October 2019

Automotive Cybersecurity Engineering - Secure Flashing with HSM

F. Retailleau – C. Poulailleau

Delphi Technologies

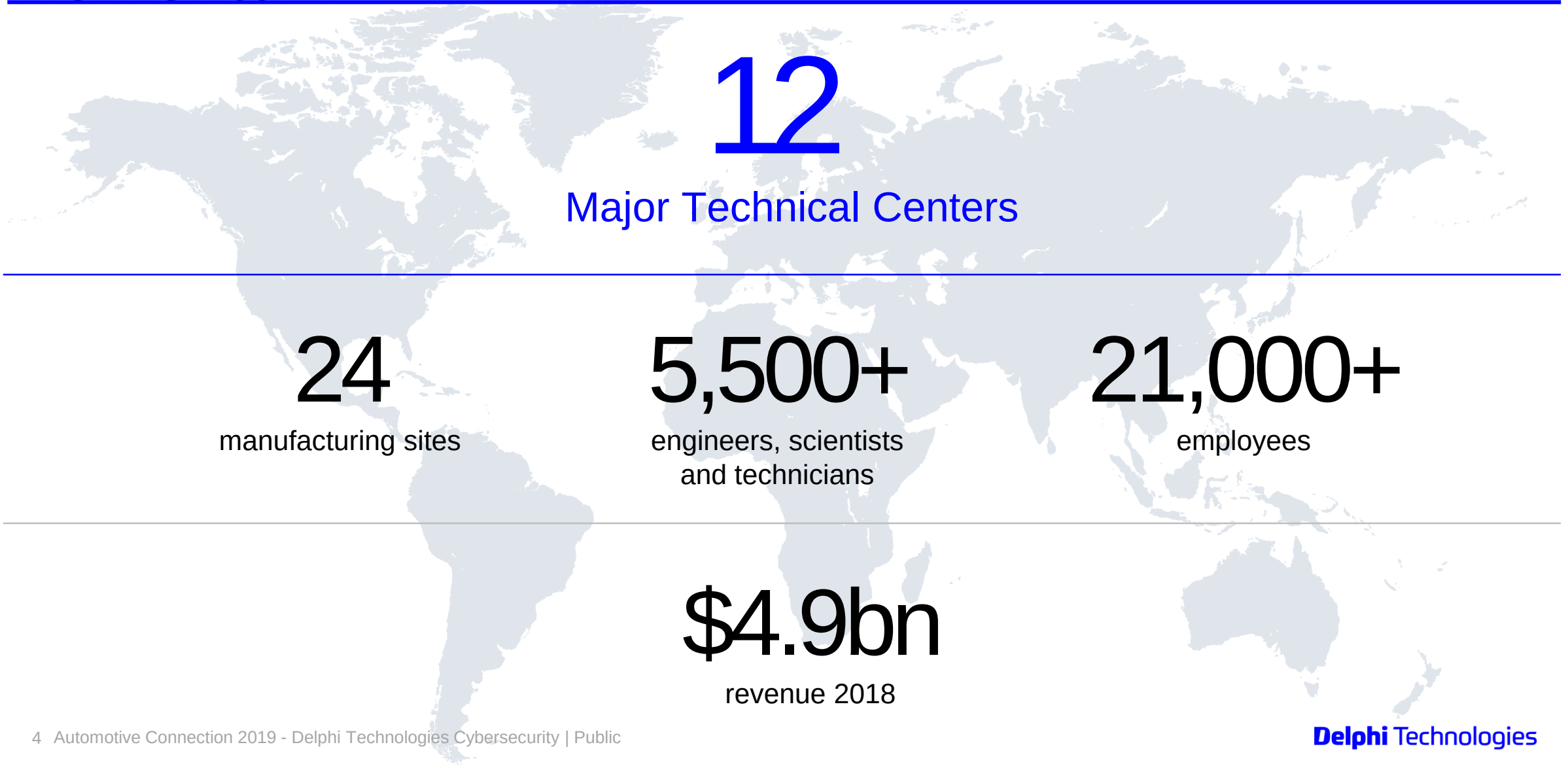
Agenda

1) Delphi Technologies
2) Cybersecurity Engineering with ISO/SAE 21434
3) Hardware Security Module in the Infineon Aurix TC3xx
4) Implementation of Secure Flashing using the HSM

1) Delphi Technologies

Delphi
Technologies

A globally diverse business with strong bookings momentum



12

Major Technical Centers

24

manufacturing sites

5,500+

engineers, scientists and technicians

21,000+

employees

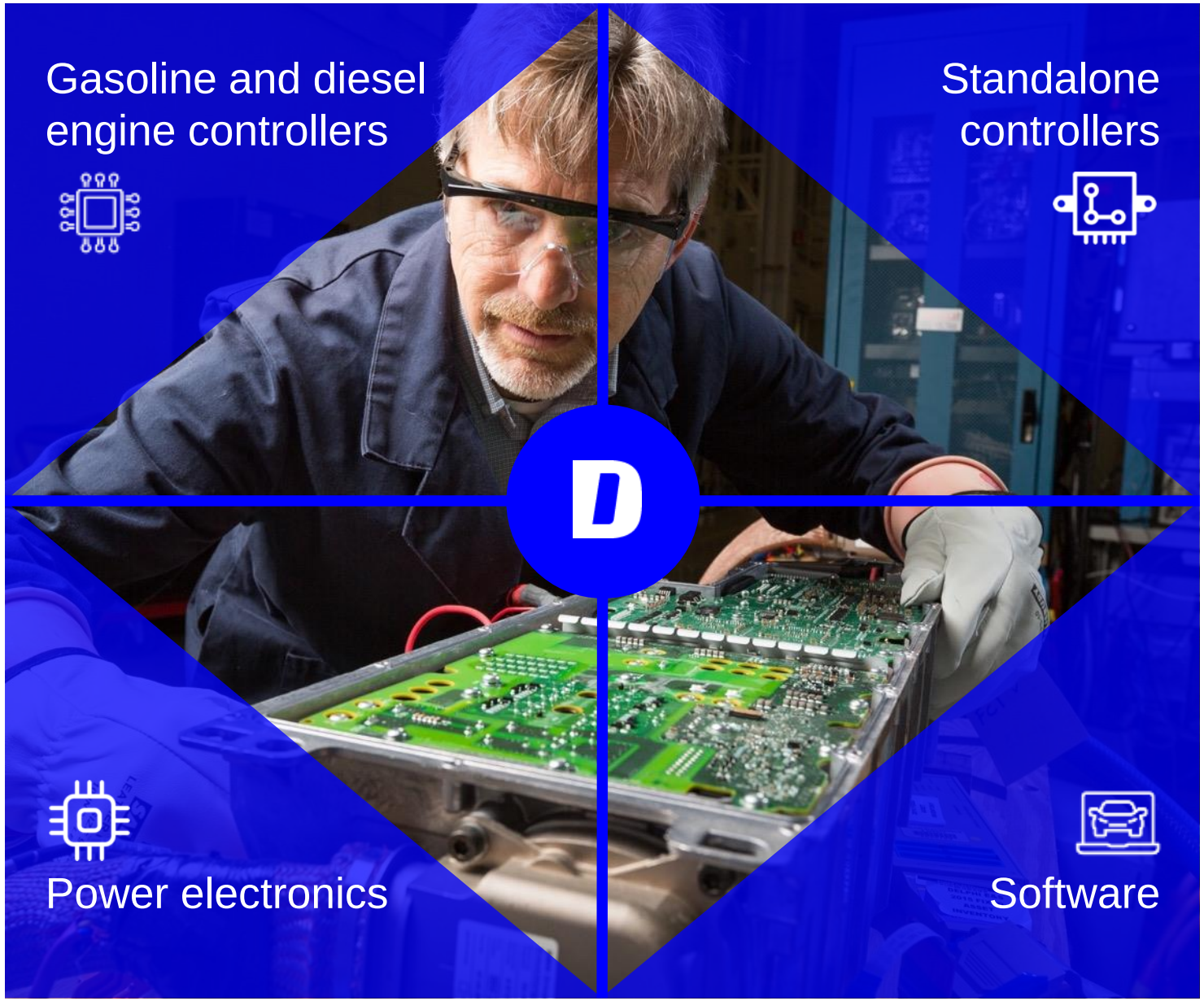
$4.9bn

revenue 2018

Electronics portfolio

Comprehensive offering of advanced systems, software and solutions.

Full suite of power electronics.



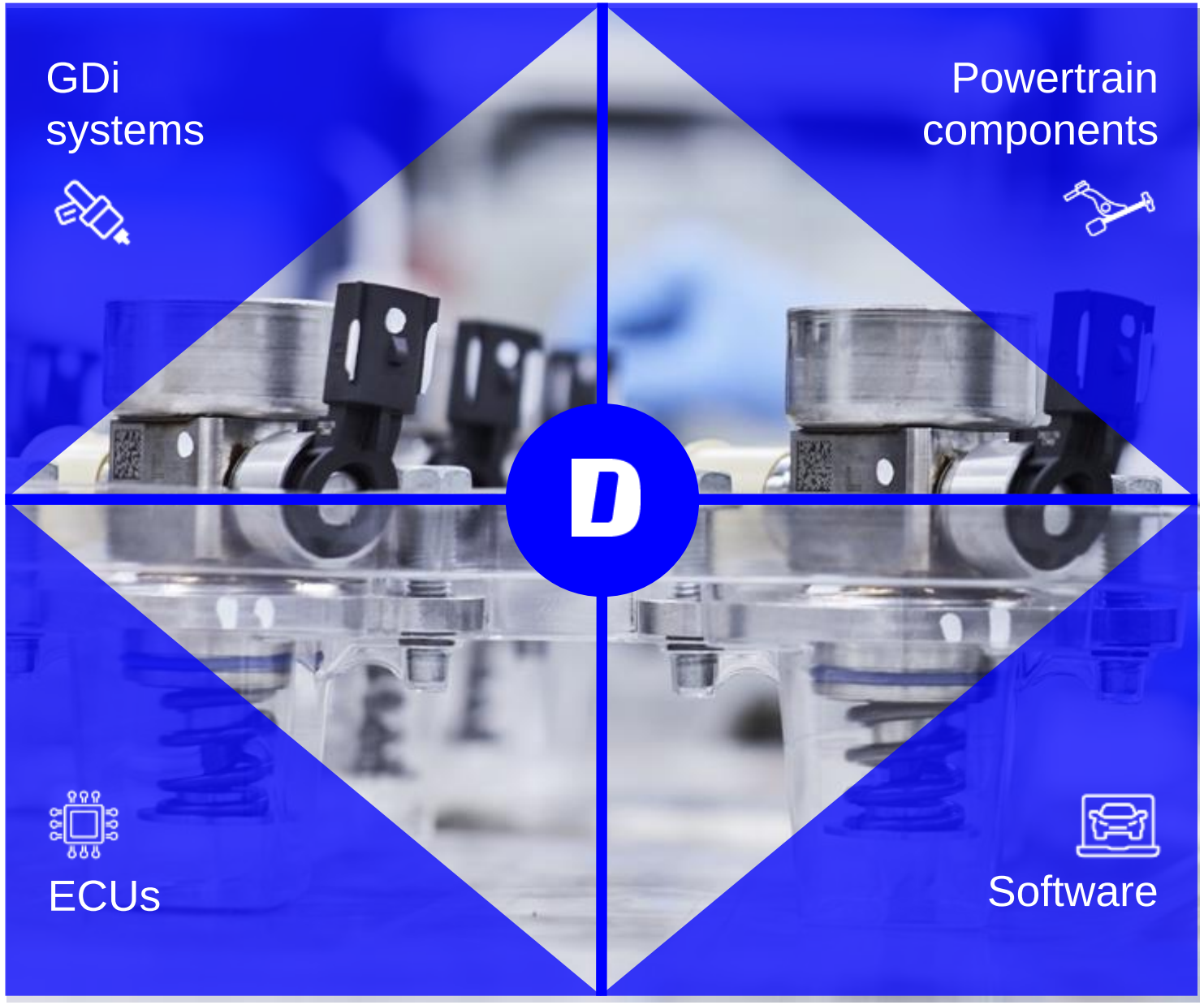
Cleaner. Better. Further.

Gasoline engine management systems

High-precision fuel delivery for low toxic emission solutions.

First to market with 350 Bar Gasoline Direct Injection (GDi) fuel system.

Cleaner. Better. Further.



Diesel engine management systems

Leverages investment for a broad range of LV and CV applications.

Flexible solutions for applications up to 18 litre engines.

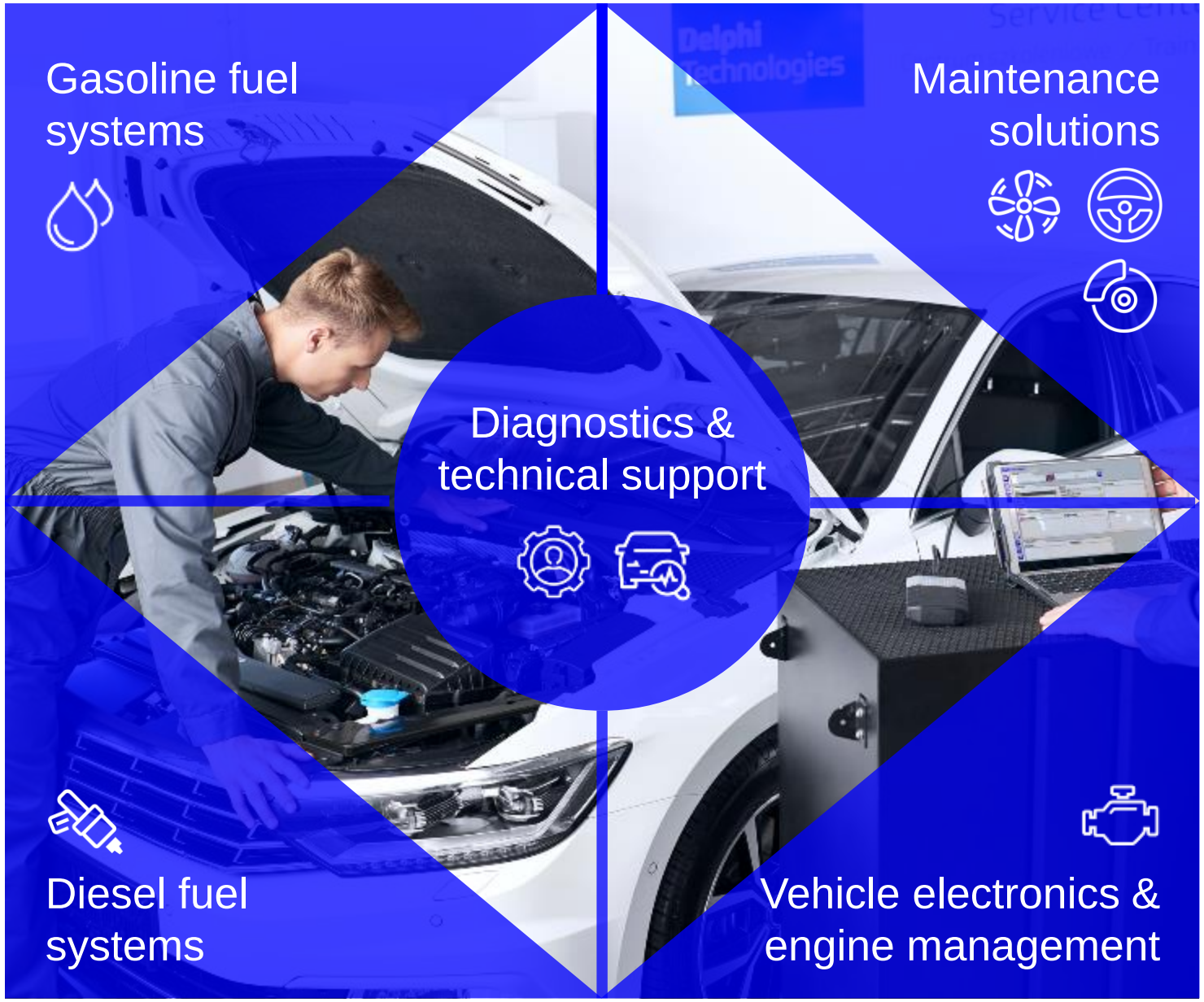
Cleaner. Better. Further.

Aftermarket

We are committed to ensuring vehicles drive as well as the day they were built, for the whole of their life. But we also know how a vehicle rides, and how a vehicle stops are just as important.

We like to call it:

‘Start. Go. Stop.’



Business units collaborating on integrated solutions

Electrification & Electronics

Supporting OEMs at each stage of their electrification journey with powerful and flexible automotive grade engine control modules and power electronics solutions.

Fuel Injection Systems

Developing advanced fuel injection systems to provide precise control of quantity and timing of fuel delivery to optimize combustion for passenger cars, on-and-off highway commercial vehicles.

Powertrain Products

Providing a wide range of engine and fuel handing components to help monitor, control and optimize powertrain efficiency in conventional and hybrid vehicles.

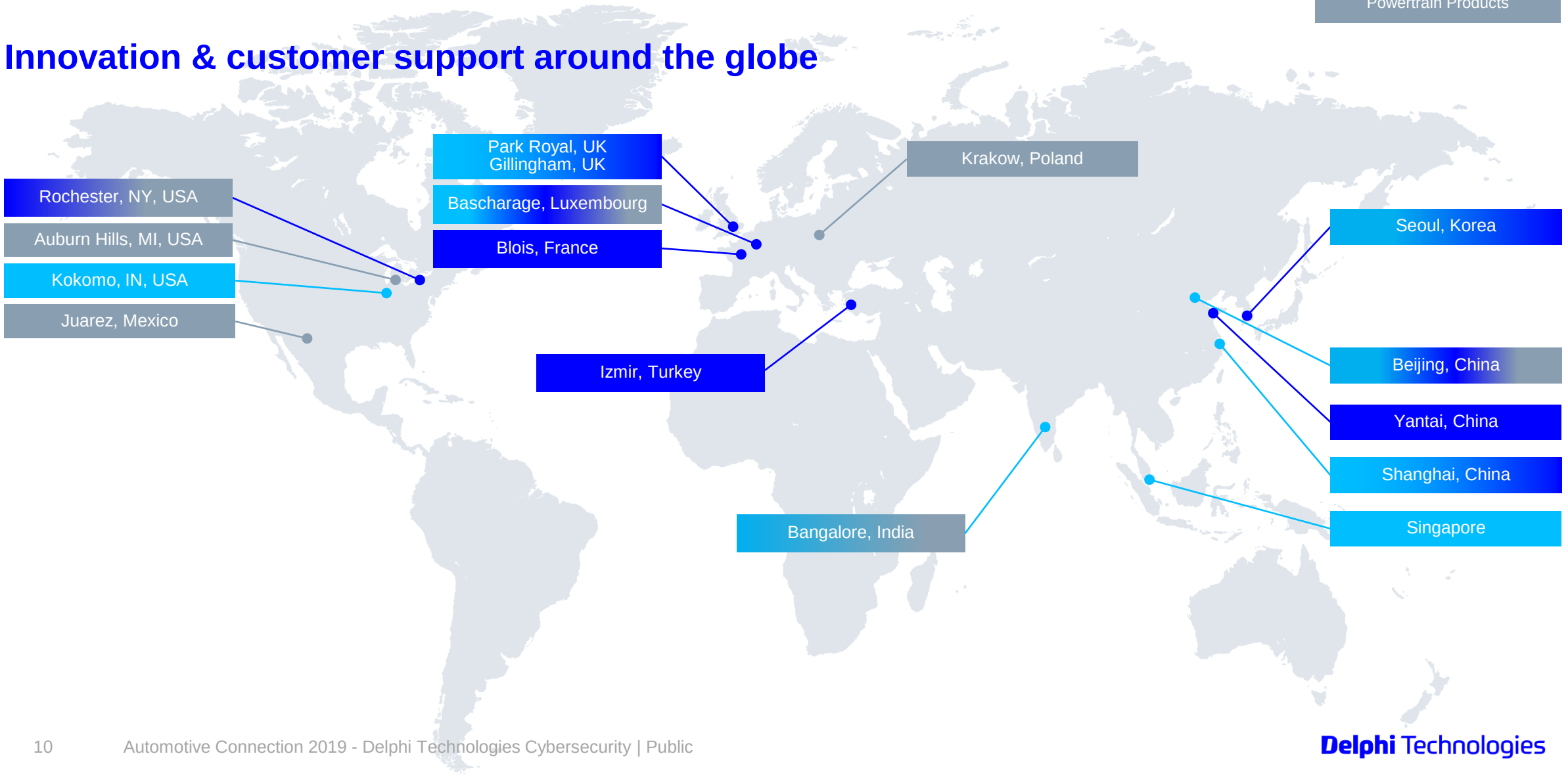
Aftermarket

Helping aftermarket customers to be a step ahead in servicing and maintaining sophisticated vehicle systems with leading service solutions.

Engineering footprint

Electrification & Electronics
Fuel Injection Systems
Powertrain Products

Innovation & customer support around the globe

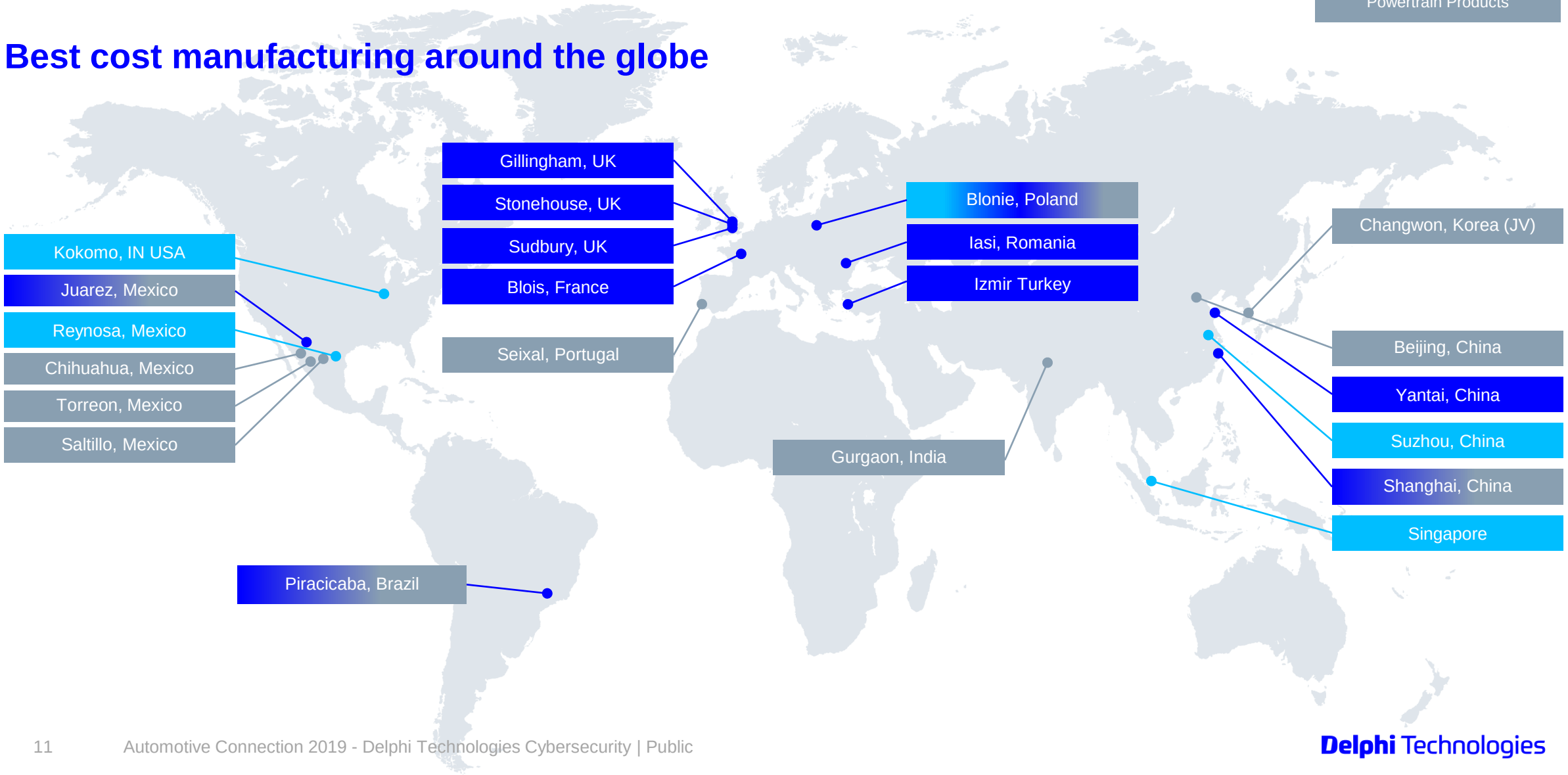


Delphi Technologies

Manufacturing footprint

Electrification & Electronics
Fuel Injection Systems
Powertrain Products

Best cost manufacturing around the globe



Delphi Technologies

2) Cybersecurity Engineering

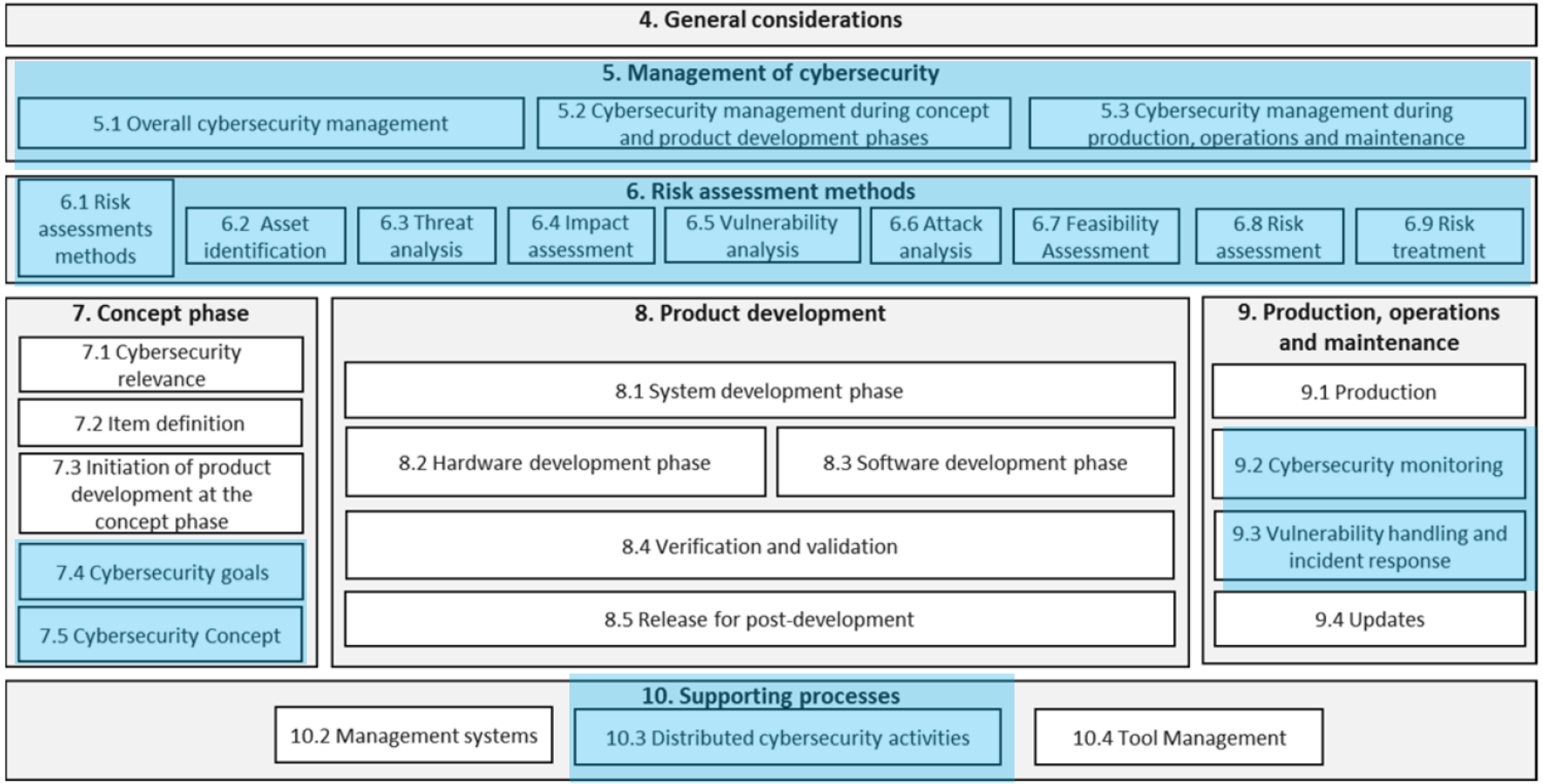
Delphi
Technologies

ISO/SAE 21434 - Introduction

SAE J3061 :	Cybersecurity Guidebook for Cyber-Physical vehicle Systems
ISO/SAE 21434 :	Road Vehicle – Cybersecurity Engineering *- Not yet approved -*
ISO 26262 :	Road Vehicle – Functional Safety

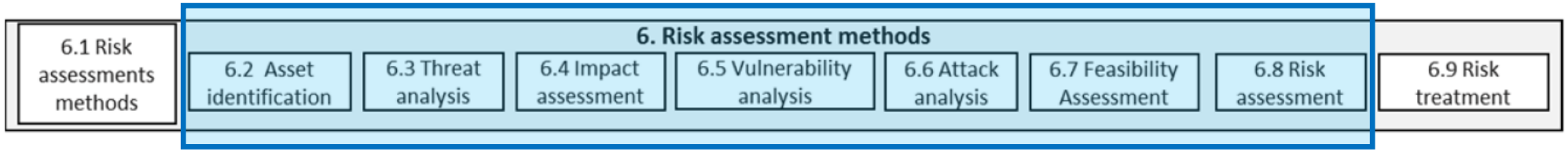
- It will be the first (joint) standard for Cybersecurity in Automotive
- Most companies are starting to understand and to apply it (will be an official standard in 2020)

ISO/SAE 21434 - Overview of structure



Delphi Technologies

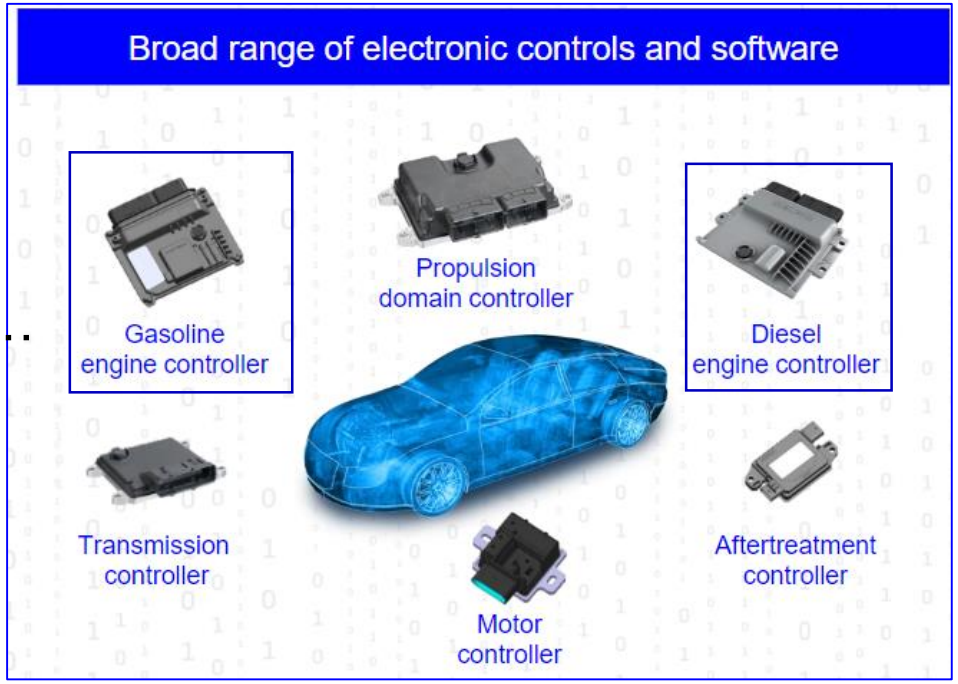
ISO/SAE 21434 - Threat Analysis and Risk Assessment (TARA)

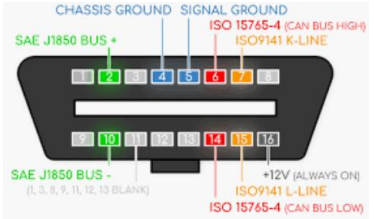


- Identify the project 'Assets' (things which have a value for project stakeholders)
- Identify potential 'Threats' which could break 'Security Objectives' of the Assets
- Based on the known 'Vulnerabilities' identify the potential 'Attack paths'
- Assess the 'Impact Level' (consequences of an Attack on the 'Security Objectives')
- Assess the 'Feasibility Level' (difficulty to realize the Attack)
- Finally Determine the 'Risk' for each Threat (Security Level)

Use Case – Vulnerability of Vehicle OBD-II Port for an ECU

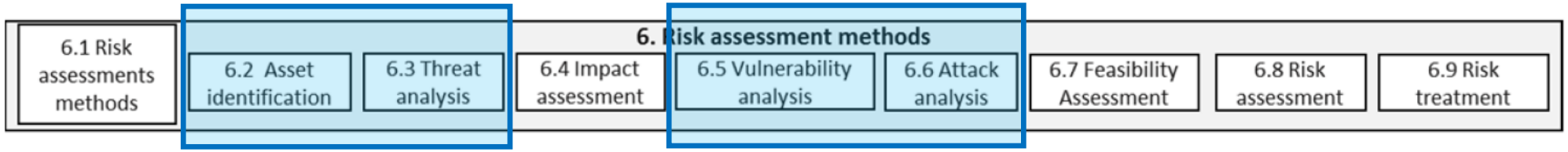
- **The Vehicle OBD-II connector provides access to the Vehicle Internal network (CAN)**
  - It's a potential **Attack vector** : Physical attack but not only….
- **It is a mandatory function**
  - For **Diagnostic** purpose (emission laws)
  - And for **Firmware updates** for the Aftermarket (cost)
- **One well known threat is :**
  - What about someone **reflashing illegitimate Firmware** ?





Delphi Technologies

TARA – Definition of the Threat



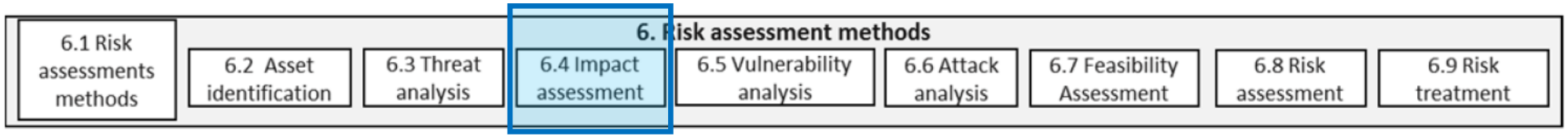
- Identify the project ‘Assets’ (things which have a value for project stakeholders)
- Identify potential ‘Threats’ which could break ‘Security Objectives’ of the Assets
- Based on the known ‘Vulnerabilities’ identify the potential ‘Attack paths’
- Assess the ‘Impact Level’ (consequences of an Attack on the ‘Security Objectives’)
- Assess the ‘Feasibility Level’ (difficulty to realize the Attack)
- Finally Determine the ‘Risk’ for each Threat (Security Level)

TARA – Threat definition

Use Case : illegitimate firmware reflashing

Violation of the Security Property :	*Authenticity (C.I.A.A.A.N)*
Of the Asset :	*ECU Firmware*
May lead to :	*Introduction of modified Firmware (with potential malicious code) in the ECU*
By using the (STRID**E**) Threat :	*Elevation of Privilege*
Due to the Vulnerability :	*Unsecure re-flashing process*
With the Attack :	*An attacker is flashing a illegitimate Firmware in the ECU by using the UDS reprogramming Services (via the vehicle OBD-II port)*
Causing impact on Security Objective :	*Operational - Safety. (S.F.O.P)*

TARA – Assess the Impact



- Identify the project 'Assets' (things which have a value for project stakeholders)
- Identify potential 'Threats' which could break 'Security Objectives' of the Assets
- Based on the known 'Vulnerabilities' identify the potential 'Attack paths'
- Assess the 'Impact Level' (consequences of an Attack on the 'Security Objectives')
- Assess the 'Feasibility Level' (difficulty to realize the Attack)
- Finally Determine the 'Risk' for each Threat (Security Level)

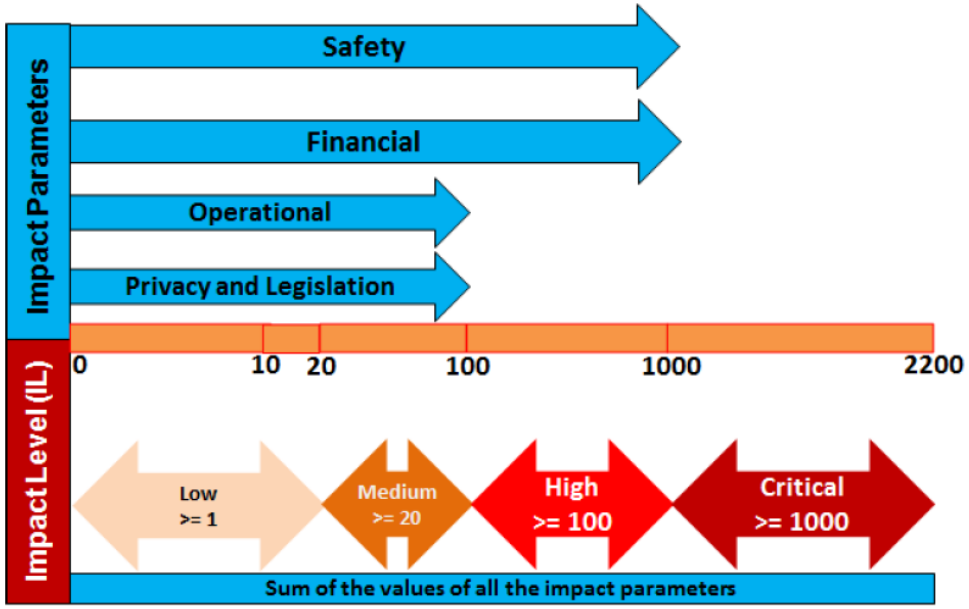
TARA - Impact Level (based on HEAVENS Security Model)

- **Four Security Objectives (parameters)** are used

- First two parameters with **high** weight
  - Safety (0-1000) → Injuries because of sudden Engine stop
  - Financial (0-1000) → Cost due to call-back and law-suits

- The other two with **lower** weight
  - Operational (0-100) → Car cannot start …
  - Privacy and Legislation (0-100) → Theft of personal data

  (usually Engine Control ECU do not manipulate personal data)



- **Impact level** for each threat = Sum of the four 'Impact Level' parameters values

TARA → Impact

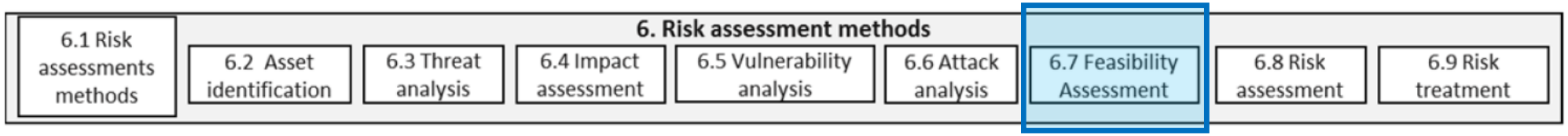
Use Case : illegitimate firmware reflashing

Impact Assessment (Impact Level - IL)					
Safety Impact	**Financial Impact**	**Operational Impact**	**Privacy and Legislation Impact**	**Impact Value**	**Impact Level**
Light to moderate (10)	Low (10)	High (100)	Low (1)	121	High (3)
[FR] Low probablitity that attacker's aim is to kill the driver by reflashing it's own vehicle	[FR] If the 'non authentic' Firmware and a tuto are made available on the Web, it can lead to some limited financial damages for Delphi Technologies (may be the attacker will be able to activate option fro free or to increase vehicle power ...)	[FR] Significant modifications of the Vehicle/Engine behaviour maybe introduced ...	[FR] Potentially violation of the Emission regulation laws (by changing the Injection behaviour)		

0	No Impact (0)
[1-19]	Low (1)
[20-99]	Medium (2)
[100-999]	High (3)
>999	Critical (4)

Brainstorming with the Project Design team

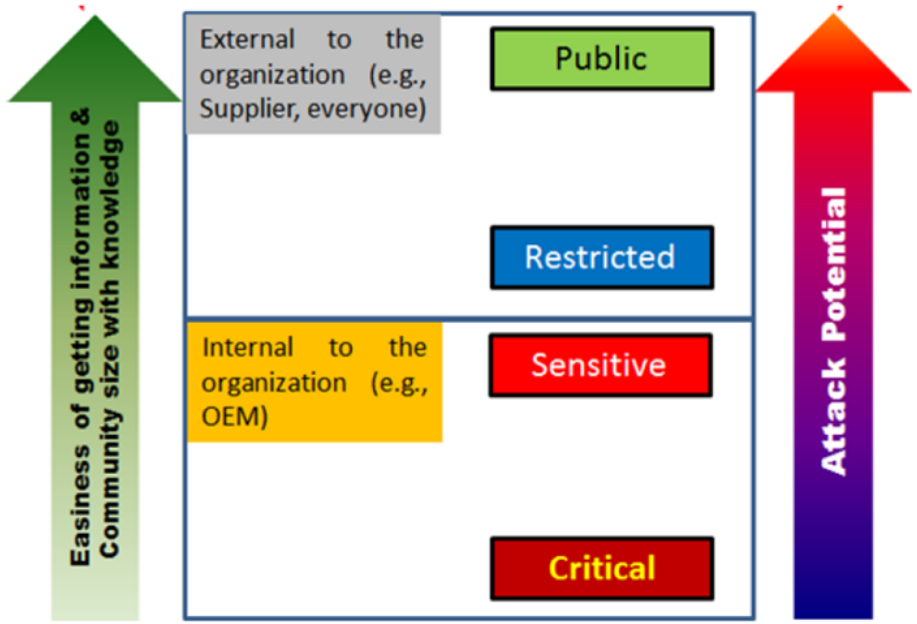
TARA – Assess the Feasibility



- Identify the project 'Assets' (things which have a value for project stakeholders)
- Identify potential 'Threats' which could break 'Security Objectives' of the Assets
- Based on the known 'Vulnerabilities' identify the potential 'Attack paths'
- Assess the 'Impact Level' (consequences of an Attack on the 'Security Objectives')
- Assess the 'Feasibility Level' (difficulty to realize the Attack)
- Finally Determine the 'Risk' for each Threat (Security Level)

TARA - Feasibility (based on HEAVENS Security Model)

- **Four Feasibility parameters** are used **(**with same weight)
  - Expertise (0-100) → Layman, Proficient, Expert, Multiple exp
  - Knowledge about TOE (0-100) → Public, Restricted, Sensitive, Critical
  - Opportunity (0-100) → Critical, High, Medium, Low
  - Equipment (0-100) → Standard, Specialized, Bespoke, Multiple bespoke
- **Feasibility level** = Sum of the four Feasibility parameters



Note : in 'Initial' assessment, **conservative rating** will be used

TARA - Feasibility

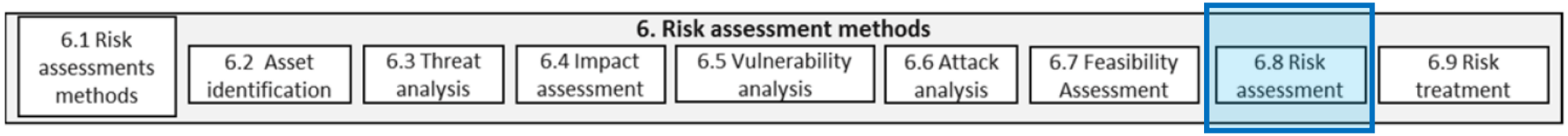
Use Case : illegitimate firmware reflashing

Threat Feasibility Assessment (Threat Level - TL)					
Expertise	**Knowledge about TOE**	**Window of opportunity**	**Equipment**	**Threat Value**	**Threat Level**
Expert (2)	Sensitive (2)	Critical (0)	Standard (0)	4	Medium (2)
[FR] The Attacker must be able to create a functional SW of his own (at least one Attacker, but not the Attackers who will re-use it ...). Real SW expertise is necessary.	[FR] Sensitive information concerning the HW mapping (Flash, RAM, ...) and internal components is necessary to build one Firmware which will be executable in the ECU without crashing	[FR] Vehicle owner can access its Vehicle OBD2 port (to re-flash) without any constraint	[FR] Commercial OBD Flashing tool are easily available		

>9	No Impact (0)
[7-9]	Low (1)
[4-6]	Medium (2)
[2-3]	High (3)
[0,1]	Critical (4)

Brainstorming with the Project Design team

TARA – Determine the Risk (Security Level)



- Identify the project 'Assets' (things which have a value for project stakeholders)
- Identify potential 'Threats' which could break 'Security Objectives' of the Assets
- Based on the known 'Vulnerabilities' identify the potential 'Attack paths'
- Assess the 'Impact Level' (consequences of an Attack on the 'Security Objectives')
- Assess the 'Feasibility Level' (difficulty to realize the Attack)
- Finally Determine the 'Risk' for each Threat (Security Level)

TARA - Security Level (Risk)

- Risk = Security Level : rating from **QM** (no security risk) to **Critical** security risk

- if Threat Impact and Threat Level are High
  → We have to manage a High priority risk

- If Threat Impact and Threat Level are Low
  → We have to manage a Low priority risk
  (or maybe no need to take it into account)

Security Level (SL)	Impact Level (IL)					
Threat Level (TL)		0	1	2	3	4
	0	QM	QM	QM	QM	Low
	1	QM	Low	Low	Low	Medium
	2	QM	Low	Medium	Medium	High
	3	QM	Low	Medium	High	High
	4	Low	Medium	High	High	Critical

Illegitimate firmware reflashing => Security Level (Risk) = **Medium**

ISO/SAE 21434 – Other CS Engineering activities (brief)

- TARA : for each potential Threat, **Assess the Risk** (Security Level) - Medium -
- For 'Not-acceptable' threats propose a **Treatment** (**Mitigate**-Transfer-Remove-Retain) - Mitigate -
- For threats to 'mitigate' define the **Cybersecurity Goals** (High Level CS requirements)

  - Only legitimate Delphi Technologies firmware should be reflashed -

- Determine the **Cybersecurity Assurance Level** (CAL) to tailor the Cybersecurity activities - CAL 3 -
- Use the **Cybersecurity Goals** (Component) and the specific **customer cybersecurity requirements** (System) to propose a **Cybersecurity Concept**

  - Secure Flashing -

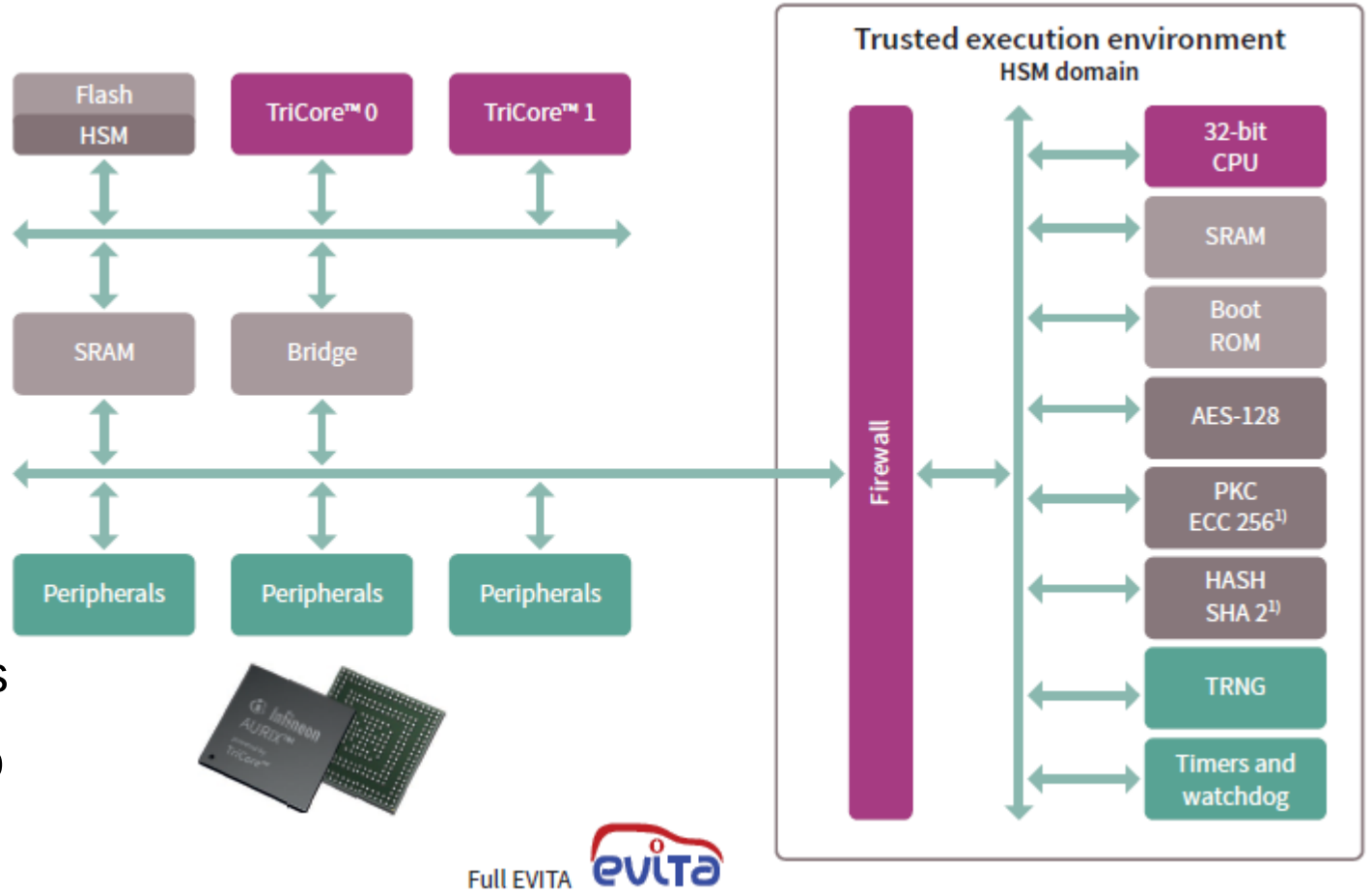
- And then System, Hardware, Software development activities …

3) Microcontroller Infineon Aurix TC3xx with HSM+ module

Delphi Technologies

Infineon Aurix TC3xx – HSM+

- Concept of **Trusted Environment**
- Host Side
  - Four 32bit Cores 300 MHz
  - Including Two lockstep cores
- HSM Side (**HW Security Module**)
  - One 32bit Core with its own resources
  - With hardware accelerators for Crypto
- Isolation by means of a Firewall



Delphi Technologies

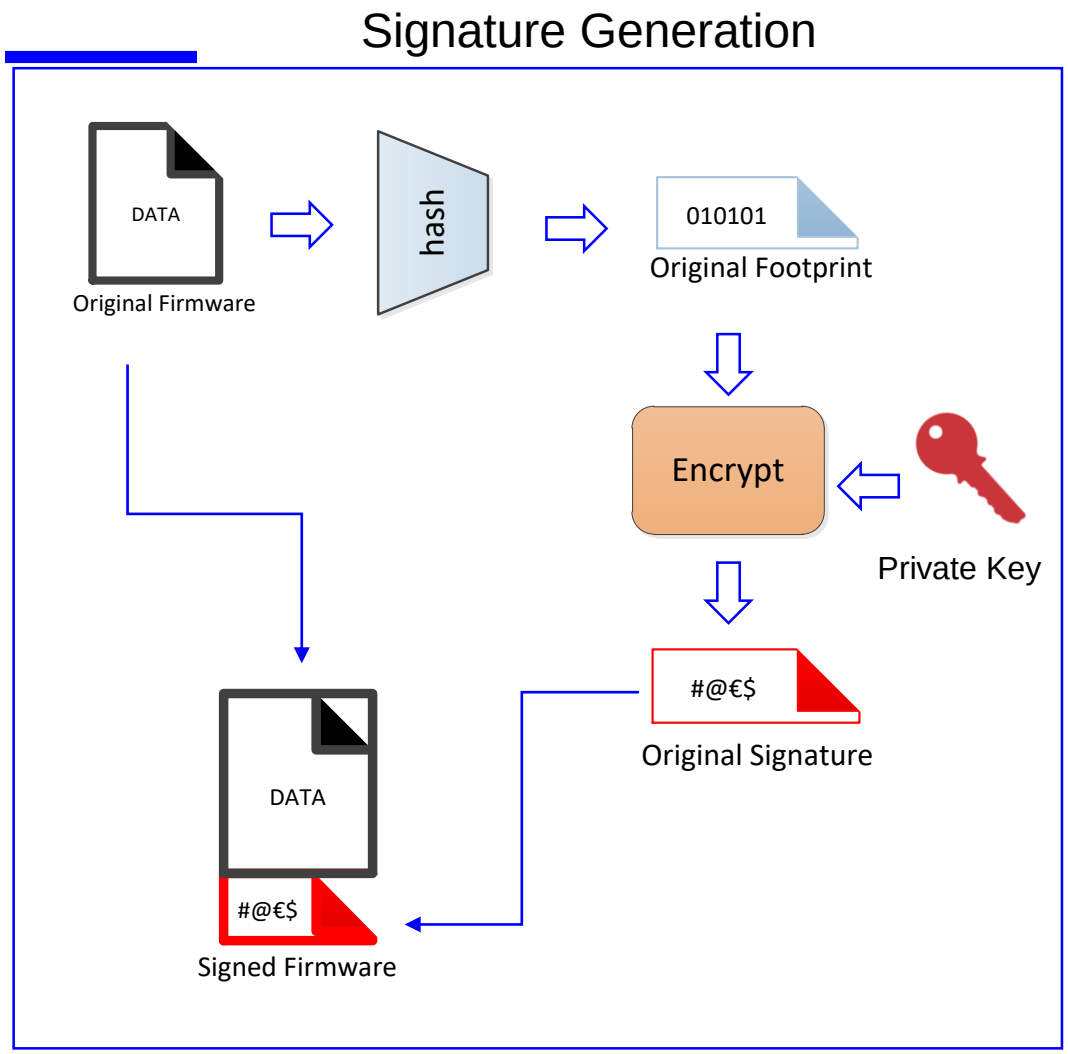
4) Secure Flashing with HSM

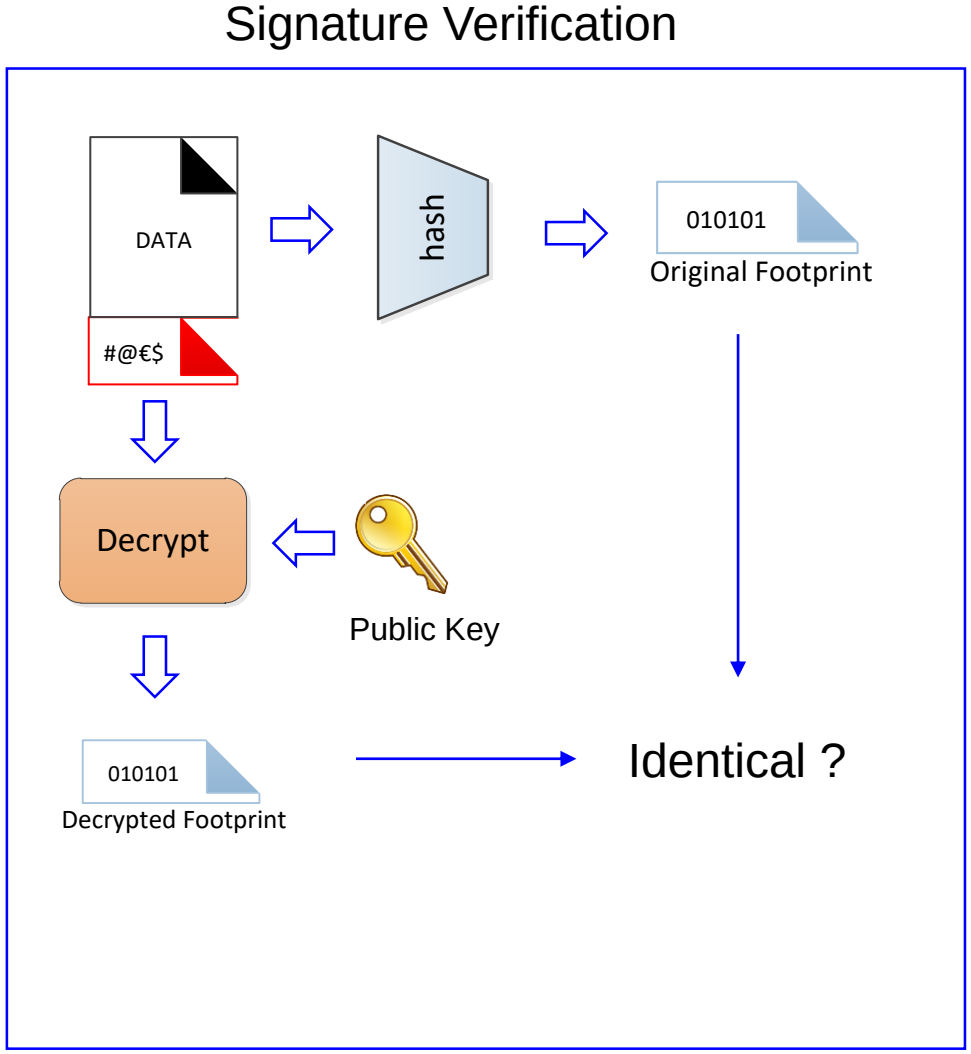
Delphi
Technologies

Concept - Secure Flashing

- It's a Cybersecurity Control developed to mitigate the 'illegitimate firmware reflashing' threat
- It's based on the Digital Signature cryptographic mechanism
  - Private Key : used at *Delphi Technologies Engineering* to sign the Firmware
  - Public Key : downloaded in the ECU (HSM) at *Delphi Technologies Manufacturing*
  - **Signed Firmware** : provided to *Customer Manufacturing* (or Aftermarket)
- Only Firmware Signed by Delphi Technologies is accepted during the Flashing process (via OBD-II)
- HSM trust anchor of the Aurix TC3xx microcontroller is the key element used to implement secure parts of this concept

Digital Signature - Trick is relying on the Private Key





Digital Signature – Security properties covered

Digital Signature covers

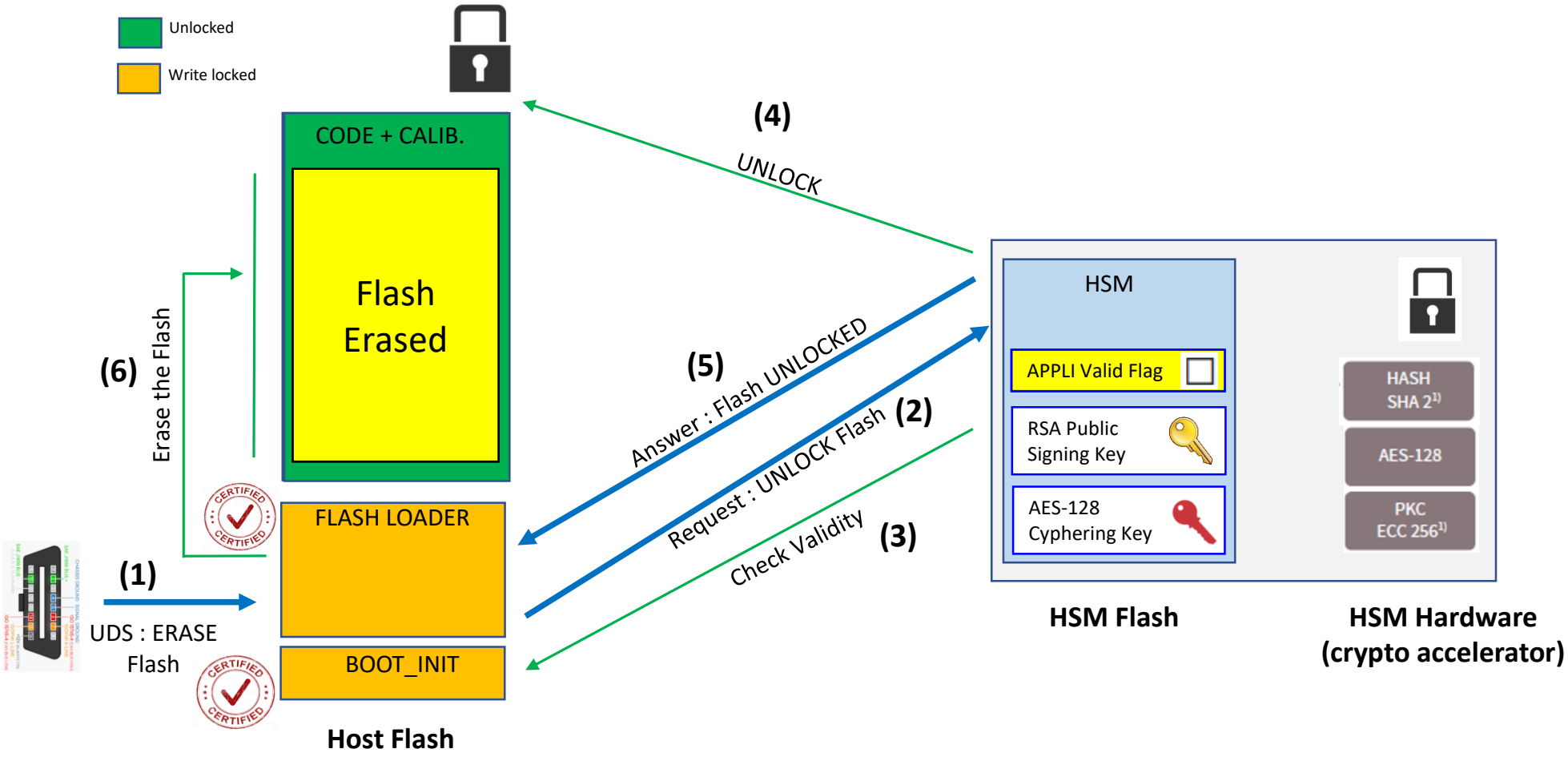
- **I**ntegrity → the received Data have not been altered or tampered with
- **A**uthentication → Identity of the sender is known
- **N**on-repudiation → The sender can't deny that he sent these Data

Digital Signature does not cover

- **C**onfidentiality → No one can read the data except the intended receiver

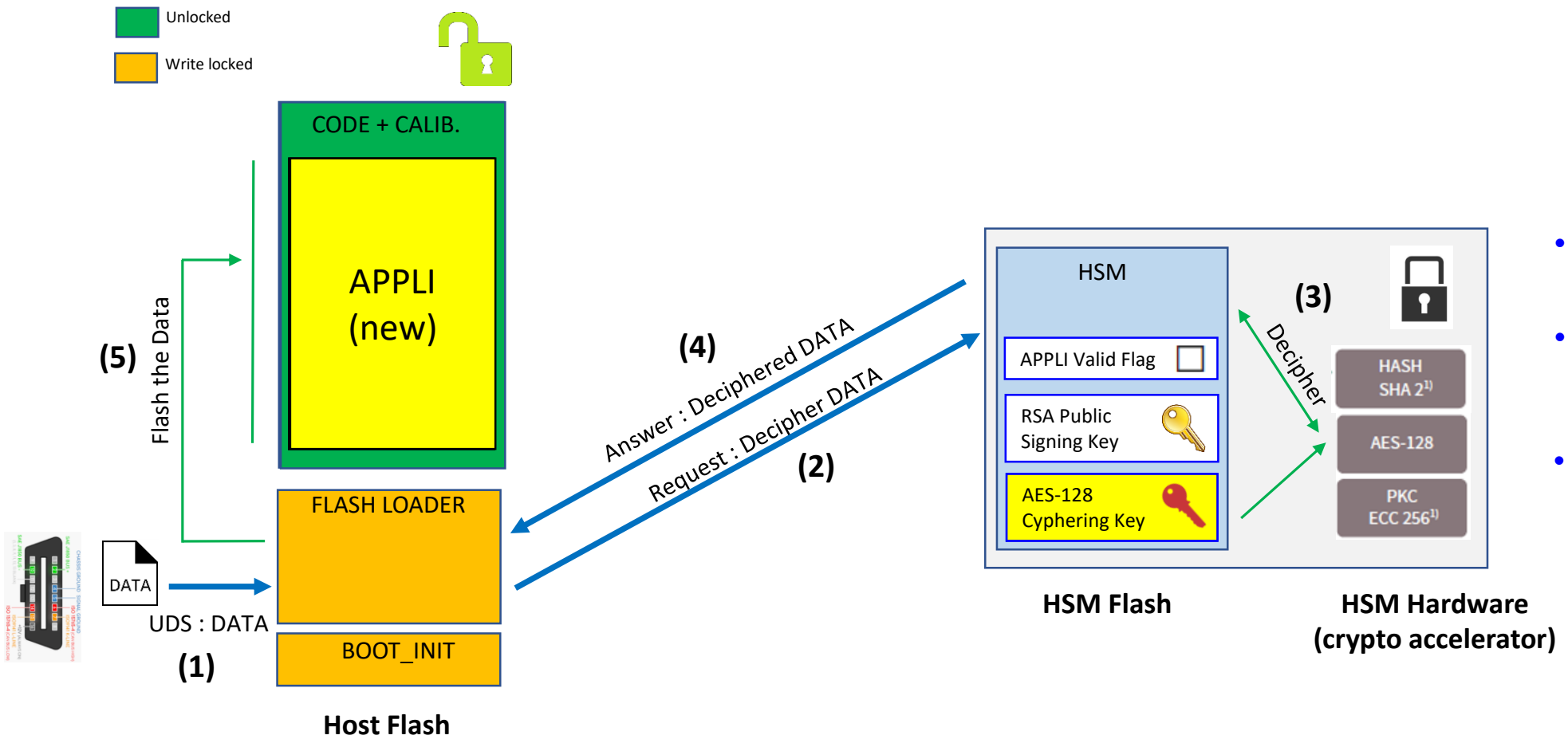
=> To cover this last property, Firmware will also be Cyphered (symmetric AES-128)
(after being compressed to reduce downloading time …)

Secure Flashing (simplified) - Erase the Flash



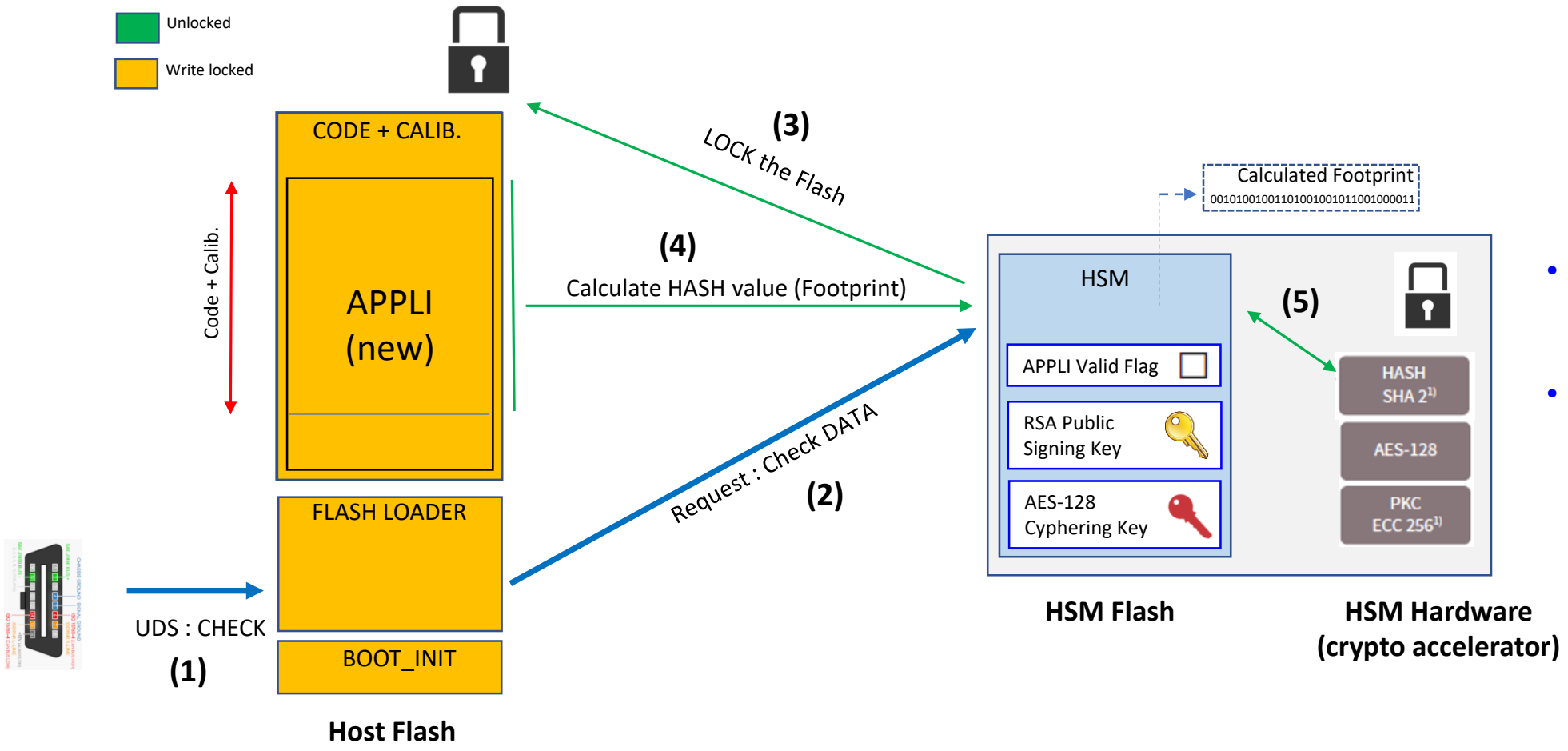
- Flash is locked by default
- Chain of Trust
- Flash is unlocked by the HSM

Secure Flashing (simplified) - Download the new Firmware



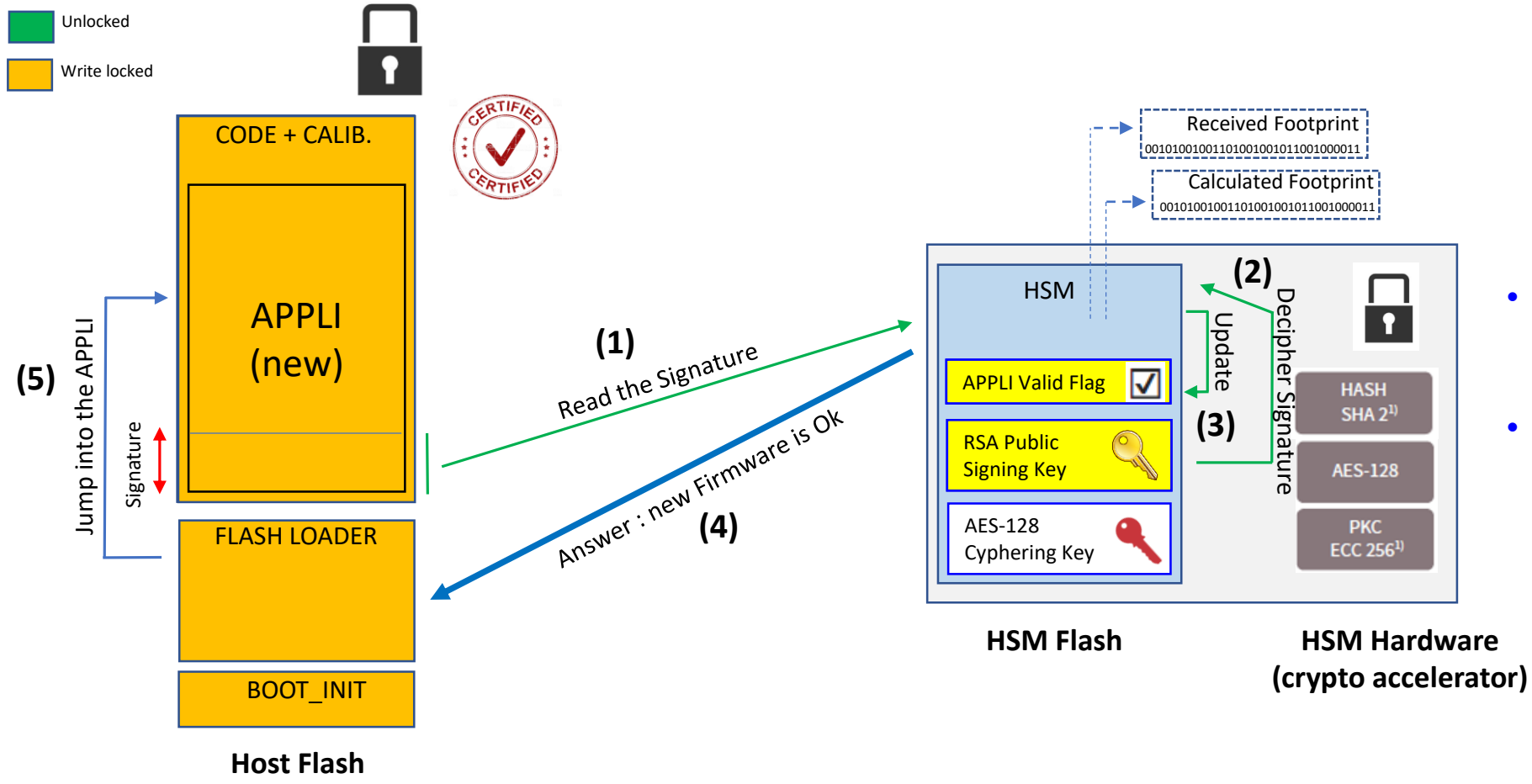
- UDS DATA are cyphered
- UDS DATA are deciphered by AES-128
- AES-128 key is stored in secure area

Secure Flashing (simplified) - Calculate the Footprint



- Flash is (re) locked by the HSM
- Hash is computed on the locked flash

Secure Flashing (simplified) - Validate the Firmware



- Signature is deciphered by the HSM
- Application is valid only if both footprints are matching

Aurix HSM usage for Secure functions

- **Security functions** performed by the Infineon **Aurix TC3xx HSM** for the **Secure Flashing** concept
  - Independent control of **Boot_Init** and **Loader validity**
  - Secure **Locking-Unlocking** of the Flash
  - On the Fly **deciphering** of the cyphered Firmware (AES-128 Hardware)
  - Verification of the **validity** of the new received Firmware (SHA-2, RSA Digital Signature)

- These capabilities are also used for **other Security Concepts**
  - Secure Boot
  - Secure Running
  - Secure Communications
  - Secure Logs, Secure Storage
  - …

Thank you

Delphi
Technologies

References

- [Ref.1] ISO/SAE 21434:2018 [X] – Road vehicles – Cybersecurity engineering

- [Ref.2] HEAVENS (HEAling Vulnerabilities to Enhance Software Security and Safety) D2 – 2.0 - 2016