WHITEPAPER

The Overarching Challenge for Autonomous and Electric Vehicles: Designing for a Future of Unknowns

A DATA-CENTRIC DATABUS PROVIDES THE FLEXIBILITY TO FUTURE-PROOF ARCHITECTURE CHOICES, WHILE STREAMLINING DEVELOPMENT RESOURCES TODAY

OVERVIEW

In a relatively short period of time, electric vehicles have taken center stage as the automotive model of the future. OEMs are basing new business models on the future capabilities that electric vehicles are introducing and which will set the foundations for autonomous vehicles. Consumers and communities are interested in ride sharing and new ownership (or non-ownership) models that will be more environmentally friendly and perhaps better suited to the lifestyle of urban and suburban dwellers.

In short, the electric vehicle is now a reality. The road to creating autonomous vehicles has the potential to change virtually everything about the traditional manufacturing and ownership model for cars.

New players will affect the course of this market, but there is no way to accurately predict who these players will be and what influence they will have.

Not surprisingly, most OEMs have multiple and even competing autonomous vehicle projects underway. Some programs are on the road already. Other projects may take five or ten years, depending on factors that OEMs simply can't put in a design plan at this stage.

For these reasons and many more, OEMs are choosing the standards-based Data Distribution Service™ (DDS®) databus as the connectivity middleware for their architectures and platforms. DDS offers the comprehensive capabilities that architects and engineering managers need to build with maximum flexibility and scalability (including future-proofing). At the same time, it provides a simplified environment that keeps designs streamlined, low maintenance and cost effective.

THE MYRIAD, MATRIXED CHALLENGES OF DESIGNING AUTONOMOUS VEHICLES

The autonomous vehicles market is growing at a tremendous rate, with a growing CAGR of over 20% during the period 2022-2027.

Typically, autonomous industrial applications are designed to operate in relatively predictable, controlled environments. By contrast, the automotive industry may well have the toughest of all autonomy design challenges: the uncontrollable, unpredictable chaos of public roads.

This white paper is for architects, engineering managers and system developers responsible for the design of electric and autonomous commercial vehicles—generally between Levels 3-5, as shown in Figure 1. However, inherent in this design focus is the consideration of IP reuse in other electric and autonomous use cases that are not yet on the drawing board.

The paper covers three major design challenges. Each challenge on its own is largely uncharted territory for OEMs. Together, they present designers with a myriad of matrixed challenges. The stakes for success are incredibly high, from the confidence of consumers and regulators to market innovation and even survival. These design challenges are:

1. Enabling sensor fusion from numerous sources that generate massive unstructured data-flows
2. Engineering increasingly complex systems, faster and at a lower development cost
3. Ensuring scalability and future-proofing

WHEN IT COMES TO NEXT GEN E/E ARCHITECTURE, THE CAN BUS IS MAXED OUT

The traditional architecture of passenger vehicles is based on the deployment of electronic control units (ECUs) for various subsystems. The Controller Area Network (CAN) bus provides the interconnection with ECUs, wherever a communications loop is required. So does the Local Interconnect Network (LIN) bus.

Today's average vehicle has 100-200 ECUs packed into different corners of the car. For the next generation of cars, that number has to come down dramatically. That means finding ways to start consolidating some of that processing power.

To achieve Level 2 autonomy, OEMs are pushing the boundaries of these buses. However, adding ECUs adds to the weight and cost of the car, reducing fuel efficiency. And because there's no way to share applications across ECUs, some features and functions must be duplicated.

In order to eliminate or consolidate ECUs, OEMs are looking at Next-Generation E/E Architectures based on Zonal controllers—a step toward creating a more flexible and efficient processing system. This is where the sensor fusion and autonomous vehicle functions will operate, adding new functionality that didn't exist before.

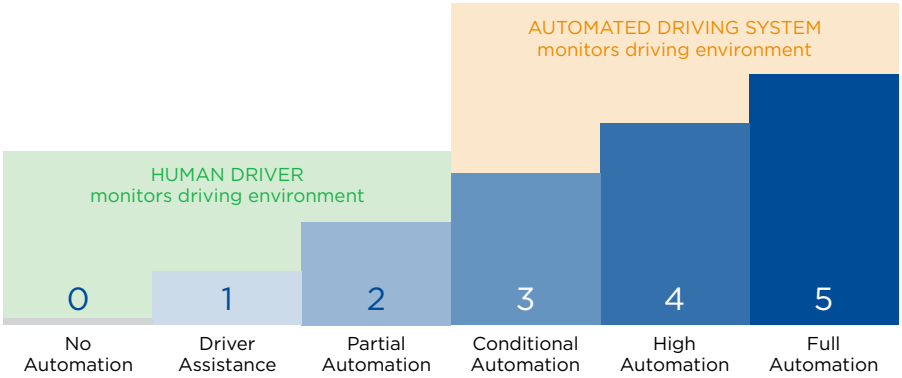


Figure 1: Accelerating progress towards Levels 3-5 autonomy, while meeting critical requirements (such as safety, security and latency) requires a standards-based. data-centric framework. This framework greatly reduces the constraints that are inherent with internally-developed, proprietary middleware.

Real-time connections to the cloud will become standard. It will be necessary to support high definition location information—well beyond consumer services Google Maps—that can locate the automobile within a centimeter of its position. High-definition maps updated in real-time, LiDAR sensor data accurate to the centimeter, and V2X data coming from other cars, people (cellphones) and road-side infrastructure are good examples of the dynamic and bandwidth-intensive data that a new system must handle.

OEMs are trying to plan architectures for the next 5-10 years that take all of this and more into account. This requires flexibility for where applications get deployed. The networking stack on top of CAN and LIN buses is very rigid and constrained in its nature, limiting its ability to provide the foundation for many of the capabilities required for autonomy. In other words, these buses are maxed out when it comes to autonomous vehicles.

In order to future-proof their architectures, designers prefer to avoid building a code base that's designed for a specific ECU. Instead, code generation should be able to run on the ECU, the domain controller, the central processing system or via the cloud. There are very few technologies that can support this level of flexibility.

The standards-based, data-centric DDS databus

Addressing all of these considerations requires a standards-based framework designed for exceptionally complex industrial systems. The DDS standard, developed by the Object Management Group® (OMG®), defines an open integration data-centric framework for software applications. Other industries with similar safety, performance and security demands (including aerospace and defense, mass transportation and healthcare), have also adopted DDS for mission-critical production systems.

DDS is transport-agnostic, provides a rich set of Quality of Service (QoS) policies and can handle any volume of data from virtually any number of sources. As shown in Figure 2, the DDS databus enables interoperability between applications, because it:

- Uses different DDS implementations
- Is written in different programming languages

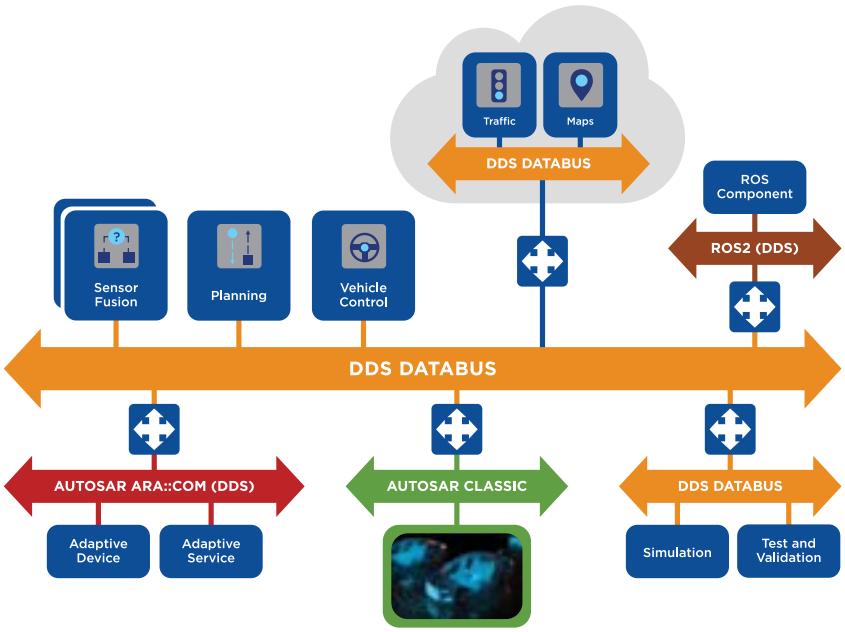


Figure 2: The DDS databus connects primary autonomy functions, such as sensor fusion and path planning. Databuses can be layered for connected car applications, such as fleet management, telematics and teleoperations. The databus shares a common data model across other systems and standards (including AUTOSAR Adaptive and ROS 2). There are no protocol translations to maintain.

- Runs on heterogeneous platforms (e.g., different hardware architecture and operating system)
- Is interconnected by different underlying communication technologies (e.g., IP-based networks, shared memory or wireless links)

A data-centric connectivity architecture is based on a software databus. The databus is designed for exceptionally complex systems, much like those used to build autonomous vehicles. The databus operates as a shared application space where all of the data lives. It is also what allows applications and devices to work together as one integrated system.

The databus distributes data in motion, unlike the database, which manages data at rest. The data serves as the interface between devices, but isn't necessarily stored anywhere. Messages do not need to be sent through brokers to access or process that data. This is one of the key attributes that enables real-time response (or ultra-low latency) with a high-volume of data from multiple sources.

ENABLING SENSOR FUSION FROM NUMEROUS SOURCES THAT GENERATE MASSIVE UNSTRUCTURED DATA-FLOWS

Data volume increases with every level of autonomy. These massive data-flows are generated by numerous sources both internal and external to the vehicle, as illustrated in Table 1.

Handling a greater volume of data means adding more processing power and memory. This, in turn, affects cost. Even the availability of 5G won't reduce the cost of connectivity, given the rate at which data volumes are escalating.

Massive data-flows also introduce more challenges with latency (the ability to process this data in near real time), high throughput and absolute safety compliance. Large data handling is a fairly straightforward problem if that's all the system requires. But sensor fusion is needed to combine multiple data sources in order to make sense of the environment and take appropriate action: It's not one application, but many layers of applications.

This complexity creates an unstructured environment that must somehow be made predictable without being inflexible.

The DDS databus and unstructured environments

The DDS databus greatly simplifies handling this unstructured environment. In fact, it is designed to handle massive data-flows of varying characteristics, including:

- Volume
- Frequency
- Latency
- Reliability
- Destination

Without a data-centric framework, designers need two or three different protocols to handle all of this variability. For example, they would require a protocol for streaming and a different protocol for control functions.

DDS is a single standard and application programming interface (API) that can handle all of these data types. All data is available everywhere and appears to an application as local memory. Architects can designate rules on how data is shared and managed on the databus, including requirements like how latency and bandwidth are handled. The connection between readers and writers (a 'contract' between publishers and subscribers) determines what to transmit, to what destination and with what priority (QoS), and how to handle collisions or failures on the network (QoS).

Following are some examples of how architects might use the DDS databus to apply QoS to different data types:

Example 1: Camera data

Camera data is sent at a certain rate. Image frames have to be processed quickly, because image refreshes are continuous and need to be interpreted for action. If part of an image is lost because of a network disruption, however, there's no point in retransmitting the image. Retransmitting takes a lot of bandwidth and adds latency, and the new image is almost identical to the previous one. With QoS, image frames can be set as "best effort" priority, which eliminates wasteful re-transmission of data.

Source	Volume (approximate)
8 cameras	2.5-20 Gbit/s
4 LiDAR sensors	30-300 Mbit/s
5 Radar sensors	~10 kB/s
16 Ultrasonic sensors	~10 kB/s
1 GPS	~10 kB/s
Control commands	~10 kB/s
Status/error handling	Whatever is needed

Table 1: Autonomous systems require sensor fusion for massive amounts of data

Example 2: Objects

With objects like LiDAR data, it's critical to know that the system has captured every object, even if it is slightly out of the video frame. Every object in the car or in the field of view is important. Using QoS to give the object guaranteed priority ensures that any data that is dropped or corrupted will be re-transmitted.

Example 3: Command and control

Command and control data requires absolute reliability, including acknowledgements that the command has been received and assurance that the data has been processed in the right order. The first level of reliability is QoS, to ensure that command and control data and acknowledgement have top transmission priority. The databus ensures reliability and robustness for redundant paths, data sources, and receivers.

The DDS databus also includes mechanisms for automatic discovery of data and sources throughout all vehicle communications. Basically, new data sources can be plugged into the databus without adding new layers of programming or configuration, including simulated modules for test environments.

It's also important to be able to change QoS priorities based on field conditions. For example, at night, the car and driver are going to rely more on LiDAR or radar data. The designer can change the priority of this data automatically with changing light or other exterior conditions.

CREATING INCREASINGLY COMPLEX SYSTEMS FASTER AND WITH LOWER DEVELOPMENT COSTS

Most OEMs don't want to be in the middleware business, especially in the era of autonomy. In-house software development requires expensive internal resources. It imposes an enormous overhead, not only for the talent, but also in the need to coordinate many design and development teams around the globe. And it creates a software maintenance burden that will inevitably constrain and slow innovation.

The DDS databus is not just a framework for sharing large volumes of data. It also provides the ability to identify, prioritize and assign values to specific data streams. It does this in an extremely efficient manner, reducing both complexity and cost.

One example is security—a critical requirement for autonomous vehicles, which also has the greatest potential for adding unsustainable complexity that could negatively affect cost and performance.

If there's a security breach in the IT world, the response is immediate: automated containment. But for autonomous vehicles, this kind of response can protect systems, but endanger lives. In the event that a security breach is detected, it is imperative to regain control of the vehicle safely and immediately.

The autonomous vehicle requires many connections to the outside world. All of these connections are vulnerable to interference and tampering. However, it is simply too expensive to secure every piece of software and every data-flow. A DDS databus provides the ability to apply tailored security policies that are specific to each application in the vehicle.

There are many dimensions to security. The DDS databus enables designers to apply security selectively, rather than integrating one-size-fits-all sweeping security policies. DDS Security™ is a standards-based security architecture that can be optionally applied without impacting application development. Security is applied after the application-level API calls, but before data reaches the network. This makes it independent of the application code, and independent of the network transport protocol.

In the DDS standard, each data piece is called a topic. Each topic can be individually tuned. Not every data-flow needs to be secured at the same level of strength. With topic-level security, the security architects can specify what level of authentication each data topic requires. What are the access control roots for this data? What's the encryption required for this data? The ability to assign security on a per-topic basis offers significant advantages for reducing cost and complexity while strengthening security.

Following are some examples:

Engine temperature data is important for maintenance, but does not require the same rigorous security as door locks, steering, braking or acceleration. Engine temperature might only require security authentication to ensure that it's coming from a reliable source. By contrast, sensitive information sent over a cellular network, such as car location, would require both authentication and encryption. Other large data streams, such as video and LiDAR, do not require encryption, but need to be authenticated.

System architects can save CPU processing power, improve latency and lower the cost of building a car by eliminating unnecessary encryption while maintaining strong security policies.

The DDS databus is also transport agnostic, so security protocols for TCP, UDP, multicast and other protocols can be applied independently of that transport.

The DDS databus and system simplicity

The DDS databus creates a greatly simplified development and runtime environment. This streamlined framework helps keep the architecture and platform free from needless complexity and added costs. Here are two examples of how the DDS framework simplifies a system, resulting in very real cost savings:

Better data sharing to reduce the instances of copying data

Traditional IT can afford a lot of data copying when there's a lot of CPU performance and memory available. That model doesn't work for autonomous vehicles. The DDS databus uses more efficient data sharing to transmit and re-use data in different languages, from different platforms across different networks, while reducing the number of copies required.

Compliance certification

In certification, cost is tied to how many lines of code there are in the system. Autonomous vehicles require the highest level of certification, which means multiple tests across many scenarios. The DDS databus can reduce the number of lines of code written in an application by as much as 60%, which can significantly reduce certification costs.

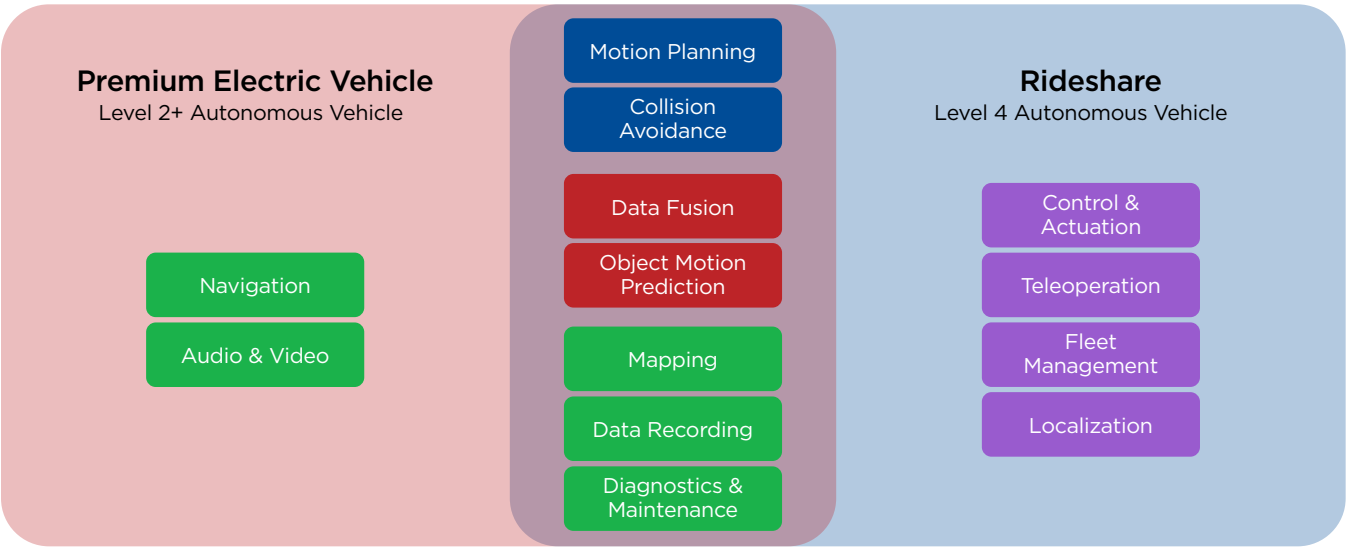


Figure 3: Different use cases will have different hardware and software designs. For example, these two vehicles don't need the same audio or navigation packages. The rideshare vehicle requires a teleoperations module, which the electric vehicle doesn't need. But they have a lot in common, too. Both will need a sensor fusion package and collision-avoidance applications. While the electric vehicle seems to be an entirely different use case for autonomy, the optimum design will reuse a lot of IP.

ENSURING SCALABILITY AND FUTURE-PROOFING

For autonomous vehicles, scalability is a multi-faceted requirement, encompassing:

- The flexibility to run new software on the ECUs, domain controller or in the cloud.
- The ability to support different devices, different systems and different networks. This requires the ability to easily plug in new sources and apply different policies (e.g., QoS and security) to the new data by topic.
- The ability to extract data across multiple, fast-moving internal and external sources.
- The ability to incorporate off-the-shelf or common components to reduce the cost of developing and maintaining custom software and hardware.
- The flexibility to swap components and upgrade the CPU or install a more powerful sensor without changing the core functionality of the component.

The DDS databus scales both horizontally and vertically. Designers don't have to pre-configure the system to try to anticipate how data might be shared someday. DDS builds that aspect of scalability into the architecture itself. Another aspect to scalability is the need to meet the design requirements of additional car models with varying requirements for autonomous features. Developing applications for single-use instances is very expensive. OEMs need the ability to design applications and functions for luxury class vehicles and adapt the same applications for economy vehicles. Beyond that, designers want to be able to reuse IP in developing autonomous vehicles for other use cases. As shown in Figure 3, the DDS databus can support the requirements for an electric vehicle operating at Level 2 autonomy, as well as a Level 4 autonomy rideshare vehicle.

Ideally, OEMs want platforms that can scale as seamlessly as possible from Level 2 and 3 up to Level 5, but also accommodate the requirement to move in increments or leaps to higher levels of autonomy.

Finally, OEMs need to plan for future scalability across teams and applications. Start with a car that can drive itself within a certain city. Then add more capabilities, so that car can drive anywhere in the country or anywhere in the world. This implies not only more software applications, but also more people and teams working on that software. It means data sharing regardless of what the transport is, whether the data is coming from backend servers or embedded systems.

The DDS databus and designing for a future of unknowns

Predicting which architectures will be dominant in the next 5-10 years is nearly impossible. Many OEMs are investing in multiple projects that all seem competitive, because they don't know which will have a chance at market success and which will be shelved. Having multiple platforms, hardware and software and technologies is the price of reducing risk.

A platform in development today may not be adequate in three or four years. The industry is making rapid advancements across the board, but currently there are no dominant ecosystems or architectures. Designing for a future of so many unknowns is really the overarching requirement that makes the DDS databus so valuable.

CONCLUSION

This paper focused on autonomous processing for developing a private autonomous vehicle. But even this, as challenging as it is, is a small part of the architectural challenge for autonomous vehicles. In reality, the automobile is evolving into a device that is integrated into a broader smart transportation infrastructure.

In other words, the car really isn't the complete system anymore, even when it's fully autonomous. It's connected to an increasingly intelligent transportation infrastructure that is local, regional and even global.

The autonomous vehicle will also change how OEMs do business. In the future, OEMs expect to make money from the data, software and value-add services in the vehicles. Increasingly, OEMs will move from the traditional model of selling cars (and walking away from the customer relationship) to offerings such as Mobility-as-a-Service, which customers would pay for on an ongoing basis.

In the future, there will be a different competitive landscape. OEMs will be competing with the technology industry. This future of unknowns is the greatest design challenge of all. For this reason, RTI Connext Drive®, based on the DDS standard, saves time and unburdens engineering teams that are already under tremendous pressure. It simplifies where complexity might introduce problems and, worse, project failures. It provides the flexibility to make and revise choices, avoiding dead ends and the inability to address new requirements. It is the first automotive-grade, safety-certified data-centric communications framework for next-generation vehicles.

For more information, please visit www.rti.com/drive.

DDS IN AUTOSAR ADAPTIVE AND ROS2

DDS is a recommended connectivity standard in the major platforms and standards in the automotive market, including:

AUTOSAR includes DDS as a middleware option in its Adaptive Platform, and has plans for introducing it into its Classic Platform as well.

ROS

DDS is an integral part of ROS 2, which relies on DDS for all its connectivity needs.

RTI CONNEXT DRIVE™

RTI Connext Drive software can accelerate the development of autonomous cars by providing an efficient path from prototyping to production with scalability across platforms, development teams and supply chains. Engineering managers and architects can leverage RTI's experience with safety-critical and robotic systems to simplify design and keep pace with this disruptive market. Connext Drive has the industry's most robust security solution for both in-car networks and back-end data with a standards-compliant databus that is certifiable up to ISO 26262 ASIL D.

ABOUT RTI

Real-Time Innovations (RTI) is the largest software framework company for autonomous systems. RTI Connext® is the world's leading architecture for developing intelligent distributed systems. Uniquely, Connext shares data directly, connecting AI algorithms to real-time networks of devices to build autonomous systems.

RTI is the best in the world at ensuring our customers' success in deploying production systems. With over 1,800 deployments, RTI software runs over 250 autonomous vehicle programs, controls the largest power plants in North America, coordinates combat management on U.S. Navy ships, drives a new generation of medical robotics, enables flying cars, and provides 24/7 intelligence for hospital patients and emergency medicine. RTI runs a smarter world.

RTI is the leading vendor of products compliant with the Object Management Group® (OMG) Data Distribution Service™ (DDS) standard. RTI is privately held and headquartered in Sunnyvale, California with regional headquarters in Colorado, Spain and Singapore.

Download a free 30-day trial of the latest, fully-functional Connext DDS software today: www.rti.com/downloads.

RTI, Real-Time Innovations and the phrase "Your systems. Working as one," are registered trademarks or trademarks of Real-Time Innovations, Inc. All other trademarks used in this document are the property of their respective owners. ©2022 RTI. All rights reserved. 50044 V3 0222

Your systems. Working as one.

CORPORATE HEADQUARTERS

232 E. Java Drive, Sunnyvale, CA 94089
Telephone: +1 (408) 990-7400
Fax: +1 (408) 990-7402
info@rti.com

rti.com
company/rti
rti_software
connextpodcast
rtisoftware
rti_software