FREEDOM TO EVOLVE

ZONAL ARCHITECTURE:

THE FOUNDATION FOR NEXT-GENERATION VEHICLES

GUARD KNOX

TABLE OF CONTENTS

FREEDOM TO EVOLVE

EXECUTIVE SUMMARY

The landscape in which the automotive industry operates is changing dramatically. The consumers of today expect drastically different experiences and functionalities from a vehicle than they did previously. The new generation of automotive consumers expect constant internet connectivity, a fully customizable driving or riding experience, as well as personalized entertainment functionalities. Simultaneously, consumers expect to feel safe and secure, while enjoying the latest, modern features on-demand, in real-time and over-the-air as they download different applications and services from vehicle "app-stores". In fact, the vehicle is no longer the focal point in the consumer's mind, rather it is the mobility *service or mobility* experience that it provides.

Learn more about the paradigm shift happening in the automotive industry.

The traditional approach of designing a vehicle's E/E Architecture, one that has proven itself for decades, is no longer viable. The automotive industry has responded to the consumer trends by gradually adding more and more ECUs, millions of lines of code, and hundreds of specialized suppliers and parts to the modern vehicle. But the current E/E Architecture has reached its scalability limits. It can only be surpassed by a technological shift, which in turn creates even more challenges. The most important vectors of evolution for the new E/E Architecture include the following:

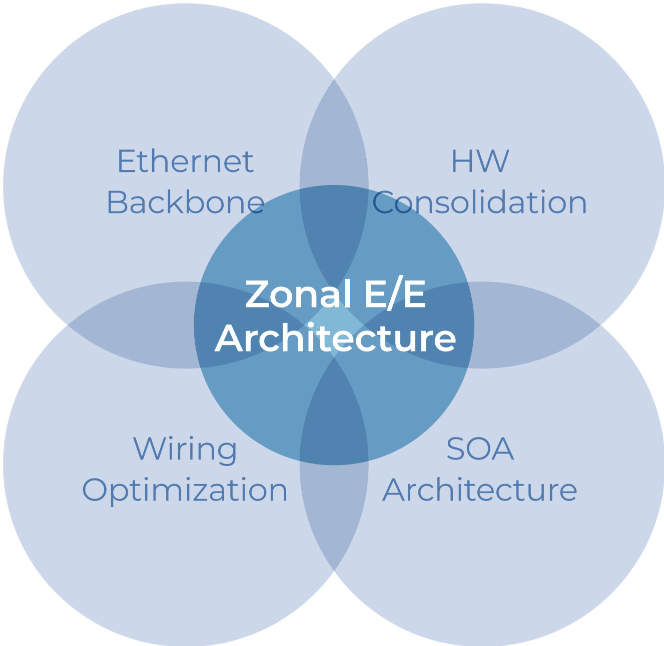
1. **Ethernet Backbone**
Moving from the CAN bus architecture to a modern high speed Ethernet communication network.

2. **Hardware Consolidation**
Consolidating multiple functions that are today served by separate ECUs into ECUs that are multi functional.

3. **Wiring Optimization**
Consolidation of ECUs together with new topologies for the vehicle networks to reduce the needed cabling length, weight, and cost to a fraction of what it currently is.

4. **Software-Driven Service-Oriented Architecture**
The vehicle software architecture is evolving towards a Service-Oriented Architecture that can accommodate the needed flexibility, security and agility for the new software-defined vehicles.

These 4 vectors make up what has become known as the "Zonal E/E Architecture".



This new approach addresses scalability, enabling automotive manufacturers to meet new consumer expectations while reducing cost and overhead altogether. However, making so many changes all at once is a risky endeavor, but not one without rewards.

The emergence of zonal networks can be seen as part of a shift in the way the automotive industry conducts its business. What seems at first as an evolutionary step addressing technical problems, is in fact a change of the vehicle definition and its supply chain - enabling new business opportunities and redefining the traditional roles of the OEM and Tier 1.

Below the reader will find a framing to:

- Current state and limitations
- Possible solutions
- Challenges
- Business opportunities
- Industry implications

HOW WE GOT HERE: A BRIEF HISTORY OF VEHICLE ELECTRONICS

ORIGINS: THE PROTO-MOTOR-VEHICLE

Car electronics are not a new concept. They were first introduced in the 1970's and were mostly a luxury or a novelty in the form of simple and superfluous trip computers, digital dashboards and the like. Through the years, car electronics became more widely used and like almost any other aspect of life, the introduction to new technology, in the form of the microprocessor as well as the ever-decreasing price of the transistor, changed everything. So much so that, by the 1990's, almost all cars had computer-controlled systems onboard.

The first mainstream adoption of computerization in the automotive industry was the Electronic Fuel Injection (EFI) system, an alternative to the antiquated carburetor as a fuel delivery system. With increased engine output, faster throttle response, better fuel economy and lower emissions all at once, the success of EFI showed just how much replacing or augmenting a traditional mechanical part with logic circuitry could offer. The original computer responsible for EFI was called an Electronic Control Unit (ECU), a term which has now become a generic term used to describe any vehicle computer unit.

Other electronic components were introduced later, with the goal of providing higher levels of satisfaction and value to the customer. Many of these became standard either through regulation or consumer demand, like Anti-lock Braking System (ABS) to increase safety, or climate control systems for greater passenger comfort. The rise of electronic controls drove the need to establish a unified, reliable means of communication with other ECUs and other electronic components such as sensors and switches. As more and more functions and intelligent features were added, they began to form a network within the vehicle, which in turn required a new automotive standard.

Introduced to the world in the 1980's by Bosch, the Controller Area Network (CAN) bus became the standard for digital communications in the automotive industry. The CAN bus is simple, cost-effective, robust and widely adopted. It was an obvious choice for the ever increasing number of electronic components as more and more functionality was digitized and modernized the car. The standard itself has evolved to meet new requirements but has remained fundamentally the same. It is still the most widely used bus system in cars to this day and is likely to remain so for the foreseeable future. Despite that, CAN has reached its limit of Electronic/Electrical (E/E) network scalability. It alone can no longer serve as the communication network backbone.

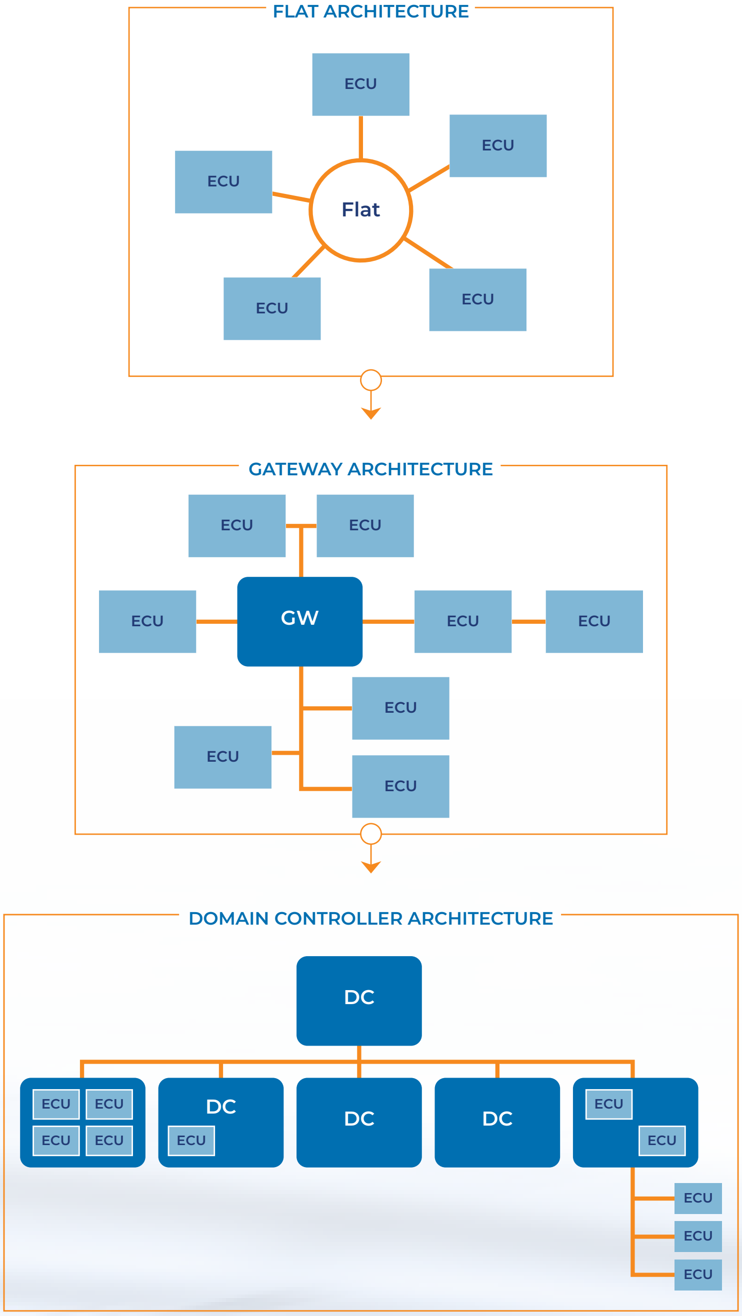
HITTING THE SCALABILITY WALL

Drivers on the road today are no longer mechanically linked to the vehicle. Instead, sensors relay various inputs to on-board electronics which in turn control the mechanical aspects of the vehicle. The gas pedal, which once was a mechanical link to a carburetor valve, is now a position sensor, designed to look and feel the same as a traditional gas pedal would to the driver. This is true not only for autonomous vehicles, but also for cars already in production with features such as automatic parking and active safety. The vehicle is already effectively driving itself.

The communication network and its associated wiring has also become a major pain point for the industry. Despite multiple attempts to replace the aging CAN bus with a modern alternative, most, if not all, failed to catch on. Designed to convey simple messages through fewer wires, CAN's low bandwidth and large broadcast domains created a limit to how large a vehicle network would be able to scale given congestion, delays, and data transmission constraints. To date, OEMs have managed to engineer around these problems - mainly with the introduction of gateways - but that too has reached its limits. Until recently, it seemed that the CAN bus was here to stay.

Generally, automotive E/E Architecture has made three major evolutionary advancements:

1. **Flat:** All ECUs are connected to the same bus.
2. **Gateway:** Several flat network segments are connected together through a dedicated gateway ECU in order to reduce broadcast domains.
3. **Domain Controller:** Hierarchical logical arrangement of ECUs, it involves some consolidation into high-performance computing platforms. This is the first architecture to start moving away from CAN as it requires a high speed backbone handled by a dedicated Gateway Domain Controller.



**LOGICAL REPRESENTATION, NOT OPTIMIZED FOR PHYSICAL LAYOUT*

Traditional E/E designs - logical layout

Like in many traditional industries, when originally introduced, vehicle electronics were treated like any other functional (mostly mechanical) part, each performing a single or limited number of functions. Each new function required the commissioning of a new part, or a new ECU. This mindset is in contrast with the path the computer industry went; where more and more functionality is abstracted and performed mostly by software.

We are now at a point where complexity is no longer manageable given consumer demands and industry trends. There are simply too many ECUs to handle. Each individual ECU requires development, maintenance, logistics, cost, variants, weight, configuration and obsolescence. Currently, a shortage of a single wire harness is all it can take to halt an entire production line.

Fortunately, however, OEMs are already looking at ways to solve this by revolutionizing the way that cars function, are built and are managed. The apex of this is the Zonal E/E Architecture concept.

THINKING ABOUT NEXT-GENERATION E/E: THE DISTRIBUTED BRAIN

INSPIRATION OUTSIDE THE AUTOMOTIVE INDUSTRY

The aviation industry was plagued with the exact same problems the automotive industry is presently facing. Several innovative ideas were introduced including a new approach to "Avionics" or aviation-electronics, the equivalent of an ECU.
Some of these changes included:

- **Hardware reduction** - high-performance multi-role mixed-criticality modular computers.
- **Wiring reduction** - high speed communication backbone.
- **Increased functionality** - modular software as a service (SaaS).

Many innovations in automotive originated in aviation a few decades prior, it is safe to assume that this case will not differ.

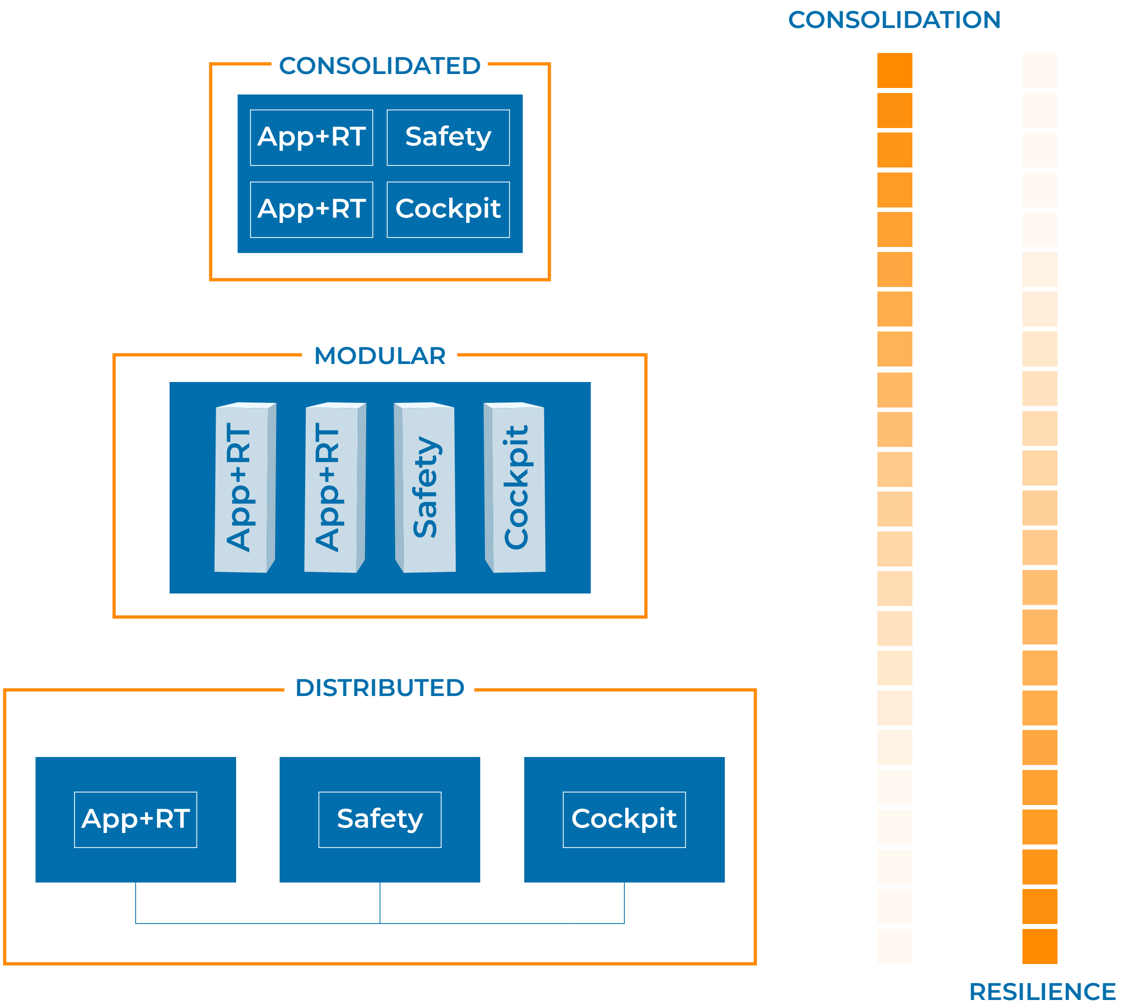
ZONAL E/E ARCHITECTURE FUNDAMENTALS

The Zonal Architecture concept is, at its core, an optimization of the computational resources and wiring of the Domain Controller Architecture. By taking into account the physical properties of the vehicle, a Zonal Architecture introduces two new device classes: *Vehicle Server* and *Zonal Gateway*.

Edge devices, such as sensors and actuators, as well as legacy ECUs, remain the same as they do in other architectures.

	Main Purpose and Function	Consolidation / Reduction	Connectivity and Interfaces	Main Attribute
Vehicle Server	Mixed criticality application host	Physical ECUs	Few high speed Few low speed	Computational power
Zonal Gateway	Localized connectivity hub with edge processing	Wiring	Many low speed / legacy Few high speed	Connectivity (Number and type of interfaces) and Latency

The purpose of the *Vehicle Server* is to reduce the number of physical ECUs in the vehicle, through consolidation of hardware components. Less silicon, less wiring and less casing will result in less weight and an overall cost reduction. The *Vehicle Server* does not have to be a single physical device; it can be implemented as a *modular server* by connecting several high-performance ECUs, clustering System on Chip (SoCs) on the same Printed Circuit Board (PCB), or by having a base backplane with expansion cards. Logically, however, they are all identical.



Vehicle Server configuration

The purpose of the *Zonal Gateway* is to act as a local connectivity hub, aggregating multiple low speed (mostly legacy) interfaces and relaying data through a single high speed Ethernet link to the backbone. The Zonal Gateway may also perform edge processing and can offload or prepare data for a Vehicle Server to process in order to reduce workload and bandwidth.

Latency is an unavoidable concern. The problem stems from using a switched network as frames need to "hop" network nodes to their destination, reception and transmission time therefore accumulates.

THE COMPUTE PLATFORM

Although specialty hardware exists such as ADAS, cockpit, autonomy or safety dedicated, a typical Vehicle Server/Domain Controller consists of two execution domains, typically consolidated to the same SoC: *application* and *real-time*.

The *application domain* is a feature-rich environment that commonly relies on existing ecosystems such as Linux or Java that focuses on functionality. Often, this domain runs a hypervisor in order to ease implementation and widen the runtime options as much as possible while separating criticalities. This domain mainly relies on hardware enforced logical isolation and is the largest domain in terms of software and therefore, functionality. Despite its name, the application domain may host time sensitive and safety-related functionality given the right configuration.

The *real-time domain* is, usually, dedicated hardware that performs similarly to traditional ECUs where time-sensitive and safety-related tasks are being handled independently of other functionalities. This domain typically runs an Real-Time Operating System (RTOS) and relies on physical isolation as hardware tends to be more simplistic and resources constrained.

	Typical processor type	Runtime / middleware	Hosted applications	Resource allocation
Application domain	Multicore microprocessor	RTOS + hypervisor	Any	Partitioned
Real-time domain	Lockstep Microcontroller	RTOS	Time sensitive and safety critical	Dedicated

SECURITY AS A FOUNDATION

Consolidation, however, comes at a cost. Putting more and more eggs into fewer baskets raises two main concerns: *safety* and *security*.

The industry is addressing both through standardization, *ISO 26262* and the upcoming *ISO 21434*.
ISO 26262: Road vehicles - Functional safety
ISO 21434: Road vehicles - Cybersecurity engineering

	ISO standard	Focus	Assessment	Measurement
Safety	ISO 26262	Functional integrity	Failure probability	Integrity
Security	ISO 21434	Information and functionality	Risk management	Assurance

Although each standard is distinct, they are two faces of the same problem. Security affects safety - a security flaw is a safety hazard - therefore addressing security is inherently addressing safety.

When it comes to similarities, both are process driven, both are iterative processes and both must be taken into consideration from concept phase to post-deployment. Sometimes both are synergistic and rely on the same measures to achieve their goals, the table below has a few examples.

	Secure boot	Resource isolation	Network protection	Secure storage
Safety	Flashing wrong firmware by mistake	Freedom from interference between components	Malfunctioning ECU transmitting random data	Private data for emergency services
Security	Flashing OEM authorized firmware only for liability	Limit attack propagation from one system component to another	ECU becomes attack vector through vulnerability exploit	Proprietary IP secrecy

However there is one major difference that should be mentioned: security is not the sum of its parts. Unlike safety, connecting two systems with the same assurance level does not necessarily result in a secure system.

Security is a fundamental aspect of consolidation and is the enabler of such platforms. Once many functional units are put together, a breach in one may propagate and end up exposing the entire system, compromising other functionalities and potentially affecting vehicle safety. Furthermore, it may be mandatory to comply with regulatory requirements such as privacy and financial data protection. History has taught us again and again that security is not an additive and cannot be patched in. It must be designed into the system, as an engineering effort, much like how safety concepts are a preliminary step.

Given that almost all vehicles produced today are connected and that commercial fleets play a pivotal role in driving the global economy, cybersecurity is likely to become a matter of national security. Governments will most likely take an interest in and therefore tighten regulation on this topic, similar to the not so distant past where seatbelts and rearview mirrors were originally factory options and were not mandated by regulatory or governmental bodies. Today, independent organizations perform their own safety tests and disclose the results to the public, the same could happen with security testing.

When it comes to designing any secure computer system, two core methodologies must be employed from the onset:

- **Security by design** - risk identification and countermeasure definition.
- **Defense in depth** - limit or nullify the propagation of a compromised system element.

There are no set rules which apply to every component or use case; every system and configuration must be evaluated individually. However one goal should be in every system - a breach must not compromise the safety of the vehicle occupants and road users.

THE ETHERNET BACKBONE: THE NERVOUS SYSTEM

Ethernet was introduced in the 1970's and commercialized in the early 1980's, and has now become a standard for digital communication in virtually all other industries. It is a late but welcome addition to the automotive industry as it frees the E/E Architecture from existing limitations, opens up new opportunities and simplifies complexity.

There is no single approach and no limitations on topology when implementing an Ethernet backbone, however there are 3 distinct leading network configurations::

	Ring	Tree	Star
Network resilience	+ + +	+ +	-
Hardware consolidation	+	+ +	+ + +
Latency (hops)	-	+ +	+ + +

As a general rule, there is a trade-off between consolidation and resilience.

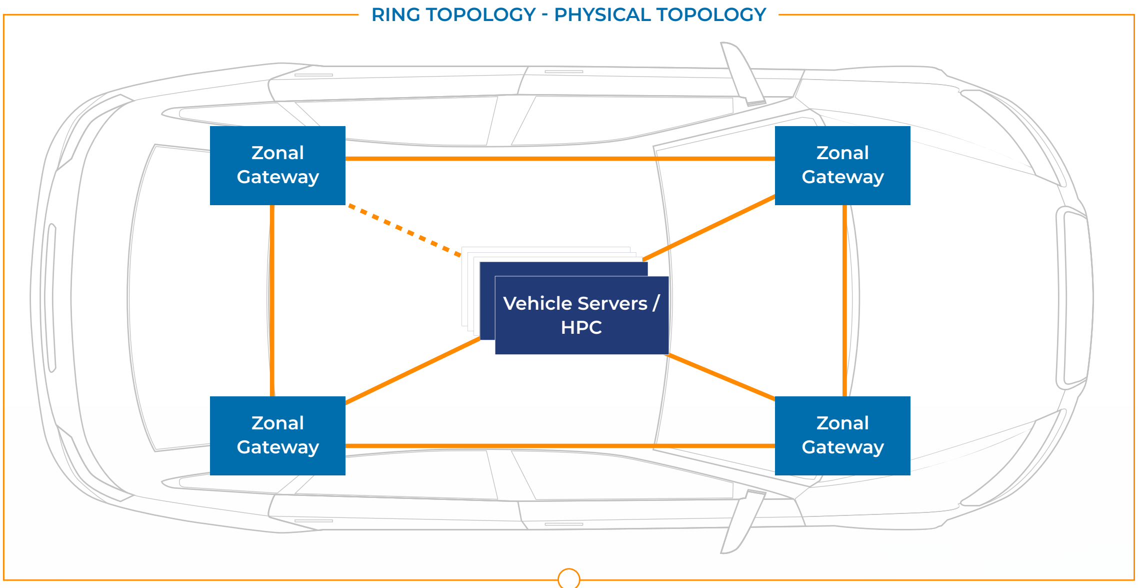
Topology trade-off axis

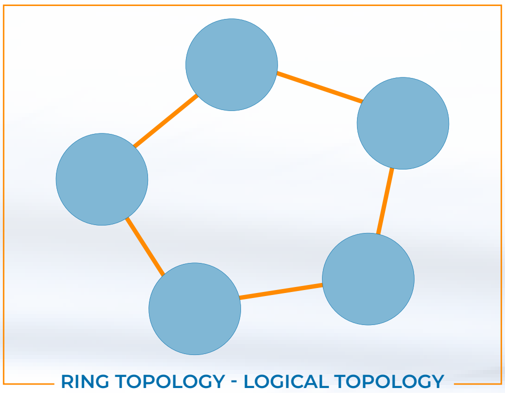
Ring Topology: Dual-Redundancy, Fail-Operational

With the Ring configuration, all or the majority of ECUs are connected in the form of a closed loop. Within that loop, there are 2 paths to reach every ECU, clockwise and counter-clockwise. In case of a single link loss, the network can still remain functional, resulting in dual-redundancy and is therefore fail-operational.

The downside of this configuration is that it maximizes hops (latency) - frames may need to cross the entire network to reach a specific ECU. The servers can be centralized or distributed to some extent as long as they will retain a ring topology.

Please note that this network configuration requires suitable equipment and configuration.





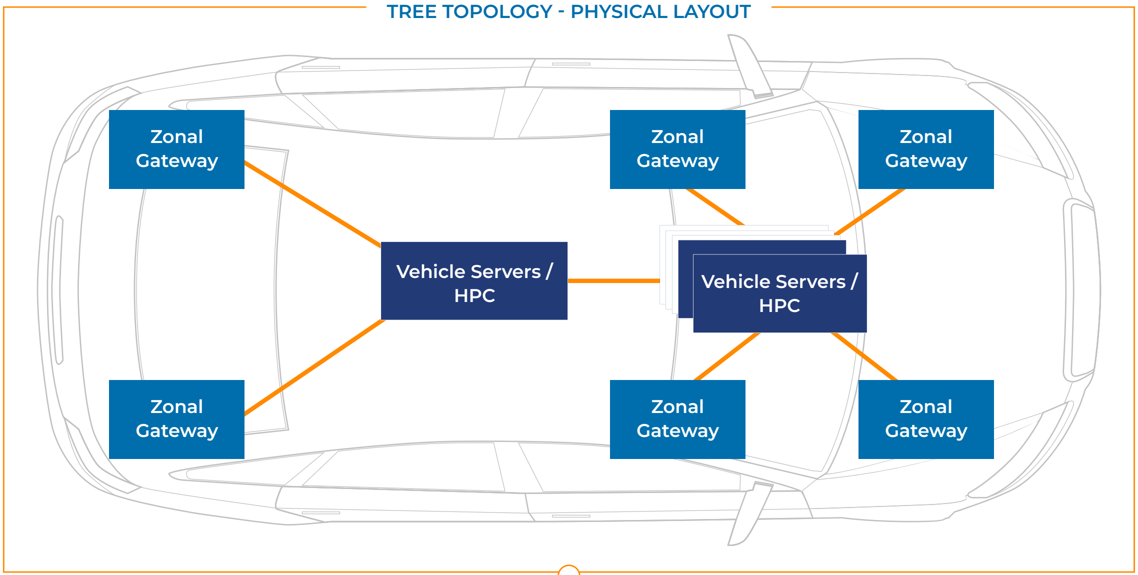
Tree Topology: Striking a Balance Between Consolidation and Hops

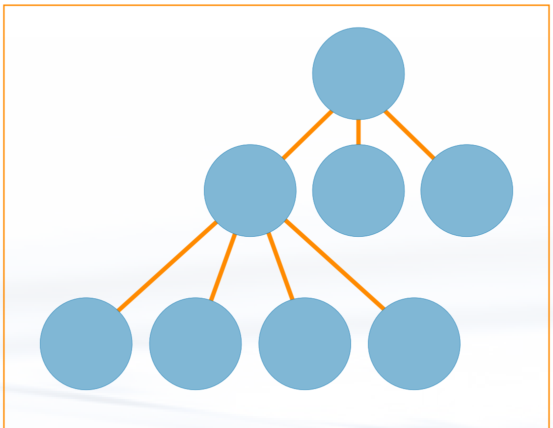
Tree Topology consists of two or more servers or clusters split across the vehicle. It appears logically as a hierarchy. The Zonal Gateways are connected as tree nodes that may be leaves. This configuration offers a good balance between hardware consolidation and number of hops.

The main drawback of the Tree Topology is that the main link between the clusters is a single point of failure. A breakage in this line would cut off a large portion of the network and leave compute elements completely isolated. It might be difficult to offer reliable redundancy to this line due to the physical layout of this configuration.

Domain Controller Architecture can be viewed as a logical Tree Topology which is not optimized for physical layout. In fact, when done right, Vehicle Servers can be made agnostic to the network configuration and can function (or rather just be labeled) as Domain Controllers as well.

TREE TOPOLOGY - PHYSICAL LAYOUT





TREE TOPOLOGY - LOGICAL LAYOUT

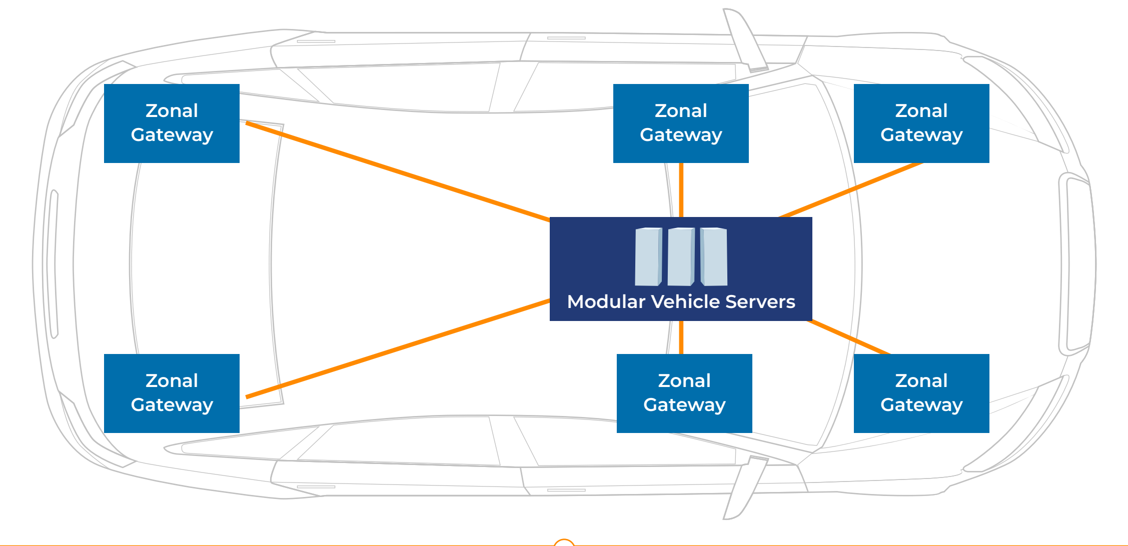
Star Topology: Maximum Consolidation With Minimum Hops

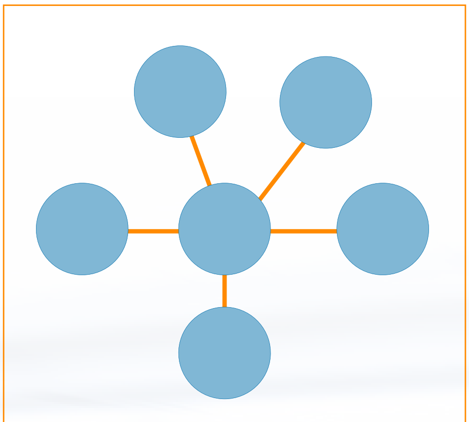
With the Star Topology, hardware consolidation can be maximized to a single ECU by combining all the necessary hardware to a single PCB (effectively a cluster computer), motherboard with extentions cards (like an enterprise blade server) or modular cabinet (like the aviation concept of Integrated Modular Avionics).

When all links radiate from this center unit, the network will end up with a minimal number of hops. Latency can be further reduced by introducing a high speed backplane for the server modules instead of automotive Ethernet (AXI, PCIe, fiber-based Ethernet etc.).

The most significant concern with the Star Topology is that the entire vehicle “brain” is placed in one place and is likely to be fed from a single power line, creating a very large single point of failure.

STAR TOPOLOGY - PHYSICAL LAYOUT





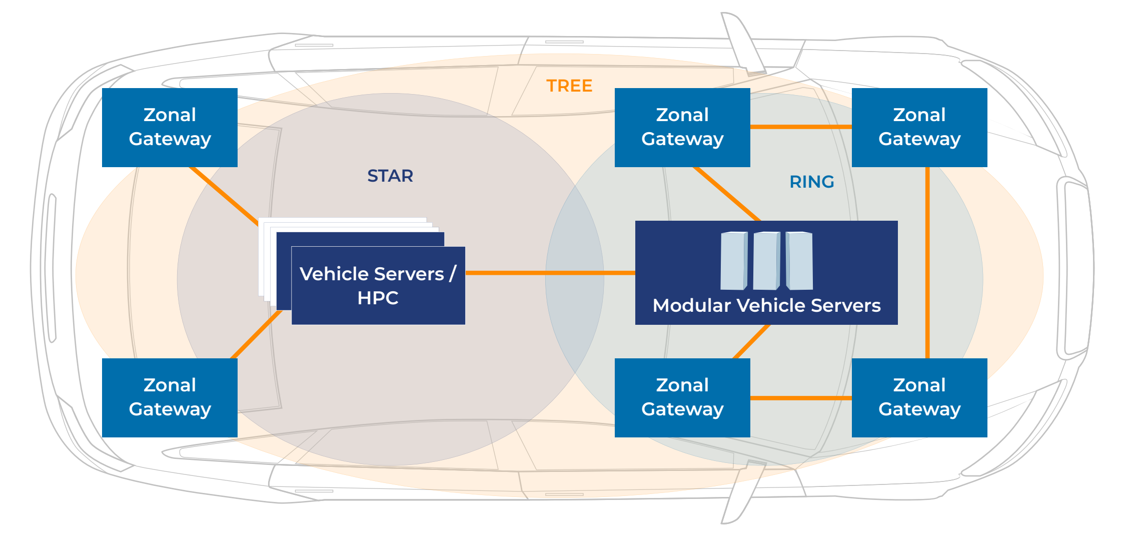
STAR TOPOLOGY - LOGICAL LAYOUT

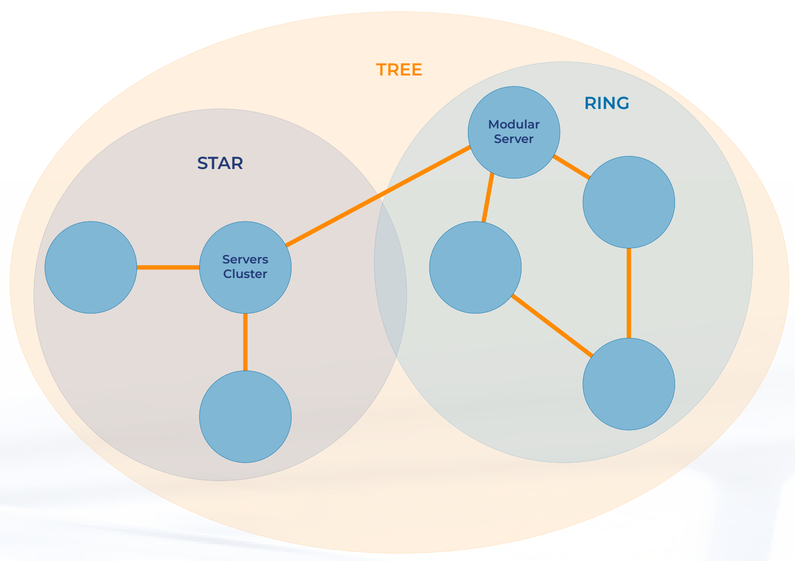
Hybrid Topology: Mix-and-Match

Realistically, the E/E design will be a combination of topologies, effectively a *hybrid*, for practical reasons. Weight distribution, Electro-Magnetic Interference (EMI), wiring tunnels, heat distribution, cooling, accessibility, aerodynamics, exposure to the elements, module placement (e.g., in vehicle crumple zones), redundancy, power distribution and legacy components are all considerations that have to be taken into account.

Below is an example of how flexible Ethernet connectivity actually is. On the left, there is a *star* connected to a *modular server* which is part of a *ring* on the right. The network itself is connected as a *tree* hierarchy.

HYBRID TOPOLOGY - PHYSICAL LAYOUT





HYBRID TOPOLOGY - LOGICAL LAYOUT

SCALING THE WALL

The transition from CAN to Ethernet is not as straight forward as it may seem. Moving from a theoretical maximum of 1 Mbps to a very real 1000 Mbps line rates poses a technological challenge.

First, Ethernet controllers are much more sophisticated and complex than CAN controllers. Second, scaling to over 1000 times the bandwidth requires major changes to hardware with microprocessors and dedicated hardware accelerators as opposed to a simple microcontroller implementation. Lastly, scaling up the computational elements requires scaling up other peripheries such as memory (which can no longer be integrated into the SoC).

	Frequency	Memory	Bandwidth	OS
Microprocessor	Up to about 1000 MHz	External 2-32 GB flash 1-16 GB DDR	500 Mbps and above	Multiple (hypervisor) rich and RTOS
Microcontroller	Up to about 600 MHz	Integrated 1-16 MB flash 256 KB - 16 MB SRAM	Up to 100 Mbps	Dedicated RTOS

Reaching full bandwidth requires changes to the programming model: the sheer number of interrupts alone will overwhelm even a powerful processor, and interface polling will require the dedication of expensive processing power. With small frame sizes, the problem is amplified. There is always a tradeoff between latency and bandwidth.

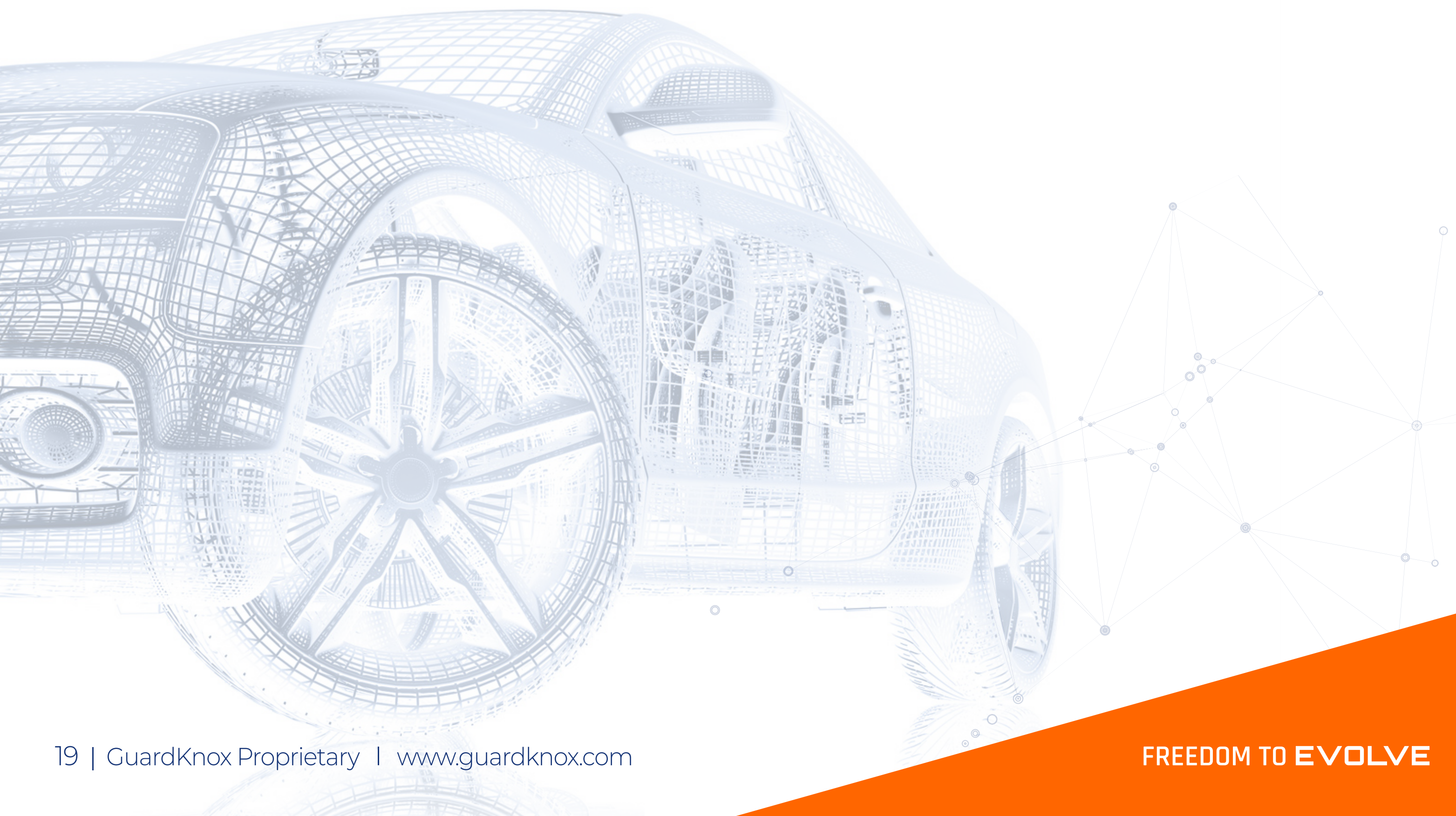
In contrast from CAN networks, achieving determinism is also different in Ethernet. What was, for the most part, handled by the media controller and transceiver in CAN is introduced into Ethernet through several standards often referred to as Time Sensitive Networking (TSN). These require careful Ethernet Media Access Controller (MAC) selection and configuration and is especially true with mixed criticality traffic as it's consolidated to fewer lines.

Another defining characteristic is that Ethernet is not a bus, it's a switched network that can incur delays as there are active network elements relaying and directing frames. These switches, unless specifically selected and configured to do so, cannot be connected in a closed loop.

	Industries	Network	Line rate	Determinism	Security
CAN/CAN-FD	Automotive Industrial Aviation	Bus	CAN: Up to 1 Mbps CAN-FD: Up to 1/8 Mbps (theoretically more in data phase)	Designed in	None
Ethernet	All	Mostly switched	Up to 100 Gbps (theoretically unlimited)	Extension standards	Standard protocols and Extension standards

One major advantage Ethernet offers over CAN is network modularity and the interchangeability of components. With complementary software, multiple network configurations can be supported with hot pluggable attachments connected any place on the network backbone - announcing its functionality is ready for use. This would greatly simplify production line assembly, drive down costs and offer an endless number of possible network configurations.

Implementing an Ethernet backbone creates the ability to offer substantially more factory options than what is currently feasible. Retrofitting also becomes a viable option, opening up the opportunity to create new and recurring revenue streams from existing vehicles after the initial sale. Vehicles already on the road can have an upgraded cellular modem or infotainment system after years of use, similar to *aftermarket* purchases of tire rims and other aesthetic upgrades.



THE SOFTWARE-DEFINED VEHICLE - THE NEURONS

EVERYTHING IS SOFTWARE (ALMOST)

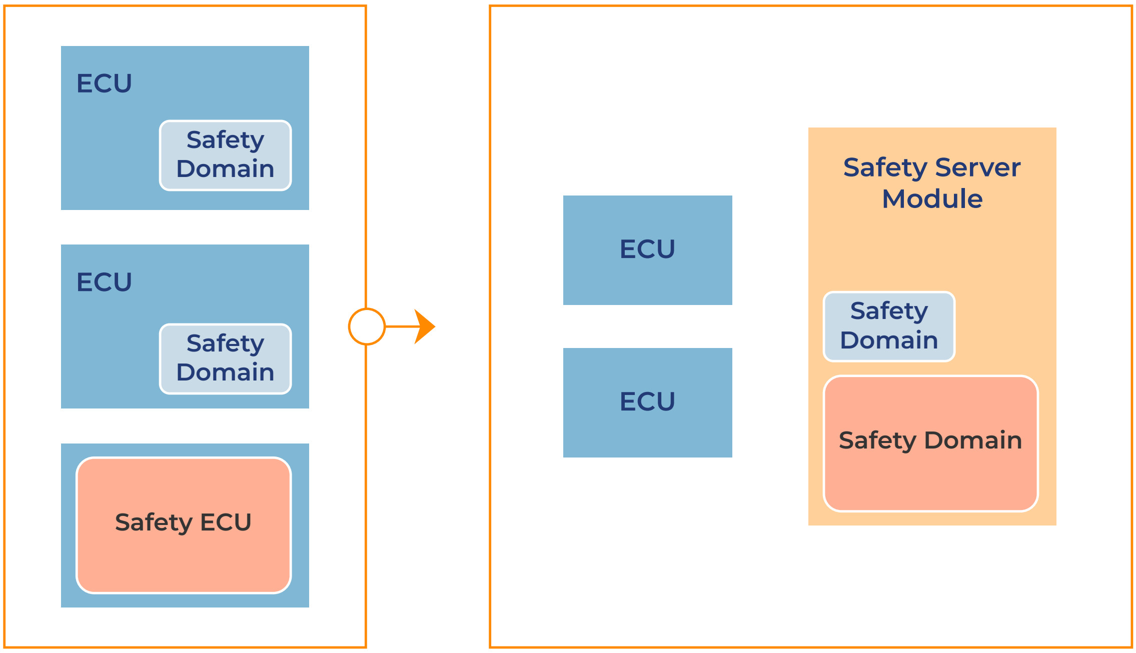
The Zonal E/E concept completely decouples functionality from physical hardware. With that level of abstraction, the computational elements are no longer treated or bound to any physical location, form or constraints. Instead it's treated as a pool or a cloud of resources ready to be distributed and assigned to tasks strictly or on-demand.

By changing the way we think of the E/E Architecture, the options are virtually unlimited. In fact, functionality is no longer bound to the vehicle itself, it can be executed on a remote cloud server, similar to the way smartphone ecosystems function today. This therefore, further reduces on-board computational resources and even eliminates some entirely.

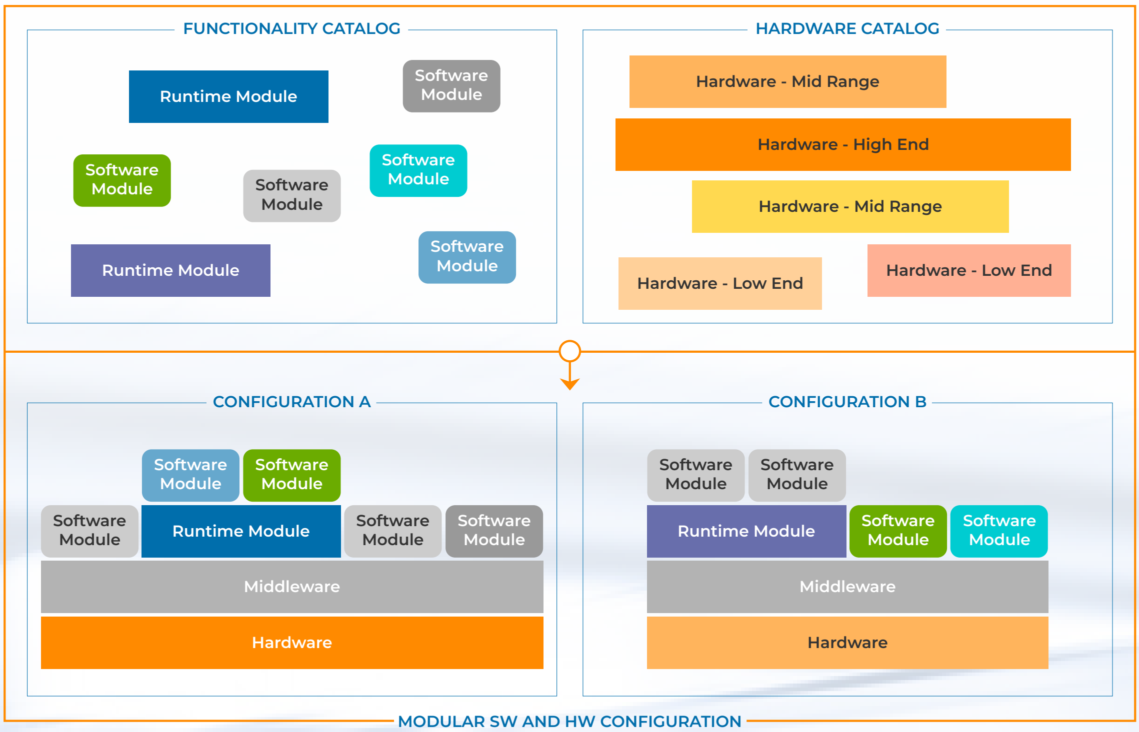
The OEM can partition and allocate resources from existing computational elements according to the functionality needed. Furthermore, dynamic in nature, the OEM may change priorities and decide to distribute such resources differently at any stage of the vehicle's lifecycle.

	Modularity	Interoperability	Upgradeability
Software	Individual applications VM images	Common protocols Abstract data exchange	Version upgrade New modules
Hardware	"Plug&Play" modules Expansion cards	Ethernet Common bus	Module addition Hardware upgrade

For example, as of today, every time an ECU has an ASIL-D functionality requirement, the supplier, in most cases, adds an ASIL-D capable SoC. With this new approach, there is an option for consolidation of the many ECU dedicated devices distributed all around the network. There will be a central high-performance ASIL-D capable server that will host all of the ASIL-D functionality, greatly reducing the number of microcontrollers onboard and simplifying hardware design.



Like the other zonal concepts, software is also modular and interchangeable. *Service-Oriented Architecture (SOA)* is not a new concept but is rapidly gaining traction as the ecosystem is being set to transition from dedicated firmware images (large and small) to abstract self contained modules, very similar to how cloud computing operates today. Reusability will maximize as software will be developed and certified once and used across multiple brands, models, network configurations, regional variants and years.



THE IMPLICATIONS ON THE AUTOMOTIVE INDUSTRY

Functionality as a service is a radical new way of thinking in automotive, one which will shake the foundations of the supply chain and vendors because, from a business operations perspective, this is akin to creating a whole new industry. Mobility as a service and shifting consumer preferences changes the nature and almost the very definition of a car. Up until recently, the engine output and exhaust noise were packaged as an exciting driving experience to entice consumers into purchasing. Meanwhile, the younger generation is more concerned with whether they can link their phone to the head unit and continue listening to their favorite music from where they left off on their personal computer. Not only do they not prioritize the mechanical aspects of the vehicle, they prefer to not be driving at all and instead use that time for other activities which they see as more valuable. Scaling up functionality is key to keeping this new generation of customers happy as they transition away from being just ‘drivers’ to becoming ‘subscribers’ of services.

THE NEW ROLE OF THE TRADITIONAL TIER 1

Fewer physical hardware components means less projects which in turn means less suppliers and vendors. As scaling up through software catches on, the industry will shift, business models will change, and new innovative and potentially smaller players and suppliers facing lower barriers to entry will emerge.

OEMs are another major player in this ecosystem as they have started developing and producing their own ECUs and technologies as a foundation or unique selling point. This is already happening with the emerging industry disruptors such as Tesla and other Silicon Valley startup style OEMs.

BUSINESS AGILITY AND INTERCHANGEABLE PARTS

We are witnessing a major shift in the way business has been conducted for years in the industry, which goes hand in hand to a greater automotive paradigm shift. Gone are the days of decades-long contractual obligations to produce and supply components to specific geographical locations. With the rapid transition to software-based, abstracted functionality, there is no physical production of parts and spares and no delivery. And there is always the possibility of replacing a software module with someone else’s potentially better and cheaper solution. This can even happen years after the vehicle is already in the customer’s hands.

The remaining physical components could be in danger of being replaced by different vendors, in turn creating OEMs that are much more resilient to shortages and supply chain disruption by necessity.

On the other hand, upgrading individual components is much more affordable than replacing the entire vehicle, unlike smartphones, which most consumers can afford to replace every few years. This is a prospect OEMs will surely exploit to create recurring revenues from vehicles already sold. This will boost the value of used yet upgraded cars and further push sales into this market space. The possibility of having feature support, which may be accomplished by a simple premium software upgrade, can be sold as a service much like a cellular plan today. This subscription-based model will convert the drivers of vehicles to subscribers of services, create new revenue streams and remain intact for the entire vehicle lifecycle.

MOVING FORWARD: EVOLVE OR BE LEFT BEHIND

THE FUTURE OF MOBILITY: THE NEO-MOTOR-VEHICLE

The standard car blueprint is now over a century old. The industry is facing the transition to new propulsion systems focused on Zero Emissions (ZE) such as battery powered Electric Vehicles (EVs) and fuel cell technology for an upgraded end-user experience. The car is transitioning from being a self-propelled wonder to a *mobility solution*, one that should be personalized, customized and entertaining rather than being a burden of operating and maintenance. This *paradigm shift* could lead to the customer not owning the vehicle in material form at all. The product may no longer be just a physical object as the role of the owner is quickly transitioning into a user.

This is an evolutionary process which will be driven by the freedom to do something completely new that fulfils new needs under new constraints and conditions. Not long ago, electricity was a World's Fair curiosity and phones were merely a means of conveying voice across long distances.

We can always turn to the past in order to get a glimpse of the future. History is rife with small, seemingly insignificant companies who supplanted the giants of the day by spurring and dominating a transformed or completely new industry:

- Apple produced home computers
- Google was a search engine
- Amazon was an online bookstore
- nVidia produced video game graphic cards

Keeping up with the times is a major opportunity, for those who move quickly and decisively, to gain market share, increase revenue and introduce new offerings we probably can't even imagine.

THOSE WHO ADAPT - WILL THRIVE

CONCLUSION - THE EVOLUTIONARY ARMS RACE

DESIGN AND TECHNOLOGICAL CHALLENGES SUMMARY

Since the first patented car 130 years ago by Carl Benz, to the cars of today, to the vehicles of tomorrow - innovation has driven the automotive industry. The current paradigm shift the automotive industry is undergoing is functionally and fundamentally changing the way consumers interact with and drive their vehicles.

Long gone are the days of purely mechanical machines transporting goods or people from point A to point B. Propelling this paradigm shift will require a fundamental re-think of the traditional automotive supply chain as we've known it over the past century. Whereas OEMs long drove vehicle innovation, the connected cars of the 2020s are software-defined computers on wheels running on hundreds of millions of lines of code. Automakers alone cannot produce the secure, sophisticated technology required to power today's vehicles.

Dynamic consumer demands are a major driving force leading the paradigm shift as well. The current automotive supply chain is struggling to implement the required technological changes and advancements to foster the high speed communication and application hosting based computing platforms that are secure by design - and answer their consumers mounting wants and needs. Rethinking and redesigning decades old architectures, networks, and components to be able to match economic and consumer trends requires synergistic relationships and partnerships with emerging, technological driven suppliers as the automotive industry shifts from the traditional vehicle blueprint to the new era of mobility.

There are a number of technological challenges in transitioning to a Zonal E/E Architecture. Below is a summary of some considerations and challenges:

HARDWARE CONSOLIDATION

- Isolation
- Platform approach
- Minimize number of components
- Interchangeable parts

ETHERNET BACKBONE

- Switched network
- Determinism
- Topology agnostic

SERVICE-ORIENTED ARCHITECTURE (SOA)

- Certification ease
- Modular design
- Reusability

ABOUT GUARDKNOX

GuardKnox's high-performance, service-oriented, customizable, and secure-by-design products usher in the 'smartphonization' of next generation vehicles. GuardKnox's SOA Framework middleware solution and developer tool suite shorten the complexity and time to market for automotive applications and updates. The CommEngine guarantees high-throughput and ultra-low latency, enabling the move to next-gen E/E architectures.

Founded in 2016, GuardKnox is based in Israel, with subsidiary locations in Munich, Germany, and Detroit, Michigan.

Please feel free to contact us at info@guardknox.com for more information!

Let us empower you with the

FREEDOM TO EVOLVE