Security of connected cars

with Tesla as example

SSTIC
SYMPOSIUM
SUR LA SÉCURITÉ DES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

SYNACKTIV

Who are we

David BERARD

SECURITY EXPERT

@_p0ly_

Vincent DEHORS

SECURITY EXPERT

@vdehors

About this talk

Why car security matters ?

“ Hackers Remotely Kill a Jeep on a Highway ”

Charlie Miller and Chris Valasek
2015

“ Over-the-Air: How we Remotely Compromised the Gateway, BCM, and Autopilot ECUs of Tesla Cars ”

Keenlab, Tencent
2017

- Embed more electronic
- More features
- More connected
- Electronic is used for driving features (emergency break, driving assistance, self driving)

In this talk :

- Security model and architecture
- Hardening
- Focused on the Tesla’s infotainment ECU

Attacker profiles

Multiple threat models

Thief

Wants to steal something valuable inside the car

Wants to steal the car itself

User

Wants to tune his own car, modifying software and hardware

Wants to bypass some paying feature

Targeted attack

Wants to locate the vehicle or record the audio/video inside the vehicle

Wants to cause an accident

Cyber criminals

Wants to steal user data like account credentials

Car ransomware ?

Car components

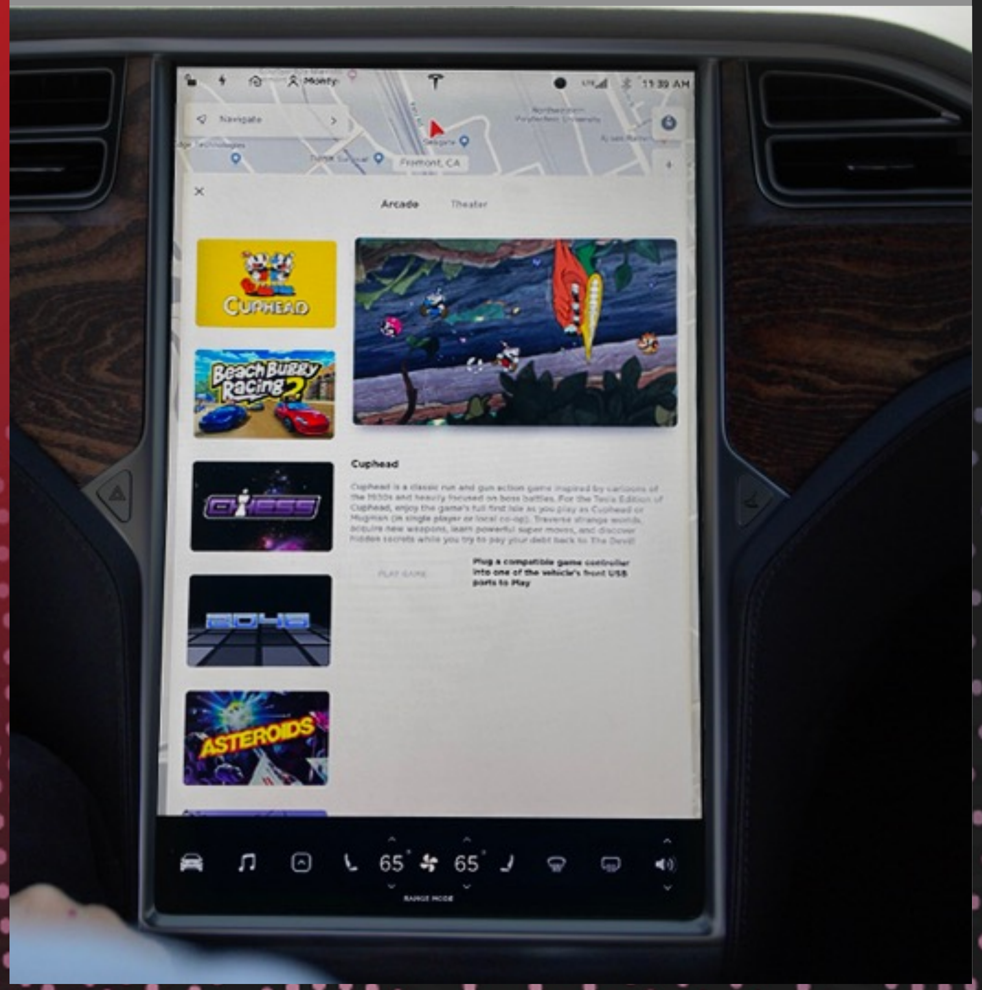
More and more connected car

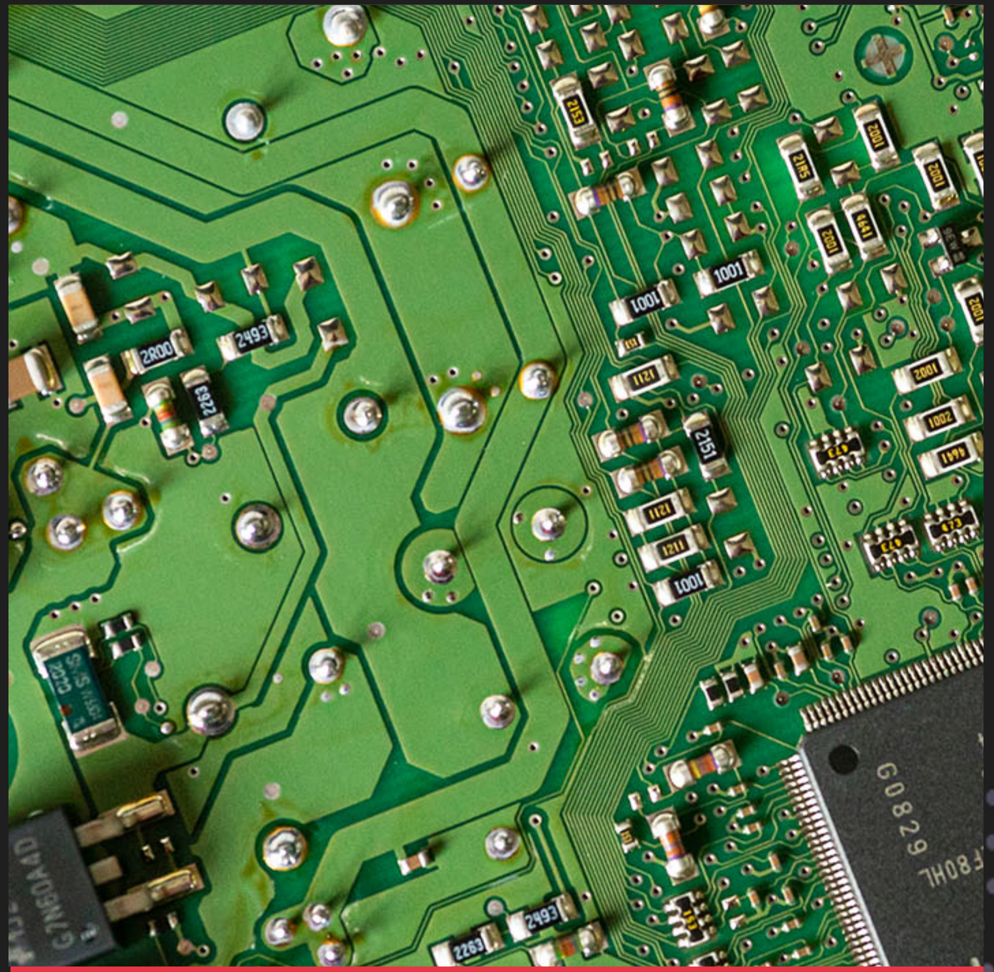
Connectivity

Antennas and connectivity chips for mobile networks, Bluetooth, Wifi, ...

Multimedia

Infotainment is the multimedia computer of the car, has a touch screen and multiple connectivities



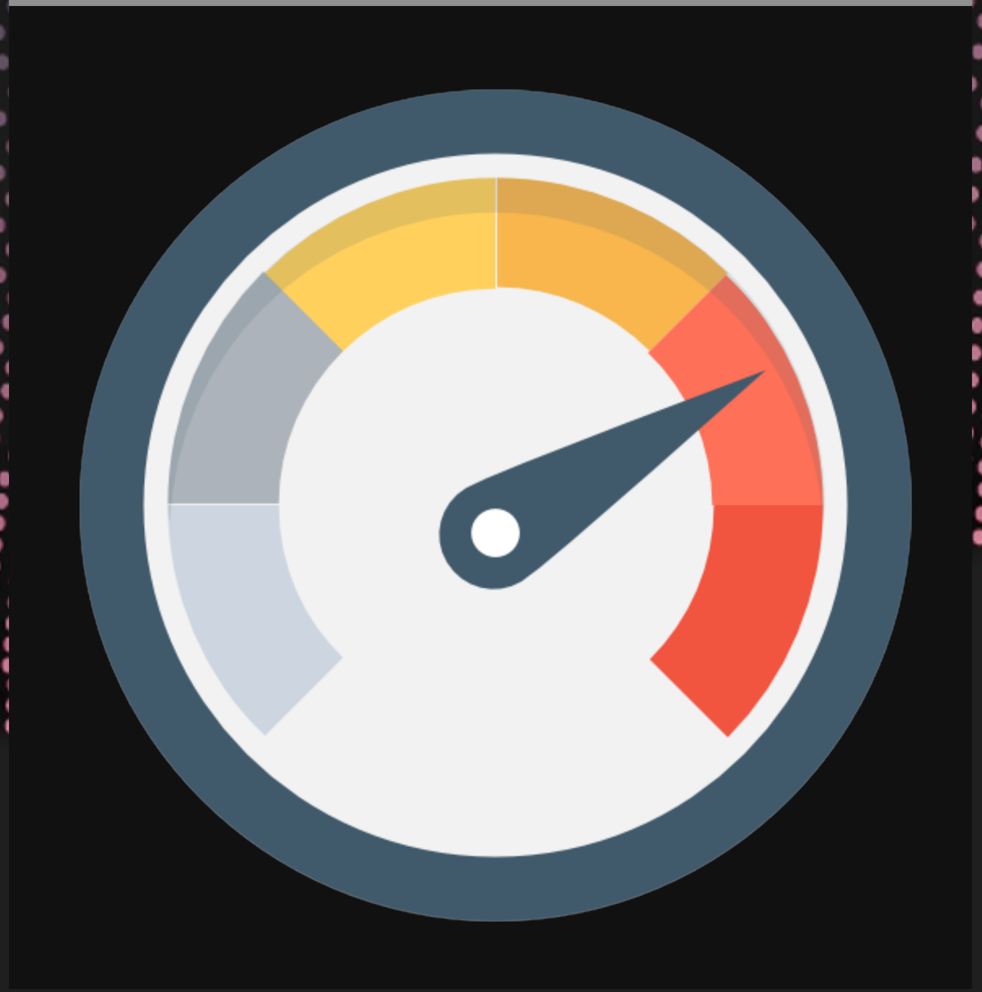


Car control

Electronic control units (ECUs) manage all the mechanic parts of the car

Sensors and Actuators

May be smart or not



Car networks

How these computers are internally connected

Local connection

Component communicating on the same PCB using an embedded protocols like SPI, I2C, UART, PCI

CAN buses

Historic car network technology

There are often multiple separate CAN networks

Ethernet network

Sometimes components are connected using Ethernet buses either on the same PCB or between two boards. The classic TCP/IP stack can be used.

ECUs are interconnected

There are a lot of ECUs in a car

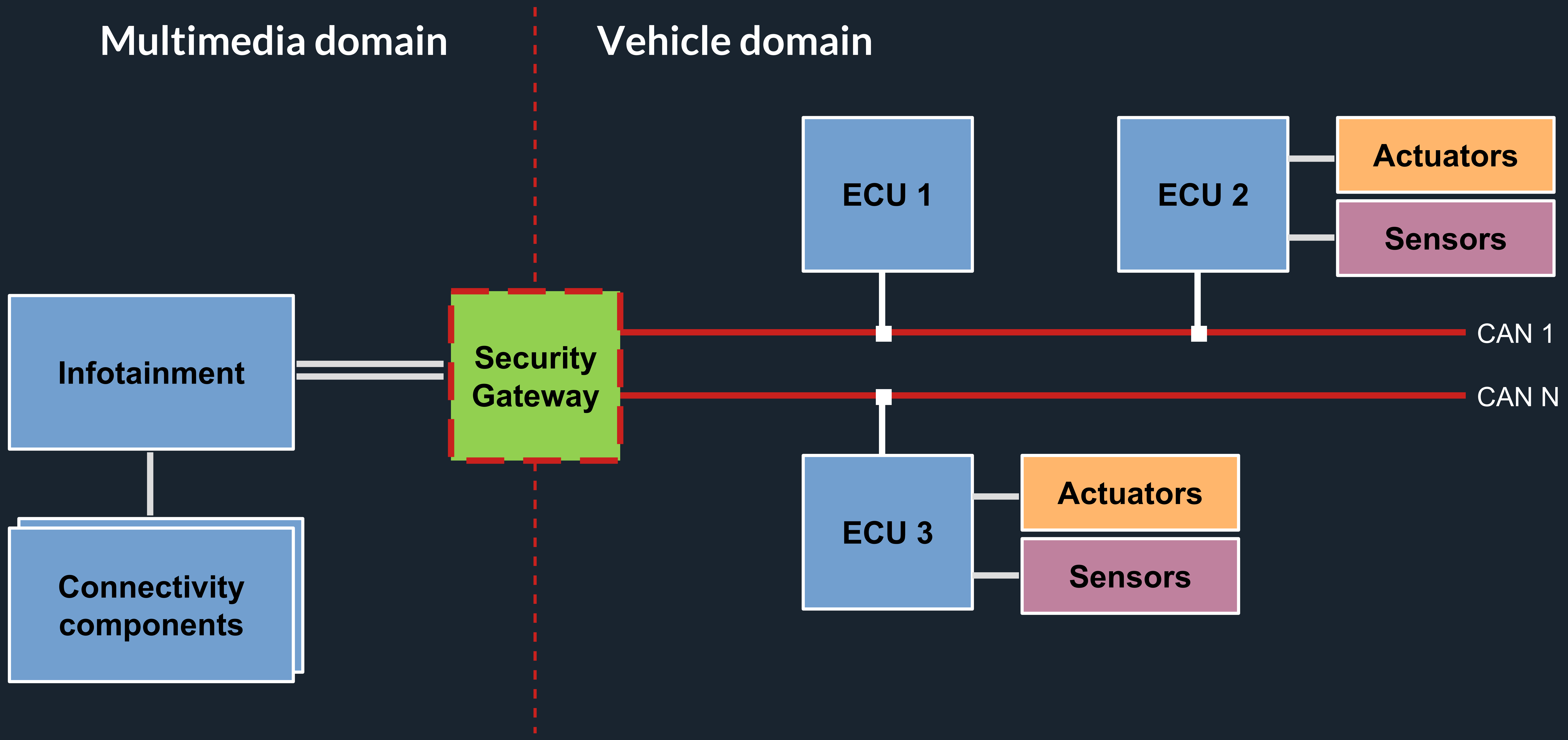
They can send and receive data to each others if they are on the same bus

An attacker can progress in theses networks by compromising multiple ECUs

Common techniques with IT pentesting

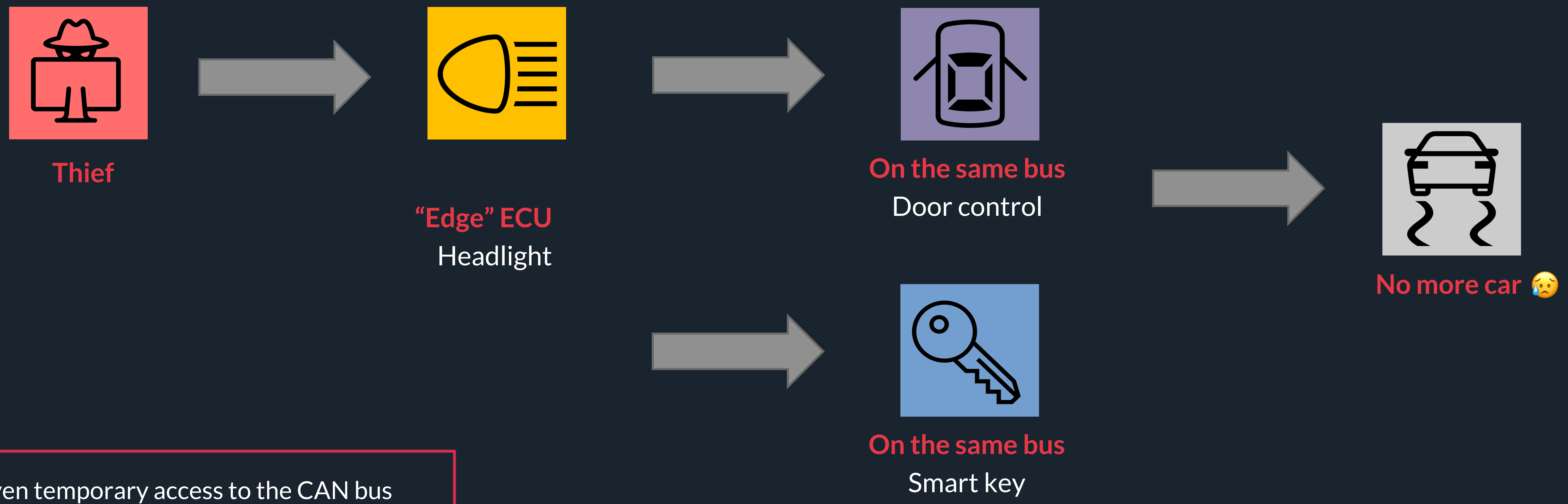
Modern architecture

Multimedia and vehicule domains separared by a gateway



In the wild scenario

Thieves with physical access

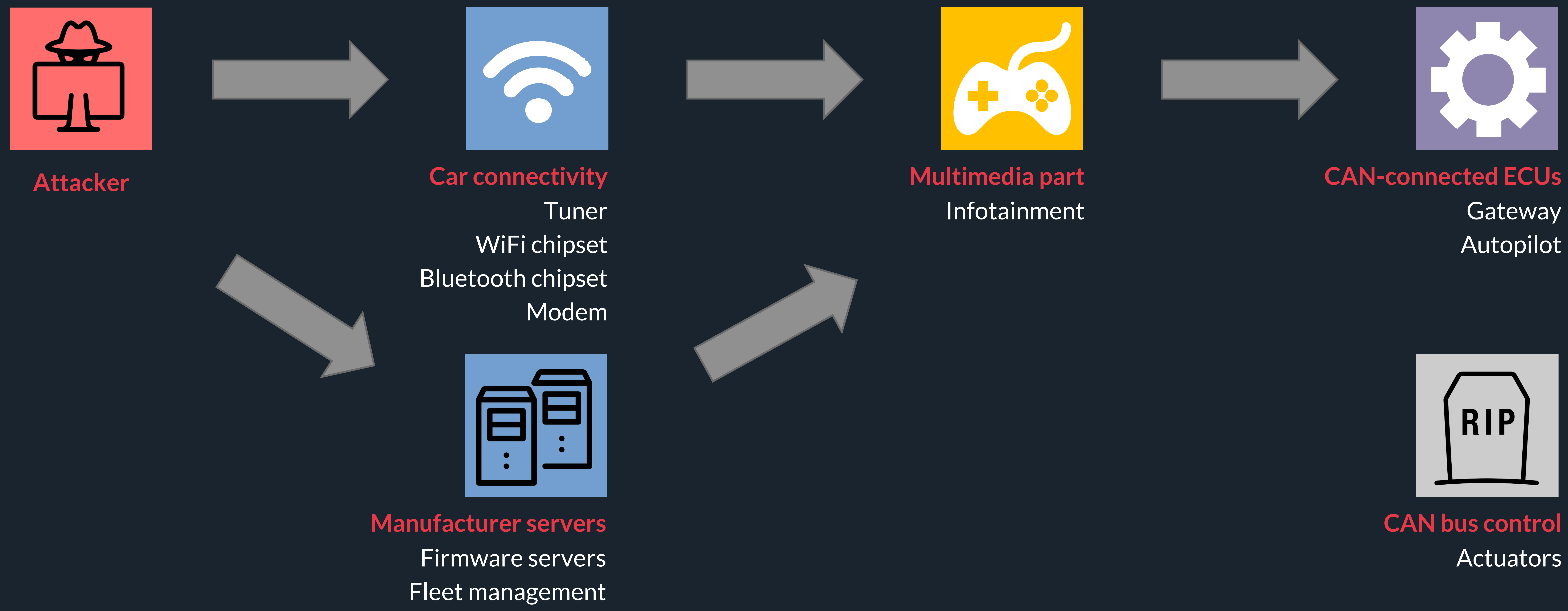


Even temporary access to the CAN bus can allow to an attacker to compromise and **persist** on an ECU

Details on Ken Tidell blogpost : https://kentindell.github.io/2023/04/03/can-injection/

Full chain

Worst case scenario : remote to CAN



Tesla

Infotainment

Genericity as a design strength

Share hardware and software between models

Model 3 / S / Y / X now share the same infotainment hardware (ICE ECU)

Limit major changes between hardware revisions

Intel and **AMD** devices share a very similar software stack and hardware design

Software long term service

The software stack is actively maintained on all hardware revisions

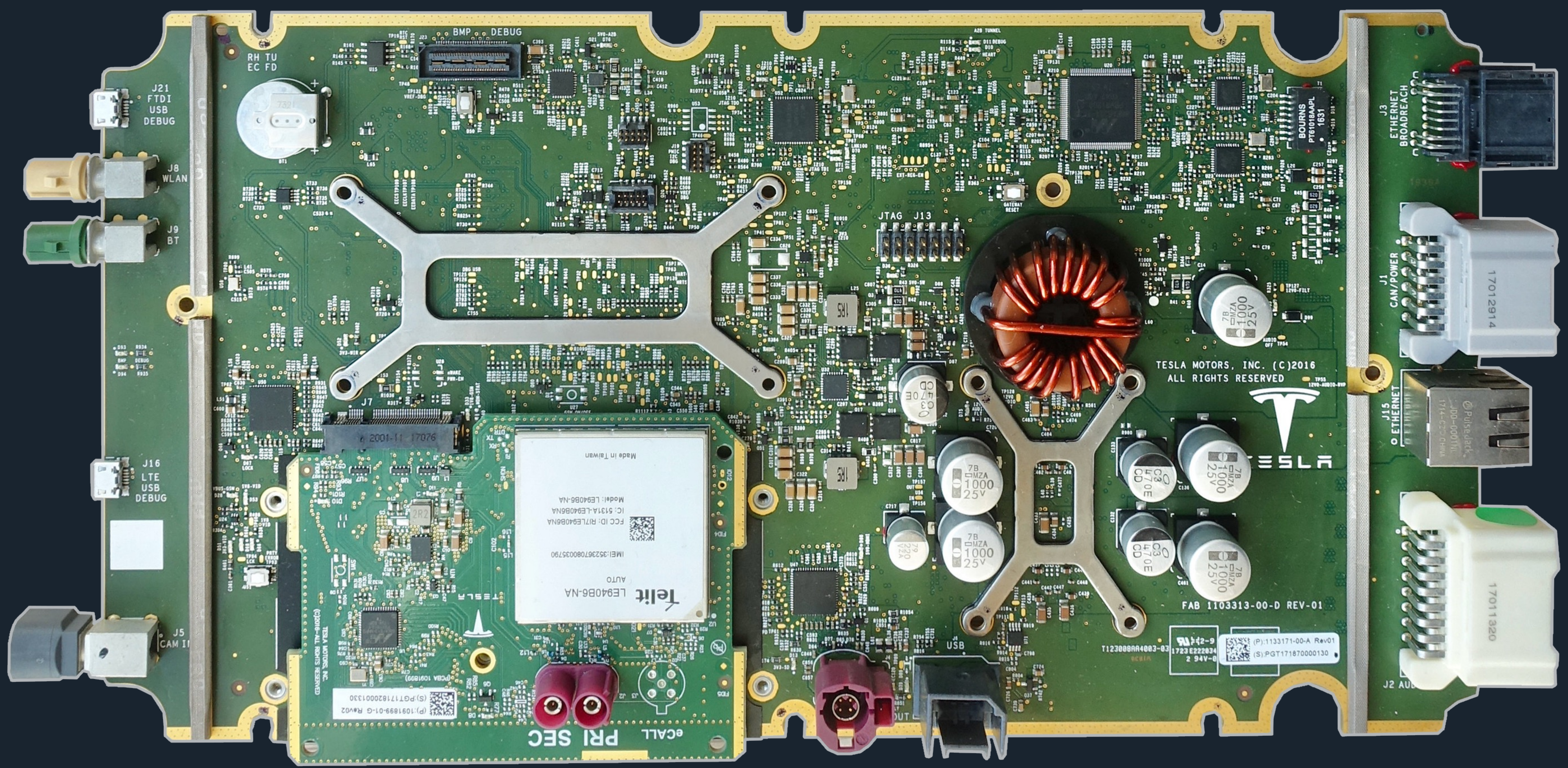
ICE Architecture

Hardware

Same enclosure for infotainment & autopilot

ICE Architecture

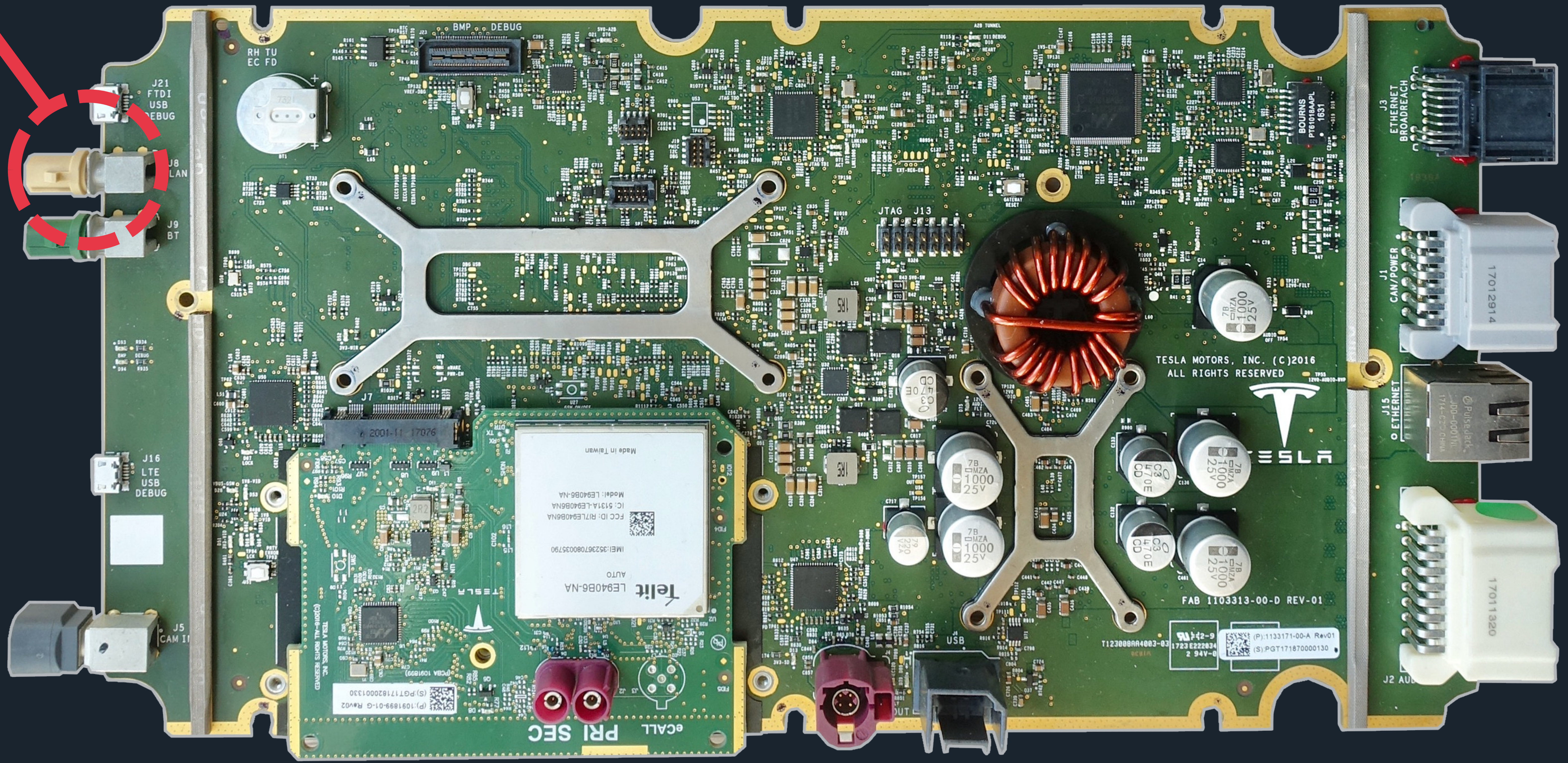
Interfaces



ICE Architecture

Interfaces

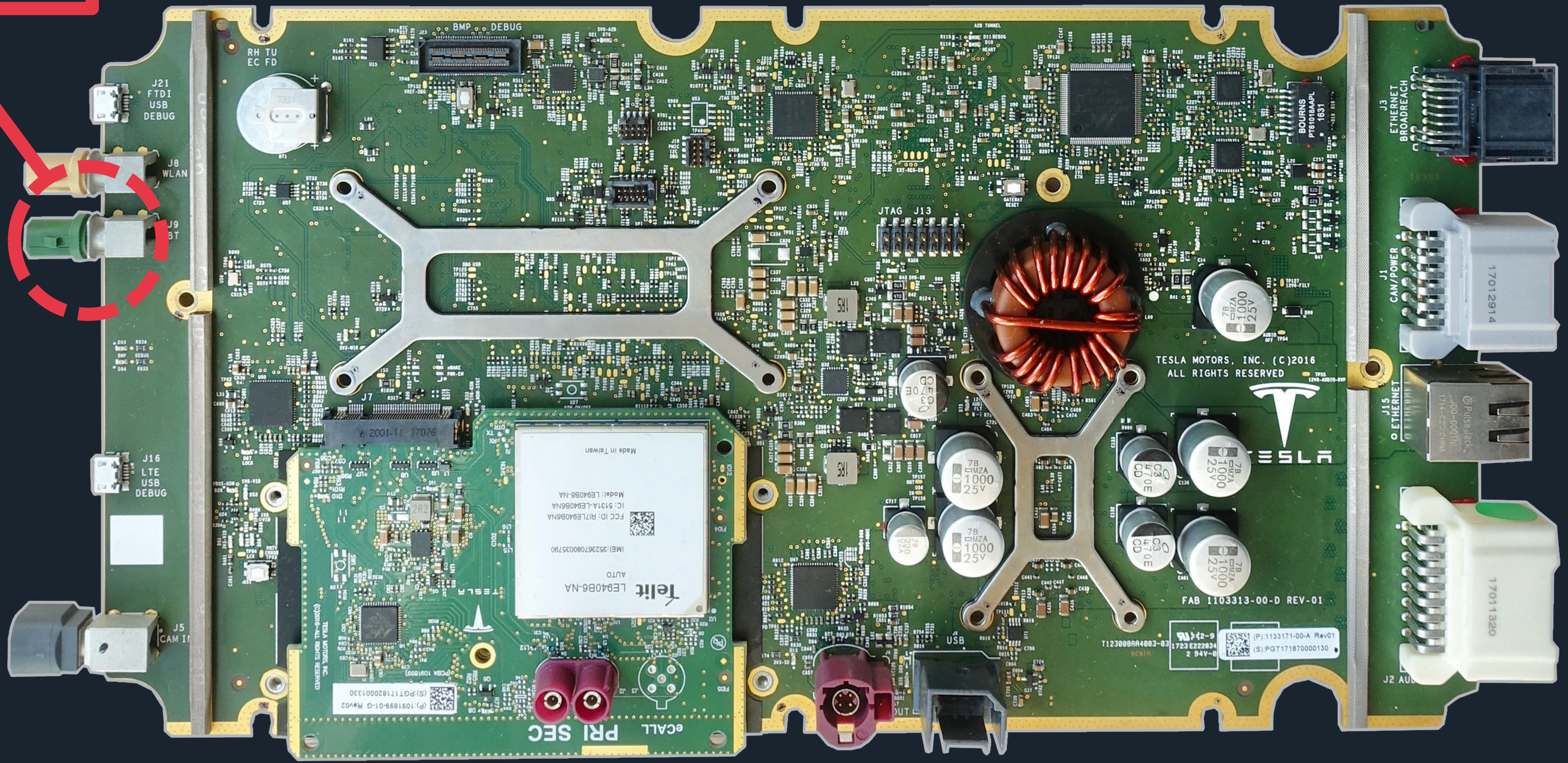
WiFi



ICE Architecture

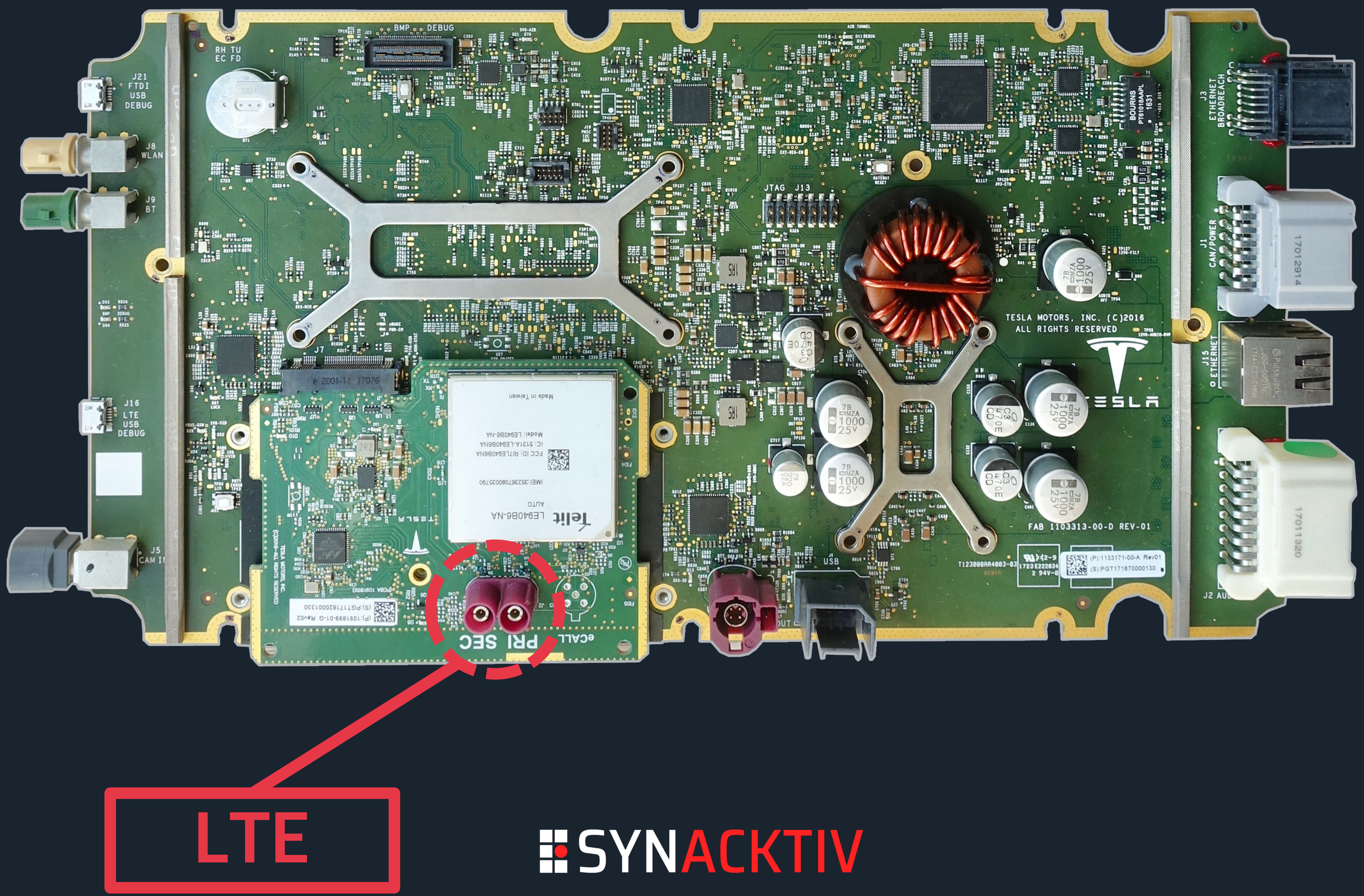
Interfaces

Bluetooth



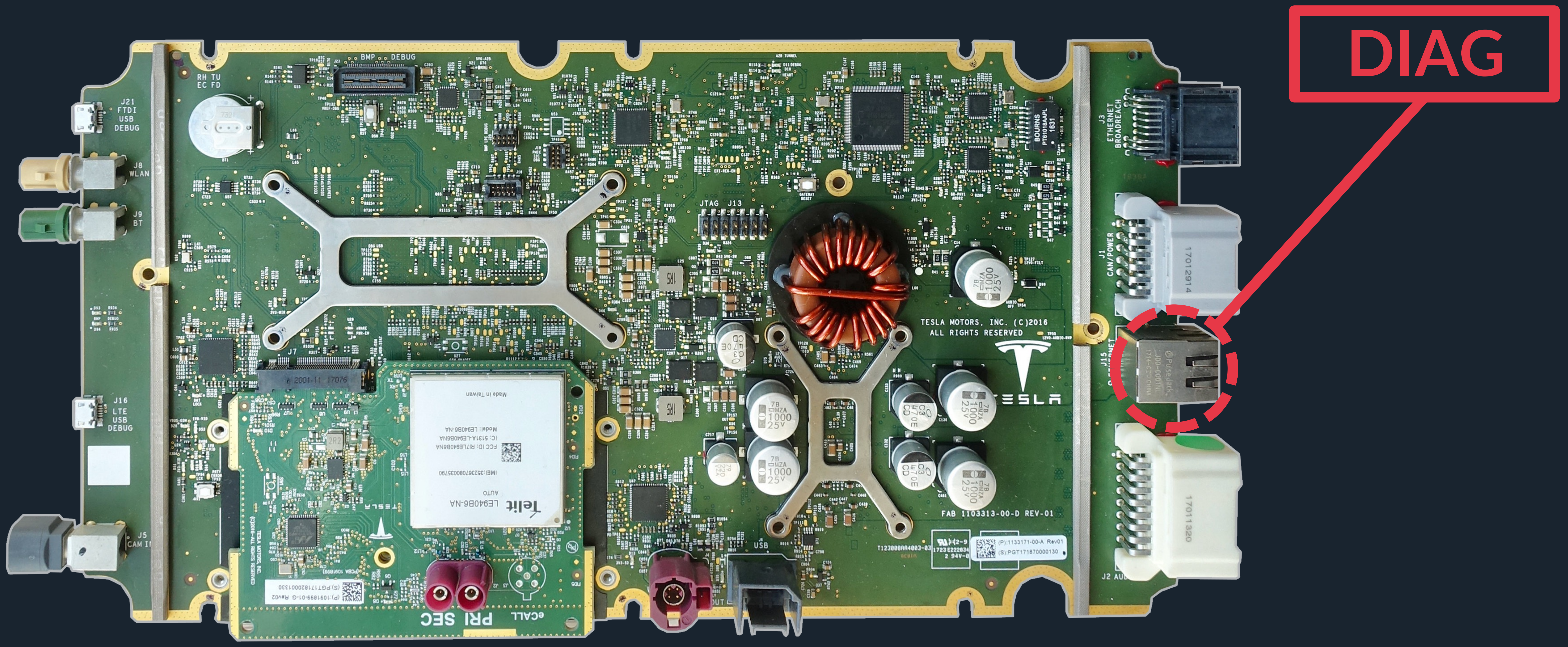
ICE Architecture

Interfaces



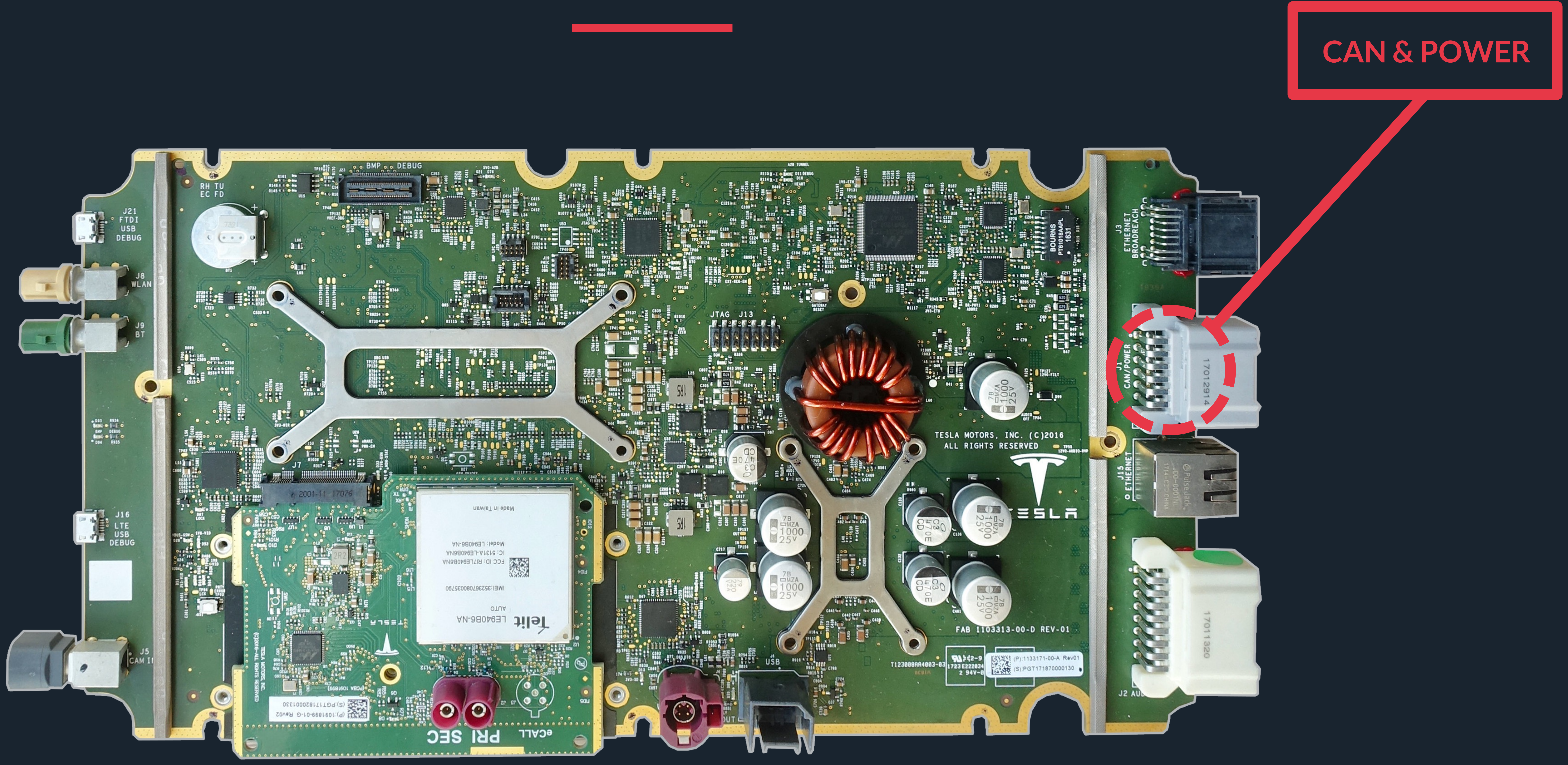
ICE Architecture

Interfaces



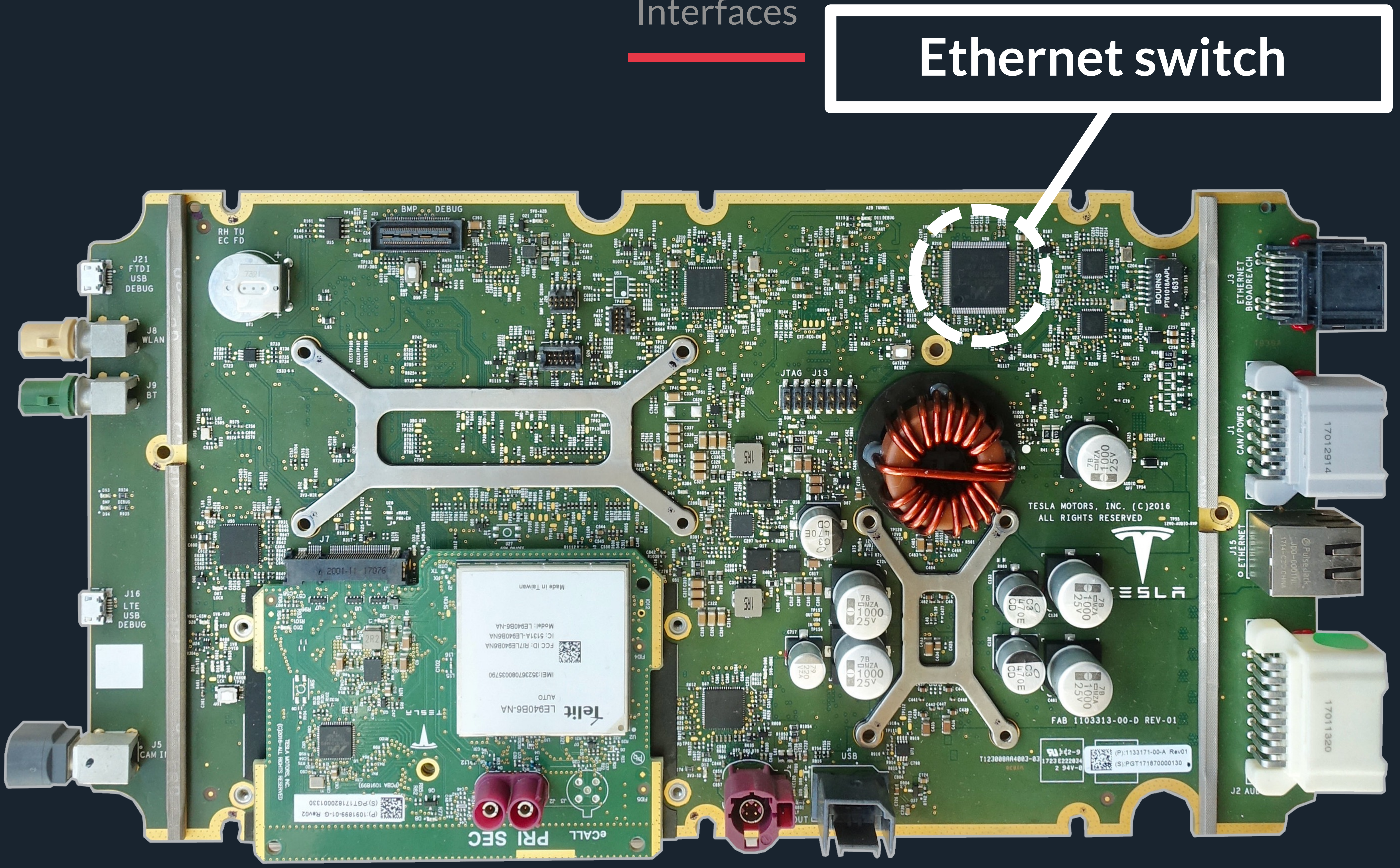
ICE Architecture

Interfaces



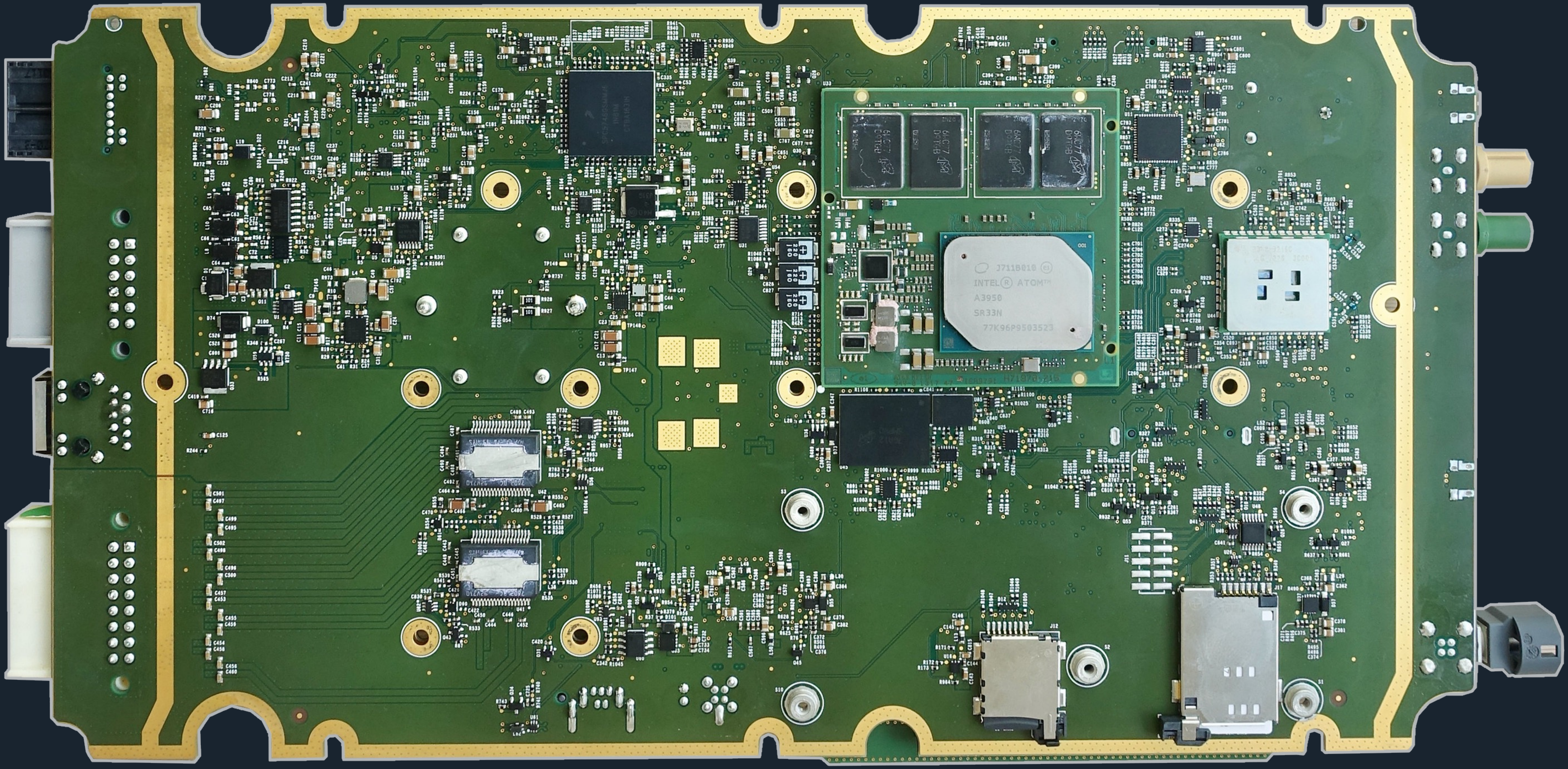
ICE Architecture

Interfaces



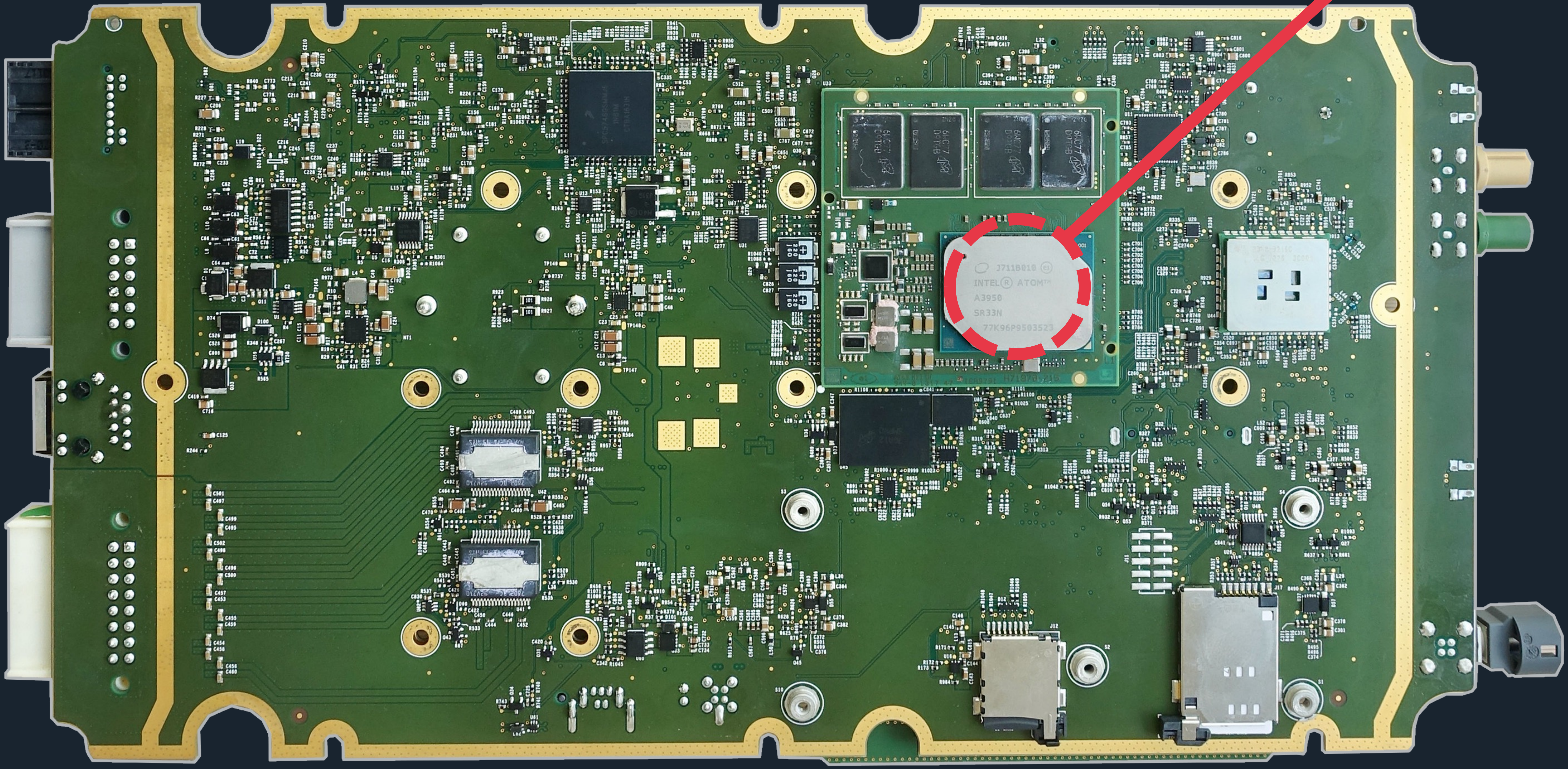
ICE Architecture

Hardware



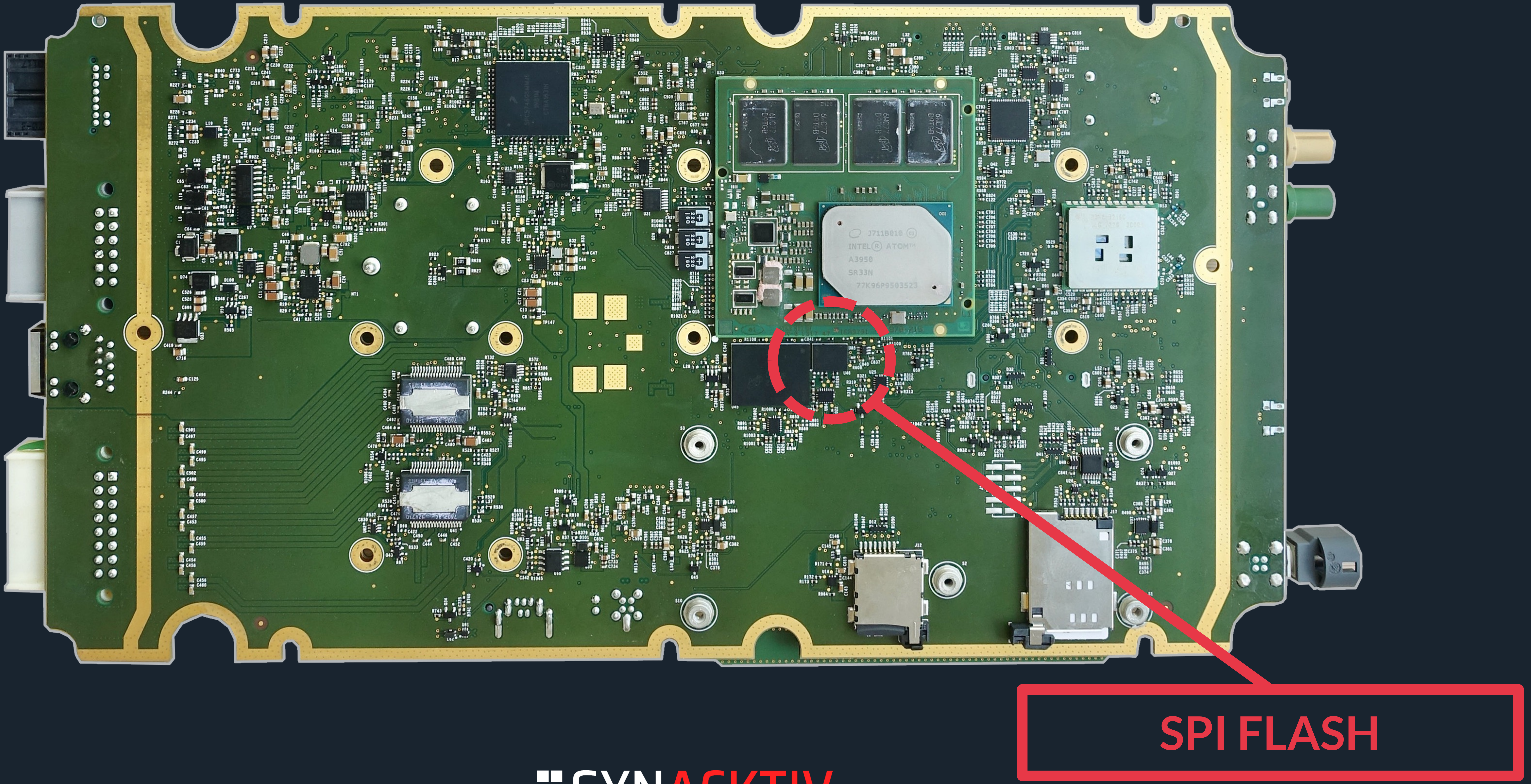
ICE Architecture

Hardware



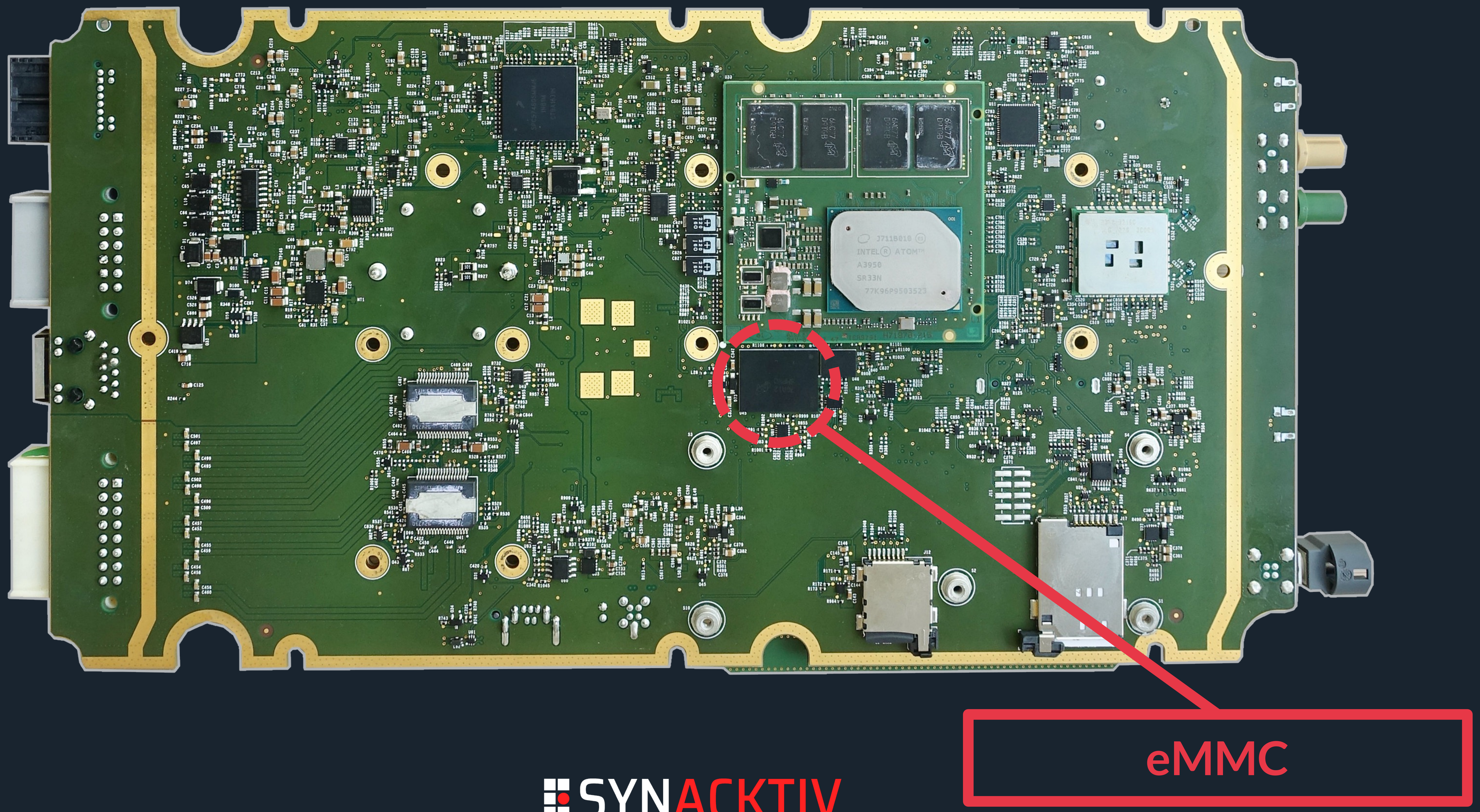
ICE Architecture

Hardware



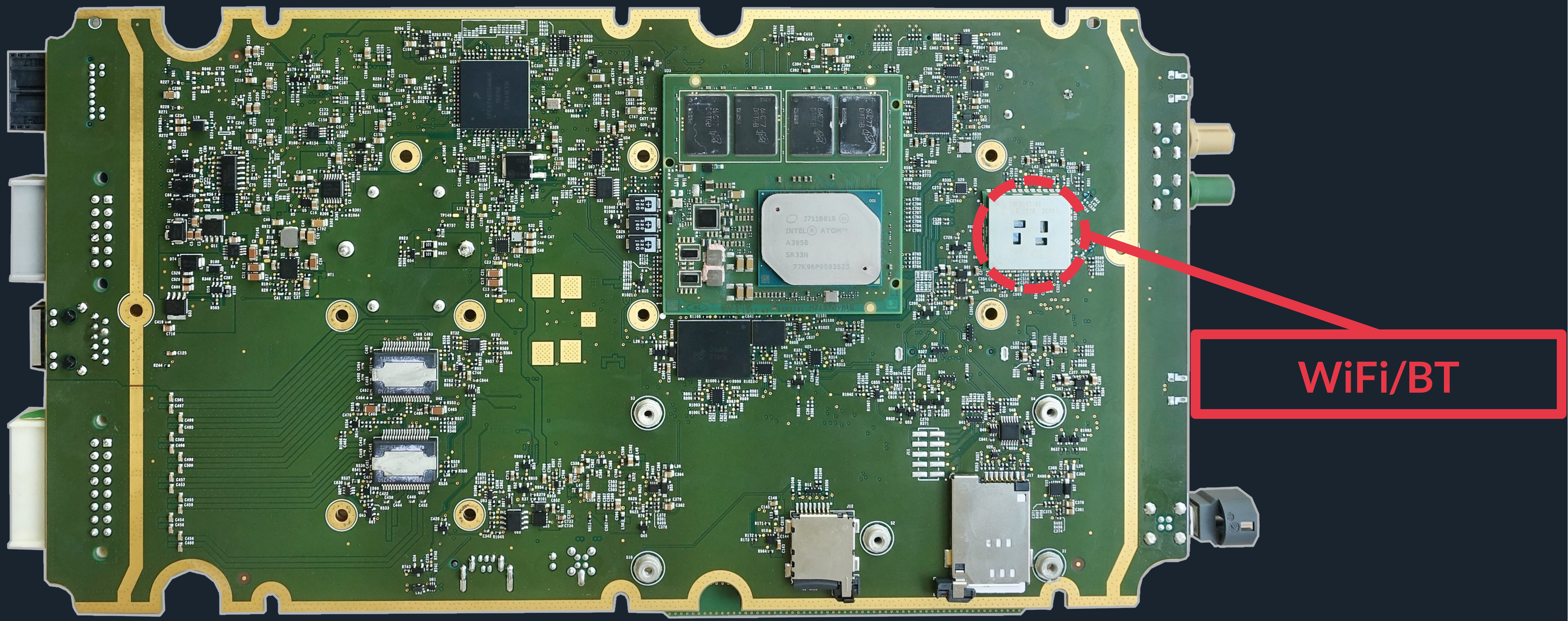
ICE Architecture

Hardware



ICE Architecture

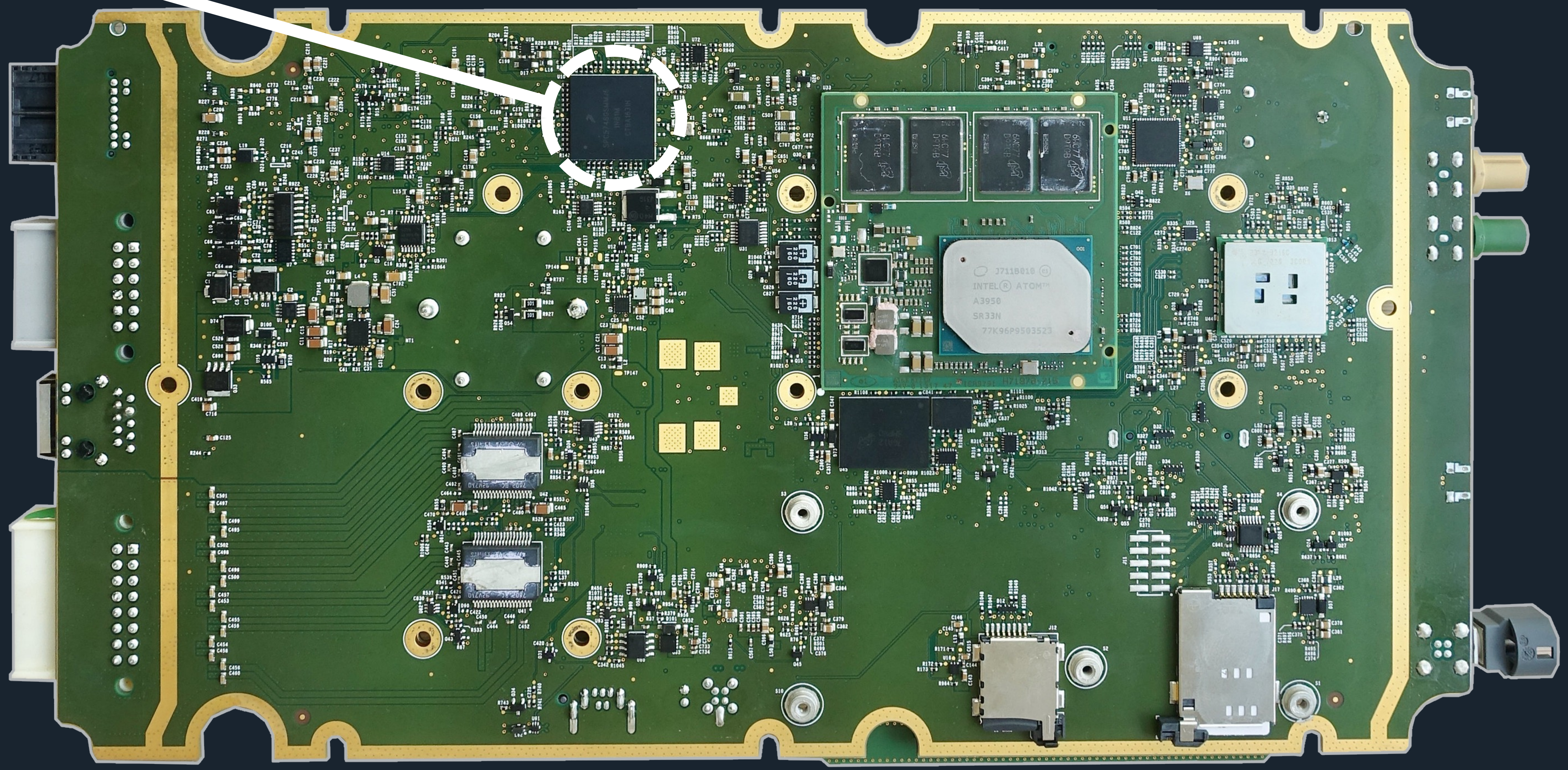
Hardware



ICE Architecture

Hardware

Gateway MCU



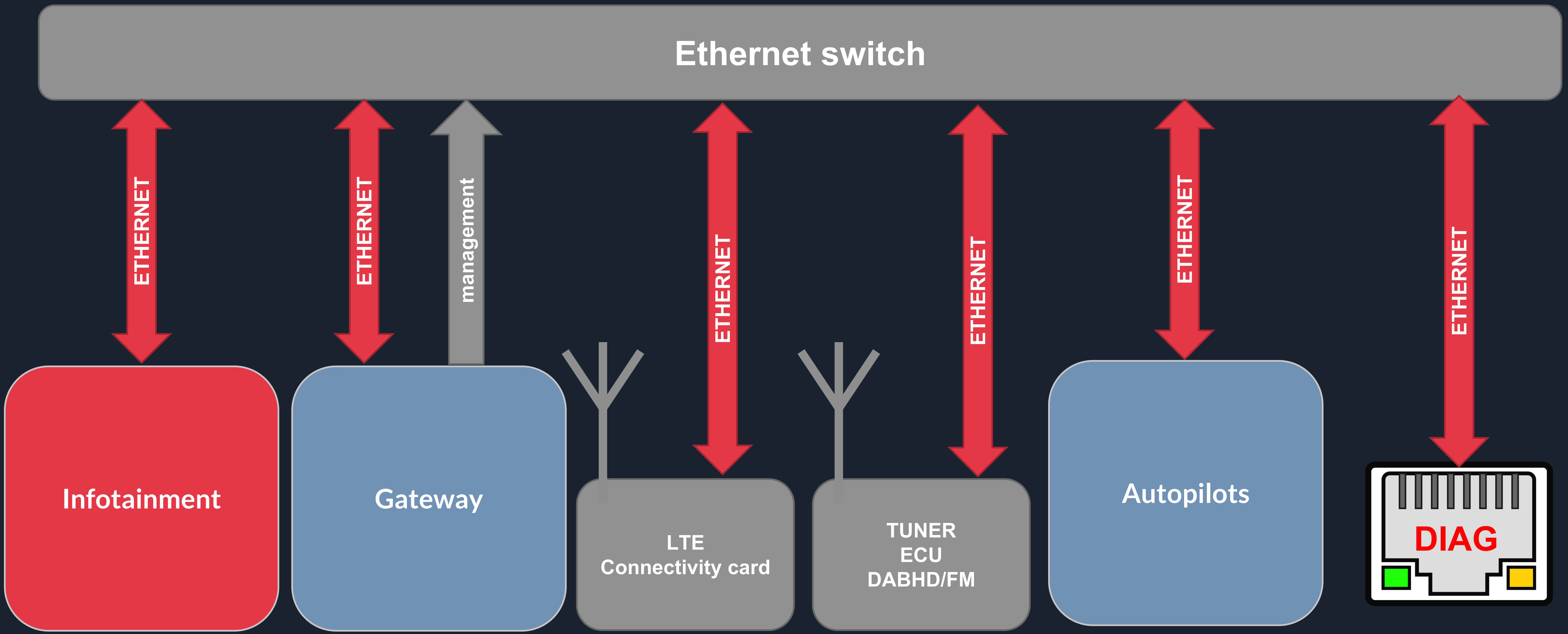
ICE Architecture

Software

Infotainment	Gateway
X64 (Intel Atom / AMD Ryzen)	PowerPC e200 in VLE mode
OS: Linux 4.14 / 5.4	OS: based on FreeRTOS
Highly customized buildroot system	
Boots on eMMC / SPI	Boots on internal flash / SDcard / tftp

ICE Architecture

Ethernet Network



Infotainment

Software System & Hardening

Principles

1. Limit the attack surface
2. Isolate and limit application rights
3. Make vulnerabilities harder to exploit
4. Patch vehicles OTA
5. Isolate and filter Ethernet and CAN networks
6. Protect user data and system integrity

Limit the attack surface

Linux

Well configured

Only required features & drivers

Software stack

Use state of the art opensource softwares

Activate only required features

Modify software

Modify source code to disable unnecessary features

Infotainment

Software System & Hardening

Principles

1. Limit the attack surface
2. Isolate and limit application rights
3. Make vulnerabilities harder to exploit
4. Patch vehicles OTA
5. Isolate and filter Ethernet and CAN networks
6. Protect user data and system integrity

Isolate and limit application rights

Process isolation

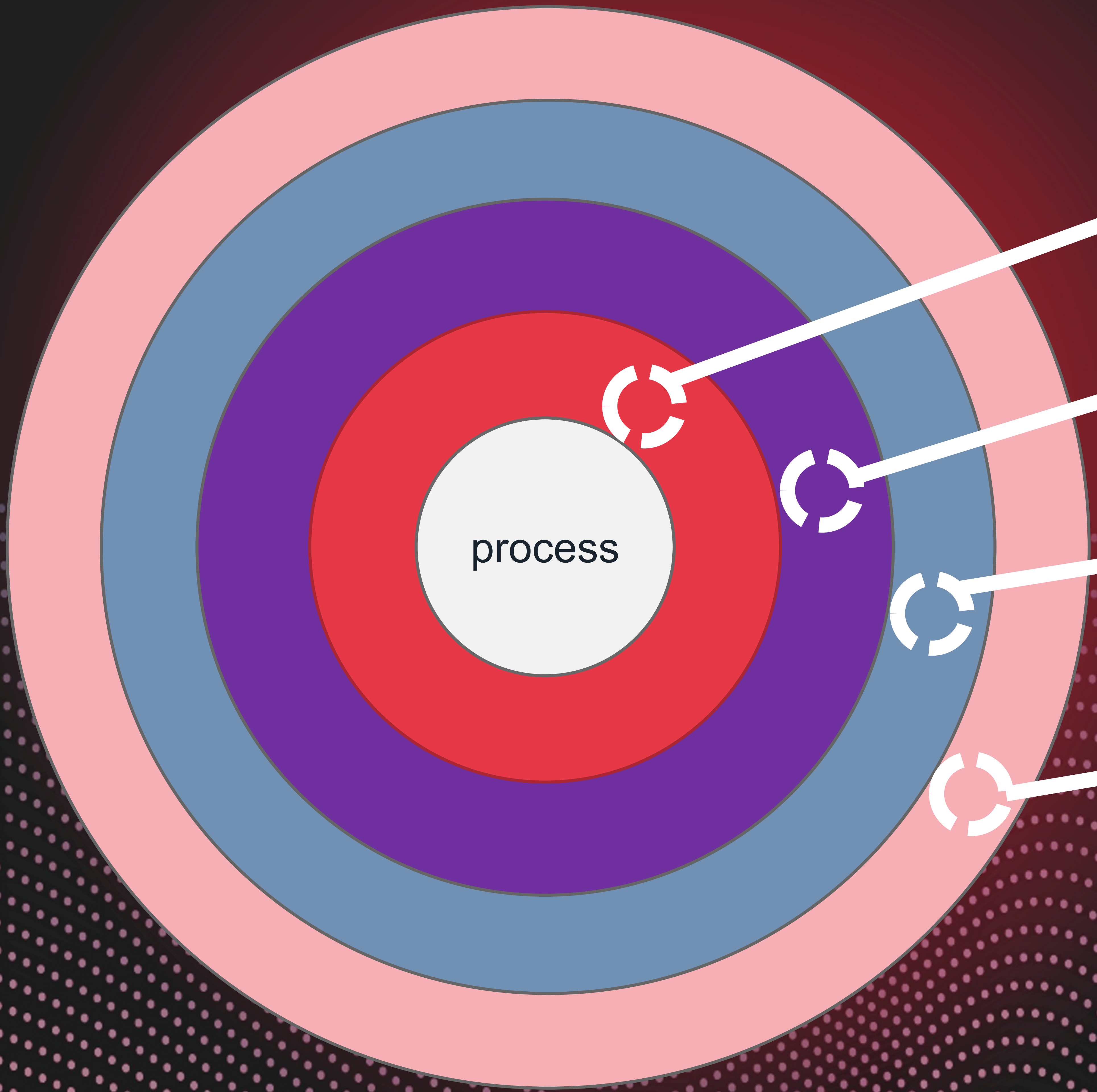
```
ssh
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
dbus        2660  0.6  0.0   3896  3008 ?        S    01:43   0:01 /usr/bin/dbus-daemon --nofork --sys
mediase+    2696  0.0  0.0   3260  2440 ?        S    01:43   0:00 /usr/bin/dbus-daemon --nofork --pri
hermes-+    2763  0.0  0.1 1304752 7876 ?        Sl   01:43   0:00 /opt/hermes/hermes_grablogs --log-l
hermes-+    2778  0.0  0.5 1312476 20308 ?       Sl   01:43   0:00 /opt/hermes/hermes_proxy --log-leve
updater     2780  0.1  0.5  36396 23080 ?        S    01:43   0:00 /bin/ice-updater
nobody      2796  0.0  0.0   2452   236 ?        S    01:43   0:00 socklog ucspi
log         2868  0.0  0.0   2932  1892 ?        S<   01:43   0:00 ulogd --uid log
chromium    3325  0.0  0.0   3260  2424 ?        S    01:43   0:00 /usr/bin/dbus-daemon --nofork --pri
infohea+    3918  0.0  0.1 1077980 5808 ?        Sl   01:44   0:00 /usr/bin/infohealthd
nobody      4449  0.2  0.0   2452   244 ?        S    01:43   0:00 socklog unix /dev/log
gpsmana+    4512  0.0  0.0   3624  1956 ?        S    01:43   0:00 /bin/bash /usr/bin/custom-cron.sh -
helios      4529  0.0  0.1  14104  7636 ?        Ss   01:43   0:00 /usr/bin/helios --daemon
gpsmana+    4567  0.0  0.0   4140  1600 ?        S    01:43   0:00 /bin/bash /usr/bin/custom-cron.sh -
autopil+    4603  0.0  0.1 1078024 5356 ?        Sl   01:43   0:00 /usr/bin/autopilot-api -country=US`
gpsmana+    5004  0.0  0.0   3312   508 ?        S    01:43   0:00 sleep 2h
webrtc      5274  0.0  0.2  27764  9116 ?        S    01:43   0:00 /usr/bin/webrtc-comms
command+    5280  0.0  0.3  96092 12376 ?        Ssl  01:43   0:00 /usr/bin/command-router
hermes-+    5308  0.0  0.5 1316508 22480 ?       Sl   01:43   0:00 /opt/hermes/hermes_client --log-lev
service+    5365  0.0  0.1 416256  5740 ?        Sl   01:43   0:00 /opt/odin/service-ui
connman     5405  0.0  0.1   5696  4108 ?        Ss   01:43   0:00 /usr/sbin/connmand -c /etc/connman/
ssh-man+    5471  0.0  0.0   3444  2292 ?        Ss   01:43   0:00 /bin/sh /usr/bin/ssh-manager
mounterd    5482  0.0  0.2 250388  8428 ?        Ssl  01:43   0:00 /usr/bin/mounterd 0 0 1
sys-mon+    5546  0.0  0.1  13880  7916 ?        S    01:43   0:00 /usr/bin/sys-monitor
shell_h+    5562  0.0  0.0   3444  2204 ?        S    01:43   0:00 /bin/sh /usr/bin/shell-history-moni
shell_h+    5573  0.0  0.0   3444   512 ?        S    01:43   0:00 tail -n0 -q -F /var/log/.ash_histor
shell_h+    5579  0.0  0.0   3312   508 ?        S    01:43   0:00 cat /tmp/user_history.pipe
inducti+    5580  0.0  0.1  87708  7716 ?        Sl   01:43   0:00 /usr/bin/inductivechargerd
dnsmasq     5716  0.0  0.0   3036  1972 ?        S    01:43   0:00 /usr/sbin/dnsmasq --conf-file=/etc/
```

Each service runs with its own Linux UID

UID can be used for filtering network as well

Isolate and limit application rights

Sandboxes



Kafel
- Filter syscalls and basic parameters

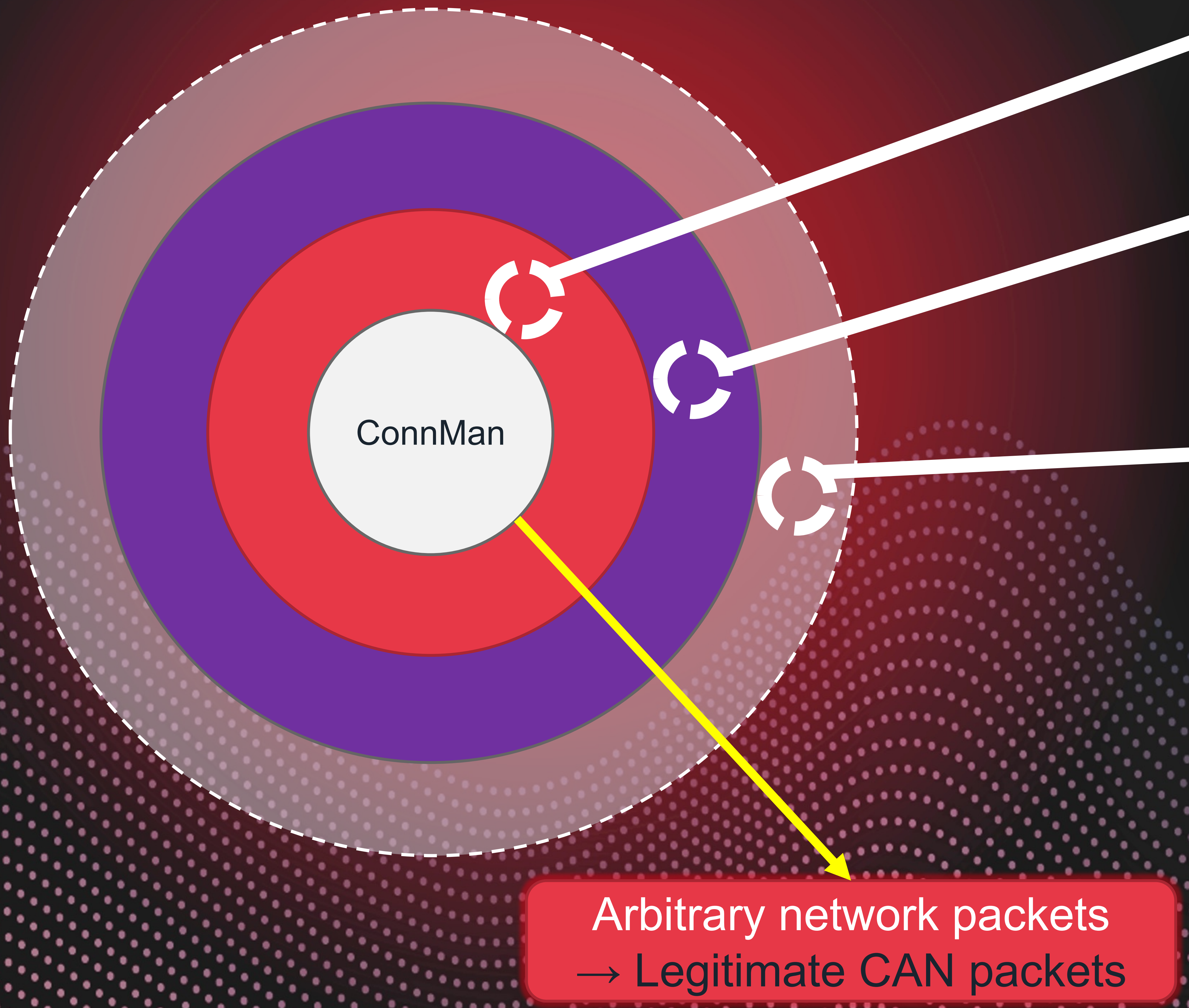
AppArmor
- Filter access to files and socket types

IPTables
- Filter network outputs

Minijail
- Isolate process into empty network namespace
- Chroot process

Isolate and limit application rights

Sandboxes escape example — **pwn2own 2022**



Kafel

- `socket` & `sendto` allowed

AppArmor

- `network packet dgram` allowed
- `capability net_raw` allowed

IPTables

- Output packets on raw socket are not filtered

Kafel & AppArmor let Connman create raw socket
Raw sockets bypass IPTables

Infotainment

Software System & Hardening

Principles

1. Limit the attack surface
2. Isolate and limit application rights
3. Make vulnerabilities harder to exploit
4. Patch vehicles OTA
5. Isolate and filter Ethernet and CAN networks
6. Protect user data and system integrity

Make vulnerabilities harder to exploit

Kernel

Kept up-to-date
—
Some hardening enabled
—
KASLR
—
ASLR for userland applications
—
No CFI

Binaries

Often built with PIE
—
Stack cookies enabled
—
Memory safe languages seem to be used for recently added services
—
No dynamic allocation for some critical services
—
No CFI

Libraries

Kept up-to-date, backport fixes
—
Libc has hardening enabled for heap management

Infotainment

Software System & Hardening

Principles

1. Limit the attack surface
2. Isolate and limit application rights
3. Make vulnerabilities harder to exploit
4. Patch vehicles OTA
5. Isolate and filter Ethernet and CAN networks
6. Protect user data and system integrity

Infotainment

OTA updates

Tesla uses an encrypted channel (Hermes) to communicate between its server and infotainment
Regular updates use this channel

From a security point of view, updates are used to:
- Fix vulnerabilities
- Add counter measures
- Improve sanbox configuration
- Update base software

Updates add features, this encourages users to apply them

Infotainment

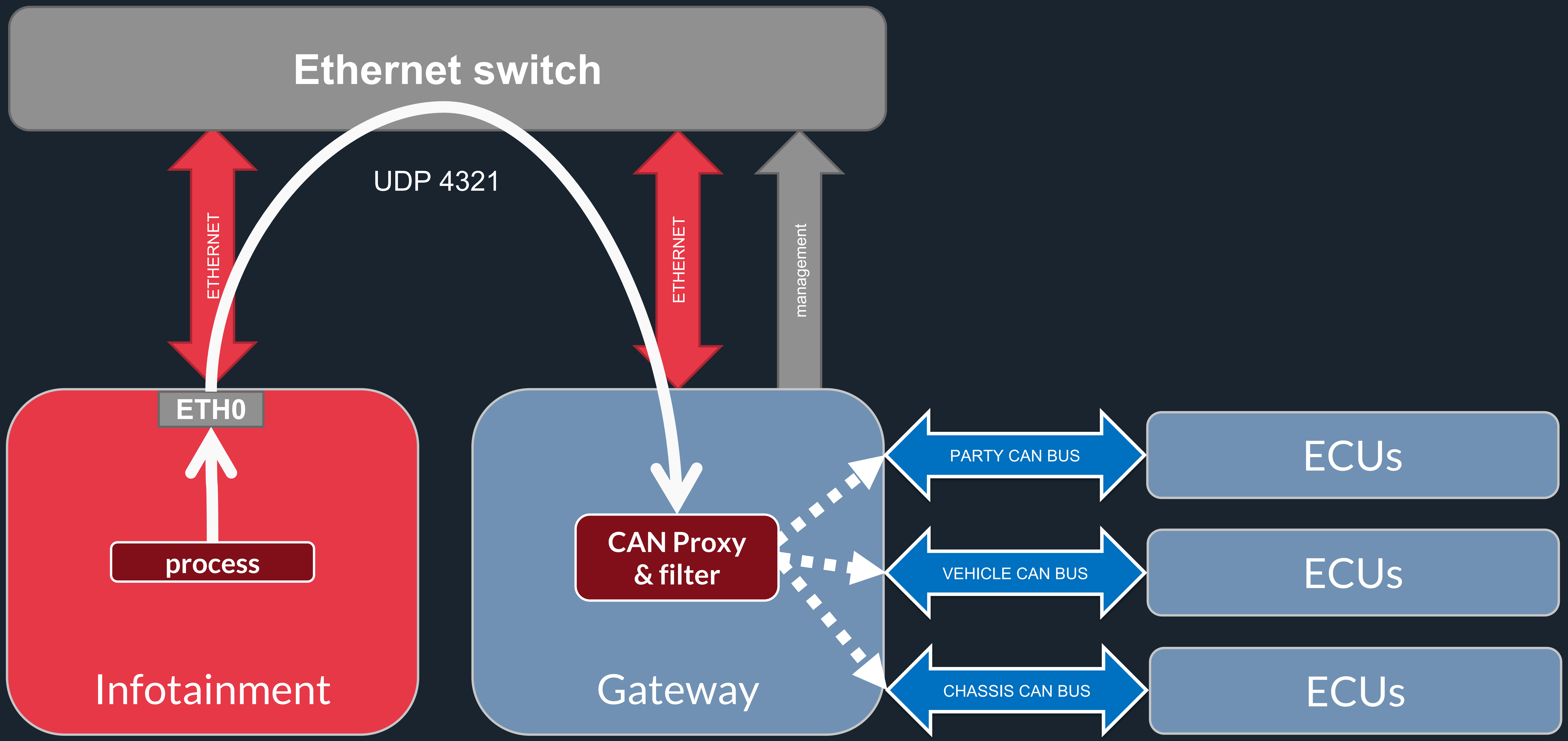
Software System & Hardening

Principles

1. Limit the attack surface
2. Isolate and limit application rights
3. Make vulnerabilities harder to exploit
4. Patch vehicles OTA
5. Isolate and filter Ethernet and CAN networks
6. Protect user data and system integrity

Isolate and filter Ethernet and CAN networks

Access to CAN buses



Isolate and filter Ethernet and CAN networks

Security Gateway

SYSTEM

Same PCB as Infotainment
—
SoC NXP MCP5748G
—
FreeRTOS PPC-VLE

NETWORKS

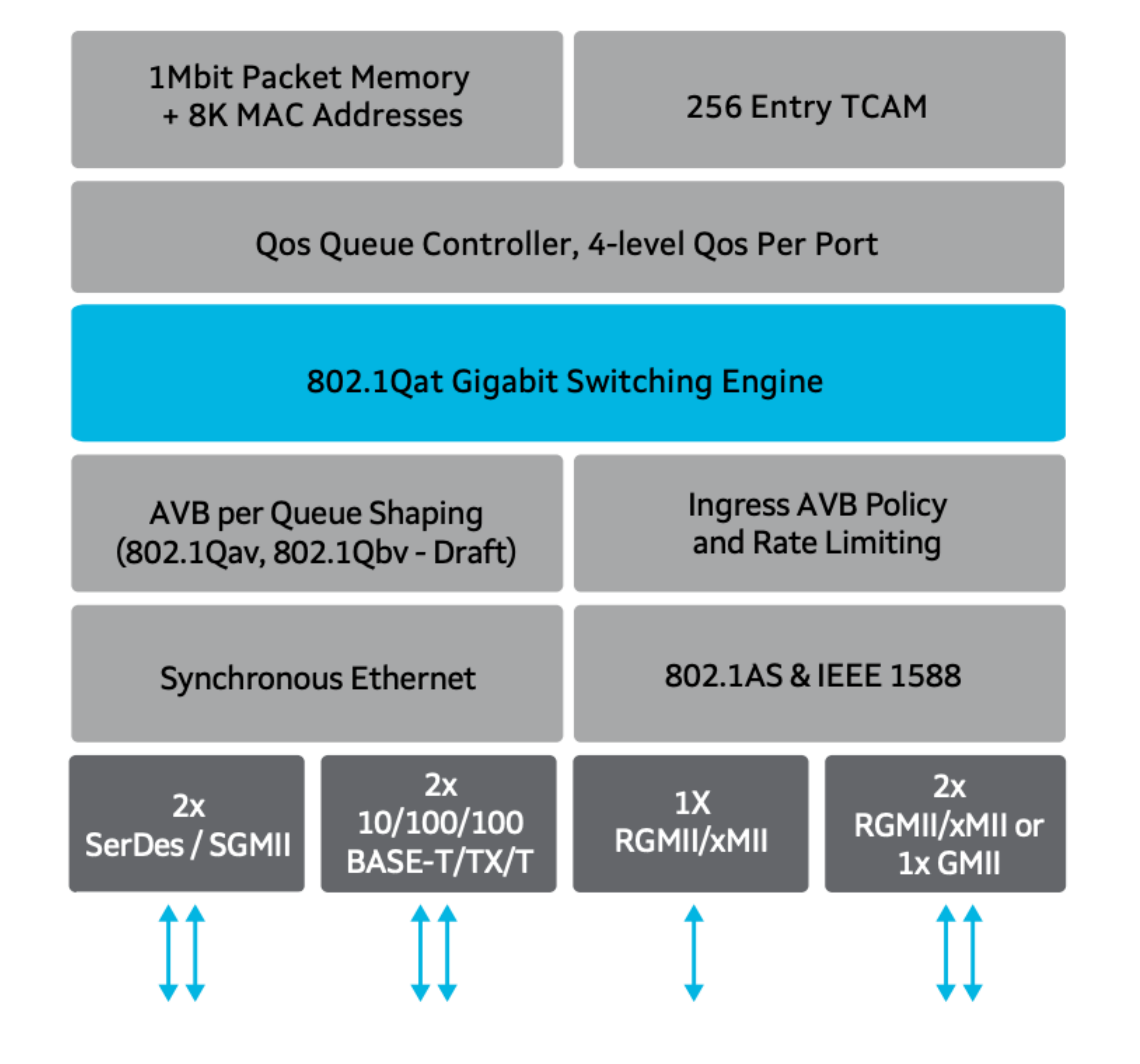
Ethernet
—
CAN buses (Chassis/Party/Vehicle)

Features

Filter CAN messages
—
Save log files
—
Update mode
Update other ECUs and itself
—
Provide sensitive information to other ECU (VIN/Serial/…)

Isolate and filter Ethernet and CAN networks

Ethernet Switch



Marvell 88ea6321

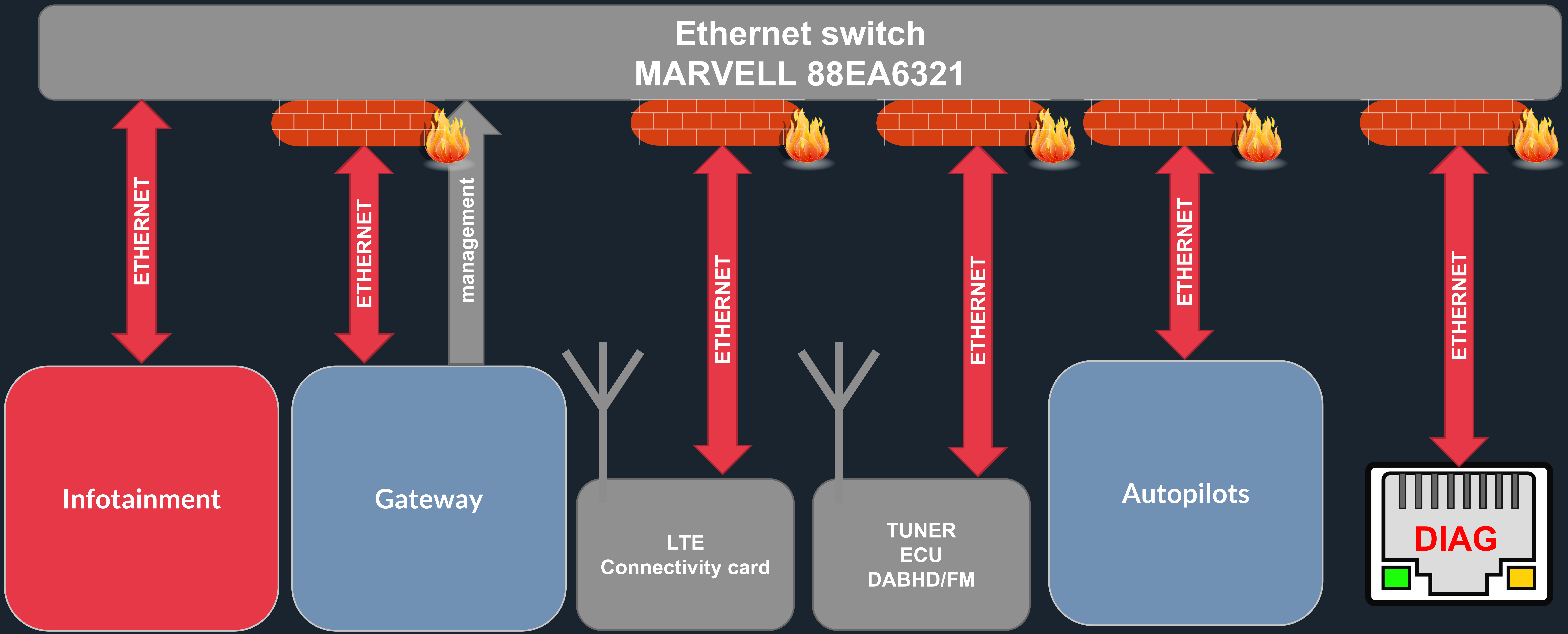
- 7 ports Switch
- Manageable
- No public datasheet
- Product brief only mention features

- 256 entries TCAM

- Gateway in charge of switch configuration over MDIO
- Ethernet remote management is disabled

Isolate and filter Ethernet and CAN networks

Ethernet Switch TCAM



Isolate and filter Ethernet and CAN networks

Ethernet Switch TCAM

```
-zsh
❯ python3 parse.py mdio.csv |egrep 'Read|Write' --color=none|
ress |head -n50
Write(phy=0x1a(0x1a), reg=0x00, value=0x9BF7)
Write(phy=PORT4(0x14), reg=PORT_IN_DISCARD_LO(0x10), value=0xF
Read(phy=PORT5(0x15), reg=PORT_STS(0x0)) = 0x0000
Write(phy=0xf(0xf), reg=0x1a, value=0xFFFF)
Read(phy=GLOBAL3(0x1d), reg=P2_DEBUG28(0x1c)) = 0x0000
Read(phy=0xf(0xf), reg=0x1d) = 0x0000
Write(phy=0x1f(0x1f), reg=0x1f, value=0xFFFF)
Write(phy=0x1f(0x1f), reg=0x1e, value=0xFFFF)
Write(phy=0x9(0x9), reg=0x1f, value=0xFFFF)
Read(phy=0xf(0xf), reg=0x1f) = 0x0000
Read(phy=0x9(0x9), reg=0x01) = 0x0000
Read(phy=0xd(0xd), reg=0x00) = 0x0000
Read(phy=0x8(0x8), reg=0x00) = 0x8000
Write(phy=0x1f(0x1f), reg=0x1f, value=0xFFFF)
Write(phy=GLOBAL1(0x1b), reg=MV88E6XXX_G1_STATS_COUNTER_01(0x1
Write(phy=0x1f(0x1f), reg=0x1f, value=0xFFFF)
Write(phy=0x1e(0x1e), reg=0x1f, value=0xFFFF)
Write(phy=0x0(0x0), reg=0x00, value=0x2FFF)
Write(phy=0xa(0xa), reg=0x00, value=0xFFFF)
Write(phy=0xc(0xc), reg=0x03, value=0xF7FF)
Read(phy=0xf(0xf), reg=0x1f) = 0xFFFF
Read(phy=0x17(0x17), reg=0x0f) = 0xFFFF
Write(phy=0x1f(0x1f), reg=0x1f, value=0xFFFF)
```

```
-zsh
❯ python3 parse_tcam.py boot_normal.csv
tcam entry    0 : src_port=3, dst_port=0, eth_type=0x0800,IPv4,TCP,tcp_dport=22,
tcam entry    1 : src_port=4, dst_port=0, eth_type=0x0800,IPv4,TCP,tcp_dport=22,
tcam entry    2 : src_port=3, dst_port=0, eth_type=0x0800,IPv4,TCP,tcp_dport=8080,
tcam entry    3 : src_port=4, dst_port=0, eth_type=0x0800,IPv4,TCP,tcp_dport=8080,
tcam entry    4 : src_port=3, dst_port=0, eth_type=0x0800,IPv4,TCP,tcp_dport=8081,
tcam entry    5 : src_port=4, dst_port=0, eth_type=0x0800,IPv4,TCP,tcp_dport=8081,
tcam entry    6 : src_port=0, dst_port=3,4, eth_type=0x0800,IPv4,TCP,tcp_sport=0,
tcam entry    7 : src_port=0, dst_port=3,4, eth_type=0x0800,IPv4,TCP,tcp_sport=7967,
tcam entry    8 : src_port=0, dst_port=3,4, eth_type=0x0800,IPv4,TCP,tcp_sport=7967,
tcam entry    9 : src_port=0, dst_port=1,2,3,4,5,6, eth_type=0x0806,
tcam entry   10 : src_port=3, dst_port=0, eth_type=0x0806,
tcam entry   11 : src_port=4, dst_port=0, eth_type=0x0806,
tcam entry   12 : src_port=1, dst_port=0, eth_type=0x0800,IPv4,ip_src=192.168.90.60/32
tcam entry   13 : src_port=1, dst_port=0, VLAN_TAG=81:00:00:14, eth_type=0x0800,
tcam entry   14 : src_port=1, dst_port=0, eth_type=0x0806,
tcam entry   15 : src_port=6, dst_port=0, eth_type=0x0800,IPv4,ip_src=192.168.90.30/32
tcam entry   16 : src_port=6, dst_port=0, eth_type=0x88f7,
tcam entry   17 : src_port=6, dst_port=0, VLAN_TAG=81:00:60:03, eth_type=0x22f0,
tcam entry   18 : src_port=6, dst_port=0, eth_type=0x0806,
tcam entry   19 : src_port=2, dst_port=0, eth_type=0x0800,IPv4,ip_src=192.168.90.103/32
tcam entry   20 : src_port=2, dst_port=0, eth_type=0x0800,IPv4,ip_src=192.168.90.104/31
tcam entry   21 : src_port=2, dst_port=0, eth_type=0x0806,
tcam entry   22 : src_port=0, dst_port=DROP ip_src=192.168.90.60/32
```

Infotainment

Software System & Hardening

Principles

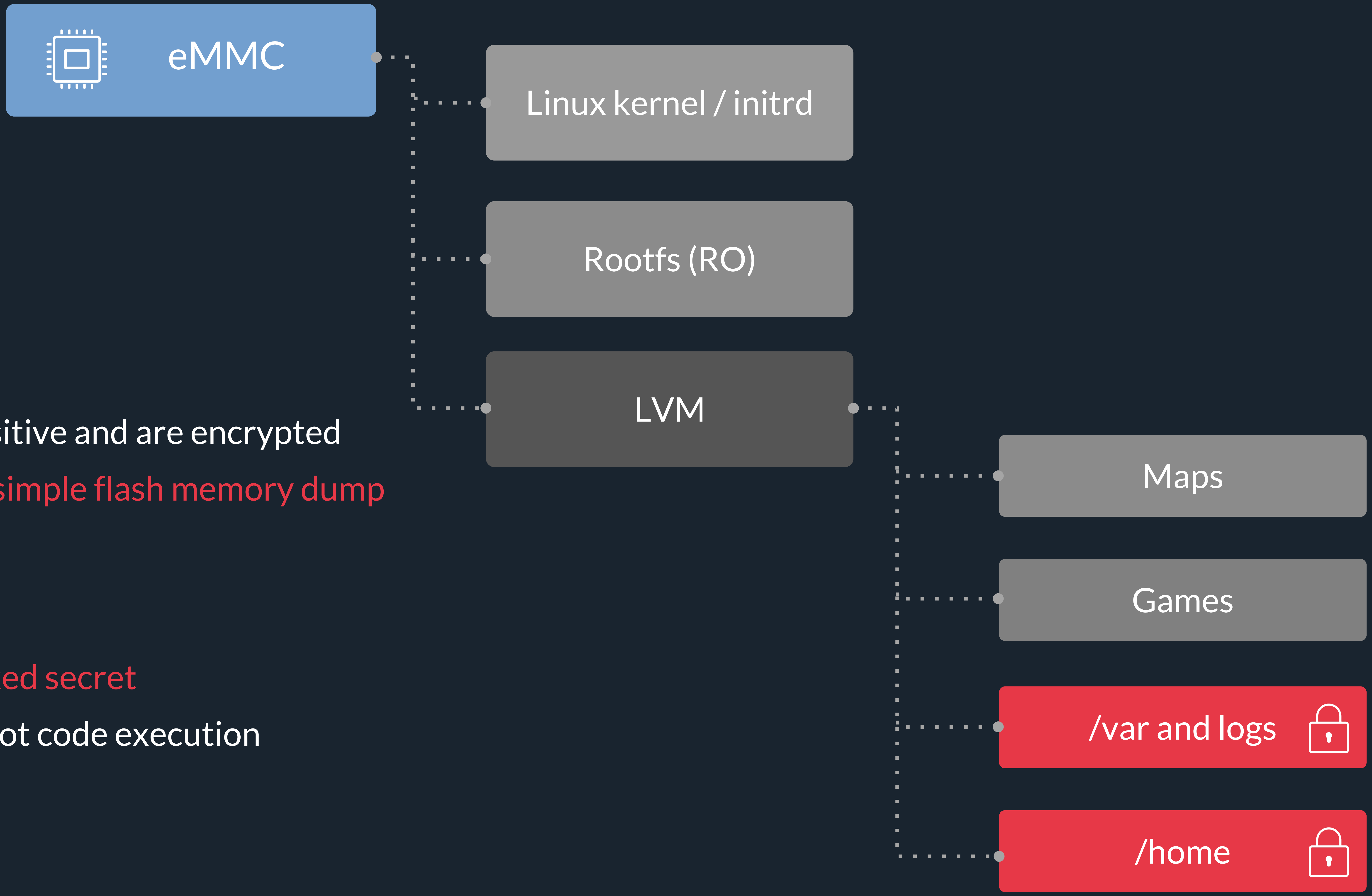
1. Limit the attack surface
2. Isolate and limit application rights
3. Make vulnerabilities harder to exploit
4. Patch vehicles OTA
5. Isolate and filter Ethernet and CAN networks
6. Protect user data and system integrity

Protect sensitive data

Partition scheme and encryption using LVM / LUKS

Which data ?

- Application data and credentials
- User data (login, passwords, cookies, history, ...)
- System files (writable configuration files, logs)



Confidentiality

- Some user or application data can be considered sensitive and are encrypted
- An attacker should not be able to access data from a simple flash memory dump

Encryption key

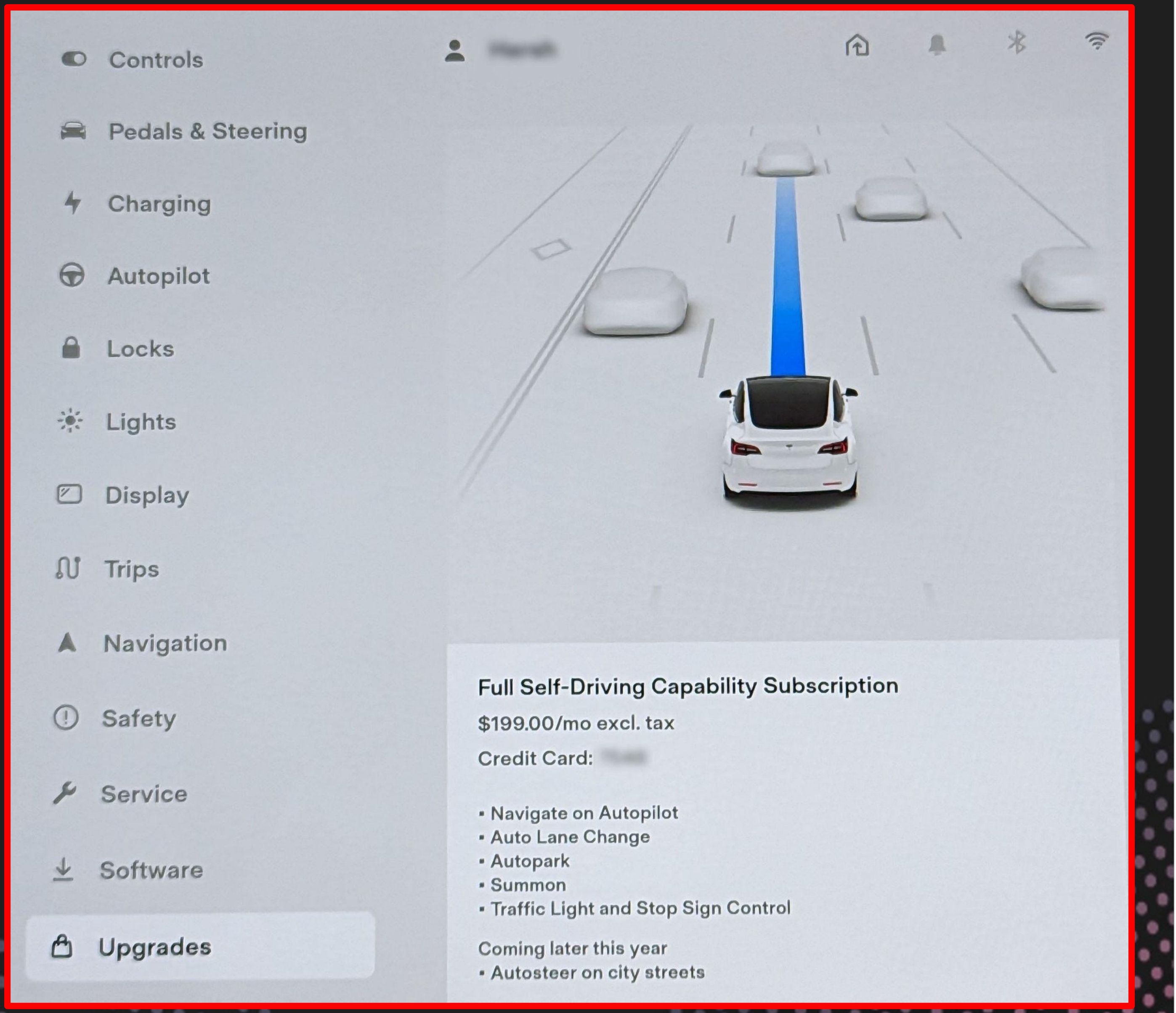
- Encryption keys are derivated from an hardware-backed secret
- Could be extracted if the attacker gains unconfined root code execution

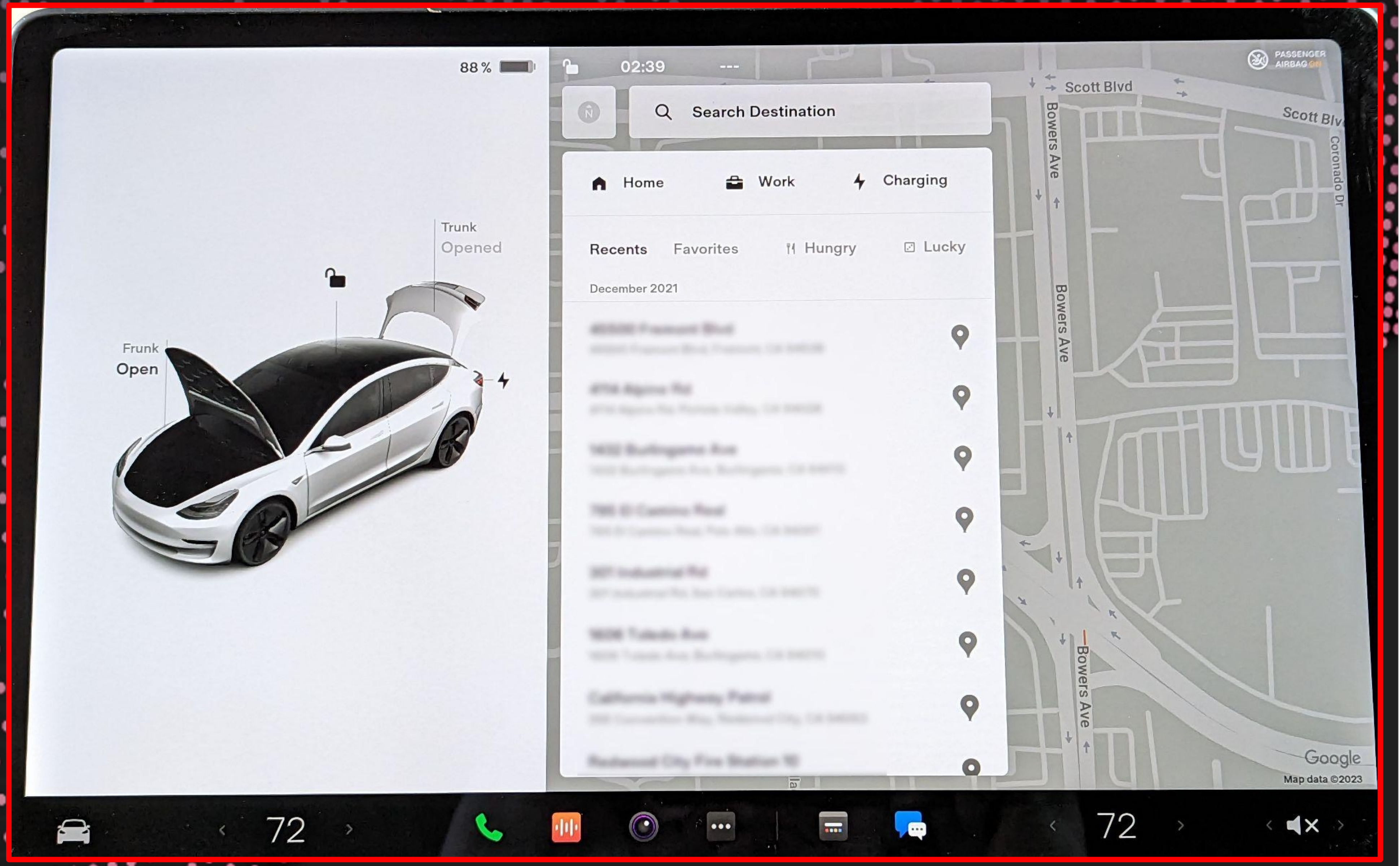
Protect sensitive data

User data embedded in the devices

In the wild products

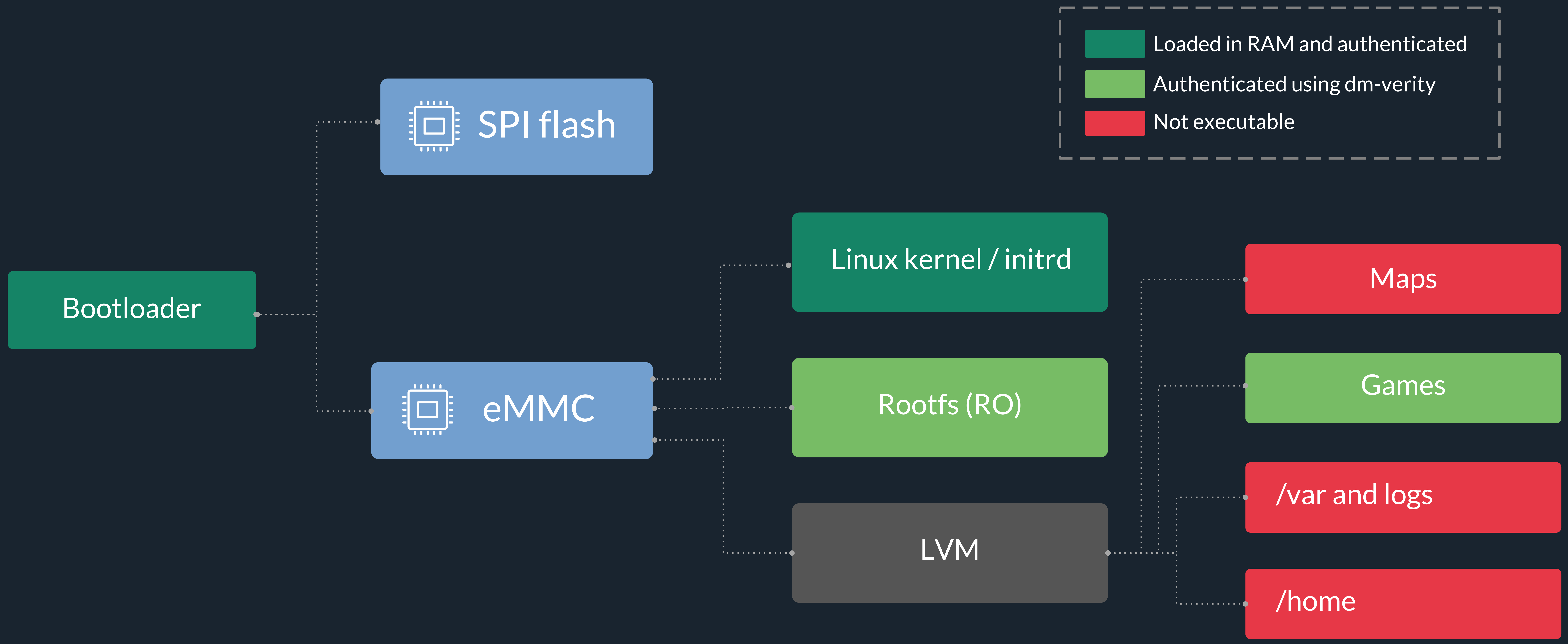
- It is hard to manage the product end of life
- ECUs bought on eBay still have personal data
- These ECUs probably come from damaged cars
- Some data are directly visible in the UI





Protect system integrity

Secure boot to authenticate executed code



Product security blue team

- Dedicated to manufactured products (cars)
- Works with the IT security team (infra)
- Handles all phases of product life

Design

Integrate the security from the early product design

Software and hardware architecture

Development

Code reviews
Recommendations

Validation

Security assessments

Production

Security updates
Product updates reviews
Detection
Incident response
Bug bounty / Pwn2Own

Intensive testing

Working with external researchers

External assessments

Ask cybersecurity companies to assess the product

Everything found can be useful (not only the impactful vulnerabilities)

Pwn2Own

International contest

Real world scenario and impactful demonstration only

Hardware (ECUs) is given to contestants

Security researcher program

- Researchers can register their car as test product
- Tesla helps researchers fix their car in case of broken ECU
- Gives a one-year root SSH key to researchers reporting vulnerabilities that allow to get root on Infotainment

Bug bounty

Ask security researchers from all around the world to find and report security issues

Conclusion

- Security has to be taken into account since the beginning. Cars have a long life cycle

- Lot of changes since 2016 on Tesla cars, they managed to update the software on production vehicle to improve their defences

- Even with a good architecture and hardening, they are still full chain attacks that allows CAN access remotely (pwn2own 2023)

- Tesla infotainment is increasingly similar to the smartphones in terms of security, with a few years late.

- Our researches only target Tesla, it is hoped that other automotive manufacturers and ECUs suppliers will conduct similar work

Questions ?