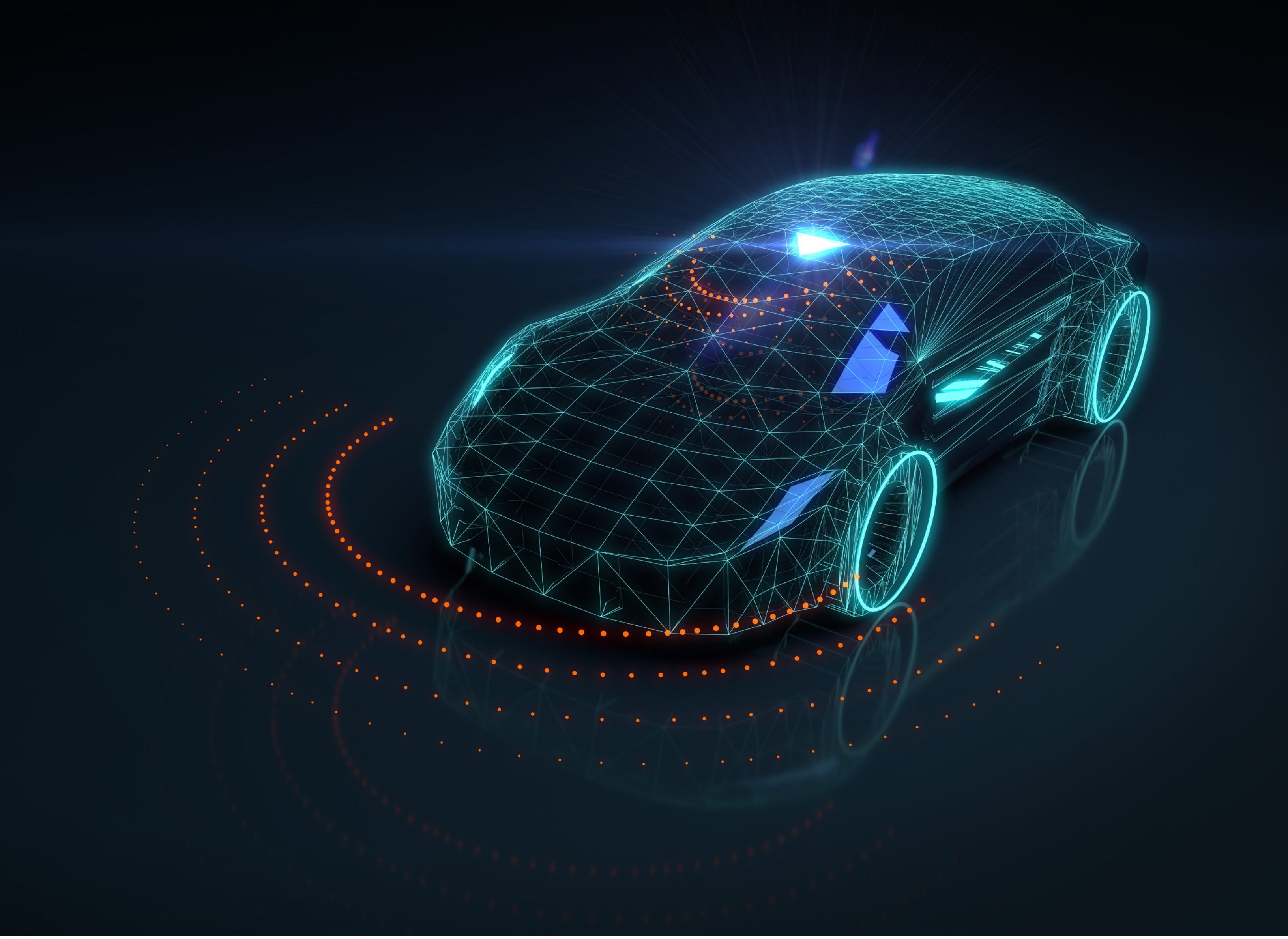


WHITE PAPER

Decoupling Hardware from Software in the Next Generation of Connected Vehicles

AUTOMOTIVE CONNECTIVITY

Contents

DEMOCRATIZING VEHICLE SOFTWARE 3

UNDERSTANDING THE TRUE DEPTH & POTENTIAL OF VEHICLE DATA 4

ONE COMPUTER, MANY MACHINES 5

TURNING THE VEHICLE INTO A CLOUD PLATFORM VIA VIRTUALIZATION 7

- Use & Manage Vehicle Functions through the Cloud 7
- Starting at the Edge, Not in the Radio 7
- Constant Connectivity Not Needed 8
- Normalize Software Development and Management for Vehicles 8

THE TRANSFORMATION BEGINS NOW—AND WILL BE ON THE ROAD IN 2022 9

VEHICLE DATA OPTIONS FOR THE CLOUD DEVELOPER 10

Democratizing Vehicle Software

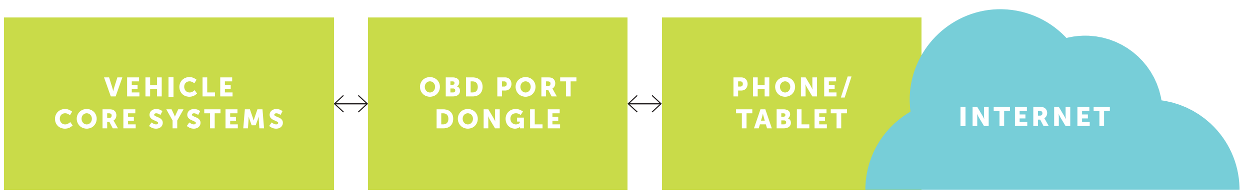
Today's vehicle connectivity solutions are similar to the very early days of internet connectivity for personal computers, via retrofitting the vehicle with add-on hardware typically connected via the diagnostic port. Other solutions are slightly more advanced, using the in-vehicle infotainment system as a connectivity hub with a smartphone display (CarPlay, Android Auto). This approach still does not fully tap into the capabilities of the vehicle as a local computing platform or effectively utilize the data the vehicle produces. We need to go beyond simple apps and dive much deeper into the vehicle itself to revolutionize connected vehicles.

Over the next few years, there will be a fundamental shift in how vehicles interact with the internet. With this transformation, vehicles will join the Cloud as new software platforms for all developers. In order to achieve this, major changes to the basic architecture of onboard computing devices in the car will occur. Fortunately, this will not require new and unproven technologies. By starting with mature hypervisor technologies used in mainframes and data centers and applying the same technology concepts and architectures to vehicle systems, it will be possible to concurrently run safety-critical software and cloud-connected software on the same physical computer. This approach democratizes vehicles for software developers, moving vehicles from a platform deeply entwined with vehicle-specific hardware and available only to embedded and specialist software engineers to a platform as open and flexible as any other cloud computing environment.

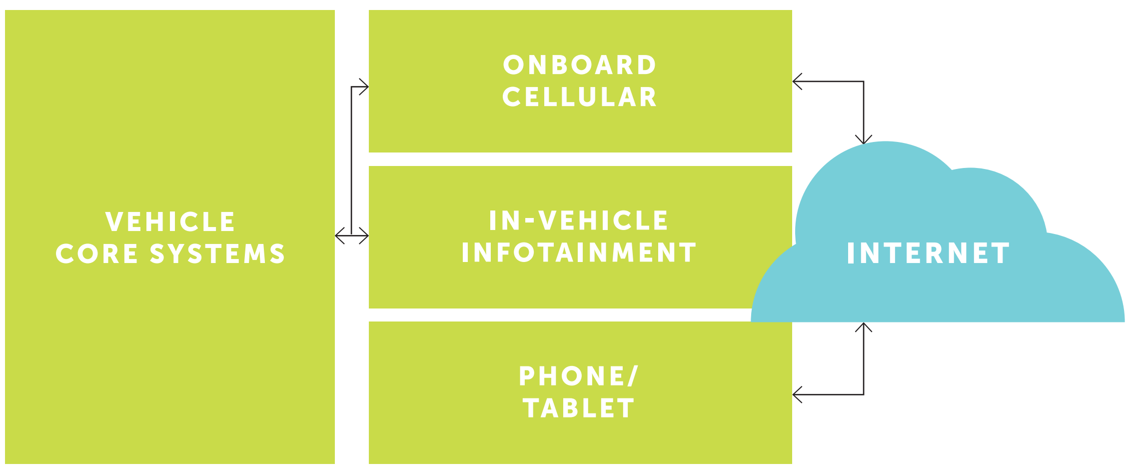
Understanding the True Depth & Potential of Vehicle Data

All vehicles produce astonishing amounts of data. According to a McKinsey, a car has the "computing power of 20 personal computers, features about 100 million lines of programming code, and processes up to 25 gigabytes of data an hour."[1] The data is used by onboard systems to operate the vehicle itself, to interact with drivers and passengers and, most importantly, to help safety systems operate.

The process of getting some of this data to the internet is similar to when an external modem was required to connect a computer to the internet – you hang an external device off the vehicle and call it "connected." There are many devices in the market that plug into the onboard diagnostics port which, in turn, is connected via Bluetooth or a cable to a smartphone which is the gateway to the outside world:



In more advanced connected vehicles, the radio, or more accurately "in-vehicle infotainment" (IVI) system, has its own connection, which may give some basic access to vehicle core systems:



Both of these approaches are simply "dumb pipe" network connections and at best, rudimentary extra screens for the smart phone that don't really take advantage of the vehicle's entire "data vocabulary." Additionally, they don't offer much (if anything) in the way of integrating modern cloud-based software solutions and business platforms into the vehicle itself. As a matter of fact, without a smartphone, many vehicles aren't actually connected at all.

There is a different approach – one that moves past dongles and radios and elevates the vehicle itself to a participating system in the connection economy.

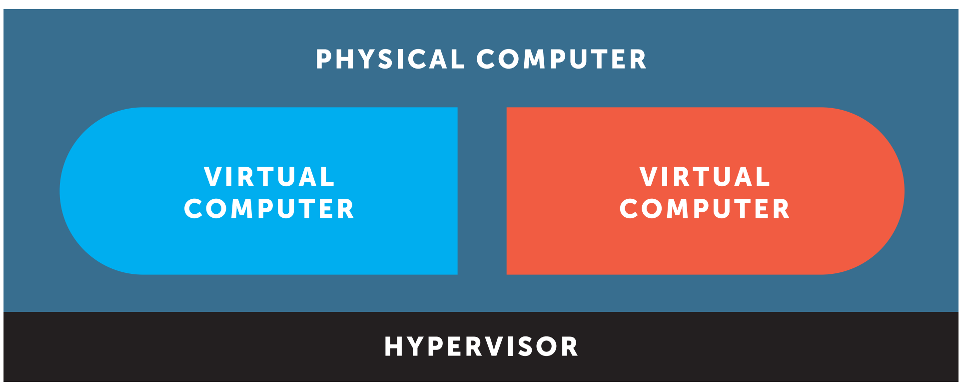
[1] https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/whats-driving-the-connected-car

One Computer, Many Machines

For decades, it has been possible to take one physical computer and make it behave as multiple "virtual machines."

As far as software is concerned, a virtual machine is real – the operating system and software does not normally "know" that it is not on its own dedicated computer hardware. A familiar example of this approach is making a Mac computer run Windows and OS X concurrently. Another example is virtually every Cloud service out there. Every physical server in every rack in a Cloud services data center is running multiple copies of various operating systems at once, and running multiple applications per operating system.

Turning one physical computer into multiple virtual machines can be accomplished through special computer code that is called a hypervisor. The hypervisor takes care of many things related to negotiating and prioritizing access to the capabilities of the physical machine by the virtual machines, and it also ensures that each virtual machine is only capable of doing what it is specifically allowed to do. Since hypervisors are a mature technology – for example, the Xen Hypervisor has run on IBM mainframe computers since 2003 – methods of isolation and control of virtual machines are well-proven.



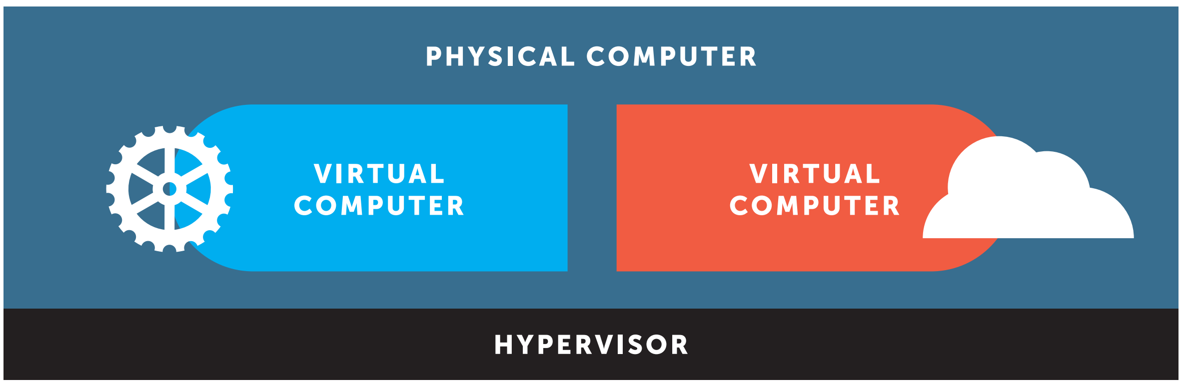
With the ability to take one physical computer and divide it into multiple virtual machines, we can do some interesting things.[2]

Let's take one physical computer in a vehicle and use a hypervisor to transform it into two virtual machines. For the sake of simplicity in this paper, we'll call them "Blue Computer" and "Red Computer" – but there can be many more.

In our example, Blue Computer is the most important computer, as it possesses the necessary software running to directly connect to and control various safety critical vehicle subsystems. The hypervisor also assigns Blue Computer highest priority for use of the physical computer and the things (like a display) that are connected to it.

[2] epa.ms/aos

Red Computer is set up differently – it is intended to be connected to the internet. Unlike Blue Computer, Red Computer lacks the software needed to talk directly to vehicle hardware subsystems, so it can only communicate with the vehicle through a special and highly limited piece of software (an API) created by the vehicle maker. That API is tightly controlled, and even then, the hypervisor is running "outside" the Blue and Red Computers making sure these machines are running only as intended. In the case of the Red Computer, this precludes direct access to any hardware in the vehicle.



Now the in-vehicle computer is, functionally speaking, two distinct computers that have the ability to access and use vehicle data in permitted ways, but only Blue Computer is able to communicate directly with vehicle systems.

Turning the Vehicle into a Cloud Platform via Virtualization

Since the Red (virtual) Computer is connected to the internet, just as a smartphone or laptop or a smart TV would be, there are several ways the vehicle opens up as a new platform for developers:

USE & MANAGE VEHICLE FUNCTIONS THROUGH THE CLOUD

With hundreds of onboard sensors, a vehicle collects and creates lots of data. Let's call some of these data items "facts" (Is the vehicle on? Is the gas tank full? How fast is the vehicle going? How many seats are occupied by adults and how many by children? Are the child locks enabled? Is it raining? Is the sunroof open? Is the "check engine" light on?). There are tens of thousands of data facts constantly flowing through the vehicle.

Additionally, there are instructions that can be sent by various systems to other systems in the vehicle. Some are deep in the engine and occur without awareness of the driver or passengers (change the fuel/air mix because the barometric pressure is low), some are visible to the driver (turn on the wipers because it is raining) and some are triggered by safety systems (turn on the hazard lights if there has been a crash).

Taken together, the collection, creation and utilization of data facts and commands in a vehicle can be called "functions." When the vehicle becomes a cloud-connected platform, vehicle functions can be transformed into something familiar to the cloud developer – an API – a simple "call and response" approach for one program to talk to another. Since the vehicle is now a connected cloud platform, we think of it from a software developer's perspective first.

RUNNING AT THE EDGE, NOT IN THE RADIO

When thinking about accessing and using vehicle functions via software, we don't need to worry (initially) about a user (or driver) facing applications. These come later, and are packaged, branded and operated according to the business strategy of the OEM.

Before apps, before interfaces, before code, we must think about how the vehicle interacts with other software and platforms. The point is to open the vehicle to the connection economy, which means providing generalized access to the vehicle via software. This, in turn, enables new business concepts and capabilities that may include many types of user experiences.

This is why the basic thinking and architecture of a fully connected vehicle needs to be structured around on-vehicle software services, rather than just driver-facing "apps." Services are computer programs that run invisibly under applications and make them work better. There are many cloud services behind virtually every business with any kind of online presence. Cloud services improve video content delivery, store data, recognize media content, pre-process data, handle eCommerce payments, ensure compliance with policies and laws, track advertising, deliver messages and much more.

RUNNING AT THE EDGE, NOT IN THE RADIO (CONT'D)

With on-vehicle software services, we can run complex software on the vehicle itself with no additional hardware or user interface. For example, both a Usage Based Insurance (UBI) service that tracks driving behavior and a Predictive Maintenance monitor can run purely as a service on the local computing hardware. These services can start and run silently in the vehicle, provide useful data to other cloud systems, and execute in-vehicle instructions and logic. With this approach, the vehicle effectively becomes a software platform for application developers, significantly simplifying application development and platform integration.

If the developer needs to create user interfaces, the services can feed and exchange data with a phone or dashboard, a web application or another vehicle. A single onboard service can concurrently feed other systems, such as an existing Fleet Management Platform or a Smart City Parking Management System via API – or any other kind of software and platform.

CONSTANT CONNECTIVITY NOT NEEDED

In the same way a smartphone still works in airplane mode, our Red Computer still works without connectivity since it is functionally its own local computer with some storage and the ability to run programs. This "edge computing" model means the examples of UBI and Predictive Maintenance services will continue to run, potentially with some limitations, without connectivity and can take specific actions once connectivity is restored.

NORMALIZE SOFTWARE DEVELOPMENT AND MANAGEMENT FOR VEHICLES

Because Red Computer is a cloud-connected virtual machine, there is considerable flexibility in how it can be configured. We can run operating systems like Automotive Grade Linux, Android and more. This also means that, rather than requiring complex embedded software development languages and processes, we can enable programming languages familiar to cloud application developers (like Python or Node.js) for the on-vehicle computing systems. Finally, as a cloud-connected system, our Red Computer also permits updates to software at any time and as many times as needed – meaning Agile development processes are practical, even for automotive solutions.

The Transformation Begins Now—and Will Be on the Road in 2022

The design-engineer-build process for vehicles is very long compared to the software industry. Vehicles in design now (mid 2018) will not hit the road until around 2022. However, the design phase of new vehicle systems includes new ways of architecting onboard systems – including connected vehicle services.

Given the lengthy process, proof of concept and testing must get underway this year, and the time to plan and design new connected services is right now.

Appendix: Vehicle Data Options for the Cloud Developer

The span and depth of vehicle data available is vast. We have created an extremely simplified subset of a typical data realm as might be found on any late-model (2015 and later) vehicle, with the intention of stimulating ideas for new services that may be created and deployed via a virtualized in-cloud/on-vehicle computing platform (not the OBD port). Not all data will be available on all vehicles. This is only a reference example, not a specification.

VEHICLE FACTS

- Vehicle Identification Number
- World Manufacturer Identifier
- Vehicle Type
- Brand
- Model
- Year
- Vehicle Size
- Number of Doors
- Fuel
- Transmission
- Wheels
- Steering Wheel
- Odometer
- Dashboard System Language
- Dashboard Units of Measurement

VEHICLE STATUS

- Driving Status
- Drive Mode
- Engine Running
- Accessory Mode
- Cranking
- Vehicle Speed
- Engine Speed (RPM)
- Wheel Speed
- Transmission Current Gear
- Accelerator Position
- Acceleration Rate
- Gyro Position
- Cruise Control Status
- Lights (Interior Lights, Dashboard Lights, Headlights, Turn Signals, Parking Lights)
- Steering
- Mirror Positions
- Seats
- Windows
- Brake
- Horn
- Interior Temp
- Exterior Temp
- Exterior Air Pressure
- All AC Settings
- Night Mode
- Distance Traveled since start
- Estimated Fuel Range
- Average Fuel Consumption
- Fuel Consumed since last start

Appendix: Vehicle Data Options for the Cloud Developer

VEHICLE MAINTENANCE & ALERTING

- General Alert (Check Engine)
- Warning Lights (oil, temp, brake, battery, low air)
- Engine Oil
- Engine Oil Wear
- Engine Coolant
- Transmission Oil Wear
- Washer Fluid Level
- Brake Pad Wear
- Battery

SAFETY SYSTEMS

- Lane Departure Sensors
- Antilock Brakes
- Traction Control
- Stability Control
- Top Speed Limit
- Airbag Status
- Child Safety Lock
- Doors
- Rain Intensity Sensor
- Wipers
- Mirror Defrost
- Convertible Roof Status
- Parking Brake

ABOUT EPAM SYSTEMS

Since 1993, EPAM Systems, Inc. (NYSE: EPAM), has leveraged its core engineering expertise to become a leading global product development and digital platform engineering services company. Through its 'Engineering DNA' and innovative strategy, consulting, and design capabilities, EPAM works in collaboration with its customers to deliver innovative solutions that turn complex business challenges into real business opportunities. EPAM's global teams serve customers in over 25 countries across North America, Europe, Asia and Australia. EPAM is a recognized market leader among independent research agencies and was ranked #8 in FORBES 25 Fastest Growing Public Tech Companies, as a top information technology services company on FORTUNE'S 100 Fastest Growing Companies, and as a top UK Digital Design & Build Agency. Learn more at www.epam.com and follow us on Twitter @EPAMSYSTEMS and LinkedIn.

CONTACT

automotive@epam.com

GLOBAL

41 University Drive,
Suite 202
Newtown, PA 18940, USA

P: +1-267-759-9000

F: +1-267-759-8989

EUROPE

Corvin Offices I. Futó street
47-53
Budapest, H-1082, Hungary

P: +36-1-327-7400

F: +36-1-577-2384

CIS

9th Radialnaya Street,
Building 2
Moscow, 115404, Russia

P: +7-495-730-6360

F: +7-495-730-6361