Retro-Ingenierie de systèmes embarqués AUTOSAR

SSTIC 2023
ETIENNE CHARRON & AXEL TILLEQUIN

Summary

01

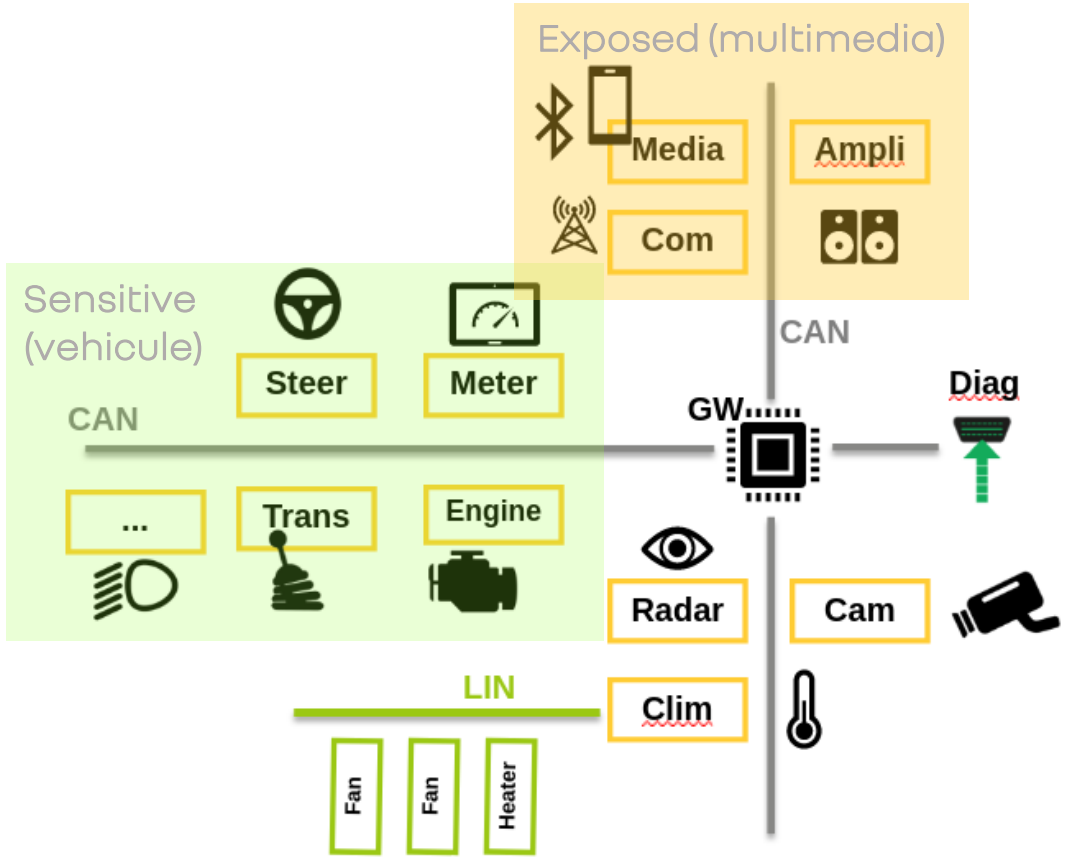
CONTEXT

Automotive architecture

- **Vehicule (= ~ 100 ECU)**

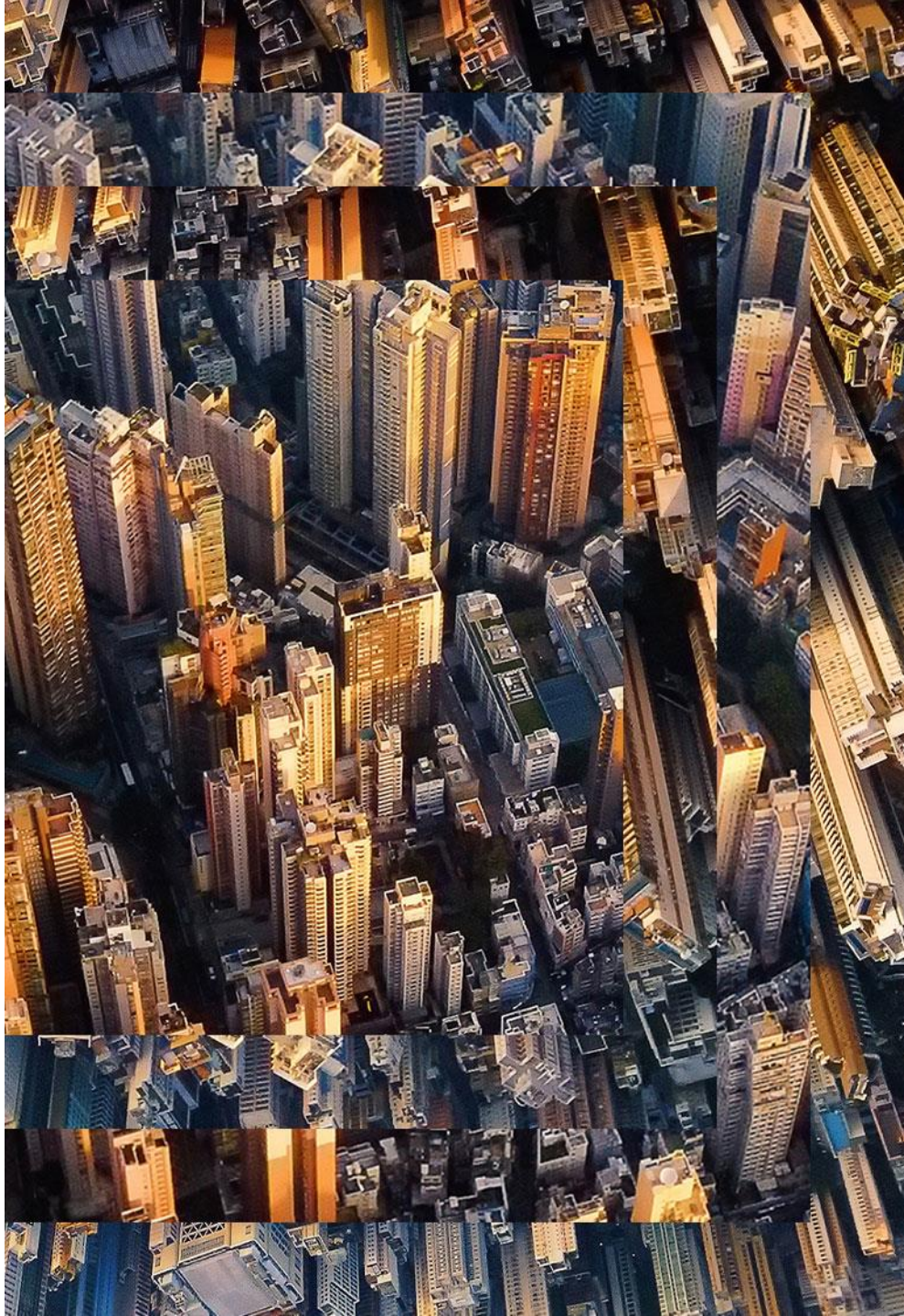
- **Cybersecurity impacts**
  - Safety (preserve passager life)
  - Data privacy (RGPD)
  - IT (Automobile knowledge)

- **Standard commonly used**
  - AUTOSAR : Software
  - UDS : Diagnostic protocol



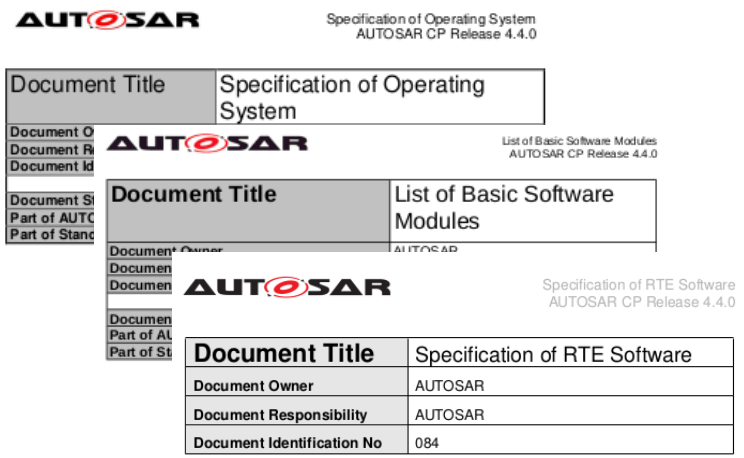
02

AUTOSAR



AUTomotive Open System ARchitecture

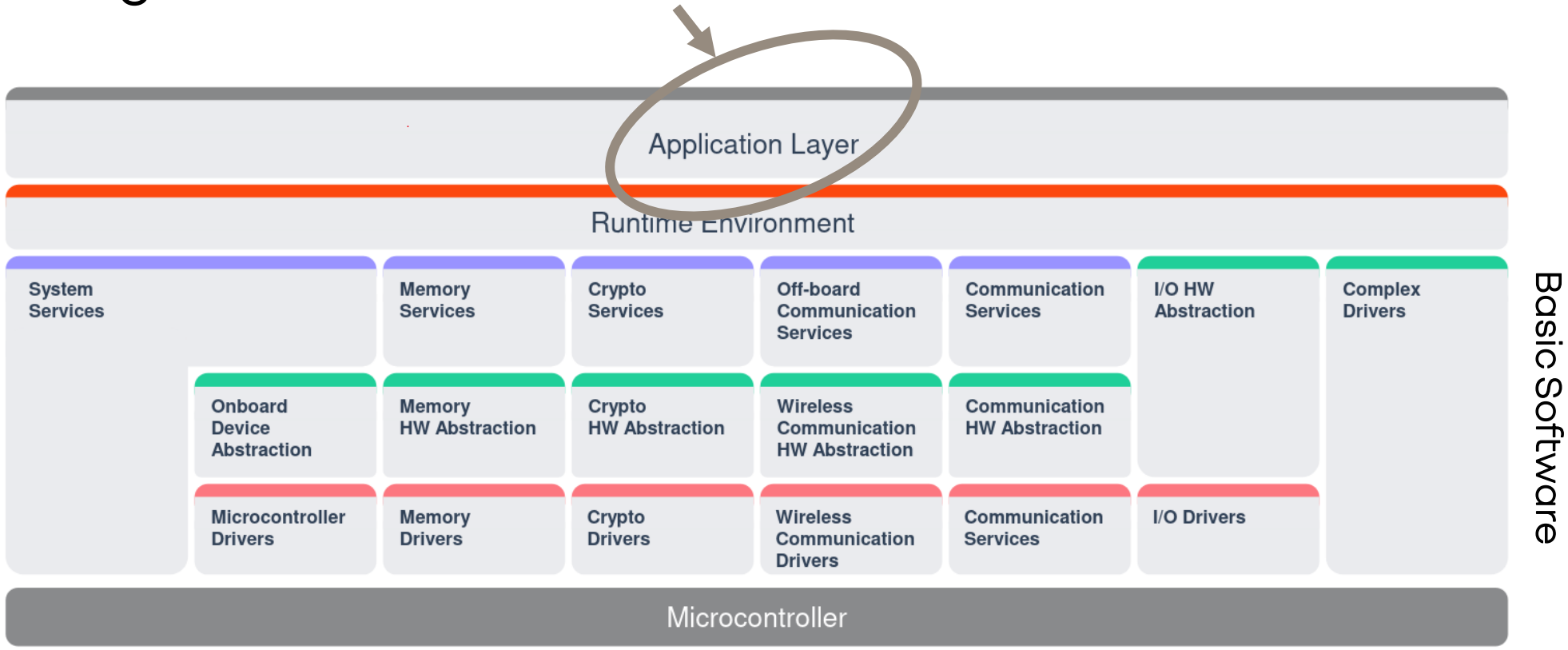
- **Publicly available set of PDFs that describes a system architecture**
- **Provides a methodology for design & build of ECU**
- **Diversity :**
  - AUTOSAR "Classic": hard real time, safety critical needs
    - Based on OSEK/VDX Standard for RTOS
  - AUTOSAR "Adaptive": high-perf, fail-operational needs
    - Based on POSIX
- **No "open-source" implementation...**



02 - AUTOSAR : AUTomotive Open System ARchitecture

RG

Goal: looking for vulnerabilities HERE



Problem: how to focus on Application Layer...
can we identify the BSW & RTE ?

Where is the data and where is the code ?

Ghidra detection

- ~ 10 k functions
- ~ 15 strings

SAMPLE

Difficulty

- No file format (PE, ELF)
- Exotic architecture
- No dependency (only one blob)
- Sometimes fully encrypted...

02 - AUTOSAR : Example of CAN specification (v4.4)

Service Name	CanIf_GetVersionInfo	
Syntax	`void CanIf_GetVersionInfo ( Std_VersionInfoType* VersionInfo )`	
Service ID [hex]	0x0b	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	VersionInfo	Pointer to where to store the version information of this module.
Return value	None	
Description	This service returns the version information of the called CAN Interface module.	

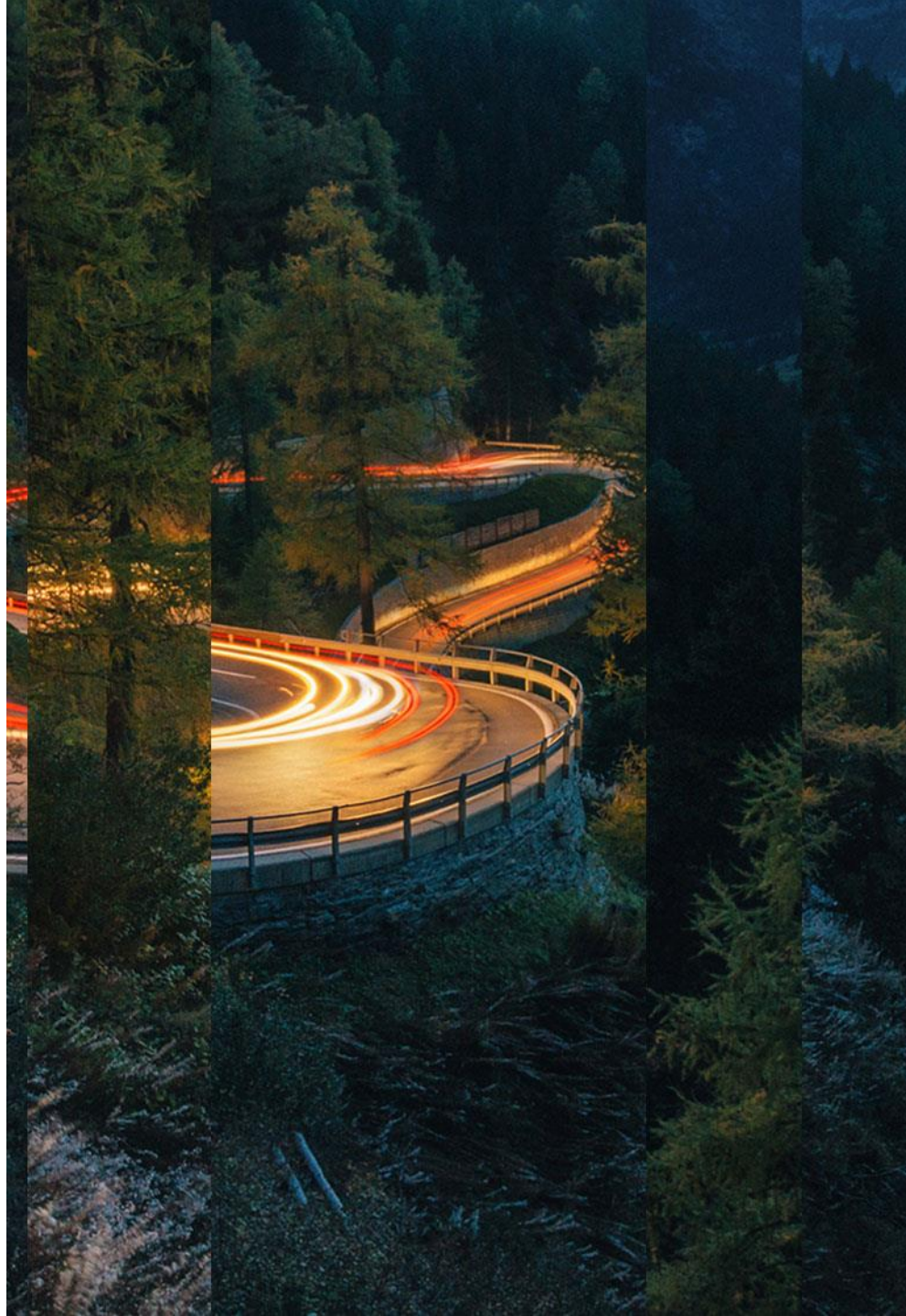
```
void FUN_80082932(int *param) {
  if (param != (int *)0x0) {
    *(undefined2 *)param = XX;
    *(undefined2 *)((int)param + 2) = 0x3c;
    *(undefined *)(param + 1) = 8;
    *(undefined *)((int)param + 5) = 1;
    *(undefined *)((int)param + 6) = 0;
    return;
  }
  FUN_8014d750(0,0x3c,0x14,0xb);
  return;
}
```


```
void CanIf_GetVersionInfo(std_VersionInfoType *info) {
  if (info != (Std_VersionInfoType *)0x0) {
    info->vendorID = XXXXXXX;
    info->moduleID = CANIF;
    info->instanceID = 8;
    info->sw_major_version = 1;
    info->sw_minor_version = 0;
    return;
  }
  Det_ReportError(0,0x3c,0x14,0xb);
  return;
}
```

03

FUNCTION IDENTIFICATION

USING DET MODULE



03 -FUNCTION IDENTIFICATION : DET "ERROR TRACER"

RG

Module

Service

CAN Interface 060

- CanIf_Init 0x01
- CanIf_SetBaudrate 0x21
- CanIf_Transmit 0x49
- ...

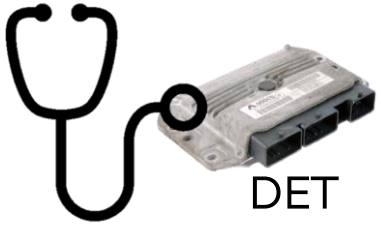
Crypto Service Manager 110

...

TCP/IP Stack 170

- TcpIp_Init 0x01
- TcpIp_TcpListen 0x07
- TcpIp_TcpReceived 0x08
- ...

...



AUTOSAR SPECIFICATION (DET)

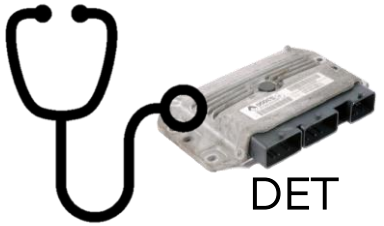
- **Module : Default Error Tracer (DET)**
  - "The API parameters allow for tracing source and kind of error (Module in which error has been detected; Function in which error has been detected; Type of error) "*

```
Det_ReportError( uint16 ModuleId, uint8 InstanceId, uint8 ApiId, uint8 ErrorId )

Det_ReportRuntimeError( uint16 ModuleId, uint8 InstanceId, uint8 ApiId, uint8 ErrorId)
```

ApiId == Service Id

- **DET characteristic function**
  - 4 parameters (type integers)
  - Log function (function used a lot of time)



* https://www.autosar.org/

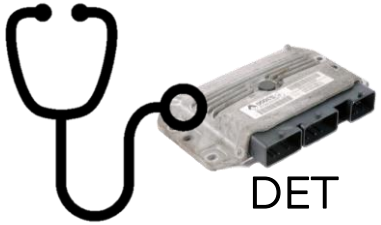
03 -FUNCTION IDENTIFICATION : DET

```
func_blue_80010200 (int param1, int param2) {
  ...
  Det_ReportError( 170, 0, 0x07, 0)
  ...
}
```

ModuleId = 170 ; ApiId = 0x07 -> TcpIp_TcpListen

```
func_blue_80010200 == TcpIp_TcpListen
Param1 ==  SocketId
Param2 == MaxChannels
```

 high probability



Example Automotive Firmware

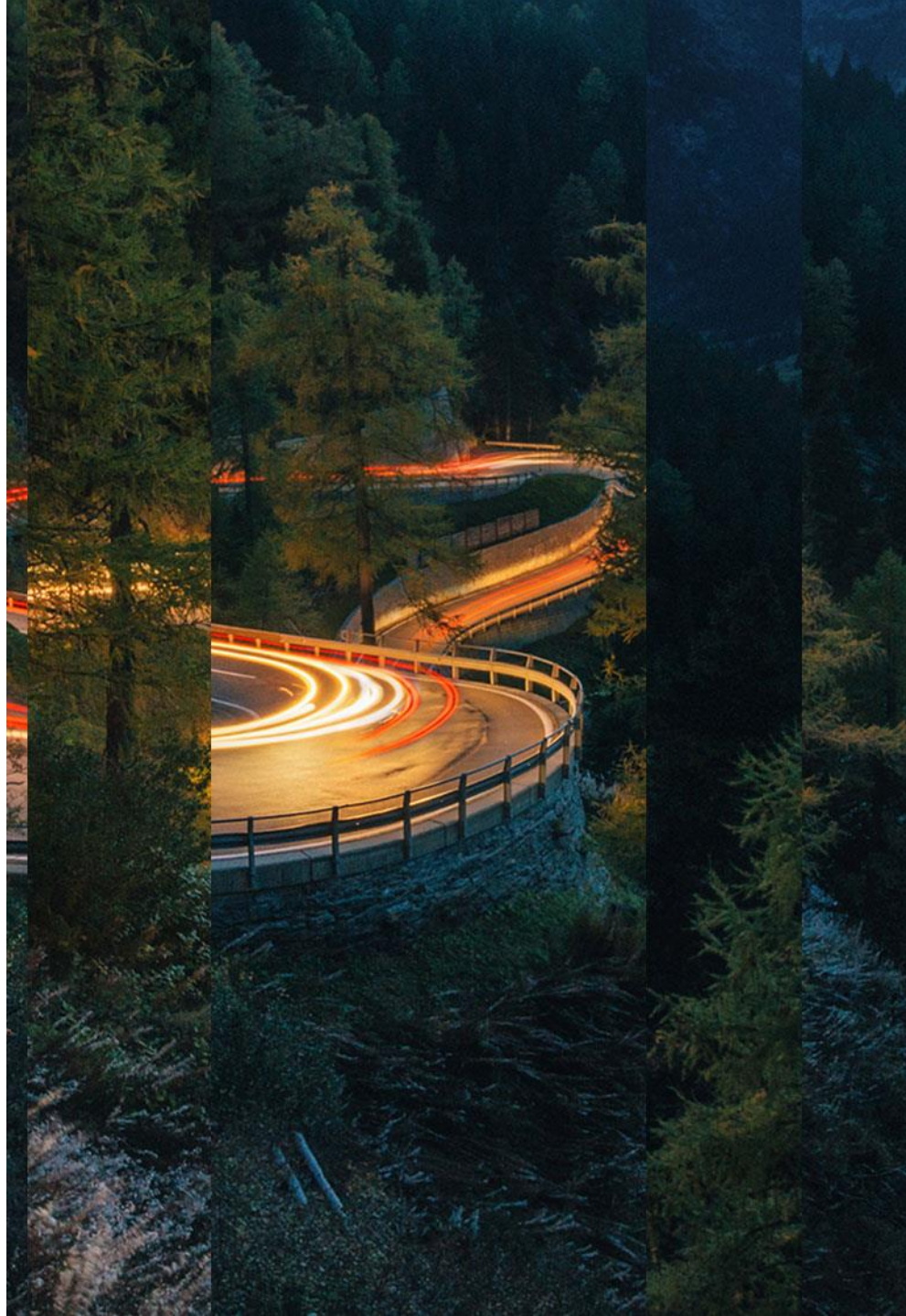
- ~10k detected functions (not documented)
- DET functions are in the TOP 3 most referenced
- ~10% functions prototype retrieved
- AUTOSAR version at least 4.3 (CryptoDriver)

Crypto, Ethernet, DOIP, OS (RTE), Memory manipulation, CAN, Socket…

03

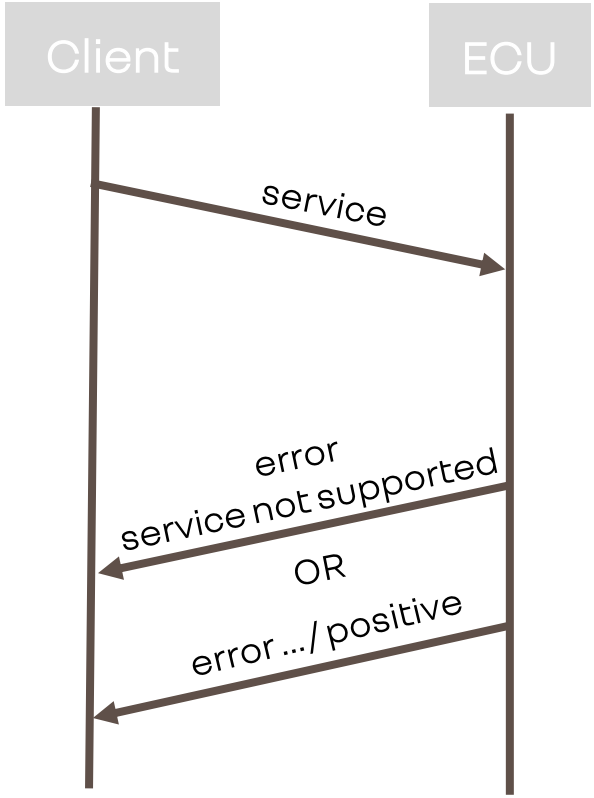
FUNCTION IDENTIFICATION

USING UDS PROTOCOL



UDS (DoCAN / DOIP)

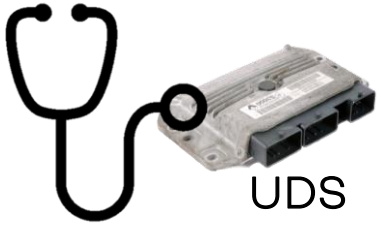
Diagnostic Protocol normalized by the ISO14229-1 allowing to reset the Equipement (ECU Reset), to update the Equipement (DataTransfer), Change configuration (Read Data By Identifier, Write Data By Identifier)




```
python nmap.py A komodo 18daf1d4 18dad4f1 services
10 Diagnostic Session Control
11 ECU Reset
14 Clear Diagnostic Information
19 Read DTC Information
22 Read Data By Identifier
27 Security Access
2e Write Data By Identifier
3e Tester Present
85 Control DTC Settings
```

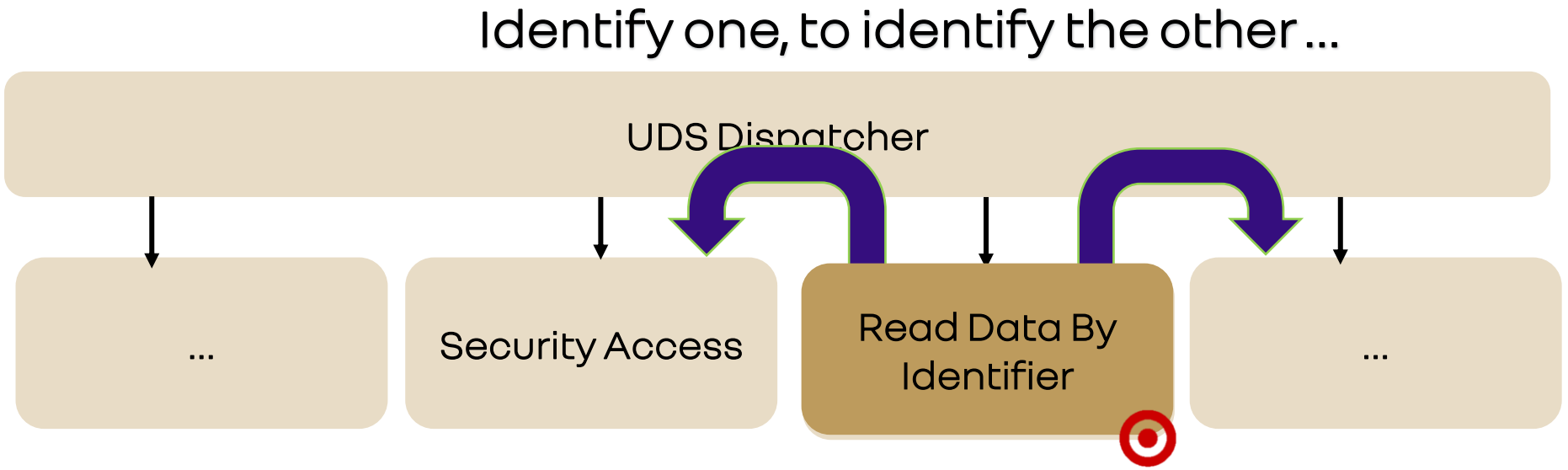
CANanalyze

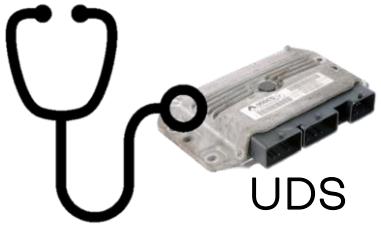
scapy



UDS

UDS (DoCAN / DOIP)

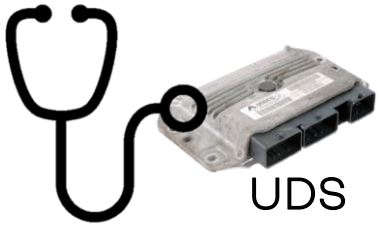




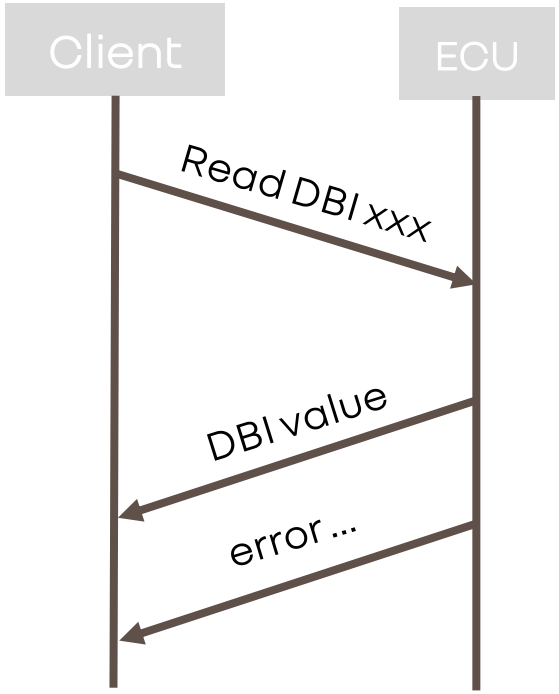
03 -FUNCTION IDENTIFICATION : UDS Dynamic Analysis

- Read Data By Identifier and Write Data By Identifier
  - Service allowing to get/set a configuration item identified by an integer (0x0 to 0xFFFF)
  - Generally, DID are readable without authentication

DBI	Content
F180	Supplier Identifier
F190	VIN
F193	Hardware Version Number
F195	Software Version Number
...	...



03 -FUNCTION IDENTIFICATION : UDS Dynamic Analysis



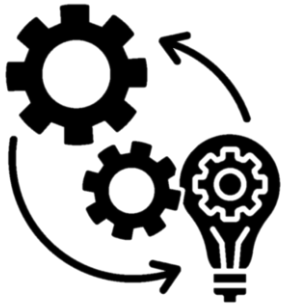
scapy CANanalyze

```
python rdbi.py A komodo 18daf1d4 18dad4f1
0x0100: XXXXXXXXXXXXXXX
0x0111: XXXXXXXXXXXXXXX
0x0112: XXXXXXXXXXXXXXX
…
0xf195: 'ver_10051'
…
0xfd14: XXXXXXXXXXXXXXX
0xfd15: XXXXXXXXXXXXXXX
0xfd16: XXXXXXXXXXXXXXX
```

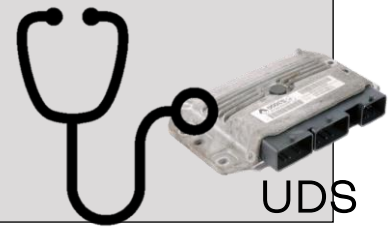
Harcoded Value

DBI LIST

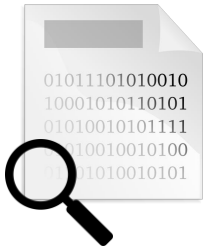
CANanalyze SAMPLE



```
strings firmware
…
ver_10051
…
```



Read Data By Identifier and Write Data By Identifier




```
'ver_10051'
```

reference

```
func_BLUE
{
…
copy(buffer,'ver_10051')
…
}
```

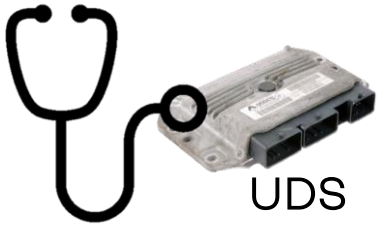
reference

```
801267bc ushort      F195h
801267be ushort      6h
801267c0 addr        func_GREEN
801267c4 addr        func_BLUE
801267c8 addr        00000000
801267cc int         0h
```

String exists in the binary
&&
Value DIDF195 == ver_10051

Find function manipulating « ver_10051 » string

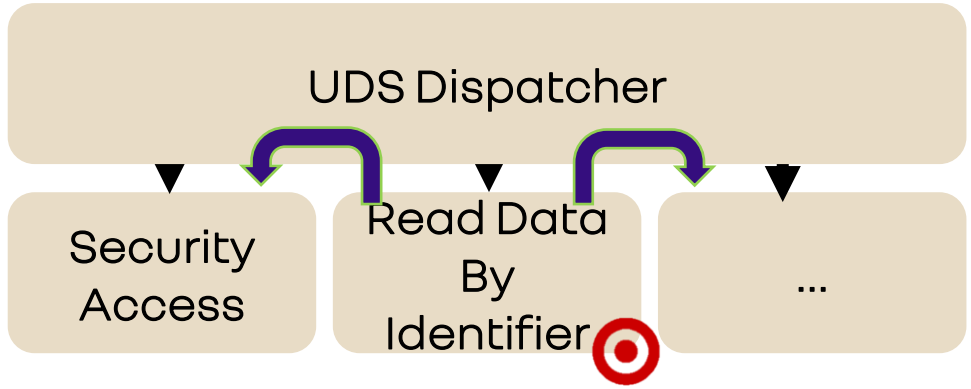
Find the DID structure array

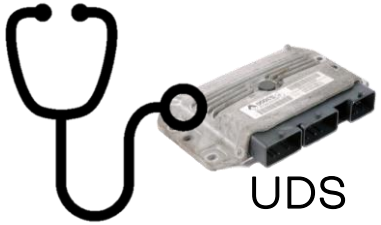


Example Automotive Firmware

- **~10k detected functions (not documented)**
- **~18% functions prototype retreived**
  - ~10% with DET method
  - ~8% with UDS read/write method
  - UDS hander retreived DBI handler
    - Security access handler
    - Ecu reset handler

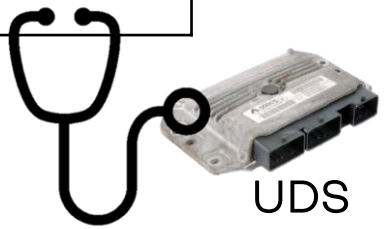
...





And other Firmwares

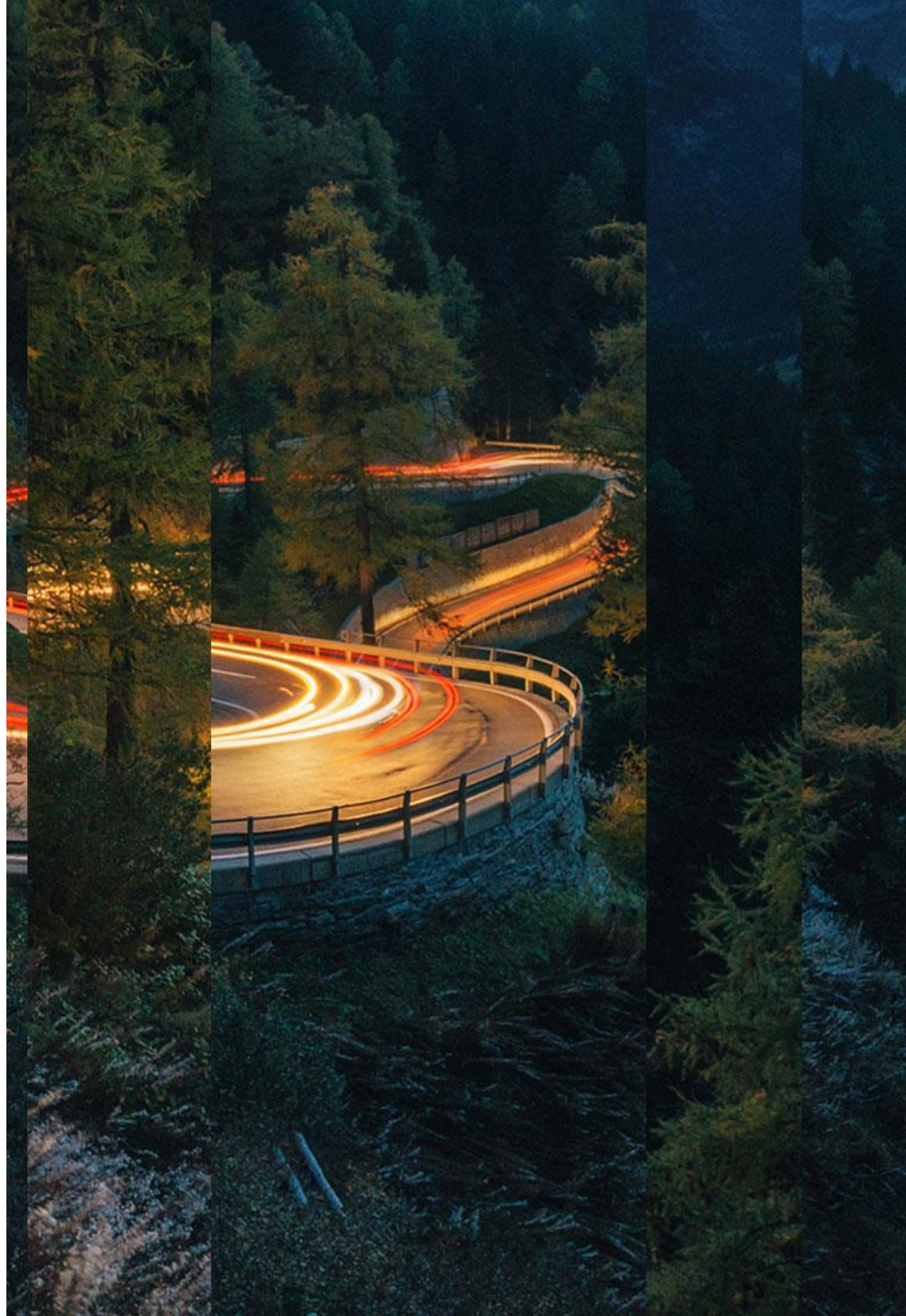
	Strings	DET % rename	DID% rename
Firm 1	15	10%	8%
Firm 2	5472	0%	1%
Firm 3	361	2%	4%
Firm 4	12	4%	2%



03

FUNCTION IDENTIFICATION

USING DATA STRUCTURES



Collecting information related to data structures:

For example:

- AURIX TC38x SoC (Tricore CPU x4) datasheet
  - describes memory mapped registers SCU, CAN, ETH, GPT, DMA, FCE, ...
  - all described by C structs

    (see https://github.com/Infineon/AURIX_code_examples)

```
$ ccrawl -g 'tricore' collect Libraries/iLLD/TC38A/Tricore/
...
```

- **Private knowledge gathered from previous work**

03 -FUNCTION IDENTIFICATION : Structure Identification

More generally: can we identify a function by how it operates on its data ?

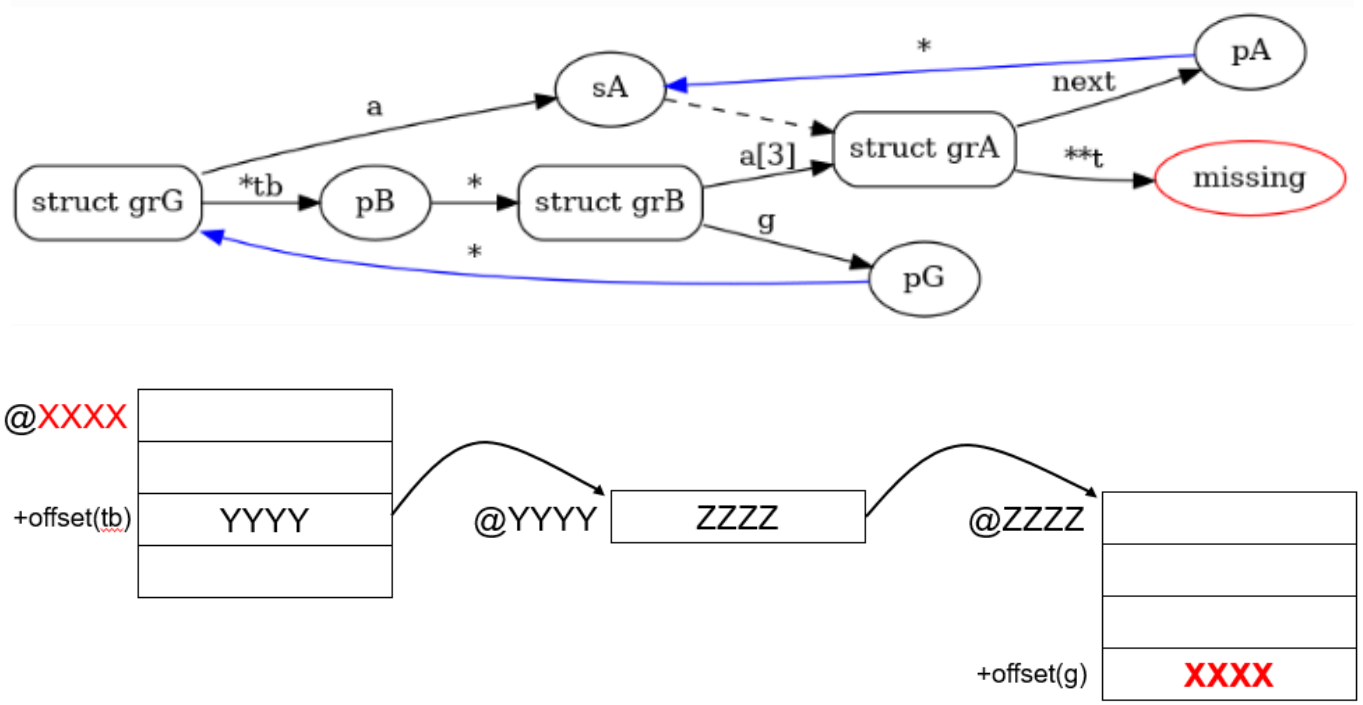
```
void FUN_80082932(int *param) {
  if (param != (int *)0x0) {
    *(undefined2 *)param = XX;
    *(undefined2 *)((int)param + 2) = 0x3c;
    *(undefined *)(param + 1) = 8;
    *(undefined *)((int)param + 5) = 1;
    *(undefined *)((int)param + 6) = 0;
    return;
  }
  FUN_8014d750(0,0x3c,0x14,0xb);
  return;
}
```

```
void CanIf_GetVersionInfo(std_VersionInfoType *info) {
  if (info != (Std_VersionInfoType *)0x0) {
    info->vendorID = XXXXXXX;
    info->moduleID = CANIF;
    info->instanceID = 8;
    info->sw_major_version = 1;
    info->sw_minor_version = 0;
    return;
  }
  Det_ReportError(0,0x3c,0x14,0xb);
  return;
}
```

03 –TASKS IDENTIFICATION : OS Structure(s) Identification

RG

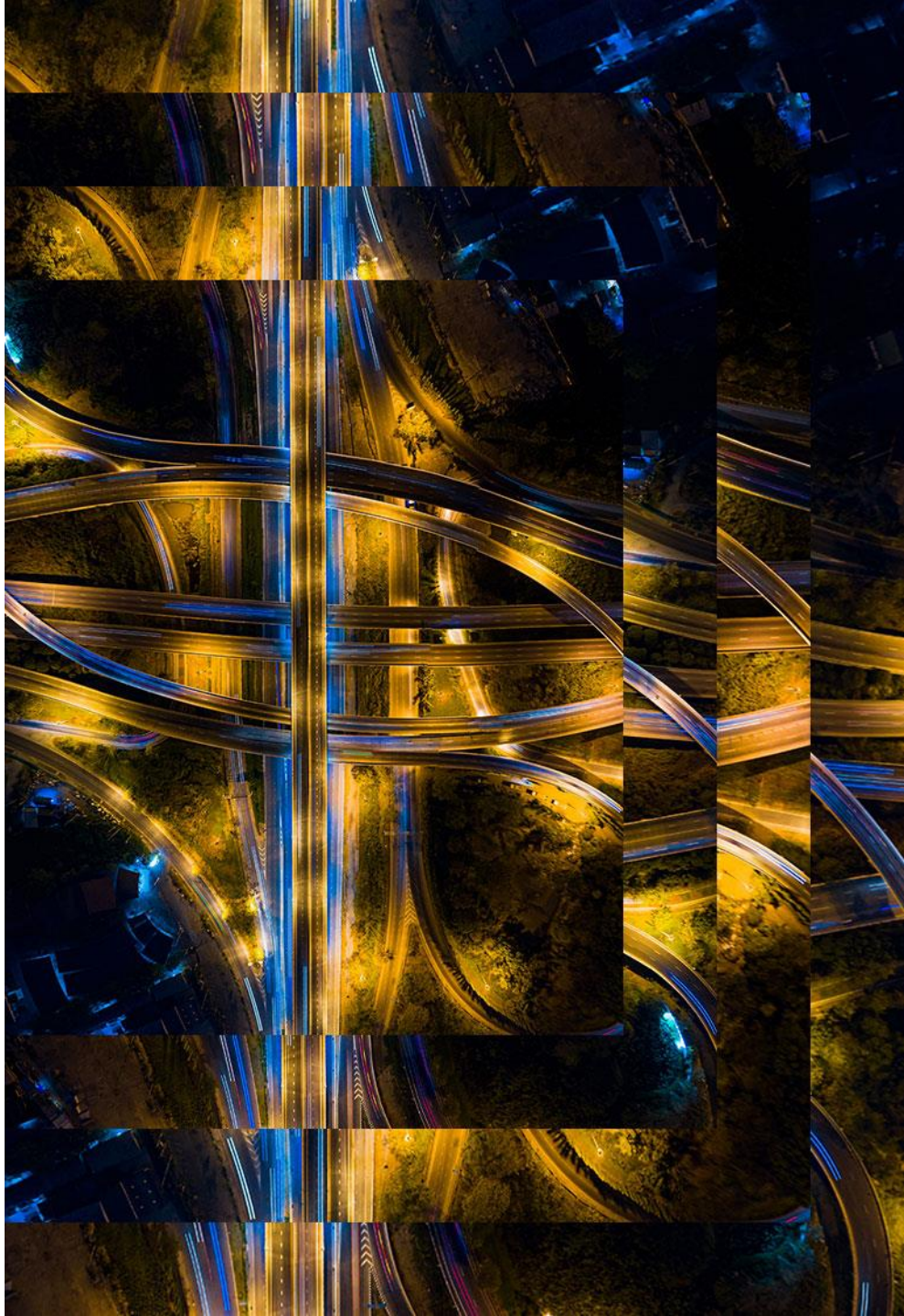
can we detect/locate OS known structures based on their cyclic-graph properties ?



04

DEMO &

PERSPECTIVES



Perspectives

- **Improving DET method**
  - subfunctions that contribute to a service may raise specific error codes…
  - Detect AUTOSAR version
- **Improving structure detection by propagating to/from call trees**
  - subfields struct pointer passed to a functions should match the subfield struct as well…

https://github.com/grouperenault/autosar-re
https://github.com/bdcht/ccrawl


RG
Thank you