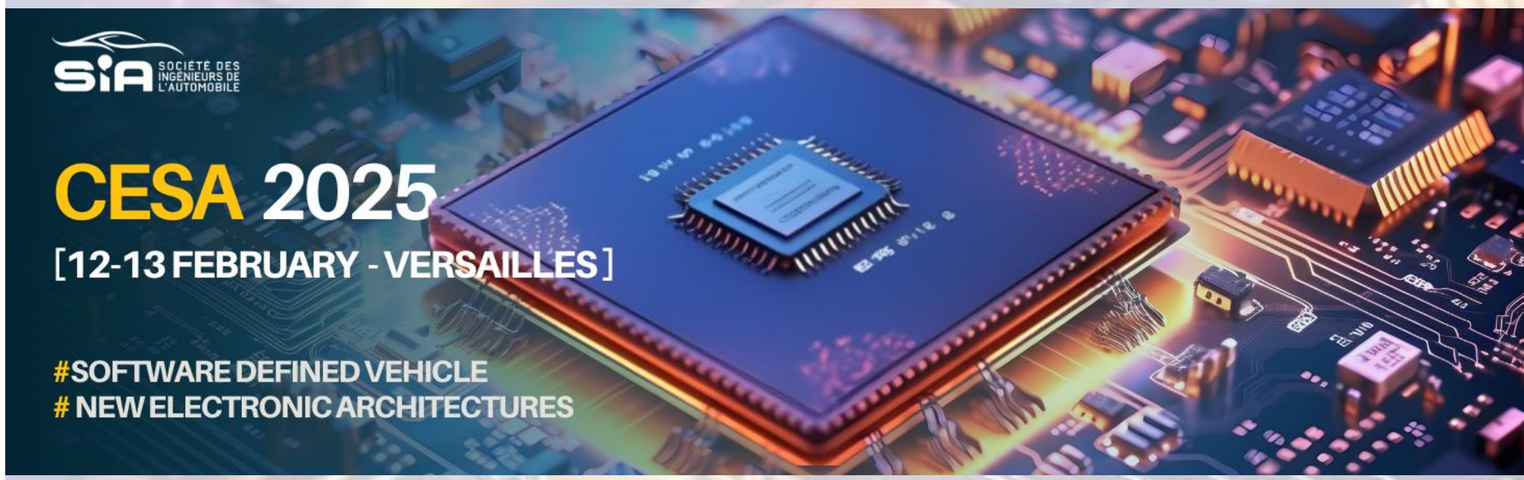
Insight on the impact of SDV on EE/A



Jean-Baptiste LAURENT

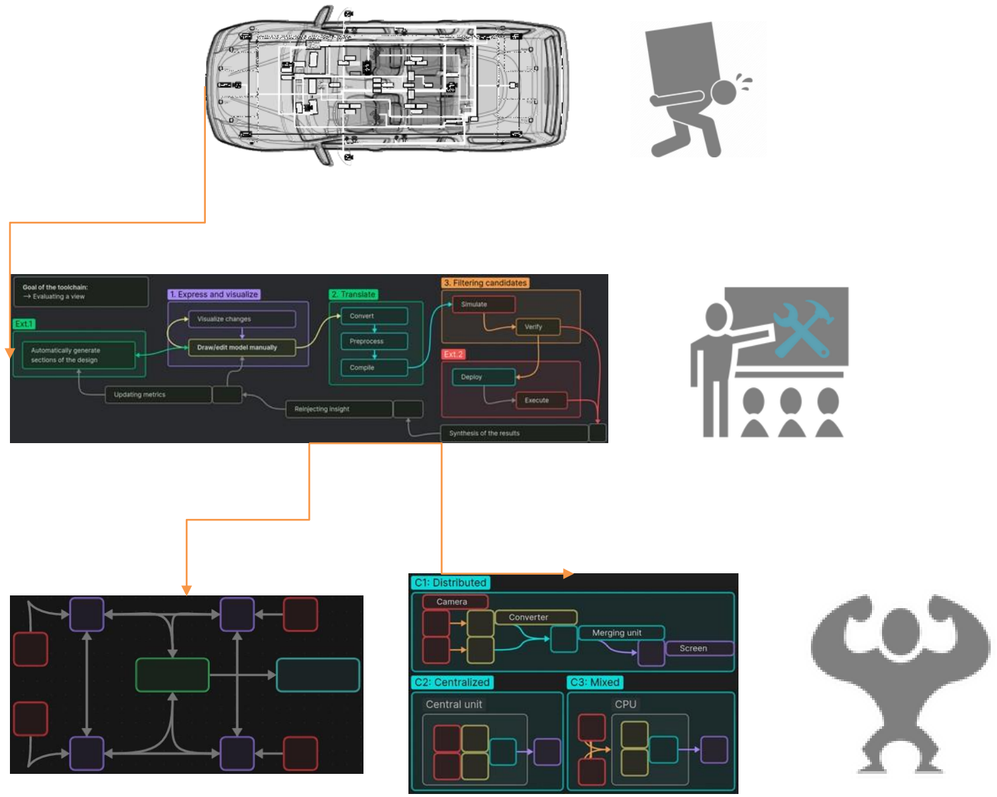
Julien RICHEFEU, Forvia

Bruno MONSUEZ, ENSTA

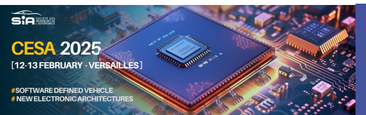
With the support of PFA FILIÈRE AUTOMOBILE & MOBILITÉS, Flev ÉQUIPER LE FUTUR DE L'AUTOMOTIVE, FISITA

Presenting Myself

- Jean-Baptiste LAURENT, 3rd year into PhD
  - Background in Cybersecurity & ASM/C/C++
- Motivation and PhD topic:
  - Research grounded in chaos
  - Research also implies finding random cool stuff
- Today's presentation topic ?
  - What is the deal with SDV ?
  - Exploring one way to craft EE architectures



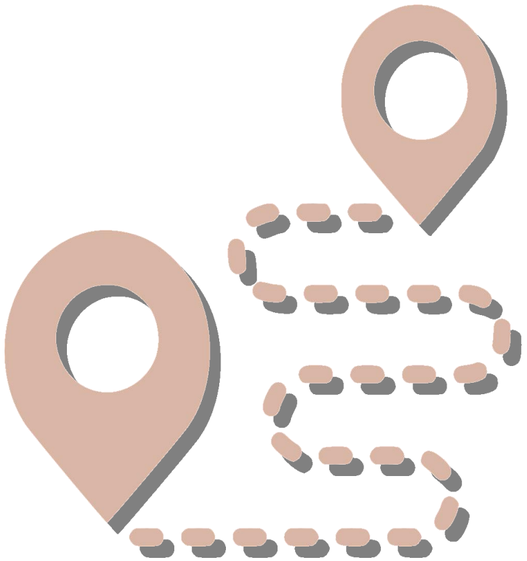
How to design with our method
(covered right next)

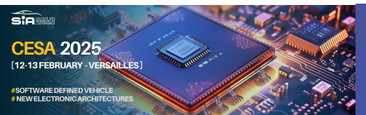


What I am going to talk about

Agenda

I. Introduction
II. Evolution of EE/A : trends and tooling needs
III. Paradigm shift with software development popularization
IV. Transformation imposed by Software Defined Vehicles
V. Conclusion



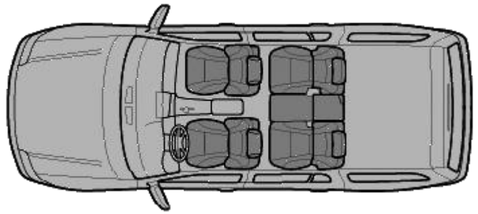


I. Introduction

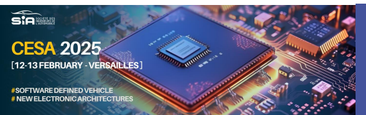
A long time ago, something happened …

- It all began with when the electronic arrived
  - We prefixed the architectures with « E/E »
  - And it changed, everything

→ “EE/A” stands for an Electrical/Electronic Architecture



The EEA has drastically changed since last 40 years, with the introduction of electronics on automotive industry

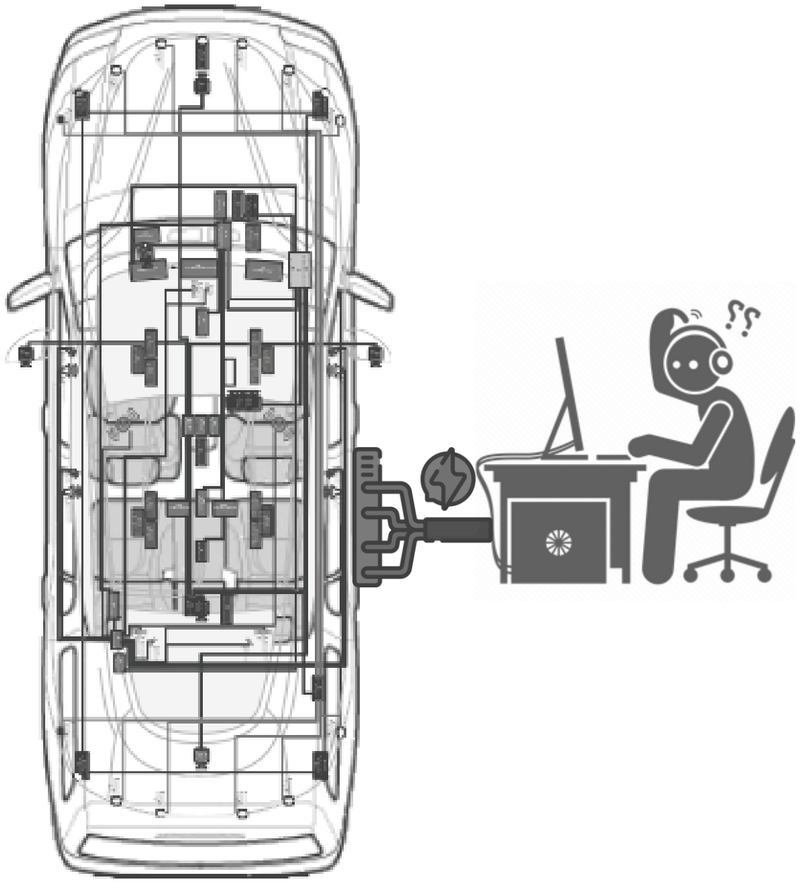


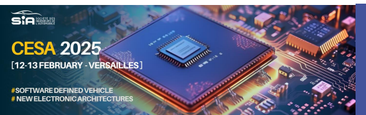
I. Introduction

… and now we got a problem !

- “It” is evolving too fast
  - Things have to be done quickly (because industry)
  - But now EE/A are unfit and too complex

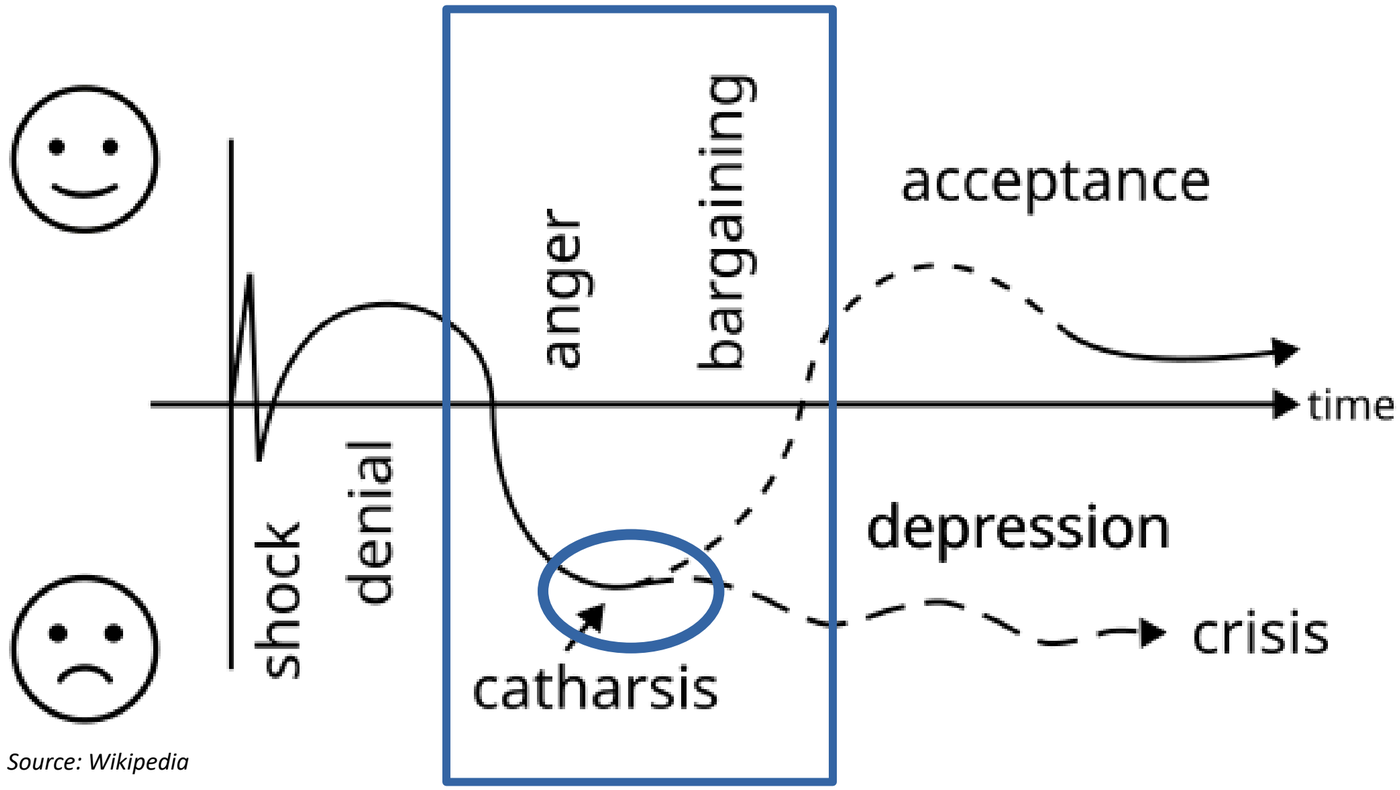
→ Need to deal with electrical and connected features





I. Introduction

Kübler Ross's stages of grief :
* the automotive stage from a researcher perspective



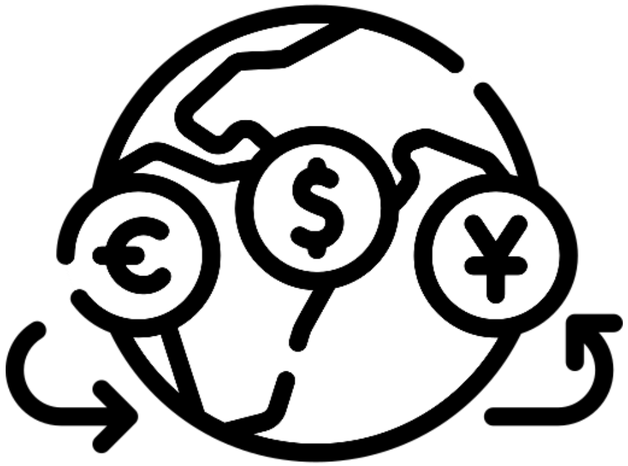
Source: Wikipedia

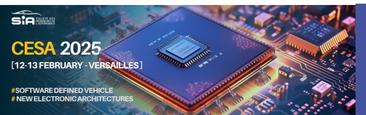
I. Introduction

My guess on what we need

- Flexibility is needed, now more than ever
  - Unless we want to multiply the designs

- New markets, such as China
- Increased automotive lifespan

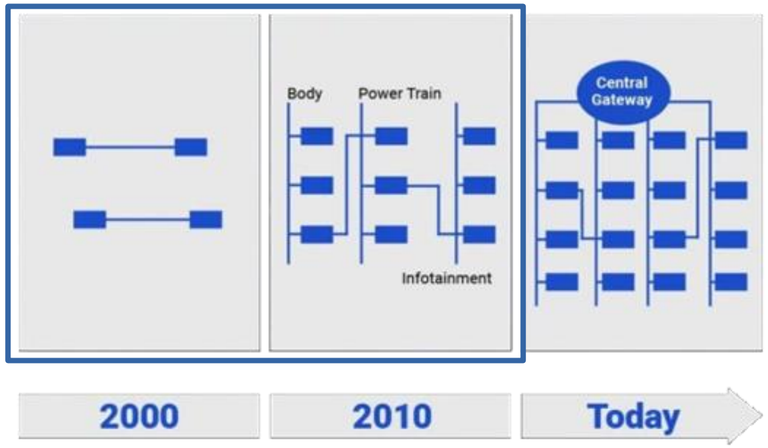


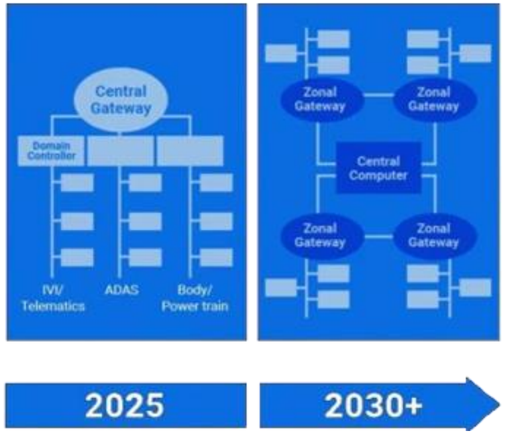


II. Evolution of EE/A : chronology

1 Distributed EE architectures

* 1 feature equals 1 ECU
* Lots of (heavy) wiring
* Combinatorics explosion

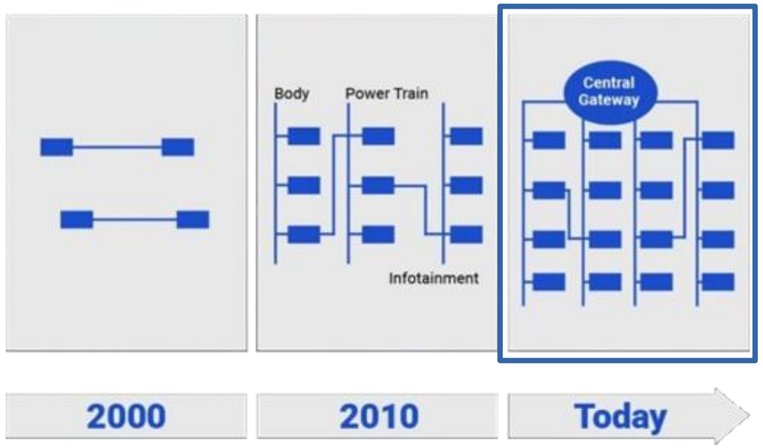


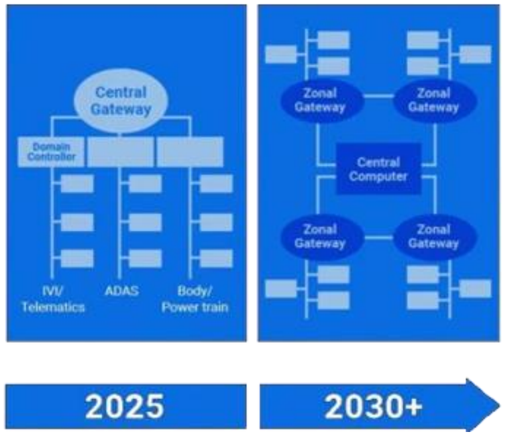


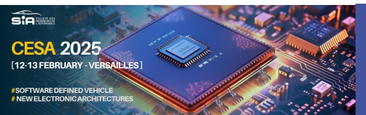


II. Evolution of EE/A : chronology

1	Distributed EE architectures	* 1 feature equals 1 ECU * Lots of (heavy) wiring * Combinatorics explosion
2	Domain oriented EE architectures	* Optimized cost by domain * Specialized ECU * Inter domain communication not possible

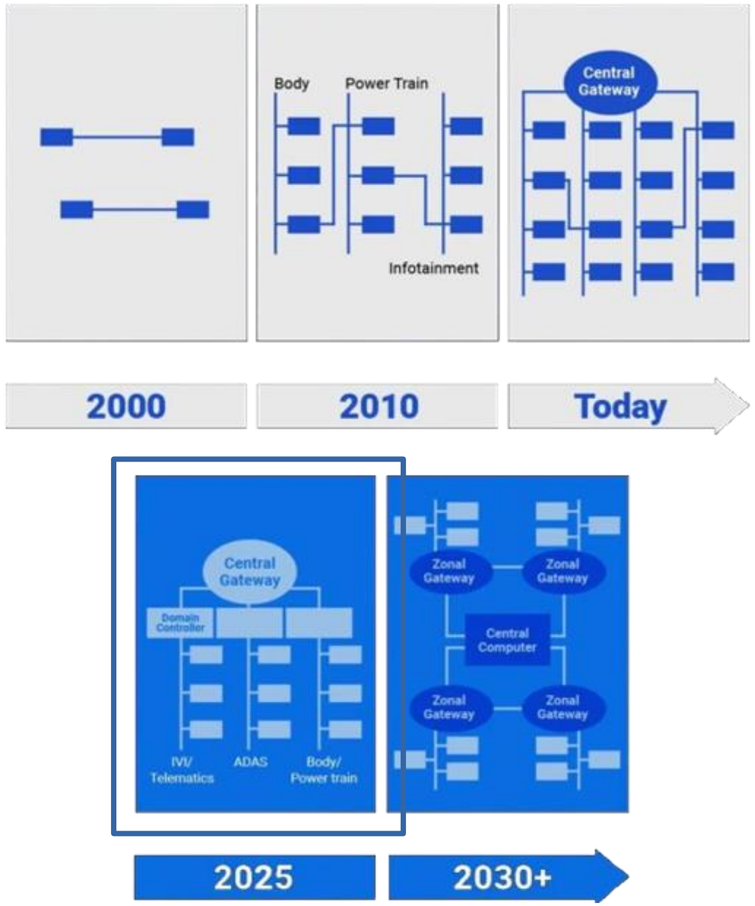






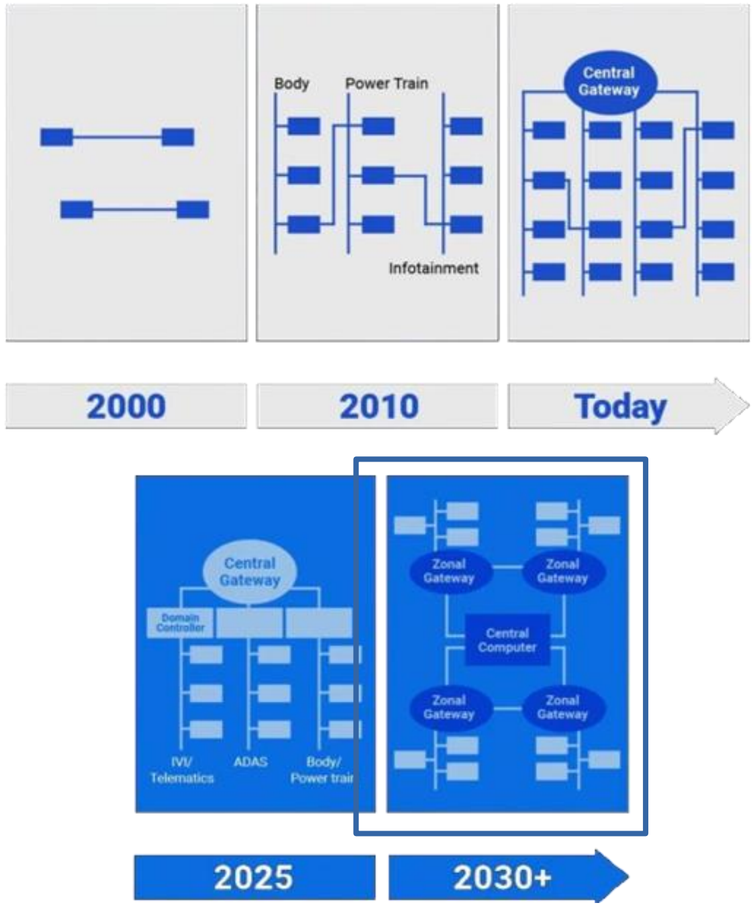
II. Evolution of EE/A : chronology

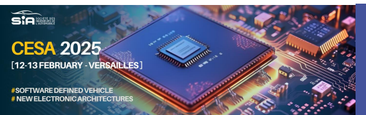
1	Distributed EE architectures	* 1 feature equals 1 ECU * Lots of (heavy) wiring * Combinatorics explosion
2	Domain oriented EE architectures	* Optimized cost by domain * Specialized ECU * Inter domain communication not possible
3	Fully centralized EE architectures	* High Power Computing * System design complexity



II. Evolution of EE/A : chronology

1	Distributed EE architectures	* 1 feature equals 1 ECU * Lots of (heavy) wiring * Combinatorics explosion
2	Domain oriented EE architectures	* Optimized cost by domain * Specialized ECU * Inter domain communication not possible
3	Fully centralized EE architectures	* High Power Computing * System design complexity
4	Zonal oriented EE architectures	* Best trade using zonal and edge computing * Still not dynamic without the help of SDV





II. Evolution of methods

Managing complexity and fitness over time

1-2 Static analysis

* Local optimization
* Tailored but inflexible
* Only accounts for static components

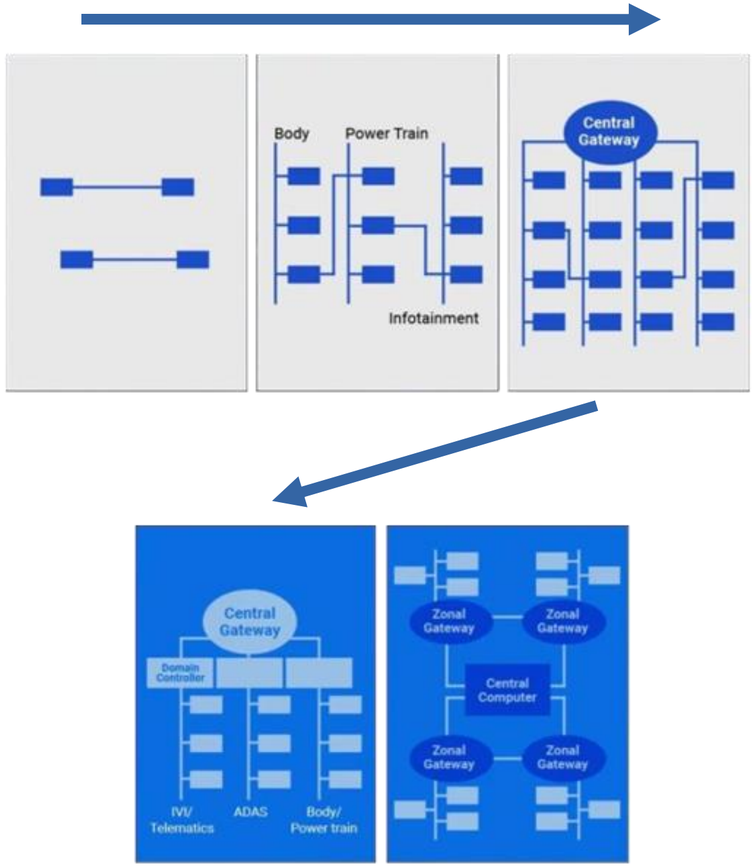
2-3 Aggregation and segregation

* Domains / zonals
* Containers / Hypervisors
* Stronger merges of ECUs and HPCs
* Accounting for safety/cyber areas

3-4 Design space exploration

* Tailoring an architecture by design

→ Accounts for all previous methods

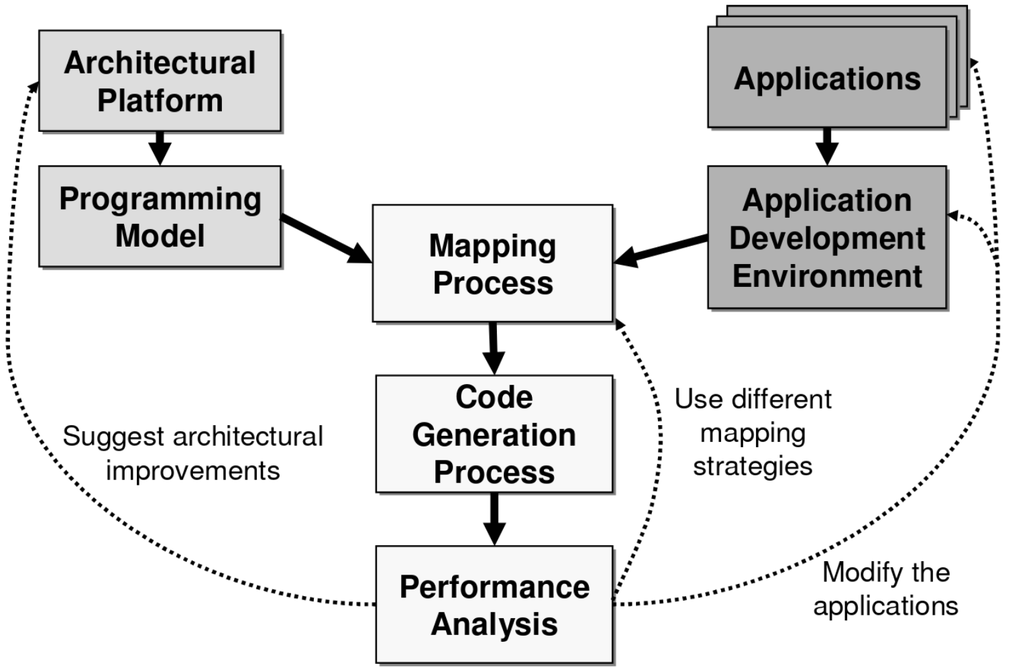




III. Paradigm shift

There is so many things we can do

- Shall we build with design-space-exploration ?
  - There must be many good designs out there
  - We need to find them
- To design the system as a single unit
  - Instead of each sub-systems
- Some tried, but didn’t pass the test of time
  - For instance, the Metropolis framework (Berkeley)



Exploring heterogeneous systems back in the days

Fig 1: “A methodology for architecture exploration of heterogeneous signal processing systems:” : • P. Lieverse, P. van der Wolf, E. Deprettere, and K. Vissers, doi: 10.1109/SIPS.1999.822323.

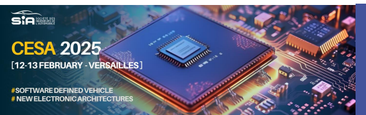


III. Paradigm shift methodologies ?

A change in the process, but we have issues deploying it

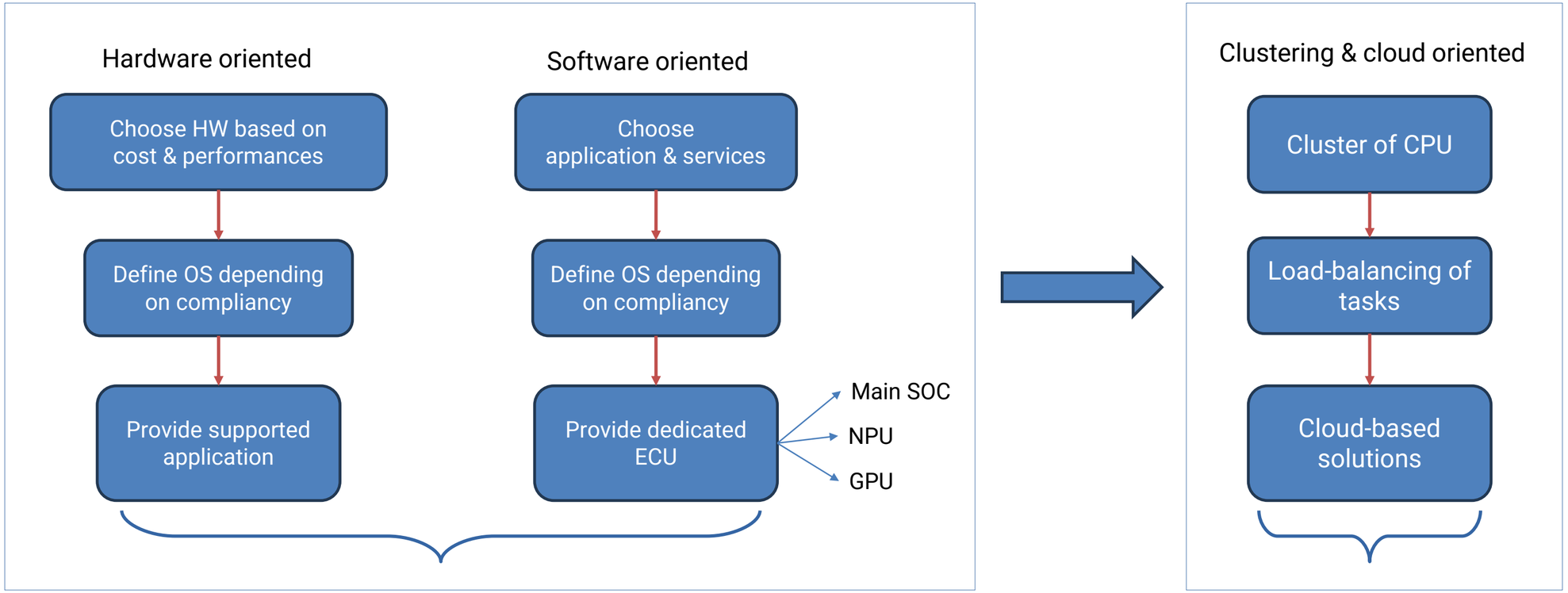
Why is it so difficult to deploy such methodology ?

- A constant race against deprecation
- System engineering must evolve
- System architecture leader role will change (he got centralized as well)
- The adoption of new methods is an ever lasting challenge



III. Simple shift or full transformation?

We got new Horizon : extending with the software and V2X



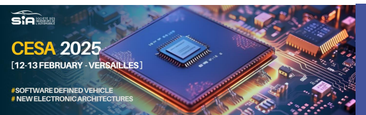
Before centralized EEA & SDV

Now

IV. Transformation imposed by SDV

Definition: *"Software Defined Vehicle (SDV) or 'Vehicle designed around software' refers to the ability to evolve a car throughout its life, thanks to a centralized architecture, by updating it and installing new applications to expand its functions", Renault Group*

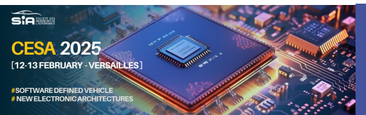
- SDV means new applications, services, features real time
  - Capability to seamlessly and continuously update the vehicle system
  - More secure and safer (AD oriented services, regulation, A-Spice & Safety)
  - Longer life cycle (or less value decreased)



IV. Transformation imposed by SDV

Definition: *"Software Defined Vehicle (SDV) or 'Vehicle designed around software' refers to the ability to evolve a car throughout its life, thanks to a centralized architecture, by updating it and installing new applications to expand its functions", Renault Group*

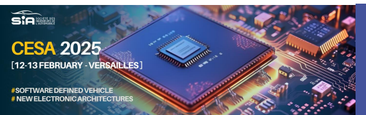
- Also, according to IEEE 1471-2000, this means E/E architectures are software-intensive systems:
  - « software contributes to the design, construction, deployment, and evolution of the system *as a whole* »
  - We can follow existing their best practices recommendations



IV. Transformation imposed by SDV

Current perspective

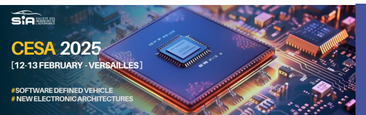
- Shall vehicle be more than just a phones on wheels ?
  - Tesla example, then Stellantis & VW, then Renault Software Lab (Ampere)
  - Google Android AAOS-SDV example with new and different Android flavors
- Speed up development time *(--lazy-evaluation)*
  - As seen in the China market
- Needs for providing dedicated services based on collected data
  - Which services, which data ?



IV. Transformation imposed by SDV

Points to worry about, and the notion of power

- Car vs phone debate
  - Phone or car navigation system ?
  - Phone or car playlist and apps ?
- The growing weight of the GAFAM, and their influence
  - Will car makers still decide ?
  - Or will the GAFAM impose their technologies ?
- Impact of the US and China automotive market
  - Regarding the European market
- Ethical considerations

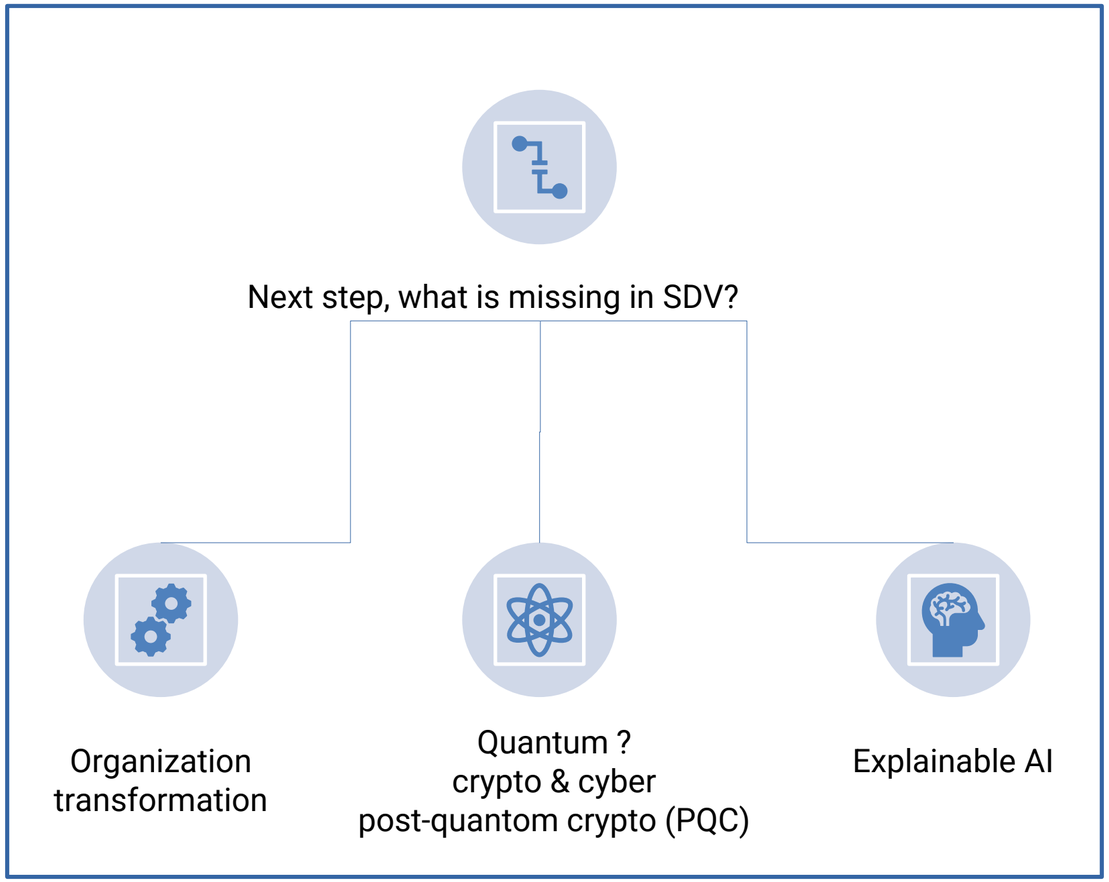


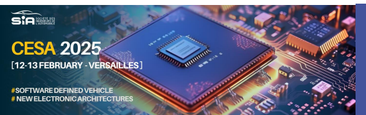
V. Conclusion

- Finally, what is the benefits for end-user ?
- Will we see changes in the Adoption rate ?
- Personalization vs regulation (customizable cars)

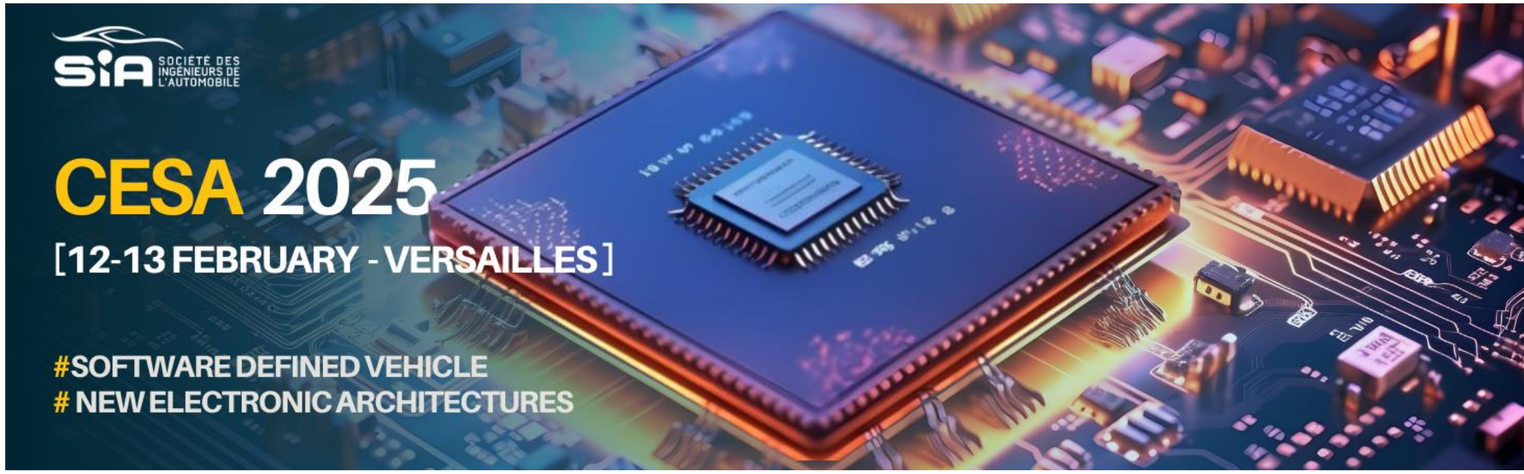
- IVI navigation maps vs mobile phone

- Predictive maintenance
- Digital car manual and AI based recommandations





THANK YOU FOR YOUR ATTENTION



Contact us at:

- jeanbaptiste.laurent@protonmail.com
- julien.richefeu@forvia.com
- bruno.monsuez@ensta-paris.fr

With the support of PFA FILIÈRE AUTOMOBILE & MOBILITÉS, Fiev ÉQUIPER LE FUTUR DE L'AUTOMOTIVE, FISITA