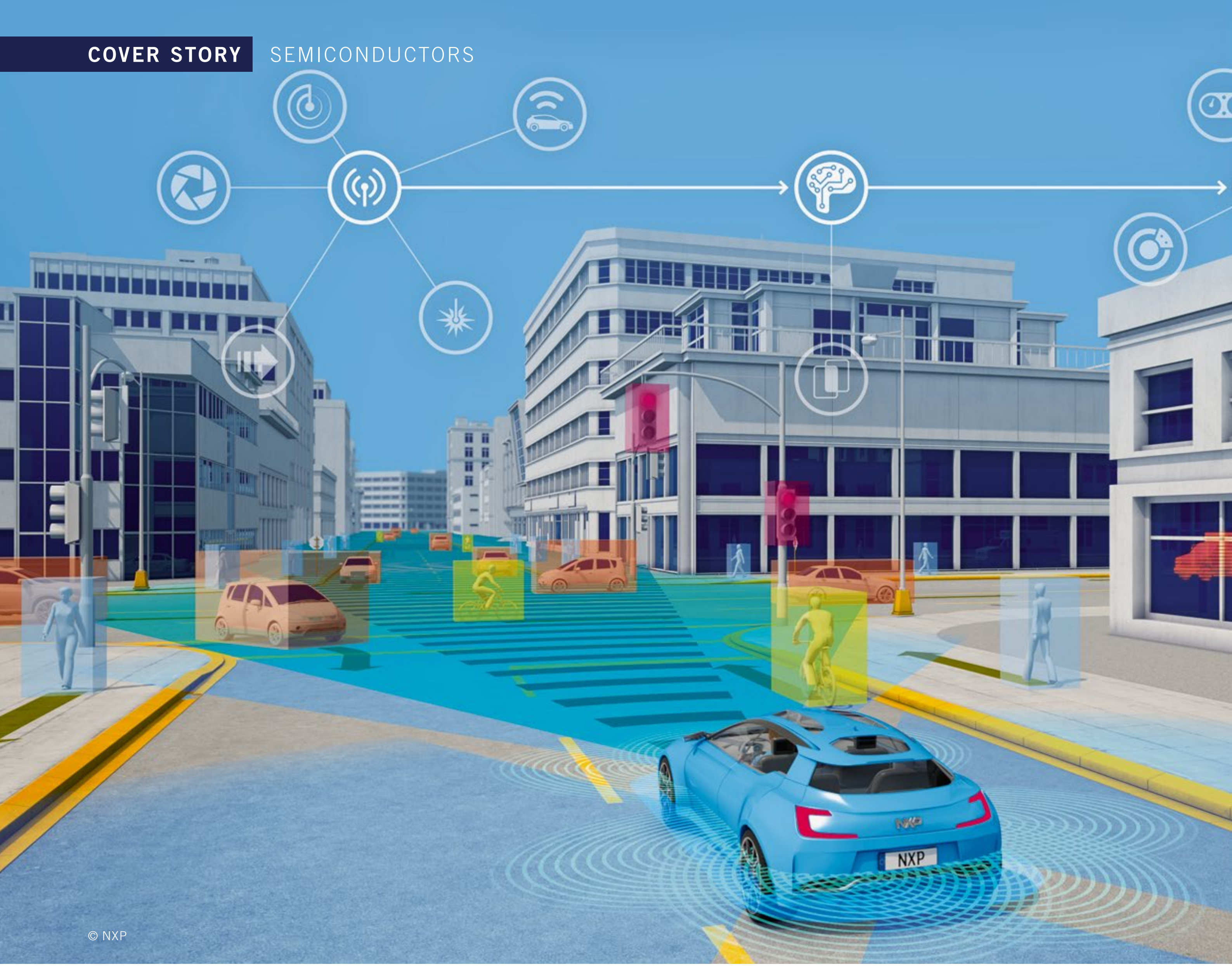

© NXP

AUTHOR

Lars Reger
is Automotive CTO at
NXP Semiconductors
in Munich (Germany).

ESTABLISHED AND NEW PLAYERS

The era of self-driving cars places semiconductor companies at the center of important discussions about standards, methodologies, and design approaches. Traditional automakers and new auto disrupters ask: "How would a semiconductor company build a self-driving car?" Long standing companies need a way to maintain business continuity and incrementally build their portfolios in areas like autonomy. Disrupters want to build from the ground up with an all new approach. Both of these vantage points seek to make sense of extremely complex and interconnected design issues that present themselves in a high pressure competitive environment.

NXP's unique position in working with traditional car makers and disrupters, as well as automakers in every part of the world, brings insights into how design considerations and global trends are changing the shape of mobility. Many have thought about how a car should be conceptualised and built and it all comes down to a domain-based architecture.

This paper presents a systems architecture approach based on domains, see **FIGURE 1** and **FIGURE 2**, and explains how this new architecture will help carmakers master the complexity of autonomous driving.

SIGNIFICANCE OF SEMICONDUCTOR COMPANIES

How a semiconductor company approaches a self-driving car does not seem intuitive until you consider how vital electronics have become to present-day automotive architectures. In fact, most of what is new and innovative in the automotive industry has to do with electronics, software and IT, **FIGURE 3**.

Today's cars are safer, more efficient, and smarter than ever before, and semiconductors are a big reason why. The vehicles rolling off of today's assembly lines are more like robots on wheels, and the very high degree of electronic sophistication is largely due to semicon-

The EE Architecture for Autonomous Driving
A Domain-based Approach

NXP Semiconductors has announced an all new control and compute concept for connected, electric and autonomous cars. The NXP S32 platform is the world's first fully-scalable automotive computing architecture. Soon to be adopted by both premium and volume automotive brands, it offers a unified architecture of microcontrollers/microprocessors and an identical software environment across application platforms.

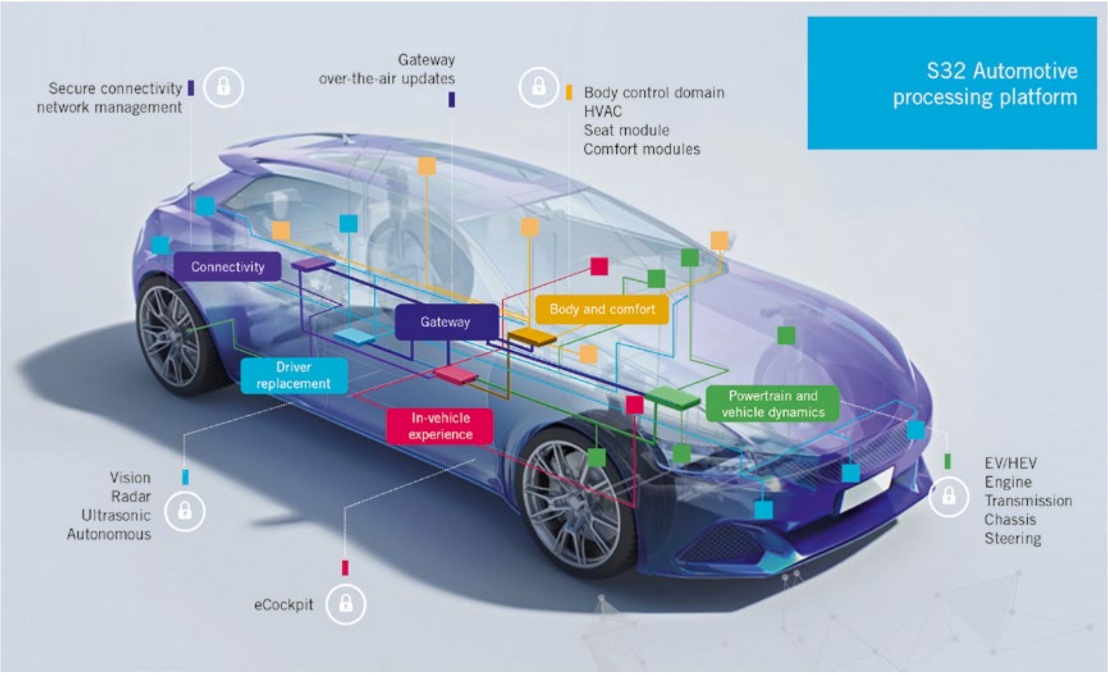

FIGURE 1 Domain-based car architecture (© NXP)

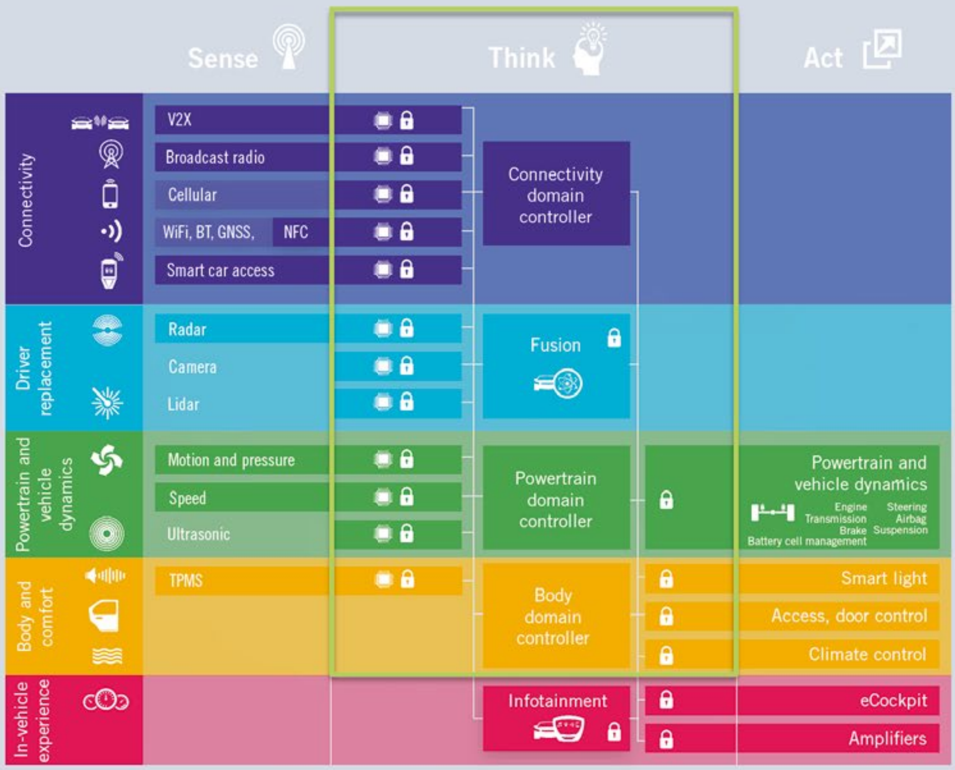

FIGURE 2 Domain-based car architecture maximises reuse of hardware and software across product and applications (© NXP)

ductor technology. As this trend toward electronification continues, semiconductor companies, **FIGURE 3**, will play an important role in vehicle design.

Semiconductor companies like NXP offer expertise that extends to processing, security, and mobile technologies that are at the core of autonomous innovation. As a result, when people ask how to build a self-driving car, NXP has a clear answer: a streamlined approach to system design through domain-based vehicle architecture.

THE DOMAIN-BASED CAR ARCHITECTURE

The domain-based architecture, **FIGURE 2**, reflects the work of NXP's automotive innovators and their collaboration with key industry stakeholders. It organises and groups together the functions that let cars sense, think, and act on behalf, and helps to manage complexity and support scalability.

The bottom three domains – in-vehicle experience, body and comfort, and powertrain and vehicle dynamics – have been part of vehicle architectures for a very long time. The top two – driver replacement and connectivity – are new and relate specifically to the functions required for autonomous operation. Taken as a whole, the domain-based architecture delivers an optimum level of autonomy while ensuring the highest degrees of safety and security. Four of the most significant advantages are:

1. Modularity: Dividing functionality into separate domains helps highlight the functional safety and cybersecurity requirements for each subsystem, simplifies the development and implementation of robotic algorithms, and makes it easier to scale features within each subsystem.
2. Easier optimisation: The domain-based architecture groups similar functions together and isolates them, so it is easier to design the right levels of safety and security based on the common requirements within each domain. In the connectivity domain, where the car communicates with the outside world, security is paramount, because external interactions need to be protected from tampering. In the powertrain and vehicle dynamics domain, security is less of an issue since the domain operates away from external factors. What is more important here is functional safety and reliability, since components need to function while exposed to extreme conditions.
3. Simpler scalability: The modular approach makes it much easier to scale, within each domain, from baseline performance to high-end operation. That means it is easier to create a range of features to meet various market requirements. The in-vehicle experience domain, for example, which includes infotainment features, might include fewer options for a budget model and many more for a luxury model. The domain-based architecture enables this kind of scalability in each section of the car, and makes it simpler to create individual building blocks that are both compatible and reusable. Development becomes more efficient and cost-effective, and manufacturing becomes more flexible and more responsive.
4. Reusability: NXP is a strong proponent of the design-reuse philosophy, **FIGURE 4**. By aiming to use the same building blocks across every automotive domain, it is easier to add or subtract functionality and evolve designs as new technologies become available. In particular, NXP's microcontroller

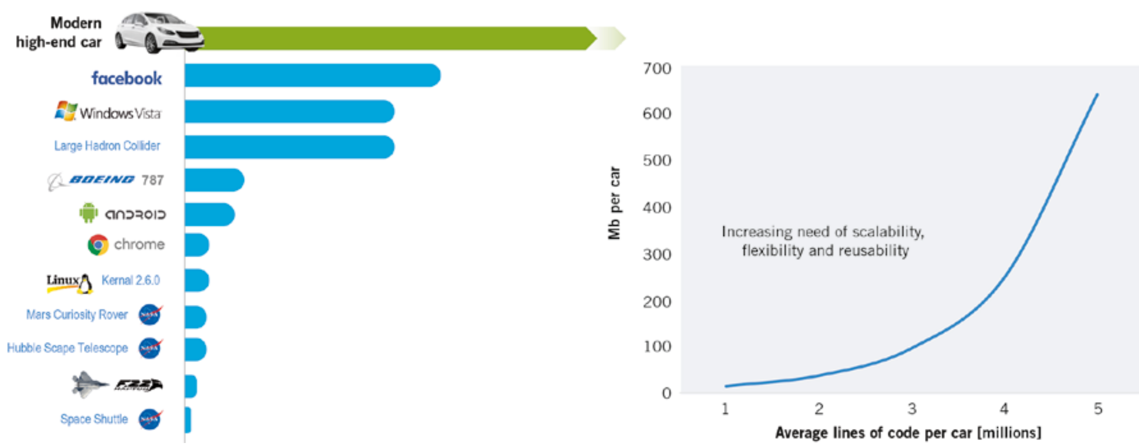

FIGURE 3 Today's vehicle contains more than 100 million lines of code, more than in branches with software-focused products (left); this amount of code lines will increase six times more in the future (right) (© NXP)

portfolio supports design reuse by employing a common architecture and a common software platform across the portfolio. Whether working on a radar system, an electronic control unit (ECU) for braking, or an automotive gateway, the one-chip microcontroller architecture means that customers can start with the same basic microcontroller hardware, and finalise their design using a familiar set of tools, IP libraries, and software code.

– Reusability: NXP is a strong proponents of the design-reuse philosophy, **FIGURE 4**. NXP aims to use the same building blocks across every automotive domain, so it is easier to add or subtract functionality and evolve designs as new technologies become available. In particular, the company's microcontroller portfolio supports design reuse by employing a common architecture and a common software platform across the portfolio. Whether engineers are working on a radar system, an electronic control unit (ECU) for braking, or an automotive gateway, NXP's one-chip microcontroller architecture means the customer can start with the same basic microcontroller hardware, and finalise his design using a familiar set of tools, IP libraries, and software code.

DOMAIN-BASED CAR ARCHITECTURE FOCUSED ON CONNECTIVITY

The connectivity domain is an overarching domain that covers a number of operations. It governs all the wireless interfaces that connect the car to the outside world. The connectivity domain seamlessly and securely deploys and collects information from all of a vehicle's externally connected interfaces. That includes familiar interfaces used by passengers and their devices, including radio, cellular, Wi-Fi, Bluetooth Low Energy (BLE), and GPS, as well as newer interfaces that relate more closely to vehicle operation, such as vehicle-to-vehicle (V-to-V) and vehicle-to-everything (V-to-X) communication. In the ideal setup, all these external interfaces are housed in a highly integrated smart antenna module that makes it easier to add or subtract interfaces as needed. Top requirements for connectivity are:

- ASIL B
- security
- reception robustness
- co-existing multi-standard transmission.

DRIVER REPLACEMENT DOMAIN

The driver replacement domain lets the automotive "robot" take over the task of driving. It provides the capabilities of sensing and thinking, and uses safeguards to ensure proper operation. The driver replacement domain is where a lot of the car's "smarts" reside, making it possible to interpret the environmental observations produced by various sensors and cameras. The "sensing" components include radar, cameras, laser-based lidar, and components for positioning and other types of environmental information. The "thinking" components include situational assessments, path planning, sensor fusion, safety-related algorithms, and more.

In today's automatic-transmission driving involves turning the steering wheel and manipulating two pedals, one for acceleration and one for braking. By just about any measure, the driver replacement domain is better at doing these things than we are. The driver substitute reacts more quickly and more consistently, without the influence of human emotions, and is always on alert. It also does not drink coffee, eat snacks, talk to passengers, answer phone calls, or otherwise get distracted while doing its job.

The driver replacement domain functions like the car's brain. And, like human brains, it can take on new knowledge gained from experience. One way to "teach" self-driving cars is to use cloud connectivity. While a self-driving car is parked in a garage overnight, for example, it can connect to the cloud and upload data accumulated during the day. That data can be aggregated with data from other cars, and can be used to optimise driving algorithms.

The "sleeping" car can download these new functions when it "wakes" in the morning, and start the new day with added capabilities. Top requirements for driver replacement are:

- ASIL D
- automotive qualification
- smart sensing
- cost, form-factor and performance trade-off.

POWERTRAIN AND VEHICLE DYNAMICS DOMAIN

Governing motion and speed, this domain is what makes the automobile move. In self-driving cars, the movement is based on inputs from the driver or the driver substitute, and can be modified and optimised based on personal preferences and environmental constraints, such as road conditions.

The powertrain has been present in cars since the earliest days of vehicle design. Whether it is part of a traditional combustion engine, an electric engine, or a hybrid of the two, the powertrain portion of this domain converts the original fuel source into power and delivers that power to the road surface. It traditionally covers the engine, transmission, drive shaft, axle, and wheels. Operating conditions in the powertrain are harsh, with high temperatures and near-constant vibration.

In automobiles, dynamics refers to the effect that forces and torques have on motion. The vehicle dynamics portion of this domain is where supporting subsystems, such as suspension and steering ensure stability and a smooth ride. This is also where many of the car's various sensor technologies can be find, including those based on sophisticated MEMS and MR technologies. Top requirements for powertrain and vehicle dynamics are:
– ASIL D
– cost, form-factor and performance trade-off
– software enablement personalisation and upgradability
– data fusion (between vehicle sensors and driver inputs).

BODY AND COMFORT DOMAIN

The body and comfort domain supports basic functions for the driver and passengers, and follows behavior to learn preferences. This is also where passive safety mechanisms (seatbelts) and access mechanisms (door locks) are typically managed.

The settings people like to have in place when they are in the car – the seat in a certain position, the mirrors just so, the climate control at the right temperature – can be tailored for automatic adjustment each time they use the car. These functions often rely on traditional auto electronics, like window controls and seat adjusters, and typically convert hardware operations into software for easier management and modification.

Sensors, microcontrollers, and new illumination techniques can work together to create smart lighting functions that increase safety and match personal preferences. With exterior lighting, headlights can automatically adjust for weather conditions or the presence of oncoming traffic. With interior lighting, programmable zones can make it easier for passengers to sleep or read or watch a video, and dashboard settings can be adjusted automatically based on time of day or who is in the car. Top requirements for body and comfort are:
– upgradable functionality
– low maintenance
– energy efficiency
– monitoring and learning abilities.

THE IN-VEHICLE EXPERIENCE DOMAIN

This is the domain that lets the car support the entertainment, productivity, and well-being of everyone onboard.

The in-vehicle experience domain can recreate the same experiences drives and passengers have in their living room. It offers seamless access to digital content, and gives the ability to create and manipulate that content. It is also a smart learning environment that adjusts to their preferences. The software used in this domain needs to be both flexible and easily upgradable, to ensure access to content via any existing hardware infrastructure. At the same time, there also needs to be an advanced, barrier-free human-machine interface (HMI), able to support voice commands, gestures, augmented reality, and advanced personalisation.

Top requirements for experience are:
– over-the-air (OTA) updates
– monitoring and learning abilities
– software upgradability/flexibility for content access
– advanced HMI.

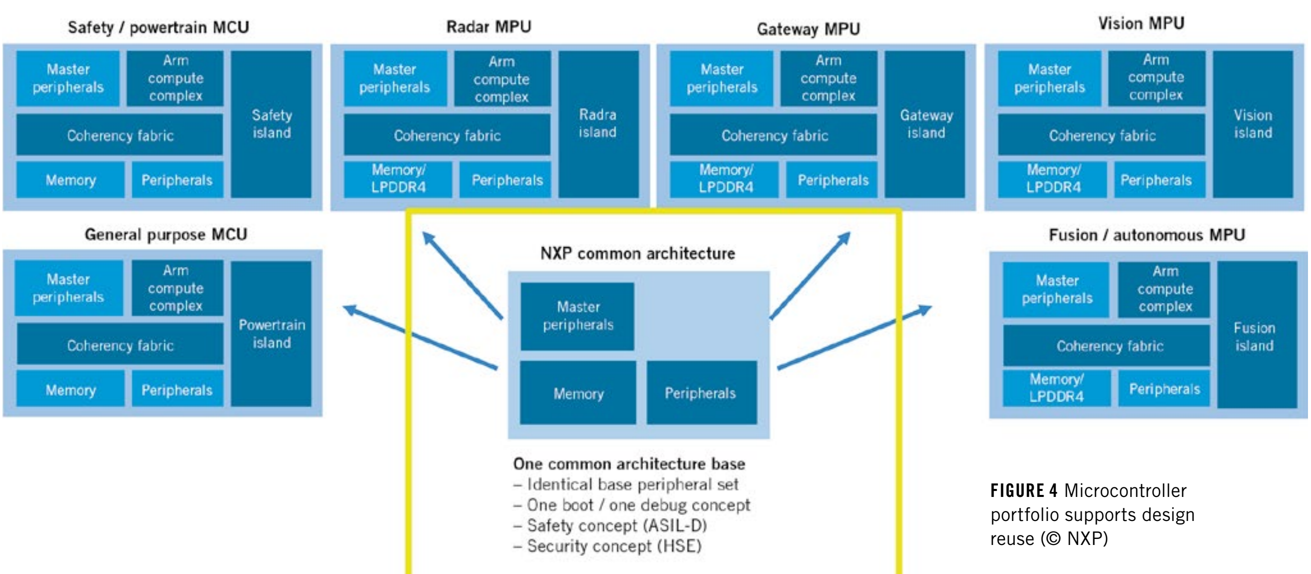

FIGURE 4 Microcontroller portfolio supports design reuse (© NXP)

GATEWAY AND IN-VEHICLE NETWORKS

The domain-based architecture is connected by a sophisticated communication network that lets the domains operate in tandem and share information. Acting as the architectural glue that holds the domains together, the internal network ensures data is shared at the right bandwidth and in a secure, reliable manner. The internal network uses many of the same technologies used in today's most advanced IT setups, including Ethernet connectivity and secure gateways.

The in-vehicle network (IVN), which includes traditional automotive technologies such as CAN, LIN, and Flexray, as well as Ethernet, securely connects the domains. The IVN lets the domains share relevant information and works with the onboard gateway to ensure proper distribution of car-generated data.

The onboard gateway keeps information inside the car, protecting it from external access and outside hacking. The gateway serves to protect (firewall) the subsystems, keeping them separated from one another to avoid unwanted interactions. That way, safety-critical systems are shielded from the actions of other systems, such as infotainment. The gateway also ensures that the high amounts of data used by each domain are routed efficiently and reliably. Top requirements for gateway and in-vehicle networks are:
- ASIL D
- security
- reception robustness
- low electromagnetic emissions
- co-existing multi-standard transmission.

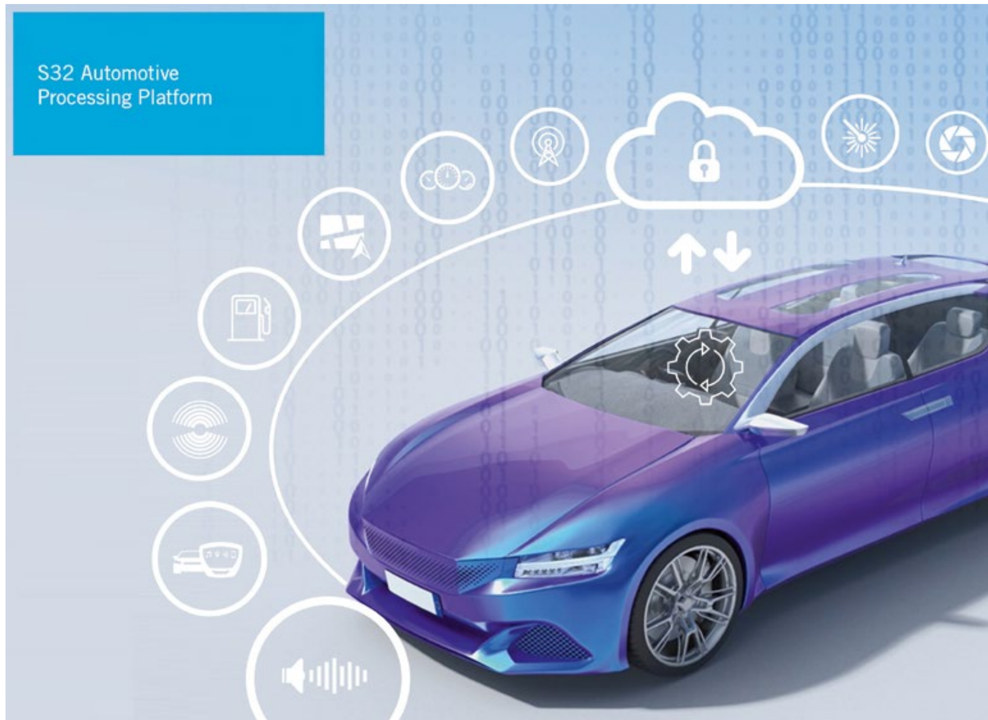

FIGURE 5 NXP uses the domains to guide the internal structur of the company (© NXP)

CONCLUSION

As part of the definition of the domain-based vehicle architecture, NXP has underscored the importance of three underlying concepts that guide decisions throughout the development process according to simplicity, reusability and scalability.

The domain-based architecture is a logical way to break down and group the hardware and software components associated with vehicle design, but it is also a way to organise the design team itself. NXP uses the domains to guide their internal structure, **FIGURE 5**. This helps focus efforts and gather expertise, and makes it easier to maximise the collaboration and technical crossovers needed to spark innovation.

In part of NXPs domain-based approach, NXP recently announced a completely new concept for control and data processing in the vehicle of the future, whether electric, connected or autonomous. The NXP S32 platform is the world's first fully-scalable automotive computing architecture and will soon be used in premium and high-volume vehicles alike. The unified architecture of microcontrollers (MCUs) and microprocessors (MPUs) possesses identical software environments across different application platforms.