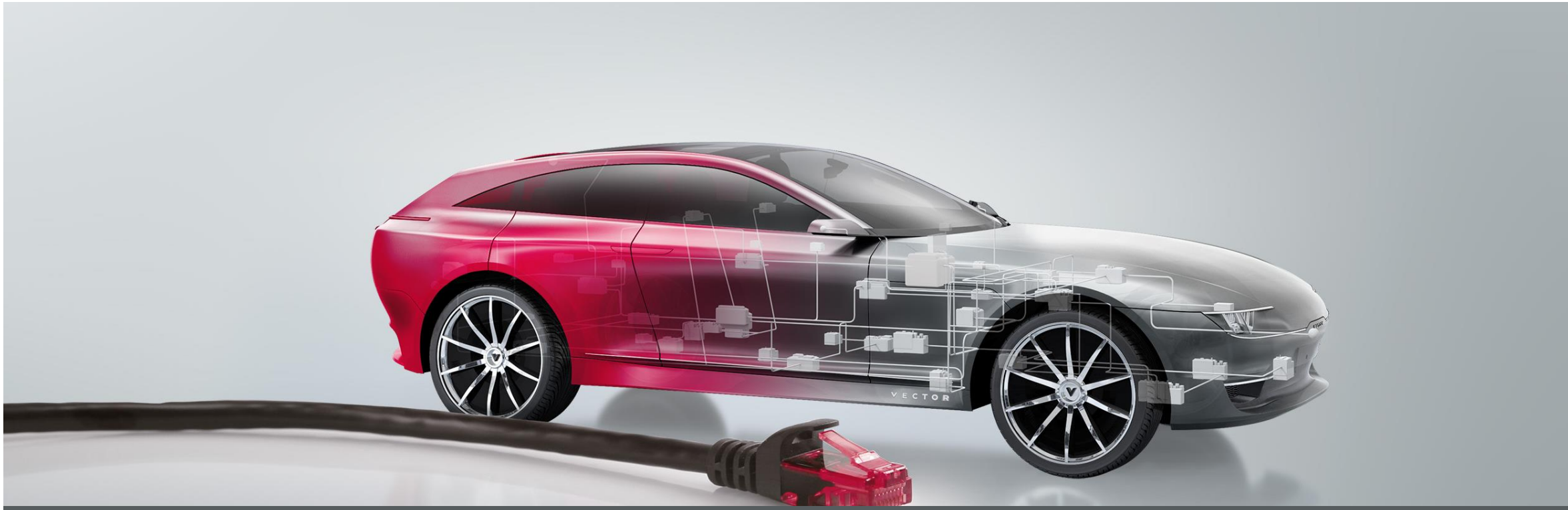
VECTOR



TLS Secured Connections in CANoe

Practical Approaches to Analyzing Data That Are Not Intended For Analysis

V1.01 | 2019-03-27

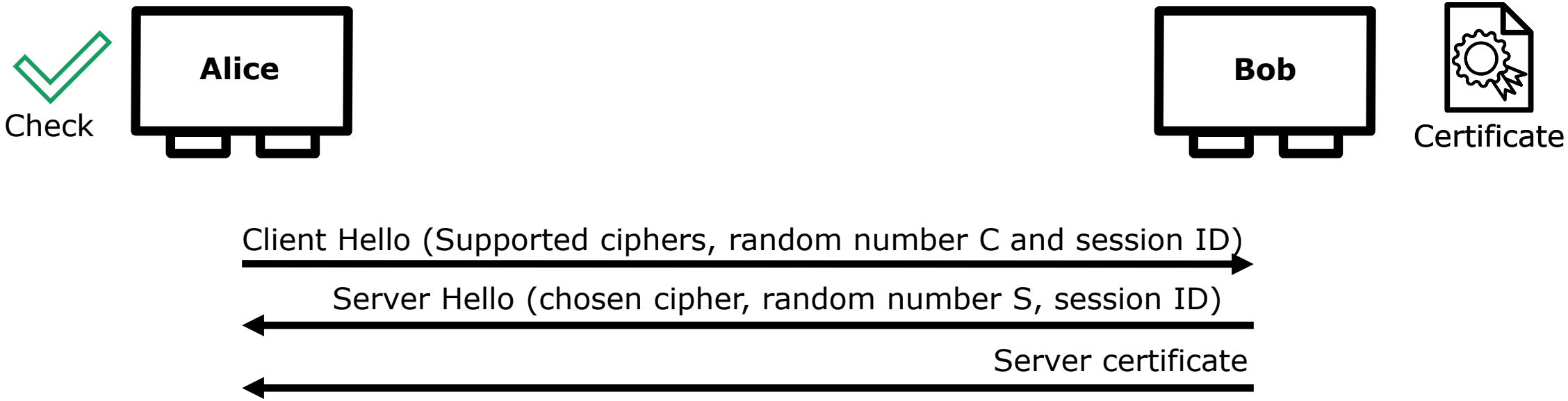
Agenda

- **Introduction**
- Software Development Impact
- End-Point Simulation
- Not End-Point Simulation
- Conclusion

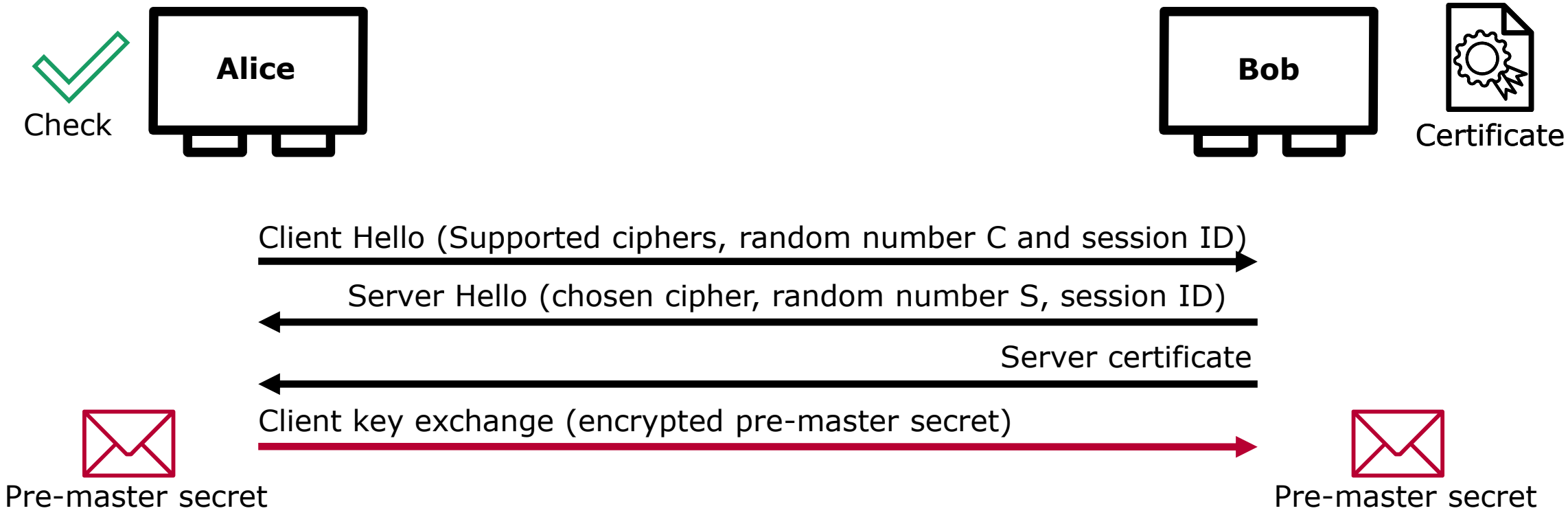
TLS in a Nutshell

- TLS stands for Transport Layer Security
- It secures the TCP communication above the Transport Layer (ISO/OSI Model Layer 4)
  - It is a pure point-to-point communication, broadcast is not possible
- Typically TLS allows Authentication of the Server and the Client and encrypts the payload content
  - The connection can still be traced
- Well known application domains are HTTPs and
- SmartCharge communication in automotive domains

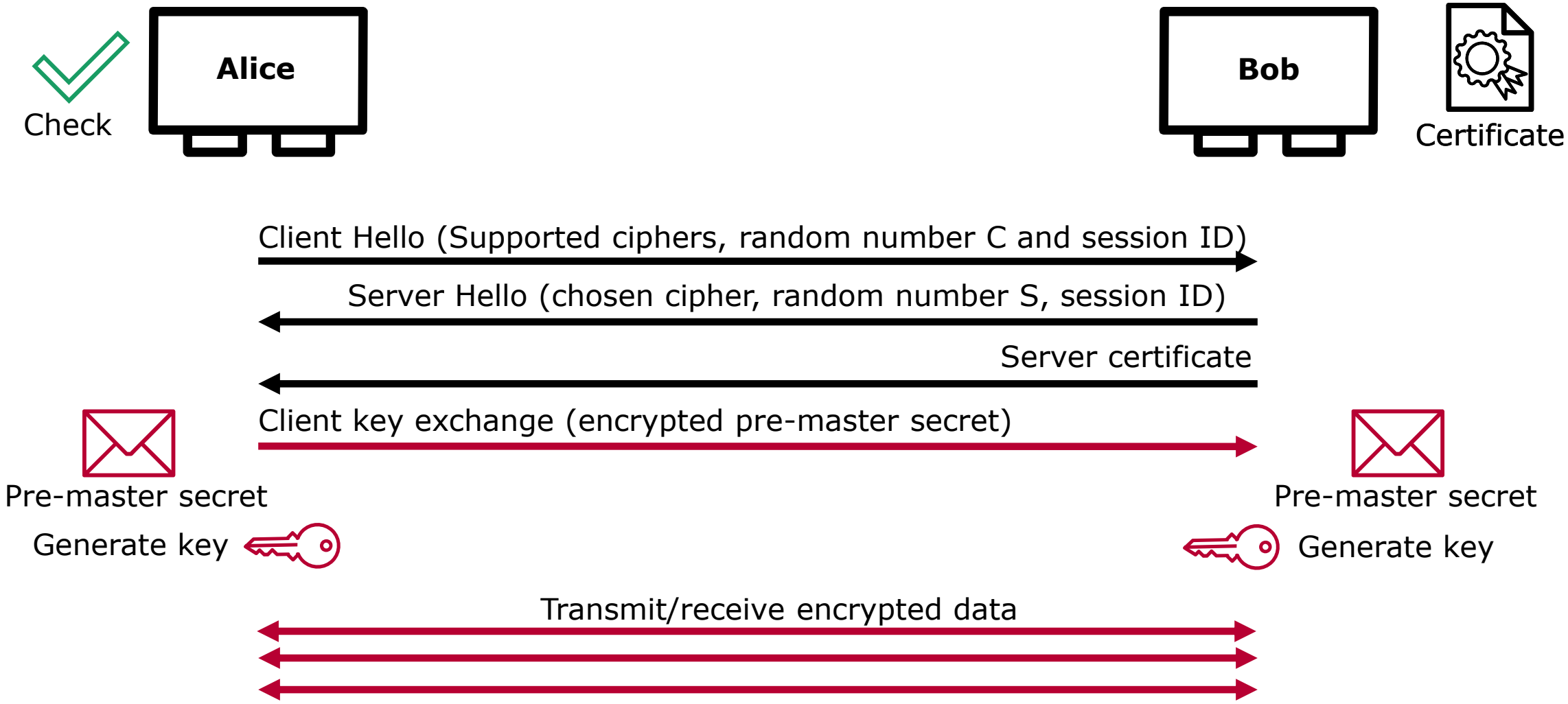
How Does TLS Work?



How Does TLS Work?



How Does TLS Work?



Agenda

Introduction

- **Software Development Impact**

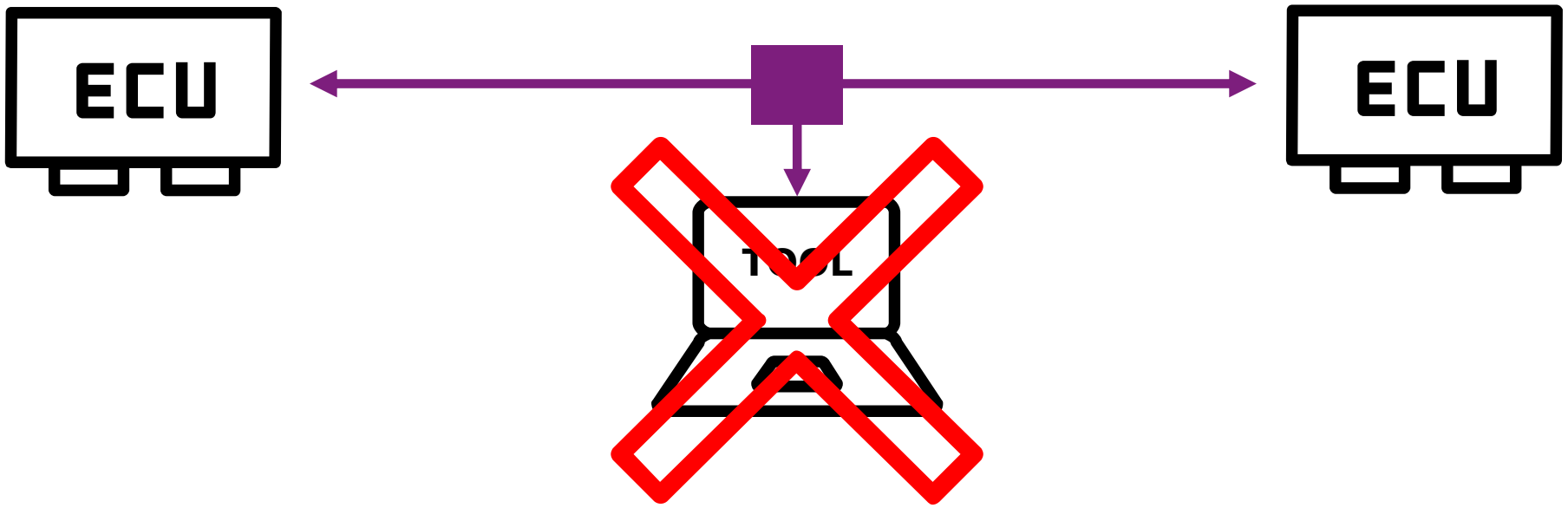
End-Point Simulation

Not End-Point Simulation

Conclusion

Consequence of Security for the Development Environment

- Handling of security material like keys and certificates
  - This has practically the highest impact, but out of scope for this presentation
  - Tools like the Security Manger of Vector may help to overcome certain issues while development phase
- Unable to observe communication



Trace-Fenster

- Example: TLS secured smart charge communication

Trace All

<Search> Initial

Time	Protocol	Name	Protocol Info	Source IP	Destination IP	Packet Length	Payload Length	Source Port	Destination Port	Payload Data ASCII	Payload Data
10.384036	sdp	SECC Discovery Request	TP version: 1	FE80::2	FF02::1	10	2	F1F7	3B0E	..	00 00
Security	TLS										
Transport Layer	TCP										
10.404043	ndp		Neighbor Solicitation	FE80::11	FF02::1:FF00:2	86	0				
10.404049	ndp		Neighbor Advertisement	FE80::2	FE80::11	86	0				
10.404057	sdp	SECC Discovery Response	TP version: 1	FE80::11	FE80::2	28	20	3B0E	F1F7	.....	FE 80 00 00 00
Security	TLS										
Transport Layer	TCP										
EVSE IP	FE80:0:0:0:0:0:0:11										
EVSE TCP Port	51111										
10.404063	tcp		CF96 -> C7A7 [SYN] Seq=32FC4660 Win=FFFF	FE80::2	FE80::11	78	0	CF96	C7A7		
10.404069	tcp		C7A7 -> CF96 [ACK, SYN] Seq=143BD5CD Ack=32FC4661 Win=FFFF	FE80::11	FE80::2	78	0	C7A7	CF96		
10.404075	tcp		CF96 -> C7A7 [ACK] Seq=32FC4661 Ack=143BD5CE Win=FFFF	FE80::2	FE80::11	74	0	CF96	C7A7		
10.404113	tls		TLS 1.0: Handshake (Client Hello)	FE80::2	FE80::11	472	398	CF96	C7A7	.....\..|):..#._|>S.p.G..I.&#R..g../w...	16 03 01 01 89
10.404126	tls		TLS 1.2: Handshake (Server Hello)	FE80::11	FE80::2	170	96	C7A7	CF96	....[...W..\..|.9...Z.....wp.....@..Z	16 03 03 00 5B
10.404248	tls		TLS 1.2: Handshake	FE80::11	FE80::2	1514	1440	C7A7	CF96	.....0...0.....F.....0...*.H....	16 03 03 07 D1
10.404253	tcp		CF96 -> C7A7 [ACK] Seq=32FC47EF Ack=143BDBCE Win=FD20	FE80::2	FE80::11	74	0	CF96	C7A7		
10.404323	tls		TLS 46.68: Type 0, Type 135	FE80::11	FE80::2	870	796	C7A7	CF96	.0D. >._.....i....YUm.~Y{.0..i..z4 .". U..Q...	00 30 44 02 20
10.404340	tls		TLS 1.2: Handshake (Client Key Exchange)	FE80::2	FE80::11	217	143	CF96	C7A7	.....p....j....{;.Gl.7.....Go.z.II.....	16 03 03 00 8A
10.500006	tcp		C7A7 -> CF96 [ACK] Seq=143BDEEA Ack=32FC487E Win=FFFF	FE80::11	FE80::2	74	0	C7A7	CF96		
10.500015	tls		TLS 1.2: Change Cipher Spec, Handshake	FE80::2	FE80::11	117	43	CF96	C7A7	..... e^..>..E.s.,.Z%...~'..(.Vp_...G@	14 03 03 00 01
10.500022	tls		TLS 1.2: Change Cipher Spec	FE80::11	FE80::2	80	6	C7A7	CF96	.....	14 03 03 00 01
10.600006	tcp		CF96 -> C7A7 [ACK] Seq=32FC48A9 Ack=143BDEF0 Win=FFFF	FE80::2	FE80::11	74	0	CF96	C7A7		
10.600015	tls		TLS 1.2: Handshake	FE80::11	FE80::2	111	37	C7A7	CF96	....F. ,.....@s.E...J..n/.z...o	16 03 03 00 20
10.700006	tcp		CF96 -> C7A7 [ACK] Seq=32FC48A9 Ack=143BDF15 Win=FFFF	FE80::2	FE80::11	74	0	CF96	C7A7		
10.900028	tls		TLS 1.2: Application Data	FE80::2	FE80::11	171	97	CF96	C7A7	....\.u|1c../..<..y}{..@A..|.}.._...z....H.J....	17 03 03 00 5C
11.000006	tcp		C7A7 -> CF96 [ACK] Seq=143BDF15 Ack=32FC490A Win=FFFF	FE80::11	FE80::2	74	0	C7A7	CF96		
11.200037	tls		TLS 1.2: Application Data	FE80::11	FE80::2	107	33	C7A7	CF96	.....u.V.....n.....fQ..+.Y.@.y<	17 03 03 00 1C
11.300006	tcp		CF96 -> C7A7 [ACK] Seq=32FC490A Ack=143BDF36 Win=FFFF	FE80::2	FE80::11	74	0	CF96	C7A7		

Trace-Fenster

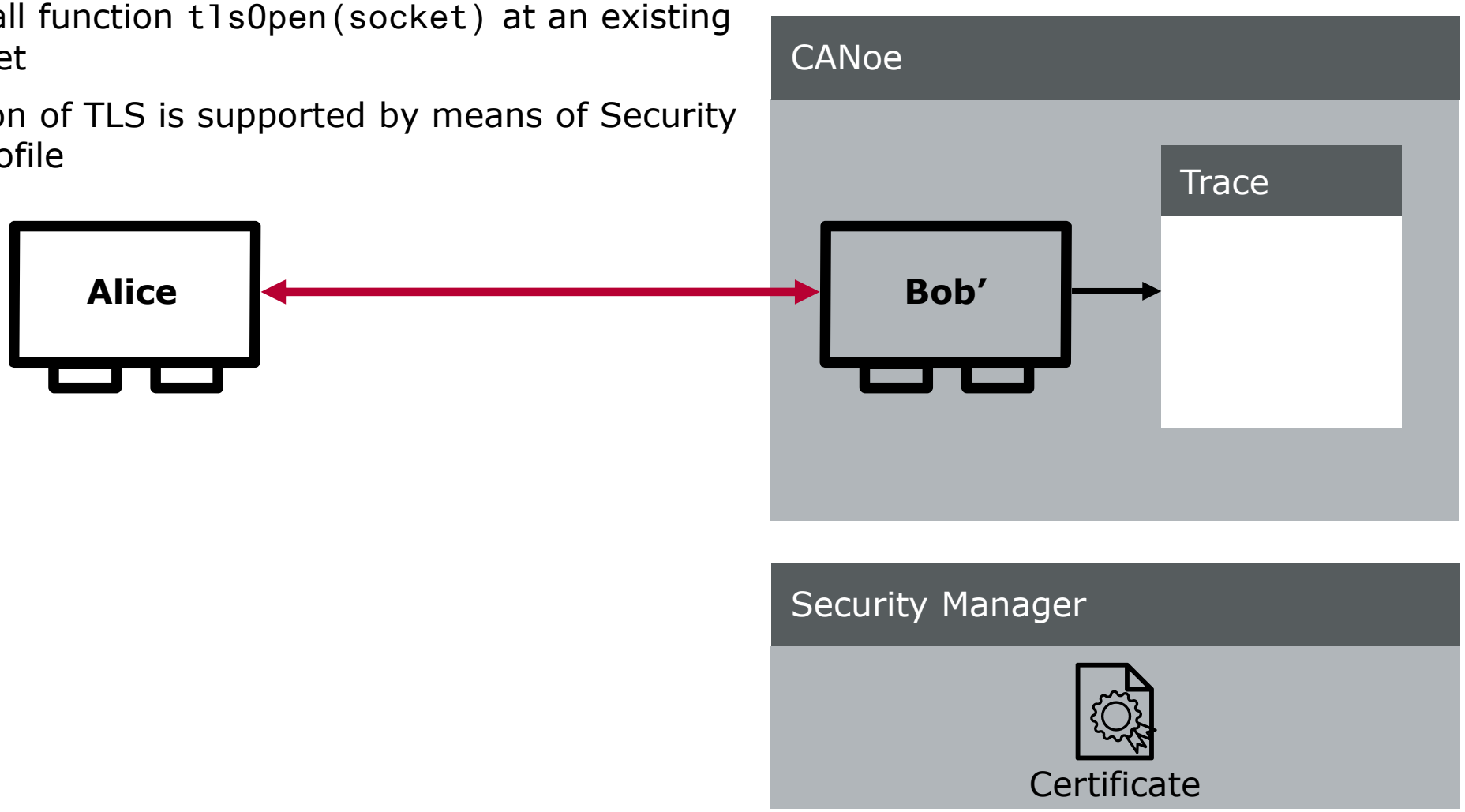
Time	Protocol	Name	Protocol Info	Source IP	Destination IP	Packet Length	Payloa
10.384036	sdp	SECC Discovery Request	TP version: 1	FE80::2	FF02::1	10	2
Security	TLS						
Transport Layer	TCP						
10.404043	ndp		Neighbor Solicitation	FE80::11	FF02::1:FF00:2	86	0
10.404049	ndp		Neighbor Advertisement	FE80::2	FE80::11	86	0
10.404057	sdp	SECC Discovery Response	TP version: 1	FE80::11	FE80::2	28	20
Security	TLS						
Transport Layer	TCP						
EVSE IP	FE80:0:0:0:0:0:0:11						
EVSE TCP Port	51111						
10.404063	tcp		CF96 -> C7A7 [SYN] Seq=32FC4660 Win=FFFF	FE80::2	FE80::11	78	0
10.404069	tcp		C7A7 -> CF96 [ACK, SYN] Seq=143BD5CD Ack=32FC4661 Win=FFFF	FE80::11	FE80::2	78	0
10.404075	tcp		CF96 -> C7A7 [ACK] Seq=32FC4661 Ack=143BD5CE Win=FFFF	FE80::2	FE80::11	74	0
10.404113	tls		TLS 1.0: Handshake (Client Hello)	FE80::2	FE80::11	472	398
10.404126	tls		TLS 1.2: Handshake (Server Hello)	FE80::11	FE80::2	170	96
10.404248	tls		TLS 1.2: Handshake	FE80::11	FE80::2	1514	1440
10.404253	tcp		CF96 -> C7A7 [ACK] Seq=32FC47EF Ack=143BDBCE Win=FD20	FE80::2	FE80::11	74	0
10.404323	tls		TLS 46.68: Type 0, Type 135	FE80::11	FE80::2	870	796
10.404340	tls		TLS 1.2: Handshake (Client Key Exchange)	FE80::2	FE80::11	217	143
10.500006	tcp		C7A7 -> CF96 [ACK] Seq=143BDEEA Ack=32FC487E Win=FFFF	FE80::11	FE80::2	74	0
10.500015	tls		TLS 1.2: Change Cipher Spec, Handshake	FE80::2	FE80::11	117	43
10.500022	tls		TLS 1.2: Change Cipher Spec	FE80::11	FE80::2	80	6
10.600006	tcp		CF96 -> C7A7 [ACK] Seq=32FC48A9 Ack=143BDEF0 Win=FFFF	FE80::2	FE80::11	74	0
10.600015	tls		TLS 1.2: Handshake	FE80::11	FE80::2	111	37
[illegible]	tcp		[illegible]	FE80::2	FE80::11	74	0
10.900028	tls		TLS 1.2: Application Data	FE80::2	FE80::11	171	97
11.000006	tcp		C7A7 -> CF96 [ACK] Seq=143BDF15 Ack=32FC490A Win=FFFF	FE80::11	FE80::2	74	0
11.200037	tls		TLS 1.2: Application Data	FE80::11	FE80::2	107	33
11.300006	tcp		CF96 -> C7A7 [ACK] Seq=32FC490A Ack=143BDF36 Win=FFFF	FE80::2	FE80::11	74	0

Agenda

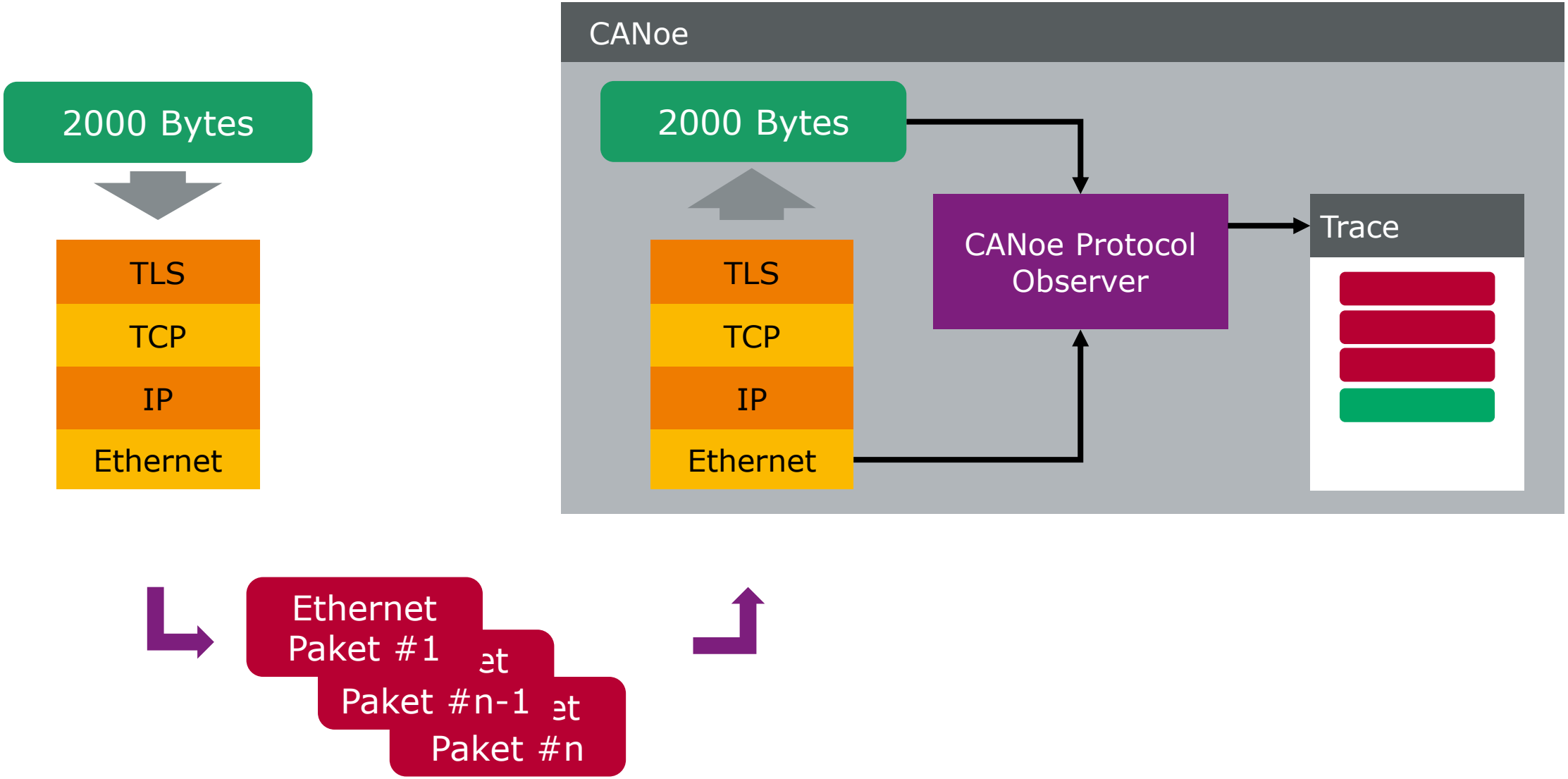
- Introduction
- Software Development Impact
- **End-Point Simulation**
- Not End-Point Simulation
- Conclusion

Stand in For Bob

- CANoe supports build in TLS layer for simulation
  - Simply call function `tls0pen(socket)` at an existing TCP socket
- Configuration of TLS is supported by means of Security Manager profile

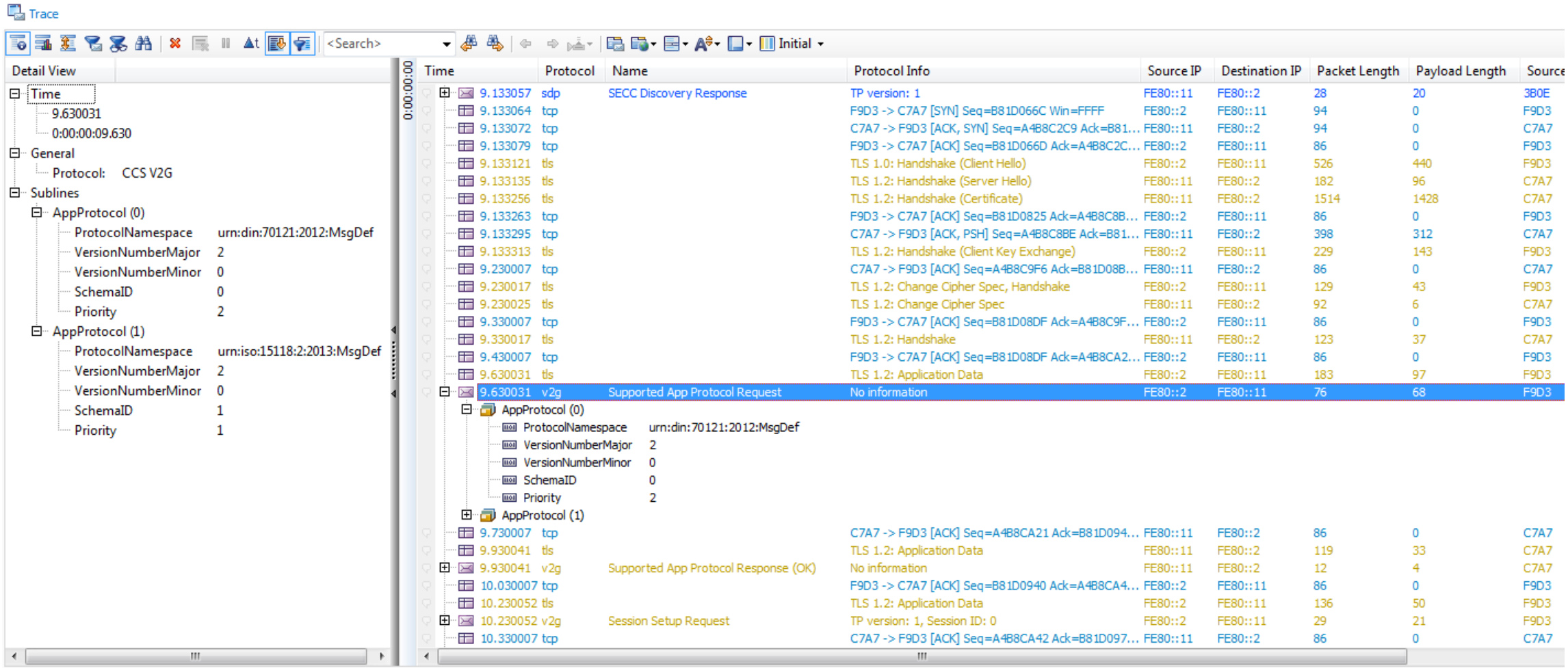


Analysis

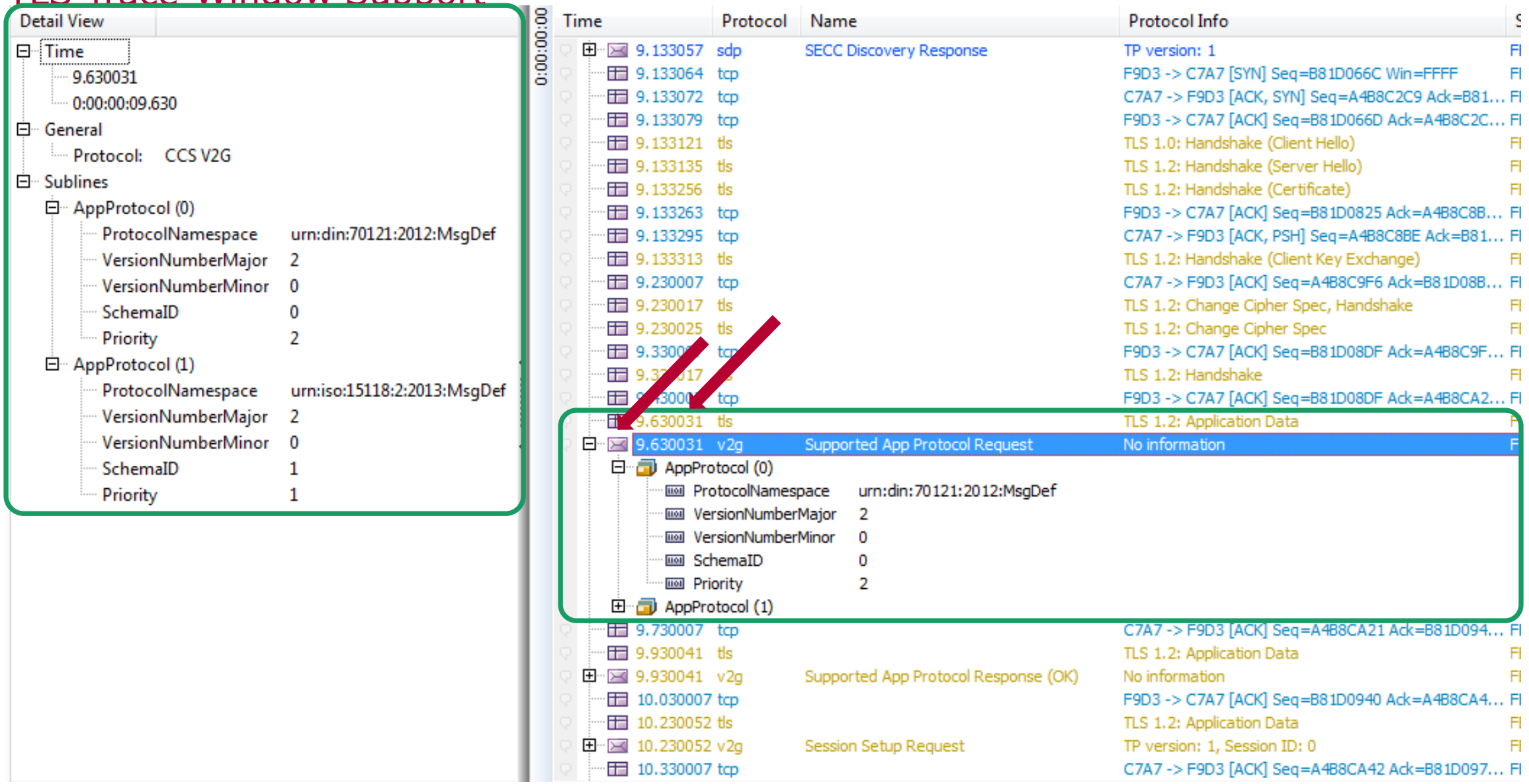


TLS Trace-Window Support

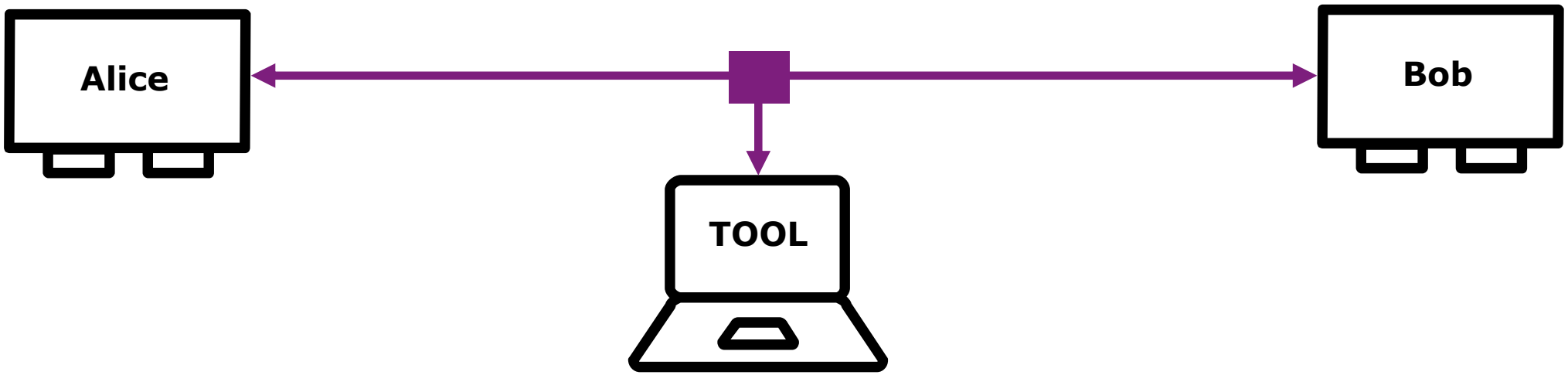
- Smart charge communication TLS secured but enhanced interpretation (hence CANoe 12)



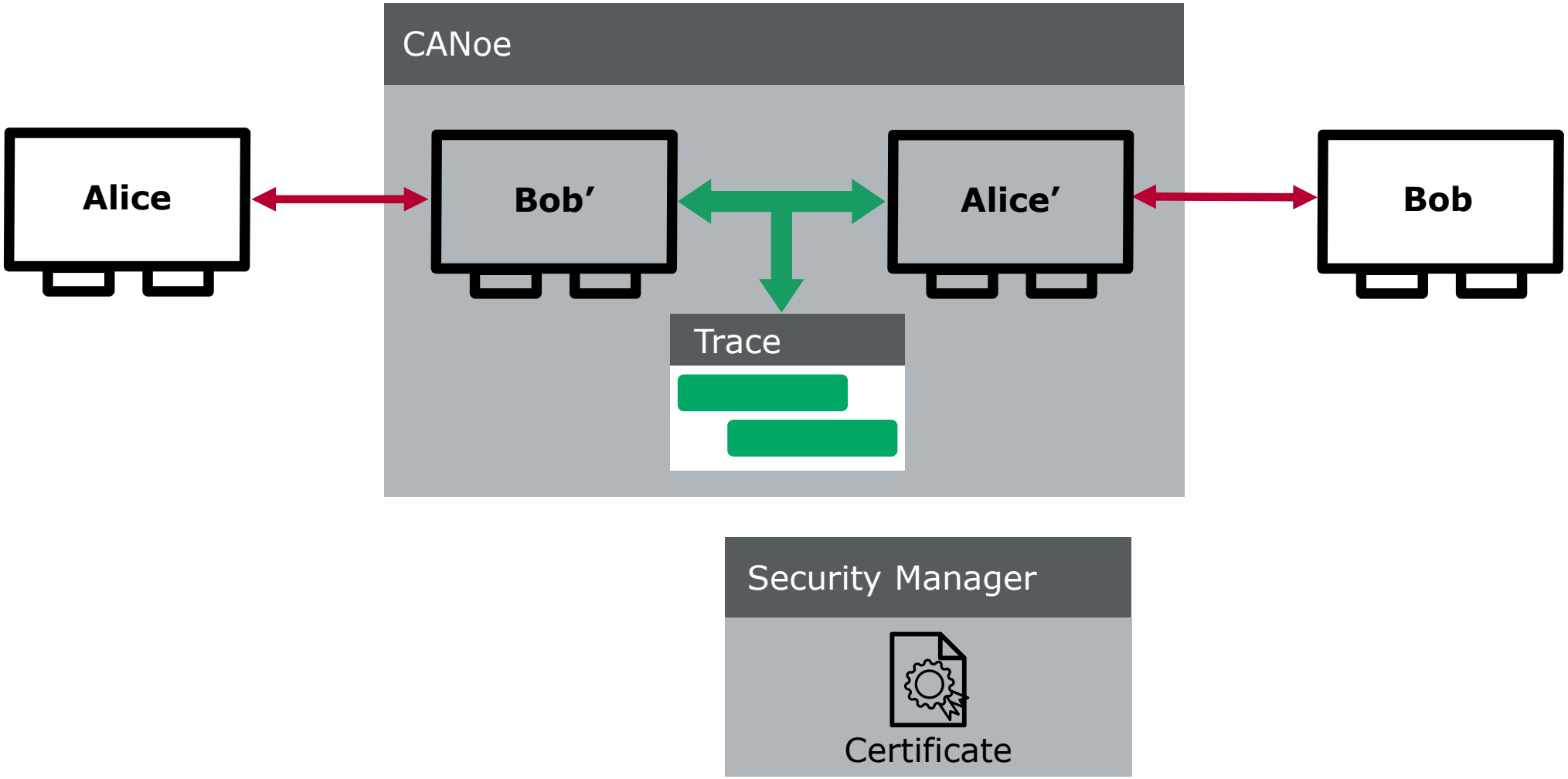
TLS Trace-Window Support



Friendly Reminder on Our Objective



Playing The Role of Alice And Bob



Playing The Role of Alice And Bob

Pro

- Issues on the application layer can be analyzed due to availability of decrypted messages
- Even messages/signals can be manipulated
- Security layer on real ECU does not need any modification
  - Works only if private certificates are shared with the tool
    - No private development certificates might be used, too
- Logging for post data analysis is less complicated

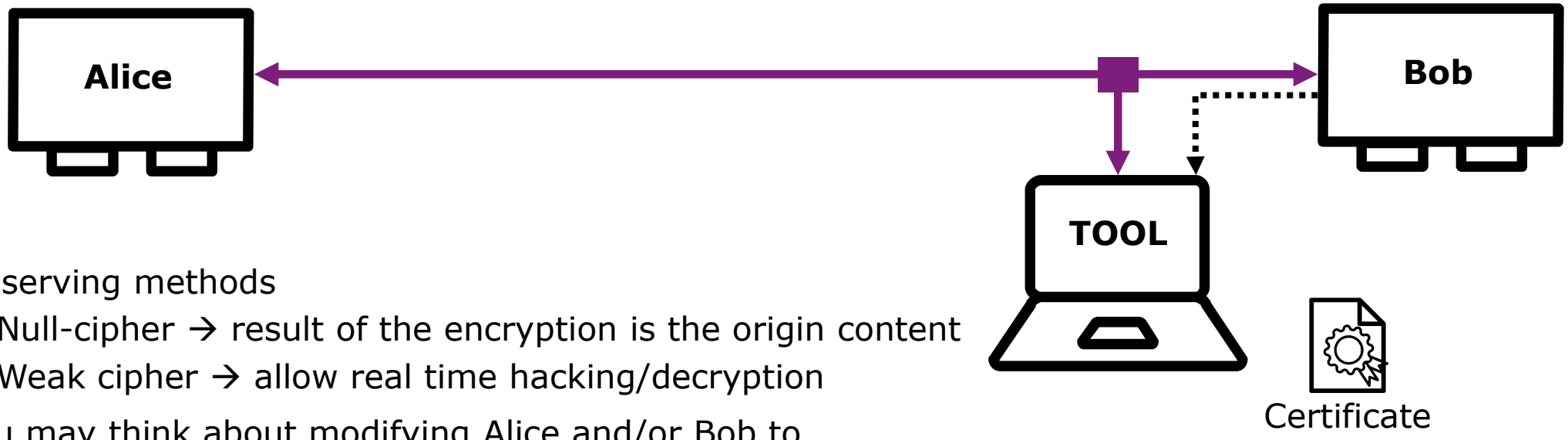
Con

- Communication layer is completely replaced by simulation
  - Issues on the communication layer may vanish when using the tool
  - Impact on timing and reaction time
- Also non TLS related communication is routed through the proxy
  - Hardware bypass might be an option for some non IP traffic
  - Partially routing of IP and related traffic may confuse the TCP/IP stack
    - Issues if Bob‘ and Bob are connected to the same network and run in parallel
    - E.g. DHCP, ARP

Agenda

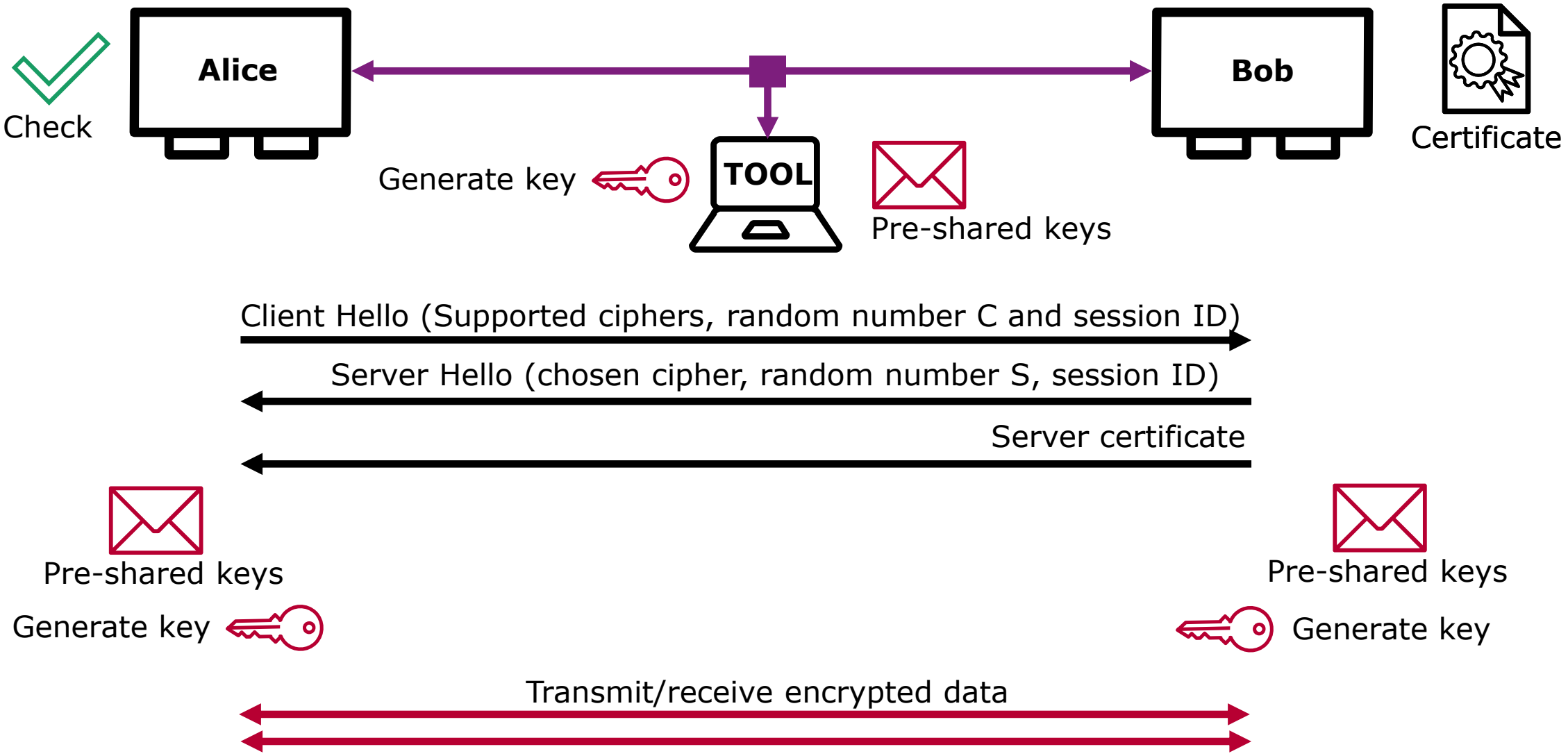
- Introduction
- Software Development Impact
- End-Point Simulation
- **Not End-Point Simulation**
- Conclusion

We Are Not an End-Point, Now What?



- Observing methods
  - Null-cipher → result of the encryption is the origin content
  - Weak cipher → allow real time hacking/decryption
- You may think about modifying Alice and/or Bob to
  - disable encryption at all
  - manipulate the crypto algorithm → weaker cryptography to allow real time hacking
  - provide a proprietary side channel for the tool → e.g. share master secret, use temporary keys
- Any modification you apply may weaken your cryptography
  - If such “debug” functionality can be enabled in the released product it may become an attack point for hackers, too
  - If the modification will be removed before release, potentially all performed tests are to be repeated

Consider to Use Standards But Weaker Security Mechanisms



Agenda

- Introduction
- Software Development Impact
- End-Point Simulation
- Not End-Point Simulation
- **Conclusion**

VECTOR

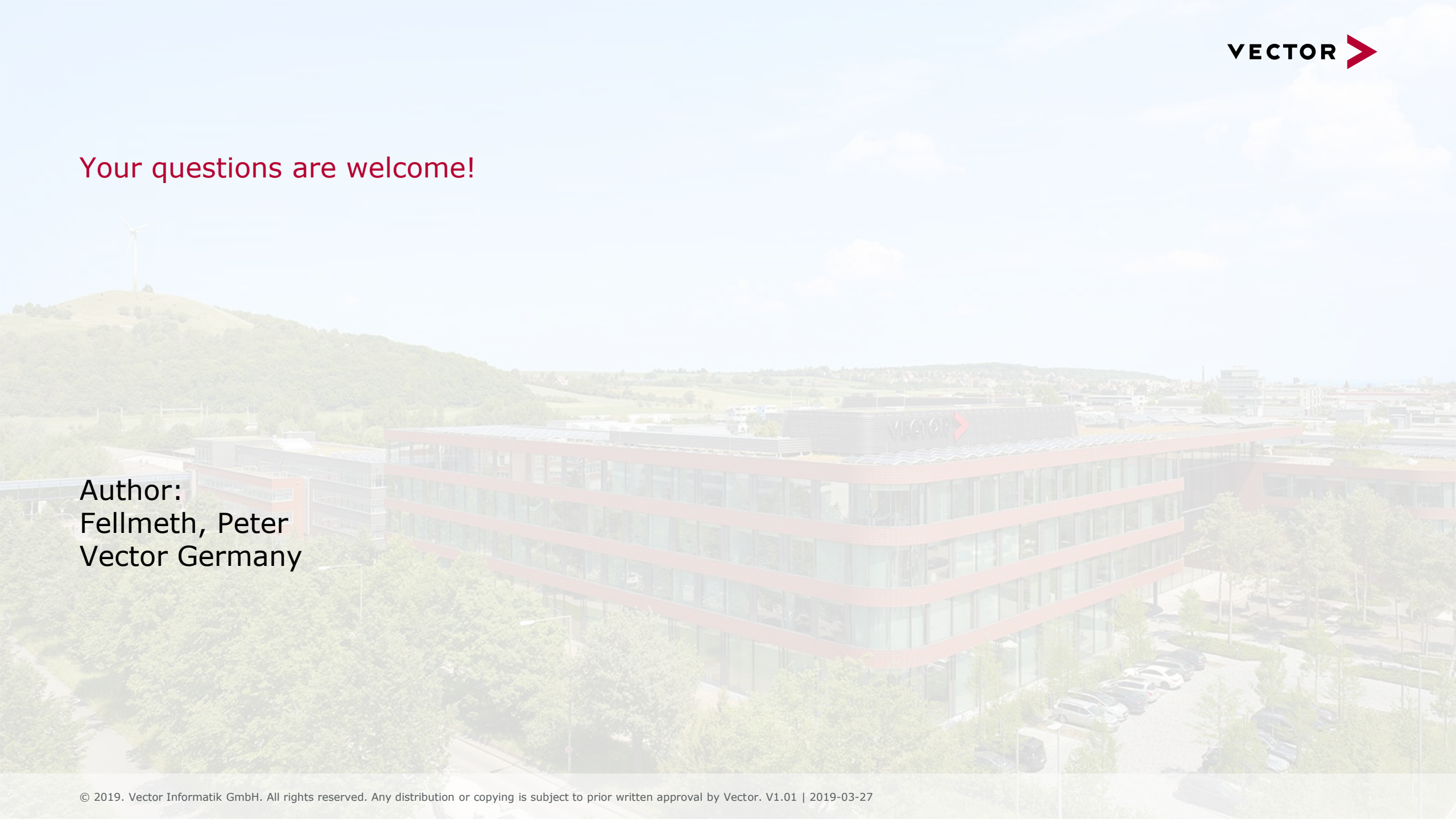
Security in The Development Environment

- Taking into account all discussed aspects, there is no single solution that fits all requirements
- The tools will not solve the problems. Tools can help to address and overcome individual issues
  - e.g. Security Manager can help to handle sensitive materials like certificates, keys to ease the distribution and make them available to users who are not trained to handle sensitive crypto material
- Make sure your measures during the development phase will not have a negative impact on the final product
  - Don’t leave backdoors open, revert any modification permanently and use temporary keys if possible
- Since most likely your security measures can be considered somehow proprietary don't forget to discuss this with your preferred tools supplier

Roadmap:

- CANoe 11.0 SP3 → supports TLS Stack
- CANoe 12.0 → supports “decryption” of TLS encrypted packets
- CANoe 12.0 SP → will support pre-shared keys
- An easy to use Man-in-the-middle (proxy) is being conceived for a future release

Your questions are welcome!



Author:
Fellmeth, Peter
Vector Germany