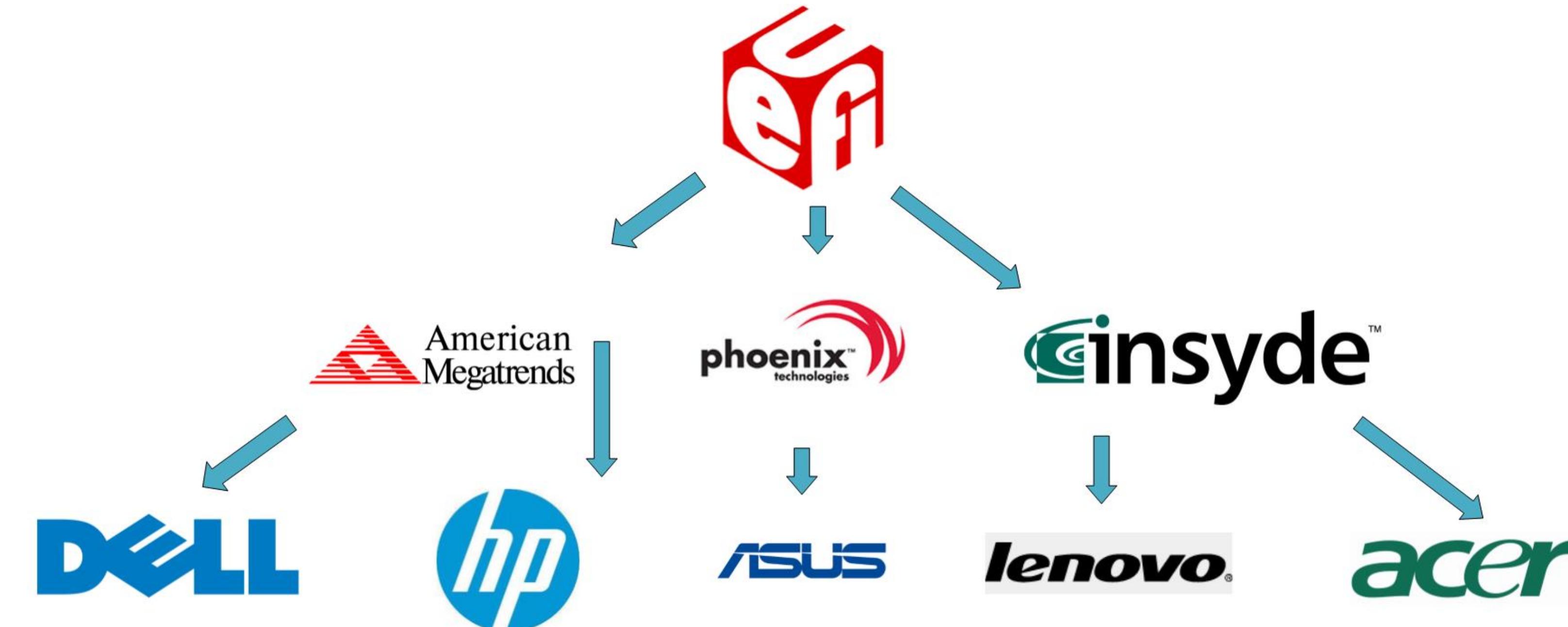




BTW

Did I mention the
Bluetooth ecosystem is
ARCHITECTURALLY
UNSECURABLE

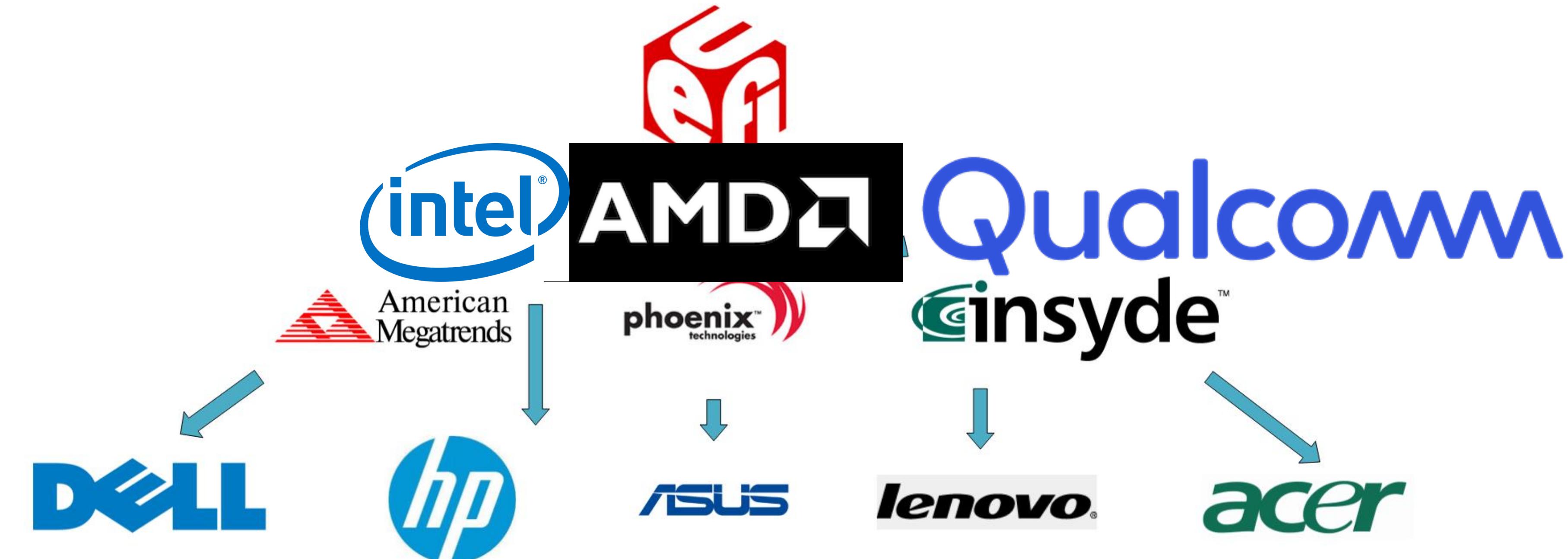
UEFI Vulnerability Proliferation



- If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.
 - More on how this theory works "in practice" later...



UEFI Vulnerability Proliferation

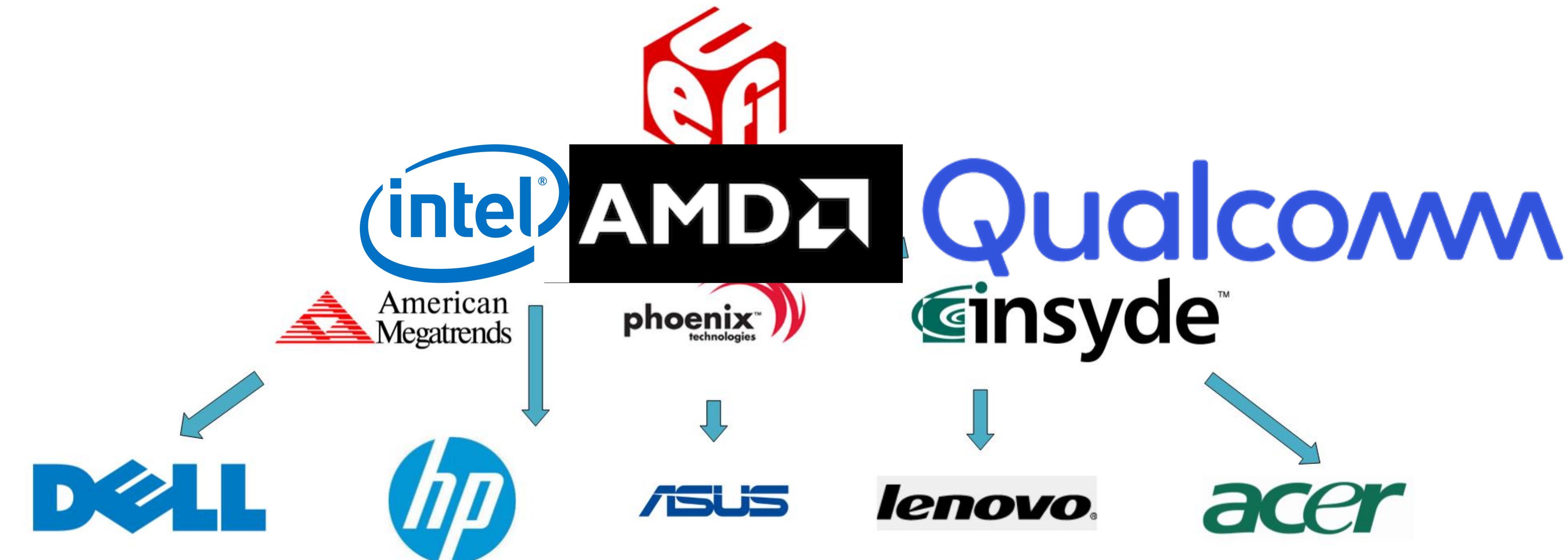


- If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.
 - More on how this theory works "in practice" later...



UEFI Vulnerability Proliferation

SIG

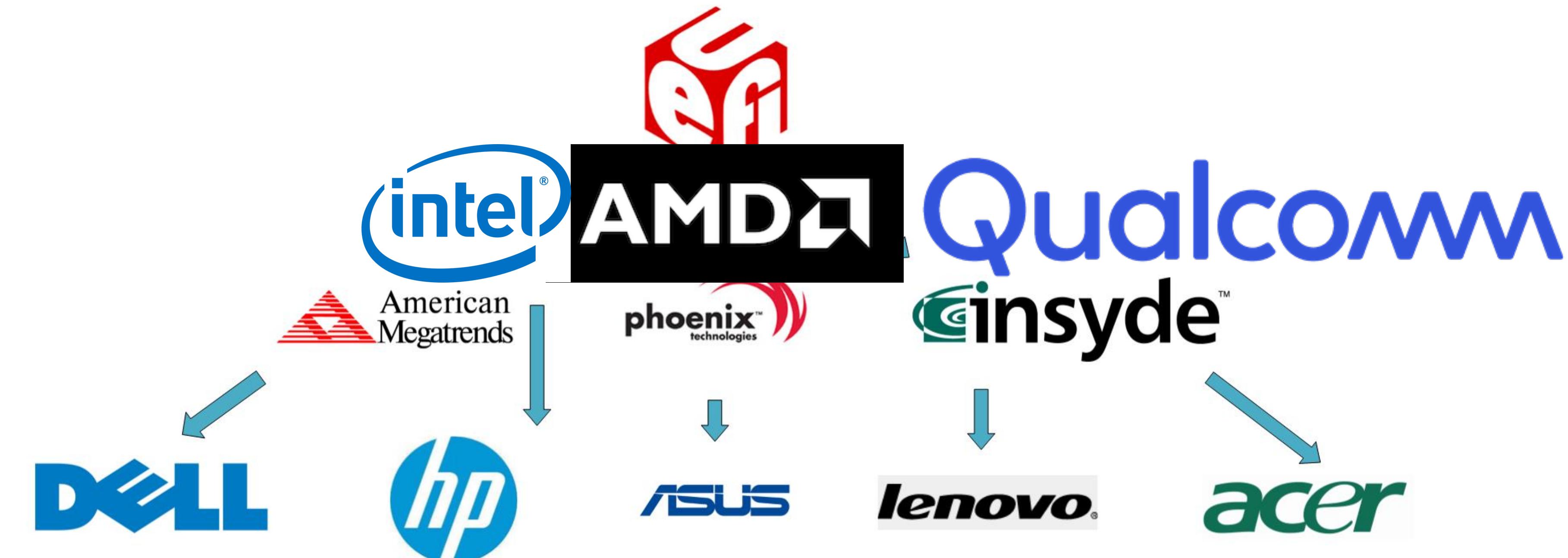


- If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.
 - More on how this theory works "in practice" later...



UEFI Vulnerability Proliferation

SIG
Silicon

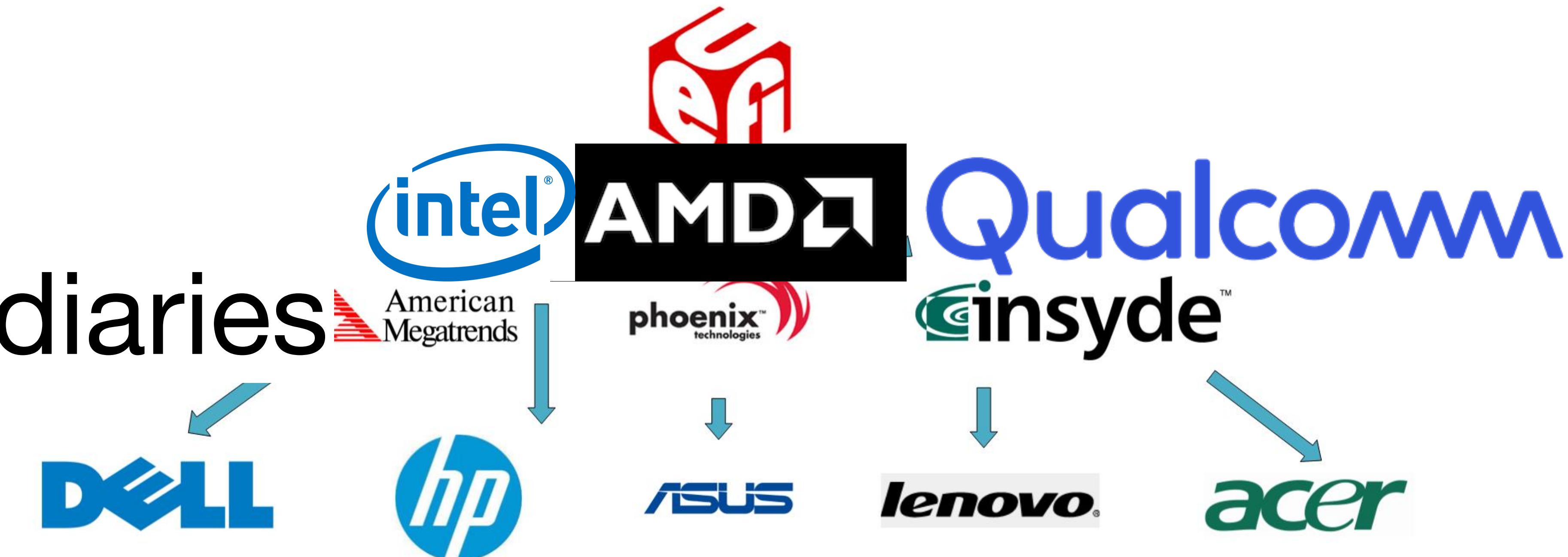


- If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.
 - More on how this theory works "in practice" later...



UEFI Vulnerability Proliferation

SIG
Silicon
Intermediaries

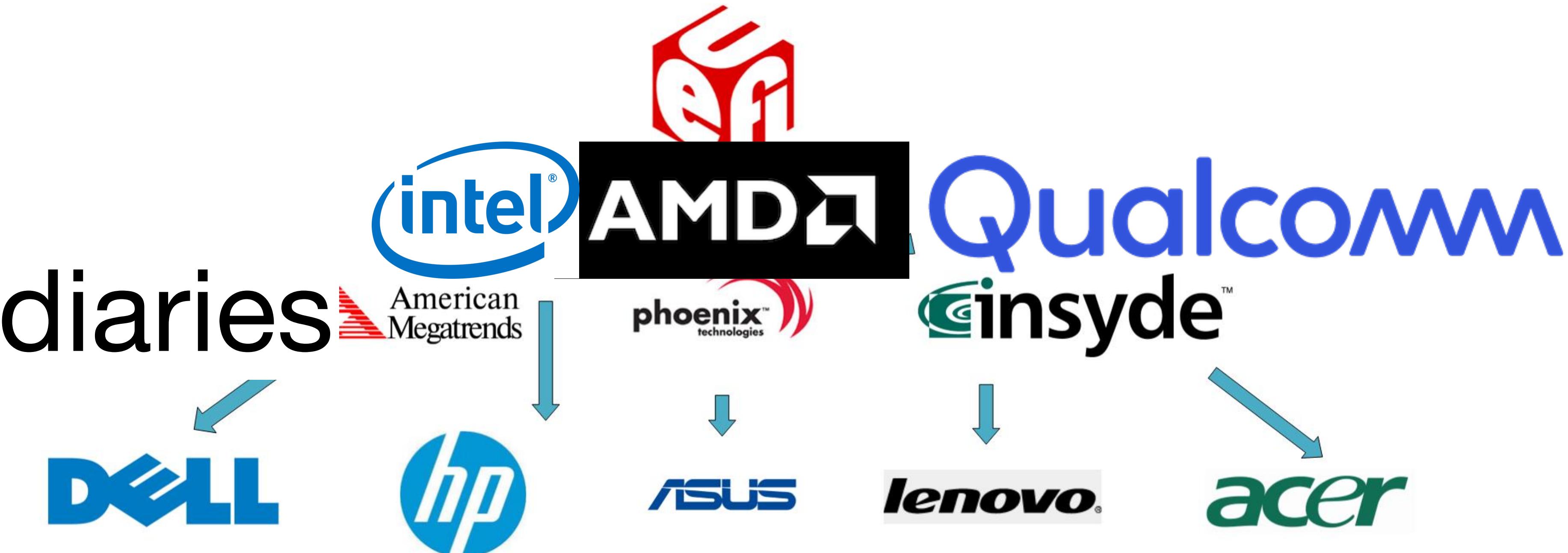


- If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.
 - More on how this theory works "in practice" later...



UEFI Vulnerability Proliferation

SIG
Silicon
Intermediaries
Device
Makers



- If an attacker finds a vulnerability in the UEFI "reference implementation," its proliferation across IBVs and OEMs would potentially be wide spread.
 - More on how this theory works "in practice" later...



 Bluetooth®

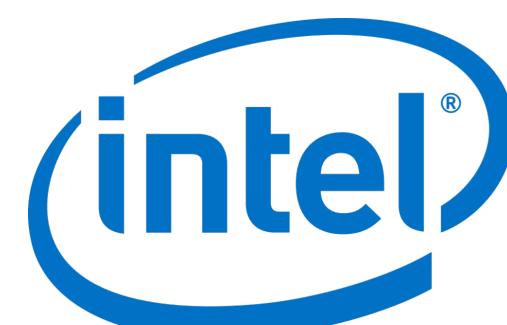
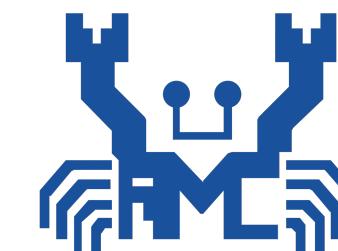
DST2
.FYI



SIG



Silicon Vendors (>20)



A COMPANY OF THE SWATCH GROUP



Telink



SIG



Silicon Vendors (>20)



OST2
.FV1

SIG



Silicon Vendors (>20)



OST2
.FYI

SIG



Silicon Vendors (>20)



Intermediaries



OST2
.FYI

SIG



Silicon Vendors (>20)



Intermediaries



Product-Makers



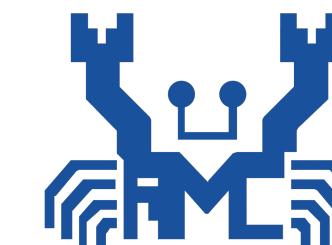
3330 registered with Bluetooth SIG as of the time of writing!

OST2
.FYI

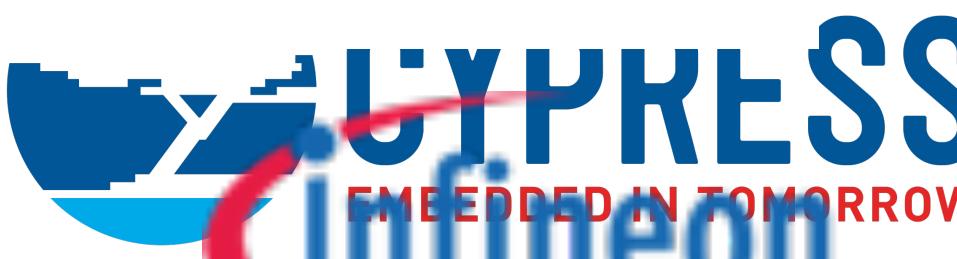
SIG



Silicon Vendors (>20)



Silicon

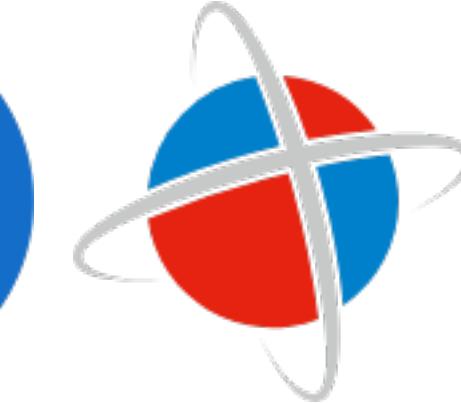


Telink



Module-Makers (IDEK how many)

Intermediaries
Device



晶讯
JINGXUN



Product-Makers

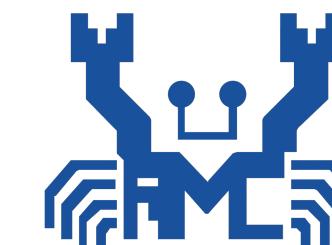
Makers 3330 registered with Bluetooth SIG as of the time of writing!

OST2
.FYI

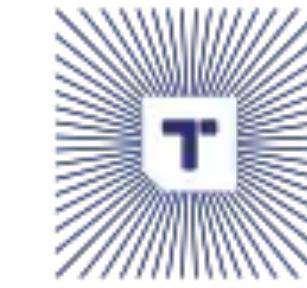
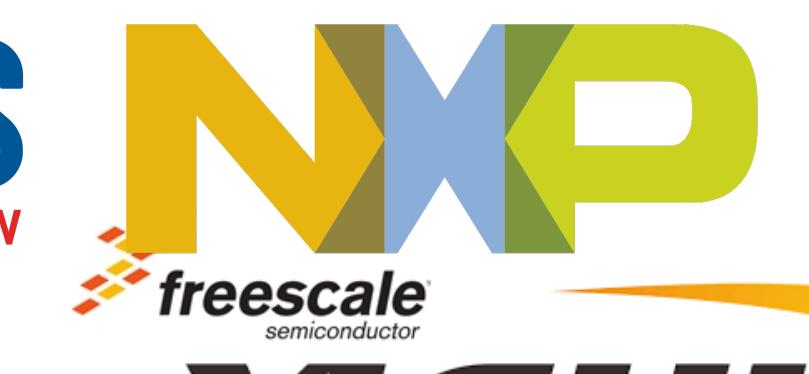
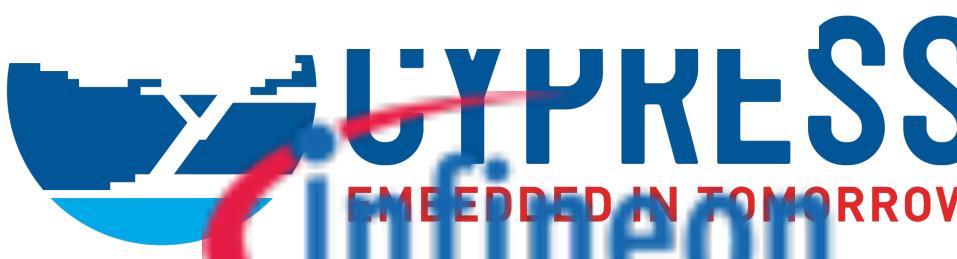
SIG



Silicon Vendors (>20)



Silicon



Telink

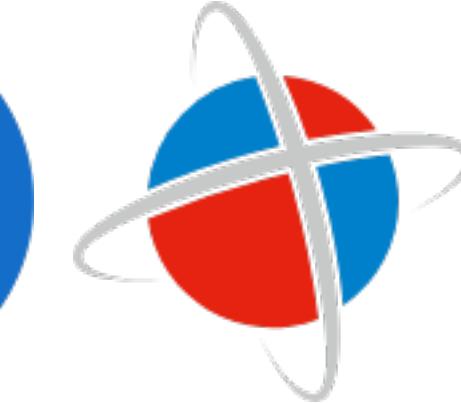


A COMPANY OF THE SWATCH GROUP



Module-Makers (IDEK how many)

Intermediaries
Device



晶讯
JINGXUN



Product-Makers

Makers 3330 registered with Bluetooth SIG as of the time of writing!

OST2
.FBI

SIG

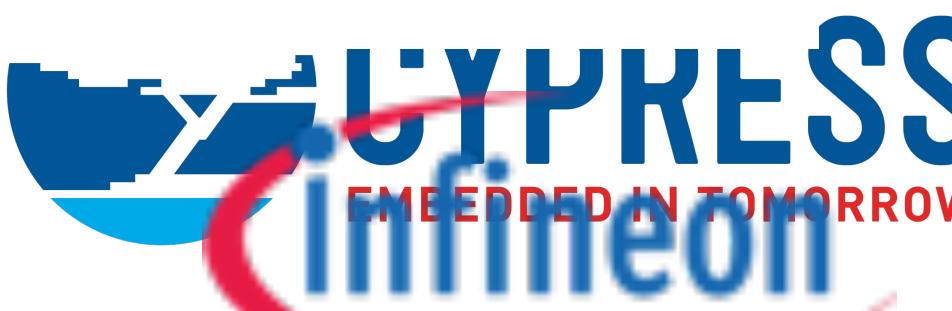
OST2
.FYI



Qualco



Silicon



Telink

RENESAS

MICROCHIP



oppo

MARVELL™

ISSC

BES TECHNIC

BEKEN

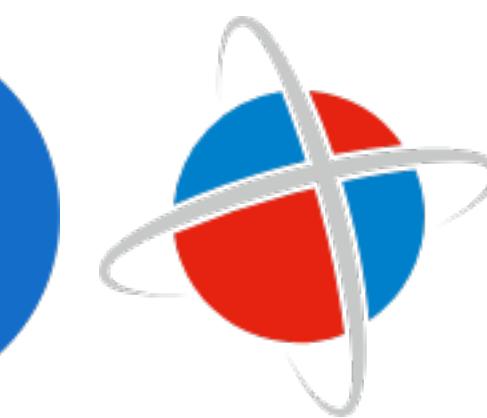
Module-Makers (IDEK how many)

Intermediaries



Device

Makers 3330 registered with Bluetooth SIG as of the time of writing!



晶讯
JINGXUN

ACKme
NETWORKS

Blue
Radios®
A Wireless World

silex
technology

stollmann

Rayson

Laird™

muRata
INNOVATOR IN ELECTRONICS

Product-Makers



DE ALTEK
FOR SECURITY

RESSIF ST



My Terminology

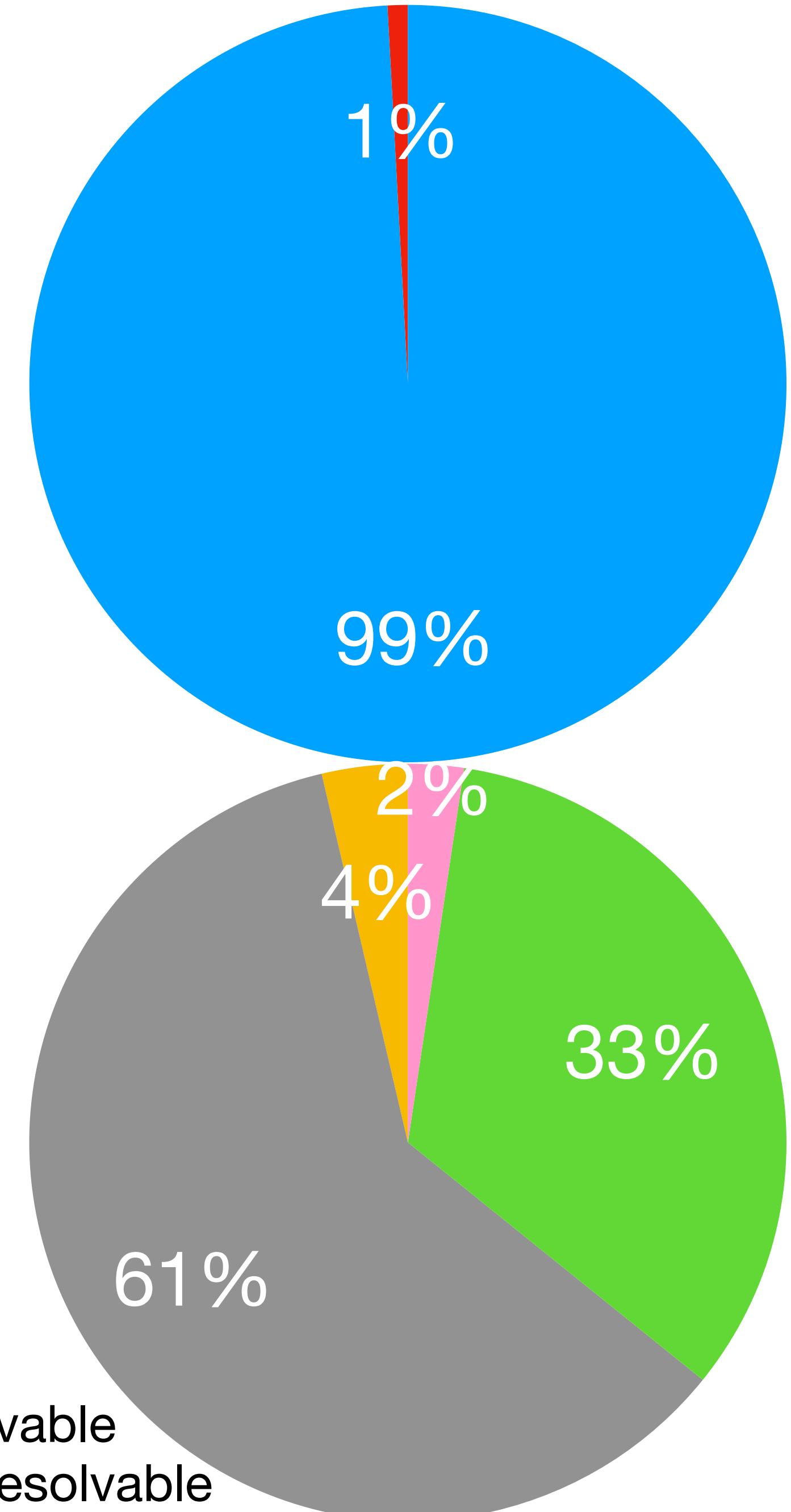
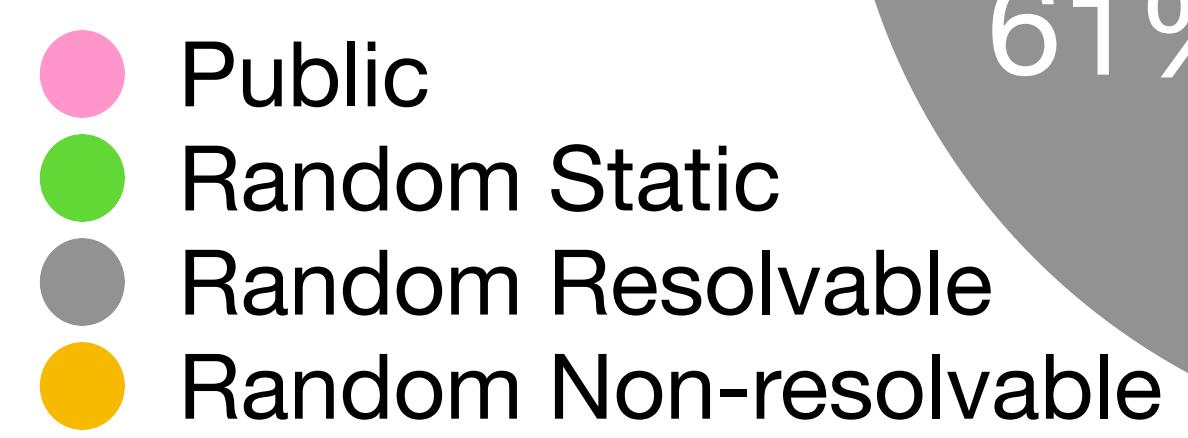
- BTC = Bluetooth Classic 
- BLE = Bluetooth Low Energy
- BDADDR = Bluetooth Device Address (like MAC address)



Overall BDADDR Data

My data as of 2023-10-26

- 73,047 *unique* BT Classic BDADDRs
- 8,392,322 *unique* BLE BDADDRs
 - 194,151 "public" BDADDRs
 - 2,731,623 "random static" BDADDRs
 - 4,950,427 "random resolvable" BDADDRs
 - 299,244 "random non-resolvable" BDADDRs





Passive

2thprinting Approaches



Mostly-Passive



Active



Bluetooth: The Gathering: 🕉️ Passive

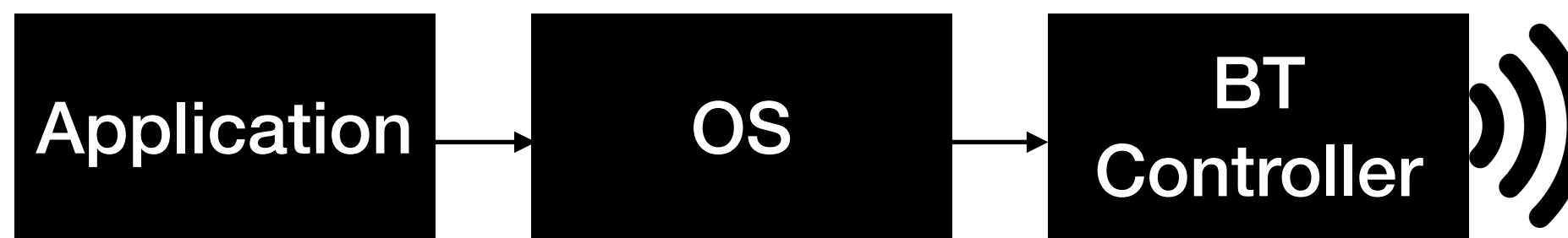
- Run a sniffer, never send *any* packets
- **BLE:** Sniffle, Ice9 Sniffer, **BTC:** Ubertooth
- BLE ADV_IND, NONCONN_ADV_IND





Bluetooth: The Gathering: *Mostly-Passive*

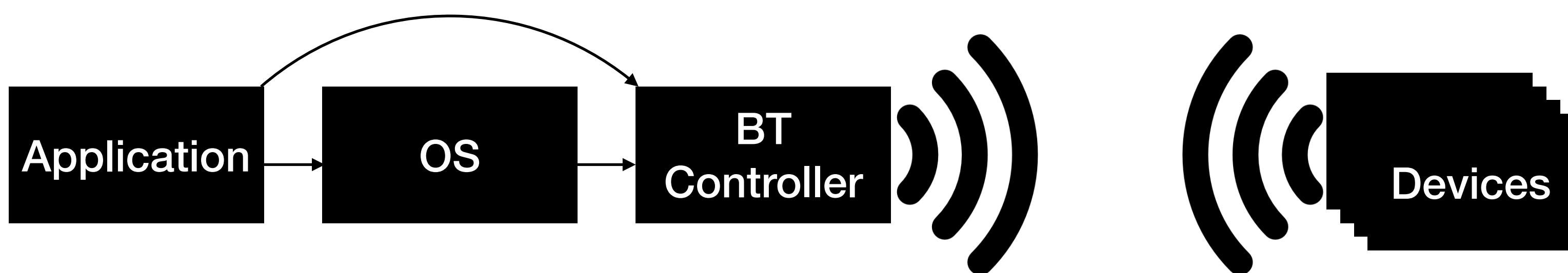
- Ask your OS "What Bluetooth devices are currently visible?"
- It will scan for any advertisements, and *potentially*, autonomously, query some limited information about devices it finds, such as device name, class of device, etc (depends on OS & Bluetooth stack version)
 - This is most of my data





Bluetooth: The Gathering: 🎈 Active

- Send custom BLE LL / BTC LMP packets (requires custom controller firmware)
- Connect to all connectable interfaces, query all queryable information



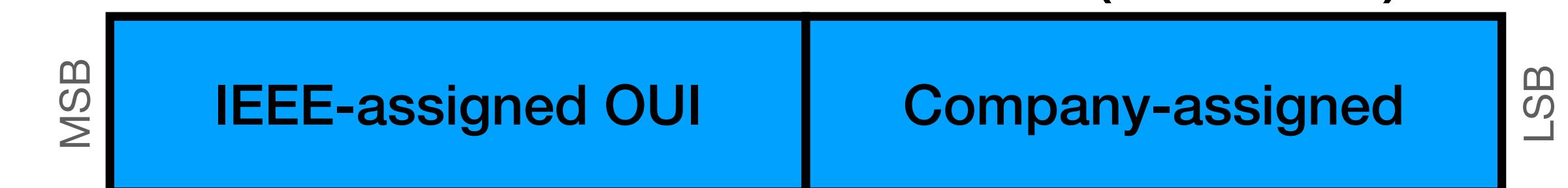
2thprint by BDADDR OUI





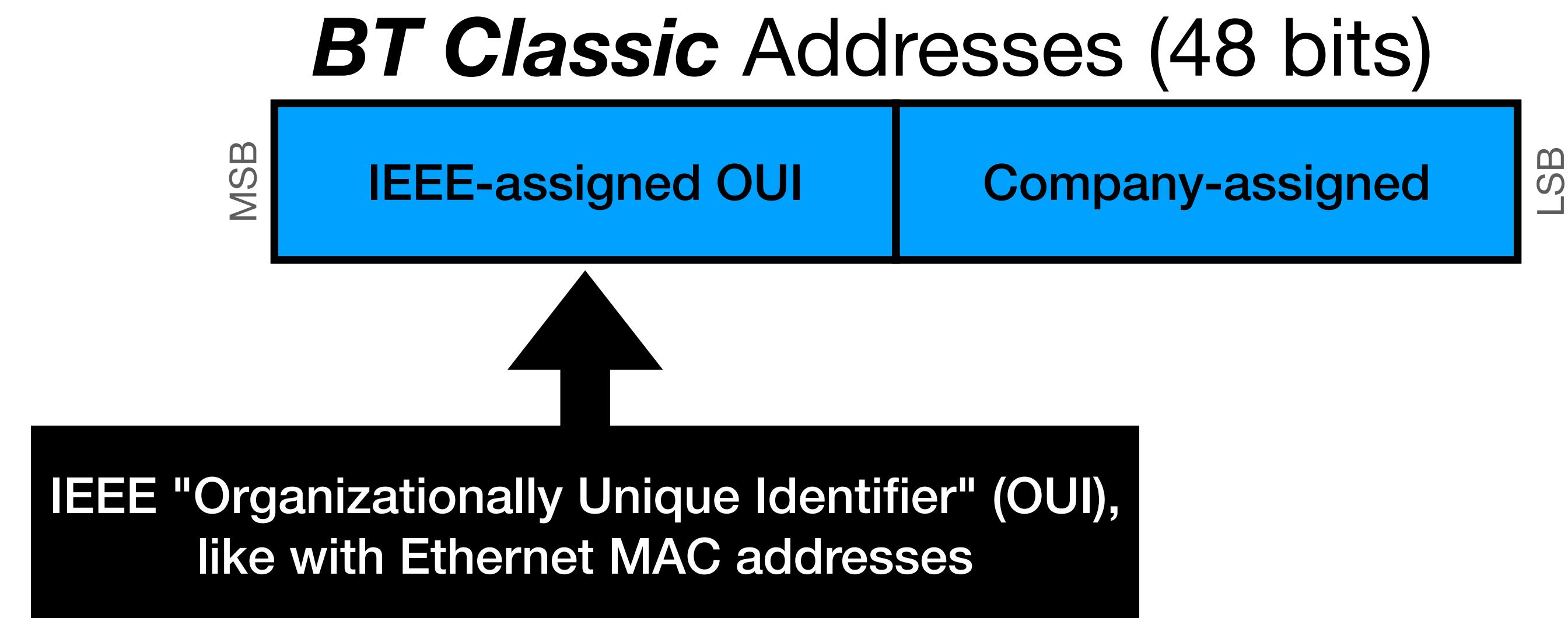
Background - BT Device Address (*BDADDR*)

BT Classic Addresses (48 bits)



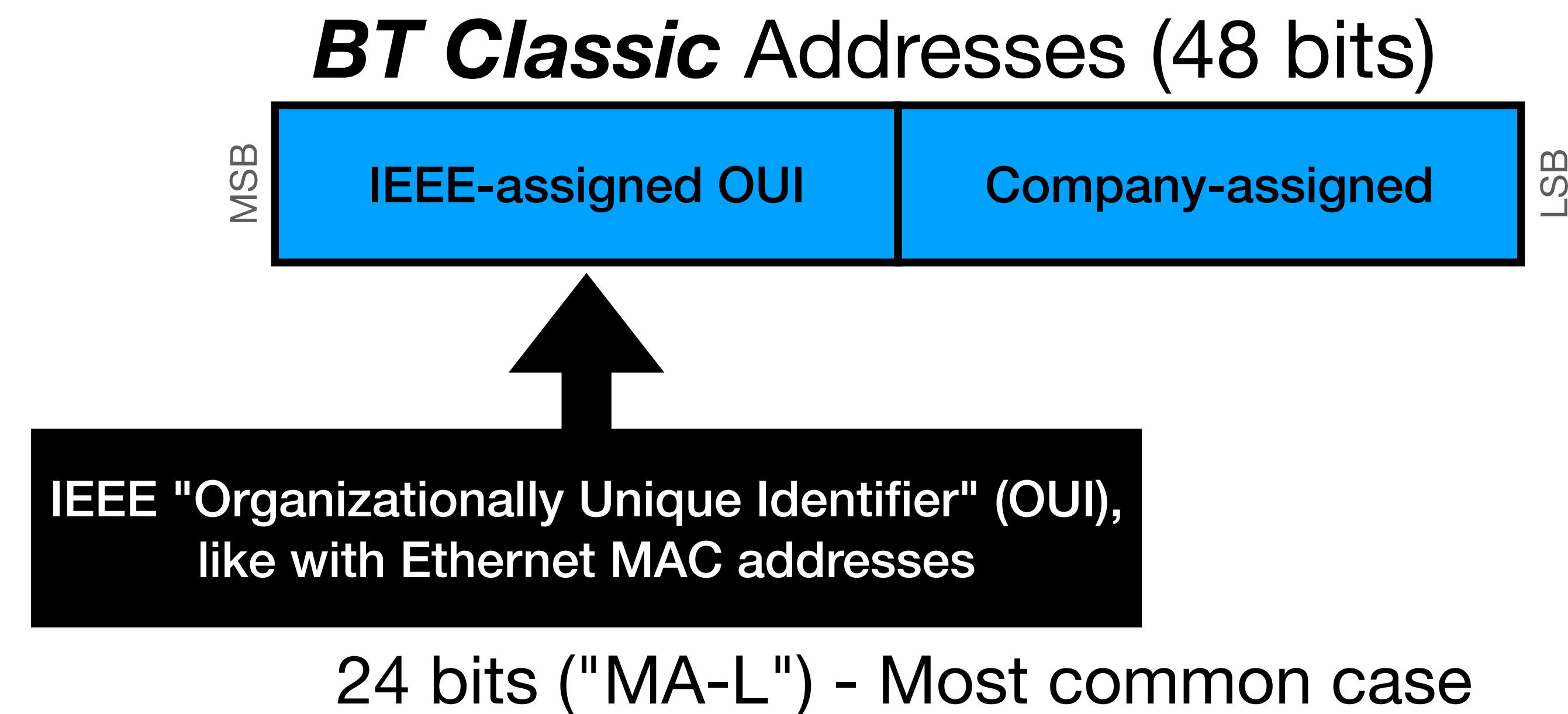


Background - BTC *BT Device Address (BDADDR)*



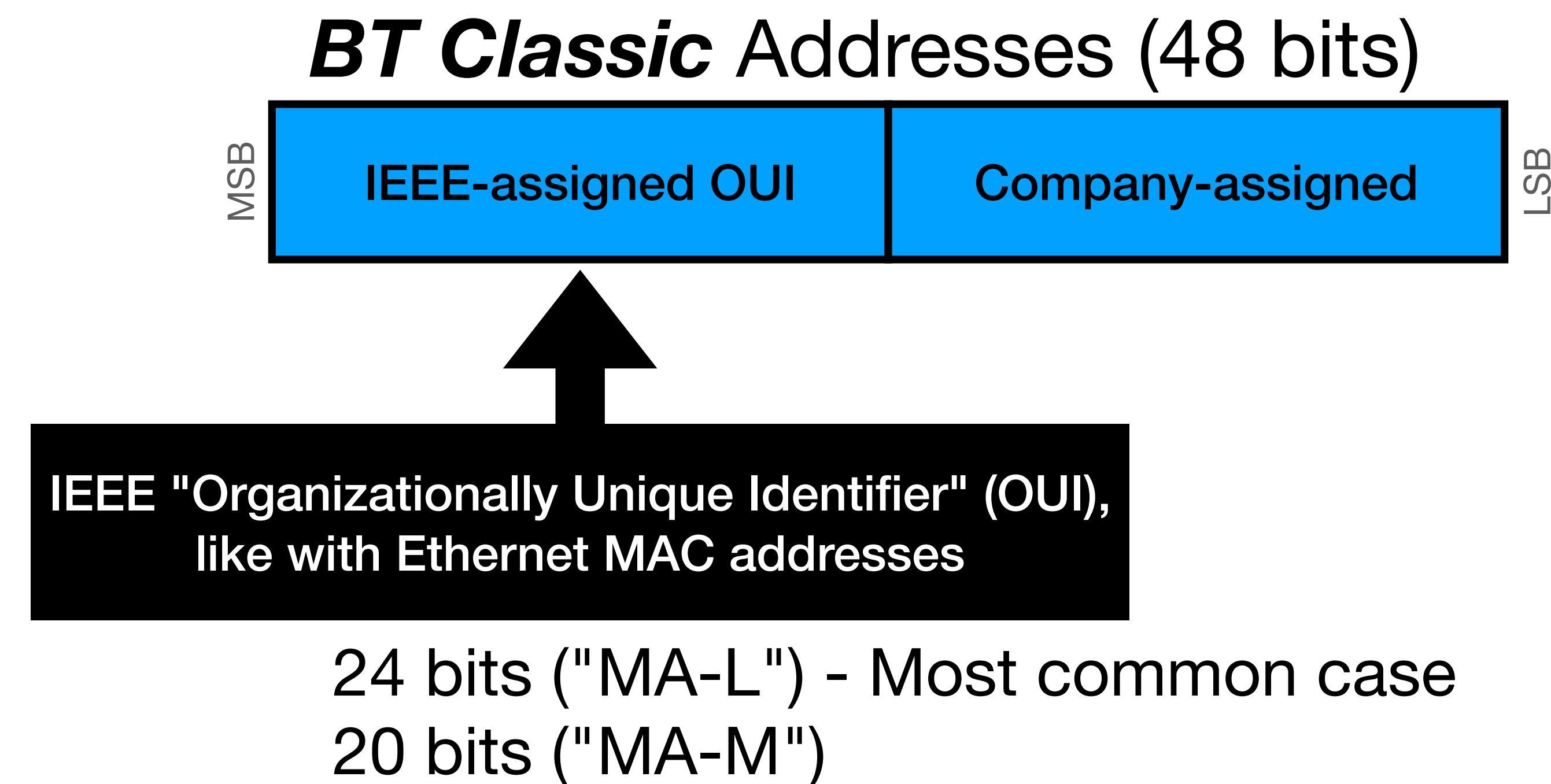


Background - BTC *BT Device Address (BDADDR)*



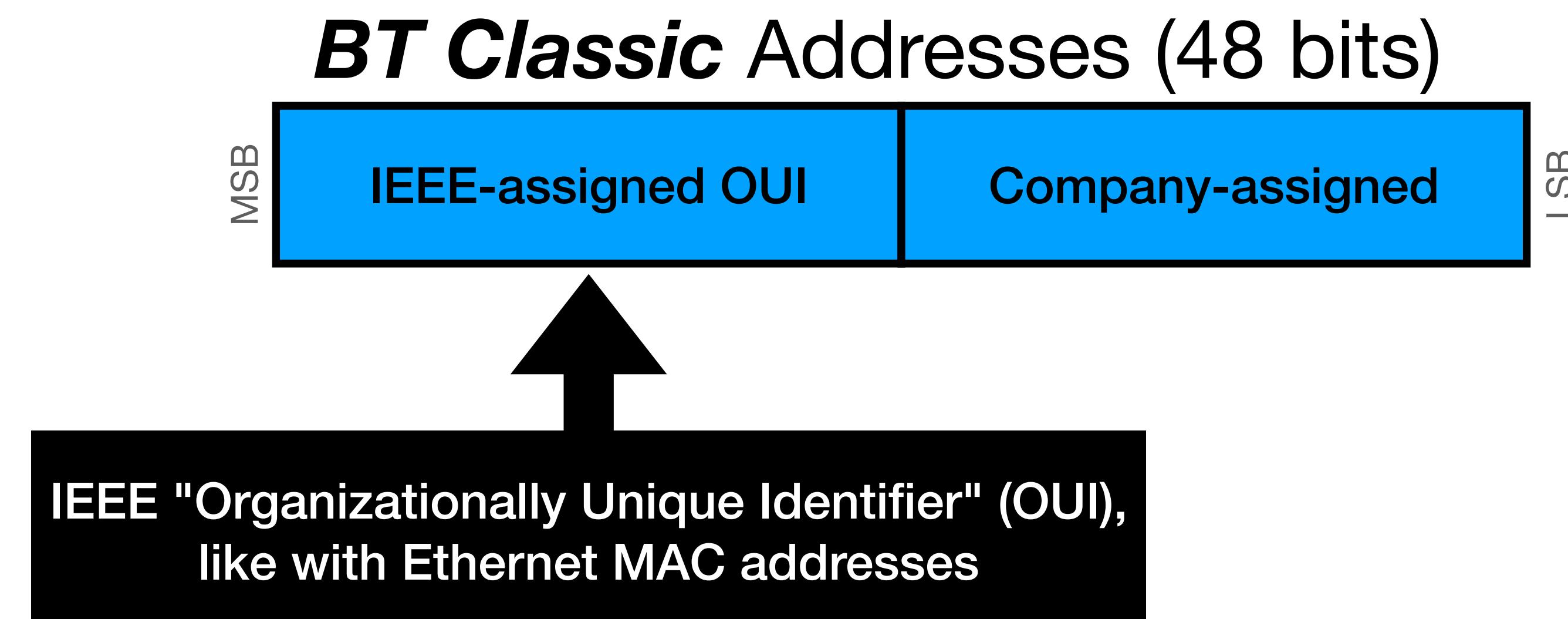


Background - BTC *BT Device Address (BDADDR)*





Background - BTC *BT Device Address (BDADDR)*



24 bits ("MA-L") - Most common case

20 bits ("MA-M")

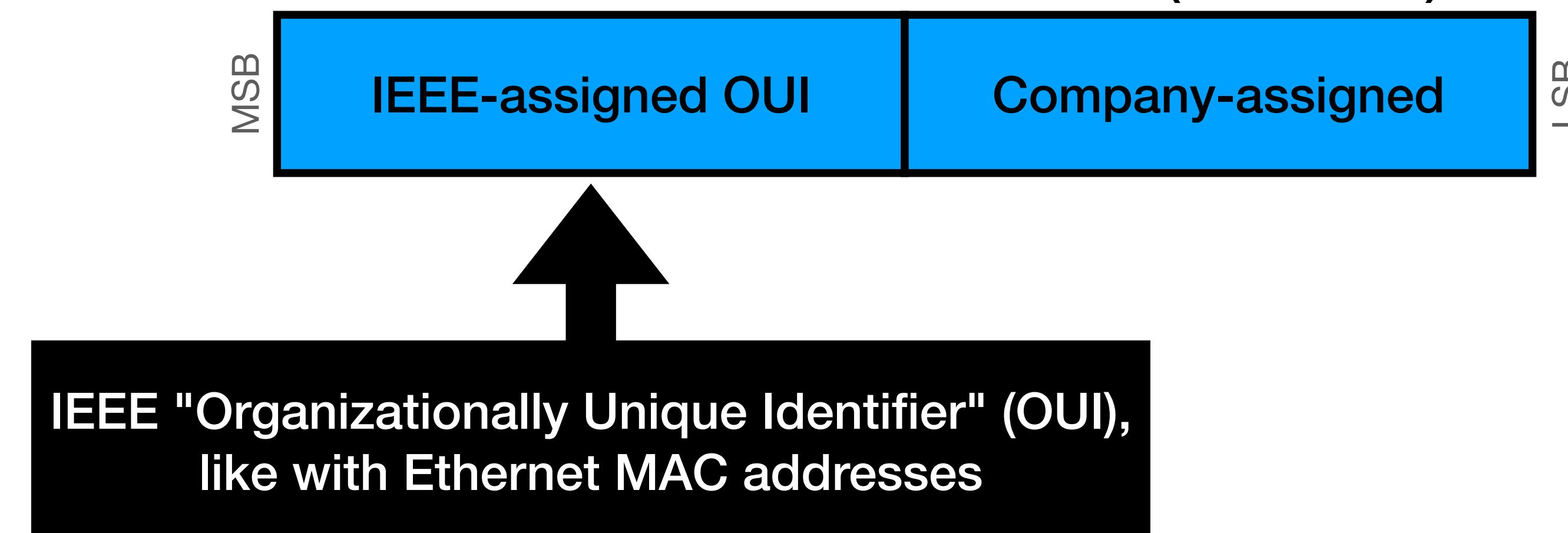
12 bits ("MA-S")



Background - BTC *BT Device Address (BDADDR)*

00:1f:ff:5f:0d:5a

BT Classic Addresses (48 bits)



24 bits ("MA-L") - Most common case

20 bits ("MA-M")

12 bits ("MA-S")

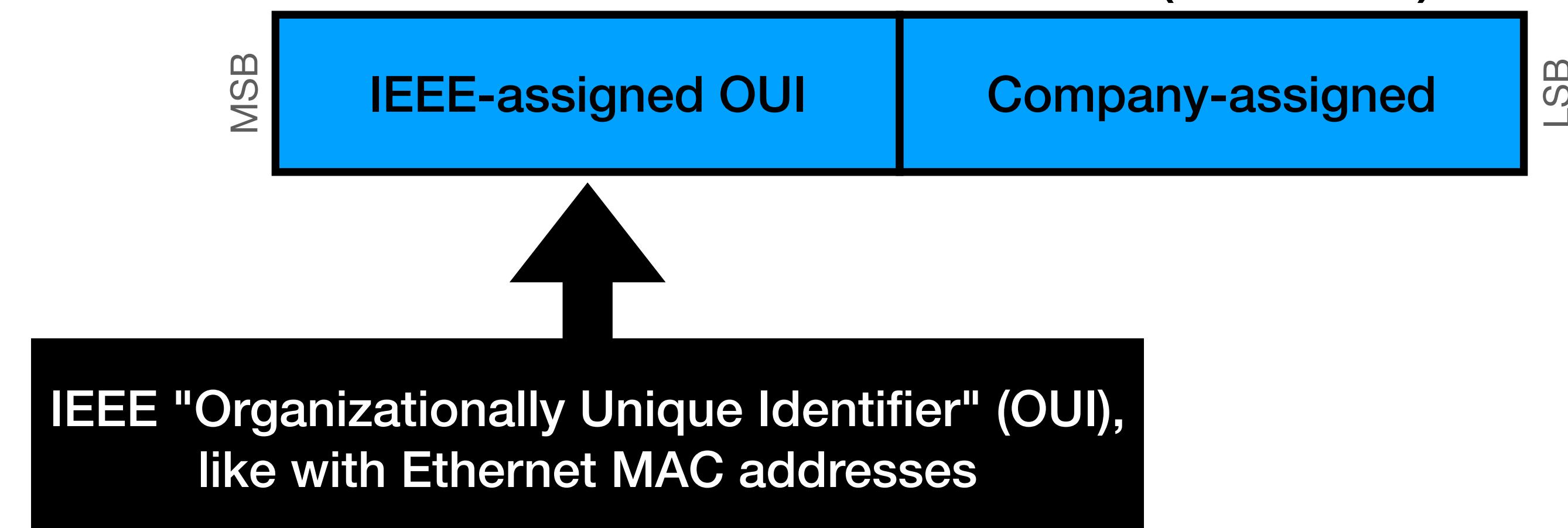


Background - BTC *BT Device Address (BDADDR)*

(00:1f:ff) == Respironics, Inc.

00:1f:ff:5f:0d:5a

BT Classic Addresses (48 bits)



24 bits ("MA-L") - Most common case

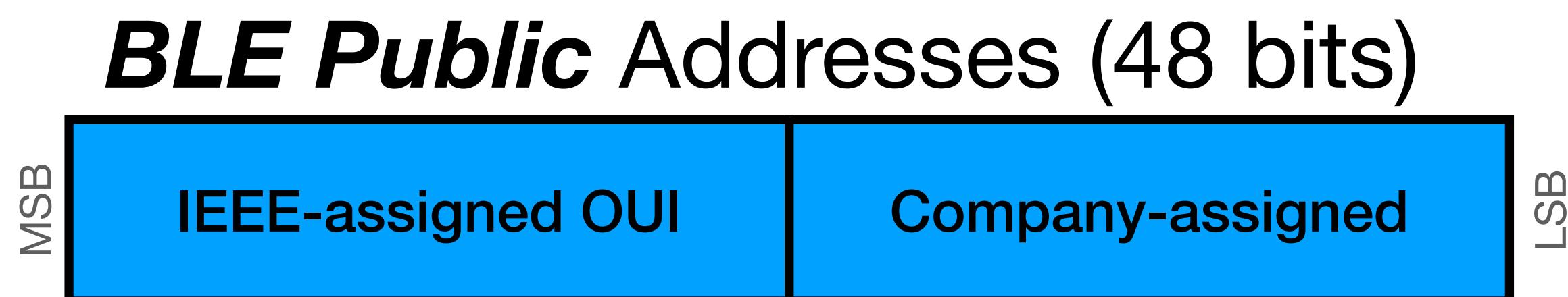
20 bits ("MA-M")

12 bits ("MA-S")



Background - BLE BT Device Address (*BDADDR*)

👋 There's a *bit* in BLE packet headers that says whether a BDADDR is "public" or "random" 👋

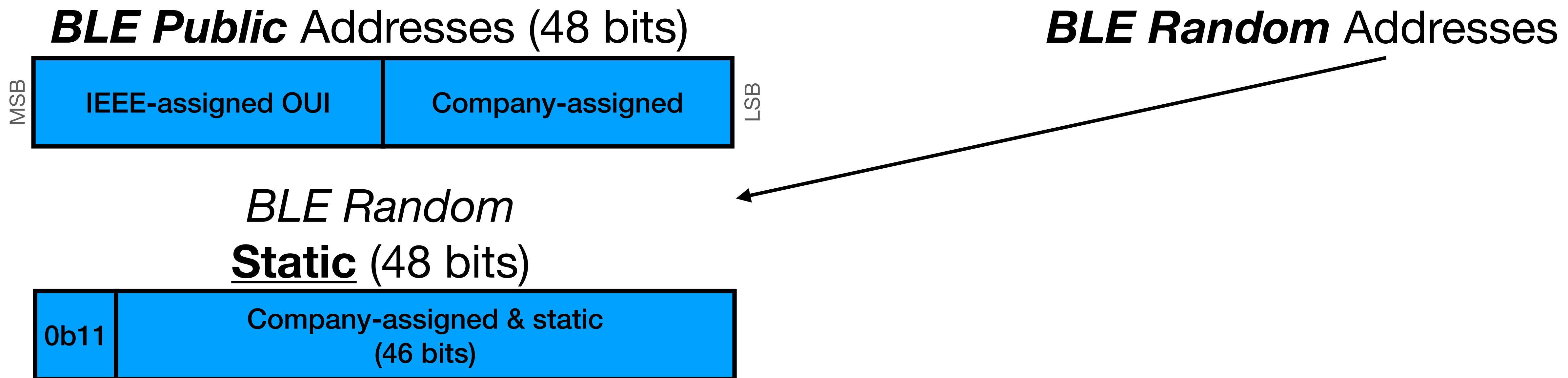


BLE Random Addresses



Background - BLE BT Device Address (**BDADDR**)

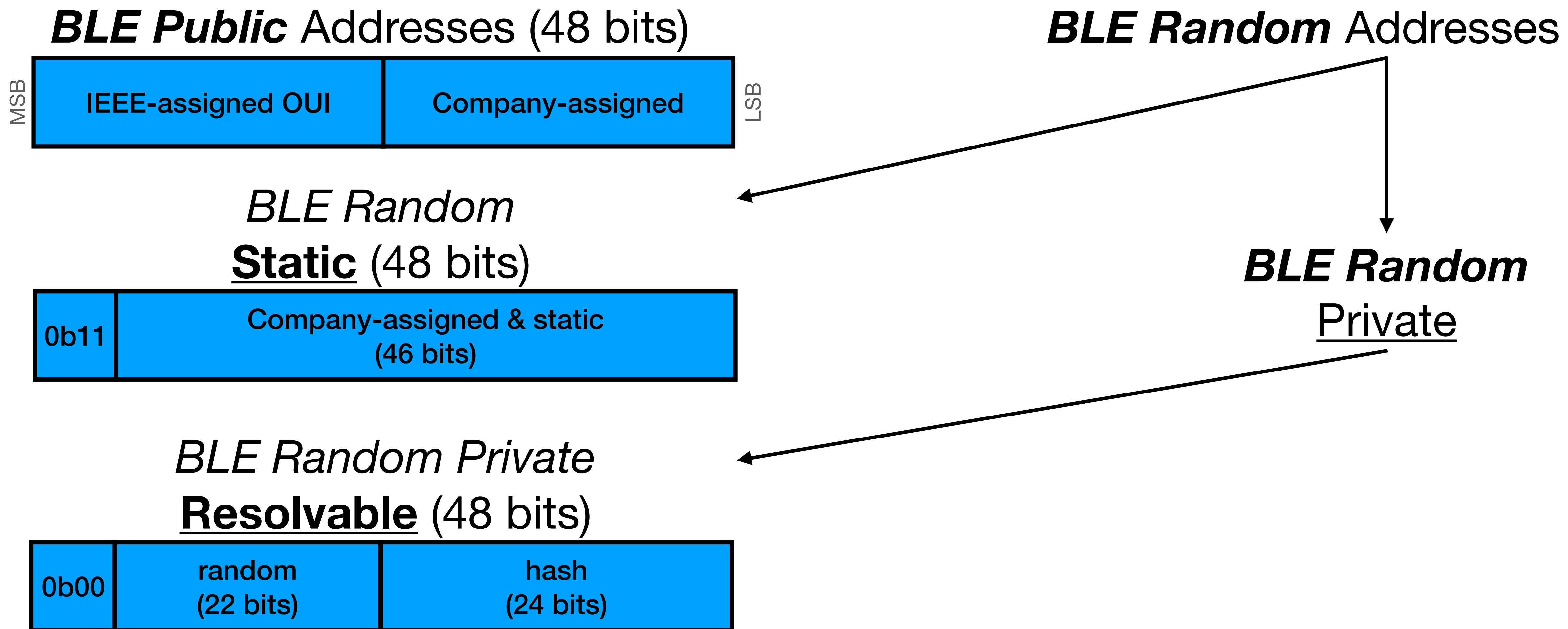
👏 There's a *bit* in BLE packet headers that says whether a BDADDR is "public" or "random" 👏





Background - BLE BT Device Address (**BDADDR**)

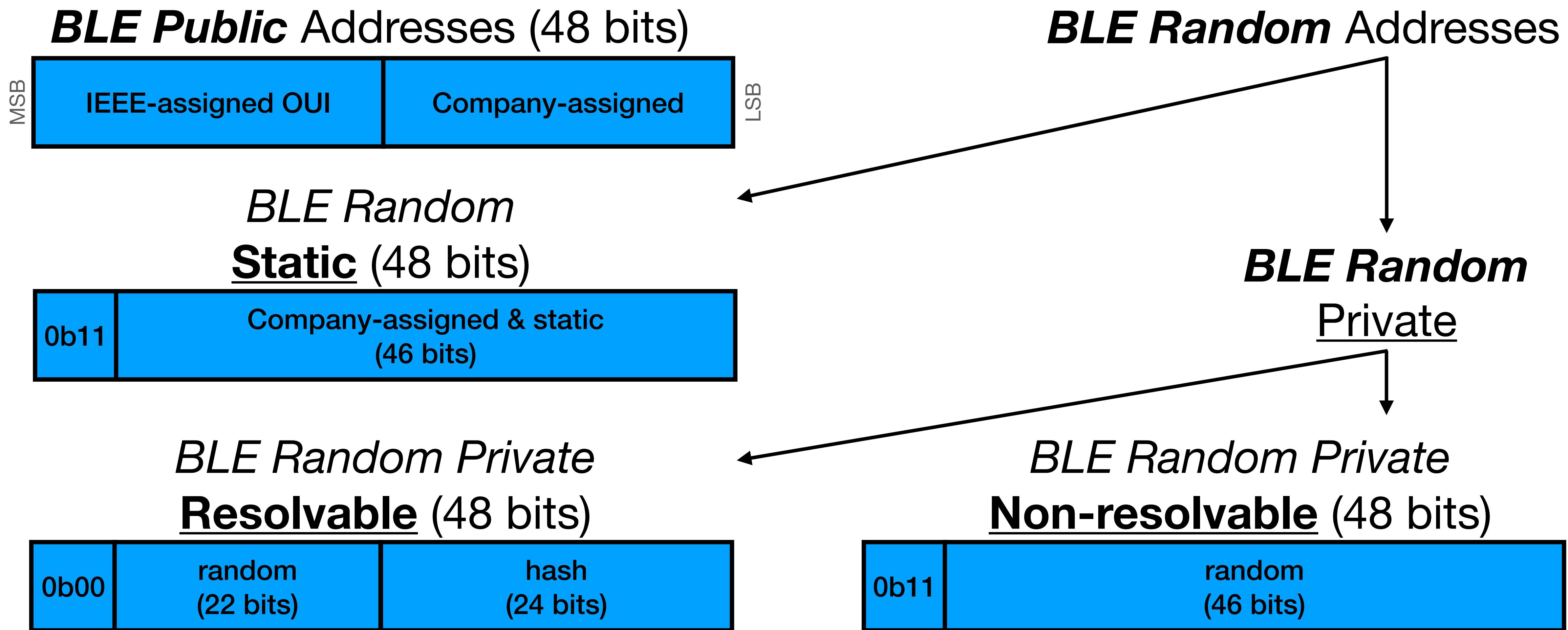
👏 There's a bit in BLE packet headers that says whether a BDADDR is "public" or "random" 👏





Background - BLE BT Device Address (**BDADDR**)

👏 There's a bit in BLE packet headers that says whether a BDADDR is "public" or "random" 👏





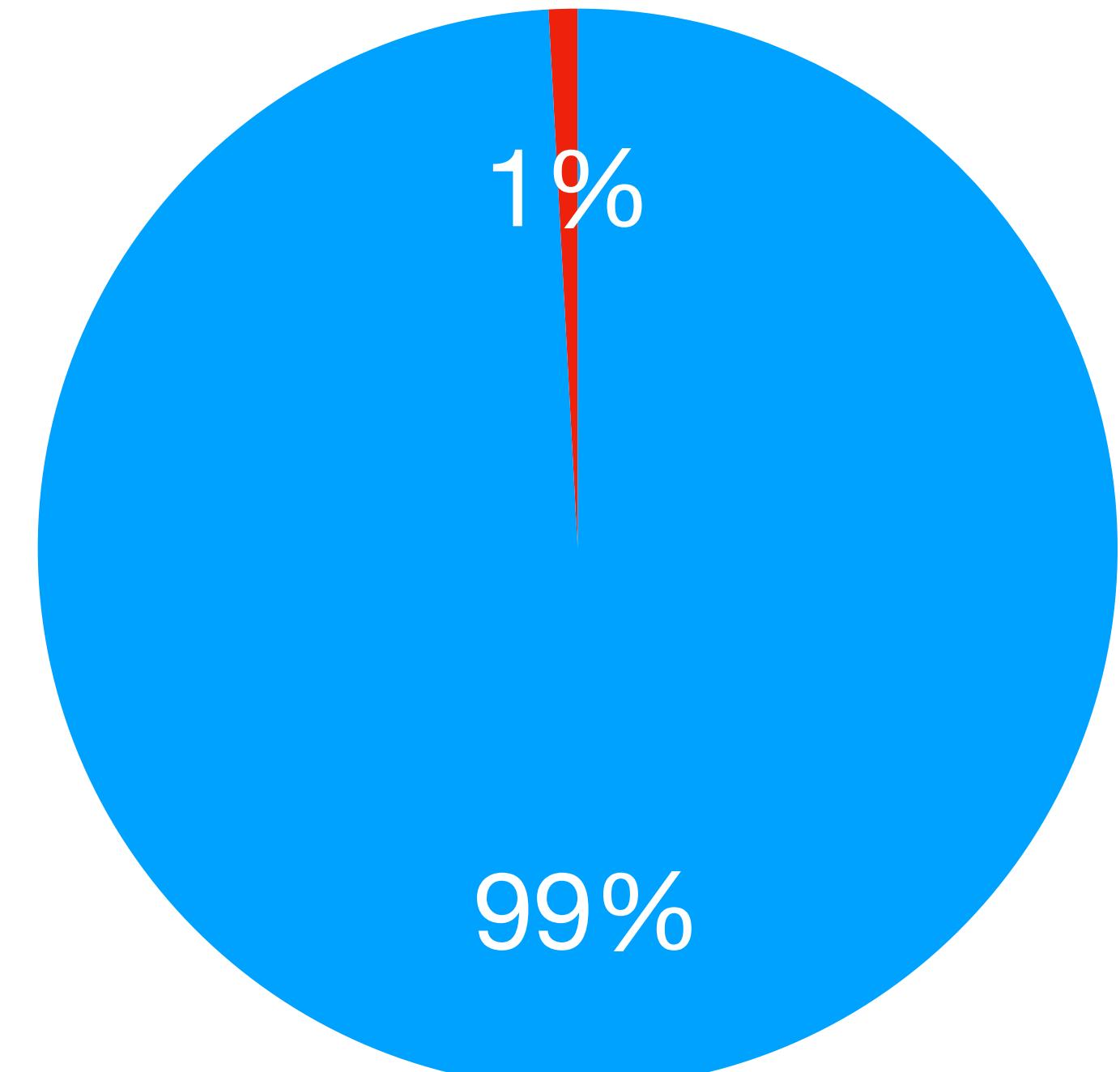
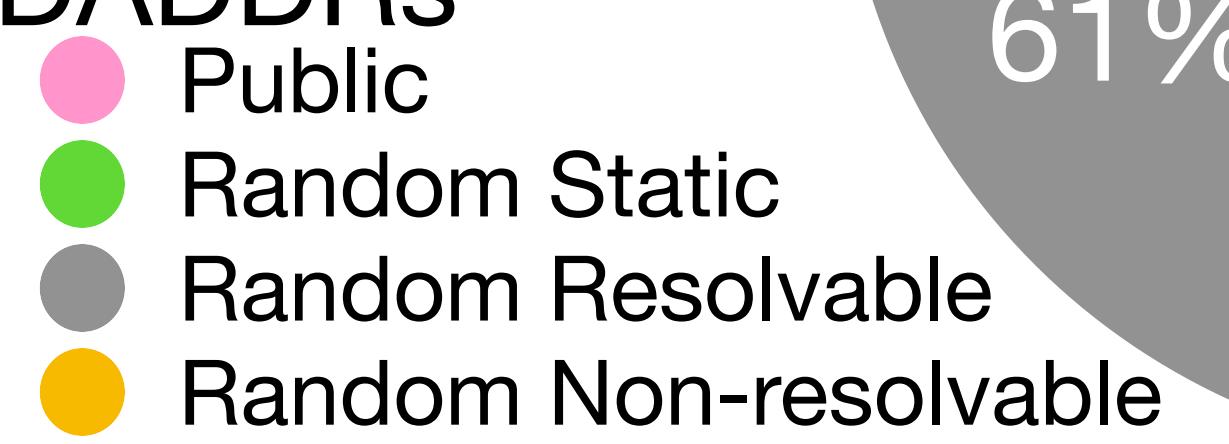
BLE

Classic

Overall BDADDR Data

IEEE OUI Applicability

- 73,047 *unique* BT Classic BDADDRs
- 8,392,322 *unique* BLE BDADDRs
 - 194,151 "public" BDADDRs
 - 2,731,623 "random static" BDADDRs
 - 4,950,427 "random resolvable" BDADDRs
 - 299,244 "random non-resolvable" BDADDRs
- So OUIPrints applicable to only ~3.2% of the BDADDRs
 $(73,047+194,151)/(73,047+8,392,322)$





BTC Data (top 20 of 593 companies seen)

+-----+		
company_by_bdaddr found		
+-----+		
GPS/Watches/etc	Garmin International	10423
📱 📺	Samsung Electronics Co.,Ltd Qualcomm, Broadcom, MediaTek, Samsung 🍪	6359
📱	OnePlus Technology (Shenzhen) Co., Ltd	3682
🍪	Actions Semiconductor Co.,Ltd.(Cayman Islands)	3036
📱	Apple, Inc. <i>Mostly Broadcom, Sometimes Apple (e.g. AirPods)</i> 🍪	2604
🎵 🚗 🧩	Panasonic Automotive Systems Co.,Ltd	2491
🎵 🚗 🧩	Laird Connectivity	2238
🎵 🚗 🧩	PIONEER CORPORATION	1977
🍪	Intel Corporate	1960
🚗	ALPSALPINE CO,.LTD	1919
📱	GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP.,LTD	1648
🧩	AzureWave Technology Inc. NXP, Cypress, MediaTek 🍪	1369
🍪	Texas Instruments	974
📱	Wistron Neweb Corporation	920
🧩	silex technology, Inc. Qualcomm 🍪	890
🎵 🚗 🧩	Shinwa Industries(China) Ltd. Qualcomm/CSR, TI, Cypress, Sunplus 🍪	858
🎵 🚗 🧩	MITSUMI ELECTRIC CO.,LTD. CSR->Qualcomm 🍪	845
🎵 🚗 🧩	PARROT SA TI 🍪	741
📱 integrator?	Wingtech Mobile Communications Co., Ltd. MediaTek? 🍪	690
🎵 🚗 🍪	Sunplus Technology Co., Ltd.	606



BTC Data (top 20 of 593 companies seen)

+-----+ company_by_bdaddr found +-----+		
GPS/Watches/etc		
📱 📺	Garmin International	10423
📱 🍪	Samsung Electronics Co., Ltd Qualcomm, Broadcom, MediaTek, Samsung 🍪	6359
📱	OnePlus Technology (Shenzhen) Co., Ltd	3682
🍪	Actions Semiconductor Co., Ltd.(Cayman Islands)	3036
📱	Apple, Inc. <i>Mostly Broadcom, Sometimes Apple (e.g. AirPods)</i> 🍪	2604
🎵 🚗 🧩	Panasonic Automotive Systems Co.,Ltd	2491
🎵 🚗 🧩	Laird Connectivity	2238
🎵 🚗 🧩	PIONEER CORPORATION	
🍪	Intel Corporate	
🚗	ALPSALPINE CO,.LTD	1919
📱	GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP., LTD	1648
🧩	AzureWave Technology Inc. NXP, Cypress, MediaTek 🍪	1369
🍪	Texas Instruments	974
📱	Wistron Neweb Corporation	920
🧩	silex technology, Inc. Qualcomm 🍪	890
🎵 🚗 🧩	Shinwa Industries(China) Ltd. Qualcomm/CSR, TI, Cypress, Sunplus 🍪	858
🎵 🚗 🧩	MITSUMI ELECTRIC CO.,LTD. CSR->Qualcomm 🍪	845
🎵 🚗 🧩	PARROT SA TI 🍪	741
📱 integrator?	Wingtech Mobile Communications Co., Ltd. MediaTek? 🍪	690
🎵 🚗 🍪	Sunplus Technology Co., Ltd.	606



Note! My data is skewed towards vehicles, because I like to put my sniffers over freeways!



BLE Data (top 20 of 608 companies seen)

		found	
	company_by_bdaddr		
	+-----+-----+		
	Texas Instruments	31799	
	VXi Corporation	25190	
	Samsung Electronics Co.,Ltd Qualcomm, Broadcom, MediaTek, Samsung	12340	
	Bose Corporation	8475	
	Logitech, Inc	4980	
	Cambridge Mobile Telematics, Inc.	CSR->Qualcomm	4958
	Apple, Inc.	Mostly Broadcom, Sometimes Apple (e.g. AirPods)	4821
	Silicon Laboratories		3802
	Espressif Inc.		2975
BLE beacons	Shenzhen Minew Technologies Co., Ltd.		2462
	Murata Manufacturing Co., Ltd. Nordic, Cypress, CSR, Onsemi, Dialog	2276	
	Telink Semiconductor (Taipei) Co. Ltd.		1943
GPS/Watches/etc	Shenzhen Jingxun Software Telecommunication Technology Co.,Ltd		1879 RealTek, Airoha
	Garmin International		1651
	Sunitec Enterprise Co.,Ltd	Broadcom	1426
BLE beacons	Aruba, a Hewlett Packard Enterprise Company		1388
LED lights & sensors	Shenzhen Intellirocks Tech co.,ltd	Telink	1139
	Shanghai Rui Rui Communication Technology Co.Ltd.		1070 (TI bought OUI?)
	ALPSALPINE CO,.LTD		969
🎵 Hearing aids	Starkey Labs Inc.		934



Of what I want to know

- If the BDADDR maps to a chip-maker, we have an idea of what chip maker is being used with high probability

"Texas Instruments"	"Silicon Laboratories"	"Intel Corporate"
"Telink Semiconductor (Taipei) Co. Ltd."	"Cambridge Silicon Radio"	"Espressif Inc."
"Nordic Semiconductor ASA"	"NXP Semiconductors"	"NXP France Semiconductors France"
"NXP Semiconductor (Tianjin) LTD."	"NXP (China) Management Ltd."	"Microchip Technology Inc."
"Broadcom"	"Qualcomm Technologies International, Ltd. (QTIL)"	"REALTEK SEMICONDUCTOR CORP."

- For some other names (like module-makers), we might know that there's only 1 or 2 chips they ever use, though of course that can change
 - Need to collect that data over time



The First Traces

Of what I want to know

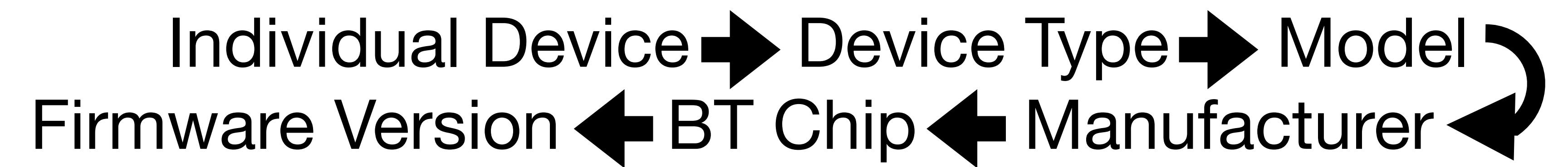
- If the BDADDR maps to a chip-maker, we have an idea of what chip maker is being used with high probability

"Texas Instruments"	"Silicon Laboratories"	"Intel Corporate"
"Telink Semiconductor (Taipei) Co. Ltd."	"Cambridge Silicon Radio"	"Espressif Inc."
"Nordic Semiconductor ASA"	"NXP Semiconductors"	"NXP France Semiconductors France"
"NXP Semiconductor (Tianjin) LTD."	"NXP (China) Management Ltd."	"Microchip Technology Inc."
"Broadcom"	"Qualcomm Technologies International, Ltd. (QTIL)"	"REALTEK SEMICONDUCTOR CORP."

- For some other names (like module-makers), we might know that there's only 1 or 2 chips they ever use, though of course that can change
 - Need to collect that data over time



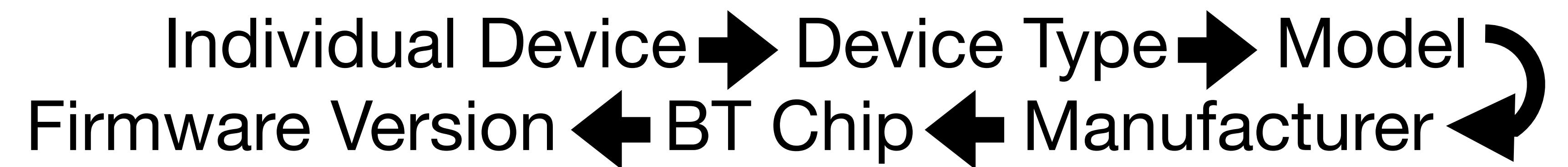
What I Want





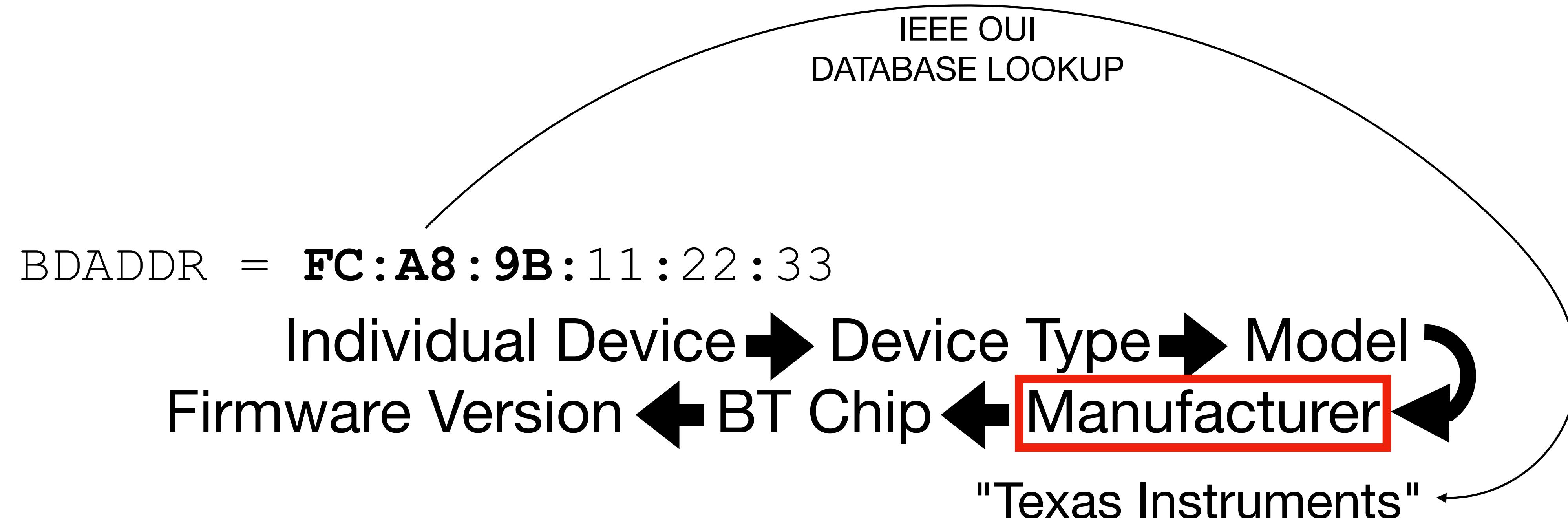
What I Want

BDADDR = **FC:A8:9B:11:22:33**



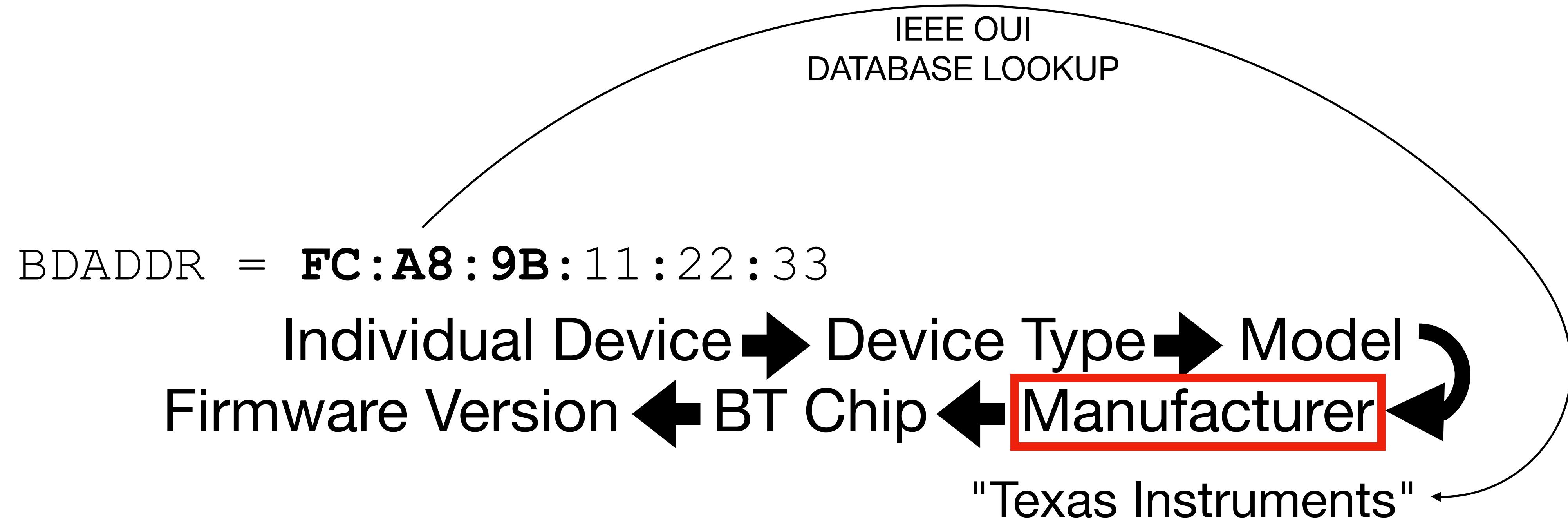


What I Want





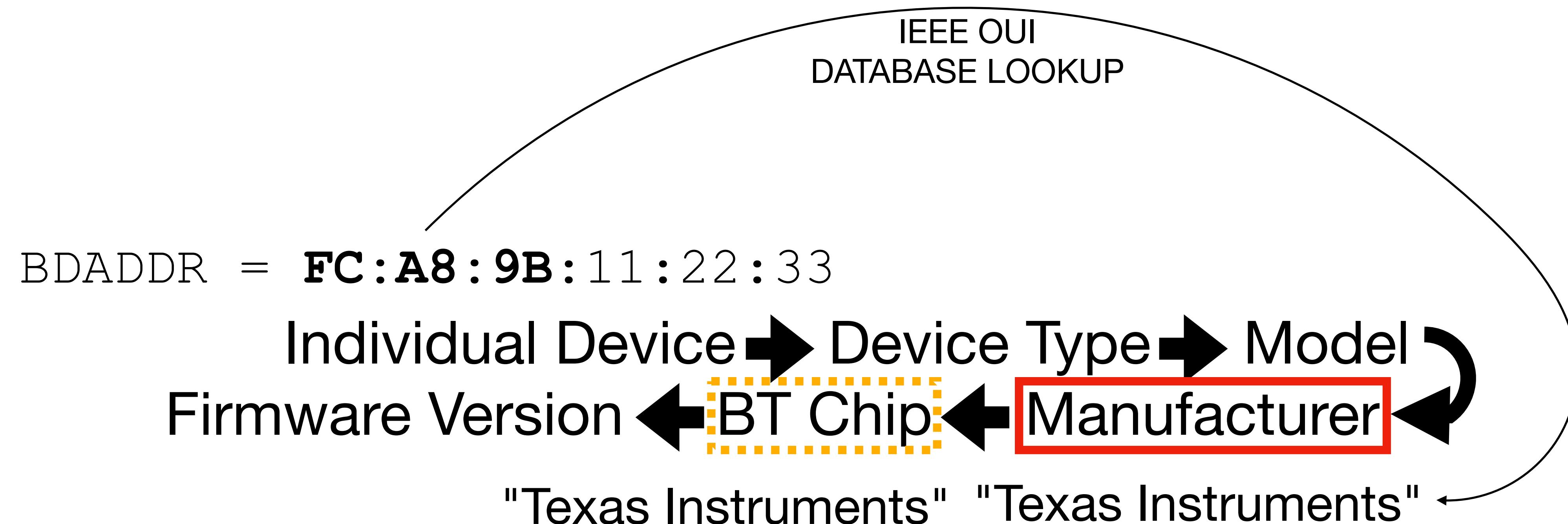
What I Want



ASSUMPTION:
OUIPrint == ChipPrint, for OUI == {Silicon Vendor OUIs}



What I Want



ASSUMPTION:
OUIPrint == ChipPrint, for OUI == {Silicon Vendor OUIs}

2thprint by Link Layer Version Info

or

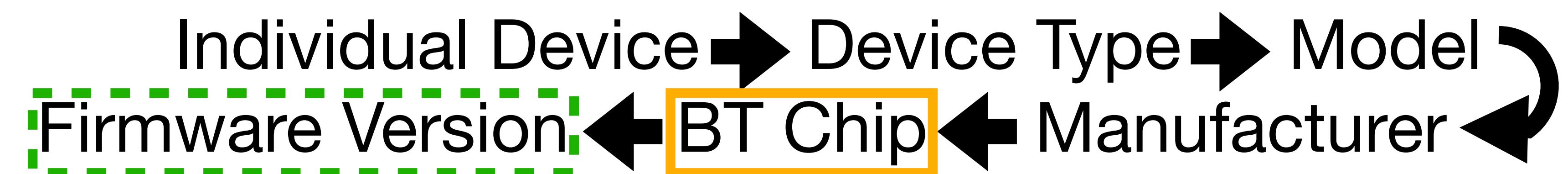




2thprint by Link Layer Version Information

LL_VERSION_IND (BLE), LMP_version_res (BTC)

- BTC has Link Management Protocol (LMP) and BLE has Link-Layer (LL) Control packets, that can be sent to request some chip and firmware version information
- These can be sent/received *without* any sort of BT pairing/bonding!

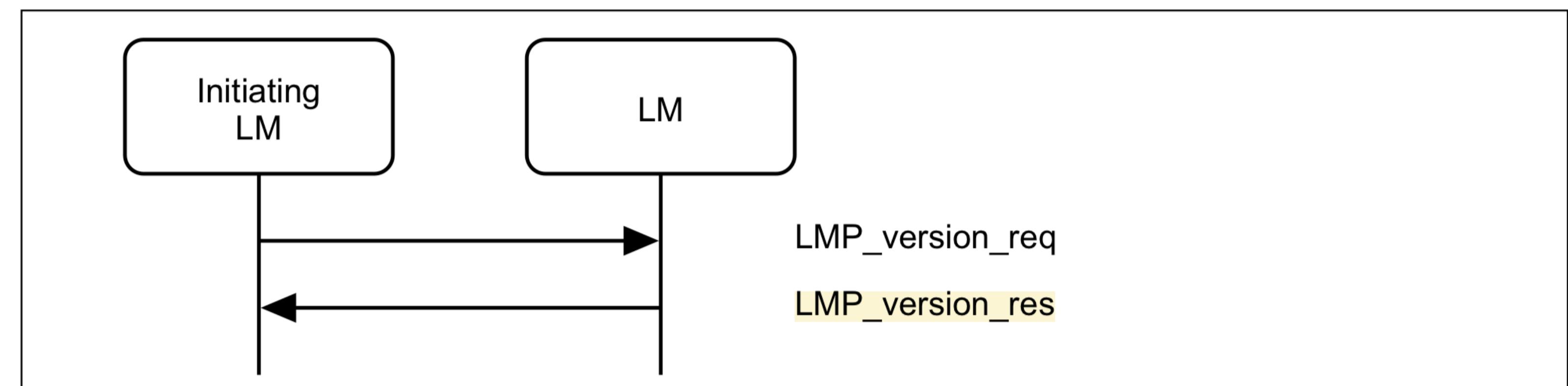




2thprint by LMP_version_res (BTC)

M/O	PDU	Contents
M	LMP_version_req	VersNr Compld SubVersNr
M	LMP_version_res	VersNr Compld SubVersNr

Table 4.26: PDUs used for LMP version request



Sequence 77: Request for LMP version



2thprint by **LL_VERSION_IND (BLE)**

2.4.2.13 **LL_VERSION_IND**

The format of the CtrData field is shown in [Figure 2.23](#).

CtrData		
VersNr (1 octet)	Compld (2 octets)	SubVersNr (2 octets)

Figure 2.23: CtrData field of the LL_VERSION_IND PDU

The **LL_VERSION_IND** CtrData consists of three fields:

- VersNr field shall contain the version of the Bluetooth Controller specification (see Bluetooth [Assigned Numbers](#)).
- Compld field shall contain the company identifier of the manufacturer of the Bluetooth Controller (see Bluetooth [Assigned Numbers](#)).
- SubVersNr field shall contain a unique value for each implementation or revision of an implementation of the Bluetooth Controller.



Prior Work

"ESPwn32: Hacking with ESP32 System-on-Chips"

- Oh no! I got scooped! (Or did I?)
- [1] by Cayre et al. from May 2023 recognized that LL_VERSION_IND packets contains useful information
- But it subsequently *assumes* this information is *sufficient* for vulnerability applicability assessment, without offering any proof / data analysis





Prior Work

"ESPwn32: Hacking with ESP32 System-on-Chips"

- Oh no! I got scooped! (Or did I?)
- [1] by Cayre et al. from May 2023 recognized that LL_VERSION_IND packets contains useful information
- But it subsequently *assumes* this information is *sufficient* for vulnerability applicability assessment, without offering any proof / data analysis





LMP/LL 2thprints useful, but not definitive

- It turns out that for MediaTek (CID = 70), the most common value for the sub version is 0

device_BT_CID	lmp_sub_version	frequency
70	0	774
70	288	189
70	4648	86
70	4101	62
70	2051	15
70	1571	14
70	613	11
70	533	10
70	2344	10
70	1030	9
70	4391	8
70	1797	7
70	4135	5
70	304	4
70	726	4
70	791	4
70	1560	4
70	1817	4
...		

device_BT_CID	ll_sub_version	frequency
70	0	36
70	534	8
70	4101	8
70	4648	6
70	2051	5
70	4355	3
70	4373	2
70	288	1
70	304	1
70	546	1
70	776	1
70	791	1
70	1033	1
70	1042	1
70	1045	1
70	1568	1
70	4097	1
70	4116	1
...		



LMP/LL 2thprints useful, but not definitive

- It turns out that for MediaTek (CID = 70), the most common value for the sub version is 0

device_BT_CID	lmp_sub_version	frequency
70	0	774
70	288	189
70	4648	86
70	4101	62
70	2051	15
70	1571	14
70	613	11
70	533	10
70	2344	10
70	1030	9
70	4391	8
70	1797	7
70	4135	5
70	304	4
70	726	4
70	791	4
70	1560	4
70	1817	4
...		

device_BT_CID	ll_sub_version	frequency
70	0	36
70	534	8
70	4101	8
70	4648	6
70	2051	5
70	4355	3
70	4373	2
70	288	1
70	304	1
70	546	1
70	776	1
70	791	1
70	1033	1
70	1042	1
70	1045	1
70	1568	1
70	4097	1
70	4116	1
...		

LMP/LL 2thprints useful, but not definitive

- It turns out that for MediaTek (CID = 70), the most common value for the sub version is 0

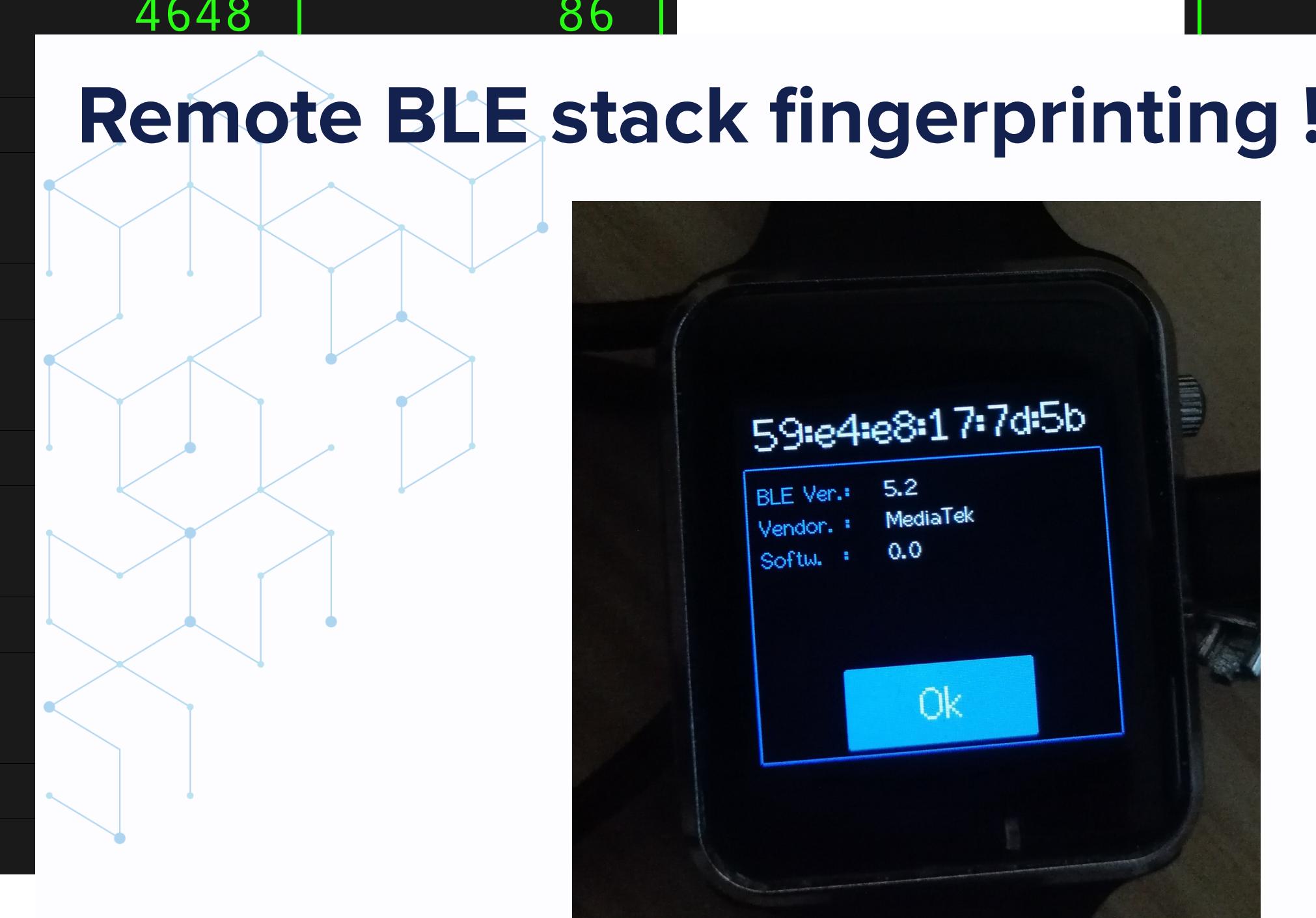
device_BT_CID	lmp_sub_version	frequency
70	0	774
70	288	189
70	4648	86
70	4101	62
70	2051	15
70	1571	14
70	613	11
70	533	10
70	2344	10
70	1030	9
70	4391	8
70	1797	7
70	4135	5
70	304	4
70	726	4
70	791	4
70	1560	4
70	1817	4
...		

device_BT_CID	ll_sub_version	frequency
70	0	36
70	534	8
70	4101	8
70	4648	6
70	2051	5
70	4355	3
70	4373	2
70	288	1
70	304	1
70	546	1
70	776	1
70	791	1
70	1033	1
70	1042	1
70	1045	1
70	1568	1
70	4097	1
70	4116	1
...		



LMP/LL 2thprints useful, but not definitive

- It turns out that for MediaTek (CID = 70), the most common value for the sub version is 0



device_BT_CID	ll_sub_version	frequency
70	0	36
70	534	8
70	4101	8
	4648	6
	2051	5
	4355	3
	4373	2
	288	1
	304	1
	546	1
	776	1
	791	1
	1033	1
	1042	1
	1045	1
	1568	1
	4097	1
	4116	1

OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

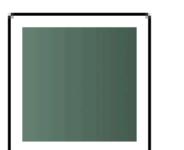
CA\$649.99



device_BT_CID	ll_sub_version	frequency
70	4101	62
70	2051	15
70	1571	14
70	613	11
70	533	10
70	2344	10
70	1030	9
70	4391	8
70	1797	7
70	4135	5
70	304	4
70	726	4
70	791	4
70	1560	4
70	1817	4
...		

But not definitive

Color: Halo Green



ROM

8 GB RAM + 128 GB Storage

most common value for the sub

device_BT_CID	ll_sub_version	frequency
70	0	36
70	534	8
70	4101	8
70	4648	6
70	2051	5
70	4355	3
70	4373	2
70	288	1
70	304	1
70	546	1
70	776	1
70	791	1
70	1033	1
70	1042	1
70	1045	1
70	1568	1
70	4097	1
70	4116	1

OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA\$649.99

Color: Halo Green



ROM

8 GB RAM + 128 GB Storage



70	4101	62
70	2051	15
70	1571	14
70	613	11
70	533	10
70	2344	10
70	1030	9
70	4391	8
70	1797	7
70	4135	5
70	304	4
70	726	4
70	791	4
70	1560	4
70	1817	4
...		





Nokia 130
Feed your playful side

OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA\$649.99

Color: Halo Green



ROM

8 GB RAM + 128 GB Storage





Nokia 130
Feed your playful side

OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA\$649.99

Color: Halo Green



ROM

8 GB RAM + 128 GB Storage





(Link Layer) Version Number

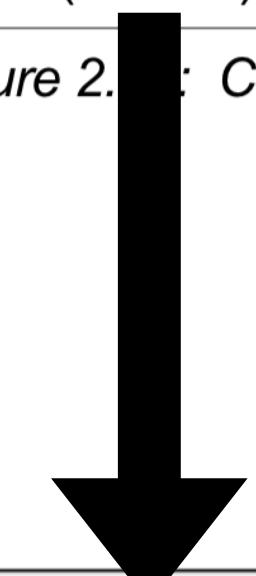
VersNr (1 byte)

Last Modified: 2023-02-21

Core Specification Name	Version
Bluetooth® Core Specification 1.0b (Withdrawn)	0x00
Bluetooth® Core Specification 1.1 (Withdrawn)	0x01
Bluetooth® Core Specification 1.2 (Withdrawn)	0x02
Bluetooth® Core Specification 2.0+EDR (Withdrawn)	0x03
Bluetooth® Core Specification 2.1+EDR (Withdrawn)	0x04
Bluetooth® Core Specification 3.0+HS (Withdrawn)	0x05
Bluetooth® Core Specification 4.0 (Withdrawn)	0x06
Bluetooth® Core Specification 4.1 (Deprecated)	0x07
Bluetooth® Core Specification 4.2	0x08
Bluetooth® Core Specification 5.0	0x09
Bluetooth® Core Specification 5.1	0x0A
Bluetooth® Core Specification 5.2	0x0B
Bluetooth® Core Specification 5.3	0x0C
Bluetooth® Core Specification 5.4	0x0D

CtrData		
VersNr (1 octet)	Compld (2 octets)	SubVersNr (2 octets)

Figure 2. CtrData field of the LL_VERSION_IND PDU





Vendor Prevalence

BLE - 2023-10-26

- In general though this seems to have a higher prevalence of silicon makers than other BT company/vendor ID fields, so a better S/N ratio for what I want to know
- I have only seen ~ 11466 / 92854 (12.3%) success in my LL2thprint log
 - Future work will take RSSI into account

+-----+ comp_by_CompId +-----+	count
Broadcom Corporation	8108
Apple, Inc.	1466
Nordic Semiconductor ASA	163
Qualcomm Technologies International, Ltd. (QTIL)	121
MediaTek, Inc.	119
Qualcomm	114
Cypress Semiconductor	85
Texas Instruments Inc.	83
Samsung Electronics Co. Ltd.	60
Dialog Semiconductor B.V.	41
RivieraWaves S.A.S	32
Telink Semiconductor Co. Ltd	29
Bestechnic(Shanghai),Ltd	28
Realtek Semiconductor Corporation	28
ST Microelectronics	26
Marvell Technology Group Ltd.	19
Casambi Technologies Oy	15
Zhuhai Jiel i technology Co.,Ltd	15
Airoha Technology Corp.	14
Silicon Laboratories	13
Ambiq	11
Atheros Communications, Inc.	10
Universal Electronics, Inc.	8
Shenzhen Goodix Technology Co., Ltd	6
Yichang Microelectronics Co., Ltd	5



Vendor Prevalence

BLE - 2023-10-26

- In general though this seems to have a higher prevalence of silicon makers than other BT company/vendor ID fields, so a better S/N ratio for what I want to know
- I have only seen ~ 11466 / 92854 (12.3%) success in my LL2thprint log
 - Future work will take RSSI into account

	count
Broadcom Corporation	8108
Apple, Inc.	1466
Nordic Semiconductor ASA	163
Qualcomm Technologies International, Ltd. (QTI)	121
MediaTek, Inc.	119
Qualcomm	114
Cypress Semiconductor	85
Texas Instruments Inc.	83
Samsung Electronics Co. Ltd.	60
Dialog Semiconductor B.V.	41
RivieraWaves S.A.S	32
Telink Semiconductor Co. Ltd	29
Bestechnic(Shanghai),Ltd	28
Realtek Semiconductor Corporation	28
ST Microelectronics	26
Marvell Technology Group Ltd.	19
Casambi Technologies Oy	15
Zhuhai Jiel i technology Co.,Ltd	15
Airoha Technology Corp.	14
Silicon Laboratories	13
Ambiq	11
Atheros Communications, Inc.	10
Universal Electronics, Inc.	8
Shenzhen Goodix Technology Co., Ltd	6
Yichip Microelectronics (Hangzhou) Co.,Ltd.	5
UNKNOWN_COMP_BY_BT_CID	5
beken	5
Actions (Zhuhai) Technology Co., Limited	5
PHYPLUS Inc	4
Intel Corp.	4
Toshiba Corp.	3
Shanghai wuqi microelectronics Co.,Ltd	2
Barrot Technology Co.,Ltd.	2
Hong Kong HunterSun Electronic Limited	2
WuXi Vimicro	2
NXP B.V.	2
Ingchips Technology Co., Ltd.	2
LAPIS Semiconductor Co.,Ltd	2
Bluetrum Technology Co.,Ltd	2
ON Semiconductor	1
MindTree Ltd.	1



Vendor Prevalence

BTC - 2023-10-26

- In general though this seems to have a higher prevalence of silicon makers than other BT company/vendor ID fields, so a better S/N ratio for what I want to know
- I have only seen ~ 7916 / 25125 (31.5%) success in my LMP 2thprint log
 - Future work will take RSSI into account

+-----+ comp_by_CompId +-----+	count
🍪 Qualcomm	35
🍪 MediaTek, Inc.	26
🍪 Broadcom Corporation	25
🍪 Intel Corp.	8
🍪 Qualcomm Technologies International, Ltd. (QTIL)	7
🍪 Zhuhai Jielì technology Co.,Ltd	7
🍪 Realtek Semiconductor Corporation	6
🍪 Cypress Semiconductor	3
🍪 Marvell Technology Group Ltd.	2
🍪 Samsung Electronics Co. Ltd.	2
🍪 RivieraWaves S.A.S	2
🍪 Bluegiga	1
✗ Equinix AG	1
✗ G-wearables inc.	1
✗ Anova Applied Electronics	1
✗ Toshiba Corp.	1
🍪 Texas Instruments Inc.	1
✗ Lumens For Less, Inc	1
✗ Nokia Mobile Phones	1
🍪 Actions (Zhuhai) Technology Co., Limited	1
✗ Shenzhen Feasycom Technology Co., Ltd.	1



Vendor Prevalence

BTC - 2023-10-26

- In general though this seems to have a higher prevalence of silicon makers than other BT company/vendor ID fields, so a better S/N ratio for what I want to know
- I have only seen ~ 7916 / 25125 (31.5%) success in my LMP 2thprint log
 - Future work will take RSSI into account

Bitflip!

	count
+-----+-----+	
comp_by_CompId	
+-----+-----+	
🍪 Qualcomm	35
🍪 MediaTek, Inc.	26
🍪 Broadcom Corporation	25
🍪 Intel Corp.	8
🍪 Qualcomm Technologies International, Ltd. (QTIL)	7
🍪 Zhuhai Jielì technology Co.,Ltd	7
🍪 Realtek Semiconductor Corporation	6
🍪 Cypress Semiconductor	3
🍪 Marvell Technology Group Ltd.	2
🍪 Samsung Electronics Co. Ltd.	2
🍪 RivieraWaves S.A.S	2
🍪 Bluegiga	1
✗ Equinix AG	1
✗ G-wearables inc.	1
✗ Anova Applied Electronics	1
✗ Toshiba Corp.	1
🍪 Texas Instruments Inc.	1
✗ Lumens For Less, Inc	1
✗ Nokia Mobile Phones	1
🍪 Actions (Zhuhai) Technology Co., Limited	1
✗ Shenzhen Feasycom Technology Co., Ltd.	1
+-----+-----+	



(Link Layer) Sub-Version Number

SubVersNr (2 bytes)

- The vendor gets to make up whatever value they want here!



2.4.2.13 LL_VERSION_IND

(Link Layer) SubVersion

The format of the CtrData field is shown in Figure 2.23.

CtrData		
VersNr (1 octet)	Compld (2 octets)	SubVersNr (2 octets)

- The vendor identifier

Figure 2.23: CtrData field of the LL_VERSION_IND PDU

The LL_VERSION_IND CtrData consists of three fields:

- VersNr field shall contain the version of the Bluetooth Controller specification (see Bluetooth [Assigned Numbers](#)).
- Compld field shall contain the company identifier of the manufacturer of the Bluetooth Controller (see Bluetooth [Assigned Numbers](#)).

SubVersNr field shall contain a unique value for each implementation or revision of an implementation of the Bluetooth Controller.



Does SubVersNr Imply OS/Firmware Version?

- **Hypothesis:** SubVersNr will increment per OS/firmware update, and thus can be used to infer the OS/firmware version
 - Conclusion:
 - **Rejected for Broadcom**



Does SubVersNr Imply OS/Firmware Version?

- From BlueZ 5.66 monitor/packet.c

```
    } broadcom_usb_subversion_table[] = {  
        { 0x210b, "BCM43142A0" }, /* 001.001.011 */  
        { 0x2112, "BCM4314A0" }, /* 001.001.018 */  
        { 0x2118, "BCM20702A0" }, /* 001.001.024 */  
        { 0x2126, "BCM4335A0" }, /* 001.001.038 */  
        { 0x220e, "BCM20702A1" }, /* 001.002.014 */  
        { 0x230f, "BCM4354A2" }, /* 001.003.015 */  
        { 0x4106, "BCM4335B0" }, /* 002.001.006 */  
        { 0x410e, "BCM20702B0" }, /* 002.001.014 */  
        { 0x6109, "BCM4335C0" }, /* 003.001.009 */  
        { 0x610c, "BCM4354" }, /* 003.001.012 */
```



Oh right...I remember something now...

- InternalBlue has firmware files per Broadcom chip, and they're ordered by numbers that look like those SubVersions!

The screenshot shows a GitHub repository interface. The URL in the address bar is github.com/seemoo-lab/internalblue/tree/master/internalblue/fw. The left sidebar shows navigation buttons for back, forward, and refresh, along with a lock icon. Below these are buttons for 'Files' (selected), 'master' (branch), and a search bar with 'Go to file' and a 't' icon. The main content area displays a list of files under the directory 'internalblue / internalblue / fw /'. The files listed are:

- fw_0x2209.py
- fw_0x220b.py
- fw_0x220c.py
- fw_0x220e.py



C

- ❑ fw_0x2033.py
- ❑ fw_0x203a.py
- ❑ fw_0x2056.py
- ❑ fw_0x21a9.py
-
- ❑ fw_0x21d0.py
- ❑ fw_0x2209.py
- ❑ fw_0x220b.py
- ❑ fw_0x220c.py
- ❑ fw_0x220e.py
- ❑ fw_0x2230.py
- ❑ fw_0x240f.py
- ❑ fw_0x3032.py

- ❑ fw_0x4196.py
- ❑ fw_0x4208.py
- ❑ fw_0x420e.py
- ❑ fw_0x420e_iphone.py
- ❑ fw_0x4228.py
- ❑ fw_0x422a.py
- ❑ fw_0x6103.py
- ❑ fw_0x6109.py
- ❑ fw_0x6119.py



C

```
} broadcom_usb_subversion_table[] = {
{ 0x210b, "BCM43142A0" }, /* 001.001.011 */
{ 0x2112, "BCM4314A0" }, /* 001.001.018 */
{ 0x2118, "BCM20702A0" }, /* 001.001.024 */
{ 0x2126, "BCM4335A0" }, /* 001.001.038 */
{ 0x220e, "BCM20702A1" }, /* 001.002.014 */
{ 0x230f, "BCM4354A2" }, /* 001.003.015 */
{ 0x4106, "BCM4335B0" }, /* 002.001.006 */
{ 0x410e, "BCM20702B0" }, /* 002.001.014 */
{ 0x6109, "BCM4335C0" }, /* 003.001.009 */
{ 0x610c, "BCM4354" }, /* 003.001.012 */
```

- fw_0x2033.py
- fw_0x203a.py
- fw_0x2056.py
- fw_0x21a9.py
-
- fw_0x21d0.py
- fw_0x2209.py
-
- fw_0x4196.py
- fw_0x4208.py
- fw_0x420e.py
- fw_0x420e_iphone.py
- fw_0x4228.py
- fw_0x422a.py
- fw_0x6103.py
- fw_0x6109.py**
- fw_0x6119.py



Broadcom

- So at least for Broadcom, the SubVersion is used to store a specific silicon chip (*& stepping revision*) information!
- ***The ideal ChipPrint!*** 😊
- (I would have preferred a VersionPrint, but oh well...)



Does SubVersNr Imply OS/Firmware Version?

- **Hypothesis:** SubVersNr will increment per OS/firmware update, and thus can be used to infer the OS/firmware version
 - Conclusion:
 - Weak reject for MediaTek, which seems to default to SubVersNr == 0, but where there are a range of versions seen, which probably then depends on the device maker's behavior
 - *Requires more investigation*



Does SubVersNr Imply OS/Firmware Version?

- **Hypothesis:** SubVersNr will increment per OS/firmware update, and thus can be used to infer the OS/firmware version
 - Overall Conclusion:
 - Requires more investigation!
 - Fundamentally requires hands on with a device + manually updating firmware and observing how the sub-version-number changes

2thprint by Link Layer Packet Combinations



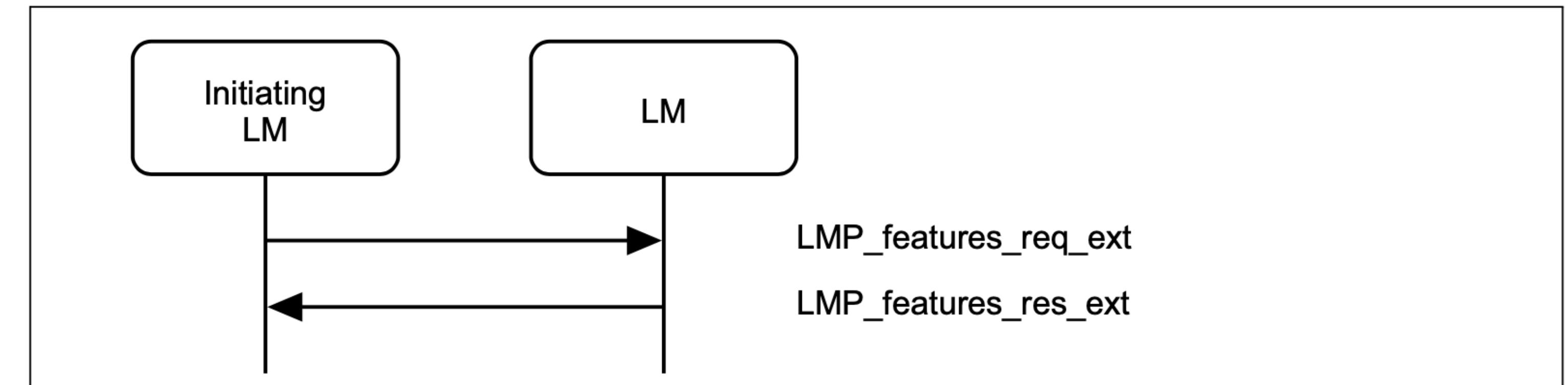


Version Information is not the only game in town

- If we want to be more nmap-OS-fingerprint-like, it makes sense to hit the target with multiple packet types, potentially in different orders, and see how it responds
- **LMP packet types:** LMP_features_req, LMP_features_req_ext, LMP_version_req, LMP_name_req, LMP_switch_req, LMP_ping_req, LMP_encryption_key_size_req, *malformed* LMP_features_req, *malformed* LMP_features_req_ext
- **BLE LL packet types:** LL_VERSION_IND, LL_LENGTH_REQ, LL_PING_REQ, LL_FEATURE_REQ



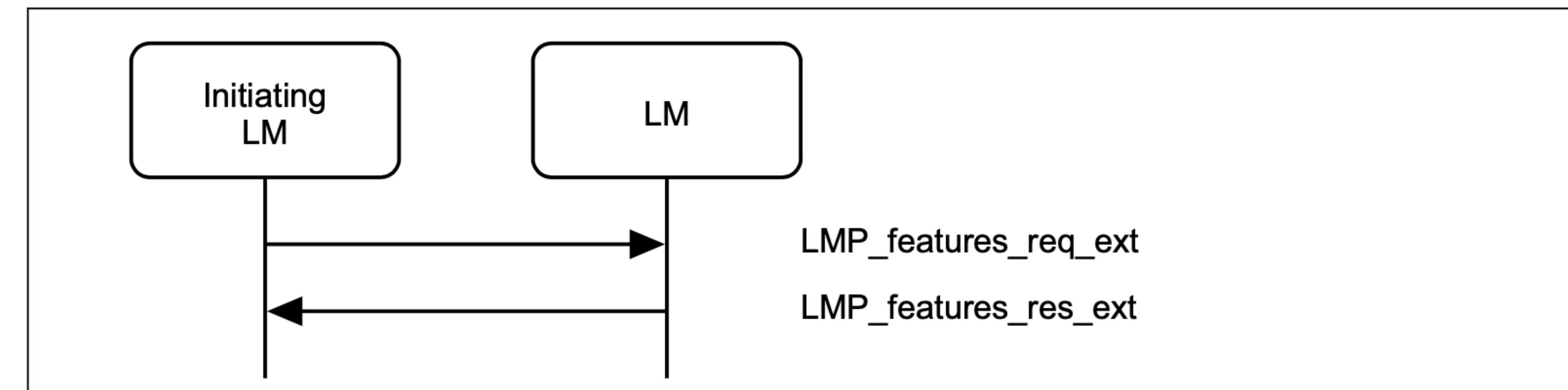
"malformed" LMP_features_ext_req?



Sequence 79: Request for extended features



"malformed" LMP_features_ext_req?



Sequence 79: Request for extended features

3.3 FEATURE MASK DEFINITION

The features are represented as a bit mask when they are transferred in LMP messages. For each feature a single bit is specified which shall be set to 1 if the feature is supported and set to 0 otherwise. The single exception is the flow control lag which is coded as a 3 bit field with the least significant bit in byte 2 bit 4 and the most significant bit in byte 2 bit 6. All removed, unknown, or unassigned feature bits shall be set to 0 and ignored upon receipt.

No.	Supported feature	Byte	Bit
0	3 slot packets	0	0
1	5 slot packets	0	1
2	Encryption	0	2



"malform"

!?

3	Slot offset	0	3
4	Timing accuracy	0	4
5	Role switch	0	5
6	Hold mode	0	6
7	Sniff mode	0	7
8	Park state	1	0
9	Power control requests	1	1
10	Channel quality driven data rate (CQDDR)	1	2
11	SCO link	1	3

...

No.	Supported feature	Byte	Bit
57	Inquiry TX Power Level	7	1
58	Enhanced Power Control	7	2
59	Reserved	7	3
60	Reserved	7	4
61	Reserved	7	5
62	Reserved	7	6
63	Extended features	7	7

Table 3.2: Feature mask definitions (page 0)



"malform"

)?

3	Slot offset	0	3
4	Timing accuracy	0	4
5	Role switch	0	5
6	Hold mode	0	6
7	Sniff mode	0	7
8	Park state	1	0
9	Power control requests	1	1
10	Channel quality driven data rate (CQDDR)	1	2
11	SCO link	1	3

...

No.	Supported feature	Byte	Bit
57	Inquiry TX Power Level	7	1
58	Enhanced Power Control	7	2
59	Reserved	7	3
60	Reserved	7	4
61	Reserved	7	5
62	Reserved	7	6
63	Extended features	7	7

Table 3.2: Feature mask definitions (page 0)



No.	Supported Feature	Byte	Bit
64	Secure Simple Pairing (Host Support)	0	0
65	LE Supported (Host)	0	1
66	Simultaneous LE and BR/EDR to Same Device Capable (Host)	0	2
67	Secure Connections (Host Support)	0	3

Table 3.3: Extended feature mask definition (page 1)

No.	Supported Feature	Byte	Bit
128	Connectionless Slave Broadcast – Master Operation	0	0
129	Connectionless Slave Broadcast – Slave Operation	0	1
130	Synchronization Train	0	2
131	Synchronization Scan	0	3
132	Inquiry Response Notification Event	0	4
133	Generalized interlaced scan	0	5
134	Coarse Clock Adjustment	0	6
135	Reserved	0	7
136	Secure Connections (Controller Support)	1	0
137	Ping	1	1
138	Reserved	1	2
139	Train nudging	1	3

Table 3.4: Extended feature mask definition (page 2)



No.	Supported Feature	Byte	Bit
64	Secure Simple Pairing (Host Support)	0	0
65	LE Supported (Host)	0	1
66	Simultaneous LE and BR/EDR to Same Device Capable (Host)	0	2
67	Secure Connections (Host Support)	0	3

Table 3.3: Extended feature mask definition (page 1)

No.	Supported Feature	Byte	Bit
128	Connectionless Slave Broadcast – Master Operation	0	0
129	Connectionless Slave Broadcast – Slave Operation	0	1
130	Synchronization Train	0	2
131	Synchronization Scan	0	3
132	Inquiry Response Notification Event	0	4
133	Generalized interlaced scan	0	5
134	Coarse Clock Adjustment	0	6
135	Reserved	0	7
136	Secure Connections (Controller Support)	1	0
137	Ping	1	1
138	Reserved	1	2
139	Train nudging	1	3

Table 3.4: Extended feature mask definition (page 2)



Pick a page, any page...

M/O	PDU	Contents
M	LMP_features_req	features
M	LMP_features_res	features
O(63)	LMP_features_req_ext	features page max supported page extended features
O(63)	LMP_features_res_ext	features page max supported page extended features

Table 4.27: PDUs used for features request

"The LMP_features_req_ext PDU contains a feature page index that specifies which page is requested and the contents of that page for the requesting device. Pages are numbered from 0-255 with page 0 corresponding to the normal features mask. "

Pick a page, any page



You should *never* request a feature page > their max supported page OR the max page in the version of the spec that they conform to.

E.g. Spec 4.2 only has pages 0, 1, 2.

So what happens if you request 3...or 255?

M/O	PDU	Contents
M	LMP_features_req	features
M	LMP_features_res	features
O(63)	LMP_features_req_ext	features page max supported page extended features
O(63)	LMP_features_res_ext	features page max supported page extended features

"The LMP_features_req_ext PDU contains a feature page index that specifies which page is requested and the contents of that page for the requesting device. Pages are numbered from 0-255 with page 0 corresponding to the normal features mask. "

Table 4.27: PDUs used for features request



How to Send Packets?

- Sweyntooth[1] (2020, BLE-only) & Braktooth[2] (2022, BTC-only)
 - Provide a way in Python and C respectively to create & send arbitrary link-layer packets in an arbitrary order

[1] <https://asset-group.github.io/disclosures/sweyntooth/>

[2] <https://asset-group.github.io/disclosures;braktooth/>



DATA ANALYSIS ETA WEN?

- Finding "weird packets/combinations" was the original idea for 2thprinting...and yet...I haven't actually analyzed this data yet _(`)_/. Why tho?
- 1) it *feels* like I haven't found the right balance yet between speed and useful signal
 - I'm prioritizing 2thprinting mechanisms that are as fast as possible, so they can be used against moving targets
 - Reinvocation of Braktooth/Sweyntooth adds 5+ seconds of overhead to every 2thprinting attempt
- 2) I want known *reference chips* to compare this data to before I start asserting "this packet combination result is indicative of vendor X"
 - Current research intern project

2thprint by Manufacturer-Specific Data



or





2thprint by Manufacturer-Specific Data (MSD): Company ID (CID)

Supplement to the Bluetooth Core Specification | v11, Part A

page 12

Data Types Specification



- BLE Advertisements and BTC Extended Inquiry Response packets can include MSD data where the manufacturer can *mostly* put whatever they want

1 DATA TYPES DEFINITIONS AND FORMATS

This part defines the basic data types used for Extended Inquiry Response (EIR), Advertising Data (AD), Scan Response Data (SRD), Additional Controller Advertising Data (ACAD), and OOB data blocks. Additional data types may be defined in profile specifications.

Each data type shall only be used in accordance with the requirements specified in [Table 1.1](#).

Data type	Context				
	EIR	AD	SRD	ACAD	OOB
Service UUID	O	O	O	O	O
Local Name	C.1	C.1	C.1	X	C.1
Flags	C.1	C.1	X	X	C.1
Manufacturer Specific Data	O	O	O	O	O



2thprint by Manufacturer-Specific Data (MSD): Company ID (CID)

Supplement to the Bluetooth Core Specification | v11, Part A

page 12

Data Types Specification



- BLE Advertisements and BTC Extended Inquiry Response packets can include MSD data where the manufacturer can *mostly* put whatever they want

1 DATA TYPES DEFINITIONS AND FORMATS

This part defines the basic data types used for Extended Inquiry Response (EIR), Advertising Data (AD), Scan Response Data (SRD), Additional Controller Advertising Data (ACAD), and OOB data blocks. Additional data types may be defined in profile specifications.

Each data type shall only be used in accordance with the requirements specified in [Table 1.1](#).

Data type	Context				
	EIR	AD	SRD	ACAD	OOB
Service UUID	O	O	O	O	O
Local Name	C.1	C.1	C.1	X	C.1
Flags	C.1	C.1	X	X	C.1
Manufacturer Specific Data	O	O	O	O	O



2thprint by Manufacturer-Specific Data (MSD): Company ID (CID)

- They're *supposed* to put their company's assigned number in the first 2 bytes, but not everyone does...
- Also, some put the company ID as little-endian, and some as big-endian, and some use both!

Supplement to the Bluetooth Core Specification | v11, Part A

page 16

Data Types Specification



1.4 MANUFACTURER SPECIFIC DATA

1.4.1 Description

The Manufacturer Specific data type is used for manufacturer specific data. The first two data octets shall contain a company identifier from [Assigned Numbers](#). The interpretation of any other octets within the data shall be defined by the manufacturer specified by the company identifier.

1.4.2 Format

Data Type	Description
«Manufacturer Specific Data» <i>uint16</i> , which may be followed by <i>struct</i>	The first value contains the Company Identifier Code. Any remainder contains manufacturer specific data.

Table 1.5: Manufacturer Specific data type



2thprint by Manufacturer-Specific Data (MSD): Company ID (CID)

- They're *supposed* to put their company's assigned number in the first 2 bytes, but not everyone does...
- Also, some put the company ID as little-endian, and some as big-endian, and some use both!

Supplement to the Bluetooth Core Specification | v11, Part A

page 16

Data Types Specification



1.4 MANUFACTURER SPECIFIC DATA

1.4.1 Description

The Manufacturer Specific data type is used for manufacturer specific data. The first two data octets shall contain a company identifier from [Assigned Numbers](#). The interpretation of any other octets within the data shall be defined by the manufacturer specified by the company identifier.

1.4.2 Format

Data Type	Description
«Manufacturer Specific Data» <i>uint16</i> , which may be followed by <i>struct</i>	The first value contains the Company Identifier Code. Any remainder contains manufacturer specific data.

Table 1.5: Manufacturer Specific data type



2thprint by Manufacturer-Specific Data (MSD): Company ID (CID)

Supplement to the Bluetooth Core Specification | v11, Part A

page 16

- They're *supposed* to put their company's assigned number in the first 2 bytes, but not everyone does...
- Also, some put the company ID as little-endian, and some as big-endian, and some use both!

Data Types Specification

1.4 MANUFACTURER SPECIFIC DATA

1.4.1 Description

The Manufacturer Specific data type is used for manufacturer specific data. The first two data octets shall contain a company identifier from [Assigned Numbers](#). The interpretation of any other octets within the data shall be specified by the manufacturer specified by the company identifier.

1.4.2 Format

It doesn't say what endianness it should use!

Data Type	Description
«Manufacturer Specific Data» <i>uint16</i> , which may be followed by <i>struct</i>	The first value contains the Company Identifier Code. Any remainder contains manufacturer specific data.

Table 1.5: Manufacturer Specific data type





Endianness info from BT Spec 4.0!

Lost in BT Spec 4.2 when they created the Core Specification Supplement doc?

8.1 EIR DATA TYPE DEFINITIONS

This section defines the basic EIR data types. Additional EIR data types may be defined in profile specifications.

All EIR data type values are listed in the Bluetooth [Assigned Numbers](#) document.

All numerical multi-byte entities and values associated with the following data types shall use little-endian byte order.



Top 20 Vendors for BTC MSD data

BTC - 2023-10-26

device_BT_CID	company_name	frequency	
0x8700	Garmin International (wrong-endian)	5942	✗
0x4C00	Apple, Inc. (wrong-endian)	2487	🍪
0x1D	Qualcomm	2437	🍪
0xFF19	Samsung Electronics Co. Ltd. (just wacky)	1938	🍪
0xF	Broadcom Corporation	950	🍪
0x75	Samsung Electronics Co. Ltd.	879	🍪
0x7500	Samsung Electronics Co.,Ltd (wrong-endian)	278	🍪
0xD906	Shanghai Mountain View Silicon Co.,Ltd. (wrong-endian)	274	🍪
0x3E0	Actions (Zhuhai) Technology Co., Limited	97	🍪
0x4C	Apple, Inc.	45	🍪
0x27D	HUAWEI Technologies Co., Ltd.	43	✗
0xA	Qualcomm Technologies International, Ltd. (QTIL)	32	🍪
0xA02	Ayxon-Dynamics GmbH	14	✗
0x200	Verifone Systems Pte Ltd. Taiwan Branch	7	✗
0x0	Ericsson AB	7	✗
0xA00	Ampler Bikes OU (wrong-endian Qualcomm?)	6	✗
0x5F0	beken	6	🍪
0x850	Yealink (Xiamen) Network Technology Co.,LTD	5	✗
0x18C	Wilo SE	3	✗
0x67	GN Audio A/S	3	✗
...			



Top 20 Vendors for BLE MSD data

BLE - 2023-10-26

device_BT_CID	company_name	frequency
0x4C	Apple, Inc.	9503999
0x6	Microsoft	94967
0x75	Samsung Electronics Co. Ltd.	83182
0x11B	Hewlett Packard Enterprise	18899
0x183	Walt Disney	14073
0x3	IBM Corp.	11937
0x87	Garmin International, Inc.	10150
0x131	Cypress Semiconductor	7872
0x0	Ericsson AB	7547
0x171	Amazon.com Services LLC	6926
0x57	Harman International Industries, Inc.	6424
0x87F	Phillips Connect Technologies LLC	5735
0x12D	Sony Corporation	5265
0xE0	Google	4370
0x310	SGL Italia S.r.l.	3730
0xB01	RESIDEO TECHNOLOGIES, INC.	3608
0xA01	Cleveron AS	3139
0x157	Anhui Huami Information Technology Co., Ltd.	2657
0x65	HP, Inc.	2412
0x4C00	Apple, Inc. (wrong-endian)	1946
...		

2thprint by GATT

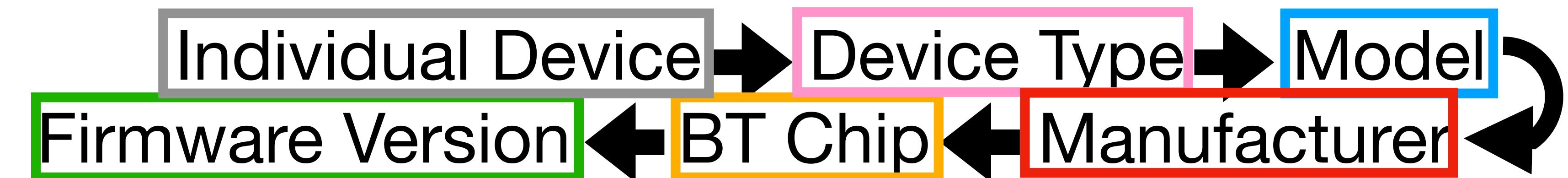




2thprint by GATT

GATTPrint

- Generic **Attribute Profile** (GATT) is a sort of weird thing that's built on top of Attribute *Protocol* (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information through GATT!

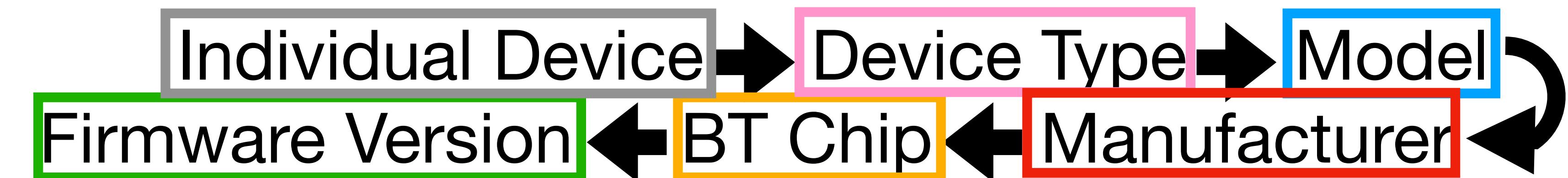




2thprint by GATT

GATTPrint

- Generic **Attribute Profile** (GATT) is a sort of weird thing that's built on top of Attribute *Protocol* (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information through GATT!

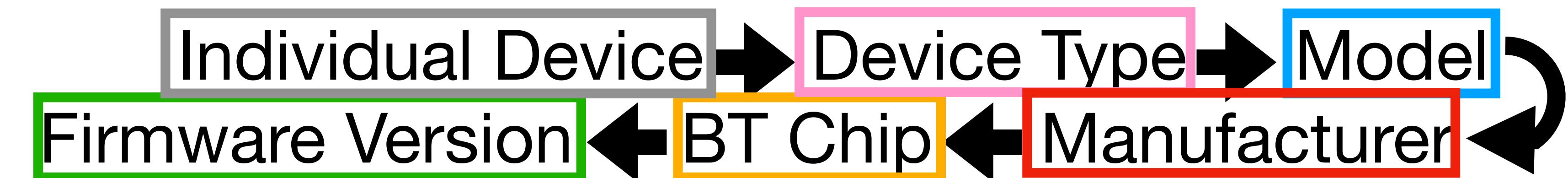




2thprint by GATT

GATTPrint

- Generic **Attribute Profile** (GATT) is a sort of weird thing that's built on top of Attribute *Protocol* (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information through GATT!





GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

The screenshot shows the BluefruitConnect app interface on a mobile device. At the top, there are two tabs: 'Disconnect' and 'Modules'. The 'Modules' tab is selected, showing a list of connected devices. The first device listed is '047_INSTACART1' with a signal strength of '-92 dBm'. Below this, there are two sections: 'MODULES' and 'Info'. The 'Info' section is currently active, displaying detailed device information. The information includes:

Model Number	ZD621
Serial Number	D9N224204243
Firmware Revision	v93.21.17Z
Hardware Revision	ZD6A042-D01L01EZ
Software Revision	6.7
Manufacturer Name	Zebra Technologies
PnP ID	38EB4A80-C570-11E3-9507-0002A5D5C51B
Characteristic	38EB4A81-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0
Characteristic	38EB4A82-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0



GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

The image shows two screenshots of the BluefruitConnect app interface on an iPhone. The top screenshot shows the 'Modules' screen for a device named '047_INSTACART1' with a signal strength of -92 dBm. The bottom screenshot shows the 'Info' screen with detailed device information.

Device Information (Bottom Screenshot):

Model Number	ZD621
Serial Number	D9N224204243
Firmware Revision	v93.21.17Z
Hardware Revision	ZD6A042-D01L01EZ
Software Revision	6.7
Manufacturer Name	Zebra Technologies
PnP ID	38EB4A80-C570-11E3-9507-0002A5D5C51B
Characteristic	38EB4A81-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0
Characteristic	38EB4A82-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0



GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

The image shows two screenshots of the BluefruitConnect app interface on an iPhone. The top screenshot shows the 'Modules' screen for a device named '047_INSTACART1' with a signal strength of -92 dBm. The bottom screenshot shows the 'Info' screen with detailed device information.

Device Information (Bottom Screenshot):

Model Number	ZD621
Serial Number	D9N224204243
Firmware Revision	v93.21.17Z
Hardware Revision	ZD6A042-D01L01EZ
Software Revision	6.7
Manufacturer Name	Zebra Technologies
PnP ID	38EB4A80-C570-11E3-9507-0002A5D5C51B
Characteristic	38EB4A81-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0
Characteristic	38EB4A82-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0



GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

The image shows two screenshots of the BluefruitConnect app on an iPhone. The top screenshot shows the 'Modules' screen for a device named '047_INSTACART1' with a signal strength of -92 dBm. The bottom screenshot shows the 'Info' screen with various device details.

Category	Value
Model Number	ZD621
Serial Number	D9N224204243
Firmware Revision	v93.21.17Z
Hardware Revision	ZD6A042-D01L01EZ
Software Revision	6.7
Manufacturer Name	Zebra Technologies
PnP ID	38EB4A80-C570-11E3-9507-0002A5D5C51B
Characteristic	38EB4A81-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0
Characteristic	38EB4A82-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0

Diagram Labels:

- Individual Device (highlighted in gray)
- Device Type
- Model (highlighted in blue)
- Firmware Version
- BT Chip
- Manufacturer

The diagram illustrates the flow of GATT information from the individual device to its type, model, and manufacturer, with the BT chip being the source of the firmware version and manufacturer information.



GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

The image shows two screenshots of the BluefruitConnect app on an iPhone. The top screenshot shows the 'Modules' screen for a device named '047_INSTACART1' with a signal strength of -92 dBm. The bottom screenshot shows the 'Info' screen with various device details.

Category	Information
Model Number	ZD621
Serial Number	D9N224204243
Firmware Revision	v93.21.17Z
Hardware Revision	ZD6A042-D01L01EZ
Software Revision	6.7
Manufacturer Name	Zebra Technologies
PnP ID	38EB4A80-C570-11E3-9507-0002A5D5C51B
Characteristic	38EB4A81-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0
Characteristic	38EB4A82-C570-11E3-9507-0002A5D5C51B
Client Characteristic Configuration	0

Diagram illustrating GATT data flow:

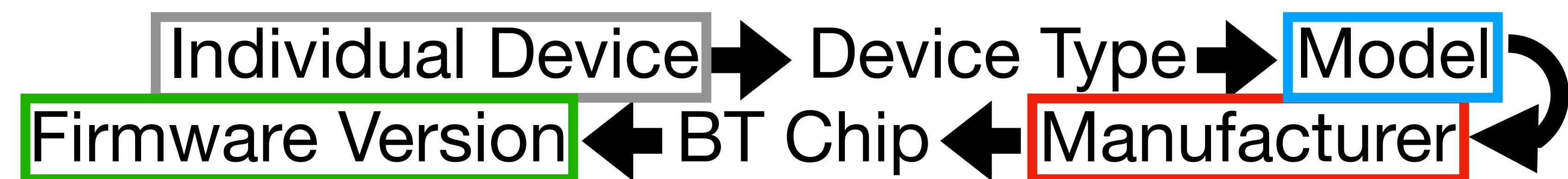
```
graph LR; A[Individual Device] --> B[Device Type]; B --> C[Model]; C --> D[Manufacturer]; D <--> E[BT Chip]; E <--> F[Firmware Version]
```

The diagram shows the flow of GATT data from an individual device through its type and model to its manufacturer. The manufacturer information is highlighted with a red box, and the BT chip and firmware version are also highlighted with red boxes.



GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect



The screenshot shows the "Modules" screen of the BluefruitConnect app. At the top, it displays the device name "047_INSTACART1" and signal strength "-92 dBm". Below this is a "MODULES" section with a single entry labeled "Info". To the right, under "DEVICE INFORMATION", are several fields: "Model Number" (ZD621) is highlighted with a blue border; "Serial Number" (D9N224204243) is in a grey box; "Firmware Revision" (v93.21.17Z) is highlighted with a green border; "Hardware Revision" (ZD6A042-D01L01EZ); "Software Revision" (6.7) is highlighted with a green border; and "Manufacturer Name" (Zebra Technologies) is highlighted with a red border. At the bottom, there are two "Characteristic" sections with IDs 38EB4A80-C570-11E3-9507-0002A5D5C51B and 38EB4A81-C570-11E3-9507-0002A5D5C51B, each with a "Client Characteristic Configuration" value of 0.

This part of the screenshot shows two "Characteristic" entries. The first is for UUID 38EB4A80-C570-11E3-9507-0002A5D5C51B, with a "Client Characteristic Configuration" value of 0. The second is for UUID 38EB4A82-C570-11E3-9507-0002A5D5C51B, also with a "Client Characteristic Configuration" value of 0.



GATT on Phones

- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

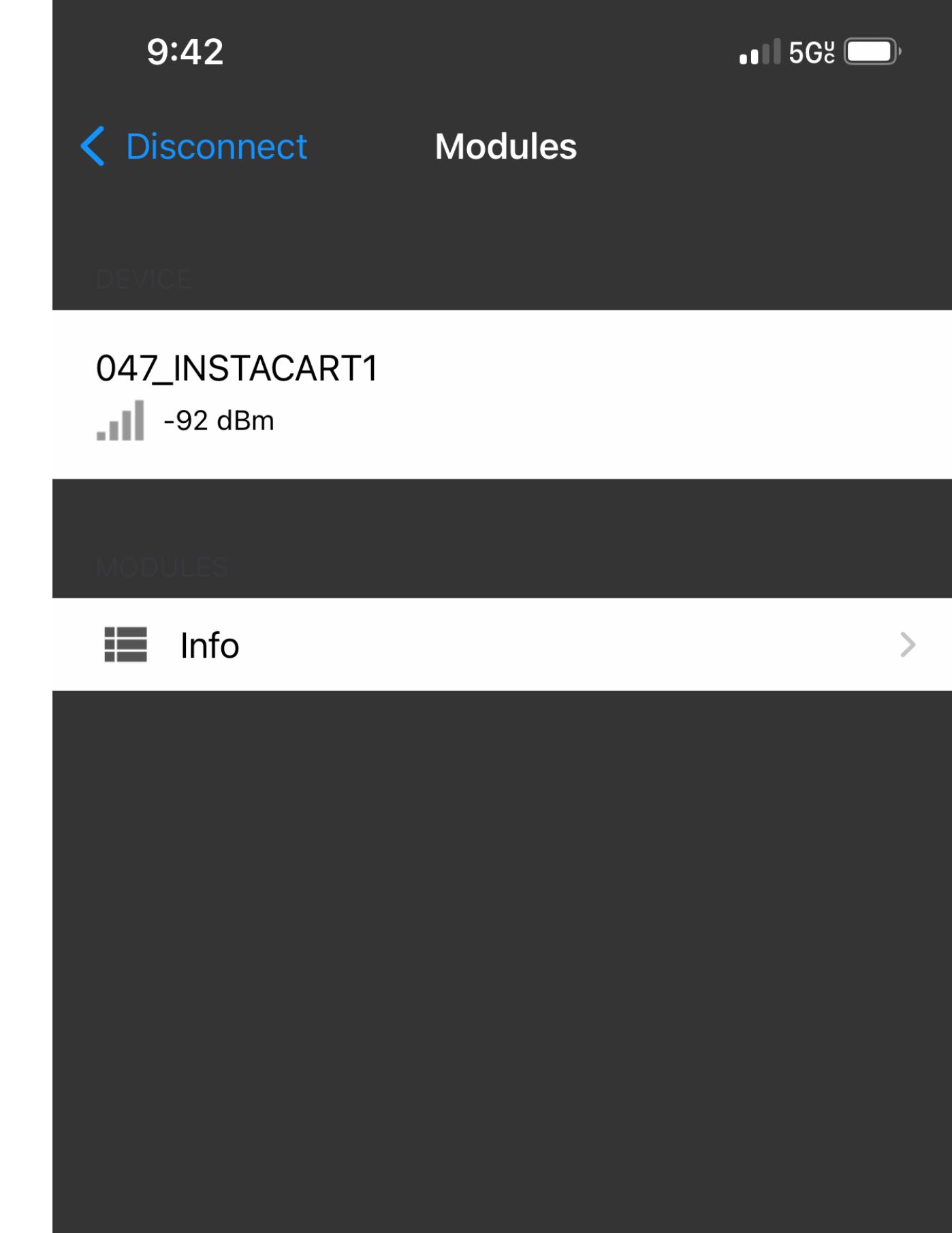
The screenshot shows the BluefruitConnect app interface on a mobile device. At the top, it displays the time as 9:42 and signal strength as 5G. The main screen shows a device named "047_INSTACART1" with a signal level of -92 dBm. Below the device name is a "MODULES" section, which is currently expanded to show an "Info" tab. The "Info" tab displays various device details. Several fields are highlighted with colored boxes: "Model Number" (blue box) is ZD621; "Firmware Revision" (green box) is v93.21.17Z; "Software Revision" (green box) is 6.7; and "Manufacturer Name" (red box) is Zebra Technologies. The "Characteristics" section at the bottom lists three entries, each with a "Characteristic" label and a "Client Characteristic Configuration" field set to 0.

Characteristic	Client Characteristic Configuration
38EB4A80-C570-11E3-9507-0002A5D5C51B	0
38EB4A81-C570-11E3-9507-0002A5D5C51B	0
38EB4A82-C570-11E3-9507-0002A5D5C51B	0



GATT on Phones

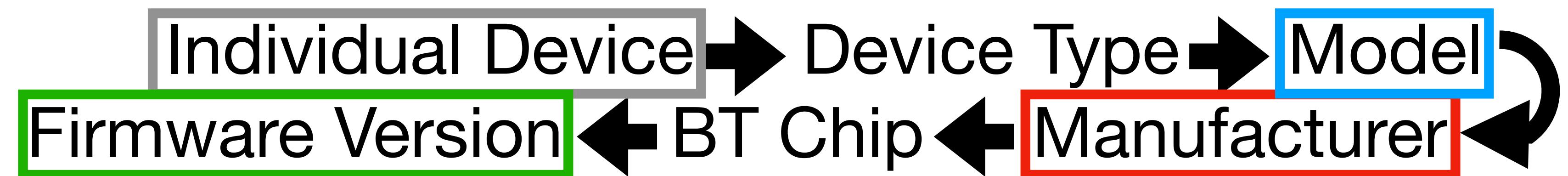
- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect



ELEVATING A PROVEN WINNER

Extending the G-Series Legacy

Zebra's GX Series printers are known for quality and premium performance. As you select your next printer, you can be confident that the next-generation ZD621 includes everything you loved in those legacy printers, and builds on this heritage to deliver best-in-class features for this new era of intelligence and forward adaptability.



EXPECT THE BEST

Premier Printing Performance

Rely on the ZD621 to help you power through – day after day. From outstanding print quality to portability for application flexibility to emerging technology to field-installable options, the ZD621 st

Chat with us



GATT in General

- On the *OTHER* hand...we may instead get a whole bunch of *nothing useful* from GATT
- Devices may not respond to GATT requests, and when they do, we're in no way guaranteed to get "characteristics" which contain the kind of information we want
 - GATT inquiries can take a few seconds. For moving targets, there is a low probability of success to perform a full information collection without a strong transmitter and good antenna



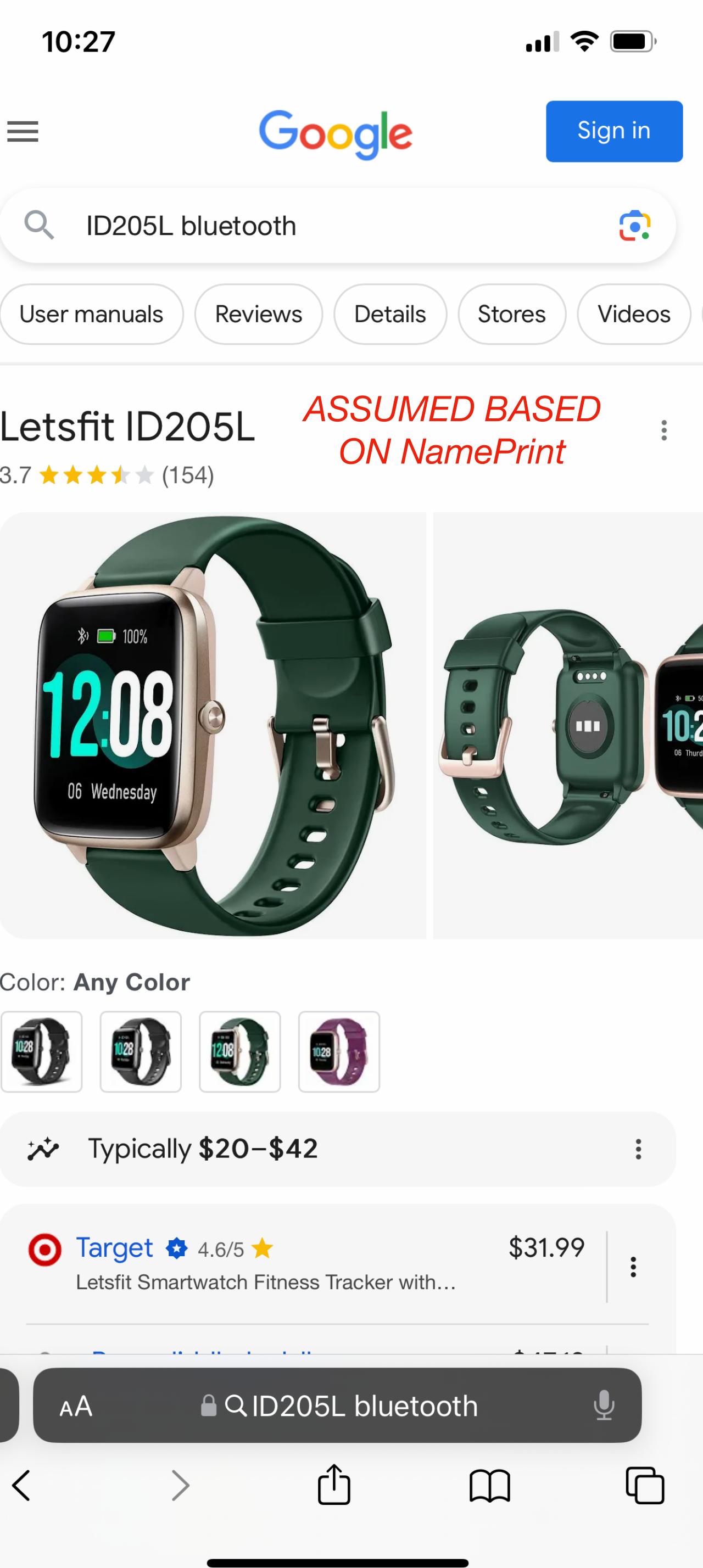
Characteristic	Value
0AF6	0AF0
0AF7	0U
Client Characteristic Configuration 0	0AF2 09 07 00
Client Characteristic Configuration 0	0AF1 09 07





GATT in General

- On the *OTHER* hand...we may instead get a whole bunch of *nothing useful* from GATT
- Devices may not respond to GATT requests, and when they do, we're in no way guaranteed to get "characteristics" which contain the kind of information we want
- GATT inquiries can take a few seconds. For moving targets, there is a low probability of success to perform a full information collection without a strong transmitter and good antenna





Prior Work

"Fingerprinting Bluetooth-Low-Energy Devices Based on the Generic Attribute Profile"

- [1] by Celosia & Cunche from 2019 connected 🚶 to BLE devices via GATT and just called *all the returned information* the fingerprint
 - Noteworthy, in my mind, for the different threat model (user privacy), that called out basically everything *except* firmware revision string as interesting

We identified that **information** found in GATT profiles can be used to infer the **following information**: **device type**, **device model**, **device manufacturer** and **user's name**. All this **information** can threaten the **privacy** of the device owner.



Prior Work

"Fingerprinting Bluetooth-Low-Energy De

You know what's really a
threat to user privacy?
Getting pwned over-the-air ;)



- [1] by Celosia & Cunche from 2019 connected 🚶 to BLE devices via GATT and just called *all the returned information* the fingerprint
- Noteworthy, in my mind, for the different threat model (user privacy), that called out basically everything *except* firmware revision string as interesting

We identified that **information** found in GATT profiles can be used to infer the **following information**: **device type**, **device model**, **device manufacturer** and **user's name**. All this **information** can threaten the **privacy** of the device owner.

Table 4: Average time to collect a GATT profile among different devices.

Device type	Device	Time (sec)
Lightbulb	Osram Smart+	6.531
Motion sensor	Eve Motion	6.468
Socket outlet	Eve Energy	5.919
Smartphone	Apple iPhone 8	4.354
Smartphone	Apple iPhone 6	4.259
Keyring	Nut	4.148
TV dongle	Google Chromecast	3.660
Fitness wristband	Fitbit Inspire	3.231
Presentation remote	Logitech Spotlight	2.860
Smartwatch	Apple Watch Series 3	2.853
Heart rate monitor	Polar H7	2.751
Fitness wristband	Fitbit Flex	2.552
Headset	Bose SoundLink Around-Ear II	2.181
Speaker	Divacore Ktulu2+	1.742
Keyring	Chipolo	1.426
	Average	3.662

**Table 4: Average time to collect a GATT profile among different devices.**

Device type	Device	Time (sec)
Lightbulb	Osram Smart+	6.531
Motion sensor	Eve Motion	6.468
Socket outlet	Eve Energy	5.919
Smartphone	Apple iPhone 8	4.354
Smartphone	Apple iPhone 6	4.259
Keyring	Nut	4.148
TV dongle	Google Chromecast	3.660
Fitness wristband	Fitbit Inspire	3.231
Presentation remote	Logitech Spotlight	2.860
Smartwatch	Apple Watch Series 3	2.853
Heart rate monitor	Polar H7	2.751
Fitness wristband	Fitbit Flex	2.552
	Sound-Ear II	2.181
		1.742
		1.426
		3.662



I've seen things take up to 24 seconds to reply to GATTprinting...



Table 4: Average time to collect a GATT profile among different devices.

Device type	Device	Time (sec)
Lightbulb	Osram Smart+	6.531
Motion sensor	Eve Motion	6.468
Socket outlet	Eve Energy	5.919
Smartphone	Apple iPhone 8	4.354
Smartphone	Apple iPhone 6	4.259
Keyring	Nut	4.148
TV dongle	Google Chromecast	3.660
Fitness wristband	Fitbit Inspire	3.231
Presentation remote	Logitech Spotlight	2.860
Smartwatch	Apple Watch Series 3	2.853
Heart rate monitor	Polar H7	2.751
Fitness wristband	Fitbit Flex	2.552
	Sound-Ear II	2.181
		1.742
		1.426
		3.662



I've seen things take up to 24 seconds to reply to GATTprinting...



	All	
	%	#
Device Name	99.49	13182
Appearance	99.48	13082
Service Changed	0.02	2
Manufacturer Name String	99.48	9177
Model Number String	99.36	9158
Battery Level	2.45	191
Current Time	0.41	31
Peripheral Preferred Connection Parameters	99.90	1051
Software Revision String	97.04	1017
Hardware Revision String	95.95	996
Serial Number String	97.12	979
Firmware Revision String	94.99	835



	All	
	%	#
Device Name	99.49	13182
Appearance	99.48	13082
Service Changed	0.02	2
Manufacturer Name String	99.48	9177
Model Number String	99.36	9158
Battery Level	2.45	191
Current Time	0.41	31
Peripheral Preferred Connection Parameters	99.90	1051
Software Revision String	97.04	1017
Hardware Revision String	95.95	996
Serial Number String	97.12	979
Firmware Revision String	94.99	835

At least 13182 devices reported a Name (and it was readable 99.49% of the time)



	All	
	%	#
Device Name	99.49	13182
Appearance	99.48	13082
Service Changed	0.02	2
Manufacturer Name String	99.48	9177
Model Number String	99.36	9158
Battery Level	2.45	191
Current Time	0.41	31
Peripheral Preferred Connection Parameters	99.90	1051
Software Revision String	97.04	1017
Hardware Revision String	95.95	996
Serial Number String	97.12	979
Firmware Revision String	94.99	835

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

Only 835 devices reported a Firmware Revision String!
(and it was readable 94.49% of the time)

	All	
	%	#
Device Name	99.49	13182
Appearance	99.48	13082
Service Changed	0.02	2
Manufacturer Name String	99.48	9177
Model Number String	99.36	9158
Battery Level	2.45	191
Current Time	0.41	31
Peripheral Preferred Connection Parameters	99.90	1051
Software Revision String	97.04	1017
Hardware Revision String	95.95	996
Serial Number String	97.12	979
Firmware Revision String	94.99	835

Note: Their dataset is heavily skewed by containing at least 9924 iPhones, and iPhones don't report a Firmware Revision String

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

Only 835 devices reported a Firmware Revision String!
(and it was readable 94.49% of the time)

	All	
	%	#
Device Name	99.49	13182
Appearance	99.48	13082
Service Changed	0.02	2
Manufacturer Name String	99.48	9177
Model Number String	99.36	9158
Battery Level	2.45	191
Current Time	0.41	31
Peripheral Preferred Connection Parameters	99.90	1051
Software Revision String	97.04	1017
Hardware Revision String	95.95	996
Serial Number String	97.12	979
Firmware Revision String	94.99	835

Note: Their dataset is heavily skewed by containing at least 9924 iPhones, and iPhones don't report a Firmware Revision String

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

Only 835 devices reported a Firmware Revision String!
(and it was readable 94.49% of the time)

	All	
	%	#
Device Name	99.49	13182
Appearance	99.48	13082
Service Changed	0.02	2
Manufacturer Name String	99.48	9177
Model Number String	99.36	9158
Battery Level	2.45	191
Current Time	0.41	31
Peripheral Preferred Connection Parameters	99.90	1051
Software Revision String	97.04	1017
Hardware Revision String	95.95	996
Serial Number String	97.12	979
Firmware Revision String	94.99	835

Note: Their dataset is heavily skewed by containing at least 9924 iPhones, and iPhones don't report a Firmware Revision String

At least 13182 devices reported a Name (and it was readable 99.49% of the time)

Only 835 devices reported a Firmware Revision String!
(and it was readable 94.49% of the time)



Prior Work

"Fingerprinting Bluetooth-Low-Energy Devices Based on the Generic Attribute Profile"

- [1] by Zuo et al. from 2019 scanned 🚶 BLE devices via Generic Attribute Profile (GATT) and just called all the top level UUID128s the fingerprint
 - Here they were referring to more like a "device-model" fingerprint, not "individual-device" fingerprint
 - Most interestingly, they scraped a bunch of Android applications to attempt to extract GATT-related UUID128s via static analysis



My GATTPrint-er

- The BlueZ (*deprecated*) "gatttool" seemed to already do a good job of collecting this information. But the output wasn't easily machine-parsable. So I modified the source to store info to a more easily machine-parsable log file
 - Pro tip: If you try to use the higher-layer BT APIs (such as those available via most Python->BT libraries), you will not be able to collect this info without first pairing. But pairing isn't actually necessary for most characteristics of most devices!
 - Some devices though may refuse read/write requests on access control grounds, due to lack of the encryption/authentication that comes as a result of pairing



Vendor-specific 128-bit UUIDs

🍪 Silicon-specific examples: Texas Instruments

- f000ffc**0**-0451-4000-b000-000000000000 - **OTA Firmware update GATT Service**
 - f000ffc**1**-0451-4000-b000-000000000000, f000ffc**2**-0451-4000-b000-000000000000, f000ffc**3**-0451-4000-b000-000000000000, f000ffc**4**-0451-4000-b000-000000000000 - Associated GATT Characteristics
- BleedingBit exploited an Aruba-customized TI OTA firmware update service (that just added some security-by-obscenity magic unlock code)



Vendor-specific 128-bit UUIDs

🍪 Silicon-specific examples: Nordic

- 6e400001-b5a3-f393-e0a9-e50e24dcca9e - **UART** GATT Service (advertised in SCAN_RSP as well as GATT)
 - 6e400002-b5a3-f393-e0a9-e50e24dcca9e - UART RX GATT Characteristic
 - 6e400003-b5a3-f393-e0a9-e50e24dcca9e - UART TX GATT Characteristic
- 00001530-1212-efde-1523-785feabcd123 - **"Legacy" (Insecure) Device Firmware Update (DFU)** GATT Service
 - 00001531-1212-efde-1523-785feabcd123, 00001532-1212-efde-1523-785feabcd123, 0000153-1212-efde-1523-785feabcd123 - Associated GATT Characteristics
- 0000fe59-0000-1000-8000-00805f9b34fb - **Secure Device Firmware Update (DFU)** GATT Service
 - 8EC90001-F315-4F60-9FB8-838830DAEA50, 8EC90002-F315-4F60-9FB8-838830DAEA50 - Associated GATT Characteristics

https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/libraries/bluetooth_services/services/nus.html

<https://nordicsemiconductor.github.io/IOSnRFConnect/assets/files/UserManual.pdf>

https://github.com/adafruit/Bluefruit_LE_Connect_Android/blob/master/app/src/main/java/com/adafruit/bluefruit/le/connect/ble/KnownUUIDs.java



Vendor-specific 128-bit UUIDs

🍪 Silicon-specific examples: Cambridge Silicon Radio (bought by Qualcomm)

- 00001100-d102-11e1-9b23-00025b00a5a5 - **GAIA (Over The Air update protocol)** GATT service
 - 00001101-d102-11e1-9b23-00025b00a5a5, 00001102-d102-11e1-9b23-00025b00a5a5, 00001103-d102-11e1-9b23-00025b00a5a5 - Associated GATT characteristics



Seeing which devices use a chip-maker

Based on UUID128Print

- Detective Work 🕵️: GATT -> Google Search "00001100-d102-11e1-9b23-00025b00a5a5" -> Cambridge Silicon Radio (CSR) CSR102x chips' over the air update protocol!

```
./TellMeEverything.py --UUID128regex "00001100-d102-11e1-9b23-00025b00a5a5" | grep DeviceName | sort | uniq -c
 2 DeviceName: BW-FYE13
 2 DeviceName: Crusher ANC
 3 DeviceName: Crusher Evo
 2 DeviceName: HK385
 1 DeviceName: Ink'd+ Active
 7 DeviceName: Jabra Elite 3
 4 DeviceName: Jabra Elite 4 Active
 1 DeviceName: LE-Device
 1 DeviceName: LE-Headset
 1 DeviceName: LE-MINOR III
 16 DeviceName: LE-OpenRun Pro
 1 DeviceName: LE_ATH-TWX9
 7 DeviceName: LG-TONE-FP9_LE
 2 DeviceName: M50
 3 DeviceName: Shure AONIC TW2
 1 DeviceName: Tribit FlyBuds C2
 1 DeviceName: [AV] MX-T40
 2 DeviceName: al hydrajolt
 2 DeviceName: mini lifejacket
 2 DeviceName: mini lifejacket jolt
```



Vendor-specific 128-bit UUIDs

🍪 Silicon-specific examples: Silicon Labs

- 331a36f5-2459-45ea-9d95-6142f0c4b307 - **BGX Xpress Streaming (arbitrary data)** GATT Service
 - a9da6040-0823-4995-94ec-9ce41ca28833, a73e9a10-628f-4494-a099-12efaf72258f, 75a9f022-af03-4e41-b4bc-9de90a47d50b - Associated GATT characteristics



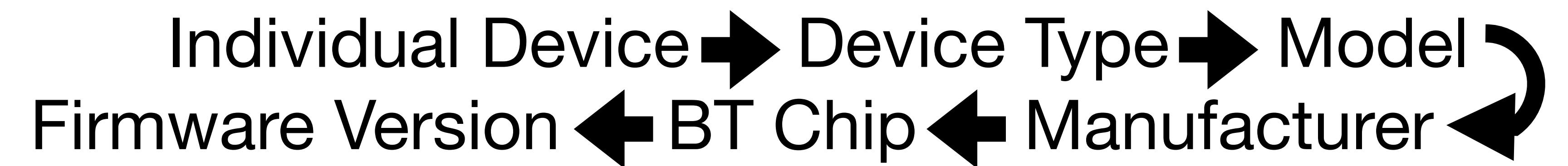
Vendor-specific 128-bit UUIDs

 Module-specific examples: [Laird](#)

- 569a1101-b87f-490c-92cb-11ba5ea5167c - **Virtual Serial Port Service** (GATT & ADV_IND)
 - 569a2000-b87f-490c-92cb-11ba5ea5167c - TX GATT Characteristic
 - 569a2001-b87f-490c-92cb-11ba5ea5167c - RX GATT Characteristic

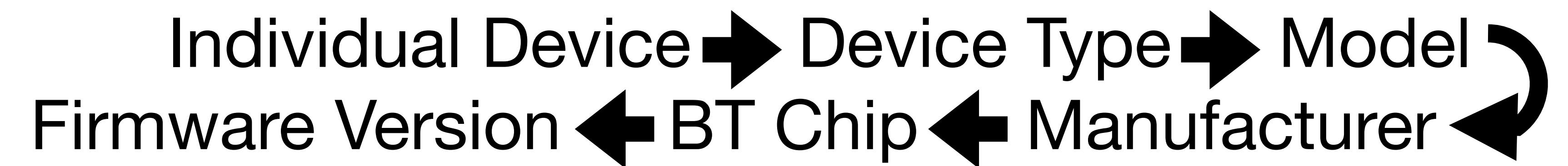


What I Want





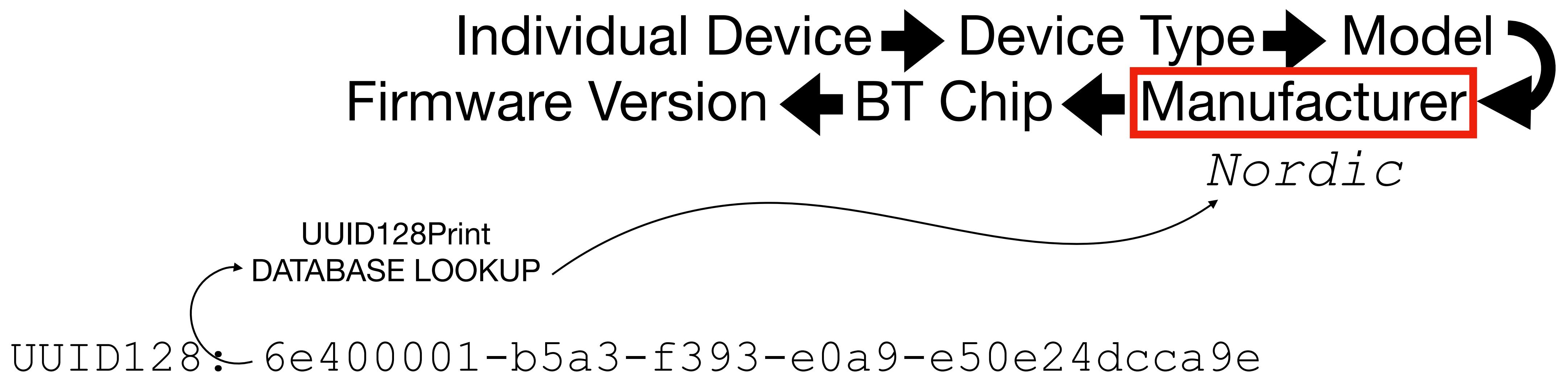
What I Want



UUID128: 6e400001-b5a3-f393-e0a9-e50e24dcca9e

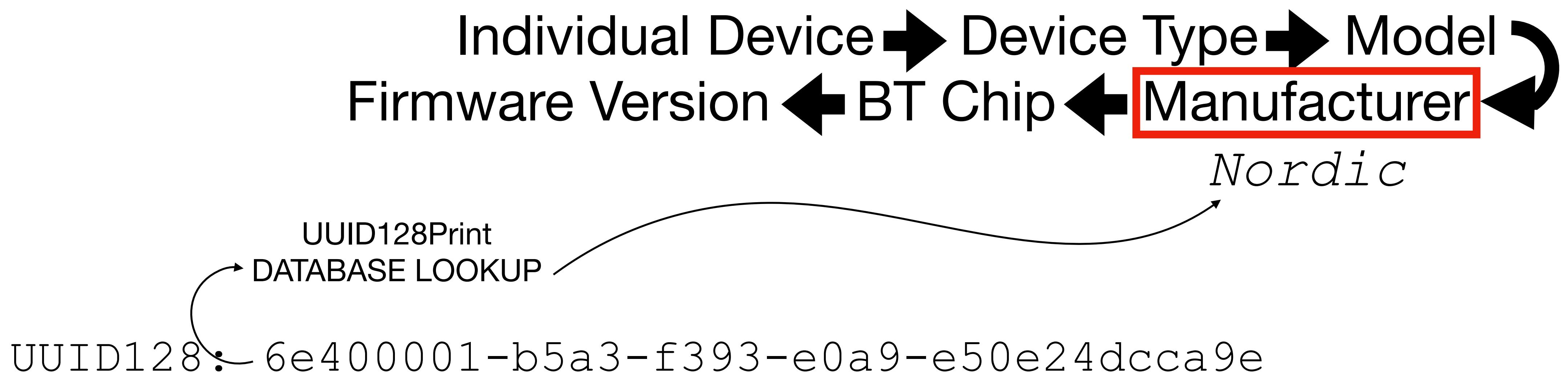


What I Want





What I Want

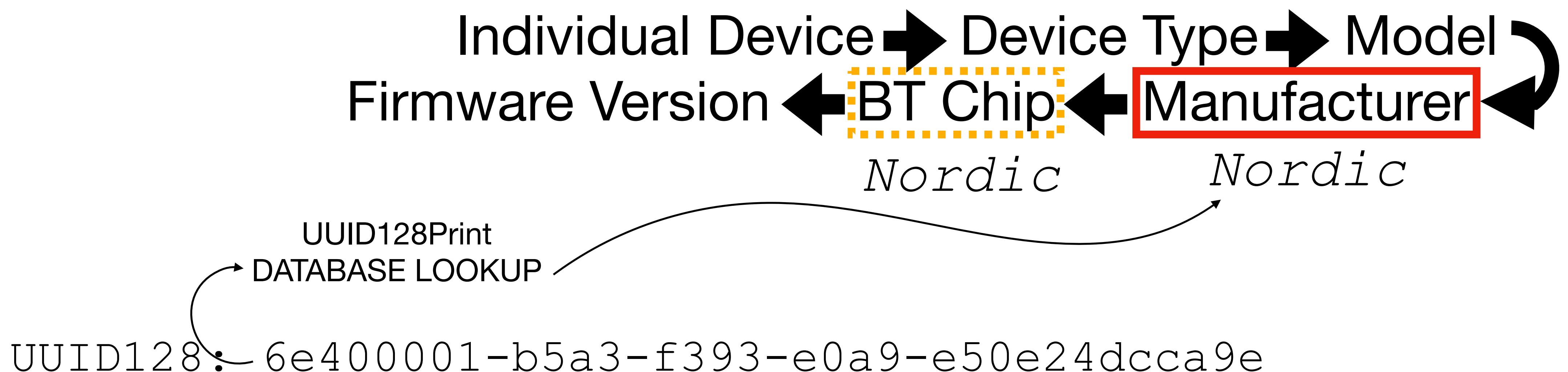


ASSUMPTION:

UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers



What I Want



ASSUMPTION:

UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers



Top 20 GATT Services in my data FWIW

2023-10-26

UUID128	Count	
00001800-0000-1000-8000-00805f9b34fb	21127	-> "Generic Access" (GAP)
00001801-0000-1000-8000-00805f9b34fb	20675	-> Generic Attribute (GATT)
0000180a-0000-1000-8000-00805f9b34fb	18044	-> Device Information
9fa480e0-4967-4542-9390-d343dc5d04ae	15738	-> Apple Nearby
d0611e78-bbb4-4591-a5f8-487910ae4366	15737	-> Apple Continuity
0000180f-0000-1000-8000-00805f9b34fb	11284	-> Battery
00001805-0000-1000-8000-00805f9b34fb	10641	-> Current Time
89d3502b-0f36-433a-8ef4-c502ad55f8dc	10636	-> Apple Media Service
7905f431-b5ce-4e99-a40f-4b1e122d00d0	10636	-> Apple Notification Center Service
0000febe-0000-1000-8000-00805f9b34fb	843	-> Bose
0000fe03-0000-1000-8000-00805f9b34fb	598	-> Amazon Alexa
0000fe2c-0000-1000-8000-00805f9b34fb	421	-> Google Fast Pair
9aa4730f-b25c-4cc3-b821-c931559fc196	359	-> Apple Watch? (my data seems to support)
eedd5e73-6aa8-4673-8219-398a489da87c	280	-> Samsung SmartTag Authentication Service
0000fd69-0000-1000-8000-00805f9b34fb	246	-> Samsung SmartTag Offline Finding Advertisement
0000feed-0000-1000-8000-00805f9b34fb	234	-> Tile
0000180d-0000-1000-8000-00805f9b34fb	222	-> Heart Rate
eed6d5cc-c3b2-4d7b-8c6b-7acbf7965bb6	204	-> Samsung Galaxy Watch (based on my data)
00001855-0000-1000-8000-00805f9b34fb	202	-> Telephony and Media Audio
0000184c-0000-1000-8000-00805f9b34fb	196	-> Generic Telephone Bearer



Top 20 GATT Characteristics in my data FWIW

2023-10-26

char_UUID128	Count	
00002a00-0000-1000-8000-00805f9b34fb	19592	-> Device Name
00002a01-0000-1000-8000-00805f9b34fb	19205	-> Appearance
00002a05-0000-1000-8000-00805f9b34fb	18443	-> Service Changed
00002a29-0000-1000-8000-00805f9b34fb	16776	-> Manufacturer Name String
00002a24-0000-1000-8000-00805f9b34fb	16650	-> Model Number String
af0badb1-5b99-43cd-917a-a77bc549e3cc	15194	-> Apple Nearby Characteristic
8667556c-9a37-4c91-84ed-54ee27d90049	15193	-> Apple Continuity Characteristic
00002a19-0000-1000-8000-00805f9b34fb	10632	-> Battery Level
00002a2b-0000-1000-8000-00805f9b34fb	10245	-> Current Time
9fbf120d-6301-42d9-8c58-25e699a21dbd	10194	-> Apple Notification Center Notification Source
69d1d8f3-45e1-49a8-9821-9bbdfdaad9d9	10194	-> Apple Notification Center Control Point
22eac6e9-24d6-4bb5-be44-b36ace7c7fb	10192	-> Apple Notification Center Data Source
9b3c81d8-57b1-4a8a-b8df-0e56f7ca51c2	10185	-> Apple Media Center Remote Command
2f7cabce-808d-411f-9a0c-bb92ba96c102	10183	-> Apple Media Center Entity Update
c6b2f38c-23ab-46d8-a6ab-a3a870bbd5d7	10183	-> Apple Media Center Entity Attribute
00002a0f-0000-1000-8000-00805f9b34fb	10176	-> Local Time Information
00002a26-0000-1000-8000-00805f9b34fb	1682	-> Firmware Revision String
00002aa6-0000-1000-8000-00805f9b34fb	1655	-> Central Address Resolution
00002a04-0000-1000-8000-00805f9b34fb	1644	-> Peripheral Preferred Connection Parameters
00002a28-0000-1000-8000-00805f9b34fb	1563	-> Software Revision String

2thprint by UUID128



or





iBeacon: "The other UUID128!"

UUID128Print

- One of the most common types of "Manufacturer-Specific Data" (MSD) is the Apple-specified iBeacon (<https://developer.apple.com/ibeacon/>)
- It contains a UUID128 that beacon-deployers are supposed to associate with themselves, a Major ID and Minor ID that they're supposed to associate with individual beacons (e.g. Major ID for a country/store, and minor ID for an individual beacon)

```
Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
  Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088
Apple iBeacon:
  UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
  Major ID: 0000
  Minor ID: 0000
  RSSI at 1 meter: -120dBm
  In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
  This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)
```



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work : GATT -> Google Search

```
GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Readable' )
GATT Characteristic value read as b'MILWAUKEE TOOL'
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
GATT Descriptor: 00002a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
    GATT Characteristic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number String), Properties: 2 ('Readable' )
    GATT Characteristic value read as b'BLE112'
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
GATT Descriptor: 00002a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
    GATT Characteristic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number String), Properties: 2 ('Readable' )
    GATT Characteristic value read as b'123456789'
```



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer
istic value read as b'MILWAUKEE TOOL'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number
istic value read as b'BLE112'
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number
istic value read as b'123456789'
```



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer  
istic value read as b'MILWAUKEE TOOL'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9  
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10  
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number)  
istic value read as b'BLE112'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11  
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12  
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number)  
istic value read as b'123456789'
```



iBeacon: "The other UUID128" UUID128Print

- Detective Work 🕵️: GATT -> Google Search



```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer  
istic value read as b'MILWAUKEE TOOL'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9  
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10  
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number  
istic value read as b'BLE112'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11  
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12  
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number  
istic value read as b'123456789'
```



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer  
istic value read as b'MILWAUKEE TOOL'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9  
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10  
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number  
istic value read as b'BLE112'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11  
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12  
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number  
istic value read as b'123456789'
```



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer  
istic value read as b'MILWAUKEE TOOL'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9  
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10  
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number  
istic value read as b'BLE112'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11  
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12  
istic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number  
istic value read as b'123456789'
```

Shopping

Images

Videos

App

News

Maps

Books

Flights

Finance



Milwaukee Tool

<https://www.milwaukeetool.com> > Products

TICK Tool and Equipment Tracker

Bluetooth tracker for tools and equipment offers multiple attachment options and a low profile design - users can glue, screw, rivet or strap the TICK™...

★★★★★ Rating: 4.7 · 901 reviews

Missing: BLE112 | Show results with: BLE112



Milwaukee Tool

<https://www.milwaukeetool.com> > Products



Shopping

Images

Videos

App

News

Maps

Books

Flights

Finance

They make an AirTag "for tools"!

Where's your anti-stalking defense now?

and a low profile design - users can glue, screw, rivet or strap the TICK™

★★★★★ Rating: 4.7 · 901 reviews

Missing: BLE112 | Show results with: BLE112



Milwaukee Tool

<https://www.milwaukeetool.com> > Products



Shopping

Images

Videos

App

News

Maps

Books

Flights

Finance



Milwaukee Tool

<https://www.milwaukeetool.com> > Products

TICK Tool and Equipment Tracker

Bluetooth tracker for tools and equipment offers multiple attachment options and a low profile design - users can glue, screw, rivet or strap the TICK™...

★★★★★ Rating: 4.7 · 901 reviews

Missing: BLE112 | Show results with: BLE112



Milwaukee Tool

<https://www.milwaukeetool.com> > Products



Can Milwaukee tools be traced? ▾

Does Milwaukee have Bluetooth? ▾

Feedback



Silicon Labs

<https://www.silabs.com> › bluetooth › device.bled112

BLED112

The **BLED112 Bluetooth Low Energy Dongle** integrates all **Bluetooth LE** features. The **USB** dongle has a **virtual COM port** that enables seamless ho...



BLE112 bluetooth -"BLED112"



Images

Videos

Manual

Shopping

News

Maps

Books

Flights

Finan

Sponsored



Mouser Electronics

<https://www.mouser.com> :

[BLED112 Bluetooth Smart Dongle - Silicon Labs | Mouser](#)

Mouser is an Authorized Silicon Labs Distributor with Inventory, Prices & Datasheets. Huge Selection of Silicon Labs In Stock with No Minimum Orders & Fast Delivery! Fast Delivery. AS6496 Certified. Authorized Distributor. ISO 9001:2015. Order by 8PM CST.

Get phone number ▾

Selection of Silicon Labs In Stock with No Minimum Orders & Fast Delivery! Fast Delivery.

AS6496 Certified. Authorized Distributor. ISO 9001:2015. Order by 8PM CST.

📞 Get phone number ▾



Silicon Labs

<https://www.silabs.com> › public › data-sheets



BLE112 Datasheet

BLE112 offers all Bluetooth Low Energy features: radio, stack, profiles and application space for customer applications, so no external processor is needed. The ...

26 pages



Digikey

<https://forum.digikey.com> › getting-started-with-the-bl...



Getting Started with the BlueGiga BLE112 Bluetooth Low ...

Mar 9, 2021 — Purpose. This page explains how to set up the Silicon Labs BLE112 Bluetooth Low Energy module to communicate with a microcontroller via UART.



BLE112

DATA SHEET

Wednesday, 02 December 2020

Version 1.8



Selection of Silicon Labs In Stock with No Minimum Orders & Fast Delivery! Fast Delivery.

AS6496 Certified. Authorized Distributor. ISO 9001:2015. Order by 8PM CST.

📞 Get phone number ▾



Silicon Labs

<https://www.silabs.com> › public › data-sheets

⋮

BLE112 Datasheet

BLE112 offers all Bluetooth Low Energy features: radio, stack, profiles and application space for customer applications, so no external processor is needed. The ...

36 pages



Digikey

<https://forum.digikey.com> › getting-started-with-the-bl...

⋮

Getting Started with the BlueGiga BLE112 Bluetooth Low ...

Mar 9, 2021 — Purpose. This page explains how to set up the Silicon Labs BLE112 Bluetooth Low Energy module to communicate with a microcontroller via UART.



Selection of Silicon Labs In Stock with No Minimum Orders & Fast Delivery! Fast Delivery.

AS6496 Certified. Authorized Distributor. ISO 9001:2015. Order by 8PM CST.

📞 Get phone number ▾



Silicon Labs

<https://www.silabs.com> › public › data-sheets

⋮

BLE112 Datasheet

BLE112 offers all Bluetooth Low Energy features: radio, stack, profiles and application space for customer applications, so no external processor is needed. The ...

36 pages



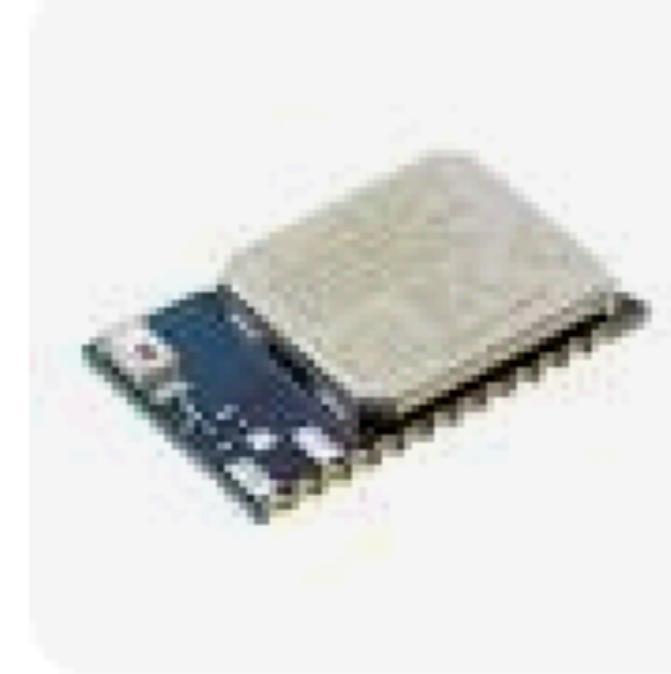
Digikey

<https://forum.digikey.com> › getting-started-with-the-bl...

⋮

Getting Started with the BlueGiga BLE112 Bluetooth Low ...

Mar 9, 2021 — Purpose. This page explains how to set up the Silicon Labs BLE112 Bluetooth Low Energy module to communicate with a microcontroller via UART





iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> **BLE112 == Bluegiga module**

n:
rvice: Begin Handle: 1 End Handle: 5 UUID128: 00001800-0000-1000-8000-00805f9b34fb (Generic Access)
GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 1
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 2
GATT Descriptor: 00002a00-0000-1000-8000-00805f9b34fb, Descriptor Handle: 3
 GATT Characteristic: 00002a00-0000-1000-8000-00805f9b34fb (Device Name), Properties: 10 ('Readable' 'Writable')
 GATT Characteristic value read as b'000C007771MKE '
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4
GATT Descriptor: 00002a01-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5
 GATT Characteristic: 00002a01-0000-1000-8000-00805f9b34fb (Appearance), Properties: 2 ('Readable')
 GATT Characteristic value read as b'\x00\x01'
 Appearance decodes as: Category (8): Tag, Sub-Category (0): Generic
rvice: Begin Handle: 6 End Handle: 12 UUID128: 0000180a-0000-1000-8000-00805f9b34fb (Device Information)
GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 6
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 7
GATT Descriptor: 00002a29-0000-1000-8000-00805f9b34fb, Descriptor Handle: 8
 GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Readable')
 GATT Characteristic value read as b'MILWAUKEE TOOL'
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
GATT Descriptor: 00002a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
 GATT Characteristic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number String), Properties: 2 ('Readable')
 GATT Characteristic value read as b'BLE112'
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
GATT Descriptor: 00002a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
 GATT Characteristic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number String), Properties: 2 ('Readable')
 GATT Characteristic value read as b'123456789'

0:c4:7a:
IEEE OUI (00:07:80): Bluegiga Technologies OY
Inquiry Result Device info.

d.

er found.

ata found.

pecific Data:
Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies pop
Endianness-flipped device company ID (in case the vendor used the wrong endian
a: 000c00777100
In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
This was found in an event of type 0 which corresponds to Connectable Undirected
ice Data found.

1:
Device: Begin Handle: 1 End Handle: 5 UUID128: 00001800-0000-1000-800

IEEE OUI (00:07:80): Bluegiga Technologies OY

Inquiry Result Device info.

d.

er found.

ata found.

pecific Data:

Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies pop
Endianness-flipped device company ID (in case the vendor used the wrong endian
a: 000c00777100

In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)

This was found in an event of type 0 which corresponds to Connectable Undirected

ice Data found.

1:

Device: Begin Handle: 1 End Handle: 5

WWID128: 00001800-0000-1000-8000

bluegiga silicon labs

X



Images

Bluetooth

App

Videos

BLED112

BLE GUI tool

UART demo

Shopp



Silicon Labs

<https://www.silabs.com> › developers › bluegiga-bluetooth... :

Bluegiga Bluetooth Smart Software Development Kit (SDK)

Bluegiga Bluetooth Smart Software is a complete Bluetooth Smart software stack for Bluegiga Legacy Bluetooth Smart products, such as BLE112, BLE113, ...

People also ask :

Is Silicon Labs safe?



Bluegiga Legacy Modules (BLE)

The **Bluegiga** Legacy Bluetooth 4.0 Low Energy (BLE) Modules were launched in 2011. For new Bluetooth 5 designs, we recommend our latest Bluetooth modules.



Silicon Labs Newsroom

<https://news.silabs.com › 2015-02-03-Silicon-Labs-Ac...> :

Silicon Labs Acquires Bluegiga, a Leader in Bluetooth and ...

Feb 3, 2015 — **Silicon Labs** completed the acquisition of **Bluegiga** Technologies Oy on January 30, 2015. Under the agreement, **Bluegiga** investors received ...

Bluegiga Legacy Modules (BLE)

The Bluegiga Legacy Bluetooth 4.0 Low Energy (BLE) Modules were launched in 2011. For new Bluetooth 5 designs, we recommend our latest Bluetooth modules.



Silicon Labs Newsroom

<https://news.silabs.com/silicon-labs-acquires-bluegiga>

Silicon Labs Acquires Bluegiga, a Leader in Bluetooth and ...

Feb 3, 2015 — Silicon Labs completed the acquisition of Bluegiga Technologies Oy on January 30, 2015. Under the agreement, Bluegiga investors received ...



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> **Bluegiga == Silicon Labs**



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer  
istic value read as b'MILWAUKEE TOOL'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9  
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10  
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number)  
istic value read as b'BLE112'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11  
a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12  
istic: 00002a25-0000-1000-8000-00805f9b34fb  
istic value read as b'123456789'
```



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs

```
istic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer  
istic value read as b'MILWAUKEE TOOL'  
803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9  
a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10  
istic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number  
istic value read as b'BLE112'  
80-000-1000-8000-00805f9b34fb, Descriptor Handle: 11  
805f9b34fb, Descriptor Handle: 12  
1000-8000-00805f9b34fb  
123456789  
Serial Number
```



Kinda seems
uninitialized

n:
rvice: Begin Handle: 1 End Handle: 5 UUID128: 00001800-0000-1000-8000-00805f9b34fb (Generic Access)
GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 1
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 2
GATT Descriptor: 00002a00-0000-1000-8000-00805f9b34fb, Descriptor Handle: 3
 GATT Characteristic: 00002a00-0000-1000-8000-00805f9b34fb (Device Name), Properties: 10 ('Readable' 'Writable')
 GATT Characteristic value read as b'000C007771MKE '
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4
GATT Descriptor: 00002a01-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5
 GATT Characteristic: 00002a01-0000-1000-8000-00805f9b34fb (Appearance), Properties: 2 ('Readable')
 GATT Characteristic value read as b'\x00\x01'
 Appearance decodes as: Category (8): Tag, Sub-Category (0): Generic
rvice: Begin Handle: 6 End Handle: 12 UUID128: 0000180a-0000-1000-8000-00805f9b34fb (Device Information)
GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 6
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 7
GATT Descriptor: 00002a29-0000-1000-8000-00805f9b34fb, Descriptor Handle: 8
 GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Readable')
 GATT Characteristic value read as b'MILWAUKEE TOOL'
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
GATT Descriptor: 00002a24-0000-1000-8000-00805f9b34fb, Descriptor Handle: 10
 GATT Characteristic: 00002a24-0000-1000-8000-00805f9b34fb (Model Number String), Properties: 2 ('Readable')
 GATT Characteristic value read as b'BLE112'
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 11
GATT Descriptor: 00002a25-0000-1000-8000-00805f9b34fb, Descriptor Handle: 12
 GATT Characteristic: 00002a25-0000-1000-8000-00805f9b34fb (Serial Number String), Properties: 2 ('Readable')
 GATT Characteristic value read as b'123456789'



iBeacon: "The other UUID128!"

UUID128Print

```
Manufacturer-specific Data:  
Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate  
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric  
Raw Data: 000c00777100  
    In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)  
    This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)  
  
s of Device Data found.
```

```
Information:  
GATT Service: Begin Handle: 1 End Handle: 5           UUID128: 00001800-0000-1000-8000-00805f9b34fb (Generic Access)  
GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 1  
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 2  
GATT Descriptor: 00002a00-0000-1000-8000-00805f9b34fb, Descriptor Handle: 3  
    GATT Characteristic: 00002a00-0000-1000-8000-00805f9b34fb (Device Name), Properties: 10 ('Readable' 'Writable')  
    GATT Characteristic value read as b'000C007771MKE '  
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4  
GATT Descriptor: 00002a01-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5  
    GATT Characteristic: 00002a01-0000-1000-8000-00805f9b34fb (Appearance), Properties: 2 ('Readable' )  
    GATT Characteristic value read as b'\x00\x01'  
        Appearance decodes as: Category (8): Tag, Sub-Category (0): Generic  
GATT Service: Begin Handle: 6 End Handle: 12          UUID128: 0000180a-0000-1000-8000-00805f9b34fb (Device Information)  
GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 6  
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 7  
GATT Descriptor: 00002a29-0000-1000-8000-00805f9b34fb, Descriptor Handle: 8  
    GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Readable' )  
    GATT Characteristic value read as b'MILWAUKEE TOOL'  
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
```



iBeacon: "The other UUID128!"

UUID128Print

```
Manufacturer-specific Data:  
Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate  
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric  
Raw Data: 000c00777100  
    In BT LE Data (LE_bdaddr_to_mf_specific), the random = 0 (Public)  
    This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)  
s of Device Data found.  
  
Information:  
GATT Service: Begin Handle: 1 End Handle: 5 UUID128: 0000180a-0000-1000-8000-00805f9b34fb, Descriptor Handle: 1  
GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 2  
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 3  
GATT Descriptor: 00002a00-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4  
    GATT Characteristic: 00002a00-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5  
        GATT Characteristic value read as b'000000///1MKE'  
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 4  
GATT Descriptor: 00002a01-0000-1000-8000-00805f9b34fb, Descriptor Handle: 5  
    GATT Characteristic: 00002a01-0000-1000-8000-00805f9b34fb (Appearance), Properties: 2 ('Readable' )  
    GATT Characteristic value read as b'\x00\x01'  
        Appearance decodes as: Category (8): Tag, Sub-Category (0): Generic  
GATT Service: Begin Handle: 6 End Handle: 12 UUID128: 0000180a-0000-1000-8000-00805f9b34fb (Device Information)  
GATT Descriptor: 00002800-0000-1000-8000-00805f9b34fb, Descriptor Handle: 6  
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 7  
GATT Descriptor: 00002a29-0000-1000-8000-00805f9b34fb, Descriptor Handle: 8  
    GATT Characteristic: 00002a29-0000-1000-8000-00805f9b34fb (Manufacturer Name String), Properties: 2 ('Readable' )  
    GATT Characteristic value read as b'MILWAUKEE TOOL'  
GATT Descriptor: 00002803-0000-1000-8000-00805f9b34fb, Descriptor Handle: 9
```



They seem to be using a big-endian BT CID, rather than the more common little-endian



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons

Manufacturer-specific Data:

Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric Tools)

Raw Data: 000c03249000

In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)

This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)

Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!

Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)

Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088

Apple iBeacon:

UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309

Major ID: 0000

Minor ID: 0000

RSSI at 1 meter: -120dBm

In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)

This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work : GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons

turer-specific Data:
Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric Tool)
Raw Data: 000c03249000
In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
Raw Data: 0215f34ebac47cd74027878e55f4e71d030900000000088
Apple iBeacon:
Major ID: 0000
Minor ID: 0000
RSSI at 1 meter: -120dBm
In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)



iBeacon: "The other UUID1 28!"

UUID128Print

- Detective Work : GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons

Kinda seems
uninitialized



iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128

Manufacturer-specific Data:

```
Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric Tool)
Raw Data: 000c03249000
    In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
    This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
Raw Data: 0215f34ebac47cd74027878e55f4e71d030900000000088
Apple iBeacon:
    UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
    Major ID: 0000
    Minor ID: 0000
    RSSI at 1 meter: -120dBm
    In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
    This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)
```





iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128

```
Manufacturer-specific Data:  
Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!  
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric Tool Corporation)  
Raw Data: 000c03249000  
    In BT LE Data (LE_bdaddr_to_mfg_specific), bdaddr_random = 0 (Public)  
    This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)  
Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!  
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)  
Raw Data: 0215f34ebac47cd74027878e55f4e71d030900000000088  
Apple iBeacon:  
    UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309  
    Minor ID: 0000  
    RSSI at 1 meter: -120dBm  
    In BT LE Data (LE_bdaddr_to_mfg_specific), bdaddr_random = 0 (Public)  
    This was found in an event of type 0 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

Where else does this UUID128 appear?





iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128

```
Manufacturer-specific Data:  
Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!  
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric Tool)  
Raw Data: 000c03249000  
    In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)  
    This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)  
Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!  
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)  
Raw Data: 0215f34ebac47cd74027878e55f4e71d030900000000088  
Apple iBeacon  
    UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309  
  
    Minor ID: 0000  
    RSSI at 1 meter: -120dBm  
    In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)  
    This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)
```





iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128

```
turer-specific Data:  
Device Company ID: 0x6501 (No Match) - take with a grain of salt, not all companies populate this accurately!  
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x0165 (Milwaukee Electric T  
Raw Data: 000c03249000  
    In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)  
    This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)  
Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!  
    Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)  
Raw Data: 0215f34ebac47cd74027878e55f4e71d030900000000088  
Apple iBeacon  
    UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309  
        Minor ID: 0000  
        RSSI: 00  
        In B  
        This  
        bdaddr_ (Public)  
        correct  
        n-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

Let's search for this UUID128, but remove anything associated with Milwaukee...



```
=====
For bdaddr = 84:71:27:69:c3:dc:
    Company Name by IEEE OUI (84:71:27): Silicon Laboratories

    No BTC Extended Inquiry Result Device info.

    No Names found.

    No UUID16s found.

    No transmit power found.

    No Appearance data found.

    Manufacturer-specific Data:
        Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
        Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
        Raw Data: 0215f34ebac47cd74027878e55f4e71d030900000000088
        Apple iBeacon:
            UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
            Major ID: 0000
            Minor ID: 0000
            RSSI at 1 meter: -120dBm
            In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
            This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)
```

No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

No BTC 2thprint Info found.

```
=====
For bdaddr = 84:fd:27:34:d2:62:
    Company Name by IEEE OUI (84:fd:27): Silicon Laboratories

    No BTC Extended Inquiry Result Device info.

    No Names found.
```

```
=====
For bdaddr = 84:71:27:69:c3:dc:
    Company Name by IEEE OUI (84:71:27) Silicon Laboratories
    No BTC Extended Inquiry Result Device info.

    No Names found.

    No UUID16s found.

    No transmit power found.

    No Appearance data found.

    Manufacturer-specific Data:
        Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!
        Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)
        Raw Data: 0215f34ebac47cd74027878e55f4e71d030900000000088
        Apple iBeacon:
            UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309
            Major ID: 0000
            Minor ID: 0000
            RSSI at 1 meter: -120dBm
            In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
            This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

    No Class of Device Data found.

    No GATT Information found.

    No BLE 2thprint Info found.

    No BTC 2thprint Info found.

=====
For bdaddr = 84:fd:27:34:d2:62:
    Company Name by IEEE OUI (84:fd:27): Silicon Laboratories
    No BTC Extended Inquiry Result Device info.

    No Names found.
```

In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

=====

For bdaddr = 88:6b:0f:0e:22:f0:

Company Name by IEEE OUI (88:6b:0f) Bluegiga Technologies OY

No BTC Extended Inquiry Result Device info.

No Names found.

No UUID16s found.

No transmit power found.

No Appearance data found.

Manufacturer-specific Data:

Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!

Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)

Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088

Apple iBeacon:

UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309

Major ID: 0000

Minor ID: 0000

RSSI at 1 meter: -120dBm

In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)

This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

No BTC 2thprint Info found.

This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

=====

For bdaddr = 88:6b:0f:0e:22:f0:

Company Name by IEEE OUI (88:6b:0f) Bluegiga Technologies OY

No BTC Extended Inquiry Result Device info.

No Names found.

No UUID16s found.

No transmit power found.

No Appearance data found.

Manufacturer-specific Data:

Device Company ID: 0x004c (Apple, Inc.) - take with a grain of Endianness-flipped device company ID (in case the vendor is Apple)

Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088

Apple iBeacon:

UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309

Major ID: 0000

Minor ID: 0000

RSSI at 1 meter: -120dBm

In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random

This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

No BTC 2thprint Info found.

Does this UUID128, ever appear with any device NOT associated with Silicon Labs or Bluegiga based on the OUI?

accurately!
(No Match)

ing (ADV_NONCONN_IND)



In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)
This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

=====

For bdaddr = 88:6b:0f:0e:22:f0:

Company Name by IEEE OUI (88:6b:0f): Bluegiga Technologies OY

No BTC Extended Inquiry Result Device info.

No Names found.

No UUID16s found.

No transmit power found.

No Appearance data found.

Manufacturer-specific Data:

Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!

Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)

Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088

Apple iBeacon:

UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309

Major ID: 0000

Minor ID: 0000

RSSI at 1 meter: -120dBm

In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)

This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

No BTC 2thprint Info found.

This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

=====

For bdaddr = 88:6b:0f:0e:22:f0:

Company Name by IEEE OUI (88:6b:0f): Bluegiga Technologies OY

No BTC Extended Inquiry Result Device info.

No Names found.

No UUID16s found.

No transmit power found.

No Appearance data found.

Manufacturer-specific Data:

Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt. This is accurately
Endianness-flipped device company ID (in case the vendor has swapped the bytes).

Raw Data: 0215f34ebac47cd74027878e55f4e71d03090000000088

Apple iBeacon:

UUID128: f34ebac4-7cd7-4027-878e-55f4e71d0309

Major ID: 0000

Minor ID: 0000

RSSI at 1 meter: -120dBm

In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random

This was found in an event of type 3 which corresponds to Non-Connectable Undirected Advertising (ADV_NONCONN_IND)

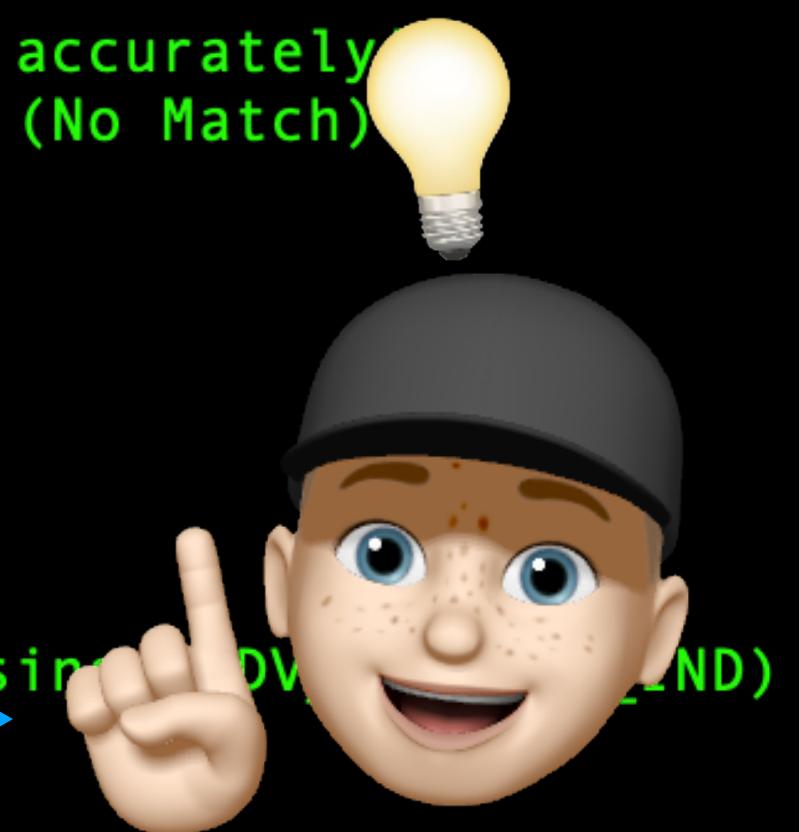
No Class of Device Data found.

No GATT Information found.

No BLE 2thprint Info found.

No BTC 2thprint Info found.

No! Therefore, there is a high probability that this UUID128, if present, is strongly indicative of Bluegiga/SiLabs chips!





iBeacon: "The other UUID128!"

UUID128Print

- Detective Work 🕵️: GATT -> Google Search -> BLE112 == Bluegiga module -> Google Search -> Bluegiga == Silicon Labs -> iBeacons -> UUID128
- *iBeacon UUID128 **f34ebac4-7cd7-4027-878e-55f4e71d0309** == Bluegiga/Silicon chips!*
 - It doesn't hurt that the OUIPrint is corroborating ;)
 - Note: I could be wrong, and it could be that this UUID128 is actually all Milwaukee equipment that is missing the other IDs like UUID16 or Milwaukee MSD or GATT info
 - This is why we need more crowdsourced data!



iBeacon: "The other UUID128!"

UUID128Print



iBeacon: "The other UUID128!"

UUID128Print

- Plot Twist! The Bluegiga *module* was based on a TI *chip*!



iBeacon: "The other UUID128!"

UUID128Print

- Plot Twist! The Bluegiga *module* was based on a TI *chip*!

7 Block diagram

BLE112 is based on TI's CC2540 chip. Embedded 32 MHz and 32.678 kHz crystals are used for clock generation. Matched balun and low pass filter provide optimal radio performance with extremely low spurious emissions. Small ceramic chip antenna gives good radiation efficiency even when the module is used in layouts with very limited space. <https://www.silabs.com/documents/public/data-sheets/BLE112-DataSheet.pdf>



iBeacon: "The other UUID128!"

UUID128Print

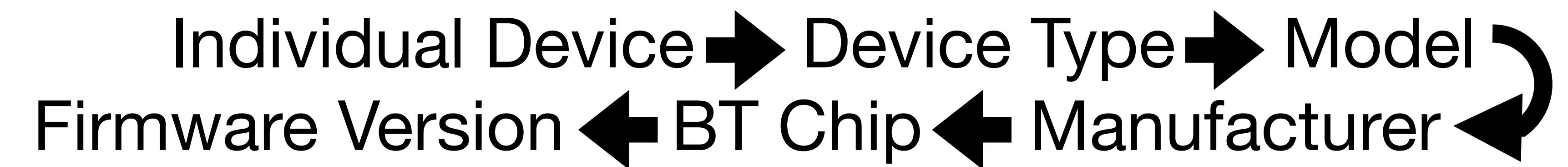
- Plot Twist! The Bluegiga *module* was based on a TI *chip*!

7 Block diagram

BLE112 is based on TI's CC2540 chip. Embedded 32 MHz and 32.678 kHz crystals are used for clock generation. Matched balun and low pass filter provide optimal radio performance with extremely low spurious emissions. Small ceramic chip antenna gives good radiation efficiency even when the module is used in layouts with very limited space. <https://www.silabs.com/documents/public/data-sheets/BLE112-DataSheet.pdf>

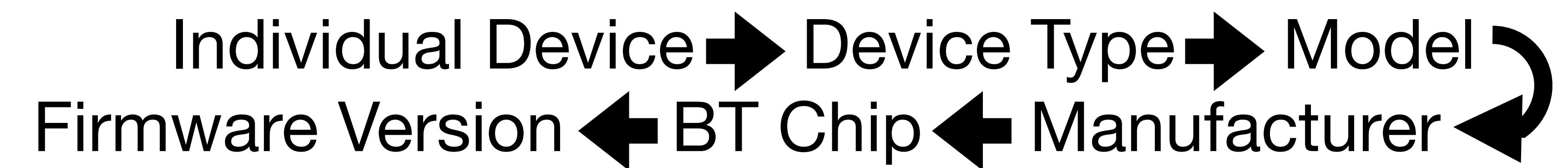


But sometimes...





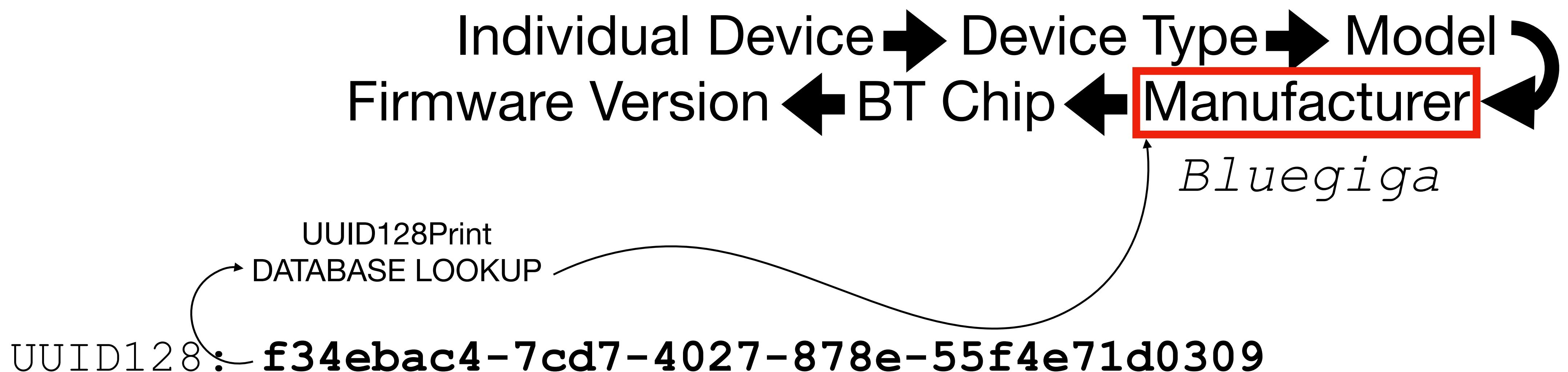
But sometimes...



UUID128: **f34ebac4-7cd7-4027-878e-55f4e71d0309**

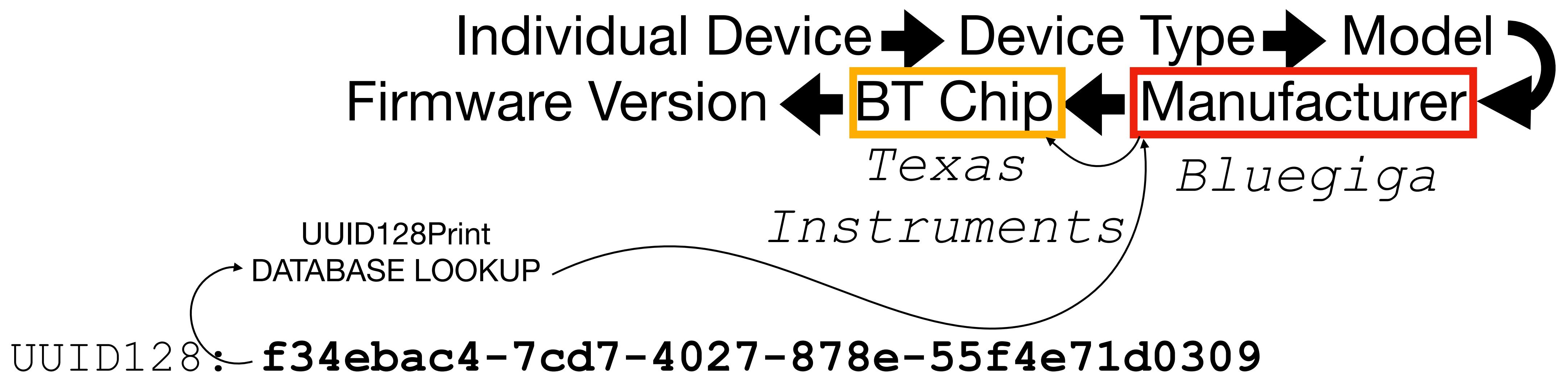


But sometimes...



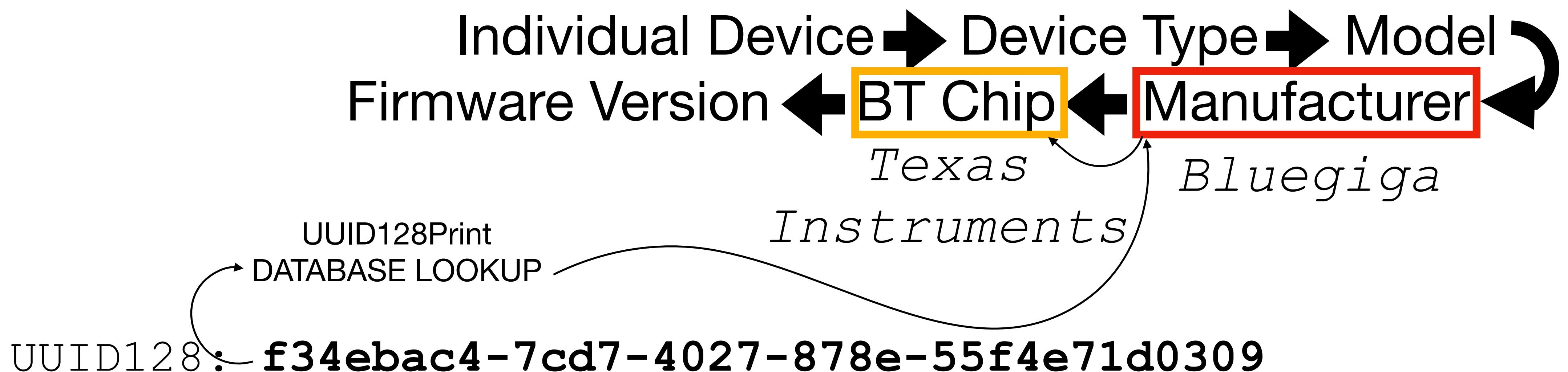


But sometimes...





But sometimes...



ASSUMPTION:

UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers

2thprint by UUID16



or





2thprint by UUID16

UUID16Print

- 16-bit UUIDs can be optionally included in BTC EIR packets or BLE Advertisements
- They are frequently used to advertise company-specific services (usually served over GATT)
- Silicon vendors are present, but not prevalent

```
./TellMeEverything.py --UUID16stats=quiet
----- BLUETOOTH CLASSIC RESULTS -----
count  uuid16  company
✗ 254  0xfe4c  Volkswagen AG
✗ 57   0xfe31  Volkswagen AG
✗ 31   0xfe35  HUAWEI Technologies Co., Ltd
✗ 24   0xfd81  Huawei Technologies Co., Ltd
🍪 12   0xfd69  Samsung Electronics Co., Ltd
✗ 1    0xeeaa  Swirl Networks, Inc.

----- BLUETOOTH LOW ENERGY RESULTS -----
count  uuid16  company
✗ 109258  0xfef3  Google LLC
🍪 108953  0xfd6f  Apple, Inc.
🍪 44895   0xfd69  Samsung Electronics Co., Ltd
✗ 29877   0xfeed  Tile, Inc.
✗ 29059   0xfebe  Bose Corporation
✗ 13392   0xfeaa  Google LLC
✗ 13184   0xfe9f  Google LLC
✗ 12533   0xfe03  Amazon.com Services, Inc.
✗ 9450    0xfd5a  Samsung Electronics Co., Ltd.
✗ 8922    0xfe50  Google LLC
✗ 6642    0xfe1f  Garmin International, Inc.
✗ 5107    0xfe2c  Google LLC
✗ 4686    0xfe61  Logitech International SA
✗ 3903    0xfe4c  Volkswagen AG
✗ 3801    0xfe26  Google LLC
✗ 2663    0xfea0  Google LLC
✗ 2451    0xfd82  Sony Corporation
✗ 2420    0xfe78  Hewlett-Packard Company
✗ 2353    0xeee7  Tencent Holdings Limited.
✗ 2018    0xfe48  General Motors
...
```



2thprint by UUID16
UUID16Print

- 16-bit UUIDs can be optionally included in BTC EIR packets or BLE Advertisements
 - They are frequently used to advertise company-specific services (usually served over GATT)
 - Silicon vendors are present, but not prevalent

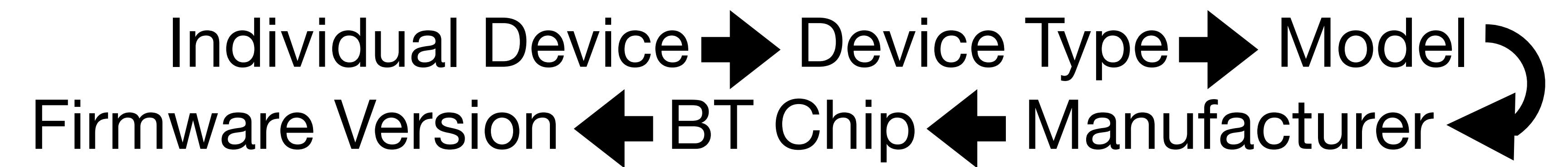
1740	0xfee0	🍪 Anhui Huami Information Technology Co., Ltd.
1558	0xfd92	🍪 Qualcomm Technologies International, Ltd. (QTIL)
749	0xfef5	🍪 Dialog Semiconductor GmbH
216	0xfe8f	🍪 CSR
189	0xefeb	🍪 Telit Wireless Solutions (Formerly Stollmann E+V)
136	0xfef1	🍪 CSR
74	0xfe59	🍪 Nordic Semiconductor ASA

```
./TellMeEverything.py --UUID16stats=quiet
----- BLUETOOTH CLASSIC RESULTS -----
count  uuid16  company
254  0xfe4c  Volkswagen AG
57   0xfe31  Volkswagen AG
31   0xfe35  HUAWEI Technologies Co., Ltd
24   0xfd11  Huawei Technologies Co., Ltd
12   0xfd69  Samsung Electronics Co., Ltd
1    0xfeeaa Swirl Networks, Inc.

----- BLUETOOTH LOW ENERGY RESULTS -----
count  uuid16  company
109258 0xfef3  Google LLC
108953 0xfd6f  Apple, Inc.
44895  0xfd69  Samsung Electronics Co., Ltd
29877  0xfeed  Tile, Inc.
29059  0xfebe  Bose Corporation
13392  0xfeaa  Google LLC
13184  0xfe9f  Google LLC
12533  0xfe03  Amazon.com Services, Inc.
9450   0xfd5a  Samsung Electronics Co., Ltd.
8922   0xfe50  Google LLC
6642   0xfe1f  Garmin International, Inc.
5107   0xfe2c  Google LLC
4686   0xfe61  Logitech International SA
      0xfe4c  Volkswagen AG
      0xfe26  Google LLC
      0xfea0  Google LLC
      0xfd82  Sony Corporation
      0xfe78  Hewlett-Packard Company
      0xee7   Tencent Holdings Limited.
      0xfe48  General Motors
```

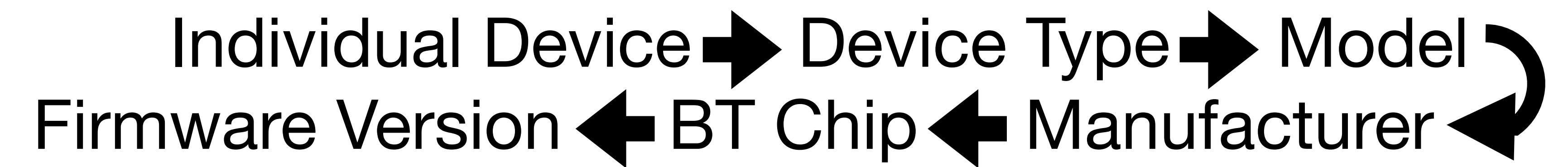


What I Want





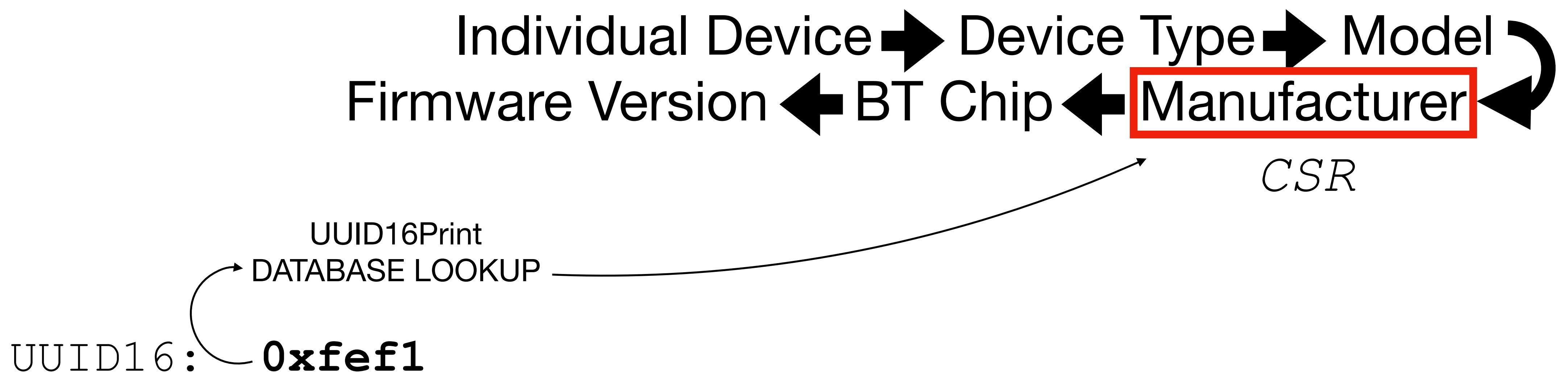
What I Want



UUID16: **0xefef1**

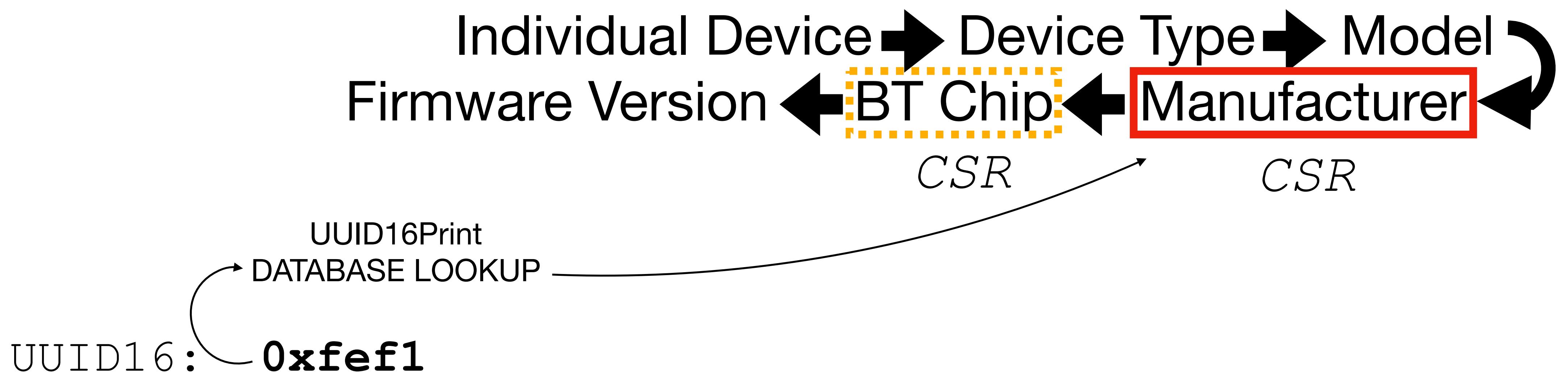


What I Want



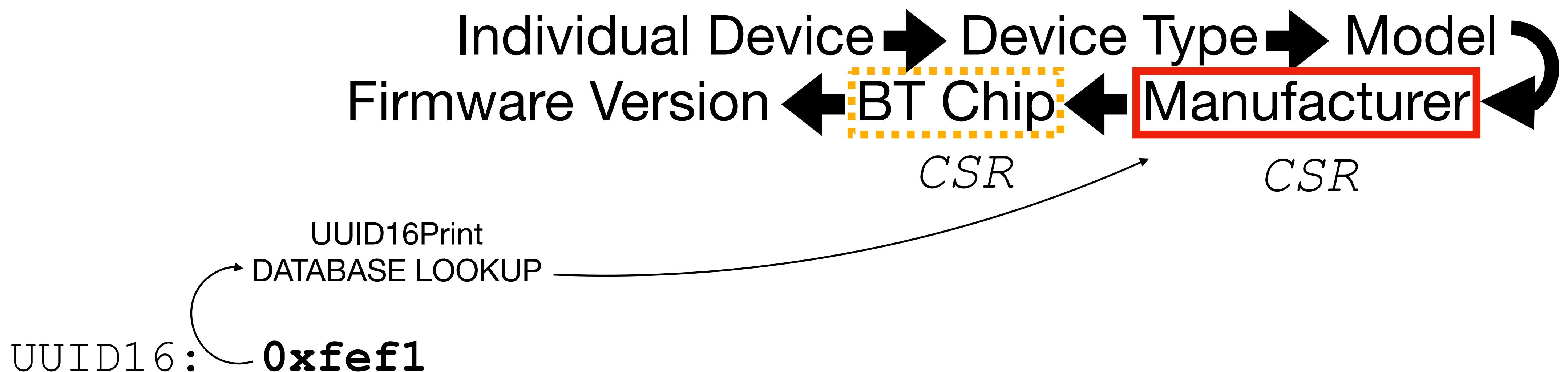


What I Want





What I Want



ASSUMPTION:

UUID16Prints are just the assigned values from the BT
public/assigned_numbers/uuids/member_uuids.yaml

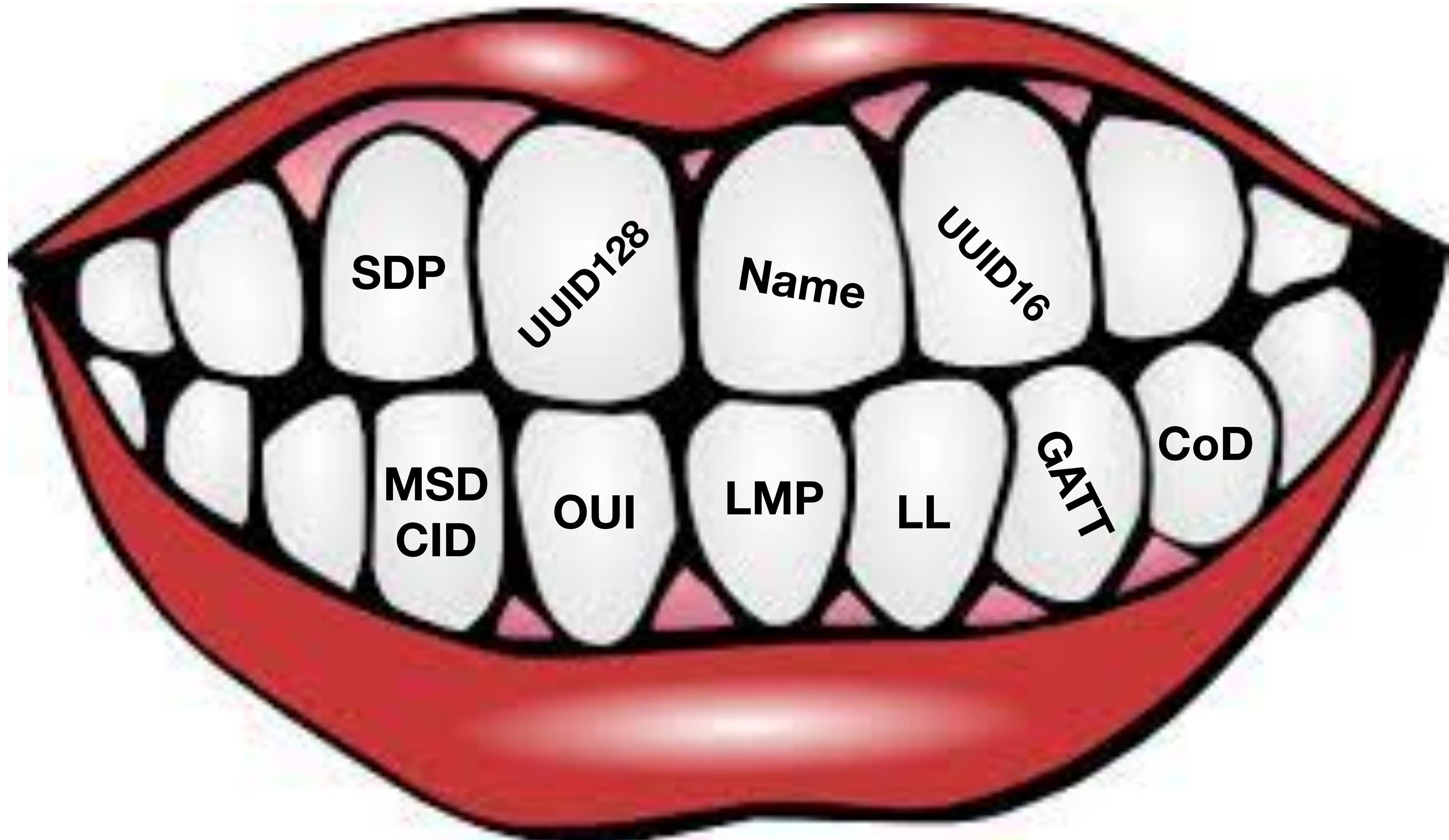


Putting It All *Tooth*gether 😊



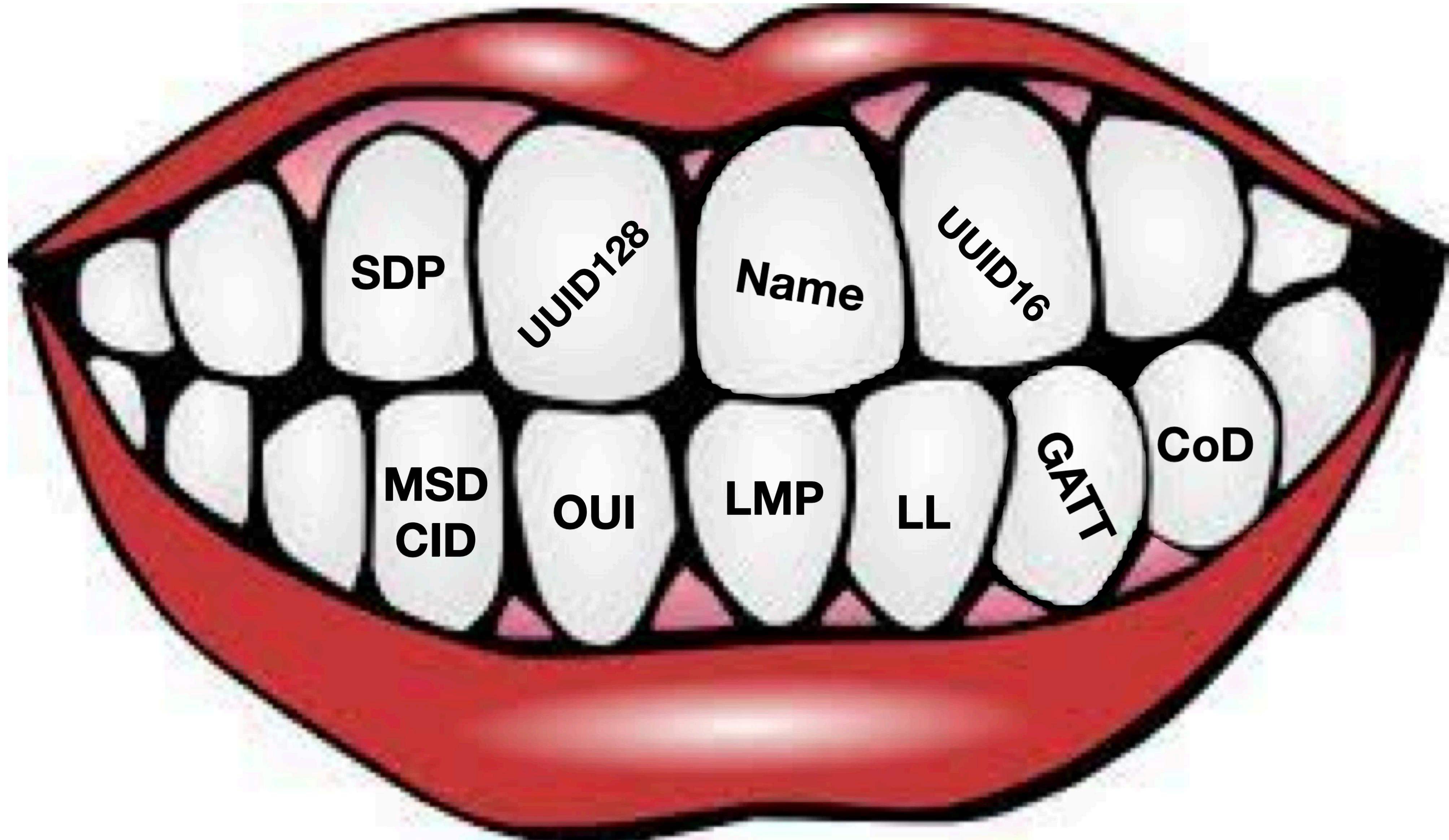


Putting It All *Tooth*gether 😜



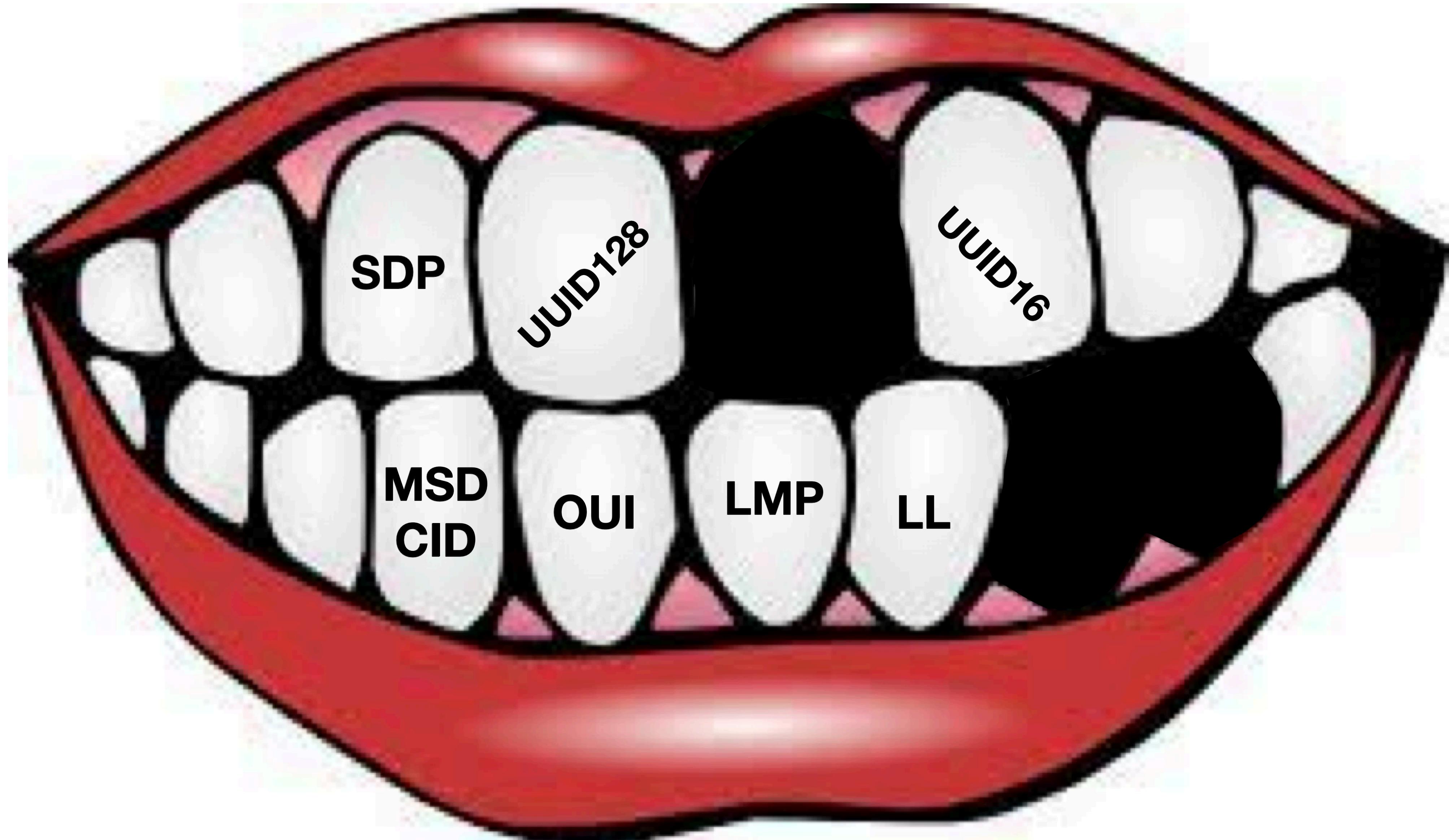


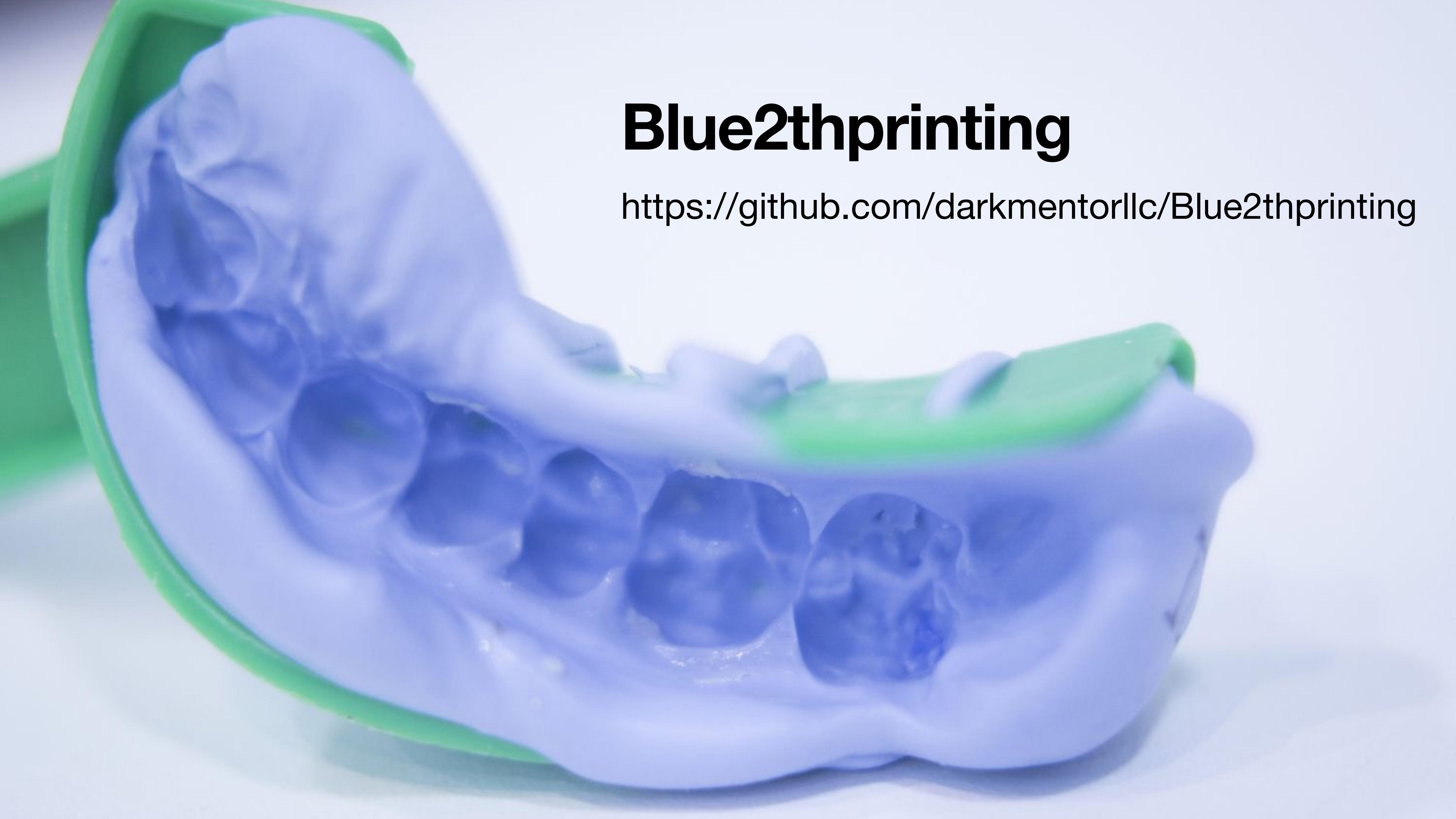
Reminder, even if wireless links weren't *lossy*,
you're not generally going to have all the data for every device





Reminder, even if wireless links weren't *lossy*,
you're not generally going to have all the data for every device





Blue2thprinting

<https://github.com/darkmentorllc/Blue2thprinting>



Call To Action!



JOIN ME! AND TOGETHER
WE CAN RULE THE
BLUETOOTH GALAXY!



Conclusion

- Bluetooth *vulnerability assessment* **is not yet a thing you can really do!**
- This is an active research topic I'm working on, but it needs more researchers working on it (because this is only 25% of my time ;))
 - "I'm puttin' together a (Blue)crew..."
- The starting point, as always, is to read related work
- I've organized the related work into a TiddlyWiki that I will continue to update over the coming years, and which others can contribute to via github PRs
- <https://darkmentor.com/bt.html>



Fin



- BT research is cool
- But *OpenSecurityTraining2* (<https://ost2.fyi> , @OpenSecTraining) *is cooler!*
 - We'll have BT classes eventually, but in the meantime there's so much other stuff to learn! Reverse Engineering, Vulnerabilities, Firmware, System Architecture!
- You should take a class, or *teach* a class!

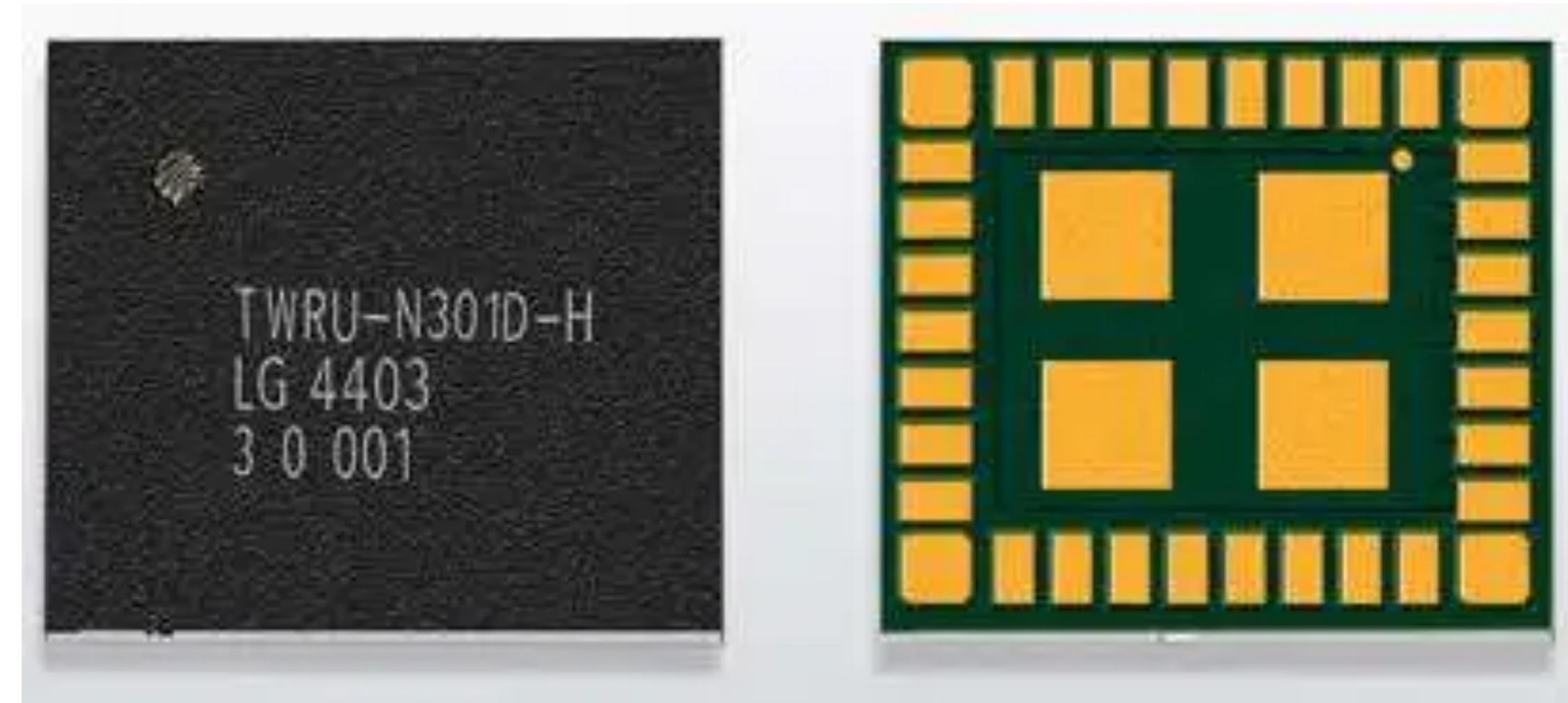


Backup



Ongoing work

- How should I structure the 2thprints data?!
- What % of devices respond to GATT/SDP/LMP2th/LL2th/etc requests?
 - Working with a student on a research project to investigate more rigorously
- What other packet types (teef!) can we add to improve the 2thprint? (e.g. L2CAP? RFCOMM?)
- Can we do the work they didn't do in [1] to *automatically* generate minimal chip-model-differentiation packet sequences from existing learned state machines?



"LG Innotek has developed a Nordic nRF51822 SoC-based Bluetooth Smart (previously known as Bluetooth low energy) module"

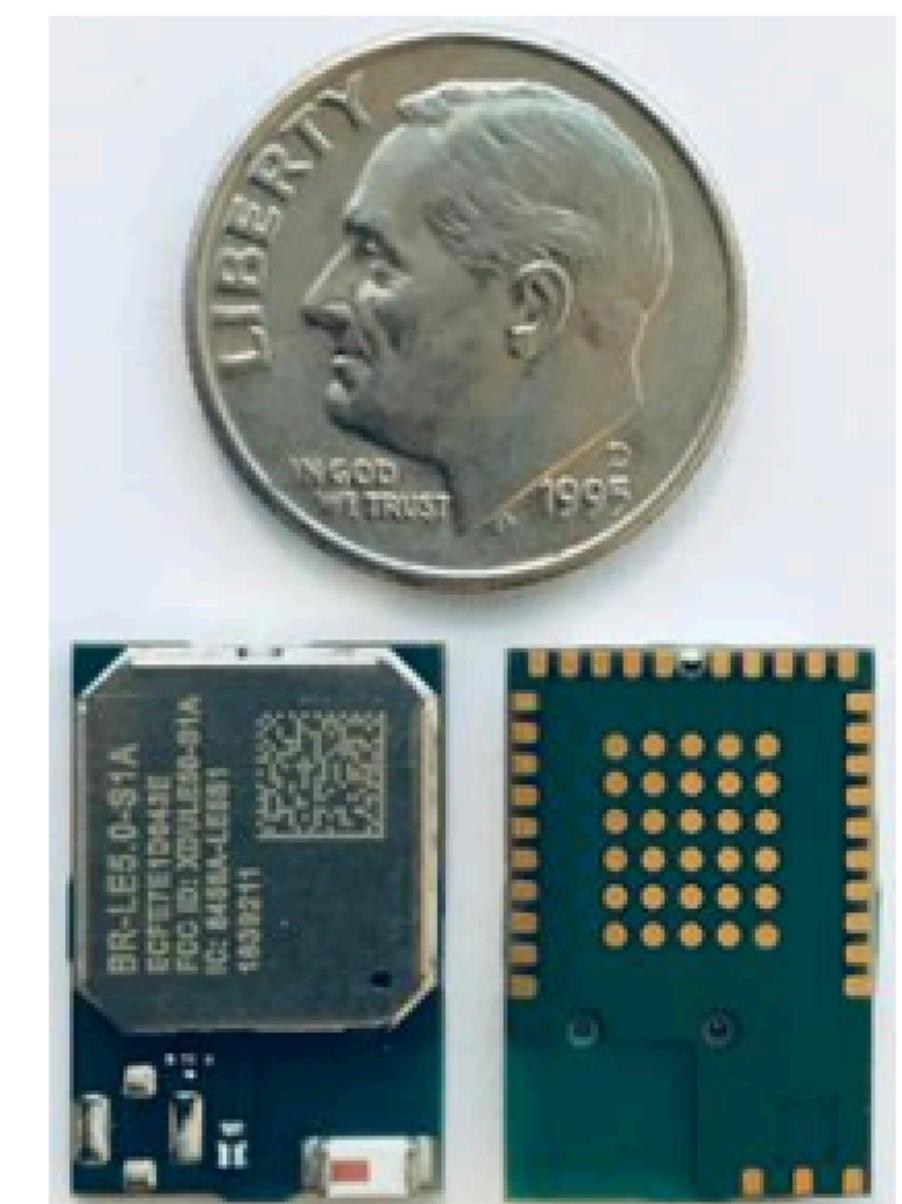
So for my purposes, it will almost certainly present as a Nordic chip, even though it's technically custom LG silicon.



BT5.0 Low Energy Single Mode Class 1 SoC Module

nBlue™ BR-LE5.0-S1A (nRF52840)

- **AT HOME. AT WORK. ON THE ROAD. USING BT5.0 LOW ENERGY WIRELESS TECHNOLOGY MEANS TOTAL FREEDOM FROM THE CONSTRAINTS AND CLUTTER OF WIRES IN YOUR LIFE.**
- FCC, IC, CE, RoHS, and BT5.0 Certified ISM 2.4GHz module supporting BT5.0 high speed mode, long range mode and advertising extensions. Can also support BT5.0 Mesh, 802.15.4 for Thread and Zigbee, ANT or proprietary 2.4Ghz.
- Utilizes the Nordic nRF52840 SoC. 64Mhz ARM® Cortex™ M4F 32-bit processor with FPU, 1MB Flash, 256K RAM, built in DC-DC converter and ARM CryptoCell cryptographic accelerator.
- Programmable output power from -40dBm to +8dBm for short to long range applications.
- Over 1000 meter line of site distance with integrated antenna. External antenna can be connected to RF_OUT pad or through optional u.FL connector (requires moving RF path resistor).
- Can be externally controlled via simple ASCII AT commands over UART, USB and BT5.0, or programmed with custom applications embedded in the module.

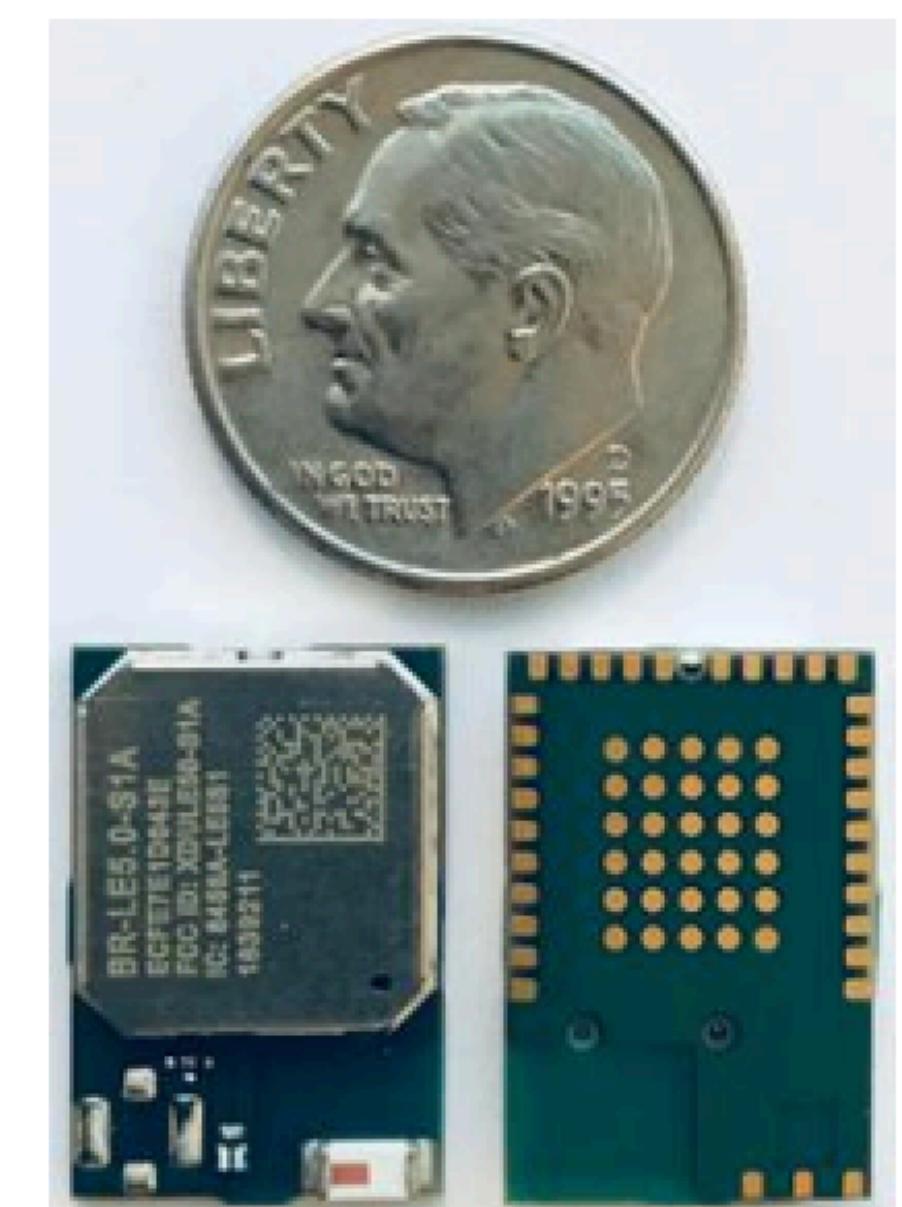




BT5.0 Low Energy Single Mode Class 1 SoC Module

nBlue™ BR-LE5.0-S1A (nRF52840)

- **AT HOME. AT WORK. ON THE ROAD. USING BT5.0 LOW ENERGY WIRELESS TECHNOLOGY MEANS TOTAL FREEDOM FROM THE CONSTRAINTS AND CLUTTER OF WIRES IN YOUR LIFE.**
- FCC, IC, CE, RoHS, and BT5.0 Certified ISM 2.4GHz module supporting BT5.0 high speed mode, long range mode and advertising extensions. Can also support BT5.0 Mesh, 802.15.4 for Thread and Zigbee, ANT or proprietary 2.4Ghz.
- Utilizes the Nordic nRF52840 SoC. 64Mhz ARM® Cortex™ M4F 32-bit processor with FPU, 1MB Flash, 256K RAM, built in DC-DC converter and ARM CryptoCell cryptographic accelerator.
- Programmable output power from -40dBm to +8dBm for short to long range applications.
- Over 1000 meter line of site distance with integrated antenna. External antenna can be connected to RF_OUT pad or through optional u.FL connector (requires moving RF path resistor).
- Can be externally controlled via simple ASCII AT commands over UART, USB and BT5.0, or programmed with custom applications embedded in the module.



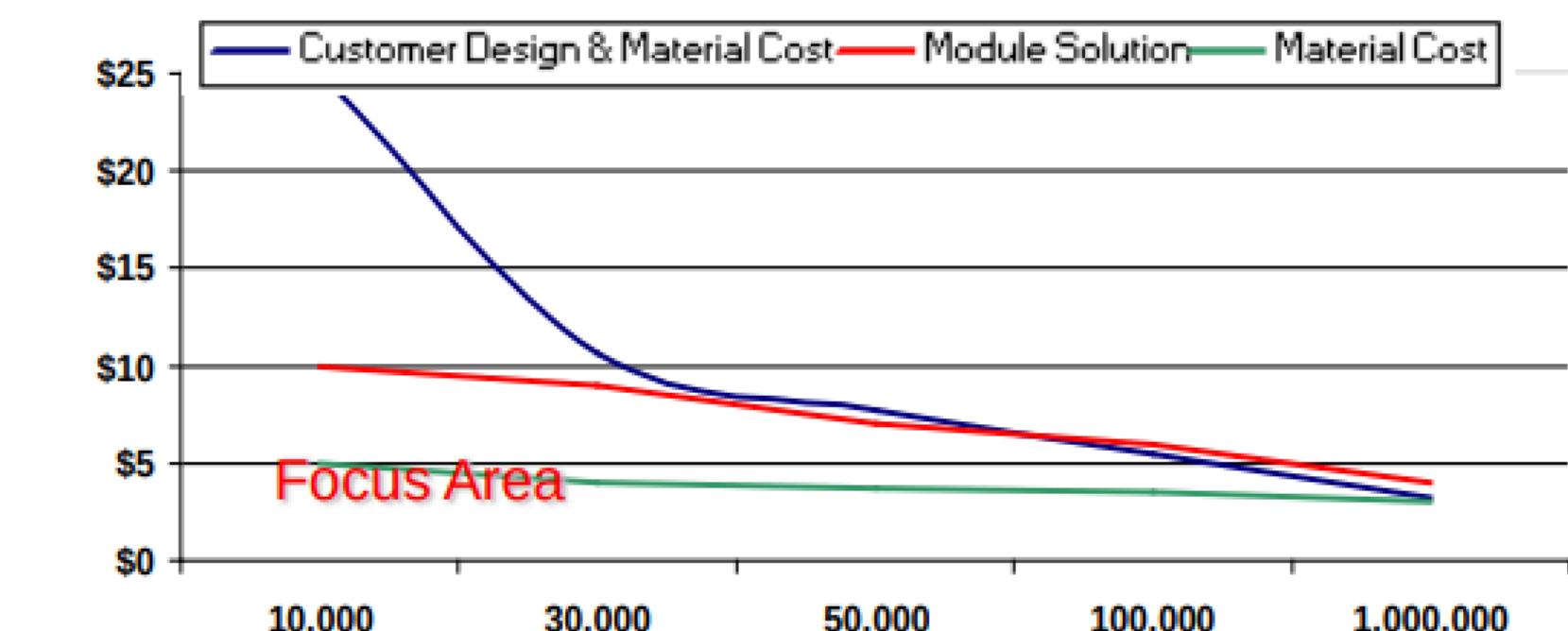


Buying Modules vs. Chips Aside

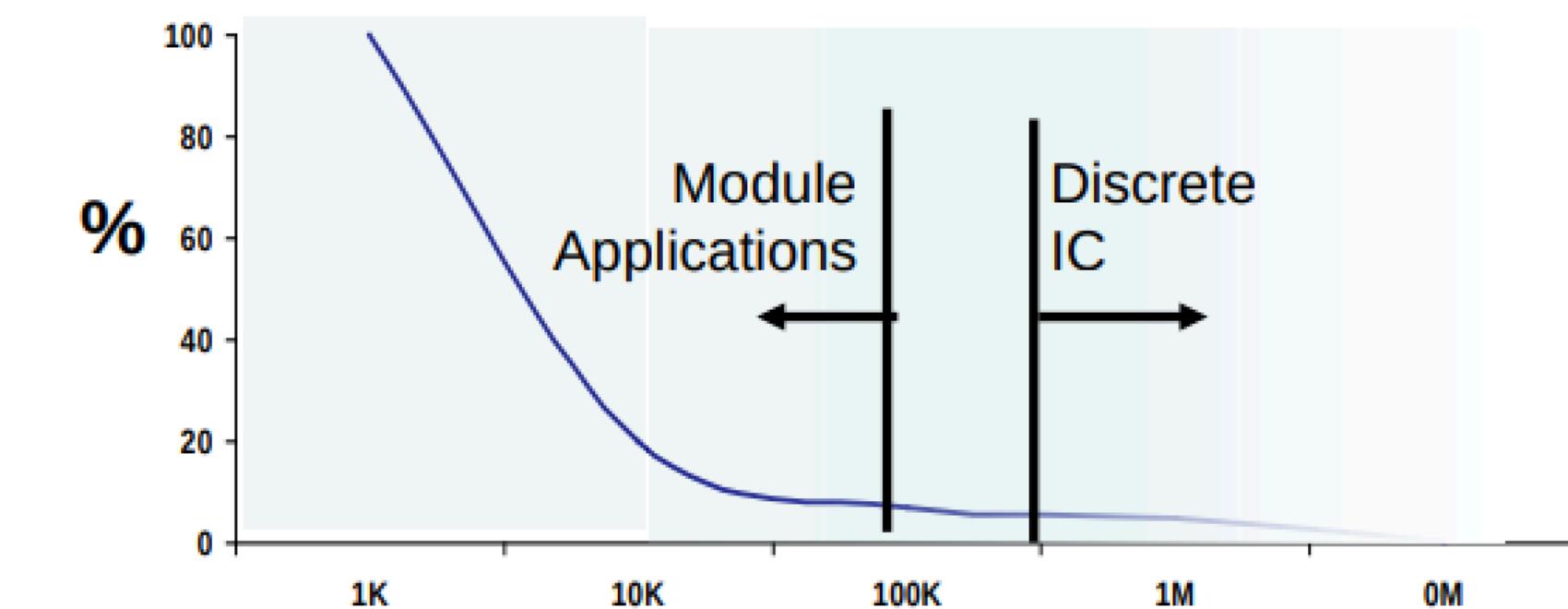
	Module	Discrete/On-board	Module Benefit
RF Design	Core competency of module vendor Heavy 1-time investment of module vendor	Expertise required for layout, signal routing, layer stack-up, interference, shielding	Lower RF team needs Less board design iterations
Size	Size optimized	Non module will require larger area on target PCB	Saves board area
Procurement	1 component	Non module will increase procured BOM elements management	Reduce operational costs
Assembly	1 component	Full BOM	Reduced production cost
Test	Module fully tested	Individually tested end-product	Reduced production cost
Quality	Modules are fully tested and provided as known good	RF expertise and test flows to cover connectivity subsystem	Increased quality
Yield Loss	Pre-yielded modules	Yield losses in production Failure analysis & rework costs	Reduced production cost

RF Mo

Module vs. Chip Cost

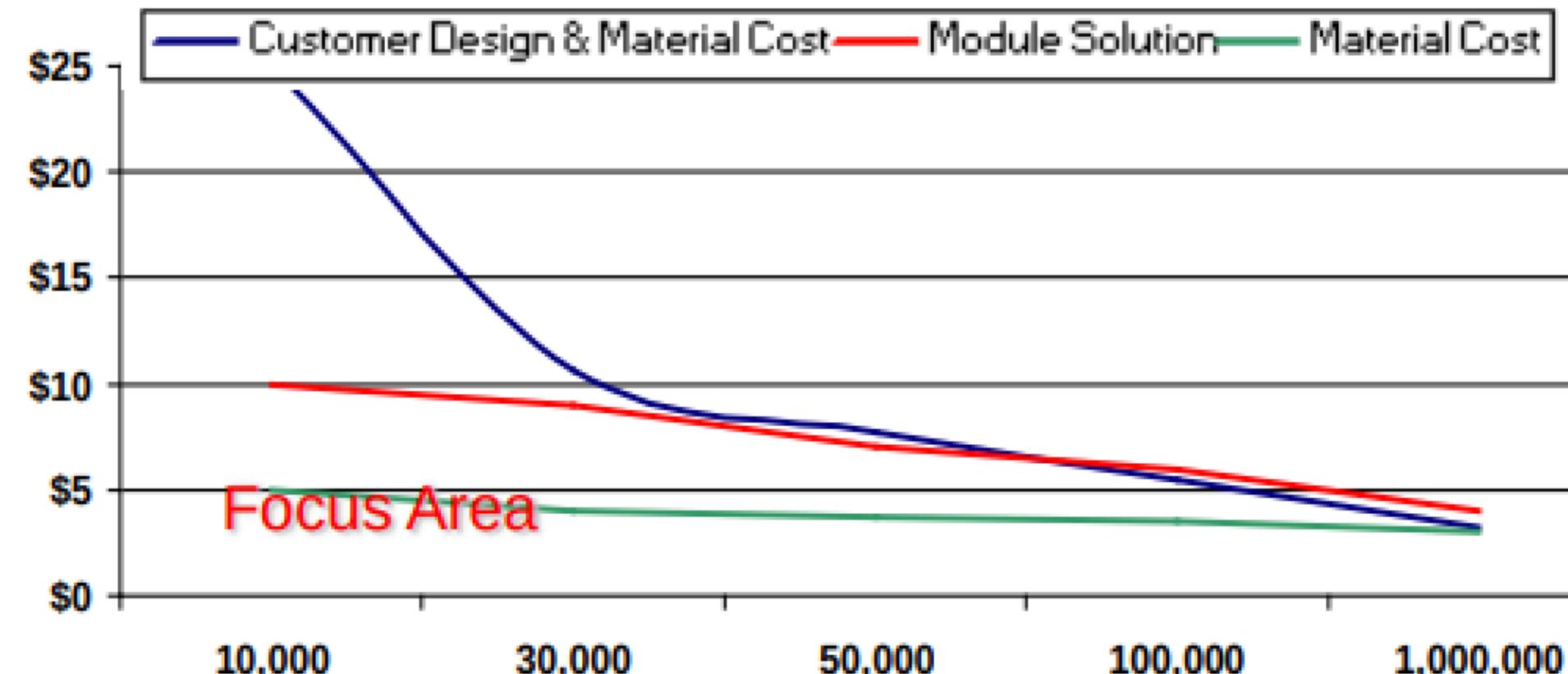


Module Adaption Rate





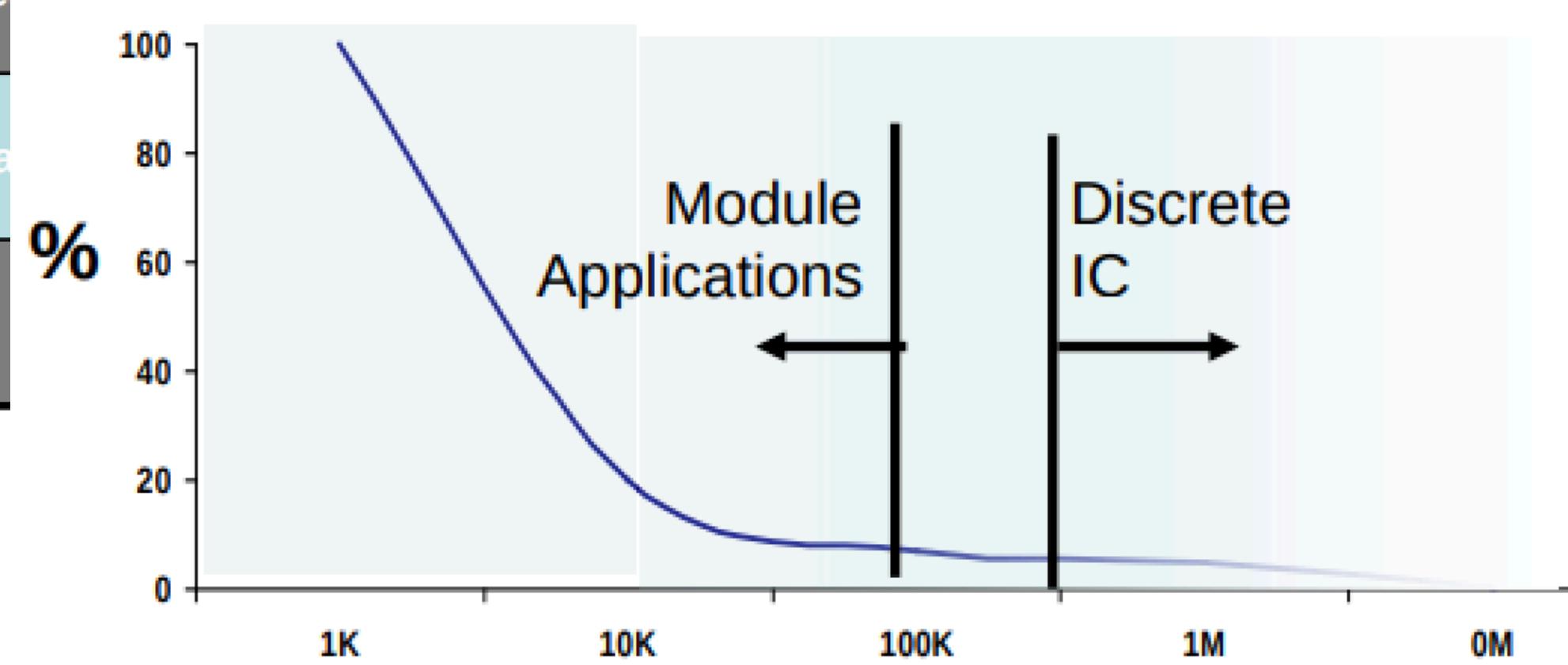
Module vs. Chip Cost



Customer Decision: Module vs Chip

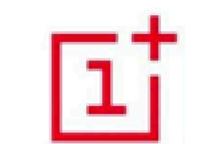
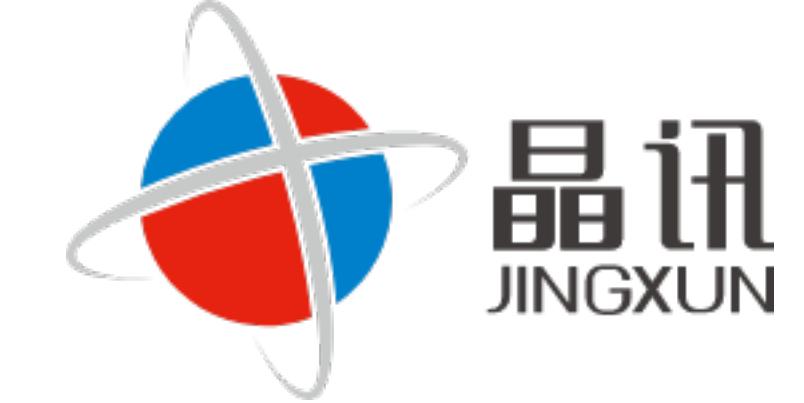
- Initial Development / RF Certification & Testing -- \$181,970
 - Respin
 - Hardware Engineering
 - Firmware
 - Layout
 - Testing & Verification
 - Certification
- TOC = BOM + Mfg & Packaging

Module Adaption Rate



Module Adaption Considerations

- Time to market – Modules reduce time to market.
- Product Life – Modules are the best choice for products with a long life cycle, as a module life can be extended beyond IC life cycle by using a single footprint over several generations.



From

<https://www.jingxun.xyz/en/portal/page/index/id/34.html>

2thprint by Name



or





2thprint by Name

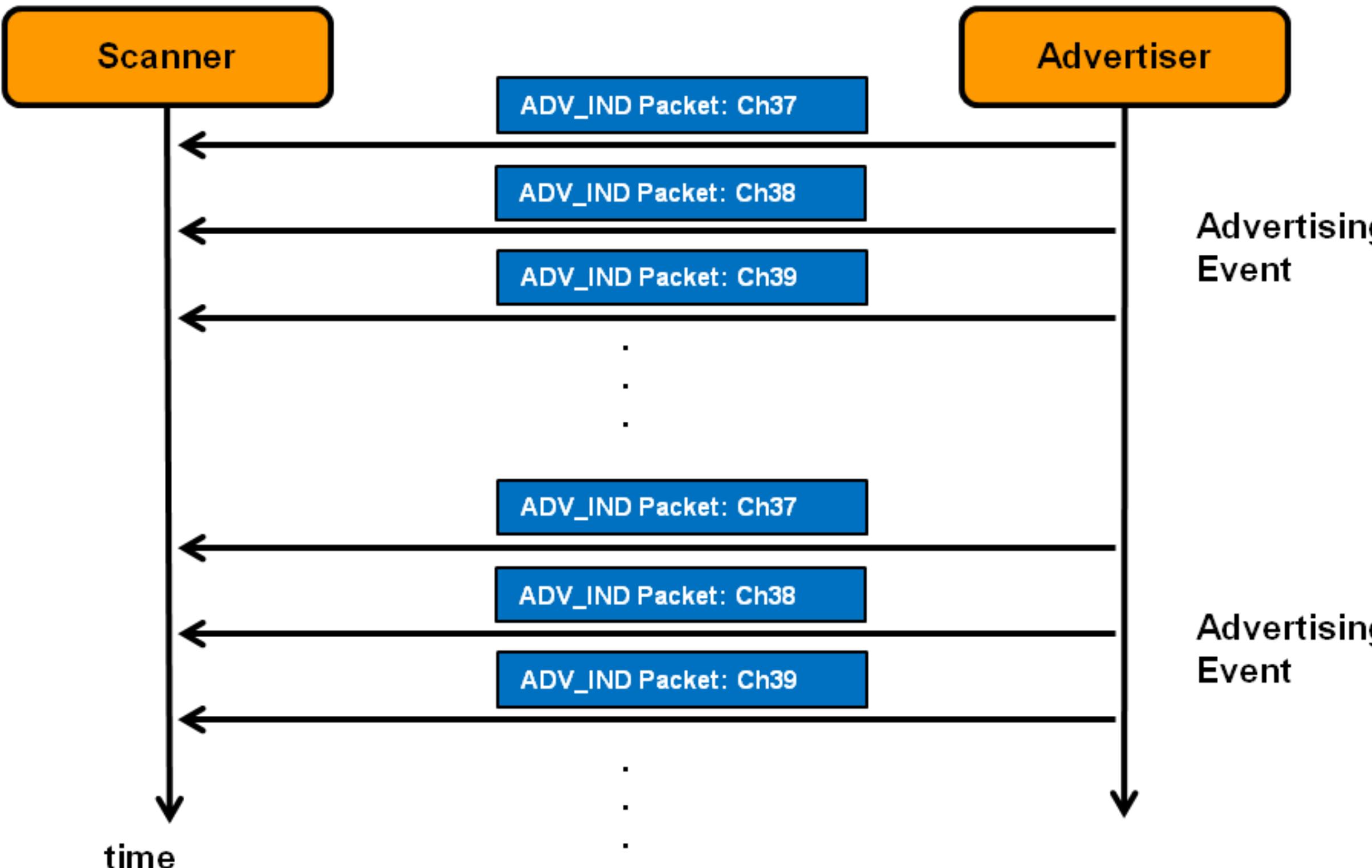
aka "*NamePrint*"

- Let's start with the easy situation, where a device more or less literally tells you what it is, based on its name
 - "Ember Ceramic Mug"
 - "Nest Cam"
 - "Versa 4"
 - "User's MacBook Pro"

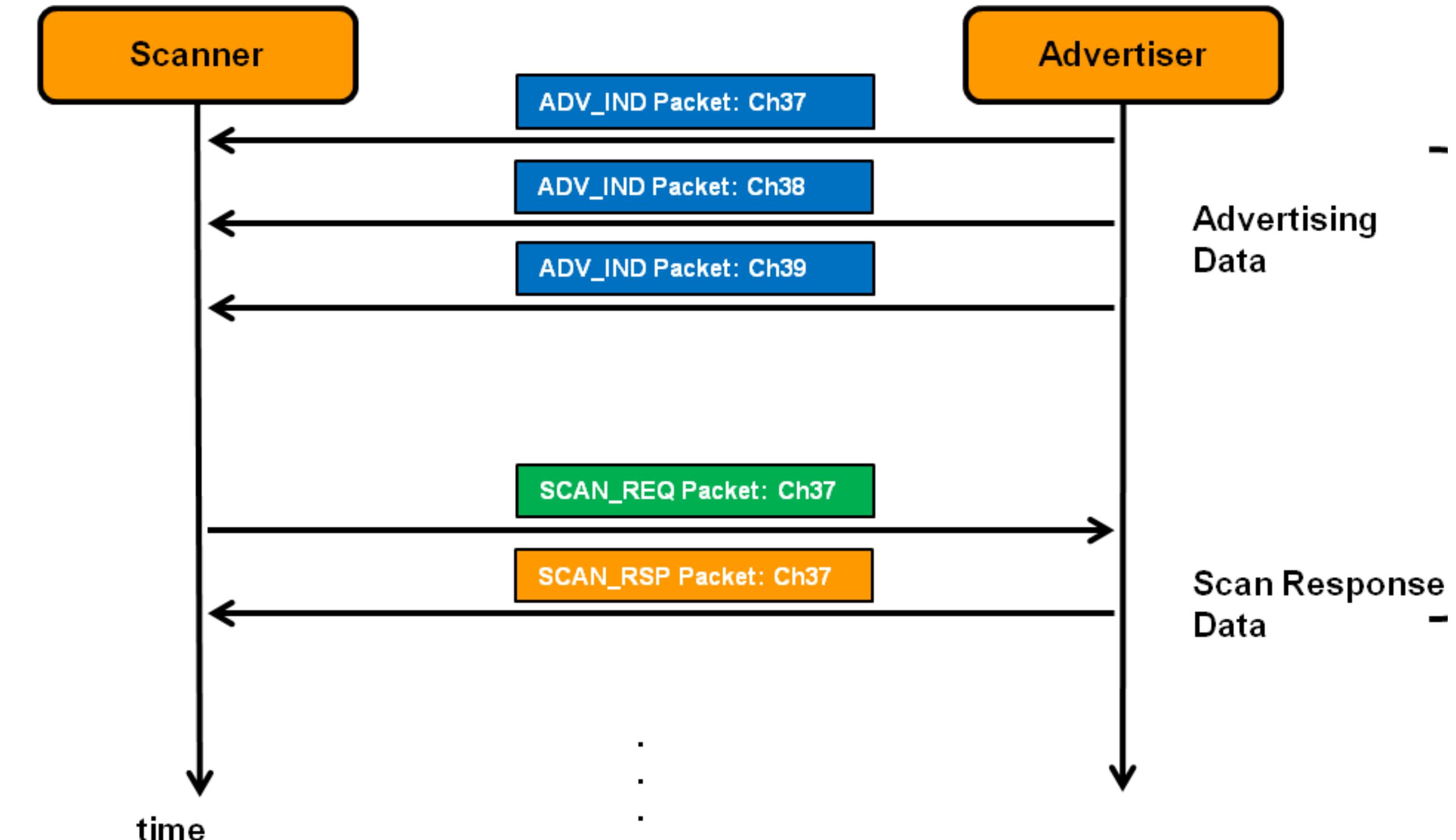


Background

Passive Scanning 🧘



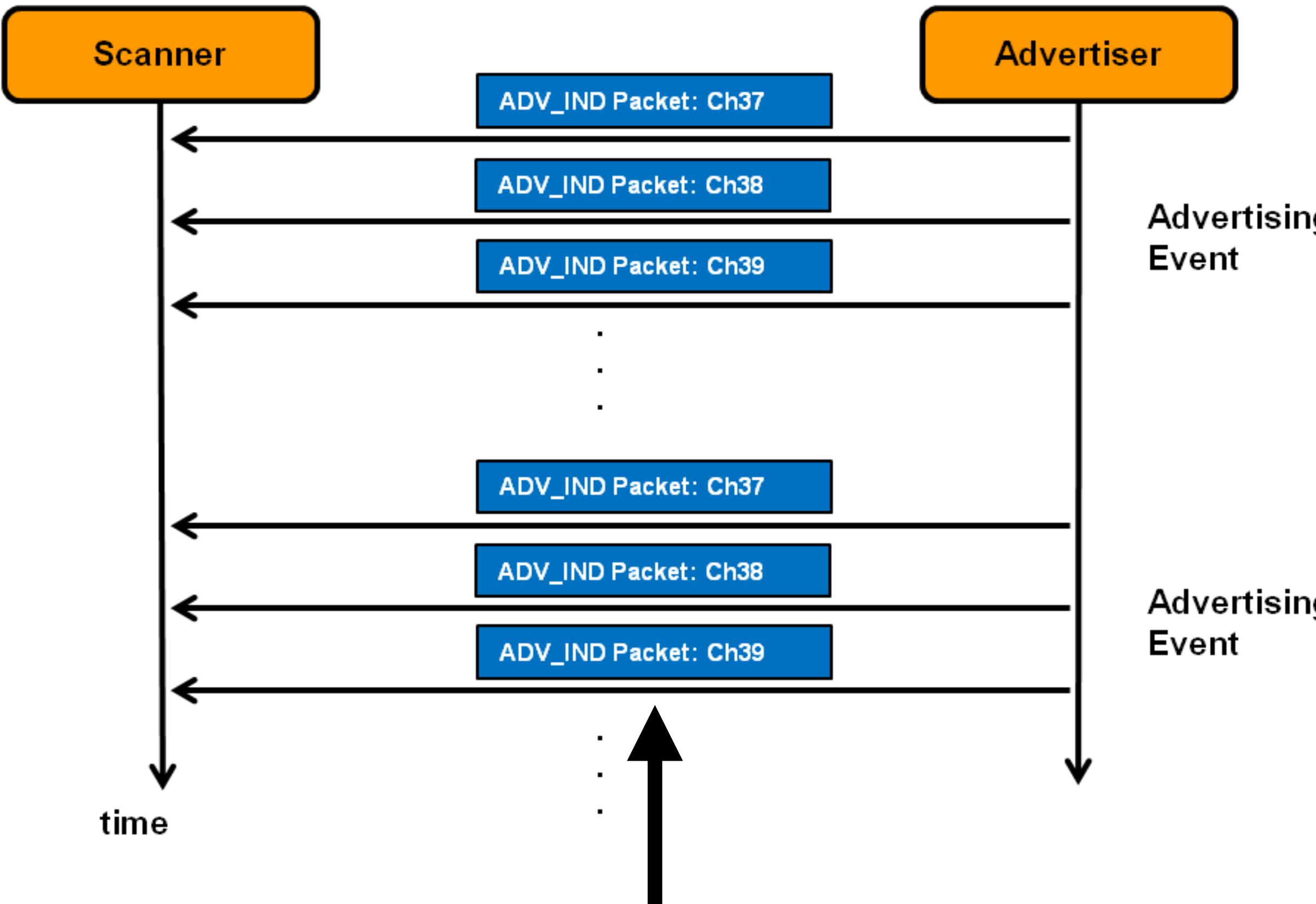
"Mostly-passive" 🚶 Scanning



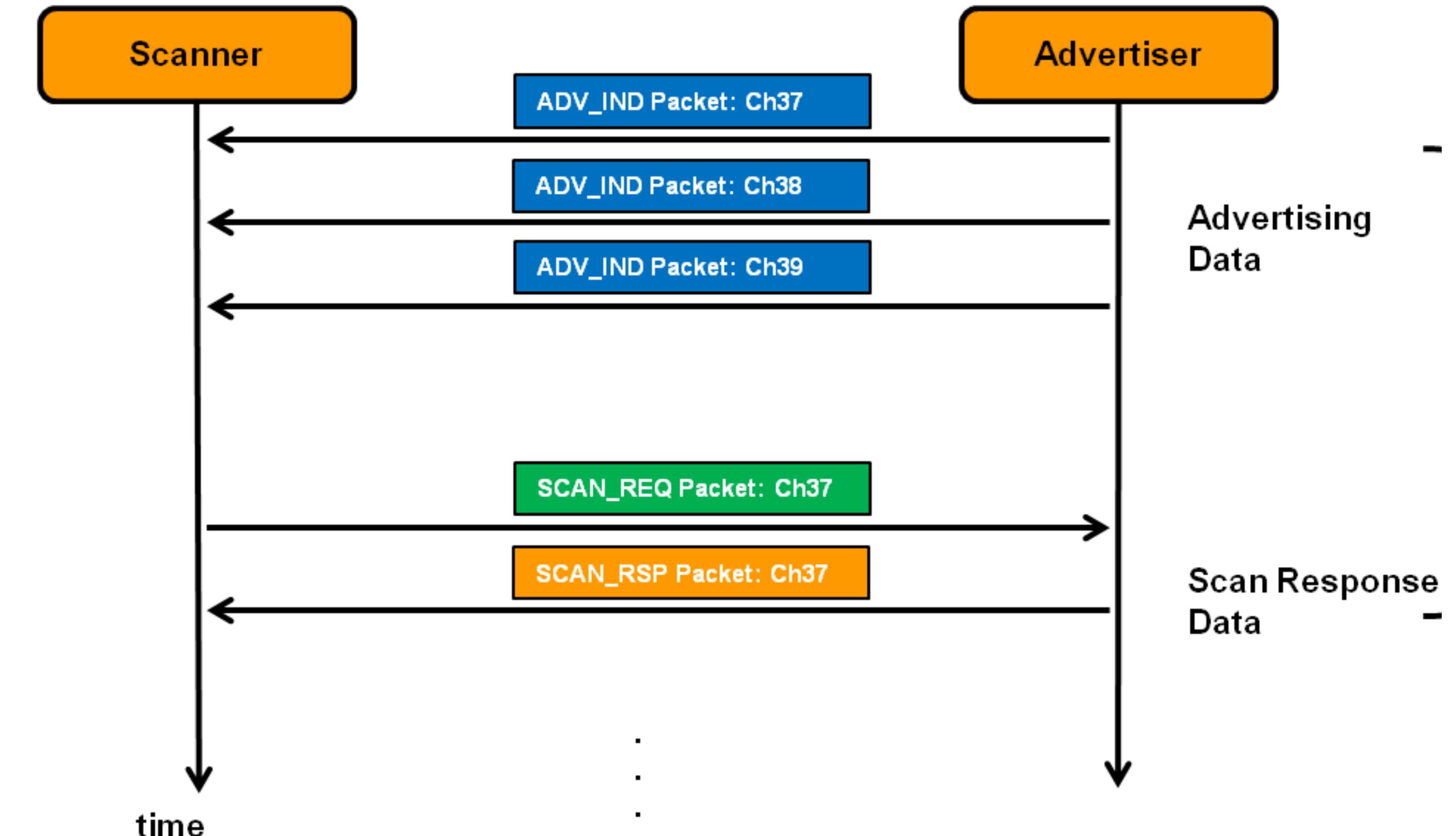


Background

Passive Scanning 🧘



"Mostly-passive" 🚶 Scanning

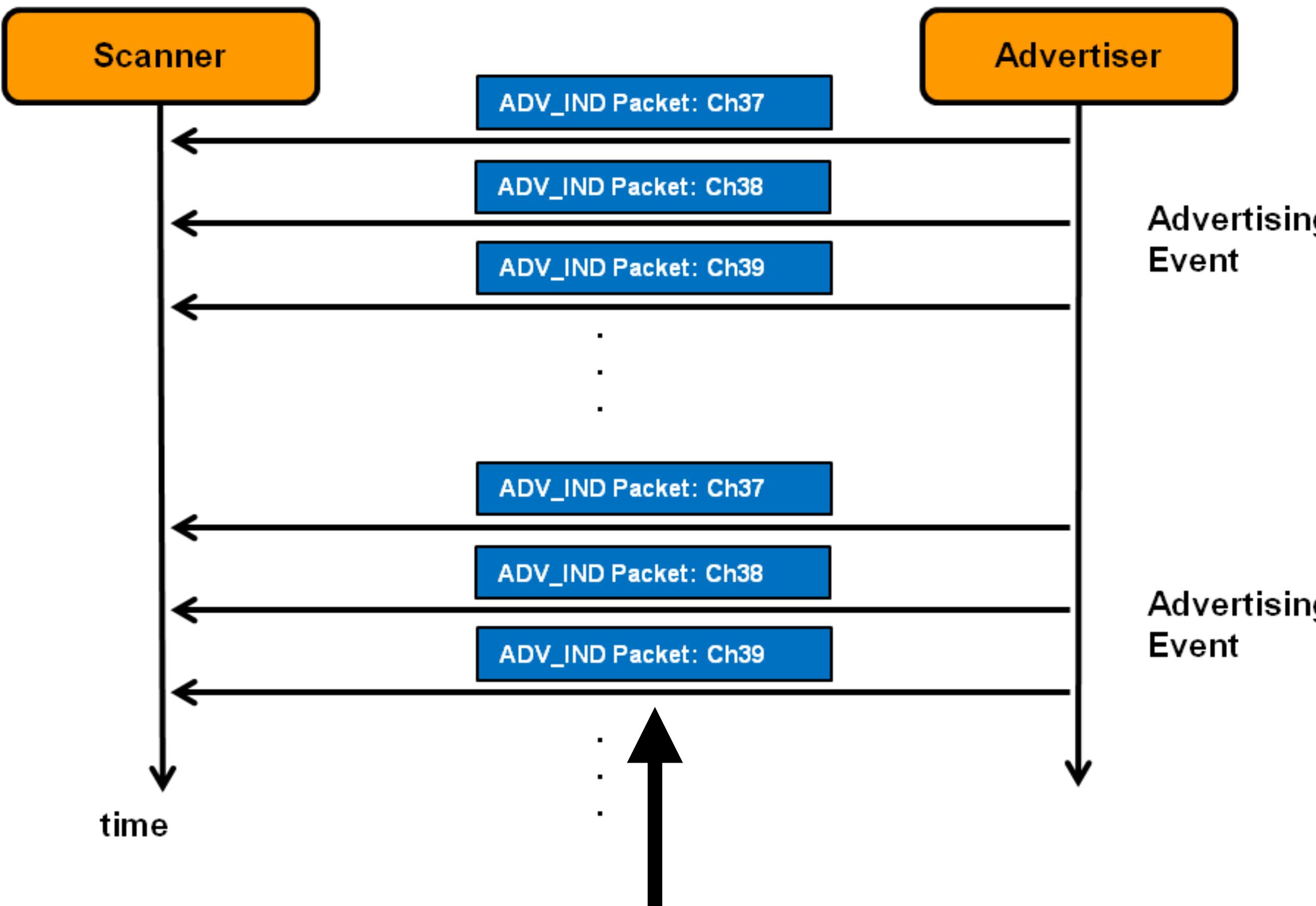


Sometimes the name will be here 🧘



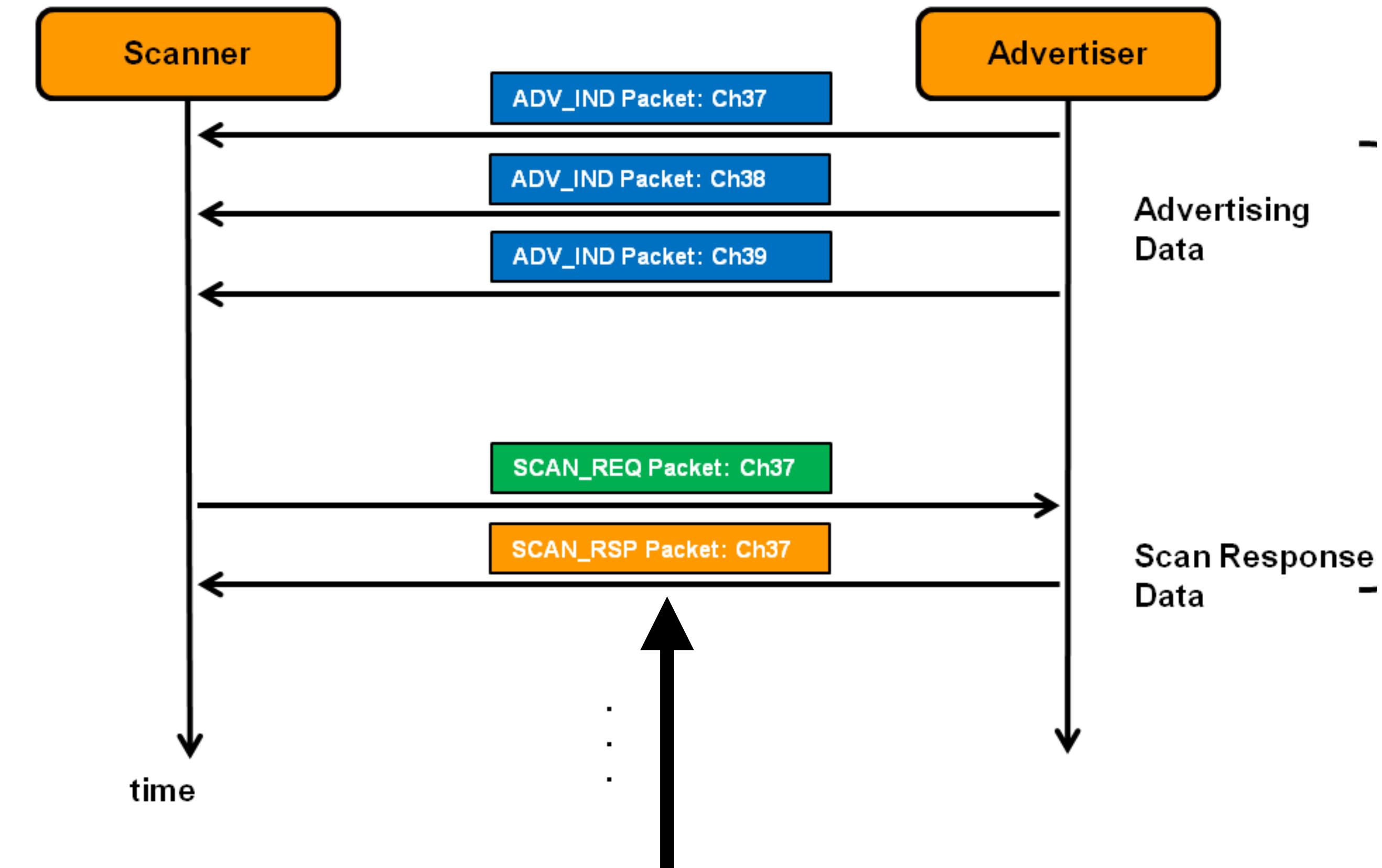
Background

Passive Scanning 🧘



Sometimes the name will be here 🧘

"Mostly-passive" 🚶 Scanning



Other times the OS will ask for it 🚶 ,
and it comes back in a SCAN_RSP



BT Name Behavior: One-to-One

One device == one *unique* name && one *unique* BDADDR



BT Name Behavior: One-to-One

One device == one *unique* name && one *unique* BDADDR

Complete BDADDR inclusion:

oura_A038F8445ED7 → a0:38:f8:44:5e:d7 (public address)

N0KE3K_C8D8DCF760F6 → c8:d8:dc:f7:60:f6 (random static)



BT Name Behavior: One-to-One

One device == one *unique* name && one *unique* BDADDR

Complete BDADDR inclusion:

oura_A038F8445ED7 → a0:38:f8:44:5e:d7 (public address)

N0KE3K_C8D8DCF760E6 → c8:d8:dc:f7:60:f6 (random static)

Partial BDADDR inclusion:

Galaxy Fit2 (987C) → 10:39:17:36:98:7c (public address)

Xiaomi Smart Band 7 34C1 → c6:05:ea:95:34:c1 (random static)



BT Name Behavior: One-to-One

One device == one *unique* name && one *unique* BDADDR

Complete BDADDR inclusion:

oura_A038F8445ED7 → a0:38:f8:44:5e:d7 (public address)

N0KE3K_C8D8DCF760E6 → c8:d8:dc:f7:60:f6 (random static)

Partial BDADDR inclusion:

Galaxy Fit2 (987C) → 10:39:17:36:98:7c (public address)

Xiaomi Smart Band 7 34C1 → c6:05:ea:95:34:c1 (random static)

(Presumed) serial number inclusion

TW370_TIA00414 → 4c:36:4e:4c:57:2a (public address)

CATBTNT-04 DKS02390 → 00:81:f9:7e:37:a6 (public address)



BT Name Behavior: One-to-One

One device == one *unique* name && one *unique* BDADDR



BT Name Behavior: One-to-One

One device == one *unique* name && one *unique* BDADDR

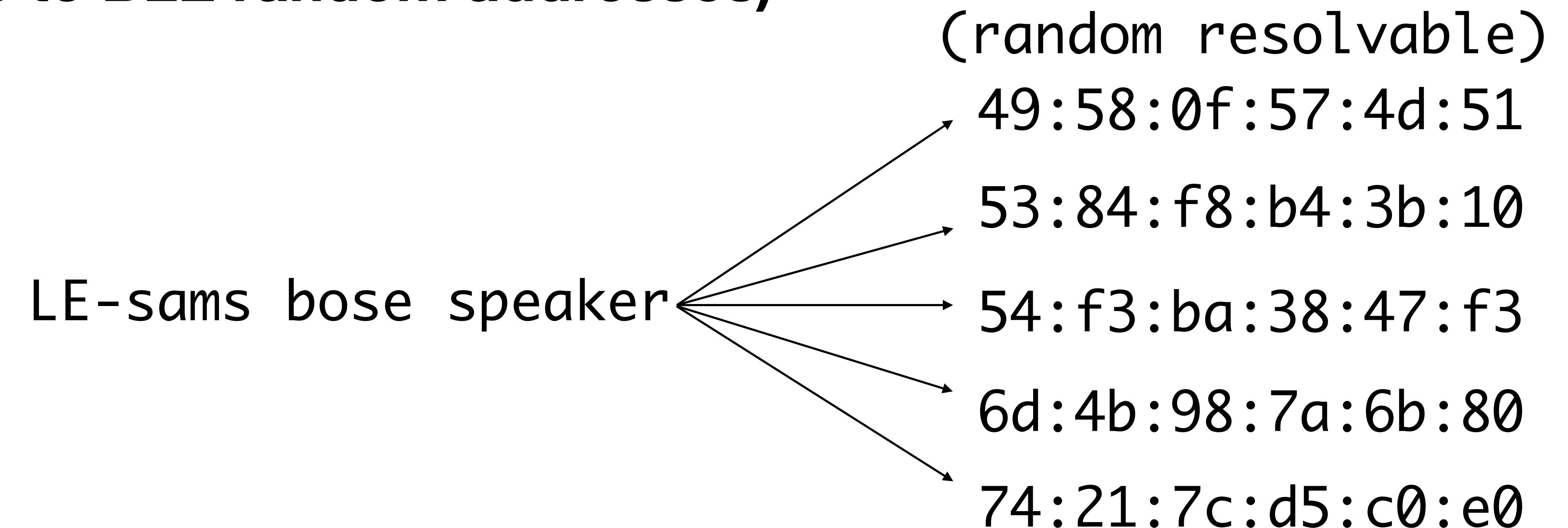
Possible partial BDADDR disclosure?:

Galaxy Watch Active2(C898) LE → c0:b9:b5:02:10:31 (random static)



BT Name Behavior: One-to-Many

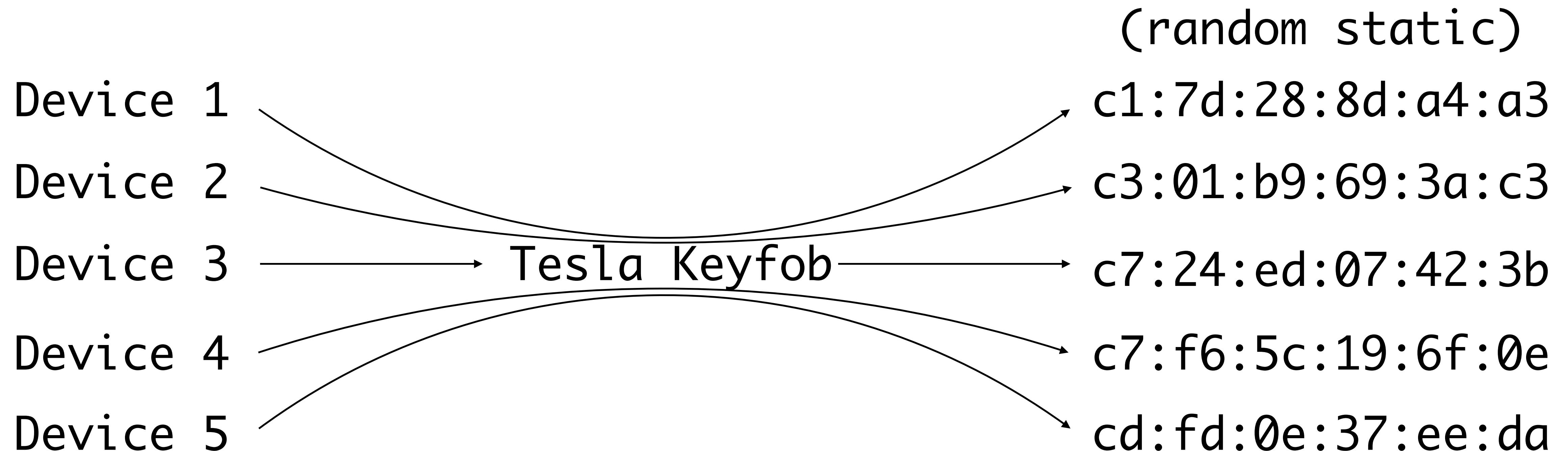
One device == one *unique* name && many BDADDRs
(exclusive to BLE random addresses)





BT Name Behavior: Many-to-Many v1

Many devices, one *shared* name, one *unique* BDADDR per device

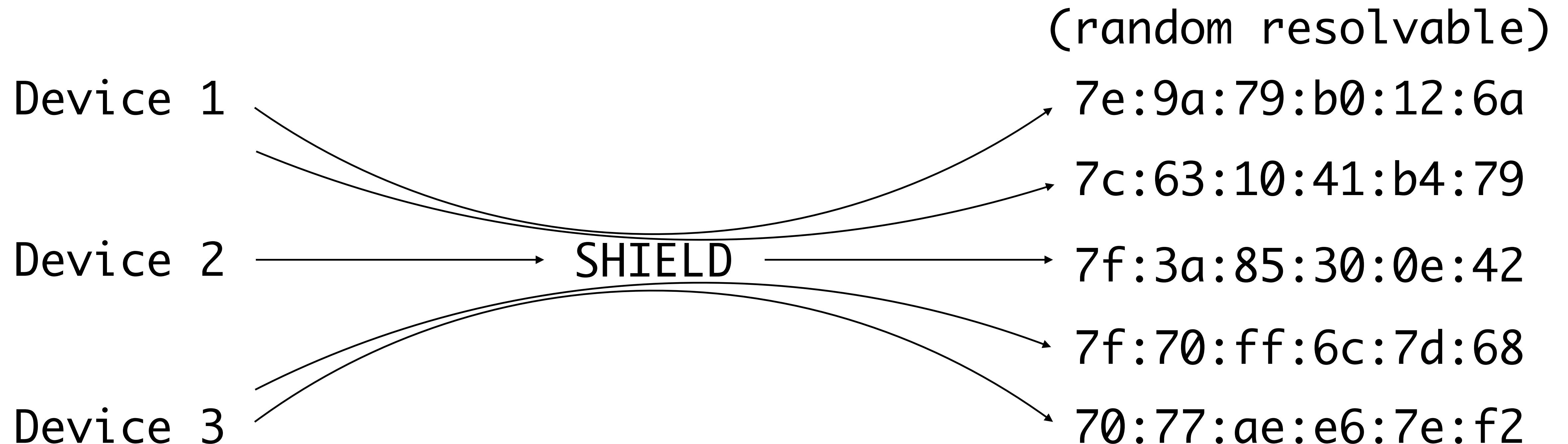


Example: Tesla Keyfob



BT Name Behavior: Many-to-Many v2

Many devices, one *shared* name, *many* BDADDR per device



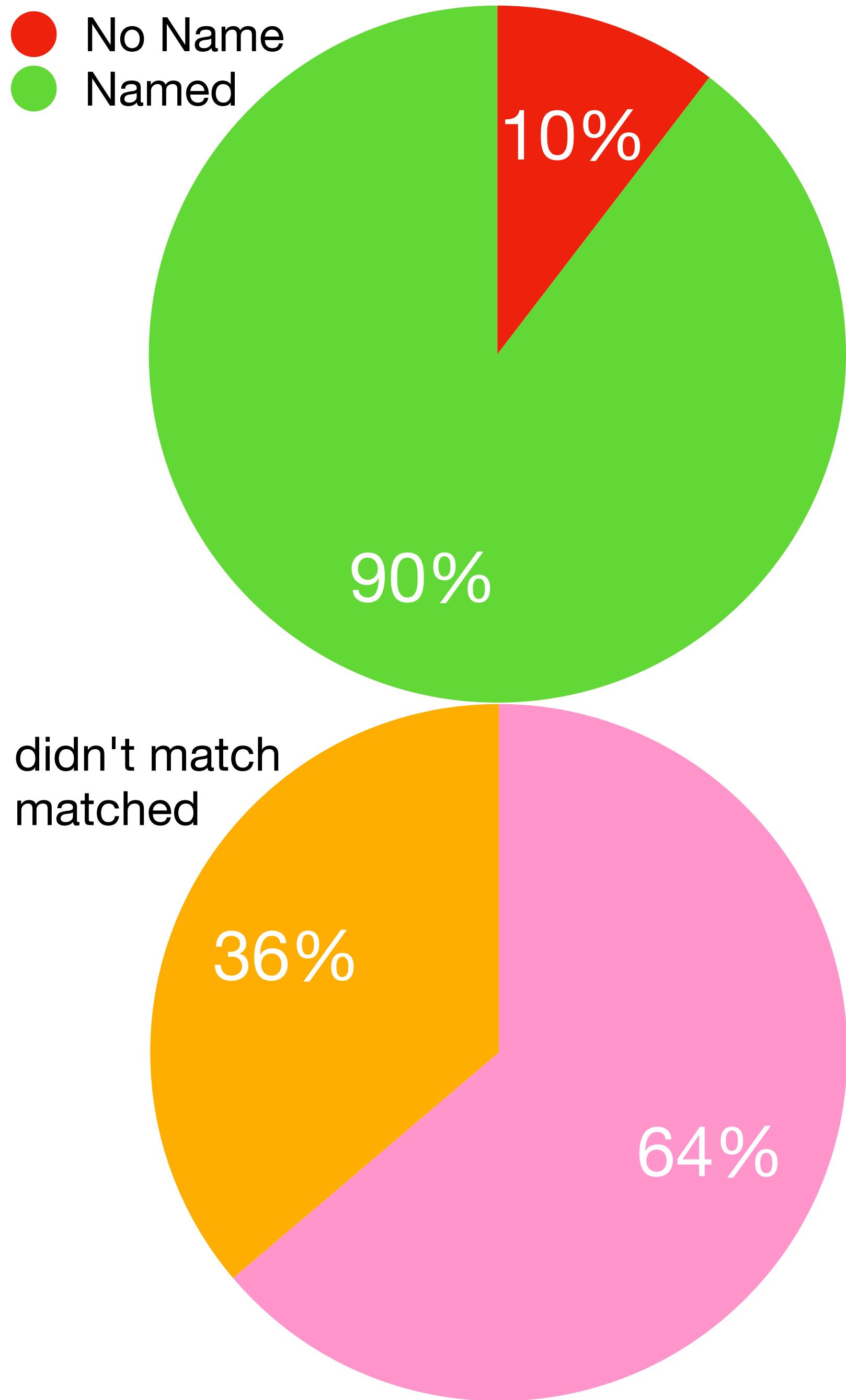
Example: Nvidia SHIELD



NamePrint Data

BTC as of 2023-10-26

- I wrote 1,447 regex "NamePrints"
 - 557 NamePrints matched on BTC data
- 65,451 *unique* BT Classic BDADDRs with a name
 - 65,744 *unique* {name:BDADDR} pairs
 - 23,803 NamePrint matches
 - $23,803 / 65,744 = 36\%$ of all BTC data
 - 33,368 *unique names* in BTC data

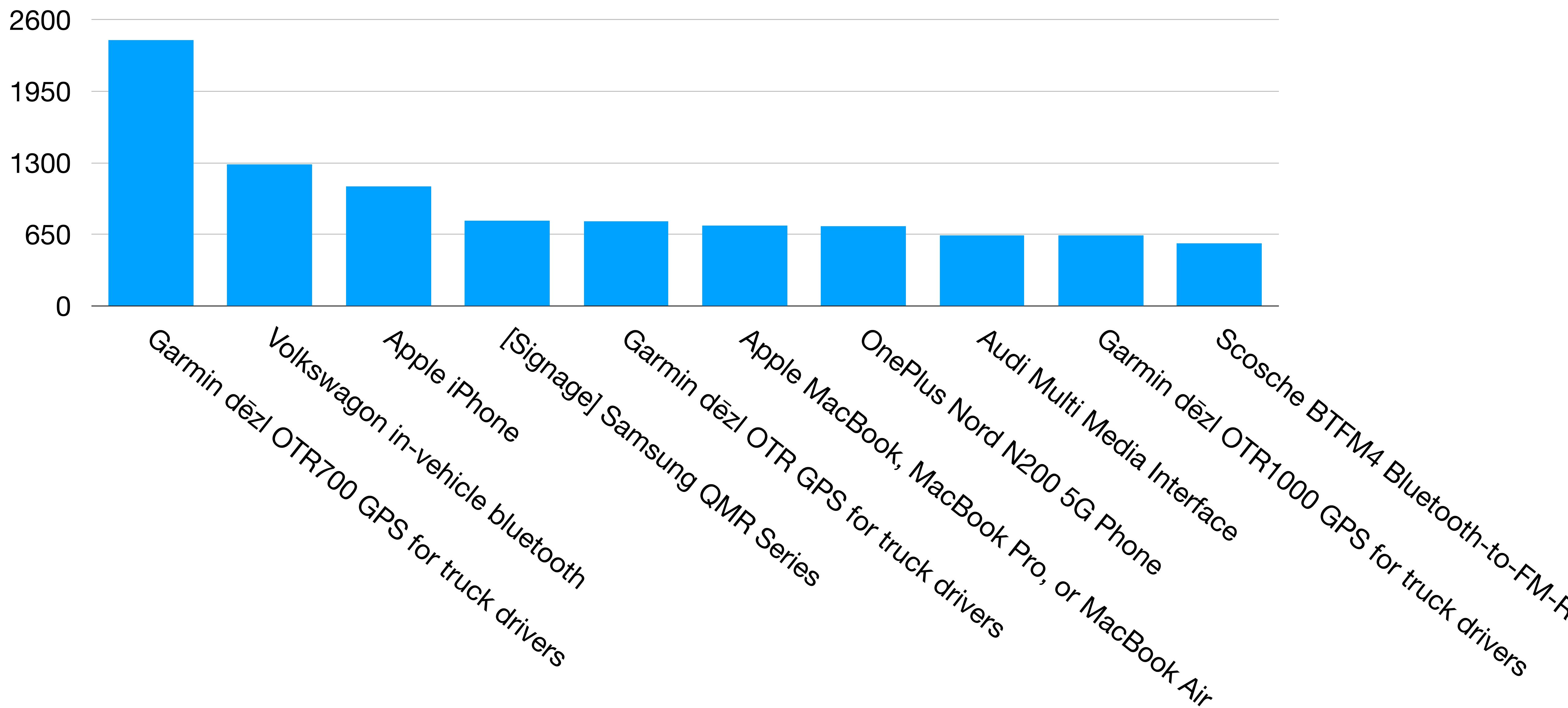


How can there be more bdaddr:name pairs than bdaddrs? Multiple names per bdaddr! (Often due to receiving corrupt name data.)

How can there be more names than matches? Because NamePrints are regexes, one NamePrint matches multiple names



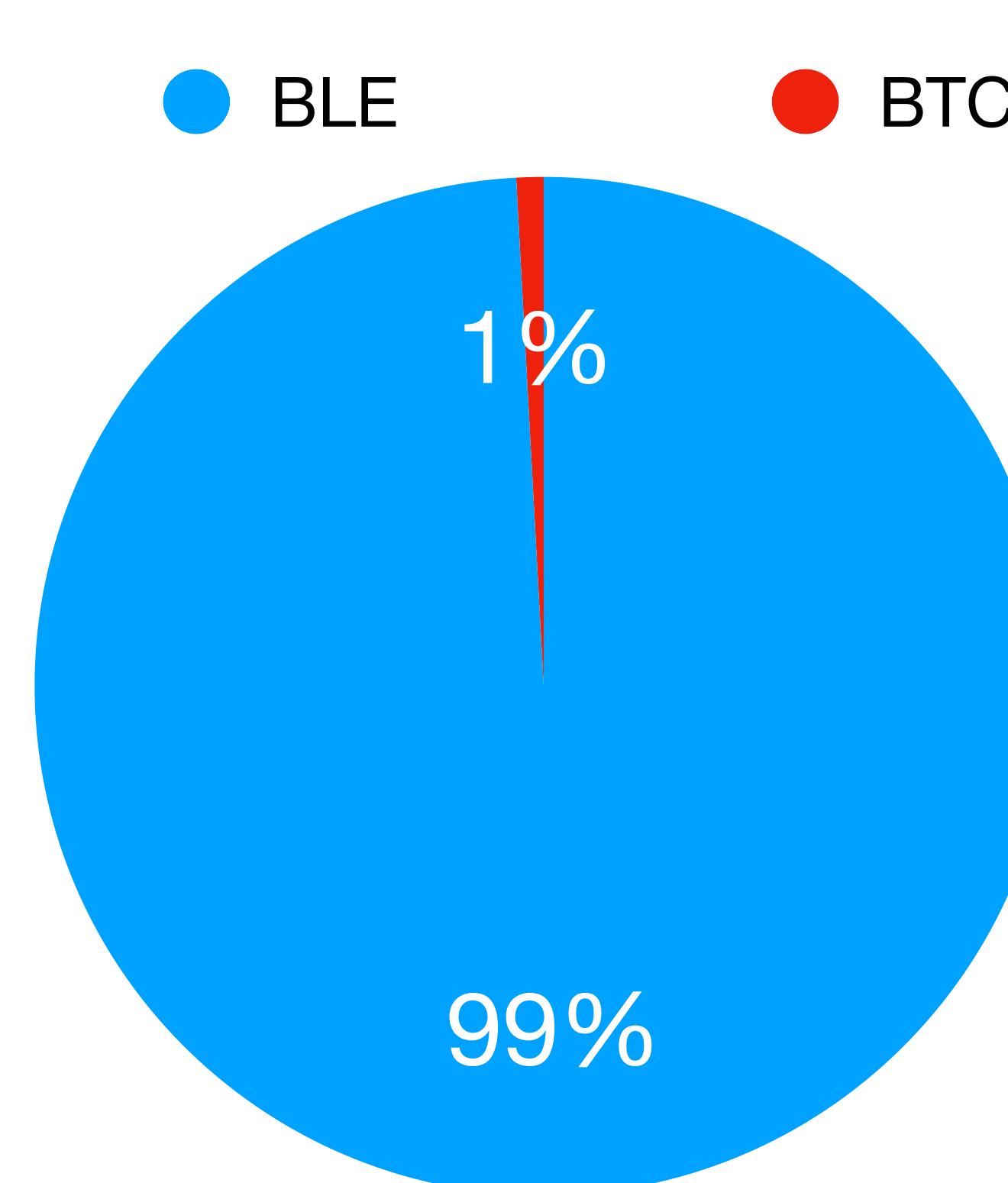
Top 10 BTC Matches





Summary of BTC NamePrint % applicability

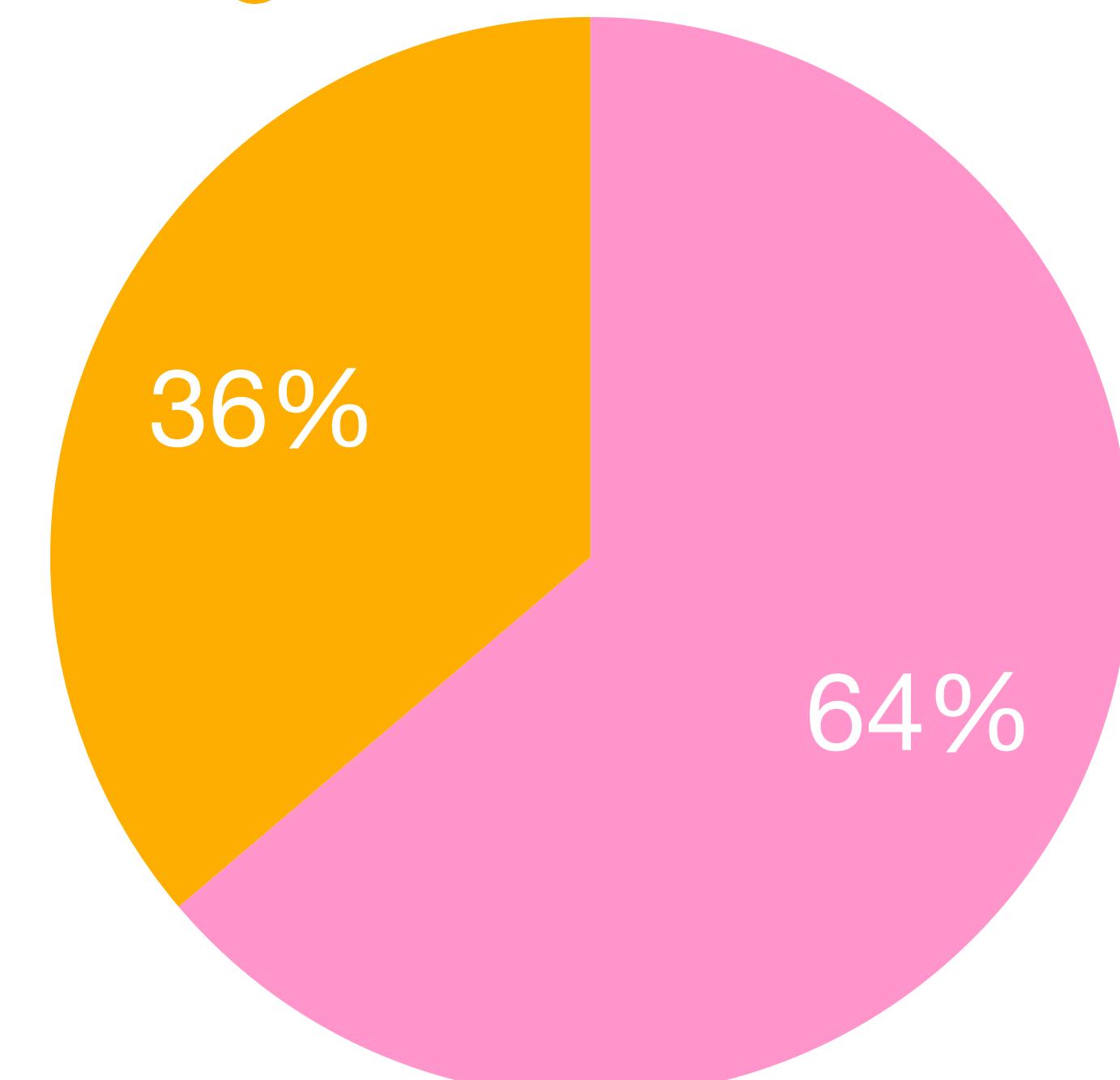
- 36% of 90% of 1% (.324%) of all my data is BTC with a name that matches a NamePrint



● BTC No Name ● BTC Named



● Name didn't match
● Name matched

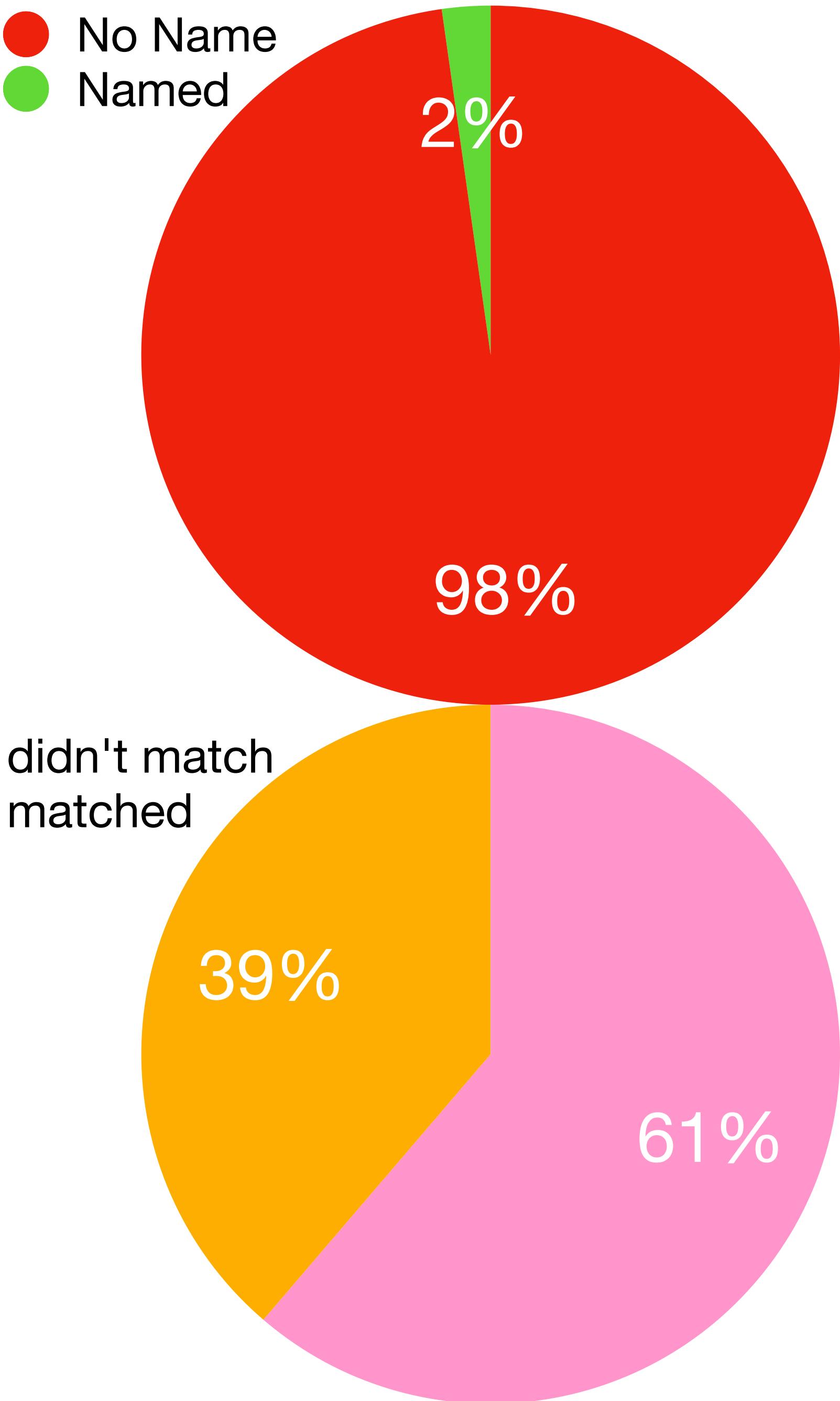




NamePrint Data

BLE as of 2023-10-26

- Made 1447 regex "NamePrints"
 - 912 NamePrints matched on BLE data
- 188,059 *unique* BLE BDADDRs with a name
 - 195,954 *unique* name:BDADDR pairs
 - 123,865 matches
 - 39% of named data, 1.7% of all BTC data
 - 18,959 *unique names*
 - So 540 regexes match $18959/37939 \sim 50\%$ of the names

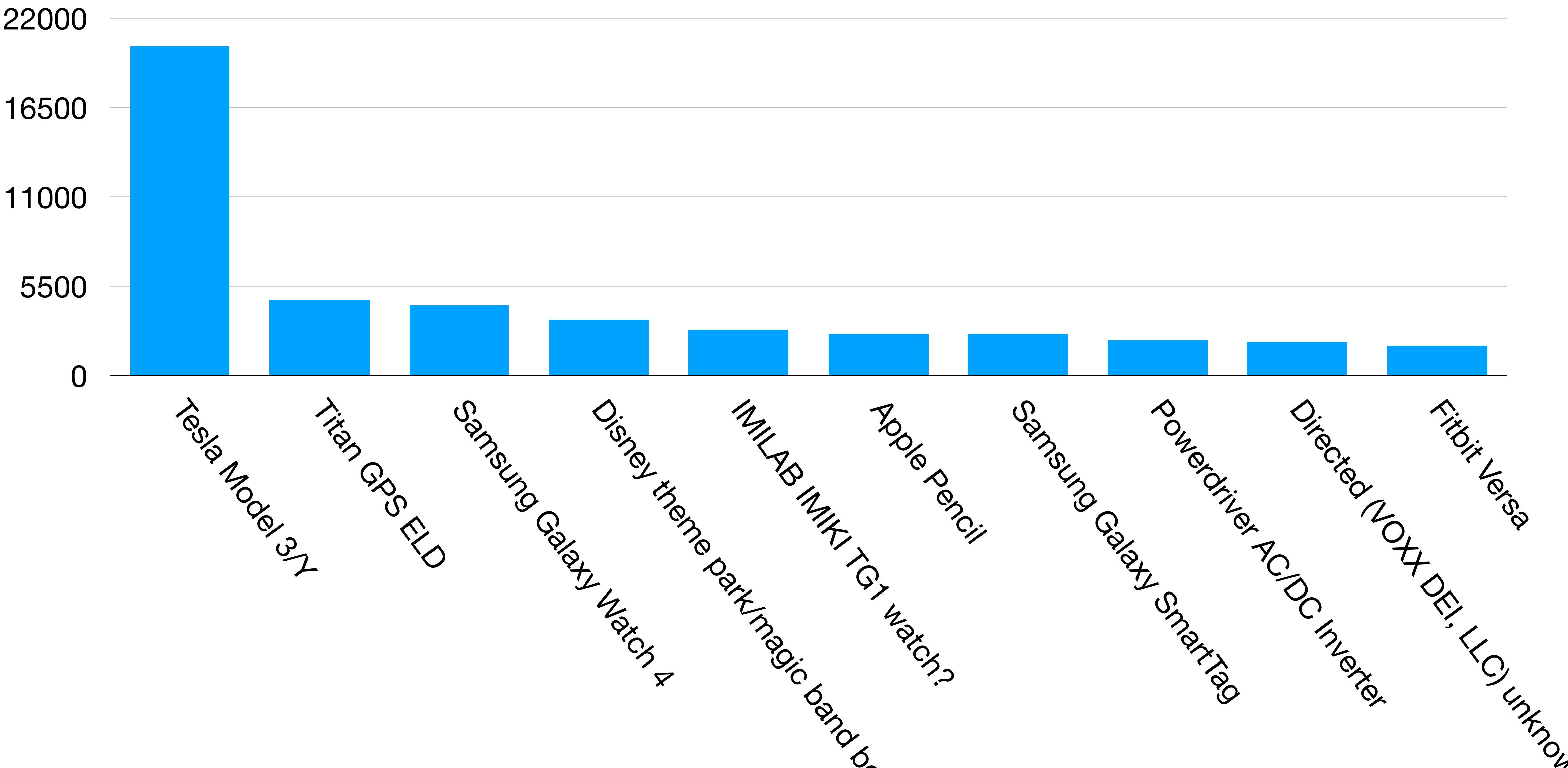


How can there be more bdaddr:name pairs than bdaddrs? Multiple names per bdaddr! (Often due to receiving corrupt name data.)

How can there be more names than matches? Because NamePrints are regexes, one NamePrint matches multiple names



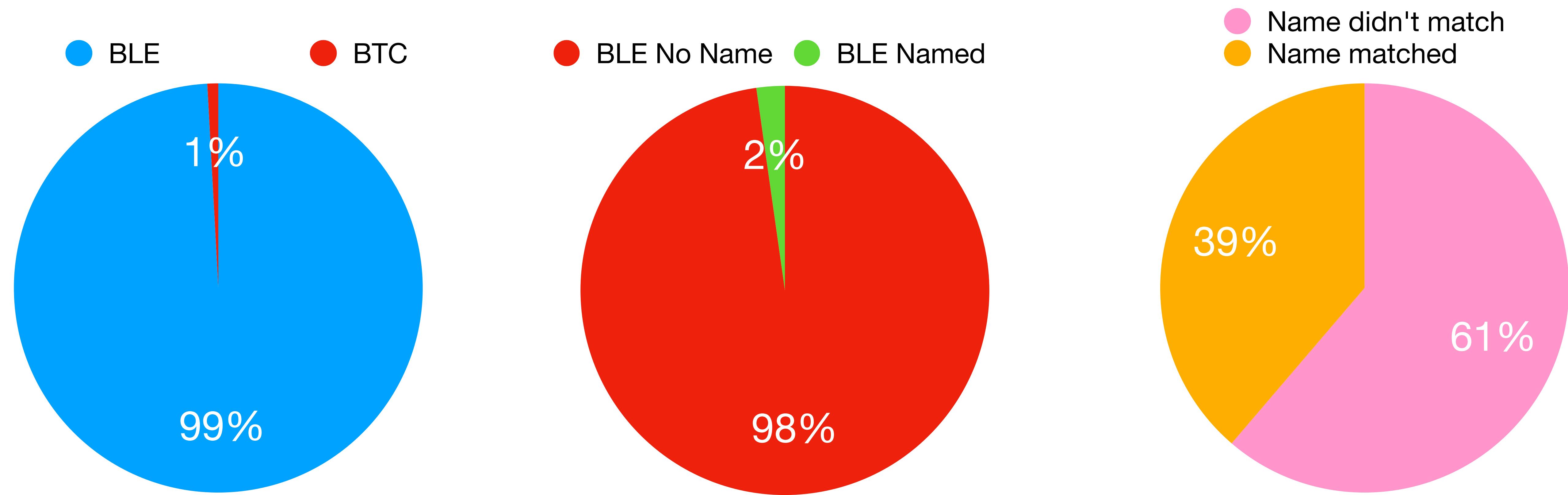
Top 10 BLE Matches





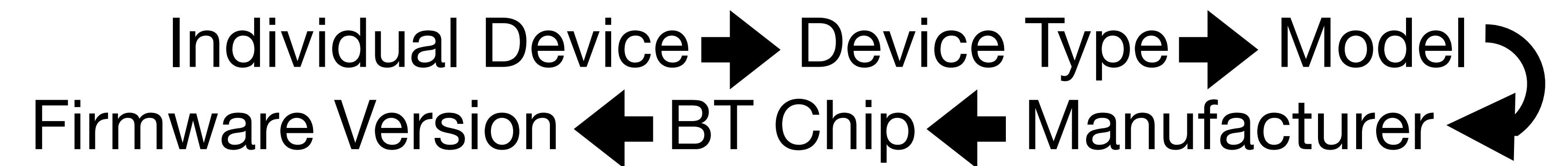
Summary of BTC NamePrint % applicability

- 39% of 2% of 99% (7.128%) of all my data is BLE with a name that matches a NamePrint





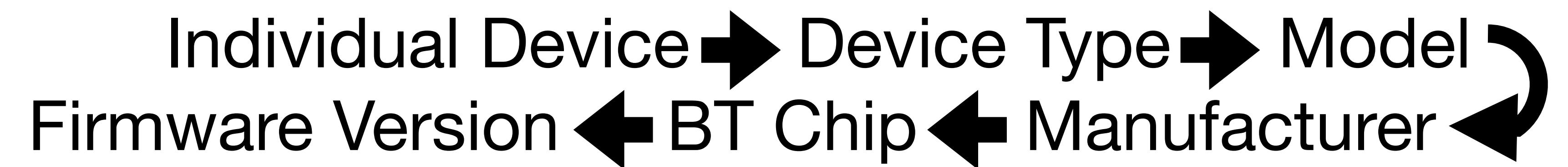
What I Want





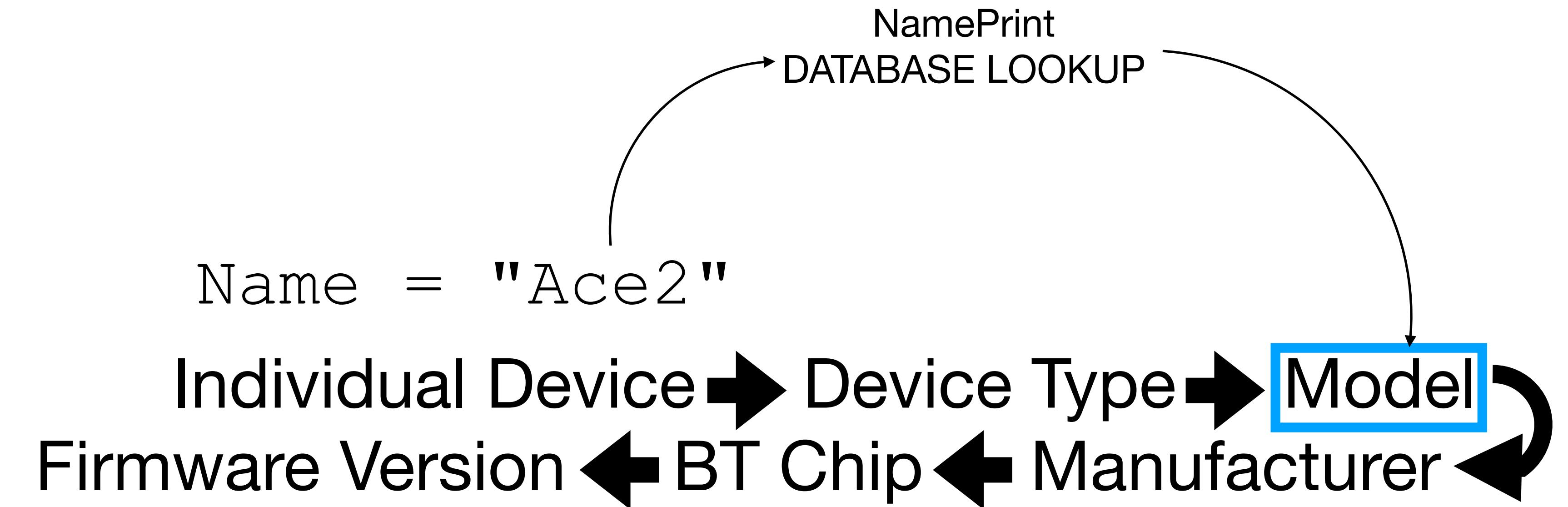
What I Want

Name = "Ace2"



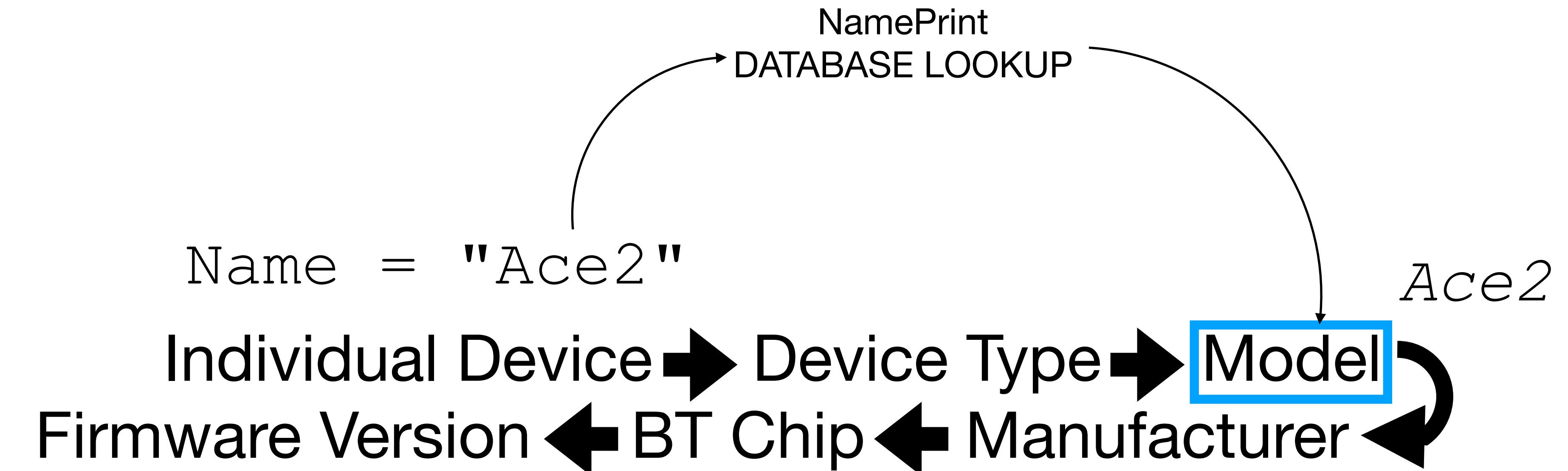


What I Want



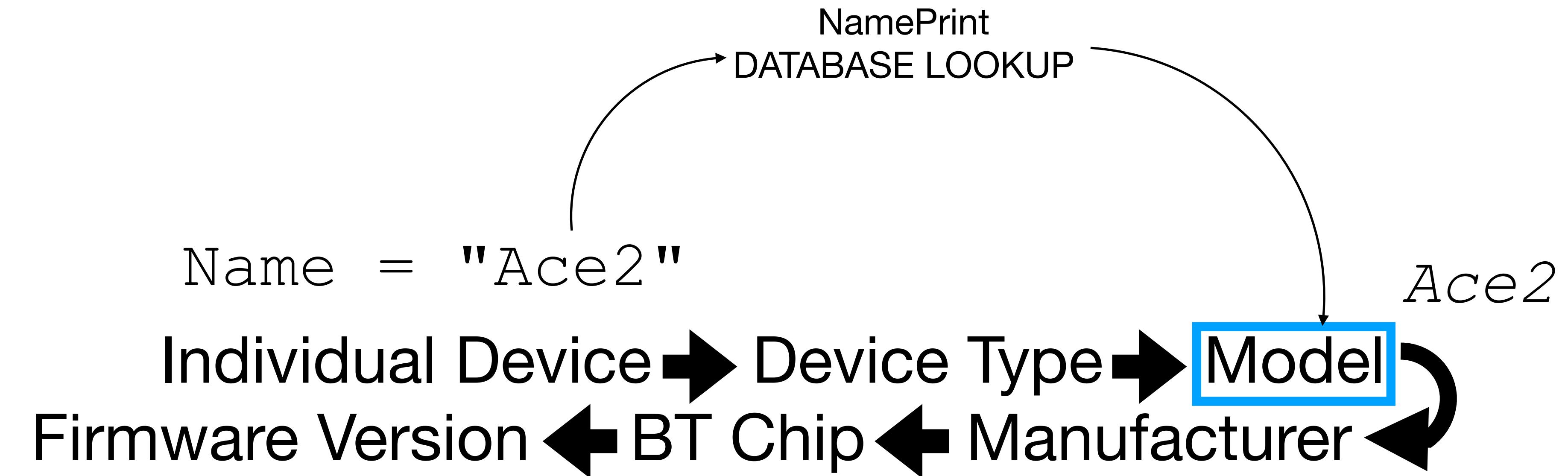


What I Want





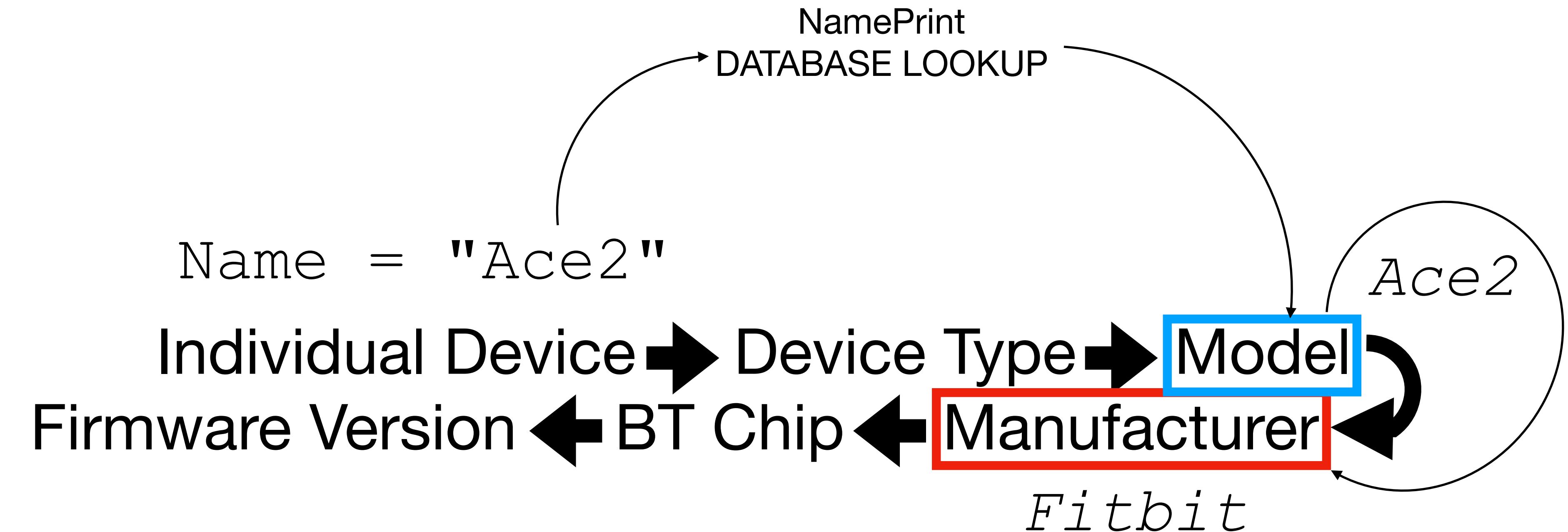
What I Want



ASSUMPTION:
Identifying a Model also allows
the identification of the Manufacturer & Device Type



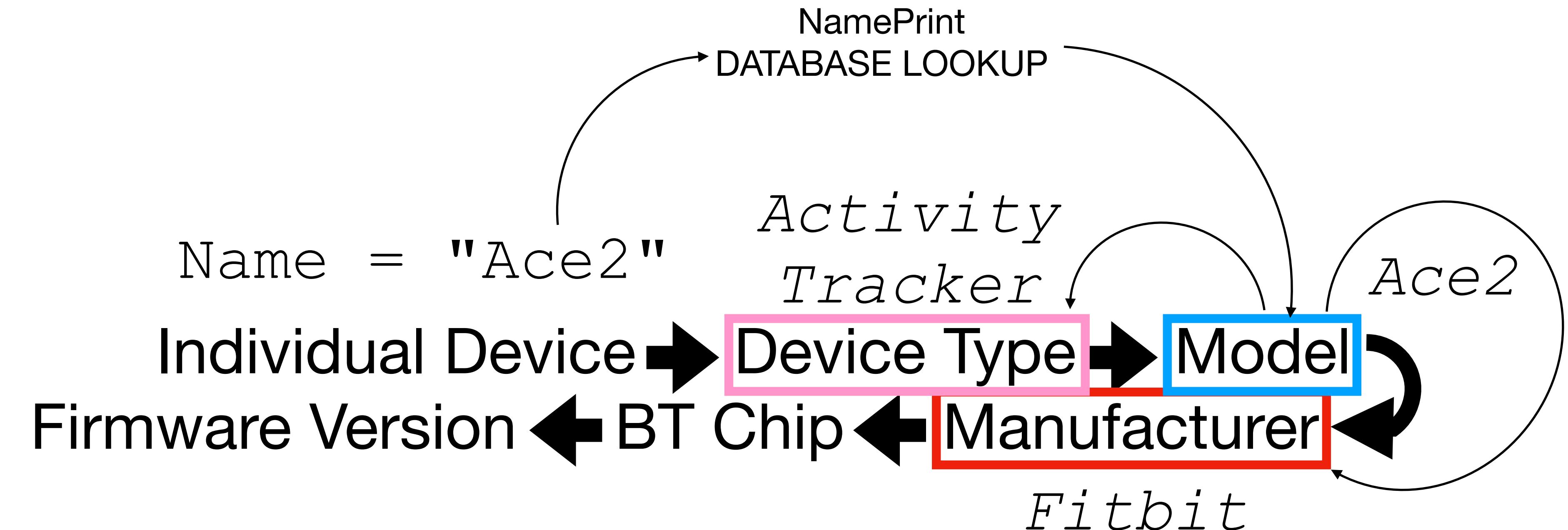
What I Want



ASSUMPTION:
Identifying a Model also allows
the identification of the Manufacturer & Device Type



What I Want



ASSUMPTION:
Identifying a Model also allows
the identification of the Manufacturer & Device Type



Occasionally NamePrint -> ChipPrint

- *Very rarely (as in, I've only seen this once so far ;)),* the vendor will make a big deal out of what chip it uses, and then you can quickly get a ChipPrint directly from the NamePrint!
- NamePrint="⁸FiiO BTR5\$",ChipPrint="CSR8675"



BTR5

Flagship Portable High-Fidelity Bluetooth Amplifier

High Performance DAC ES9218P*2

Flagship Bluetooth chip CSR8675

FPGA clock management, dual independent crystal oscillators

Bluetooth 5.0 with full format support

Independent control chip XMOS XUF208

USB DAC supporting up to 384kHz/DSD256 native

Double-sided 2.5D glass with OLED display

3.5mm+2.5mm headphone outputs

Intelligent control with FiiO Music app

One-touch NFC pairing





BTR5

Flagship Portable High-Fidelity Bluetooth Amplifier

High Performance DAC ES9218P*2

Flagship Bluetooth chip CSR8675

FPGA clock management, dual independent crystal oscillators

Bluetooth 5.0 with full format support

Independent control chip XMOS XUF208

USB DAC supporting up to 384kHz/DSD256 native

Double-sided 2.5D glass with OLED display

3.5mm+2.5mm headphone outputs

Intelligent control with FiiO Music app

One-touch NFC pairing



2thprint by Device ID Profile





BTC EIR Device ID

- Some devices seem to include Device ID information as shown here in BTC Extended Inquiry Response packets
 - Despite the fact that I can't find anything in the spec saying that's supposed to be a thing? (The "Supplement to the Bluetooth Core Specification" doesn't list it as valid...)
 - If VendorIDSource = 1, the VendorID is looked up in the Bluetooth assigned company IDs. If it's 2, it's looked up in the USB assigned company IDs

5.1.10 Device Identification Profile (DID)

Applicable to Service Class UUIDs:

- PnPInformation: 0x1200

Last Modified: 2023-02-02

Attribute ID	Attribute Name
0x0200	SpecificationID
0x0201	VendorID
0x0202	ProductID
0x0203	Version
0x0204	PrimaryRecord
0x0205	VendorIDSource



BTC EIR Device ID

- Some devices seem to include Device ID information as shown here in BTC Extended Inquiry Response packets
- Despite the fact that I can't find anything in the spec saying that's supposed to be a thing? (The "Supplement to the Bluetooth Core Specification" doesn't list it as valid...)
- If VendorIDSource = 1, the VendorID is looked up in the Bluetooth assigned company IDs. If it's 2, it's looked up in the USB assigned company IDs

▼ Extended Inquiry Response Data
 ▼ Device Name: SD-7000T_96
 Length: 12
 Type: Device Name (0x09)
 Device Name: SD-7000T_96
 ▼ Tx Power Level
 Length: 2
 Type: Tx Power Level (0x0a)
 Power Level (dBm): 6
 ▼ Device ID / Security Manager TK Value
 Length: 9
 Type: Device ID / Security Manager TK Value (0x10)
 Vendor ID Source: USB Implementer's Forum (0x0002)
 Vendor ID: Linux Foundation (0x1d6b)
 Product ID: 0x0246 (Unknown)
 Version: 0x053c

Last Modified: 2023-02-02

Attribute ID	Attribute Name
0x0200	SpecificationID
0x0201	VendorID
0x0202	ProductID
0x0203	Version
0x0204	PrimaryRecord
0x0205	VendorIDSource



BTC EIR Device ID

Product: SD-7000T1

User Manual



- Some devices seem to include Device ID information as shown here in BTC Extended Inquiry Response packets
- Despite the fact that I can't find anything in the spec saying that's supposed to be a thing? (The "Supplement to the Bluetooth Core Specification" doesn't list it as valid...)
- If VendorIDSource = 1, the VendorID is looked up in the Bluetooth assigned company IDs. If it's 2, it's looked up in the USB assigned company IDs

▼ Extended Inquiry Response Data
 ▼ Device Name: SD-7000T_96
 Length: 12
 Type: Device Name (0x09)
 Device Name: SD-7000T_96
 ▼ Tx Power Level
 Length: 2
 Type: Tx Power Level (0x0a)
 Power Level (dBm): 6
 ▼ Device ID / Security Manager TK Value
 Length: 9
 Type: Device ID / Security Manager TK Value (0x10)
 Vendor ID Source: USB Implementer's Forum (0x0002)
 Vendor ID: Linux Foundation (0x1d6b)
 Product ID: 0x0246 (Unknown)
 Version: 0x053c

Last Modified: 2023-02-02

Attribute ID	Attribute Name
0x0200	SpecificationID
0x0201	VendorID
0x0202	ProductID
0x0203	Version
0x0204	PrimaryRecord
0x0205	VendorIDSource



Top 20 Vendors for BTC Device ID data

BTC - 2023-10-26

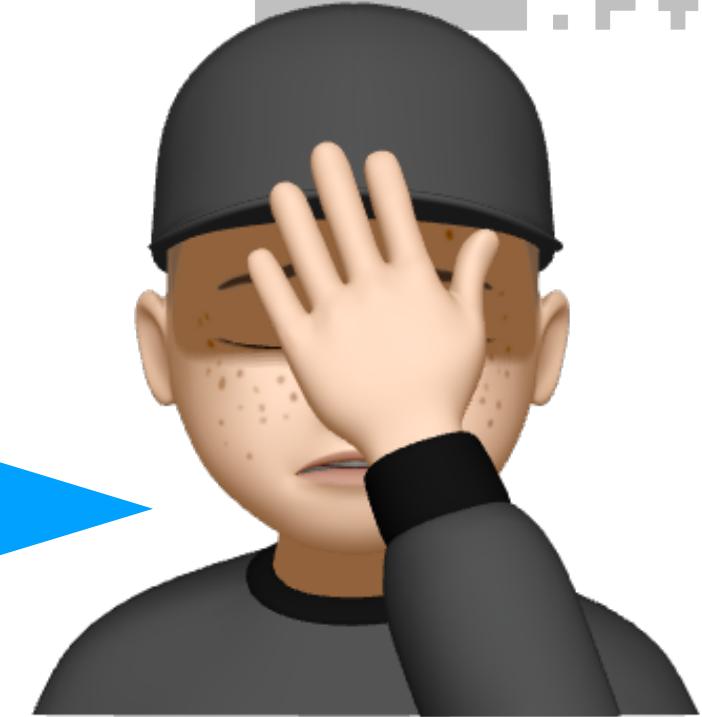
vendor_source	vendor_id_hex	company_name	frequency	
Bluetooth	0x4E8	STABIL0 International	4462	<- actually Samsung
USB	0x1D6B	Linux Foundation	433	
Bluetooth	0x0	Ericsson AB	95	
USB	0x44E	Alps Electric Co., Ltd	85	🔧
Bluetooth	0xA	Qualcomm Technologies International, Ltd. (QTI)	62	🍪
Bluetooth	0x75	Samsung Electronics Co. Ltd.	60	
USB	0xA12	Cambridge Silicon Radio, Ltd	58	🍪
Bluetooth	0x4C	Apple, Inc.	23	🍪
Bluetooth	0x3E0	Actions (Zhuhai) Technology Co., Limited	21	🍪
USB	0x108C	Robert Bosch GmbH	16	
992	0xFFFF	NULL	15	
Bluetooth	0x9E	Bose Corporation	12	
Bluetooth	0x474C	NULL	12	
USB	0x54C	Sony Corp.	5	
Bluetooth	0x850	Yealink (Xiamen) Network Technology Co., LTD	5	
256	0x4E8	NULL	5	
USB	0xA9	NULL	4	
Bluetooth	0x418	Reserved	4	
USB	0xA	NULL	4	
Bluetooth	0x103	Bang & Olufsen A/S	4	



Top 20 Vendors for BTC Device ID data

BTC - 2023-10-26

Samsung's setting their vendor source to Bluetooth but then using their USB ID



vendor_source	vendor_id_hex	company_name	frequency	
Bluetooth	0x4E8	STABIL0 International	4462	<- actually Samsung
USB	0x1D6B	Linux Foundation	433	
Bluetooth	0x0	Ericsson AB	95	
USB	0x44E	Alps Electric Co., Ltd	85	🔧
Bluetooth	0xA	Qualcomm Technologies International, Ltd. (QTIL)	62	🍪
Bluetooth	0x75	Samsung Electronics Co. Ltd.	60	
USB	0xA12	Cambridge Silicon Radio, Ltd	58	🍪
Bluetooth	0x4C	Apple, Inc.	23	🍪
Bluetooth	0x3E0	Actions (Zhuhai) Technology Co., Limited	21	🍪
USB	0x108C	Robert Bosch GmbH	16	
992	0xFFFF	NULL	15	
Bluetooth	0x9E	Bose Corporation	12	
Bluetooth	0x474C	NULL	12	
USB	0x54C	Sony Corp.	5	
Bluetooth	0x850	Yealink (Xiamen) Network Technology Co., LTD	5	
256	0x4E8	NULL	5	
USB	0xA9	NULL	4	
Bluetooth	0x418	Reserved	4	
USB	0xA	NULL	4	
Bluetooth	0x103	Bang & Olufsen A/S	4	



- This snippet from the 4.0 spec (before they started using the Core Spec Supplement) suggests that Device ID profile information can be included in EIR

8.1 EIR DATA TYPE DEFINITIONS

This section defines the basic EIR data types. Additional EIR data types may be defined in profile specifications.

- Basically all the other data types made their way into the CSS but not this...

7 EIR Transactions to Obtain Device ID Information

If Extended Inquiry Response is supported by a given device that supports the Device ID Profile, then the device may expose the Device ID information in the Extended Inquiry Response when discoverable.

If an implementer chooses to expose a Device ID EIR record, the following Device ID attribute values shall be exposed:

Attribute	Attribute Value Type
VendorIDSource	Uint16
VendorID	Uint16
ProductID	Uint16
Version	Uint16

See section 8.2 for details on the format of the Device ID EIR Record.

8.2 Device ID Extended Inquiry Response

The inclusion of the Device ID EIR Record is optional. However, for each Device ID EIR Record that is included in the EIR record, a corresponding Device ID Service Record (meaning that all attributes that appear in both the Service Record and the EIR Record shall be equal) shall be included in the SDP database. The format of the Device ID EIR Record shall be as shown in Table 8.2.

Value	Notes
0x09	Length of this Data
0xTT	Device ID EIR Tag
0xYYYY	Uint16 Vendor ID Source
0xYYYY	Uint16 VendorID
0xYYYY	Uint16 ProductID
0xYYYY	Uint16 Version

Table 8.2: Device ID Extended Inquiry Response Tags



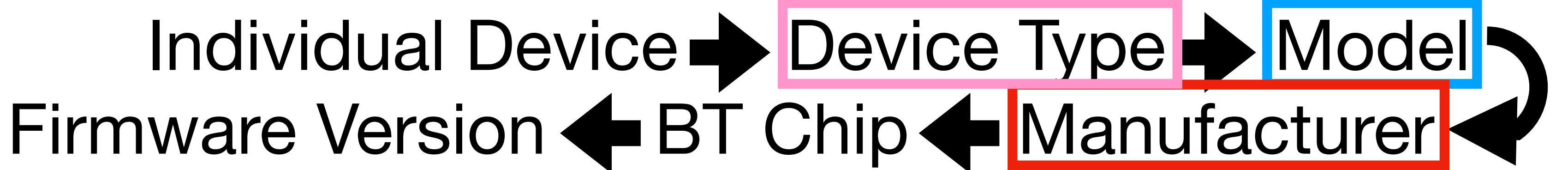
- From assigned numbers doc

Common Data Type	Name	Reference
0x01	Flags	Core Specification Supplement, Part A, Section 1.3
0x02	Incomplete List of 16-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x03	Complete List of 16-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x04	Incomplete List of 32-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x05	Complete List of 32-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x06	Incomplete List of 128-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x07	Complete List of 128-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x08	Shortened Local Name	Core Specification Supplement, Part A, Section 1.2
0x09	Complete Local Name	Core Specification Supplement, Part A, Section 1.2
0x0A	Tx Power Level	Core Specification Supplement, Part A, Section 1.5
0x0D	Class of Device	Core Specification Supplement, Part A, Section 1.6
0x0E	Simple Pairing Hash C-192	Core Specification Supplement, Part A, Section 1.6
0x0F	Simple Pairing Randomizer R-192	Core Specification Supplement, Part A, Section 1.6
0x10	Device ID	Device ID Profile
0x10	Security Manager TK Value	Core Specification Supplement, Part A, Section 1.8
0x11	Security Manager Out of Band Flags	Core Specification Supplement, Part A, Section 1.7
0x12	Peripheral Connection Interval Range	Core Specification Supplement, Part A, Section 1.9



- From assigned numbers doc

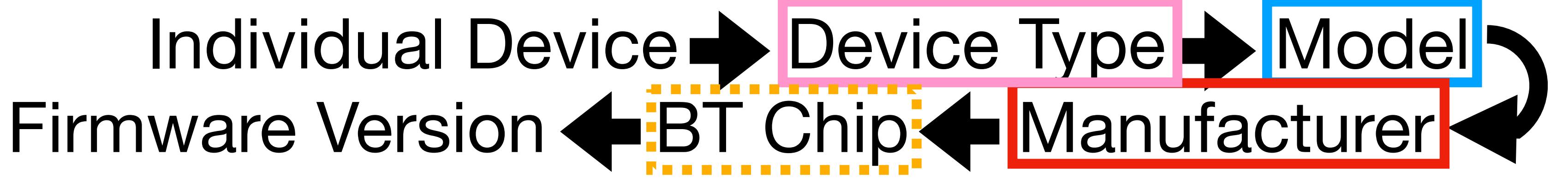
Common Data Type	Name	Reference
0x01	Flags	Core Specification Supplement, Part A, Section 1.3
0x02	Incomplete List of 16-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x03	Complete List of 16-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x04	Incomplete List of 32-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x05	Complete List of 32-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x06	Incomplete List of 128-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x07	Complete List of 128-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x08	Shortened Local Name	Core Specification Supplement, Part A, Section 1.2
0x09	Complete Local Name	Core Specification Supplement, Part A, Section 1.2
0x0A	Tx Power Level	Core Specification Supplement, Part A, Section 1.5
0x0D	Class of Device	Core Specification Supplement, Part A, Section 1.6
0x0E	Simple Pairing Hash C-192	Core Specification Supplement, Part A, Section 1.6
0x0F	Simple Pairing Randomizer R-192	Core Specification Supplement, Part A, Section 1.6
0x10	Device ID	Device ID Profile
0x10	Security Manager TK Value	Core Specification Supplement, Part A, Section 1.8
0x11	Security Manager Out of Band Flags	Core Specification Supplement, Part A, Section 1.7
0x12	Peripheral Connection Interval Range	Core Specification Supplement, Part A, Section 1.9





- From assigned numbers doc

Common Data Type	Name	Reference
0x01	Flags	Core Specification Supplement, Part A, Section 1.3
0x02	Incomplete List of 16-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x03	Complete List of 16-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x04	Incomplete List of 32-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x05	Complete List of 32-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x06	Incomplete List of 128-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x07	Complete List of 128-bit Service Class UUIDs	Core Specification Supplement, Part A, Section 1.1
0x08	Shortened Local Name	Core Specification Supplement, Part A, Section 1.2
0x09	Complete Local Name	Core Specification Supplement, Part A, Section 1.2
0x0A	Tx Power Level	Core Specification Supplement, Part A, Section 1.5
0x0D	Class of Device	Core Specification Supplement, Part A, Section 1.6
0x0E	Simple Pairing Hash C-192	Core Specification Supplement, Part A, Section 1.6
0x0F	Simple Pairing Randomizer R-192	Core Specification Supplement, Part A, Section 1.6
0x10	Device ID	Device ID Profile
0x10	Security Manager TK Value	Core Specification Supplement, Part A, Section 1.8
0x11	Security Manager Out of Band Flags	Core Specification Supplement, Part A, Section 1.7
0x12	Peripheral Connection Interval Range	Core Specification Supplement, Part A, Section 1.9



2thprint by UUID128



or





2thprint by UUID128

UUID128Print

- Detective Work : UUID128Print -> NamePrint



KFTC BANKPOS

Point of Sale terminal

- Regex: ^KFTC BANKPOSS\$
- "Korea Financial Telecommunications and Clearings Institute (Korean: 금융결제원, KFTC) is a non-profit organization which manages several inter-bank payment systems in South Korea."
- [https://en.wikipedia.org/wiki/
Korea_Financial_Telecommunications_%26_Clearings_Institute](https://en.wikipedia.org/wiki/Korea_Financial_Telecommunications_%26_Clearings_Institute)



KFTC BANKPOS

Point of Sale terminal

- Regex: ^KFTC BANKPOSS\$
- "Korea Financial Telecommunications and Clearings Institute (Korean: 금융결제원, KFTC) is a non-profit organization which manages several inter-bank payment systems in South Korea."
- [https://en.wikipedia.org/wiki/
Korea_Financial_Telecommunications_%26_Clearings_Institute](https://en.wikipedia.org/wiki/Korea_Financial_Telecommunications_%26_Clearings_Institute)



KFTC BANKPOS

Point of Sale terminal

```
For bdaddr = 04:32:f4:18:2e:d8:  
    Company Name by IEEE OUI (04:32:f4): Partron  
  
    No BTC Extended Inquiry Result Device info.  
  
    DeviceName: KFTC BANKPOS  
        In BT LE Data (LE_bdaddr_to_name), bdaddr_random = 0 (Public)  
        This was found in an event of type 4 which corresponds to Scan Response (SCAN_RSP)  
  
    No UUID16s found.  
  
    No transmit power found.  
  
    No Appearance data found.  
  
    Manufacturer-specific Data:  
        Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!  
            Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)  
        Raw Data: 0215585cde931b0142cc9a1325009bedc65e00010002c5  
        Apple iBeacon:  
            UUID128: 585cde93-1b01-42cc-9a13-25009bedc65e  
            Major ID: 0001  
            Minor ID: 0002  
            RSSI at 1 meter: -59dBm  
            In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)  
            This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
```



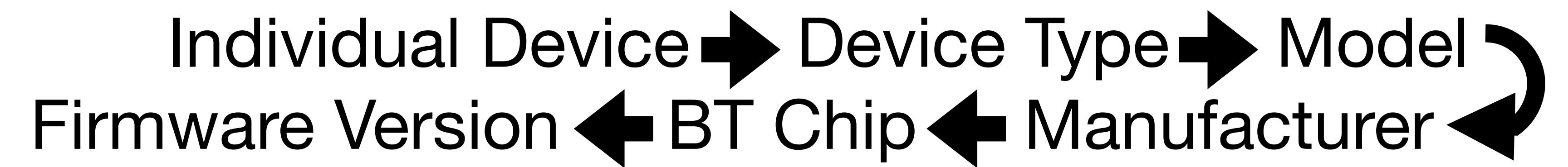
KFTC BANKPOS

Point of Sale terminal

```
For bdaddr = 04:32:f4:18:48:d7:  
    Company Name by IEEE OUI (04:32:f4): Partron  
  
    No BTC Extended Inquiry Result Device info.  
  
    No Names found.  
  
    No UUID16s found.  
  
    No transmit power found.  
  
    No Appearance data found.  
  
    Manufacturer-specific Data:  
        Device Company ID: 0x004c (Apple, Inc.) - take with a grain of salt, not all companies populate this accurately!  
        Endianness-flipped device company ID (in case the vendor used the wrong endianness): 0x4c00 (No Match)  
        Raw Data: 0215585cde931b0142cc9a1325009bedc65e00010002c5  
        Apple iBeacon:  
            UUID128: 585cde93-1b01-42cc-9a13-25009bedc65e  
            Major ID: 0001  
            Minor ID: 0002  
            RSSI at 1 meter: -59dBm  
            In BT LE Data (LE_bdaddr_to_mf_specific), bdaddr_random = 0 (Public)  
            This was found in an event of type 0 which corresponds to Connectable Undirected Advertising (ADV_IND)
```

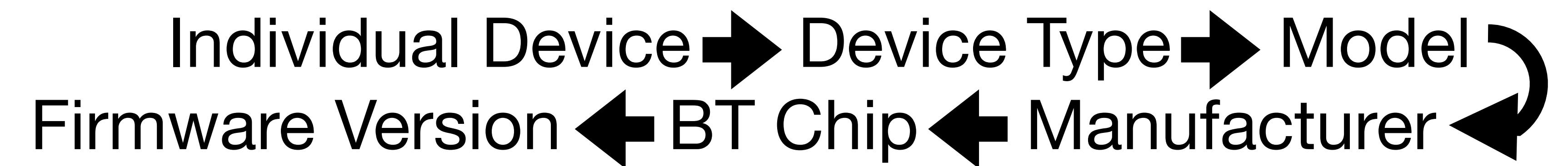


What I Want





What I Want



UUID128: **585cde93-1b01-42cc-9a13-25009bedc65e**



What I Want

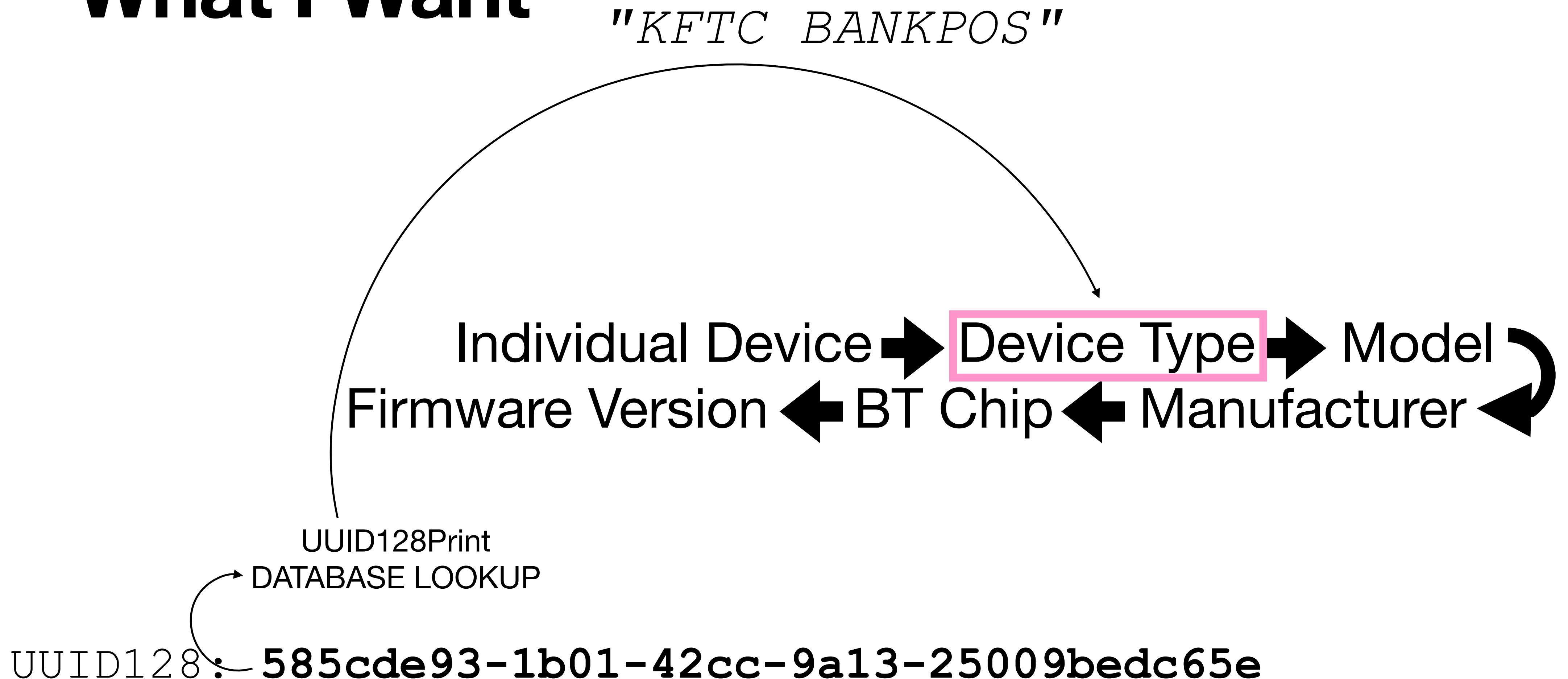
Individual Device → Device Type → Model
Firmware Version ← BT Chip ← Manufacturer ←

UUID128Print
DATABASE LOOKUP

UUID128: 585cde93-1b01-42cc-9a13-25009bedc65e

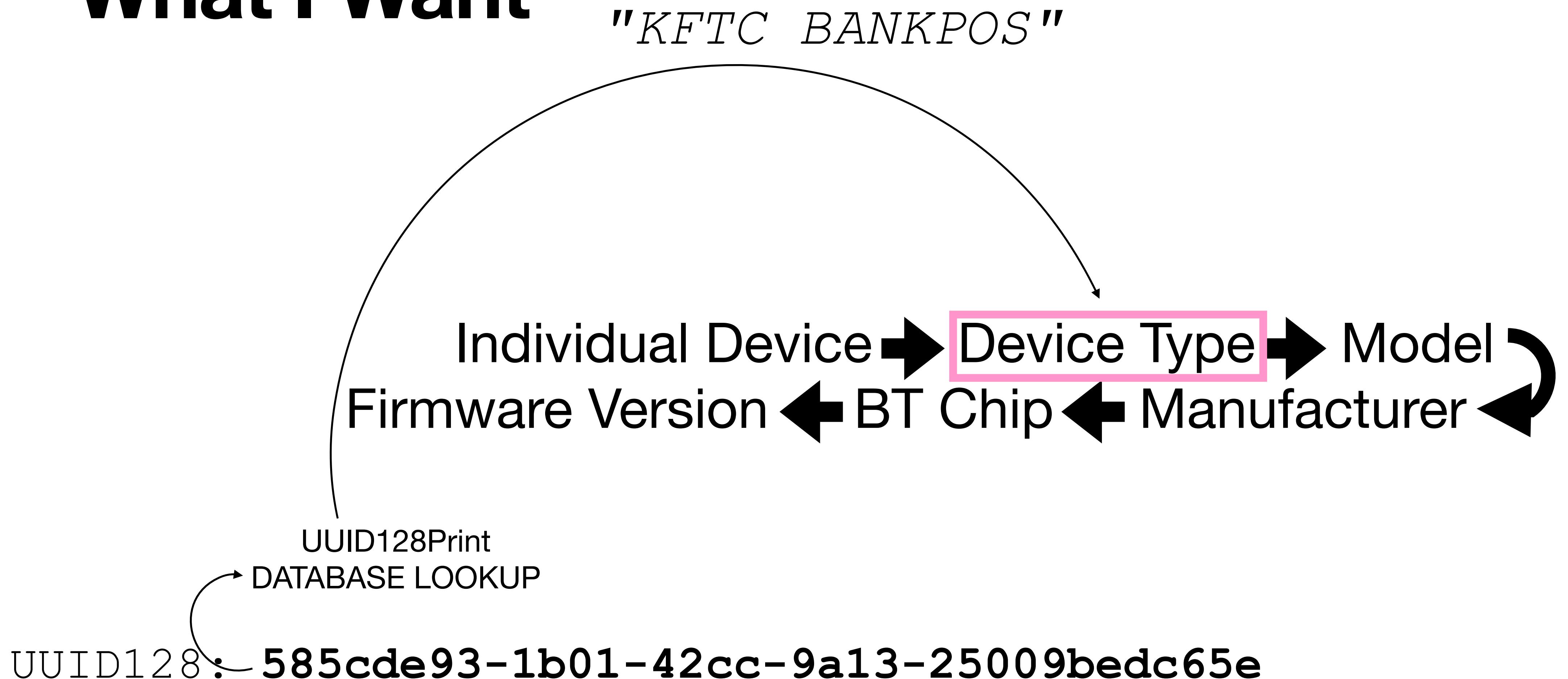


What I Want





What I Want



ASSUMPTION:

UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers



Vendor-specific 128-bit UUIDs

- abbaff00-e56a-484c-b832-8b17cf6cbfe8
 - Versa (|2|Lite), Ionic
- adab*fb00*-6e7d-4601-bda2-bffaa68956ba
 - Inspire HR, Flex 2
- adab*0d57*-6e7d-4601-bda2-bffaa68956ba
- adab*6552*-6e7d-4601-bda2-bffaa68956ba
 - One
- adab*5b8c*-6e7d-4601-bda2-bffaa68956ba
 - Flex



Vendor-specific 128-bit UUIDs

- abbaff00-e56a-484c-b832-8b17cf6cbfe8
 - Versa (|2|Lite), Ionic
- adab*fb00*-6e7d-4601-bda2-bffaa68956ba
 - Inspire HR, Flex 2
- adab*0d57*-6e7d-4601-bda2-bffaa68956ba
- adab*6552*-6e7d-4601-bda2-bffaa68956ba
 - One
- adab*5b8c*-6e7d-4601-bda2-bffaa68956ba
 - Flex

> HCI Event: LE Meta Event (0x3e) plen 42
LE Advertising Report (0x02)
Num reports: 1
Event type: Connectable undirected - **ADV_IND** (0x00)
Address type: Random (0x01)
Address: F5:6E:B2:C3:73:D2 (Static)
Data length: 30
Flags: 0x06
LE General Discoverable Mode
BR/EDR Not Supported
128-bit Service UUIDs (partial): 1 entry
Vendor specific (**abbaff00-e56a-484c-b832-8b17cf6cbfe8**)
Service Data (UUID 0x180a): 2604329303
RSSI: -93 dBm (0xa3)



Vendor-specific 128-bit UUIDs

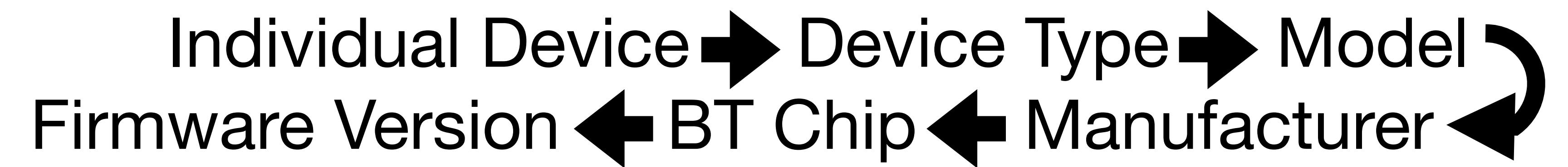
- abbaff00-e56a-484c-b832-8b17cf6cbfe8
 - Versa (|2|Lite), Ionic
- adab*fb00*-6e7d-4601-bda2-bffaa68956ba
 - Inspire HR, Flex 2
- adab*0d57*-6e7d-4601-bda2-bffaa68956ba
- adab*6552*-6e7d-4601-bda2-bffaa68956ba
 - One
- adab*5b8c*-6e7d-4601-bda2-bffaa68956ba
 - Flex

> HCI Event: LE Meta Event (0x3e) plen 42
LE Advertising Report (0x02)
Num reports: 1
Event type: Connectable undirected - **ADV_IND** (0x00)
Address type: Random (0x01)
Address: F5:6E:B2:C3:73:D2 (Static)
Data length: 30
Flags: 0x06
LE General Discoverable Mode
BR/EDR Not Supported
128-bit Service UUIDs (partial): 1 entry
Vendor specific (**abbaff00-e56a-484c-b832-8b17cf6cbfe8**)
Service Data (UUID 0x180a): 2604329303
RSSI: -93 dBm (0xa3)



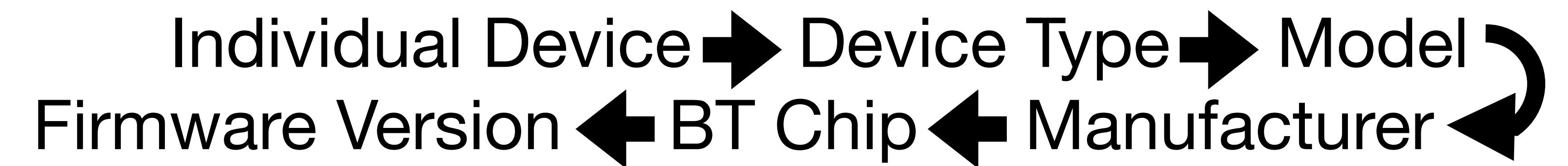


What I Want





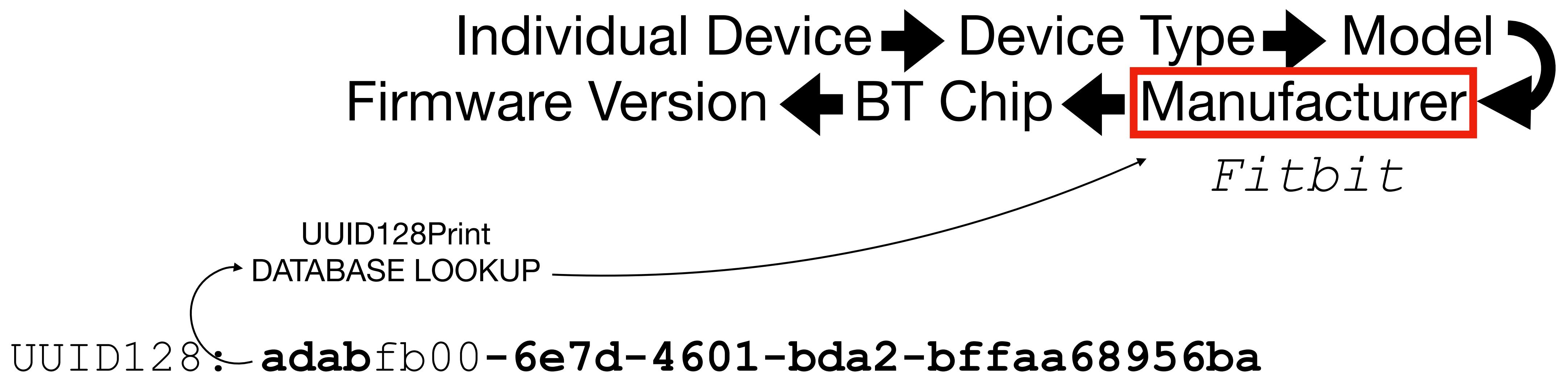
What I Want



UUID128: **adabfb00-6e7d-4601-bda2-bffaa68956ba**

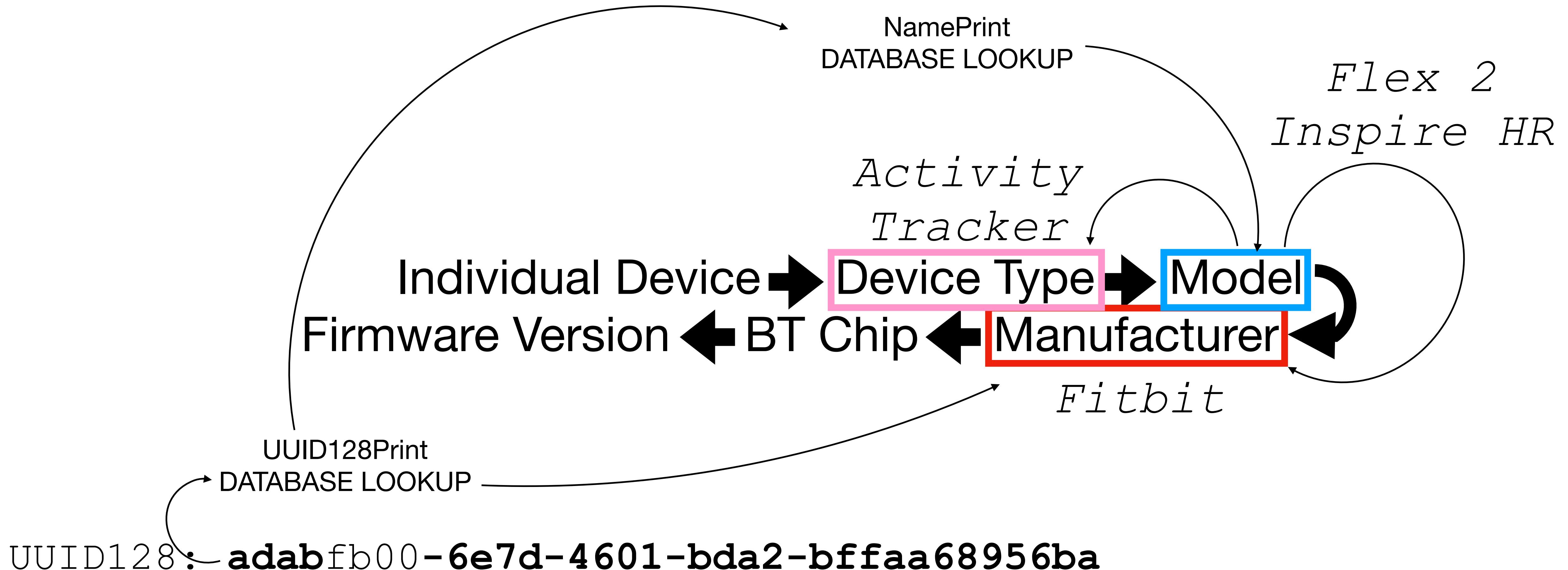


What I Want



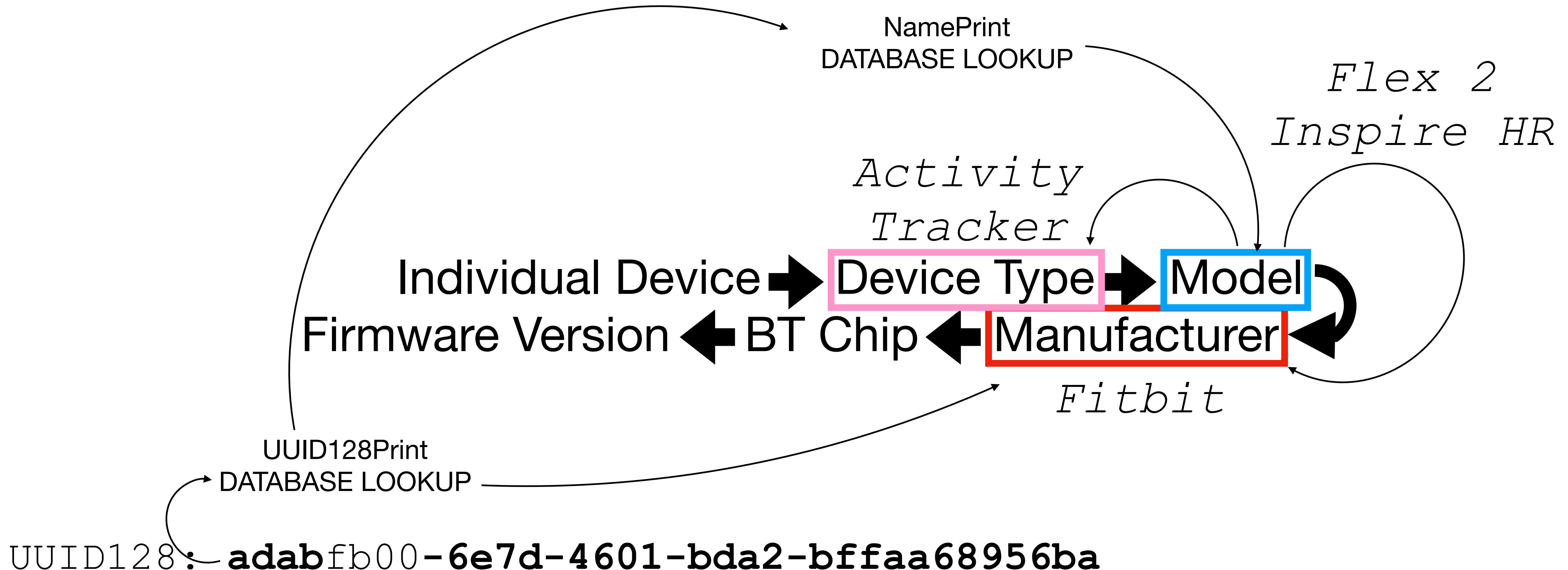


What I Want





What I Want

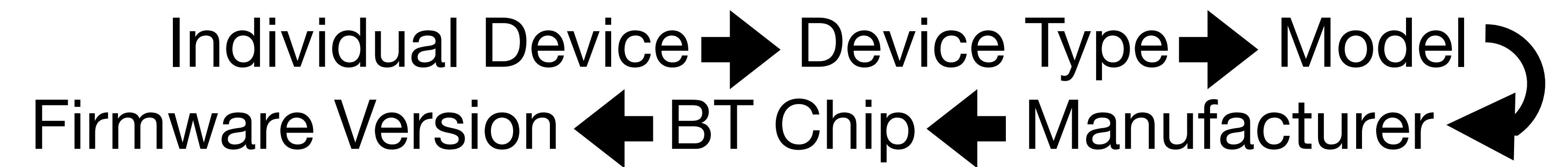


ASSUMPTION:

UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers

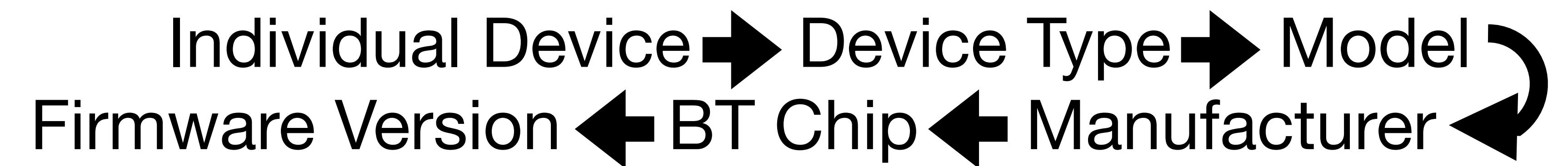


What I Want





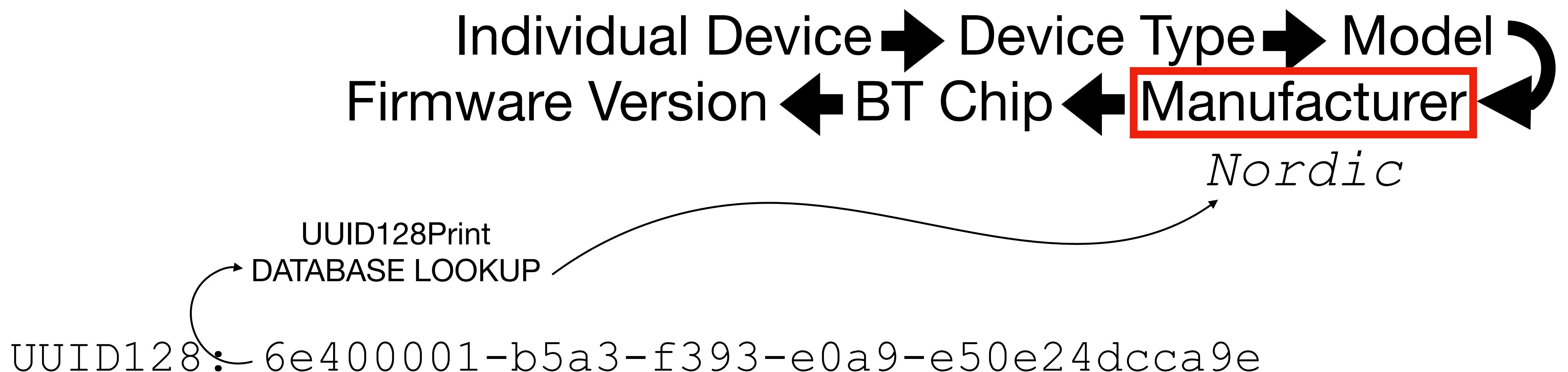
What I Want



UUID128: 6e400001-b5a3-f393-e0a9-e50e24dcca9e

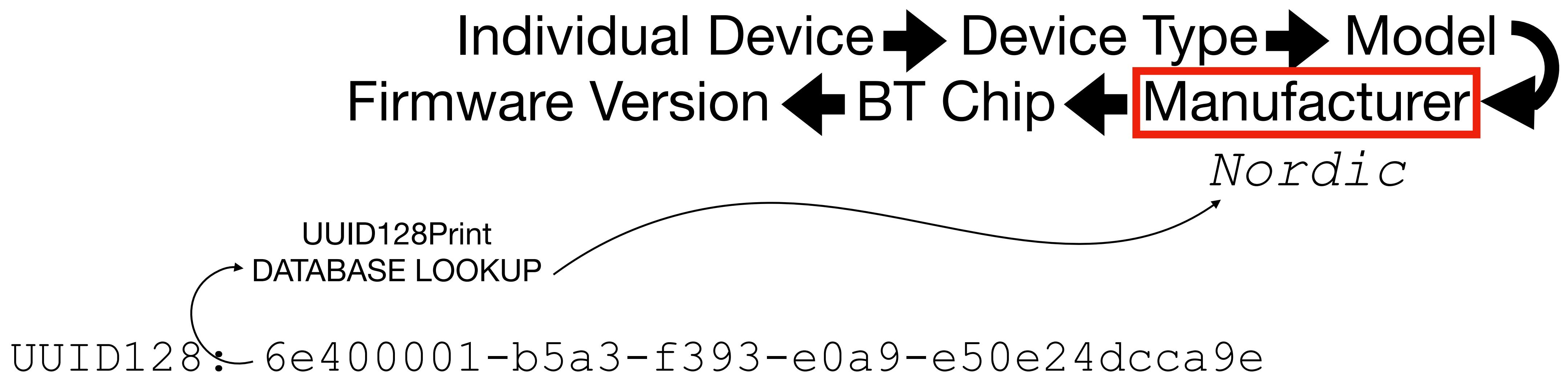


What I Want





What I Want

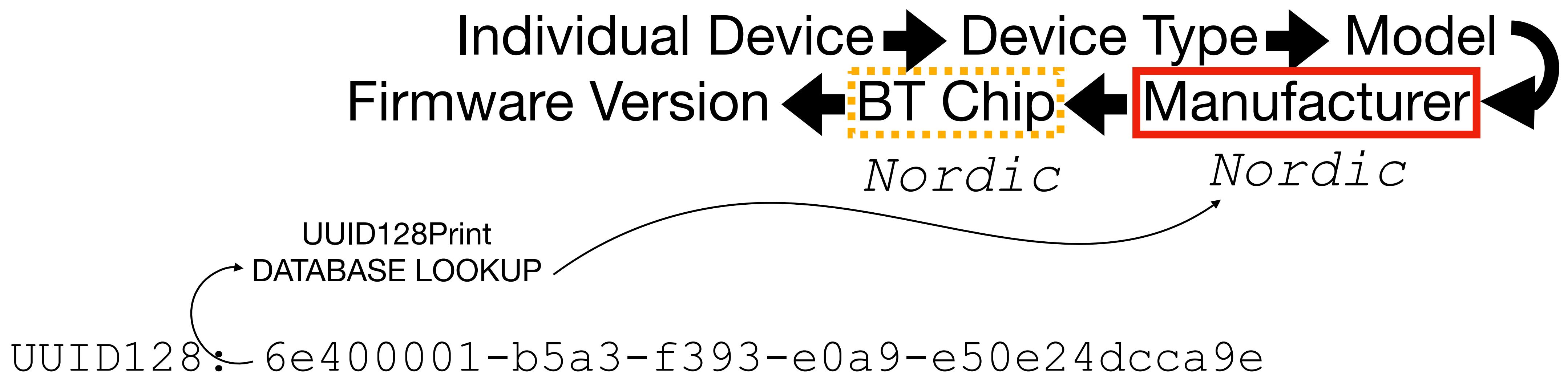


ASSUMPTION:

UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers



What I Want



ASSUMPTION:

UUID128Prints can be reused *within* Manufacturers,
but not reused *between* Manufacturers

2thprint by SDP





Prior Work

"Blueprinting - Remote Device Identification based on Bluetooth Fingerprinting Techniques"

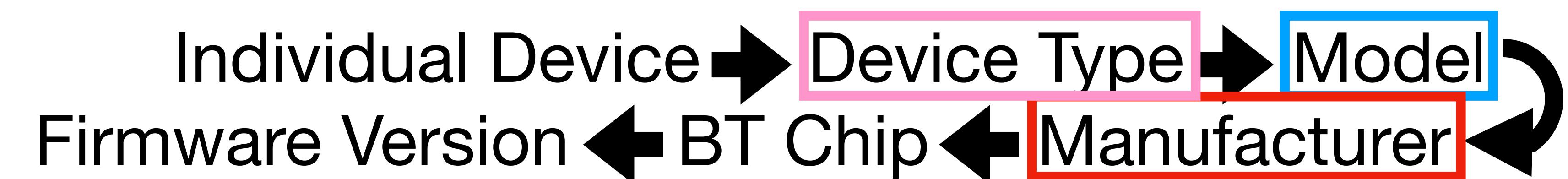
- [1] by Herfert & Mulliner from 2004 connected to BTC devices via Service Discovery Protocol (SDP) and created a hash from selected data within the available profiles
- I'm simply using the deprecated "sdptool" from BlueZ with its existing XML output option. *But I haven't decided how to process the data yet!*



Prior Work

"Blueprinting - Remote Device Identification based on Bluetooth Fingerprinting Techniques"

- [1] by Herfert & Mulliner from 2004 connected to BTC devices via Service Discovery Protocol (SDP) and created a hash from selected data within the available profiles
 - I'm simply using the deprecated "sdptool" from BlueZ with its existing XML output option. *But I haven't decided how to process the data yet!*

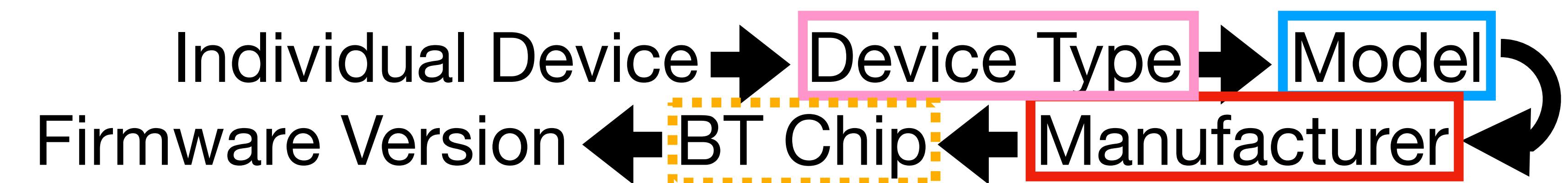




Prior Work

"Blueprinting - Remote Device Identification based on Bluetooth Fingerprinting Techniques"

- [1] by Herfert & Mulliner from 2004 connected to BTC devices via Service Discovery Protocol (SDP) and created a hash from selected data within the available profiles
 - I'm simply using the deprecated "sdptool" from BlueZ with its existing XML output option. *But I haven't decided how to process the data yet!*



2thprint by Class of Device





2thprint by Class of Device (CoD)

Primarily applicable to BTC (but some rare BLE devices use it too)

- BTC Extended Inquiry Response (EIR) packets contain a 24-bit CoD value

2.8 Class of Device

Referenced from the following:

- Bluetooth Core Specification [Vol 2] Part B, Section 6.5.1.4 [4].
- Supplement to the Bluetooth Core Specification Part A, Section 1.6.2 [22].

The Class of Device is composed of four fields: A Major Service Classes bitfield, a Major Device Class enumerated value, the Minor Device Classes, and a fixed value of 0b00 in the two least significant bits. The format of the Minor Device Class is determined by the Major Device Class value. The structure of the Class of Device is defined below:

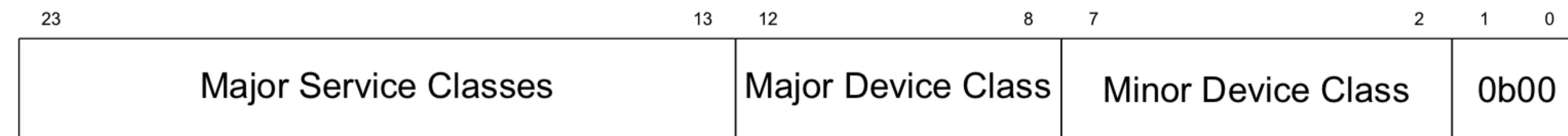


Figure 2.2: Class of Device format



2.8.2 Major Device Classes

The Miscellaneous major device class is used where a more specific Major Device Class code is not suitable. A device that does not have a major class code assigned can use the Uncategorized: device code not specified until "classified."

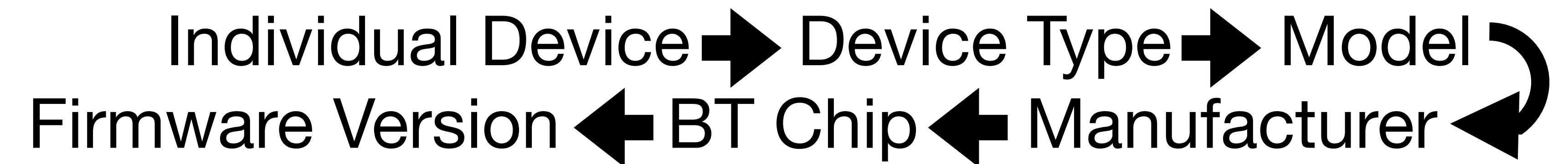
Last Modified: 2022-05-25

12	11	10	9	8	Major Device Class
0	0	0	0	0	Miscellaneous
0	0	0	0	1	Computer (desktop, notebook, PDA, organizer, ...)
0	0	0	1	0	Phone (cellular, cordless, pay phone, modem, ...)
0	0	0	1	1	LAN/Network Access point
0	0	1	0	0	Audio/Video (headset, speaker, stereo, video display, VCR, ...)
0	0	1	0	1	Peripheral (mouse, joystick, keyboard, ...)
0	0	1	1	0	Imaging (printer, scanner, camera, display, ...)
0	0	1	1	1	Wearable
0	1	0	0	0	Toy
0	1	0	0	1	Health
1	1	1	1	1	Uncategorized: device code not specified

2.8.1 Major Service Classes

Last Modified: 2022-05-25

Bit	Class of Device Major Service Class
13	Limited Discoverable Mode
14	LE audio
15	Reserved for future use
16	Positioning (Location identification)
17	Networking (LAN, Ad hoc, ...)
18	Rendering (Printing, Speakers, ...)
19	Capturing (Scanner, Microphone, ...)
20	Object Transfer (v-Inbox, v-Folder, ...)
21	Audio (Speaker, Microphone, Headset service, ...)
22	Telephony (Cordless telephony, Modem, Headset service, ...)
23	Information (WEB-server, WAP-server, ...)





2.8.2 Major Device Classes

The Miscellaneous major device class is used where a more specific Major Device Class code is not suitable. A device that does not have a major class code assigned can use the Uncategorized: device code not specified until "classified."

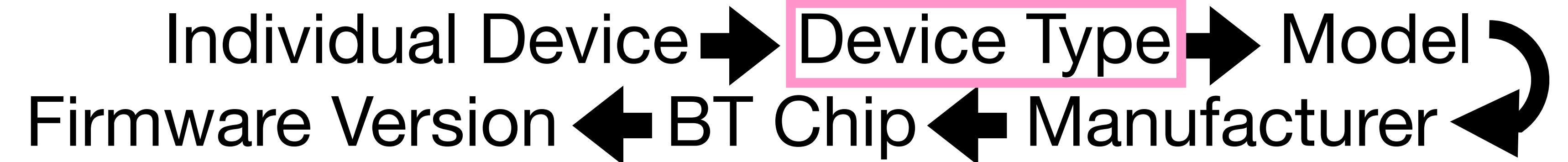
Last Modified: 2022-05-25

12	11	10	9	8	Major Device Class
0	0	0	0	0	Miscellaneous
0	0	0	0	1	Computer (desktop, notebook, PDA, organizer, ...)
0	0	0	1	0	Phone (cellular, cordless, pay phone, modem, ...)
0	0	0	1	1	LAN/Network Access point
0	0	1	0	0	Audio/Video (headset, speaker, stereo, video display, VCR, ...)
0	0	1	0	1	Peripheral (mouse, joystick, keyboard, ...)
0	0	1	1	0	Imaging (printer, scanner, camera, display, ...)
0	0	1	1	1	Wearable
0	1	0	0	0	Toy
0	1	0	0	1	Health
1	1	1	1	1	Uncategorized: device code not specified

2.8.1 Major Service Classes

Last Modified: 2022-05-25

Bit	Class of Device Major Service Class
13	Limited Discoverable Mode
14	LE audio
15	Reserved for future use
16	Positioning (Location identification)
17	Networking (LAN, Ad hoc, ...)
18	Rendering (Printing, Speakers, ...)
19	Capturing (Scanner, Microphone, ...)
20	Object Transfer (v-Inbox, v-Folder, ...)
21	Audio (Speaker, Microphone, Headset service, ...)
22	Telephony (Cordless telephony, Modem, Headset service, ...)
23	Information (WEB-server, WAP-server, ...)



2thprint by "Pics or it didn't happen"





2thprint by looking up teardown pictures :P

This is not scalable...unless we crowdsource it!

- When one wants to know what a very specific device has for a chip, one can just google for teardown pictures!
- Sometimes the FCC (or other wireless regulatory authorities') database "internal photos" are useful in this regard

E Tu Rivian?

- Regex: ^Rivian Sensor [1234]\$ e.g. Rivian Sensor 1
- Regex: ^Rivian Phone Key\$
- Regex: ^Rivian Camp Speaker\$



E Tu Rivian?



- Regex: ^Rivian Sensor [1234]\$ e.g. Rivian Sensor 1
- Regex: ^Rivian Phone Key\$
- Regex: ^Rivian Camp Speaker\$



Address type: **Public** (0x00)

Address: AC:4D:16:FD:40:93 (OUI AC-4D-16)

Name (complete): Rivian Sensor 3

E Tu Rivian?



- Regex: ^Rivian Sensor [1234]\$ e.g. Rivian Sensor 1
- Regex: ^Rivian Phone Key\$
- Regex: ^Rivian Camp Speaker\$



Address type: **Public** (0x00)

Address: AC:4D:16:FD:40:93 (OUI AC-4D-16) ← Actually Texas Instruments

Name (complete): Rivian Sensor 3

btmon just didn't have it in
its vendor database

```
For bdaddr = AC:4D:16:FD:40:93:  
    Company Name by IEEE OUI (AC:4D:16): Texas Instruments  
  
    No BTC Extended Inquiry Result Device info.  
  
    DeviceName: Rivian Sensor 3  
        In BT LE Data (LE_bdaddr_to_name), bdaddr_random = 0 (Public)
```

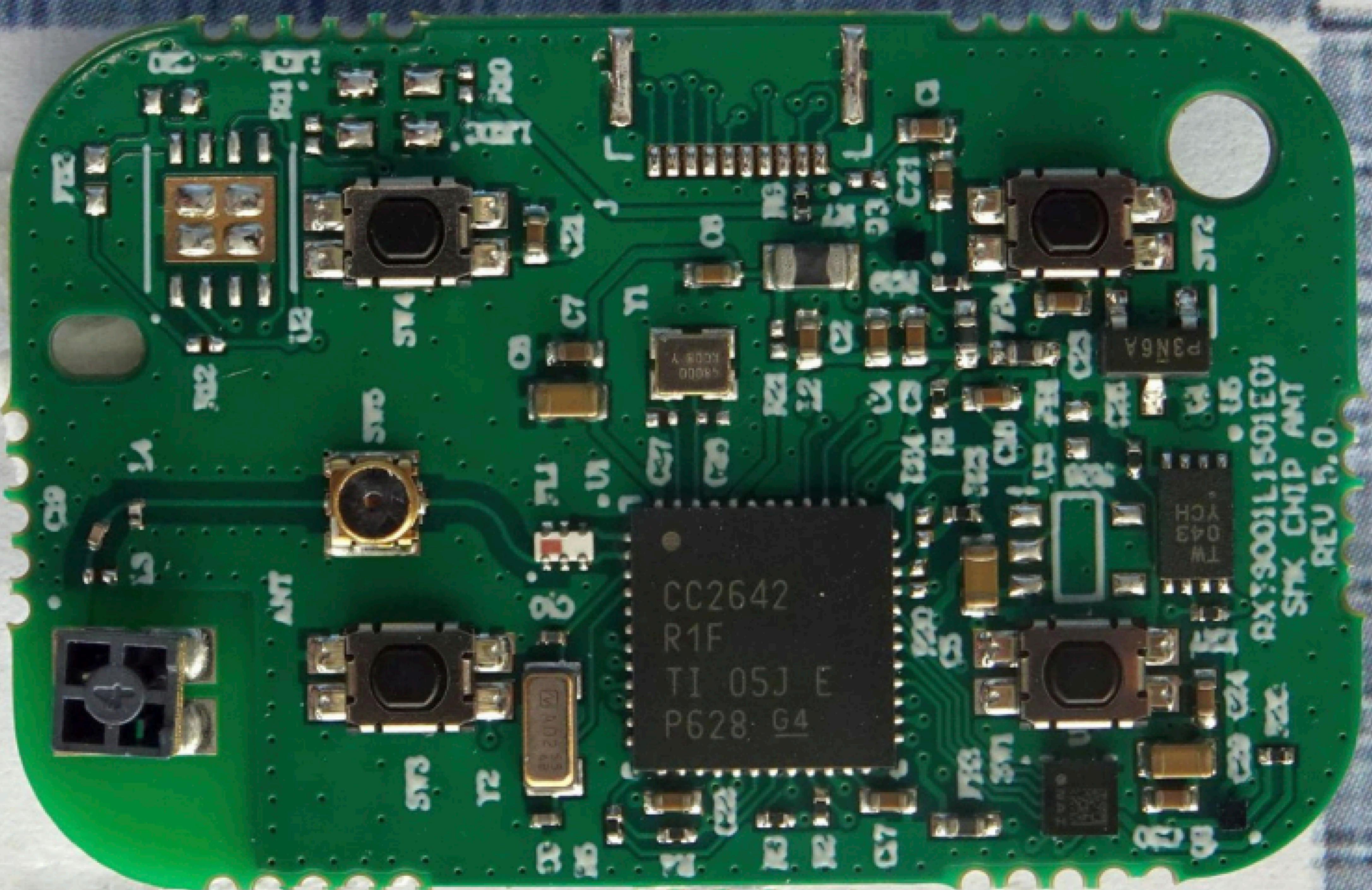
ETU

- Register
- Register
- Register



Address
Address
Name (c)

For bdaddr = A
Company
No BTC
Device



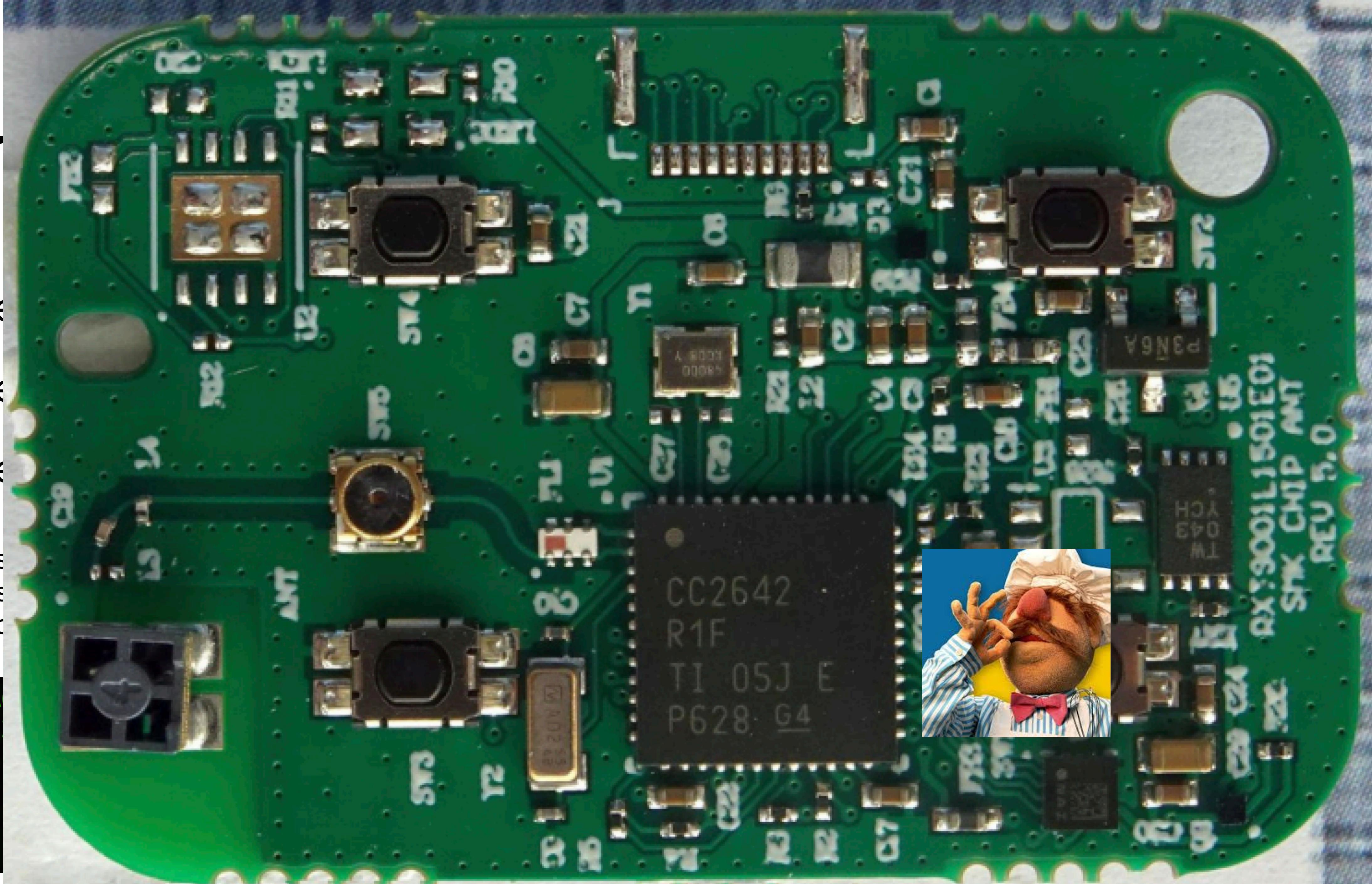
ETU

- Register
- Register
- Register



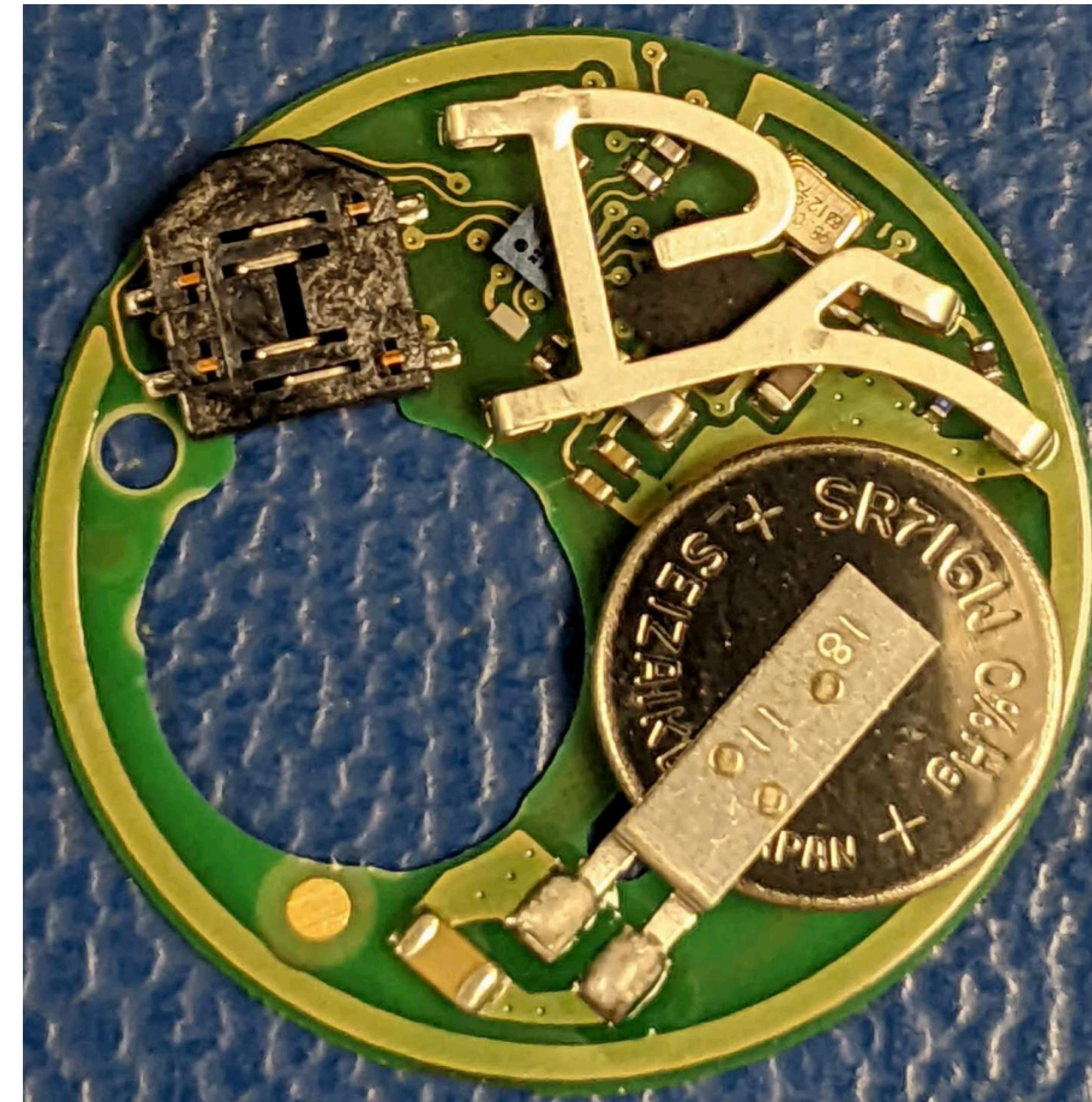
Address
Address
Name (c)

For bdaddr = A
Company
No BTC
Device



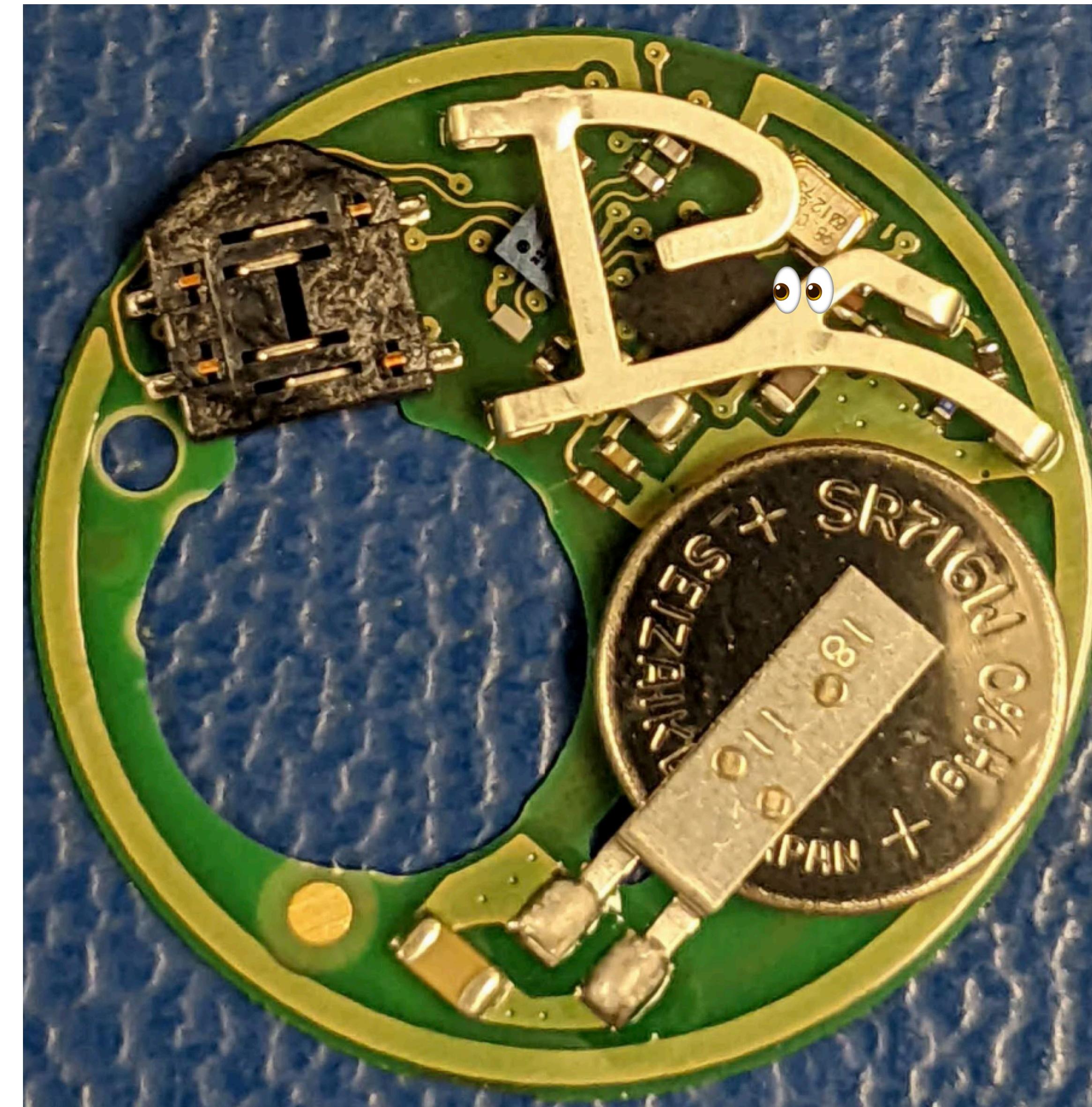


Common Case





Common Case





Common Case

