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)**

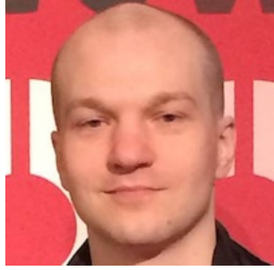
About Me

- 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 ;)

OST2 Crew

In order of appearance



Xeno Kovah

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

Piotr Król

Arch4021 ~6 hours
Arch4031 ~6 hours

Michał Żygowski

Arch4021 ~6 hours

Thaís Moreira Hamasaki

RE3201 ~6 hours

Gal Zaban

RE3011 ~6 hours

Kc Udonsi

Vulns1001 ~15 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


DARK MENTOR


nest

Texas Instruments

Silicon Labs

Broadcom

?

Why I Want To Know It:

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

TEXAS INSTRUMENTS

SILICON LABS

BROADCOM®

ARMIS®

DST2.FYI

TEXAS INSTRUMENTS

SILICON LABS®

BROADCOM®

ARMIS®

DST2.FYI

TEXAS INSTRUMENTS

SEMC
SECURE MOBILE NETWORKING

SILICON LABS®

BROADCOM®

ARMIS®

DARK MENTOR

TEXAS INSTRUMENTS

SEEMO

SECURE MOBILE NETWORKING

SILICON LABS®

BROADCOM®

ARMIS®

DARK MENTOR

TEXAS INSTRUMENTS

SEEMO

SECURE MOBILE NETWORKING

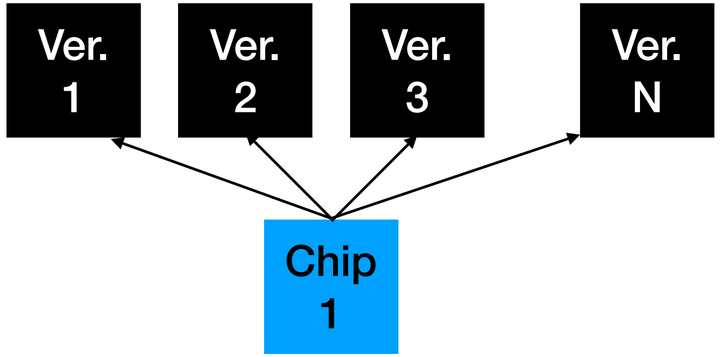
BROADCOM®

DARK MENTOR

SILICON LABS®

VersionPrint

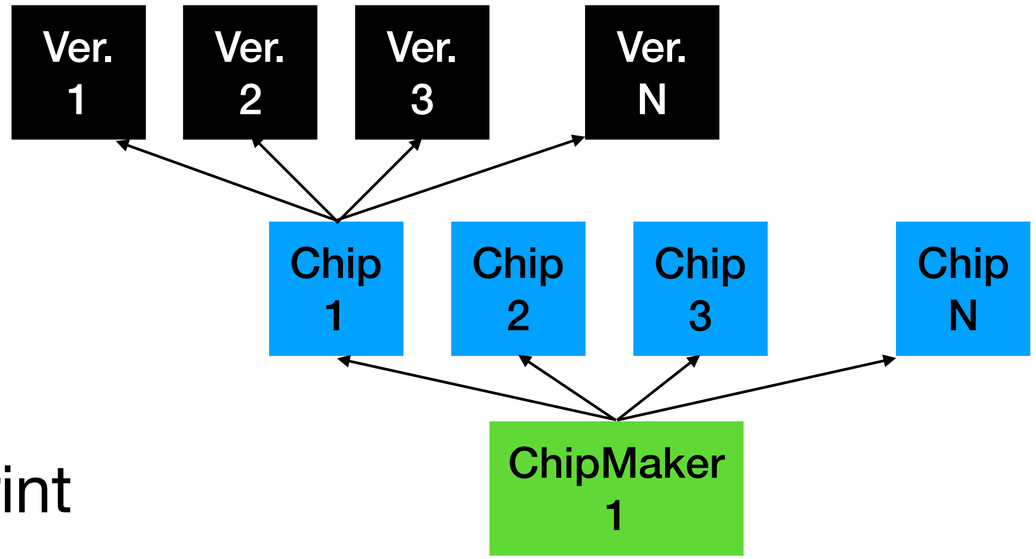
ChipPrint



VersionPrint

ChipPrint

ChipMakerPrint

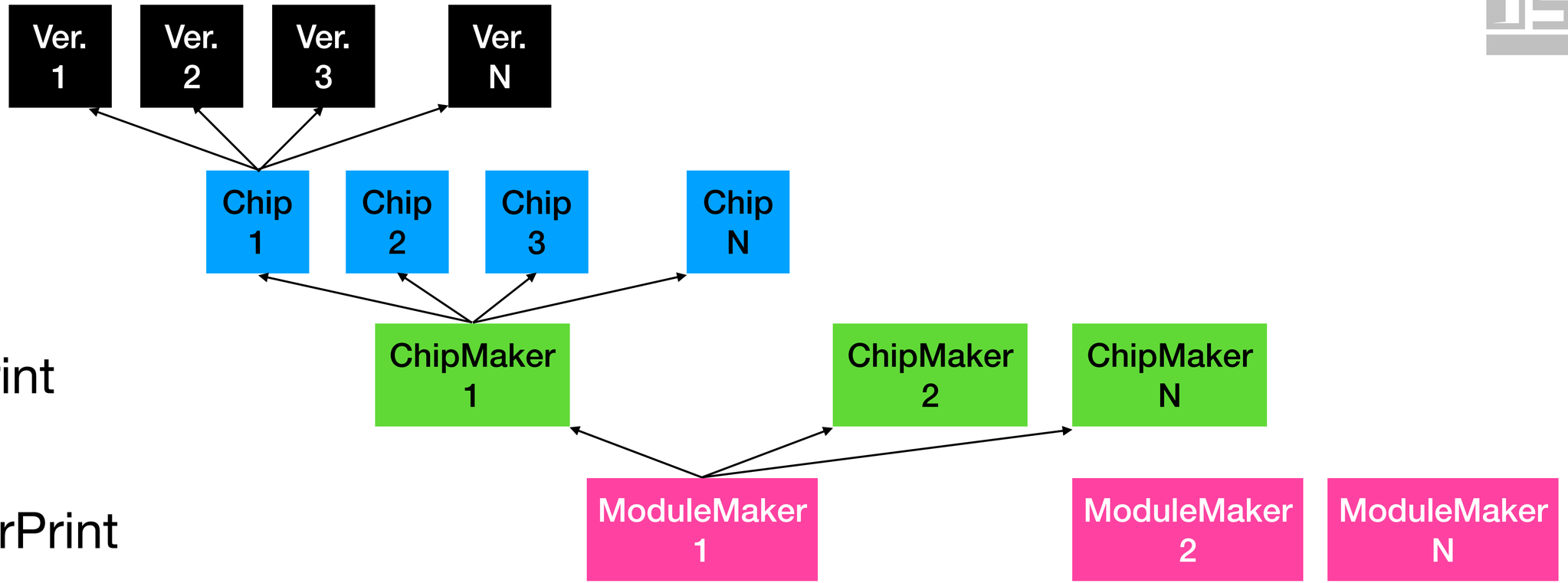


VersionPrint

ChipPrint

ChipMakerPrint

ModuleMakerPrint



Realtek

ChipMaker
1

From
https://docs.ai-thinker.com/en/rtl87xx

REALTEK

ChipMaker
1

Ai-Thinker Technology

ModuleMaker
1

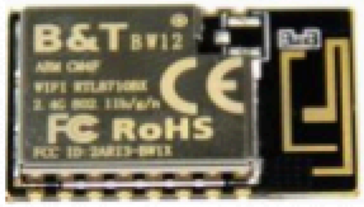
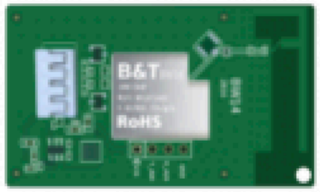
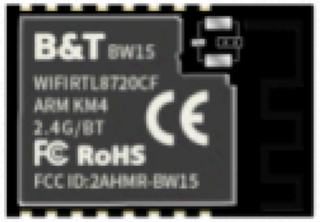
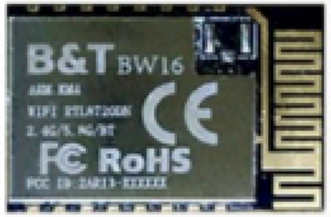
From
https://docs.ai-thinker.com/en/rtl87xx

REALTEK

ChipMaker 1

Ai-Thinker Technology

ModuleMaker 1

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	24 Module 1 lz	240 Module 2 Hz	2.40 Module 3 GHz	2400-248 Module 4 180-5825MHz
Bluetooth			Bl LE	5.0
Operating temperature	-20~+85° C	-20℃~70℃	-40 ℃ ~ 85 ℃	-20 ℃ ~ 70 ℃
Storage temperature	-40 ~125°C	-40℃~125℃	-40 ℃ ~ 125 ℃ , < 90%RH	-40 ℃ ~ 125 ℃ , < 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

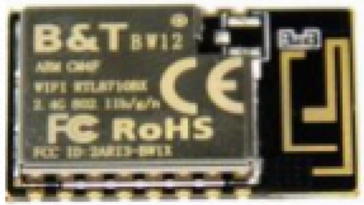
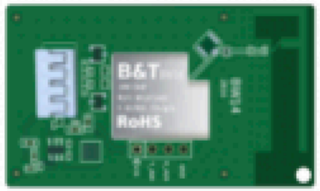
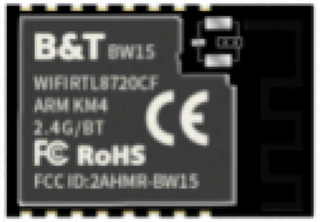
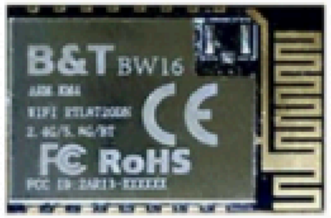
From
https://docs.ai-thinker.com/en/rtl87xx

REALTEK

ChipMaker 1

Ai-Thinker Technology

ModuleMaker 1

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	24 Module 1 lz	240 Module 2 Hz	2.40 Module 3 GHz	2400-248 Module 4 180-5825MHz
Bluetooth			Bl LE	5.0
Operating temperature	-20~+85° C	-20℃~70℃	-40 ℃ ~ 85 ℃	-20 ℃ ~ 70 ℃
Storage temperature	-40 ~125°C	-40℃~125℃	-40 ℃ ~ 125 ℃ , < 90%RH	-40 ℃ ~ 125 ℃ , < 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

???

From
https://docs.ai-thinker.com/en/rtl87xx

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

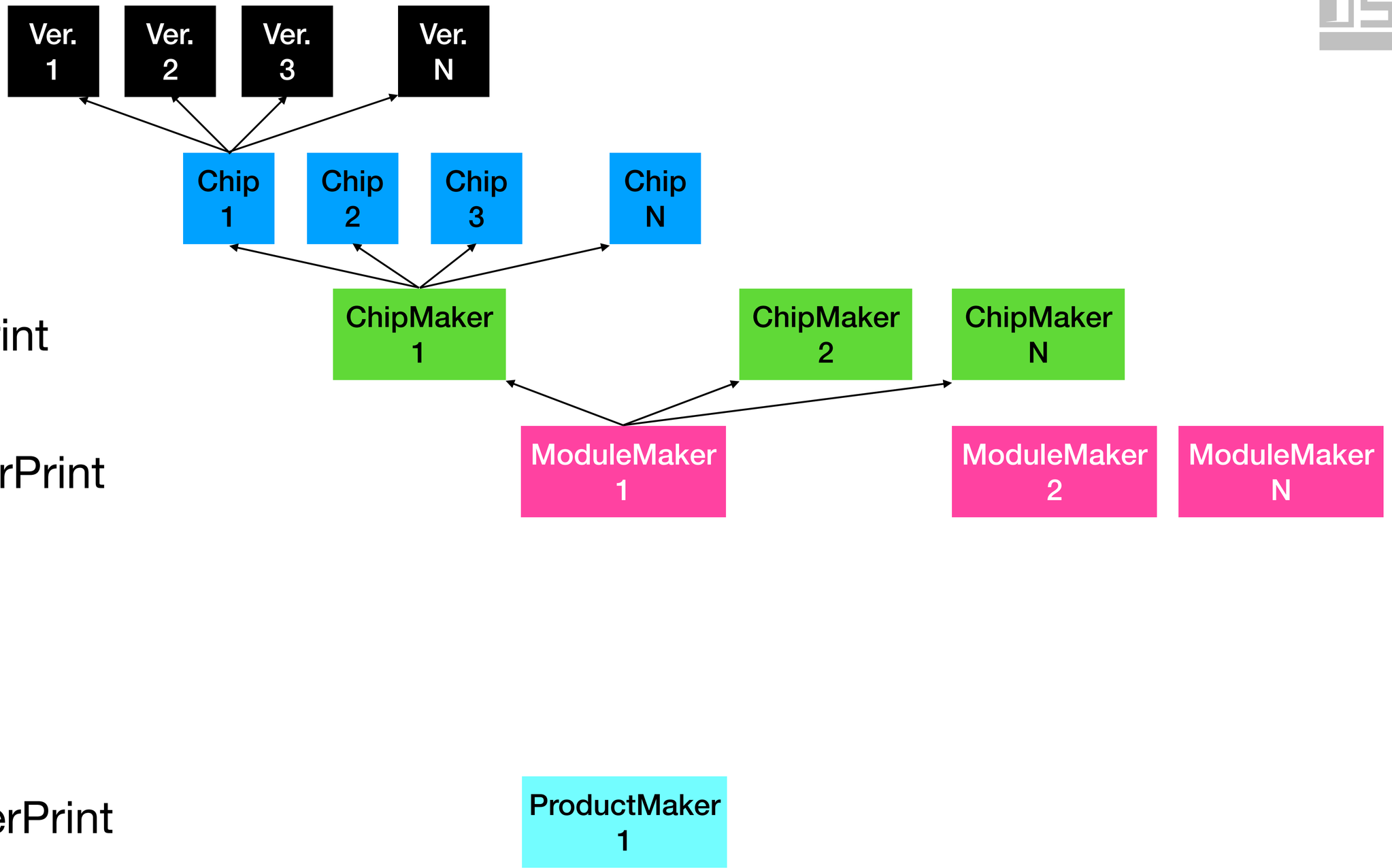
VersionPrint

ChipPrint

ChipMakerPrint

ModuleMakerPrint

ProductMakerPrint



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

VersionPrint

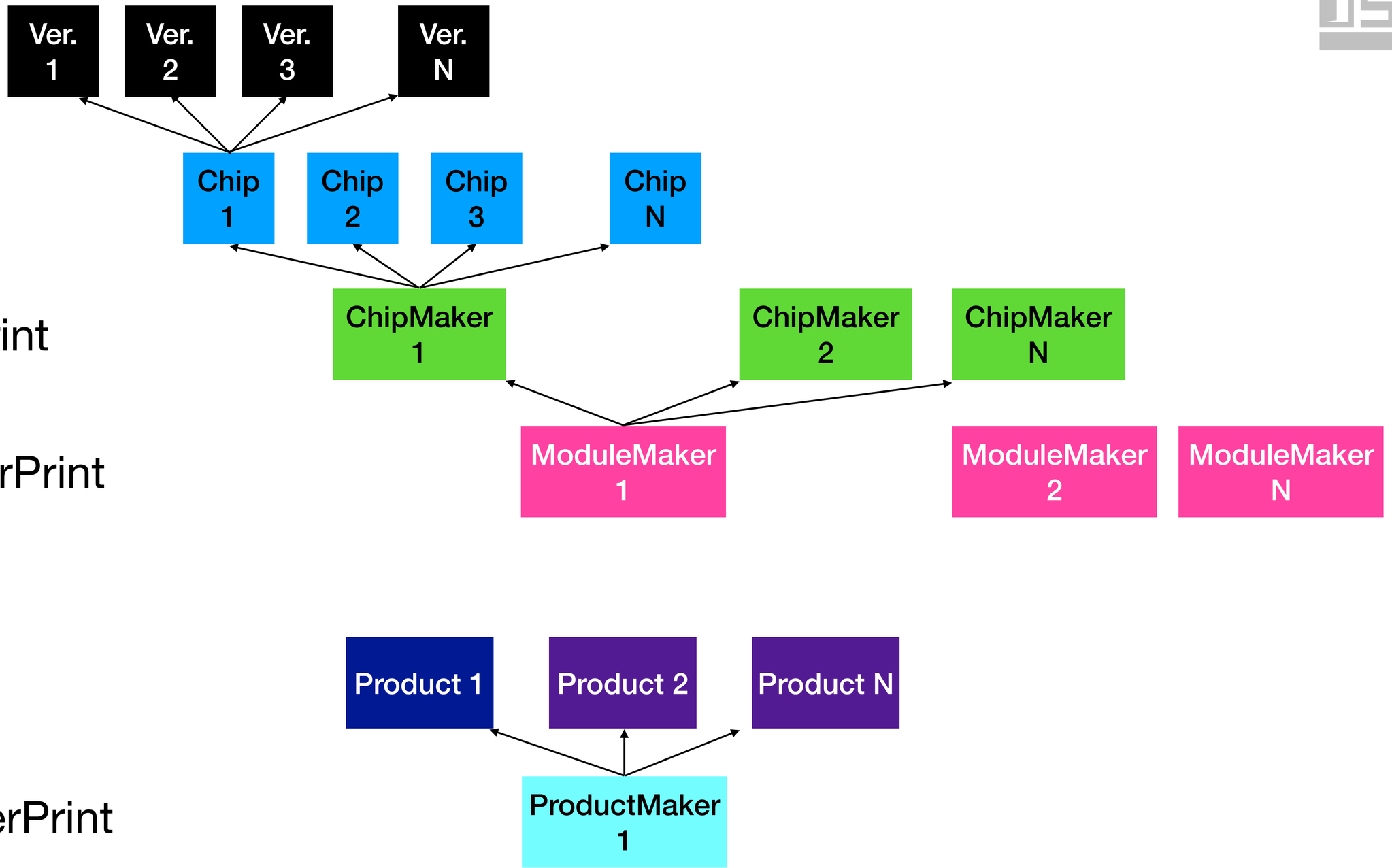
ChipPrint

ChipMakerPrint

ModuleMakerPrint

ProductPrint

ProductMakerPrint



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

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

VersionPrint

ChipPrint

ChipMakerPrint

ModuleMakerPrint

ProductPrint

ProductMakerPrint

Ver. 1
Ver. 2
Ver. 3
Ver. N

Chip 1
Chip 2
Chip 3
Chip N

ChipMaker 1
ChipMaker 2
ChipMaker N

ModuleMaker 1
ModuleMaker 2
ModuleMaker N

Product 1
Product 2
Product N

ProductMaker 1

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

VersionPrint

ChipPrint

ChipMakerPrint

ModuleMakerPrint

ProductPrint

ProductMakerPrint

Ver. 1
Ver. 2
Ver. 3
Ver. N

Chip 1
Chip 2
Chip 3
Chip N

ChipMaker 1
ChipMaker 2
ChipMaker N

ModuleMaker 1
ModuleMaker 2
ModuleMaker N

Product 1
Product 2
Product N

ProductMaker 1

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

Ver. 1
Ver. 2
Ver. 3
Ver. N

Chip 1
Chip 2
Chip 3
Chip N

ChipMaker
ChipMaker 2
ChipMaker N

ModuleMaker 1
ModuleMaker 2
ModuleMaker N

Product 1
Product 2
Product N

ProductMaker 1

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

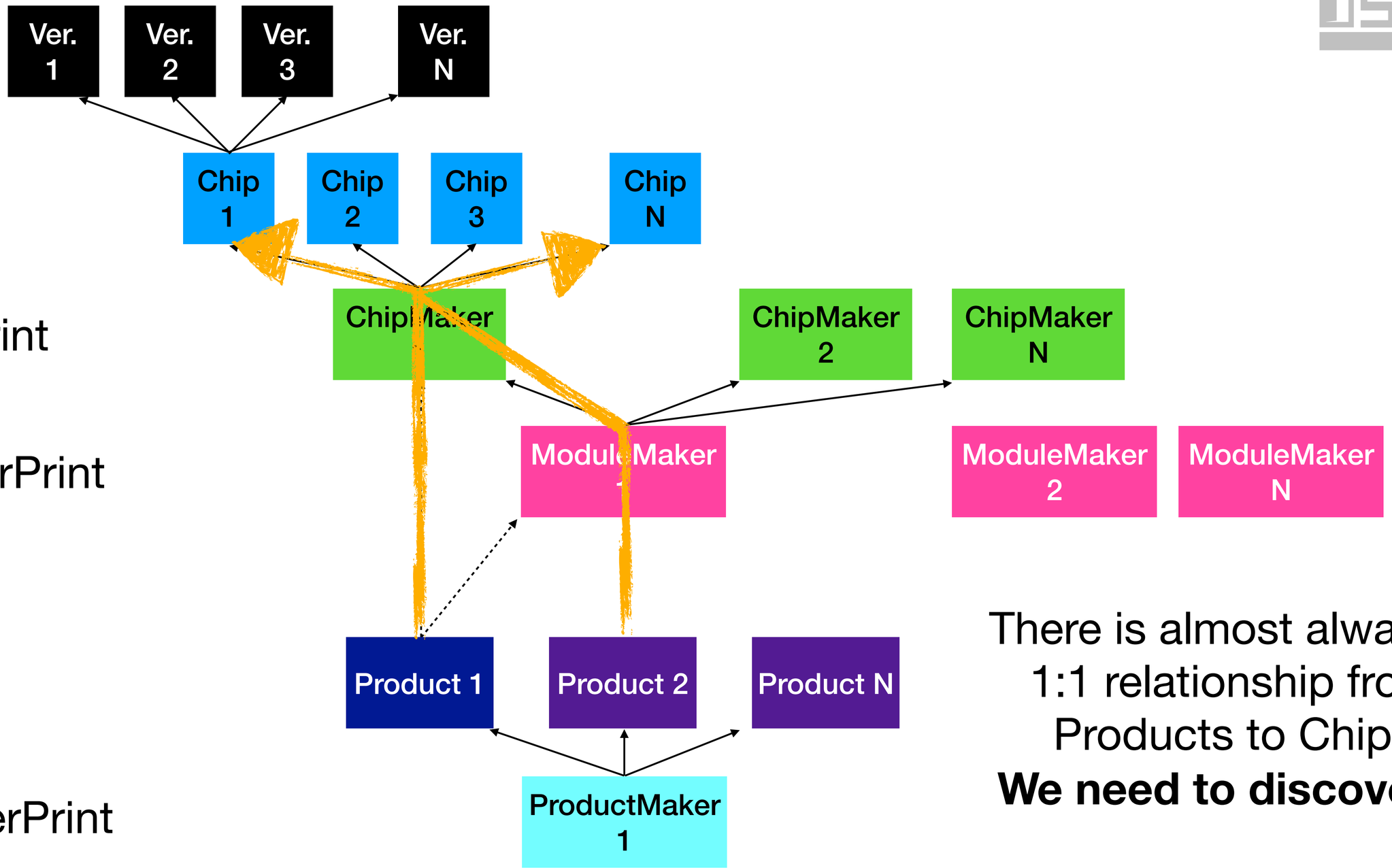
ChipPrint

ChipMakerPrint

ModuleMakerPrint

ProductPrint

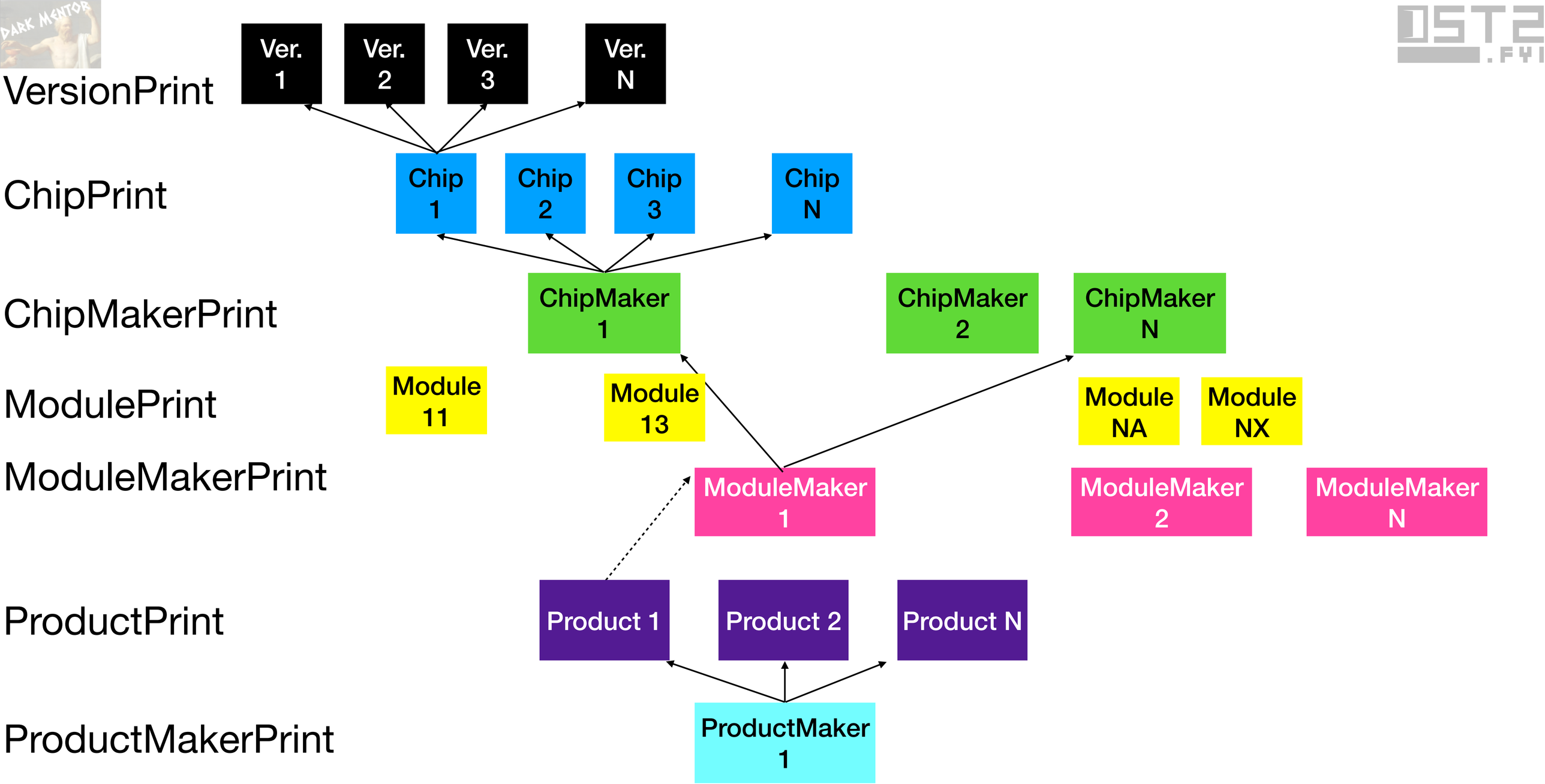
ProductMakerPrint



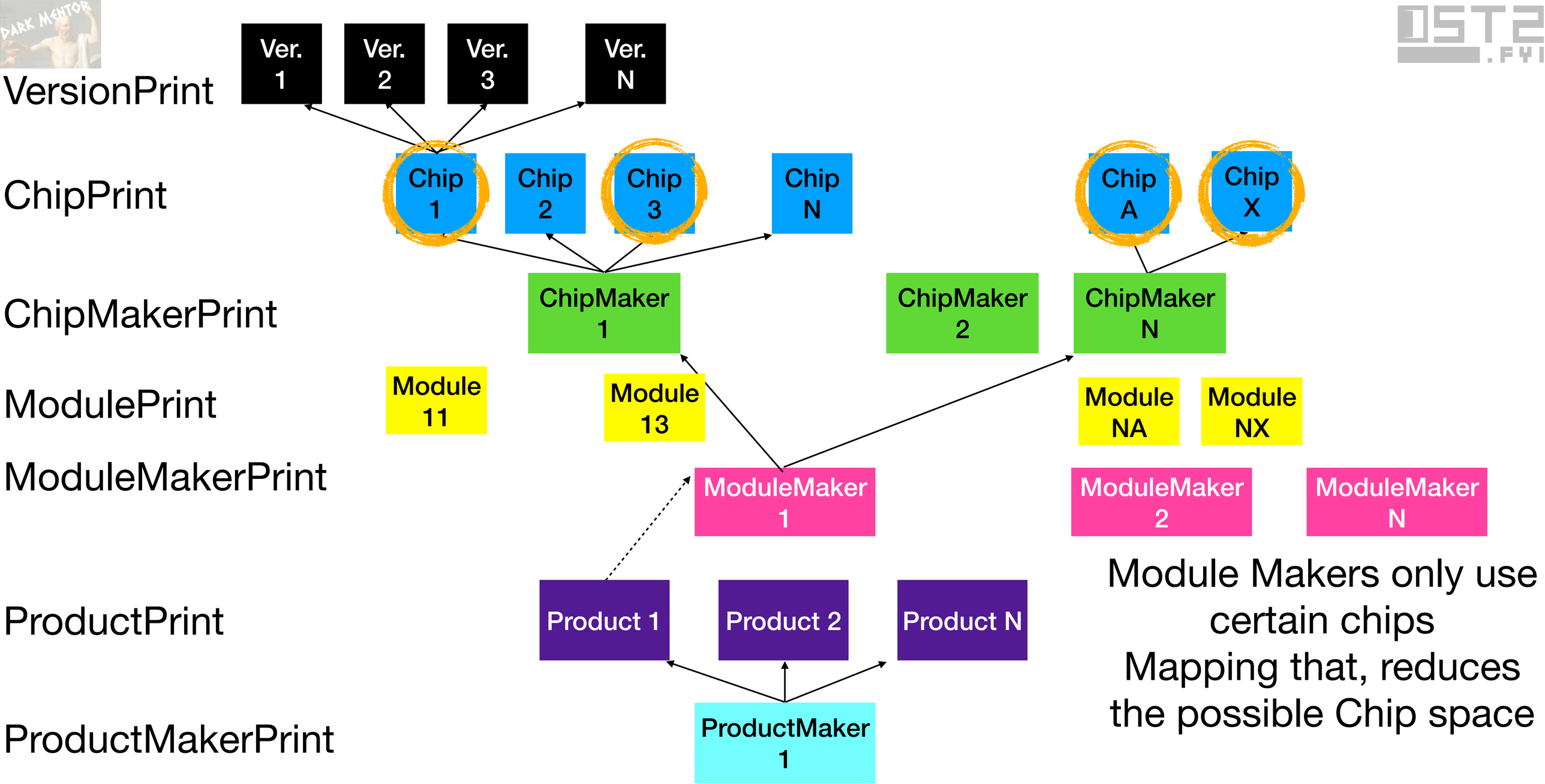
There is almost always a 1:1 relationship from Products to Chips

We need to discover it

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

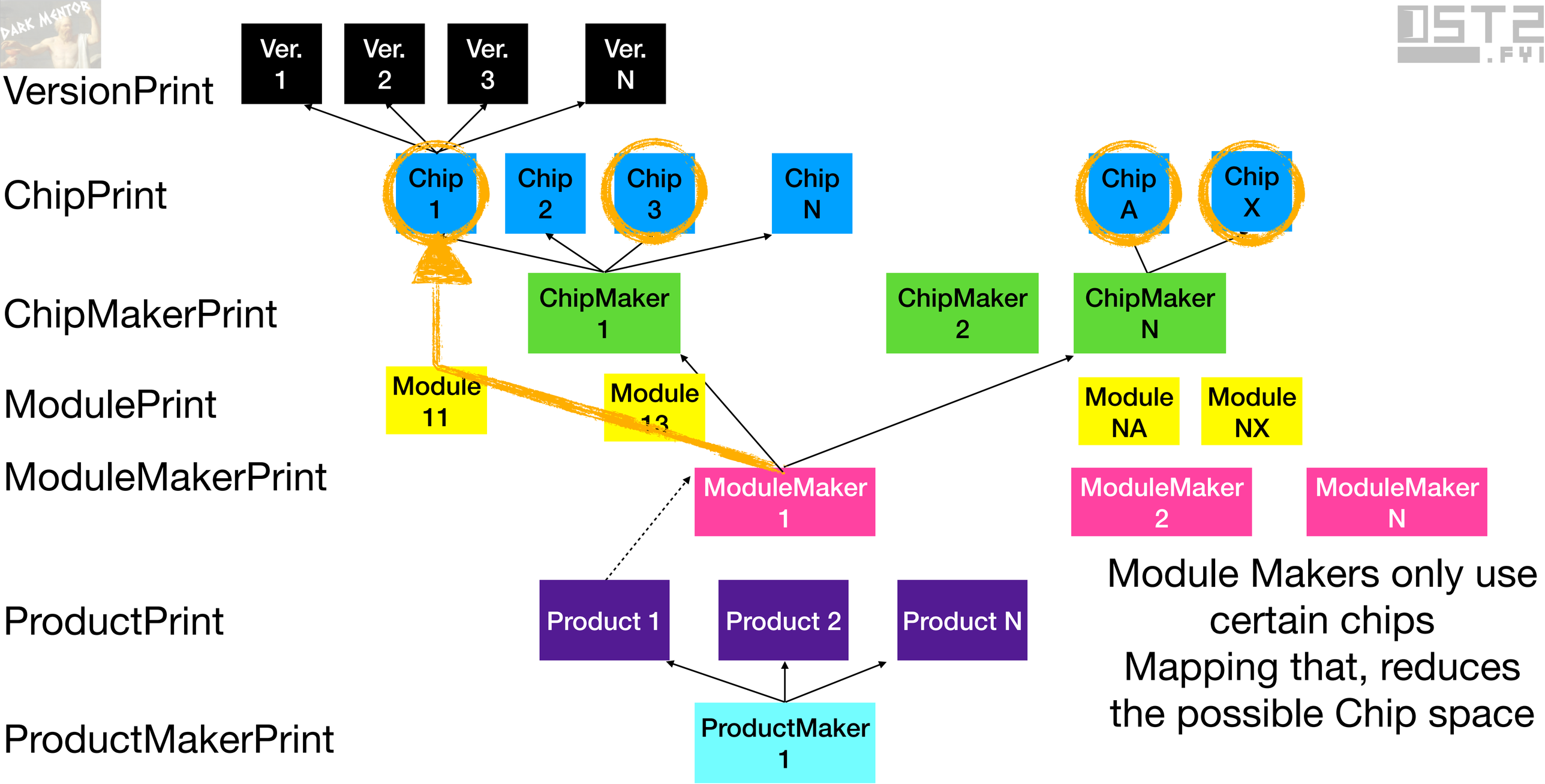


3330 *Product Makers registered* with Bluetooth SIG as of the time of writing!



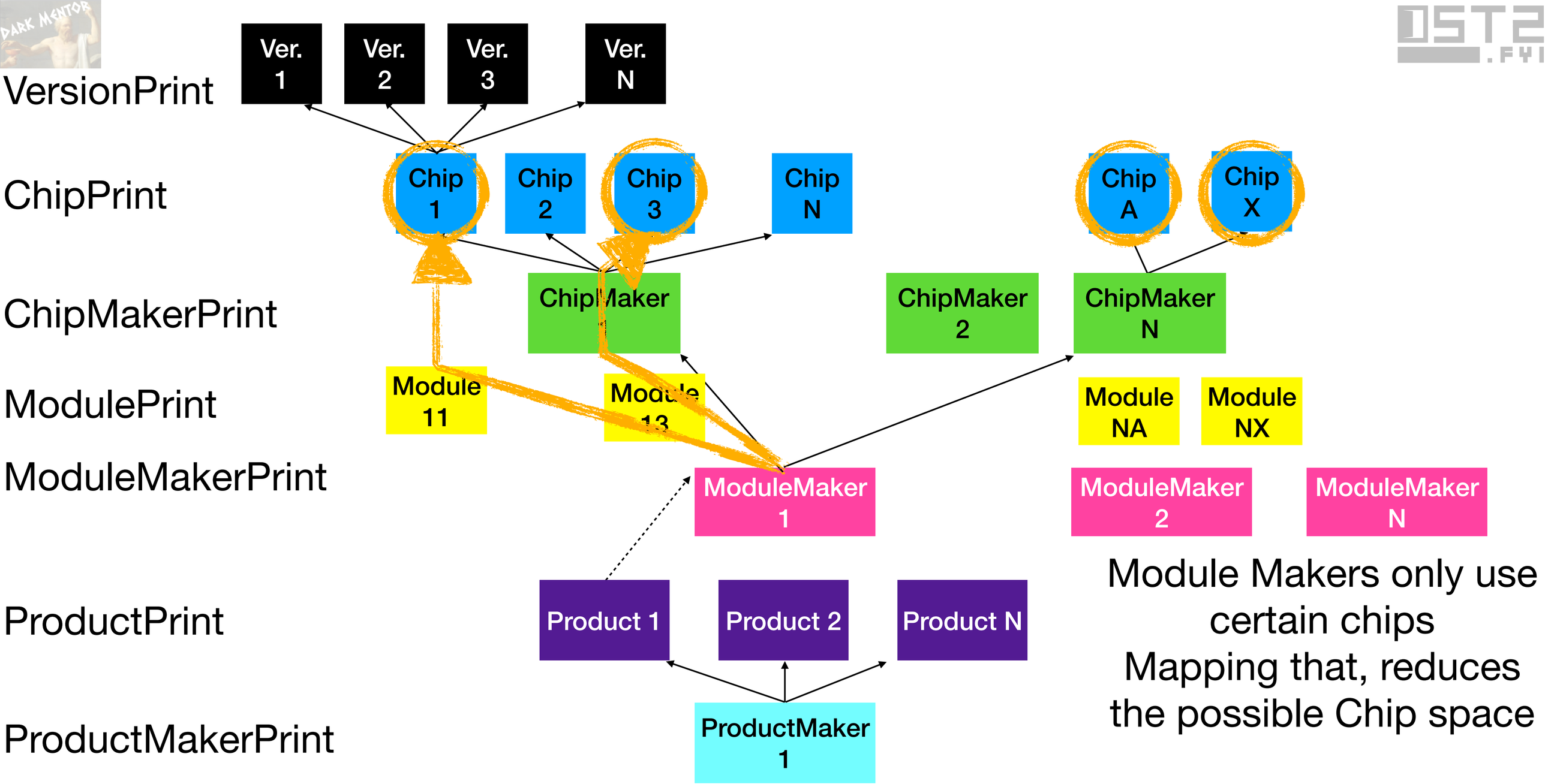
Module Makers only use certain chips
Mapping that, reduces the possible Chip space

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



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!



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

VersionPrint

ChipPrint

ChipMakerPrint

ModulePrint

ModuleMakerPrint

ProductPrint

ProductMakerPrint

Ver. 1
Ver. 2
Ver. 3
Ver. N

Chip 1
Chip 2
Chip 3
Chip N

ChipMaker 1
ChipMaker 2
ChipMaker N

Module 11
Module 13
Module NA
Module NX

ModuleMaker 1
ModuleMaker 2
ModuleMaker N

Product 1
Product 2
Product N

ProductMaker 1

WHAT I WANT!

VersionPrint

Ver. 1 | Ver. 2 | Ver. 3 | Ver. N

ChipPrint

Chip 1 | Chip 2 | Chip 3 | Chip N

ChipMakerPrint

ChipMaker 1 | ChipMaker 2 | ChipMaker N

ModulePrint

Module 11 | Module 13 | Module NA | Module NX

ModuleMakerPrint

ModuleMaker 1 | ModuleMaker 2 | ModuleMaker N

ProductPrint

Product 1 | Product 2 | Product N

ProductMakerPrint

ProductMaker 1

VersionPrint

Ver. 1 | Ver. 2 | Ver. 3 | Ver. N

WHAT I WANT!

ChipPrint

Chip 1 | Chip 2 | Chip 3 | Chip N

WHAT I MOSTLY GET 😔

ChipMakerPrint

ChipMaker 1 | ChipMaker 2 | ChipMaker N

ModulePrint

Module 11 | Module 13 | Module NA | Module NX

ModuleMakerPrint

ModuleMaker 1 | ModuleMaker 2 | ModuleMaker N

ProductPrint

Product 1 | Product 2 | Product N

ProductMakerPrint

ProductMaker 1

My Terminology

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

So you've got a couple million datapoints...

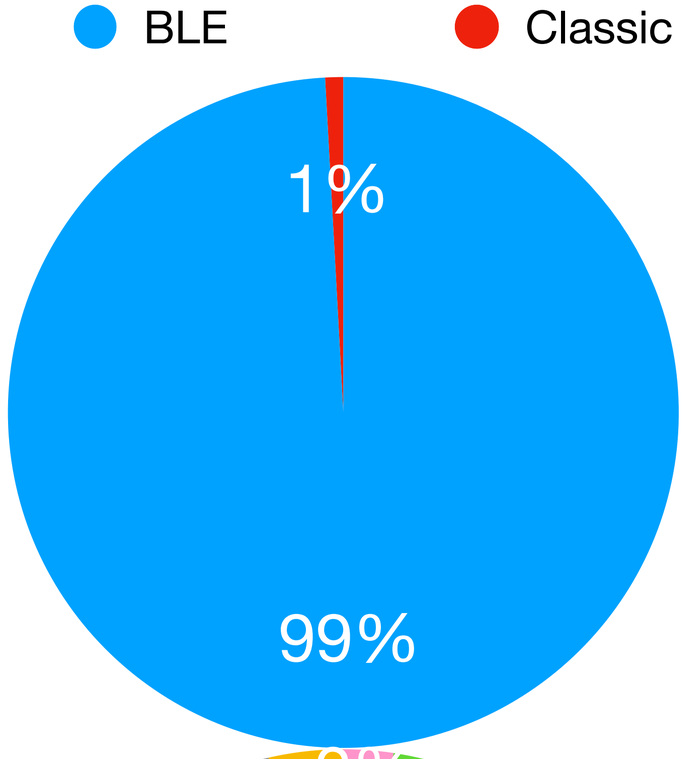
Is any of it useful?

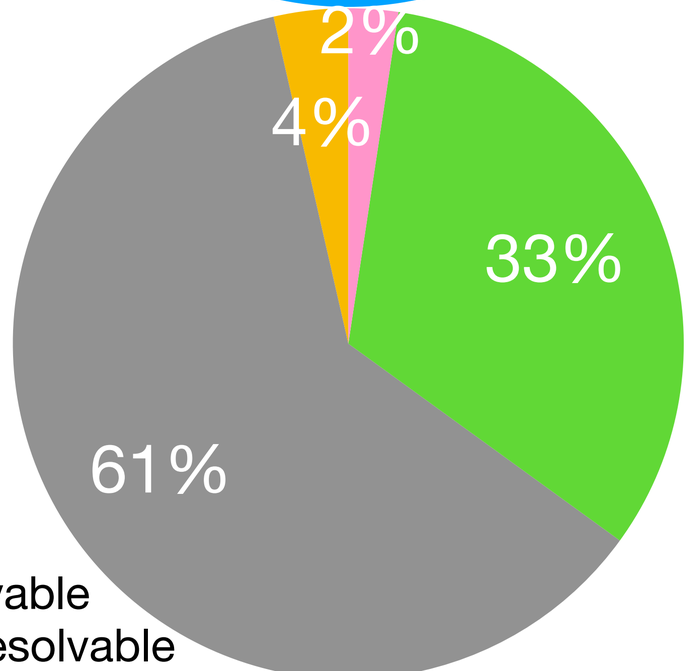
- I started doing basic naïve BT data collection as a hobby during the pandemic as a way to get out of the house and drive around
  - See "It was harder to sniff Bluetooth through my mask during the Pandemic" talk from summer 2023
- The majority of my data is Linux "HCI" logs
  - HCI is the Host-Controller-Interface, between Linux, and the BT chip on my logging platform (mostly Raspberry Pi Zeros or 4Bs)
- I started work on customized 2thprinting in May 2023, once I had a chance to start looking at what the HCI logs could, and couldn't, provide for 2thprinting

Overall BDADDR Data

My data as of 2024-01-12

- 74,934 *unique* BT Classic BDADDRs
- 8,569,483 *unique* BLE BDADDRs
  - 204,708 "public" BDADDRs
  - 2,793,274 "random static" BDADDRs
  - 5,264,247 "random resolvable" BDADDRs
  - 307,030 "random non-resolvable" BDADDRs





🧘 Passive

2thprinting Approaches

🚶 Mostly-Passive

🤾 Active

Bluetooth: The Gathering: 🧘Passive

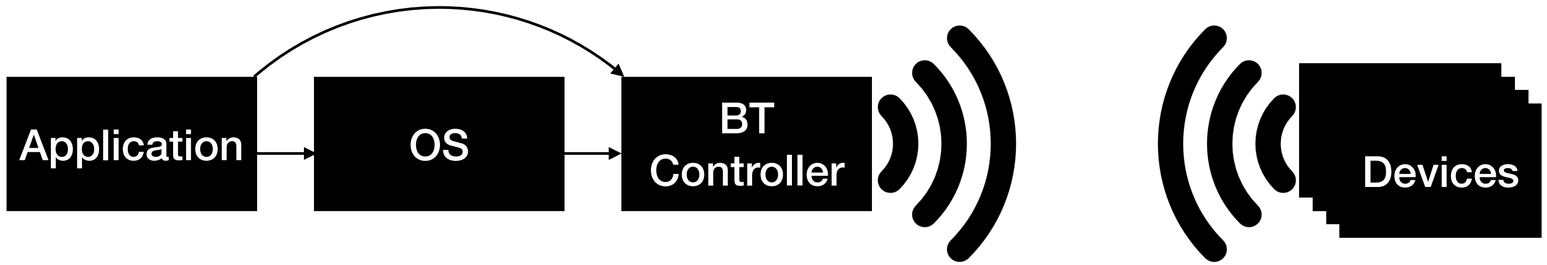
- Run a sniffer, never send *any* packets
- **BLE:** Sniffle, Ice9 Sniffer, **BTC:** Ubertooth
- BLE ADV_IND, NONCONN_ADV_IND

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

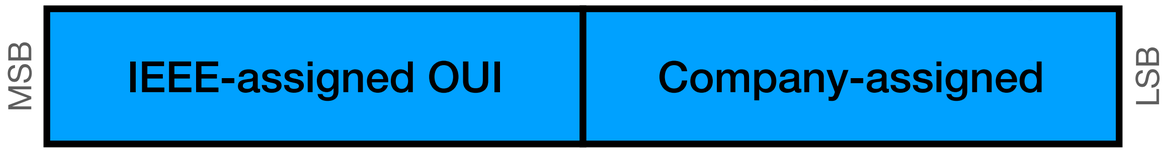
- Send custom BLE LL / BTC LMP packets (requires custom controller firmware)
- Connect to all connectable interfaces, query all queryable information



2thprint by BDADDR OUI

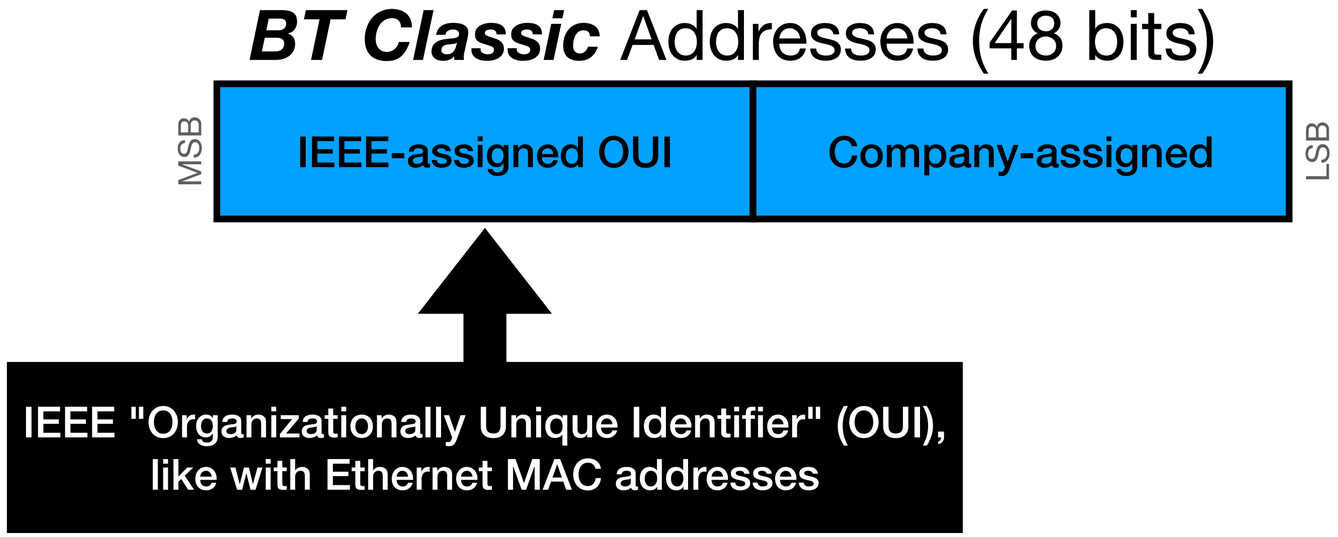
Background - BTC *BT Device Address (BDADDR)*

BT Classic Addresses (48 bits)



https://standards.ieee.org/faqs/regauth

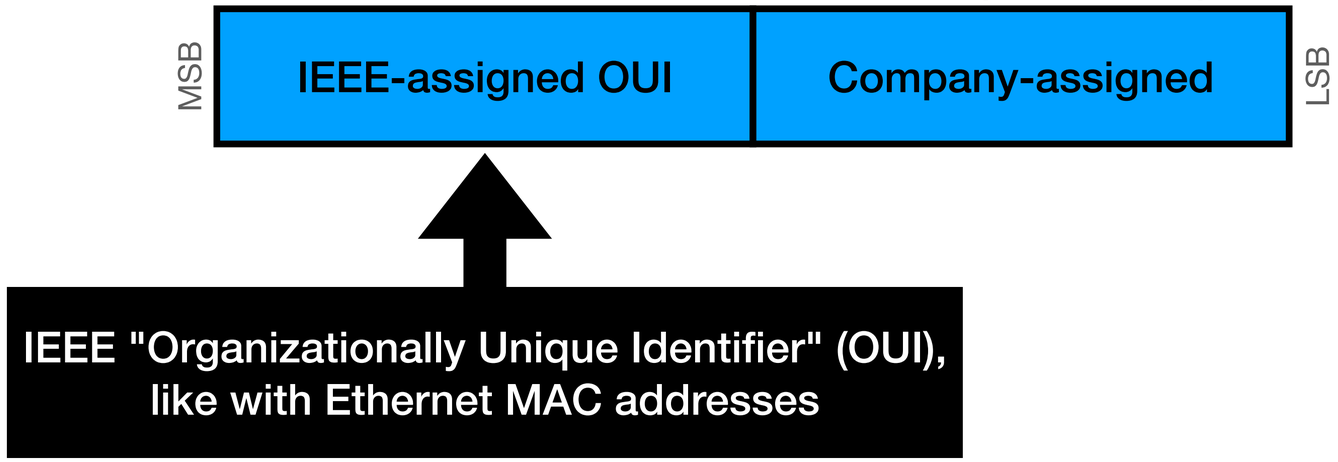
Background - BTC *BT Device Address (BDADDR)*



https://standards.ieee.org/faqs/regauth

Background - BTC *BT Device Address (BDADDR)*

BT Classic Addresses (48 bits)

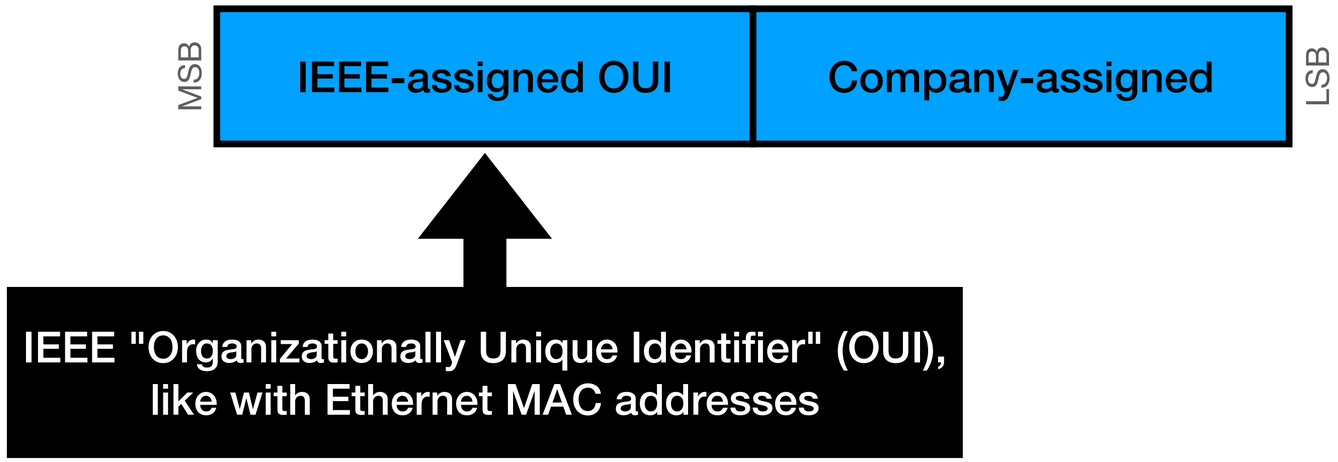


24 bits ("MA-L") - Most common case

https://standards.ieee.org/faqs/regauth

Background - BTC *BT Device Address (BDADDR)*

BT Classic Addresses (48 bits)

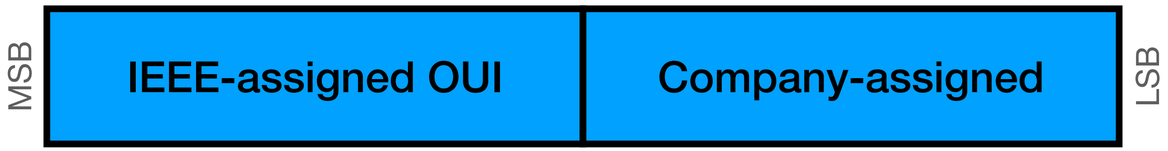


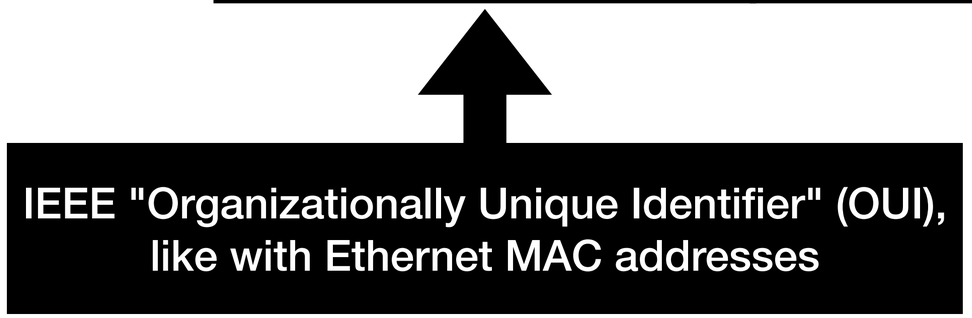
24 bits ("MA-L") - Most common case
20 bits ("MA-M")

https://standards.ieee.org/faqs/regauth

Background - BTC *BT Device Address (BDADDR)*

BT Classic Addresses (48 bits)





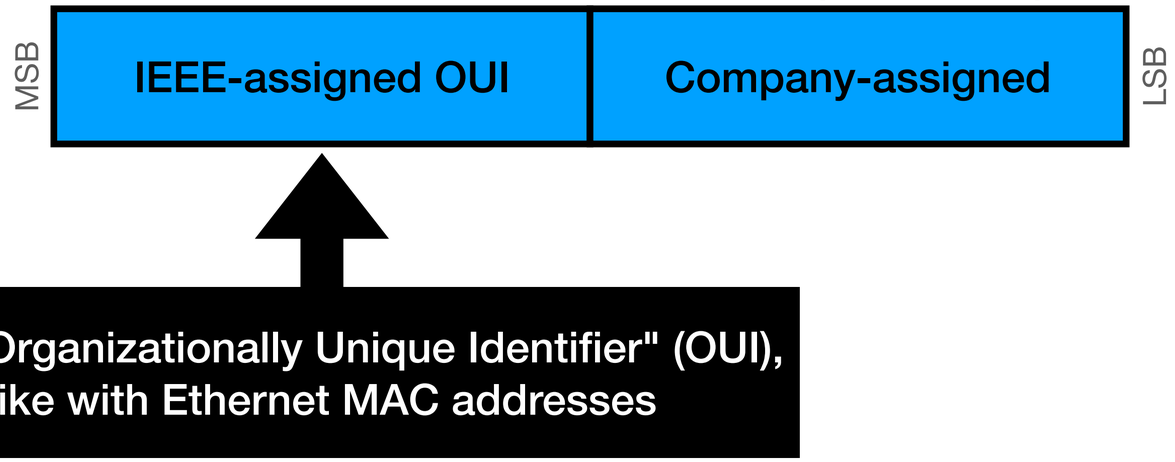
24 bits ("MA-L") - Most common case
20 bits ("MA-M")
12 bits ("MA-S")

https://standards.ieee.org/faqs/regauth

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")

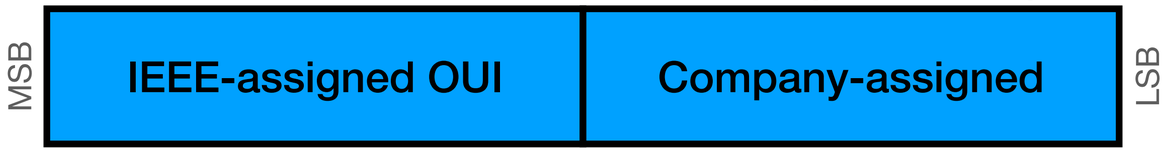
https://standards.ieee.org/faqs/regauth

Background - BTC *BT Device Address (BDADDR)*

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

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

BT Classic Addresses (48 bits)



IEEE "Organizationally Unique Identifier" (OUI), like with Ethernet MAC addresses

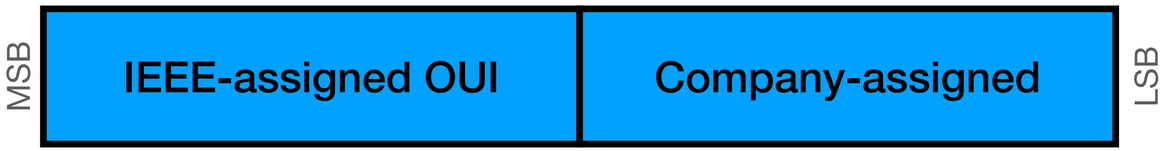
24 bits ("MA-L") - Most common case
20 bits ("MA-M")
12 bits ("MA-S")

https://standards.ieee.org/faqs/regauth

Background - BLE *BT Device Address (BDADDR)*

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

BLE Public Addresses (48 bits)



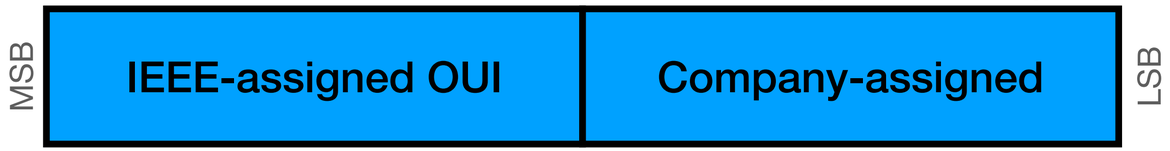
BLE Random Addresses

https://standards.ieee.org/faqs/regauth

Background - BLE *BT Device Address (BDADDR)*

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

BLE Public Addresses (48 bits)



BLE Random Addresses

BLE Random

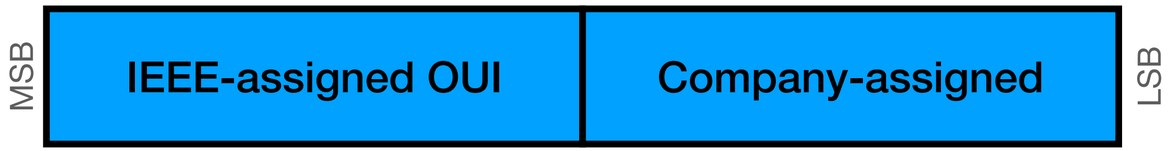
Static (48 bits)

https://standards.ieee.org/faqs/regauth

Background - BLE *BT Device Address (BDADDR)*

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

BLE Public Addresses (48 bits)

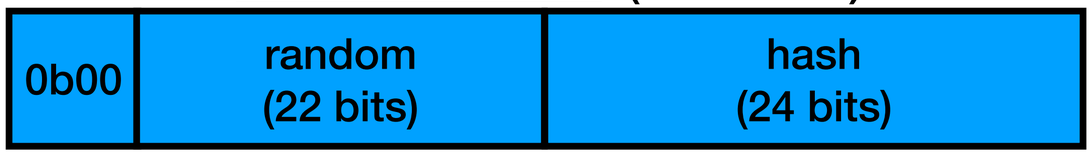


BLE Random Addresses

BLE Random
Static (48 bits)

BLE Random
Private

BLE Random Private
Resolvable (48 bits)

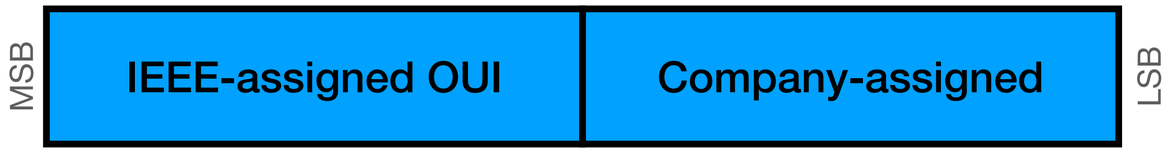


https://standards.ieee.org/faqs/regauth

Background - BLE *BT Device Address (BDADDR)*

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

BLE Public Addresses (48 bits)

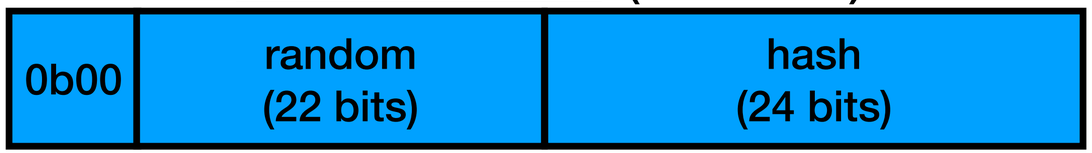


BLE Random Addresses

BLE Random
Static (48 bits)

BLE Random
Private

BLE Random Private
Resolvable (48 bits)



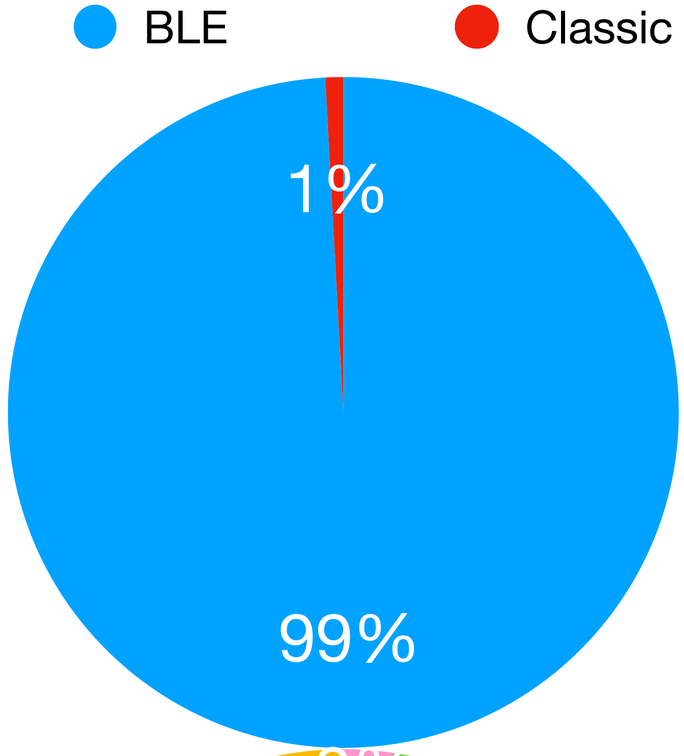
BLE Random Private
Non-resolvable (48 bits)

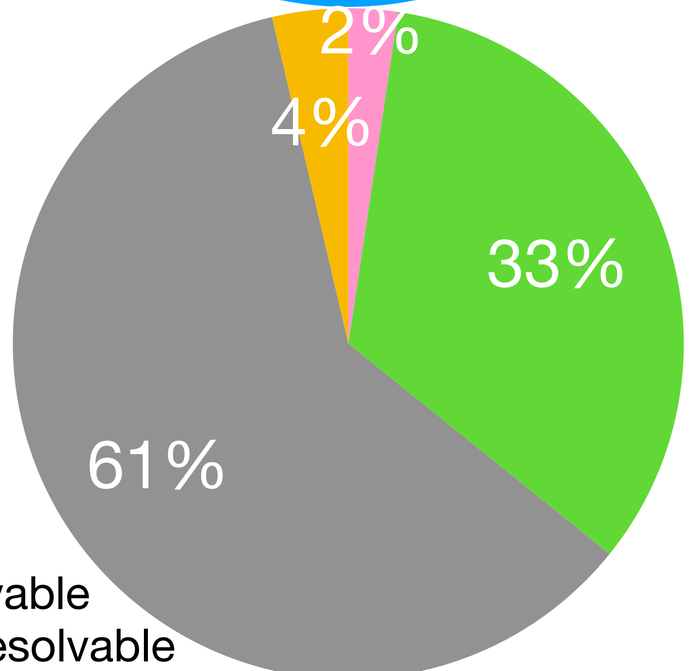
https://standards.ieee.org/faqs/regauth

Overall BDADDR Data

IEEE OUI Applicability

- **74,934 *unique* BT Classic BDADDRs**👍
- 8,569,483 *unique* BLE BDADDRs
  - **204,708 "public" BDADDRs**👍
  - ~~2,793,274 "random static" BDADDRs~~
  - ~~5,264,247 "random resolvable" BDADDRs~~
  - ~~307,030 "random non-resolvable" BDADDRs~~
- So OUIPrints applicable to only ~3.2% of the BDADDRs (74,934+ 204,708)/(74,934+8,569,483)





📱 = phone-related, 🚗 = car-related, 🎶 = audio-related, 🍪 = chip-related, 🧩 = module-related

BTC Data (top 20 of 604 companies seen)

	company_by_bdaddr		found
GPS/Watches/etc	Garmin International		10454
📱📺	Samsung Electronics Co.,Ltd	*Qualcomm, Broadcom, MediaTek, Samsung* 🍪	7057
📱	OnePlus Technology (Shenzhen) Co., Ltd		3757
🍪	Actions Semiconductor Co.,Ltd.(Cayman Islands)		3054
📱	Apple, Inc.	*Mostly Broadcom, Sometimes Apple (e.g. AirPods)* 🍪	2697
🎶🚗🧩	Panasonic Automotive Systems Co.,Ltd		2499
🎶🚗🧩	Laird Connectivity		2244
🍪	Intel Corporate		2042
🎶🚗🧩	PIONEER CORPORATION		1992
🚗	ALPSALPINE CO,.LTD		1944
📱	GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP.,LTD		1696
🧩🍪	AzureWave Technology Inc.	*NXP, Cypress, MediaTek* 🍪	1378
🍪	Texas Instruments		995
📱	Wistron Neweb Corporation		932
🧩	silex technology, Inc.	*Qualcomm* 🍪	893
🎶🚗🧩	Shinwa Industries(China) Ltd.	*Qualcomm/CSR, TI, Cypress, Sunplus* 🍪	861
🎶🚗🧩	MITSUMI ELECTRIC CO.,LTD.	*CSR->Qualcomm* 🍪	849
🎶🚗🧩	PARROT SA	*TI* 🍪	749
📱 integrator?	Wingtech Mobile Communications Co., Ltd.	*MediaTek?* 🍪	692
🎶🚗🍪	Sunplus Technology Co., Ltd.		608

📱 = phone-related, 🚗 = car-related, 🎶 = audio-related, 🍪 = chip-related, 🧩 = module-related

BTC Data (top 20 of 604 companies seen)

	company_by_bdaddr		found
GPS/Watches/etc	Garmin International		10454
📱📺	Samsung Electronics Co.,Ltd	*Qualcomm, Broadcom, MediaTek, Samsung* 🍪	7057
📱	OnePlus Technology (Shenzhen) Co., Ltd		3757
🍪	Actions Semiconductor Co.,Ltd.(Cayman Islands)		3054
📱	Apple, Inc.	*Mostly Broadcom, Sometimes Apple (e.g. AirPods)* 🍪	2697
🎶🚗🧩	Panasonic Automotive Systems Co.,Ltd		2499
🎶🚗🧩	Laird Connectivity		2244
🍪	Intel Corporate		
🎶🚗🧩	PIONEER CORPORATION		
🚗	ALPSALPINE CO,.LTD		
📱	GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP.,LTD		1696
🧩🍪	AzureWave Technology Inc.	*NXP, Cypress, MediaTek* 🍪	1378
🍪	Texas Instruments		995
📱	Wistron Neweb Corporation		932
🧩	silex technology, Inc.	*Qualcomm* 🍪	893
🎶🚗🧩	Shinwa Industries(China) Ltd.	*Qualcomm/CSR, TI, Cypress, Sunplus* 🍪	861
🎶🚗🧩	MITSUMI ELECTRIC CO.,LTD.	*CSR->Qualcomm* 🍪	849
🎶🚗🧩	PARROT SA	*TI* 🍪	749
📱 integrator?	Wingtech Mobile Communications Co., Ltd.	*MediaTek?* 🍪	692
🎶🚗🍪	Sunplus Technology Co., Ltd.		608

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

📱 = phone-related, 🚗 = car-related, 🎶 = audio-related, 🍪 = chip-related, 🧩 = module-related

BLE Data (top 20 of 631 companies seen)

	company_by_bdaddr		found	
🍪	Texas Instruments		32252	
🎶	VXi Corporation		25243	
📱🖥️	Samsung Electronics Co.,Ltd	*Qualcomm, Broadcom, MediaTek, Samsung* 🍪	13165	
🎶	Bose Corporation		8584	
🐀⌨️	Logitech, Inc		5031	
🚗	Cambridge Mobile Telematics, Inc.	*CSR->Qualcomm* 🍪	5024	
📱	Apple, Inc.	*Mostly Broadcom, Sometimes Apple (e.g. AirPods)* 🍪	4970	
🍪	Silicon Laboratories		3849	
🍪	Espressif Inc.		3027	
BLE beacons	Shenzhen Minew Technologies Co., Ltd.		2609	
🧩	Murata Manufacturing Co., Ltd.	*Nordic, Cypress, CSR, Onsemi, Dialog* 🍪	2288	
🍪	Telink Semiconductor (Taipei) Co. Ltd.		1965	
🍪🧩 + 🍪?	Shenzhen Jingxun Software Telecommunication Technology Co.,Ltd		1891	*RealTek, Airoha* 🍪
GPS/Watches/etc	Garmin International		1740	
🧩	Sunitec Enterprise Co.,Ltd	*Broadcom* 🍪	1442	
BLE beacons	Aruba, a Hewlett Packard Enterprise Company		1409	
LED lights & sensors	Shenzhen Intellirocks Tech co.,ltd	*Telink* 🍪	1164	
Teslas	Shanghai Rui Rui Communication Technology Co.Ltd.		1073	(TI bought OUI?)
🎶🚗🧩	ALPSALPINE CO,.LTD		979	
🎶 Hearing aids	Starkey Labs Inc.		956	

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

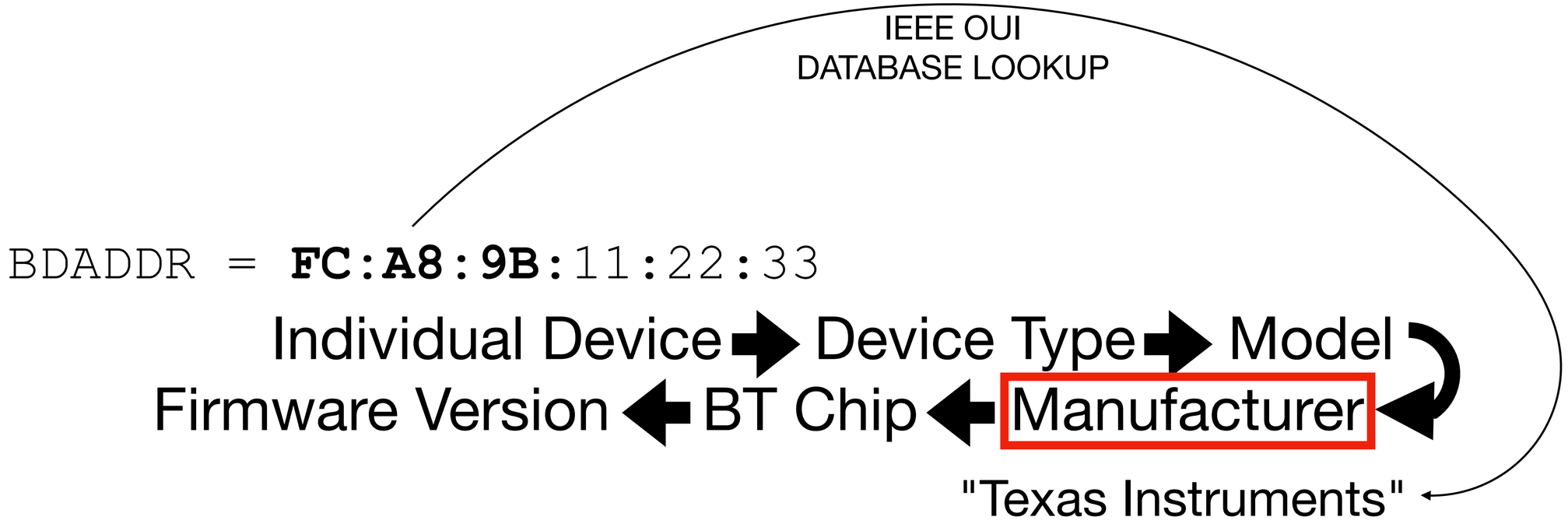
Individual Device → Device Type → Model ↷
Firmware Version ← BT Chip ← Manufacturer

What I Want

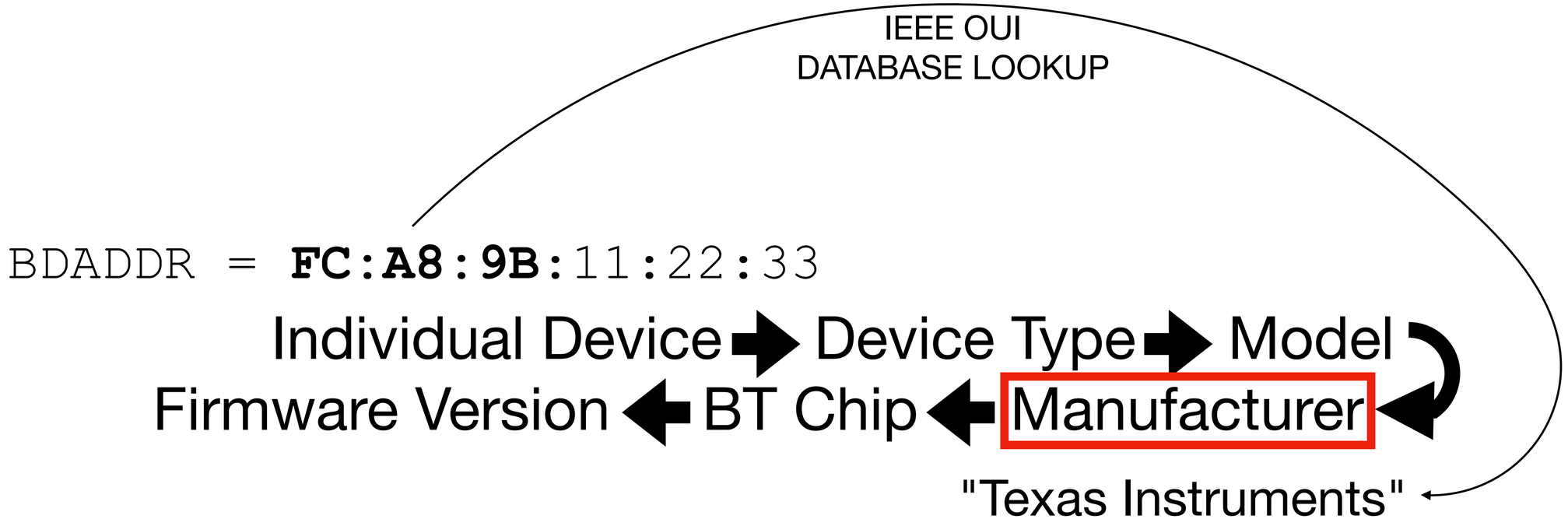
`BDADDR = FC:A8:9B:11:22:33`

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

What I Want

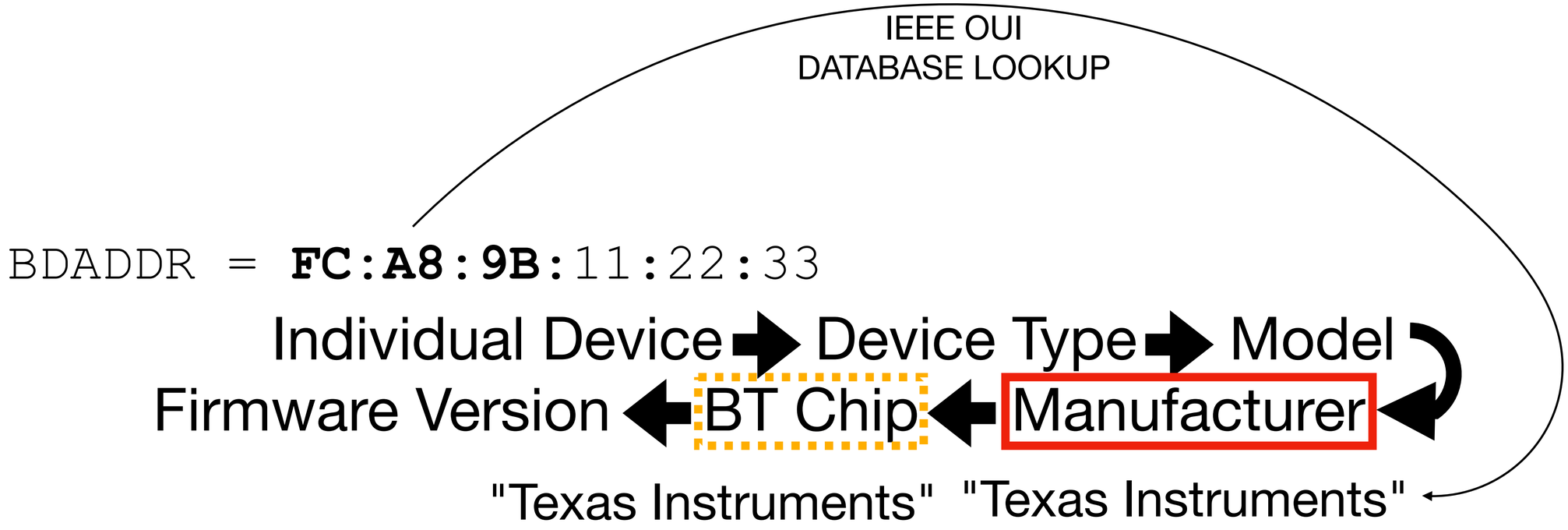

IEEE OUI
DATABASE LOOKUP
BDADDR = FC:A8:9B:11:22:33
Individual Device
Device Type
Model
Firmware Version
BT Chip
Manufacturer
"Texas Instruments"

What I Want



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

What I Want



ASSUMPTION:

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

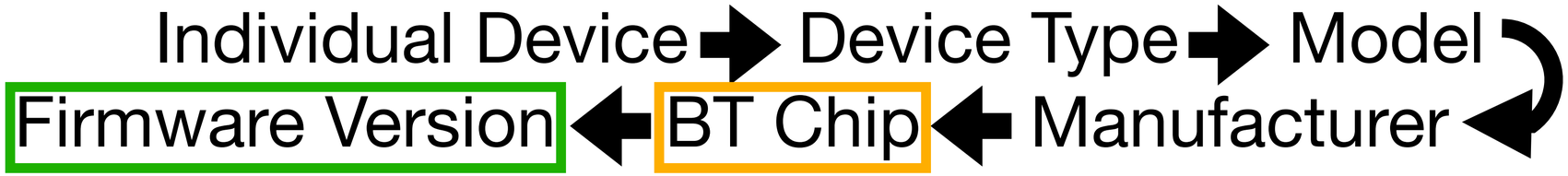
2thprint by Link Layer Version Info

🚶 or 🤾

2thprint by Link Layer Version Information

LL_VERSION_IND (BLE), LMP_version_res (BTC)

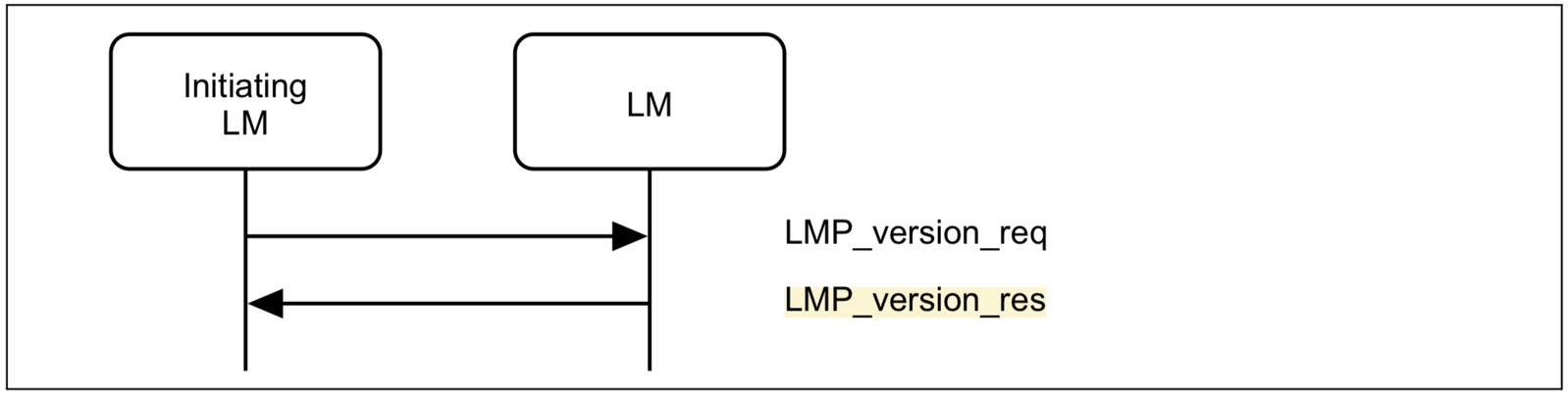
- BTC has Link Management Protocol (LMP) and BLE has Link-Layer (LL) Control packets, that can be sent to request some chip and firmware version information
- These can be sent/received *without* any sort of BT pairing/bonding!



2thprint by
LMP_version_res (BTC)

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

Table 4.26: PDUs used for LMP version request



Sequence 77: Request for LMP version

From BT
Core Spec v4.2

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)	CompId (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).
- CompId 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.

From BT
Core Spec v4.2

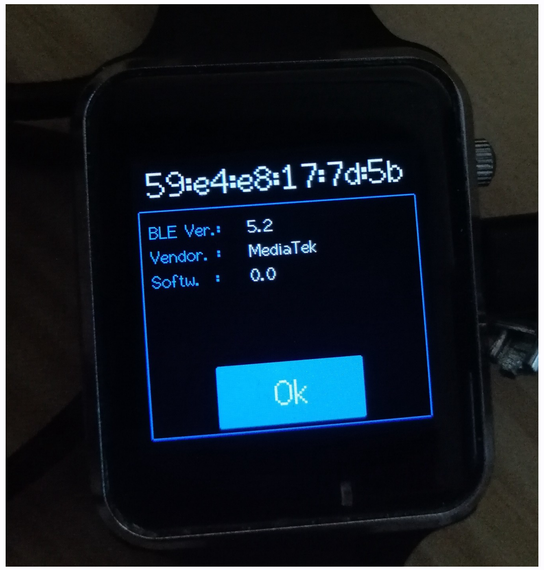
Prior Work

"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

Remote BLE stack fingerprinting !

EURECOM Quarkslab



[1] https://wootconference.org/papers/woot23-paper22.pdf

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

[1] https://wootconference.org/papers/woot23-paper22.pdf

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

OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA$649.99

Color: Halo Green

ROM

8 GB RAM + 128 GB Storage

...but not definitive

...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

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

...

OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA$649.99

Color: Halo Green

ROM

8 GB RAM + 128 GB Storage

OnePlus Trade-In Open for Every

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

...

OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA$649.99

Color: Halo Green

ROM

8 GB RAM + 128 GB Storage

62
15
14
11
10
10
9
8
7
5
4
4
4
4
4

OnePlus Pad

Halo Green | 8 GB RAM + 128 GB Storage

CA$649.99

Color: Halo Green

ROM

8 GB RAM + 128 GB Storage

OnePlus Trade-In Open for Every

62
15
14
11
10
10
9
8
7
5
4
4
4
4
4

(Link Layer) Version Number

VersNr (1 byte)

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

Figure 2.[illegible]: CtrData field of the LL_VERSION_IND PDU

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

From BT
Assigned Numbers document

Vendor Prevalence

BLE - 2024-01-12

- 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 LL 2thprint 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 Jieli 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

Vendor Prevalence

BLE - 2024-01-12

- 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 LL 2thprint 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 Jieli 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 - 2024-01-12

- 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 Jieli 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
Equinux 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 - 2024-01-12

- 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!

comp_by_CompId	count
Qualcomm	35
MediaTek, Inc.	26
Broadcom Corporation	25
Intel Corp.	8
Qualcomm Technologies International, Ltd. (QTIL)	7
Zhuhai Jieli 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
Equinux 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!

From BT
Assigned Numbers document

(Link I

SubVersI

- The venc

2.4.2.13 LL_VERSION_IND

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

CtrData		
VersNr (1 octet)	CompId (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).
- CompId 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.

From BT
Assigned Numbers document

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

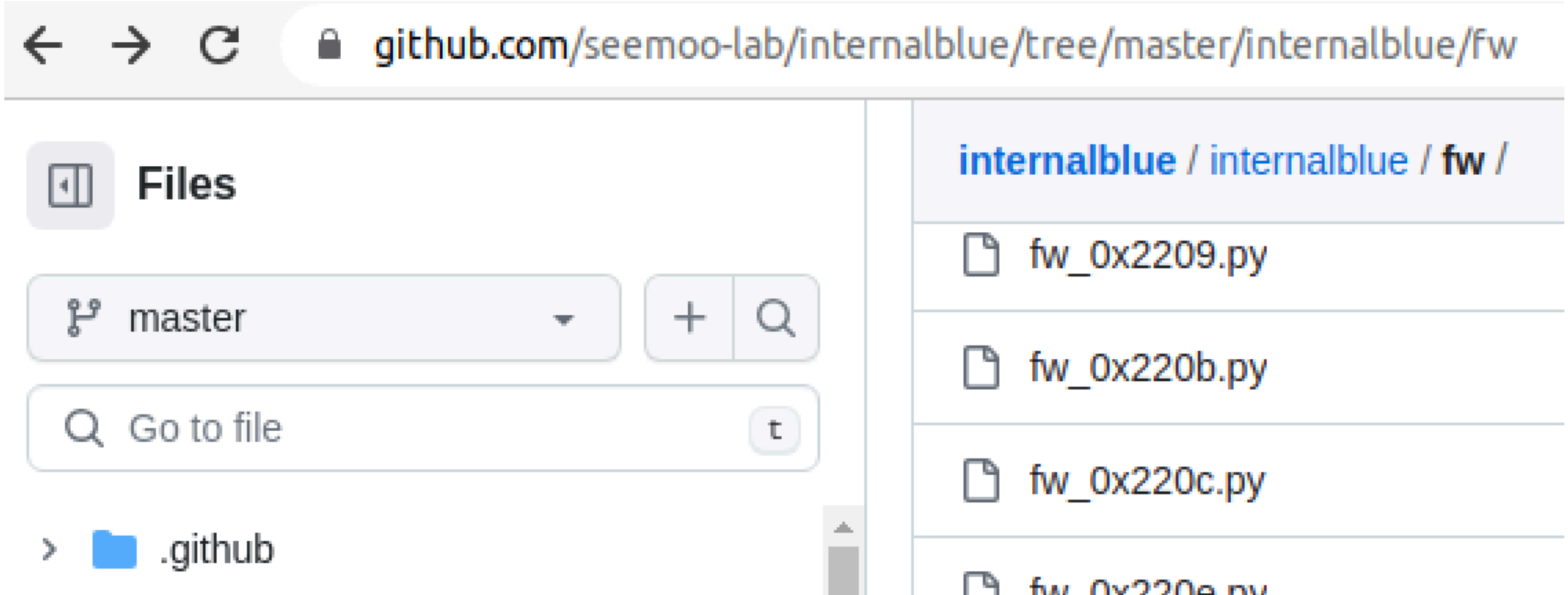
Does SubVersNr Imply OS/Firmware Version?

- 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!



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

- fw_0x2033.py
- fw_0x203a.py
- fw_0x2056.py
- fw_0x21a9.py
- fw_0x21d0.py
- fw_0x2209.py

```
} 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_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...)

VersionPrint

Ver. 1 | Ver. 2 | Ver. 3 | Ver. N

WHAT I WANT!

ChipPrint

Chip 1 | Chip 2 | Chip 3 | Chip N

WHAT I MOSTLY GET 😔

ChipMakerPrint

ChipMaker 1 | ChipMaker 2 | ChipMaker N

ModulePrint

Module 11 | Module 13 | Module NA | Module NX

ModuleMakerPrint

ModuleMaker 1 | ModuleMaker 2 | ModuleMaker N

ProductPrint

Product 1 | Product 2 | Product N

ProductMakerPrint

ProductMaker 1

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?

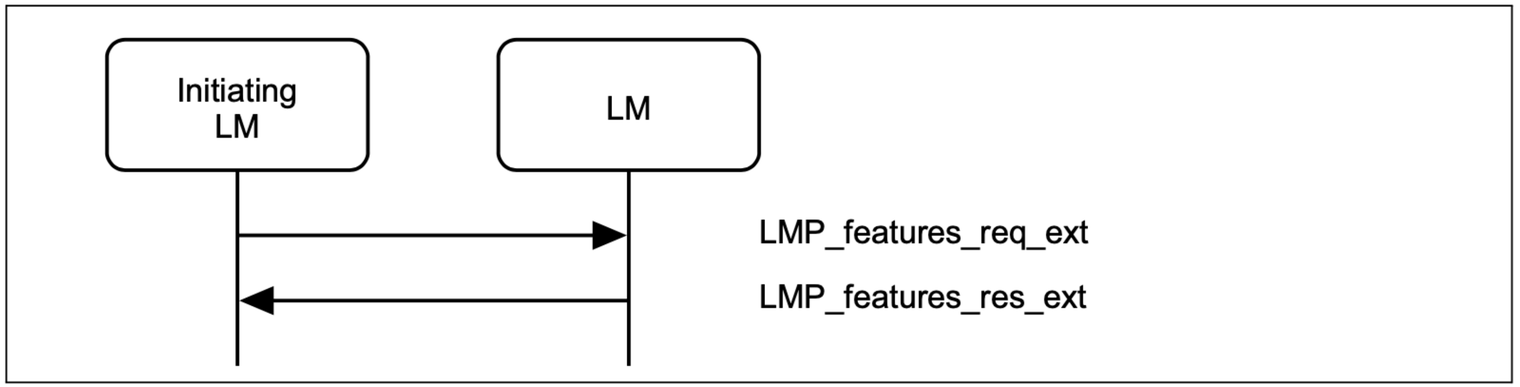
- **Hypothesis:** SubVersNr will increment per OS/firmware update, and thus can be used to infer the OS/firmware version
  - Overall Conclusion:
    - Requires more investigation!
      - Fundamentally requires hands on with a device + manually updating firmware and observing how the sub-version-number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

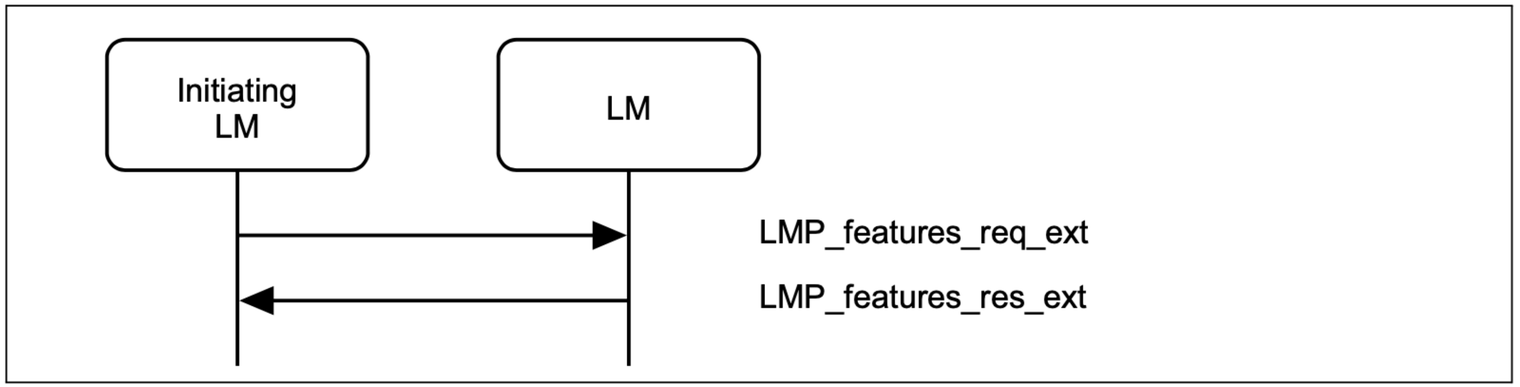
"*malformed*" LMP_features_ext_req?



Sequence 79: Request for extended features

From BT
Core Spec v4.2

"*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

From BT Core Spec v4.2

"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)

From BT
Core Spec v4.2

"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)

From BT
Core Spec v4.2

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)

From BT
Core Spec v4.2

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 definitio (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 definitio (page 2)

From BT
Core Spec v4.2

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. "

From BT
Core Spec v4.2

Pick a page, any pag

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

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. "

From BT
Core Spec v4.2

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 ¯_(ツ)_/¯. 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)

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

Data Types Specification

Bluetooth®

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.

	Context				
Data type	**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)

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

Data Types Specification

Bluetooth®

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's *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!

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» *uint16*, which may be followed by *struct*	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's *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!

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» *uint16*, which may be followed by *struct*	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's *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 Ass Numbers. The interpretation of any other octets within the data shall d by the manufacturer specified by the company identifier.

It doesn't say what endianness it should use!

1.4.2 Format

Data Type	Description
«Manufacturer Specific Data» *uint16*, which may be followed by *struct*	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 - 2024-01-12

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 - 2024-01-12

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

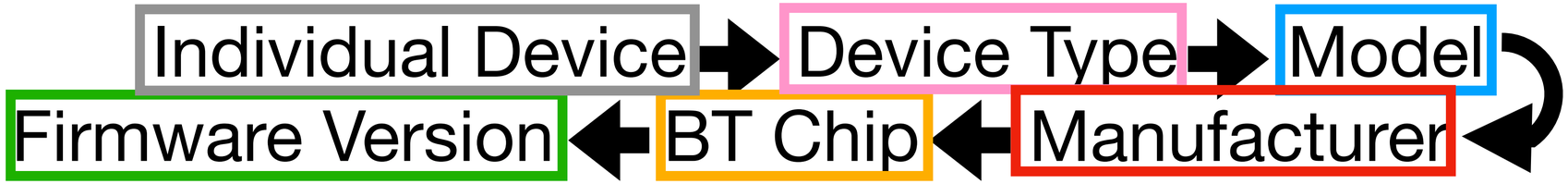
 *

* I would tend to argue though that GATT UUID16/UUID128s included in ADV_IND or SCAN_RSP are passive and semi-passive respectively

2thprint by GATT

GATTPrint

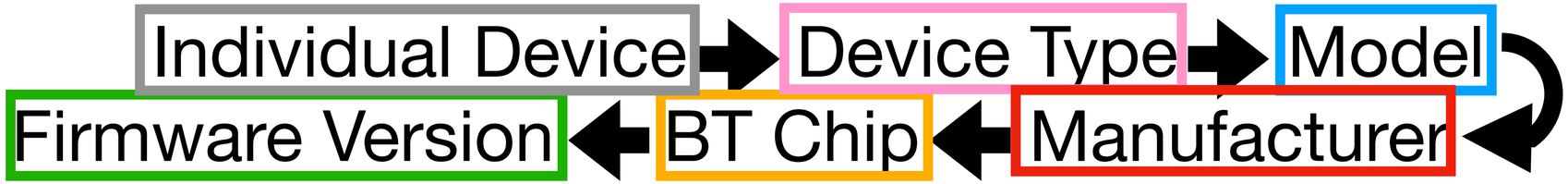
- **G**eneric **Att**ribute *Profile* (GATT) is a sort of weird thing that's built on top of Attribute Protocol (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information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



2thprint by GATT

GATTPrint

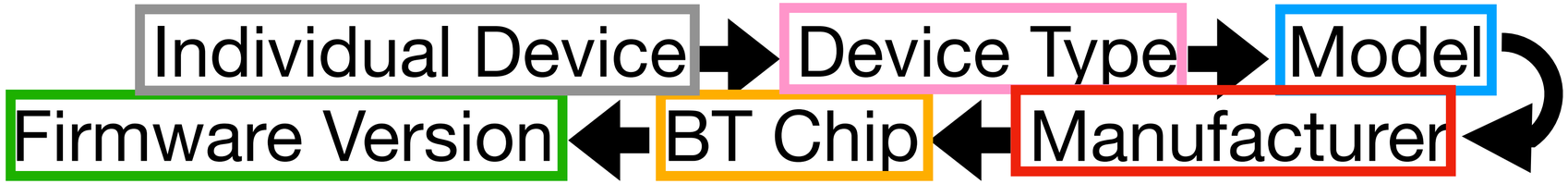
- **G**eneric **Att**ribute *Profile* (GATT) is a sort of weird thing that's built on top of Attribute Protocol (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information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



2thprint by GATT

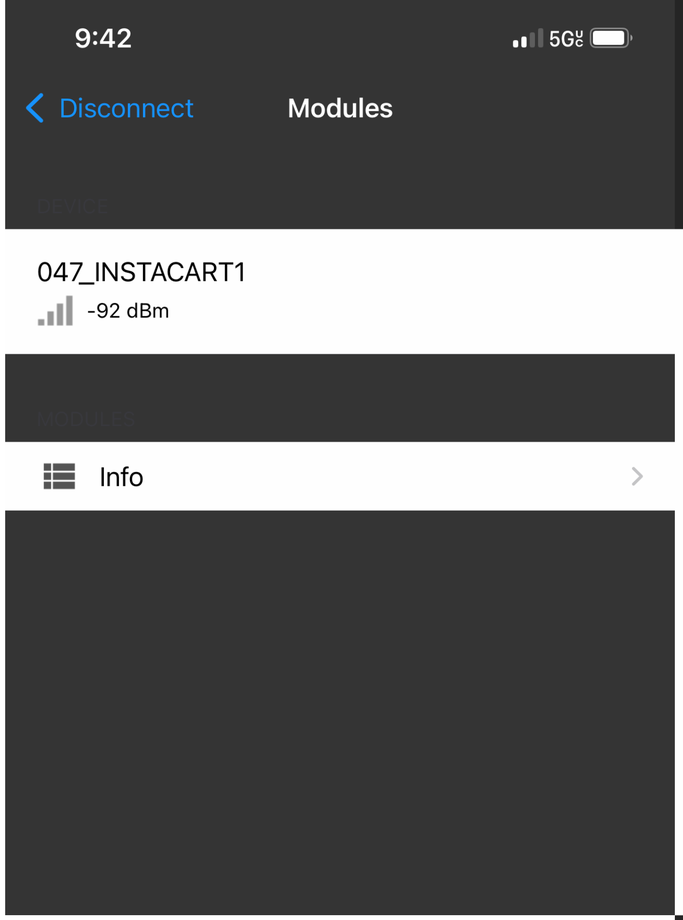
GATTPrint

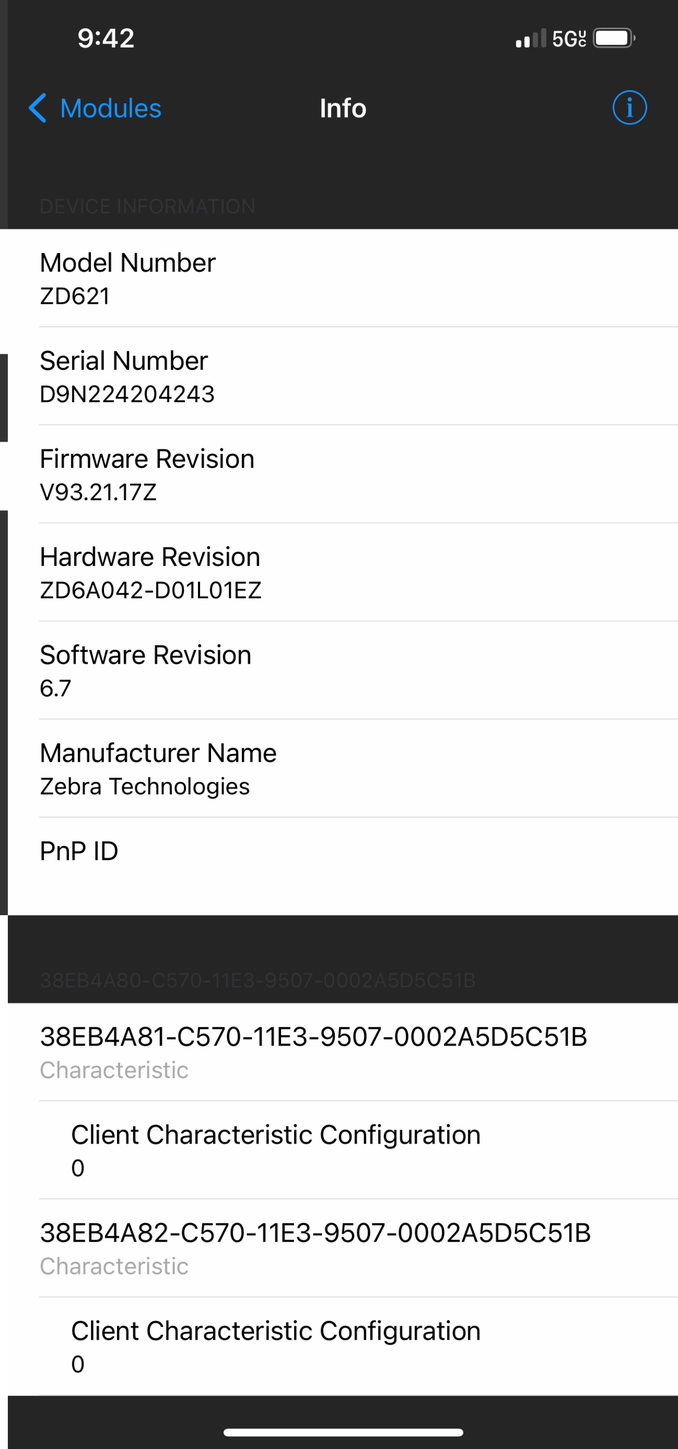
- **G**eneric **Att**ribute *Profile* (GATT) is a sort of weird thing that's built on top of Attribute Protocol (ATT), which is the actual protocol for sending & receiving data
- Mostly used on BLE, though it can technically be used on BTC devices too
- You can *theoretically* get ALL the types of information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



GATT on Phones

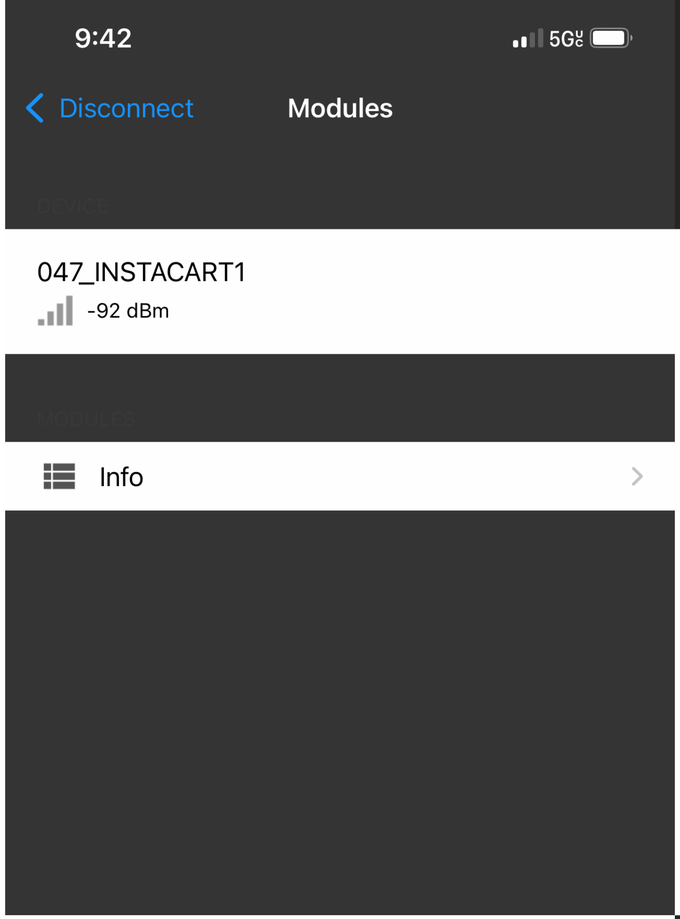
- When you open a Bluetooth scanner app like LightBlue or BluefruitConnect on your phone, and they show you information, usually that is GATT information
- This is from BluefruitConnect

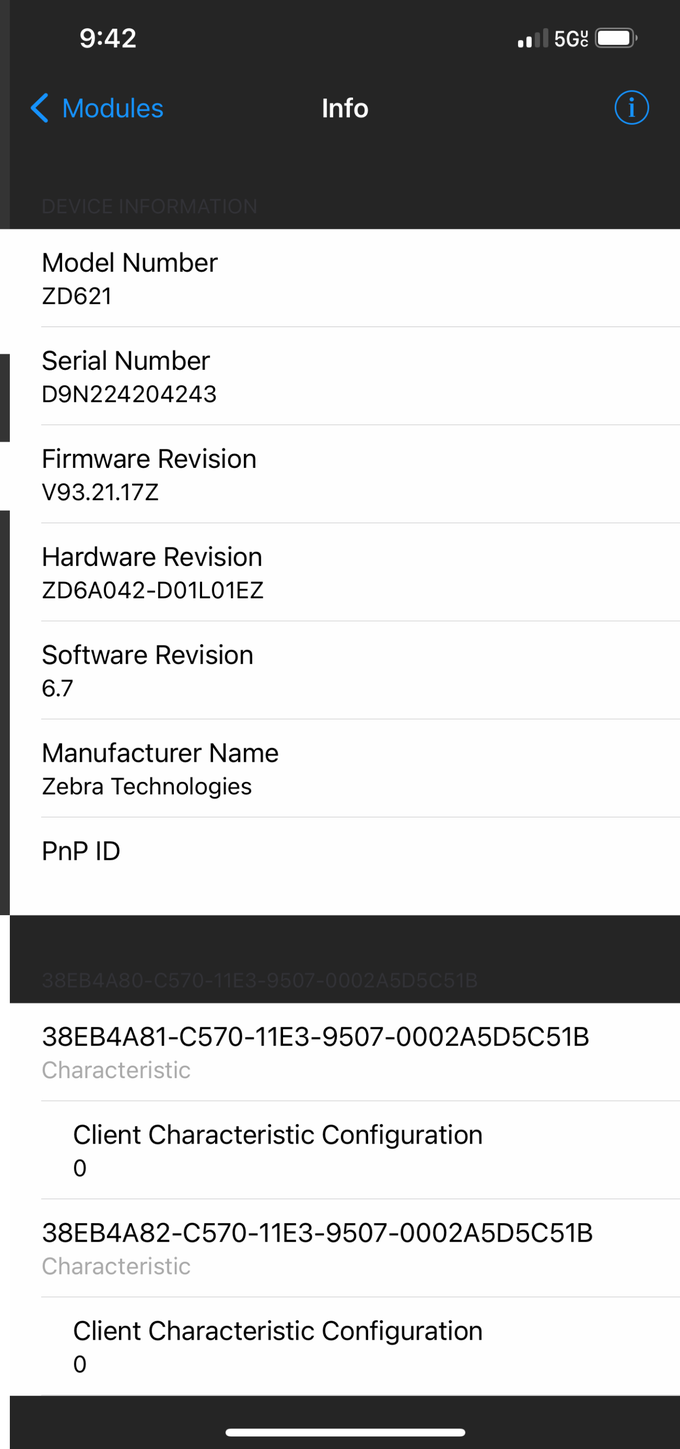




GATT on Phones

- 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

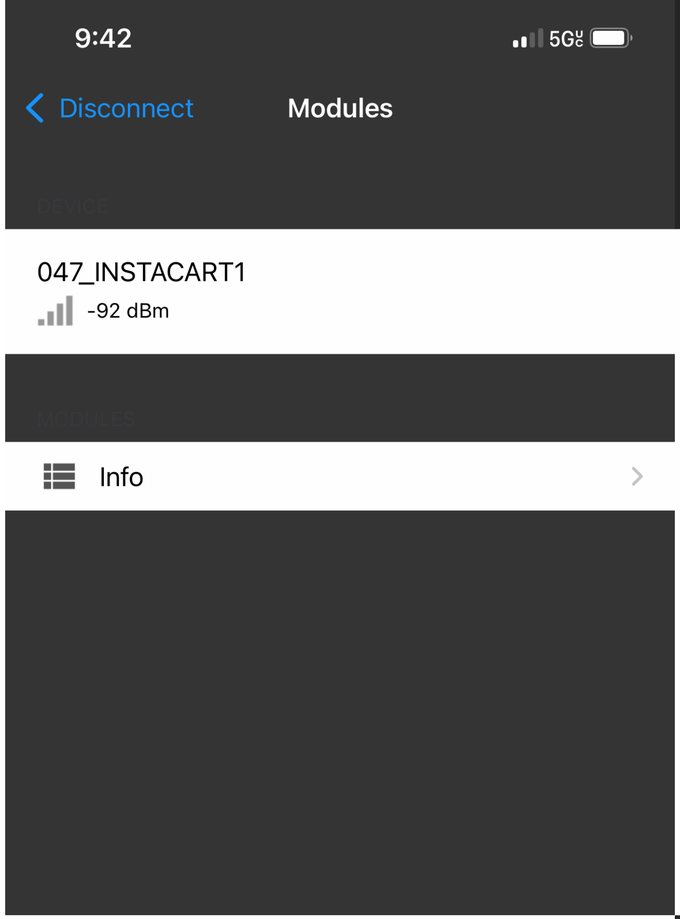


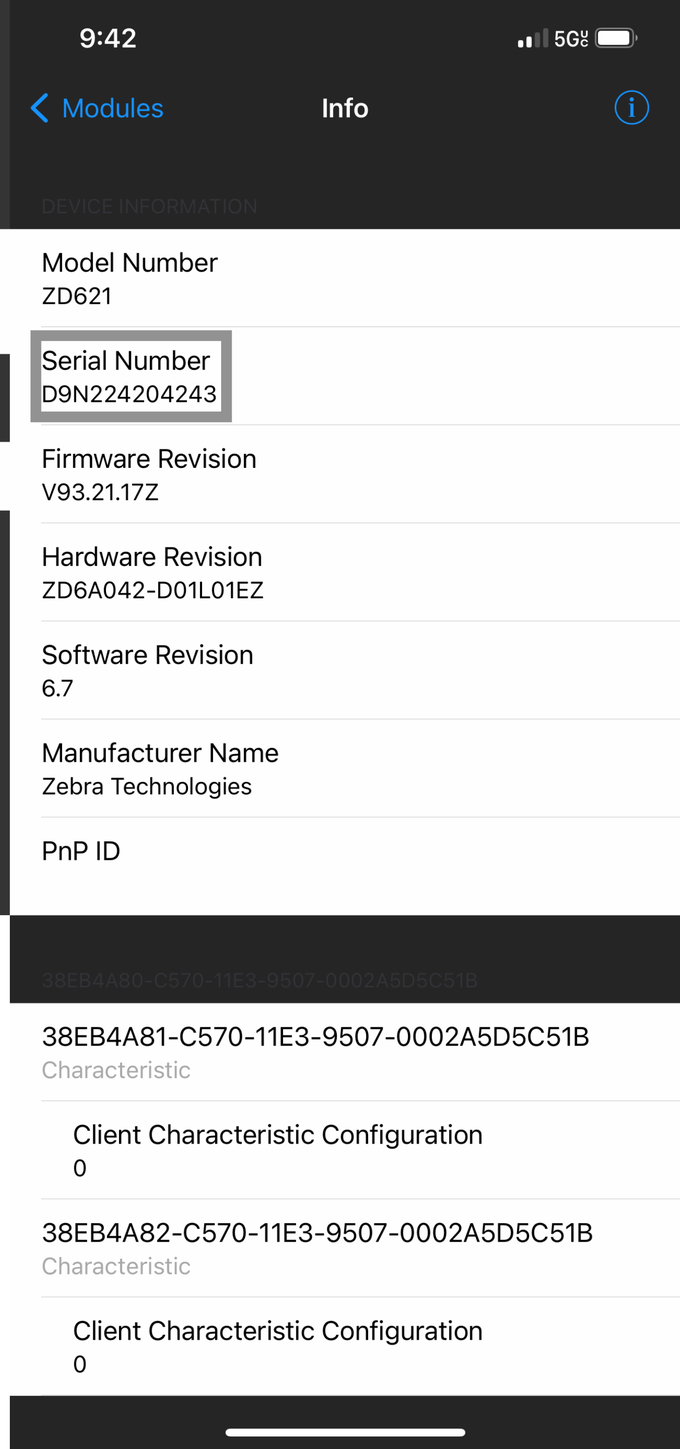


Individual Device → Device Type → Model → Manufacturer → BT Chip → Firmware Version

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

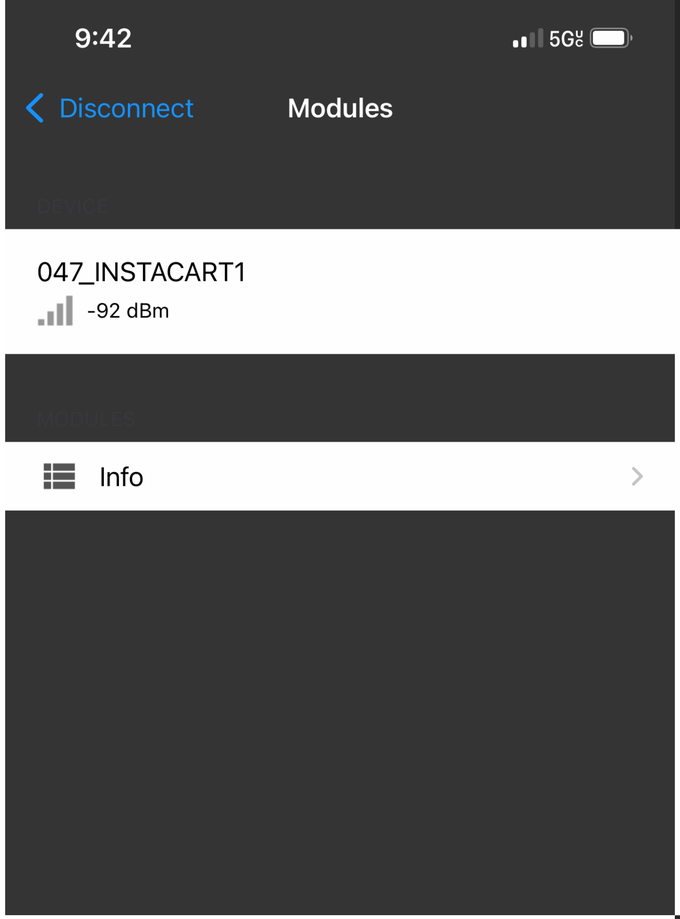


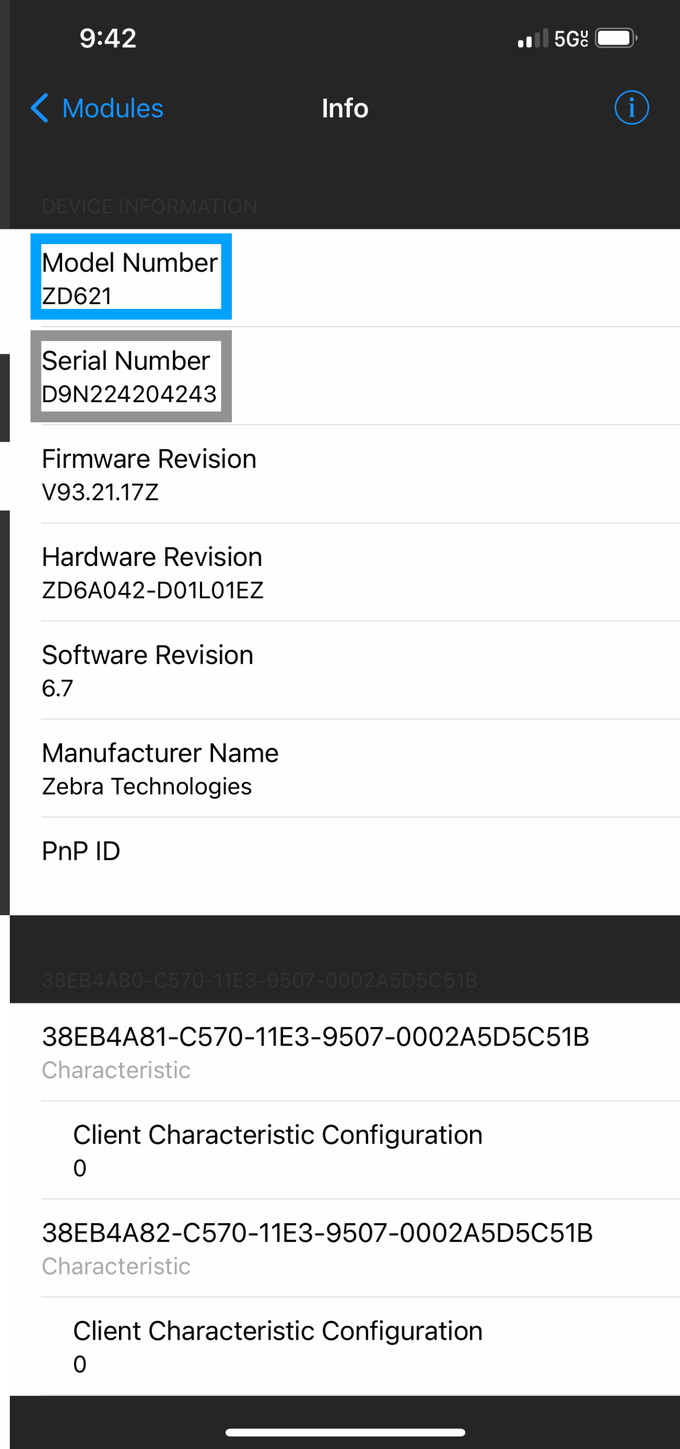


Individual Device → Device Type → Model → Manufacturer → BT Chip → Firmware Version

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

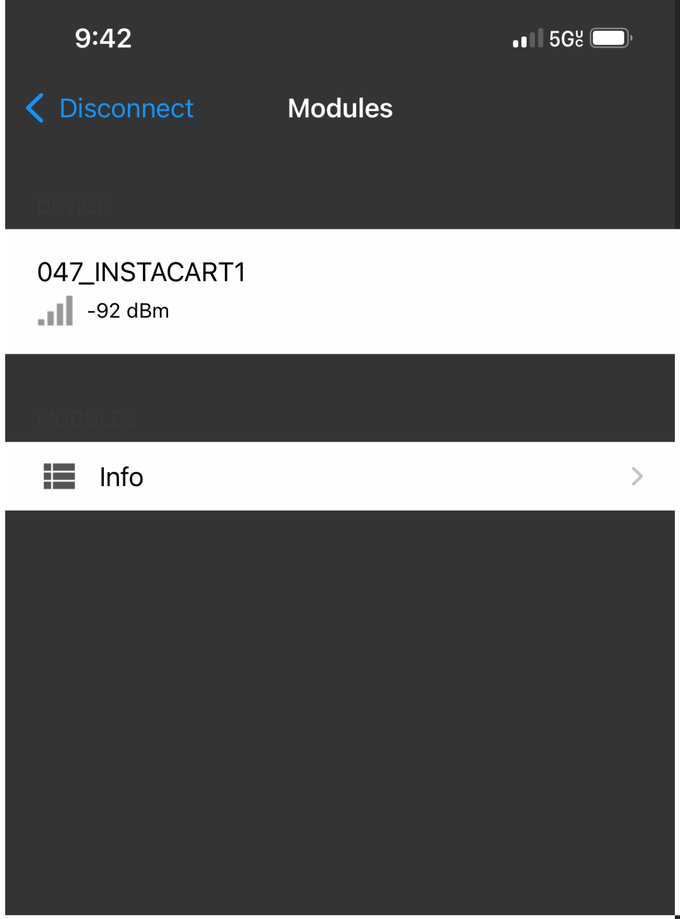


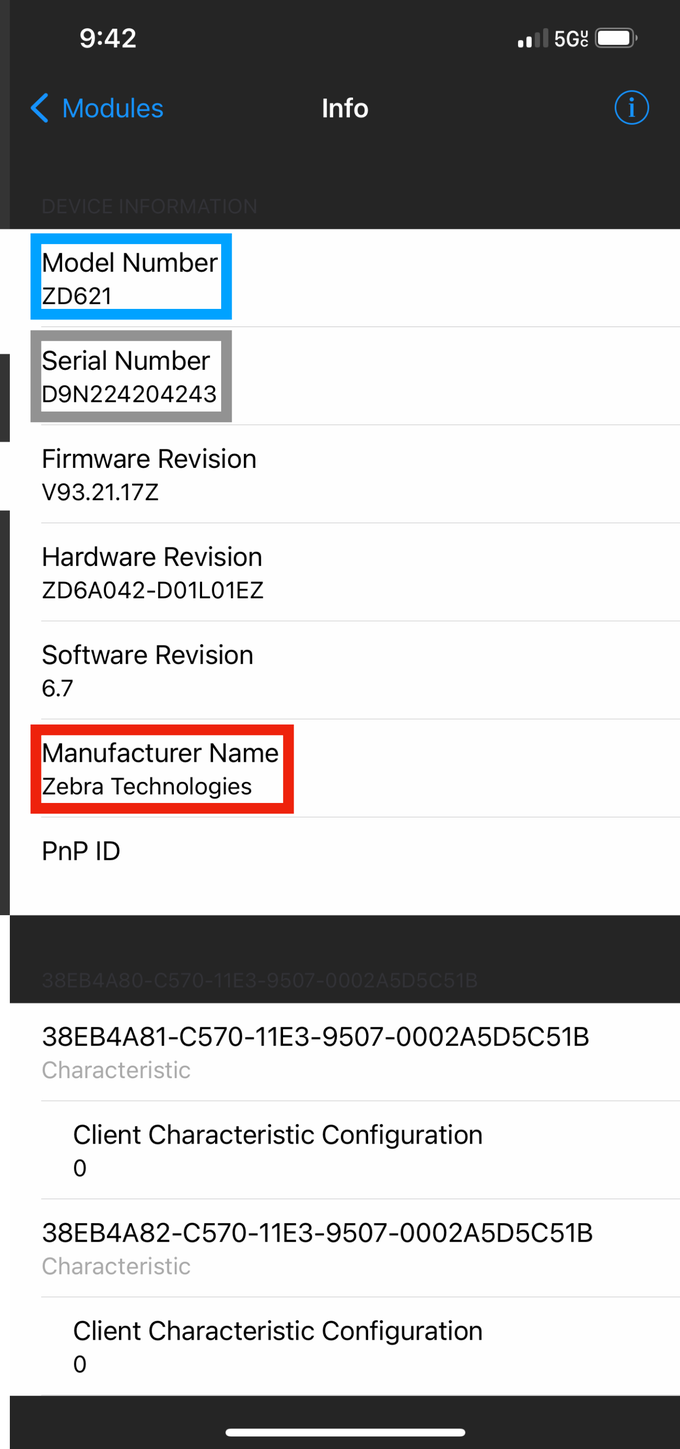


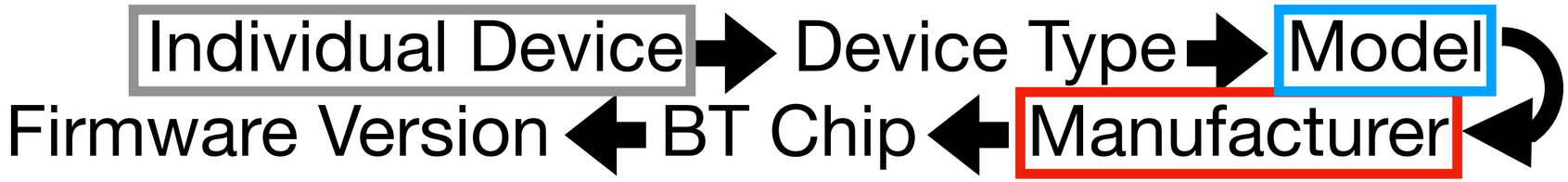
Individual Device → Device Type → Model → Manufacturer → BT Chip → Firmware Version

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

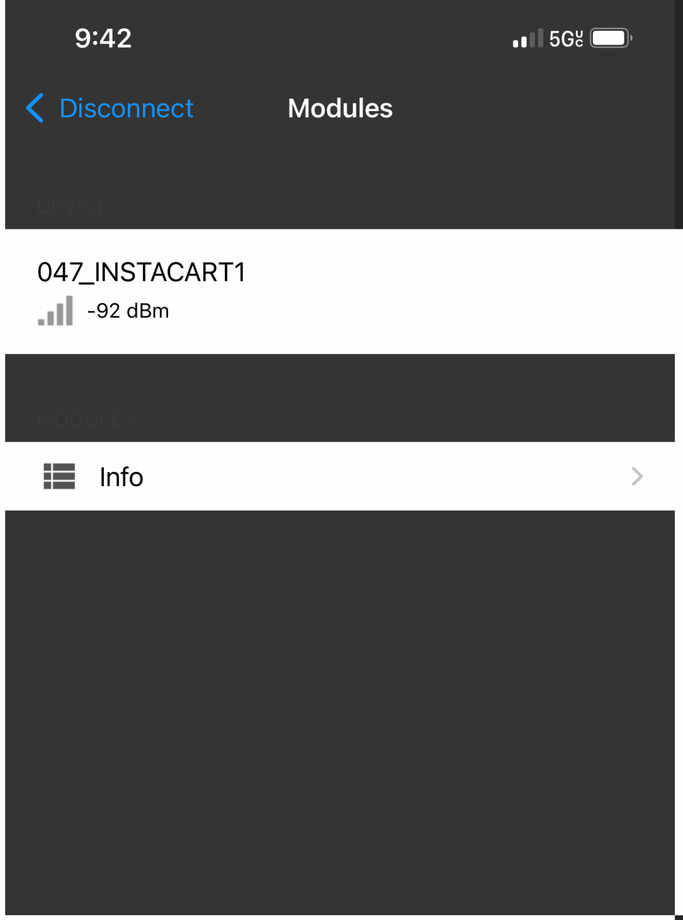


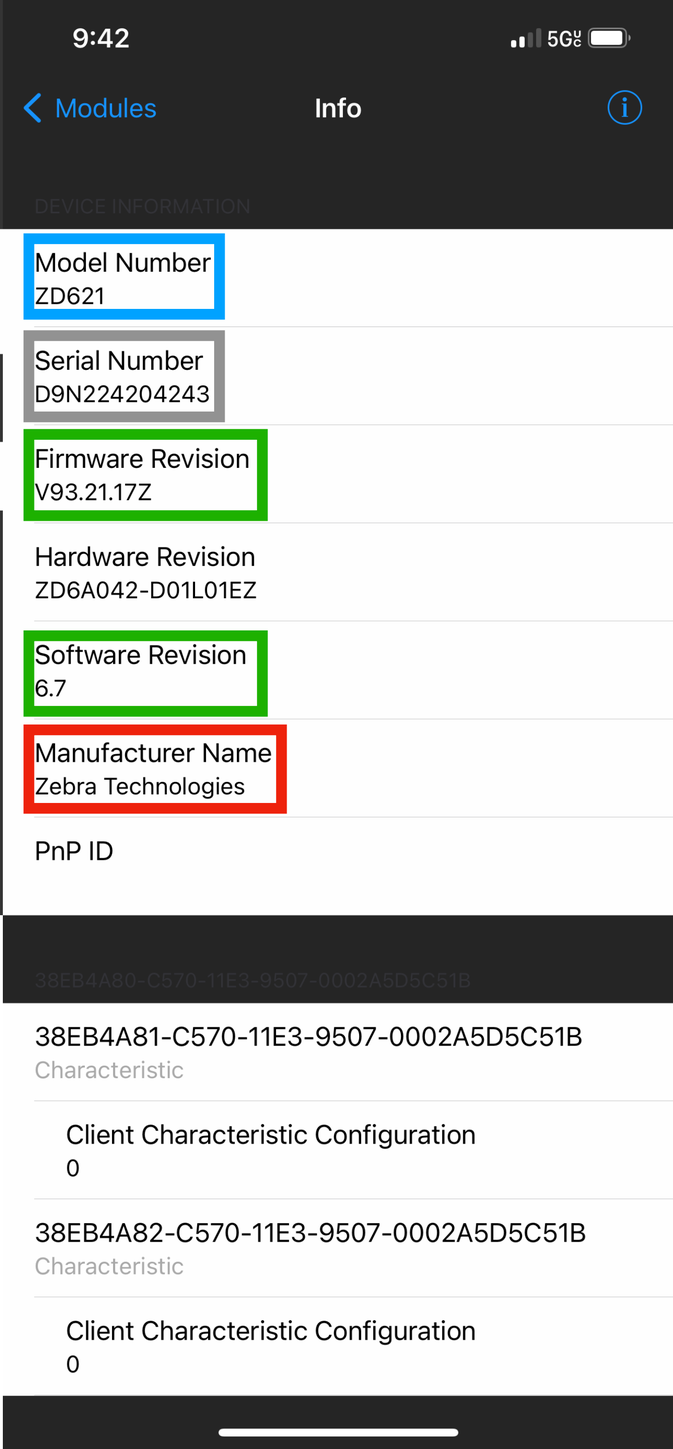


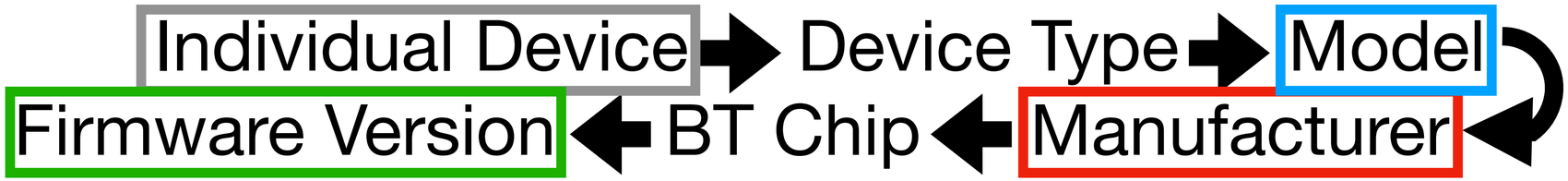


GATT on Phones

- 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

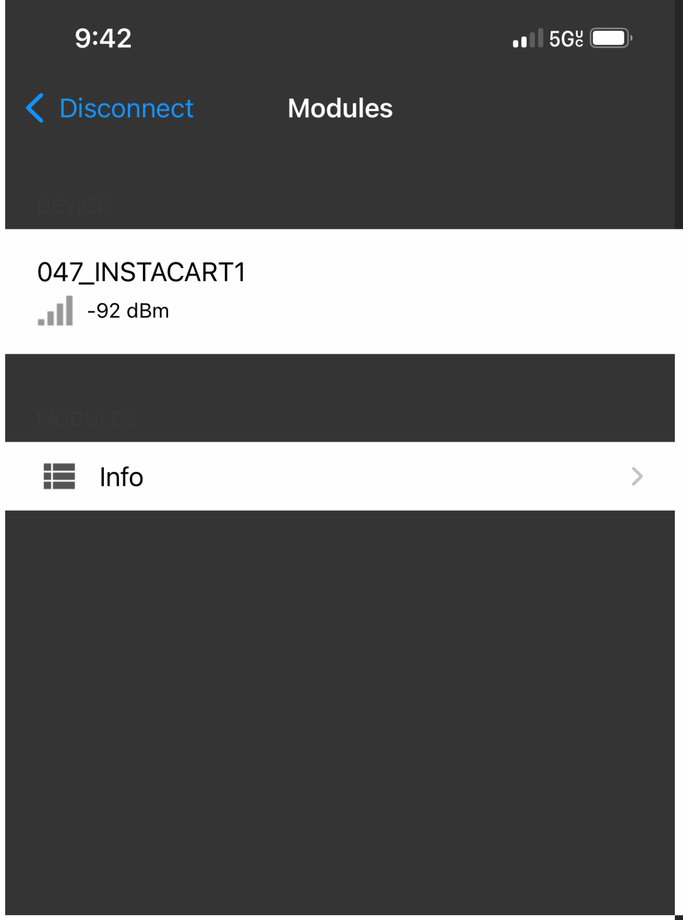


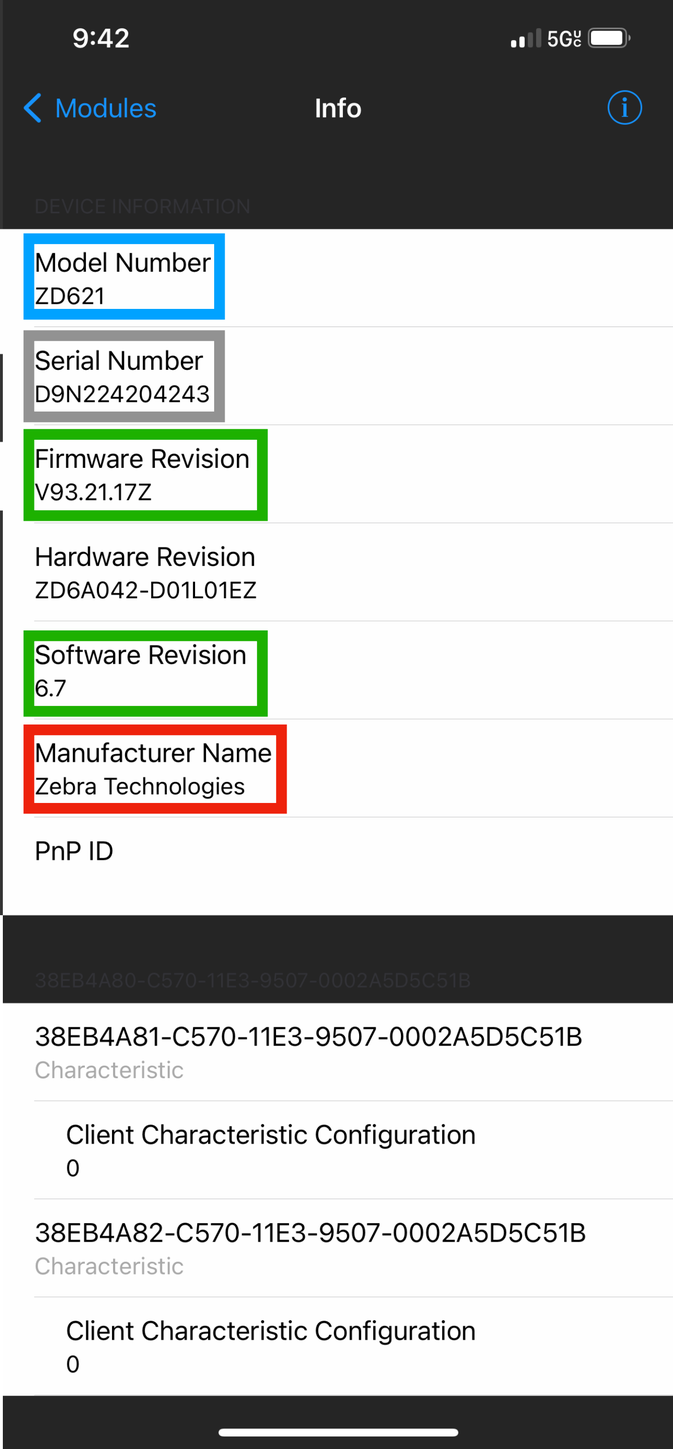


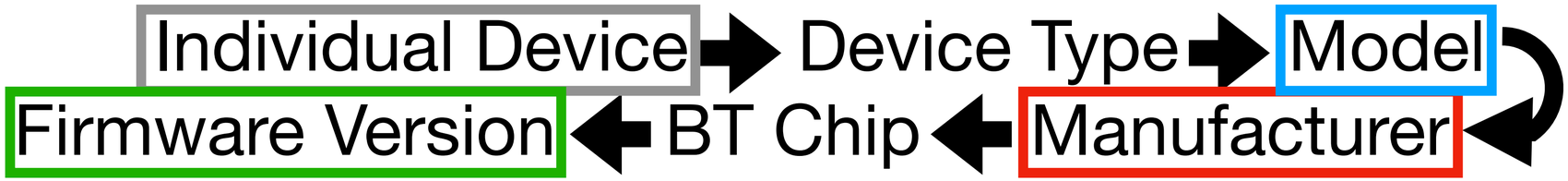


GATT on Phones

- 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

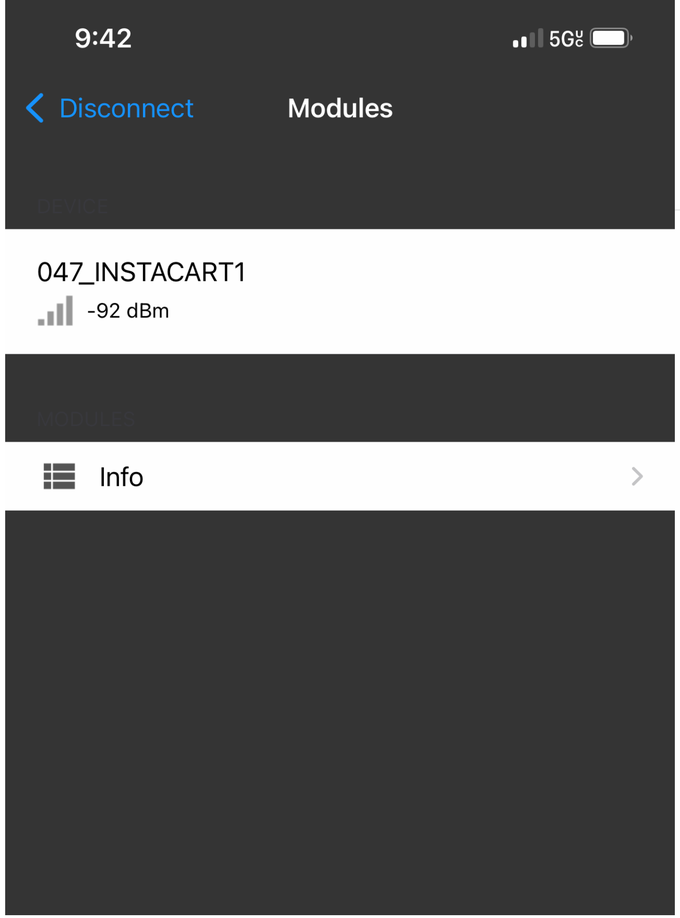


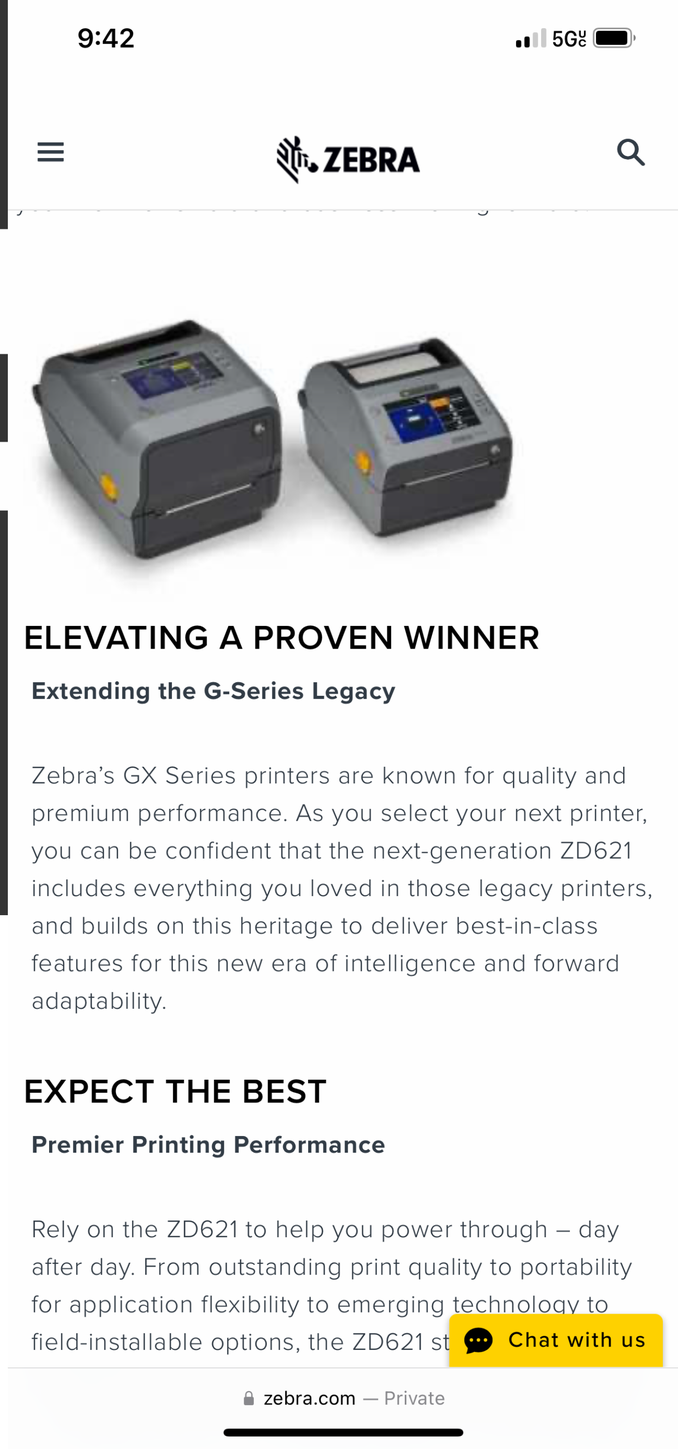


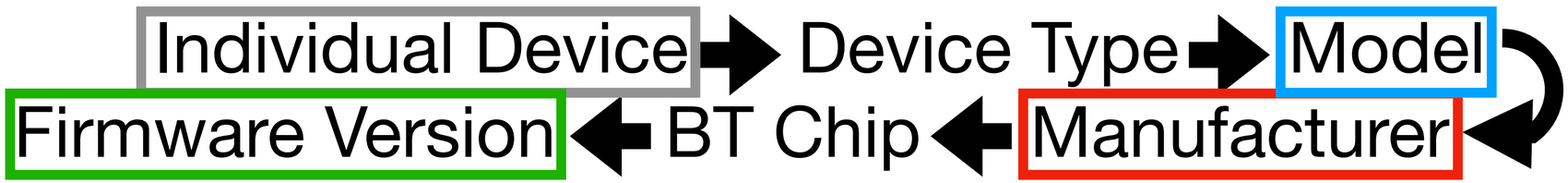


GATT on Phones

- 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

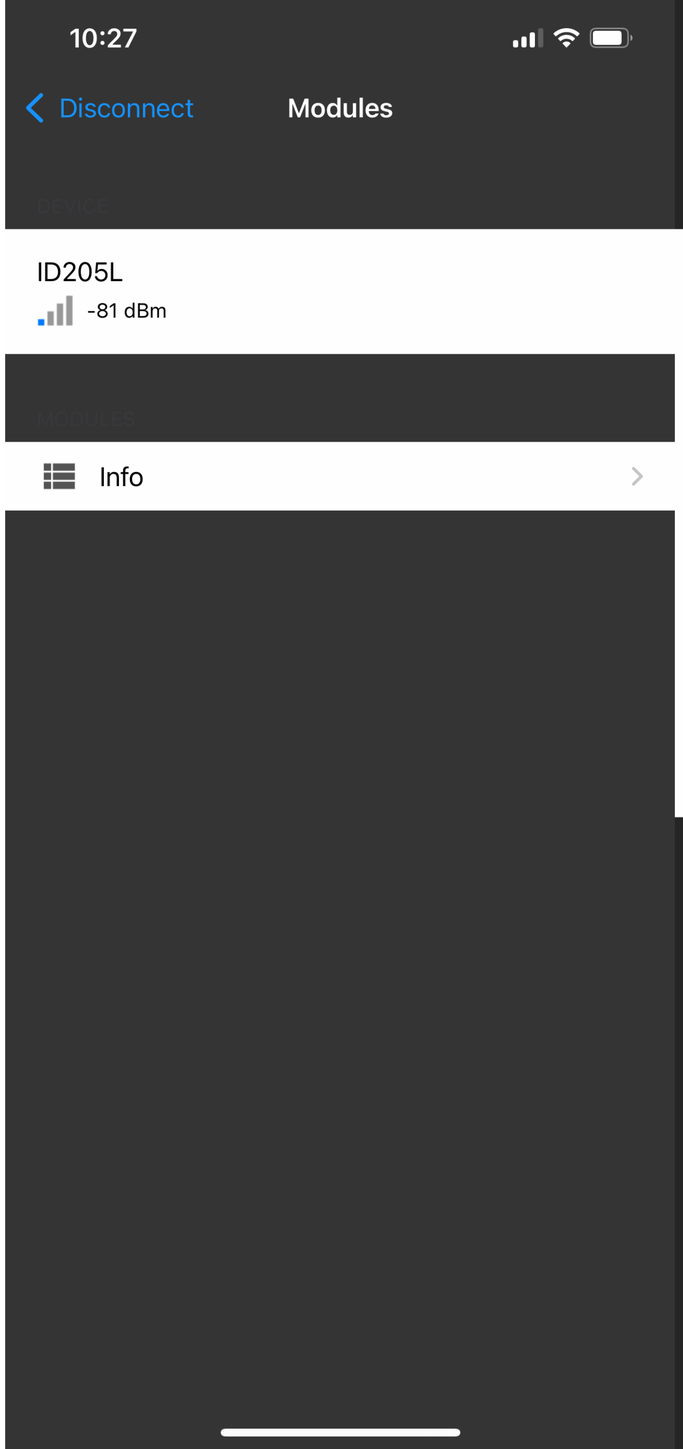


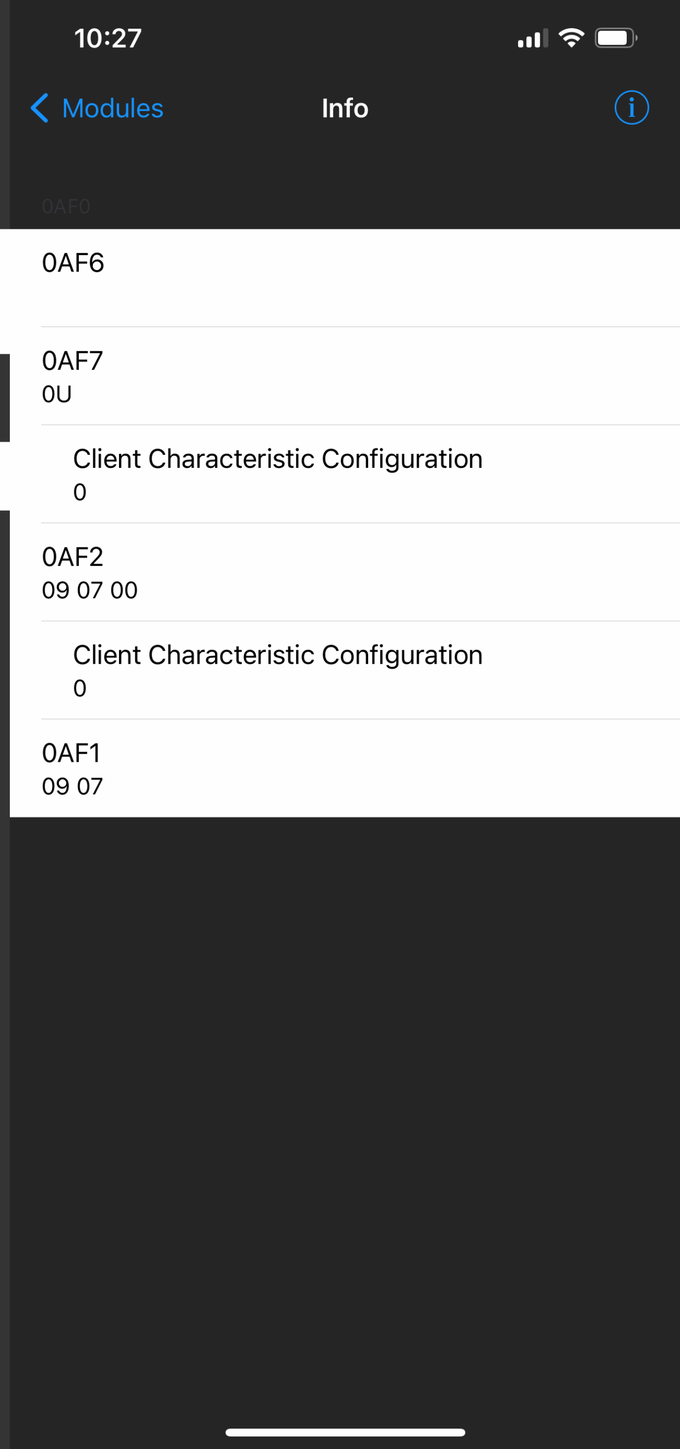




GATT in General

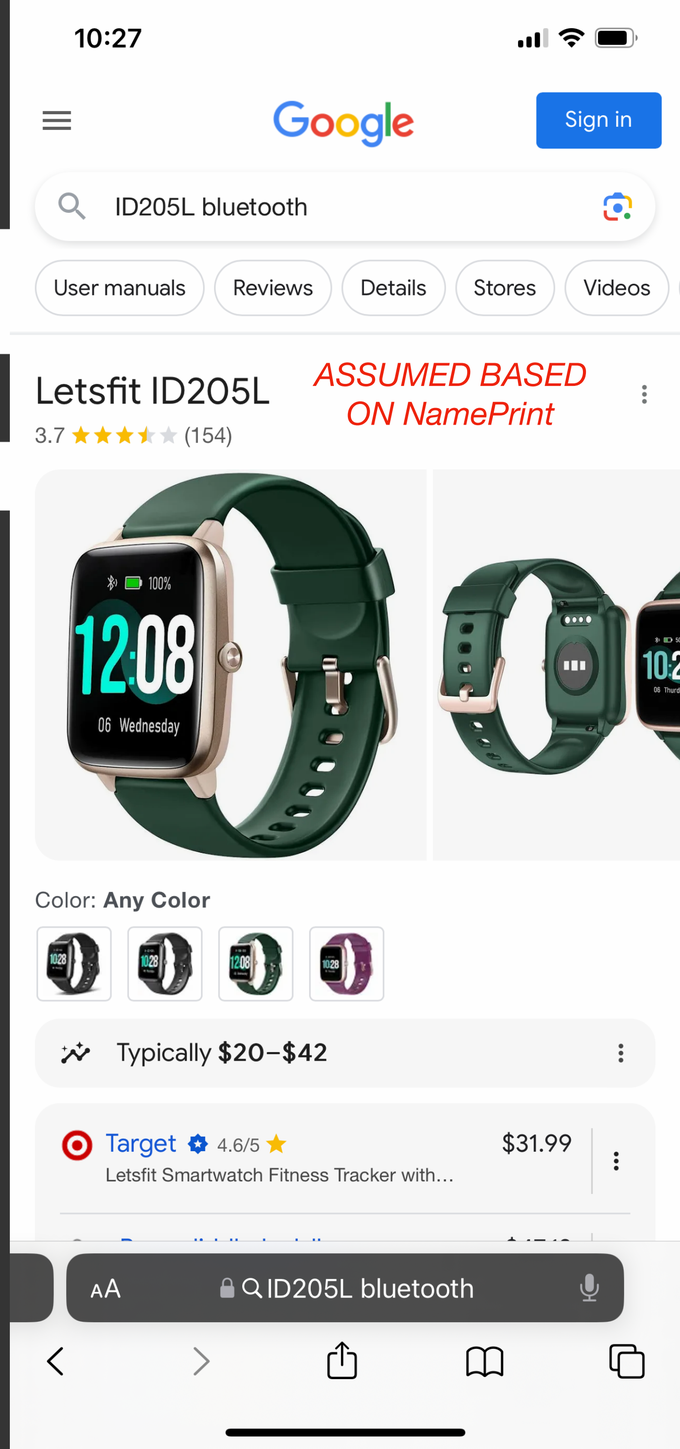
- On the *OTHER* hand...we may instead get a whole bunch of *nothing useful* from GATT
- Devices may not respond to GATT requests, and when they do, we're in no way guaranteed to get "characteristics" which contain the kind of information we want
  - GATT inquiries can take a few seconds. For moving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





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.

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

Prior Work

"Fingerprinting Bluetooth-Low-Energy De **Prof**

- [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.

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

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

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

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
	ound-Ear II	2.181
		1.742
		1.426
		3.662

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

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

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
	ound-Ear II	2.181
		1.742
		1.426
		3.662

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

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

	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

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

	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)

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

	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)

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

	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)

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

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

	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)

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

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

	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)

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

[1] https://inria.hal.science/hal-02359914/file/paper.pdf

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

[1] https://web.cse.ohio-state.edu/~lin.3021/file/CCS19a.pdf

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-obscurity magic unlock code)

https://software-dl.ti.com/lprf/simplelink_cc2640r2_sdk/1.35.00.33/exports/docs/ble5stack/ble_user_guide/html/oad/oad_concepts.html#oad-service-0xffc0

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)**

- 0000110**0**-d102-11e1-9b23-00025b00a5a5 - **GAIA (Over The Air update protocol)** GATT service
  - 0000110**1**-d102-11e1-9b23-00025b00a5a5, 0000110**2**-d102-11e1-9b23-00025b00a5a5, 0000110**3**-d102-11e1-9b23-00025b00a5a5 - Associated GATT characteristics

https://developer.qualcomm.com/qfile/34081/csr102x_otau_overview.pdf

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

https://docs.silabs.com/gecko-os/1/bgx/latest/ble-services

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

https://www.mouser.com/pdfDocs/ApplicationNoteLairdBLESerialPortServicev1_0_0.pdf

What I Want

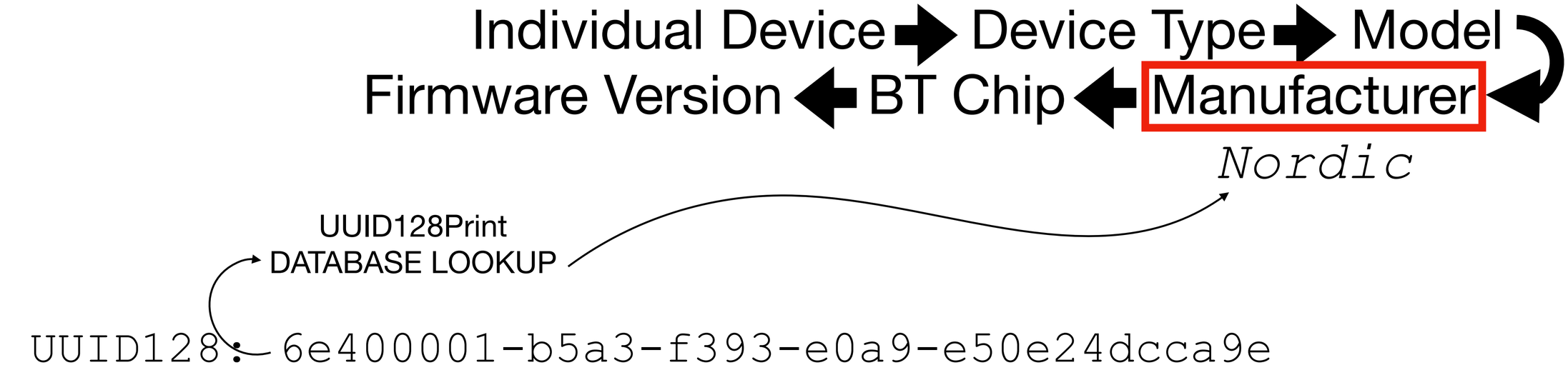
Individual Device → Device Type → Model ↷
Firmware Version ← BT Chip ← Manufacturer

What I Want

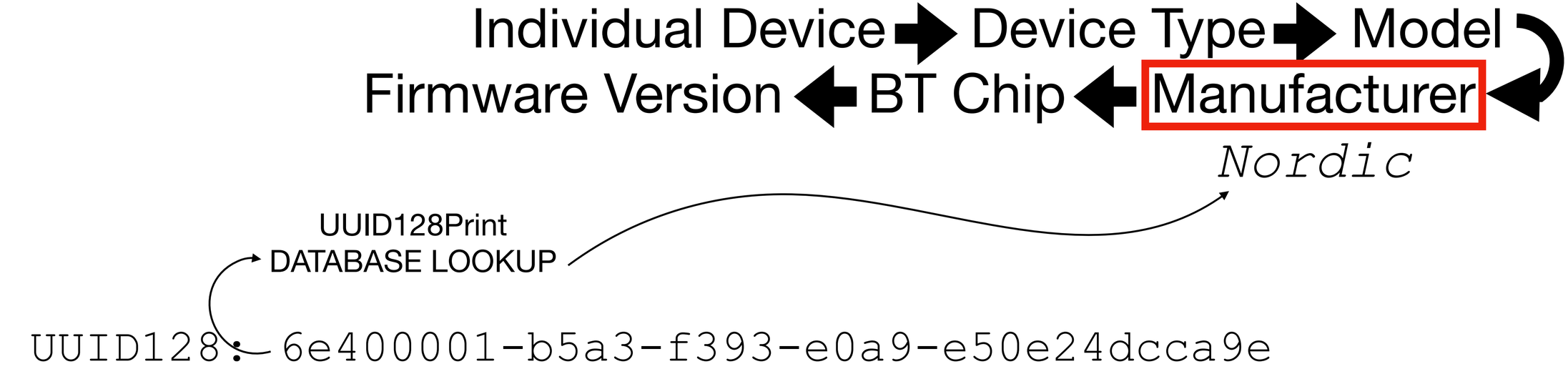

```
UUID128: 6e400001-b5a3-f393-e0a9-e50e24dcca9e
```


DARK MENTOR

What I Want

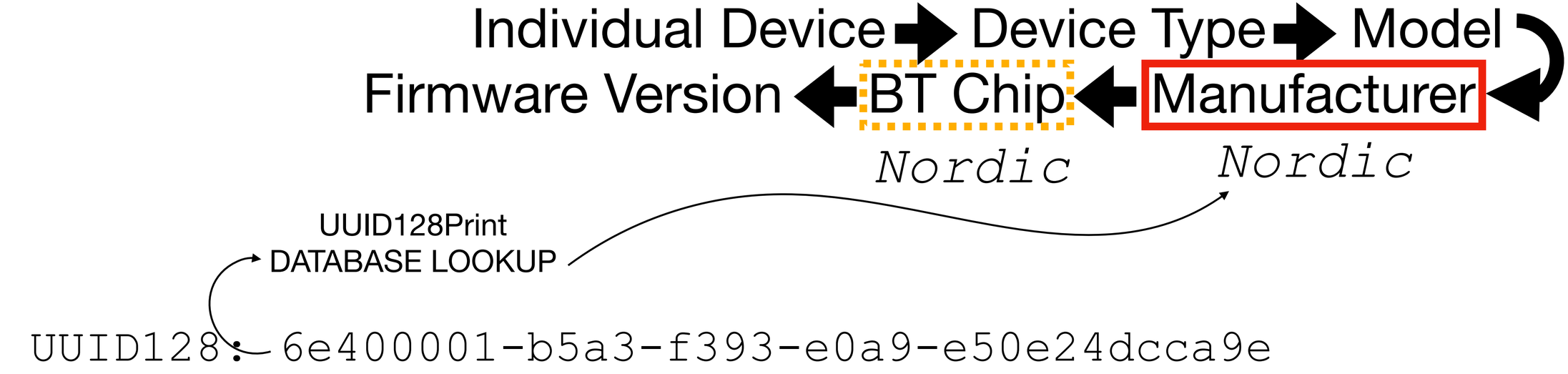

Individual Device
Device Type
Model
Firmware Version
BT Chip
Manufacturer
Nordic
UUID128Print
DATABASE LOOKUP
UUID128: 6e400001-b5a3-f393-e0a9-e50e24dcca9e

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

2024-01-12

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

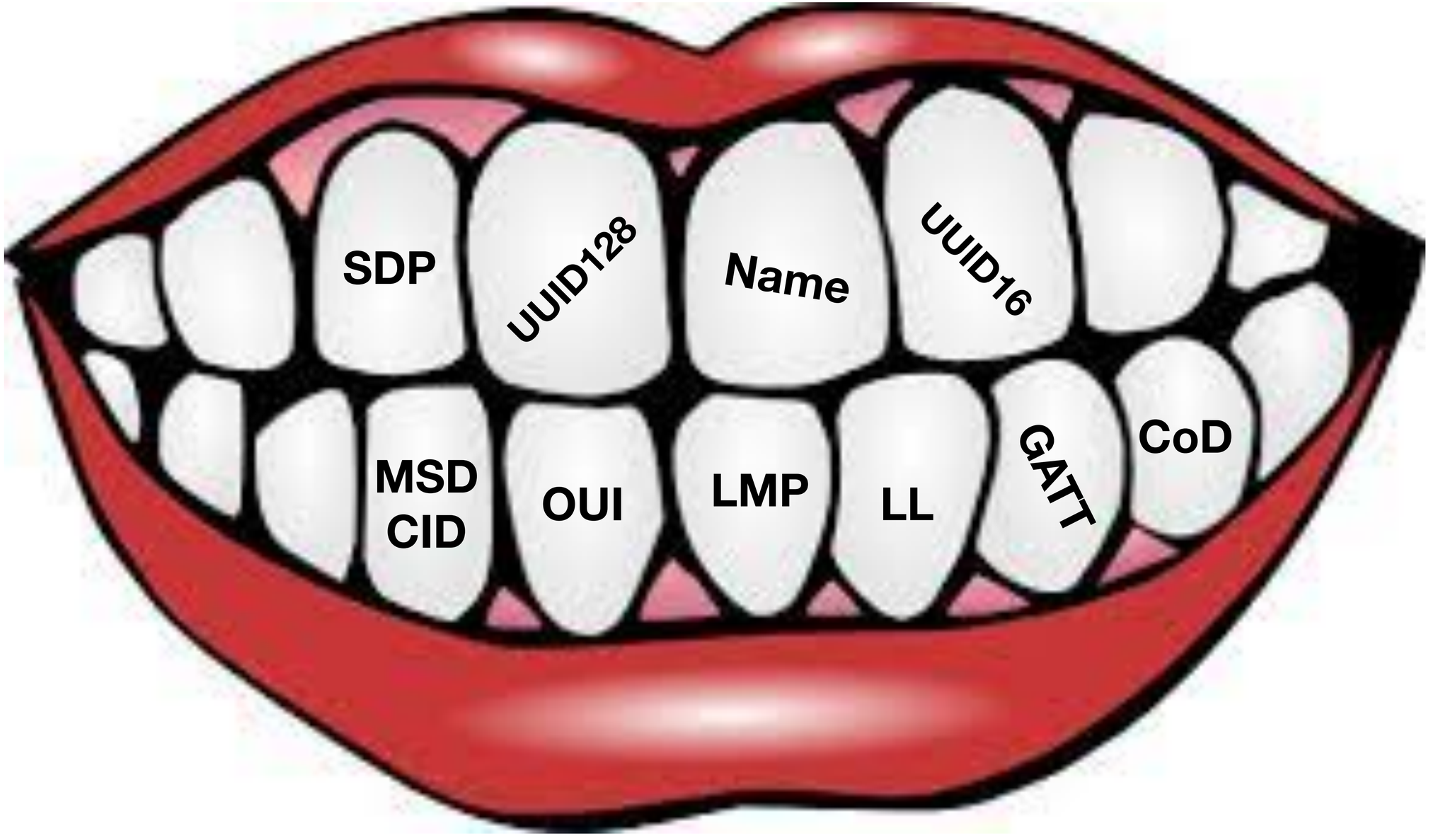
Top 20 GATT Characteristics in my data FWIW

2024-01-12

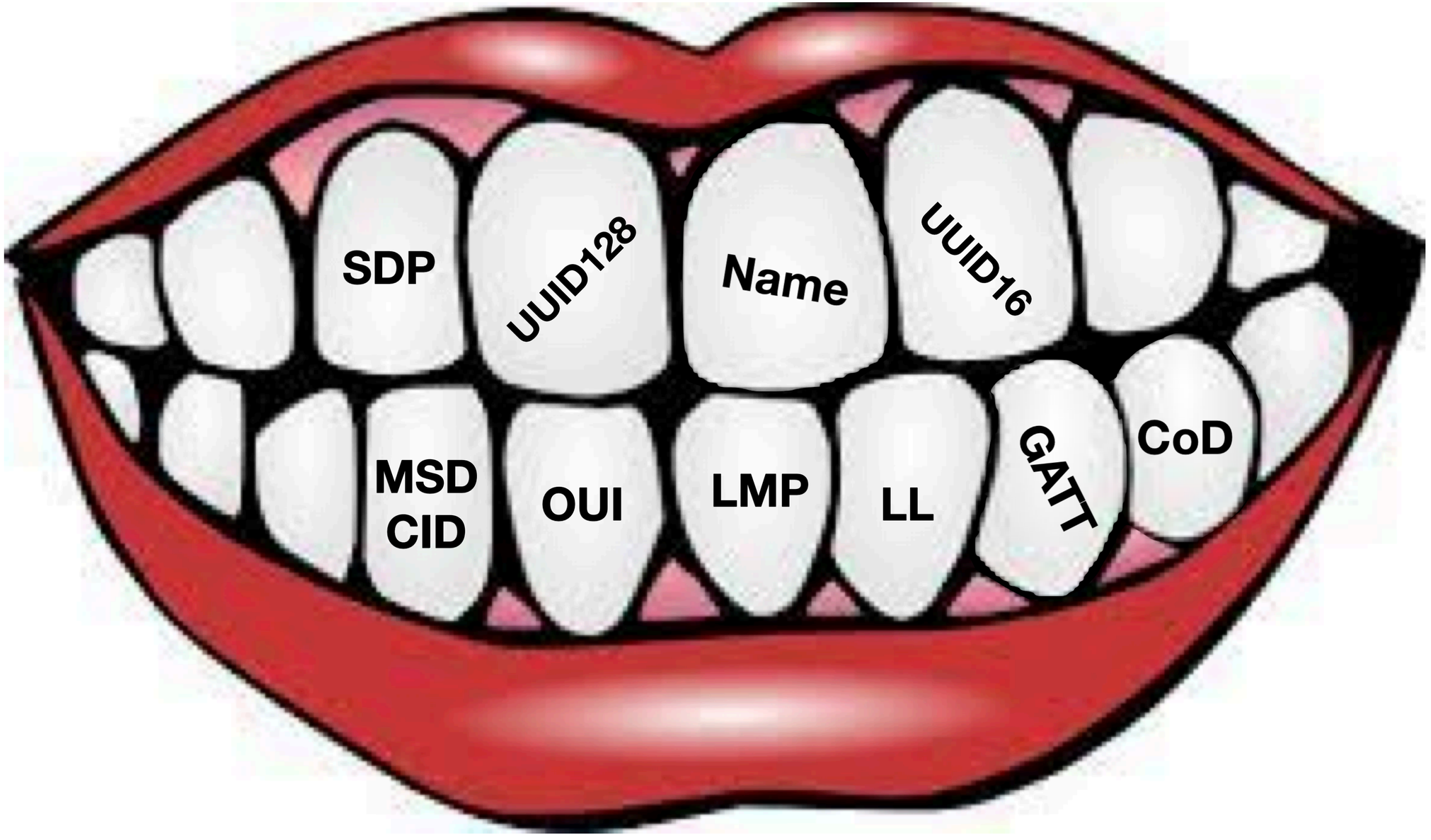
char_UUID128	Count	
00002a00-0000-1000-8000-00805f9b34fb	19592	-> Device Name
00002a01-0000-1000-8000-00805f9b34fb	19205	-> Appearance
00002a05-0000-1000-8000-00805f9b34fb	18443	-> Service Changed
00002a29-0000-1000-8000-00805f9b34fb	16776	-> Manufacturer Name String
00002a24-0000-1000-8000-00805f9b34fb	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-b36ace7c7bfb	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

Putting It All *Toothgether* 😜

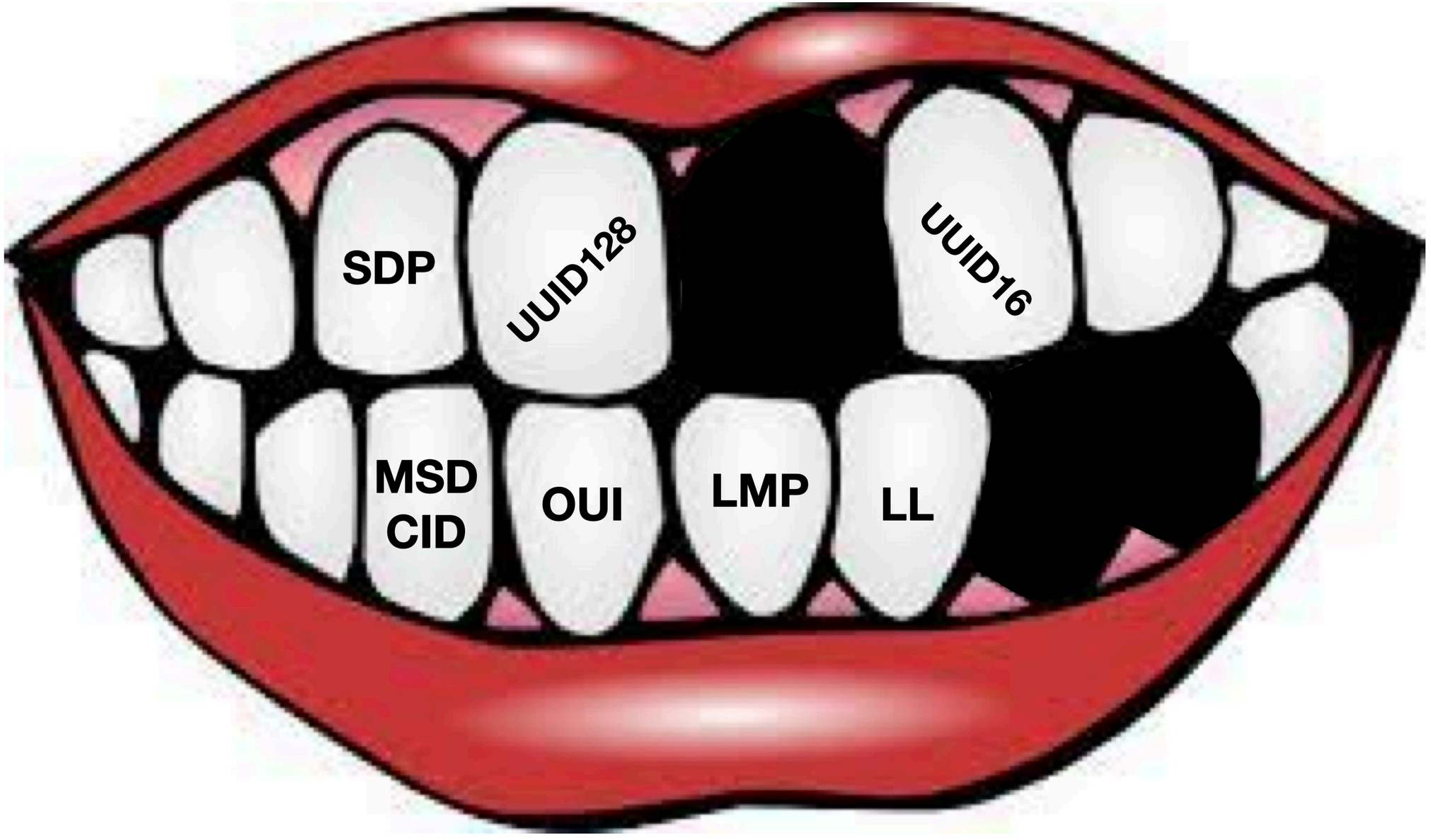
Putting It All *Toothgether* 😜

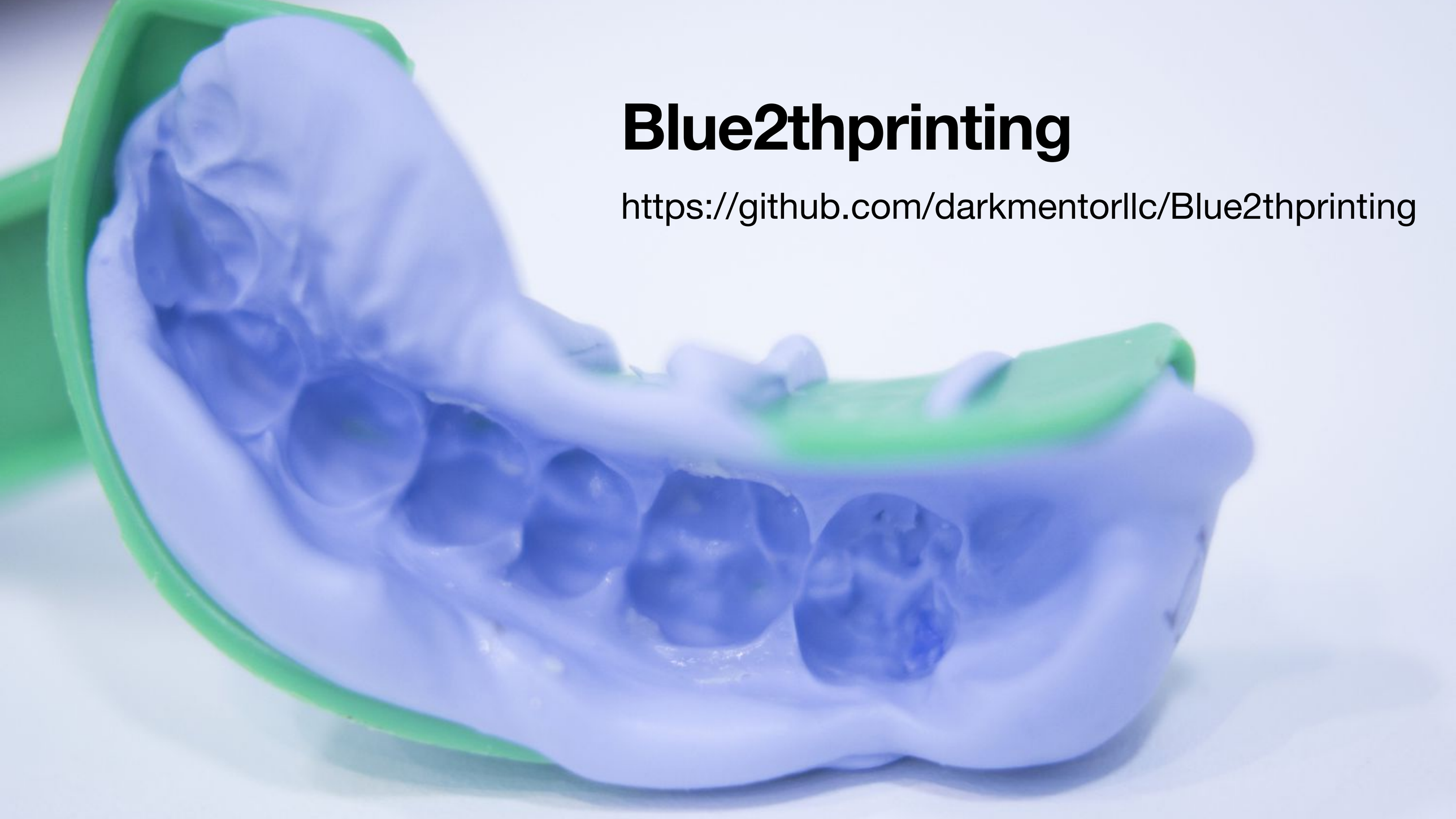


Reminder, even if wireless links weren't *lossy*, you're not generally going to have all the data for every device



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!

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!